



Universidad
Rey Juan Carlos

ENSEÑANZA, EVALUACIÓN Y ANÁLISIS DE HABILIDADES DE PENSAMIENTO COMPUTACIONAL EN ETAPAS TEMPRANAS

**Tesis
doctoral**

Editorial Academia Abierta

Tesis doctoral

Enseñanza, evaluación
y análisis de habilidades
de pensamiento
computacional
en etapas tempranas

Autora:

María Zapata Cáceres

Directora:

Estefanía Martín Barroso

**Programa de Doctorado en
Tecnologías de la Información y las Comunicaciones.
Escuela Internacional de Doctorado
2022**

Agradecimientos

Me gustaría agradecer en primer lugar a mi directora de tesis, Estefanía Martín, su ayuda, amistad y apoyo incondicional, sin los cuales no habría sido posible de ninguna manera realizar este trabajo. Gracias por prestarme tus grandes conocimientos y por estar ahí en momentos de desánimo, dudo que pueda existir una directora mejor.

Quiero mostrar mi agradecimiento también a las personas con las que he establecido colaboraciones a lo largo de estos años y que se han convertido en amigos. Gracias a Marcos Román, por confiar en mí sin conocerme de nada y apoyarme con total generosidad, profesionalidad y cercanía. A Nardie Fanchamps, por convertirse en un apoyo fundamental, no solo para el trabajo, sino también para la vida. Gracias a Laila El- Hamamsy y Pedro Marcelino, por trabajar tan duro en nuestro proyecto común y ser siempre tan comprensivos.

Además, quisiera agradecer a mis compañeros y amigos de la Universidad Rey Juan Carlos por hacer que este camino pueda ser agradable y divertido, y no importe tanto llegar o no a alguna parte. Gracias también a Ángel Velázquez por su crítica sincera y sus acertados consejos que, sin duda, han fortalecido la investigación.

Gracias a mi familia y amigos de siempre por la comprensión y la espera y, sobre todo, por el tiempo robado.

Finalmente, mi agradecimiento a la red e-Madrid, a las instituciones, los estudiantes de grado, y los colegios que han colaborado, y, sobre todo, ¡muchas gracias a los niños que han participado en las actividades o que han jugado con nuestras aplicaciones!

Resumen

El Pensamiento Computacional (PC) es una destreza cognitiva que debe adquirirse desde una edad temprana para poder así adaptarse a la sociedad informatizada actual. Muchas de las habilidades comúnmente asociadas al PC pueden desarrollarse al aprender a programar como, por ejemplo, la abstracción, la descomposición de problemas, la algoritmia y la depuración de programas. El aprendizaje y evaluación de habilidades que contribuyen al desarrollo del PC en la educación reglada está cobrando una gran importancia a nivel internacional, habiendo consenso en que este aprendizaje debe comenzar en edades tempranas. Para incluir la enseñanza de la programación y el desarrollo de habilidades de PC en el currículo escolar, es necesario determinar qué debe aprenderse a cada edad y cómo. Al ser una competencia tan reciente, la definición del concepto es aún difusa, así como los marcos de referencia operativos que permitan establecer métodos de enseñanza y de evaluación eficaces. Sin embargo, no existen instrumentos validados y consensuados para el aprendizaje y la evaluación del PC y, además, es necesario establecer cuáles son las habilidades alcanzables y si hay diferencias en la adquisición de estas habilidades según la edad y género de los niños.

En esta tesis, se realiza una revisión sistemática de la literatura en cuanto al PC teórico y operacional, para extraer una propuesta de definición y descomposición del PC, así como varias hipótesis sobre los sistemas que combinan varias estrategias para el aprendizaje y evaluación del PC. Del estado del arte se observa que en los últimos años han surgido nuevos instrumentos de evaluación, los cuales no suelen orientarse a etapas tempranas. En el momento

de la realización del presente estudio, no existía, en la comunidad científica, ninguna prueba de evaluación independiente (no relacionada con un entorno de aprendizaje concreto) enfocada a alumnos de Educación Primaria.

Por este motivo, en primer lugar, se diseñó y se validó, a través de una experiencia en colegios, un test de estas características que permitiera una correcta evaluación del PC en etapas tempranas. Este test: **Beginners Computational Thinking test (BCTt)** fue validado mediante una experiencia con alumnos de Educación Primaria, con resultados muy satisfactorios en cuanto a fiabilidad y con gran acogida internacional, estableciéndose numerosos contactos e iniciando proyectos comunes para la mejora y desarrollo de nuevos instrumentos de evaluación del PC. Estas colaboraciones se orientan a un objetivo común para obtener instrumentos de evaluación que cubran el rango educativo desde etapas prescolares hasta la educación superior. Además, el BCTt se ha traducido a numerosos idiomas, y se han llevado a cabo varias experiencias en diferentes poblaciones y países (como Portugal, Bélgica o Singapur). También se han planteado líneas de actuación futuras, las cuales se encuentran en diferentes fases de desarrollo.

Dado que las primeras preguntas del BCTt resultaron demasiado fáciles para el último ciclo de Educación Primaria, se desarrolló, en una colaboración con la *Ecole Polytechnique Fédérale de Lausanne (EPFL)* en Suiza, una nueva adaptación del test para estos cursos, el **competence Computational Thinking test (cCTt)**, a través de una experiencia en centros escolares suizos. También se realizó una experiencia, en colaboración con la *EPFL* de Suiza y la ONG *Tree Tree 2* de Portugal, para establecer los límites de edad para utilizar el BCTt y el cCTt, en alumnos de 3º y 4º de Educación Primaria, que es donde existe duda en cuanto a la aplicación de uno u otro test.

Para establecer el límite inferior de edad para la administración del BCTt, se realizó una nueva experiencia en Educación Infantil, basada en la enseñanza de la programación a través de la robótica, en colaboración con la *Fontys University of Applied Science* y la *Zuyd University of Applied Science*, ambas en Países Bajos. Esta experiencia permitió validar el BCTt para niños de 4 y 5 años. De esta forma, el BCTt y el cCTt se confirman como instrumentos autónomos, independientes de cualquier entorno, y fiables para la evaluación del PC en Educación Primaria, pudiendo ser administrados como pre-test y post-test en las investigaciones que lo requieran.

También en el contexto de la tesis, y para explorar nuevas estrategias de aprendizaje y evaluación del PC, se ha desarrollado el videojuego colaborativo **Blue Ant Code (BAC)**, así como diversos módulos para el análisis y filtrado de datos de juego. Esta herramienta se basa en el aprendizaje basado en juegos para la enseñanza y evaluación automática del PC mediante analíticas de aprendizaje. Se realizó una experiencia en tres colegios de Educación

Primaria y se validó BAC utilizando el BCTt desarrollado previamente. Esta experiencia también permite validar un sistema que combina diferentes estrategias de aprendizaje y evaluación del PC. Los resultados muestran que este sistema es adecuado, especialmente en edades tempranas, cubriendo las tres dimensiones clave del PC (conceptos, prácticas y perspectivas computacionales). Los resultados de la experiencia fueron muy positivos, resultando en una mejora significativa del aprendizaje y de la retención en el tiempo de lo aprendido. BAC, además, pareció mejor adaptado a los alumnos más jóvenes y a aquellos con necesidades especiales o bajo percentil. Uno de los hallazgos más relevantes es que los alumnos de los primeros cursos de Educación Primaria fueron capaces de superar, en las habilidades de PC, a los alumnos de cursos superiores al final de la experiencia, con lo que en esta tesis se resalta la importancia de desarrollar el PC desde edades muy tempranas y se recomienda la incorporación de ciertos conceptos computacionales, como los bucles anidados o los condicionales, en el currículo ya en primer curso de Educación Primaria o incluso en el último curso de Educación Infantil.

Para poder evaluar, además de las habilidades estrictamente académicas, también aspectos relativos a la motivación intrínseca, intereses, persistencia y actitudes frente a las recompensas para la enseñanza de la programación y el PC, se realizó una versión de BAC, con nuevas funcionalidades, para el despliegue masivo en un entorno no escolar y voluntario. En esta experiencia masiva, se examinaron más de 28.000 partidas para analizar el comportamiento de usuarios que voluntariamente jugaron con BAC. A través de analíticas de aprendizaje basadas en juegos, se extrajeron también hallazgos relevantes como por ejemplo que, en un contexto no escolar de juego voluntario, los conceptos computacionales de bucles simples, anidados y condicionales pueden ser superados entre los 3 y los 6 años, lo que podría ser un aspecto a tener en cuenta para la elaboración de los programas escolares. Además, es destacable que todos estos conceptos pueden dominarse completamente a los 4 años si el niño persiste en un reto de forma voluntaria, intrínsecamente motivado.

Por otro lado, también se obtuvieron hallazgos que pueden ayudar a guiar el diseño de mejores herramientas para el aprendizaje de la programación y el desarrollo del PC. Dado que la edad y el género influyen en el rendimiento, es aconsejable adaptar las herramientas de aprendizaje en consecuencia, así como el currículo escolar. Además, los entornos de aprendizaje podrían adecuar la forma de proporcionar refuerzos y recompensas, especialmente para los niños en los retos más complejos y para las niñas a partir de los 5 años, ya que se detectaron diferencias significativas en estos comportamientos según la edad y el género de los niños.

En general, todos los trabajos llevados a cabo y presentados en la tesis, han enriquecido el estado del arte en el nuevo campo del Pensamiento

Computacional y especialmente en lo relativo a instrumentos de enseñanza-evaluación y a la determinación de rangos de edad/género y comportamientos en el aprendizaje de esta competencia.

Abstract

Computational Thinking (CT) is a cognitive skill that must be acquired from an early age in order to adapt to today's computerised society. Many of the abilities commonly associated with CT can be developed by learning to program, such as abstraction, problem decomposition, algorithms, and program debugging. The learning and assessment of skills that contribute to CT development in formal education is of increasing importance on an international scale. Moreover, there is a consensus that this learning should start at an early age. In order to include the teaching of programming and the development of CT skills in the school curricula, it is necessary to determine what should be learned at what age and how. Being such a recent competence, the definition of the concept is still unclear, as well as the operational frameworks for effective teaching and assessment. Furthermore, there are no validated and agreed instruments for the learning and assessment of CT. In addition, it is necessary to establish which skills are achievable and whether there are differences in the acquisition of these skills according to the age and gender of the children.

In this thesis, a systematic review of the literature on theoretical and operational CT is carried out in order to provide a suggested definition and decomposition of CT, as well as hypothesis about systems that combine different strategies for the learning and assessment of CT. From the state of the art, it can be found that new assessment tools have emerged in recent years, and these are not usually oriented to early stages. At the time of the present study, there was no stand-alone assessment test (not related to a specific learning environment) for primary school students in the scientific community.

For this reason, we designed and validated, through an experience in schools, a standalone test that would provide a correct assessment of CT in early stages. This test: **Beginners Computational Thinking test (BCTt)** was validated through a case study with primary school students, with very successful results in terms of reliability and with wide international acceptance, establishing several connections and starting collaboration projects for the improvement and development of new instruments for the assessment of CT. These collaborations are oriented towards a common interest in obtaining assessment instruments that cover the educational range from pre-school to higher education. In addition, the BCTt has been translated into multiple languages, and several experiences have been carried out in different populations and countries (such as Portugal, Belgium or Singapore). Future lines of action have also been proposed and are in different stages of development.

Since the first questions of the BCTt were too easy for the last stage of Primary Education, a new adaptation of the test for these grades, the **competence Computational Thinking test (cCTt)**, was developed in collaboration with the *Ecole Polytechnique Fédérale de Lausanne (EPFL)* in Switzerland, through a case study in several schools in Switzerland. Moreover, a case study was also carried out, in collaboration with the EPFL in Switzerland and the NGO *Tree Tree 2* in Portugal, to establish the age limits for using the BCTt and the cCTt, in 3rd and 4th grades of Primary Education, where there is doubt as to the application of one or the other test.

In order to establish the lower age limit for the administration of the BCTt, a new case study was carried out in Early Childhood Education, based on the teaching of programming through robotics, in collaboration with *Fontys University of Applied Science* and *Zuyd University of Applied Science*, both in the Netherlands. This experience made it possible to validate the BCTt for children aged 4 and 5. In this way, the BCTt and the cCTt are validated as autonomous instruments, independent of any environment, and reliable for the assessment of CT in Primary Education and, therefore, can be administered as a pre-test and post-test in the investigations that require it.

Within the context of the thesis, and in order to explore new CT learning and assessment strategies, the collaborative video game **Blue Ant Code (BAC)** has been developed, as well as several modules for game data analysis and filtering. This tool is based on game-based learning for the learning and assessment of CT through learning analytics. A case study was conducted in three primary schools in which BAC was validated using the previously developed BCTt. Through this study, a system that combines different CT learning and assessment strategies, was also validated. The results show that this system is suitable, especially at early ages, covering the three key dimensions of CT (computational concepts, practices, and perspectives). The results obtained were very strong, resulting in a significant improvement

in learning and retention over time. BAC also seemed to be better suited to younger students and those with special needs or low percentile. One of the most relevant findings is that students in the first years of Primary Education were able to outperform students in higher grades in CT skills at the end of the study, which highlights the importance of developing CT from a very early age, and suggests the incorporation of certain computational concepts, such as nested loops or conditionals, into the school curricula in the first year of Primary Education or even earlier.

In order to be able to assess, in addition to strictly academic skills, also aspects related to intrinsic motivation, interests, persistence and behaviours in response to rewards for learning programming and CT, a version of BAC, with new functionalities, was developed for large-scale deployment in a non-school and voluntary environment. In this massive study, more than 28,000 games were examined to analyse the behaviour of users who voluntarily played with BAC. Through game-based learning analytics, relevant findings were also drawn such as that, in a non-school context of voluntary play, the computational concepts of simple, nested, and conditional loops can be mastered between the ages of 3 and 6, which could be an aspect to take into account for the development of school curricula. Moreover, it is noteworthy that all these concepts can be fully mastered by the age of 4 if the children persist in a challenge on a voluntary, intrinsically motivated basis.

On the other hand, other findings can help to guide the design of improved learning tools for programming and CT development. Since age and gender have an impact on performance, it is advisable to adapt learning tools accordingly, as well as the school curricula. In addition, learning environments could adapt the way reinforcement and rewards are provided, especially for boys for more complex challenges and for girls from the age of 5, as significant differences in these behaviours were identified according to children's age and gender.

In overall, all the work carried out and presented in the thesis has enriched the state of the art in the new field of Computational Thinking and especially regarding teaching-assessment tools and the specification of age/gender ranges and behaviours in the learning of this competence.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	27
• 1.1. Motivación	27
• 1.2. Objetivos generales	29
• 1.3. Experiencias realizadas	32
• 1.4. Estructura del documento	32
CHAPTER 1. INTRODUCTION	35
• 1.5. Motivación	35
• 1.6. General aims	37
• 1.7. Studies conducted	40
• 1.8. Document structure	40
CAPÍTULO 2. MARCO TEÓRICO Y ESTADO DEL ARTE	43
• 2.1. Pensamiento Computacional	43
• 2.1.1. Definiciones genéricas	47
• 2.1.2. Definiciones operativas	49
• 2.1.3. Marcos de referencia educativos	57
• 2.2. PC en etapas tempranas	61
• 2.3. Teorías y estrategias para el desarrollo del PC	67
• 2.4. Entornos de aprendizaje	72
• 2.4.1. Robótica	75
• 2.4.2. Aplicaciones y juegos	76
• 2.4.3. Pensamiento computacional desenchufado	83
• 2.5. Herramientas de evaluación	85
CAPÍTULO 3. INVESTIGACIÓN EMPÍRICA: HIPÓTESIS Y EXPERIENCIAS	91
• 3.1. Propuesta de definición de PC	91
• 3.2. Hipótesis generales de la investigación empírica	92
• 3.3. Estructura de la investigación empírica	97
CHAPTER 3. EMPIRICAL RESEARCH: HYPOTHESES AND CASE STUDIES	103
• 3.4. Suggested definition of CT	103
• 3.5. General hypotheses of the empirical investigation	104
• 3.6. Empirical research structure	109
CAPÍTULO 4. DESARROLLO Y VALIDACIÓN DE UNA PRUEBA INDEPENDIENTE PARA LA EVALUACIÓN DEL PC	115
• 4.1. Introducción y objetivos específicos	115

• 4.2. Procedimiento y diseño del BCTt	117
• 4.2.1. BCTt v.1	117
• 4.2.2. Evaluación de expertos	123
• 4.2.3. BCTt v.2	128
• 4.2.4. Administración del BCTt v.2	132
• 4.3. Resultados y discusión	133
• 4.3.1. Transiciones	133
• 4.3.2. Estadística descriptiva	134
• 4.3.3. Fiabilidad	137
• 4.4. Conclusiones sobre la validación del BCTt	138
• 4.5. Limitaciones del BCTt y líneas de actuación	140
• 4.6. Competent Computational Thinking Test (cCtt)	141
• 4.6.1. Diseño y validación del cCtt	141
• 4.6.2. Comparativa entre el BCTt y el cCtt	143
• 4.7. Publicaciones y avance de objetivos	144
CAPÍTULO 5. VALIDACIÓN DEL BCTT EN ALUMNOS DE EDUCACIÓN INFANTIL	147
• 5.1. Introducción y objetivos específicos	147
• 5.2. Metodología y participantes	149
• 5.3. Materiales	149
• 5.4. Procedimiento	150
• 5.5. Resultados y análisis de datos	151
• 5.5.1. Análisis cualitativo	151
• 5.5.2. Análisis cuantitativo mediante el BCTt	153
• 5.6. Conclusiones	157
• 5.7. Limitaciones y líneas de actuación	159
CAPÍTULO 6. DESARROLLO Y VALIDACIÓN DE VIDEOJUEGO PARA LA ENSEÑANZA Y EVALUACIÓN DEL PC: BLUE ANT CODE (BAC)	163
• 6.1. Introducción y objetivos	163
• 6.1.1. Contexto general	163
• 6.1.2. Objetivos específicos	165
• 6.2. BAC como herramienta de aprendizaje	165
• 6.3. Adaptación de BAC al marco 3D	168
• 6.4. Desarrollo del entorno BAC	170
• 6.4.1. Diseño de juego	170
• 6.4.2. Implementación	177
• 6.4.3. Base de datos y módulos de análisis	180
• 6.5. Caso de estudio en Educación Primaria	182

• 6.5.1. Sistema de evaluación propuesto	182
• 6.5.2. Participantes y procedimiento	185
• 6.6. Resultados y discusión	187
• 6.6.1. BCTt	187
• 6.6.2. Entorno BAC	195
• 6.6.3. Datos cualitativos	197
• 6.7. Conclusiones	199
• 6.8. Limitaciones y líneas de actuación	204
 CAPÍTULO 7. ANÁLISIS DE COMPETENCIAS: MOTIVACIÓN INTRÍNSECA, PERSISTENCIA Y RECOMPENSAS MEDIANTE ANALÍTICAS DE APRENDIZAJE CON BAC	 207
• 7.1. Introducción y objetivos	207
• 7.1.1. Analíticas de aprendizaje en juegos	208
• 7.1.2. Motivación intrínseca en entornos GBL	209
• 7.1.3. Objetivos específicos y preguntas de investigación	211
• 7.2. Metodología	213
• 7.2.1. Adaptación del entorno BAC para su despliegue masivo	213
• 7.2.2. Variables de investigación	214
• 7.2.3. Participantes y procedimiento	216
• 7.3. Resultados y discusión	217
• 7.3.1. General	217
• 7.3.2. Conceptos computacionales	219
• 7.3.3. Progresión	223
• 7.3.4. Persistencia	226
• 7.3.5. ¿Persistencia para superar un reto o persistencia para obtener una recompensa?	228
• 7.3.6. Recompensas	232
• 7.4. Conclusiones	236
• 7.4.1. Interés en aprender a programar y resolver problemas computacionales	236
• 7.4.2. Habilidades y logros a través de los intentos en el juego	237
• 7.4.3. Comportamientos de juego: persistencia y recompensas	237
• 7.5. Limitaciones y líneas de actuación	240
 CAPÍTULO 8. CONCLUSIONES Y LÍNEAS DE ACTUACIÓN	 243
• 8.1. Beginners Computational Thinking Test (BCTt)	247
• 8.2. Evaluación del PC: líneas actuales y futuras	250
• 8.3. Blue Ant Code (BAC)	252

• 8.4. Aprendizaje del PC: líneas actuales y futuras	256
• 8.5. Publicaciones	258
• 8.5.1. Revistas JCR (Q1 y Q2)	258
• 8.5.2. Congresos internacionales	259
• 8.5.3. Congresos nacionales	259
• 8.5.4. Capítulos de libro	259
• 8.5.5. Divulgación	260
CHAPTER 8. CONCLUSIONS AND LINES OF ACTION	261
• 8.6. Beginners Computational Thinking Test (BCTt)	265
• 8.7. CT assessment: current and future directions	268
• 8.8. Blue Ant Code (BAC)	271
• 8.9. CT learning: current and future directions	274
• 8.10. Publications	276
• 8.10.1. JCR journals (Q1 y Q2)	276
• 8.10.2. International congresses	276
• 8.10.3. National congresses	277
• 8.10.4. Book chapters	277
• 8.10.5. Dissemination	277
BIBLIOGRAFÍA	279
ANEXOS	301
• A. BCTt	301
• A.1. BCTt v.1	301
• A.2. BCTt v.2	314
• A.3. Protocolo de actuación del BCTt	327
• A.4. Ejemplos	339
• B. BAC	366
• B.1. Protocolo de actuación	366
• B.2. Base de datos	340

ÍNDICE DE TABLAS

• Tabla 1. Resumen de las principales descomposiciones del PC y propuesta	55
• Tabla 2. Modelo de PC: Qué debe ser capaz de hacer el estudiante para considerar que ha desarrollado cada habilidad de PC según Atmatzidou y Demetriadis [55]	58
• Tabla 3. Dimensiones clave del marco 3D de PC de Brennan y Resnick [46]	60
• Tabla 4. Estrategias de aprendizaje más utilizadas y con más impacto en el aprendizaje del PC	70
• Tabla 5. Ejemplos de aplicaciones e instrumentos para enseñar a programar y desarrollar el PC	73
• Tabla 6. Conceptos de PC que se utilizan en ScratchJr. según Rose et al. [57]	79
• Tabla 7. Iniciativas para aprender programación y desarrollar PC a nivel internacional y nacional	80
• Tabla 8. Relación entre los elementos de juego, la RV, el PC, y el impacto en el aprendizaje [127]	82
• Tabla 9. Seguimiento de los objetivos, capítulo 2	89
• Tabla 10. Relación entre las experiencias, las hipótesis y los objetivos de la tesis	90
• Tabla 11. Seguimiento de los objetivos, capítulo 3	99
• Tabla 12. Conceptos computacionales del marco 3D abordados en cada ítem del BCTt.	121
• Tabla 13. Perfiles profesionales de los expertos	124
• Tabla 14. Resumen del cuestionario para la validación de los expertos	124
• Tabla 15. Submuestras de administración del BCTt en centros de Educación Primaria (n: número de estudiantes)	133
• Tabla 16. Estadística de submuestras y prueba t de Student para comparar las variaciones del BCTt (1: con transiciones, y 2: sin transiciones)	134
• Tabla 17. Puntuaciones en el BCTt. Estadísticas por curso	135
• Tabla 18. Estadísticas de fiabilidad de la muestra completa 1	138
• Tabla 19. Estadísticas de fiabilidad del BCTt por curso	138
• Tabla 20. Seguimiento de los objetivos, capítulo 4	145
• Tabla 21. Diferencias por concepto computacional abordado en la experiencia en Educación Infantil	154
• Tabla 22. Análisis de varianza (ANOVA) en la experiencia en Educación Infantil	155
• Tabla 23. Diferencias por edad según el concepto computacional en Educación Infantil.	156
• Tabla 24. Análisis t-test comparando entre 4 y 5 años las diferencias en la comprensión de los conceptos computacionales	157
• Tabla 25. Conceptos computacionales alcanzables para niños de 4 y 5 años según el estudio	158

• Tabla 26. Seguimiento de los objetivos, capítulo 5	160
• Tabla 27. Dimensiones clave del marco 3D y su adaptación al nivel de principiante	169
• Tabla 28. Conceptos del marco 3D implementados en BAC por nivel	172
• Tabla 29. Diagrama entidad relación de la base de datos 1	181
• Tabla 30. Resumen del contenido de los cuestionarios	184
• Tabla 31. Ciclo de Educación Primaria respecto a cada colegio y edad de los alumnos	185
• Tabla 32. Número de estudiantes (n) en cada submuestra	185
• Tabla 33. Submuestras consideradas e instrumentos utilizados en los diferentes puntos de tiempo de la investigación (por semanas)	186
• Tabla 34. Resultados de la prueba t de Student de la puntuación del BCTt como pre-test	187
• Tabla 35. Comparación entre los resultados de las puntuaciones del pre-test y del posttest entre las distintas submuestras dependiendo del modo de juego (prueba t de Student)	189
• Tabla 36. Percentiles por curso	190
• Tabla 37. Comparación entre los resultados de las puntuaciones del pre-test y del posttest entre las diferentes submuestras dependiendo de la modalidad de juego (prueba t de Student). Q1, percentil 25	191
• Tabla 38. Comparación entre los resultados de las puntuaciones del pre-test y del posttest entre las diferentes submuestras dependiendo de la modalidad de juego (prueba t de Student). Q2, percentil 50	192
• Tabla 39. Comparación entre los resultados de las puntuaciones del pre-test y el test de retención y del post-test y el test de retención entre las diferentes submuestras dependiendo de la modalidad de juego	194
• Tabla 40. Seguimiento de los objetivos, capítulo 6	205
• Tabla 41. Relación entre las trazas recogidas y las variables de investigación	215
• Tabla 42. Distribución por género, normalizado por clúster	228
• Tabla 43. Distribución de clústeres normalizada por género	230
• Tabla 44. Distribución de clústeres normalizada por género	232
• Tabla 45. Persistencia-reto y persistencia-recompensa por nivel, edad y sexo. porcentaje de jugadores que repiten el nivel	235
• Tabla 46. Seguimiento de los objetivos, capítulo 7	241
• Tabla 47. Resumen de objetivos, hipótesis y experiencias realizadas	245
• Tabla 48. Tabla de los instrumentos en línea con el objetivo común OC	248
• Tabla 49. Habilidades de PC alcanzables y/o nivel de adquisición posible en primera instancia	249

ÍNDICE DE FIGURAS

• Figura 1. Motivación de la tesis e instrumentos desarrollados	29
• Figura 2. Objetivos generales de la tesis	31
• Figura 3. Experiencias realizadas en relación con los objetivos generales	32
• Figura 4. Relación entre el Pensamiento Computacional, la Informática, y la Programación [30]	44
• Figura 5. Relación entre el Pensamiento Computacional y el Pensamiento Matemático según Sneider et al. [35]	45
• Figura 6. Relación dual entre el Pensamiento Computacional y el Pensamiento de Diseño según Kelly et al. [36]	45
• Figura 7. Ejemplos de robots educativos: Cubetto (fuente: primotoys.com), Bee-bot y Blue-bot (fuente: bee-bot.us y blue-bot.us)	76
• Figura 8. Interfaces de las herramientas Scratch (izquierda) y ScratchJr (derecha)	78
• Figura 9. Actividades de PCD: Crazy characters (izquierda) y Winter Warmers de Barefoot computing (derecha)	85
• Figura 10. Impacto esperado de la estrategia de aprendizaje en las dimensiones clave del marco 3D del PC	94
• Figura 11. Impacto esperado de la estrategia de evaluación en las dimensiones clave del marco 3D del PC	96
• Figura 12. Procedimiento de validación del BCTt	117
• Figura 13. Laberinto sin transiciones (izquierda) y con transiciones entre las casillas (derecha), queda convertido en un diagrama de estados	119
• Figura 14. Ítem de tipo lienzo con problemas en la lectura. La dirección de la línea de puntos crea confusión sobre la dirección en la que deberían leerse las posibles respuestas	119
• Figura 15. Ítem de tipo lienzo. La dirección de dibujo de la línea de puntos es la misma dirección en la que deben leerse las posibles respuestas	120
• Figura 16. BCTt v.1. Ítem número 3	120
• Figura 17. BCTt v.1. Ítem número 18	122
• Figura 18. BCTt v.1. Ítem número 21	122
• Figura 19. BCTt v.1. Ítem número 24	123
• Figura 20. Eje de ordenadas: Puntuación obtenida en el test (por ejemplo, 41 significa que 41 de 45 expertos respondieron correctamente); por concepto computacional (eje de abscisas: 1. Secuencias, 2. Bucle simple, 3. Bucle anidado, 4. If-then, 5. If-then-else, 6. While)	126
• Figura 21. Dificultad de los ítems del BCTt v.1 percibida por los expertos (eje de ordenadas: escala Likert de 1 a 5), por ítem del BCTt (eje de abscisas)	126

• Figura 22. Relevancia del concepto computacional del BCTt para medir el PC, percibida por los expertos (eje de ordenadas: escala Likert de 1 a 5) por concepto computacional (eje de abscisas: 1. Secuencias, 2. Bucle simple, 3. Bucle anidado, 4. If-then, 5. If-then. 6. While)	127
• Figura 23. BCTt v.2. Ejemplo de ítem (ítem número 3)	130
• Figura 24. BCTt v.2. Ejemplo de ítem (ítem número 18)	130
• Figura 25. BCTt v.2. Ejemplo de ítem (ítem número 21)	131
• Figura 26. BCTt v.2. Ejemplo de ítem (ítem número 24)	131
• Figura 27. Ejemplo de ítem relativo al concepto de secuencias "paso a paso"	132
• Figura 28. Puntuación en el BCTt por curso y concepto. Eje de abscisas: concepto computacional. Eje de ordenadas: Puntuación del ítem BCTt, normalizado de 0 a 5: 5 puntuación máxima	135
• Figura 29. Índice de dificultad del ítem (eje de ordenadas) para cada ítem del BCTt (eje de abscisas) 99	
• Figura 30. Índice de dificultad del ítem (eje de ordenadas) para cada ítem del BCTt (eje de abscisas), en la primera etapa educativa (primer y segundo curso de Educación Primaria)	136
• Figura 31. Histograma de puntuaciones en el BCTt para el primer ciclo de Educación Primaria	137
• Figura 32. Ejemplos de interpretación de los pictogramas del test, y pregunta número 22 del cCTt, que combina varios conceptos diferentes (versión en inglés)	142
• Figura 33. Entorno preparado para la experiencia con Cubetto	150
• Figura 34. Procedimiento de la experiencia en Educación Infantil	151
• Figura 35. Pantalla de selección de modo de juego: individual o colaborativo	171
• Figura 36. Pantalla de selección de nivel	172
• Figura 37. Ejemplos de partida aleatoria en niveles 3 (izquierda) y 5 (derecha)	173
• Figura 38. Ejemplo de partida en el nivel 3	175
• Figura 39. Pantalla de registro y creación de nuevo usuario, mediante apodo y número de identificación	176
• Figura 40. Pantalla de estado emocional en el modo colaborativo	177
• Figura 41. Esquema de arquitectura REST	178
• Figura 42. Actividades de una iteración del PU y diagrama MVC	178
• Figura 43. Algunas clases de BAC (diagrama simplificado de algunas clases, UML)	180
• Figura 44. Diagrama simplificado del módulo para el tratamiento de ficheros locales generado por BAC	181
• Figura 45. Ejemplo de gráfico de visualización de datos de BAC	182

• Figura 46. Mejora de la puntuación en el BCTt (eje de ordenadas: puntuación post-test menos puntuación pre-test) por conjunto concepto computacional. Comparación entre submuestras A (modalidad de juego individual)	193
• Figura 47. Tiempo medio empleado en segundos (eje de ordenadas) para resolver el problema en cada partida jugada: siendo el 1 la primera partida (eje de abscisas), por submuestras	196
• Figura 48. Tiempo medio empleado en segundos (eje de ordenadas) para resolver el problema por concepto computacional (eje de abscisas), en dos momentos (momento primer día del periodo de juego; final del momento 1: tras 5 semanas de periodo de juego)	197
• Figura 49. Impacto confirmado de la estrategia de aprendizaje en las dimensiones clave del marco 3D del PC	200
• Figura 50. Impacto confirmado de la estrategia de evaluación en las dimensiones clave del marco 3D del PC	201
• Figura 51. Mapa conceptual: conexión entre los conceptos tratados en esta sección.	213
• Figura 52. Distribución por edad	217
• Figura 53. Tiempo medio en minutos totales de juego por edad y sexo	218
• Figura 54. Eje izquierdo: número medio de intentos por edad y sexo. Eje de la derecha: tiempo medio en minutos dedicado a cada intento	219
• Figura 55. Rendimiento o logro: porcentaje del número de partidas ganadas sobre las jugadas	219
• Figura 56. Eje izquierdo: habilidad - número medio de intentos por edad y sexo (normalizado). Eje derecho: logro - porcentaje medio del número de partidas ganadas sobre las jugadas por edad y género (normalizado)	220
• Figura 57. Logro por nivel y por género	221
• Figura 58. Persistencia media o número de intentos (eje vertical), por nivel y edad (eje horizontal)	222
• Figura 59. Rendimiento (logro) medio (eje vertical) por nivel y edad (eje horizontal).	222
• Figura 60. Eje izquierdo: logro medio, representado gráficamente por líneas. Eje derecho: habilidad media por intento por género, en segundos, representado gráficamente por barras. Eje horizontal: intentos progresivos agrupados de 5 en 5	224
• Figura 61. Logros por intento (10 primeros intentos) y nivel, para 4 años	225
• Figura 62. Logros por intento (10 primeros intentos) y nivel, para 5 años	225
• Figura 63. Logros por intento (10 primeros intentos) y nivel, para 6 años	225
• Figura 64. Primera división en clústeres	227

• Figura 65. Distribución en cada clúster (normalizado), de los 3 a los 12 años de edad.	228
• Figura 66. Segunda división de clústeres	229
• Figura 67. Tendencia a lo largo de los intentos por clúster	230
• Figura 68. Tercera división en clústeres	231
• Figura 69. Distribución de las variables (normalizadas) en cada clúster	232
• Figura 70. Trayectorias generales del juego, donde los nodos son niveles	233
• Figura 71. Trayectorias del juego. Persistencia-recompensa (después de ganar la partida)	233
• Figura 72. Trayectorias del juego. Persistencia-reto (después de perder la partida)	234
• Figura 73. Versión desenchufada del BCTt [234]	252
• Figura 74. Pantallas de prototipos para módulos de BAC: sudokus (izquierda), cifrado y descifrado (derecha)	257

CAPÍTULO 1. INTRODUCCIÓN

A lo largo del presente capítulo se expone la motivación general que ha llevado a la realización de los trabajos presentados en esta tesis. A continuación, se establecen los objetivos generales propuestos, que se irán desglosando a lo largo del documento. Por último, se incluye la estructura organizativa de la memoria.

1.1. Motivación

El Pensamiento Computacional (PC) es una destreza cognitiva para resolver problemas que debe adquirirse desde una edad temprana, de forma similar a la adquisición de una segunda lengua, para poder así adaptarse a la sociedad informatizada actual. Muchas de las habilidades comúnmente asociadas al PC pueden desarrollarse al aprender a programar como, por ejemplo, la abstracción, la descomposición de problemas, la algoritmia y la depuración de programas [1]. La importancia del PC reside en que estas habilidades para resolver problemas, que podrían parecer solamente relacionados con la informática, pueden emplearse para hacer frente a problemas en muchos de los ámbitos productivos de la sociedad actual, así como a problemas de la vida cotidiana [2]-[4].

Para incluir la enseñanza de la programación y el desarrollo de habilidades de PC en el currículo escolar, es necesario determinar qué debe aprenderse a cada edad y cómo. Sin embargo, al ser una competencia tan reciente, la definición del concepto es aún difusa [5], así como los marcos de referencia operativos que permitan establecer métodos de enseñanza y de evaluación eficaces [4]. Por este motivo, es necesario realizar una investigación teórica sobre este tema y marcar una base operacional en cuanto a definiciones, estrategias y entornos para el desarrollo del PC, así como de instrumentos o herramientas existentes para el aprendizaje y evaluación de esta competencia.

Para establecer si un método de enseñanza del PC es adecuado, es necesario medir, a través de un instrumento específico para la evaluación del PC, el efecto de ese determinado método de enseñanza en el aprendizaje de los estudiantes. En los últimos años han surgido nuevos instrumentos de evaluación del PC, los cuales no suelen orientarse a etapas tempranas [1], [5]. En el momento de la realización del presente estudio, no existía, en la comunidad científica, ninguna prueba de evaluación (test) independiente (no relacionada con un entorno de aprendizaje concreto) y enfocada a alumnos de Educación Primaria. Con esta motivación, en primer lugar, se plantea el diseño y la validación de un test de estas características que nos permita una correcta evaluación del PC en etapas tempranas, el *Beginners Computational Thinking*

test (BCTt) y su adaptación a niños más mayores, competence Computational Thinking test (cCTt), así como su validación y límites de edad para su administración [6]-[8].

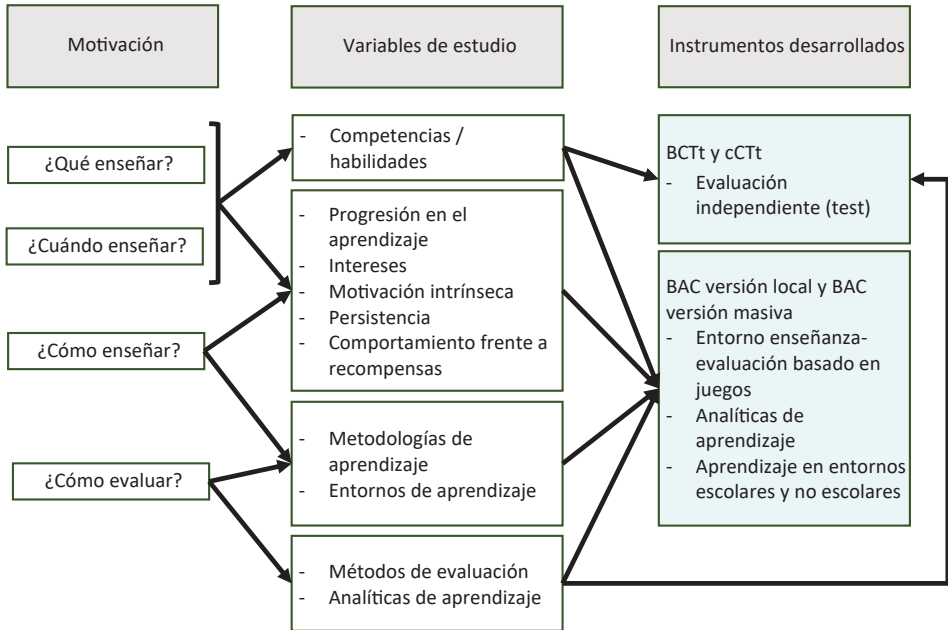
Además, con la motivación de evaluar una herramienta de aprendizaje basado en juegos [9]-[11], fundamentada en el constructivismo [12]-[15], para la enseñanza y evaluación automática del PC mediante analíticas de aprendizaje en juegos [16], [17], se desarrolló y validó *Blue Ant Code* (BAC) [18], [19], que incluye modos de juego individual y colaborativo.

Mediante los instrumentos de evaluación (BCTt y cCTt) y el entorno de aprendizaje y evaluación (BAC), es posible extraer y analizar datos para poder avanzar en el conocimiento del desarrollo de las habilidades de PC en diferentes edades y género. Mediante BAC en entornos escolares y no escolares (versión masiva de BAC), y analíticas de aprendizaje en juegos, es posible también extraer y analizar no solamente las habilidades estrictamente académicas, sino también la progresión en el aprendizaje y disposiciones y predisposiciones para el desarrollo del PC en cuanto a aspectos relativos a la motivación intrínseca, intereses, persistencia y actitudes frente a las recompensas para la enseñanza de la programación y el PC.

La validación de los instrumentos de evaluación y de BAC, permite su utilización con una buena fiabilidad y, además, facilita la realización de nuevos instrumentos y metodologías de enseñanza-evaluación del PC, gracias a las conclusiones que se han obtenido de los procesos de validación. De las experiencias realizadas, se han extraído conclusiones concretas sobre las habilidades de PC asociadas a cada edad y género del alumno y, más allá, sobre la motivación intrínseca de los usuarios para el aprendizaje del PC, sus intereses, persistencia, y actitud frente a las recompensas, también dependiendo de su edad y su género.

De esta forma, los instrumentos y herramientas desarrolladas en esta tesis, así como las experiencias realizadas para su validación han sido motivadas por la búsqueda de diferentes variables relacionadas con el desarrollo del PC para, una vez analizadas, poder no solamente ampliar el conocimiento de estas habilidades, sino también mejorar el diseño del currículo escolar con respecto a esta competencia, así como poder progresar en el desarrollo de herramientas de aprendizaje y evaluación del PC. En la Figura 1 se muestra un esquema de las variables a analizar que motivan el trabajo realizado para esta tesis, y su relación con los instrumentos desarrollados.

Figura 1. Motivación de la tesis e instrumentos desarrollados.



A partir de esta motivación, es importante establecer los objetivos generales y específicos de la tesis, que se abordarán en el siguiente apartado.

1.2. Objetivos generales

Actualmente la definición del concepto de PC es aún difusa, así como los marcos de referencia operativos que permitan establecer **métodos de enseñanza y de evaluación eficaces, sobre todo a etapas tempranas**. Por lo tanto, el objetivo principal de la tesis es:

OP: Avanzar en el conocimiento sobre el PC, sobre todo en su componente operacional, es decir, en cuanto a métodos de enseñanza y de evaluación eficaces, sobre todo en etapas tempranas.

El objetivo principal OP de la tesis se puede dividir en una serie de objetivos específicos que se describen a continuación:

Investigación teórica:

- *OP1*: Recopilar un amplio marco teórico y estado del arte de publicaciones y estudios sobre PC. Se deben considerar los siguientes aspectos:
 - o *OP1a*: Definiciones teóricas y operacionales del PC.
 - o *OP1b*: Marcos, currículos escolares e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas.
 - o *OP1c*: Marco teórico y estrategias para el desarrollo/aprendizaje del PC, así como herramientas y entornos existentes.
 - o *OP1d*: Marco teórico y estrategias para la evaluación del PC, así como instrumentos existentes.

Investigación empírica:

- *OP2*: Establecer una propuesta operativa del PC en base al *OP1* y así constituir la dirección de la investigación empírica mediante hipótesis y objetivos base. Posteriormente, en cada experiencia empírica, se detallarán las hipótesis y objetivos específicos de la misma.
- *OP3*: Diseñar y validar, empíricamente, herramientas para el aprendizaje y la evaluación del PC:
 - o *OP3a*: Diseñar y validar un instrumento de evaluación del PC en edades tempranas.
 - o *OP3b*: Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas.
- *OP4*: Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC:
 - o *OP4a*: Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género.
 - o *OP4b*: Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género.
 - o *OP4c*: Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género.

Además, existe un objetivo común con otros investigadores a nivel internacional, que es el **diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones**, desde la etapa de Educación Infantil hasta la etapa universitaria. Dentro de este marco común, esta tesis se enfoca en edades tempranas: **Educación Infantil y primaria**. De aquí se desprende el objetivo común:

OC: Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones

En la Figura 2 se resumen los objetivos generales de esta tesis:

Figura 2. Objetivos generales de la tesis.

OP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.	
	Marco teórico y estado del arte.	
	OP1	OP1a Definiciones teóricas y operacionales del PC.
		OP1b Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas.
		OP1c Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes.
		OP1d Marco teórico y estrategias para la evaluación del PC e instrumentos existentes.
	OP2	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.
		Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC.
	OP3	OP3a Diseñar y validar un instrumento de evaluación del PC en edades tempranas.
		OP3b Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas.
	OP4	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC.
		OP4a Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género.
		OP4b Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género.
		OP4c Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género.
OC	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones.	

1.3. Experiencias realizadas

En cuanto a la investigación empírica que se desprende de los objetivos OP2, OP3 y OP4, se han realizado varias experiencias que se resumen en la Figura 3. En cada uno de los capítulos a los que se direcciona en la tabla, se detallará cada una de las experiencias, no sin antes establecer una base operacional clara y las hipótesis base generales a todas ellas en el Capítulo 3.

Figura 3. Experiencias realizadas en relación con los objetivos generales.

	Experiencia	Objetivos generales				Capítulo
E1	Desarrollo y validación de prueba independiente para la evaluación del PC en Educación Primaria: Beginners Computational Thinking test (BCTt).	OP3a	OP4a	OP4c	OC	4
E2	Estimación de la validez del BCTt en alumnos de Educación Infantil.	OP3a	OP4a	OP4c	OC	5
E3	Validación y análisis del entorno educativo basado en el juego BAC, en modos individual y colaborativo y análisis de habilidades de PC en Educación Primaria.	OP3b	OP4a	OP4b	OP4c	6
E4	Análisis de motivación intrínseca, intereses, persistencia y comportamiento frente a recompensas en el aprendizaje de la programación en etapas tempranas utilizando el entorno BAC.		OP4a	OP4b	OP4c	7

1.4. Estructura del documento

Tras este primer capítulo de introducción, donde se detallan las hipótesis y los objetivos de este trabajo, el documento se divide en 8 capítulos más cuyo contenido se estructura de la siguiente forma:

- Capítulo 2: Se realiza un estado del arte y se establece un marco teórico, mediante una revisión crítica de la literatura, con respecto a definiciones

teóricas y operacionales del PC; marco, currículos escolares e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas educativas tempranas; y marco teóricos y estrategias para el desarrollo del PC. También se exploran herramientas y entornos existentes; marcos teórico y estrategias para la evaluación del PC; así como instrumentos existentes.

- Capítulo 3: Se exponen las conclusiones que se extraen del capítulo 2 y la dirección operativa para la investigación empírica, especificando las hipótesis generales y las experiencias a realizar.
- Capítulo 4: Se describe la experiencia de desarrollo y validación del instrumento de evaluación del PC: Beginners Computational Thinking Test (BCTt) dirigido a alumnos de Educación Primaria, así como la posterior adaptación del test para los alumnos más mayores en los que el BCTt resulta menos fiable, mediante una colaboración internacional que resulta en el instrumento de evaluación: Competence Computational Thinking Test (cCTt).
- Capítulo 5: Se describe la experiencia dirigida a alumnos de Educación Infantil, en colaboración internacional, mediante el uso de entornos de aprendizaje basados en la robótica, para establecer el límite inferior de edad para el que el BCTt es adecuado.
- Capítulo 6: Se describe la experiencia de desarrollo y validación de un entorno de aprendizaje y evaluación basado en juegos (BAC), con el objetivo general de analizar estrategias para el aprendizaje y evaluación del PC y las habilidades de PC de alumnos de Educación Primaria, entre otros objetivos.
- Capítulo 7: Se describe la experiencia de despliegue de BAC en un entorno no escolar, con el objetivo general de analizar estrategias para el aprendizaje y evaluación del PC, así como analizar las habilidades de PC de alumnos de Educación Primaria, con especial atención en las disposiciones y predisposiciones para el PC basadas en la motivación intrínseca y el juego voluntario.
- Capítulo 8: Se resumen las principales colaboraciones, contribuciones, conclusiones y hallazgos en el contexto de la tesis, así como producción científica a la que ha dado lugar el trabajo.
- Bibliografía y Anexos: Finalmente se incluyen las referencias bibliográficas citadas a lo largo de este documento, así como documentación adicional de la tesis.

CHAPTER 1. INTRODUCTION

Throughout this chapter, the general motivation that has led to the work presented in this thesis is set out. Next, the general objectives are established and be described in detail. Finally, the organisational structure of the document is presented.

1.5. Motivación

Computational Thinking (CT) is a cognitive problem-solving skill that must be acquired from an early age, similar to second language acquisition, in order to adapt to today's computerised society. Many of the skills commonly associated with CT can be developed by learning to program, such as abstraction, problem decomposition, algorithmia, and debugging [1]. The importance of CT relies on the fact that these problem-solving skills, which might seem only computer-related, can be used to address problems in many of the productive domains of today's society as well as problems in everyday life [2]-[4].

In order to include the teaching of programming and the development of CT skills in the school curricula, it is necessary to determine what should be learned at each age and how. However, being such a recent competence, the definition of the concept is still unclear [5], as well as the operational frameworks for effective teaching and assessment strategies [4]. For this reason, it is necessary to conduct theoretical research on this topic and to establish an operational basis in terms of definitions, strategies, and environments for the development of CT, as well as existing instruments or tools for the learning and assessment of this competence.

To establish whether a CT learning strategy is appropriate, it is first necessary to establish whether it impacts learning, which requires a CT assessment instrument that can measure such learning. In recent years, new assessment instruments have emerged, but these are not usually focused on the early stages [1], [5]. At the time of the present study, there was no stand-alone assessment test in the scientific community - not related to a specific learning environment - focused on primary school students. With this motivation, first, we proposed the design and validation of a test of this type that would allow us to properly assess CT in early stages, the Beginners Computational Thinking test (BCTt) and its adaptation to older children, the competence Computational Thinking test (cCTt), as well as its validation and age limits for its administration [6]-[8].

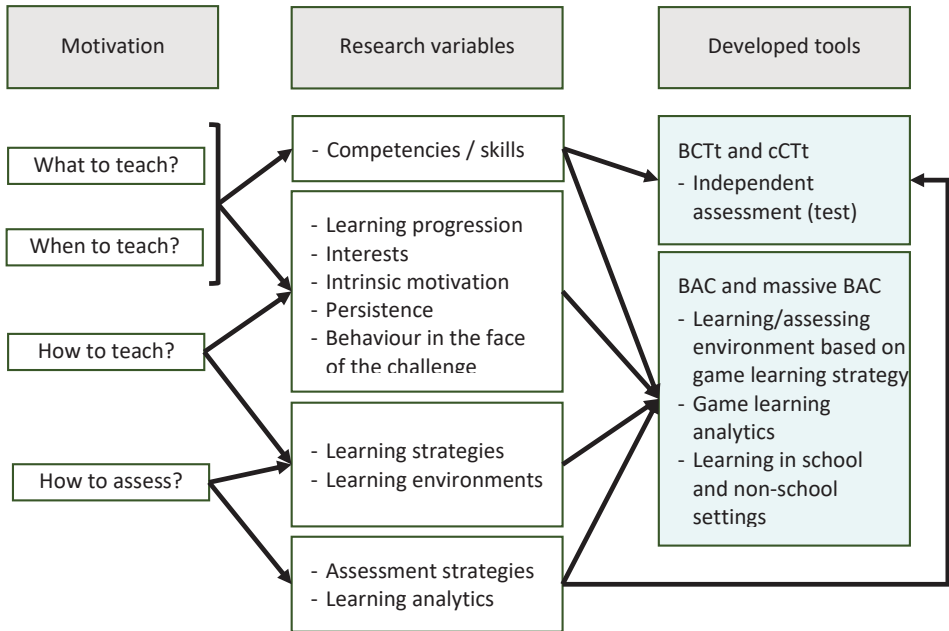
Furthermore, with the motivation of evaluating a game-based learning tool [9]-[11], based on constructivism [12]-[15], for the learning and assessment of CT through game learning analytics [16], [17], Blue Ant Code (BAC) [18], [19], which includes individual and collaborative game modes, was developed and validated.

Using the assessment instruments (both BCTt and cCTt) and the learning and assessment environment (BAC), it is possible to collect and analyse data in order to advance knowledge of the development of CT skills at different age groups and gender. Through BAC in school and non-school environments (massive version of BAC), and game learning analytics, it is also possible to collect and analyse not only strictly academic skills, but also learning progression and dispositions and predispositions for CT development in terms of aspects related to intrinsic motivation, interests, persistence, and behaviours in response to rewards for learning to program and developing CT.

The validation of the assessment instruments and BAC, allows their administration with a good reliability and, in addition, facilitates the development of new instruments and methodologies for learning-assessment of CT, according to the conclusions that have been obtained from the validation procedures. From the studies carried out, specific conclusions have been drawn about the CT skills associated with each student's age and gender and, furthermore, about the students' intrinsic motivation for learning CT, their interests, persistence, and behaviour towards rewards, also depending on their age and gender.

In addition, the instruments and tools developed in this thesis, as well as the studies carried out for their validation, have been motivated by the search for different variables related to the development of CT in order to, once analysed, not only broaden the knowledge of these skills, but also to improve the design of the school curricula with regard to this competence, as well as to progress in the development of tools for learning and assessing CT. Figure 1 shows an outline of the variables to be analysed that motivate the work carried out for this thesis, and their relationship with the instruments developed.

Figure 1. Motivation of the thesis and instruments developed.



Based on this motivation, it is important to establish the general and specific aims of the thesis, which will be addressed in the following section.

1.6. General aims

Currently, the definition of the concept of CT is still unclear, as well as the operational frameworks for **effective learning and assessment strategies**, especially at early stages. Therefore, the main aim of the thesis is:

OP: To advance knowledge about CT, especially in its operational component, i.e., in terms of effective learning and assessment strategies, especially at early stages.

The main aim *OP* of the thesis can be divided into several specific aims which are described below:

Theoretical research:

- *OP1*: Compile a broad theoretical framework and state of the art of publications and studies on CT. The following aspects should be considered:
 - o *OP1a*: Theoretical and operational definitions of CT.

- OP1b: Frameworks, school curricula and initiatives for the development of CT at international and national level at early stages.
- OP1c: Theoretical framework and strategies for CT development/learning, as well as existing tools and environments.
- OP1d: Theoretical framework and strategies for the assessment of CT, as well as existing instruments.

Empirical research:

- OP2: Establish an operational proposal of CT based on OP1 and thus constitute the direction of the empirical research by means of hypotheses and base aims. Subsequently, in each empirical study, the specific hypotheses and aims of the study will be detailed.
- OP3: To design and validate, empirically, tools for learning and assessment of CT:
 - OP3a: To design and validate a CT assessment tool aimed at early ages.
 - OP3b: To develop and validate a tool for learning and assessment of CT at an early age.
- OP4: To carry out empirical studies to analyse data related to CT skills as well as CT dispositions and predispositions:
 - OP4a: To collect and analyse data to improve the design of the school curricula related to CT according to age and gender.
 - OP4b: To collect and analyse data to improve the design of learning tools related to CT according to age and gender.
 - OP4c: To collect and analyse data to improve the design of CT assessment tools according to age and gender.

In addition, there is a common aim with other researchers at international scale, which is the **design and validation of CT assessment tools covering a wide range of ages and populations**, from Early Childhood Education to University. Within this common framework, this thesis focuses on early ages: **pre-school and primary school**. Thus, the common aim is the following:

OC: Design and validation of CT assessment instruments covering a wide range of ages and populations.

Figure 2 summarises the general aims of this thesis:

Figure 2. General aims of the thesis.

OP	To advance knowledge about CT, especially in its operational component, i.e., in terms of effective learning and assessment strategies, especially at early stages.		
	Compile a broad theoretical framework and state of the art of publications and studies on CT		
	OP1	OP1a Theoretical and operational definitions of CT	
		OP1b Frameworks, school curricula and initiatives for the development of CT at international and national level at early stages.	
		OP1c Theoretical framework and strategies for CT development/learning, as well as existing tools and environments.	
		OP1d Theoretical framework and strategies for the assessment of CT, as well as existing instruments.	
	OP2	Establish an operational proposal of CT based on OP1 and thus constitute the direction of the empirical research by means of hypotheses and base aims.	
		To design and validate, empirically, tools for learning and assessment of CT.	
	OP3	OP3a To design and validate a CT assessment tool aimed at early ages.	
		OP3b To develop and validate a tool for learning and assessment of CT at an early age.	
	OP4	To carry out empirical studies to analyse data related to CT skills as well as CT dispositions and predispositions.	
		OP4a	To collect and analyse data to improve the design of the school curricula related to CT according to age and gender.
		OP4b	To collect and analyse data to improve the design of learning tools related to CT according to age and gender.
		OP4c	To collect and analyse data to improve the design of CT assessment tools according to age and gender.
OC	Design and validation of CT assessment instruments covering a wide range of ages and populations.		

1.7. Studies conducted

With regard to the empirical research arising from the aims OP2, OP3 and OP4, several studies have been conducted, which are summarised in Figure 3. In each of the chapters addressed in the table, each of the studies will be detailed, but before that, in Chapter 3, a clear operational basis and the general basic hypotheses for all of them will be established.

Figure 3. Studies carried out in relation to the general aims.

Study	General aims	Chapter
E1 Development and validation of a stand-alone test for the assessment of CT in Primary Education: Beginners Computational Thinking test (BCTt).	OP3a OP4a OP4c OC	4
E2 Estimation of the validity of the BCTt in Early Childhood Education students.	OP3a OP4a OP4c OC	5
E3 Validation and analysis of the educational environment based on the BAC game, in individual and collaborative modes and analysis of CT skills in Primary Education.	OP3b OP4a OP4b OP4c	6
E4 Analysis of intrinsic motivation, interests, persistence, and behaviour in response to rewards in the learning of programming in early stages using the BAC environment.	OP4a OP4b OP4c	7

1.8. Document structure

After this first introductory chapter, which details the hypotheses and the aims of this work, the document is divided into 8 chapters structured as follows:

- Chapter 2: A state of the art and theoretical framework is conducted, through a critical literature review, with respect to theoretical and operational definitions of CT; framework, school curricula and initiatives for CT development at international and national levels in early educational stages; and theoretical framework and strategies for CT development. It also explores

existing tools and environments; theoretical frameworks and strategies for CT assessment; and existing instruments.

- Chapter 3: The conclusions drawn from Chapter 2 and the operational direction for the empirical research are presented, specifying the general hypotheses and the studies to be carried out.
- Chapter 4: The study of the development and validation of the CT assessment instrument: Beginners Computational Thinking Test (BCTt) aimed at Primary School students is described, as well as the subsequent adaptation of the test for older students for whom the BCTt is less reliable, through an international collaboration resulting in the assessment instrument: Competence Computational Thinking Test (cCTt).
- Chapter 5: Describes the study aimed at pre-school pupils in an international collaboration using robotics-based learning environments to establish the lower age limit for which the BCTt is suitable.
- Chapter 6: Describes the study of the development and validation of a game-based learning and assessment environment (BAC), with the general aim of analysing strategies for learning and assessing CT, as well as CT skills of Primary School students, among other aims.
- Chapter 7: The study describes the BAC deployment in a non-formal setting, with the general aim of analysing strategies for learning and assessing CT, as well as analysing the CT skills of Primary School pupils, with a special focus on dispositions and predispositions for CT based on intrinsic motivation and voluntary play.
- Chapter 8: The main collaborations, contributions, conclusions, and findings in the context of the thesis are summarised, as well as the scientific production related to it.
- Bibliography and Annexes: Finally, the bibliographical references cited throughout this document are included, as well as additional documentation of the thesis.

CAPÍTULO 2. MARCO TEÓRICO Y ESTADO DEL ARTE

Tal y como se establece en el objetivo general *OP1: Recopilar un amplio marco teórico y estado del arte de publicaciones y estudios sobre PC (ver apartado 1.2. Objetivos)*, en este capítulo se realiza una revisión crítica de la literatura con respecto a: definiciones teóricas y operacionales del PC (objetivo OP1a); marco, currículos escolares e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas (objetivo OP1b); marco teóricos y estrategias para el desarrollo del PC, así como herramientas y entornos existentes (objetivo OP1c); y marco teórico y estrategias para la evaluación del PC, así como instrumentos existentes (objetivo OP1d).

2.1. Pensamiento Computacional

Existe un creciente movimiento internacional que se centra en las habilidades del siglo XXI, necesarias para que los estudiantes tengan éxito en una sociedad digital en constante cambio. Éstas incluyen una amplia gama de habilidades, como, por ejemplo: aprendizaje e innovación, información, y medios de comunicación y tecnología [20], de las cuales, las habilidades de aprendizaje e innovación suelen clasificarse en cuatro bloques conocidos como las cuatro "C": pensamiento crítico, creatividad, colaboración, y comunicación [21]. En las últimas décadas, se ha producido un creciente reconocimiento de estas habilidades como elementos esenciales de los planes de estudio [11].

Por otro lado, el pensamiento computacional (PC) es una habilidad cognitiva fundamental para la resolución de problemas mediante estrategias computacionales como la abstracción o la algoritmia, pero que cuya aplicación práctica no se limita a las actividades que puedan llevarse a cabo en el ámbito de la informática, sino que puede aplicarse ampliamente en la vida cotidiana. El PC es esencial para adaptarse a la sociedad tecnológica actual y futura [4] y puede ser considerado como una habilidad básica más o la quinta "C" del siglo XXI, por lo que hay consenso general en que el PC debe ser incluido como una competencia básica en el currículo escolar [1], [22]-[26].

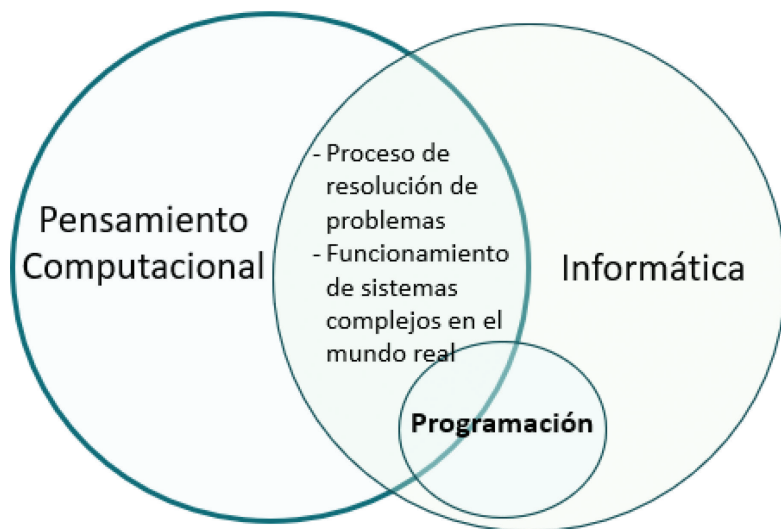
A pesar de que el aprendizaje de habilidades de PC se está incluyendo en los planes de estudio de países de todo el mundo, la definición genérica del concepto aún no está clara. Sin embargo, en las últimas décadas se han propuesto multitud de definiciones, lo que ha dado lugar a un amplio debate [5].

En algunos casos se han resaltado los riesgos de no contar con una definición consensuada [27], [28]. La relación entre programar, la informática y el PC es

evidente, incluso en ocasiones se ha considerado que aprender a programar, o incluso la informática, podrían ser sinónimos del PC. Para resolver problemas, los ordenadores tienen que recibir instrucciones y la habilidad que se requiere para transmitir esas instrucciones a los ordenadores es la programación. Uno de los beneficios del PC es que habilita para poder programar, pero el PC abarca muchas otras habilidades [29]. Por otro lado, el proceso que subyace a la programación es el PC y, además, tanto la programación como el PC son necesarios en el estudio de la informática [30]. El solapamiento entre la informática y el PC se produce en el proceso de resolución de problemas y en la intención de entender cómo funcionan los sistemas complejos en el mundo real [3].

Sin embargo, el PC no abarca solamente el proceso mental necesario para programar o para aprender a hacerlo, sino que lo trasciende, como se muestra en la Figura 4, e incluye habilidades que todo el mundo debería adquirir y no sólo los informáticos o los programadores [31]. Tampoco el PC se adquiere exclusivamente al aprender a programar, ni se trata de aprender a “pensar” como un ordenador, sino que es una destreza exclusivamente humana que incluye habilidades para, por ejemplo, manejar información con eficiencia, o la capacidad de abstracción para la resolución de problemas [1], [32], [33]. Mills *et al.* [34] amplían el esquema de la Figura 4, creando el superconjunto “computación” que definen como cualquier actividad o área de estudio que aprovecha los métodos, modelos o sistemas computacionales, como por ejemplo la gestión de la información, la ingeniería informática, la inteligencia artificial, la ciencia de los datos, los medios de entretenimiento, etc. Implica, en definitiva, cualquier habilidad o práctica de las ciencias de la computación y/o el PC.

Figura 4. Relación entre el Pensamiento Computacional, la Informática, y la Programación [30].



También pueden encontrarse similitudes con otros tipos de pensamiento. En el caso del pensamiento matemático hay cierto solapamiento con el PC en los procesos de resolución de problemas, modelado, interpretación y análisis de datos, y estadística, como se puede ver en la Figura 5 [35]. En cuanto al Pensamiento de Diseño (*Design Thinking*), que requiere pensar como un diseñador, tiene en común con el PC el objetivo de solucionar problemas, pero enfoca el proceso de diferente forma como puede verse en la Figura 6. Kelly *et al.* conciben los dos tipos de pensamiento como un proceso dual no excluyente [36].

Figura 5. Relación entre el Pensamiento Computacional y el Pensamiento Matemático según Sneider *et al.* [35].

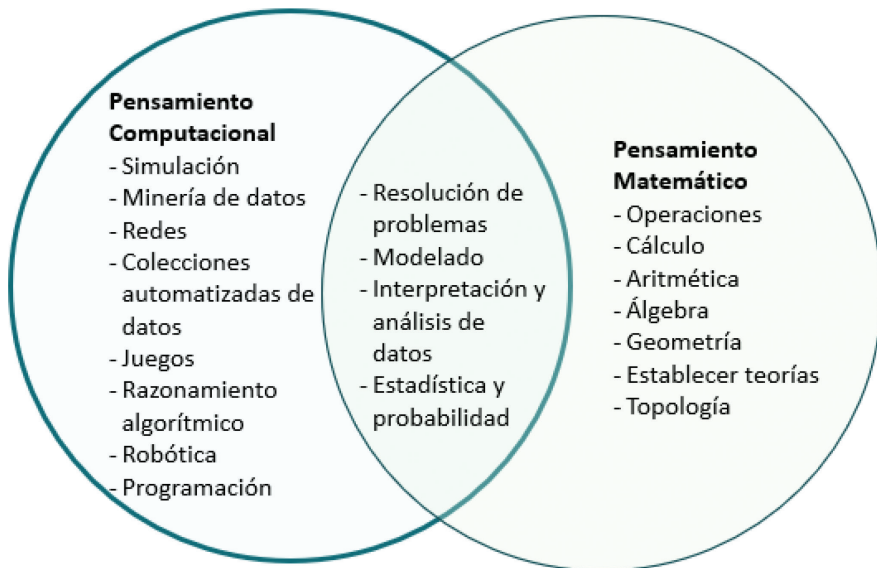


Figura 6. Relación dual entre el Pensamiento Computacional y el Pensamiento de Diseño según Kelly *et al.* [36].

Pensamiento Computacional	Pensamiento de Diseño
Se intenta ampliar el marco del problema para captar su complejidad.	Se intenta reducir el marco del problema para abstraer la complejidad innecesaria.
Se intenta crear una solución específica para problema.	Se pretende crear una solución general para el problema.

Como se ha visto, aunque el PC tiene su origen en la informática y es la disciplina en la que podemos encontrar mayores similitudes, se diferencia

de ella en que habilita a las personas a transferir las habilidades a dominios diferentes [2]. Así, el PC se puede aplicar para la resolución de problemas en muchas disciplinas como, por ejemplo, para calcular la trayectoria de una tormenta o para predecir la evolución geográfica de un virus; y también para para resolver problemas cotidianos como, por ejemplo, para optimizar el flujo en la cocina de un restaurante y aportar un servicio adecuado, para reducir el tiempo de espera en un centro de atención telefónica, o para planificar una buena estrategia en un juego de mesa, entre otras. Esta habilidad del PC, en que se transfiere la capacidad de resolver problemas a situaciones cotidianas, y que más adelante llamaremos *generalización*, es relevante también en el caso de la motivación para los profesores, ya que una de sus inquietudes frecuentes es sobre la necesidad de aprender a programar y su aplicación real [37], así como su aplicación instrumental en el contexto escolar [38].

Un posible proceso de pensamiento para resolver problemas que implica PC sería por ejemplo el siguiente: cuando se presenta un problema, se reformula para representarlo de forma abstracta con modelos de datos o con algoritmos; después se analizan críticamente y se prueban varias soluciones antes de seleccionar y aplicar la más adecuada, eficiente o exacta, y además se cuestiona si se podría mejorar o entender el problema de otra forma [30].

Mientras que la programación se considera una habilidad técnica y la informática una disciplina académica, el PC es un proceso de resolución de problemas fundamental para la informática, pero que puede aplicarse de forma más amplia a otras disciplinas y ámbitos, y en la vida cotidiana. Adquirir habilidades de PC puede ser motivador para los estudiantes en el estudio de la informática y otras profesiones relacionadas con la tecnología [39], además de tener aplicaciones importantes en otras áreas de ciencia, tecnología, ingeniería y matemáticas (conocidas como áreas STEM) [40], [41]. Las habilidades de PC también están ligadas a la creatividad y la innovación [42]. Son tanto habilidades que deben aprenderse, como una forma de aprender, crear, descubrir y dar sentido al mundo, que es crítica para adaptarse a la sociedad altamente informatizada actual.

El PC se definió formalmente por primera vez por Jeannette M. Wing como un *«proceso humano de resolución de problemas que utiliza la descomposición y requiere pensar en múltiples niveles de abstracción»* [43] y, a partir de entonces, se han sugerido multitud de nuevas definiciones en un esfuerzo por concretar el concepto, que aún es difuso debido a su carácter reciente.

Se considera que las definiciones de PC pueden clasificarse en tres tipos [44]:

- **Definiciones genéricas** que se centran en las habilidades implicadas en la resolución de problemas (por ejemplo, la definición de Aho [45]).

- **Definiciones operativas** que proporcionan diferentes descomposiciones en factores o habilidades de PC (por ejemplo, la descomposición que proponen Shute *et al.* [1]).
- **Definiciones o marcos de referencia educativos** para el desarrollo del PC en el aula (por ejemplo, el marco tridimensional (3D) [46]).

A continuación, se explorarán estas definiciones y se seleccionarán o pondrán definiciones concretas como marco o contexto para el desarrollo de la tesis.

2.1.1. Definiciones genéricas

El término "Pensamiento Computacional" fue mencionado en 1980 por Seymour Papert en su libro *Mindstorms* [47], pero no fue hasta 2006 cuando Jeannette M. Wing lo definió formalmente de la siguiente forma:

«El PC implica la resolución de problemas, el diseño de sistemas y la comprensión del comportamiento humano, recurriendo a los conceptos fundamentales de la informática. El PC incluye una serie de herramientas mentales que reflejan la amplitud del campo de la informática [...]. Representa una actitud y un conjunto de habilidades de aplicación universal que todo el mundo, no sólo los informáticos, estaría dispuesto a aprender y utilizar» [43].

A partir de ese momento se han propuesto multitud de definiciones. Por ejemplo, Lu y Fletcher, en 2009, resaltaron la importancia de desarrollar el PC de forma que ayude a procesar la información y las tareas eficientemente y de forma sistemática. Además, apuntan a la existencia de una dicotomía entre el PC y la informática como asignatura académica, defendiendo la siguiente tesis: *«La programación es a la informática lo que la construcción de pruebas es a las matemáticas y lo que el análisis literario es al inglés»*. Y añaden que: *«Antes de que los estudiantes se inscriban en los cursos de programación, se requiere una preparación sustancial en el pensamiento computacional»* [48]. Estos autores proponen la creación de un lenguaje de pensamiento computacional (LPC), diferente a un lenguaje de programación, que incluya vocabulario y símbolos que puedan utilizarse para anotar y describir la computación y la abstracción, sugerir información y ejecución, y proporcionar una notación en torno a la cual pueda describirse la comprensión semántica de los procesos computacionales.

Una de las definiciones más citadas en la literatura es la de Cuny, Snyder y Wing, acuñada en 2010: *«El pensamiento computacional es el proceso de pensamiento que implica la formulación de problemas y sus soluciones, de*

manera que éstas se representen de forma que puedan ser llevadas a cabo eficazmente por un agente procesador de información» [49].

Aho, en 2011, define nuevamente el PC como *«el proceso de pensamiento que implica la formulación de problemas para que sus soluciones puedan representarse como pasos computacionales y algoritmos»*, y considera como una parte importante de este proceso el encontrar modelos de computación adecuados con los que formular el problema y derivar sus soluciones [45].

Barr *et al.* en 2011 definen el PC en un contexto educativo, como un enfoque para la resolución de problemas de tal forma que podría ser implementada con un ordenador. Consideran que los estudiantes no son sólo usuarios de herramientas sino constructores de herramientas, que utilizan un conjunto de conceptos como la abstracción, la recursividad y la iteración, para procesar y analizar datos y crear artefactos reales y virtuales. También consideran que el PC es una metodología de resolución de problemas que puede automatizarse, transferirse y aplicarse en todas las materias. El PC se puede aplicar en cualquier tipo de razonamiento y dominio como, por ejemplo, en física cuántica, biología avanzada, sistemas informáticos humanos, y desarrollo de herramientas computacionales útiles. Además, estos investigadores añaden también conceptos como áreas de valor, motivaciones, disposición personal, actitudes y sentimientos aplicables a PC como la confianza en uno mismo, la persistencia ante los problemas difíciles, la colaboración con los compañeros o el autoconocimiento. [41].

Por su parte, la British Royal Society definió en 2012 el PC como *«el proceso de reconocer los aspectos de la computación en el mundo que nos rodea, y aplicar las herramientas y técnicas de la Informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales.»* y animó a la enseñanza del PC en los colegios de Reino Unido [50].

En 2015, Berland and Wilensky definieron el PC como *«la capacidad de pensar con el ordenador como herramienta»* [2] en un contexto de aprendizaje a través de la robótica. Por su parte, en un entorno orientado a la programación en Educación Primaria, Israel *et al.*, en 2015 utilizan el término PC para referirse a las habilidades de estudiantes que *«usan los ordenadores para modelar sus ideas y desarrollar programas que potencien esos mismos programas»*, incluyendo así la programación informática como una parte del PC [29]. Estas definiciones, como muchas otras propuestas en la literatura, están enfocadas a investigaciones concretas o a contextos determinados, por lo que son difícilmente generalizables.

Tedre y Denning, en 2016 consideran que el PC es *«una frase popular que se refiere a un conjunto de ideas computacionales y hábitos mentales que*

las personas de las disciplinas informáticas adquieren a través de su trabajo en el diseño de programas, software, simulaciones y cálculos realizados por máquinas» [51]. Por su parte, Shute et al. en 2017, después de un extenso meta-estudio, definen el PC como «la base conceptual necesaria para resolver problemas de forma eficaz y eficiente (es decir, algorítmicamente, con o sin la ayuda de ordenadores) con soluciones que son reutilizables en diferentes contextos» [1]. Finalmente, según Grover (2018), puede definirse como las habilidades de pensamiento que preceden a la codificación y la programación, y que se aplican para comprender un problema y formular una solución como un informático [26].

En 2019, Moreno-León et al. realizaron un análisis de las principales definiciones de PC, extrayendo los elementos más influyentes en ellas que resultaron ser "problema", "ordenador", "solución" y "proceso". Desde este prisma, proponen una nueva definición para PC: *«la capacidad de formular y representar problemas para resolverlos haciendo uso de herramientas, conceptos y prácticas de la disciplina informática, como la abstracción, la descomposición o el uso de simulaciones»*. En su opinión, esta nueva definición basada en el análisis de datos aclara la relación entre el PC y la programación, siendo la primera una capacidad cognitiva del sujeto, y la segunda sólo uno de los medios para desarrollarla. Sin embargo, los investigadores se extrañaron de que las palabras "cognición" y "habilidad cognitiva" no apareciesen como términos clave, quizá debido a que las definiciones son acuñadas por informáticos en lugar de por psicólogos [52].

Con el objetivo de proporcionar una definición más breve y concisa, y basándonos en el estudio anterior, en esta tesis se propone la siguiente definición: **El pensamiento computacional es una habilidad cognitiva que permite resolver problemas utilizando estrategias computacionales.**

2.1.2. Definiciones operativas

De la misma forma en la que han ido surgiendo definiciones genéricas de PC, también encontramos diferentes definiciones operativas de PC. Estas definiciones incluyen descomposiciones del PC en distintos factores, componentes o habilidades. En este apartado, exploraremos las que consideramos más relevantes en la literatura.

Inicialmente, Wing (2006) [43] incluye cinco procesos cognitivos con el objetivo de resolver problemas de forma eficiente y creativa:

1. *Abstracción* - Modelar los aspectos centrales de los problemas o sistemas complejos.

2. *Reformulación de problemas* - Reformular un problema para convertirlo en uno solucionable y conocido.
3. *Recursión* - Construir un sistema de forma incremental a partir de la información precedente.
4. *Descomposición de problemas* - Descomponer el problema en unidades manejables.
5. *Pruebas sistemáticas* - Realizar acciones intencionadas para obtener soluciones.

De entre estos componentes o procesos de PC, Wing destaca la abstracción como el más importante [3].

En 2010, el *National Research Council* [31], divide el PC en cinco elementos esenciales y universales en diferentes dominios:

1. *Pruebas de hipótesis* – seleccionar los parámetros adecuados para las pruebas de entre todas las posibilidades.
2. *Gestión de datos* – recopilar datos de varias fuentes, procesar patrones y representar los datos de forma significativa.
3. *Paralelismo* – procesar la información simultáneamente desde varias fuentes o dimensiones.
4. *Abstracción* – modelar el funcionamiento de problemas o sistemas complejos.
5. *Depuración* – encontrar y corregir errores después de haber construido el modelo.

Barr y Stephenson (2011) [41] dividen el PC en 6 componentes, coincidiendo con la descomposición propuesta por Wing [43] en los dos primeros:

1. *Abstracción*.
2. *Reformulación de problemas*.
3. *Organización y análisis de datos* - organizar los datos recopilados y analizarlos.
4. *Automatización* - Lograr que un proceso o sistema opere de forma automática.

5. *Eficiencia* - Crear soluciones óptimas.
6. *Generalización* - Aplicar estrategias de PC para la resolución de problemas nuevos.

Además, se considera muy relevante que estos autores también incluyan otras características que consideran importantes al enfrentarse al PC como por ejemplo las capacidades previas, disposiciones y predisposiciones de los estudiantes, así como la "cultura de la clase". Las capacidades que ellos consideran importantes son: diseñar soluciones para un problema; implementar dichas soluciones; probar y depurar; modelar, ejecutar simulaciones y analizar sistemas; reflexionar en las prácticas y comunicar; usar el vocabulario específico; reconocer abstracciones y moverse entre los distintos niveles de abstracción; innovar, explorar y ser creativo a través de diferentes disciplinas; resolver problemas en grupo; y utilizar diversas estrategias de aprendizaje. Por otra parte, en cuanto a las disposiciones y predisposiciones, se intentan valorar áreas como los valores, la motivación, los sentimientos, los estereotipos y las actitudes de los estudiantes frente al PC. Estas disposiciones y predisposiciones incluyen: confianza frente a la complejidad; persistencia cuando se trabaja con problemas difíciles; habilidad para manejar la ambigüedad; habilidad para manejar problemas con finales abiertos; poder dejar de lado las diferencias para trabajar con otras personas en pos de un fin o solución común; y ser consciente las propias fortalezas y debilidades a la hora de trabajar en grupo [41].

También en 2011, la Sociedad Internacional para la Tecnología en la Educación (ISTE) y la Asociación de Profesores de Ciencias de la Computación (CSTA) colaboraron con líderes de la educación superior, la industria, y profesores de las etapas de Educación Infantil, primaria y secundaria (K-12) para desarrollar una definición de PC. La ISTE y la CSTA recabaron la opinión de casi 700 profesores, investigadores y profesionales de la informática para llegar a una definición operacional conjunta, y definen el PC como el proceso de resolución de problemas que incluye las siguientes características:

1. Formular problemas de manera que nos permita utilizar un ordenador y otras herramientas para ayudar a resolverlos.
2. Organizar y analizar lógicamente los datos.
3. Representar los datos a través de abstracciones como modelos y simulaciones.
4. Automatizar las soluciones a través del pensamiento algorítmico, como una serie de pasos ordenados.

5. Identificar, analizar y poner en práctica posibles soluciones con el objetivo de lograr la combinación más eficiente y eficaz de pasos y recursos.
6. Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas.

Además, consideran también, al igual que Barr y Stephenson [41], que estas habilidades se apoyan y potencian en una serie de disposiciones o actitudes que son dimensiones esenciales del PC. Estas disposiciones o actitudes incluyen: la confianza en el manejo de la complejidad; la persistencia en el trabajo con problemas difíciles; la tolerancia a la ambigüedad; la capacidad de tratar con problemas abiertos y la capacidad de comunicar y trabajar con otros para lograr un objetivo o solución común [53].

Grover y Pea, en 2013, teniendo en cuenta diferentes investigaciones en contextos educativos, categorizan el PC en los siguientes componentes: abstracción y generalización de patrones; procesamiento sistemático de la información; sistemas simbólicos y representaciones; nociones de algoritmia en los flujos de control; descomposición estructurada de problemas; iteración, recursión y pensamiento paralelo; lógica condicional; eficiencia y límites de ejecución; y depuración y detección sistemática de errores [5].

En 2014, Bers *et al.* categorizan el PC en *abstracción, generalización y actividades de prueba y error*. De esta forma, resaltan la depuración, es decir, la identificación y corrección de errores cuando la solución aportada no funciona tal y como se había esperado, como uno de los aspectos más importantes del PC [54].

Atmatzidou y Demetriadis (2016) realizan un estudio sobre PC en el contexto de la robótica educativa y desde el análisis de diferentes descomposiciones de otros autores en el mismo contexto, desglosan el PC en cinco facetas:

1. *Abstracción* – Proceso de obtener algo simple desde algo complejo, obviando los detalles, encontrando patrones, e identificando ideas de los detalles tangibles.
2. *Generalización* – Transferencia de un proceso de resolución de problemas a una variedad grande de problemas
3. *Algoritmia* – Identificación de instrucciones específicas y explícitas que, paso a paso, llevan a cabo un proceso.
4. *Modularidad* - Desarrollo de procesos autónomos que encapsulan una serie de comandos de uso frecuente, que ejecutan una función específica y pueden usarse en problemas diferentes.

5. *Descomposición* - Proceso de descomponer problemas en otros más pequeños que pueden resolverse con mayor facilidad.

Además, estos autores especifican cuales son las habilidades que deben poseer los estudiantes para poder ser capaces de realizar cada una de las facetas anteriores. Por ejemplo, para la abstracción, deben poder separar la información importante de la redundante; analizar y especificar comportamientos comunes en estructuras de programación en diferentes fragmentos de código; e identificar abstracciones entre diferentes entornos de programación [55].

También en 2016, Anderson, divide el PC en cinco componentes: descomposición de problemas, reconocimiento de patrones, abstracción, diseño de algoritmos, y evaluación de las soluciones [56].

Shute *et al.* (2017) categorizan las habilidades de PC en las cuatro facetas que surgen con más frecuencia en la literatura: *abstracción*, *descomposición*, *algoritmos* y *depuración* y, además, identifican la *iteración* y la *generalización* como dos componentes más [1]. En el mismo año, Rose *et al.* [57] proponen la siguiente descomposición:

1. *Abstracción y generalización*: eliminar los detalles de un problema y formular una solución genérica.
2. *Algoritmia y procedimientos*: secuencias de pasos y reglas para resolver un problema.
3. *Recogida, análisis y representación de datos*: uso y análisis de datos para ayudar a resolver un problema.
4. *Descomposición*: división de un problema en partes.
5. *Paralelismo*: más de una evento ocurre al mismo tiempo.
6. *Depuración, pruebas y análisis*: identificar, eliminar y corregir errores.
7. *Estructuras de control*: uso de sentencias condicionales y bucles.

Zapata-Ros & Pérez-Paredes (2018) incluyen en las habilidades de PC multitud de componentes como por ejemplo *análisis ascendente*, *análisis descendente*, *creatividad*, *heurística*, *pensamiento divergente*, *resolución de problemas*, *pensamiento abstracto*, *recursividad*, *iteración*, *método ensayo-error*, *método colaborativo*, *patrones*, *sinéctica*, *metacognición*, *cinesesia* y *metacognición*. Bajo este último enfoque, el PC se superpone, en cierta medida, a las ya consideradas habilidades del siglo XXI, en lugar de

constituir por sí mismo la quinta "C" de dichas habilidades, como propone Grover (2018) [26].

En la Tabla 1 se puede ver un resumen de las principales descomposiciones encontradas en la literatura. Como puede apreciarse, se distinguen seis bloques claros. El primero de ellos, engloba la abstracción, el reconocimiento de patrones y el simbolismo o representación; todas las descomposiciones en la literatura incluyen la abstracción como uno de los elementos del PC y muchas de ellas lo identifican como el más importante. En el segundo bloque, destaca la organización y el análisis de la información o los datos, ya que gran parte de las descomposiciones incluyen este elemento. En un tercer bloque se da importancia a la descomposición del problema en otros más pequeños y a la reformulación de los problemas, siempre para encontrar más fácilmente una solución; nuevamente, prácticamente todas las descomposiciones incluyen esta faceta del PC. En cuarto lugar, la mayor parte de las descomposiciones incluyen la generalización, como una forma de poder aplicar las soluciones encontradas a problemas en otros ámbitos, en esta tesis se considera este elemento como uno de los pilares del PC, ya que el desarrollo de éste implica que puedan aplicarse las habilidades adquiridas en otras disciplinas y en los problemas de la vida diaria. En el bloque cinco, se sugieren distintos elementos o estrategias que claramente están vinculadas a la programación y que sugieren la estructura de la solución a un problema en pasos ordenados mediante la algoritmia. Finalmente, el sexto bloque alude a la necesidad de probar la solución propuesta y depurar los errores.

Wing	National Research Council	Barr y Stephenson	ISTE y CSTA	Grover y Pea	Bers et al.	Atmatzidou et al.	Anderson	Shute et al.	TESIS
		Automatización	Automatización / algoritmia	Algoritmia		Algoritmia	Diseño de algoritmos	Algoritmos	Algoritmia
Recursión	Paralelismo			Iteración, recursión, paralelismo				Iteración	
				Lógica condicional					
						Modularidad			
		Eficiencia		Eficiencia y límites de ejecución					
Pruebas sistemáticas	Pruebas de hipótesis			Depuración / detección errores	Actividades de prueba y error		Evaluación de las soluciones	Depuración	Depuración de errores
	Depuración								

En esta tesis, se propone la siguiente descomposición del PC en seis factores y sus definiciones, extrayendo el elemento más significativo de cada uno de los bloques, que en la mayor parte de los casos engloba al resto de elementos del bloque (Tabla 1):

1. **Abstracción** – Proceso de obtener algo simple desde algo complejo, obviando los detalles, encontrando patrones e identificando ideas.
2. **Análisis de datos** – Buscar, seleccionar, organizar y analizar lógicamente los datos.
3. **Descomposición de problemas** – Descomponer problemas en otros más pequeños que pueden resolverse con mayor facilidad.
4. **Algoritmia** – Identificar instrucciones específicas y explícitas que, paso a paso, llevan a cabo un proceso.
5. **Depuración de errores** – Identificar y corregir los errores en la solución aportada.
6. **Generalización** – Transferir un proceso de resolución de problemas a una variedad grande de problemas.

Además, proponemos también la identificación de disposiciones y predisposiciones para el aprendizaje del PC, tal y como se plantea en la literatura [41], [53]. Esta identificación tendrá especial relevancia en el marco de la tesis, ya que se analizarán estos aspectos motivacionales y de persistencia ante los retos en el *Capítulo 7. Análisis de competencias: motivación intrínseca, persistencia y recompensas mediante analíticas de aprendizaje con BAC*.

2.1.3. Marcos de referencia educativos

En los últimos tiempos, ha habido también multitud de intentos por describir **marcos de referencia educativos** para el desarrollo del PC en el aula. En este apartado, trataremos de destacar los más relevantes de los últimos años.

En 2011, Barr y Stephenson [41], además de la definición operativa descrita en el apartado previo de esta tesis, cuya importancia radica en la inclusión de disposiciones y predisposiciones de los estudiantes frente al desarrollo del PC, proponen un modelo didáctico para el desarrollo del PC en el aula. Este modelo destaca porque plantea incluir conceptos de PC en Educación Infantil y primaria, además de en informática, en otras asignaturas y disciplinas: matemáticas, ciencias naturales, ciencias sociales y lengua. Incluye una tabla de posibles actividades para incorporar el desarrollo del PC en estas disciplinas

ligadas a cada uno de los componentes del PC, por ejemplo, plantea incorporar gráficos o estadísticas de batallas en ciencias sociales para desarrollar el componente de “recolección de datos”; reconocer patrones en frases del lenguaje para “análisis de datos”; o usar símiles y metáforas en lengua para desarrollar la “abstracción”. También animan a los gobiernos para que realicen esfuerzos por incluir el desarrollo del PC en los currículos escolares y aboga por una mayor formación a los profesores. Sin embargo, este modelo es incompleto, ya que no define claramente cada componente; hay cierta confusión entre los conceptos y las habilidades; y solamente ofrece un ejemplo de actividad para cada componente de PC por asignatura [1].

En 2016, Atmatzidou y Demetriadis [55], proponen un modelo simple y descriptivo que se basa en el desarrollo del PC en el aula mediante actividades de robótica educativa. Su modelo se apoya en la definición operativa que ellos mismos proponen y que se describió en el apartado 2.1.2. Estos autores especifican cuales son las habilidades que deben poseer los estudiantes para poder ser capaces de alcanzar un desarrollo adecuado en cada faceta propuesta de PC (Tabla 2). También proponen preguntas asociadas a cada habilidad de PC para la autorreflexión del estudiante sobre su programa creado (en el entorno del estudio de actividades de robótica. Por ejemplo, para reflexionar sobre la abstracción, el estudiante puede preguntarse: ¿cuáles son los detalles irrelevantes y no necesarios en la solución aportada? Este modelo es únicamente descriptivo y se circunscribe a las actividades de robótica realizadas demasiado específicas, por lo que es difícilmente generalizable.

Tabla 2. Modelo de PC: Qué debe ser capaz de hacer el estudiante para considerar que ha desarrollado cada habilidad de PC según Atmatzidou y Demetriadis [55].

Habilidades de PC	Capacidades del estudiante
Abstracción	1. Separar la información importante de la redundante. 2. Analizar y especificar comportamientos o estructuras de programación comunes entre diferentes fragmentos de programación. 3. Identificar las abstracciones entre los diferentes entornos de programación.
Generalización	Ampliar una solución existente en un problema determinado para cubrir más posibilidades/casos.

Habilidades de PC	Capacidades del estudiante
Algoritmia	1. Indicar explícitamente los pasos del algoritmo. 2. Identificar diferentes algoritmos eficaces para un problema dado. 3. Encontrar el algoritmo más eficiente.
Modularidad	Desarrollar secciones de código autónomas para su uso en los mismos o diferentes problemas.
Descomposición	Dividir un problema en partes más pequeñas/simplificadas que sean más fáciles de gestionar.

Weintrop *et al.* [58], en 2016, elaboraron un marco que está formado por cuatro categorías de PC: prácticas con datos; modelado y simulación; resolución de problemas de computación; y pensamiento sobre sistemas. Los autores, al igual que Atmatzidou y Demetriadis [55], proponen ejemplos concretos para desarrollar cada una de estas categorías mediante actividades en el aula dirigidos a cursos de STEM en la educación superior. Es un modelo muy prometedor, que podría adaptarse a educación secundaria pero más difícilmente a Educación Primaria con lo que se descarta su utilización en esta tesis.

Uno de los marcos de PC más citados en la literatura y más utilizados empíricamente es el marco tridimensional (3D). Este marco surgió a partir de los estudios de Brennan y Resnick (2012) [46] sobre las actividades diseñadas por jóvenes programadores de todo el mundo utilizando Scratch, que es el lenguaje de programación basado en bloques visuales más popular utilizado en Educación Primaria como herramienta de aprendizaje y como instrumento para el diseño de actividades de PC [4], [59]. El marco 3D se basa en el pensamiento de visualización y el pensamiento de proceso, hace hincapié en el atributo social y colaborativo del PC y tiene una fuerte operatividad en las prácticas educativas [60]. Este marco clasifica el PC según tres dimensiones (Tabla 3): 1) **conceptos computacionales** (conceptos que utilizan los programadores); 2) **prácticas computacionales** (prácticas de resolución de problemas que se necesitan/producen en el proceso de programación); y 3) **perspectivas computacionales** (perspectivas que se forman los diseñadores sobre sí mismos y el mundo que les rodea). Este marco es adecuado para considerar el PC en entornos que utilizan la programación informática y en Educación Primaria [61], por lo que parece adecuado como punto de partida en esta tesis.

Tabla 3. Dimensiones clave del marco 3D de PC de Brennan y Resnick [46].

Conceptos computacionales	Secuencias	Instrucciones para que el ordenador ejecute comportamientos.
	Bucles	Repetir la misma instrucción un número determinado de veces.
	Paralelismo	Concurrencia de múltiples instrucciones.
	Eventos	Desencadenantes de determinadas acciones para crear entornos interactivos.
	Condicionales	Restricciones en la ejecución de instrucciones, que permiten diferentes resultados.
	Operadores	Operaciones matemáticas y de cadenas.
	Datos	Almacenamiento, recuperación y actualización de datos.
Prácticas computacionales	Ser incremental e iterativo	Procesos iterativos para diseñar e implementar soluciones, paso a paso.
	Pruebas y depuración	Procesos de prueba y error para comprobar y eliminar los fallos de funcionamiento que lo justifiquen.
	Reutilización y remezcla	Construir instrucciones reutilizables; construir nuevos productos sobre el trabajo de otros.
	Abstracción y modularidad	Modelización de sistemas complejos con elementos básicos.
Perspectivas computacionales	Expresar	Percepción de la computación como forma de expresión y creación.
	Conectar	Percepción de la computación como una forma de interactuar y trabajar con otros.
	Cuestionar	Plantear preguntas y utilizar la tecnología para resolver problemas de la vida real.

Una vez exploradas las definiciones del PC y con un marco de referencia educativo adecuado, en el siguiente apartado se explorará la necesidad de desarrollar el PC en edades tempranas y las iniciativas tanto internacionales como nacionales que se están llevando a cabo en este sentido.

2.2. PC en etapas tempranas

Hoy en día, el PC es esencial para los alumnos que estudian materias relacionadas con la informática, pero también es crucial para los alumnos de otras disciplinas o áreas. En un mundo altamente tecnificado y en constante cambio, es necesario que los alumnos sean capaces de pensar de forma crítica y resolver problemas complejos para asegurarse de que tienen las herramientas necesarias para enfrentarse de forma productiva al mundo en el que se espera que vivan [62]. En la etapa de aprendizaje temprano, *«además de la lectura, la escritura y la aritmética, el pensamiento computacional debe añadirse a la capacidad de análisis de cada niño»* [43]. A partir de la revisión de la literatura, se extrae la conclusión de que el desarrollo del PC en etapas tempranas es un tema importante en el currículo educativo de muchos países. Algunos de ellos consideran el PC como un programa nacional o han preparado libros de texto y nuevos contenidos de enseñanza [4]. De esta forma, la inclusión del PC en la educación reglada ha hecho grandes progresos en la última década, comenzando incluso en Educación Infantil en algunos países. Se está formando a los niños en PC, para después aplicar habilidades relacionadas con este desarrollo, como pensamiento autónomo y resolución de problemas, a diferentes disciplinas o a la vida cotidiana. Sin embargo, todavía hay dudas sobre cómo enseñar el PC, cuándo empezar o qué enseñar [4], particularmente, hay poca investigación sobre cómo debe enseñarse el PC en las etapas de infantil y en el primer ciclo de Educación Primaria [63], y como indican Hsu *et al.* [4], la formación debe adaptarse en su contenido y diseñarse con diferentes estrategias de enseñanza adecuadas a cada edad a la que vaya dirigida.

En multitud de países se han tomado iniciativas para incorporar la enseñanza de la programación y el desarrollo del PC a nivel formal en sus currículos escolares, como por ejemplo EEUU, Suiza, Estonia, Finlandia, Singapur, Israel o el Reino Unido [64], y en algunos casos, se comienza con este desarrollo desde Educación Infantil, como por ejemplo en Australia o Japón [65].

En 2006, el Parlamento Europeo ya establecía la competencia digital como una de las ocho competencias clave que cualquier joven debe haber desarrollado al finalizar la enseñanza obligatoria para poder incorporarse a la vida adulta de manera satisfactoria y ser capaz de desarrollar un aprendizaje permanente a lo largo de la vida. En 2012, la Comisión Europea presentó *Replantear la Educación* en donde se destaca la importancia de formar en las competencias

necesarias en la sociedad actual del siglo XXI y entornos futuros, la necesidad de que la tecnología se aproveche plenamente y se integre de forma eficaz en los centros formativos, mejorar también el acceso a la educación a través de recursos educativos abiertos y las oportunidades sin precedentes que los nuevos medios ofrecen para la colaboración profesional, la resolución de problemas y la mejora de la calidad y equidad de la educación. La competencia digital es un prerrequisito para que los estudiantes de todas las edades puedan beneficiarse por completo de las nuevas posibilidades que ofrece la tecnología para un aprendizaje más eficaz, motivador e inclusivo.

Más adelante, en 2018, la Comisión Europea estableció como objetivo clave el incluir el aprendizaje de la programación en los colegios europeos en su Plan de Acción de la Educación Digital. En la actualización de 2021-2027 de este plan, se refuerza la idea y se incluye la programación en las competencias necesarias para adaptarse a la sociedad digital. Este plan incluye como objetivos y en el ámbito prioritario, entre otros: *Mejorar las competencias y capacidades digitales para la transformación digital*, que requiere: *capacidades y competencias digitales básicas desde una edad temprana y educación informática y buen conocimiento y comprensión de las tecnologías intensivas en datos tales como la inteligencia artificial* [66]. Además, la Comisión Europa también publica el *Marco europeo de competencias digitales* DIGCOMP [67], para el desarrollo de la competencia digital de los ciudadanos en Europa. El marco proporciona la descripción detallada de todas las habilidades necesarias para ser competente en entornos digitales y las describe en términos de conocimientos, habilidades y actitudes y aporta los niveles dentro de cada competencia.

A nivel de iniciativas no oficiales, la organización Code.org [68], es uno de los referentes mundiales en cuanto a la enseñanza de la programación. Esta organización sin ánimo de lucro, apoyada por donantes como Amazon, Facebook, Google o Microsoft, se dedica a ampliar el acceso a las ciencias de la computación en las escuelas y aumentar la participación de las mujeres jóvenes y estudiantes de otros grupos no suficientemente representados. Plantean que la oportunidad de aprender programación debe ser tan accesible como el aprendizaje de otras materias ampliamente trabajadas en los currículos actuales, como matemáticas o química. Además, organizan una campaña anual: *La Hora del Código* [69], un movimiento global en el que participan más de 180 países y 200.000 educadores.

En España, La Sociedad Científica Informática (SCIE), con el apoyo del Consejo de Decanos de Escuelas de Informática (CDDII) creó en septiembre de 2017 un grupo de trabajo formado por expertos en informática y en educación informática con el objetivo de elaborar un informe con recomendaciones específicas sobre la inclusión del aprendizaje de la informática en las etapas de Educación Primaria y Educación Secundaria [70]. El informe fue emitido en

julio de 2018 y recomienda, de acuerdo con las leyes educativas españolas, establecer una materia denominada "Informática". Esta materia se implantaría preferentemente como una asignatura obligatoria ofertada desde la etapa de Educación Primaria. Los contenidos de la asignatura comprenden seis áreas: programación, ordenadores y sistemas operativos, redes e Internet, datos, contenidos digitales, y seguridad. Incluye cuestiones tanto de alfabetización digital como de informática como disciplina, con contenidos de alfabetización digital basados en el marco DIGCOMP [67].

En el año 2018 la competencia digital se estableció como una de las ocho competencias clave que cualquier joven debe haber desarrollado al finalizar la enseñanza obligatoria para poder incorporarse a la vida adulta de manera satisfactoria y ser capaz de desarrollar un aprendizaje permanente a lo largo de la vida. La enseñanza de la programación a nivel nacional a partir de educación secundaria forma parte de la asignatura optativa de Bachillerato: *Tecnologías de la Información y la Comunicación I*, y se concreta de forma diferente en cada Comunidad Autónoma, así, por ejemplo, en Madrid en 1º, 2º y 3º de la ESO; en Cataluña se incluye en segundo ciclo de la ESO; en Castilla y León mediante una asignatura optativa de 3º y 4º de la ESO [64].

Sin embargo, la enseñanza de la programación no se incluye en Educación Primaria en la legislación a nivel nacional, sino que solamente se hace referencia a las TIC desde el prisma de la competencia digital o creación de contenidos. Sin embargo, en los Reales Decretos de algunas comunidades autónomas, de forma heterogénea, se incluyen contenidos de programación. La Comunidad de Madrid es la única que incluye la asignatura nueva: *Tecnología y recursos digitales para la mejora del aprendizaje* para toda la etapa de Educación Primaria. El resto de las Comunidades solamente incluyen la programación de manera transversal. Por ejemplo, Cataluña establece en el R.D. 119/2015, de 23 de junio, que deben incluirse *contenidos de programación y pensamiento computacional en las áreas de matemáticas y conocimiento del medio*. En la Comunidad Foral de Navarra, la enseñanza de la programación está integrada en las materias de Matemáticas de 4º y 5º de primaria, aunque también establece en su normativa que los centros educativos podrán comenzar itinerarios desde cualquier curso de primaria [71]. En Andalucía, en 2019 se incluyó media hora de robótica a la semana en 5º y 6º de primaria para trabajar el pensamiento matemático. En el resto de los territorios, se ha incluido la enseñanza de la programación en las aulas de primaria de manera aislada y mediante actuaciones asociadas a proyectos promovidos por instituciones públicas y, en su mayor parte, privadas. Por ejemplo, el proyecto de innovación TIC-STEAM [72], en Castilla y León, en el que se trabajan las competencias STEAM (Ciencias, Tecnología, Ingeniería, Artes y Matemáticas), mediante la aplicación de técnicas de programación y robótica, utilizando como medio transmisor las tecnologías de la información y la comunicación, facilitando al profesorado los medios formativos de apoyo que sean neces-

rios para su posterior aplicación en el aula.

La Ley Orgánica de Educación (LOMLOE) 3/2020 por la que se modifica la Ley de Orgánica 2/2006 entró en vigor el 19 de enero de 2021. Actualmente, están publicados, para debate público, los proyectos de Real Decreto para la reforma del currículo en el marco de la LOMLOE. En estos proyectos, se da importancia **al aprendizaje de la programación y el desarrollo del PC desde la etapa de Educación Infantil**, lo que indica la tendencia a incorporar estos elementos en la educación formal que redundará positivamente en los alumnos y, en un futuro, en la sociedad en su conjunto.

Desde el proyecto de reforma del currículo, se destaca, como el elemento más perturbador que motiva esta reforma, la incertidumbre ante el futuro que provoca el cambio a gran velocidad de los procesos económicos, sociales, tecnológicos, políticos y culturales. En este contexto, los alumnos deben incorporarse a la sociedad como ciudadanos activos en contextos temporales, espaciales y digitales llenos de incertidumbre. Por ello, se apuesta por una nueva propuesta curricular competencial de aprendizaje por disciplinas, y definiendo aprendizajes esenciales y otros específicos en los que se debe hacer hincapié en el uso de **tecnologías de la información y las comunicaciones, y el fomento del espíritu crítico y científico**, entre otras cuestiones que el currículo debe integrar desde un enfoque inclusivo.

En concreto, en Educación Infantil se incide en la competencia digital, iniciando en esta etapa, el proceso de alfabetización digital que conlleva, entre otros, el acceso a la información, la comunicación y la creación de contenidos a través de medios digitales, así como el uso saludable y responsable de herramientas digitales. Además, se promueve el uso e integración de estas herramientas en las actividades, experiencias y materiales del aula, ya que esto puede contribuir a aumentar la motivación, la comprensión y el progreso en la adquisición de aprendizajes de niños y niñas. Esta competencia digital del proyecto engloba habilidades de PC.

Además, hay diversas competencias específicas que incluyen explícitamente el PC como habilidad a desarrollar, como, por ejemplo, en el segundo ciclo de **Educación Primaria** las competencias: 2.4. *Utilizar diferentes estrategias para la toma de decisiones con progresiva autonomía, afrontando el proceso de creación de soluciones originales en respuesta a los retos que se le planteen*; 2.5. *Programar secuencias de acciones o instrucciones para la resolución de tareas analógicas y digitales, **desarrollando habilidades básicas de pensamiento computacional***; y 2.6. *Participar en proyectos colaborativos compartiendo y valorando opiniones propias y ajenas, expresando conclusiones personales a partir de ellas*. Como puede verse, se incluye ya explícitamente el PC como habilidad a desarrollar.

Se plantea en el proyecto desarrollar estas habilidades de pensamiento computacional de manera progresiva, a través de procesos de observación y manipulación de objetos, es decir, mediante PC desenchufado (ver apartado 2.4.3), para iniciarse en la interpretación del entorno y responder de forma creativa a las situaciones y retos que se plantean. El alumno deberá encontrar soluciones o alternativas originales y creativas a diferentes cuestiones, retos o situaciones, mediante la aplicación de procesos inicialmente sencillos y manipulativos, que progresivamente ganarán en complejidad y requerirán mayor capacidad de abstracción. Dichos procesos son propios **tanto de las destrezas de pensamiento computacional y de diseño como del método científico**, y se aplicarán descomponiendo una tarea en otras más simples, formulando y comprobando hipótesis, explorando e investigando, relacionando conocimientos y planteando ideas o soluciones originales.

En Educación Primaria, además de la competencia matemática y en STEM, también se añade la competencia digital, que incluye explícitamente **la resolución de problemas y el pensamiento computacional y crítico**. Por ejemplo, esta competencia incluye la CD5. *Se inicia en el desarrollo de soluciones digitales sencillas y sostenibles (reutilización de materiales tecnológicos, programación informática por bloques, robótica educativa...) para resolver problemas concretos o retos propuestos de manera creativa, solicitando ayuda en caso necesario.*

En las áreas de conocimiento del medio, natural, social y cultural, se incluye, en el bloque B: Tecnología y digitalización, el *desarrollo de las estrategias propias del pensamiento de diseño y pensamiento computacional para resolver problemas cooperativamente, que den solución a una necesidad concreta*. Asimismo, en las competencias específicas, se incluye *resolver problemas a través de proyectos interdisciplinares, utilizando el pensamiento de diseño y el pensamiento computacional, para generar cooperativamente un producto creativo e innovador que responda a necesidades concretas*. De esta forma, se incluyen las habilidades de PC como saberes básicos y como competencias específicas evaluables en los tres ciclos de Educación Primaria.

También en el área de matemáticas se incide en el pensamiento computacional como competencia esencial para la sociedad actual y la resolución de problemas de la vida diaria, habilidad de PC que hemos llamado "generalización" en esta tesis. Así, el PC se engloba en uno de los 6 saberes básicos que estructuran el sentido matemático y que es **el sentido algebraico y pensamiento computacional**, *que proporciona el lenguaje en el que se comunican las matemáticas. Reconocer patrones y relaciones entre 118 variables, expresar regularidades o modelizar situaciones con expresiones simbólicas son sus características fundamentales*. También se considera una competencia específica el **utilizar el pensamiento computacional** organizando datos, descomponiendo en partes, reconociendo patrones, generalizando e inter-

pretando, modificando y creando algoritmos de forma guiada para modelizar y automatizar situaciones de la vida cotidiana.

En cuanto a las iniciativas no formales, en 2018, en colaboración con las diferentes Comunidades Autónomas y como iniciativa del Ministerio de Educación y Formación Profesional a través del Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, se creó la Escuela de Pensamiento Computacional [73] y el sitio web Programamos [74], donde pueden encontrarse multitud de recursos gratuitos para el aprendizaje de la programación y el desarrollo del PC. Más de 10.000 docentes en España han tenido acceso a los recursos de este proyecto y además ha tenido una gran acogida internacional.

Con respecto a los docentes, se publica el *Marco de Referencia de Competencia Digital Docente* (BOE 191 de 2020) [75] como respuesta al *Marco europeo de competencias digitales DIGCOMP* [67] y al hecho de que la tecnología afecta todos los aspectos de nuestra vida cotidiana y ha cambiado la forma de comunicarnos. Pretende ser un marco común de referencia para estandarizar la competencia digital. Alcanzar un adecuado grado de desarrollo de la competencia digital de los docentes y de los centros educativos es imprescindible para garantizar la adquisición y desarrollo de las competencias clave por parte de los estudiantes y, en último término, de todos los ciudadanos. Uno de sus objetivos es influir para que se produzca un cambio metodológico tanto en el uso de los medios tecnológicos como en los métodos educativos en general. Hasta ese momento, la formación TIC estaba centrada en los aspectos más instrumentales de la tecnología y no tanto en el uso real de los nuevos medios y recursos digitales en el aula o su función en el desarrollo profesional docente, la actualización continua y la generación de comunidades virtuales de aprendizaje y colaboración profesional en el uso de recursos educativos. El marco divide las competencias en cinco áreas: Información y alfabetización Informacional, Comunicación y colaboración, Creación de contenidos digitales (se da importancia a la programación informática, tanto en la dimensión de comprender qué hay detrás de un programa como en entender los principios y la lógica existente en la programación), Seguridad y Resolución de problemas. Cada competencia se divide en tres niveles (básico, intermedio, avanzado) y está definida por diferentes descriptores. Considera que la competencia digital es una de las capacidades clave del aprendizaje y, por ello, una de las cuestiones a las que se debe prestar especial atención, según las Recomendaciones del Consejo Europeo, siendo un elemento indispensable de la alfabetización elemental del siglo XXI.

Como se ha visto, existen esfuerzos a todos los niveles para introducir la enseñanza de la programación y el desarrollo del PC en las aulas, a edades cada vez más tempranas. Sin embargo, sigue habiendo algunas dudas sobre cuál es la mejor manera de enseñar PC, cuándo comenzar dicha instrucción o qué enseñar [76] y la investigación es especialmente escasa sobre cómo

desarrollar las prácticas computacionales en los primeros grados [63]. Desde la literatura, se recomienda introducir el PC mediante distintas actividades para niños entre 3 y 6 años [53], como, por ejemplo:

- 3-4 años en adelante: Se recomiendan la programación tangible mediante actividades en las que se produzcan y ejecuten instrucciones, que en su mayor parte estén vinculadas al propio cuerpo. También se recomienda trabajar con objetos manipulables.
- 4-5 años: De igual forma que en el punto anterior, se recomienda la programación tangible manipulativa, incorporando elementos de programación a través de interfaces naturales y táctiles, por ejemplo, comandos que se puedan arrastrar y soltar, representaciones visuales e instrucciones gráficas.
- 5-6 años: Además de la programación tangible y táctil, se plantea la introducción de instrucciones simples mediante comandos con algunas palabras sencillas.

Se recomienda también mantener un enfoque global, bien fundamentado con un enfoque constructivista y con metodologías de aprendizaje activo, colaborativo, basado en juegos o basado en proyectos. También es posible desarrollar el PC mediante actividades que no requieran dispositivos electrónicos, conocidas como actividades desenchufadas o *unplugged* [77], [78]. En los siguientes apartados exploraremos estos enfoques y metodologías, y los relacionaremos con el marco de referencia tridimensional del PC [46].

2.3. Teorías y estrategias para el desarrollo del PC

Como se ha expuesto, el desarrollo del PC es esencial para adaptarse a la sociedad de hoy en día y es necesario la introducción de esta habilidad desde etapas tempranas. Sin embargo, no hay evidencias claras de cuáles son las estrategias de aprendizaje más adecuadas para este desarrollo. Parece clave, por tanto, el estudio de metodologías de aprendizaje para esta competencia y el desarrollo de nuevas estrategias para el desarrollo integral de los estudiantes. Durante esta última década, la enseñanza constructivista, que se basa en la teoría del constructivismo, ha destacado como un enfoque clave para el aprendizaje de asignaturas de ciencias, así como para el desarrollo del PC [79].

La teoría del constructivismo postula que el aprendizaje es esencialmente activo y los nuevos conocimientos se incorporan a las experiencias previas y a los esquemas mentales preexistentes, diferentes en cada persona, por lo que el aprendizaje es subjetivo y se va modificando según la experiencia individual

[13]-[15]. En el socio-constructivismo [12], por su parte, se da importancia al individuo y sus relaciones con la sociedad ya que se postula que el conocimiento se construye entre las personas a medida que éstas interactúan. Además, se considera que el lenguaje es la herramienta psicológica que más influye en el desarrollo cognitivo. Según esta teoría, las personas cuentan con ciertas habilidades y conocimientos que dominan gracias a la interacción social, pero en un determinado momento pueden llegar a una *zona de desarrollo próximo* (distancia entre lo que los individuos ya saben hacer y lo que aún no logran realizar de forma independiente) en la cual aparecen problemas que deben resolver para seguir desarrollando habilidades cognitivas. Es en este momento cuando otros individuos, como puede ser el profesor, orientan el aprendizaje nuevo que, una vez queda establecido, da lugar a la aparición de una nueva *zona de desarrollo próximo*.

Cuando el constructivismo y el socio-constructivismo se aplican a un contexto educativo, se da especial importancia a las percepciones, emociones y pensamientos del estudiante. Un aprendizaje constructivista debe ser 1) activo: la responsabilidad del resultado del aprendizaje es del alumno y éste se compromete con el proceso de aprendizaje; 2) constructivo: ya que los alumnos aportan ideas nuevas a su conocimiento previo para dar sentido o significado a una discrepancia; 3) colaborativo: se trabaja en grupos de aprendizaje y construcción del conocimiento, se aprovechan las habilidades del resto y el apoyo social; 4) intencional: se intenta conseguir un objetivo cognitivo; 5) conversacional: es un aprendizaje social y dialógico; 6) contextualizado: las actividades de aprendizaje se contextualizan en tareas significativas del mundo real o simulado, por ejemplo mediante un entorno de aprendizaje basado en problemas; 7) reflexivo: los estudiantes reflexionan sobre lo aprendido y los procesos y decisiones implicadas. El profesor, por su parte, es el promotor de actividades y guía al alumno para que éste explore y resuelva los problemas. Así, el estudiante construye su propio aprendizaje, aprende a pensar y razonar, y reflexiona sobre propio proceso de aprendizaje [80].

De esta forma, el aprendizaje constructivista se basa en la construcción del conocimiento cuando se conectan nuevas experiencias con las ideas preexistentes [79]. De manera sencilla, el enfoque constructivista evoca la idea de *aprender haciendo* mientras los estudiantes trabajan en un proyecto concreto, experimentando y descubriendo por ellos mismos. Es un aprendizaje en el que el alumno *construye* su propio conocimiento mientras interactúa con el entorno [81]. El aprendizaje resulta más eficiente desde este enfoque que desde estrategias más tradicionales [82]. Además, esta estrategia impacta directamente en el rendimiento académico, la motivación, el desarrollo personal, la autoevaluación y las habilidades comunicativas del estudiante; y mejora la disposición del estudiante frente al aprendizaje de la informática y por lo tanto, del PC [83]. Además, las TIC proporcionan herramientas para el

desarrollo de actividades de colaboración y cooperación en la enseñanza, facilitando la interacción de los estudiantes desde una perspectiva constructivista vinculada, de manera inexorable, a la teoría socio-constructivista de Vygotsky [12].

En respuesta a las innovaciones tecnológicas y pedagógicas que se han ido produciendo, han surgido también marcos operativos de aprendizaje como el marco de aprendizaje constructivista de Papert [47] en el que se destaca la interacción social de los estudiantes, la cognición y el pensamiento de alto nivel y, por lo tanto, el desarrollo del PC. Otro marco operativo destacado y ampliamente utilizado para el diseño de entornos de aprendizaje constructivistas es la teoría de la actividad de Jonassen [14], en donde se analizan las necesidades, tareas y resultados esperados cuando se diseñan estos entornos.

El constructivismo también puede ser una base para formar metodologías de enseñanza como las metodologías activas, que inciden en la autoconstrucción del conocimiento por parte de los estudiantes y en la responsabilidad de los estudiantes por su propio aprendizaje. Las metodologías activas están recibiendo un creciente interés como enseñanza instruccional centrada en el alumno mediante actividades dirigidas por un profesor, en las que el estudiante afronta problemas y se esfuerza para encontrar las soluciones, como por ejemplo actividades de resolución de problemas, tareas de escritura, discusión en grupo, o actividades de reflexión [84]. Los componentes comunes a estas metodologías activas se pueden resumir de la siguiente forma [85]:

- **El escenario:** Establece el contexto para el problema, caso o proyecto.
- **Trabajo en grupo:** Los estudiantes tienen una responsabilidad con el trabajo eficiente del grupo, además de con el desarrollo de su aprendizaje individual.
- **Solución de problemas:** Los problemas planteados en un entorno de metodologías activas suelen ser complejos y necesitarán en general razonamiento e indagación.
- **Descubrimiento de nuevos conocimientos:** Los estudiantes deben determinar qué saben y qué necesitan saber para poder continuar.
- **Conexión con el mundo real:** Se puede extrapolar lo aprendido al mundo real.

Basadas en el constructivismo, durante la última década se han ensayado diferentes estrategias de aprendizaje como metodologías activas de enseñanza para ayudar a los estudiantes a aprender a programar y así mejorar

desarrollar el PC [61]. Las investigaciones más recientes destacan, tanto por su efecto positivo en el aprendizaje como por ser las más empleadas para el desarrollo del PC, las siguientes estrategias: el aprendizaje basado en problemas, el aprendizaje colaborativo, el aprendizaje basado en proyectos y el aprendizaje basado en juegos [4]. Se muestra en la Tabla 4 la definición de cada una de estas estrategias.

Destacamos en este punto el aprendizaje colaborativo, ya que suele ser una estrategia transversal al resto y resulta fundamental para desarrollar las perspectivas computacionales del PC según el marco 3D de PC de Brennan y Resnick [46]. Dillenbourg [86] establece cuatro condiciones para establecer un contexto colaborativo activo en el que es probable que se activen los mecanismos de aprendizaje: establecer las condiciones iniciales, especificar el tipo de colaboración con un escenario basado en roles, estructurar y guiar las interacciones productivas mediante reglas de interacción, y supervisar y regular las interacciones. El aprendizaje colaborativo fomenta las habilidades de pensamiento crítico de forma mucho más eficaz que los entornos individuales [85]. Esto da lugar a mayores niveles de rendimiento académico y mayor productividad, así como a una mayor salud psicológica, competencia social y autoestima que el aprendizaje individual [87]. Además, los equipos cooperativos alcanzan niveles de pensamiento más altos y retienen la información durante más tiempo que los alumnos que trabajan individualmente [88], [89].

Tabla 4. Estrategias de aprendizaje más utilizadas y con más impacto en el aprendizaje del PC

Estrategia de aprendizaje	Definición
Aprendizaje basado en problemas	Los problemas son el estímulo y el foco de la actividad de los estudiantes: se les presenta un escenario para la resolución de un problema y deben explorar la solución por sí mismos, estableciendo sus propios objetivos de aprendizaje [90].
Aprendizaje colaborativo	En este tipo de aprendizaje se colabora en un proyecto común. Se contemplan los siguientes elementos: situación, interacción, procesos y efectos. Las relaciones entre estos cuatro elementos son fundamentales, ya que los miembros del grupo realizan una tarea o situación juntos, negocian y comparten opiniones y conocimientos para resolver el problema [86], [91].

Aprendizaje basado en proyectos	Los alumnos participan en la investigación de una tarea compleja basada en una pregunta desafiante o un problema. En este marco, los alumnos buscan soluciones al problema mediante preguntas, debates, diseño de planes, toma de decisiones o investigación [92].
Aprendizaje basado en juegos	Es un entorno de juego o un escenario de resolución de problemas mediante la creación de un reto, para que los estudiantes busquen soluciones utilizando mecánicas de juego, con una sensación de logro, a la vez que mejoran la adquisición de conocimientos y habilidades [9]-[11].

En esta tesis se va a plantear el aprendizaje del PC mediante la estrategia de aprendizaje basado en problemas como enfoque principal; la estrategia basada en juegos como vehículo de andamiaje para guiar y facilitar el proceso de resolución a través de la mecánica del juego; y la estrategia colaborativa como instrumento transversal para mejorar el proceso de aprendizaje. Es importante no confundir la estrategia de aprendizaje basado en juegos (*Game Based learning* o GBL) con la gamificación. Esta última aplica los sistemas de puntuación-recompensa-objetivo propios de los juegos a la metodología del aula, mientras que la metodología GBL consiste en la utilización directa de juegos como vehículos y herramientas de apoyo al aprendizaje, la asimilación o la evaluación de conocimientos.

Mediante la metodología GBL se trabaja la alfabetización digital y el desarrollo del PC. Algunas de sus ventajas son la motivación del alumno, ya que resulta divertido y dinamizador; el desarrollo del pensamiento crítico, la autosuficiencia y la resolución de problemas, ya que durante el juego el estudiante debe reflexionar y tomar decisiones adecuadas, depurar errores y reponerse frente a las derrotas; el aprendizaje activo, experimentando mediante la prueba y error, estableciendo conexiones entre los conocimientos previos y los nuevos, y tomando decisiones para mejorar; el alumno regula su propio aprendizaje, como propugna el constructivismo, ya que se ofrece una retroalimentación instantánea, el estudiante es consciente de su nivel de conocimientos y habilidades, y puede guiar su propio aprendizaje; el docente puede ser consciente en tiempo real de las fortalezas y debilidades de sus alumnos, pudiendo intervenir en consecuencia en fases iniciales, guiando y adaptando el aprendizaje; se potencia la creatividad y la imaginación, ya que existe una componente de improvisación y capacidad de imaginar soluciones a los retos, con lo que el

estudiante puede ampliar su mente y percepción del mundo; y puede beneficiarse del aprendizaje colaborativo si el juego permite jugadores simultáneos que cooperen para un fin común.

Con el objetivo de **relacionar las estrategias de aprendizaje con el marco de referencia educativo para el desarrollo del PC** (apartado 2.1.3), se ha realizado un análisis de la literatura. Se ha detectado que la mayoría de las investigaciones y actuaciones para el desarrollo del PC en el aula, se centran en la dimensión *conceptos computacionales* del marco 3D de PC de Brennan y Resnick [46], [61], [93] y, solamente en algunas investigaciones muy recientes, sobre el aprendizaje y la evaluación de las dimensiones *prácticas computacionales* y *perspectivas computacionales* [93]–[97]. Esto nos lleva a la necesidad de más investigación y de prestar especial atención a estas dos últimas dimensiones, especialmente más allá del entorno de Scratch [98] en el cual se ubican la mayor parte de los casos de estudio en las investigaciones, ya que el PC es un conjunto interdisciplinario de habilidades y es necesario que existan métodos operativos de aprendizaje y evaluación del PC a través de diversas estrategias de aprendizaje [61], [96]. Las prácticas y perspectivas computacionales son incluso más pertinentes en las etapas de Educación Infantil y Educación Primaria, ya que como se ha visto, es importante desarrollar el PC (por ejemplo, aprendiendo a programar) desde etapas tempranas, para dotar a los estudiantes de habilidades de resolución de problemas que puedan transferir posteriormente a dominios no relacionados con la programación [41], y estas dos últimas dimensiones del marco 3D son las que nos permiten trabajar esta componente del PC que, en nuestra descomposición propuesta (apartado 2.1.2), hemos llamado *generalización*. A este respecto, en el capítulo 3 de esta tesis se han establecido **hipótesis sobre la posible relación del aprendizaje que se obtiene de las estrategias de aprendizaje más relevantes con las dimensiones del marco 3D**.

2.4. Entornos de aprendizaje

Como se ha visto, la programación se utiliza con frecuencia para desarrollar el PC, ya que, como se ha visto, se desarrollan muchas de las componentes que se identifican con el PC. Programar requiere analizar el problema y descomponerlo en otros más sencillos y resolverlo mediante pasos ordenados, para ello se requiere análisis, abstracción, descomposición, algoritmia, y habilidad para la depuración de errores. Uno de los objetivos docentes a nivel internacional es enseñar programación de forma efectiva y atractiva a estudiantes de Educación Infantil y primaria, utilizando metodologías activas basadas en el constructivismo y empleando estrategias de aprendizaje adecuadas. Existen numerosas experiencias en esta dirección en las cuales se utilizan diferentes contextos y entornos de aprendizaje. Uno de los principales problemas que surgen en etapas tempranas es que las habilidades de lectura y escritura de

estos estudiantes pueden no estar aun suficientemente establecidas como para poder apoyar el aprendizaje con textos escritos y, sin embargo, el desarrollo de esta competencia debe comenzar muy pronto, de forma similar al aprendizaje de una lengua no materna [99], [100].

El software educativo y las aplicaciones multimedia pueden ayudar a mejorar el aprendizaje de la programación y el desarrollo del PC en estas etapas [101], al igual que otros entornos e instrumentos como los robots programables, juegos, videojuegos, realidad virtual, realidad aumentada, o comics. Silva *et al.* (2021) [102], proponen una taxonomía de entornos para el desarrollo del PC en etapas tempranas diferenciando entre basados en bloques; robots; combinación de bloques con robots; y prototipos. Como subcategorías diferencian entre tangible (se puede manipular físicamente) o web (no-tangible). Además, también indican que las palabras que más aparecen en la literatura sobre entornos para el desarrollo del PC son: computacional, pensamiento, programar, robótica, enseñanza, y codificar. En nuestro caso (Tabla 5), se ha diferenciado entre entornos que utilizan: robótica (la programación del robot puede realizarse de forma tangible o mediante una aplicación); aplicaciones que normalmente se basan en juegos (debe usarse ordenador o tablet); aplicación en la que es necesaria una parte tangible; y tangible, aunque este último caso no se recoge completamente en la tabla porque se considera otra categoría completa (PC desenchufado). Esta tabla se encuentra ordenada por el rango de edad recomendado, de etapas más tempranas a niños más mayores.

Tabla 5. Ejemplos de aplicaciones e instrumentos para enseñar a programar y desarrollar el PC

Aplicación	Sitio web	Rango de edad (años)	Instrumento
Cubetto	primotoys.com	3 +	Robótica
Bee-bot	bee-bot.us	3 +	Robótica
Code a pillar	fisher-price.com	3 a 8	Robótica
Robot Turtles	robotturtles.com	4 +	Juego de mesa
Hello Ruby	helloruby.com	4 +	Cuento
LightbotJr	lightbot.com	4 +	Aplicación
KIBO Robots	qiibo.com	4 a 7	Robótica
Kodable	kodable.com	4 a 11	Aplicación
Minecraft Education	education.minecraft.net	5 +	Aplicación

Aplicación	Sitio web	Rango de edad (años)	Instrumento
Scratch Junior	scratchjr.org	5 a 7	Aplicación
Roamer	roamer-educational-robot.com	5 a 10	Robótica
Osmo Coding	playosmo.com	5 a 12	App + tangible
Osmo Music Symbols & Notes	playosmo.com	5 a 12	App + tangible
True-True	prodel.es	6 +	Robótica
Blue-Bot	blue-bot.us	6 +	Robótica
Scratch	scratch.mit.edu	8 +	Aplicación
Agent Sheets	agentsheets.com	8 +	Aplicación
Code Monkey Island	codemonkeyplanet.com	8 +	Juego de mesa
Code Monkey	playcodemonkey.com	9 +	Aplicación
Code Combat	codecombat.com	9 +	Aplicación
Cargobot	twolivesleft.com/CargoBot	9 a 11	Aplicación
Swift Playgrounds	apple.com/swift/playgrounds	10 +	Aplicación
Hopscotch	gethopscotch.com	10 +	Aplicación
Flexbot	flexbot.es	10 +	Robótica
Lego Mindstorms	lego.com	10 +	Robótica
Colobot	ceebot.com/colobot	10 a 15	Aplicación
Mimo	getmimo.com	12 +	Aplicación
Py – Learn to Code	getpy.com	12 +	Aplicación
Elegoo	elegoo.com	12 +	Robótica
Makeblock ultimate	makeblock.es	12 +	Robótica
Robocode	robocode.sourceforge.net	14 a 18	Aplicación
SpaceChem	zachtronics.com/spacechem	15 +	Aplicación

2.4.1. Robótica

El uso de robots educativos en los colegios está cobrando una gran importancia como un modo de acercar la tecnología y la programación a los niños de manera práctica, ya que el resultado de sus creaciones o programas se traslada al mundo real de forma inmediata y tangible, incluso usando simuladores [103]. Mediante la robótica, se facilita el trabajo en equipo, se desarrolla la comprensión conceptual, se mejora el pensamiento crítico y se promueve el aprendizaje de orden superior en los ámbitos de las matemáticas y las ciencias [83]. Del mismo modo, la robótica proporciona a los alumnos experiencias táctiles para resolver problemas a través del PC [104], [105]. Los alumnos tienen que identificar el problema u objetivo general del robot, descomponerlo, por ejemplo, calculando el número de pasos o subobjetivos para lograr el objetivo. Para ello, los alumnos desarrollan algoritmos para que el robot pueda seguir las instrucciones y actuar en consecuencia. Cuando el robot no actúa como se espera, entra en juego la depuración que requiere procesos iterativos de pruebas y modificaciones sistemáticas.

Mediante el uso de robots dotados de sensores que actúan como nuestros sentidos de la vista, el oído o el tacto; y actuadores que permiten al robot realizar diversas acciones por ejemplo de movimiento o sonido; se pueden realizar experiencias de aprendizaje siguiendo las bases de la programación de tipo *sensor-razonamiento-actuador* o *SenseReason-Act* (SRA). La programación SRA y, sobre todo, si se utiliza un entorno de programación visual que complementa al robot, permite un desarrollo significativo del PC, sobre todo en relación a la resolución de tareas o problemas complejos [54], [83], [106]–[108]. Numerosos investigadores han destacado la ventaja de utilizar la robótica como medio para un desarrollo significativo del PC desde un enfoque constructivista, por ejemplo, Atmatzidou *et al.* (2016) destacan la facilidad para superar dificultades iniciales gracias a la robótica y, en consecuencia, el desarrollo de las habilidades de PC de los estudiantes [55]. Bers *et al.* (2014) subrayan la gran implicación y motivación de los alumnos, el aprendizaje de conceptos de programación de habilidades de resolución de problemas complejos que se logra en alumnos de Educación Infantil con un enfoque constructivista de aprendizaje y el uso de la robótica en las aulas, aunque resaltan la necesidad de adecuar las estrategias de aprendizaje y los dispositivos que se utilizan en estas etapas tan tempranas para poder apreciar un desarrollo del PC [54].

En este sentido, hay una gran oferta de dispositivos para el desarrollo del PC a través de la robótica educativa. En la Tabla 5 se muestran algunos de ellos orientados a las etapas educativas de infantil, primaria y secundaria. En el caso de infantil y primaria, los robots suelen funcionar de forma independiente, pudiendo programarlos físicamente, sin usar una aplicación en un ordenador o tablet para ello. Por ejemplo, Cubetto, diseñado por Primo Toys

y basado en la metodología Montessori [109], se puede utilizar ya en la etapa de Educación Infantil, e incluye fichas tangibles que se insertan en diferentes ranuras de un tablero y que se utilizan para programar una secuencia de acciones para el robot, representado por un cubo de madera, que ejecuta la secuencia programada (véase la parte izquierda de la Figura 7). Este tipo de robots para niños pequeños también pueden incluir botones en el propio robot para programar la secuencia de movimientos como por ejemplo en el caso de Bee-bot y Blue-bot (véase la parte central y derecha de la Figura 7).

Cabe destacar también el kit de robótica KIBO, para niños de 4 a 7 años, desarrollado el Grupo de Investigación DevTech de la Universidad de Tufts, que propone involucrar a los niños pequeños tanto en la construcción con piezas como en la programación, además de desarrollar el PC. Melin *et al.* [110] presentan un caso de uso exitoso de esta herramienta con niños de 3 años.

Figura 7. Ejemplos de robots educativos: Cubetto (fuente: primotoys.com), Bee-bot y Blue-bot (fuente: bee-bot.us y blue-bot.us).



Cuando las habilidades de lecto-escritura estén ya asentadas, el siguiente paso a los entornos visuales de programación por bloques, es el uso del pseudocódigo [111], que utiliza las estructuras de un lenguaje de programación real, pero está diseñado para que se pueda leer por una persona en un lenguaje natural (mejor si es la lengua materna del estudiante), en lugar de por un ordenador. Más adelante, pueden utilizarse entornos de simulación visual del código programado [112]. De esta forma, se va alcanzando gradualmente la competencia en programación al mismo tiempo que se van desarrollando las habilidades asociadas de PC.

2.4.2. Aplicaciones y juegos

En cuanto a las aplicaciones, normalmente juegos de tipo puzzle, en los que es necesaria la utilización de un dispositivo digital como un ordenador o una tablet, también se trabaja el PC ya que, al igual que con la robótica, el juego

implica varios objetivos, a menudo puzzles, que los jugadores deben resolver. Para tener éxito, los jugadores emplean varias habilidades o componentes del PC: descomposición, para elaborar estrategias de resolución de problemas; depuración de errores ya que, si un plan falla, deben poder ser capaces de depurarlo y al probar sistemáticamente varios planes, los jugadores encuentran la estrategia más eficaz para superar los retos del juego; y la generalización cuando los jugadores adoptan las estrategias utilizadas anteriormente para resolver nuevos problemas. Muchas de estas aplicaciones o juegos sirven además para aprender a programar con lo que se potencia el aprendizaje del PC añadiendo los beneficios de aprender programación con las habilidades que se desarrollan mientras se juega.

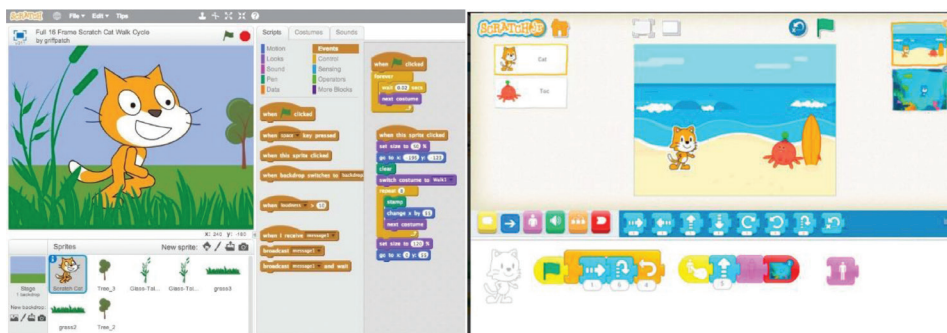
Los entornos de software educativo basados en bloques visuales de programación son muy populares actualmente para el desarrollo del PC en etapas tempranas (véase la Tabla 5). Los estudiantes emplean tablets u ordenadores y utilizan elementos visuales (bloques) que se arrastran y sueltan para construir programas de código ejecutable, sin necesidad de escribir o leer en un lenguaje de programación concreto o incluso solamente con gráficos auto explicativos sin ningún texto. Uno de los entornos basados en bloques más utilizados para el desarrollo y la evaluación sistemática del PC a partir de los 8 años, es Scratch [98], que incluye una componente creativa, ya que permite a los estudiantes crear sus propios proyectos en forma de aplicaciones o juegos autónomos que pueden jugarse por otras personas posteriormente. Brennan y Resnick [46] afirman en su investigación que la realización creativa de proyectos con Scratch puede suponer un aprendizaje significativo a largo plazo, además, la evaluación del PC a través del marco 3D se fundamenta inicialmente en el análisis de las habilidades de los estudiantes a lo largo del tiempo mientras utilizaban Scratch. Se analizaron tanto la evolución cuantitativa de los proyectos que creaban los estudiantes como sus prácticas, motivaciones y actitudes mediante observación directa y entrevistas cualitativas. Los resultados revelaron una progresión muy positiva de desarrollo de habilidades de PC en todas las dimensiones del marco 3D. Scratch ha sido utilizado como elemento para evaluar el desarrollo del PC por numerosos investigadores [1]. Por ejemplo, Seiter y Foreman en 2013 [113] confirman el desarrollo del PC en alumnos de Educación Primaria mediante el análisis de los comandos, conceptos, variables y patrones en proyectos realizados con Scratch. Grover *et al.* en 2015 [114] diseñaron una serie de instrumentos de evaluación del PC que incluyen el análisis de fragmentos de código de proyectos realizados con Scratch.

La herramienta AgentSheets (véase la Tabla 5) combina, al igual que Scratch, el modo creativo en el que el estudiante puede crear sus propios mundos, con la programación visual por bloques. Esta herramienta desarrolla las habilidades en informática y su curva de aprendizaje es suave, con lo que estudiantes

inexpertos pueden fácilmente iniciarse en la programación de la misma forma que estudiantes más avanzados pueden crear juegos complejos. Basawapatna *et al.* [115] constataron que los estudiantes podían reconocer mejor patrones de PC después de interactuar con el juego.

Para niños más pequeños existen entornos basados en bloques visuales para su uso desde los 4 años (véase la Tabla 5), como por ejemplo Kodable. Pero es, de nuevo, Scratch, con su versión para niños de 5 a 7 años, ScratchJr (véase la Figura 8), el más utilizado para el desarrollo del PC en el primer ciclo de Educación Primaria [57]. Mantiene los elementos de programación creativa de Scratch que permiten a los niños crear fácilmente historias y juegos cortos, pero mediante una interfaz completamente simbólica en la que no hay texto escrito. Se pueden añadir diferentes personajes a una escena en la que hay una cuadrícula para facilitar en cálculo de distancias. Para dotar a los personajes de comportamientos se pueden combinar sencillos bloques visuales de instrucciones que están ideados para su uso en tablets. En ScratchJr las instrucciones se ordenan de izquierda a derecha en lugar del enfoque de arriba a abajo utilizado en Scratch. Rose *et al.* (2017) resaltan ScratchJr como un entorno adecuado (frente a otros como LightbotJr) para el desarrollo del PC gracias a su diseño sencillo, pero con un techo alto de aprendizaje, de forma que estudiantes más avanzados puedan también seguir aprendiendo y creando proyectos más difíciles [57]. En la Tabla 6 se incluyen los conceptos de PC que se utilizan en ScratchJr según la descomposición propuesta por estos autores (ver apartado 2.1.2).

Figura 8. Interfaces de las herramientas Scratch (izquierda) y ScratchJr (derecha).



Otras aplicaciones se basan en juegos serios o puzzles que, para completar, es necesario realizar alguna tarea de programación, codificación o estrategia computacional. Los juegos serios o educativos utilizan la metodología GBL y son un enfoque prometedor para introducir la programación y el PC [116]. La mayoría de estos juegos utilizan la programación por bloques al igual que Scratch [117]-[119], aunque también hay casos en los que debe

introducirse directamente código en texto o pseudocódigo para poder progresar en el juego, como CodeCombat [120], dirigidos al último ciclo de Educación Primaria.

Tabla 6. Conceptos de PC que se utilizan en ScratchJr. según Rose et al. [57].

Componentes del PC	ScratchJr.
Abstracción y generalización	Comprensión de la cuadrícula y del movimiento de los personajes. Identificación de comportamientos comunes mediante instrucciones y bloques de instrucciones.
Algoritmia y procedimientos	Secuenciación de instrucciones para crear algoritmos. Utilizar procedimientos para repetir instrucciones comunes.
Recogida, análisis y representación de datos	Contar el movimiento necesario utilizando la cuadrícula.
Descomposición	Aplicación de comportamientos a diferentes personajes. Tener varios bloques de instrucciones en un mismo personaje. Aplicación de comportamientos por pasos.
Paralelismo	Los bloques de instrucciones se ejecutan en paralelo.
Depuración, pruebas y análisis	La instrucción que se está ejecutando en ese momento aparece resaltada. Los programas pueden volver a ejecutarse para comprobar si hay errores. Las instrucciones se pueden pulsar individualmente para comprobar el efecto.
Estructuras de control	Uso de bloques como repetir y esperar para controlar la ejecución. Instrucciones en bucle mediante procedimientos.

Además de aplicaciones y juegos autónomos, hay multitud de iniciativas de gran cobertura a nivel internacional y nacional. En la Tabla 7 se muestran

algunas de ellas. La mayor parte se orientan a toda la etapa de infantil y primaria, y el objetivo es que las ciencias de la computación y la programación sean parte habitual de la educación del mismo modo que lo son la biología, las matemáticas o la lengua. Además de iniciativas dirigidas a colectivos vulnerables sin acceso a oportunidades para el aprendizaje de la programación como CodeNow o ScriptEd, cabe resaltar que algunas de estas iniciativas se enfocan en las niñas, con el objetivo de que haya una representación y aumento de la participación de las mujeres en los campos STEM a través de la exposición temprana a la informática y la tecnología, debido a la brecha de género en este sentido. Por ejemplo, las iniciativas Black Grils CODE, Girls in Tech y Girls Who Code trabajan en este sentido.

Tabla 7. Iniciativas para aprender programación y desarrollar PC a nivel internacional y nacional.

Iniciativa	Sitio web	Ámbito	Plataforma
Bebras	bebras.org	Internacional	Dispositivo
Black Girls CODE (BGC)	blackgirlscode.com	Internacional	Dispositivo
Code Week	codeweek.eu	Internacional	Dispositivo
Code.org	code.org	Internacional	Dispositivo
CodeNow	codenow.org	Internacional	Dispositivo
CoderDojo	coderdojo.com	Internacional	Dispositivo
Coding pirates	codingpirates.dk	Internacional	Dispositivo
Computing at School	computingatschool.org.uk	Internacional	Dispositivo
CSforAll Consortium	csforall.org	Internacional	Dispositivo
CSNYC	csnyc.org	Internacional	Dispositivo
CT en Google for Education	edu.google.com/resources/programs/exploring-computational-thinking	Internacional	Dispositivo
Exploring Computer Science (ECS)	exploringcs.org	Internacional	Dispositivo
Girls in Tech	girlsintech.org	Internacional	Dispositivo
Girls Who Code	girlswhocode.com	Internacional	Dispositivo

Iniciativa	Sitio web	Ámbito	Plataforma
Hour of Code	hourofcode.com/ uk/es	Internacional	Dispositivo
Made with code	madewithcode.com	Internacional	Dispositivo
ScriptEd	scripted.org	Internacional	Dispositivo
Code.Educalab	code.educalab.es	Nacional	Dispositivo
Genios	educagenios.com	Nacional	Dispositivo
Programamos	programamos.es	Nacional	Dispositivo
Computer Science Unplugged	csunplugged.org	Internacional	Desenchufado
Barefoot computing	barefootcomputing.org	Internacional	Desenchufado

Hay también iniciativas en las que la herramienta que se utiliza como vehículo para la enseñanza es una adaptación de videojuego, como por ejemplo como es el caso de Second Life o Minecraft Education, que funciona como una herramienta colaborativa, de aprendizaje y evaluación de la programación, pero también cuenta con multitud de actividades y contenidos educativos orientados a otras áreas como lengua o historia.

Los entornos que utilizan la estrategia basada en juegos GBL potencian el PC y en general las habilidades del siglo XXI, por lo que existe una tendencia a incorporar esta estrategia cada vez más en las aulas, aunque según Qian y Clark [11] en 2016, solamente un 20% de las experiencias en la literatura se orientan a la etapa de Educación Infantil y primaria. Estos autores indican que los juegos educativos suelen basarse en el constructivismo y la teoría del flujo [121] en su mayor parte, y los efectos en el aprendizaje suelen ser muy significativos en un 85% de los casos, así como en el desarrollo de las habilidades del siglo XXI, entre ellas el PC.

Para potenciar el nivel de inmersión en el juego, la estrategia GBL puede incluir realidad extendida (RX), tecnología que está renovando la forma en que las personas experimentan los entornos físicos y virtuales, desde la observación hasta la inmersión. La RX engloba tanto la realidad aumentada (RA) como la virtual (RV) [122]. Por ejemplo, la compañía *universiv.com* plantea recorridos virtuales educativos e inmersivos por escenarios históricos o por el interior del cuerpo humano a través de la RV.

Este tipo de tecnologías permiten desarrollar nuevos tipos de juegos, como los pervasivos, que son aquellos en los que se rompen los límites tradicionales del juego, en términos de dimensiones espaciales, temporales y sociales [123]. Por

ejemplo, el Grupo AffectiveLab de la Universidad de Zaragoza, en el espacio interactivo e inmersivo JUGUEMOS [124], presentó el juego multijugador El viaje fantástico, para trabajar la atención, la planificación y las habilidades sociales, utilizando la RX. En este juego se incide especialmente en la dimensión espacial de los juegos pervasivos, combinando elementos virtuales y físicos, y a través de la interacción multimodal [125]. En el mismo espacio, y a través de la utilización de mesas interactivas que trazan tanto objetos como los dedos de los niños, también se presentó el juego tangible y colaborativo, StarLoop, para el aprendizaje de conceptos de programación como bucles y procedimientos [126].

La investigación con respecto al desarrollo del PC mediante la RV se encuentra aún en un estado inicial dado su carácter reciente. Sukirman *et al.* realizan una revisión profunda de la literatura actual incluyendo el año 2021 y plantean un marco conceptual (VRGACT) para la inclusión de la VR para el desarrollo del PC [127]. Basándose en su revisión, relacionan los elementos de juego y la VR con el PC y el impacto en el aprendizaje como se puede ver en la Tabla 8.

Tabla 8. Relación entre los elementos de juego, la RV, el PC, y el impacto en el aprendizaje [127].

Elementos de juego	Características de RV	Conceptos de PC	Impacto en el aprendizaje
Jugabilidad	Telepresencia	Descomposición	Creatividad
Compromiso	Presencia social	Reconocimiento de patrones	Aprendizaje auto-dirigido / autonomía
Emoción	Representación metafórica	Abstracción	Independencia
Interés	Presencia	Pensamiento/ diseño de algoritmos	Persistencia / mantenimiento
Disfrute	Inmersión	Programación de conjuntos de respuestas	Experiencia de aprendizaje
Personalización	Interacción	Máquina de estado finito	Interés por el aprendizaje
Colaboración	Interacción social	Resolución de problemas	Motivación
Interacción	Abstracción	Pensamiento recursivo Heurística	Conocimiento Comprensión

Elementos de juego	Características de RV	Conceptos de PC	Impacto en el aprendizaje
Diseño visual	Aspectos visuales	Representación de datos Depuración Secuencias, bucles, variables	Motivación Sentido común Pensamiento crítico Pensamiento creativo Razonamiento sistémico

De igual forma, se han desarrollado herramientas de RA para que el aprendizaje de la programación sea más atractiva y fácil, además de desarrollar el PC. Por ejemplo, con Osmo coding (Tabla 5) se desarrollan habilidades lógicas y de resolución de problemas [128]. Aplicaciones que incluyen RA pueden mostrar en el dispositivo del estudiante objetos y elementos en tres dimensiones de forma que se pueden mover y explorar, por ejemplo, con la aplicación Star Walk de Vito Technology para la visualización de las estrellas o con las diversas aplicaciones que ofrece la compañía *onirix.com* para aplicar en diversas asignaturas como, por ejemplo, ciencias naturales.

2.4.3. Pensamiento computacional desenchufado

Existen otras aproximaciones estratégicas para el aprendizaje de la programación y el desarrollo del PC que no se apoyan en ningún dispositivo digital. Así, el PC Desenchufado o *unplugged* (PCD) se refiere, en términos generales, al aprendizaje de conceptos de PC e informática a través de juegos de rol, manipulación de objetos del mundo real (por ejemplo, fichas, tarjetas, bloques de madera) y acciones físicas del cuerpo, entre otros. Tim Bell *et al.* [78] afirma, que aprender de esta manera no consiste simplemente en simular los procesos del ordenador, sino que se trata de proporcionar a los estudiantes la oportunidad de explorar las ideas fundamentales de las ciencias de la computación sin necesitar los conocimientos técnicos necesarios para programar. Está demostrado que este tipo de aprendizaje sin dispositivos digitales potencia el desarrollo del PC en Educación Primaria [129]. Zapata-Ros [77] argumenta, en 2014, que las actividades de PCD deberían incluirse en el currículo desde Educación Infantil, ya que es fundamental que se fomente una práctica formativa del PC desde las primeras etapas de desarrollo, de la misma forma que se habla de prelectura, preescritura o precálculo, es necesario trabajar fases previas a la codificación desde edades tempranas.

Los robots tangibles como bee-bot, blue-bot, Kibo o Cubetto incluidos en la Tabla 5, pueden utilizarse como materiales para actividades de PCD. También existen juegos de mesa diseñados específicamente para desarrollar las habilidades de PC de los jugadores. Por ejemplo, con Robot Turtles, niños de entre

los 3 y los 8 años pueden aprender conceptos de programación atravesando laberintos en un tablero, utilizando para ello los conceptos computacionales de secuencia y condicionales para resolver los problemas que se van encontrando. Estos juegos de mesa no incluyen ningún agente electrónico, sino que incluyen manipuladores de programación para expresar los códigos, normalmente en forma de tarjetas. En Code Monkey Island, jugadores de 8 años en adelante deben mover las figuras por el tablero aplicando conceptos computacionales, como condicionales, bucles, operadores booleanos, y operadores lógicos. En general, los juegos de mesa que no requieren el uso de ningún dispositivo informático son formas económicas de desarrollar el PC que pueden alcanzar espacios de aprendizaje informal como los hogares o los programas extraescolares. Los juegos de mesa suelen implicar varios jugadores, lo que potencia la colaboración. Una de las limitaciones de los juegos de mesa es que suelen ser menos completos y sólo se centran en un conjunto limitado de conceptos computacionales [128].

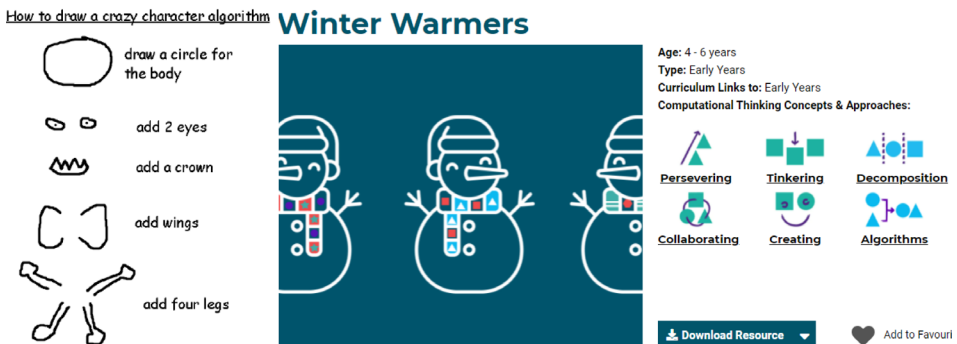
Otra posibilidad es la enseñanza de la programación a etapas tempranas a través de cuentos, como Hello Ruby, en el que cada lenguaje de programación está representado por un personaje. Algunos investigadores proponen crear comics [130] para aprender programación ya que consideran que el lenguaje visual de los cómics puede utilizarse para apoyar la enseñanza y el aprendizaje de conceptos computacionales.

Existen numerosas iniciativas que proporcionan ideas y materiales para llevar a la práctica actividades que trabajen el PCD, por ejemplo, CS Unplugged (véase la Tabla 7) que fue desarrollado a finales de los años 90 por profesores de ciencias de la computación y maestros de escuela, en la Universidad de Canterbury en Nueva Zelanda, y promueve el aprendizaje de conceptos de informática sin ordenadores, utilizando estrategias constructivistas de aprendizaje. Consideraron que los conceptos computacionales podían enseñarse sin usar un ordenador y que éste era muchas veces una distracción para el aprendizaje. El curso se diseñó por primera vez para alumnos de Educación Primaria y ha sido utilizado con éxito por estudiantes de todas las edades, incluso en contextos de educación superior y de personas mayores, y en entornos educativos formales e informales [131]. Además, la enseñanza se puede complementar con materiales multimedia.

La iniciativa Barefoot computing (véase la Tabla 7), proporciona también multitud de actividades para trabajar el PCD, por ejemplo, la actividad Crazy characters [132] dirigida a niños de entre 5 y 7 años, se utiliza habitualmente en colegios del Reino Unido y consiste en que un alumno o grupo de alumnos crean un "algoritmo", a través de instrucciones breves, para el diseño de un personaje inventado. Este algoritmo se lee en alto y otro alumno o grupo dibuja ese personaje según las instrucciones que recibe. Normalmente, el personaje no resulta como se esperaba y hay que mejorar el "algoritmo" (véase la Figura

9). Barefoot computing ofrece muchas otras ideas y materiales en su web y en cada una de ellas especifica qué aspectos del PC se trabajan en ella, por ejemplo, en la actividad de la Figura 9, se trabaja la perseverancia, las alternativas, la descomposición, la colaboración, la creatividad y la algoritmia.

Figura 9. Actividades de PCD: Crazy characters (izquierda) y Winter Warmers de Barefoot computing (derecha).



Finalmente, algunos autores como Adreas Dengel [133] proponen un PC re-enchufado (replugged) que permita realizar actividades propias de PCD, pero beneficiándose de la capacidad de inmersión que nos ofrecen la RV y la RA. De esta forma, se podrían mejorar, o incluso facilitar o permitir actividades de PCD que podrían ser imposibles, peligrosas o costosas de llevar a cabo en la realidad. Cuando se proporcionan medios educativos inmersivos que inducen una sensación de presencia en el entorno virtual, se puede ofrecer la ilusión de una experiencia de aprendizaje no mediada. De cualquier forma, esta propuesta aún debe investigarse más a fondo.

2.5. Herramientas de evaluación

La evaluación, tanto cuantitativa como cualitativa, del progreso de los estudiantes es indispensable para la introducción del PC en el currículo [31]. Aunque, de nuevo, hay una falta de consenso sobre qué estrategias son las más adecuadas para evaluar el PC en edades tempranas, ni existe acuerdo en qué debe evaluarse. Esto dificulta poder medir de la eficacia de las intervenciones para el aprendizaje del PC de forma fiable y válida [1], [5]. Tang *et al.* [134] identifican cuatro formas posibles de evaluar el PC: test tradicional, portafolio (proyecto), cuestionario y entrevista.

La mayoría de los estudios e investigaciones emplean cuestionarios y entrevistas para evaluar el PC, por ejemplo, Atmatzidou y Demetriadis en 2016 [55] o Denner en 2014 [135]. Cetin [136] en 2016 utilizó cuestionarios para medir la

comprensión conceptual de los profesores en formación sobre los constructos de programación, así como sus actitudes hacia la informática. Jun *et al.* [137] en 2014 utilizaron cuestionarios y entrevistas para medir la alfabetización computacional en estudiantes de Educación Primaria.

Los instrumentos para la evaluación del PC de tipo portafolio o proyecto, dependen de actividades o entornos concretos de aprendizaje, es decir, se basan en el análisis de un proyecto o actividad realizada por los alumnos en un entorno concreto de aprendizaje. El marco 3D de Brennan y Resnick [46] surge al aplicar este tipo de evaluación, ya que se basa en el análisis de proyectos creados con Scratch, aunque también incluyen evaluación cualitativa por medio de observación directa y entrevistas para evaluar las prácticas y perspectivas de los alumnos mientras interactúan con el entorno. Hay muchas otras iniciativas de evaluación de tipo portafolio (gran parte utilizan proyectos de Scratch), por ejemplo, las siguientes:

- CTP-Quiz [115] que consta de ocho preguntas relacionadas con actividades previas realizadas por los estudiantes con varios juegos concretos (por ejemplo, Frogger). La respuesta a estas preguntas permite a los investigadores reconocer patrones de PC en los estudiantes.
- La evaluación Fairy en Alice [138] es también de tipo portafolio. Alice es un entorno en tres dimensiones en el que los alumnos controlan a los personajes mediante programación basada en bloques visuales. Se analizan después los proyectos de los estudiantes en cuanto a la algoritmia y el uso eficaz de la abstracción y el modelado.
- Garneli y Chorianopoulos [139] basan su evaluación del PC en la creación de proyectos concretos en Scratch, diferenciando entre crear un videojuego o una simulación. Su evaluación se basa sobre todo en el análisis del código construido, aunque también realizan un análisis cualitativo mediante observaciones y entrevistas a los estudiantes.
- Otra iniciativa basada en Scratch es la evaluación PECT, propuesta por Seiter y Foreman [113], en 2013. PECT está diseñado para estudiantes de primaria y evalúa el PC a través del análisis del código en los proyectos de Scratch, separando los resultados en 3 componentes: evidencias, como operadores y condicionales; patrones, como movimiento e interactividad; y conceptos, como descomposición, abstracción y algoritmia. Los niveles de competencia de los estudiantes (básico, en desarrollo y competente) por componente, se establecen mediante rúbricas.
- Dr. Scratch [94], es una herramienta estática de análisis del código en proyectos de Scratch. Mediante el análisis de estos proyectos se pueden establecer categorías y niveles diferentes de habilidades de PC. Esto puede ser

de interés, por ejemplo, para crear itinerarios de aprendizaje de dificultad creciente, o para preparar actividades y retos centrados en el desarrollo de dimensiones específicas según el marco 3D de PC.

Además de instrumentos de evaluación de tipo portafolio, es importante que haya instrumentos de tipo test tradicional, independientes de entornos de programación concretos para utilizarlos de forma autónoma y así poder evaluar a los estudiantes sin depender de la utilización de un entorno en particular, así como para evaluar el efecto en el aprendizaje de futuros nuevos instrumentos. No obstante, existen pocos instrumentos en este sentido y no suelen incluir datos relativos a fiabilidad y validez de la evaluación realizada. En este particular, el *Test for Measuring Basic Programming Abilities* de Mühling *et al.* [140] y el *Commutative Assessment* de Weintrop y Wilensky [58] son ambos instrumentos validados bajo un enfoque psicométrico pero dirigidos a estudiantes de educación secundaria y universidad. Korkmaz, Çakir, y Özden [141] en 2017 desarrollaron una escala compuesta por 29 preguntas que evalúa cinco áreas de PC según una descomposición del PC propia de estos autores (creatividad, cooperación, pensamiento algorítmico, pensamiento crítico y resolución de problemas). El test mostró una buena fiabilidad y un buen índice de ajuste de los datos y evidencias de que los ítems medían constructos específicos.

El *Computational Thinking Test* (CTt) de Román-González *et al.* [44], destaca como instrumento para la evaluación del PC, ya que se aportan evidencias de fiabilidad y validez de criterio, bajo un enfoque psicométrico [142], [143]. Es consistente con las pruebas de Mühling [140] y de Weintrop [58] y está alineado con los estándares internacionales para pruebas psicológicas y educativas [144]. El CTt incluye 28 preguntas y se tarda 45 minutos de media en completar. Además, muestra correlaciones significativas con otras pruebas estandarizadas sobre habilidades mentales y de resolución de problemas. Aunque el CTt está dirigido a alumnos de entre 10 y 16 años, puede ser una base firme para desarrollar nuevos instrumentos de evaluación en etapas más tempranas.

Al comienzo de este trabajo de investigación, en el año 2019, no existían instrumentos de tipo test tradicional, independientes de entornos concretos, para evaluar el PC a lo largo de todos los cursos de Educación Primaria. Por ejemplo, el test de Chen *et al.* está dirigido a estudiantes de último ciclo de Educación Primaria [145], pero había una carencia clara de instrumentos orientados a niños más pequeños. Sin embargo, es manifiesta la necesidad de una evaluación fiable y válida del PC en todos los contextos educativos para que los investigadores puedan determinar si sus intervenciones son eficaces o no, por lo que nuestros esfuerzos en esta investigación han ido orientados hacia esta dirección mediante el desarrollo del *Beginners Computational Thinking Test* (BCTt), publicado en 2020 [8], cuyo diseño y validación se describe en el Capítulo 4, así como el *competence Computational Thinking test* (cCTt) como

adaptación para los niños más mayores, para los que el CTt fue una base consolidada y firme. Posteriormente a la publicación del BCTt ha habido otros intentos a destacar de instrumentos de tipo test tradicional para evaluar el PC en etapas tempranas, como por ejemplo el TechCheck CT unplugged [146]; o el Computational Thinking Assessment of Chinese Elementary School Students [147], también validado en cuanto a la fiabilidad y validez de constructo.

Una limitación de la evaluación por medio de test tradicional es que no abarca todas las dimensiones del PC del marco 3D. Según Grover y Pea [148], es necesario contar con *sistemas de evaluación*, para evaluar un aprendizaje más profundo combinando diferentes medidas de datos [142], por lo que es necesario combinar otros tipos de evaluación con el test tradicional.

La evaluación del PC también puede ser dinámica y así reflejar las capacidades y la progresión de los alumnos a lo largo del tiempo. Una forma de evaluar este aspecto es mediante analíticas de aprendizaje en juegos o *Game Learning Analytics* (GLA). La disciplina GLA es una extensión de analíticas de aprendizaje [149] pero adaptadas a entornos educativos basados en juegos. Con la técnica de GLA se recogen los datos de juego sin que el estudiante perciba que le están evaluando. Estos datos se analizan en tiempo real o después de jugar para proporcionar retroalimentación, y también después de que se hayan recogido suficientes datos durante varias partidas para que el análisis pueda arrojar conocimientos adicionales como la progresión de los estudiantes a lo largo de las partidas. Además, los datos de las interacciones del juego contienen información valiosa sobre cómo los jugadores interactuaron con el juego, el progreso, las habilidades, la persistencia, los problemas encontrados o la comprensión de los conceptos [16], [17]. Existen casos de éxito de GLA para la evaluación en juegos educativos [16], [119], [150], por ejemplo, CMX que es un MMORPG educativo dirigido a la educación secundaria para la enseñanza de la programación [151], [152]. Del mismo modo, existen varios marcos y estándares para el correcto análisis de los datos recogidos por los juegos [16], [17], [119], [152]-[154]. Para alcanzar nuestro objetivo de analizar el PC en todas sus dimensiones, planteamos también el uso de GLA en nuestro *sistema de evaluación*.

En esta tesis se trabaja para abarcar todas las dimensiones del marco 3D mediante diferentes enfoques de evaluación, creando así un *sistema de evaluación*. Así, se plantea una hipótesis de impacto de los diferentes enfoques de evaluación según el marco 3D (véase el Capítulo 3).

En este punto, el grado de cumplimiento de los objetivos generales de la tesis es el que se muestra en la Tabla 9. En el siguiente capítulo, se extraerán conclusiones a partir de esta investigación teórica y, a partir de ellas, se establecerán las hipótesis generales de la tesis.

Tabla 9. Seguimiento de los objetivos, capítulo 2.

Objetivos			Grado de cumplimiento			
			Bajo	Medio	Alto	
OP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.					
	OP1	Marco teórico y estado del arte.				
		OP1a	Definiciones teóricas y operacionales del PC.			
		OP1b	Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas.			
		OP1c	Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes.			
		OP1d	Marco teórico y estrategias para la evaluación del PC e instrumentos existentes.			
	OP2	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.				
		Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC.				
	OP3	OP3a	Diseñar y validar un instrumento de evaluación del PC en edades tempranas.			
		OP3b	Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas.			

Tabla 9. Seguimiento de los objetivos, capítulo 2. (continuada)

Objetivos		Grado de cumplimiento		
		Bajo	Medio	Alto
OP4		Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC.		
	OP4a	Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género.		
	OP4b	Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género.		
	OP4c	Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género.		
OC	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones.			

CAPÍTULO 3. INVESTIGACIÓN EMPÍRICA: HIPÓTESIS Y EXPERIENCIAS

Una vez definido el marco teórico y estudiado el estado de arte, a continuación, se exponen las conclusiones que se extraen de esta investigación teórica y, en base a las mismas, la dirección operativa que se establece para la investigación empírica, tal y como se establece en el objetivo general OP2: *Establecer una propuesta operativa del PC en base al OP1 y así constituir la dirección de la investigación empírica mediante hipótesis y objetivos base*. Posteriormente, en cada experiencia empírica, se detallarán las hipótesis y objetivos específicos de la misma (ver apartado 1.2. Objetivos).

De esta forma, en este capítulo se plantean, en relación con los objetivos generales de la tesis, las **hipótesis, desarrollos y experiencias realizadas en la investigación empírica**. En los capítulos siguientes, se irán detallando en profundidad dichos desarrollos y experiencias, así como las hipótesis y objetivos específicos relativas a cada una de ellas.

3.1. Propuesta de definición de PC

Dado que no existe una definición consensuada del PC, se ha realizado una revisión de la literatura y, basándonos en los puntos en común de las definiciones existentes (apartado 2.1), en esta tesis se propone la siguiente definición:

El pensamiento computacional es una habilidad cognitiva que permite resolver problemas utilizando estrategias computacionales.

De igual forma, en esta tesis, se propone la siguiente descomposición del PC en seis factores y sus definiciones: 1) **abstracción** (proceso de obtener algo simple desde algo complejo, obviando los detalles); 2) **análisis de datos** (buscar, seleccionar, organizar y analizar lógicamente los datos); 3) **descomposición de problemas** (descomponer problemas en otros más pequeños que pueden resolverse con mayor facilidad); 4) **algoritmia** (identificar instrucciones específicas y explícitas que, paso a paso, llevan a cabo un proceso); 5) **depuración de errores** (identificar y corregir los errores en la solución aportada); y 6) **generalización** (transferir un proceso de resolución de problemas a una variedad grande de problemas).

Además, se propone también la identificación de disposiciones y predisposiciones para el aprendizaje del PC, tal y como se plantea en la literatura [41], [53]. Esta identificación tendrá especial relevancia en el marco de la tesis,

ya que se analizarán estos aspectos motivacionales y de persistencia ante los retos en el Capítulo 7.

En cuanto al marco de referencia educativo, se ha seleccionado uno de los marcos de PC más citados en la literatura y más utilizados empíricamente, que es el marco 3D de Brennan y Resnick (2012) [46]. Este marco clasifica el PC según tres dimensiones: 1) **conceptos computacionales** (conceptos que utilizan los programadores); 2) **prácticas computacionales** (prácticas de resolución de problemas que se necesitan/producen en el proceso de programación); y 3) **perspectivas computacionales** (perspectivas que se forman los diseñadores sobre sí mismos y el mundo que les rodea). Este marco es adecuado para considerar el PC en entornos que utilizan la programación informática y en Educación Primaria [61], por lo que parece adecuado como punto de partida en esta tesis.

3.2. Hipótesis generales de la investigación empírica

A continuación, se describen las hipótesis generales que se plantean tras la investigación teórica y que constituyen la base para la investigación empírica realizada. A continuación, se especifican las hipótesis empíricas (HE) planteadas y las experiencias realizadas en relación a cada una de ellas.

*HE1 – sobre **la evaluación** del PC en Educación Primaria.*

Es posible evaluar el PC en la etapa de Educación Primaria mediante un test tradicional independiente de un entorno concreto.

Como se ha evidenciado en el estado del arte, pese a que evaluar del progreso de los estudiantes es indispensable para la introducción del PC en el currículo [31], hay una falta de consenso sobre qué estrategias son las más adecuadas para esta evaluación y, además, existen pocos instrumentos en este sentido y no suelen incluir datos relativos a su fiabilidad y validez. Sin embargo, es manifiesta la necesidad de una evaluación fiable y válida del PC en todos los contextos educativos para que los investigadores puedan determinar si sus intervenciones son eficaces o no.

Al comienzo de este trabajo de investigación, en el año 2019, no existía ningún instrumento de tipo test tradicional, independiente de entornos concretos, para evaluar el PC en Educación Primaria, por lo que los esfuerzos en esta investigación se han orientado hacia esta dirección mediante el desarrollo del test de PC para principiantes: *Beginners Computational Thinking test (BCTt)*, publicado en 2020 [8], y que se detalla en el Capítulo 4, para el que el *Computational Thinking Test (CTt)* [44] ha sido una base consolidada y firme.

El BCTt ha tenido gran repercusión internacional y, junto a otros equipos de investigación de países como Suiza, Portugal o Países Bajos, se están realizando esfuerzos para determinar los límites de edad para la aplicación del test o desarrollando nuevas versiones para adaptarse a estudiantes de diferentes edad y poblaciones. De esta forma, en función de los resultados de fiabilidad del BCTt para los ciclos superiores de Educación Primaria, y gracias a la colaboración internacional, se ha desarrollado y validado el *competence Computational Thinking test* (cCTt) para así, junto con el BCTt y el CTt, cubrir todos los rangos de edad correspondientes a esta etapa educativa.

Por otro lado, se plantea poder utilizar el BCTt en también en la etapa de Educación Infantil, por lo que en el Capítulo 5 se describe una experiencia en la que trata de obtener el límite inferior de edad en la que se puede administrar el test, validando su fiabilidad en niños de 4 y 5 años.

*HE2 – sobre la relación de **estrategias de aprendizaje** con marco 3D. Mediante diferentes estrategias de aprendizaje, pueden abarcarse las tres dimensiones del PC, de forma que la relación entre el marco 3D y las estrategias de aprendizaje es como se muestra en la Figura 10.*

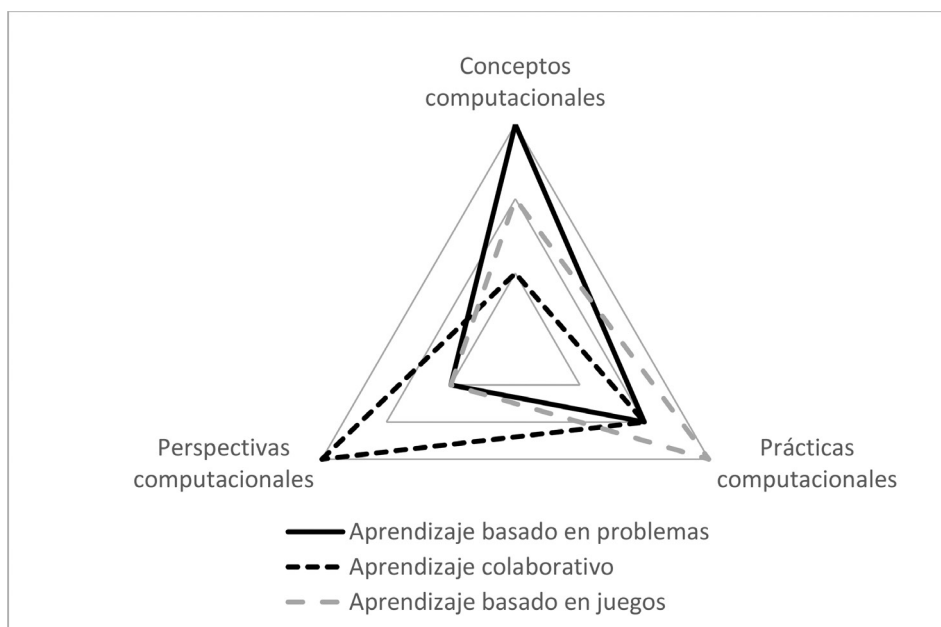
Del estado del arte, se ha detectado que la mayoría de las investigaciones y actuaciones para el desarrollo del PC en el aula, se centran en la dimensión *conceptos computacionales* del marco 3D de PC de Brennan y Resnick [46], [61], [93] y, solamente en algunas investigaciones muy recientes, sobre el aprendizaje y la evaluación de las dimensiones *prácticas computacionales* y *perspectivas computacionales* [93]–[97]. Esto nos lleva a la necesidad de más investigación y de prestar especial atención a estas dos últimas dimensiones, especialmente más allá del entorno de Scratch [98] en el cual se ubican la mayor parte de los casos de estudio en las investigaciones. Además, el PC es un conjunto interdisciplinario de habilidades y es necesario que existan métodos operativos de aprendizaje a través de diversas estrategias [61], [96].

Además, para dotar a los estudiantes de habilidades de resolución de problemas que puedan transferir posteriormente a dominios no relacionados con la programación [41], es necesario desarrollar el PC desde etapas tempranas y es especialmente importante desarrollar las prácticas y perspectivas computacionales. Estas dos últimas dimensiones del marco 3D son las que nos permiten trabajar la componente de esencial del PC que, en la descomposición propuesta en esta tesis, se ha descrito como la *generalización*.

En el marco teórico de esta tesis se exploraron las estrategias de *aprendizaje basado en problemas*, *aprendizaje colaborativo* y *aprendizaje basado en juegos* como las estrategias de aprendizaje más utilizadas y con más impacto en el aprendizaje del PC. En esta tesis se va a tratar de relacionar el posible impacto de cada una de estas estrategias de aprendizaje en cada dimensión

del marco 3D del PC. En la Figura 10 se muestra nuestra hipótesis sobre dicha relación: las dimensiones computacionales se sitúan en los vértices del triángulo, de modo que cuanto más cerca de un vértice aparezca la estrategia, mayor será la probabilidad de que esta dimensión se vea afectada por la estrategia.

Figura 10. Impacto esperado de la estrategia de aprendizaje en las dimensiones clave del marco 3D del PC.



A diferencia de la estrategia de aprendizaje basada en problemas, en la estrategia de aprendizaje basada en juegos el alumno no recibe necesariamente una definición previa de los conceptos o contenidos a tratar, sino que guía su propio aprendizaje libremente, motivado por la sensación de logro (véase la Tabla 4), por lo que nuestra hipótesis es que la adquisición de conceptos podría no ser completa. Por el contrario, en el aprendizaje basado en juegos, las prácticas computacionales podrían verse especialmente potenciadas, ya que dominarlas es esencial para progresar en los retos o niveles que se proponen [61].

Por otro lado, la hipótesis es que las prácticas y perspectivas computacionales se pueden ver reforzadas si se utilizan una estrategia colaborativa. Esta estrategia se utiliza a menudo en los estudios que implican experiencias de programación y puede mejorar tanto las prácticas como las dimensiones de las perspectivas computacionales, debido a que los estudiantes revisan y piensan sobre su experiencia de programación, mejorando así su propio

proceso de aprendizaje [61]. En un entorno colaborativo enfocado a aprender programación, los estudiantes no sólo reflexionan sobre el funcionamiento de su programa, sino que también revisan y piensan en los programas que han creado sus compañeros. El debate y discusión, en voz alta, de todas las soluciones de programación aportadas, propias y ajenas, puede mejorar estas dos dimensiones del PC, especialmente las perspectivas computacionales [15], [60], [61].

Posteriormente, mediante la experiencia en colegios descrita en el Capítulo 6, en el que se utilizarán las tres estrategias de aprendizaje simultáneamente, se comprobará esta hipótesis.

*HE3 – sobre la **estrategia basada en juegos** para el aprendizaje del PC*
Los entornos que utilizan la estrategia basada en juegos (GBL) potencian el aprendizaje de habilidades de PC en Educación Primaria en contextos escolares, sobre en los niños más pequeños y si se aplica una componente colaborativa.

Solamente un 20% de las experiencias que incorporan GBL en la literatura se orientan a la etapa de Educación Infantil y primaria, pese a que esta estrategia suele arrojar resultados muy positivos en el aprendizaje en un 85% de los casos [11]. La hipótesis planteada es que la enseñanza de habilidades de PC y su evaluación es posible y fiable en estas etapas escolares utilizando GBL.

Esta hipótesis se comprueba en el Capítulo 6, mediante la validación a través experiencias en colegios de un videojuego educativo, desarrollado específicamente en el marco de la tesis, para la enseñanza de la programación y desarrollo del PC.

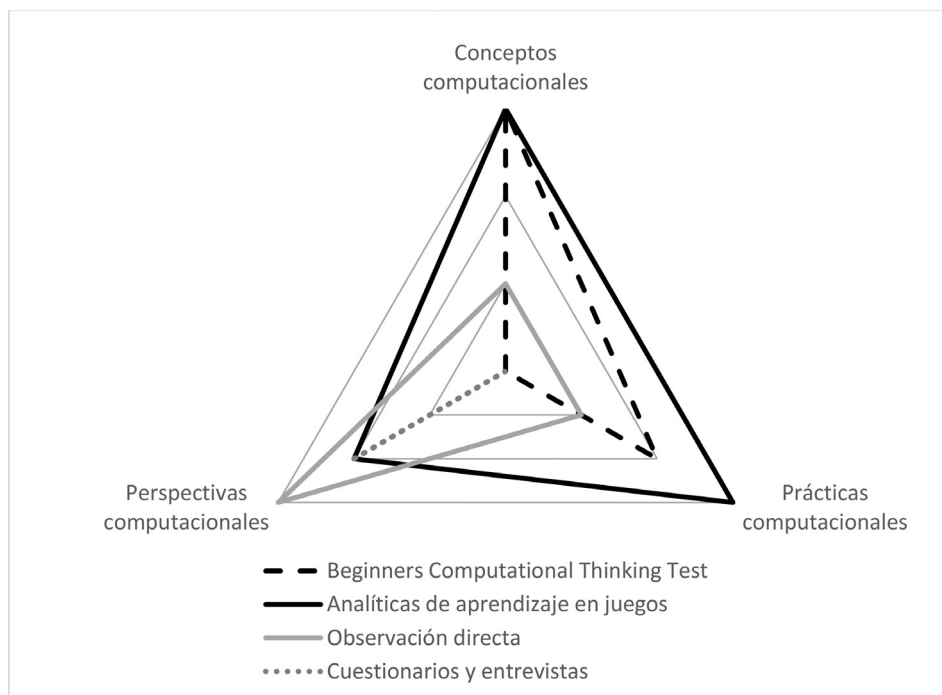
*HE4 – sobre la relación entre las **estrategias de evaluación** y el marco 3D.*
Es posible abarcar las tres dimensiones del marco 3D del PC mediante diferentes estrategias de evaluación tal y como se muestra en la Figura 11.

Una limitación de la evaluación por medio de test tradicional es que no abarca todas las dimensiones del PC del marco 3D. Según Grover y Pea [148], es necesario contar con *sistema de evaluación*, para evaluar un aprendizaje más profundo combinando diferentes medidas de datos [142], además del test tradicional.

Por otro lado, la evaluación del PC también puede ser dinámica y así reflejar las capacidades y la progresión de los alumnos a lo largo del tiempo. Este aspecto es evaluable mediante analíticas de aprendizaje en juegos o *Game Learning Analytics* (GLA). Para alcanzar el objetivo de analizar el PC en todas sus dimensiones, en esta tesis se plantea el uso de GLA como parte del *sistema de evaluación* propuesto. La hipótesis de impacto de las diferentes

estrategias de evaluación según el marco 3D se muestra en la Figura 11, en la que las dimensiones computacionales se sitúan en los vértices del triángulo, de modo que cuanto más cerca se represente la estrategia de evaluación de un vértice, mayor será la probabilidad de que esta dimensión se vea afectada por la estrategia.

Figura 11. Impacto esperado de la estrategia de evaluación en las dimensiones clave del marco 3D del PC.



En los capítulos 6 y 7, mediante la combinación de diferentes estrategias de evaluación, se explorará esta hipótesis.

*HE5 – sobre el análisis de las **predisposiciones y disposiciones del PC** con GLA. Los entornos de aprendizaje basados en juegos voluntarios pueden aportar información sobre disposiciones y predisposiciones para el aprendizaje del PC según las características del individuo en cuanto a edad y género.*

La estrategia GBL se utiliza con éxito en entornos escolares, formalmente controlados y dirigidos, en tiempo y forma. La hipótesis que se plantea es que, mediante entornos GBL voluntarios, fuera del contexto escolar, y utilizando GLA, es posible la identificación de disposiciones y predisposiciones [41], [53] para el aprendizaje del PC, como por ejemplo intereses, motivación,

persistencia ante los retos o comportamiento ante las recompensas. Esta identificación tendrá especial relevancia en el marco de la tesis y se detallará en el Capítulo 7.

3.3. Estructura de la investigación empírica

A continuación, se enumeran las experiencias que se plantean tras la investigación teórica y que constituyen la base de la investigación empírica realizada. Estas experiencias se relacionan directamente con los objetivos generales de la tesis y con las hipótesis previamente descritas. En los capítulos siguientes, se detallarán los objetivos específicos de cada una de las experiencias.

E1 – Desarrollo y validación de prueba independiente para la evaluación del PC en Educación Primaria: Beginners Computational Thinking test (BCTt).

En base a la HE1, se plantea el desarrollo de un test de PC que abarque la etapa de Educación Primaria y su validación en cuanto a fiabilidad, realizando para ello un procedimiento de validación de contenido a través de un panel de expertos y, después, una experiencia de aplicación en alumnos de Educación Primaria para comprobar su fiabilidad.

Además, según los resultados, se plantean otras experiencias, a través de colaboraciones internacionales, para el desarrollo y validación de una adaptación del test para los niños más mayores: competence Computational Thinking test (cCTt), de forma que, al finalizar todos los desarrollos y validaciones, se disponga de instrumentos fiables de evaluación que cubran la etapa completa de Educación Primaria.

E2 – Estimación de la validez del BCTt en alumnos de Educación Infantil.

Una vez desarrollado el test y sus adaptaciones en la experiencia anterior, cubriendo toda la etapa de Educación Primaria, se plantea, nuevamente a través de colaboraciones internacionales, el estudio de los límites inferiores de aplicación en cuanto a la edad de los niños. Se plantea la validación del BCTt mediante una experiencia en alumnos de Educación Infantil para concretar este límite inferior.

E3 – Validación y análisis del entorno educativo basado en el juego BAC, en modos individual y colaborativo y análisis de habilidades de PC en Educación Primaria.

En base a las hipótesis HE2, HE3 y HE4, se plantea el desarrollo de un videojuego que se basa en GBL e incluye GLA, como herramienta para el

aprendizaje y la evaluación del PC en alumnos de Educación Primaria, así como su validación a través de experiencias en distintos colegios.

E4 – Análisis de motivación intrínseca, intereses, persistencia y comportamiento frente a recompensas en el aprendizaje de la programación en etapas tempranas utilizando el entorno BAC.

En base a las hipótesis HG4 y HG5, se plantea el despliegue masivo de BAC para que los usuarios puedan jugar voluntariamente y, de esta forma, analizar con GLA la progresión del aprendizaje a lo largo del tiempo, así como predisposiciones y disposiciones del PC relacionadas con la motivación intrínseca, como intereses, persistencia ante los retos, y comportamiento ante las recompensas.

En la Tabla 10 se muestra la relación entre las experiencias, los objetivos generales y las hipótesis base para la investigación empírica. Los objetivos generales mostrados están incluidos en el objetivo principal *OP: Avanzar en el conocimiento sobre el PC, sobre todo en su componente operacional, es decir, en cuanto a métodos de enseñanza y de evaluación eficaces, sobre todo a etapas tempranas*. Las experiencias, también persiguen transversalmente el objetivo común *OC: Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones*. En la tabla también se indica en qué capítulo se describe cada experiencia.

Tabla 10. Relación entre las experiencias, las hipótesis y los objetivos de la tesis.

Experiencia	Objetivos generales				Hipótesis inv. empírica	Capítulo
E1	Desarrollo y validación de prueba independiente para la evaluación del PC en Educación Primaria: Beginners Computational Thinking test (BCTt).	OP3a	OP4a	OP4c	OC	4
E2		OP3a	OP4a	OP4c	OC	
E3	Validación y análisis del entorno educativo basado en el juego BAC, en modos individual y colaborativo y análisis de habilidades de PC en Educación Primaria	OP3b	OP4a	OP4b	OP4c	6
E4		OP3b	OP4a	OP4b	OP4c	
E4	Análisis de motivación intrínseca, intereses, persistencia y comportamiento frente a recompensas en el aprendizaje de la programación en etapas tempranas utilizando el entorno BAC.	OP4a	OP4b	OP4c	HE4	7

En este punto, el grado de cumplimiento de los objetivos es el que se muestra en la Tabla 11.

Tabla 11. Seguimiento de los objetivos, capítulo 3.

Objetivos			Grado de cumplimiento		
			Bajo	Medio	Alto
OP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.				
	OP1	Marco teórico y estado del arte			
		OP1a Definiciones teóricas y operacionales del PC			
		OP1b Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas			
		OP1c Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes			
		OP1d Marco teórico y estrategias para la evaluación del PC e instrumentos existentes			
	OP2	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.			

	OP3	Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC
		OP3a Diseñar y validar un instrumento de evaluación del PC en edades tempranas
		OP3b Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas
	OP4	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC
		OP4a Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género
		OP4b Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género
		OP4c Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género
	OC	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones

CHAPTER 3. EMPIRICAL RESEARCH: HYPOTHESES AND CASE STUDIES

Once the theoretical framework has been defined and the state of the art has been reviewed, the conclusions drawn from this theoretical research are set out below and, based on these, the operational direction established for the empirical research, as set out in the general aim OP2: To establish an operational proposal for the CT based on OP1, and thus constitute the direction of the empirical research by means of hypotheses and basic objectives. Subsequently, in each empirical case study, the specific hypotheses and specific aims will be detailed (see section 1.2. General aims).

This chapter thus sets out, in relation to the general aims of the thesis, the **hypotheses, developments and case studies carried out in the empirical research**. In the following chapters, these developments and case studies will be described in depth, as well as the specific hypotheses and aims related to each one of them.

3.4. Suggested definition of CT

Given that there is no agreed definition of CT, a review of the literature has been carried out and, based on the commonalities of the existing definitions (section 2.1), the following definition is proposed in this thesis:

Computational thinking is a cognitive ability that enables problem solving using computational strategies.

Similarly, in this thesis, the following decomposition of CT into 6 components is suggested: 1) **abstraction** (process of obtaining something simple from something complex, obviating details); 2) **data analysis** (searching, selecting, organising and logically analysing data); 3) **problem decomposition** (decomposing problems into smaller ones that can be solved more easily); 4) **algorithms** (identifying specific and explicit instructions that, step by step, implement a process); 5) **error debugging** (identifying and correcting mistakes in a given solution); and 6) **generalisation** (transferring a problem-solving strategy to a large variety of problems).

In addition, the **identification of dispositions and predispositions for learning CT**, as proposed in the literature [41], [53]., is also addressed. This identification will be of particular relevance in the framework of the thesis, as these motivational aspects and persistence in the face of challenges will be analysed in Chapter 7.

Regarding the educational frame of reference, one of the most cited in the literature and most empirically used CT frameworks has been selected, which is Brennan and Resnick's (2012) 3D framework [46]. This framework classifies CT along three dimensions (see Table 3): 1) **computational concepts** (concepts that programmers use); 2) **computational practices** (problem-solving practices that are needed/produced in the programming process); and 3) **computational perspectives** (perspectives that designers form about themselves and the world around them). This framework is suitable for considering CT in environments that use computer programming and in Primary Education [61], so it seems appropriate as a starting point in this thesis.

3.5. General hypotheses of the empirical investigation

The following is a description of the general hypotheses that emerged from the theoretical research and that form the basis for the empirical research carried out. The empirical hypotheses (HE) and the studies carried out in relation to each of them are as specified below.

HE1 – on the assessment of CT in Primary Education.

It is possible to assess CT in primary education using a traditional test independent of a specific environment.

As has been shown in the state of the art, although assessment of student progress is essential for the introduction of CT in the curriculum [31], there is a lack of consensus on which strategies are the most appropriate for this assessment and, moreover, there are few such instruments and they do not usually include data on their reliability and validity. However, there is a clear need for reliable and valid assessment of CT in all educational contexts so that researchers can determine whether their interventions are effective.

At the beginning of this research work in 2019, there was no traditional, setting-independent test-type instrument for assessing CT in Primary Education, so efforts in this research have been directed in this direction by developing the Beginners CT test: Beginners Computational Thinking test (BCTt), published in 2020 [8], and detailed in Chapter 4, for which the Computational Thinking Test (CTt) [44] has been a consolidated and firm basis.

The BCTt has had considerable international impact and, together with other research teams in countries such as Switzerland, Portugal and the Netherlands, efforts are being made to determine the age limits for the application of the test or to develop new versions to adapt it to students of different ages and populations. In this way, based on the reliability results of the BCTt for upper stages of Primary Education, and thanks to international collaboration,

the competence Computational Thinking test (cCTt) has been developed and validated to cover, together with the BCTt and the CTt, all the age ranges corresponding to this educational stage.

On the other hand, the BCTt can also be used in the Infant Education stage, so Chapter 5 describes a study in which the aim is to obtain the lower age limit at which the test can be administered, validating its reliability in children of 4 and 5 years of age.

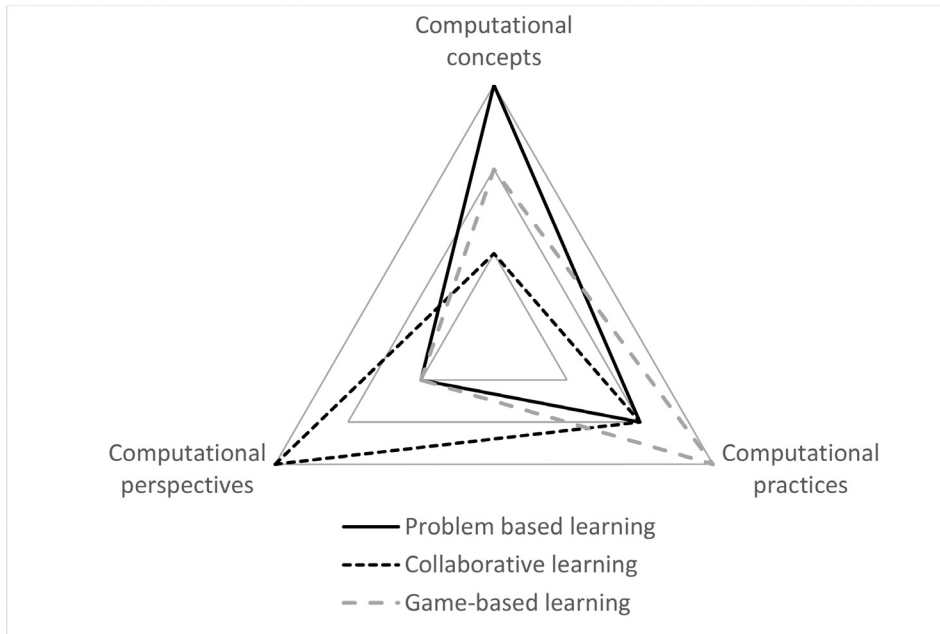
HE2 – on the relationship of learning strategies to the 3D frame. Through different learning strategies, the three dimensions of CT can be covered, so that the relationship between the 3D framework and the learning strategies is as shown in Figure 10. Expected impact of the learning strategy on the key dimensions of the CT 3D framework.

From the state of the art, it has been noticed that most of the research and actions for the development of CT in the classroom, focus on the computational concepts dimension of Brennan and Resnick's CT 3D framework [46], [61], [93] and, only in some very recent research, on the learning and assessment of computational practices and computational perspectives dimensions [93]–[97]. This leads us to the need for more research and special attention to the latter two dimensions, especially out of the Scratch environment [98] in which most of the case studies in the research are conducted. Moreover, CT is an interdisciplinary set of skills and there is a need for operational methods of learning through a combination of various strategies [61], [96].

Furthermore, to equip students with problem-solving skills that they can later transfer to non-programming domains [41], it is necessary to develop CT from early stages and it is especially important to develop computational practices and perspectives. These last two dimensions of the 3D framework are the ones that allow us to work on the essential component of the CT which, in the decomposition proposed in this thesis, has been described as *generalisation*.

In the theoretical framework of this thesis, problem-based learning, collaborative learning, and game-based learning strategies were identified as the most commonly used learning strategies with the greatest impact on CT learning. In this thesis we are going to try to link the possible impact of each of these learning strategies on each dimension of the CT 3D framework. Figure 10 shows our hypothesis about this relationship: the computational dimensions are located at the vertices of the triangle, so that the closer to a vertex the strategy appears, the higher the probability that this dimension will be affected by the strategy.

Figure 10. Expected impact of the learning strategy on the key dimensions of the CT 3D framework.



Unlike the problem-based learning strategy, in the game-based learning strategy the learner does not necessarily receive a prior definition of the concepts or content to be covered, but guides his or her own learning freely, motivated by a sense of achievement (see Table 4), so the hypothesis is that the acquisition of concepts may not be complete. On the contrary, in game-based learning, computational practices could be particularly enhanced, since mastering them is essential to progress in the challenges or levels proposed [61].

On the other hand, we hypothesise that computational practices and perspectives can be enhanced by using a collaborative strategy. This strategy is often used in studies involving programming experiences and can enhance both the practices and computational perspectives dimensions, since students review and think about their programming experience, thus enhancing their own learning process [61]. In a collaborative environment focused on learning programming, students not only reflect on how their program works, but also review and think about the programs their peers have created. Debate and discussion, aloud, of their own and others' programming solutions, can enhance both dimensions of CT, especially computational perspectives [15], [60], [61].

Subsequently, through the case study in schools described in Chapter 6, in which all three learning strategies are used simultaneously, this hypothesis will be tested.

*HE3 – on the **game-based strategy** for learning CT*

Environments using the Game-Based Learning approach (GBL) enhance the learning of CT skills in Primary Education in school settings, especially in younger children and if a collaborative component is applied.

Just 20% of the studies that incorporate GBL in the literature are aimed at the pre-school and primary school stages, even though this strategy tends to yield very positive learning results in 85% of cases [11]. The hypothesis posed is that the teaching of CT skills and their assessment is possible and reliable at these school stages using GBL

This hypothesis is confirmed in Chapter 6, by validating through studies in schools an educational video game, developed specifically within the framework of the thesis, for the teaching of programming and CT development.

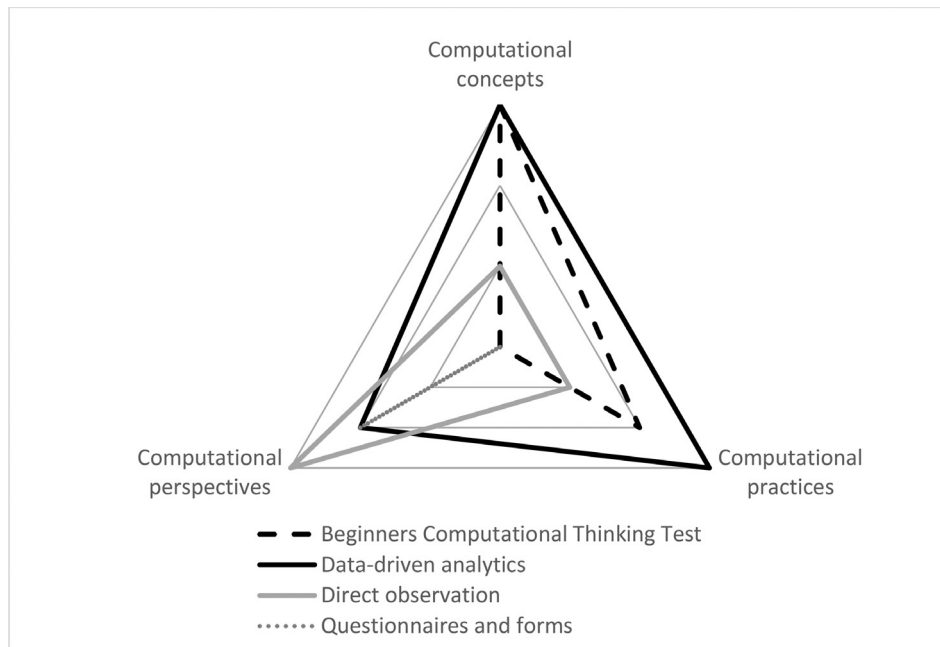
HE4 – on the relationship between assessment strategies and the 3D framework

It is possible to cover the three dimensions of the CT 3D framework through different assessment strategies as shown in Figure 11. Expected impact of the assessment strategy on the key dimensions of the CT 3D framework.

A limitation of traditional test-based assessment is that it does not cover all dimensions of the CT of the 3D framework. According to Grover and Pea [148], a *system of assessments* is needed to assess deeper learning by combining different data measures [142], in addition to the traditional test.

Moreover, the assessment of CT can also be dynamic, reflecting learners' abilities and progression over time. This aspect can be evaluated by means of Game Learning Analytics (GLA). In order to achieve the aim of analysing CT in all its dimensions, this thesis proposes the use of GLA as part of the proposed *system of assessments*. The impact hypothesis of the different assessment strategies according to the 3D framework is shown in Figure 11, in which the computational dimensions are placed at the vertices of the triangle, so that the closer the assessment strategy is represented to a vertex, the higher the probability that this dimension will be affected by the strategy.

Figure 11. Expected impact of the assessment strategy on the key dimensions of the CT 3D framework.



In chapters 6 and 7, by combining different assessment strategies, this hypothesis will be explored.

*HE5 – sobre el análisis de las **predisposiciones y disposiciones del PC** con GLA*

Los entornos de aprendizaje basados en juegos voluntarios pueden aportar información sobre disposiciones y predisposiciones para el aprendizaje del PC según las características del individuo en cuanto a edad y género.

The GBL strategy is successfully used in school settings, formally controlled and directed, in time and form. It is hypothesised that through voluntary GBL settings, outside the school context, and using GLA, it is possible to identify dispositions and predispositions [41], [53] for CT learning, such as interests, motivation, persistence in the face of challenges or behaviour towards rewards. This analysis will be of relevance in the framework of the thesis and will be detailed in Chapter 7.

3.6. Empirical research structure

Following the theoretical research, a series of case studies were carried out that are directly related to the general aims of the thesis and to the hypotheses previously described. In the following chapters, the specific aims of each of the case studies will be detailed.

E1 – Development and validation of a stand-alone test for the assessment of CT in Primary Education: Beginners Computational Thinking test (BCTt).

Based on HE1, we propose the development of a CT test that covers the Primary Education stage and its validation in terms of reliability, carrying out a content validation procedure through a panel of experts and, later, an administration on Primary Education students to check its reliability.

In addition, depending on the results, other case studies are planned, through international collaborations, for the development and validation of an adaptation of the test for older children: competence Computational Thinking test (cCTt), so that, when all the developments and validations are completed, reliable assessment instruments covering the entire Primary Education stage would be available.

E2 – Estimation of the validity of the BCTt in Infant Education students.

Once the test and its adaptations have been developed in the previous study, covering the entire Primary Education stage, research on the lower limits of application of the tests in terms of the age of the children is planned, once again through international collaborations. Validation of the BCTt is proposed by means of a case study with pre-school students in order to determine this lower limit.

E3 – Validation and analysis of the educational environment based on the BAC game, in individual and collaborative modes and analysis of CT skills in Primary Education.

Based on hypotheses HE2, HE3 and HE4, we propose the development of a video game based on GBL and including GLA, as a tool for learning and assessment of CT in Primary Education students, as well as its validation through case studies in different schools.

E4 – Analysis of intrinsic motivation, interests, persistence, and behavior in response to rewards, in learning programming in early stages using the BAC environment.

Based on hypotheses HG4 and HG5, we propose the massive deployment of BAC so that users can play voluntarily and, in this way, analyse with GLA the progression of learning over time, as well as predispositions and dispositions of the CT related to intrinsic motivation, such as interests, persistence in the face of challenges, and behavior in response to rewards.

Table 10 shows the relationship between the case studies, the general aims, and the base hypotheses for the empirical research. The general aims shown are included in the main aim *OP: To advance knowledge about CT, especially in its operational component, i.e., in terms of effective teaching and assessment methods, especially at early stages*. The studies also pursue transversally the common aim *OC: Design and validation of CT assessment instruments covering a wide range of ages and populations*. The table also indicates in which chapter each case study is described.

Table 10. Relationship between the case studies, the hypotheses and the aims of the thesis.

Case studies	Aims	Hypotheses	Chapter
E1	Development and validation of a standalone test for the assessment of CT in Primary Education: Beginners Computational Thinking test (BCTt).	HE1	4
E2	Estimation of the validity of the BCTt in Infant Education students.	HE1	5
E3	Validation and analysis of the educational environment based on the BAC game, in individual and collaborative modes and analysis of CT skills in Primary Education.	HE2 HE3 HE4	6
E4	Analysis of intrinsic motivation, interests, persistence, and behavior in response to rewards, in learning programming in early stages using the BAC environment.	HE4 HE5	7

At this point, the degree of achievement of the aims is as shown in Table 11.

Table 11. Tracking the aims, chapter 3.

Aims			Compliance degree		
			Low	Medium	High
OP	OP1		To advance knowledge about CT, especially in its operational component, i.e., in terms of effective learning and assessment strategies, especially at early stages.		
			Compile a broad theoretical framework and state of the art of publications and studies on CT.		
		OP1a	Theoretical and operational definitions of CT		
		OP1b	Frameworks, school curricula and initiatives for the development of CT at international and national level at early stages.		
		OP1c	Theoretical framework and strategies for CT development/learning, as well as existing tools and environments.		
		OP1d	Theoretical framework and strategies for the assessment of CT, as well as existing instruments.		
		OP2	Establish an operational proposal of CT on the basis of OP1 and thus constitute the direction of the empirical research by means of hypotheses and base aims.		

OC	OP3		To design and validate, empirically, tools for learning and assessment of CT.
		OP3a	To design and validate a CT assessment tool aimed at early ages.
		OP3b	To develop and validate a tool for learning and assessment of CT at an early age.
	OP4		To carry out empirical studies to analyse data related to CT skills as well as CT dispositions and predispositions.
		OP4a	To collect and analyse data to improve the design of the school curricula related to CT according to age and gender.
		OP4b	To collect and analyse data to improve the design of learning tools related to CT according to age and gender.
		OP4c	To collect and analyse data to improve the design of CT assessment tools according to age and gender.
			Design and validation of CT assessment instruments covering a wide range of ages and populations.

CAPÍTULO 4. DESARROLLO Y VALIDACIÓN DE UNA PRUEBA INDEPENDIENTE PARA LA EVALUACIÓN DEL PC

En este capítulo, se describe la experiencia de desarrollo y validación del instrumento de evaluación del PC: Beginners Computational Thinking Test (BCTt) dirigido a alumnos de Educación Primaria, así como la posterior adaptación del test para los alumnos más mayores en los que el BCTt resulta menos fiable, mediante una colaboración internacional que resulta en el instrumento de evaluación: Competence Computational Thinking Test (cCTt).

4.1. Introducción y objetivos específicos

Como se ha descrito anteriormente, junto con el aprendizaje, la evaluación del PC es una parte indispensable a tener en cuenta para introducir el PC en el currículo escolar, ya que la evaluación de los estudiantes con fines pedagógicos es esencial [31]. Desafortunadamente, aunque hay múltiples investigadores que describen experiencias de integración del pensamiento computacional en el currículo, los esfuerzos que implican la evaluación formal del pensamiento computacional se han centrado principalmente en la educación secundaria, el bachillerato y la universidad [113]. Además, los instrumentos de evaluación propuestos por las investigaciones recientes se basan principalmente en el análisis de proyectos realizados por los estudiantes en entornos de programación específicos. De esta forma, no existían, en el comienzo de este trabajo de investigación, instrumentos de tipo test tradicional, independientes de entornos concretos, para evaluar el PC a lo largo de todos los cursos de Educación Primaria. Sin embargo, es manifiesta la necesidad de una evaluación fiable y válida del PC en todos los contextos educativos para poder mejorar el currículo escolar, para que los investigadores puedan determinar si sus intervenciones son eficaces o no, y para poder desarrollar cada vez mejores instrumentos de aprendizaje y evaluación del PC.

En este sentido, se ha diseñado y validado el BCTt, dirigido a etapas tempranas y publicado en 2020 [7], [8]. Se parte la hipótesis base *HE1: es posible evaluar el PC en la etapa de Educación Primaria mediante un test tradicional independiente de un entorno concreto*. A continuación, se enumeran los **objetivos generales** asociados a esta experiencia:

- *OP3: Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC, y en particular OP3a: Diseñar y validar un instrumento de evaluación del PC en edades tempranas.*
- *OP4: Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC; en concreto el OP4a: Extraer y analizar datos para mejorar el diseño del currículo escolar relativo*

a PC según edad y género; y el OP4c: Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género.

- OC: Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones.

En el apartado 2.5, se describieron algunos de los intentos e instrumentos de evaluación del PC, de entre los que destaca el Computational Thinking Test (CTt) [44] como un instrumento autónomo, independiente de cualquier entorno de programación, para la evaluación del PC. El CTt está diseñado bajo un enfoque psicométrico y aporta evidencia sobre su fiabilidad y contenido [155], validez de criterio [142] y predictiva [143]; es consistente con el *Test for Measuring Basic Programming Abilities* [140] y con el *Commutative Assessment* [58]; y está alineado con los estándares internacionales para pruebas psicológicas y educativas [144]. En términos del marco 3D, el CTt de Román *et al.* se centra en los conceptos computacionales, parcialmente en las prácticas computacionales e ignora las perspectivas computacionales [44].

El **objetivo específico** de esta experiencia es el diseño y validación en cuanto a fiabilidad de un instrumento independiente de un entorno concreto dirigido a alumnos de Educación Primaria. Para la consecución de este objetivo y, aunque el CTt está dirigido a estudiantes de entre 10 y 16 años, se ha tomado como una base consolidada y firme para el BCTt, dirigido a niños de Educación Primaria, desarrollado y validado en esta experiencia. Como la población objetivo del BCTt es más joven que la del CTt, el test se ha adaptado tanto en su forma como en su contenido. Además, el diseño del BCTt incluye varias innovaciones de fondo que pretenden ser mejoras sustanciales con respecto a otros test tradicionales. En este capítulo se presentan las pautas que se han seguido para el diseño del BCTt como instrumento de evaluación autónomo, independiente de cualquier entorno de programación; su proceso de validación de contenidos mediante el procedimiento de juicio de expertos y la administración a alumnos de Educación Primaria. Los análisis estadísticos preliminares muestran la prometedora consistencia del BCTt para evaluar el PC en Educación Primaria.

Por otro lado, y dado que los resultados de la experiencia, presentados en el apartado 4.3, indican que existe un efecto techo en los alumnos más mayores, también se ha desarrollado y validado una versión del test para estos alumnos: **Competence Computational Thinking Test (cCTt)**, en una colaboración con la *Ecole Polytechnique Fédérale de Lausanne (EPFL)*¹ en Suiza, y un análisis comparativo del BCTt y en cCTt en colaboración la EPFL y la ONG *Tree Tree 2*, en Portugal².

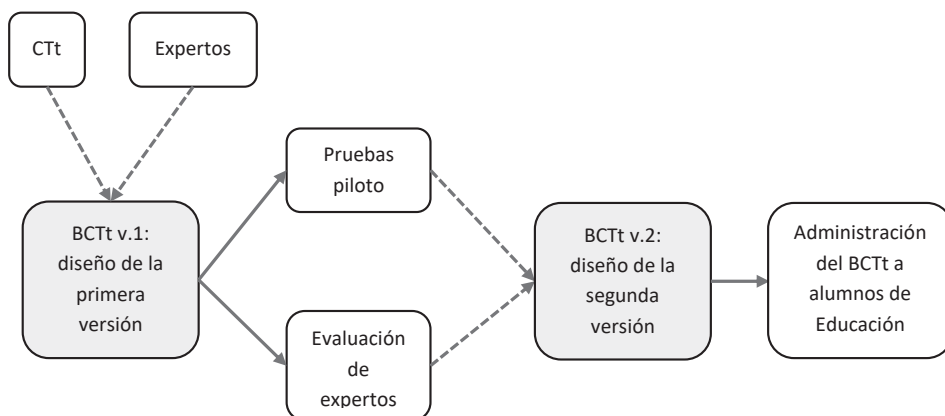
¹ <https://www.epfl.ch/about/>

² <https://www.treetree2.org/>

4.2. Procedimiento y diseño del BCTt

Se diseñó una primera versión del BCTt dirigida a alumnos de Educación Primaria. Basándose en el CTt, el test se adaptó tanto en su forma como en su contenido a los alumnos de entre 5 y 12 años, por lo que se realizaron varios cambios sustanciales. A continuación, el BCTt se sometió a una prueba piloto en pequeñas submuestras y fue evaluada por expertos en la materia. A partir de estos resultados preliminares, se introdujeron cambios para obtener una segunda versión mejorada que se administró a 299 estudiantes de tres colegios en España. Los resultados obtenidos se analizaron estadísticamente. En la Figura 12 se muestra el procedimiento de validación.

Figura 12. Procedimiento de validación del BCTt.



4.2.1. BCTt v.1

Como la población destinataria del BCTt (en principio niños de 5-6 a 12 años) es más joven que la de la CTt, hay que adaptarla tanto en la forma como en el contenido. Además, el BCTt incluye varias innovaciones que pretenden ser mejoras sustanciales. Se realizaron pruebas piloto en pequeñas submuestras ($n=3$ a 5 sujetos, de edades comprendidas entre los 5 y los 10 años) a lo largo del diseño inicial.

Esta versión inicial el BCTt tiene 25 ítems, con un tiempo estimado de 40 minutos para su resolución, tiempo que pareció ser adecuado en las pruebas piloto. Los ítems están diseñados con el menor texto posible y los símbolos pretenden ser auto explicativos para aumentar la legibilidad de la prueba en edades tempranas. La prueba está dirigida a niños pequeños que quizá no hayan adquirido aún la capacidad de leer y escribir, por lo que se diseñó incluyendo símbolos y dibujos que se explicaran por sí mismos. Teniendo en cuenta que también está dirigido a niños de las últimas etapas de Educación

Primaria que sí que han adquirido ya competencias de lecto-escritura, se incluyeron textos mínimos de ayuda que redundan en las explicaciones basadas en símbolos. La simbología es clara y pretende conectar emocionalmente con los niños para que se potencie el proceso de aprendizaje [156] ya que es un pollito que debe llegar hasta su madre, la gallina.

La primera versión de el BCTt está dividida en seis conjuntos de preguntas y cada conjunto está relacionado con un concepto computacional básico. Las preguntas son de tipo opción múltiple, inicialmente con tres alternativas de respuesta para cada ítem, que se plantean en dos diseños gráficos diferentes: tipo lienzo o laberinto. El tipo lienzo es un diseño de “seguir la línea de puntos” al que los niños de estas etapas educativas están acostumbrados, por su trabajo escolar cotidiano. El diseño tipo laberinto consiste en una matriz de cuadrados en la que el alumno debe trazar un camino desde la casilla de salida para alcanzar una casilla de meta, pasando de un cuadrado a otro en un orden determinado, evitando y/o recogiendo objetos por el camino, por ejemplo, otro pollito (véase la Figura 17).

En las preguntas de tipo laberinto se han incluido transiciones visuales entre los cuadrados (véase la Figura 13). Esto pretende ser una mejora sustancial este tipo de preguntas, ya que nuestra hipótesis es que las dificultades con este tipo de trazados en edades tempranas están relacionadas con la desorientación y las dudas sobre si la casilla actual y la de destino, en cada paso, deben formar parte de la secuencia del camino o ser ignoradas. Además, añadir transiciones convierte el laberinto en un diagrama de estados, un elemento principal en los algoritmos y la codificación que ha demostrado mejorar la capacidad de comprensión de los problemas [157]-[159].

Las alternativas de respuesta del BCTt v.1 están dispuestas en forma de secuencias de flechas gruesas, números y colores, dependiendo del concepto computacional implicado en cada pregunta. Cada respuesta tiene una disposición vertical arriba-abajo, y no horizontal de izquierda a derecha como en el CTt. Esta decisión se tomó teniendo en cuenta que la dirección de lectura de las secuencias de código es arriba-abajo. Además, la disposición arriba-abajo resultó ser una disposición adecuada en las pruebas piloto, aunque se encontraron problemas relacionados con la disposición de los ítems de tipo lienzo: si la línea de puntos se dibujaba de abajo a arriba, los alumnos tendían a leer las respuestas de abajo a arriba (véase la Figura 14). Este problema se resolvió evitando estas direcciones de dibujo (tal y como se muestra en la Figura 15).

El BCTt v.1 contiene los conceptos computacionales básicos del marco 3D, ordenados en dificultad creciente, de acuerdo con las etapas educativas objetivo (véase la Tabla 12): secuencias (6 ítems), bucles simples (5 ítems), bucles

anidados (7 ítems) y condicionales (7 ítems). En cada ítem del laberinto, la tarea requerida es llevar un pollito hasta su madre (la gallina) a través del laberinto que puede ser de pequeño o gran tamaño, permitiendo retos de diferente complejidad. Puede haber obstáculos que hay que evitar (un gato) u objetos que hay que recoger por el camino, como otro pollito. Las figuras 16 a 19 muestran ejemplos de las distintas tareas que tienen que resolver los niños, y en el anexo A.1 se pueden consultar todos los ítems de esta primera versión del test.

Figura 13. Laberinto sin transiciones (izquierda) y con transiciones entre las casillas (derecha), queda convertido en un diagrama de estados.

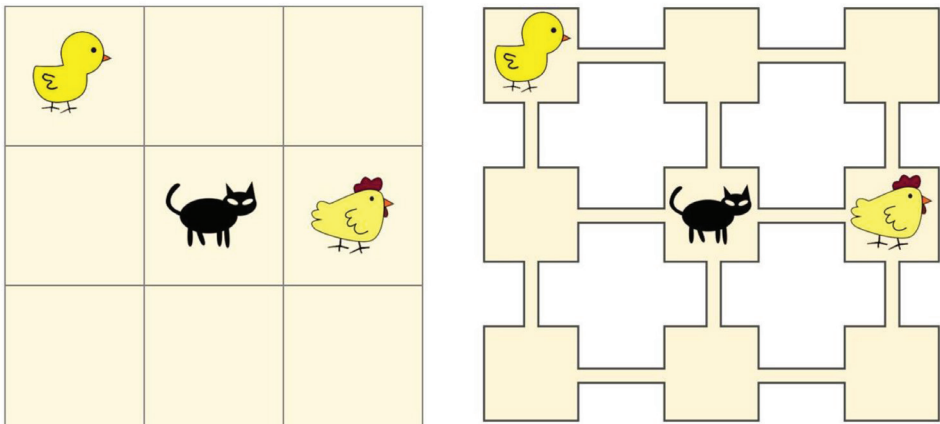


Figura 14. Ítem de tipo lienzo con problemas en la lectura. La dirección de la línea de puntos crea confusión sobre la dirección en la que deberían leerse las posibles respuestas.

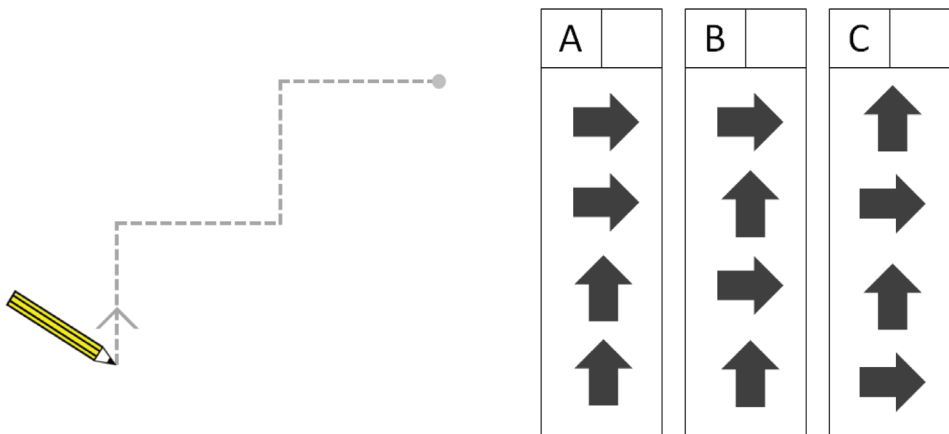


Figura 15. Ítem de tipo lienzo. La dirección de dibujo de la línea de puntos es la misma dirección en la que deben leerse las posibles respuestas.

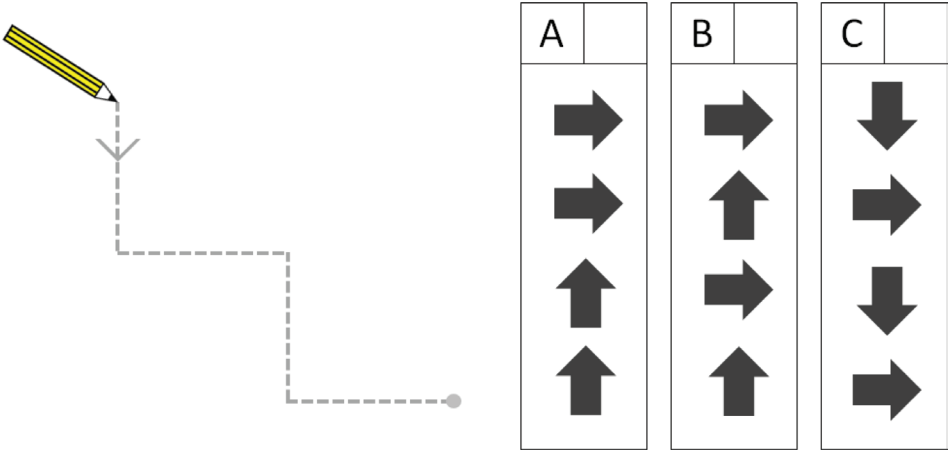


Figura 16. BCTt v.1. Ítem número 3.

3

Lleva al pollito con su mamá.
Cuidado con el gato.

Marca la secuencia correcta:

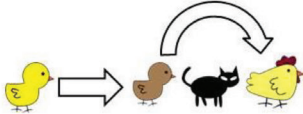
A	B	C
→	→	↓
→	↓	↓
↓	↓	→
↓		→

Tabla 12. Conceptos computacionales del marco 3D abordados en cada ítem del BCTt.

Item	Tipo	Tamaño	Obstáculos	Objetos a recoger	Conceptos computacionales				Condicionales		
					Secuencias	Bucles	Anidado	Simple	If- Then	If- Then- Else	While
1	Laberinto	Pequeño			x						
2	Lienzo		-	-	x						
3	Laberinto	Pequeño	x		x						
4	Laberinto	Pequeño	x	x	x						
5	Laberinto	Grande	x	x	x						
6	Lienzo		-	-	x						
7	Laberinto	Pequeño				x					
8	Laberinto	Pequeño				x					
9	Laberinto	Pequeño	x			x					
10	Laberinto	Grande				x					
11	Laberinto	Grande		x		x					
12	Laberinto	Grande					x				
13	Lienzo		-	-			x				
14	Laberinto	Grande	x				x				
15	Laberinto	Grande		x			x				
16	Laberinto	Grande		x			x				
17	Lienzo		-	-			x				
18	Laberinto	Grande	x	x			x				
19	Laberinto	Pequeño						x			
20	Laberinto	Grande						x			
21	Laberinto	Grande							x		
22	Laberinto	Grande							x		
23	Laberinto	Pequeño								x	x
24	Laberinto	Grande								x	x
25	Laberinto	Grande									x

Figura 17. BCTt v.1. Ítem número 18.

18



Lleva a los pollitos con su mamá.
Cuidado con el gato.

Marca la secuencia correcta:

A

2x





1x



B

2x



3x



1x



C

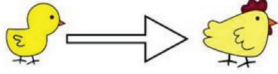
3x





Figura 18. BCTt v.1. Ítem número 21.

21



Lleva al pollito con su mamá:



Si el pollito está en una casilla sin nube, avanza una casilla a la derecha

Marca la secuencia correcta:

A







B





C



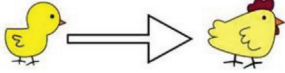




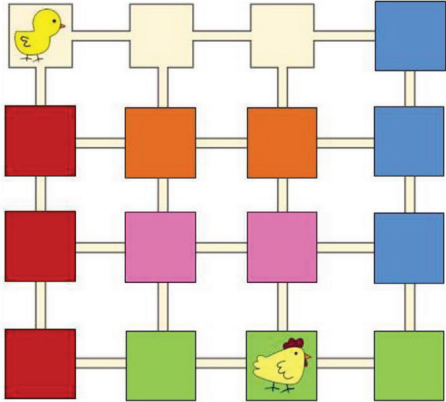
122

Figura 19. BCTt v.1. Ítem número 24.

24



Lleva al pollito con su mamá:





Mientras que el pollito esté en casillas azules, **siempre** avanza a la derecha

Marca la secuencia correcta:

A



B



C



4.2.2. Evaluación de expertos

Esta primera versión del BCTt se sometió a un procedimiento de evaluación en el que se consultó a 45 expertos, tanto profesionales de la informática y la educación, como profesores de Educación Primaria. Los datos sobre el perfil de los expertos se muestran en la Tabla 13.

En primer lugar, los expertos completaron el BCTt y después fueron consultados, a través de un formulario en línea de 66 ítems o preguntas cuyos resultados se resumen en la Tabla 14, sobre la dificultad y la adecuación de las preguntas en cuanto a su relevancia para medir cada uno de los diferentes conceptos computacionales incluidos en el test. Además, también se preguntó a los expertos sobre la longitud, la simbología utilizada y otras cuestiones. También se realizó una encuesta específica sobre las transiciones incluidas en las preguntas tipo laberinto. Por último, se recogieron los comentarios y sugerencias de mejora de los expertos.

123

Tabla 13. Perfiles profesionales de los expertos.

Grupo o grupos profesionales (se permite respuesta múltiple)	Número de expertos
Profesional de la informática	21
Profesor de informática	14
Profesor de Educación Primaria	9
Profesor de Educación Infantil	1
Profesor de universidad	9
Sin respuesta	4
Edad	
menos de 30	3
de 31 a 50	37
Más de 51	1
Sin respuesta	4
Género	
Mujer	13
Hombre	28
Sin respuesta	4
Nivel de conocimientos en metodologías de enseñanza de la informática	
Muy bajo	8
Bajo	2
Medio	9
Alto	13
Muy alto	9
Sin respuesta	4

Tabla 14. Resumen del cuestionario para la validación de los expertos.

Número de ítem del formulario	Concepto abordado	Elemento valorado por los expertos	Tipo de pregunta en el formulario
1 a 12	Secuencias	Dificultad percibida en cada pregunta del test y en el concepto en general	Opción múltiple y escala Likert
13	Secuencias	Relevancia del concepto para evaluar el PC	Escala Likert

Número de ítem del formulario	Concepto abordado	Elemento valorado por los expertos	Tipo de pregunta en el formulario
14 a 23	Bucle simple	Dificultad percibida en cada pregunta del test y en el concepto en general	Opción múltiple y escala Likert
24	Bucle simple	Relevancia del concepto para evaluar el PC	Escala Likert
25 a 38	Bucle anidado	Dificultad percibida en cada pregunta del test y en el concepto en general	Opción múltiple y escala Likert
39	Bucle anidado	Relevancia del concepto para evaluar el PC	Escala Likert
40 a 43	If-then	Dificultad percibida en cada pregunta del test y en el concepto en general	Opción múltiple y escala Likert
44	If-then	Relevancia del concepto para evaluar el PC	Escala Likert
45 a 48	If-then-else	Dificultad percibida en cada pregunta del test y en el concepto en general	Opción múltiple y escala Likert
49	If-then-else	Relevancia del concepto para evaluar el PC	Escala Likert
50 a 55	While	Dificultad percibida en cada pregunta del test y en el concepto en general	Opción múltiple y escala Likert
56	While	Relevancia del concepto para evaluar el PC	Escala Likert
57 a 60	Datos perfil	Datos sobre el perfil profesional del experto	Texto
61	Transiciones	Preferencia del laberinto con o sin las transiciones	Dicotomía
62	Transiciones	Justificación de la preferencia anterior	Texto
63	Longitud	Evaluación sobre la longitud del test	Escala Likert
64	Adecuación	Evaluación sobre la adecuación del test a la edad objetivo	Escala Likert
65	Interfaz	Evaluación sobre la simbología, aspecto gráfico y experiencia de usuario	Escala Likert
66	Generales	Comentarios finales y sugerencias	Texto

En la Figura 20 se muestran las puntuaciones obtenidas por los expertos en el BCTt dividiendo los ítems del test en conjuntos de conceptos computacionales y contando cuántos expertos respondieron correctamente los ítems de cada conjunto (por ejemplo, una puntuación de 41 en un concepto significa que 41 de 45 expertos respondieron correctamente en las preguntas relacionadas con ese concepto). Se observa que las puntuaciones disminuyen a lo largo del test, como era de esperar, ya que las preguntas se ordenaron en supuesta dificultad creciente. Esta hipótesis queda confirmada ya que los expertos consideran que los ítems aumentan en cuanto a dificultad a medida que se avanza en el test, como puede observarse en la Figura 21.

Figura 20. Eje de ordenadas: Puntuación obtenida en el test (por ejemplo, 41 significa que 41 de 45 expertos respondieron correctamente); por concepto computacional (eje de abscisas: 1. Secuencias, 2. Bucle simple, 3. Bucle anidado, 4. If-then, 5. If-then-else, 6. While).

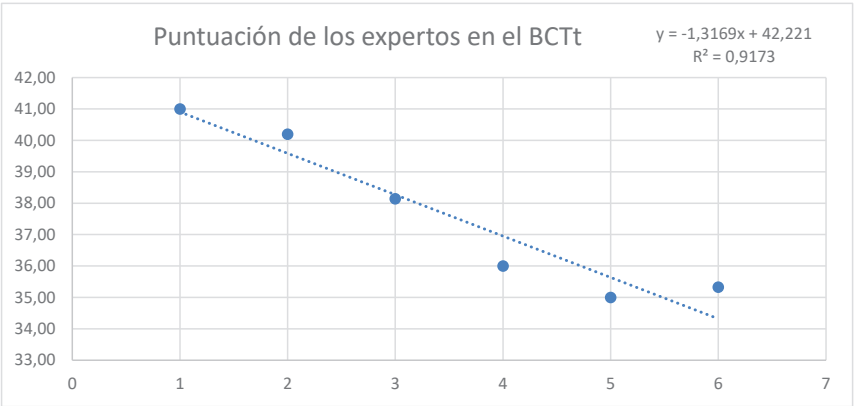
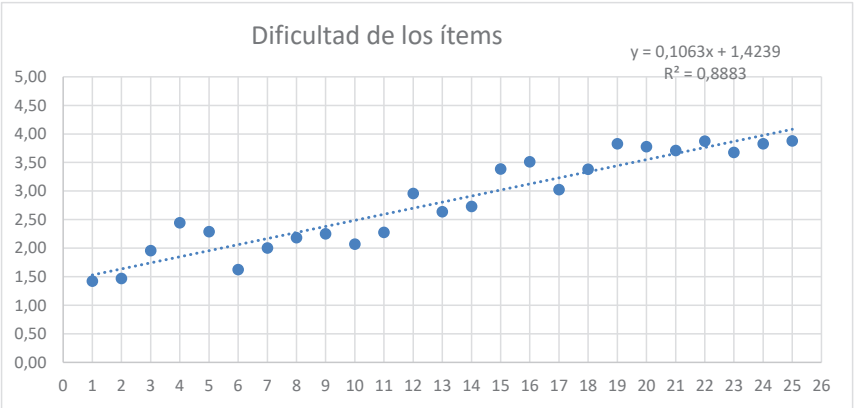
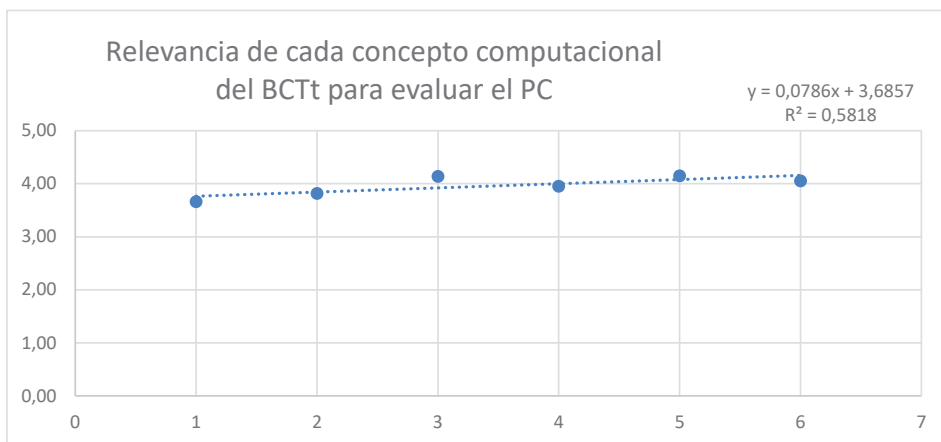


Figura 21. Dificultad de los ítems del BCTt v.1 percibida por los expertos (eje de ordenadas: escala Likert de 1 a 5), por ítem del BCTt (eje de abscisas).



La relevancia para medir el PC crece de forma similar a lo largo de los ítems del test (véase la Figura 22), donde cada concepto computacional tiene una relevancia percibida media o alta. En concreto, los expertos consideran que las preguntas del BCTt miden los conceptos computacionales abordados y valoran los bucles anidados como el concepto más relevante de todos (relevancia para medir el PC, escala Likert de 1 a 5: Secuencias=3,66; Bucles simples=3,82; Bucles anidados=4,14; If-then=3,95; If-then-else=4,15; While=4,05). En cuanto a si el test en su conjunto evalúa por completo la dimensión de conceptos computacionales del PC en alumnos de Educación Primaria, el 73,1% de los expertos consideró que completamente o casi completamente (escala Likert de 1 a 5: 34,1%: completamente, 39%: casi completamente, 22%: a medias, 4,9%: un poco, 0%: nada).

Figura 22. Relevancia del concepto computacional del BCTt para medir el PC, percibida por los expertos (eje de ordenadas: escala Likert de 1 a 5) por concepto computacional (eje de abscisas: 1. Secuencias, 2. Bucle simple, 3. Bucle anidado, 4. If-then, 5. If-then-else. 6. While).



La mayoría de los expertos consultados (68,3%) consideraron que el número de ítems (longitud del test) era adecuada, el 24,4% estimó que el test contenía demasiadas preguntas. Sólo un 7,3% añadiría más preguntas. Además, en cuanto a la simbología utilizada, aspecto gráfico y experiencia de usuario en alumnos de Educación Primaria (escala Likert de 1 a 5), un 75,6% de expertos consideran que el BCTt es adecuado en este aspecto (39%: muy adecuado, 36,6%: adecuado, 22%: medianamente adecuado, 2,4%: poco adecuado, 0%: nada adecuado).

Una de las mejoras sustanciales del BCTt fue la inclusión de transiciones en las preguntas del laberinto. En este sentido, los expertos se mostraron muy de acuerdo con la mejora y consideraron que ayudaría a los niños, sobre todo a los más pequeños, a comprender mejor las respuestas del test. Algunas

respuestas recogidas fueron: “las transiciones se asocian fácilmente a las flechas en las respuestas”, “las transiciones incorporan aristas a los laberintos, a modo de diagrama de estados, con una mejor comprensión del problema”, “los caminos permitidos son claros, excluyendo los movimientos diagonales”, “al incluir transiciones se distingue claramente el movimiento y el lugar de llegada. En el diseño sin transiciones, se generan dudas sobre cuándo un personaje llega a otro (¿es cuando llega a la casilla anterior o cuando llega a la casilla del otro personaje?)”.

Los comentarios y sugerencias de los expertos se tuvieron en cuenta para diseñar y mejorar la siguiente versión del BCTt. Por ejemplo, algunos de sus comentarios más frecuentes se referían a la necesidad de explicar primero oralmente a los niños el significado de cada conjunto de preguntas, de añadir una respuesta más a cada pregunta, de sustituir los elementos elegibles (pollitos) por otro símbolo que no induzca a confusión, o de reformular las preguntas relativas a algunos conceptos computacionales para acercarse a su definición exacta. Se reproducen a continuación las sugerencias más relevantes:

- “Es necesario un ejemplo oral explicativo de cada tipo de ítem”.
- “En cuanto al número de alternativas de respuesta, sugiero 4 alternativas en lugar de 3, ya que sólo 3 alternativas de respuesta por ítem pueden influir negativamente en la fiabilidad total del test”.
- “Los niños podrían intentar pasar por la casilla del gato, saltándolo: se necesita una indicación expresa de no tocar el gato”.
- “No está claro si dos pollitos pueden moverse juntos después de encontrarse”.
- “Las últimas preguntas (condicionales “while”) necesitan más explicación”.
- “Los ítems if-else y if-then-else no se corresponden exactamente con el concepto computacional”.
- Seis de los nueve profesores de Educación Primaria, consideraron el test demasiado difícil para los alumnos de esta edad. Esta impresión de los expertos se refutó más tarde, cuando el BCTt se administró a alumnos de Educación Primaria de varios colegios.

4.2.3. BCTt v.2

Teniendo en cuenta los resultados de la evaluación de expertos, se diseñó la segunda versión del BCTt con varias modificaciones y adiciones. Por ejemplo, se añadió una alternativa más de respuesta a cada pregunta, se refinaron los

enunciados de las preguntas, se sustituyeron los elementos que se pueden recoger en los ítems de tipo laberinto por otros, y se reformularon las preguntas de algunos de los conjuntos para acercarse más fielmente a la definición formal del concepto computacional. Uno de los cambios más notables fue el refuerzo de los colores en las preguntas con una simbología de formas, para permitir que los alumnos con problemas en la visualización de los colores puedan realizar la prueba. Esta mejora hace que el test pueda imprimirse en blanco y negro. A continuación, se enumeran los cambios más relevantes:

- Se añade una respuesta a cada ítem, de forma que cada pregunta contiene 4 respuestas alternativas en lugar de 3, para reducir la probabilidad de responder correctamente al azar (por ejemplo, Figura 23).
- En los ítems que contienen un gato que hay que evitar, se especifica que no se cruza la casilla ocupada por el gato (por ejemplo, Figura 23).
- Los ítems que incluían otro pollito resultaban ambiguos ya que no quedaba claro si los dos pollitos debían continuar juntos después de encontrarse o no, por lo que en el BCTt v.2, el segundo pollito se sustituye por una flor para recoger (por ejemplo, Figura 24).
- Los ejemplos de significado que se incluyen en cada ítem se clarifican en las últimas preguntas (por ejemplo, Figura 25).
- Los ítems if-else y if-then-else se reformulan para una mejor correspondencia con el concepto computacional (por ejemplo, Figura 25).
- Para garantizar que los alumnos con dificultades visuales asociadas al color puedan leer los símbolos de cada ítem, se asocia una forma específica a cada color diferente (por ejemplo, triángulo y azul). Esta mejora permite imprimir el test en formato blanco y negro (por ejemplo, Figura 26).
- Se elabora un protocolo de administración en el que se especifica que se debe dar una explicación oral a los alumnos antes de realizar la prueba, con un ejemplo explicativo de cada uno de los conceptos computacionales que se abordan en la prueba. El protocolo completo puede encontrarse en el anexo A.3.
- Antes de realizar la prueba, se debe explicar oralmente a los alumnos un ejemplo de ítem por cada uno de los 6 conceptos computacionales diferentes que se contemplan en el test, estos ejemplos se incluyen en el protocolo de actuación y también como archivo separado en forma de presentación de diapositivas "paso a paso" (por ejemplo, Figura 27). El archivo completo puede encontrarse en el anexo A.4.

Figura 23. BCTt v.2. Ejemplo de ítem (ítem número 3).

3



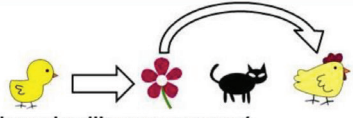
Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A	B	C	D
→	→	↓	↓
→	↓	↓	↓
↓	↓	→	→
↓		→	↑

Figura 24. BCTt v.2. Ejemplo de ítem (ítem número 18).

18



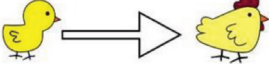
Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

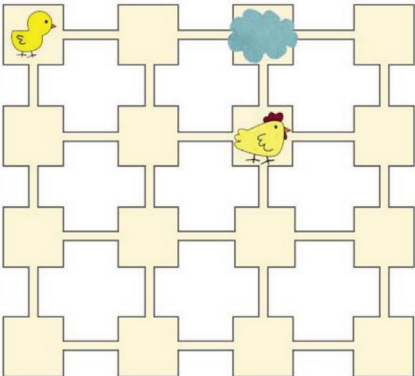
A	B	C	D
2x → 1x ↑	2x ↑ 3x → 1x ↑	3x → 1x ↑	3x → 3x ↑

Figura 25. BCTt v.2. Ejemplo de ítem (ítem número 21).

21



Lleva al pollito con su mamá:



Ejemplo de significado:



Si el pollito está en una casilla con nube, avanza una casilla a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

A



B



C

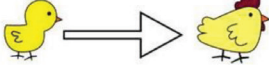


D

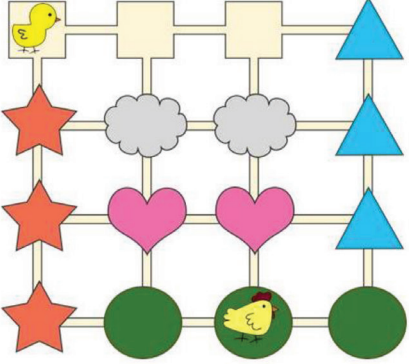


Figura 26. BCTt v.2. Ejemplo de ítem (ítem número 24).

24



Lleva al pollito con su mamá:



Ejemplo de significado:



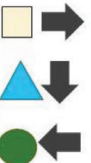
Mientras que el pollito esté en casillas triángulo, siempre avanza a la derecha

Marca la secuencia correcta:

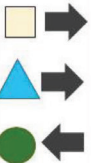
A



B



C

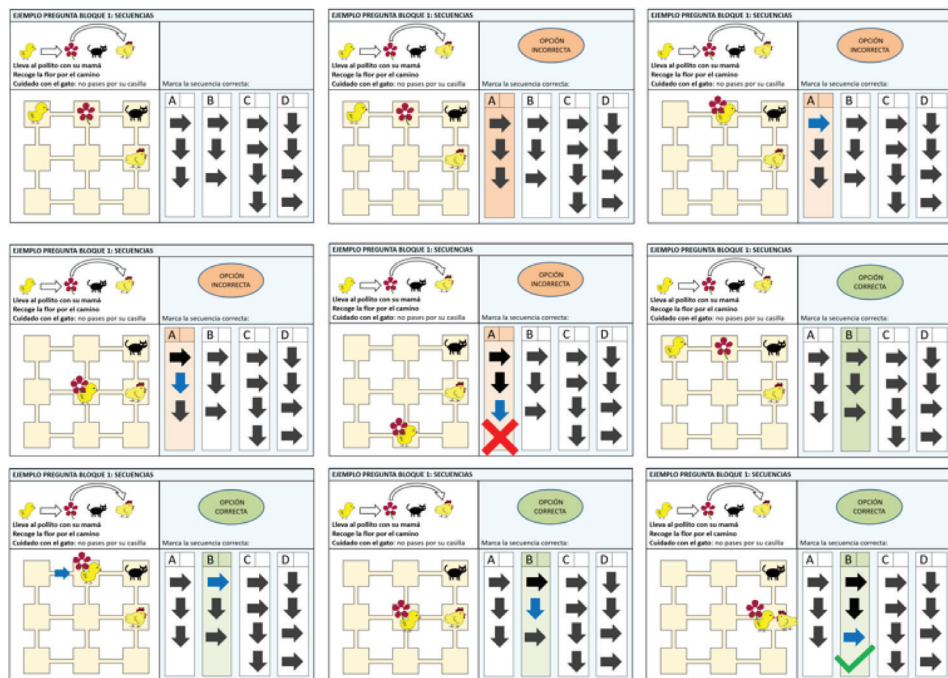


D



131

Figura 27. Ejemplo de ítem relativo al concepto de secuencias “paso a paso”.



4.2.4. Administración del BCTt v.2

La segunda versión del BCTt se administró, siguiendo el protocolo de actuación, a una muestra de N=299 estudiantes de Educación Primaria (edades comprendidas entre los 5 y los 12 años) de tres colegios españoles. En cada colegio, la investigación se centró en una etapa educativa.

Se realizaron dos variaciones diferentes de la segunda versión del BCTt, una incluyendo las transiciones entre las casillas (variación 1), y otra sin ellas (variación 2), para poder comparar el rendimiento de los alumnos con y sin esta ayuda.

La muestra de alumnos se dividió en varias submuestras, como se muestra en la Tabla 15, teniendo en cuenta la edad de los alumnos, la variación del test (con o sin transiciones) que realizarían y el grupo escolar al que pertenecían. La variación 1 del test (con transiciones) se administró a las submuestras A1, B1, C1, D1, E1 y F1. La variación 2 del test (sin transiciones), se administró a las submuestras B2, D2 y F2. Una de las submuestras (D1) fue sometida a una segunda prueba 5 semanas después.

La investigación se llevó a cabo en las mismas condiciones en cada colegio, ya que se siguió el protocolo de actuación, de esta forma, antes de realizar la

prueba, se explicó oralmente delante de los alumnos un ejemplo de un ítem de cada uno de los 6 conceptos computacionales diferentes.

Las pruebas se administraron simultáneamente a todos los sujetos de cada submuestra. Para garantizar que las habilidades de los estudiantes o su experiencia previa en el uso de dispositivos informáticos no interfirieran en los resultados, los test se imprimieron y rellenaron por los estudiantes individualmente en papel. Además, las pruebas se imprimieron en escala de grises, para que fueran accesibles a los alumnos con dificultades visuales asociadas al color.

Tabla 15. Submuestras de administración del BCTt en centros de Educación Primaria (n: número de estudiantes).

Ciclo de Educación Primaria	Curso	Identificador de submuestra	BCTt variación 1 (con transiciones)	BCTt variación 2 (sin transiciones)
1	1	A	A1: n=52	
	2	B	B1: n=18	B2: n=18
2	4	C	C1: n=54	
	4	D	D1: n=28	D2: n=28
3	5	E	E1: n=51	
	6	F	F1: n=25	F2: n=25

4.3. Resultados y discusión

El BCTt v.2 se administró a alumnos de Educación Primaria para analizar empíricamente la adecuación del diseño, realizar un análisis estadístico de los ítems y comprobar su fiabilidad.

4.3.1. Transiciones

Las transiciones entre casillas en las preguntas de tipo laberinto pretendían ser una mejora sustancial del BCTt. Para comprobarlo, se compararon las puntuaciones (suma de las preguntas respondidas correctamente en el test) entre la variación 1 del BCTt (con transiciones) y la variación 2 del BCTt (sin transiciones). Como puede verse en la Tabla 16, los resultados indican que no hay diferencias significativas entre las puntuaciones de las variaciones del test obtenidas en las muestras de alumnos a partir de cuarto curso, pero sí hay una diferencia muy significativa entre las puntuaciones de los niños de cursos inferiores ($p=0,005<0,01$), lo que indica que esta ayuda es muy notable para los niños más pequeños, pero no así para los mayores.

Tabla 16. Estadística de submuestras y prueba t de Student para comparar las variaciones del BCTt (1: con transiciones, y 2: sin transiciones).

Curso	Submuestra	Versión del BCTt	n	Mean	Std. Dev.	t-test for Equality of Means	
						t	Sig.
2	B1	1	18	16.778	2.487	3.042	0.005
	B2	2	18	14.278	2.445		
4	D1	1	28	21.357	2.438	0.122	0.904
	D2	2	28	21.286	1.922		
6	F1	1	25	21.720	2.622	0.499	0.620
	F2	2	25	21.280	3.542		

4.3.2. Estadística descriptiva

Considerando la puntuación en el BCTt como la suma de respuestas correctas de cada alumno a lo largo de los 25 ítems del test, y buscando un equilibrio entre el número de sujetos en cada submuestra y los ciclos de Educación Primaria, se realizó un análisis estadístico de los resultados de la puntuación del test en una submuestra de cada curso: submuestras A1, B1, C1, E1 y F1 (N=200).

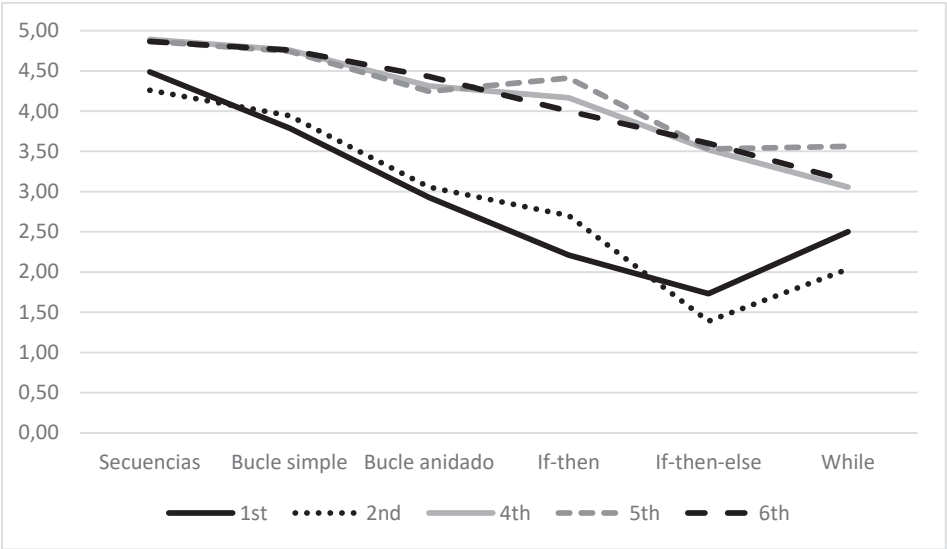
Los resultados de este análisis se muestran en la Tabla 17. El análisis preliminar de toda la muestra revela una media global alta (19,9150), y las puntuaciones a lo largo de las submuestras sugieren que la prueba podría estar dirigida a las primeras etapas educativas de primaria, ya que no se muestran diferencias significativas entre las medias de 4º y 5º curso (prueba t de Student, $t=0.485$, $p=0.63 > 0.05$) ni entre las medias de 5º y 6º curso (prueba t de Student, $t=0.193$; $p=0.85 > 0.05$).

No obstante, se realizó un segundo análisis más profundo, esta vez, dividiendo los ítems del BCTt según los conceptos computacionales, y contando cuántos alumnos contestaron correctamente los ítems de cada conjunto (es decir, si la submuestra es $n=23$, 21 puntos de puntuación en un ítem significan que 21 de los 23 alumnos lo contestaron correctamente). Esta puntuación por ítem, relacionada con cada concepto computacional, muestra resultados interesantes, ya que con respecto a los ítems relacionados con los conceptos de bucles anidados y condicionales se obtienen puntuaciones bajas a lo largo de todos los cursos como se muestra en la Figura 28, pero con respecto a las secuencias y los bucles simples se obtienen puntuaciones muy altas en 5º y 6º grado, lo que nos lleva a concluir que las primeras preguntas del test podrían ser demasiado fáciles para los alumnos a partir de 3º de primaria, y que podría ser necesario añadir a la prueba ítems más difíciles para esta etapa.

Tabla 17. Puntuaciones en el BCTt. Estadísticas por curso.

Submuestra	Muestra completa	A1	B1	C1	E1	F1
Curso	1 a 6	1	2	4	5	6
N	200	52	18	54	51	25
Media	19.92	16.52	16.78	21.57	21.84	21.72
Mediana	20.00	16.00	18.00	23.00	23.00	22.00
Desv. Estándar	3.79	3.31	2.49	3.044	2.61	2.62
Varianza	14.36	10.96	6.183	9.268	6.815	6.88
Mínimo	8.00	8.00	11.00	14.00	13.00	15.00
Máximo	25.00	24.00	20.00	25.00	25.00	25.00
Percentiles 25	17.00	14.00	15.75	19.00	20.00	19.50
50	20.00	16.00	18.00	23.00	23.00	22.00
75	23.00	19.00	18.00	24.00	24.00	24.00

Figura 28. Puntuación en el BCTt por curso y concepto. Eje de abscisas: concepto computacional. Eje de ordenadas: Puntuación del ítem BCTt, normalizado de 0 a 5: 5 puntuación máxima.



La puntuación por ítem (o índice de dificultad) confirma empíricamente la dificultad progresiva anticipada por la evaluación de los expertos (índice de dificultad medio = 0,81) considerando toda la muestra (véase la Figura 29).

Asimismo, los resultados, aislando a los alumnos más jóvenes: primer ciclo de Educación Primaria (submuestras A1 y B1, N=70), muestran una dificultad creciente a lo largo de los elementos (N=25 ítems; Mínimo=0,27; Máximo=0,96; índice de dificultad medio =0,70; puntuación media total = 16,59) (véase la Figura 30).

Figura 29. Índice de dificultad del ítem (eje de ordenadas) para cada ítem del BCTt (eje de abscisas).

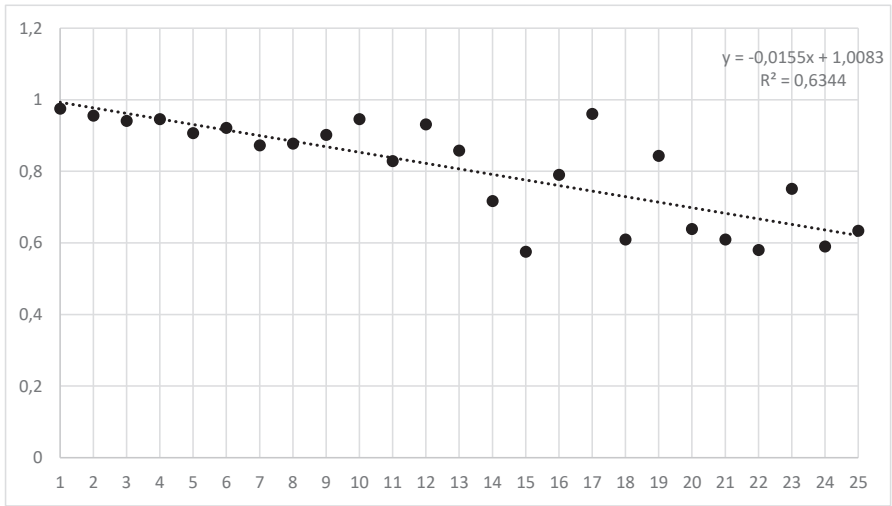
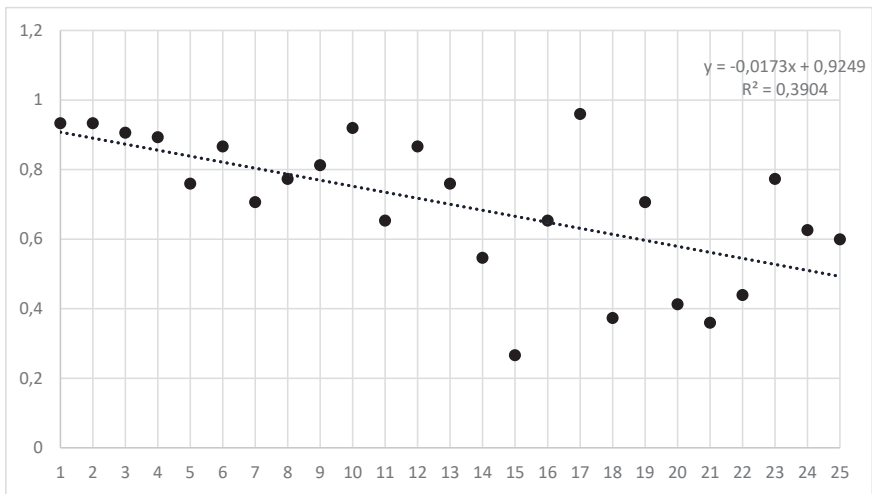
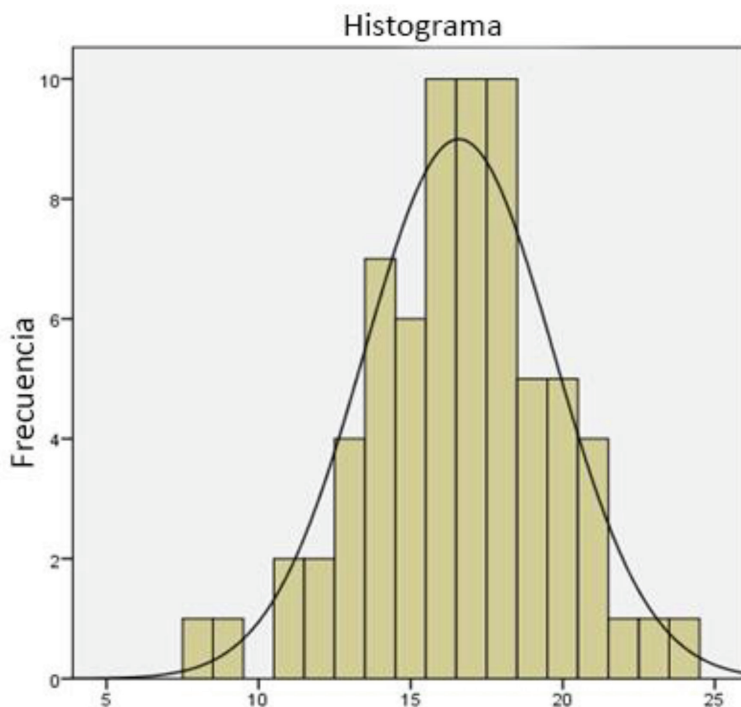


Figura 30. Índice de dificultad del ítem (eje de ordenadas) para cada ítem del BCTt (eje de abscisas), en la primera etapa educativa (primer y segundo curso de Educación Primaria).



El histograma que muestra la distribución de la puntuación del BCTt a lo largo de las submuestras de 1º y 2º curso (véase la Figura 31), se ajusta a la curva normal y es bastante simétrico (media=16,59; desviación estándar=3,104; n=70), lo que sugiere que el BCTt está equilibrada en cuanto a la dificultad de sus ítems para el primer ciclo de Educación Primaria.

Figura 31. Histograma de puntuaciones en el BCTt para el primer ciclo de Educación Primaria.



4.3.3. Fiabilidad

Para evaluar la consistencia interna asociada a las puntuaciones del BCTt, considerando todos los cursos (N=200), se ha realizado un análisis de fiabilidad. El Alfa de Cronbach es $\alpha=0,824$ (véase la Tabla 18), que puede considerarse como una fiabilidad muy buena [160]. Los resultados de fiabilidad por curso muestran un Alfa de Cronbach más bajo cuanto más alto es el curso (véase la Tabla 19), lo que nos lleva a concluir que el BCTt se dirige principalmente a las primeras etapas de primaria (cursos de 1 a 4) donde muestra una mayor fiabilidad.

Tabla 18. Estadísticas de fiabilidad de la muestra completa.

Muestra N	N de Ítems	Estadísticas de fiabilidad		Estadísticas de ítem			
		Cronbach's Alpha	Cr. 's Alpha Based on Stand. Items	Mean	Min.	Max.	Variance
200	25	0.824	0.829	0.807	0.576	0.976	0.021

Tabla 19. Estadísticas de fiabilidad del BCTt por curso.

Submuestras				Estadísticas de fiabilidad		Estadísticas de ítem	
Ciclo de Educación Primaria	Curso	Id.	n	Cronbach's Alpha	Cr. 's Alpha Based on Stand. Items	Mean	Variance
1	1	A1	52	0.833	0.838	0.742	0.041
1	2	B1	18	0.793	0.801	0.630	0.042
2	4	C1	54	0.771	0.735	0.837	0.022
3	5	E1	51	0.660	0.683	0.863	0.012
3	6	F1	25	0.657	0.648	0.844	0.015

Se realizó un segundo análisis de fiabilidad. Esta vez mediante el método de test y re-test con la muestra D1 ($n=28$). El BCTt se administró a los sujetos de D1 (2ª ciclo, 4º curso) y, 5 semanas después, se volvió a administrar en las mismas condiciones a los mismos sujetos. Como las puntuaciones del BCTt no se distribuyen normalmente en las submuestras de D1 (prueba de normalidad de Shapiro-Wilk: test Sig.=0,03; re-test Sig.=0,01), se utilizó la prueba no paramétrica de Spearman, que mostró una correlación muy fuerte, positiva y significativa ($r_s=0,93$; $p<0,01$). Por lo tanto, se encontró una excelente fiabilidad como estabilidad para el BCTt en esta submuestra.

4.4. Conclusiones sobre la validación del BCTt

Los expertos consideraron el diseño del BCTt como adecuado para la evaluación del PC en alumnos de Educación Primaria, tanto en la forma como en el contenido, y con mejoras relevantes con respecto al CTt como las transiciones entre casillas. Además, los expertos opinaron que los ítems del test parecían estar ordenados en dificultad creciente y relevancia. Estas consideraciones

se confirmaron en el análisis estadístico de cuando se administró el test a alumnos de Educación Primaria.

Las transiciones entre casillas se muestran como una mejora relevante en los trazados de los laberintos para los alumnos más jóvenes (primer ciclo de Educación Primaria), dando lugar a puntuaciones superiores muy significativas en comparación con los alumnos que realizaron la variación del test sin transiciones. Sin embargo, los alumnos de 4º y 6º curso no se benefician de la inclusión de transiciones ni se vieron afectados negativamente por ellas. Esto nos lleva a la conclusión de que las transiciones son una mejora que se puede incluir en este tipo de problemas que funciona un andamiaje importante para los alumnos más jóvenes sin afectar negativamente a los mayores.

La media global de toda la muestra y las puntuaciones del BCTt de las submuestras sugieren que el test podría estar dirigido a las primeras etapas educativas de primaria, ya que se muestran medias elevadas en las submuestras del segundo y tercer ciclo, y no se muestran diferencias significativas entre las medias de las puntuaciones totales en los alumnos de mayor edad. Así, el BCTt parece estar dirigida a los cursos 1º a 4º y especialmente a los cursos 1º y 2º, ya que está equilibrada en cuanto a la dificultad de sus ítems. Sin embargo, existen diferencias significativas entre los cursos en las medias de las puntuaciones de las submuestras de todos los ciclos de Educación Primaria a partir de la pregunta 12, lo que nos lleva a concluir que la prueba podría estar dirigida a todas las etapas de primaria, pero la primera parte de la prueba podría incluir ítems demasiado fáciles para los alumnos de mayor edad, al contrario de lo que se esperaba según la opinión de los expertos, por lo que incluir más ítems con una dificultad mayor para los alumnos de tercer ciclo podría ser una mejora a considerar para una futura versión del test.

El BCTt demostró una alta fiabilidad en toda la muestra con un Alfa de Cronbach = 0,824, sin embargo, se obtuvo un coeficiente mayor en los alumnos más jóvenes que en los mayores. Como era de esperar, el test es más fiable en 1º y 2º curso que en los superiores, ya que el nivel de dificultad se ajusta mejor a los cursos inferiores. Así, el BCTt, al igual que el CTt es un instrumento autónomo, independiente de cualquier entorno, que se ha revelado fiable para la evaluación del PC en Educación Primaria y puede ser administrado como pre-test y post-test en las investigaciones que lo requieran.

A partir de esta investigación se puede concluir que el BCTt puede ser utilizado en alumnos de Educación Primaria, particularmente en los primeros cursos (5 a 7 años), ya que los resultados de los alumnos de mayor edad (8 a 12 años) revelan que el BCTt podría ser demasiado fácil para los alumnos de los cursos más altos, aunque puede ser utilizado centrándose sólo en los

ítems más complejos del test. Por lo tanto, el BCTt puede considerarse una extensión fiable del CTt de Román *et al.* para estudiantes más jóvenes, ya que el CTt está dirigida a estudiantes de 10 a 16 años.

4.5. Limitaciones del BCTt y líneas de actuación

Una de las limitaciones del BCTt es que, en relación con el marco 3D, se centra en los conceptos computacionales, sólo parcialmente en las prácticas computacionales e ignora las perspectivas computacionales. Por este motivo, se recomienda su uso en paralelo con otros instrumentos de evaluación para cubrir sus limitaciones, dentro de un *sistema de evaluaciones* [161], junto con otros instrumentos que evalúan otras dimensiones del PC, como se ha visto en el marco teórico. Este aspecto se tiene en cuenta en experiencias posteriores y se contempla en capítulos posteriores de esta tesis.

Por otra parte, es necesario establecer el límite inferior de edad en el que el BCTt puede ser administrado. En este sentido, se ha realizado una experiencia, en colaboración con la *Fontys University of Applied Science* y la *Zuyd University of Applied Science*, ambas de Países Bajos, que se describe en el Capítulo 5 y concluye en que el BCTt muestra una fiabilidad muy alta en alumnos de 4 años (α de Cronbach = 0,802) y 5 años (α de Cronbach = 0,889).

En cuanto al límite superior de edad para la administración del BCTt, a través de la experiencia realizada se demuestra que el BCTt tiene muy buena fiabilidad para la evaluación del PC en niños hasta los 8 años, pero en niños a partir de esta edad se produce un efecto techo. En este sentido, y dado que el CTt está dirigido al rango de edad de los 10 a los 16 años, se plantea el desarrollo y validación de una adaptación del BCTt que muestre una muy buena fiabilidad en el rango de edad en el que ésta decrece.

Por este motivo, se ha desarrollado y validado una versión del test que ha demostrado ser muy fiable en el rango de los 7 a los 9 años: **Competence Computational Thinking Test (cCTt)**, en una colaboración con la *Ecole Polytechnique Fédérale de Lausanne (EPFL)* en Suiza, en la que se realizó un estudio a principios de 2021 con 37 expertos y 1519 alumnos de estos cursos en centros escolares suizos. Después, se realizó un análisis comparativo del BCTt y en cCTt en colaboración la EPFL y la ONG *Tree Tree 2*, en Portugal, para concretar cuándo debe administrarse uno u otro test. Estas experiencias se resumen en el siguiente apartado.

De esta forma, se pretende avanzar en el objetivo OC: *Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones.*

4.6. Competent Computational Thinking Test (cCtt)

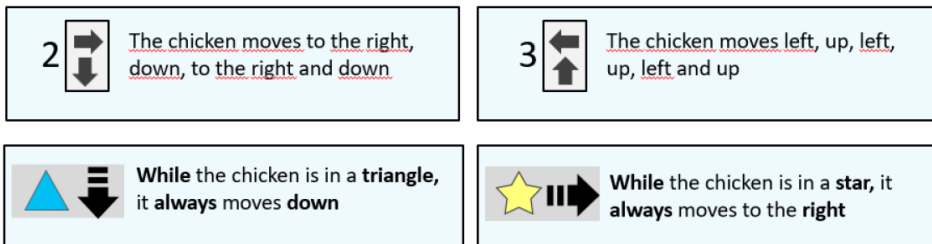
Dado que las primeras preguntas del BCTt resultaron demasiado fáciles para los alumnos a partir de 8 años, y se observó un efecto techo, es necesaria una nueva adaptación del test para estos cursos, incluyendo preguntas más difíciles.

Para ello se realizó el desarrollo y validación del cCtt a través de una experiencia en colegios, en una colaboración liderada por los siguientes equipos de la EPFL en Suiza: *MOBOTS Group of the BIOROB Laboratory*; *LEARN – Center for Learning Sciences*; y *Computer Human Interaction in Learning and Instruction (CHILI)*.

4.6.1. Diseño y validación del cCtt

Al igual que el BCTt, el cCtt es un test que se compone de 25 preguntas de dificultad progresiva. La mayoría de las preguntas (21 de 25) son de tipo laberinto y el resto, de tipo lienzo. Cada pregunta se presenta con cuatro posibles respuestas entre las que los alumnos deben elegir. El test aborda los conceptos de secuencias, bucles simples, bucles anidados, condicionales (if-then y if-then-else) y sentencias while, con un número variable de preguntas por concepto.

El diseño del cCtt, incluye adaptaciones que tienen como objetivo hacer la prueba más compleja, para ello, se emplean principalmente laberintos de cuadrícula grande; se eliminan las preguntas de baja dificultad que producían efecto techo en la experiencia anterior; se añaden preguntas relacionadas con conceptos complejos; se crea un nuevo subconjunto de preguntas con combinaciones de conceptos (véase la Figura 32); y se altera la disposición de los objetos (punto de partida, objetivos y obstáculos), y/o la selección de respuestas, de modo que la identificación de la respuesta correcta requiera más reflexión por parte de los estudiantes.



La validación del cCtt se realizó en dos fases, mediante un procedimiento similar al utilizado en la validación del BCTt descrita anteriormente. En primer

lugar, se realizó una evaluación por 37 expertos, procedentes del panel de expertos que participaron en la validación del BCtT, además de nuevos investigadores expertos en el PC, y de profesionales que participan en la reforma de la educación digital en Suiza (expertos del departamento de educación de la EPFL, profesores universitarios del grado de formación del profesorado, y profesores de Educación Primaria que enseñan informática o PC en el grupo de edad objetivo). Los resultados de la evaluación por expertos indicaron que el test tenía una buena validez constructo y de contenido, además de sugerir modificaciones en diseño del test.

Figura 32. Ejemplos de interpretación de los pictogramas del test, y pregunta número 22 del cCTt, que combina varios conceptos diferentes (versión en inglés).

24

Take the chicken to his mother

For example:

While the chicken is in a triangle, it always moves to the right

Try A, B, C and D and choose the correct one

A	B	C	D

Una vez rediseñado el test según las sugerencias de los expertos, se administró a 1519 alumnos (77 clases) de 3º y 4º curso de Educación Primaria (edades de 7 a 9 años), procedentes de 7 centros escolares en Suiza.

Para el análisis psicométrico del test (en términos de validez y fiabilidad), se utilizó el análisis de factores confirmatorios (CFA) [162], la Teoría Clásica de los Test [163] y la Teoría de la Respuesta al Ítem [164]. Los resultados del análisis mostraron una buena fiabilidad con buena consistencia interna (α de Cronbach = 0.85), niveles de discriminación (correlación punto-biserial > 0.2) y un rango amplio de dificultad como proporción de respuestas correctas, para los estudiantes de 7 a 9 años. Además, el test resultó mejor

adaptado para la evaluación y discriminación de los alumnos con habilidades medias o bajas.

Finalmente, se diseñaron variantes del cCTt acortadas, de forma que pueda utilizarse el test si no se tiene suficiente tiempo para la evaluación con el test completo, ya que esta es una de las inquietudes que más comúnmente resaltaron los profesores de Educación Primaria consultados. Para diseñar estas variantes, se utilizó nuevamente CFA [162] para eliminar los elementos que estaban altamente correlacionados o cargados en múltiples factores.

En conclusión, el cCTt resulta ser un instrumento válido y fiable para la evaluación del PC en los cursos 3º y 4º de Educación Primaria (estudiantes de 7 a 9 años). El test es fácil de administrar y calificar a gran escala. Por lo tanto, con este instrumento se amplía el rango de instrumentos para la evaluación del PC diseñados para ser utilizadas por investigadores y profesores en la educación formal. **En concreto, el BCTt (5 a 7 años), el cCTt (7 a 9 años) y el CTt (10 a 16 años) abarcarían conjuntamente el abanico necesario para la evaluación del PC en las etapas educativas de primaria y secundaria.**

No obstante, era deseable ampliar el estudio administrando el cCTt a otras poblaciones, y estudiar específicamente los límites de edad para utilizar uno u otro test. Por este motivo, se realizó una comparativa entre los dos test que se resume en el siguiente apartado.

4.6.2. Comparativa entre el BCTt y el cCTt

Para establecer los límites de edad para utilizar el BCTt y el cCTt, se realizó una experiencia en colaboración con la EPFL de Suiza y la ONG *Tree Tree 2* de Portugal, en alumnos de 3º y 4º de Educación Primaria, que es donde existe duda en cuanto a la aplicación de uno u otro test.

Mediante un análisis psicométrico comparativo se establecieron los límites de validez del BCTt y el cCTt para los cursos 3º y 4º de Educación Primaria. Más concretamente, se empleó la Teoría Clásica de los Test [163] y la Teoría de la Respuesta al Ítem [164] sobre los datos obtenidos de 575 alumnos de centros escolares en Portugal.

Los resultados del análisis psicométrico indican que se debería **preferir el cCTt para los alumnos de 3º y 4º curso**, ya que los alumnos ya tienen una buena asimilación de los conceptos básicos de PC relativos a las secuencias y los bucles, y además obtienen buenos resultados en el BCTt, que es más fácil. No obstante, **el BCTt puede utilizarse si el objetivo es discriminar entre alumnos con capacidades bajas**. Estos resultados subrayan aún

más la importancia de realizar comparaciones entre los instrumentos de PC existentes y superpuestos para establecer sus límites de validez. Mediante comparaciones detalladas, los investigadores podrán elegir el instrumento de evaluación de acuerdo con los requisitos de su estudio.

Por último, instrumentos como el BCTt y el cCTt deberían combinarse con otras formas de evaluación dentro de un *sistema de evaluaciones* [161], junto con otros instrumentos que evalúan otras dimensiones del PC, como se ha visto en el marco teórico, para medir con precisión toda la gama de competencias que están en juego cuando se considera el PC, especialmente las relativas a la observación de los procesos de pensamiento y las estrategias de los estudiantes. Estos aspectos se analizan en los capítulos 6 y 7 de la tesis.

4.7. Publicaciones y avance de objetivos

Las publicaciones relativas a estas experiencias son las siguientes:

- M. Zapata-Cáceres, E. Martín-Barroso, and M. Román-González, "Computational thinking test for beginners: Design and content validation," in *IEEE Global Engineering Education Conference (EDUCON'20)*, Porto, Portugal, pp. 1905–1914, 2020, doi: 10.1109/EDUCON45650.2020.9125368.
- Zapata-Cáceres María, E. Martín-Barroso and M. Román-González, "BCTt: Beginners computational thinking test," in *Raspberry Pi Foundation Research Seminars*, Apr 2021, pp. 46-56. ISSN 2514-586X (16).
- L. El-Hamamsy, M. Zapata-Caceres, E. Martín, F. Mondada, J. Dehler Zufferey, and B. Bruno, "The competent Computational Thinking test (cCTt): Development and validation of an unplugged Computational Thinking test for upper primary school," *Journal of Educational Computing Research*, 2022. In press.
- L. El-Hamamsy, M. Zapata-Caceres, P. Marcelino, J. Dehler Zufferey, B. Bruno, and E. Martin, "Comparing the psychometric properties of two primary school
- Computational Thinking (CT) assessments for grades 3 and 4: the Beginners' CT test (BCTt) and the competent CT test (cCTt)" in *ITiCSE 2022, 27th Annual Conference on Innovation and Technology in Computer Science Education*. En proceso de revisión.

Mediante las experiencias realizadas se confirma la HE1: *es posible evaluar el PC en la etapa de Educación Primaria mediante un test tradicional independiente de un entorno concreto*. Por otro lado, los objetivos que se habían marcado en este capítulo (OP3, OP4 y OC) quedan parcialmente completados y se avanzará en ellos en sucesivos capítulos. En este punto, el grado de cumplimiento de los objetivos es el que se muestra en la Tabla 20.

Tabla 20. Seguimiento de los objetivos, capítulo 4.

Objetivos		Grado de cumplimiento		
		Bajo	Medio	Alto
QP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.			
OP1	Marco teórico y estado del arte			
OP1a	Definiciones teóricas y operacionales del PC			
OP1b	Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas			
OP1c	Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes			
OP1d	Marco teórico y estrategias para la evaluación del PC e instrumentos existentes			
OP2	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.			
OP3	Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC			
OP3a	Diseñar y validar un instrumento de evaluación del PC en edades tempranas			
OP3b	Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas			

			Grado de cumplimiento		
Objetivos			Bajo	Medio	Alto
	OP4	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC			
	>OP4a	Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género			
	OP4b	Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género			
	OP4c	Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género			
OC	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones				

Los límites superiores de administración del BCTt han quedado suficientemente definidos mediante las experiencias descritas en este capítulo, incluso, dada la pérdida de fiabilidad para los alumnos más mayores, ha sido necesario complementar el BCTt con otra herramienta de evaluación más compleja dirigida a estos alumnos (el cCTt). Sin embargo, el límite inferior de edad para la administración del BCTt no queda aún establecido, por lo que, en el siguiente capítulo, se describe la experiencia realizada con este fin en alumnos de Educación Infantil.

A raíz de la publicación del BCTt, éste ha generado gran impacto internacional, con lo que se ha traducido a numerosos idiomas y se han realizado (y actualmente se están llevando a cabo) varias experiencias en diferentes poblaciones y países (como Portugal, Bélgica o Singapur) y planteado líneas de actuación futuras, las cuales se encuentran en diferentes fases de desarrollo (véase el apartado 8.2). En el anexo A se pueden encontrar las versiones completas del BCTt, así como el protocolo de actuación.

CAPÍTULO 5. VALIDACIÓN DEL BCTT EN ALUMNOS DE EDUCACIÓN INFANTIL

En este capítulo se describe la experiencia dirigida a alumnos de Educación Infantil, en colaboración con la *Fontys University of Applied Science* y la *Zuyd University of Applied Science*, ambas en Países Bajos, mediante el uso de entornos de aprendizaje basados en la **robótica**, para **establecer el límite inferior de edad para el que el BCTt es adecuado**.

5.1. Introducción y objetivos específicos

Una vez diseñado y validado el Beginners Computational Thinking test (BCTt), así como el competence Computational Thinking test (cCTt) [7], [8], [165] y en base a la confirmación de la hipótesis base *HE1: es posible evaluar el PC en la etapa de Educación Primaria mediante un test tradicional independiente de un entorno concreto*, se plantea una experiencia para establecer el límite de edad inferior del BCTt.

En esta experiencia, en colaboración con la *Fontys University of Applied Science* y la *Zuyd University of Applied Science*, Netherlands ambas en Países Bajos, también se plantea la posible confirmación de la *HE4: es posible abarcar las tres dimensiones del marco 3D del PC mediante diferentes estrategias de evaluación* como se muestra en la Figura 11. A continuación, se enumeran los **objetivos generales** asociados a esta experiencia:

- *OP3: Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC, en particular: OP3a: Diseñar y validar un instrumento de evaluación del PC en edades tempranas,*
- *OP4: Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC. En concreto el OP4a: Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género; y el OP4c: Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género.*
- *OC: Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones.*

Basándonos en la investigación teórica previa se espera, con relación al marco 3D, que los alumnos sean capaces de comprender los conceptos asociados al PC y de desarrollar prácticas y perspectivas de PC con una madurez creciente en función de su edad [166].

Además, aprender programación es una forma de desarrollar estas habilidades desde una edad muy temprana [61] y los robots programables son ideales para este propósito, ya que estos entornos pueden hacer que el resultado de un programa informático creado sea concreto y tangible [107]. Además, los entornos colaborativos entre pares potencian la reflexión y mejoran las experiencias de aprendizaje relacionadas con el PC [167]. En esta experiencia, se investiga sobre el desarrollo del PC a través de una mejor comprensión de los conceptos de programación subyacentes, y utilizando entornos de robótica colaborativa en alumnos de Educación Infantil.

Las investigaciones previas apuntan a que los alumnos de Educación Infantil son capaces de programar proyectos sencillos de robótica de forma lúdica y atractiva, mejorando sus habilidades de PC, mientras trabajan de forma colaborativa [54], ya que, a través de estos entornos, resuelven problemas descomponiéndolos en pasos; utilizan la abstracción y los algoritmos para traducir el problema a un lenguaje codificable; y aplican la depuración, ya que deben modificar su solución si no es correcta. Utilizar Blue-bot parece ser adecuado para introducir la robótica a los alumnos de Educación Infantil [168]. De igual manera, hay evidencia del uso de Cubetto en los colegios como un entorno robótico adecuado para los alumnos de Educación Infantil para una primera experiencia de programación [169].

Esta experiencia se apoya en las siguientes **hipótesis**:

- La utilización de entornos de programación basados en la colaboración entre iguales, que ofrecen la oportunidad de comprender los conceptos de programación a una edad temprana, motiva a los niños pequeños a aprender a programar.
- Los niños de Educación Infantil que aprenden a programar utilizando los entornos robóticos: Blue-bot o Cubetto muestran una comprensión evaluable de los conceptos de programación.
- La programación con Blue-bot o Cubetto contribuye al desarrollo de las habilidades de PC en alumnos de Educación Infantil.

Los **objetivos específicos** o preguntas de investigación de esta experiencia se enumeran a continuación:

- En línea con el *OP3*: establecer el límite inferior de edad para el que el BCTt es adecuado, realizando una experiencia en niños de 4 y 5 años del último curso de Educación Infantil.
- En línea con el *OP4*: establecer el límite inferior de edad en la que los niños son capaces de alcanzar ciertos conceptos computacionales.
- En línea con el *OP4*: hasta qué punto puede evaluarse un desarrollo del PC en niños de 4 y 5 años.

5.2. Metodología y participantes

En esta experiencia se utiliza un diseño metodológico de pre-test y post-test. Esto incluye: a) una preevaluación de las habilidades PC entre los alumnos de Educación Infantil, b) una intervención basada en la robótica utilizando dos tipos diferentes de entornos de programación, y c) una prueba posterior entre los alumnos para evaluar las habilidades PC. Además de los datos obtenidos a través del BCTt, también se recogen datos cualitativos mediante la observación directa y entrevistas, haciendo a los alumnos diversas preguntas sobre la experiencia al final de las sesiones de programación.

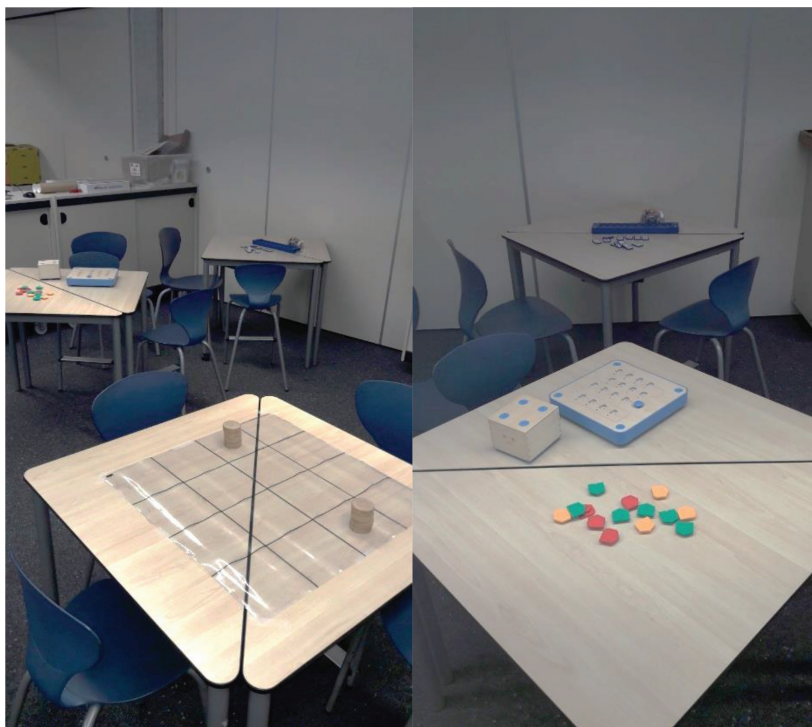
Este estudio se llevó a cabo con alumnos de Educación Primaria de 4 y 5 años, de un colegio de los Países Bajos. Los alumnos fueron divididos aleatoriamente en parejas y asignados a una submuestra para realizar sesiones de programación con Blue-bot (n=10) o Cubetto (n=10). El grupo de control (n=36) no participó en las sesiones.

5.3. Materiales

En las submuestras se utilizaron Blue-bot y Cubetto (véase la Figura 33) como entornos de programación con el objetivo de programar dos tipos diferentes de robots. La programación en sí se realiza mediante tarjetas de comandos físicas (tangibles), cada una de las cuales contiene un comando de programación específico (por ejemplo, hacia delante, hacia atrás, hacia la izquierda, hacia la derecha, etc.). Para componer un programa, estas tarjetas de comandos deben secuenciarse en el orden correcto en un lector de comandos, creando así una secuencia de programación. Este programa creado puede enviarse entonces al robot Blue-bot o al robot Cubetto a través de una conexión *bluetooth*. En ambos entornos se abordaron los siguientes conceptos: "secuencia", "bucle-simple", "bucle-anidado" y "condicional if-then". El condicional "while" no se aborda en ninguno de los entornos, aunque sí se evalúa la posible adquisición del concepto mediante el BCTt.

Para medir el efecto sobre las habilidades de PC de los alumnos, se utilizó el BCTt validado anteriormente, como pre-test y post-test. Además, se han recogido datos cualitativos para abordar otras dimensiones del marco 3D: observación directa durante las sesiones de programación; y entrevistas al final de las sesiones en las que se pregunta oralmente a los alumnos sobre sus sensaciones al interactuar con los entornos y con sus compañeros, y otras cuestiones relacionadas con lo que debían aprender y las características de los entornos.

Figura 33. Entorno preparado para la experiencia con Cubetto.



5.4. Procedimiento

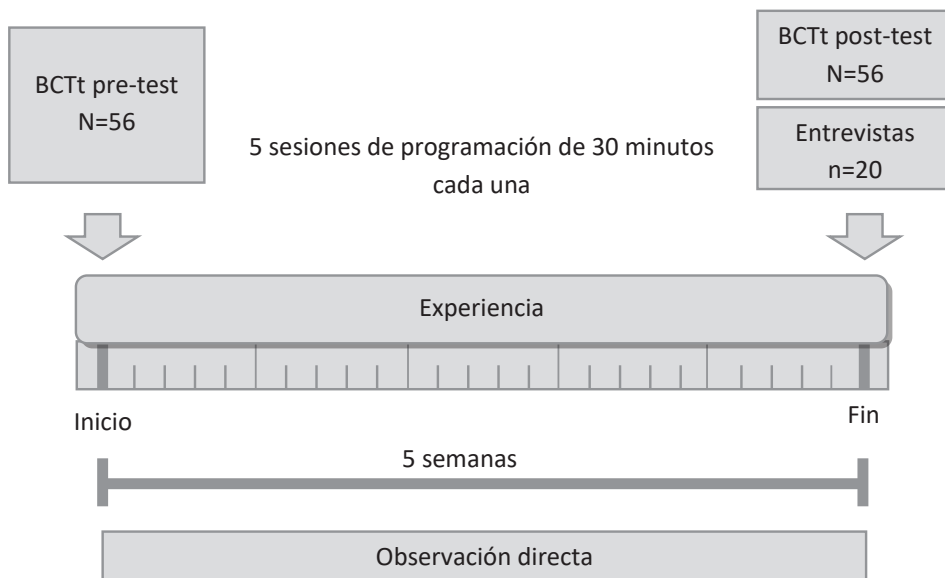
Para que los niños, que no han tenido ningún contacto previo con los conceptos informáticos y que tienen un nivel de lectura y escritura bajo o nulo, puedan realizar el test, el BCTt incluye muy pocos textos sólo con el objetivo de apoyar los símbolos utilizados en las preguntas. De todas formas, la traducción al neerlandés de estos textos ha sido calibrada por un grupo de investigación formado por 8 expertos en educación.

Antes de las sesiones de programación, se administró el BCTt a toda la muestra (N=56). Dado que los alumnos de Educación Infantil de esta categoría de edad aún no saben leer, de acuerdo con el protocolo adjunto, cada ejemplo de tipología de pregunta distinta se leyó en voz alta dos veces. Al mismo tiempo, los alumnos miraban la imagen correspondiente y determinaban qué secuencia de programación les parecía correcta. Posteriormente, los alumnos se dividieron de forma aleatoria entre las dos submuestras y el grupo de control.

Durante cinco semanas, un grupo recibió cinco sesiones de programación de 30 minutos cada una con Blue-bot. En las mismas condiciones, el otro

grupo recibió cinco sesiones de programación utilizando Cubetto. Durante la sesión final, los alumnos que utilizaron Blue-bot tuvieron que resolver un problema de programación aplicando el bucle simple. De forma similar, los alumnos que utilizaron Cubetto tuvieron que resolver un problema de programación aplicando el condicional “if-then”. Tras la finalización de las sesiones de programación, se administró de nuevo el BCTt como post-test en todos los grupos en las mismas condiciones para poder identificar las diferencias en el desarrollo del PC. Además, se recogieron datos cualitativos a través de la observación directa mientras se interactuaba con el entorno de programación y con entrevistas realizadas después de la experiencia de programación. La Figura 34 presenta un esquema del resumen del procedimiento que se llevó a cabo en la experiencia en Educación Infantil.

Figura 34. Procedimiento de la experiencia en Educación Infantil.



5.5. Resultados y análisis de datos

A continuación, se muestra el análisis de los datos recogidos, tanto cualitativos (inventario) como cuantitativos (SPSS).

5.5.1. Análisis cualitativo

Mediante el BCTt se evalúan los conceptos computacionales del marco 3D y, parcialmente, las prácticas computacionales, pero se ignoran las perspectivas computacionales. Por este motivo se recogen datos cualitativos, a través de la

observación directa y se realizan entrevistas al final de la experiencia de programación, estableciendo así un *sistema de evaluación* del PC [148]. Según la hipótesis general de la tesis *HE4*, mediante estos tres tipos de evaluación se pueden abarcar las tres dimensiones del marco 3D.

Los estudiantes se mostraron muy motivados y entusiastas durante toda la experiencia de programación, mostrando una fuerte autorregulación y una atención profunda durante largos períodos de tiempo. El grupo que utilizó Cubetto estaba interesado en utilizar Blue-bot más adelante y viceversa. La observación directa muestra que los alumnos comprenden las preguntas en el BCTt, que están basadas en pictogramas, y son capaces de establecer el vínculo entre la definición visual del problema y las soluciones que deben encontrarse, representadas por combinaciones de flechas direccionales entre las que debe elegirse la respuesta correcta. No se observaron diferencias en cuanto al género, el enfoque, la motivación o las habilidades, pero sí entre los alumnos de 4 y 5 años, ya que se encontró un límite en lo que los alumnos de 4 años eran capaces de comprender.

En cuanto las habilidades de PC que mostraron los alumnos, se observó que podían pensar en una tarea que debía ejecutar el robot, hacer una abstracción descomponiendo el problema en partes más pequeñas y plasmarlo en un algoritmo. Asimismo, su persistencia en la búsqueda de la respuesta fue notable, ya que fueron capaces de cambiar de opinión y corregir sus errores en el código, utilizando el proceso de depuración. Además, el entorno colaborativo entre compañeros fue un gran apoyo para la reflexión, ya que pensar en voz alta y encontrar la solución junto a un compañero fue clave para resolver los problemas.

Otras observaciones muestran que ambas categorías de edad comprenden los principios subyacentes al concepto de "secuencia" para poder compilar un programa y que éste se ejecute mediante la concatenación de las flechas de dirección como comandos. Para los alumnos de 4 años el límite parece estar en las preguntas que incluyen bucles. Los alumnos de 5 años presumiblemente pueden entender y responder a las preguntas relacionadas con bucles, así como a las relacionadas con condicionales. En cuanto al primer objetivo o pregunta de investigación sobre hasta qué punto se puede utilizar el BCTt entre los alumnos de Educación Primaria, queda claro que este instrumento es aplicable entre los alumnos de 4 y 5 años. Esto se confirmó posteriormente a través del análisis de datos cuantitativo.

Independientemente del entorno de programación utilizado, las entrevistas orales realizadas al final de la experiencia de programación también muestran una **motivación muy alta** en los alumnos, ya que los sentimientos más comunes que expresaron al hablar de la experiencia fueron los de felicidad y alegría, y que resolver problemas les hace sentir inteligentes. A todos los

alumnos les gusta utilizar las tarjetas o la tableta para programar. El 90% de los niños que utilizaron Blue-bot pensaron que entendían perfectamente el concepto de “repetición”, en comparación con el 70% de los que utilizaron Cubetto. Aunque esperar su turno hace que el 65% de los alumnos se sientan “impacientes” o “descontentos” y que el 20% tenga dificultades para pedir ayuda a sus compañeros, la metodología colaborativa fue bien valorada por el 100%, que se sintió “feliz”, “contento” o “bien” por trabajar con un compañero. Todos los alumnos sintieron que habían aprendido algo nuevo y que conocían mejor su derecha y su izquierda después de trabajar con los robots.

5.5.2. Análisis cuantitativo mediante el BCTt

Para determinar la fiabilidad del BCTt para la muestra específica de esta experiencia, se calculó el alfa global de Cronbach ($\alpha = 0,911$). El alfa global de Cronbach calculado para el BCTt en la experiencia descrita en el capítulo anterior es $\alpha = 0,824$, por lo que se puede concluir que se cumplen ampliamente los requisitos iniciales de fiabilidad interna.

Para determinar si, y en caso afirmativo, cuál de los dos entornos de programación utilizados, Blue-bot o Cubetto, conduce a diferencias significativas con respecto al grupo de control, y/o si pueden producirse diferencias significativas en una comparación entre los dos entornos de programación, se realizó un análisis de la varianza (Anova) con la prueba de Levene. Posteriormente, se realizaron pruebas post-hoc para demostrar los posibles efectos significativos y confirmar o rechazar las hipótesis. Se calculó la Eta al cuadrado (η^2) para revelar la magnitud de los efectos. Todos los análisis estadísticos asumen un nivel de significación del 5% ($p = \leq 0,05$). En relación a la segunda hipótesis sobre si los niños que aprenden a programar, utilizando los entornos robóticos, muestran una comprensión evaluable de los conceptos de programación, se observa un desarrollo en las medias de los resultados en el post-test respecto al pre-test, como puede verse en la Tabla 21, lo que indica que sí que se puede evaluar si los conceptos abordados se han comprendido.

Los alumnos programaron con Blue-bot o Cubetto muestran un aprendizaje en los conceptos de programación comparados con el grupo de control. La Tabla 21 muestra que todos los alumnos que utilizaron Blue-bot o Cubetto resolvieron con éxito más problemas de pensamiento computacional y desarrollaron más la comprensión de los conceptos de programación “secuencia”, “bucles” (simples y anidados) y “condicional” (if-then, if-then-else). No se pudo demostrar una mejora en “condicional while”, lo cual era de esperar ya que este concepto no se trabaja en los entornos robóticos utilizados.

Tabla 21. Diferencias por concepto computacional abordado en la experiencia en Educación Infantil.

Blue-Bot	Pre-test		Post-test	
	M	SD	M	SD
Total	0.27	.21	0.42	.27
Secuencia	0.36	.23	0.72	.22
Bucle simple	0.36	.28	0.54	.25
Bucle anidado	0.11	.16	0.23	.30
Condicional if-then	0.15	.24	0.25	.35
Condicional if-then-else	0.00	.00	0.25	.35
Condicional while	0.60	.74	0.33	.19

Cubetto	Pre-test		Post-test	
	M	SD	M	SD
Total	0.30	.21	0.48	.27
Secuencia	0.44	.19	0.87	.15
Bucle simple	0.42	.33	0.64	.34
Bucle anidado	0.16	.23	0.21	.27
Condicional if-then	0.10	.21	0.25	.35
Condicional if-then-else	0.05	.16	0.30	.42
Condicional while	0.50	.85	0.37	.48

Control group	Pre-test		Post-test	
	M	SD	M	SD
Total	0.32	.20	0.27	.19
Secuencia	0.48	.18	0.55	.28
Bucle simple	0.43	.33	0.33	.31
Bucle anidado	0.15	.23	0.13	.19
Condicional if-then	0.11	.27	0.10	.20
Condicional if-then-else	0.10	.23	0.10	.23
Condicional while	0.50	.74	0.10	.19

Total = número de preguntas correctas en el BCT; M = media; SD = desviación estándar.

La respuesta a la tercera hipótesis, es decir, si la programación con Blue-bot o Cubetto contribuye al desarrollo de las habilidades de PC entre los alumnos de Educación Infantil, puede deducirse de los datos presentados en la Tabla 22. Estos datos muestran que en ambos grupos que interactuaron con los entornos, en contraste con el grupo de control, se puede medir un desarrollo significativo según el número total de problemas de PC resueltos (Total) y para los conceptos computacionales abordados "secuencia", "bucle simple" y "condicional while". Un examen más detallado de los datos aplicando pruebas post-hoc revela que: 1) para el "número total de problemas de PC resueltos" tanto Blue-bot como Cubetto causan un efecto significativo en el aprendizaje, 2) para "secuencia" Cubetto causa la diferencia significativa, 3) para "bucle-simple" Cubetto causa la diferencia significativa, y 4) para "condicional while" Cubetto causa la diferencia significativa. A pesar de un fuerte desarrollo en los promedios, como se muestra en la Tabla 21, no se puede demostrar un aumento significativo para los conceptos computacionales abordados "bucle anidado", "condicional if-then" y "condicional if-then-else".

Tabla 22. Análisis de varianza (ANOVA) en la experiencia en Educación Infantil.

	Cantidad						
	SS	df	MS	F	p	SD	η^2
Total*	.485	2,53	.243	4.91	.011	.24	.156
Secuencia*	.858	2,53	.429	6.53	.003	.28	.198
Bucle simple*	.934	2,53	.467	4.99	.010	.33	.158
Bucle anidado	.097	2,53	.049	9.23	.404	.23	.034
Condicional if-then	.300	2,53	.150	2.17	.124	.27	.076
Condicional if-then-else	.419	2,53	.209	2.40	.101	.30	.083
Condicional while*	.797	2,53	.399	4.09	.022	.33	.134

*Cantidad = valor medido; Total = número de preguntas correctas en el BCTt; Concepto computacional abordado = secuencia, bucle simple, bucle anidado, condicional if-then, condicional if-then-else, condicional while; SS = suma de cuadrados; DF = grados de libertad; MS = cuadrado medio; F = valor f; p = nivel de significación; SD = desviación estándar; η^2 = Eta al cuadrado; *Efecto significativo.*

Para determinar si los alumnos de 4 y 5 años son capaces de comprender los conceptos de programación subyacentes al BCTt, se realizó una comparación entre ambas edades. Los resultados de la Tabla 23 muestran que el límite para

los niños de 4 años es la comprensión de los conceptos "secuencia" y "bucle simple". En los demás conceptos ("bucle anidado", "condicional if-then", "condicional if-then-else", y "condicional while") no es posible evaluar el desarrollo. Esto contrasta con los niños de 5 años que muestran un desarrollo en todos los conceptos, excepto en el de "condicional while".

Tabla 23. Diferencias por edad según el concepto computacional en Educación Infantil.

Edad: 4 años	Pre-test		Post-test	
	M	SD	M	SD
Total	0.17	.06	0.19	.10
Secuencia	0.37	.16	0.53	.26
Bucle simple	0.24	.18	0.28	.26
Bucle anidado	0.00	.00	0.00	.02
Condicional if-then	0.00	.00	0.00	.00
Condicional if-then-else	0.00	.00	0.00	.00
Condicional while	0.00	.00	0.00	.00
Edad: 5 años	Pre-test		Post-test	
	M	SD	M	SD
Total	0.52	.14	0.56	.21
Secuencia	0.59	.17	0.80	.24
Bucle simple	0.69	.30	0.64	.31
Bucle anidado	0.38	.18	0.42	.17
Condicional if-then	0.30	.33	0.39	.31
Condicional if-then-else	0.18	.29	0.41	.37
Condicional while	1.32	.65	0.48	.37

Total = número de preguntas correctas del cuestionario BCTt; M = media; SD = desviación estándar.

Además, la Tabla 24 presenta un análisis t-test para evaluar si existen diferencias significativas entre los alumnos de 4 y 5 años en cuanto a la comprensión de los conceptos computacionales abordados en el BCTt. De estos resultados se desprende que los niños de 5 años obtienen una puntuación significativamente mejor que los de 4 años en todos los conceptos presentes en el BCTt, como era esperable.

Tabla 24. Análisis t-test comparando entre 4 y 5 años las diferencias en la comprensión de los conceptos computacionales.

	Cantidad				
	t	df	p	CI	d
Total*	-7.69	27.29	0.000	-0.47- -0.27	2.24
Secuencia*	-3.85	47.29	0.000	-0.40- -0.12	1.04
Bucle simple*	-4.44	38.62	0.000	-0.52- -0.19	1.24
Bucle anidado*	-11.26	21.56	0.000	-0.49- -0.34	3.38
Condicional if-then*	-5.92	21.00	0.000	-0.52- -0.25	1.79
Condicional if-then-else*	-5.24	21.00	0.000	-0.57- -0.25	1.57
Condicional while*	-6.20	21.00	0.000	-0.65- -0.32	1.87

Total = número de preguntas correctas del cuestionario BCTT; Concepto computacional abordado = secuencia, bucle simple, bucle anidado, condicional if-then, condicional if-then-else, condicional while; t = valor t; df = grados de libertad; p = nivel de significación; CI = intervalo de confianza; d = tamaño del efecto d de Cohen; *efecto significativo medido

5.6. Conclusiones

Los alumnos de Educación Infantil mostraron comprensión de los conceptos de programación subyacentes a los entornos de programación utilizados, y una mejora general significativa en lo que respecta a las habilidades de PC, con efectos de tamaño mayor en comparación con los del grupo de control. Además, los niños estaban muy motivados cuando trabajaban en parejas y al utilizar entornos de programación con robótica. Por lo tanto, se puede concluir que los entornos de programación basados en la robótica y la colaboración entre pares son apropiados y mejoran el aprendizaje de los conceptos de programación a una edad temprana, por lo que es posible comenzar a desarrollar las habilidades de PC al menos desde los 4 años.

Tanto Blue-bot como Cubetto resultaron adecuados para los niños de Educación Infantil y se observó una mejora general significativa en las habilidades relacionadas con el PC. Cubetto causa las mejoras más significativas en conceptos computacionales específicos. Esto puede deberse a que su diseño permite a los niños abstraer (una de las prácticas computacionales del marco 3D) más fácilmente el código propuesto, ya que a) las tarjetas de comandos se muestran de izquierda a derecha, y b) las tarjetas de comandos tienen la

forma del símbolo utilizado. Por el contrario, en Blue-bot a) las secuencias se muestran de arriba a abajo, y b) los símbolos se dibujan dentro de las tarjetas de comando. Aunque Cubetto parece más adecuado para los niños de Educación Infantil, nuestra sugerencia es que se utilicen diferentes entornos de programación con el mismo grupo de alumnos, para que puedan beneficiarse de ellos de forma transversal, ya que ambos contribuyen al desarrollo de las habilidades de PC, pero de distinta forma.

De esta experiencia también se concluye que los conceptos que se pueden enseñar a cada grupo de edad son diferentes. Se extrae de los resultados que los alumnos de 5 años son capaces de comprender todos los conceptos computacionales abordados en el BCTt, y muestran una mejora significativa, tras las sesiones de programación, en cada uno de ellos (excepto en el “condicional while” que no se trabajó con los entornos), e incluyendo el “condicional if-then-else”, aunque este concepto no se practicó en ninguna de las sesiones de programación. La diferencia entre los resultados de los dos grupos de edad es muy significativa, ya que los alumnos de 4 años sólo muestran comprensión y mejora en dos de los seis conceptos abordados en el BCTt (“secuencia” y “bucle-simple”). Dado que en las sesiones de programación se practicó el “condicional if-then” y no se mostró comprensión ni mejora, se puede concluir que el “condicional if-then” y, por tanto, el “condicional if-then-else” podrían no ser conceptos alcanzables para los alumnos de 4 años. Aunque no mostraron comprensión ni mejora en el “condicional-while”, dado que este concepto no se abordó en los entornos de programación, no se puede afirmar con certeza que no esté al alcance de los niños de 4 o 5 años (Tabla 25). Sin embargo, sería de esperar que el “condicional-while” no estuviese al alcance de los niños de 4 años, ya que no mostraron comprensión ni mejora en los demás condicionales.

Tabla 25. Conceptos computacionales alcanzables para niños de 4 y 5 años según el estudio.

	secuencias	bucle simple	bucle anidado	if-then	if-then-else	while
4 años de edad	SI	SI	NO	NO	NO	En duda
5 años de edad	SI	SI	SI	SI	SI	En duda

El BCTt muestra una fiabilidad muy alta como instrumento para evaluar las habilidades de PC en alumnos de 4 ($\alpha = 0,802$) y 5 años ($\alpha = 0,889$) y, en combinación con los datos cualitativos, proporciona una evaluación de PC

adecuada para estos grupos de edad. Tanto los alumnos de 4 como los de 5 años pueden completar el test, comprenden los pictogramas y son capaces de interpretar las secuencias de programación planteadas como posibles soluciones. Sin embargo, cuatro de los seis conceptos computacionales que se abordan en el test parecen no ser alcanzables para los alumnos de 4 años, ya que no fueron capaces de responder a las preguntas relacionadas con ellos ni mostraron ninguna mejora en el post-test respecto a dichos conceptos.

5.7. Limitaciones y líneas de actuación

La publicación relativa a esta experiencia es la siguiente:

- M. Zapata-Cáceres and N. Fanchamps, "Using the beginners computational thinking test to measure development on computational concepts among preschoolers," in *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*. Singapore: National Institute of Education, 2021, pp. 32-37. **Cabe destacar que esta publicación recibió el premio "Best full paper".**

Sin embargo, la investigación arroja nuevas preguntas que se discuten en este apartado.

En base a la HE1: *es posible evaluar el PC en la etapa de Educación Primaria mediante un test tradicional independiente de un entorno concreto*, los objetivos que se habían marcado en este capítulo (OP3, OP4 y OC) quedan parcialmente completados y se avanzará en ellos en sucesivos capítulos. En este punto, el grado de cumplimiento de los objetivos es el que se muestra en la Tabla 26.

En cuanto a los instrumentos de evaluación, dado que los niños de 4 años no son capaces de responder a las preguntas del test que contemplan algunos conceptos computacionales resulta necesario **el diseño de una nueva versión del BCTt dirigida a los alumnos de 4 años e incluso para niños menores de esta edad**. En este sentido, se plantea una versión de una versión del test para niños menores de 4 años en colaboración con Países Bajos. Este proyecto se encuentra en su fase inicial y contempla incluir otros elementos sensoriales como el tacto o el oído en el instrumento de evaluación. Además, sería necesario realizar nuevas experiencias con muestras más grandes, con más sesiones de programación (con una duración acorde con la limitada capacidad de atención de los niños muy pequeños) y en otros países para confirmar los hallazgos. Estas y otras experiencias, en proceso y futuras, se justifican y describen brevemente en el capítulo 8.

Tabla 26. Seguimiento de los objetivos, capítulo 5.

Objetivos		Grado de cumplimiento		
		Bajo	Medio	Alto
OP1	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.			
	Marco teórico y estado del arte			
	OP1a Definiciones teóricas y operacionales del PC			
	OP1b Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas			
	OP1c Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes			
OP2	OP1d Marco teórico y estrategias para la evaluación del PC e instrumentos existentes			
	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.			
	Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC			
OP3	OP3a Diseñar y validar un instrumento de evaluación del PC en edades tempranas			
	OP3b Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas			
	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC			

OC	OP4a	Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género	
	OP4b	Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género	
	OP4c	Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género	
	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones		

En cualquier caso, a través de las experiencias descritas en los capítulos 4 y 5, se han logrado diseñar y validar **instrumentos de evaluación del PC, fiables e independientes, y con proyección internacional, para la etapa de Educación Primaria (BCTt y cCTt) y, además, se han validado para esta población estableciéndose los límites de utilización para cada edad y los protocolos de administración correspondientes**. Sin embargo, estos instrumentos deberían combinarse con otras formas de evaluación dentro de un *sistema de evaluaciones* [161], junto con otros instrumentos que evalúan otras dimensiones del PC, como se ha visto en el marco teórico, para medir con precisión toda la gama de competencias que están en juego cuando se considera el PC, especialmente las relativas a la observación de los procesos de pensamiento y las estrategias de los estudiantes. Estos aspectos se analizan en los capítulos 6 y 7 de esta tesis.

En cuanto a los entornos de aprendizaje, el hecho de que los alumnos de 4 y 5 años no hayan sido capaces de alcanzar algunos de los conceptos computacionales propuestos no demuestra que éstos sean inalcanzables con otro tipo de estrategias de aprendizaje o enfoques diferentes de aprendizaje. En los siguientes capítulos se describen experiencias con entornos de aprendizaje basados en juegos tanto inmersos en un contexto escolar como totalmente voluntarios para así comprobar si realmente estos conceptos son alcanzables o no, así como detectar diferencias en cuanto a edad y género en los estudiantes de Educación Infantil y Primaria.

CAPÍTULO 6. DESARROLLO Y VALIDACIÓN DE VIDEOJUEGO PARA LA ENSEÑANZA Y EVALUACIÓN DEL PC: BLUE ANT CODE (BAC)

En este capítulo, se describe la experiencia de desarrollo y validación de un entorno de **aprendizaje y evaluación basado en juegos**, con el objetivo general de analizar estrategias para el aprendizaje y evaluación del PC y las habilidades de PC de alumnos de Educación Primaria, además de otros objetivos que se detallan en el siguiente apartado.

6.1. Introducción y objetivos

Una vez validado los instrumentos independiente del entorno, BCTt y cCTt, descritos en capítulos anteriores, de evaluación del PC en Educación Primaria, es posible la validación de otras estrategias de aprendizaje y evaluación dependientes de entornos concretos, ya que puede comprobarse si ha habido o no aprendizaje utilizando estos instrumentos como herramienta de pre-test y post-test y, de la misma forma, puede compararse los resultados de la evaluación del entorno con la evaluación a través del test.

Como se vio en el estado del arte de esta tesis (ver capítulo 2), los entornos que utilizan la estrategia basada en juegos (GBL) potencian el PC, pero ¿hasta qué punto? El hecho de que los alumnos de 4 y 5 años no hayan sido capaces de alcanzar algunos de los conceptos computacionales propuestos en la experiencia anterior con entornos robóticos (ver capítulo 5), no demuestra que éstos sean inalcanzables o no evaluables con otro tipo de estrategias de aprendizaje o enfoques de evaluación.

En este capítulo se diseña, desarrolla y valida mediante una experiencia en colegios de Educación Primaria, un **entorno de aprendizaje y evaluación del PC basado en GBL y mediante analíticas de aprendizaje en juegos o Game Learning Analytics (GLA)** como enfoque prometedor para evaluar el desarrollo del PC.

6.1.1. Contexto general

De esta forma, esta experiencia se apoya en las hipótesis, que se recuerdan a continuación (ver apartado 3.2 para más detalles):

- *HE2: Mediante diferentes estrategias de aprendizaje, pueden abarcarse las tres dimensiones del PC. La relación entre el marco 3D y las estrategias de aprendizaje*

es como se muestra en la Figura 10, de forma que se espera un aprendizaje que abarque las tres dimensiones del marco 3D mediante este tipo de entornos.

- *HE3: Los entornos que utilizan la estrategia basada en juegos (GBL) potencian el aprendizaje de habilidades de PC en Educación Primaria en contextos escolares, sobre en los niños más pequeños y si se aplica una componente colaborativa.*
- *HE4: es posible abarcar las tres dimensiones del marco 3D del PC mediante diferentes estrategias de evaluación como se muestra en la Figura 11, de forma que también se espera una evaluación que abarque las tres dimensiones del marco 3D mediante este tipo de entornos*

A continuación, se enumeran los **objetivos generales** asociados a esta experiencia:

- *OP3: Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC, en particular: **OP3b**: Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas,*
- *OP4: Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC; en concreto el **OP4a**: Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género; el **OP4b**: Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género; y el **OP4c**: Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género.*

Para alcanzar los objetivos generales y como contribución a la investigación en este área, se ha desarrollado un entorno basado en el marco teórico previo; las teorías del constructivismo social [12] y del desarrollo cognitivo [13]; en marcos como la Teoría de la Actividad [14]; y en metodologías colaborativas (apartado 2.3). En base a la HE2, se ha desarrollado el entorno como instrumento de aprendizaje del PC en etapas tempranas a través de la programación visual por bloques, basado en el aprendizaje basado en problemas, el aprendizaje colaborativo, y el aprendizaje basado en juegos. Este entorno se llama **Blue Ant Code (BAC)** e incluye modos de juego individual y colaborativo.

Además, apoyándonos en la HE3, el entorno se ha diseñado en base a la estrategia GBL [5], [61] como vehículo demostrado para el desarrollo del PC (ver capítulo 2).

Después, se realizó un estudio de caso exploratorio en el que participaron 176 alumnos de educación. Se utilizaron los siguientes instrumentos: a) el BCTt, dirigido a evaluar las habilidades de PC en edades tempranas [8]; b) el

entorno desarrollado (BAC) como instrumento de aprendizaje y evaluación; y c) varios cuestionarios para complementar la evaluación. La aplicación de los tres instrumentos en el mismo caso de estudio pretende cubrir las tres dimensiones computacionales del marco 3D, basándonos en la *HE4* en lo que respecta a los enfoques de evaluación y funcionando como un *sistema de evaluación* (ver apartado 2.5 para más detalles).

6.1.2. Objetivos específicos

Este estudio pretende responder a las siguientes preguntas de investigación u objetivos específicos:

- **OE1:** ¿Es adecuada una estrategia basada en el juego como metodología de aprendizaje de la programación y herramienta de evaluación (a través de GLA) de las habilidades de PC a lo largo de la etapa de Educación Primaria?
- **OE2:** ¿Qué habilidades de PC puede n empezar a desarrollarse en cada etapa de Educación Primaria?
- **OE3:** ¿Cuáles son las diferencias entre los enfoques de aprendizaje colaborativo e individual basado en juegos en su aplicación en alumnos de Educación Primaria y qué estrategia es mejor para cada edad?

A continuación, se describe el desarrollo del entorno, para después detallar la experiencia en colegios que se realizó para recoger datos y, posteriormente el análisis de los mismos y los hallazgos obtenidos.

6.2. BAC como herramienta de aprendizaje

El diseño del entorno BAC se fundamenta en el marco teórico del capítulo 2. En este apartado se contextualizarán las teorías, estrategias y técnicas generales al entorno basado en juegos desarrollado. Además, se describen las premisas básicas en las que se fundamenta el diseño.

Dado que numerosos estudios indican que las habilidades relacionadas con el PC deben enseñarse a edades tempranas para fomentar el desarrollo cognitivo de los alumnos (apartado 2.2), **el entorno se dirige a la etapa de Educación Primaria**. Asimismo, como se ha analizado en el capítulo 2, el PC se puede desarrollar eficazmente a través del **aprendizaje de la programación** [4], [61], sobre todo a través de **bloques visuales** (apartado 2.4) que, además, si son integrados en **entornos de juego** mejoran la creatividad y las habilidades de resolución de problemas de los estudiantes [5]. Estos entornos de programación visuales pueden mejorar las habilidades y estrategias de

programación de los principiantes, así como fomentar un mayor compromiso con las tareas de programación. Además, pueden ayudar a los estudiantes en la modelización, la simulación y la resolución de problemas, que son clave en el desarrollo del PC [61].

Por lo tanto, merece la pena integrar elementos de juego en entornos visuales y explorar su influencia en la experiencia del alumno y su motivación para resolver problemas computacionales. Los entornos de programación basados en el juego, como BAC, pueden potenciar todas las dimensiones del marco 3D y mejorar especialmente las prácticas computacionales (Figura 10).

En cuanto a las estrategias de aprendizaje adecuadas para la enseñanza del PC en Educación Primaria, basándonos en nuestra hipótesis *HE2*, en este diseño de entorno combinamos tres de las cuatro estrategias más aplicadas y representadas en la literatura para este aprendizaje [4], esperando que mediante esta combinación se abarquen todas las dimensiones del marco 3D. La estrategia basada en proyectos se ha descartado, ya que los estudiantes necesitan estar comprometidos cognitivamente con la materia durante un largo periodo de tiempo [92], lo que excede el alcance de esta investigación. En el entorno BAC, la **estrategia basada en problemas** ha sido el principal enfoque adoptado; la **estrategia basada en el juego** se ha aplicado como vehículo de andamiaje para guiar y facilitar el proceso de resolución a través de la mecánica del juego; y la **estrategia colaborativa** se ha seleccionado como instrumento transversal para mejorar el proceso de aprendizaje.

Según Qian y Clark, la eficacia de un entorno de **aprendizaje basado en juegos** depende de que los diseños de los juegos se basen en teorías de aprendizaje establecidas y en elementos de diseño de juegos que hayan demostrado tener éxito [11]. Además, los resultados positivos de la **estrategia de aprendizaje basado en juegos** predominan en los estudios teóricos sobre el aprendizaje [170], ya que *"el diseño de juegos educativos es un proceso interdisciplinar que requiere una profunda comprensión de la teoría del diseño de juegos, conocimientos sobre el tema académico y fundamentos en las teorías del aprendizaje pertinentes"* [171]. Por estos motivos, se ha realizado una investigación teórica en el capítulo 2. De dicha investigación resaltamos, para el diseño del entorno basado en juegos, el constructivismo social de Vygotsky [12] y la teoría del desarrollo cognitivo de Piaget [13], ya que el constructivismo en educación postula un proceso de aprendizaje dinámico, colaborativo e interactivo mediante el cual el conocimiento es creado activamente por el propio alumno. Así, la teoría sociocultural del aprendizaje, que sostiene que el aprendizaje tiene lugar en entornos sociales, activos y situados [11], [12], es uno de los principales fundamentos teóricos utilizados por los investigadores del aprendizaje basado en juegos [23], [172], [173].

El marco de aprendizaje constructivista de Papert es también relevante para entornos de aprendizaje basados en juegos, ya que ayuda a los alumnos a potenciar la interacción social, la cognición, el pensamiento de alto nivel y el PC [47], además, en consonancia con este marco, en los entornos de juegos virtuales los estudiantes resuelven los retos de aprendizaje mediante la interacción y la colaboración [4]. Por otra parte, la Teoría de la Actividad [14], proporciona un marco adecuado para analizar las necesidades, las tareas y los resultados a la hora de diseñar entornos de aprendizaje constructivistas, ya que sus supuestos son consonantes con los del constructivismo, el aprendizaje situado, el razonamiento basado en casos, la cognición social y la cognición cotidiana. Además, la Teoría de la Actividad ha sido ampliamente utilizada para proporcionar un marco operativo claro para el diseño de este tipo de entornos. Finalmente, la teoría del flujo es clave para fomentar la motivación en los juegos y el aprendizaje, ya que la actividad del jugador está impulsada por el placer y no por recompensas externas [11], [121], [174].

En cuanto al **aprendizaje colaborativo**, éste fomenta las habilidades de pensamiento crítico de forma mucho más eficaz que los entornos individuales [85]. Esto se traduce en mayores niveles de logro y mayor productividad, así como en relaciones más afectuosas, solidarias y comprometidas. Así, mediante este tipo de aprendizaje, se promueve una mayor salud psicológica, competencia social y autoestima que a través del aprendizaje individual [87]. Además, los equipos cooperativos alcanzan niveles de pensamiento más altos y retienen la información durante más tiempo que los alumnos que trabajan individualmente [88], [89].

En cuanto al aprendizaje de la programación de forma colaborativa, es destacable la importancia de los entornos de colaboración entre iguales para lograr experiencias de aprendizaje de programación significativas [167]. La reflexión es una estrategia que se utiliza a menudo en los estudios sobre experiencias de programación. Está relacionada con la mejora de las dimensiones de las prácticas computacionales y de las perspectivas computacionales, ya que los estudiantes revisan y piensan sobre su experiencia de programación, mejorando así su propio proceso de aprendizaje [61]. En un entorno colaborativo, los estudiantes no sólo reflexionan sobre su experiencia de programación, sino que también revisan y piensan sobre el rendimiento de la programación de sus compañeros. Además, pensar en voz alta, así como responder y aclarar las preocupaciones de los compañeros, también puede mejorar estas dos dimensiones del marco 3D de PC, especialmente las perspectivas computacionales [15], [60], [61]. Hainey *et al.* presentan una revisión bibliográfica sistemática de la investigación sobre el aprendizaje basado en juegos, y concluyen que son necesario más estudios que comparen el juego individual y el colaborativo para identificar sus beneficios pedagógicos en la etapa de Educación Primaria [175]. El entorno BAC se ha diseñado para que se pueda

cambiar entre el modo de juego individual y el colaborativo con el objetivo de poder comparar estas dos estrategias de aprendizaje.

Con las estrategias de **aprendizaje basadas en problemas**, la adquisición de conceptos se produce a través de escenarios definidos, guiados y presentados por los profesores. Sin embargo, en una estrategia de aprendizaje basada en juegos el alumno no recibe necesariamente una definición previa de los conceptos, sino que guía su propio aprendizaje libremente, motivado por la sensación de logro (Figura 10), por lo que la adquisición de conceptos puede no ser completa. Por el contrario, nuestra hipótesis es que, en un entorno de aprendizaje basado en el juego en el que el alumno debe superar retos, el dominio de las prácticas es esencial para progresar en el juego [61]. Por ejemplo, puede que los alumnos no sepan que están utilizando *secuencias*, porque no ha habido una explicación previa, pero si han sido capaces de completar un nivel en el que tienen que averiguar cuál es la secuencia óptima entre diferentes alternativas, queda patente que la práctica computacional de *probar y depurar* ha sido dominada. Además, nuestra hipótesis es que las prácticas y, sobre todo, las perspectivas computacionales se ven reforzadas por si el aprendizaje es colaborativo, dado que la negociación social del significado de los conceptos a aprender es una parte importante de los entornos colaborativos de resolución de problemas [176], [177].

Por último, en cuanto al despliegue, las tabletas digitales parecen ser dispositivos adecuados para los entornos de aprendizaje en edades tempranas ya que no requieren de dispositivos externos como el ratón o el teclado y el usuario manipula directamente los objetos. La integración de múltiples funciones dentro de un dispositivo y las interfaces táctiles son facetas clave que hacen que las tabletas sean adecuadas como herramientas de aprendizaje en las escuelas [178]. La colaboración entre compañeros mejora, dado que las tabletas digitales proporcionan una interacción más natural y directa con el contenido [179], [180] Además, Lin *et al.* concluyen que el uso de las tabletas potencia los resultados del aprendizaje, especialmente en los casos de grupos numerosos, ya que se produce una mayor colaboración entre compañeros y resultados de mayor calidad [181].

6.3. Adaptación de BAC al marco 3D

El entorno BAC se basa en las estrategias y teorías de aprendizaje discutidas anteriormente, toma como referencia el marco 3D del PC, y está en línea con la propuesta de Lye y Koh [61] de crear un entorno de aprendizaje de resolución de problemas basado en el construccionismo para apoyar todas las dimensiones computacionales del marco 3D.

Como se ha visto en el marco teórico de esta tesis, el PC es una asignatura importante en los planes de estudio nacionales de muchos países y su inclusión en la educación ha avanzado mucho en la última década, empezando ya en Educación Infantil en algunos países, como Australia o Japón [65]. Se está formando a los niños en PC, de forma que desarrollen el pensamiento independiente y la resolución de problemas y luego puedan aplicar estas habilidades a diferentes disciplinas o a la vida cotidiana. Sin embargo, siguen existiendo algunas dudas sobre la mejor manera de enseñar PC, cuándo comenzar dicha instrucción o qué enseñar [62], y son especialmente escasas las investigaciones sobre las prácticas computacionales en los primeros cursos [63]. Es importante, por tanto, adaptar el marco de referencia educativo a edades tempranas.

Para la categorización de las dimensiones del PC, nos referiremos al marco 3D [46] escogido para todas las experiencias en esta tesis, ya que su contexto es similar al del presente estudio. Sin embargo, como el entorno BAC está dirigido a alumnos más jóvenes que los del estudio de Brennan y Resnick y el entorno no es Scratch, el marco 3D se ha adaptado a un nivel de principiante y al entorno BAC según la Tabla 27.

Tabla 27. Dimensiones clave del marco 3D y su adaptación al nivel de principiante.

Dimensiones del marco 3D	Aspectos	Nivel principiante
Conceptos computacionales	Secuencias	Sí
	Bucles	Sí
	Eventos	No
	Paralelismo	No
	Condicionales	Sí
	Operadores	No
	Datos	No
Prácticas computacionales	Ser incremental e iterativo	Sí
	Probar y depurar	Sí
	Reutilizar y remezclar	No
	Abstraer y modularizar	Sí
Perspectivas computacionales	Expresar	Sí
	Conectar	Sí
	Cuestionar	Sí

Aunque existen datos sobre el aprendizaje progresivo de conceptos cuando los alumnos interactúan con Scratch o ScratchJr en diferentes grupos de edad [93], los conceptos computacionales seleccionados como adecuados para un nivel principiante, son aquellos que se ha demostrado que pueden ser alcanzados inicialmente por estudiantes de primaria [93], [95], [96], [182], y, por tanto, pueden ser evaluados por el BCTt [8]. Por ejemplo, aunque los eventos sencillos parecen alcanzables para los alumnos de primaria en un contexto de ScratchJr, el verdadero reto es dominar los eventos complejos que los alumnos de esta edad no demuestran [93]. Las prácticas y perspectivas computacionales seleccionadas son las que están relacionadas con los conceptos abordados [96]. Por ejemplo, aunque se sabe que abstraer es difícil para los estudiantes de Educación Primaria, esta práctica podría observarse mientras los estudiantes interactúan con el entorno BAC y con sus compañeros en el modo colaborativo [46], [96]. Sin embargo, la reutilización y la remezcla no se han incluido en este nivel principiante ya que BAC no es un entorno de creación de juegos, como lo es Scratch, por lo que no existe la opción de reutilización de código.

6.4. Desarrollo del entorno BAC

El desarrollo del entorno BAC incluye el videojuego compatible con múltiples plataformas, una API REST para la conexión con la base de datos relacional y módulos complementarios para el filtrado, la visualización y análisis de datos con GLA.

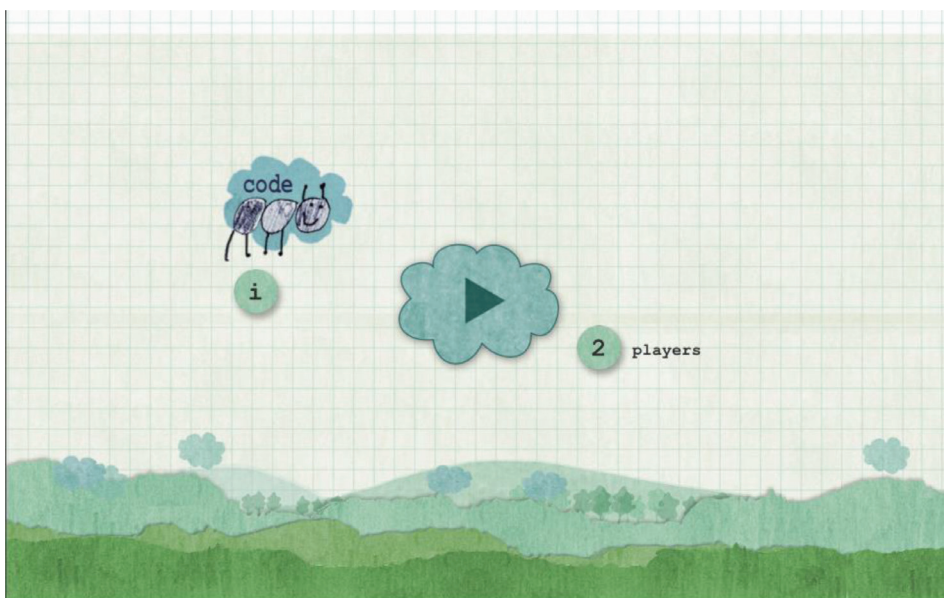
6.4.1. Diseño de juego

Teniendo en cuenta que el entorno se utilizaría en principio en colegios pero que tal vez se escalara en el futuro a otras plataformas, se decidió orientar el desarrollo hacia PC, Android, e iOS, de forma que se pudiese utilizar por ejemplo en el aula de ordenadores del centro escolar o bien en tablets iOS o Android si el centro dispusiera de ellas y, además, pudiera desplegarse en el futuro en plataformas como Google Play o Apple Store.

BAC está diseñado en base a los fundamentos teóricos analizados, para ser instrumento de aprendizaje como una herramienta de evaluación a través GLA. Se prestó especial atención a los elementos de diseño del juego, ya que la eficacia del aprendizaje basado en juegos depende de ellos [11]. Así, la colaboración, la complejidad, la competición, la estrategia, los retos adaptados para mantener la fluidez, los objetivos claros, la comunicación, la interactividad y el andamiaje son los principales elementos de diseño del juego implementados en el entorno. Asimismo, se han tenido muy en cuenta los preceptos de la experiencia del usuario y se ha adaptado el diseño a la

edad objetivo (por ejemplo, las transiciones en los diseños de los puzles, los bloques grandes que se pueden arrastrar, los decorados y los personajes coloridos y animados). Además, BAC tiene **dos modos de juego: individual y colaborativo**, que se seleccionan antes de empezar a jugar, como puede verse en la Figura 35.

Figura 35. Pantalla de selección de modo de juego: individual o colaborativo.



El juego se divide en **seis niveles** de resolución de problemas con grados de dificultad creciente, a los que se accede desde un menú principal (véase la Figura 36). Cada nivel está ambientado en un **puzle de tipo laberinto** diferente representado por un hormiguero en el que hay varias cámaras o salas conectadas, para que el protagonista pueda explorarlas. En el modo individual, los elementos del puzle se generan **aleatoriamente de forma procedimental**, con lo que el usuario se encuentra un puzle diferente cada vez que juega. Hay 4 elementos de juego: hormiga azul (personaje principal), hormiga roja (enemigo), hoja (objetos a recoger), y montón de hojas (meta). En cada nivel, el alumno se enfrenta al reto de resolver un problema computacional mediante instrucciones visuales basadas en bloques que pueden arrastrarse para ensamblar una pieza de código secuenciado, con el fin de guiar a la hormiga azul a través de varias cámaras del hormiguero, recogiendo una o más hojas, evitando eventualmente a la hormiga roja en el camino y, finalmente, llegando al montón de hojas. Cada nivel de dificultad creciente se centra en un concepto computacional del marco 3D, como se muestra en la Tabla 28.

Figura 36. Pantalla de selección de nivel.

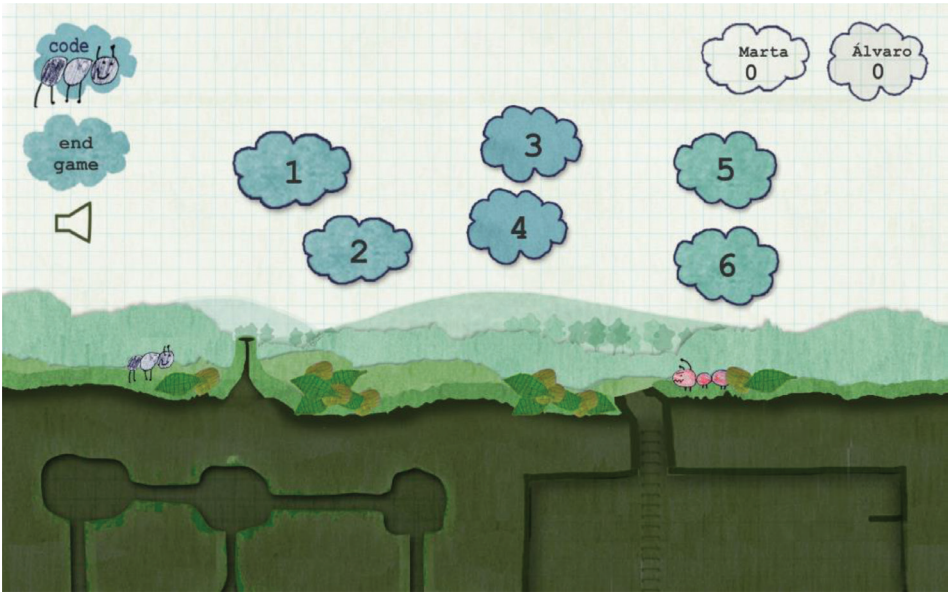


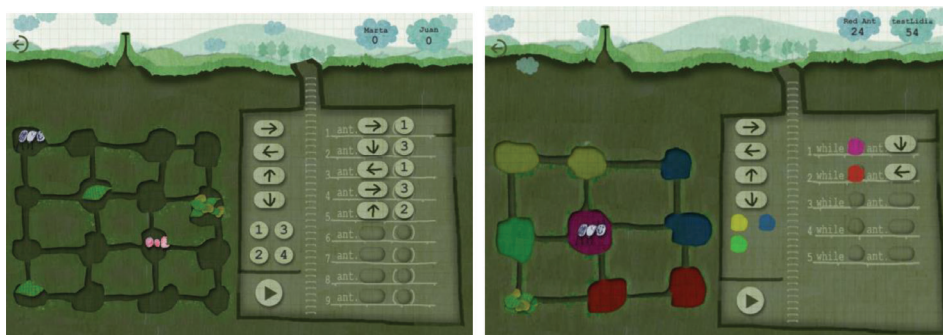
Tabla 28. Conceptos del marco 3D implementados en BAC por nivel.

Nivel	Concepto computacional					
	Secuencias		Bucles		Condicional (while)	
	Simple	Complejo	Simple	Complejo (anidado)	Simple	Complejo
1	x					
2		x				
3			x			
4				x		
5					x	
6						x

La interfaz del juego se divide verticalmente en dos áreas (véase la Figura 37). El puzle (hormiguero) aparece en la parte izquierda de la pantalla; los bloques de programación con los que construir el código y resolver el puzle están disponibles en la parte derecha. En el modo individual, los 4 elementos del juego

se colocan aleatoriamente en las cámaras del hormiguero, y el estudiante debe construir el código para resolver el reto. El modo de juego colaborativo se basa en las cuatro condiciones que Dillenbourg indica para establecer un contexto colaborativo activo en el que es probable que se activen los mecanismos de aprendizaje [86]: establecer las condiciones iniciales; sobre especificar el contrato de "colaboración" con un escenario basado en roles; guiar mediante andamiaje las interacciones productivas englobando las reglas de interacción en el medio; y supervisar y regular las interacciones. En cuanto a la asignación de roles en el modo colaborativo, el primer alumno establece un reto, colocando los elementos del juego donde desee; y el segundo asume el papel de resolver el reto (Figura 37). La hipótesis es que los dos alumnos, independientemente de su rol, aprenden el uno del otro, que no se trata de un juego competitivo sino cooperativo ya que ambos interactúan y se ayudan mutuamente.

Figura 37. Ejemplos de partida aleatoria en niveles 3 (izquierda) y 5 (derecha).



Para tener éxito y completar el nivel de juego, los estudiantes deben probar varios planes y encontrar la estrategia más eficaz para resolver el reto. Los jugadores más hábiles pueden adoptar estrategias que ya han utilizado para resolver nuevos problemas, trabajando en prácticas computacionales del marco 3D como la descomposición de problemas, la prueba, la abstracción o la iteración. Como elemento de juego atractivo, se ha establecido un sistema de puntuación, de modo que el alumno que plantea el reto gana más puntos cuanto más difícil es ese reto; y el estudiante que resuelve el reto gana más punto cuanto más óptima resulta la solución a ese reto. En cuanto a la dificultad del reto, será más difícil cuantos más pasos sean necesarios para su resolución; mientras que la solución óptima será la que menos instrucciones y, en segunda instancia, pasos, se necesiten para resolver el reto.

Al acceder a un nivel por primera vez, deberá realizarse una breve explicación a los niños del objetivo y mecánica del juego. A continuación, se detalla la

explicación por cada nivel. En el caso del modo individual, el rol del jugador 1 es asumido por el propio juego:

- Nivel 1:
Objetivo del juego: La hormiga azul debe llevar la hoja hasta el montón de hojas sin pasar por la casilla de la hormiga roja
Mecánica: El jugador 1 coloca los elementos en el puzle. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado.
- Nivel 2:
Objetivo del juego: La hormiga azul debe llevar las hojas hasta el montón de hojas sin pasar por la casilla de la hormiga roja
Mecánica: Idéntica a la del nivel 1, aunque el laberinto contiene más casillas y se puede necesitar un mayor número de instrucciones para resolverlo que en caso del nivel 1.
- Nivel 3:
Objetivo del juego: La hormiga azul debe llevar las hojas hasta el montón de hojas sin pasar por la casilla de la hormiga roja.
Mecánica: El jugador 1 coloca los elementos en el puzle. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado, incorporando el número de veces que se repite cada una.
- Nivel 4:
Objetivo del juego: La hormiga azul debe llevar las hojas hasta el montón de hojas
Mecánica: El jugador 1 coloca los elementos en el puzle. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado, incorporando el número de veces que se repiten cada dos instrucciones agrupadas. Este nivel entraña bastante dificultad, ya que las instrucciones agrupadas se repiten las veces indicadas de forma secuencial, pero en grupos de dos. Por ejemplo, si un grupo en el que se han incorporado las instrucciones "arriba" y "abajo" y se ha indicado que se repita tres veces, la ejecución será la siguiente: "arriba", "abajo", "arriba", "abajo", "arriba", "abajo". Es posible que este nivel requiera más tiempo de explicación a los alumnos.
- Nivel 5:
Objetivo del juego: La hormiga azul debe llegar hasta el montón de hojas
Mecánica: El jugador 1 coloca los elementos en el puzle. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda",

“derecha”, incorporando un color que determinará en qué casillas se realizará la acción. Por ejemplo, si se incorpora “amarillo” y “derecha”, todas las veces en las que la hormiga esté en una casilla amarilla, avanzará hacia la derecha. Este nivel entraña bastante dificultad sobre todo la primera vez por lo que es posible que se requiera más tiempo de explicación a los alumnos.

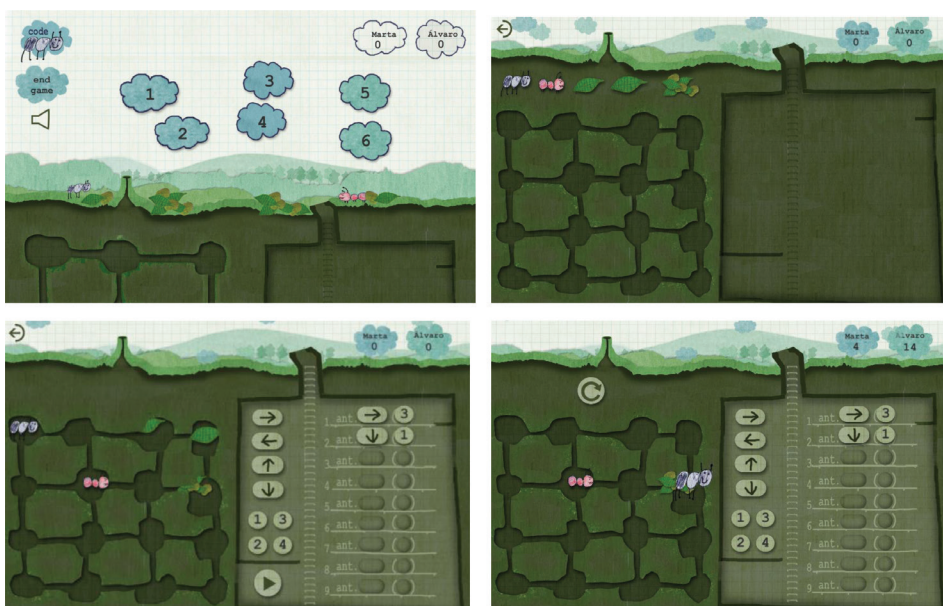
- Nivel 6:

Objetivo del juego: La hormiga azul debe llegar hasta el montón de hojas

Mecánica: Es la misma que en el nivel 5 pero el laberinto contiene más casillas con lo que es más difícil de resolver y probablemente se necesiten más instrucciones para ello.

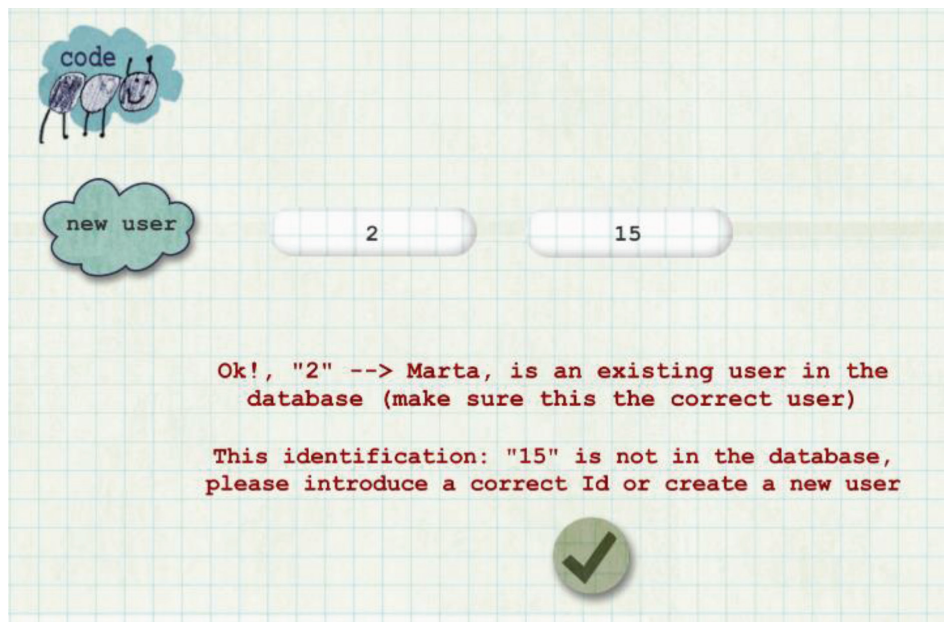
En la Figura 38, se muestra un ejemplo de partida colaborativa en el nivel 3.

Figura 38. Ejemplo de partida en el nivel 3.



BAC cuenta con un sistema de registro de estudiantes por el que se identifica a los jugadores en una base de datos para poder seguir su progresión en el tiempo. Los estudiantes pueden elegir un apodo o pseudónimo, y se les da un número de identificación único a cada uno de ellos, que es con el que la aplicación trabaja.

Figura 39. Pantalla de registro y creación de nuevo usuario, mediante apodo y número de identificación.

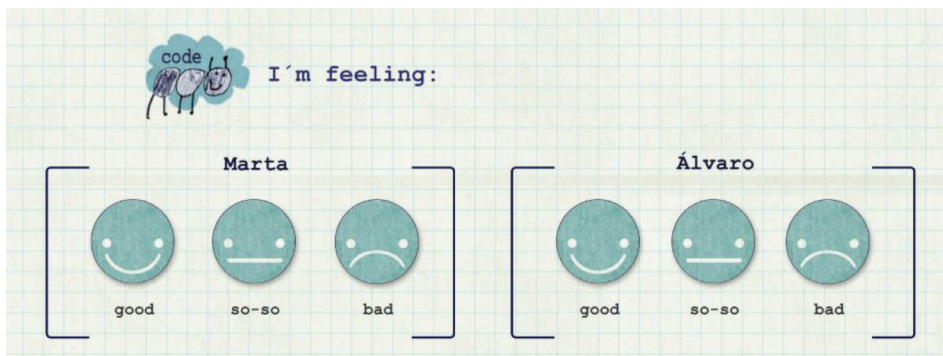


El sistema de monitorización registra cada paso del juego a bajo nivel (cada micro acción del jugador), mientras se juega, recogiendo datos en tiempo real en una base de datos remota para ser analizados y mediante los cuales se pueden evaluar posteriormente los conceptos computacionales del marco 3D, las prácticas computacionales y, hasta cierto

punto, las perspectivas computacionales (según la hipótesis *HE4*). Los datos recogidos incluyen información sobre el tiempo que se ha tardado en plantear y resolver los problemas, el grado de optimización del camino planteado en la solución o el número de partidas ganadas por un jugador. En el anexo B.2 se detalla el esquema completo de la base de datos.

Por último, para evaluar las perspectivas computacionales, se pueden recoger datos cualitativos mediante la observación directa de los alumnos mientras interactúan tanto con el entorno como con sus compañeros, especialmente en el modo de juego colaborativo. Además, para recoger datos adicionales sobre la motivación, se pregunta a los estudiantes a través de la interfaz de usuario sobre su estado de ánimo antes y después de la sesión de juego como se puede observar en la Figura 40.

Figura 40. Pantalla de estado emocional en el modo colaborativo.



6.4.2. Implementación

Para el desarrollo del entorno BAC, se utilizó Unity, que es un motor de videojuego multiplataforma creado por Unity Technologies, que permite a los desarrolladores de videojuegos asociar la implementación de imágenes, escenarios y mecánicas de juego de una manera rápida y eficaz gracias a un conjunto de librerías incluidas en el programa¹. Además, cuenta con una gran flexibilidad para la exportación a diferentes plataformas y dispositivos.

Para el proyecto se han utilizado versiones LTS (Long Term Support), por su estabilidad y soporte a largo plazo. Por otro lado, cabe destacar también el uso del programa Unity Hub² de Unity Technologies, que permite una mejor administración del proyecto y de sus distintas versiones a lo largo del tiempo de desarrollo. Como entorno de desarrollo, se ha utilizado el IDE Visual Studio³ compatible con múltiples lenguajes de programación (en este caso se ha optado por el lenguaje C#) y entornos de desarrollo web.

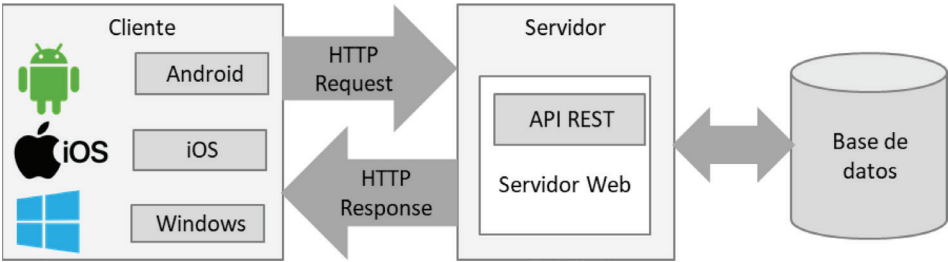
Para conectar con la base de datos SQL creada en servidor Web, también se desarrolló una API REST basado en el framework .NET. El protocolo REST permite separar totalmente la interfaz de usuario del servidor y el almacenamiento de datos, con lo que se mejora la portabilidad de la interfaz a otras plataformas, aumenta la escalabilidad y seguridad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente y escalable. El framework .NET permite la utilización del mismo IDE Visual Studio que se utilizó para la programación de la lógica de juego, para el desarrollo de la API (véase la Figura 35).

¹ <https://unity.com/es>

² <https://unity.com/es/unity-hub>

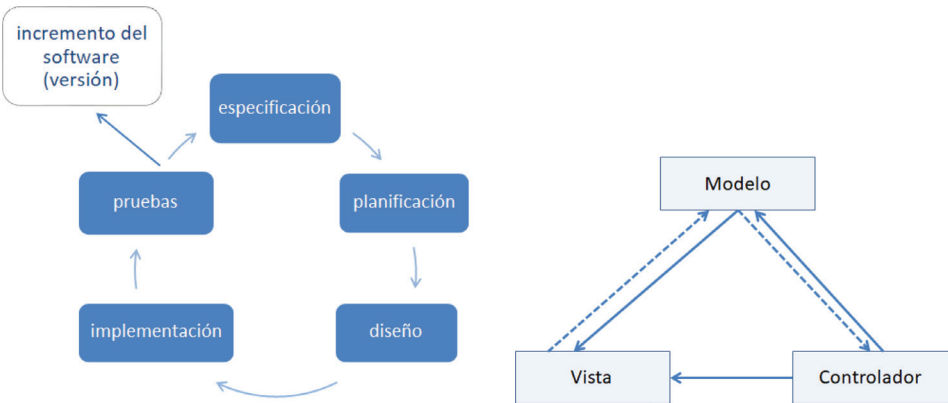
³ <https://visualstudio.microsoft.com/es/>

Figura 41. Esquema de arquitectura REST.



Para el desarrollo del juego, se siguió un proceso unificado ágil (PU), con una estrategia de desarrollo iterativa [183] en la que se van atravesando las fases clásicas del PU, las cuales contienen varias iteraciones cada una, en las cuales se recorren las diferentes actividades de forma flexible, refinando el proyecto y obteniendo un producto ejecutable o software incremental al final de cada iteración, con algunas de las funciones de la aplicación. En las primeras iteraciones se comienza por construir las funciones elementales de la aplicación, con lo que predominan las disciplinas de especificación y planificación. En posteriores iteraciones, se va ampliando el producto y se va dando más o menos peso a otras disciplinas según sea necesario. Según el PU, se combina el ciclo de vida iterativo y el desarrollo dirigido por el riesgo en una descripción consistente y bien documentada. Así, cada una de las iteraciones da como resultado un incremento de software consistente en un subconjunto estable y ejecutable del producto final [184] como puede verse en la parte izquierda de la Figura 42.

Figura 42. Actividades de una iteración del PU y diagrama MVC.



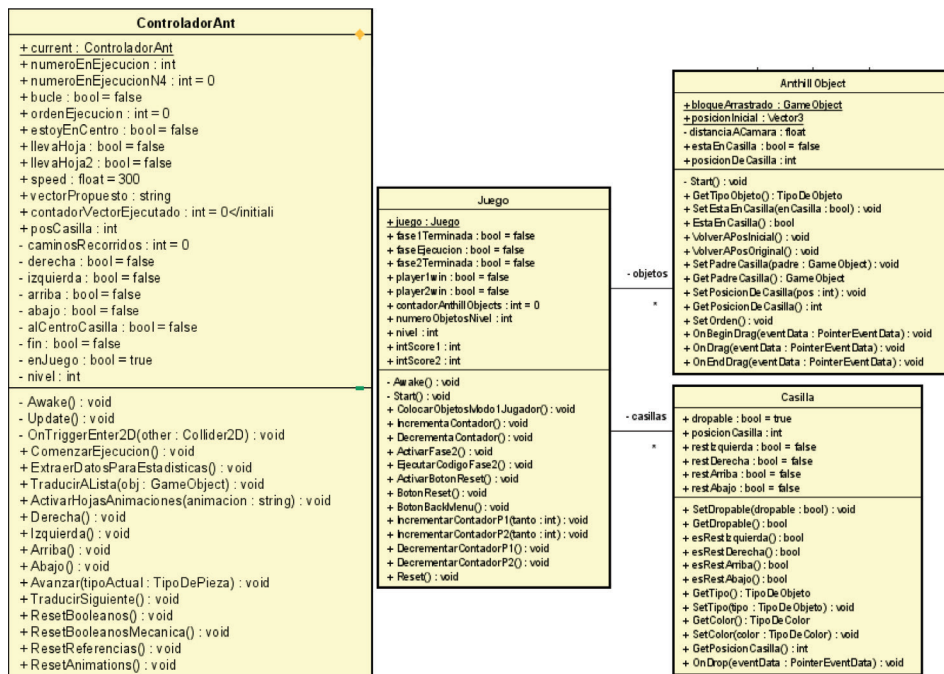
El motor de juego Unity utiliza como base el patrón de diseño entidad-componente (EC), en un nivel por encima del diseño orientado a objetos (POO), ya que, en Unity, a cada entidad (instancia de *GameObject*) se le puede asociar un componente de tipo script, donde se programaría el comportamiento de esa entidad. En diseño de videojuegos, este tipo de patrón se suele utilizar con considerable frecuencia. De esta forma, se suele definir la jerarquía de los elementos que forman parte de la aplicación o juego y después se definen los comportamientos de dichas entidades. Este tipo patrón es útil en el caso de haber herencia múltiple, ya que resuelve algunos problemas relacionados con ello.

Sin embargo, pese a que el patrón entidad-relación ayuda a desfragmentar y organizar el código, en proyectos complejos este tipo de estructura produce demasiada desfragmentación, disminuyendo la cohesión del proyecto y dificultando el desarrollo y las posibles actualizaciones futuras. Por ello, se ha considerado más adecuado abordar el proyecto bajo una perspectiva de patrón Modelo-Vista-Controlador (MVC), que separa el software en tres componentes: Modelo (datos/representación de la información), Vista (Interfaz/detección) y Controladores (Decisión/acción) (véase la Figura 42).

Este patrón de arquitectura de software MVC se basa en las ideas de reutilización de código y separación de conceptos, características que se buscan en este proyecto y que facilitan la tarea de desarrollo y de mantenimiento posterior. En el siguiente diagrama se muestra la relación entre las componentes del MVC. Las líneas sólidas indican asociación directa y las punteadas, indirecta:

El desarrollo de juegos tiene un flujo de acción típico de espera por una entrada del usuario u otra acción o disparador (*trigger*) que envía una notificación a los eventos apropiados y se decide qué hacer en respuesta, actualizando los datos correspondientes. Este flujo de acción concuerda perfectamente con el patrón MVC, ya que se divide el proceso en datos, interfaz y decisiones. Dado que el motor de juego Unity, tienen una estructura EC, como se ha mencionado anteriormente, algunas clases que encapsulan funcionalidades no encajan en el modelo MVC y se tratarán como componentes aislados, incrementando la cohesión y flexibilizando el patrón MVC para adaptarse al diseño de videojuegos con un motor como Unity. En la Figura 43 puede verse la clase que controla al personaje principal (ControladorAnt), las clases principales que regulan el progreso del jugador junto a la lógica de las casillas de cada laberinto y los objetos de su interior.

Figura 43. Algunas clases de BAC (diagrama simplificado de algunas clases, UML).



6.4.3. Base de datos y módulos de análisis

A medida que se va desarrollando una partida entre dos usuarios (modo experiencia en colegios) o sólo con un usuario (modo masivo), se van guardando los datos de juego en una base de datos SQL relacional. Estos datos también se guardan en un archivo de registro (log) como copia de seguridad local, ya que se encontraron problemas con la estabilidad de la conexión a internet de los colegios. Los archivos de registro que se van generando siguen la misma estructura que hay en la base de datos, de forma que, si posteriormente se desea copiar los datos a la base de datos, la tarea sea lo más automática posible. En la Tabla 29 se muestra el diagrama entidad-relación de la base de datos (para más información sobre cada campo de la base de datos ver anexo B.2).

Además de la API creada para la acceder a la base de datos, se han creado también los siguientes módulos asociados para el análisis de datos, de forma que se puedan realizar dicho análisis de forma sistemática y automatizada:

- Aplicación para el **tratamiento de ficheros locales** generados por BAC, en donde se analiza cada archivo local de registro (log). Esta aplicación se desarrolló con el

lenguaje Python y el entorno de desarrollo PyCharm. La aplicación recibe los datos de un puzle (partida) y devuelve la solución óptima mediante implementaciones del algoritmo A* adaptadas a cada nivel del juego (véase la Figura 44) [185].

Tabla 29. Diagrama entidad relación de la base de datos.

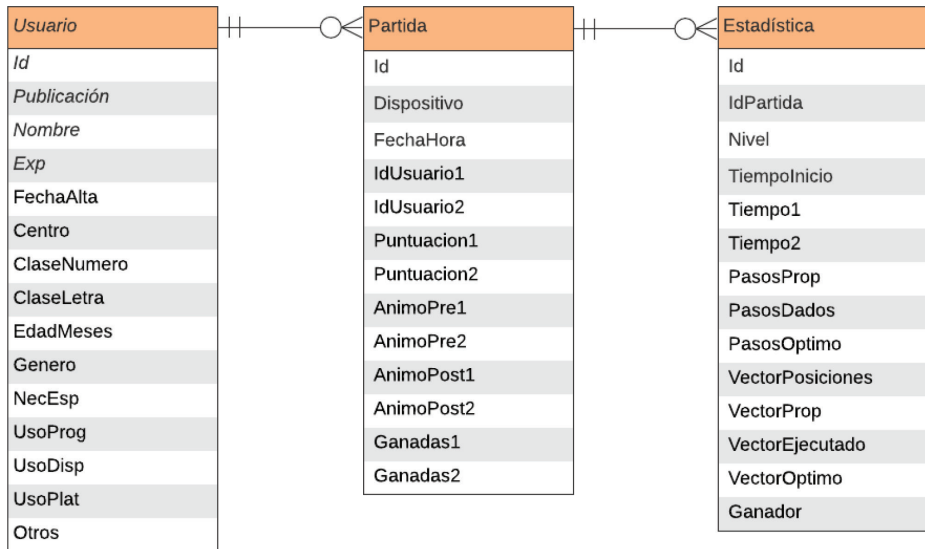
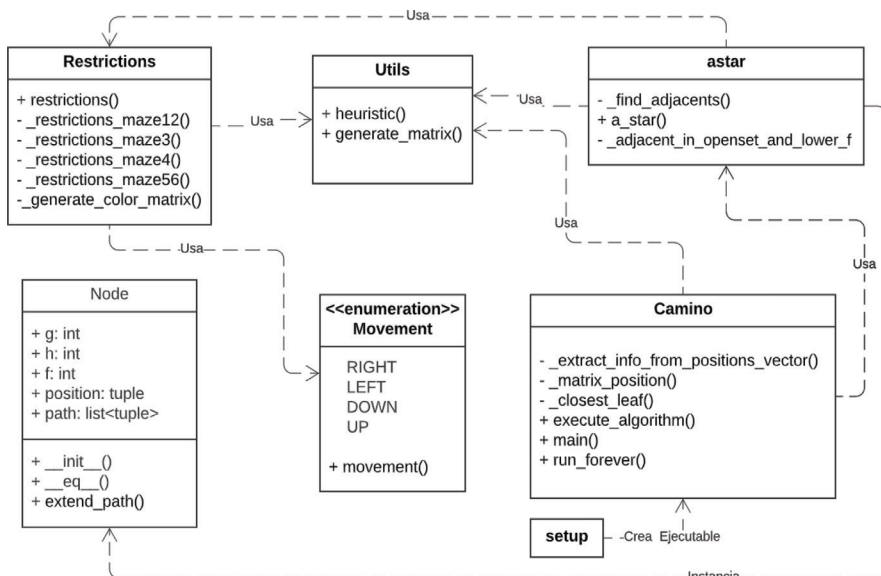
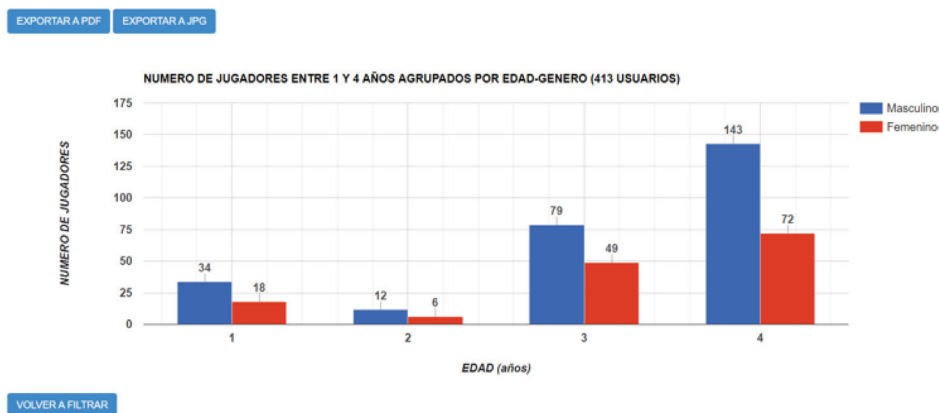


Figura 44. Diagrama simplificado del módulo para el tratamiento de ficheros locales generado por BAC.



- Aplicación web para facilitar la **visualización y análisis** de datos de BAC. Esta aplicación utiliza la arquitectura MVC, se desarrolló utilizando Spring y Google Charts, y muestra la información de la base de datos en forma de gráficas (véase la Figura 45) [186].

Figura 45. Ejemplo de gráfico de visualización de datos de BAC.



- Finalmente, para el **postproceso de datos** pormenorizado de la base de datos en un momento estático de tiempo, se ha realizado un módulo utilizando el entorno PyCharm con el lenguaje Python y las librerías matplotlib, pandas, sklearn (para la **clusterización** de los datos) y mpl_toolkits.

6.5. Caso de estudio en Educación Primaria

Para poder validar BAC como instrumento de aprendizaje y evaluación y, además, responder a las preguntas de investigación, se realizó un caso de estudio en 3 colegios diferentes, en la etapa de Educación Primaria, utilizando BAC en modos individual y colaborativo.

6.5.1. Sistema de evaluación propuesto

Como se vio en el apartado 2.5, evaluar el PC es fundamental para la inclusión del PC en el currículo y hacerlo mediante *sistemas de evaluación* es recomendable para así evaluar un aprendizaje más profundo combinando diferentes medidas de datos [148], [161]. En este caso, planteamos un *sistema*

de evaluación compuesto por tres medidas de datos que se enumeran a continuación, junto la descripción del impacto esperado de cada uno de los enfoques de evaluación en el marco 3D de referencia según la hipótesis HE4 (véase la Figura 11):

- 1) **BCTt** [8]: el instrumento independiente del entorno BCTt, desarrollado y validado en el marco de esta tesis (capítulos 4 y 5), está enfocado a la etapa de Educación Primaria, se centra en los conceptos computacionales y, parcialmente, en las prácticas computacionales, mientras que ignora las perspectivas computacionales.
- 2) **Entorno BAC**: en apartados previos se subrayó la idoneidad de evaluar el PC a través de información dinámica en tiempo real a través de GLA y de esta forma analizar las habilidades y progresión del alumno a lo largo del tiempo. A través del análisis de las acciones o del código propuesto por el alumno en el entorno BAC, se podrían identificar las prácticas computacionales utilizadas [187] y, a través de las acciones realizadas (por ejemplo, analizando el estilo de juego de los alumnos en el modo colaborativo), se podrían evaluar en cierta medida las perspectivas computacionales. La evaluación del PC a través de GLA está en consonancia con la teoría del flujo, ya que el aprendizaje a través del juego puede continuar sin problemas mientras la recogida de datos se maneja de forma discreta para que el flujo se mantenga [188][190].
- 3) **Datos cualitativos**: mediante la recogida de datos cualitativos, a través de cuestionarios y observación directa de los alumnos mientras interactuaban con el entorno y sus compañeros se pueden evaluar las perspectivas computacionales del marco 3D. Para ello, se elaboraron varios cuestionarios para completar y comparar los datos recogidos, en concreto los relacionados con la motivación de los alumnos, sus características, habilidades, intereses y cuestiones de contexto (véase la Tabla 30). Estos cuestionarios estaban dirigidos tanto a los alumnos como a los profesores y se utilizaron en diferentes momentos de la experiencia. Los resultados procedentes de los cuestionarios se reportan descriptivamente sólo para ítems únicos e independientes (no para conjuntos de ítems que supuestamente componen una escala). En este sentido, no es necesario analizar la fiabilidad de los cuestionarios en su conjunto.

Tabla 30. Resumen del contenido de los cuestionarios.

Número	Objetivo	Punto temporal	Submuestra	Título descriptivo
1	Estudiantes	0	A, B, C	Datos del alumno (por ejemplo, edad, sexo) y experiencia previa con juegos, dispositivos y aplicaciones digitales y programación
2	Estudiantes	0	A, B, C	Rendimiento y calificaciones anteriores del colegio, preferencias e intereses por las distintas asignaturas y aspectos motivacionales antes del periodo de juego
3	Estudiantes	2	A, B	Aspectos motivacionales y preferencias después del periodo de juego
4	Profesores	0	A, B, C	Características de los alumnos, necesidades especiales, rendimiento escolar y otros
5	Profesores	0	A, B, C	Características de la escuela, metodologías y materias relacionadas con la programación
6	Profesores	0	A, B, C	Preocupaciones de los profesores sobre la programación y las metodologías basadas en el juego (antes de jugar)
7	Profesores	2	A, B	Preocupaciones y opiniones de los profesores sobre el entorno BAC después del periodo de juego

6.5.2. Participantes y procedimiento

Los participantes en este estudio fueron una muestra de 176 alumnos de primaria de tres colegios públicos españoles. Como se muestra en la Tabla 31, la investigación se centró en una etapa educativa de cada colegio. El procedimiento de muestreo es intencional y la muestra se subdivide según las clases naturales en los colegios, tal y como se muestra en la Tabla 32.

Tabla 31. Ciclo de Educación Primaria respecto a cada colegio y edad de los alumnos.

Número de colegio	Ciclo de Educación Primaria	Cursos	Edad de los alumnos
Colegio 1	Primero	1º y 2º	5 - 8
Colegio 2	Segundo	3º y 4º	7 - 10
Colegio 3	Tercero	5º y 6º	9 - 12

Tabla 32. Número de estudiantes (n) en cada submuestra.

Descripción	Identificador	Ciclo de Educación Primaria		
		Primero	Segundo	Tercero
Modo individual	A	A1: n = 23	A2: n = 28	A3: n = 25
Modo colaborativo	B	B1: n = 23	B2: n = 26	B3: n = 23
Grupo de control	C		C2: n = 28	

Para garantizar que la investigación se realice en las mismas condiciones en cada escuela, se elaboró un protocolo de actuación (véase el anexo B.1). Como se muestra en la Tabla 33, en el momento 0 los alumnos y los profesores rellenaron los cuestionarios (Tabla 30). En consecuencia, el BCTt se administró como pre-test de forma concurrente a cada sujeto siguiendo el protocolo de actuación. Para garantizar que las habilidades o la experiencia previa de los estudiantes en el uso de dispositivos informáticos no interfirieran en los resultados del test, los BCTt se imprimieron y se rellenaron por los estudiantes en papel.

Tabla 33. Submuestras consideradas e instrumentos utilizados en los diferentes puntos de tiempo de la investigación (por semanas).

Instrumento		Submuestras			
Cuestionarios	A, B, C		A, B		
BCTt	A, B, C		A, B, C		A, B
BAC			A, B		
Semana	0	1 a 5	6	7 a 15	16
	Tiempo 0 Pre-test	Tiempo 1	Tiempo 2 Post-test		Tiempo 3 retención

A continuación, los sujetos de las submuestras A y B jugaron con el entorno BAC durante 5 semanas utilizando tabletas digitales. Cada semana, cada sujeto de cada submuestra jugó durante 10 minutos, independientemente del modo de juego. La submuestra A jugó en el modo individual; la submuestra B jugó en el modo colaborativo, en el que dos estudiantes emparejados al azar juegan al mismo tiempo en la misma tableta digital, uno de los cuales asume el papel de establecer el reto mientras el otro intenta resolver el problema planteado. Los roles en el modo colaborativo se intercambiaron entre los estudiantes para que cada sujeto asumiera ambos roles de forma equitativa. Durante el tiempo de juego (Tiempo 1), los datos de juego cuantitativos de cada estudiante fueron recogidos automáticamente por BAC, sin alterar el flujo del juego, y guardados en la base de datos remota. También se recogieron datos cualitativos durante el tiempo de juego a través de la observación directa, en cuanto a la forma de reflexionar ante los retos propuestos, la interacción con los compañeros o las dificultades que expresaron oralmente.

Al final del periodo de juego de 5 semanas (tiempo 2), se volvió a administrar el BCTt a los sujetos de las submuestras A y B, como post-test, así como a la submuestra C como grupo de control. Además, los alumnos y los profesores de las submuestras A y B rellenaron los cuestionarios.

Por último, 10 semanas después de la finalización del periodo de juego, en el momento 3, se administró el BCTt a los sujetos de las submuestras A y B, para comprobar los aspectos relacionados con la retención del aprendizaje.

Al final de la semana 16, el equipo de investigación analizó cuidadosamente todos los datos cuantitativos y cualitativos recogidos.

6.6. Resultados y discusión

A continuación, se describen los resultados de la experiencia en los colegios.

6.6.1. BCTt

El BCTt se administró en los tiempos 0, 2 y 3 (véase la Tabla 33). Como se esperaba, la fiabilidad como consistencia interna asociada a las puntuaciones del BCTt es alta (Alfa de Cronbach, submuestras A y B: tiempo 0, $\alpha = .803$; tiempo 2, $\alpha = .801$; tiempo 3, $\alpha = .837$). A continuación, se analizaron los datos del BCTt recogidos para determinar si se había producido el aprendizaje esperado. Considerando la puntuación del BCTt como la suma de todas las respuestas correctas a lo largo de los 25 ítems del test, se realizó un análisis inicial de los resultados de la puntuación en el pre-test. Dado que la población de la muestra parece estar distribuida normalmente, los sujetos se dividieron intencionadamente en submuestras (véase la Tabla 32). A continuación, se aplicó la prueba *t* de Student a los resultados, por separado en cada etapa educativa, para las submuestras A y B (sujetos que jugarían con BAC en una etapa posterior: A: modalidad de juego individual; B: modalidad de juego colaborativo), con el fin de evaluar la significación estadística de la diferencia entre las dos medias poblacionales. Los resultados no muestran diferencias significativas en las puntuaciones de las pruebas ($p > .05$) entre los grupos del mismo grado, por lo que las submuestras A y B están equilibradas y pueden ser comparadas en una fase posterior como se muestra en la Tabla 34.

Tabla 34. Resultados de la prueba *t* de Student de la puntuación del BCTt como pre-test.

Ciclo de Educación Primaria	Curso	Edad de los alumnos	Submuestra	<i>n</i>	Media	Desviación estándar	<i>t</i>	<i>p</i>
Primera	1	5 - 7	A1	23	17.00	3.03	0.24	0.81
			B1	23	16.78	3.03		
Segunda	4	8 - 10	A2	28	21.79	2.99	0.53	0.60
			B2	26	21.35	3.15		
Tercera	5	9 - 11	A3	25	21.40	2.47	-	0.12
			B3	23	22.43	1.95		

Tras el periodo de juego de cinco semanas, se realizó un análisis primario de los resultados: se aplicó la prueba *t* de Student entre los resultados de las puntuaciones del pre-test y del post-test, obteniendo en general mejores

resultados en las puntuaciones medias del post-test en las submuestras A y B (véase la Tabla 35). Dado que $p < .05$, existe una significación estadística en la diferencia entre las puntuaciones del pre-test y post-test en casi todas las submuestras A y B, en comparación con los resultados obtenidos entre pre y post-test en la submuestra C (grupo de control) en la que no hubo una ganancia de puntuación significativa ($p > .05$). Estos resultados nos llevan a concluir que se produjo un aprendizaje efectivo durante el periodo de juego, considerando las puntuaciones globales de cada alumno, tanto en la modalidad individual como en la colaborativa; sin embargo, en la modalidad colaborativa de 5º curso (submuestra B3), no se observó una ganancia significativa ($p = .878 > .05$). Esto era de esperar, ya que los resultados del pre-test en esta submuestra ya eran altos, lo que dejaba poco margen de mejora y, además, los alumnos en general jugaron menos tiempo en la modalidad colaborativa; sin embargo, una hipótesis alternativa es que los alumnos mayores aprovechan menos la modalidad colaborativa que los más jóvenes.

Además, el tamaño del efecto fue grande (d de Cohen $> 0,8$) en los cursos inferiores y medio en los superiores (d de Cohen $> 0,5$), quizá debido a que el BCTt tiene una fiabilidad menor en los alumnos de más edad, excepto en la submuestra B3, que es un resultado muy favorable, ya que el tiempo total de juego fue de sólo 50 minutos por alumno (diez minutos cada semana).

Los resultados de las puntuaciones en el BCTt en todos los cursos fueron más altos de lo esperado, e incluso se observó un efecto techo en los grados superiores, como se puede observar en la Tabla 36, por lo que se llevó a cabo un segundo análisis, centrándose exclusivamente en los sujetos de percentiles bajos en cada submuestra experimental (calculado a partir de los resultados del pre-test). Los resultados muestran ganancias muy significativas entre las puntuaciones del pre-test y el post-test cuando se consideran los primeros cuartiles: Q1 (véase la Tabla 37), y Q2 (véase la Tabla 38), en todas las submuestras y modalidades de juego, considerando $\alpha = .01$ (prueba t de Student entre las puntuaciones de la modalidad individual pre-test y post-test Q1: grado 1, $p = .0065$; grado 5, $p = .0015$). La única excepción fue en la modalidad colaborativa de 5º curso, en la que no hubo una ganancia significativa considerando $\alpha = .01$ ($p = .0248$), pero sí cuando se consideró $\alpha = .05$. Resultados similares se obtuvieron en el cuartil Q2, que muestra resultados significativos entre las puntuaciones pre-test y post-test en todas las submuestras, considerando $\alpha = .01$ (prueba t de Student entre las puntuaciones pre-test y post-test de la modalidad individual: curso 1, $p = .0042$; curso 5, $p = .0007$), excepto, de nuevo, en la modalidad colaborativa de 5º curso ($p = .447$).

Tabla 35. Comparación entre los resultados de las puntuaciones del pre-test y del post-test entre las distintas submuestras dependiendo del modo de juego (prueba t de Student).

Curso	Modo de juego BAC	Submuestra	n	Modo BCTT	Media	Desv. Std.	t	p	Tamaño del efecto Cohen d
1	Individual	A1	23	pretest	17.00	3.03	-2.63	0.012	0.903
				posttest	19.74	3.97			
	Colaborativo	B1	23	pretest	16.78	3.03	-2.29	0.027	0.804
				posttest	18.96	3.40			
4	Individual	A2	28	pretest	21.79	2.99	-3.38	0.001	0.522
				posttest	23.89	1.40			
	Colaborativo	B2	26	pretest	21.35	3.15	-3.83	0.000	0.615
				posttest	23.92	1.35			
	Grupo de control	C2	28	pretest	22.29	2.27	-0.372	0.711	0.292
				posttest	22.54	2.73			
5	Individual	A3	25	pretest	21.40	2.47	-2.62	0.012	0.619
				posttest	22.96	1.67			
	Colaborativo	B3	23	pretest	22.43	1.95	-0.15	0.878	0.037
				posttest	22.52	1.88			

Tabla 36. Percentiles por curso.

Curso	Submuestra	Mínimo	Máximo	Media	Desviación estándar	Percentiles		
						25	50	75
1	A1	12.00	24.00	17.00	3.03	14.00	17.00	20.00
	B1	11.00	23.00	16.78	3.03	15.00	16.00	19.00
4	A2	14.00	25.00	21.79	2.99	19.00	23.00	24.00
	B2	15.00	25.00	21.35	3.15	18.00	22.50	24.00
5	A3	16.00	25.00	21.40	2.47	19.50	22.00	23.50
	B3	19.00	25.00	22.43	1.95	21.00	23.00	24.00

Además, se realizó una tercera comparación, esta vez dividiendo los ítems del BCTt según los conceptos computacionales que se abordan, y contando cuántos estudiantes contestaron correctamente los ítems en cada conjunto (es decir, si la submuestra es $n = 23$, la puntuación de 21 puntos en un conjunto significa que 21 de los 23 estudiantes lo contestaron correctamente).

Esta nueva puntuación, que está relacionada con cada concepto computacional, muestra resultados interesantes, ya que los ítems tienen una dificultad creciente y se esperan mayores ganancias en la puntuación en los conceptos más difíciles. De hecho, esta tendencia se confirmó empíricamente. Hubo una ganancia significativa en la prueba t de Student en los últimos 4 conceptos considerando $\alpha = .01$ en cada submuestra experimental, por lo que el aprendizaje efectivo tiene lugar particularmente en relación con los ítems más difíciles del BCTt. La Figura 46 muestra las mejoras en la puntuación en el BCTt (diferencia de puntuación entre el pre-test y el post-test). Por ejemplo, para el nivel 6, la media de puntuación de la submuestra A1 fue de 11.90 en el pretest y de 18.20 en el post-test, por lo que la mejora en la puntuación fue de 6.30 puntos) para cada conjunto de conceptos computacionales (submuestras A), por lo que hubo una correlación positiva entre la dificultad de los ítems y la mejora en la puntuación observada (coeficiente de determinación: A1: $R^2 = .90$; A2: $R^2 = .88$; A3: $R^2 = .69$). La Figura 46 también muestra que hay mayores incrementos de puntuación para cada concepto computacional en los cursos inferiores (A1) y que esto disminuye proporcionalmente en los cursos superiores (A2, A3).

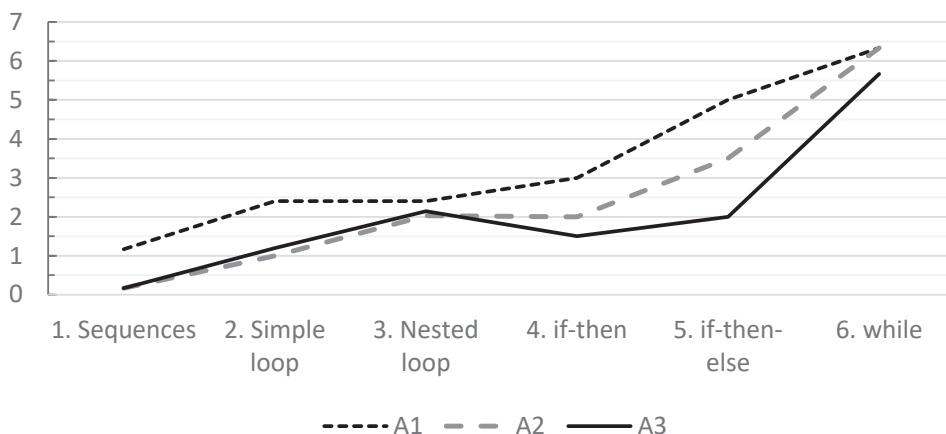
Tabla 37. Comparación entre los resultados de las puntuaciones del pre-test y del post-test entre las diferentes submuestras dependiendo de la modalidad de juego (prueba t de Student). Q1, percentil 25.

Curso	Modo de juego BAC	Submuestra	Perc.	n	Modo BCT	Media	Desv. Std.	t	p	Tamaño efecto d Cohen	
1	Individual	A''1	< 25	6	pre-test	13.50	0.84	-3.42	.007	2.690	
					post-test	17.50	2.74				
	Colaborativo	B''1		8	pre-test	13.88	1.36	-2.66	.019	2.925	
					post-test	17.13	3.18				
4	Individual	A''2	< 25	9	pre-test	17.89	1.62	-8.28	.000	3.045	
					post-test	23.89	1.45				
	Colaborativo	B''2		7	pre-test	16.86	1.21	-7.70	.000	5.242	
					post-test	23.43	1.90				
5	Individual	A''3	< 25	6	pre-test	17.83	1.17	-4.32	.002	4.345	
					post-test	21.83	1.94				
	Colaborativo	B''3		6	pre-test	19.67	0.82	-2.64	.025	1.859	
					post-test	22.33	2.34				

Tabla 38. Comparación entre los resultados de las puntuaciones del pre-test y del post-test entre las diferentes submuestras dependiendo de la modalidad de juego (prueba t de Student). Q2, percentil 50.

Curso	Modo de juego de BAC	Submuestra	Perc.	n	Modo BCTt	Media	Desv. Std.	t	p	Tamaño efecto d Cohen
1	Individual	A'1	< 50	13	pre-test	14.92	1.61	-3.16	.004	1.666
					post-test	18.54	3.80			
	Colaborativo	B'1		13	pre-test	14.69	1.49	-2.90	.008	2.053
					post-test	17.77	3.52			
4	Individual	A'2	< 50	17	pre-test	20.12	2.71	-4.45	.000	0.812
					post-test	23.53	1.62			
	Colaborativo	B'2		13	pre-test	18.77	2.39	-6.66	.000	2.304
					post-test	24.00	1.53			
5	Individual	A'3	< 50	15	pre-test	19.87	1.96	-3.82	.001	1.292
					post-test	22.47	1.77			
	Colaborativo	B'3		17	pre-test	21.65	1.62	-0.77	.447	0.197
					post-test	22.12	1.93			

Figura 46. Mejora de la puntuación en el BCTt (eje de ordenadas: puntuación post-test menos puntuación pre-test) por conjunto concepto computacional. Comparación entre submuestras A (modalidad de juego individual).



Se observó que se necesitó un tiempo considerablemente menor para realizar el post-test que para realizar el pretest; de hecho, los tiempos se redujeron a la mitad en todas las submuestras experimentales. Este resultado es coherente con la mejora de las habilidades de PC, en particular la automatización [191].

Finalmente, se analizaron los resultados obtenidos en la tercera administración del BCTt para comprobar aspectos relacionados con la retención de lo presuntamente aprendido (véase la Tabla 39). Se compararon las puntuaciones del pretest y del test de retención para comprobar si, tras un lapso de diez semanas de inactividad, el aprendizaje previamente adquirido se mantenía y seguía siendo significativo. Del mismo modo, también se compararon las puntuaciones de post-test y de la prueba de retención, para ver si se había producido alguna pérdida significativa de aprendizaje. Las comparaciones entre las submuestras A y B del pretest y la prueba de retención muestran ganancias de puntuación significativas ($\alpha = .05$) en todas las submuestras a pesar del lapso de tiempo, excepto en la submuestra B3, como fue el caso entre las puntuaciones del pre-test y el post-test (prueba *t* de Student entre las puntuaciones del pre-test y la prueba de retención: A1, $p = 0,042$; A2, $p = 0,011$; A3, $p = 0,043$; B1, $p = 0,040$; B2, $p = 0,000$; B3, $p = 0,889$). Los tamaños del efecto (d de Cohen) entre el pre-test y la prueba de retención en todos los grados (excepto en la submuestra B3) estaban en la zona de efectos deseados de Hattie (2009) ($d > 0,4$); además, los tamaños del efecto en las submuestras correspondientes al modo colaborativo de los cursos inferiores (B1: $d = 0.694$; B2: $d = 0.700$) eran mayores que los de las submuestras individuales (A1: $d = 0.548$; A2: $d = 0.555$). Estos resultados reflejan la retención de lo aprendido en todos los grupos, especialmente en las submuestras colaborativas, a pesar de que había un cansancio acumulado ya que la prueba de retención se realizó al final del curso escolar.

Tabla 39. Comparación entre los resultados de las puntuaciones del pre-test y el test de retención y del post-test y el test de retención entre las diferentes submuestras dependiendo de la modalidad de juego.

Curso	Modo de juego de BAC	Submuestra	n	Modo de BCTt	Media	Desv. Std.	t	p	Tamaño efecto d Cohen
1	Individual	A1	23	pre-test	17.00	3.03	-2.098	.042	0.548
				test de retención	18.83	2.87			
				post-test	19.74	3.97	0.894	.376	-0.258
	Colaborativo	B1	23	test de retención	18.83	2.87			
				pre-test	16.78	3.03	-2.115	.040	0.694
				test de retención	18.57	2.68			
4	Individual	A2	28	post-test	18.96	3.40	0.433	.667	-0.168
				test de retención	18.57	2.68			
				pre-test	21.79	2.99	-2.648	.011	0.555
	Colaborativo	B2	26	test de retención	23.46	1.53			
				post-test	23.89	1.40	1.096	.278	-0.310
				test de retención	23.46	1.53			
5	Individual	A3	25	pre-test	21.35	3.15	-3.856	.000	0.700
				test de retención	23.85	1.01			
				post-test	23.92	1.35	0.232	.817	-0.070
	Colaborativo	B3	23	test de retención	23.85	1.01			
				pre-test	21.40	2.47	-2.080	.043	0.435
				test de retención	22.76	2.15			
	Individual			post-test	22.96	1.67	0.368	.715	-0.101
				test de retención	22.76	2.15			
				pre-test	22.43	1.95	-0.141	.889	0.044
	Colaborativo			test de retención	22.52	2.23			
				post-test	22.52	1.88	0.000	1	0.000
				test de retención	22.52	2.23			

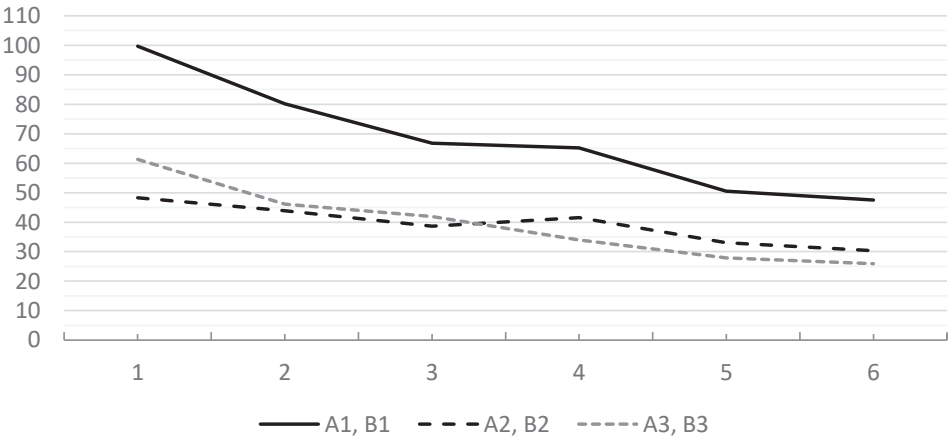
Además, las comparaciones entre el post-test y la prueba de retención en las submuestras A y B no muestran una pérdida de aprendizaje significativa (considerando $\alpha = .05$) en ninguna de las submuestras (prueba *t* de Student entre las puntuaciones del post-test y la prueba de retención: A1, $p = .376$; A2, $p = .278$; A3, $p = .715$; B1, $p = .667$; B2, $p = .817$; B3, $p = 1$). Hay que destacar que en todas las submuestras B hubo menos pérdida de puntuación entre el post-test y el test de retención que en las submuestras A, así como menores tamaños del efecto, especialmente en los grados inferiores y en los percentiles bajos. Esto nos lleva a concluir que la modalidad de juego colaborativo fomenta un aprendizaje más eficaz y duradero, tal y como se esperaba [89], aunque los resultados iniciales del pre-test y el post-test puedan parecer acompañados de incrementos de puntuación más bajos en comparación con la modalidad individual.

Al analizar los resultados de las submuestras de percentil bajo, se observaron resultados más favorables en todos los casos, considerando $\alpha = .01$, aumentos de puntuación significativos entre el pre-test y la prueba de retención a pesar del lapso de tiempo, que fue aún más significativo en las submuestras del modo colaborativo (por ejemplo, la diferencia entre el pre-test de la submuestra de percentil B3 Q2 y la prueba de retención fue de $p = .0093$, a pesar de ser el peor caso anterior); y no hubo pérdidas significativas (considerando $\alpha = .05$) entre el post-test y la prueba de retención. Estos resultados nos llevan a concluir que la población de percentil bajo y los estudiantes más jóvenes se benefician aún más de la herramienta de aprendizaje, y especialmente del modo de juego colaborativo.

6.6.2. Entorno BAC

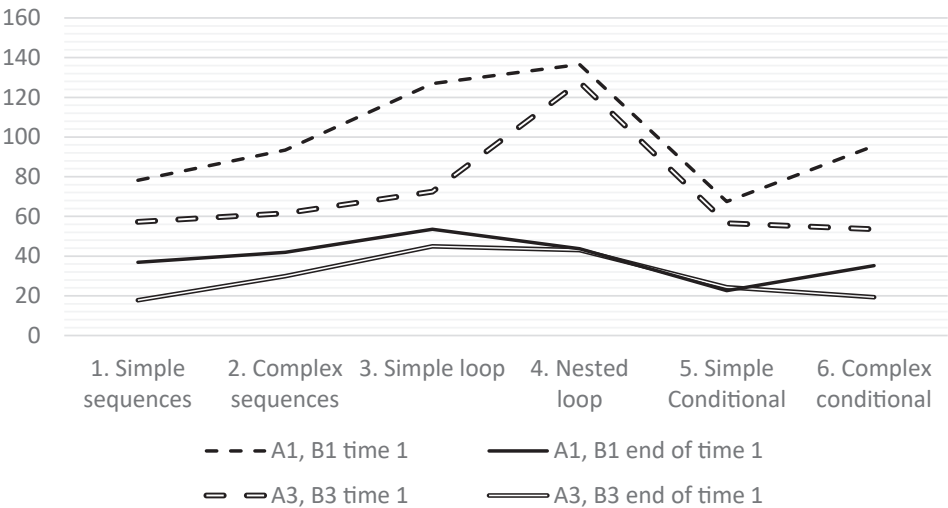
Se llevó a cabo un análisis preliminar de los datos recogidos en el entorno BAC mientras los estudiantes jugaban. En primer lugar, independientemente del modo de juego o del nivel jugado, se analizó para cada submuestra el tiempo medio necesario para resolver el problema (en el modo colaborativo, el tiempo del jugador 2: alumno que resuelve el reto) a lo largo de la partida. Como muestra la Figura 47, existe una correlación negativa en todas las submuestras entre el número de partidas jugadas y el tiempo medio de resolución del problema (coeficiente de determinación: A1, B1: $R^2=0.935$; A2, B2: $R^2=0.897$; A3, B3: $R^2=0.937$).

Figura 47. Tiempo medio empleado en segundos (eje de ordenadas) para resolver el problema en cada partida jugada: siendo el 1 la primera partida (eje de abscisas), por submuestras.



Asimismo, se comparó el tiempo medio de resolución de cada nivel entre los alumnos de cada curso escolar, a lo largo del periodo de juego. En la Figura 48, las submuestras A1 y B1 se comparan con las submuestras A3 y B3 según el tiempo medio empleado, en segundos, para resolver el problema de cada nivel del juego en dos momentos: tiempo 1 - primer momento, el primer día del periodo de juego y, final del tiempo 1 - último momento, tras un periodo de juego de 5 semanas. Estos resultados nos llevan a concluir que el aprendizaje efectivo se produjo progresivamente a lo largo del periodo de juego. Hay que destacar especialmente que en las submuestras A1 y B1 (primer curso), en el tiempo 1, los alumnos tardaron más en resolver el problema, en todos los niveles, que los alumnos de las submuestras A3 y B3; pero, al final del tiempo 1, cuando terminó el periodo de juego, los sujetos de las submuestras A1 y B1 fueron capaces de resolver los problemas en casi el mismo tiempo que los alumnos de las submuestras A3 y B3. De hecho, en algunos niveles incluso necesitaron menos tiempo. Este resultado es coherente con los resultados del BCTt, donde se observaron mejoras mucho mayores en los estudiantes más jóvenes. También podría añadirse que las habilidades computacionales adquiridas por los alumnos de primer curso son similares o incluso mejores (en lo que respecta a ciertos conceptos) que las de los alumnos de mayor edad.

Figura 48. Tiempo medio empleado en segundos (eje de ordenadas) para resolver el problema por concepto computacional (eje de abscisas), en dos momentos (momento 1: primer día del periodo de juego; final del momento 1: tras 5 semanas de periodo de juego).



6.6.3. Datos cualitativos

Para evaluar las perspectivas computacionales del marco 3D, se administraron cuestionarios y se recogieron datos cualitativos mediante la observación directa mientras los alumnos jugaban. Las perspectivas computacionales implican que los estudiantes desarrollen una comprensión de sí mismos y de sus relaciones con los demás y con el mundo tecnológico. De esta forma, trabajar por parejas puede mejorar las prácticas y perspectivas computacionales del PC, como sugieren Hsu *et al.* [4] y también confirma este estudio.

Los datos de los cuestionarios muestran que las escuelas participantes están moderadamente involucradas con los ordenadores y la tecnología. Aunque hay pizarras digitales en todas las aulas, así como salas de informática, el aprendizaje de la programación no se considera una asignatura en las etapas de Educación Primaria. Además, los profesores de primaria declaran tener un nivel muy bajo de conocimientos informáticos (16.7%), bajo (16.7%) o medio (66.7%). Sin embargo, el 100% cree que aprender a programar para hacer frente a las demandas del futuro es de muy alta (33.3%) o alta importancia (66.7%). Independientemente de la experiencia previa con juegos (el 76.6% de los alumnos ha jugado antes), aplicaciones de programación (el 83.7% de los alumnos no conocía ninguna, y el 83% nunca ha programado) o dispositivos digitales (el 95.5% de los alumnos utiliza un dispositivo digital de forma habitual); el 70.2% de los alumnos considera importante aprender a programar

y el 52.2% piensa que los videojuegos podrían ayudarles a aprender esta habilidad.

Se observó una fuerte motivación e interés por jugar con BAC desde el principio, y se mantuvo durante toda la experiencia, independientemente del modo de juego. Esto es coherente con los resultados de los cuestionarios, en los que el 100% de los participantes de cada grupo de muestra disfrutaron de la experiencia y quisieron seguir aprendiendo con entornos basados en juegos. Los profesores también informaron de que la experiencia fue positiva y agradable para los alumnos y pensaron que éstos sí aprendían con el BAC (83.3%).

En cuanto a los resultados procedentes de la observación directa, el nivel 4 de BAC (bucle anidado) fue en el que, en el momento 1, los alumnos informaron de más problemas para encontrar soluciones, lo que coincide con los datos cuantitativos que indican que es el nivel para el que los alumnos necesitan más tiempo (véase la Figura 48). Algunos alumnos de cursos inferiores intentaban evitar el nivel 4 y otros lo percibían como un reto e intentaban superarlo; por el contrario, todos los alumnos de cursos superiores preferían jugar a los niveles difíciles. No obstante, al final del tiempo 1, los tiempos necesarios para el nivel 4 se redujeron significativamente a un tercio. Se comprobó que los elementos de juego (como los objetivos claros, la complejidad, los retos, la competición o el andamiaje), estaban equilibrados y mejoraban el compromiso en el juego, ya que los alumnos elegían un nivel superior si superaban el anterior y un nivel inferior si no podían resolver el problema. En los cursos superiores, se mostró más interés por el sistema de puntuación como elemento motivador del juego, ya que los alumnos intentaban superar la puntuación de sus compañeros o la suya propia durante todo el tiempo de juego.

La preferencia por el modo de juego colaborativo se observó en todas las muestras, ya que el 100% de los alumnos manifestó su preferencia por este modo de juego. La comprensión y la dinámica entre los compañeros fueron muy positivas, a pesar de su falta de familiaridad con las estrategias de GBL. Se confirmó la hipótesis de que los alumnos, independientemente de su rol, aprenden los unos de los otros y que no es un juego competitivo sino colaborativo, ya que los alumnos interactúan y se ayudan mutuamente. La mayoría de los alumnos prefirieron el rol de resolver problemas. En los cursos superiores, los alumnos que establecían los problemas se esforzaban más por hacer los retos más difíciles para sus compañeros.

Todos los alumnos, independientemente del modo de juego, reflexionan mientras juegan. En el modo individual, piensan en voz alta, y en el modo colaborativo también discuten los posibles retos y soluciones con su compañero. Al final, los estudiantes reflexionan sobre el juego terminado; por

ejemplo, el que resuelve el reto suele aconsejar al compañero sobre cómo podría hacerlo más complejo. Además, la mayoría de los alumnos afirman que reflexionan sobre futuras partidas, por ejemplo, pensando en cómo plan-tear un reto más difícil o resolver un puzle específico la próxima vez. Se descubrió que esta necesidad de revisar y pensar en el proceso de programación de sus compañeros fomenta la revisión de su propio rendimiento de aprendizaje, lo que hace que los estudiantes se dediquen a pensar, como concluyeron Søndergaard y Mulder en su estudio [192]. Como sugieren Lye y Koh [61], se confirma que la revisión por pares puede mejorar las prácticas y perspectivas computacionales de las dimensiones del pensamiento computacional.

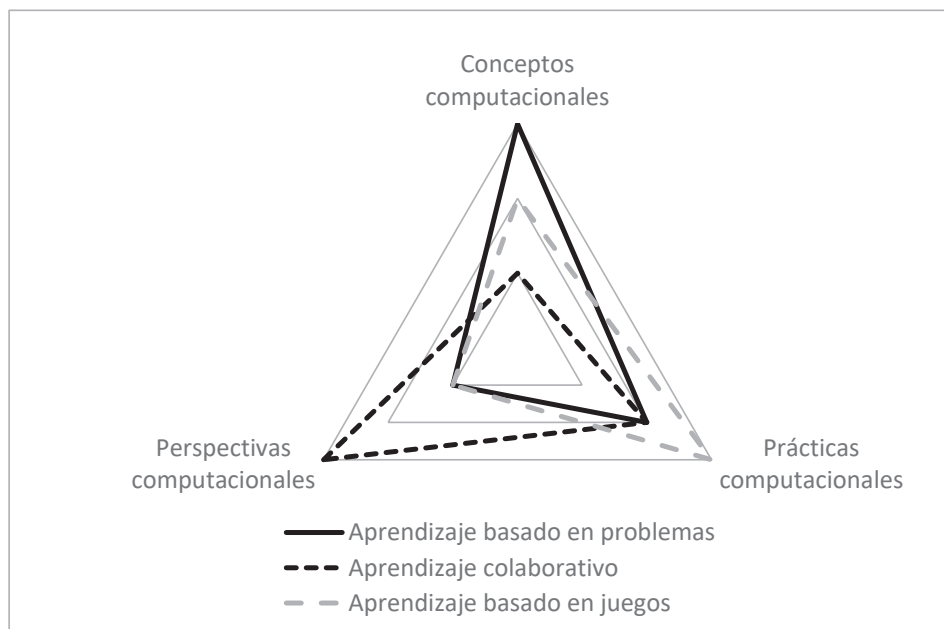
En todas las muestras experimentales, los alumnos con necesidades especiales (8.3% de los participantes), con bajos resultados académicos (15.1% de los participantes), o con un percentil bajo según los resultados del BCTt pre-post, mostraron un gran interés por el juego y motivación para superar los retos que les planteaba. Además, estos alumnos demostraron mejoras muy significativas en los resultados cuantitativos de la prueba BCTt pre-post. Estos resultados son coherentes con los obtenidos por Snodgrass *et al.*, que concluyen que los estudiantes con necesidades especiales pueden entrenar sus habilidades de PC utilizando entornos de programación visual [193]. Por ejemplo, un estudiante con discalculia y dislexia (submuestra B2) que no había obtenido buenos resultados en el pre-test del BCTt fue capaz de superar los retos del juego en un tiempo considerablemente menor que la media de la muestra B2. De hecho, mejoró significativamente su rendimiento en el post-test del BCTt y también mostró una notable mejora en su autoestima, sorprendiéndose de su capacidad para resolver los retos mejor que sus compañeros con mayor rendimiento escolar, en línea con la investigación de Laal y Ghodsi [89].

6.7. Conclusiones

El aprendizaje y la evaluación del PC en las escuelas se encuentra todavía en una fase experimental, especialmente en las edades tempranas. La mayoría de las investigaciones recientes se centran en las etapas escolares de educación secundaria y superiores, y en la dimensión del marco tridimensional de los conceptos computacionales. Sin embargo, no hay suficientes investigaciones sobre las edades tempranas y las otras dos dimensiones: las prácticas computacionales y las perspectivas computacionales [61]. En este estudio, se han combinado tres de las estrategias de aprendizaje más utilizadas en un entorno colaborativo basado en el juego, construido sobre una estrategia constructivista basada en problemas, que ha sido probado en la etapa de Educación Primaria. Los resultados muestran que este enfoque es adecuado para el aprendizaje del PC, especialmente en edades tempranas, bajo

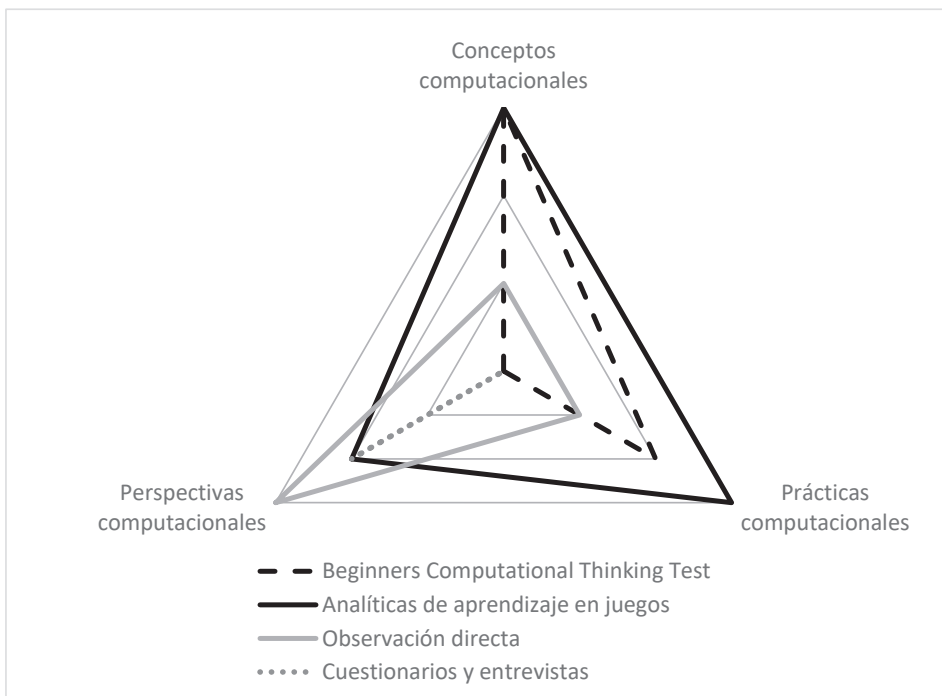
el marco de las 3D, cubriendo sus tres dimensiones clave como se planteó en la hipótesis *HE2*, y se ha confirmado mediante el caso de estudio (véase la Figura 49).

Figura 49. Impacto confirmado de la estrategia de aprendizaje en las dimensiones clave del marco 3D del PC.



La combinación de diferentes metodologías de evaluación, como un *sistema de evaluaciones* [148]: BCTt, GLA a través de GBL, y datos cualitativos (observación directa y cuestionarios), permite una evaluación tridimensional, como se planteó en la *HE4* y se ha confirmado mediante este caso de estudio (véase la Figura 50). En concreto, el BCTt evalúa los conceptos computacionales y, parcialmente, las prácticas computacionales. La técnica de GLA parece adecuada para evaluar los conceptos y las prácticas computacionales y, en parte, las perspectivas computacionales, aunque en mayor grado si el juego es colaborativo. Los cuestionarios y la observación directa, especialmente en la modalidad de juego colaborativo, cubren la evaluación de las perspectivas computacionales. Además, la evaluación del PC a través de GLA ha demostrado estar en consonancia con la teoría del flujo, ya que el aprendizaje a través del juego puede continuar con fluidez mientras que las evaluaciones se manejan discretamente manteniendo el flujo en todo momento.

Figura 50. Impacto confirmado de la estrategia de evaluación en las dimensiones clave del marco 3D del PC.



La mejora de la puntuación entre los test pre y post del BCTt fue significativa en las submuestras A1, B1, A2, B2 y A3, mientras que el tiempo dedicado a la realización del test se redujo a la mitad, lo que sugiere que los estudiantes mejoraron sus habilidades de PC después del periodo de juego, en contraste con la muestra de control que no mostró ninguna mejora significativa ni reducción de tiempo. Estos resultados son coherentes con el análisis de los datos recogidos por BAC, que muestran una mejora muy significativa a lo largo del tiempo. Los resultados similares obtenidos de BCTt y BAC de cada estudiante y muestra, revelan GLA como una herramienta de evaluación adecuada que, además, puede ofrecer datos en tiempo real que pueden ser útiles para una evaluación personal detallada, un fuerte análisis de datos de bajo nivel, y la identificación de conceptos computacionales inalcanzables para cada edad específica. Se obtuvieron resultados positivos en la prueba de retención, lo que sugiere que esta metodología basada en el juego fomenta el aprendizaje duradero. No obstante, el BCTt y el entorno BAC parecen estar mejor adaptados a los estudiantes más jóvenes, ya que se produjeron mayores mejoras en las habilidades de PC en las muestras de los cursos inferiores (primer ciclo de Educación Primaria) y no se observó ninguna mejora significativa

en la submuestra B3 (lo cual era de esperar, ya que en los cursos superiores algunos sujetos alcanzaron la puntuación máxima en el pre-test dejando poco margen de mejora). Dado que la investigación está dirigida a edades tempranas, este resultado desfavorable era de esperar, y suponemos que el BCTt y la BAC podrían incluso ser adecuadas para cursos inferiores de Educación Infantil.

Además, se encontró un resultado inesperado y relevante, ya que se observaron puntuaciones mayores al final del periodo de juego, tanto en el BCTt como en BAC en los estudiantes más jóvenes en comparación con los mayores. Aunque se observó una mejora significativa en la mayoría de los conceptos más difíciles en todas las submuestras; también fue mayor la mejora en la puntuación global, en cada concepto computacional, en los cursos escolares más bajos (A1) y hubo una disminución gradual de esta mejora en los cursos superiores (A2, A3). Además, las habilidades del PC de los alumnos de primer curso eran similares o incluso mejores en lo que respecta a ciertos conceptos, como los bucles anidados) que las de los alumnos de mayor edad después del periodo de juego. Esto nos lleva a sugerir que es importante desarrollar el PC desde edades muy tempranas y que es recomendable incorporar ciertos conceptos, como los bucles anidados o los condicionales, en el currículo ya en primer curso o incluso antes, en lugar de esperar hasta el último ciclo de Educación Primaria (o incluso secundaria), como ocurre actualmente en la mayoría de los centros educativos en España. Sin embargo, es necesario seguir investigando debido al posible sesgo en la selección de los colegios para cada ciclo educativo.

La investigación ha demostrado que los alumnos con necesidades especiales o de bajo percentil de cada muestra se benefician aún más de la herramienta de aprendizaje, y especialmente de la modalidad de juego colaborativo, ya que en estas submuestras se obtuvieron mejoras muy significativas (considerando $\alpha = .01$). Además, estos alumnos mostraron un gran interés por el juego y motivación para superar los retos. Esto nos lleva a concluir que la estrategia basada en el juego y los entornos de programación visual son adecuados y atractivos para estos estudiantes, lo que coincide con la investigación de Snodgrass *et al.* [193].

Se confirmó a través de la observación directa que todos los estudiantes reflexionan mientras juegan en ambos modos de juego, especialmente en el colaborativo. La reflexión y el juego colaborativo entre pares fomentan el rendimiento del aprendizaje y hacen que los estudiantes reflexionen; por lo tanto, la revisión entre pares puede mejorar las dimensiones de las prácticas y perspectivas computacionales del PC, como sugieren Lye y Koh [61]. Los resultados de retención del aprendizaje fueron mejores en la modalidad

colaborativa que en la individual, sobre todo en los cursos inferiores, lo que nos lleva a la conclusión de que la modalidad de juego colaborativo fomenta un aprendizaje más eficaz y duradero (aunque a partir de los resultados iniciales del pre-test y el post-test parece que se produce un incremento de las puntuaciones menor en comparación con la modalidad individual). Estos resultados están en consonancia con la investigación realizada por Laal y Ghodsi [87], en la que concluyen que el aprendizaje colaborativo se traduce no sólo en un mayor rendimiento y una mayor productividad, en relaciones más afectivas, solidarias y comprometidas, sino también en una mayor salud psicológica, competencia social y autoestima. Los resultados también son coherentes con Johnson y Johnson [88], y Laal y Laal [89] que sugieren que los equipos cooperativos retienen la información durante más tiempo que los alumnos que trabajan como individuos.

Hainey *et al.*, en su revisión bibliográfica sistemática de la investigación sobre el aprendizaje basado en el juego, concluyen que se necesitan más estudios que comparen las estrategias individuales y las colaborativas para identificar sus beneficios pedagógicos en la escuela primaria [175]. Los resultados obtenidos en esta investigación muestran que la estrategia colaborativa beneficia a los estudiantes más jóvenes, fomenta la reflexión a través de la comunicación entre pares, mejora las perspectivas computacionales (como se sugirió en investigaciones anteriores) y se consigue un aprendizaje más eficaz y duradero. Sin embargo, es posible que la estrategia colaborativa no sea tan eficaz para el último ciclo de Educación Primaria, ya que estos alumnos parecen ser capaces de una reflexión individual más eficaz. Se necesitan más investigaciones para comprobar esta hipótesis.

Esta investigación muestra resultados de aprendizaje significativos cuando se utilizan entornos informáticos educativos basados en juegos, especialmente con alumnos de bajo percentil, alumnos con necesidades especiales y en etapas educativas tempranas. Además, la combinación de estrategias de aprendizaje basadas en el juego y en la colaboración cubren las tres dimensiones del PC. Así, cuando se diseñan entornos basados en GBL para el desarrollo del PC y dirigidos a las primeras etapas educativas de primaria, se recomienda incluir estrategias de colaboración explícitas, teniendo siempre en cuenta la teoría del flujo y los elementos de diseño del juego para los alumnos más jóvenes. Por lo tanto, la hipótesis *HE3*, en la que se planteaba que los entornos basados en GBL potencian el aprendizaje del PC en Educación Primaria, en contextos escolares, sobre en los niños más pequeños y si se aplica una componente colaborativa, queda también confirmada. Sin embargo, cuando el entorno se dirige a las últimas etapas de primaria se recomienda complementarlas con actividades centradas en la reflexión individual.

6.8. Limitaciones y líneas de actuación

Se han detectado algunas limitaciones en este estudio. Se ha llegado a la conclusión de que los alumnos de 1º y 2º de primaria (primer ciclo), se benefician más de la estrategia de aprendizaje GBL que los alumnos de mayor edad, pero puede ser necesaria una investigación adicional sobre los alumnos de 3º de primaria para determinar con precisión a qué edad es más probable que los alumnos aprovechen este tipo de estrategias. Las investigaciones adicionales deben centrarse también en Educación Infantil, ya que no se ha establecido el límite de edad más bajo para utilizar entornos GBL. Además, sería recomendable replicar el estudio incluyendo grupos de control en cada grado. Dado que este estudio revela un alto beneficio del uso de GBL para los alumnos con necesidades especiales, también sería conveniente realizar más investigaciones sobre este aspecto. El uso de un *sistema de evaluación* pretende cubrir las tres dimensiones computacionales, pero es necesario un análisis más profundo para identificar exactamente qué prácticas y perspectivas computacionales se evalúan con cada enfoque. Además, sería esclarecedor replicar el estudio en otros países y poblaciones.

Aunque se han confirmado las hipótesis *HE2*, *HE3* y *HE4* y se han alcanzado los objetivos específicos de este capítulo, es necesaria más investigación en cuanto a las técnicas de GLA, incidiendo en la progresión en el juego y analizando los datos del juego con mayor profundidad y no sólo en contextos escolares, para así poder analizar disposiciones y predisposiciones para el desarrollo del PC, como se plantea en la *HE5*. Se avanzará en este sentido en el capítulo siguiente.

En cuanto a los objetivos generales de la tesis, se ha completado el objetivo OP3 y el OP4 parcialmente. En este punto, el grado de cumplimiento de los objetivos generales es el que se muestra en la Tabla 40.

Tabla 40. Seguimiento de los objetivos, capítulo 6.

Objetivos			Grado de cumplimiento		
			Bajo	Medio	Alto
OP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.				
	OP1	Marco teórico y estado del arte			
		OP1a Definiciones teóricas y operacionales del PC			
		OP1b Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas			
		OP1c Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes			
		OP1d Marco teórico y estrategias para la evaluación del PC e instrumentos existentes			
	OP2	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.			
		Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC			
	OP3	OP3a Diseñar y validar un instrumento de evaluación del PC en edades tempranas			
		OP3b Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas			

Objetivos			Grado de cumplimiento		
			Bajo	Medio	Alto
OC	OP4	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC			
		OP4a Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género			
		OP4b Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género			
		OP4c Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género			
	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones				

Los resultados de las experiencias realizadas en colegios de Educación Primaria presentadas en este capítulo se han recogido en la publicación:

- M. Zapata-Caceres, E. Martin, and M. Roman-Gonzalez, "Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study," in *IEEE Transactions on Learning Technologies*, vol. 14, no. 5, pp. 576-589, 1 Oct. 2021, doi: 10.1109/TLT.2021.3111108.

CAPÍTULO 7. ANÁLISIS DE COMPETENCIAS: MOTIVACIÓN INTRÍNSECA, PERSISTENCIA Y RECOMPENSAS MEDIANTE ANALÍTICAS DE APRENDIZAJE CON BAC

A continuación, se describe la experiencia de despliegue masivo de BAC en un entorno no escolar, con el objetivo general de analizar estrategias para el aprendizaje y evaluación del PC y las habilidades de PC de alumnos de Educación Primaria, con especial atención en las disposiciones y predisposiciones para el PC basadas en la motivación intrínseca y el juego voluntario.

7.1. Introducción y objetivos

Una vez validado desarrollado el entorno Blue Ant Code (BAC) mediante la experiencia en entornos escolares en el capítulo anterior, se plantea el análisis de las habilidades relacionadas con la motivación intrínseca a edades tempranas. Esto no es posible en un entorno escolar, con lo que se ha desplegado BAC en las tiendas virtuales para que niños de todo el mundo puedan descargarse el juego y jugar de forma voluntaria.

El **objetivo general** de la tesis asociado a esta experiencia es el **OP4**: *Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC*; en concreto el **OP4a**: *Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género*; el **OP4b**: *Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género*; y el **OP4c**: *Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género*.

Este objetivo está basado en la **hipótesis general** de la tesis **HE5**: los entornos de aprendizaje basados en juegos voluntarios pueden aportar información sobre disposiciones y predisposiciones para el aprendizaje del PC según las características del individuo en cuanto a edad y género.

Aprender a programar a una edad temprana es un vehículo para el desarrollo del PC. Los entornos basados en juegos o GBL se utilizan a menudo para desarrollar estas habilidades, pero carecen de la suficiente voluntariedad para evaluar aspectos relacionados con la motivación intrínseca, como los

intereses, las habilidades, la persistencia en la resolución de un problema y el comportamiento en respuesta a las recompensas. Estos aspectos afectan directamente al rendimiento y al desempeño académico, por lo que es necesario analizar las posibles diferencias de edad y género para ajustar los planes de estudio e incluir el PC. Por ejemplo, diversos estudios demuestran que las chicas están menos interesadas en las áreas STEM que los chicos durante la etapa de Educación Secundaria, y que esta situación podría revertirse durante las primeras etapas educativas [194].

Mediante el despliegue de BAC de forma que se pueda jugar voluntariamente, se pueden analizar datos de motivación intrínseca en edades tempranas. De esta forma, se recogieron datos de 4124 usuarios que jugaron durante 15 meses 28187 partidas.

En este capítulo, se detalla la experiencia realizada, primero realizando una contextualización teórica y enumerando los objetivos de la experiencia (preguntas de investigación), para después analizar los datos recogidos en profundidad.

7.1.1. Analíticas de aprendizaje en juegos

Como se ha visto en anteriormente, el aprendizaje basado en GBL ha sido ampliamente adoptado en la educación para la enseñanza de varios temas en muchas áreas diferentes, como las matemáticas [195], la minería de datos [196] y el idioma inglés [197]. Además, es una de las principales estrategias más utilizadas y reportadas en la literatura para el aprendizaje de la programación y el desarrollo del PC en Educación Primaria [4]. Se trata de un marco de resolución de problemas en el que se crea un reto para que los alumnos busquen soluciones utilizando mecánicas de juego con una sensación de logro, a la vez que se potencia la adquisición de conocimientos y habilidades [9]-[11]. El aprendizaje significativo se produce cuando el diseño de los juegos se fundamenta en teorías de aprendizaje relevantes y cuando se incluyen elementos de diseño de juegos específicos, como la interacción, la retroalimentación inmediata, los objetivos claros o el fracaso de bajo riesgo [11], [198]. Los principales fundamentos teóricos utilizados por GBL son el constructivismo, donde el conocimiento es creado activamente por el propio estudiante en un proceso de aprendizaje interactivo [12]; y la teoría del flujo, donde la actividad del jugador está impulsada por el placer más que por recompensas externas y donde se crea un equilibrio entre la dificultad del reto del juego y la habilidad del jugador [11], [121].

Teniendo en cuenta la teoría del flujo, la evaluación de los estudiantes en un entorno GBL no debería afectar la experiencia de juego, por lo que la recogida de datos debería ser transparente para los jugadores. Esto se logra mediante

evaluación sigilosa integrada en los entornos de juego, que se basa en los modelos de diseño centrado en la evidencia (ECD) [199] y permite almacenar las interacciones de bajo nivel de los jugadores y utilizar este flujo de evidencia del juego para evaluar las habilidades y conocimientos de los estudiantes. Estos datos se recogen de forma discreta para el jugador, sin interrumpir su participación, y en tiempo real, lo que permite una retroalimentación inmediata si es necesario. Está comprobado que la evaluación de las competencias de los jugadores es precisa gracias a esta evaluación sigilosa, y que puede utilizarse para diversos fines [200].

Los entornos GBL que incorporan la evaluación sigilosa pueden ser potentes herramientas de evaluación siempre que los datos recogidos se analicen adecuadamente. Para ello, a partir de la investigación se sabe que las soluciones basadas en datos que aprovechan las analíticas de aprendizaje basadas en juegos (GLA) son esenciales para guiar el desarrollo de entornos GBL [20]. La disciplina GLA es una extensión de las analíticas de aprendizaje en entornos educativos [201], combinada con analíticas de juegos [149] que se centra en el análisis de los datos de interacción del usuario en el juego. Hay muchas integraciones exitosas de GLA para la evaluación en juegos educativos [16], [119], [150]. Por ejemplo, CMX que es un MMORPG educativo dirigido a la educación secundaria para la enseñanza de la programación [151], [152]. Del mismo modo, existen varios marcos y normas para el correcto análisis de los datos recogidos por los juegos [16], [17], [119], [152]–[154].

GLA podría utilizarse para analizar los datos 1) en tiempo real para proporcionar una retroalimentación urgente, 2) inmediatamente después de la sesión de juego, y 3) después de que se hayan recogido suficientes datos durante varias partidas para que el análisis pueda arrojar conocimientos adicionales, a través de procesos de minería de datos, que pueden aplicarse para extraer patrones de uso, motivación o intereses. Por ejemplo, podrían extraerse grupos de jugadores que muestren diferentes comportamientos o características, como la edad o el género [202]. Además, los datos de las interacciones con juego contienen información valiosa sobre cómo los jugadores interactuaron con el juego, su progreso, las habilidades, la persistencia, los problemas encontrados o la comprensión de los conceptos abordados [16], [17]. Además, los enfoques de visualización comparativa de GLA facilitan la comprensión de las diferencias entre individuos y submuestras [202].

7.1.2. Motivación intrínseca en entornos GBL

Los entornos GBL tienen muchas ventajas, ya que la motivación y el compromiso del alumno son mayores que en los entornos escolares tradicionales y, por tanto, se consigue una comprensión más profunda y auténtica de los

principios subyacentes que se enseñan [17]. Zimmerman (2004) define los juegos como *"una actividad interactiva voluntaria, en la que uno o más jugadores siguen reglas que limitan su comportamiento, representando un conflicto artificial que termina en un resultado cuantificable"* [203]. Este juego autónomo responde a una motivación intrínseca (MI) hacia el rendimiento o el logro bajo la perspectiva de la expectativa de control y eficacia [204], [205]. Además, la MI correlaciona positivamente con el aprendizaje, el motivo de logro y la auto competencia [206], así como con la reducción de la ansiedad [207]. La teoría de la autodeterminación [208] explica la MI como una tendencia innata a realizar conductas que despiertan interés, más que las obligatorias. Es bien sabido que las personas con una alta autoeficacia están más motivadas intrínsecamente por las tareas, se fijan metas de aprendizaje más altas, persisten más y experimentan menos reacciones adversas [209].

Los juegos serios se han introducido en contextos educativos como entornos GBL. Sin embargo, se circunscriben a escenarios académicos controlados o a circunstancias externas al entretenimiento del jugador, ya que se diseñan como un soporte en el que introducir contenidos educativos, por lo que contrastan con las características distintivas de un juego: la diversión y la voluntariedad, y su carácter motivacional tradicional. Por otro lado, los videojuegos pueden definirse como una actividad lúdica, delimitada por reglas, ejercida voluntariamente a través de un hardware específico [210].

Los videojuegos utilizan el fracaso como herramienta motivacional y sólo ofrecen oportunidades intermitentes de éxito que van acompañadas de recompensas [211], [212], ya sean informativas o de control [213]. Estas experiencias de fracaso pueden conducir a sentimientos de frustración o, por el contrario, a una respuesta de perseverancia e interés por superar los retos propuestos en los que el jugador se muestra persistente [214]. La persistencia se define como la aspiración a completar un reto y superar las metas, y se ha demostrado que es una habilidad importante para la resolución de problemas y el aprendizaje en general [215]. La autoeficacia, la satisfacción y el MI son variables que conducen a la persistencia de los estudiantes [216]. Además, la micropersistencia es la aspiración para completar una tarea individual con éxito, por ejemplo, en entornos GBL, y puede mejorar el desarrollo del PC y los procesos de aprendizaje implicados, de esta forma, ha sido reconocida como un enfoque prometedor para mejorar los resultados de este aprendizaje [215], [217], [218]. Así, el comportamiento en la respuesta de un determinado reto en función de las características de los jugadores podría proporcionar pistas sobre sus capacidades e intereses, e incluso predecir su rendimiento escolar [212]. Como sugieren Ventura y Matthew (2013), se necesitan más investigaciones que tengan en cuenta estas complejidades para avanzar significativamente en este campo [212].

Sin embargo, en la gran mayoría de las investigaciones realizadas hasta el momento con entornos GBL, los jugadores son evaluados en sesiones supervisadas en los colegios [219], donde la interacción con el entorno de aprendizaje no es voluntaria ni autónoma, ya que es difícil enseñar y a la vez crear un clima de autonomía cuando hay que alcanzar unos objetivos curriculares marcados [220], por lo que puede haber una falta de MI y los datos obtenidos pueden no reflejar los intereses de los alumnos y su persistencia y micropersistencia en el aprendizaje de determinados conceptos, o su comportamiento en respuesta a las recompensas. Además, puede haber interacciones entre los propios alumnos que influyan en los resultados [117]. Además, el tamaño de las muestras utilizadas en estas sesiones supervisadas es en general bastante bajo, Alonso-Fernández *et al.* (2019) afirman que, a partir de una revisión sistemática sobre juegos serios con LA o GLA, solo el 8% de los estudios reportan muestras mayores a 1000 participantes [219]. Las muestras pequeñas podrían restringir la significación y la generalización de los resultados [118], así como la aplicación de los complejos algoritmos de GLA, que requieren grandes cantidades de puntos de datos para ser aplicados adecuadamente [219] y visualizados [202].

7.1.3. Objetivos específicos y preguntas de investigación

En base a la fundamentación anterior, consideramos que la mayor parte de las investigaciones recientes sobre entornos GBL carecen de la suficiente voluntariedad para poder evaluar aspectos relacionados con la MI, por lo que proponemos el despliegue de BAC como entorno GBL en plataformas anónimas a través de Internet sin supervisión ni límites de tiempo o intentos, dirigido a jugadores de 3 a 12 años. Se concibe como una actividad voluntaria y autónoma para aprender a programar, en la que la motivación es intrínseca, y está dirigida a una muestra mucho mayor que en un entorno escolar habitual. Se pretende evaluar aspectos como los intereses, las habilidades, la micropersistencia en la resolución de un problema y el comportamiento en respuesta a las recompensas; basándose en la evaluación sigilosa y utilizando algoritmos GLA, y visualizaciones comparativas [202]. Este análisis de datos podría proporcionar conclusiones sobre la motivación para aprender a programar entre los jugadores jóvenes de diferentes edades y géneros, así como conocimientos sobre los efectos de las recompensas según las características de los jugadores. Además, identificar las características, como la edad y el género, de los jugadores con habilidades y/o intereses en determinados conceptos de programación es esencial para poder ajustar los programas curriculares de PC, ya que cuanto mayor sea el interés por la programación, más probable es que sea más fácil aprender, tener una mayor autoeficacia creativa y, por lo tanto, mostrar una mayor potencial o empoderamiento para el aprendizaje de la programación [221].

Teniendo en cuenta las investigaciones ya realizadas, nos interesa la siguiente pregunta principal de investigación:

Basándonos en la MI, ¿qué hallazgos en términos de intereses tempranos, habilidades, micropersistencia y comportamiento en respuesta a las recompensas en el aprendizaje de la programación se pueden extraer de la aplicación de GLA a un videojuego voluntario?

Para responder a esta pregunta principal de la investigación, se han formulado las siguientes subpreguntas u objetivos específicos de la investigación, **según la edad y género** de los jugadores:

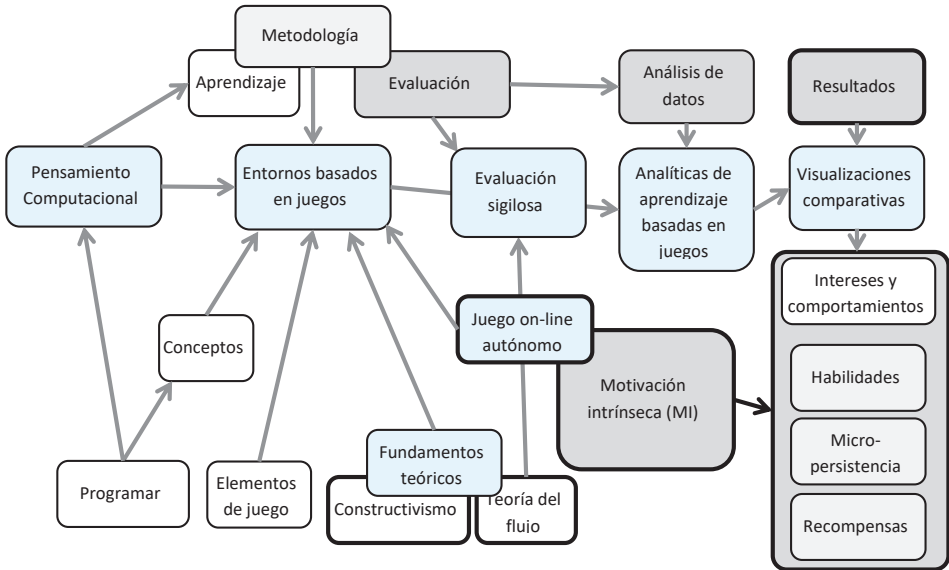
OE1: ¿Existen diferencias en el interés por aprender a programar y por resolver problemas computacionales específicos?

OE2: ¿Existen diferencias en cuanto a las habilidades y los logros a lo largo de los intentos de juego?

OE3: ¿Existen diferentes comportamientos de juego en cuanto a la micropersistencia y las recompensas?

La investigación se ha llevado a cabo bajo la siguiente hipótesis específicas: (1) la motivación para jugar a un juego autónomo en línea es intrínseca; (2) los datos pueden recogerse a través de una evaluación sigilosa en un entorno GBL en línea; y (3) los intereses, las habilidades, la micropersistencia y el comportamiento en respuesta a las recompensas pueden explorarse aplicando técnicas de GLA y visualizaciones comparativas. La Figura 51 muestra la conexión entre los conceptos discutidos en esta sección y cómo, a través de un entorno de juego online voluntario y autónomo, se pueden analizar los intereses y comportamientos de los jugadores, basados en su motivación intrínseca.

Figura 51. Mapa conceptual: conexión entre los conceptos tratados en esta sección.



7.2. Metodología

A continuación, se muestra la metodología que se ha seguido para la recogida y análisis de los datos.

7.2.1. Adaptación del entorno BAC para su despliegue masivo

En esta experiencia se utiliza BAC [51], desarrollado y validado en el marco de esta tesis como un entorno GBL que es tanto un instrumento de aprendizaje como una herramienta de evaluación sigilosa a través de la recogida de datos en tiempo real. Este juego puede utilizarse en las aulas escolares, en dispositivos como las tabletas, tanto de forma individual como en modo colaborativo, para el aprendizaje de la programación, así como para la evaluación posterior de los datos de juego mediante técnicas de GLA. Los niveles del juego y su funcionamiento se encuentran detallados en el capítulo anterior.

Para analizar el micro-comportamiento de los jugadores y con el objetivo de la implantación masiva del juego, se han añadido o actualizado las siguientes funcionalidades a BAC:

- 1) Pantalla de creación/selección de usuarios: cada dispositivo en el que se instala el juego puede tener hasta 6 usuarios diferentes.

- 2) Pantalla inicial para la recogida de datos básicos anónimos de edad y sexo, más apodo opcional.
- 3) Personalización del personaje principal (hormiga azul) a través de una tienda con acceso desde el menú principal, donde los puntos obtenidos se pueden gastar en accesorios y ropa.
- 4) Adaptación de los datos a recoger en el juego y la vinculación a la base de datos remota para el despliegue masivo.

La versión masiva solamente cuenta con el modo individual de juego. El sistema de puntuación para la versión masiva de BAC para cada nivel es el siguiente: 2 puntos por alcanzar un objetivo parcial (puede haber de 0 a 2 objetivos parciales en cada nivel) y 10 puntos por completar el nivel. Estos puntos se pueden canjear por objetos para personalizar al personaje principal del juego en una tienda virtual. Esta funcionalidad se ha implementado para analizar el efecto de la micropersistencia y las recompensas en el comportamiento del jugador en una fase posterior.

7.2.2. Variables de investigación

Para la creación de evaluaciones en entornos GBL es muy recomendable, antes de aplicar GLA, especificar y determinar las trazas de juego que se recogerán [219], [222], y éstas deben ser lo suficientemente potentes como para alimentar el sistema GLA [53]. Por esta razón, las variables de investigación para analizar el comportamiento de los jugadores fueron bien definidas antes de establecer las trazas de juego a recoger:

- **Persistencia:** es una habilidad valiosa a la hora de resolver problemas y puede mejorar la adquisición del PC y los procesos de aprendizaje que conlleva [217], [223]. Puede definirse como la voluntad de completar un proceso de aprendizaje y alcanzar los objetivos de éste. La micropersistencia es la aspiración para completar con éxito una tarea individual, por ejemplo, un nivel, y se centra en el proceso de aprendizaje de cada uno de los conceptos y no en el proceso de aprendizaje en su conjunto [224]. Cuantificaremos esta variable con el número de intentos realizados por el jugador.
- **Logro o rendimiento:** Definimos el logro como el porcentaje del número de partidas ganadas sobre las jugadas, tanto en total como en un nivel determinado.
- **Habilidad:** Grover et. al (2016) encontraron que el tiempo que se tarda en resolver un nivel predice el nivel de habilidad con una precisión de

~63% [225]. Cuantificaremos esta variable como el tiempo necesario para completar un nivel, siendo mayor sea la habilidad cuanto menor el tiempo necesario.

- **Dominio:** consideraremos el dominio de un nivel (concepto computacional) a una edad determinada, si observamos un alto rendimiento y habilidad, y/o un alto rendimiento y pocos intentos necesarios para completar un nivel, en comparación con la tendencia observada a edades más tempranas.
- **Progresión:** consideraremos una progresión positiva cuando los niveles de dominio aumenten a lo largo de los intentos de juego.
- **Recompensas:** cuantificaremos el grado de recompensas en función del número de puntos canjeados por objetos en la tienda. Analizaremos la relación de esta variable con la de persistencia.

Por último, sacaremos conclusiones sobre los intereses y la motivación, concretamente la MI, de los jugadores a partir de las variables anteriores y de las observaciones generales.

Los rastros recogidos de BAC y su relación con las variables de la investigación se muestran en la Tabla 41. Además, se asocian identificadores únicos a cada usuario, a cada sesión de juego y a cada nivel jugado. También se recoge información sobre la edad, el sexo, el tipo de dispositivo, la fecha y la hora de cada partida, el nivel y el reto propuesto (ya que los retos se establecen de forma aleatoria y son diferentes para cada partida).

Tabla 41. Relación entre las trazas recogidas y las variables de investigación.

Área	Traza	Variable primaria relacionada	Variable secundaria relacionada
Sesión de juego	Tiempo inicial de sesión	Intereses	Persistencia
	Puntuación	Motivación	Persistencia
	Número de partidas ganadas	Logro	Dominio
	Número de partidas jugadas	Persistencia	Progresión
	Puntos canjeados en la tienda	Recompensas	Persistencia

Área	Traza	Variable primaria relacionada	Variable secundaria relacionada
Nivel jugado	Tiempo inicial de partida	Habilidades	Dominio
	Tiempo final de partida	Habilidades	Dominio
	Código propuesto	Dominio	
	Código ejecutado	Dominio	
	Código óptimo	Dominio	
	Puntuación	Persistencia	Recompensas
	Indicador de partida ganada	Progresión	Logro

Los datos se almacenaron en una base de datos relacional SQL remota que se conecta al juego a través de una API .NET para garantizar la seguridad de los datos. Los datos se analizaron utilizando Python (con la biblioteca scikit-learn), Microsoft Excel y el software SPSS. Existen diferentes técnicas de visualización que se han aplicado a los datos de análisis de juegos [219]. En este caso se ha utilizado la biblioteca matplotlib de Python, Microsoft Excel y el software Gephi para diversas visualizaciones, como las métricas de rendimiento, las trayectorias del juego o las curvas de aprendizaje.

BAC ha sido desarrollado para los sistemas operativos Android e iOS y ha sido desplegado en las plataformas virtuales “Google Play” y “Apple Store” [226], estando disponible para su descarga gratuita en todo el mundo. No se han realizado campañas publicitarias dirigidas a ningún grupo o población específica para evitar sesgos previos.

7.2.3. Participantes y procedimiento

Los datos recogidos abarcan un periodo de 15 meses durante los cuales cualquier usuario era libre de descargar el juego y jugar tantas veces como quisiera. Desde Google Play se han realizado 741 descargas, procedentes de 53 países diferentes, de ellos, España fue el país con más descargas (24.5% del total), seguido de Estados Unidos (5.98%). Desde la Apple Store se han realizado 2130 descargas en 42 países diferentes, siendo Finlandia el país con más descargas (51.64% del total), seguido de México (23.90%). Se registraron 4124 usuarios en total (Google Play = 2294; Apple Store = 1829) y se jugaron 28187 partidas, teniendo en cuenta que cada dispositivo puede registrar hasta 6 usuarios, y cada usuario puede jugar cualquier número de veces.

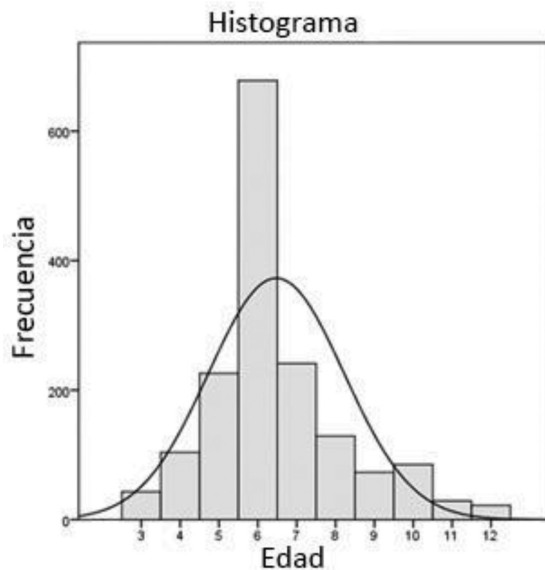
7.3. Resultados y discusión

Se analizarán los datos obteniendo primero una visión general y luego profundizando en cada uno de los aspectos del estudio y en las edades clave en las que se han observado cambios significativos en cuanto a intereses y comportamiento de juego.

7.3.1. General

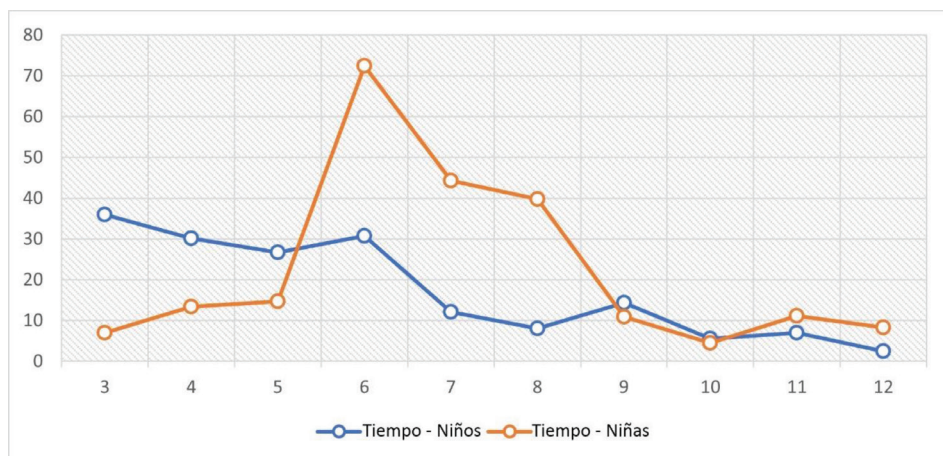
Tras el preproceso de filtrado de datos, en el que se han excluido valores atípicos (*outliers*) numéricamente distantes del resto de los datos (por ejemplo, demasiado tiempo de juego, que puede interpretarse con dejarse el dispositivo encendido por error), se han analizado 1875 usuarios y 25118 partidas. El tiempo total de juego es de 49563,85 minutos (por usuario: Media = 26,43; Mediana = 3,87; SD = 199,81). Como nos interesa el análisis del PC en edades tempranas, nos centraremos en el grupo de edad de 3 a 12 años en el que se han jugado un total de 23651 partidas. La Figura 52 muestra que la distribución de los jugadores por edad de 3 a 12 años se ajusta a una curva normal independientemente del género (Tamaño de la muestra (n) = 1630, Media = 6,47; SD = 1,744), aunque hay el triple de usuarios masculinos (70% del total). De la distribución de las descargas por género, se deduce que los niños están a priori más interesados que las niñas en aprender a programar, en línea con investigaciones anteriores [227]. Sin embargo, hay más usuarios de 6 años de lo esperado, tanto niños como niñas, quizás en parte porque es la edad que se puede introducir en la aplicación de forma más inmediata.

Figura 52. Distribución por edad.



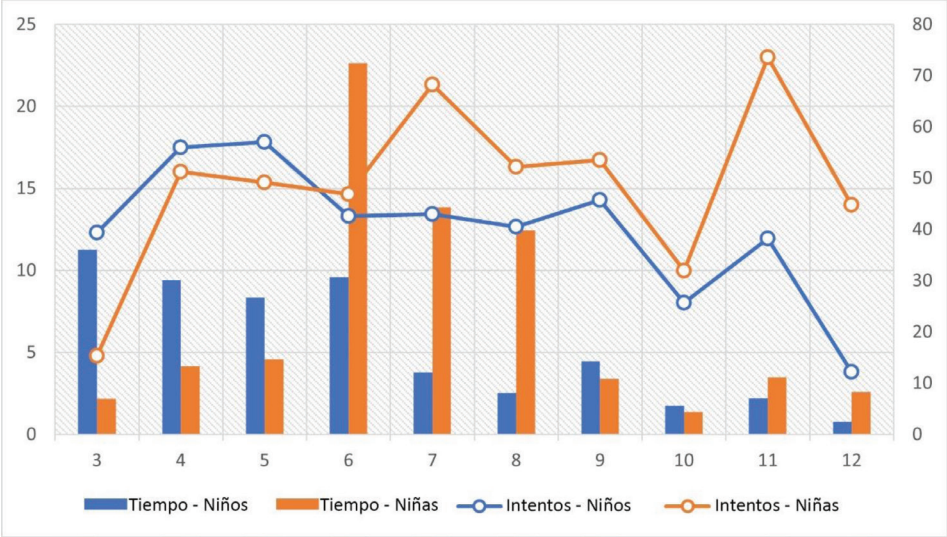
En un primer análisis, podemos ver que la media de tiempo total de juego sumando todas las partidas es mayor para las niñas (22,63 minutos) que para los usuarios (17,32 minutos), pero si dividimos la muestra por edades, podemos ver que los niños juegan más tiempo hasta los 5 años y, a partir de ahí, hay un cambio de tendencia muy marcado y, a los 6 años, las niñas juegan una media de 72,44 minutos en total (véase la Figura 53). Consideramos este hallazgo como una indicación para prestar especial atención en las edades de 4 a 6 años, especialmente en las niñas.

Figura 53. Tiempo medio en minutos totales de juego por edad y sexo.



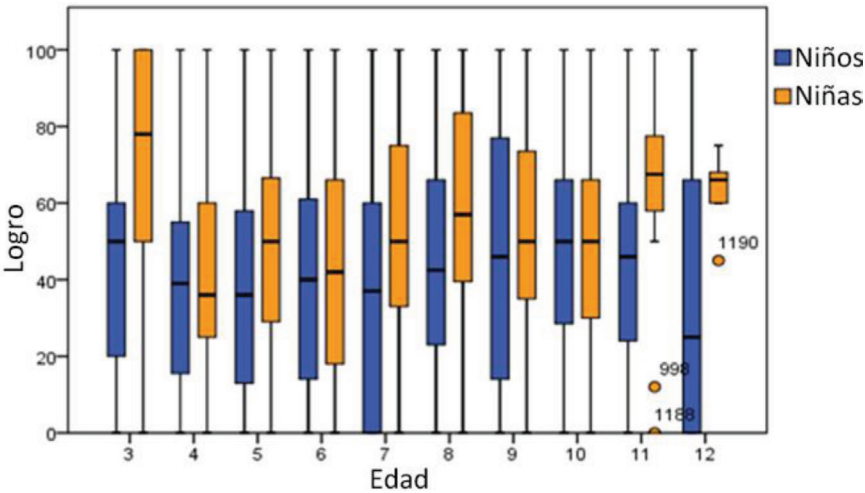
Cuando comparamos el número medio de intentos o partidas (persistencia) y el tiempo medio dedicado a cada intento (relacionado con la habilidad), por edad y género (véase la Figura 54) podemos observar que las niñas jugaron más partidas por usuario en total (media de intentos por parte de los niños = 13,76; media de intentos por parte de las niñas = 15,70). Aunque hasta los 5 años la persistencia es mayor para los niños, a partir de esa edad la tendencia se invierte, siendo las niñas las que juegan significativamente más partidas que los niños (de 6 a 8 años: prueba t de Student $p = .011 < .05$). Del mismo modo, en el rango de edad de 3 a 5 años, los niños pasaron más tiempo jugando por usuario, pero en el rango de edad de 6 a 8 años, el tiempo dedicado por partida es mucho mayor para las chicas (de 6 a 8 años: prueba t de Student $p = .002 < .01$). Esto proporciona el primer indicio de persistencia, ya que, aunque los niños parecen estar más interesados en descargar el juego, una vez instalado, las niñas dedican más tiempo y realizan un mayor número de intentos, especialmente a partir de los 6 años. Además, parece que la persistencia en el juego es mayor en las niñas a partir de los 6 años, mientras que en los niños disminuye con la edad.

Figura 54. Eje izquierdo: número medio de intentos por edad y sexo. Eje de la derecha: tiempo medio en minutos dedicado a cada intento.



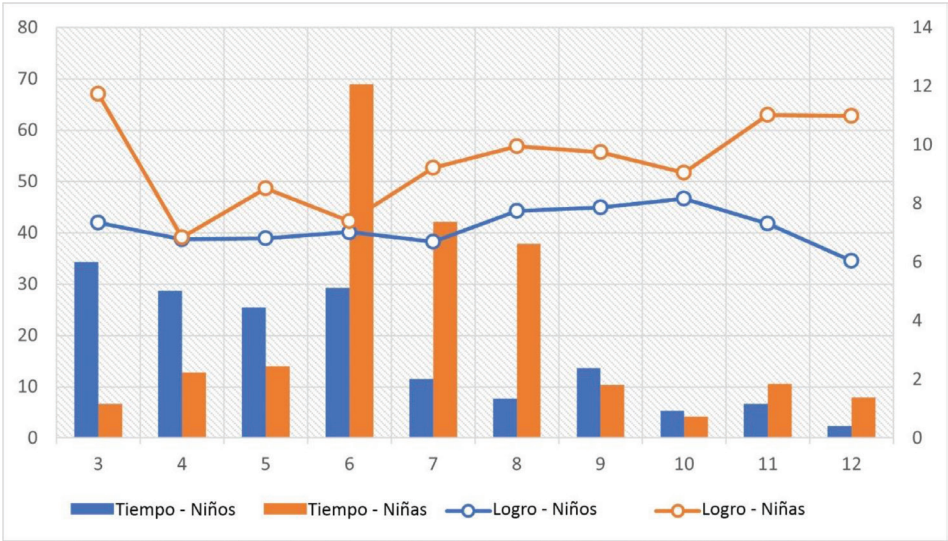
En cuanto al logro, medido como porcentaje de partidas ganadas sobre las jugadas (Media = 42,89; SD = 31,27), se observa en la Figura 55 que siempre es mayor para las niñas, independientemente de la edad o del tiempo medio empleado. En general, el logro es significativamente mayor en las niñas (prueba *t* de Student $p = .000 < .01$), especialmente a la edad de 3 años.

Figura 55. Rendimiento o logro: porcentaje del número de partidas ganadas sobre las jugadas.



Por último, para evaluar el dominio global del juego, se compararon tanto el logro como la habilidad por edad y género. Se puede observar en la Figura 56, que de los 3 a los 5 años, las niñas tienen un mayor logro con menos tiempo invertido por intento, y menos intentos necesarios (véase la Figura 54), por lo que deducimos que podrían tener un claro mayor dominio que los niños. Sin embargo, de los 6 a los 8 años, aunque su logro es mejor, las niñas pasan mucho más tiempo jugando, por lo que no podemos deducir un mayor dominio y consideramos que este comportamiento debe ser explorado con más detalle, lo que haremos más adelante en este capítulo. De 9 a 12 años, el tiempo dedicado es similar en niñas y niños, con un mayor logro en las niñas.

Figura 56. Eje izquierdo: habilidad - número medio de intentos por edad y sexo (normalizado). Eje derecho: logro - porcentaje medio del número de partidas ganadas sobre las jugadas por edad y género (normalizado).



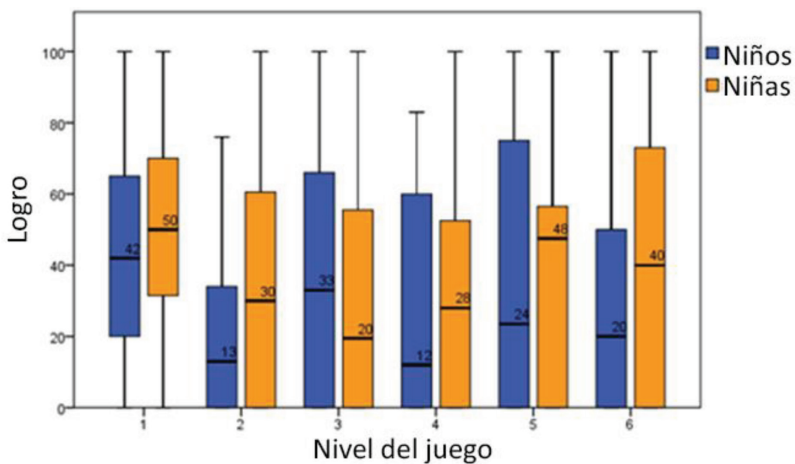
7.3.2. Conceptos computacionales

Tras estos resultados preliminares, se han analizado las variables de investigación por niveles del juego (correspondientes a distintos conceptos computacionales). En primer lugar, el nivel 1 (secuencias simples) es el de mayor rendimiento o logro (Media = 45,53; Mediana = 46,00; SD = 30,34), seguido del nivel 3 (bucles simples), 5 (while simple) y 6 (while complejo). Los niveles con menos logro son el 2 (secuencias largas) y el 4 (bucles anidados). Fue una sorpresa encontrar dificultades en el nivel 2 (Media = 24,20; Mediana = 15,00; SD = 27,45) ya que este nivel trata el concepto de secuencias, que es

el más sencillo que se aborda en el juego. Sin embargo, la particularidad de este nivel y su única diferencia con el nivel 1 es que las secuencias a construir son muy largas. Nuestra hipótesis es que esta dificultad en el nivel 2 está relacionada con la memoria de trabajo, que es crucial para almacenar información. Esta memoria de trabajo se desarrolla a lo largo de los años escolares y tiene importantes consecuencias en la capacidad de aprendizaje de los niños [58]. Por último, el nivel 4 también tiene un bajo rendimiento (Media = 29,44; Mediana = 17,00; SD = 32,67) y se considera el nivel más difícil incluido en el juego, ya que los bucles anidados se consideran un concepto inalcanzable a los 4 años [59], [60].

En cuanto al género, como se muestra en la Figura 57, las niñas tienen un mayor rendimiento que los niños en los niveles más difíciles, especialmente en los niveles 2 y 4, donde los niños tienen una mediana muy baja. El único nivel en el que los niños tienen un mayor rendimiento que las niñas es el nivel 3, de dificultad media.

Figura 57. Logro por nivel y por género.



Para obtener una visión global del dominio en cada concepto, se ha analizado el logro y la persistencia en cada uno de los niveles y por edad (Figura 58 y Figura 59). Se observa que, a los 3 años, el nivel 1 (secuencias simples) se domina con una media en el caso de los niños del 42,8% de logro y 14,9 intentos medios por usuario, y, en las niñas, un 72,2% de éxito y 5 intentos por usuario. A partir de este momento, la persistencia y el logro son similares en todas las edades, independientemente del género, en el nivel 1. Además, a los 3 años, los niños se interesan más por el nivel 3, con una media de 23 intentos y un 37% de logro.

Figura 58. Persistencia media o número de intentos (eje vertical), por nivel y edad (eje horizontal).

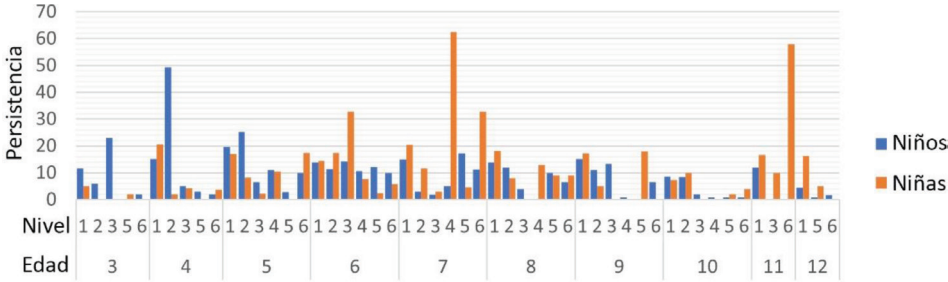
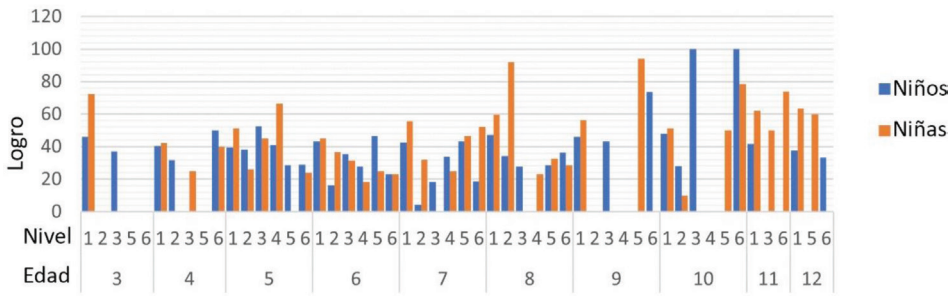


Figura 59. Rendimiento (logro) medio (eje vertical) por nivel y edad (eje horizontal).



A la edad de 4 años, los usuarios masculinos muestran una alta persistencia en el nivel 2 (49,3 intentos de media por usuario) a pesar del bajo rendimiento (32%), en comparación con los 2 intentos por usuario de las niñas y el 0% de rendimiento. Cabe destacar que, a esta edad, en el nivel 3, las chicas obtienen un logro del 25% frente al 0% de los chicos. En el nivel 6 (concepto while complejo) se obtiene un logro elevado (50%) con pocos intentos.

A los 5 años, se observa el mismo comportamiento en el nivel 2; el nivel 3 tiene un logro elevado con menos intentos tanto por parte de los chicos como de las chicas; y el nivel 4 empieza a despertar interés en ambos géneros por igual (11 juegos de media por usuario), aunque el logro es mayor para las chicas (66,5% frente al 41% de los chicos). En los niveles 5 y 6 no hay diferencias notables en cuanto al género.

A los 6 años, los niños no están tan interesados en el nivel 2, pero las niñas sí (67% de logro). El nivel 3 despierta más interés en las niñas (32,8 intentos,

frente a 14,2 en los niños) con un rendimiento similar (en torno al 35%). El nivel 4 está igualado por sexos (10,6 intentos con un 27,6% de consecución por parte de los niños, frente a 7,7 intentos y un 18,3% de consecución por parte de las niñas). Los niveles 5 y 6 despiertan más interés en los niños a esta edad.

A los 7 años, el nivel 3 deja de ser interesante para ambos sexos y el nivel 4 pasa a ser muy interesante para las niñas, con una media de 62,5 intentos por usuario, frente a los 5 intentos de los niños. Sin embargo, el rendimiento es similar (en torno al 30%).

Por último, a los 8 años, el nivel 2 es totalmente dominado por las niñas, con un 92% de éxito, frente al 34,3% de los niños. Sólo las niñas juegan al nivel 4 (13 juegos por usuario de media) con un logro del 23%, y los chicos pierden el interés por él. A partir de este punto, no hay diferencias significativas en el resto de los niveles.

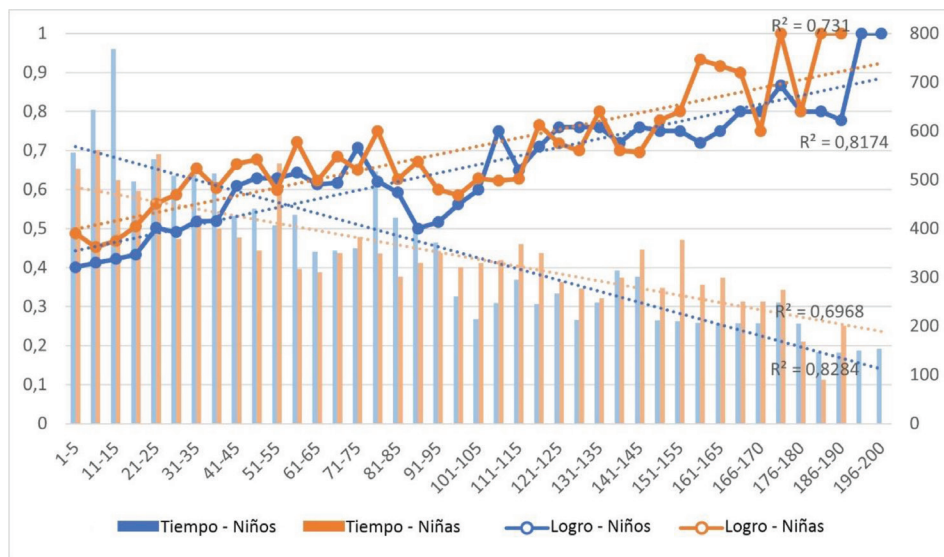
En conclusión, las secuencias simples parecen dominarse a partir de los 3 años, pero el nivel 2 (secuencias largas) no parece dominarse hasta los 5 años. En el nivel 3, en cambio, el rendimiento aumenta alrededor de los 3 años en el caso de los niños y de los 4 años en el caso de las niñas. En el nivel 4 (bucles anidados) parece que el rendimiento es mayor a partir de los 5 años en el caso de las niñas y de los 6 años en el caso de los niños, con un mejor rendimiento por parte de las niñas. Por otra parte, en los niveles 5 y 6 (condicionales), el rendimiento aumenta a los 4 años, independientemente del sexo. Estos resultados están en consonancia con investigaciones anteriores [8], [61], en las que los niños obtuvieron mejores resultados que las niñas en los niveles intermedios, mientras que esta tendencia se invirtió en los niveles más difíciles.

A los 4 años, las niñas juegan en promedio muchas partidas en el nivel 2 (secuencias largas) a pesar de un bajo rendimiento, lo que sugiere un comportamiento de automejora en respuesta a un reto. Un comportamiento similar se observa en las niñas a los 6 años en el nivel 3, y a los 7 años en el nivel 4, que ofrece el reto más difícil.

7.3.3. Progresión

Hasta ahora hemos obtenido una imagen general, pero es importante observar la progresión de los jugadores a lo largo de los intentos de juego. En la Figura 60 se muestran los logros y el tiempo por partida a lo largo de los intentos y por género.

Figura 60. Eje izquierdo: logro medio, representado gráficamente por líneas. Eje derecho: habilidad media por intento por género, en segundos, representado gráficamente por barras. Eje horizontal: intentos progresivos agrupados de 5 en 5.



Tanto las niñas como los niños mejoran a lo largo de los intentos, pero en los niños la mejora es más significativa (progresión positiva), con correlaciones más altas en comparación con las chicas, tanto en la mejora del tiempo (habilidad) (chicos, coeficiente de determinación (R^2) = 0,83; chicas R^2 = 0,70) como en el logro (chicos R^2 = 0,82; chicas R^2 = 0,73). Desde la primera vez que juegan (primer intento), las niñas tienen un mayor logro y habilidad (emplean menos tiempo) que los niños en completar el reto. A partir del intento número 50, el logro es similar.

Teniendo en cuenta los resultados obtenidos hasta el momento, nos centraremos en el logro en los diferentes niveles (Figuras 61 a 63), durante los 10 primeros intentos, y prestaremos especial atención al tramo entre los 4 y los 6 años de edad de los jugadores, ya que es el tramo en el que se ha observado previamente el mayor cambio en el rendimiento.

Figura 61. Logros por intento (10 primeros intentos) y nivel, para **4 años**.

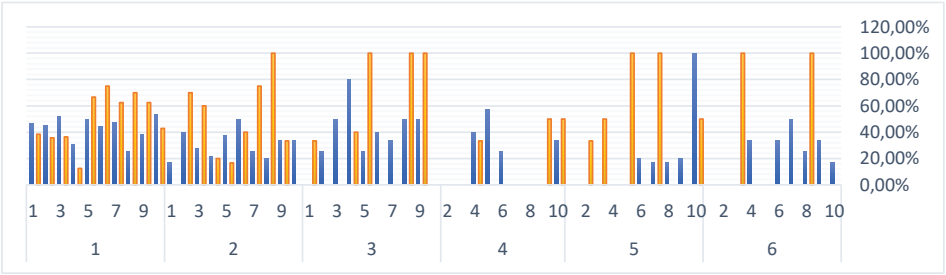


Figura 62. Logros por intento (10 primeros intentos) y nivel, para **5 años**.

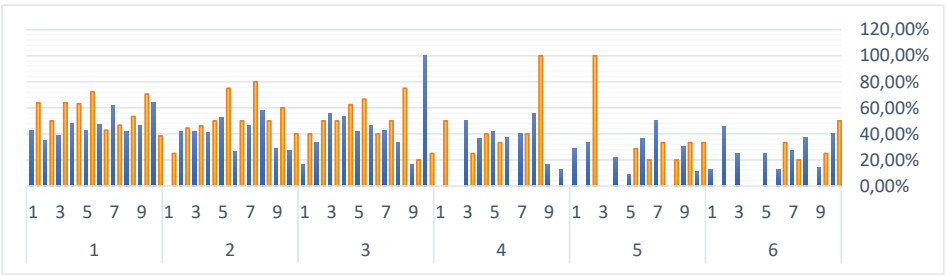
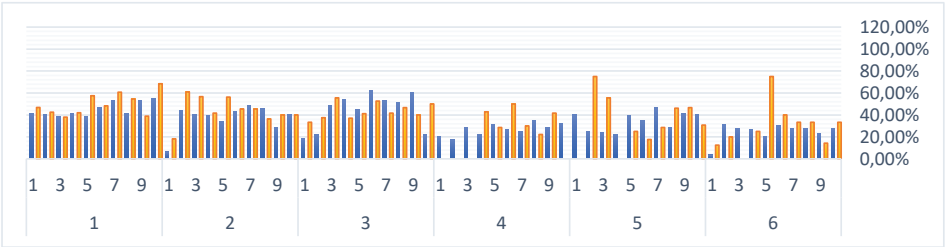


Figura 63. Logros por intento (10 primeros intentos) y nivel, para **6 años**.



Se observa que a la edad de 4 años (véase la Figura 61), las niñas, a medida que juegan, consiguen el 100% de éxito en todos los niveles, excepto en los niveles 1 y 4. En el nivel 1, el 100% de los logros se alcanza a los 3 años, y es posible que este nivel no se vuelva a alcanzar a los 4 años porque es aburrido

para los niños de esta edad, por lo que puede haber una falta de motivación para superar el reto. A la edad de 4 años y en el nivel 4, tal vez debido a la falta de interés en jugar porque se percibe como difícil en primera instancia, sólo hay un máximo de 57,14% de logro para los niños en el intento 5, y un 50% de logro en el intento 9 para las niñas en los primeros 10 intentos. Sin embargo, son capaces de alcanzar el 100% de logro en este nivel en los intentos número 13 (niñas) y número 14 (niños).

A los 5 años (véase la Figura 62), las niñas son capaces de completar el nivel 4 con un 100% de logro, en el intento 8. Cabe destacar que las niñas, a los 5 años, obtienen un 50% de logro en el primer intento, mientras que los niños sólo alcanzan esta misma tasa de éxito en el tercer intento, y un máximo de 55,6% en el intento 8.

Uno de los comportamientos más destacables es que las niñas obtienen un logro significativamente mayor que los niños (a los 4 y 5 años, 10 primeros intentos: $p = .02 < .05$; niños: $n = 641$, media = 0.40; niñas: $n = 284$; Media = 0.47). Las niñas alcanzan el 100% de logro en varias ocasiones y en todos los niveles en los 10 primeros intentos entre los 3 y los 5 años, mientras que este comportamiento no se observa en el caso de los niños. Este comportamiento se sigue observando incluso después de los primeros 20 intentos. A partir de los 6 años (véase la Figura 63), las niñas tardan más tiempo e intentos en resolver los retos.

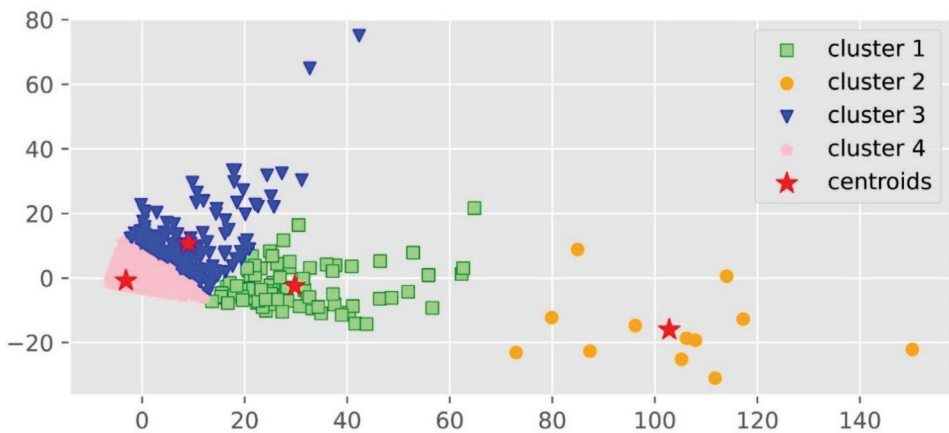
7.3.4. Persistencia

Para seguir analizando el comportamiento de los usuarios en el juego con respecto a la persistencia, se ha dividido la muestra en diferentes clústeres aplicando la técnica PCA (Análisis de Componentes Principales) para reducir la dimensionalidad y utilizando el método del codo para determinar el número de clústeres. Dado que este análisis se realiza después de haber recogido suficientes datos para poder extraer información adicional que muestre los diferentes comportamientos, y no en tiempo real, no es esencial que los tiempos de ejecución sean bajos [20], por lo que esta variable no se tiene en cuenta.

En primer lugar, el objetivo es distinguir entre los jugadores que siguen trayectorias de juego similares y también entre los jugadores que muestran persistencia frente a otros que juegan sólo unas pocas partidas y abandonan el juego. Se realizó una primera división de la muestra en clústeres que se presentan en la Figura 64. El clúster 1 agrupa a los jugadores que juegan

varias partidas, con una media de 58 partidas por usuario, especialmente en los niveles más fáciles. El clúster 2 agrupa a los jugadores con más intentos (mayor persistencia), con una media de 147 partidas por usuario y concentrados mayoritariamente en el nivel 1. El clúster 3 agrupa a los usuarios con una media de intentos de 46 por usuario, pero se distribuyen uniformemente en todos los niveles. Por último, el clúster 4 agrupa a los usuarios con un perfil bajo (media de 7 juegos por usuario), distribuidos entre los 6 niveles.

Figura 64. Primera división en clústeres.

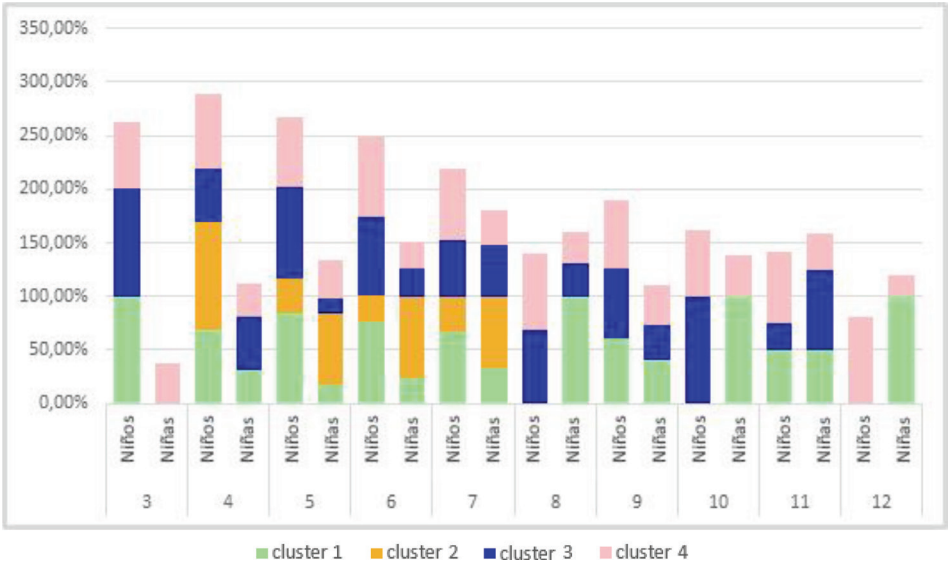


La Tabla 42 muestra que la proporción de niñas respecto a niños en todos los clústeres es similar a la proporción total (70% de chicos en el grupo de usuarios de 3 a 12 años), mientras que en el clúster 2 la proporción no se conserva y hay un 66,5% de niñas, con edades comprendidas entre los 5 y los 7 años. A partir de los 7 años, también hay más niñas que la proporción total en el clúster 1. La Figura 65 muestra, para cada edad, qué porcentaje de cada clúster está compuesto por usuarios de cada sexo. Podemos confirmar, por tanto, que el número de partidas por jugador es mayor en el caso de las niñas con diferencia, en los casos concretos en los que se han alcanzado más de 200 intentos y especialmente en el nivel 1, y que este comportamiento de las jugadoras con una persistencia muy alta se observa sobre todo entre los 5 y los 7 años, y a partir de ahí también es apreciable, pero no es tan pronunciado. En el caso de los chicos, parece haber un cambio a partir de los 8 años, cuando hay una menor proporción de usuarios masculinos en los grupos 1 y 2. Estos usuarios juegan en todos los niveles sin alcanzar un número muy elevado de intentos.

Tabla 42. Distribución por género, normalizado por clúster.

	cluster 1	cluster 2	cluster 3	cluster 4
Male	69,51%	33,33%	68,35%	70,79%
Female	30,49%	66,67%	31,65%	29,21%

Figura 65. Distribución en cada clúster (normalizado), de los 3 a los 12 años de edad.



7.3.5. ¿Persistencia para superar un reto o persistencia para obtener una recompensa?

Un análisis más profundo nos lleva a analizar si la repetición inmediata de un nivel está causada por la intención de superar un reto, en cuyo caso se repetiría el nivel después de haberlo perdido, lo que llamaremos **persistencia-reto**; o la intención de consolidar el conocimiento (satisfacción por ganar) y/o obtener una recompensa (puntos de victoria), en cuyo caso se repetiría el nivel después de haberlo ganado, lo que llamaremos **persistencia-recompensa**. Para analizar estos comportamientos, hemos tomado las primeras 30 partidas, de forma que podamos analizar el comportamiento a lo largo de suficientes repeticiones, considerando si en el caso de ganar o perder una partida se ha producido una repetición inmediatamente posterior.

En primer lugar, se ha creado un vector de características en el que, para cada usuario, se ha determinado un valor acumulado en cada componente del vector que varía en función de si se ha ganado o perdido la partida actual en relación con el resultado de la partida anterior. La Figura 66 muestra la división en clústeres y la Figura 67 muestra la tendencia acumulada en términos de éxito en las partidas de cada clúster (la curva sube en caso de victorias sucesivas y baja en caso de pérdidas sucesivas). En cuanto a la edad, no hay diferencias significativas entre los clústeres, ya que la edad media de todos ellos oscila en torno a los 6,5 años. Los clústeres centrales (1 y 2) son los que tienen un menor número de partidas en total. Se observa un mayor porcentaje de niños en el clúster 1 (véase la Tabla 43) que pierden la partida y vuelven a intentarlo a pesar de haber perdido en numerosas ocasiones, lo que sugiere un comportamiento de persistencia-reto, sin embargo, estos usuarios apenas tienen partidas exitosas. En el clúster 2 la mayoría son chicas, estos usuarios también muestran un comportamiento de persistencia-reto, aunque mucho menor, ya que hay muchas menos partidas seguidas sin éxito y, en un momento dado, la tendencia se invierte y hay partidas ganadoras. Los clústeres 3 y 4 agrupan a los usuarios con muchas partidas. En el clúster 4, en el que hay mayoría de jugadores niños, se observa de nuevo el comportamiento de persistencia-reto, con muchas partidas consecutivas perdidas, y con una inversión de la tendencia alrededor de la partida número 20, donde los usuarios empiezan a ganar muchas partidas. Finalmente, en el clúster 3, con la mayoría de las chicas, se observa un comportamiento de persistencia-recompensa, ya que ganan la mayoría de las partidas desde el principio y a lo largo de todas ellas. Se puede concluir que, por tanto, el comportamiento de repetición a pesar de haber perdido (persistencia-reto) acumula más niños y el comportamiento de persistencia-recompensa acumula más niñas.

Figura 66. Segunda división de clústeres.

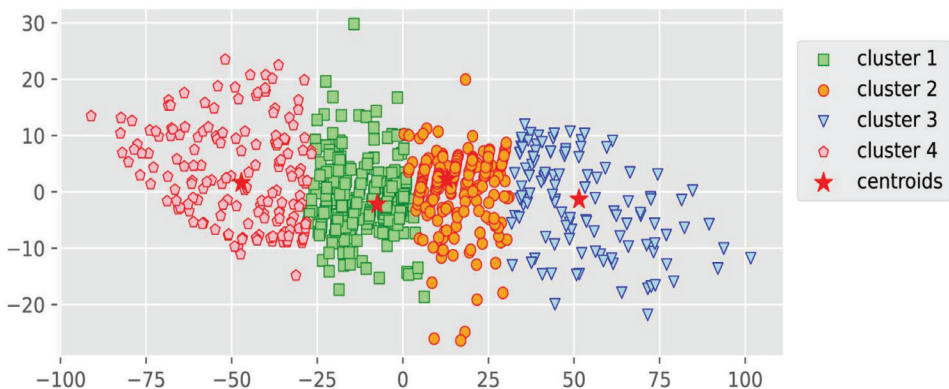


Figura 67. Tendencia a lo largo de los intentos por clúster.

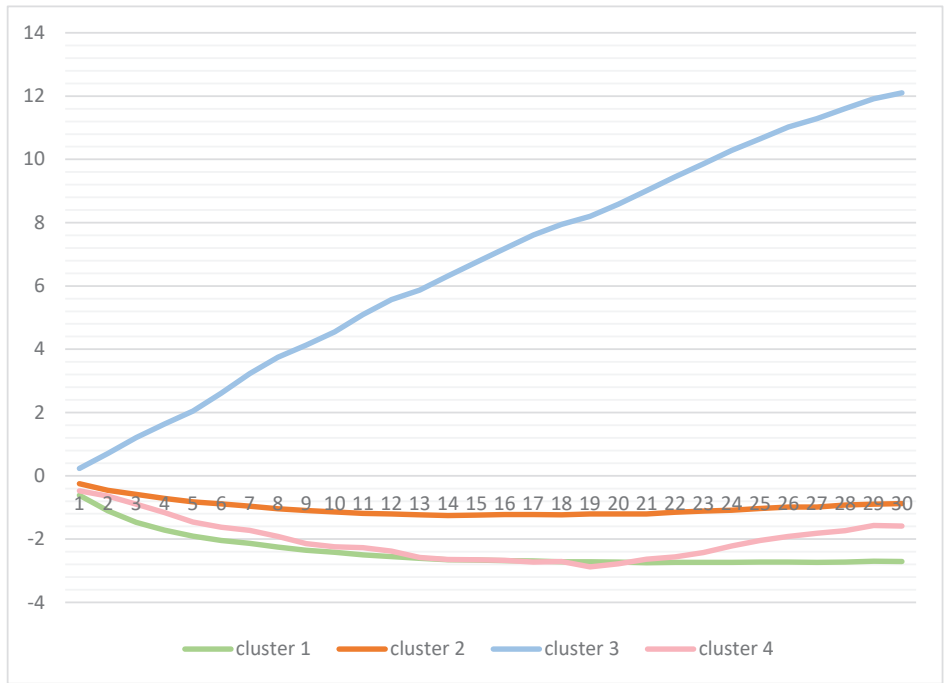


Tabla 43. Distribución de clústeres normalizada por género.

	Niños	Niñas
cluster 1	46,71%	37,42%
cluster 2	34,53%	45,19%
cluster 3	5,96%	8,18%
cluster 4	12,80%	9,20%

Finalmente, para comprobar estos resultados, se ha vuelto a analizar el comportamiento de cada clúster (véase la Figura 68), esta vez teniendo en cuenta las siguientes variables, cuya distribución se muestra en la Figura 69:

- $nMaxL$ = Número máximo de partidas consecutivas perdidas;
- $nMaxW$ = Número máximo de partidas consecutivas ganadas;

- $nSec3L$ = Número de partidas consecutivas perdidas $\Rightarrow 3$;
- $nSec3W$ = Número de partidas consecutivas ganadas $\Rightarrow 3$;
- $nCtLaW$ = Número de cambios de tendencia, de perdidas a ganadas;
- $nCtWaL$ = Número de cambios de tendencia, de ganadas a perdidas;
- nL = Número total de partidas perdidas;
- nW = Número total de partidas ganadas.

El clúster 3 incluye a los usuarios con muy pocas partidas jugadas en general. Los clústeres 1 y 5 son donde, proporcionalmente, hay más niñas. Estos clústeres aglutinan un comportamiento en el que hay una mayor sucesión de partidas ganadas, alcanzando una media de 7,71 partidas sucesivas ganadas en el clúster 5 y una alta tasa de éxito (64,56%). Como se observa en la Tabla 44, en los clústeres 2 y 4 es donde hay proporcionalmente más niños con un comportamiento en el que hay mayor sucesión de partidas ganadas, alcanzando, en el clúster 2, una media de 11,65 partidas sucesivas perdidas y una tasa de éxito muy baja (23,38%). Se confirman, por tanto, los resultados anteriores: la conducta de persistencia-reto es más frecuente en los niños, frente a la conducta de persistencia-recompensa, más frecuente en las niñas.

Figura 68. Tercera división en clústeres.

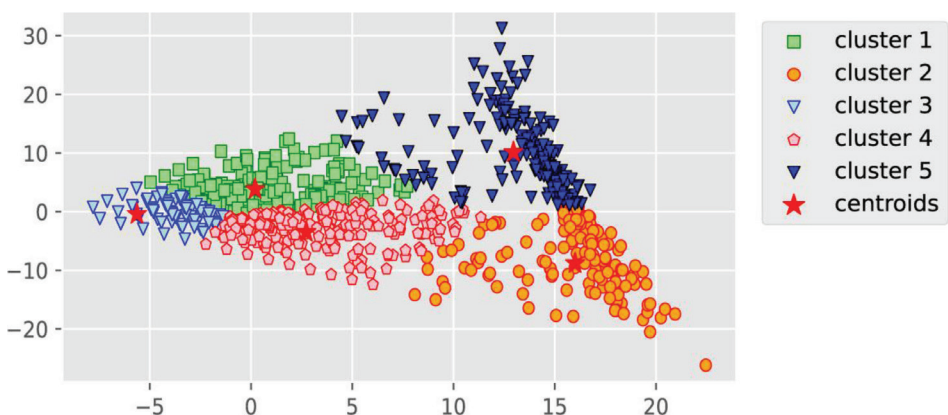


Figura 69. Distribución de las variables (normalizadas) en cada clúster.

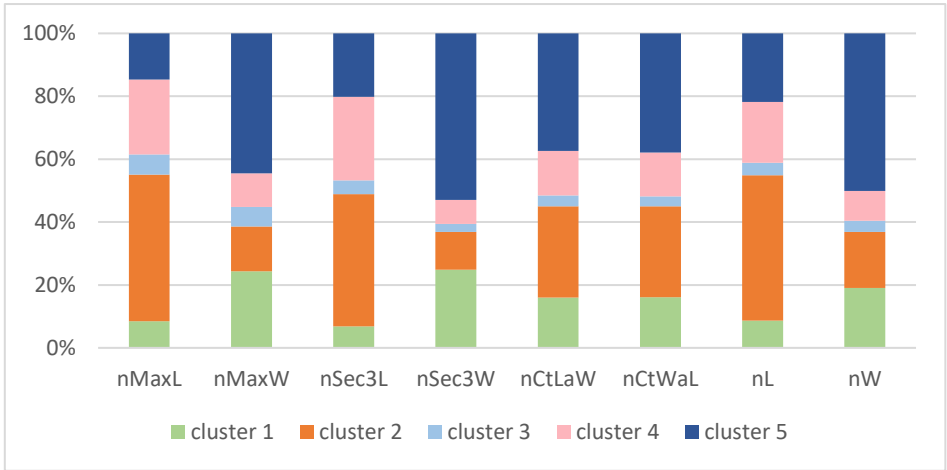


Tabla 44. Distribución de clústeres normalizada por género.

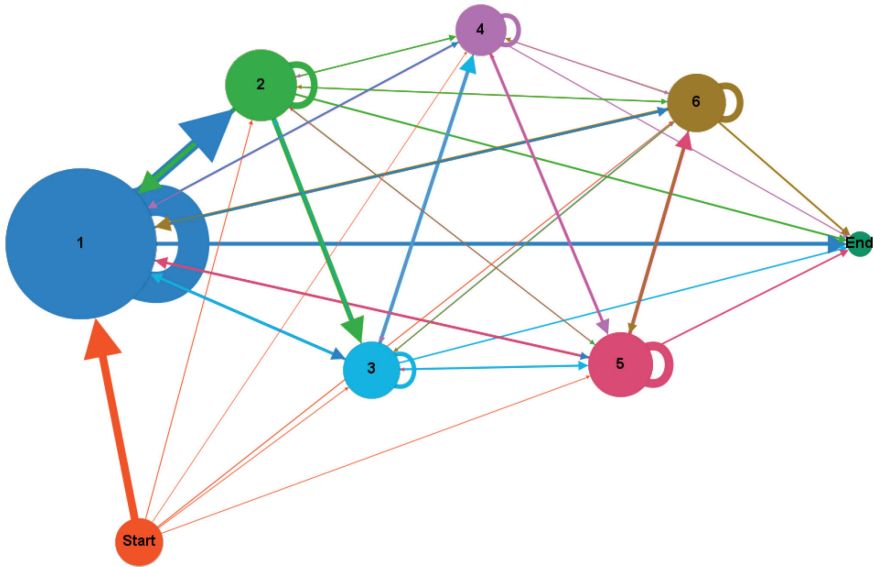
	Niños	Niñas
cluster 1	34,19%	42,16%
cluster 2	26,99%	20,54%
cluster 3	4,37%	3,24%
cluster 4	5,66%	1,62%
cluster 5	28,79%	32,43%

7.3.6. Recompensas

Por último, se ha analizado en profundidad el comportamiento persistencia-recompensa, con el fin de aclarar si este comportamiento se debe a la satisfacción de ganar el reto para afianzar los conocimientos, o si existe un deseo de obtener puntos de recompensa para canjearlos en la tienda virtual y personalizar al personaje principal.

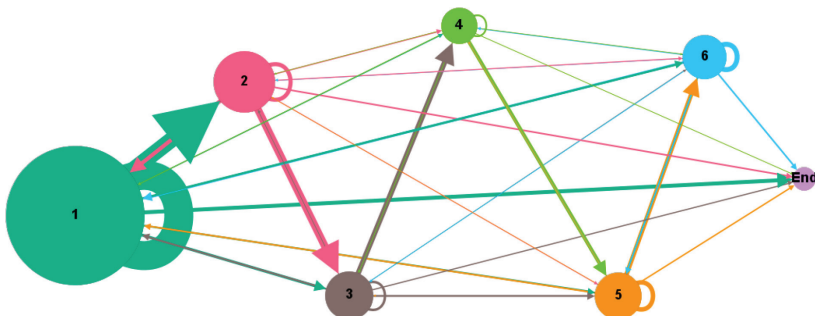
En primer lugar, se analizan los gráficos de trayectorias entre los niveles del juego (véase la Figura 70), donde los nodos (niveles) son más grandes según el número de intentos (persistencia) que se han realizado en ese nivel y las aristas son más gruesas según su grado de salida (grado de salida del nivel: 1 = 10467; 2 = 3835; 3 = 2628; 4 = 2135; 5 = 3302; 6 = 2751). Hay conexiones entre todos los niveles, especialmente del nivel 1 al resto, y de cada nivel al siguiente superior (la arista saliente es del mismo color que el nodo de origen).

Figura 70. Trayectorias generales del juego, donde los nodos son niveles.



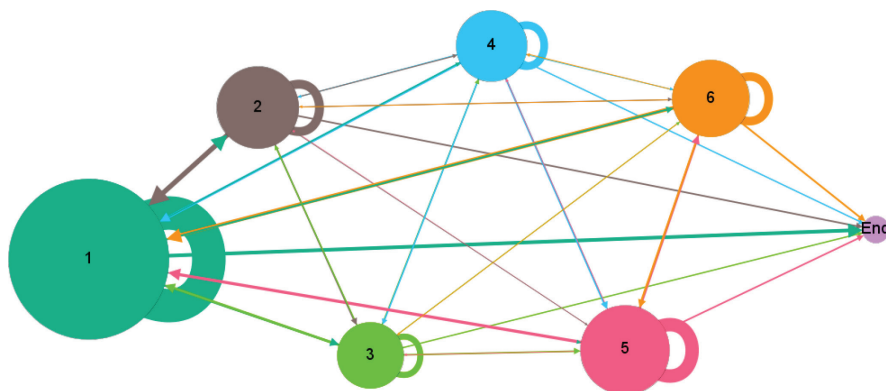
La Figura 71 muestra las trayectorias después de ganar una partida (grado de salida: 1 = 6017; 2 = 1993; 3 = 1308; 4 = 672; 5 = 1228; 6 = 1075). Se puede observar una clara progresión entre niveles en orden del nivel 1 al 6, ya que después de ganar, los usuarios suelen probar el siguiente nivel. Se observan pocas repeticiones de un nivel después de ganar, especialmente en el nivel 4, considerado difícil; y excepto en el nivel 1, el más fácil, con un gran número de repeticiones (bucle grueso en el nodo 1), lo que refuerza la hipótesis de que la repetición después de ganar se debe a un comportamiento de persistencia-recompensa (que se observó anteriormente, especialmente en las chicas), y no a la voluntad de consolidar los conocimientos.

Figura 71. Trayectorias del juego. Persistencia-recompensa (después de ganar la partida).



La Figura 72 muestra las trayectorias después de haber perdido una partida (nivel fuera de grado: 1 = 4450; 2 = 1842; 3 = 1320; 4 = 1463; 5 = 2074; 6 = 1676); se puede observar que hay más persistencia en cada nivel (bucles gruesos en los nodos) si se ha perdido el juego. Aunque el nivel 4 es el más difícil y el menos jugado, no es el que presenta un comportamiento de persistencia-reto más bajo, los jugadores persisten en el reto, incluso después de perder, y a pesar de que es el nivel más difícil, en el que pierden más veces, y en el que los chicos tienen un rendimiento especialmente bajo en comparación con las chicas. Además, el nivel 3 es el nivel en el que se observa la menor persistencia después de perder (bucle fino en el nodo 3) y el nivel en el que los chicos muestran el mayor rendimiento en comparación con las chicas. Estas observaciones refuerzan nuestra hipótesis de que este comportamiento de persistencia-reto es más frecuente en los chicos que en las chicas.

Figura 72. Trayectorias del juego. Persistencia-reto (después de perder la partida).



La Tabla 45 muestra el comportamiento de persistencia-reto en términos del porcentaje de jugadores que, habiendo perdido el nivel, lo repiten; y el comportamiento de persistencia-recompensa en términos del porcentaje de jugadores que, habiendo ganado el nivel, lo repiten, por edad, género y nivel. El comportamiento de persistencia-reto es claramente menor en las niñas, especialmente en los niveles más difíciles (2 y 4), y a la edad de 4 años (prueba de Chi-Cuadrado $p = .018 < .05$). Por el contrario, la conducta de persistencia-recompensa se observa más en las niñas, especialmente en los niveles más fáciles, lo que nos lleva a nuestra hipótesis de que el objetivo final de esta conducta es obtener puntos de recompensa para canjearlos en la tienda por objetos. Esta hipótesis se confirma ya que el análisis del número de puntos gastados y de las visitas a la tienda es significativamente mayor en las niñas (prueba t de Student $p = .003 < .01$). También se ha detectado un comportamiento particular extremo de persistencia-recompensa, en el que se recogen objetivos parciales de poco valor y se abandona la partida antes

de superar el reto, continuadamente. Este comportamiento se ha detectado sobre todo en los niños.

Tabla 45. Persistencia-reto y persistencia-recompensa por nivel, edad y sexo. porcentaje de jugadores que repiten el nivel.

Edad	Nivel	Comportamiento			
		Persistencia-reto		Persistencia-recompensa	
		Niños	Niñas	Niños	Niñas
4	1	81,15%	73,50%	67,50%	50,65%
	2	45,99%	30,77%	21,43%	10,64%
	3	40,74%	48,28%	16,67%	0,00%
	4	40,82%	13,04%	0,00%	14,29%
	5	44,23%	24,14%	9,52%	10,34%
	6	35,29%	39,47%	52,00%	30,00%
5	1	71,18%	79,65%	56,87%	62,53%
	2	43,93%	34,25%	29,10%	11,69%
	3	44,51%	52,63%	38,03%	22,03%
	4	47,80%	33,90%	53,13%	27,27%
	5	57,53%	36,96%	61,43%	20,69%
	6	39,08%	48,00%	50,00%	15,79%
6	1	70,10%	70,64%	47,88%	60,22%
	2	50,51%	38,96%	19,84%	11,32%
	3	43,72%	40,48%	12,68%	21,92%
	4	50,96%	43,59%	26,60%	11,29%
	5	61,31%	62,13%	41,35%	32,54%
	6	51,04%	46,11%	45,49%	48,94%
7	1	71,85%	78,75%	61,13%	50,99%
	2	44,65%	54,10%	14,58%	23,73%
	3	47,00%	53,33%	5,94%	28,99%
	4	53,68%	47,46%	14,81%	5,71%
	5	66,85%	65,43%	41,67%	20,51%
	6	51,65%	45,22%	24,56%	50,82%

7.4. Conclusiones

A partir de los resultados obtenidos en BAC y apoyados en la base teórica inicial, podemos concluir que el uso de un videojuego voluntario basado en la MI, al que se han aplicado técnicas de GLA y visualizaciones comparativas, puede aportar hallazgos que pueden ser relevantes en cuanto a intereses, habilidades, micropersistencia y comportamiento en respuesta a recompensas.

7.4.1. Interés en aprender a programar y resolver problemas computacionales

En línea con investigaciones anteriores [227], a priori parece haber un mayor interés por aprender a programar en los niños que en las niñas, ya que el 70% de las descargas corresponden a usuarios masculinos y, una vez instalado BAC, los niños juegan más a menudo y pasan más tiempo jugando hasta los 5 años. Sin embargo, a partir de los 6 años, el tiempo de juego y el número de partidas es mucho mayor en las niñas que en los niños. Este drástico cambio de comportamiento ha requerido un análisis más profundo en términos de persistencia.

El rendimiento en todas las edades es significativamente mayor en las niñas que en los niños, especialmente de los 3 a los 5 años, cuando las niñas parecen ser más hábiles que los niños en la resolución de los retos, ya que necesitan mucho menos tiempo e intentos para superar los retos y su rendimiento (número de partidas ganadas sobre las jugadas) es mayor, por lo que deducimos que podrían tener un claro mayor dominio que los niños. Sin embargo, de los 6 a los 8 años, aunque su rendimiento sigue siendo mejor, las niñas pasan mucho más tiempo jugando y, según los resultados de nuestra investigación, este comportamiento no está relacionado con el dominio de los conceptos abordados, sino que se debe a un comportamiento de persistencia- recompensa.

Con respecto a la resolución de problemas computacionales específicos, las chicas tienen un mayor interés y rendimiento en los niveles más difíciles (bucles anidados y secuencias largas), en comparación con los chicos, que tienen un interés similar por cada concepto, pero un mayor rendimiento sólo en el nivel que aborda los bucles simples, que es de dificultad media. Estos resultados coinciden con investigaciones anteriores [117], [228], en las que los chicos obtuvieron mejores resultados que las chicas en los niveles intermedios, mientras que esta tendencia se invirtió en los niveles más difíciles.

Las secuencias simples parecen dominarse a partir de los 3 años de edad, independientemente del género. Sin embargo, la memoria de trabajo [44], [229],

que se considera un indicador de la adquisición de habilidades de programación [1], parece desempeñar un papel importante cuando se trabaja con secuencias largas y esto puede influir en el aprendizaje, ya que no se dominan hasta la edad de 5 años, independientemente del género. Los bucles simples se dominan en torno a los 3 años para los niños y a los 4 años para las niñas. En cambio, los bucles anidados parecen superarse a partir de los 5 años para las niñas y de los 6 años para los niños. Por último, en lo que respecta a los condicionales, el rendimiento aumenta a los 4 años, independientemente del sexo. **Todos los conceptos abordados pueden ser superados entre los 3 y los 6 años, lo que podría ser un aspecto a tener en cuenta para la elaboración de los programas escolares.**

7.4.2. Habilidades y logros a través de los intentos en el juego

En cuanto a la progresión, tanto las niñas como los niños mejoran a lo largo de los intentos, pero la mejora de los niños es más significativa tanto en tiempo como en rendimiento. Sin embargo, las niñas obtienen un logro significativamente mayor que los niños y alcanzan el logro completo en todos los niveles en los primeros 10 intentos entre los 3 y 5 años, mientras que este comportamiento no se observa en el caso de los niños. A partir de los 6 años, las niñas tardan más tiempo e intentos en resolver los retos debido a un comportamiento de persistencia-recompensa.

Es destacable que, aunque los bucles anidados parecen ser difíciles para los alumnos de primaria [215], [224], es posible dominar completamente este concepto a los 4 años, independientemente del género, si se realizan suficientes intentos y, además, los niños son capaces de persistir en el reto voluntariamente, intrínsecamente motivados, hasta dominarlo. Este podría ser también un aspecto a tener en cuenta para el desarrollo de los programas escolares, ya que todos los conceptos abordados (secuencias, bucles simples y anidados, y condicionales) podrían dominarse a los 4 años si existe una persistencia intrínsecamente motivada.

7.4.3. Comportamientos de juego: persistencia y recompensas

En un entorno basado en el juego, una reacción positiva al fracaso, en forma de persistencia ante un reto, puede predecir un mejor rendimiento académico [230], [231], y, a su vez, los jugadores están muy motivados para repetir la tarea y superar el reto, y el desarrollo de este estilo motivacional positivo puede conducir al éxito educativo [212]. Por otra parte, las recompensas en los videojuegos son muy eficaces para entrenar conductas, ya que enseñan a los jugadores que la persistencia ante el fracaso tiene recompensas [211], [212].

Este comportamiento de persistencia ante el fracaso, que hemos llamado comportamiento de persistencia-reto (repetición a pesar de perder) se ha observado claramente en ambos géneros, pero es más pronunciado en los chicos, lo que podría explicar por qué su progreso está altamente correlacionado con la persistencia. En cambio, el comportamiento de persistencia-recompensa (repetición después de ganar) se observa significativamente más en las niñas, especialmente entre los 5 y los 7 años, cuando ya dominan todos los conceptos abordados.

De nuestro análisis podemos concluir que este comportamiento de persistencia-recompensa es principalmente para la acumulación de puntos de recompensa y su intercambio por objetos virtuales en el juego, y no para la asimilación de conceptos. Creemos que este puede ser un hallazgo relevante de nuestra investigación, ya que estos usuarios no juegan con el objetivo de superar retos o aprender conceptos, sino con el único objetivo de recibir una recompensa. Aunque el aprendizaje derivado de este comportamiento es posible, este aprendizaje podría ser reducido ya que el objetivo no es superar el reto, de hecho, se ha detectado un comportamiento extremo en este sentido en el que los jugadores juegan sucesivos intentos y en todos ellos abandonan el reto tras haber acumulado recompensas parciales con el claro objetivo de acumular puntos para canjear, pero sin superar el reto en ninguno de los intentos. Dado que los niños son capaces de dominar los conceptos abordados antes de los 5 años mediante la práctica continuada, nos preguntamos si no es contraproducente retrasar la adquisición de estos conceptos hasta después de los 5 años, cuando la conducta de persistencia-reto es menor y surge la de persistencia-recompensa, especialmente en las niñas.

Por otro lado, para el aprendizaje de la programación a partir de los 5 años, también se podría sugerir la motivación a través de recompensas, especialmente en las niñas, ya que se sabe que las recompensas tienen un efecto motivacional o metacognitivo positivo en el aprendizaje de forma que aumenta el compromiso con el juego [215], pero siempre teniendo en cuenta no sobrepasar el límite en el que las recompensas pueden ser perjudiciales para el aprendizaje, disminuyendo la motivación intrínseca [208] y llegando potencialmente al caso extremo detectado en el que ya no hay intención de aprender, y también teniendo cuidado de no sobrepasar el “punto dulce” [12], [232] en el que hay un equilibrio entre el reto y la frustración.

Las posibles aplicaciones derivadas de los resultados de la investigación se basan en la recomendación de analizar los aspectos de persistencia y recompensas en los juegos educativos. El entorno BAC podría utilizarse en contextos escolares, además de para el aprendizaje de la programación y su evaluación, para analizar aspectos relacionados con la motivación intrínseca, pero, en este caso, siempre que su uso sea totalmente voluntario. En este mismo sentido, se recomienda incorporar juegos educativos basados en

la motivación intrínseca en los entornos escolares. Los currículos escolares podrían diseñarse teniendo en cuenta que, a través de esta motivación, los niños pueden ser capaces de superar los conceptos computacionales abordados (incluidos los bucles anidados) antes de los 5 años, existiendo diferencias de género en la adquisición de estos conceptos. También se recomienda diseñar juegos educativos teniendo en cuenta estos aspectos, y de esta forma potenciar la conducta de persistencia-reto en y, a partir de los 5 años, controlar o potenciar la conducta de persistencia-recompensa respecto a las niñas.

En conclusión, al utilizar un juego completamente voluntario y no controlado, se pueden explorar en detalle aspectos conductuales asociados a la motivación intrínseca, como la progresión en los intentos voluntarios, la persistencia y el comportamiento en respuesta a las recompensas. Creemos que nuestros hallazgos podrían guiar el diseño de mejores herramientas para el aprendizaje de la programación y el desarrollo del PC. Dado que la edad y el género influyen en el rendimiento en estos retos, es aconsejable adaptar las herramientas de aprendizaje en consecuencia, así como el currículo escolar. Además, los entornos de aprendizaje podrían adaptar la forma de proporcionar refuerzos y recompensas, especialmente para los niños en los retos más complejos y para las niñas a partir de los 5 años. Sin embargo, se necesitan nuevos estudios que tengan en cuenta estas complejidades.

Aunque en esta investigación se han aportado nuevos enfoques de aprendizaje basados en la motivación intrínseca, siguen existiendo preguntas abiertas, por ejemplo: ¿cómo podrían incluirse en los entornos escolares los entornos de aprendizaje basados en la motivación intrínseca? ¿Podrían extraerse conclusiones relevantes del análisis del código construido por los jugadores para superar los retos? Además de analizar los datos posteriores al juego, como se ha hecho en este análisis, es posible utilizar nuestras conclusiones para mejorar los entornos de aprendizaje que hacen uso del GLA para predecir y adaptar el aprendizaje, pero ¿cómo podrían implementarse directamente los elementos predictivos y adaptativos en un videojuego voluntario sin reducir la motivación intrínseca? Además, hay otros conceptos y habilidades relacionados con el PC que no se han abordado y podrían desarrollarse en otras versiones de BAC en futuras investigaciones.

La motivación intrínseca correlaciona positivamente con el aprendizaje, el interés, la persistencia, la auto-competencia y la autoeficacia [204], [205], [213], [220]. Además, los estilos motivacionales específicos afectan directamente a los logros y al rendimiento académico, y estos estilos podrían detectarse y desarrollarse a través de los videojuegos [211]. Dados los hallazgos que se han reportado en esta investigación y las preguntas que quedan abiertas, creemos que se debe investigar más sobre los comportamientos que surgen de la motivación intrínseca, utilizando entornos voluntarios de aprendizaje y evaluación basados en juegos. También consideramos que las técnicas de

GLA y las visualizaciones comparativas son apropiadas para analizar estos comportamientos.

7.5. Limitaciones y líneas de actuación

Esta investigación tiene algunas limitaciones, por ejemplo, es posible que algunos usuarios se hayan conectado dos veces con distintos dispositivos y hayan sido tratados como dos personas diferentes; que hayan introducido datos erróneos con respecto a la edad o el sexo; o que hayan dejado el juego encendido por error, lo que ha repercutido en el tiempo de juego. Sin embargo, dado el gran volumen de datos recogidos de diferentes poblaciones y el procesamiento de los datos, este impacto se ha minimizado.

Como se ha visto en las conclusiones, de esta experiencia se extraen hallazgos relevantes que pueden ayudar tanto al diseño de juegos educativos con elementos de MI, persistencia y recompensas, así como al diseño del currículo escolar en cuanto al aprendizaje de la programación y el PC, determinando qué puede aprenderse a cada edad y las diferencias que hay en cuanto al género en la adquisición de los diferentes conceptos computacionales.

Según las conclusiones anteriores, las líneas futuras de investigación podrían ser:

- Inclusión de entornos basados en la MI en contextos escolares.
- Análisis del código propuesto por los estudiantes en términos de optimización y estilos de programación utilizando GLA.
- Ampliación de BAC incluyendo los hallazgos sobre comportamientos de juego.
- Ampliación de BAC para hacer uso de GLA adaptativo sin reducir la MI.
- Ampliación de BAC con más conceptos computacionales.

Gracias a esta experiencia, se ha confirmado la *HE5: Los entornos de aprendizaje basados en juegos voluntarios pueden aportar información sobre disposiciones y predisposiciones para el aprendizaje del PC según las características del individuo en cuanto a edad y género*, y se han alcanzado los objetivos específicos de este capítulo.

En cuanto a los objetivos generales de la tesis, se ha completado el objetivo OP4 y el objetivo global de la tesis OP. En este punto, el grado de cumplimiento de los objetivos generales es el que se muestra en la Tabla 46.

Tabla 46. Seguimiento de los objetivos, capítulo 7.

Objetivos			Grado de cumplimiento		
			Bajo	Medio	Alto
OP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.				
	OP1	Marco teórico y estado del arte			
		OP1a Definiciones teóricas y operacionales del PC			
		OP1b Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas			
		OP1c Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes			
		OP1d Marco teórico y estrategias para la evaluación del PC e instrumentos existentes			
	OP2	Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.			
		Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC			
	OP3	OP3a Diseñar y validar un instrumento de evaluación del PC en edades tempranas			
		OP3b Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas			

	OP4	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC	
		OP4a Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género	
		OP4b Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género	
		OP4c Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género	
OC		Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones	

Las publicación relativa a esta experiencia es la siguiente:

- M. Zapata-Cáceres and E. Martín-Barroso, "Applying Game Learning Analytics to a Voluntary Video Game: Intrinsic Motivation, Persistence, and Rewards in Learning to Program at an Early Age," in *IEEE Access*, vol. 9, pp. 123588-123602, 2021, doi: 10.1109/ACCESS.2021.3110475.

CAPÍTULO 8. CONCLUSIONES Y LÍNEAS DE ACTUACIÓN

Como se ha descrito en esta tesis, el aprendizaje y evaluación de habilidades que contribuyen al desarrollo del PC en la educación reglada está cobrando una gran importancia a nivel internacional, habiendo consenso en que este aprendizaje debe comenzar en edades tempranas. Debido a su carácter reciente, aún no existen instrumentos validados y consensuados para el aprendizaje y la evaluación del PC y, además, es necesario establecer cuáles son las habilidades alcanzables y si hay diferencias en la adquisición de estas habilidades según la edad y género de los niños.

A través de la revisión sistemática de la literatura en cuanto al PC teórico y operacional, se extrajo una propuesta de definición y descomposición del PC fundamentada, así como varias hipótesis sobre los sistemas de aprendizaje y de evaluación que se comprobaron más adelante. Una de las contribuciones más importantes existentes en la literatura era la necesidad de un *sistema de evaluación* [148], [161] constituido por varios instrumentos de medida diferentes. Una de las hipótesis de la tesis planteaba el posible impacto de cada uno de estos instrumentos de evaluación en las dimensiones del marco 3D [46] (siendo éste el marco de referencia más utilizado para el PC) y, a través de las experiencias realizadas, se comprobó el impacto esperado (véase la Figura 11). En esta tesis se propone no sólo la creación de un *sistema de evaluación*, sino también la creación de un *sistema de aprendizaje* del PC, en donde se utilicen diferentes estrategias pedagógicas de aprendizaje combinadas. Mediante el desarrollo expreso de un entorno de aprendizaje, se evalúa también el impacto en el aprendizaje del PC de cada una de las estrategias en las dimensiones del marco 3D (véase la Figura 10).

Mediante los **instrumentos desarrollados** en esta tesis, tanto para el aprendizaje como para la evaluación del PC y los hallazgos en cuanto a las habilidades relacionadas, en etapas tempranas, extraídos de los sucesivos desarrollos y **experiencias** llevadas a cabo en **centros escolares y no escolares**, se ha cumplido el objetivo principal (*OP*) de la tesis, así como todos los objetivos específicos indicados en el apartado 1.2, avanzando en

el conocimiento del PC e incidiendo en su componente operacional. También se han confirmado las hipótesis propuestas (apartado 3.2) a lo largo de los desarrollos instrumentales y experiencias realizadas (apartado 3.3). En cuanto al objetivo común OC, dado que corresponde a una colaboración internacional, consideramos que se ha realizado una gran contribución a la investigación global, y se tiene la intención de seguir avanzando en este sentido manteniendo las colaboraciones y trabajos en marcha con otros países (apartado 8.2). En la Tabla 47 puede verse un resumen de la estructura de la tesis en cuanto a objetivos, hipótesis y experiencias realizadas (que se enumeran a continuación):

- **E1:** Desarrollo y validación de prueba independiente para la evaluación del PC en Educación Primaria: Beginners Computational Thinking test (BCTt).
- **E2:** Estimación de la validez del BCTt en alumnos de Educación Infantil.
- **E3:** Desarrollo y validación de videojuego para la enseñanza y evaluación del PC: Blue Ant Code (BAC) en Educación Primaria.
- **E4:** Análisis de motivación intrínseca, intereses, persistencia y comportamiento frente a recompensas, en etapas tempranas, utilizando el entorno BAC para el aprendizaje de la programación de forma voluntaria.

Tabla 47. Resumen de objetivos, hipótesis y experiencias realizadas.

Objetivos		Grado cuml.	Inv.	Exp.	Hip.	Cap.
OP	Avanzar en el conocimiento sobre el PC, en su componente operacional, en cuanto a métodos de enseñanza y de evaluación eficaces en etapas tempranas.	Alto	Global	-	-	Todos
	Marco teórico y estado del arte	Alto	Teórica			2
	OP1a Definiciones teóricas y operacionales del PC	Alto				2
	OP1b Marcos e iniciativas para el desarrollo del PC a nivel internacional y nacional en etapas tempranas	Alto				2
	OP1c Marco teórico y estrategias para el desarrollo del PC y herramientas y entornos existentes	Alto				2
	OP1d Marco teórico y estrategias para la evaluación del PC e instrumentos existentes	Alto				2
	OP2 Establecer una propuesta operativa del PC en base al OP1 para constituir la dirección de la investigación empírica mediante hipótesis y objetivos base.	Alto	Pro- puesta operativa	E1, E2, E3, E4	H1, H2, H3, H4, H5	3
	Diseñar y validar empíricamente, herramientas para el aprendizaje y la evaluación del PC	Alto	Empírica			4, 5, 6 7
	OP3a Diseñar y validar un instrumento de evaluación del PC en edades tempranas	Alto		E1, E2	H1	4, 5
	OP3b Desarrollar y validar una herramienta de aprendizaje y evaluación del PC en edades tempranas	Alto				6, 7 H4, H5

Objetivos		Grado cumpl.	Inv.	Exp.	Hip.	Cap.
OP4	Realizar experiencias empíricas que permitan analizar datos relativos tanto a habilidades de PC como a disposiciones y predisposiciones para el PC	Alto		E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
	OP4a Extraer y analizar datos para mejorar el diseño del currículo escolar relativo a PC según edad y género	Alto		E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
	OP4b Extraer y analizar datos para mejorar el diseño de herramientas de aprendizaje relativo a PC según edad y género	Alto		E3, E4	H2, H3, H4, H5	6, 7
	OP4c Extraer y analizar datos para mejorar el diseño de herramientas de evaluación relativo a PC según edad y género	Alto		E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
OC	Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones	medio		E1, E2	H1	4, 5

Grado cumpl.: Grado de cumplimiento del objetivo planteado; Inv.: tipo de investigación realizada; Exp.: número de experiencia; Hip.: hipótesis contemplada; Cap: capítulo en donde describe la experiencia.

En los siguientes apartados, se resumen las principales contribuciones, conclusiones y hallazgos en el contexto de la tesis.

8.1. Beginners Computational Thinking Test (BCTt)

El principal obstáculo encontrado al inicio de la investigación fue la inexistencia, en ese momento, de un instrumento autónomo (independiente de un entorno de aprendizaje) para la evaluación del PC en Educación Primaria, lo que dificultaba enormemente evaluar o validar nuevos instrumentos tanto de aprendizaje como de evaluación no-autónomos, sin posibilidad de poder realizar experiencias de tipo pre y post-test para evaluar la idoneidad de nuevos entornos y herramientas para el desarrollo del PC.

Esta carencia de instrumentos hacía muy difícil reconocer de forma aséptica las habilidades a cada edad, ya que podía haber sesgos debido a que ciertas habilidades pudieran ser más o menos alcanzables debido a las características del entorno particular utilizado. Por este motivo se llevó a cabo las experiencias enmarcadas en E1 (capítulo 4), en las que se **desarrolló un instrumento autónomo, el Beginners Computational Thinking Test (BCTt)** [7], [8], y se validó en la etapa de Educación Primaria, en tres centros educativos. Los datos obtenidos mostraron una alta fiabilidad con un Alfa de Cronbach=0,824 en toda la etapa de Educación Primaria, aunque la recomendación final es que el BCTt es idóneo para 1º y 2º de primaria principalmente. Al constatar una reducción de la fiabilidad del test en los alumnos más mayores, se desarrolló, a través la colaboración con un equipo de la *Ecole Polytechnique Fédérale de Lausanne (EPFL)* en Suiza, el **competence Computational Thinking test (cCTt)** [165], y se validó en centros educativos de Suiza. Además, se establecen los límites de edad para la administración del BCTt y el cCTt a través de una experiencia administrando ambos test a estudiantes de Educación Primaria en centros educativos de Portugal. Los resultados del análisis psicométrico indican que se debería preferir el cCTt para los alumnos de 3º y 4º curso. No obstante, el BCTt puede utilizarse si el objetivo es discriminar entre alumnos con capacidades bajas de estos dos cursos.

Por otro lado, para establecer el límite inferior de edad en el que se puede administrar el BCTt y a través de la colaboración internacional con la *Fontys University of Applied Science* y la *Zuyd University of Applied Science*, ambas en Países Bajos, se llevó a cabo la experiencia E2 (capítulo 5) utilizando entornos de aprendizaje robóticos en Educación Infantil, con niños de 4 y 5 años, en un centro escolar de los Países Bajos [6]. Mediante esta experiencia, se determina la idoneidad de estos entornos de aprendizaje para el desarrollo del PC. Al mismo tiempo, se confirma la fiabilidad del BCTt en niños de 4 años (Alfa de Cronbach = 0,802). Sin embargo, cuatro de los seis conceptos

computacionales que se abordan en el test parecen no ser alcanzables para los alumnos de 4 años, ya que no fueron capaces de responder a las preguntas relacionadas con ellos ni mostraron ninguna mejora en el post-test respecto a dichos conceptos. Por ello, es necesario el diseño de otra versión del BCTt dirigida a los alumnos de 4 años o menos. Esta versión se está actualmente realizando en colaboración con un equipo de la *Open University Netherlands (OU)*, en Países Bajos.

El BCTt y el cCTt son instrumentos autónomos, independientes de cualquier entorno, que se han revelado fiables para la evaluación del PC en Educación Primaria y pueden ser administrados como pre-test y post-test en las investigaciones que lo requieran. Estos test pueden considerarse una extensión fiable del CTt de Román *et al.* para estudiantes más jóvenes, y se enmarcan en el objetivo internacional común OC de creación de instrumentos independientes de evaluación del PC para todos los rangos de edad y poblaciones. En la Tabla 48 se muestran los instrumentos aplicables a cada edad en el contexto del objetivo OC.

Tabla 48. Tabla de los instrumentos en línea con el objetivo común OC.

Marco de desarrollo para el OC	Actual, colaboración Países Bajos	Tesis	Tesis, colaboración Suiza	Equipo de la UNED ⁶	Equipo Suizo
Tests independientes	En progreso	BCTt	cCTt	CTt	ATTA
Edades diana	4 y menos	4 a 7	7 a 9	10 a 16	16 +
Publicaciones	En progreso	[6]-[8]	[165]	[44], [142], [143], [161]	[233]

En la Tabla 49 se muestran las habilidades de PC alcanzables y/o nivel de adquisición posible, que se desprende de forma preliminar de la administración simple del test, en las experiencias E1 y E2.

⁶ <https://www.uned.es/>

Tabla 49. Habilidades de PC alcanzables y/o nivel de adquisición posible en primera instancia.

Exp.	Etapa	Ciclo	Edad (años)	Secuencias	Bucle simple	Bucle anidado	If-then	If-then-else	While
E2	Infantil	3	4	Alcanzable	Alcanzable	No alcanzable	No alcanzable	No alcanzable	No alcanzable
		3	5	Alcanzable	Alcanzable	Alcanzable	Alcanzable	Alcanzable	Alcanzable
E1	Primaria	1	5 a 8	Alto	Alto-Medio	Medio	Medio-Bajo	Bajo	Medio-Bajo
		2	7 a 10	Alto	Alto	Alto	Medio	Medio	Medio
		3	9 a 12	Alto	Alto	Alto	Alto	Medio	Medio

Además, el BCTt se ha traducido a **numerosos idiomas, se han realizado y actualmente se están llevando a cabo varias experiencias en diferentes poblaciones y países (como Portugal, Bélgica o Singapur) y planteado líneas de actuación futuras.**

8.2. Evaluación del PC: líneas actuales y futuras

Como se ha visto, el BCTt [7], [8] ha tenido gran impacto internacional. Las primeras colaboraciones con Suiza, Países Bajos y Portugal, han resultado en el desarrollo y validación del **competence computational Thinking test (cCTt)** junto a la *Ecole Polytechnique Fédérale de Lausanne (EPFL)*: *MOBOTS Group of the BIOROB Laboratory*; y *LEARN – Center for Learning Sciences*; y *Computer Human Interaction in Learning and Instruction (CHILI)* [165].

Además, se estableció una colaboración entre la *Ecole Polytechnique Fédérale de Lausanne (EPFL)* y la ONG *Tree Tree* ⁷, para la realización de un **estudio comparativo del BCTt y el cCTt** en centros escolares de Portugal. Este estudio está actualmente en fase de revisión para su publicación en la *ITiCSE 2022 (27th Annual Conference on Innovation and Technology in Computer Science Education)* (apartado 4.6.2).

Por otro lado, mediante la colaboración con la *Fontys University of Applied Science* y la *Zuyd University of Applied Science*, ambas en Países Bajos, se pudo establecer el límite de edad inferior para la administración del BCTt [6].

En la actualidad, se están llevando a cabo otras colaboraciones con otros países y poblaciones con respecto al OC: *Diseño y validación de instrumentos de evaluación del PC que cubran un amplio rango de edades y poblaciones, a continuación, se enumeran las más relevantes:*

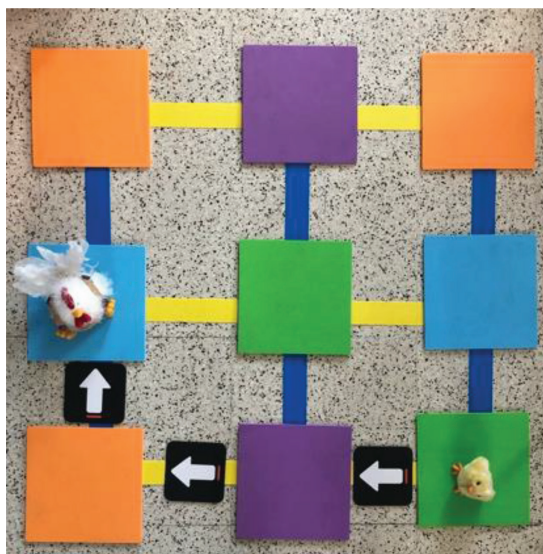
- Realización de un **estudio sobre la posibilidad de desarrollar el PC en Educación Primaria a través de actividades Bebras** [161], mediante otra experiencia en centros escolares de Portugal, utilizando el BCTt como herramienta de pre y post-test. Esta experiencia está en fase de redacción del artículo científico.
- Desarrollo de una **versión del test para niños menores de 4 años** en colaboración con la *Open University (OU)*, en Países Bajos, incluyendo aspectos sensoriales diferentes al visual. Este proyecto se encuentra en su fase inicial y contempla incluir otros elementos sensoriales como el tacto o el oído en el instrumento de evaluación.

⁷ <https://www.treetree2.org/>

- **Validación de límites del BCTt y estudio de habilidades del PC en Educación Primaria:** colaboración con la **UNED y el INTEF** (Ministerio de Educación) en la administración del BCTt en 2.500 estudiantes de entre alumnos de infantil (5 años) y 3º de primaria. Los estudios preliminares indican una fiabilidad muy buena (Alfa de Cronbach = 0.87). Este estudio está en fase de redacción.
- **Validación del BCTt en otras poblaciones**, en colaboración con *el Instituto Universitario de Lisboa (ISCTE)*, en la que se administró el BCTt en colegios de Portugal a +600 alumnos. Estos resultados están en fase de análisis.
- Integrar los conceptos e ideas de la educación del PC en las prácticas de los profesores de Educación Primaria es un reto difícil. Los profesores deben adquirir nuevos conocimientos y habilidades relacionados con la forma en que los contenidos, las estrategias pedagógicas y las herramientas TIC, deben combinarse para introducir el PC en sus prácticas educativas de manera significativa [38]. En este contexto, se plantea un estudio de **percepciones del profesorado sobre el PC en el currículo escolar**. Esta investigación resulta de una colaboración entre 4 países: **España, Países Bajos, Portugal y Singapur**, y está en estos momentos en la fase de recogida de datos. Se prevee la presentación del estudio en el congreso CTE-STEM 2022 (Sixth APSCE International Conference on Computational Thinking and STEM Education) con el título: "Understanding Teachers' Attitudes and Self-Assessment Towards Computational Thinking".
- **Creación de un equipo internacional** (España, Países Bajos, Bélgica, Portugal, Suiza), a través de una convocatoria **Erasmus +** para la inclusión del PC en el currículo de forma estandarizada, desarrollo y validación de instrumentos para su evaluación y atención especial a la diversidad, ya que la integración del PC en la educación formal puede ser una oportunidad para redefinir el currículo escolar en la educación informática, de forma hacia una mayor inclusión y diversidad [34]. Este proyecto se encuentra en fase de propuesta y está enmarcado especialmente en el objetivo común de la tesis OC.

Además, está habiendo algunas iniciativas independientes y se están realizando experiencias utilizando el BCTt en multitud de países como, por ejemplo, en **Bélgica**, el *Service d'Ingénierie Pédagogique et du Numérique éducatif*, de la Universidad Umons-FPSE ha desarrollado una versión desenchufada del BCTt (véase la Figura 73) para niños de 3 y 4 años: **Beginners Computational Thinking Test: Adaptation pour un public de moins de 5 ans** [234], que ha resultado muy fiable para la evaluación del PC.

Figura 73. Versión desenchufada del BCTt [234].



También se están desarrollando investigaciones a gran escala, por ejemplo, en colegios de **Italia** utilizando el BCTt, a través de un proyecto financiado por el **Ministerio Italiano de Educación, Universidades e Investigación**. En **Suiza**, la *Ecole Polytechnique Fédérale de Lausanne (EPFL)* incluyó el BCTt en el sitio Web de Roteco⁸, que aúna las siguientes instituciones: Academias Suizas de Ciencias, SUPSI, la EPFL, la ETHZ y escuelas St. Gallen, Lucerna, Schwyz y Valais. También hay un inicio de colaboración con profesores e **investigadores de China y el Ministerio de Educación de Singapur**, así como comienzos de colaboraciones con Cornell University (EEUU), University of Hradec Kralove (República Checa), University of Tübingen (Alemania), Vilnius University (Lituania), y la Université Lumière Lyon 2 (ISPEF) (Francia), entre otras.

Asimismo, para estas investigaciones en múltiples poblaciones, el BCTt se ha traducido a **numerosos idiomas**, por ejemplo, italiano, neerlandés, portugués, alemán, francés, chino o farsi.

8.3. Blue Ant Code (BAC)

En el contexto de la tesis, se ha **desarrollado un entorno de aprendizaje y evaluación** a través de analíticas de aprendizaje. El desarrollo del entorno

⁸ <https://www.roteco.ch/fr/nouveautes/comment-evaluer-lefficacite-de-la-pensee-computationnelle-chez-les-eleves-du-primaire>

BAC utilizado incluye el videojuego compatible con múltiples plataformas, una api REST para la conexión con la base de datos relacional y módulos complementarios para el filtrado, la visualización y análisis de datos con GLA. Este entorno se utilizó en la experiencia *E3: Validación y análisis del entorno educativo basado en el juego BAC, en modos individual y colaborativo y análisis de habilidades de PC en Educación Primaria* (véase capítulo 6). Además, se desarrolló una **segunda versión de BAC para su adaptación al entorno masivo no escolar**, incluyendo nuevas funcionalidades como la inclusión de un sistema de recompensas, que se utilizó en la *E4: Análisis de competencias: motivación intrínseca, persistencia y recompensas mediante analíticas de aprendizaje con BAC* (véase capítulo 7).

La investigación sobre cómo enseñar PC y especialmente en las dimensiones de prácticas y perspectivas computacionales del marco 3D es escasa, por lo que en esta tesis se propone un *sistema de aprendizaje* combinando tres de las estrategias de aprendizaje más utilizadas en el entorno BAC, siendo éste un entorno colaborativo basado en el juego, construido sobre una estrategia constructivista basada en problemas, que ha sido validado en la etapa de Educación Primaria en la experiencia *E3*. Los resultados de esta experiencia muestran que este enfoque es adecuado para el aprendizaje del PC, especialmente en edades tempranas, cubriendo las tres dimensiones clave del marco 3D (véase la Figura 49), como se planteó en la hipótesis *HE2*, y se ha confirmado mediante el caso de estudio. La combinación de diferentes metodologías de evaluación, como un *sistema de evaluación* [148]: BCTt, GLA a través de GBL, y datos cualitativos (observación directa y cuestionarios), permite una evaluación tridimensional, como se planteó en la *HE4* y se ha confirmado también mediante este caso de estudio (véase la Figura 50).

En cuanto a la **validación de BAC en Educación Primaria**, los resultados de la experiencia *E3* confirmaron que es un **entorno de aprendizaje apropiado**, con mejoras muy significativas en el desarrollo del PC incluso con poco tiempo de interacción con el entorno. Además, también resultó **adecuado como herramienta de evaluación** con resultados coherentes con el análisis de los datos recogidos por BAC y, además, puede ofrecer datos en tiempo real que pueden ser útiles para una evaluación personal detallada, un fuerte análisis de datos de bajo nivel, y la identificación de conceptos computacionales inalcanzables para cada edad específica no solamente en un instante determinado, sino a lo largo y al final de un proceso de aprendizaje específico. No obstante, tal y como ocurrió con el BCTt, **BAC parece estar mejor adaptado a los estudiantes más jóvenes e incluso ser adecuado para cursos inferiores** (Educación Infantil), sobre todo en el modo colaborativo. Además, BAC es especialmente adecuado y motivador para alumnos con **necesidades especiales o de bajo percentil** particularmente en la modalidad de juego colaborativo.

También se obtuvieron resultados positivos en la prueba de retención del aprendizaje, lo que sugiere que esta **metodología basada en el juego fomenta el aprendizaje duradero**. Los resultados de retención del aprendizaje fueron mejores en la modalidad colaborativa que en la individual, sobre todo en los cursos inferiores, lo que nos lleva a la conclusión de que la modalidad de juego colaborativo fomenta un aprendizaje más eficaz y duradero. Sin embargo, **es posible que la estrategia colaborativa no sea tan eficaz para el último ciclo de Educación Primaria**, ya que estos alumnos parecen ser capaces de una reflexión individual más eficaz. Se necesitan más investigaciones para comprobar esta hipótesis.

Uno de los hallazgos más relevantes es que los alumnos de los primeros cursos de Educación Primaria fueron capaces de superar las habilidades de PC los alumnos de cursos superiores al final de la experiencia, con lo que en esta tesis se resalta la **importancia de desarrollar el PC desde edades muy tempranas** y la **recomendación de incorporar ciertos conceptos, como los bucles anidados o los condicionales, en el currículo ya en primer curso o incluso antes**, en lugar de esperar hasta el último ciclo de Educación Primaria (o incluso secundaria), como ocurre actualmente en la mayoría de los centros educativos en España.

La experiencia *E4*, de **despliegue masivo de BAC en un entorno no escolar**, se realizó con el objetivo general de analizar estrategias para el aprendizaje y evaluación del PC y las habilidades de PC de alumnos de Educación Primaria, con especial atención en las disposiciones y predisposiciones para el PC basadas en la motivación intrínseca y el juego voluntario. Para esta experiencia, se utilizó el entorno adaptado de BAC para el despliegue masivo. A continuación, se resumen los principales hallazgos de esta experiencia:

- **Interés inicial:** A priori parece haber un mayor interés por aprender a programar en los niños que en las niñas, los niños juegan más a menudo y pasan más tiempo jugando hasta los 5 años; sin embargo, a partir de los 6 años, el tiempo de juego y el número de partidas es mucho mayor en las niñas que en los niños.
- **Rendimiento:** En todas las edades es significativamente mayor en las niñas que en los niños, especialmente de los 3 a los 5 años. Sin embargo, de los 6 a los 8 años, aunque su rendimiento sigue siendo mejor, las niñas pasan mucho más tiempo jugando y, según los resultados de la investigación realizada, este comportamiento no está relacionado con el dominio de los conceptos abordados, sino que se debe a un comportamiento de persistencia-recompensa.
- **Resolución de problemas computacionales específicos:** Las chicas tienen un mayor interés y rendimiento en los niveles más difíciles (bucles

anidados y secuencias largas), en comparación con los chicos, que tienen un interés similar por cada concepto, pero un mayor rendimiento sólo en bucles simples, de dificultad media. Las secuencias simples parecen dominarse a partir de los 3 años de edad, independientemente del género. Sin embargo, debido a la memoria de trabajo las secuencias largas no se dominan hasta la edad de 5 años, independientemente del género. Los bucles simples se dominan en torno a los 3 años para los niños y a los 4

- años para las niñas. En cambio, los bucles anidados parecen superarse a partir de los 5 años para las niñas y de los 6 años para los niños. Por último, en lo que respecta a los condicionales, el rendimiento aumenta a los 4 años, independientemente del sexo. **Todos los conceptos abordados pueden ser superados entre los 3 y los 6 años, lo que podría ser un aspecto a tener en cuenta para la elaboración de los programas escolares.**
- **Progresión en el aprendizaje:** Tanto las niñas como los niños progresan, pero la mejora de los niños es más significativa tanto en tiempo como en rendimiento. Sin embargo, las niñas obtienen un logro significativamente mayor que los niños y alcanzan el dominio completo en todos los niveles en los primeros 10 intentos entre los 3 y 5 años, mientras que este comportamiento no se observa en el caso de los niños. A partir de los 6 años, las niñas tardan más tiempo e intentos en resolver los retos debido a un comportamiento de persistencia-recompensa. Es destacable que, **aunque los bucles anidados parecen ser difíciles para los alumnos de primaria, es posible dominar completamente este concepto a los 4 años**, independientemente del género, si se realizan suficientes intentos y, además, los niños son capaces de **persistir en el reto voluntariamente, intrínsecamente motivados, hasta dominarlo. Este resultado contrasta con el obtenido en la experiencia E2, en el que muchos de estos conceptos no fueron alcanzables a los 4 años, con lo que se confirma la importancia de la motivación intrínseca en los procesos de aprendizaje.** Este podría ser también un aspecto a tener en cuenta para el desarrollo de los programas escolares, ya que **todos los conceptos abordados (secuencias, bucles simples y anidados, y condicionales) podrían dominarse a los 4 años si existe una persistencia intrínsecamente motivada.**
- **Comportamientos de persistencia ante el reto y frente a las recompensas:** La persistencia-reto (repetición a pesar de perder) se ha observado claramente en ambos géneros, pero es más pronunciado en los niños, lo que podría explicar por qué su progreso está altamente correlacionado con la persistencia. En cambio, el comportamiento de persistencia-recompensa (repetición después de ganar) se observa significativamente más en las niñas, especialmente entre los 5 y los 7 años, cuando ya dominan todos los conceptos abordados. Este es un hallazgo relevante de nuestra investigación ya que, aunque el aprendizaje derivado de este comportamiento

es posible, este aprendizaje podría ser reducido ya que el objetivo no es superar el reto sino obtener recompensas. Dado que los niños son capaces de dominar los conceptos abordados antes de los 5 años mediante la práctica continuada, nos preguntamos si no es **contraproducente retrasar la adquisición de estos conceptos hasta después de los 5 años, cuando la conducta de persistencia- reto es menor y surge la de persistencia-recompensa, especialmente en las niñas**. Por otro lado, **a partir de los 5 años, también se podría sugerir la motivación a través de recompensas, especialmente en las niñas**, pero siempre teniendo en cuenta no sobrepasar el límite en el que las recompensas pueden ser perjudiciales para el aprendizaje, disminuyendo la motivación intrínseca.

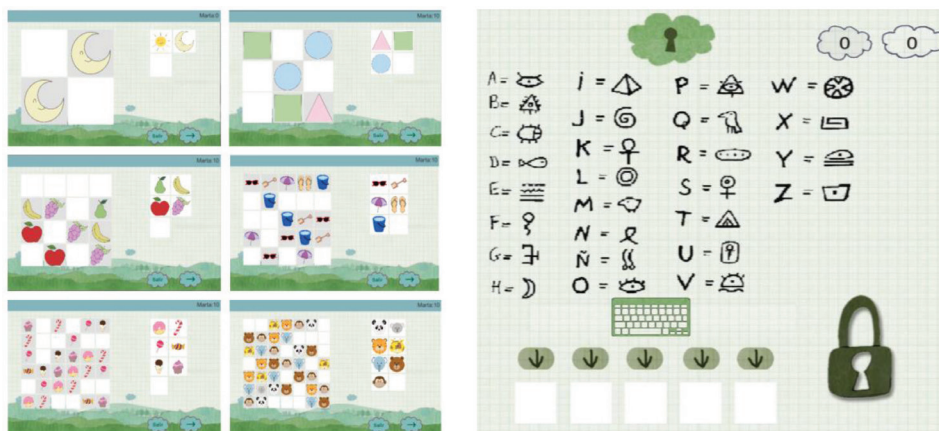
Las posibles **aplicaciones** derivadas de los resultados de la investigación se basan en la recomendación de analizar los aspectos de persistencia y recompensas en los juegos educativos. El entorno BAC podría utilizarse en contextos escolares, además de para el aprendizaje de la programación y su evaluación, para analizar aspectos relacionados con la motivación intrínseca, pero, en este caso, siempre que su uso sea totalmente voluntario. En este mismo sentido, se recomienda incorporar juegos educativos basados en la motivación intrínseca en los entornos escolares. **Los currículos escolares podrían diseñarse teniendo en cuenta que, a través de esta motivación**, los niños pueden ser capaces de superar los conceptos computacionales abordados (incluidos los bucles anidados) antes de los 5 años, existiendo diferencias de género en la adquisición de estos conceptos. También **se recomienda diseñar juegos educativos teniendo en cuenta estos aspectos, y de esta forma potenciar la conducta de persistencia-reto en los niños y, a partir de los 5 años, controlar o potenciar la conducta de persistencia-recompensa respecto a las niñas**. Creemos que los hallazgos podrían guiar el diseño de mejores herramientas para el aprendizaje de la programación y el desarrollo del PC. Dado que la edad y el género influyen en el rendimiento en estos retos, es aconsejable adaptar las herramientas de aprendizaje en consecuencia, así como el currículo escolar. Además, los entornos de aprendizaje podrían adaptar la forma de proporcionar refuerzos y recompensas, especialmente para los niños en los retos más complejos y para las niñas a partir de los 5 años. Sin embargo, se necesitan nuevos estudios que tengan en cuenta estas complejidades.

8.4. Aprendizaje del PC: líneas actuales y futuras

Actualmente se están realizando nuevos módulos de BAC para explorar el aprendizaje y evaluación de otros conceptos y habilidades de PC, además de módulos de análisis y visualización de datos con GLA, aparte de los existentes (apartado 6.4.3). Algunos de los proyectos actualmente en desarrollo son:

- Módulo de juego de sudokus adaptativos: se plantea el desarrollo de la habilidad de resolución de problemas mediante la descomposición a través de la resolución de sudokus. Para ello, se han desarrollado a nivel de prototipo jugable y validado, varios niveles de juego con sudokus de creciente dificultad (véase la Figura 74, parte izquierda) [235].
- Módulo de juego de cifrado y descifrado de códigos: también se ha realizado un prototipo jugable con dos niveles de dificultad para su futura implementación (véase la Figura 74, parte derecha) [236].
- Módulo para el análisis de datos automático en cuanto a caminos óptimos en BAC. En este caso, el módulo analizará los datos de la base de datos y se crea un **informe estadístico periódico** sobre la progresión de los jugadores en cuanto a soluciones óptimas. Este módulo se está desarrollando en C# con el IDE Visual Studio Code.

Figura 74. Pantallas de prototipos para módulos de BAC: sudokus (izquierda), cifrado y descifrado (derecha).



Algunas de las líneas futuras de investigación son:

- Ampliación del entorno de juego BAC con más conceptos computacionales.
- Inclusión y evaluación mediante experiencias de entornos de aprendizaje basados en la motivación intrínseca en contextos escolares.
- Análisis del código propuesto por los estudiantes en términos de optimización y estilos de programación utilizando GLA.

- Ampliación de BAC incluyendo los hallazgos sobre comportamientos de juego.
- Ampliación de BAC para hacer uso de GLA adaptativo sin reducir la MI.

Además, para la evaluación de nuevas metodologías de aprendizaje, se ha iniciado una colaboración con la *Universidad de Harvard* para guiar la creación de un videojuego para el aprendizaje y evaluación del CT en alumnos de educación secundaria.

En general, se pretende seguir investigando y colaborando internacionalmente para la creación de instrumentos y herramientas de aprendizaje y evaluación para el desarrollo del PC en etapas tempranas, con especial atención a la diversidad.

8.5. Publicaciones

Esta tesis ha dado lugar a publicaciones en artículos en revistas, artículos en congresos y un capítulo de libro. En este apartado, se detallan las publicaciones y acciones más relevantes.

8.5.1. Revistas JCR (Q1 y Q2)

- M. Zapata-Caceres, E. Martin, and M. Roman-Gonzalez, "Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study," in *IEEE Transactions on Learning Technologies*, vol. 14, no. 5, pp. 576-589, 1 Oct. 2021, doi: 10.1109/TLT.2021.3111108.
- M. Zapata-Cáceres and E. Martín-Barroso, "Applying Game Learning Analytics to a Voluntary Video Game: Intrinsic Motivation, Persistence, and Rewards in Learning to Program at an Early Age," in *IEEE Access*, vol. 9, pp. 123588-123602, 2021, doi: 10.1109/ACCESS.2021.3110475.
- L. El-Hamamsy, M. Zapata-Caceres, E. Martín, F. Mondada, J. Dehler Zufferey, and B. Bruno, "The competent Computational Thinking test (cCTt): Development and validation of an unplugged Computational Thinking test for upper primary school," *Journal of Educational Computing Research*, 2022. In press.

8.5.2. Congresos internacionales

- M. Zapata-Cáceres, E. Martín-Barroso, and M. Román-González, "Computational thinking test for beginners: Design and content validation," in *IEEE Global Engineering Education Conference (EDUCON'20)*, Porto, Portugal, pp. 1905–1914, 2020, doi: 10.1109/EDUCON45650.2020.9125368.
- M. Zapata-Cáceres and N. Fanchamps, "Using the beginners computational thinking test to measure development on computational concepts among preschoolers," in *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*, Singapore, National Institute of Education, 2021, pp. 3237. **Premiado como "Best full paper"**.
- L. El-Hamamsy, M. Zapata-Caceres, P. Marcelino, J. Dehler Zufferey, B. Bruno, and E. Martin, "Comparing the psychometric properties of two primary school Computational Thinking (CT) assessments for grades 3 and 4: the Beginners' CT test (BCTt) and the competent CT test (cCTt)" in *ITiCSE 2022, 27th Annual Conference on Innovation and Technology in Computer Science Education*. Under peer review.
- M. Zapata-Caceres, N. Fanchamps, I. H. Yeter, P. Marcelino, and E. Martín-Barroso, "Understanding Teachers' Attitudes and Self-Assessment Towards Computational Thinking" in *CTE-STEM 2022, Sixth APSCE International Conference on Computational Thinking and STEM Education*. In submission process.

8.5.3. Congresos nacionales

- Presentación de póster "Computational Thinking at early stages: learning assessment tolos and school curricula approach", en *II Congreso de la Escuela Internacional de Doctorado de la Universidad Rey Juan Carlos*, Noviembre 2019.
- Presentación "Videojuegos para enseñar a programar: analíticas y motivación" en *Congreso CE + Ed Educare + Educere de Formación del Profesorado*, 2021.

8.5.4. Capítulos de libro

- Zapata-Cáceres María, E. Martín-Barroso and M. Román-González, "BCTt: Beginners computational thinking test," in *Raspberry Pi Foundation Research Seminars*, Apr 2021, pp. 46-56. ISSN 2514-586X

(16). Available: <https://www.raspberrypi.org/app/uploads/2021/05/Understanding-computing-education-Volume-1-%E2%80%93-Raspberry-Pi-Foundation-Research-Seminars.pdf>.

8.5.5. Divulgación

- Presentación de la experiencia "Experiencias de enseñanza de conceptos computacionales en Educación Primaria a través de un videojuego en modos colaborativo e individual" *en las IV Jornadas de Innovación Educativa iEDU - CRIF Las Acacias*, mayo 2019.
- Presentación "Experiencia de enseñanza y evaluación del pensamiento computacional en Educación Primaria a través de un videojuego colaborativo", *en SIMO Educación*, noviembre 2019.
- Presentación y panel de discusión Q&A "Computational thinking test for beginners" *en the online seminars on computing education research from the Raspberry Pi Foundation*, Cambridge UK, julio 2020.
- Taller "Recursos Tecnológicos en Educación Infantil, Primaria y NEE", *en XIX Semana de la Ciencia y de la Innovación de Madrid, Universidad Rey Juan Carlos*, noviembre 2019.
- Taller "Aprende a programar jugando a videojuegos con amigos", *en XIX Semana de la Ciencia y de la Innovación de Madrid, Universidad Rey Juan Carlos*, noviembre 2019.
- Taller "Cómo aprender a programar jugando", *en XI Noche Europea de los Investigadores de Madrid, Universidad Rey Juan Carlos*, noviembre 2020.
- Taller "Aprende a programar jugando a videojuegos", *en XI Noche Europea de los Investigadores de Madrid, Universidad Rey Juan Carlos*, noviembre 2020.

CHAPTER 8. CONCLUSIONS AND LINES OF ACTION

As described in this thesis, the learning and assessment of skills that contribute to the development of CT in formal education is gaining importance at an international scale, and there is a consensus that this learning should begin at an early age. Due to its recent nature, there are still no validated and agreed instruments for learning and assessing CT, and it is also necessary to establish which skills are achievable and whether there are differences in the acquisition of these skills according to the age and gender of the children.

Through the systematic literature review on theoretical and operational CT, a proposal for a grounded definition and decomposition of CT was provided, as well as several hypotheses on learning and assessment systems that were further tested. One of the most important contributions in the literature was the need for a *system of assessments* [148], [161] consisting of several different measuring instruments. One of the

hypotheses of the thesis posed the possible impact of each of these assessment instruments on the dimensions of the 3D framework [46] (this being the most widely used reference framework for CT) and, through the studies carried out, the expected impact was verified (Figure 11). Moreover, this thesis proposes not only the creation of a *system of assessments*, but also the creation of a CT *learning system*, where different pedagogical learning strategies are used in combination. By developing a specific learning environment, the impact on the CT learning of each of the strategies in the dimensions of the 3D framework is also evaluated (Figure 10).

By means of the **instruments developed** in this thesis, both for the learning and assessment of CT, and the findings regarding the CT skills in early stages, extracted from the successive developments and **studies carried out in schools and non-school settings**, the main aim (OP) of the thesis has been fulfilled, as well as all the specific aims indicated in section 1.2, advancing in the knowledge of CT and impacting on its operational component. The hypotheses proposed (section 3.2) have also been confirmed throughout the

instrumental developments and studies carried out (section 3.3). As for the common aim OC, given that it corresponds to an international collaboration, we consider that a great contribution has been made to global research, and we intend to continue advancing in this direction by maintaining the collaborations and work in progress with other countries (section 8.2). Table 47 shows a summary of the structure of the thesis in terms of aims, hypotheses and studies carried out (listed below).

- **E1:** Development and validation of a stand-alone test for the assessment of CT in Primary Education: Beginners Computational Thinking test (BCTt).
- **E2:** Estimation of the validity of the BCTt in Infant Education students.
- **E3:** Validation and analysis of the educational environment based on the BAC game, in individual and collaborative modes and analysis of CT skills in Primary Education.
- **E4:** Analysis of intrinsic motivation, interests, persistence and behavior in response to rewards, in learning programming in early stages using the BAC environment.

Tabla 47. Summary of aims, case studies and hypothesis.

Objetivos		Comp. degree.	Research	Case study	Hypothesis	Chapter
OP	To advance knowledge about CT, especially in its operational component, i.e., in terms of effective learning and assessment strategies, especially at early stages.	High	Global	-	-	All
	Compile a broad theoretical framework and state of the art of publications and studies on CT.	High	Theoretical			2
	OP1a Theoretical and operational definitions of CT	High				2
	OP1b Frameworks, school curricula and initiatives for the development of CT at international and national level at early stages.	High				2
	OP1c Theoretical framework and strategies for CT development/learning, as well as existing tools and environments.	High				2
OP2	OP1d Theoretical framework and strategies for the assessment of CT, as well as existing instruments.	High				2
	Establish an operational proposal of CT on the basis of OP1 and thus constitute the direction of the empirical research by means of hypotheses and base aims.	High	Operational proposal			3

Objetivos		Comp. degree.	Research	Case study	Hypothesis	Chapter
OP3	To design and validate, empirically, tools for learning and assessment of CT.	High	Empirical	E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
	OP3a To design and validate a CT assessment tool aimed at early ages.	High		E1, E2	H1	4, 5
	OP3b To develop and validate a tool for learning and assessment of CT at an early age.	High		E3, E4	H2, H3, H4, H5	6, 7
OP4	To carry out empirical studies to analyse data related to CT skills as well as CT dispositions and predispositions.	High		E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
	OP4a To collect and analyse data to improve the design of the school curricula related to CT according to age and gender.	High		E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
	OP4b To collect and analyse data to improve the design of learning tools related to CT according to age and gender.	High		E3, E4	H2, H3, H4, H5	6, 7
OC	OP4c To collect and analyse data to improve the design of CT assessment tools according to age and gender.	High		E1, E2, E3, E4	H1, H2, H3, H4, H5	4, 5, 6, 7
	Design and validation of CT assessment instruments covering a wide range of ages and populations.	Medium		E1, E2	H1	4, 5

Comp. degree.: Compliance degree of the specific aim.

The following sections summarise the main contributions, conclusions, and findings in the context of the thesis.

8.6. Beginners Computational Thinking Test (BCTt)

The main obstacle encountered at the beginning of the research was the non-existence, at that time, of an autonomous instrument (independent of a specific learning environment) for the assessment of CT in Primary Education, which made it very difficult to evaluate or validate new learning and non-autonomous assessment instruments, with no possibility of carrying out pre- and post-test type studies to assess the suitability of new environments and tools for the development of CT.

This lack of instruments made it very difficult to aseptically recognise skills at each age, as there could be biases due to the fact that certain skills could be more or less achievable due to the characteristics of the particular environment used. For this reason, the research framed in E1 (chapter 4) was carried out, in which an autonomous instrument, **the Beginners Computational Thinking Test (BCTt)** [7], [8], was developed and validated at the Primary Education stage, in three different schools. The data obtained showed a high reliability with a Cronbach's alpha = 0.824 across the entire primary education stage, although the final recommendation is that the BCTt is suitable mainly for the 1st and 2nd years of primary school. Since a decrease in test reliability was observed in older students, **the competence Computational Thinking test (cCTt)** [165] was developed in collaboration with a research team from the Ecole Polytechnique Fédérale de Lausanne (EPFL) in Switzerland, and validated in several schools in Switzerland. Furthermore, age limits for the administration of the BCTt and the cCTt are established through a study administering both tests to primary school students in several schools in Portugal. The results of the psychometric analysis indicate that the cCTt should be preferred for 3rd and 4th grade students. However, the BCTt can be used if the aim is to discriminate between low-ability students in these two grades.

On the other hand, in order to establish the lower age limit at which BCTt can be administered, and through international collaboration with Fontys University of Applied Science and Zuyd University of Applied Science, both in the Netherlands, the E2 case study (chapter 5) was carried out using robotic learning environments in Early Childhood Education, with children aged 4 and 5 years old, in a school in the Netherlands [6]. Through this study, the suitability of these learning environments for the development of CT is determined. At the same time, the reliability of the BCTt with 4-year-olds is confirmed (Cronbach's alpha = 0.802). However, four of the six computational concepts addressed in the test seem not to be achievable for 4-year-olds, as they were not able to answer the related questions and did not show any improvement

in the post-test with respect to these concepts. Therefore, it is necessary to design another version of the BCTt for students aged 4 years and younger. This version is currently being developed in collaboration with a team from the Open University Netherlands (OU) in the Netherlands.

The BCTt and the cCTt are stand-alone instruments, independent of any environment, which have proven reliable for the assessment of CT in Primary Education and can be administered as a pre-test and post-test in research that require it. These tests can be considered a reliable extension of the CTt by Román *et al.* for younger students, and are part of the common international aim of developing stand-alone CT assessment instruments for all age ranges and populations. Table 48 shows the instruments applicable to each age range in the context of the OC aim.

Table 48. Table of instruments in line with the common aim OC.

Development framework for the OC	Current, collaboration Netherlands	Thesis	Thesis, collaboration Switzerland	UNED ⁹ research team	Swiss research team
Stand-alone instruments	In progress	BCTt	cCTt	CTt	ATTA
Edades diana	4 and below	4 to 7	7 to 9	10 to 16	16 +
Publicaciones	En progreso	[6]-[8]	[165]	[44], [142], [143], [161]	[233]

Table 49 shows the achievable CT skills and/or possible level of acquisition, preliminarily derived from the simple administration of the test, in the E1 and E2 case studies.

⁹ <https://www.uned.es/>

Table 49. Achievable CT skills and/or level of acquisition possible in the first instance.

Case	Stage	Substage	Age (years)	Sequence	Simple loop	Nested loop	If-then	If-then-else	While
E2	Infant	3	4	Achievable	Achievable	Not achievable	Not achievable	Not achievable	Not achievable
		3	5	Achievable	Achievable	Achievable	Achievable	Achievable	Achievable
E1	Primary	1	5 a 8	High	High-Medium	Medium	Medium-Low	Low	Medium-Low
		2	7 a 10	High	High	High	Medium	Medium	Medium
		3	9 a 12	High	High	High	High	Medium	Medium

In addition, the BCTt has been **translated into numerous languages, several studies have been carried out and are currently being carried out in different countries** (such as Portugal, Belgium, or Singapore) and future lines of action have been proposed, which are in different stages of development, these actions are also described in more detail in the following section.

8.7. CT assessment: current and future directions

The BCTt [7], [8] has had a significant international impact. The first collaborations with Switzerland, the Netherlands and Portugal have resulted in the development and validation of the **competence computational Thinking test (cCTt)** together with the *Ecole Polytechnique Fédérale de Lausanne (EPFL)*: *MOBOTS Group of the BIOROB Laboratory*; and *LEARN - Center for Learning Sciences*; and *Computer Human Interaction in Learning and Instruction (CHILI)* [165].

In addition, a collaboration was established between the *Ecole Polytechnique Fédérale de Lausanne (EPFL)* and the NGO *Tree Tree 2*¹⁰ for a **comparative study of BCTt and cCTt** in schools in Portugal. This study is currently under review for publication at the ITiCSE 2022 (27th Annual Conference on Innovation and Technology in Computer Science Education) (section 8.5).

Furthermore, through collaboration with *Fontys University of Applied Science* and *Zuyd University of Applied Science*, both in the The Netherlands, it was possible to establish the lower age limit for the administration of the BCTt [6].

Other collaborations with other countries and populations are currently underway regarding the OC aim: Design and validation of CT assessment instruments covering a wide range of ages and populations, the most relevant of which are listed below.

Currently, other collaborations are being carried out with other countries and populations with respect to the CT: Design and validation of CT assessment instruments covering a wide range of ages and populations, the most relevant of which are listed below:

- Conducting a **case study on the possibility of developing CT in Primary Education through Bebras activities** [161], through a study in schools in Portugal, using the BCTt as a pre- and post-test tool. This study is currently in the drafting stage of a scientific paper.

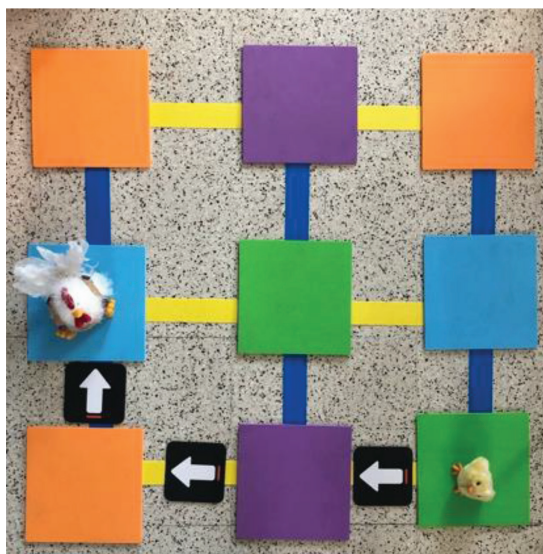
¹⁰ <https://www.treetree2.org/>

- Development of a **version of the test for children under 4 years** of age in collaboration with the *Open University* (OU) in The Netherlands, including sensory aspects other than visual, such as sound. This project is in its initial phase and includes other sensory elements such as touch and hearing in the assessment instrument.
- Validation of the **BCTt limits and study of CT skills in Primary Education**: collaboration with the **UNED and INTEF** (Ministry of Education) in the administration of the BCTt to 2500 students between infant (5 years old) and 3rd grade of primary school. Preliminary studies indicate very good reliability (Cronbach's $\alpha = 0.87$). This study is in the drafting phase.
- **Validation of the BCTt in other populations**, in collaboration with the University Institute of Lisbon (ISCTE), in which the BCTt was administered in schools in Portugal to +600 pupils. These results are currently being analysed.
- Integrating the concepts and ideas of CT education into the practices of primary school teachers is a difficult challenge. Teachers need to acquire new knowledge and skills related to how content, pedagogical strategies and ICT tools should be combined to introduce CT into their educational practices in a meaningful way [38]. In this context, a study of **teachers' perceptions of CT about the school curricula** is proposed. This research is the result of a collaboration between 4 countries: **Spain, The Netherlands, Portugal, and Singapore**, and is currently in the data collection phase. The study is expected to be presented at the CTE-STEM 2022 conference (Sixth APSCE-International Conference on Computational Thinking and STEM Education) with the title: "Understanding Teachers' Attitudes and Self-Assessment Towards Computational Thinking".
- **Creación de un equipo internacional** (España, Países Bajos, Bélgica, Portugal, Suiza), a través de una convocatoria **Erasmus +** para la inclusión del PC en el currículo de forma estandarizada, desarrollo y validación de instrumentos para su evaluación y atención especial a la diversidad, ya que la integración del PC en la educación formal puede ser una oportunidad para redefinir el currículo escolar en la educación informática, de forma hacia una mayor inclusión y diversidad [34]. Este proyecto se encuentra en fase de propuesta. Este proyecto se enmarca especialmente en el objetivo común de la tesis OC.
- Foundation of an **international team** (Spain, Netherlands, Belgium, Portugal, Switzerland), through an **Erasmus +** call for the inclusion of CT in the curriculum in a standardised way, development, and validation of instruments for its assessment and special attention to diversity, since the integration of CT in formal education can be an opportunity to redefine

the school curricula in computer education, towards greater inclusion and diversity [34]. This project is at the proposal stage. This project is particularly framed by the common aim of the OC thesis.

In addition, there are some independent initiatives and studies using the BCTt in many countries, e.g., in Belgium, the *Service d'Ingénierie Pédagogique et du Numérique éducatif*, Umons-FPSE University has developed an unplugged version of the BCTt (see Figure 73) for 3- and 4-year-olds: **Beginners Computational Thinking Test: Adaptation pour un public de moins de 5 ans** [234], which has proved very reliable for the assessment of CT.

Figure 73. Unplugged version of the BCTt [234].



Large-scale research is also being carried out, for example, in schools in Italy using the BCTt, through a project funded by the **Italian Ministry of Education**, Universities and Research. In Switzerland, the *Ecole Polytechnique Fédérale de Lausanne (EPFL)* included the BCTt on the *Roteco*¹¹ website, which brings together the following institutions: Swiss Academies of Sciences, SUPSI, the EPFL, the ETHZ and schools St. Gallen, Lucerne, Schwyz, and Valais. There is also an initial collaboration with professors and researchers from **China and the Ministry of Education of Singapore**, as well as initial collaborations with Cornell University (USA), University of Hradec Kralove (Czech Republic),

¹¹ <https://www.roteco.ch/fr/nouveautes/comment-evaluer-lefficacite-de-la-pensee-computationnelle-chez-les-eleves-du-primaire>

University of Tübingen (Germany), Vilnius University (Lithuania), and: Université Lumière Lyon 2 (ISPEF) (France), among others.

In addition, for this multi-population research, the BCTt has been translated into **several languages**, e.g., Italian, Dutch, Portuguese, German, French, Chinese, Farsi and other languages.

8.8. Blue Ant Code (BAC)

In the context of the thesis, a **learning and assessment environment** through learning analytics, has been developed. The development of the BAC environment includes a video game compatible with multiple platforms, a REST api for the connection with the relational database, and complementary modules for filtering, visualisation and data analysis with GLA. This environment was used in the case study *E3: Validation and analysis of the educational environment based on the BAC game, in individual and collaborative modes and analysis of CT skills in Primary Education* (see chapter 6). In addition, a **second version of BAC was developed adapted to a non-school setting**, including new features such as the inclusion of a rewards system, which was used in *E4: Analysis of competences: intrinsic motivation, persistence and rewards through learning analytics with BAC* (see chapter 7).

Research on how to teach CT and especially on the computational practices and perspectives dimensions of the 3D framework is scarce, so in this thesis we propose a *learning system* combining three of the most used learning strategies in the BAC environment, this being a collaborative game-based environment built on a constructivist problem-based strategy, which has been validated at the Primary Education stage in the E3 case study. The results of the research show that this approach is suitable for CT learning, especially at early ages, covering the three key dimensions of the 3D framework (see Figure 49), as posed in hypothesis HE2, and confirmed by the case study. The combination of different assessment methodologies, such as a *system of assessments* [148]: BCTt, GLA through GBL, and qualitative data (direct observation and questionnaires), allows for a three-dimensional assessment, as hypothesised in HE4, and confirmed by this case study (see Figure 50)

Regarding the **validation of BAC in Primary Education**, the results of the E3 case study confirmed BAC as an **appropriate learning environment**, with very significant improvements in CT development even with little interaction time with the environment. Moreover, it was also **suitable as an assessment tool** with results consistent with the analysis of the data collected by BAC and, in addition, it can provide real-time data that can be useful for detailed personal assessment, strong analysis of low-level data, and the identification of age-specific unachievable computational concepts not only at a specific

moment in time, but throughout and at the end of a specific learning process. However, as was the case with BCTt, **BAC seems to be better adapted to younger learners and even suitable for lower grades** (Early Childhood Education), especially in collaborative mode. Moreover, BAC is especially suitable and motivating for students with **special needs or low percentile** especially in the collaborative play mode

Positive results were also obtained in the learning retention test, suggesting that this **game-based methodology promotes long-lasting learning**. Learning retention results were better in the collaborative mode than in the individual mode, especially in the lower grades, leading to the conclusion that the collaborative game mode promotes more effective and long-lasting learning. However, it is possible that the **collaborative strategy may not be as effective for upper primary school students**, as these students seem to be capable of more effective individual reflection. Further research is needed to test this hypothesis.

One of the most relevant findings is that students in the first years of Primary Education outperform the CT skills of students in higher grades at the end of the study, thus highlighting in this thesis the **importance of developing CT from a very early age and the recommendation to incorporate certain concepts, such as nested loops or conditionals, in the school curricula already in the first year or even earlier**, instead of waiting until the last year of Primary (or even secondary) Education, as it is currently the case in most schools in Spain.

The case study *E4*, **mass deployment of BAC in a non-school setting**, was carried out with the general aim of analysing strategies for the learning and assessment of CT, and identifying CT skills of Primary School students, with a special focus on dispositions and predispositions for CT based on intrinsic motivation and voluntary play. For this study, the adapted BAC mass deployment environment was used. The main findings of this study are summarised below:

- **Initial interest:** A priori there seems to be a greater interest in learning to program in boys than in girls, since boys play more often and spend more time playing until the age of 5; however, from the age of 6 onwards, the time spent playing and the number of games is much greater in girls than in boys.
- **Performance:** At all ages is significantly higher in girls than in boys, especially from 3 to 5 years old. However, from 6 to 8 years of age, although their performance is still better, girls spend much more time playing and, according to the results of the research carried out, this behaviour is not related to the mastery of the concepts addressed but is due to a persistence-reward behaviour.

- **Specific computational problem solving:** Girls have a higher interest and performance in the more difficult levels (nested loops and long sequences), compared to boys, who have a similar interest in each concept, but higher performance only in simple, medium-difficulty loops. Simple sequences seem to be mastered from the age of 3 years onwards, regardless of gender. However, due to working memory, long sequences are not mastered until the age of 5 years, regardless of gender. Simple loops are mastered around the age of 3 years for boys and 4 years for girls. On the other hand, nested loops seem to be mastered from the age of 5 years for girls and 6 years for boys. Finally, as far as conditionals are concerned, performance increases at the age of 4, regardless of gender. **All the concepts addressed can be mastered between the ages of 3 and 6, which could be an aspect to be considered in the design of school curricula.**
- **Learning progression:** Both girls and boys make learning progress, but boys' improvement is more significant in both time and performance. However, girls achieve significantly more than boys and reach full mastery at all levels in the first 10 attempts between the ages of 3 and 5, while this behaviour is not observed for boys. From the age of 6 years onwards, girls take more time and attempts to solve the challenges due to persistence-reward behaviour. It is noteworthy that, **although nested loops seem to be difficult for primary school pupils, it is possible to fully master this concept by the age of 4, regardless of gender**, if enough attempts are made and, in addition, children can **persist with the challenge voluntarily, intrinsically motivated, until they master it. This result contrasts with that obtained in the E2 case study, in which many of these concepts were not attainable at age 4, thus confirming the importance of intrinsic motivation in learning processes.** This could also be an aspect to consider for the development of school curricula, since **all the concepts addressed (sequences, simple and nested loops, and conditionals) could be mastered at the age of 4 if there is intrinsically motivated persistence.**
- **Persistence-challenge and persistence-reward behaviours:** Persistence-challenge (repetition despite losing) is clearly observed in both genders, but is more pronounced in boys, which may explain why their progress is highly correlated with persistence. In contrast, persistence-reward behaviour (repetition after winning) is observed significantly more in girls, especially between the ages of 5 and 7, when they have mastered all the concepts addressed. This is a relevant finding of our research because, although learning derived from this behaviour is possible, this learning could be reduced as the aim is not to overcome the challenge but to obtain rewards. Given that children are able to master the concepts addressed before the age of 5 through continued practice, we wonder whether it is not **counterproductive to delay the acquisition of these concepts until after the age of 5, when persistence-challenge behaviour is lower and**

persistence-reward behaviour emerges, especially in girls. On the other hand, **from the age of 5 onwards, motivation through rewards could also be suggested, especially in girls,** but always bearing in mind not to exceed the limit at which rewards can be detrimental to learning, diminishing intrinsic motivation.

The potential **applications** derived from the research results are based on the recommendation to analyse aspects of persistence and rewards in educational games. The BAC environment could be used in school contexts, to develop and assess CT, as well as to analyse aspects related to intrinsic motivation, but, in this case, provided that its use is completely voluntary. In the same direction, it is recommended to incorporate educational games based on intrinsic motivation in school settings. **School curricula could be designed considering that, through this motivation,** children may be able to overcome computational concepts (including nested loops) before the age of 5, there being gender differences in the acquisition of these concepts. It is also recommended to **design educational games with these aspects in mind, and in this way to enhance persistence-challenge behaviour in boys and, from the age of 5, to control or enhance persistence-reward behaviour with respect to girls.** We believe that the findings could guide the design of better tools for learning programming and CT development. Since age and gender impact performance on these challenges, it is advisable to adapt learning tools accordingly, as well as the school curricula. In addition, learning environments could adapt the way reinforcement and rewards are provided, especially for boys in the more complex challenges and for girls from the age of 5. However, further studies are needed to take these complexities into account.

8.9. CT learning: current and future directions

New BAC modules are currently being developed to explore the learning and assessment of other CT concepts and skills, as well as data analysis and visualisation modules with GLA, in addition to the existing ones (section 6.4.3). Some of the projects currently under development are the following:

- Adaptive Sudoku game module: the aim is to develop problem-solving skills through decomposition by solving Sudokus. To this end, several levels of Sudoku games of increasing difficulty have been developed as a playable and validated prototype (see Figure 74, left-hand side) [235].
- Code encryption and decryption game module: a playable prototype with two levels of difficulty has also been developed for future implementation (see Figure 74, right-hand side) [236].

- Module for automatic data analysis in terms of optimal paths in BAC, player progression and statistical evaluation This module is being developed in C# with the Visual Studio Code IDE.

Figure 74. Prototype screenshots for modules: Sudoku (left), encryption and decryption (right).



Some of the future research lines are:

- Extending the BAC game environment with more computational concepts.
- Integration and evaluation of learning environments based on intrinsic motivation in school contexts, through new case studies
- Analysis of the code proposed by students in terms of optimality and programming styles using GLA.
- Extension of BAC to include findings on gaming behaviours.
- Extension of BAC to make use of adaptive GLA without reducing intrinsic motivation.

In addition, for the evaluation of new learning methodologies, a collaboration has been initiated with Harvard University to guide the creation of a video game for learning and assessing CT in secondary school students.

In general, the aim is to continue research and international collaboration for the creation of learning and assessment instruments and tools for the development of CT in early stages, with special attention to diversity.

8.10. Publications

This thesis has resulted in publications in journal papers, conference papers and a book chapter. In this section, the most relevant publications and actions are detailed below.

8.10.1. JCR journals (Q1 y Q2)

- M. Zapata-Caceres, E. Martin, and M. Roman-Gonzalez, "Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study," *IEEE Transactions on Learning Technologies*, 2021.
- M. Zapata-Cáceres and E. Martín-Barroso, "Applying Game Learning Analytics to a Voluntary Video Game: Intrinsic Motivation, Persistence, and Rewards in Learning to Program at an Early Age," *IEEE Access*, vol. 9, pp. 123588-123602, 2021.
- L. El-Hamamsy, M. Zapata-Caceres, E. Martín, F. Mondada, J. Dehler Zufferey, and B. Bruno, "The competent Computational Thinking test (cCTt): Development and validation of an unplugged Computational Thinking test for upper primary school," *Journal of Educational Computing Research*, 2022. In press.

8.10.2. International congresses

- M. Zapata-Cáceres, E. Martín-Barroso, and M. Román-González, "Computational thinking test for beginners: Design and content validation," in *IEEE Global Engineering Education Conference (EDUCON'20)*, Porto, Portugal, pp. 1905–1914, 2020, doi: 10.1109/EDUCON45650.2020.9125368.
- M. Zapata-Cáceres and N. Fanchamps, "Using the beginners computational thinking test to measure development on computational concepts among preschoolers," in *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*. Singapore: National Institute of Education, 2021, pp. 3237. **Awarded as "Best full paper"**.
- L. El-Hamamsy, M. Zapata-Caceres, P. Marcelino, J. Dehler Zufferey, B. Bruno, and E. Martin, "Comparing the psychometric properties of two primary school Computational Thinking (CT) assessments for grades 3 and 4: the Beginners' CT test (BCTt) and the competent CT test (cCTt)" in *ITiCSE 2022, 27th Annual Conference on Innovation and Technology in Computer Science Education*. Under peer review.

- M. Zapata-Caceres, N. Fanchamps, I. H. Yeter, P. Marcelino, and E. Martín-Barroso, "Understanding Teachers' Attitudes and Self-Assessment Towards Computational Thinking" in *CTE-STEM 2022, Sixth APSCE International Conference on Computational Thinking and STEM Education*. Work in progress.

8.10.3. National congresses

- Poster presentation "Computational Thinking at early stages: learning assessment tolos and school curricula approach", in *II Congreso de la Escuela Internacional de Doctorado de la Universidad Rey Juan Carlos*, Nov. 2019.
- Presentation "Videojuegos para enseñar a programar: analíticas y motivación" in *Congreso CE + Ed Educare + Educere de Formación del Profesorado*, 2021.

8.10.4. Book chapters

- Zapata-Cáceres María, E. Martín-Barroso and M. Román-González, "BCTt: Beginners computational thinking test," in *Raspberry Pi Foundation Research Seminars*, Apr 2021, pp. 46-56. ISSN 2514-586X (16). Available: <https://www.raspberrypi.org/app/uploads/2021/05/Understanding-computing-education-Volume-1-%E2%80%93-Raspberry-Pi-Foundation-Research-Seminars.pdf>.

8.10.5. Dissemination

- Presentation "Experiencias de enseñanza de conceptos computacionales en Educación Primaria a través de un videojuego en modos colaborativo e individual" in *las IV Jornadas de Innovación Educativa iEDU - CRIF Las Acacias*, May 2019.
- Presentation "Experiencia de enseñanza y evaluación del pensamiento computacional en Educación Primaria a través de un videojuego colaborativo", in *SIMO Educación*, Nov. 2019.
- Presentation y discussion panel Q&A "Computational thinking test for beginners" in *the online seminars on computing education research from the Raspberry Pi Foundation*, Cambridge UK, Julio 2020.

- Workshop "Recursos Tecnológicos en Educación Infantil, Primaria y NEE", in *XIX Semana de la Ciencia y de la Innovación de Madrid, Universidad Rey Juan Carlos*, Nov. 2019.
- Workshop "Aprende a programar jugando a videojuegos con amigos", in *XIX Semana de la Ciencia y de la Innovación de Madrid, Universidad Rey Juan Carlos*, Nov. 2019.
- Workshop "Cómo aprender a programar jugando", in *XI Noche Europea de los Investigadores de Madrid, Universidad Rey Juan Carlos*, Nov. 2020.
- Workshop "Aprende a programar jugando a videojuegos", in *XI Noche Europea de los Investigadores de Madrid, Universidad Rey Juan Carlos*, Nov. 2020.

BIBLIOGRAFÍA

- [1] V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying computational thinking," *Educational Research Review*, vol. 22, pp. 142-158, 2017. Available: <https://www.sciencedirect.com/science/article/pii/S1747938X17300350>, doi: 10.1016/j.edurev.2017.09.003.
- [2] M. Berland and U. Wilensky, "Comparing virtual and physical robotics environments for supporting complex systems and computational thinking," *Journal of Science Education and Technology*, vol. 24, no. 5, pp. 628-647, 2015.
- [3] J. M. Wing, "Computational thinking and thinking about computing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1881, pp. 3717-3725, 2008. Available: <http://rsta.royalsocietypublishing.org/content/366/1881/3717.abstract>, doi: 10.1098/rsta.2008.0118.
- [4] T. Hsu, S. Chang, and Y. Hung, "How to learn and how to teach computational thinking: Suggestions based on a review of the literature," *Computers & Education*, vol. 126, pp. 296-310, 2018. Available: <http://www.sciencedirect.com/science/article/pii/S0360131518301799>, doi: //doi.org/10.1016/j.compedu.2018.07.004.
- [5] S. Grover and R. Pea, "Computational Thinking in K—12: A Review of the State of the Field," *Educational Researcher*, vol. 42, no. 1, pp. 38-43, 2013. Available: <https://www.jstor.org/stable/23360476>, doi: 10.3102/0013189X12463051.
- [6] M. Zapata-Cáceres and N. Fanchamps, "Using the beginners computational thinking test to measure development on computational concepts among preschoolers," in *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*, Singapore, National Institute of Education, 2021, pp. 32-37.
- [7] Zapata-Cáceres María, E. Martín-Barroso and M. Román-González, "BCtT: Beginners computational thinking test," in *Raspberry Pi Foundation Research Seminars*, Apr 2021, pp. 46-56. ISSN 2514-586X (16). Available: <https://www.raspberrypi.org/app/uploads/2021/05/Understanding-computing-education-Volume-1-%E2%80%93-Raspberry-Pi-Foundation-Research-Seminars.pdf>.
- [8] M. Zapata-Cáceres, E. Martín-Barroso, and M. Román-González, "Computational thinking test for beginners: Design and content validation," in *IEEE Global Engineering Education Conference*

- (*EDUCON'20*), Porto, Portugal, pp. 1905–1914, 2020, doi: 10.1109/EDUCON45650.2020.9125368.
- [9] J. Kirriemuir and A. Mcfarlane, *Literature Review in Games and Learning*. Bristol: Futurelab, 2004. Available: <https://telearn.archives-ouvertes.fr/hal-00190453>.
- [10] M. Prensky, "Digital game-based learning," *Computers in Entertainment (CIE)*, vol. 1, no. 1, pp. 21, 2003.
- [11] M. Qian and K. R. Clark, "Game-based Learning and 21st century skills: A review of recent research," *Computers in Human Behavior*, vol. 63, pp. 50–58, 2016. Available: <https://www.sciencedirect.com/science/article/pii/S0747563216303491>, doi: 10.1016/j.chb.2016.05.023.
- [12] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA, USA: Harvard University Press, 1978.
- [13] J. Piaget, *Psychology and Epistemology: Towards a Theory of Knowledge*. CA, USA: Allen Lane, 1972.
- [14] D. H. Jonassen and L. Rohrer-Murphy, "Activity theory as a framework for designing constructivist learning environments," *Educational Technology Research & Development*, vol. 47, no. 1, pp. 61–79, 1999. Available: <https://www.jstor.org/stable/30220230>, doi: 10.1007/BF02299477.
- [15] D. H. Jonassen, *Learning to Solve Problems with Technology: A Constructivist Perspective*. Upper Saddle River, N.J.: Merrill Prentice Hall, 2003.
- [16] C. Alonso-Fernandez *et al*, "Systematizing game learning analytics for serious games," in *2017 IEEE global engineering education conference (EDUCON)*, Apr 2017, pp. 1111–1118.
- [17] M. Freire *et al*, "Game Learning Analytics: Learning Analytics for Serious Games," *Springer Nature Switzerland AG*, pp. 1–29, Dec 1, 2016.
- [18] M. Zapata-Caceres, E. Martin, and M. Roman-Gonzalez, "Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study," in *IEEE Transactions on Learning Technologies*, vol. 14, no. 5, pp. 576–589, 1 Oct. 2021, doi: 10.1109/TLT.2021.3111108.
- [19] M. Zapata-Cáceres and E. Martín-Barroso, "Applying Game Learning Analytics to a Voluntary Video Game: Intrinsic Motivation, Persistence, and Rewards in Learning to Program at an Early Age," in *IEEE Access*, vol. 9, pp. 123588–123602, 2021, doi: 10.1109/ACCESS.2021.3110475.

- [20] M. Binkley *et al*, "Defining twenty-first century skills," in *Assessment and Teaching of 21st Century Skill*. Dordrecht: Springer Netherlands, 2012, pp. 17-66.
- [21] B. Trilling and C. Fadel, *21st Century Skills: Learning for life in our times*. San Francisco: John Wiley & Sons, 2009.
- [22] H. Kim *et al*, "Enhancing teachers' ICT capacity for the 21st century learning environment: Three cases of teacher education in Korea," *Australasian Journal of Educational Technology*, vol. 28, no. 6, 2012.
- [23] K. Tang, T. Chou, and C. Tsai, "A Content Analysis of Computational Thinking Research: An International Publication Trends and Research Typology," *The Asia-Pacific Education Researcher*, pp. 1-11, 2019. Available: <https://search.proquest.com/docview/2195733480>, doi: 10.1007/s40299-019- 00442-8.
- [24] S. Grover *et al*, "A Framework for Using Hypothesis-Driven Approaches to Support Data-Driven Learning Analytics in Measuring Computational Thinking in Block-Based Programming Environments," *ACM Transactions on Computing Education (TOCE)*, vol. 17, no. 3, pp. 1-25, 2017. Available: <http://dl.acm.org/citation.cfm?id=3105910>, doi: 10.1145/3105910.
- [25] S. Grover and A. Korhonen, "Unlocking the Potential of Learning Analytics in Computing Education," *ACM Transactions on Computing Education (TOCE)*, vol. 17, no. 3, pp. 1-4, 2017. Available: <http://dl.acm.org/citation.cfm?id=3122773>, doi: 10.1145/3122773.
- [26] S. Grover. "The 5th 'C' of 21st Century skills? Try computational thinking (not coding)", *edsurge.com*. <http://bit.ly/346qK78> (accessed Feb. 1, 2018). Available: <http://scholar.aci.info/view/1544f0f08b300170003/161ce93d34000012ef80942>.
- [27] J. A. Velazquez-Iturbide, "Towards an analysis of computational thinking," in Sep 2018, pp. 1-6.
- [28] J.A. Velázquez-Iturbide, "Towards an analysis of computational thinking," in *Int. Symp. on Computers in Education (SIIE'18)*, pp. 1-6, Sep. 2018.
- [29] M. Israel *et al*, "Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis," *Computers & Education*, vol. 82, pp. 263-279, 2015. Available: <https://www.sciencedirect.com/science/article/pii/S0360131514002759>, doi: 10.1016/j.compedu.2014.11.022.
- [30] C. Angevine *et al*, "Computational Thinking for a Computational World," 2017. Available: www.digitalpromise.org.

- [31] National Research Council, *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*. National Academies Press, 2011.
- [32] Q. Burke, W. I. O'Byrne and Y. B. Kafai, "Computational participation: Understanding coding as an extension of literacy instruction," *Journal of Adolescent & Adult Literacy*, vol. 59, no. 4, pp. 371-375, 2016.
- [33] J. F. Sanford and J. T. Naidu, "Computational thinking concepts for grade school," *Contemporary Issues in Education Research (CIER)*, vol. 9, no. 1, pp. 23-32, 2016.
- [34] K. Mills *et al*, *Computational Thinking for an Inclusive World: A Resource for Educators to Learn and Lead*. Digital Promise, 2021. Available: <https://doi.org/10.51388/20.500.12265/140>.
- [35] C. Sneider *et al*, "Computational thinking in high school science classrooms," *The Science Teacher*, vol. 81, no. 5, pp. 53, 2014.
- [36] N. Kelly and J. S. Gero, "Design thinking and computational thinking: a dual process model for addressing design problems," *Design Science*, vol. 7, 2021.
- [37] T. C. Pargman *et al*, "Teaching Computational Thinking in K-9: Tensions at the Intersection of Technology and Pedagogical Knowledge," in *14th International Conference of the Learning Sciences (ICLS) 2020*, vol. 4, pp. 2409-2410, International Society of the Learning Sciences, 2020.
- [38] D. Kohen-Vacs and M. Milrad, "Computational thinking education for in-service elementary swedish teachers: Their perceptions and implications for competence development," in *International Conference on Computational Thinking Education, 13-15 June 2019, Hong Kong*, 2019.
- [39] V. Allan *et al*, "Computational thinking in high school courses," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 2010.
- [40] P. Sengupta *et al*, "Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework," *Education and Information Technologies*, 2013, vol. 18, no 2, p. 351-380. Available: <https://search.proquest.com/docview/1323979287>, doi: 10.1007/s10639-012- 9240-x.
- [41] V. Barr and C. Stephenson, "Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community?" *ACM Inroads*, vol. 2, no. 1, pp. 48-54, 2011.
- [42] P. Mishra, A. Yadav, and Deep-Play Research Group, "Rethinking technology & creativity in the 21st century," *TechTrends*, vol. 57, no. 3, pp. 10-14, 2013.

- [43] J. M. Wing, "Computational Thinking Test," *CACM Viewpoint*, pp. 33-35, 2006. Available: <http://www.cs.cmu.edu/~wing/>.
- [44] M. Román-González, J. Pérez-González and C. Jiménez-Fernández, "Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test," *Computers in Human Behavior*, vol. 72, pp. 678-691, 2017. Available: <https://www.sciencedirect.com/science/article/pii/S0747563216306185>, doi: 10.1016/j.chb.2016.08.047.
- [45] A. V. Aho, "Computation and computational thinking," *The Computer Journal*, vol. 55, no. 7, pp. 832-835, 2012.
- [46] K. Brennan, M. Resnick, and MIT Media Lab, "New frameworks for studying and assessing the development of computational thinking," *American Educational Research Association Meeting, Vancouver, BC, Canada.*, 2012.
- [47] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., 1980.
- [48] J. J. Lu and G. H. Fletcher, "Thinking about computational thinking," in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 2009.
- [49] J. Wing, "Research notebook: Computational thinking—What and why," *The Link Magazine*, vol. 6, pp. 20-23, 2011.
- [50] S. Furber, "Shut down or restart? The way forward for computing in UK schools," *The Royal Society*, 2012. Available: <https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>.
- [51] M. Tedre and P. J. Denning, "The long quest for computational thinking," in *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, pp. 120-129, 2016.
- [52] J. Moreno *et al*, " No es lo mismo: un análisis de red de texto sobre definiciones de pensamiento computacional para estudiar su relación con la programación informática," *Revista Interuniversitaria De Investigación En Tecnología Educativa*, 2019, doi: 10.6018/riite.397151.
- [53] ISTE and CSTA, "Operational definition of computational thinking for K-12 education," 2011. Available: https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf.
- [54] M. U. Bers *et al*, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," *Computers & Education*, vol. 72, pp. 145-157, 2014. Available: <https://search.datacite.org/works/10.1016/j.compedu.2013.10.020>, doi: 10.1016/j.compedu.2013.10.020.

- [55] S. Atmatzidou and S. Demetriadis, "Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences," *Robotics and Autonomous Systems*, vol. 75, pp. 661-670, 2016. Available: <https://www.sciencedirect.com/science/article/pii/S0921889015002420>, doi: 10.1016/j.robot.2015.10.008.
- [56] N. D. Anderson, "A call for computational thinking in undergraduate psychology," *Psychology Learning & Teaching*, vol. 15, no. 3, pp. 226-234, 2016.
- [57] S. Rose, J. Habgood and T. Jay, "An exploration of the role of visual programming tools in the development of young children's computational thinking," *Electronic journal of e-learning*, 2017, vol. 15, no 4, p. 297-309, Aug 1, 2017.
- [58] D. Weintrop and U. Wilensky, "Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs," in *ICER*, pp. 101-110, Aug 9, 2015.
- [59] D. Statter and M. Armoni, "Teaching abstract thinking in introduction to computer science for 7th graders," in *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*, pp. 80-83, 2016.
- [60] B. Zhong *et al*, "An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking," *Journal of Educational Computing Research*, vol. 53, no. 4, pp. 562-590, 2016. Available: <https://journals.sagepub.com/doi/full/10.1177/0735633115608444>, doi: 10.1177/0735633115608444.
- [61] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?" *Computers in Human Behavior*, vol. 41, pp. 51-61, 2014. Available: <https://www.sciencedirect.com/science/article/pii/S0747563214004634>, doi: 10.1016/j.chb.2014.09.012.
- [62] E. Hunsaker, *Computational Thinking*. The K-12 Educational Technology Handbook, A. Ottenbreit-Leftwich and R. Kimmons, Eds. EdTech Books, 2020, pp. 80-95. [Online]. Available: <https://bit.ly/2UP0gZi>.
- [63] P. J. Rich *et al*, "Coding in K-8: International Trends in Teaching Elementary/Primary Computing," *TechTrends*, pp. 1-19, 2018.
- [64] C. S. González-González, "State of the art in the teaching of computational thinking and programming in childhood education," *Education in the Knowledge Society*, vol. 20, pp. 1-15, 2019.
- [65] S. Bocconi, A. Chiocciariello, G. Dettori, A. Ferrari, and K. Engelhardt, "Developing computational thinking in compulsory education," *Eur. Commission, JRC Sci. Policy Rep.*, pp. 68, 2016.

- [66] Unión Europea, "Plan de Acción de Educación Digital (2021-2027)", Unión Europea, European Commission, European Education Area, 2021. [Online]. Available: https://ec.europa.eu/education/education-in-the-eu/digital-education-action-plan_es. [Accessed: Dic. 12, 2021].
- [67] Unión Europea, "DigComp. The Digital Competence Framework 2.0", EU SCIENCE HUB, 2021. [Online]. Available: <https://ec.europa.eu/jrc/en/digcomp/digital-competence-framework>. [Accessed: Dic. 12, 2021].
- [68] Code.org, "Learn computer science. Change the world", *Code.org*. [Online]. Available: <https://code.org/>. [Accessed: Jan. 20, 2022].
- [69] La hora del código, "Celebremos las Ciencias de la Computación en todas partes", *hourofcode.com*. [Online]. Available: <https://hourofcode.com/es>. [Accessed: Jan. 20, 2022].
- [70] J. Á Velázquez-Iturbide, "Report of the Spanish computing scientific society on computing education in pre-university stages," in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2018.
- [71] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), "Programación, robótica y Pensamiento Computacional en el aula. Situación en España y propuesta normativa", Ministerio de Educación y Formación Profesional, Gobierno de España, Oct, 2018. [Online]. Available: <https://code.intef.es/wp-content/uploads/2018/10/Ponencia-sobre-Pensamiento-Computacional.-Informe-Final.pdf>.
- [72] Junta de Castilla y León, "Proyecto de innovación 'TIC STEAM' - Curso 2016-2017", *educacyl Portal de Educación*, 2017. [Online]. Available: <https://www.educa.jcyl.es/profesorado/es/formacion-profesorado/proyectos-relacionados-formacion-permanente-profesorado/competencia-digital/convocatorias/proyecto-innovacion-tic-steam-curso-2016-2017>.
- [73] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), "Escuela de Pensamiento Computacional e Inteligencia Artificial", Ministerio de Educación y Formación Profesional, Gobierno de España, 2021. [Online]. Available: <https://intef.es/tecnologia-educativa/pensamiento-computacional/>.
- [74] Programamos, "Es necesario que todo el mundo desarrolle su Pensamiento Computacional, desde niños a mayores", *programamos.com*. [Online]. Available: <https://programamos.es/>. [Accessed: Jan. 20, 2022].
- [75] Boletín Oficial del Estado, "Marco de referencia de la competencia digital docente", Ministerio de la Presidencia, Relaciones con las Cortes

- y Memoria Democrática, Gobierno de España. [Online]. Available: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2020-7775#top.
- [76] E. Hunsaker, *Integrating computational thinking*. K-12 technology integration. Pressbooks. Available: [https://k12techintegration.pressbooks.com/chapter/integrating-computational-t hinking](https://k12techintegration.pressbooks.com/chapter/integrating-computational-thinking/), 2018.
- [77] M. Zapata-Ros, "Computational thinking unplugged," *Education in the Knowledge Society*, vol. 20, pp. 1-29, 2019.
- [78] T. Bell *et al*, "Computer science unplugged: School students doing real computing without computers," *The New Zealand Journal of Applied Computing and Information Technology*, vol. 13, no. 1, pp. 20-29, 2009.
- [79] S. Adak, "Effectiveness of Constructivist Approach on Academic Achievement in Science at Secondary Level." *Educational Research and Reviews*, vol. 12, no. 22, pp. 1074-1079, 2017.
- [80] V. Richardson, "Constructivist pedagogy," *Teachers College Record*, vol. 105, no. 9, pp. 1623-1640, 2003.
- [81] C. T. Fosnot and R. S. Perry, "Constructivism: A psychological theory of learning," *Constructivism: Theory, Perspectives, and Practice*, vol. 2, no. 1, pp. 8-33, 1996.
- [82] S. Papert and I. Harel, "Situating constructionism," *Constructionism*, vol. 36, no. 2, pp. 1-11, 1991.
- [83] C. W. Istikomah and C. Budiyo, "The contribution of educational robotics and constructivist approach to computational thinking in the 21st century," in *Proceedings of the the 1st International Conference on Computer Science and Engineering Technology Universitas Muria Kudus*, 2018.
- [84] S. Hartikainen *et al*, "The concept of active learning and the measurement of learning outcomes: A review of research in engineering higher education," *Education Sciences*, vol. 9, no. 4, pp. 276, 2019.
- [85] R. T. Johnson and D. W. Johnson, "Active learning: Cooperation in the classroom," *The Annual Report of Educational Psychology in Japan*, vol. 47, pp. 29-30, 2008.
- [86] P. Dillenbourg, Ed., *Collaborative Learning: Cognitive and Computational Approaches*. Oxford, U.K.: Pergamon, 1999.
- [87] M. Laal and S. M. Ghodsi, "Benefits of collaborative learning," *Procedia - Social and Behavioral Sciences*, vol. 31, pp. 486-490, 2012. Available: <http://www.sciencedirect.com/science/article/pii/S1877042811030205>, doi: [//doi.org/10.1016/j.sbspro.2011.12.091](https://doi.org/10.1016/j.sbspro.2011.12.091).

- [88] D. W. Johnson and R. T. Johnson, "Effects of cooperative and individualistic learning experiences on interethnic interaction," *Journal of Educational Psychology*, vol. 73, (3), pp. 444-449, 1981. Available: <https://search.proquest.com/docview/614371667>. DOI: 10.1037/00220663.73.3.444.
- [89] M. Laal and M. Laal, "Collaborative learning: what is it?" *Procedia - Social and Behavioral Sciences*, vol. 31, pp. 491-495, 2012. Available: <http://www.sciencedirect.com/science/article/pii/S1877042811030217>. DOI: [//doi.org/10.1016/j.sbspro.2011.12.092](https://doi.org/10.1016/j.sbspro.2011.12.092).
- [90] D. Boud and G. E. Feletti, Eds., *The Challenge of Problem-Based Learning*. London, U.K.: Routledge, 1997, doi: 10.4324/9781315042039.
- [91] J. Roschelle and S. D. Teasley, "The construction of shared knowledge in collaborative problem solving," in *Computer Supported Collaborative Learning*, 1995.
- [92] P. C. Blumenfeld *et al*, "Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning," *Educational Psychologist*, vol. 26, no. 3-4, pp. 369-398, 1991. Available: <http://www.tandfonline.com/doi/abs/10.1080/00461520.1991.9653139>, doi: 10.1080/00461520.1991.9653139.
- [93] L. Zhang and J. Nouri, "A systematic review of learning computational thinking through Scratch in K-9," *Comput. Educ.*, vol. 141, pp. 103607, 2019.
- [94] J. Moreno-León, G. Robles, and M. Román-González, "Towards data-driven learning paths to develop computational thinking with scratch," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 193-205, 2017.
- [95] J. Nouri *et al*, "Development of computational thinking, digital competence and 21st century skills when learning programming in K-9," *Education Inquiry*, vol. 11, no. 1, pp. 1-17, 2020.
- [96] J. Fagerlund *et al*, "Computational thinking in programming with scratch in primary schools: A systematic review," *Computer Applications in Engineering Education*, vol. 29, no. 1, pp. 12-28, 2021.
- [97] G. Falloon, "An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad," *J. Comput. Assisted Learn.*, vol. 32, no. 6, pp. 576-593, 2016.
- [98] Massachusetts Institute of Technology: MIT, "Scratch", scratch.mit.edu, [Online]. Available: <https://scratch.mit.edu/>.
- [99] P. Mozelius and L. Öberg, "Play-based learning for programming education in primary school: The östersund model," in *European Conference on E-Learning- ECEL 2017, Porto, Portugal, 26-27 October 2017*, 2017.

- [100] A. G. Soosai Raj *et al*, "Does native language play a role in learning a programming language?" in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018.
- [101] M. Jancheski, "Improving teaching and learning computer programming in schools through educational software," *Olympiads in Informatics*, vol. 11, pp. 55-75, 2017.
- [102] E. F. Silva, B. J. Dembogurski and G. S. Semaan, "A Systematic Review of Computational Thinking in Early Ages," *arXiv Preprint arXiv:2106.10275*, 2021.
- [103] J. M. Cañas Plaza *et al*, "Entorno docente universitario para la programación de los robots," *Revista Iberoamericana De Automática E Informática Industrial*, vol. 15, no. 4, pp. 404-415, 2018.
- [104] A. Hamidi, M. Milrad and S. Tavajoh, "Combining maker technologies to promote computational thinking and heart-ware skills through project-based activities: Design considerations and empirical outputs," in *Cte-Stem 2021*, 2021.
- [105] R. Zerega *et al*, "A co-design approach for developing computational thinking skills in connection to STEM related curriculum in swedish schools," in *Cte-Stem 2021*, 2021.
- [106] M. Riedmiller and T. Gabel, "On experiences in a complex and competitive gaming domain: Reinforcement learning meets robocup," in *2007 IEEE Symposium on Computational Intelligence and Games*, 2007.
- [107] N. L. Fanchamps *et al*, "The influence of SRA programming on algorithmic thinking and self-efficacy using Lego robotics in two types of instruction," *International Journal of Technology and Design Education*, vol. 31, no. 2, pp. 203-222, 2021.
- [108] N. L. Fanchamps *et al*, "The Impact of SRA-Programming on Computational Thinking in a Visual Oriented Programming Environment," *Education and Information Technologies*, vol. 26, no. 5, pp. 6479-6498, 2021.
- [109] M. Montessori, J. M. Hunt and J. Valsiner, *The Montessori Method*. Routledge, 2017.
- [110] M. Elkin, A. Sullivan, and M. U. Bers, "Programming with the KIBO robotics kit in preschool classrooms," *Computers in the Schools*, vol. 33, no. 3, pp. 169-186, 2016.
- [111] L. L. Román, "Metodología para la enseñanza aprendizaje de la lógica de la programación orientada a objetos," in *5ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI*, International Institute of Informatics and Systemics, EE. UU, 2007.

- [112] J. Urquiza-Fuentes and J. Á Velázquez-Iturbide, "A survey of successful evaluations of program visualization and algorithm animation systems," *ACM Transactions on Computing Education (TOCE)*, vol. 9, no. 2, pp. 1-21, 2009.
- [113] L. Seiter and B. Foreman, "Modeling the learning progressions of computational thinking of primary grade students," in *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 2013, Available: <http://doi.acm.org/10.1145/2493394.2493403>, doi: 10.1145/2493394.2493403.
- [114] S. Grover, R. Pea and S. Cooper, "Designing for deeper learning in a blended computer science course for middle school students," *Computer Science Education*, vol. 25, no. 2, pp. 199-237, 2015.
- [115] A. Basawapatna *et al*, "Recognizing computational thinking patterns," in *Proceedings of the 42nd ACM technical symposium on Computer science education*, pp. 245-250 Mar 9, 2011.
- [116] S. Papadakis and M. Kalogiannakis, "Evaluating a course for teaching advanced programming concepts with scratch to preservice kindergarten teachers: A case study in greece," in *Early Childhood Education* Anonymous 2018.
- [117] A. Eguiluz *et al*, "Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis," *Tetc*, pp. 1, 2017. Available: <https://ieeexplore.ieee.org/document/8093668>, doi: 10.1109/TETC.2017.2768550.
- [118] G. Petri and C. Gresse von Wangenheim, "How games for computing education are evaluated? A systematic literature review," *Computers & Education*, vol. 107, pp. 68-90, 2017. Available: <http://dx.doi.org/10.1016/j.compedu.2017.01.004>, doi: 10.1016/j.compedu.2017.01.004.
- [119] A. Vahldick, A. J. Mendes and M. J. Marcelino, "Dynamic Difficulty Adjustment through a Learning Analytics Model in a Casual Serious Game for Computer Programming Learning," *EAI Endorsed Transactions on Serious Games*, vol. 4, no. 13, 2017.
- [120] C. Kroustalli and S. Xinogalos, "Studying the effects of teaching programming to lower secondary school students with a serious game: a case study with Python and CodeCombat," *Education and Information Technologies*, pp. 1-27, 2021.
- [121] M. Csikszentmihalyi, *Finding Flow: The Psychology of Engagement with Everyday Life*. United States: 1997.
- [122] S. H. Chuah, "Why and who will adopt extended reality technology? Literature review, synthesis, and future research agenda," *Literature Review, Synthesis, and Future Research Agenda*, Dec. 13, 2018.

- [123] M. Montola, "Exploring the edge of the magic circle: Defining pervasive games," in *Proceedings of DAC*, 2005.
- [124] C. Bonillo, J. Marco, and E. Cerezo, "Developing pervasive games in interactive spaces: the JUGUEMOS toolkit," *Multimedia Tools Appl*, vol. 78, no. 22, pp. 32261–32305, 2019.
- [125] N. M. Medina *et al*, "Hacia una propuesta de evaluación heurística de experiencias de juego pervasivas," *Revista De La Asociación Interacción Persona Ordenador (AIPO)*, vol. 2, no. 2, pp. 42–53, 2021.
- [126] J. Marco, C. Bonillo, and E. Cerezo, "A tangible interactive space odyssey to support children learning of computer programming," in *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, 2017.
- [127] S. SUKIRMAN *et al*, "A Strategy of Learning Computational Thinking through Game Based in Virtual Reality: Systematic Review and Conceptual Framework," *Informatics in Education*, 2021.
- [128] Y. Ching, Y. Hsu and S. Baldwin, "Developing computational thinking with educational technologies for young learners," *TechTrends*, vol. 62, no. 6, pp. 563–573, 2018.
- [129] C. P. Brackmann *et al*, "Development of computational thinking skills through unplugged activities in primary school," in *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 2017.
- [130] S. Suh, M. Lee and G. Xia, "Coding strip: A pedagogical tool for teaching and learning programming concepts through comics," in *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2020.
- [131] G. Aranda and J. P. Ferguson, "Unplugged Programming: The future of teaching computational thinking?" *Pedagogika*, vol. 68, no. 3, 2018.
- [132] J. Waite *et al*, "Unplugged computing and semantic waves: Analysing crazy characters," in *Proceedings of the 1st UK & Ireland Computing Education Research Conference*, 2019.
- [133] A. Dengel, "Computer science replugged: What is the use of virtual reality in computer science education?" in *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, 2019.
- [134] X. Tang *et al*, "Assessing computational thinking: A systematic review of empirical studies," *Computers & Education*, vol. 148, pp. 103798, 2020. Available: <http://www.sciencedirect.com/science/article/pii/S0360131519303483>, doi: <https://doi.org/10.1016/j.compedu.2019.103798>.

- [135] J. Denner, L. Werner and E. Ortiz, "Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?" *Computers & Education*, vol. 58, no. 1, pp. 240-249, 2012. Available: <https://www.sciencedirect.com/science/article/pii/S0360131511001849>, doi: 10.1016/j.compedu.2011.08.006.
- [136] I. Cetin and E. Dubinsky, "Reflective abstraction in computational thinking," *Journal of Mathematical Behavior*, vol. 47, pp. 70-80, 2017. Available: <https://www.sciencedirect.com/science/article/pii/S0732312316301511>, doi: 10.1016/j.jmathb.2017.06.004.
- [137] S. Jun *et al*, "Assessing the computational literacy of elementary students on a national level in Korea," *Educational Assessment, Evaluation and Accountability*, vol. 26, no. 4, pp. 319-332, 2014.
- [138] L. Werner *et al*, "The fairy performance assessment," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pp. 215-220, 2012.
- [139] V. Garneli and K. Chorianopoulos, "Programming video games and simulations in science education: exploring computational thinking through code analysis," *Interactive Learning Environments*, vol. 26, no. 3, pp. 386-401, 2018. Available: <http://www.tandfonline.com/doi/abs/10.1080/10494820.2017.1337036>, doi: 10.1080/10494820.2017.1337036.
- [140] A. Mühling, A. Ruf and P. Hubwieser, "Design and first results of a psychometric test for measuring basic programming abilities," in *Proceedings of the workshop in primary and secondary computing education*, pp. 2-10, Nov 9, 2015.
- [141] Ö Korkmaz, R. Çakir and M. Y. Özden, "A validity and reliability study of the computational thinking scales (CTS)," *Computers in Human Behavior*, vol. 72, pp. 558-569, 2017. Available: <https://www.sciencedirect.com/science/article/pii/S0747563217300055>, doi: 10.1016/j.chb.2017.01.005.
- [142] M. Román-González *et al*, "Extending the nomological network of computational thinking with non-cognitive factors," *Computers in Human Behavior*, vol. 80, pp. 441-459, 2018. Available: <http://www.sciencedirect.com/science/article/pii/S0747563217305563>, doi: <https://doi.org/10.1016/j.chb.2017.09.030>.
- [143] M. Román-González *et al*, "Can computational talent be detected? Predictive validity of the Computational Thinking Test," *International Journal of Child Computer Interaction*, vol. 18, pp. 47-58, 2018. Available: <http://www.sciencedirect.com/science/article/pii/S221286891730034X>, doi: <https://doi.org/10.1016/j.ijcci.2018.06.004>.
- [144] AERA, APA and NCME, *Standards for Educational and Psychological Testing*. 2014. Available: http://www.fachportal-paedagogik.de/fis_bildung/suche/fis_set.html?FId=1050398.

- [145] G. Chen *et al*, "Assessing elementary students' computational thinking in everyday reasoning and robotics programming," *Computers & Education*, vol. 109, pp. 162–175, 2017. Available: <http://www.sciencedirect.com/science/article/pii/S0360131517300490>, doi: <https://doi.org/10.1016/j.compedu.2017.03.001>.
- [146] E. Relkin, L. de Ruiter and M. U. Bers, "TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education," *Journal of Science Education and Technology*, vol. 29, no. 4, pp. 482–498, 2020. Available: <https://search.proquest.com/docview/2413230852>, doi: 10.1007/s10956-020-09831-x.
- [147] Y. Li, S. Xu, and J. Liu, "Development and Validation of Computational Thinking Assessment of Chinese Elementary School Students," *Journal of Pacific Rim Psychology*, vol. 15, pp. 18344909211010240, 2021. Available: <https://doi.org/10.1177/18344909211010240>, doi: 10.1177/18344909211010240.
- [148] S. Grover, R. Pea and S. Cooper, "«Systems of assessments" for deeper learning of computational thinking in K-12," in *Proceedings of the 2015 Annual Meeting of the American Educational Research Association*, 2015.
- [149] G. Siemens and P. Long, "Penetrating the Fog: Analytics in Learning and Education," *EDUCAUSE Review*, vol. 46, no. 5, pp. 30, 2011.
- [150] A. Mathrani, S. Christian and A. Ponder-Sutton, "PlayIT: Game based learning approach for teaching programming concepts," *Journal of Educational Technology & Society*, vol. 19, no. 2, pp. 5-17, 2016.
- [151] C. Malliarakis, M. Satratzemi and S. Xinogalos, "CMX: The effects of an educational MMORPG on learning and teaching computer programming," *IEEE Transactions on Learning Technologies*, vol. 10, no. 2, pp. 219-235, 2016.
- [152] C. Malliarakis, M. Satratzemi and S. Xinogalos, "Integrating learning analytics in an educational MMORPG for computer programming," in *2014 IEEE 14th International Conference on Advanced Learning Technologies*, 2014.
- [153] W. Greller and H. Drachsler, "Translating learning into numbers: A generic framework for learning analytics," *Journal of Educational Technology & Society*, vol. 15, no. 3, pp. 42-57, 2012.
- [154] M. A. Chatti *et al*, "Learning Analytics: Challenges and Future Research Directions," *eleed*, vol. 10, no 1. Nov 6, 2014.
- [155] M. Román González, "Computational thinking test: Design guidelines and content validation," in *Proceedings of EDULEARN15 conference, Barcelona*, pp. 2436-2444, 201.

- [156] D. Căprioară, "Emotions and the Learning of School Mathematics," *Bulletin of the Transilvania University of Braşov, Series VII: Social Sciences and Law*, vol. 10, no. 1, pp. 9-18, 2017. Available: <http://www.ceeol.com/search/article-detail?id=566146>.
- [157] C. Chen and P. Herbst, "The interplay among gestures, discourse, and diagrams in students' geometrical reasoning," *Educational Studies in Mathematics*, vol. 83, no. 2, pp. 285-307, 2013. Available: <https://doi.org/10.1007/s10649-012-9454-2>, doi: 10.1007/s10649-012-9454-2.
- [158] H. Y. Durak and M. Saritepeci, "Analysis of the relation between computational thinking skills and various variables with the structural equation model," *Computers & Education*, vol. 116, pp. 191-202, 2018. Available: <http://www.sciencedirect.com/science/article/pii/S0360131517302087>, doi: [//doi.org/10.1016/j.compedu.2017.09.004](https://doi.org/10.1016/j.compedu.2017.09.004).
- [159] T. Watanabe, "Visual Reasoning Tools in Action: Double Number Lines, Area Models, and Other Diagrams Power Up Students' Ability to Solve and Make Sense of Various Problems," *Mathematics Teaching in the Middle School*, vol. 21, no. 3, pp. 152, 2015.
- [160] C. E. Lance, M. M. Butts and L. C. Michels, "The sources of four commonly reported cutoff criteria: What did they really say?" *Organ. Res. Methods*, vol. 9, no. 2, pp. 202-220, 2006.
- [161] M. Román-González, J. Moreno-León and G. Robles, "Combining assessment tools for a comprehensive evaluation of computational thinking interventions," in *Computational Thinking Education*, S. Kong and H. Abelson, Eds. 2019, Available: https://doi.org/10.1007/978-981-13-6528-7_6, doi: 10.1007/978-981-13-6528-7_6.
- [162] A. A. De Souza *et al*, "Data mining framework to analyze the evolution of computational thinking skills in game building workshops," *IEEE Access*, vol. 7, pp. 82848-82866, 2019.
- [163] R. F. DeVellis, "Classical test theory," *Med. Care*, pp. S50-S59, 2006.
- [164] R. K. Hambleton and R. W. Jones, "Comparison of classical test theory and item response theory and their applications to test development," *Educational Measurement: Issues and Practice*, vol. 12, no. 3, pp. 38-47, 1993.
- [165] L. El-Hamamsy, M. Zapata-Caceres, E. Martín, F. Mondada, J. Dehler Zufferey, and B. Bruno, "The competent Computational Thinking test (cCTt): Development and validation of an unplugged Computational Thinking test for upper primary school," *Journal of Educational Computing Research*, 2022. In press.
- [166] S. Kong, "A framework of curriculum design for computational thinking development in K-12 education," *Journal of Computers in Education*, vol. 3, no. 4, pp. 377-394, 2016.

- [167] F. Buitrago Flórez *et al*, "Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming," *Review of Educational Research*, vol. 87, no. 4, pp. 834, 2017. Available: <https://search.proquest.com/docview/1917804191>, doi: 10.3102/0034654317710096.
- [168] J. Alvarez *et al*, "Blue Bot Project Experiment," *8th International Toy Research Association World Conference* Jul 11, 2018.
- [169] C. Sáez Fernández, G. Viera López and D. Pérez Marín, "Propuesta metodológica de la enseñanza de la programación en Educación Infantil con Cubetto," *IE Comunicaciones: Revista Iberoamericana De Informática Educativa*, no. 28, pp. 1-8, 2018. Available: <https://dialnet.unirioja.es/servlet/oaiart?codigo=6750332>.
- [170] W. Wu *et al*, "Investigating the learning-theory foundations of game-based learning: a meta-analysis," *Journal of Computer Assisted Learning*, vol. 28, no. 3, pp. 265-279, 2012. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2729.2011.00437.x>, doi: 10.1111/j.1365-2729.2011.00437.x.
- [171] E. Boyle, T. M. Connolly, and T. Hainey, "The role of psychology in understanding the impact of computer games," *Entertainment Computing*, vol. 2, no. 2, pp. 69-74, 2011. Available: <https://www.sciencedirect.com/science/article/pii/S1875952110000200>, doi: 10.1016/j.entcom.2010.12.002.
- [172] M. Li and C. Tsai, "Game-Based Learning in Science Education: A Review of Relevant Research," *Journal of Science Education and Technology*, vol. 22, no. 6, pp. 877-898, 2013. Available: <https://doi.org/10.1007/s10956-013-9436-x>, doi: 10.1007/s10956-013-9436-x.
- [173] W. Wu *et al*, "Re-exploring game-assisted learning research: The perspective of learning theoretical bases," *Computers & Education*, vol. 59, no. 4, pp. 1153-1161, 2012. Available: <https://www.sciencedirect.com/science/article/pii/S0360131512001248>, doi: 10.1016/j.compedu.2012.05.003.
- [174] M. Csikszentmihalyi, "Flow and the psychology of discovery and invention," *HarperPerennial, New York*, vol. 39, 1997.
- [175] T. Hainey *et al*, "A systematic literature review of games-based learning empirical evidence in primary education," *Computers & Education*, vol. 102, pp. 202-223, 2016.
- [176] J. R. Savery and T. M. Duffy, "Problem based learning: An instructional model and its constructivist framework," *Educational Technology*, vol. 35, no. 5, pp. 31-38, 1995.
- [177] J. R. Savery, "Overview of Problem-based Learning: Definitions and distinctions," *Essential readings in problem-based learning: Exploring*

- and extending the legacy of Howard S. Barrows*, vol. 9, no 2, pp. 5-15, 2015. Available: <https://library.biblioboard.com/viewer/620aa903-49eb-49d1-8fc2-b02dbeda193c>.
- [178] B. Haßler, L. Major, and S. Hennessy, "Tablet use in schools: A critical review of the evidence for learning outcomes," *Journal of Computer Assisted Learning*, vol. 32, no. 2, pp. 139-156, 2016.
 - [179] Y. Rogers *et al*, "Equal opportunities: Do shareable interfaces promote more group participation than single user displays?" *Human-Computer Interaction*, vol. 24, no. 1-2, pp. 79-116, 2009.
 - [180] D. Roldan-Alvarez *et al*, "DEDOS: An Authoring Toolkit to Create Educational Multimedia Activities for Multiple Devices," *IEEE Transactions on Learning Technologies*, vol. 11, no. 4, pp. 493-505, 2018. Available: <https://ieeexplore.ieee.org/document/8244297>, doi: 10.1109/TLT.2017.2788867.
 - [181] C. Lin, L. Wong and Y. Shao, "Comparison of 1: 1 and 1: m CSCL environment for collaborative concept mapping," *Journal of Computer Assisted Learning*, vol. 28, no. 2, pp. 99-113, 2012.
 - [182] M. Zapata-Cáceres, E. Martín-Barroso and M. Román-González, "Computational Thinking Test for Beginners: Design and Content Validation," *2020 IEEE Global Engineering Education Conference (EDUCON)*. in Press, 2020.
 - [183] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. Palgrave macmillan, 2005.
 - [184] C. Larman, *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, 2004.
 - [185] L. García-de-la-Fuente, "Tratamiento de ficheros generados por Blue Ant Code," *Trabajo De Fin De Grado En Ingeniería Informática, Universidad Rey Juan Carlos*, 2020.
 - [186] G. Matilla-Jiménez, "Desarrollo una aplicación web para facilitar la visualización y análisis se datos de Blue Ant Code," *Trabajo De Fin De Grado En Ingeniería Informática, Universidad Rey Juan Carlos*, 2020.
 - [187] J. Moreno and G. Robles, "Automatic detection of bad programming habits in scratch: A preliminary study," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pp. 1-4, Oct. 2014.
 - [188] N. Diana *et al*, "An instructor dashboard for real-time analytics in interactive programming assignments," in *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pp. 272-279, 2017.

- [189] L. Gal *et al*, "Suggesting a log-based creativity measurement for online programming learning environment," in *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pp. 273-277, 2017.
- [190] D. Olivares and C. Hundhausen, "Supporting learning analytics in computing education," in *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pp. 584-585, 2017.
- [191] M. Zapata-Ros and P. Pérez-Paredes, *El Pensamiento Computacional. Análisis De Una Competencia Clave*. New York: Create Space Independent Publishing, 2018.
- [192] H. Søndergaard and R. A. Mulder, "Collaborative learning through formative peer review: Pedagogy, programs and potential," *Computer Science Education*, vol. 22, no. 4, pp. 343-367, 2012.
- [193] M. R. Snodgrass, M. Israel and G. C. Reese, "Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis," *Computers & Education*, vol. 100, pp. 1-17, 2016.
- [194] N. Ayuso *et al*, "Gender Gap in STEM: A Cross-Sectional Study of Primary School Students' Self-Perception and Test Anxiety in Mathematics," *IEEE Transactions on Education*, vol. 64, no. 1, pp. 40-49, 2020.
- [195] L. Sun *et al*, "Primary school students' perceptions of scaffolding in digital game-based learning in mathematics," *Learning, Culture and Social Interaction*, vol. 28, pp. 100457, 2021.
- [196] M. Cengiz *et al*, "Development of an interactive game-based learning environment to teach data mining," *The International Journal of Engineering Education*, vol. 33, no. 5, pp. 1598-1617, 2017.
- [197] H. Lu, "Rethinking game-based learning from a gender perspective: a case study of a male English language learner in China," *Gender, Technology and Development*, vol. 16, no. 1, pp. 49-70, 2012.
- [198] E. A. Boyle *et al*, "An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games," *Computers & Education*, vol. 94, pp. 178-192, 2016. Available: <http://www.sciencedirect.com/science/article/pii/S0360131515300750>, doi: [//doi.org/10.1016/j.compedu.2015.11.003](https://doi.org/10.1016/j.compedu.2015.11.003).
- [199] R. J. Mislevy, L. S. Steinberg and R. G. Almond, "Focus Article: On the Structure of Educational Assessments," *Measurement (Mahwah, N.J.)*, vol. 1, no. 1, pp. 3-62, 2003. Available: http://www.tandfonline.com/doi/abs/10.1207/S15366359MEA0101_02, doi: [10.1207/S15366359MEA0101_02](https://doi.org/10.1207/S15366359MEA0101_02).
- [200] V. J. Shute and G. R. Moore, "Consistency and validity in game-based stealth assessment," *Technology Enhanced Innovative Assessment*:

- Development, Modeling, and Scoring from an Interdisciplinary Perspective*, vol. 296, 2017.
- [201] C. Alonso Fernández *et al*, "Improving serious games analyzing learning analytics data: lessons learned," in *International Conference on Games and Learning Alliance*. Springer, Cham Dec 5, 2018.
 - [202] G. Wallner and S. Kriglstein, "Comparative visualization of player behavior for serious game analytics," in *Serious Games Analytics*, Cham: Springer International Publishing, pp. 159-179, 2015.
 - [203] E. Zimmerman, "Narrative, interactivity, play, and games: Four naughty concepts in need of discipline," *First Person: New Media as Story, Performance, and Game*, vol. 154, 2004.
 - [204] E. Carratala Sánchez, "Análisis de la teoría de las metas de logro y de la autodeterminación en los planes de especialización deportiva de la Generalitat Valenciana." 2004.
 - [205] C. J. Ferguson and C. K. Olson, "Friends, fun, frustration and fantasy: Child motivations for video game play," *Motivation and Emotion*, vol. 37, no. 1, pp. 154-164, 2013.
 - [206] P. R. Pintrich and A. Zusho, "The development of academic self-regulation: The role of cognitive and motivational factors," in *Development of Achievement Motivation*, 2002.
 - [207] A. E. Gottfried, "Academic intrinsic motivation in young elementary school children." *J. Educ. Psychol.*, vol. 82, no. 3, pp. 525, 1990.
 - [208] R. M. Ryan and E. L. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being." *Am. Psychol.*, vol. 55, no. 1, pp. 68, 2000.
 - [209] E. L. Deci and R. M. Ryan, "Intrinsic motivation," *The Corsini Encyclopedia of Psychology*, pp. 1-2, 2010.
 - [210] J. R. Calvo-Ferrer, "Juegos, videojuegos y juegos serios: Análisis de los factores que favorecen la diversión del jugador," *Miguel Hernández Communication Journal*, no. 9, pp. 191-226, 2018.
 - [211] I. Granic, A. Lobel and R. C. Engels, "The benefits of playing video games." *American Psychologist Journal*, vol. 69, no. 1, pp. 66, 2014.
 - [212] M. Ventura, V. Shute and W. Zhao, "The relationship between video game use and a performance-based measure of persistence," *Computers & Education*, vol. 60, no. 1, pp. 52-58, 2013.
 - [213] R. M. Ryan, C. S. Rigby and A. Przybylski, "The motivational pull of video games: A self-determination theory approach," *Motivation and Emotion*, vol. 30, no. 4, pp. 344-360, 2006.

- [214] M. Salminen and N. Ravaja, "Increased oscillatory theta activation evoked by violent digital game events," *Neuroscience Letters - Journals*, vol. 435, no. 1, pp. 69-72, 2008.
- [215] R. Israel-Fishelson and A. HersHKovitz, "Shooting for the stars: Micro-persistence of students in game-based learning environments," in *Early Warning Systems and Targeted Interventions for Student Success in Online Courses*, 2020.
- [216] C. Hart, "Factors associated with student persistence in an online program of study: A review of the literature." *Journal of Interactive Online Learning*, vol. 11, no. 1, 2012.
- [217] R. Israel-Fishelson and A. HersHKovitz, "Micro-persistence and difficulty in a game-based learning environment for computational thinking acquisition," *Journal of Computer Assisted Learning*, 2021.
- [218] Y. Fang *et al*, "Online Learning Persistence and Academic Achievement." *International Educational Data Mining Society*, 2017.
- [219] C. Alonso-Fernández *et al*, "Applications of data science to game learning analytics data: A systematic literature review," *Computers & Education*, vol. 141, pp. 103612, 2019. Available: <https://search.datacite.org/works/10.1016/j.compedu.2019.103612>, doi: 10.1016/j.compedu.2019.103612.
- [220] P. R. Pintrich and A. Zusho, "Student motivation and self-regulated learning in the college classroom," in *Higher Education: Handbook of Theory and Research* Anonymous 2002.
- [221] K. Weber and B. R. Patterson, "Student interest, empowerment and motivation," *Communication Research Reports*, vol. 17, no. 1, pp. 22-29, 2000.
- [222] Á Serrano-Laguna *et al*, "Applying standards to systematize learning analytics in serious games," *Computer Standards & Interfaces*, vol. 50, pp. 116-123, 2017.
- [223] Y. Tran, "Computational thinking equity in elementary classrooms: What third- grade students know and can do," *Journal of Educational Computing Research*, vol. 57, no. 1, pp. 3-31, 2019.
- [224] R. Israel-Fishelson and A. HersHKovitz, "Persistence and achievement in acquiring computational thinking concepts: A large-scale log-based analysis," in *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 2019.
- [225] S. Grover *et al*, "Assessing problem-solving process at scale," in *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pp. 245-248, Apr 25, 2016.

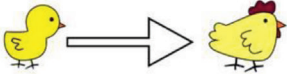
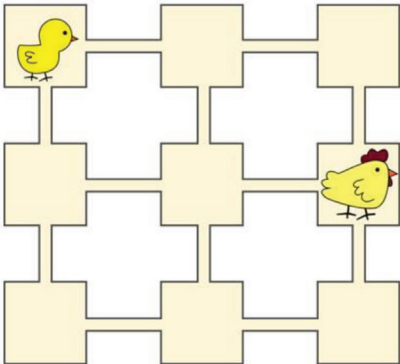
- [226] M. Zapata-Cáceres and E. Martín-Barroso, "Blue Ant Code," 2020. Available: <https://play.google.com/store/apps/details?id=com.artbabygame.blueAnt> <https://apps.apple.com/es/app/blue-ant-code/id1495021110>.
- [227] S. Kong, M. M. Chiu and M. Lai, "A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education," *Computers & Education*, vol. 127, pp. 178-189, 2018. Available: <http://dx.doi.org/10.1016/j.compedu.2018.08.026>, doi: 10.1016/j.compedu.2018.08.026.
- [228] M. Guenaga *et al*, "How do students develop computational thinking? Assessing early programmers in a maze-based online game," *Computer Science Education*, vol. 31, no 2, pp. 259-289, 2021. Available: <https://doi.org/10.1080/08993408.2021.1903248>, doi: 10.1080/08993408.2021.1903248.
- [229] T. P. Alloway, "How does working memory work in the classroom?" *Educational Research and Reviews*, vol. 1, no. 4, pp. 134-139, 2006.
- [230] C. S. Dweck and D. C. Molden, "Their impact on competence motivation and acquisition," *Handbook of Competence and Motivation: Theory and Application*, vol. 2, pp. 135-154, 2017.
- [231] L. S. Blackwell, K. H. Trzesniewski and C. S. Dweck, "Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention," *Child Dev.*, vol. 78, no. 1, pp. 246-263, 2007.
- [232] P. Sweetser and P. Wyeth, "GameFlow: a model for evaluating player enjoyment in games," *Computers in Entertainment (CIE)*, vol. 3, no. 3, pp. 3, 2005.
- [233] M. Lafuente Martínez *et al*, "Assessing Computational Thinking: Development and Validation of the Algorithmic Thinking Test for Adults," 2021.
- [234] L. Narcisi *et al*, "Beginners Computational Thinking Test: Adaptation pour un public de moins de 5 ans." *Service D'Ingénierie Pédagogique Et Du Numérique Éducatif, FPSE, UMONS*, s.f., 2021.
- [235] M. Lázaro-Álvarez, "Sudokus adaptativos para Educación Infantil o Primaria," *Trabajo De Fin De Grado En Ingeniería Informática, Universidad Rey Juan Carlos*, 2021.
- [236] J. Plaza-Carro, "Blue-encode: Juego colaborativo de cifrado y descifrado de mensajes en Educación Primaria," *Trabajo De Fin De Grado En Ingeniería Informática, Universidad Rey Juan Carlos*, 2021.

ANEXOS

A. BCTt

A continuación, se incluyen las versiones del BCTt. El test debe administrarse siguiendo las pautas del protocolo de actuación. Preferentemente se marcarán las respuestas en el test, o bien se utilizará una hoja de respuestas por alumno.

A.1. BCTt v.1

1																																									
 Lleva al pollito con su mamá:	Marca la secuencia correcta:																																								
	<table border="1"><tr><td>A</td><td></td><td>B</td><td></td><td>C</td><td></td><td>D</td><td></td></tr><tr><td colspan="2">→</td><td colspan="2">→</td><td colspan="2">↓</td><td colspan="2">→</td></tr><tr><td colspan="2">→</td><td colspan="2">↓</td><td colspan="2">↓</td><td colspan="2">↓</td></tr><tr><td colspan="2">↓</td><td colspan="2">↓</td><td colspan="2">→</td><td colspan="2">←</td></tr><tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr></table>	A		B		C		D		→		→		↓		→		→		↓		↓		↓		↓		↓		→		←									
A		B		C		D																																			
→		→		↓		→																																			
→		↓		↓		↓																																			
↓		↓		→		←																																			

2

Dibuja la figura:

Marca la secuencia correcta:

A

B

C

D

3

Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A

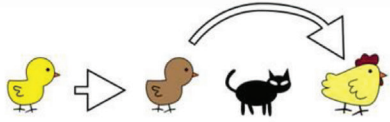
B

C

D

302

4

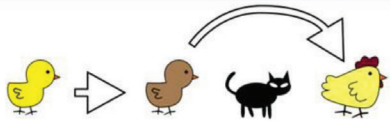


Lleva a los pollitos con su mamá: recoge al pollito marrón por el camino.
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A	B	C	D
→	→	→	↓
↓	↓	→	↓
↓	↓	↓	→
	→	↓	→

5



Lleva a los pollitos con su mamá: recoge al pollito marrón por el camino.
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A	B	C	D
↑	→	→	→
→	→	→	→
→	→	↑	↑
→	↑	→	

6

Dibuja la figura:

Marca la secuencia correcta:

A

B

C

D

7

Lleva al pollito con su mamá:

Marca la secuencia correcta:

A

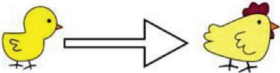
B

C

D

304

8



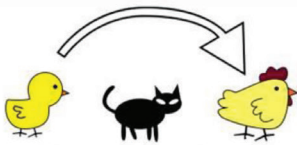
Lleva al pollito con su mamá:

Marca la secuencia correcta:

Marca la secuencia correcta:

1x	1x	2x	1x
2x	2x	1x	2x

9

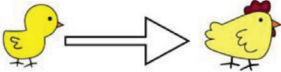


Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

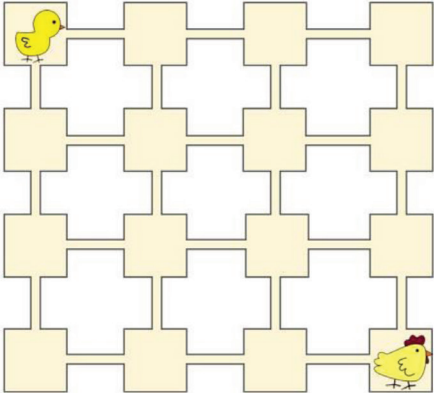
Marca la secuencia correcta:

1x	1x	2x	2x
2x	1x	2x	1x
	1x		

10



Lleva al pollito con su mamá:



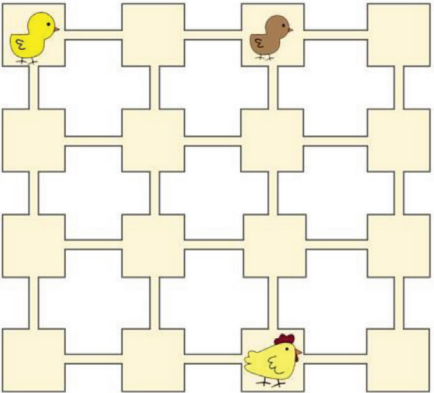
Marca la secuencia correcta:

A		B		C		D	
2x	→	3x	→	3x	↓	2x	↓
1x	↓			3x	→	2x	→

11



Lleva a los pollitos con su mamá: recoge al pollito marrón por el camino.

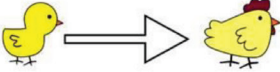


Marca la secuencia correcta:

A		B		C		D	
3x	↓	2x	→	3x	→	3x	→
2x	→	3x	↓	3x	↓	2x	↓

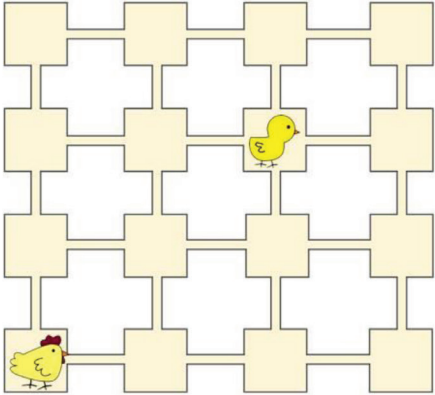
306

12



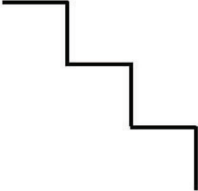
Lleva al pollito con su mamá:

Marca la secuencia correcta:



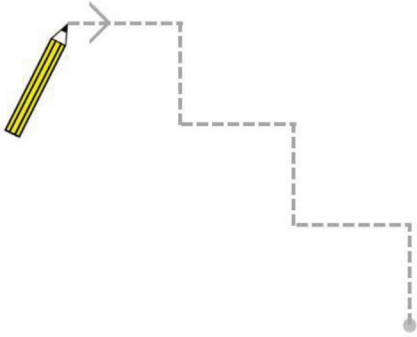
A		B		C		D	
2x		1x 1x	2x	2x 1x			

13



Dibuja la figura:

Marca la secuencia correcta:



A		B		C		D	
3x	3x	3x	3x				

307

14



Lleva al pollito con su mamá.
Cuidado con el gato.

A		B		C		D	
3x →		3x →		3x →		2x ↓	
3x ↓		3x ↓		3x ↓		2x →	

Marca la secuencia correcta:

15



Lleva a los pollitos con su mamá: recoge al
pollito marrón por el camino.

A		B		C		D	
2x →		2x →		2x →		1x →	
2x ↓		2x ↓		2x ↓		2x ↓	
1x →				1x →		2x →	

Marca la secuencia correcta:

308

16



Lleva a los pollitos con su mamá: recoge al pollito marrón por el camino.

Marca la secuencia correcta:

A

2x



B

1x





2x



C

2x





D

2x





2x



17

Dibuja la figura:



Marca la secuencia correcta:

A

2x









B

2x









C

2x







D

2x



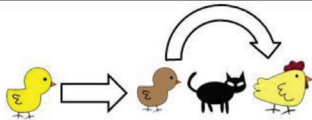






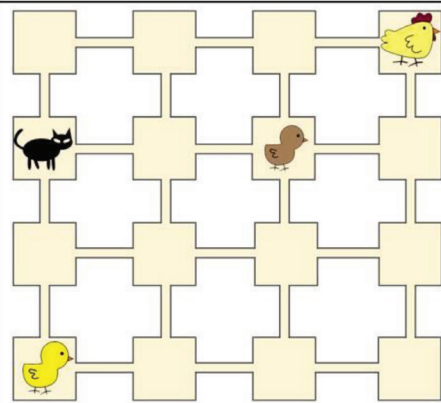
309

18

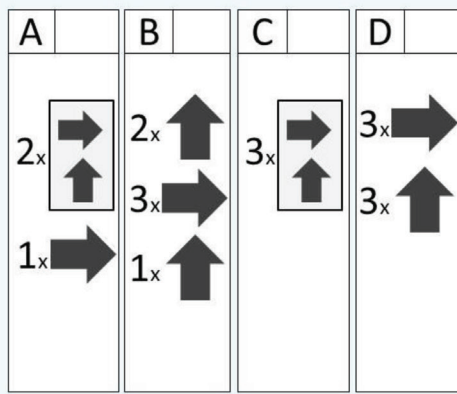


Lleva a los pollitos con su mamá: recoge al pollito marrón por el camino.

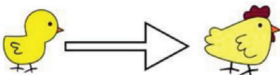
Cuidado con el gato: no pases por su casilla



Marca la secuencia correcta:



19



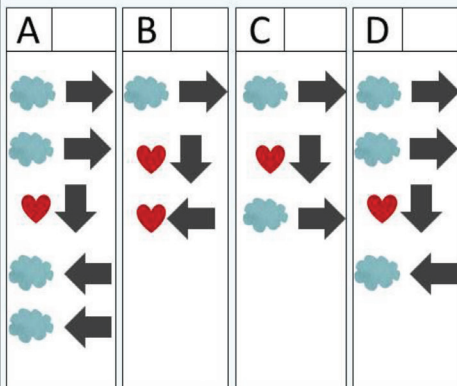
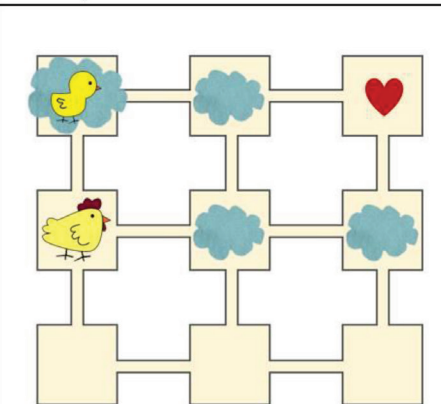
Lleva al pollito con su mamá:

Ejemplo de significado:

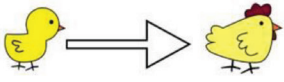


Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:



20



Lleva al pollito con su mamá:

Ejemplo de significado:

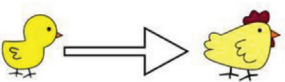


Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

A		B		C		D	
							
							
							
							

21



Lleva al pollito con su mamá:

Ejemplo de significado:

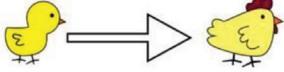


Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

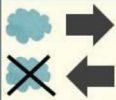
A		B		C		D	
							
							
							
							

22



Lleva al pollito con su mamá:

Ejemplo de significado:

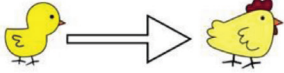


Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

A	B	C	D

23



Lleva al pollito con su mamá:

Ejemplo de significado:

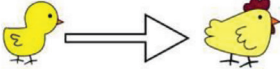


Mientras que el pollito esté en casillas azules, **siempre** avanza a la derecha

Marca la secuencia correcta:

A	B	C	D

24

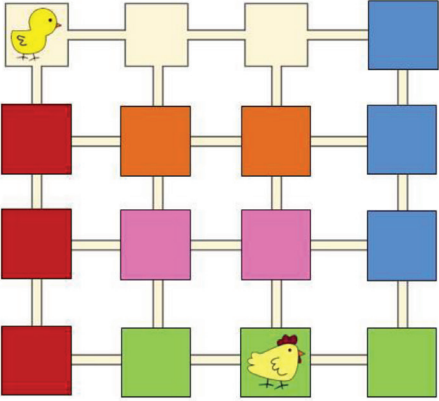


Lleva al pollito con su mamá:

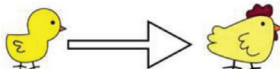
Ejemplo de significado:

 **Mientras que** el pollito esté en casillas azules, **siempre** avanza a la derecha

Marca la secuencia correcta:

	<table border="1"><tr><td>A</td><td></td><td>B</td><td></td><td>C</td><td></td><td>D</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	A		B		C		D																									
A		B		C		D																											
																																	
																																	
																																	

25

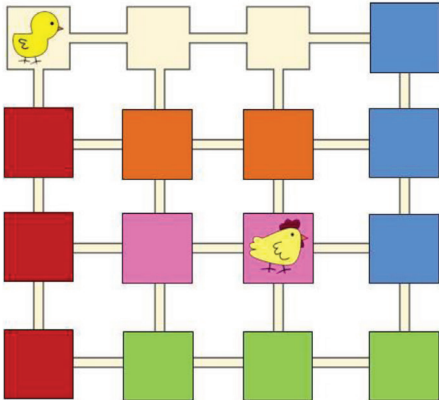


Lleva al pollito con su mamá:

Ejemplo de significado:

 **Mientras que** el pollito esté en casillas azules, **siempre** avanza a la derecha

Marca la secuencia correcta:

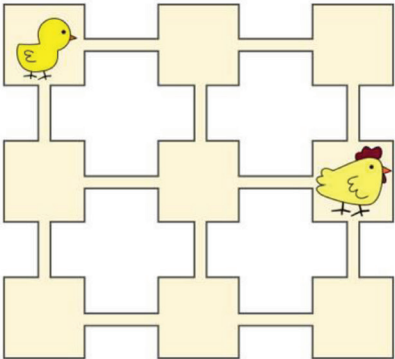
	<table border="1"><tr><td>A</td><td></td><td>B</td><td></td><td>C</td><td></td><td>D</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	A		B		C		D																																	
A		B		C		D																																			
																																									
																																									
																																									
																																									

A.2. BCTt v.2

1



Lleva al pollito con su mamá:



Marca la secuencia correcta:

A



B



C



D



2



Dibuja la figura:



Marca la secuencia correcta:

A



B



C



D



314

3



Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A

B

C

D









4



Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A

B

C

D

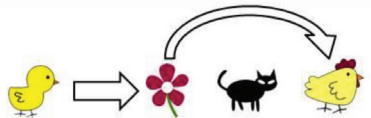








5



Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A	B	C	D
↑	→	→	→
→	→	→	→
→	→	↑	↑
→	↑	→	

6

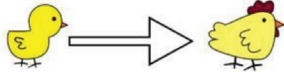


Dibuja la figura:

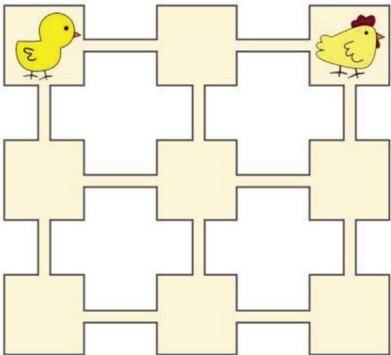
Marca la secuencia correcta:

A	B	C	D
→	↑	↓	↓
→	→	→	→
↓	↑	↑	↑
↑	→		←

7



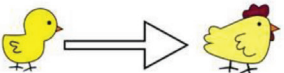
Lleva al pollito con su mamá:



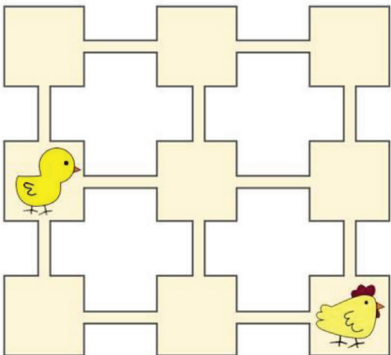
Marca la secuencia correcta:

A	B	C	D
1x →	2x →	3x →	4x →

8



Lleva al pollito con su mamá:



Marca la secuencia correcta:

Marca la secuencia correcta:

A	B	C	D
1x ↓	1x →	2x ↓	1x ↑
2x →	2x ↓	1x →	2x →

317

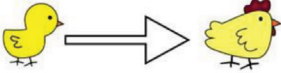
9



Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

A		B		C		D	
1x	↓	1x	→	2x	↓	2x	→
2x	→	1x	↓	2x	→	1x	↓
		1x	→				

10



Lleva al pollito con su mamá:

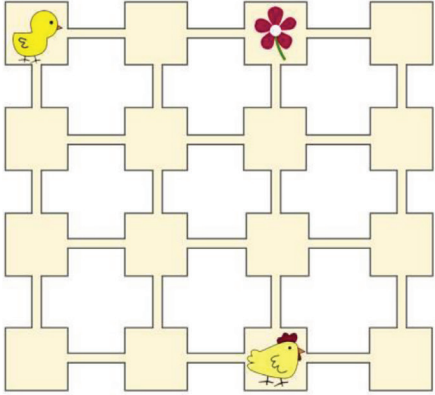
A		B		C		D	
2x	→	3x	→	3x	↓	2x	↓
1x	↓			3x	→	2x	→

318

11



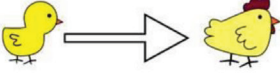
Lleva al pollito con su mamá
Recoge la flor por el camino



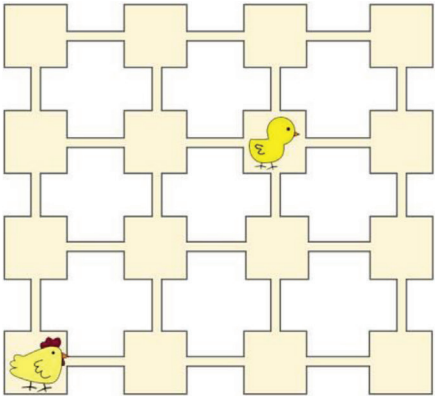
Marca la secuencia correcta:

A	B	C	D
3x ↓	2x →	3x →	3x →
2x →	3x ↓	3x ↓	2x ↓

12



Lleva al pollito con su mamá:

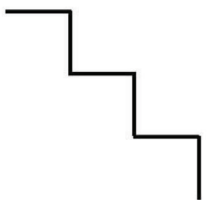


Marca la secuencia correcta:

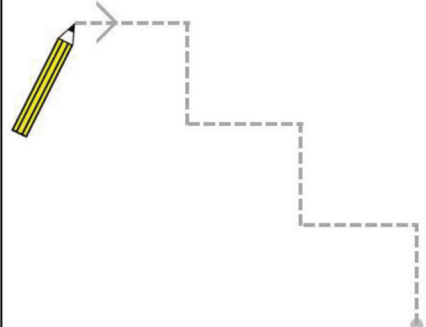
A	B	C	D
2x ← 2x ↓	1x ← 1x ↓	2x ←	2x ← 1x ↓

319

13



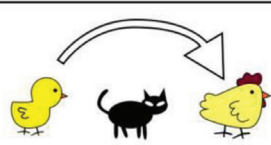
Dibuja la figura:



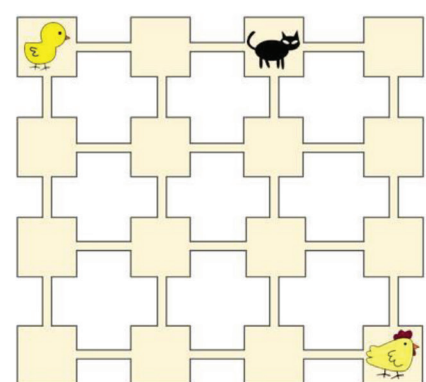
Marca la secuencia correcta:

A	B	C	D
3x ← ↓	3x ↑ →	3x → ↓	3x ↓ →

14



Lleva al pollito con su mamá.
Cuidado con el gato.



Marca la secuencia correcta:

A	B	C	D
3x → ↓	3x → ↓	3x → →	2x ↓ →

15



Lleva al pollito con su mamá
Recoge la flor por el camino

Marca la secuencia correcta:

A

2x →
2x ↓
1x →

B

2x →
↓

C

2x →
↓
1x →

D

1x →
↓
2x →

16



Lleva al pollito con su mamá
Recoge la flor por el camino

Marca la secuencia correcta:

A

2x ↑

B

1x ←
↑
2x →

C

2x ←
↑

D

2x ←
↑
2x →

321

17

Dibuja la figura: 

Marca la secuencia correcta:

A	B	C	D
			

18

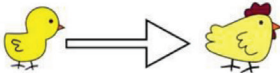


Lleva al pollito con su mamá
 Recoge la flor por el camino
 Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:

A	B	C	D
			

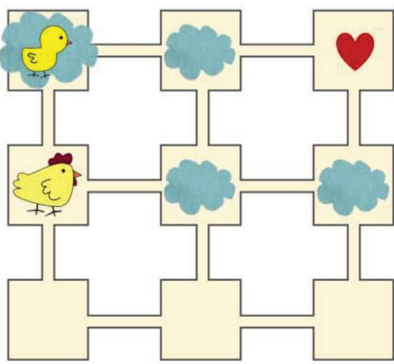
19



Lleva al pollito con su mamá:

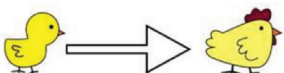
Ejemplo de significado:
 Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:



A		B		C		D	
	→		→		→		→
	→				→		→
	↓		↓		↓		↓
	←						←
	←						

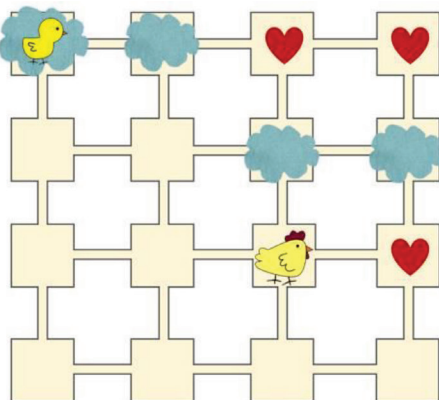
20



Lleva al pollito con su mamá:

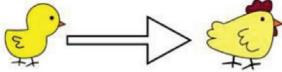
Ejemplo de significado:
 Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

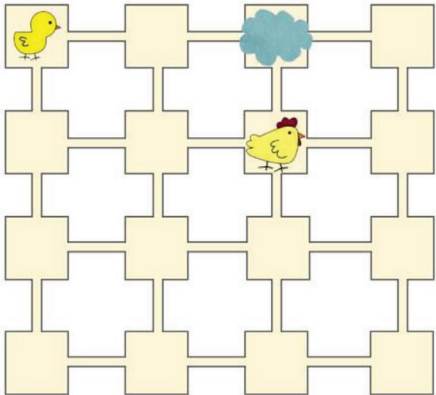


A		B		C		D	
	→		→		→		→
	→		→		→		→
	↓		↓		↓		↓
	↓		←		↓		↓

21



Lleva al pollito con su mamá:



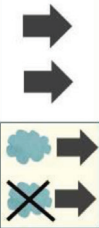
Ejemplo de significado:



Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

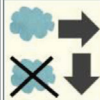
A



B



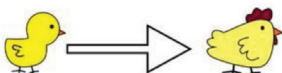
C



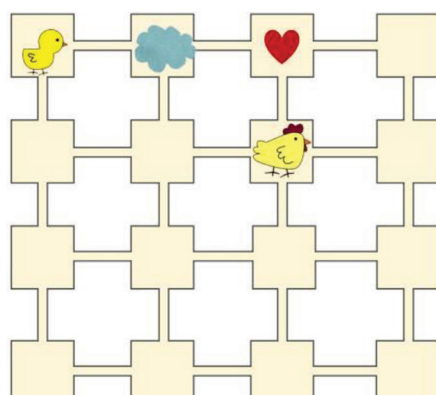
D



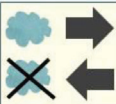
22



Lleva al pollito con su mamá:



Ejemplo de significado:



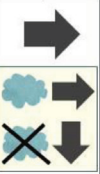
Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

A



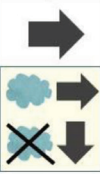
B



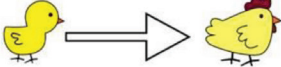
C



D



23



Lleva al pollito con su mamá:

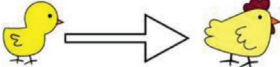
Ejemplo de significado:

 **Mientras que el pollito esté en casillas triángulo, siempre avanza a la derecha**

Marca la secuencia correcta:

A	B	C	D
			
			

24



Lleva al pollito con su mamá:

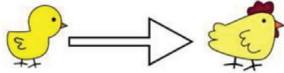
Ejemplo de significado:

 **Mientras que el pollito esté en casillas triángulo, siempre avanza a la derecha**

Marca la secuencia correcta:

A	B	C	D
			
			
			

25



Lleva al pollito con su mamá:

Ejemplo de significado:
 ➡ Mientras que el pollito esté en casillas triángulo, siempre avanza a la derecha

Marca la secuencia correcta:

A	B	C	D
 ➡	 ➡	 ⬇	 ⬇
 ⬇	 ⬇	 ➡	 ➡
 ⬅	 ⬆	 ⬇	 ➡
 ➡		 ➡	 ⬅

A.3. PROTOCOLO DE ACTUACIÓN DEL BCTT

Introducción

Debe realizarse un esfuerzo para incluir el aprendizaje de la programación en las aulas de educación primaria y educación infantil y, con ello, contribuir al desarrollo del pensamiento computacional [43], ya que es una habilidad fundamental que puede aplicarse a los problemas del día a día y se requiere para adaptarse a las necesidades de la sociedad del siglo XXI [26]. En paralelo con el aprendizaje de esta competencia, es esencial contar con instrumentos de evaluación. Desafortunadamente, la mayor parte de las investigaciones recientes se centran en la evaluación de las habilidades relacionadas con el pensamiento computacional en alumnos a partir de los 10 años [113]. Por este motivo, se ha desarrollado una herramienta de evaluación del pensamiento computacional, dirigido a edades tempranas (Educación Infantil y Educación Primaria): **BCTt (Beginners Computational Thinking Test)**.

En este documento se detalla el protocolo a seguir para la evaluación del pensamiento computacional (PC) en Educación Infantil y Educación Primaria mediante el test de pensamiento computacional para principiantes BCTt.

Descripción del BCTt

El test se ha adaptado a edades tempranas, con lo que incluye muy poco texto solamente como apoyo a los pictogramas que, por sí mismos, tienen la suficiente expresividad como para que un niño que aún no haya adquirido la competencia lectora pueda entender las preguntas.

El test consta de **25 preguntas**, que se dividen en **6 bloques**, cada uno de ellos aborda un tipo diferente de concepto computacional según la siguiente tabla:

Número de pregunta	Concepto computacional						
	BLOQUE 1	BLOQUE 2		BLOQUE 3	BLOQUE 4	BLOQUE 5	BLOQUE 6
	Secuencia	Bucle		Anidado	If-then	Condicional	
		Simple					
1 a 6	x						
7 a 11		x					
12 a 18				x			
19 a 20					x		
21 a 22						x	
23 a 25							x

Cada pregunta tiene **4 opciones posibles de respuesta** con sólo una de ellas correcta.

Aplicación del BCTt

En la etapa de Educación Primaria, el BCTt debe administrarse al grupo de alumnos a los que se desee evaluar en un tiempo medio de alrededor de **45-60 minutos**, ajustándose, preferentemente a un periodo de clase lectiva. Es posible que los alumnos de menor edad requieran de más tiempo para completar las preguntas y, asimismo, los de mayor edad podrán completarlo más rápidamente. Esta diferencia en el tiempo de resolución del test no se considerará significativa siempre que el tiempo de resolución del test sea **mayor de 30 minutos y menor de 90 minutos**.

Para evitar que las habilidades previas en el manejo de dispositivos móviles u ordenadores interfiera en los resultados del test, se administrarán en **formato de papel impreso**. El test se ha desarrollado teniendo en cuenta a las personas con daltonismo con lo que puede imprimirse en color (preferentemente) o en blanco y negro. En caso de que no se pueda realizar en formato de papel impreso, se puede realizar a través de un formulario on-line.

Cada alumno dispondrá de una copia individual del BCTt para responder. Si no es posible, podrá usarse una hoja de respuestas: en el caso de Educación Primaria, cada alumno responderá las preguntas del test en una hoja individual de respuestas; en el caso de Educación Infantil, será la persona que lo esté administrando quien vaya anotando las respuestas en la hoja individual de respuestas de cada alumno.

Previamente a la administración del test, se debe realizar una explicación oral con un ejemplo de cada uno de los seis bloques de preguntas que se incluyen en el test y que corresponden a cada uno de los tipos de concepto computacional a evaluar. Cada ejemplo de pregunta se debe mostrar en una pantalla o pizarra digital mientras se explica. En caso de no disponer de ella, estos ejemplos se explicarán mostrándolos en papel.

Cuando los alumnos ya estén realizando el test, es posible que puedan hacer preguntas. Se les podrá responder individualmente de forma similar a la explicación oral, indicando la mecánica de cada pregunta, pero nunca se les sugerirá o indicará cuál es la respuesta correcta. No se pueden dar pistas de cuál es la respuesta correcta a los alumnos. Se puede volver a usar los ejemplos de cada uno de los tipos para volver a explicar la dinámica de cada tipo de pregunta.

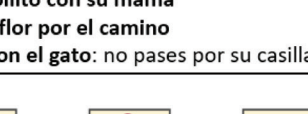
A continuación, se incluye un ejemplo de pregunta de cada bloque con orientaciones para su explicación oral. Las imágenes para los ejemplos se aportan también de forma aislada anexas a este documento.

BLOQUE 1: SECUENCIAS

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS



Lleva al pollito con su mamá
 Recoge la flor por el camino
 Cuidado con el gato: no pases por su casilla



Marca la secuencia correcta:

A	B	C	D
→	→	→	↓
↓	↓	→	↓
↓	→	↓	→
		↓	→

Ejemplo de explicación oral:

Se señalan los pictogramas que explican la pregunta: - "En esta pregunta hay que llevar al pollito con su mamá. Hay que recoger la flor por el camino y... ¡cuidado! no podéis pasar por la casilla donde está el gato. Probamos primero con la opción A:" – se recorre con el dedo la secuencia de flechas de la respuesta A: "Si elegís la respuesta A, el pollito iría a la derecha, luego abajo, luego abajo otra vez" – se señala con el dedo el recorrido que seguiría el pollito: "el pollito, haría: derecha (recoge la flor), abajo, abajo (no llega a su mamá); por lo que esta respuesta no es correcta".

Si es posible, se señalan con el dedo al mismo tiempo, las flechas de la opción A y el recorrido que sigue el pollito según esas flechas.

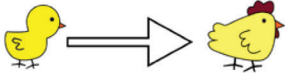
"Probamos ahora con la B" – se recorre con el dedo la secuencia de flechas de la respuesta B-: "Si elegís la respuesta B, el pollito iría a la derecha, luego abajo, luego derecha otra vez" – se señala con el dedo el recorrido que

“Probamos ahora con la D” – se recorre con el dedo la secuencia de flechas de la respuesta D-: “Si elegís la respuesta D, el pollito iría **UNA VEZ** a la derecha y **DOS VECES** hacia arriba” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: uno a la derecha y dos hacia arriba (llega a su mamá); por lo que esta respuesta ¡es correcta!”.

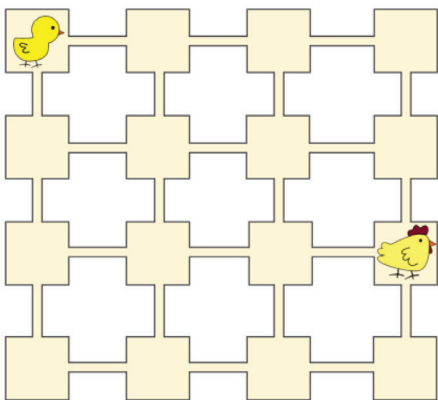
Si es posible, se señalan con el dedo al mismo tiempo, las flechas de la opción D y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar el número que indica la repetición del bucle.

BLOQUE 3: BUCLE ANIDADO

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



Lleva al pollito con su mamá:



Marca la secuencia correcta:

A

2x




B

1x




1x



C

2x




1x



D

2x




1x



Ejemplo de explicación oral:

Se señalan los pictogramas que explican la pregunta: - “En esta pregunta otra vez hay que llevar al pollito con su mamá, probamos primero con la A” – se recorre con el dedo la secuencia de flechas de la respuesta A-: “Si elegís la respuesta A, veis que las flechas están metidas dentro de un recuadro, **esas dos flechas** se repetirán, en este caso **DOS** veces, el pollito haría: derecha, abajo; derecha abajo.” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: derecha, abajo; derecha abajo, y... no llego hasta la casilla de su mamá, por lo que la respuesta no es correcta”.

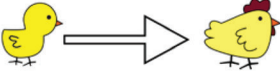
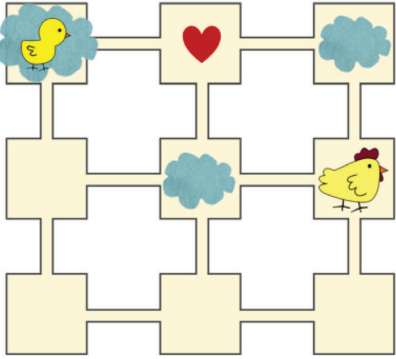
332

Si es posible, se señalan con el dedo al mismo tiempo, las flechas de la opción A y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar el número que indica la repetición del bucle, y decir rápidamente (para separar del resto de la explicación) y gesticulando, la repetición del interior del bucle.

“En este caso, **la respuesta correcta sería la C**” – se recorre con el dedo la secuencia de flechas de la respuesta C-: “Si elegís la respuesta C, veís que dos flechas están metidas dentro de un recuadro, **esas dos flechas** se repetirán, en este caso **DOS** veces, el pollito haría: derecha, abajo; derecha abajo. Finalmente iría **UNO** a la derecha” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: derecha, abajo; derecha abajo; y después, uno a la derecha (llega a su mamá); por lo que esta respuesta ¡es correcta!”.

Si es posible, se señalan con el dedo al mismo tiempo, las flechas de la opción C y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar el número que indica la repetición del bucle, y decir rápidamente (para separar del resto de la explicación) y gesticulando, la repetición del interior del bucle.

BLOQUE 4: IF-THEN

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN																					
 Lleva al pollito con su mamá:	Ejemplo de significado:  Si el pollito está en una casilla con nube, avanza una casilla a la derecha Marca la secuencia correcta:																				
	<table border="1"><thead><tr><th>A</th><th>B</th><th>C</th><th>D</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	A	B	C	D																
A	B	C	D																		
																					
																					
																					
																					

Ejemplo de explicación oral:

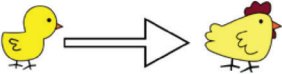
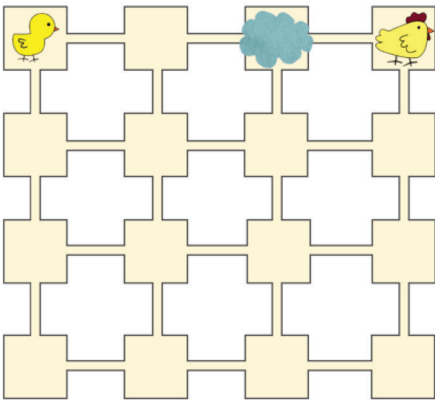
Se señalan los pictogramas que explican la pregunta: - “En esta pregunta otra vez hay que llevar al pollito con su mamá, probemos con la respuesta A” – se recorre con el dedo la secuencia de flechas de la respuesta A-: “Si elegís la respuesta A, si hay una nube en la casilla donde está el pollito, el pollito avanza una casilla a la derecha, después si el pollito está encima de un corazón, avanza hacia abajo, y después si está en una nube, avanza otra vez a la izquierda” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: nube-derecha, corazón-abajo; nube-izquierda. No llega hasta su mamá, por lo que esta respuesta no es correcta”.

Si es posible, se señalan con el dedo al mismo tiempo, las flechas de la opción C y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar que sólo si el pollito está encima del símbolo se avanzará una casilla en la dirección que la flecha indique, y el pollito no hará nada si no está encima de ese símbolo.

“En esta pregunta otra vez hay que llevar al pollito con su mamá, **la respuesta correcta sería la C**” – se recorre con el dedo la secuencia de flechas de la respuesta C-: “Si elegís la respuesta C, si hay una nube en la casilla donde está el pollito, el pollito avanza una casilla a la derecha, después si el pollito está encima de un corazón, avanza hacia abajo, y después si está en una nube, avanza otra vez a la derecha” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: nube-derecha, corazón-abajo; nube-derecha (llega a su mamá); por lo que esta respuesta ¡es correcta!”.

Si es posible, se señalan con el dedo al mismo tiempo, las flechas de la opción C y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar que sólo si el pollito está encima del símbolo se avanzará una casilla en la dirección que la flecha indique, y el pollito no hará nada si no está encima de ese símbolo.

BLOQUE 5: IF-THEN-ELSE

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE																	
	Ejemplo de significado:  Si el pollito está en una casilla con nube, avanza una casilla a la derecha, si no hay nube, avanza una casilla a la izquierda																
Lleva al pollito con su mamá:	Marca la secuencia correcta:																
	<table border="1"><tr><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	A	B	C	D												
A	B	C	D														
																	
																	

Ejemplo de explicación oral:

Se señalan los pictogramas que explican la pregunta: - “En esta pregunta otra vez hay que llevar al pollito con su mamá, **la respuesta correcta sería la A**” – se recorre con el dedo la secuencia de flechas de la respuesta A-: “Si elegís la respuesta A, primero, el pollito avanzará derecha, derecha, y después miramos el recuadro: si el pollito está encima de una nube, avanzará a la derecha y si no está encima de una nube, avanzará hacia abajo” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: derecha, derecha, y ahora está en una nube, con lo que va otra vez a la derecha (si no hubiera nube, iría hacia abajo), llega a su mamá, por lo que esta respuesta ¡es correcta!”.

En esta pregunta, es especialmente importante señalar con el dedo al mismo tiempo, las flechas de la opción A y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar que ocurre en el caso de estar encima del símbolo y qué ocurre si no está encima del símbolo.

“Vamos a probar con la respuesta D, que no es correcta” – se recorre con el dedo la secuencia de flechas de la respuesta D-: “Si elegís la respuesta D, primero, el pollito avanzará derecha, derecha, y después miramos el recuadro: si el pollito está encima de una nube, avanzará hacia abajo, y si no está

encima de una nube, avanzará hacia la derecha” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: derecha, derecha, y ahora está en una nube, con lo que avanza hacia abajo (si no hubiera nube, iría hacia la derecha), y no llega a su mamá, por lo que esta respuesta no es correcta”.

En esta pregunta, es especialmente importante señalar con el dedo al mismo tiempo, las flechas de la opción D y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar que ocurre en el caso de estar encima del símbolo y qué ocurre si no está encima del símbolo.

BLOQUE 6: WHILE

EJEMPLO PREGUNTA BLOQUE 6: WHILE

Lleva al pollito con su mamá:

Ejemplo de significado:

Mientras que el pollito esté en casillas triángulo, **siempre** avanza a la derecha

Marca la secuencia correcta:

A	B	C	D

Ejemplo de explicación oral:

Se señalan los pictogramas que explican la pregunta: - “En esta pregunta otra vez hay que llevar al pollito con su mamá, **la respuesta correcta sería la A**” – se recorre con el dedo la secuencia de flechas de la respuesta A-: “Si elegís la respuesta A, SIEMPRE que el pollito esté en una casilla cuadrada, el pollito avanzará a la derecha, SIEMPRE que esté en una casilla con un triángulo, avanzará a la derecha, y SIEMPRE que esté en una casilla con un círculo, a la izquierda.” – se señala con el dedo el recorrido que seguiría el pollito-: “el pollito, haría: está en un cuadrado: siempre a la derecha, y ahora está en un

triángulo: siempre abajo, y ahora está en un círculo: siempre a la izquierda, llega a su mamá, por lo que esta respuesta ¡es correcta!".

En esta pregunta, es especialmente importante señalar con el dedo al mismo tiempo, las flechas de la opción A y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar qué ocurre en el caso de estar encima del símbolo, que es SIEMPRE la misma acción.

"Vamos a probar con la B, que no es correcta" – se recorre con el dedo la secuencia de flechas de la respuesta B-: "Si elegís la respuesta B, SIEMPRE que el pollito esté en una casilla cuadrada, el pollito avanzará a la derecha, y SIEMPRE que esté en una casilla con un triángulo, avanzará a la derecha." – se señala con el dedo el recorrido que seguiría el pollito-: "el pollito, haría: está en un cuadrado: siempre a la derecha, y ahora está en un triángulo: siempre abajo, y ahora ha llegado a una casilla con un círculo, pero ya no haría nada más, así que la respuesta no es correcta".

En esta pregunta, es especialmente importante señalar con el dedo al mismo tiempo, las flechas de la opción B y el recorrido que sigue el pollito según esas flechas. En este caso, es recomendable enfatizar qué ocurre en el caso de estar encima del símbolo, que es SIEMPRE la misma acción.

Recogida de datos

Preferiblemente antes de la explicación oral y la administración del test a los alumnos, debe rellenarse la primera página de cada test con los datos anónimos del alumno. Algunos de estos datos es posible que deba rellenarlos el responsable de la experiencia porque los niños sean demasiado pequeños para rellenarlos ellos mismos. Estos datos son:

- **Responsable de la experiencia:** nombre de la persona encargada de la explicación oral y administración del test
- **Fecha:** fecha en la que se administra el test
- **Identificador del alumno:** identificador que el responsable pueda posteriormente corresponder con el alumno concreto
- **Fecha de nacimiento:** fecha de nacimiento del alumno
- **Etapas educativas:** indicar si la etapa educativa es Educación Infantil o Educación Primaria
- **Curso escolar:** curso escolar del alumno

- **Género:** género del alumno
- **Comentarios:** se especificará si existe algún diagnóstico del alumno, como por ejemplo si es una persona con necesidades educativas especiales, altas capacidades, dislexia, discalculia, autismo, discapacidad visual, discapacidad auditiva, discapacidad motora, discapacidad intelectual, etc.

Adicionalmente, si es posible, se pueden adjuntar al documento las **calificaciones del alumno** o cualquier otro dato que pueda ser relevante para la investigación. Si se adjuntan las calificaciones, preferiblemente se recopilarán de las siguientes asignaturas:

Educación Infantil:

- Matemáticas
- Lectura
- Escritura

Educación Primaria:

- Matemáticas
- Lengua
- Música
- Arte
- Educación física

Hoja de respuestas

Las respuestas correctas del BCTt son las siguientes:

Numero de pregunta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Respuesta correcta	A	D	C	B	B	D	B	A	D	C	B	A	C	B	C	D	A	C	A	C	D	B	A	B	C

Se adjunta como anexo la hoja de respuestas en el caso de necesitarse para los alumnos.

A.4. EJEMPLOS

Ejemplo pregunta bloque 1: SECUENCIAS

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS																					
 Lleva al pollito con su mamá Recoge la flor por el camino Cuidado con el gato: no pases por su casilla	Marca la secuencia correcta: <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>→</td><td>→</td><td>→</td><td>↓</td></tr><tr><td>↓</td><td>↓</td><td>→</td><td>↓</td></tr><tr><td>↓</td><td>→</td><td>↓</td><td>→</td></tr><tr><td></td><td></td><td>↓</td><td>→</td></tr></table>	A	B	C	D	→	→	→	↓	↓	↓	→	↓	↓	→	↓	→			↓	→
A	B	C	D																		
→	→	→	↓																		
↓	↓	→	↓																		
↓	→	↓	→																		
		↓	→																		

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS																					
 Lleva al pollito con su mamá Recoge la flor por el camino Cuidado con el gato: no pases por su casilla	OPCIÓN INCORRECTA Marca la secuencia correcta: <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>→</td><td>→</td><td>→</td><td>↓</td></tr><tr><td>↓</td><td>↓</td><td>→</td><td>↓</td></tr><tr><td>↓</td><td>→</td><td>↓</td><td>→</td></tr><tr><td></td><td></td><td>↓</td><td>→</td></tr></table>	A	B	C	D	→	→	→	↓	↓	↓	→	↓	↓	→	↓	→			↓	→
A	B	C	D																		
→	→	→	↓																		
↓	↓	→	↓																		
↓	→	↓	→																		
		↓	→																		

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS

Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

OPCIÓN INCORRECTA

Marca la secuencia correcta:

A	B	C	D
→	→	→	↓
↓	↓	→	↓
↓	→	↓	→

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS

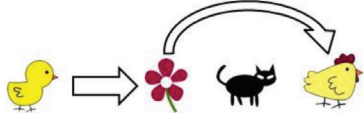
Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

OPCIÓN INCORRECTA

Marca la secuencia correcta:

A	B	C	D
→	→	→	↓
↓	↓	→	↓
↓	→	↓	→

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS



Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

OPCIÓN INCORRECTA

Marca la secuencia correcta:

A

→

↓

↓

✗

B

→

↓

→

C

→

→

↓

↓

D

↓

↓

→

→

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS



Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

OPCIÓN CORRECTA

Marca la secuencia correcta:

A

→

↓

↓

B

→

↓

→

C

→

→

↓

↓

D

↓

↓

→

→

341

EJEMPLO PREGUNTA BLOQUE 1: SECUENCIAS

Lleva al pollito con su mamá

Recoge la flor por el camino

Cuidado con el gato: no pases por su casilla

OPCIÓN
CORRECTA

Marca la secuencia correcta:

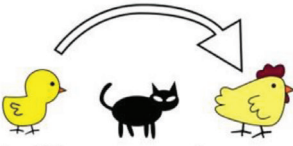
A

B

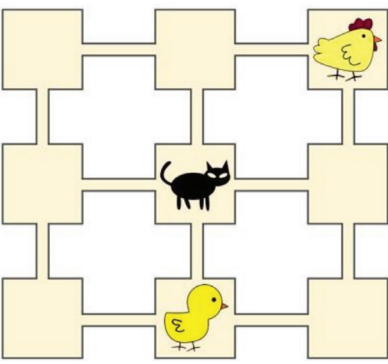
C

D

Ejemplo pregunta bloque 2: BUCLE SIMPLE

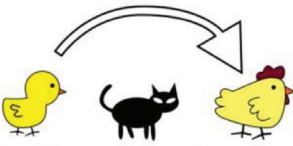


Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

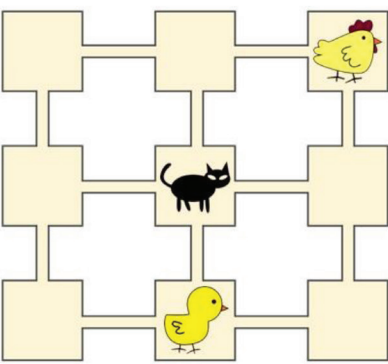


Marca la secuencia correcta:

A		B		C		D	
1x	↑	1x	→	2x	↑	1x	→
2x	→	1x	↑	2x	→	2x	↑
		1x	→				



Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla



OPCIÓN INCORRECTA

Marca la secuencia correcta:

A		B		C		D	
1x	↑	1x	→	2x	↑	1x	→
2x	→	1x	↑	2x	→	2x	↑
		1x	→				

EJEMPLO DE PREGUNTA BLOQUE 2: BUCLE SIMPLE

Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

OPCIÓN INCORRECTA

Marca la secuencia correcta:

A	B	C	D
1x ↑	1x →	2x ↑	1x →
2x →	1x ↑	2x →	2x ↑
1x →			

EJEMPLO DE PREGUNTA BLOQUE 2: BUCLE SIMPLE

Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

OPCIÓN CORRECTA

Marca la secuencia correcta:

A	B	C	D
1x ↑	1x →	2x ↑	1x →
2x →	1x ↑	2x →	2x ↑
	1x →		

EJEMPLO DE PREGUNTA BLOQUE 2: BUCLE SIMPLE

Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

OPCIÓN CORRECTA

Marca la secuencia correcta:

A	B	C	D
1x ↑	1x →	2x ↑	1x →
2x →	1x ↑	2x →	2x ↑
	1x →		

EJEMPLO DE PREGUNTA BLOQUE 2: BUCLE SIMPLE

Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

OPCIÓN CORRECTA

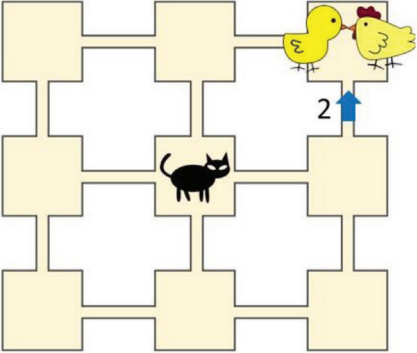
Marca la secuencia correcta:

A	B	C	D
1x ↑	1x →	2x ↑	1x →
2x →	1x ↑	2x →	2x ↑
	1x →		

EJEMPLO DE PREGUNTA BLOQUE 2: BUCLE SIMPLE



Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla



OPCIÓN CORRECTA

Marca la secuencia correcta:

A		B		C		D	
1x	↑	1x	→	2x	↑	1x	→
2x	→	1x	↑	2x	→	2x	↑
		1x	→				

✓

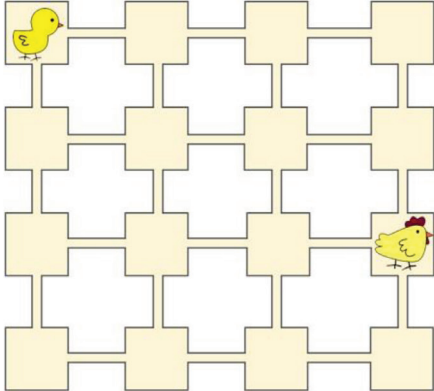
347

Ejemplo pregunta bloque 3: BUCLE ANIDADO

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



Lleva al pollito con su mamá:



Marca la secuencia correcta:

A

2x




B

1x




1x



C

2x




1x



D

2x



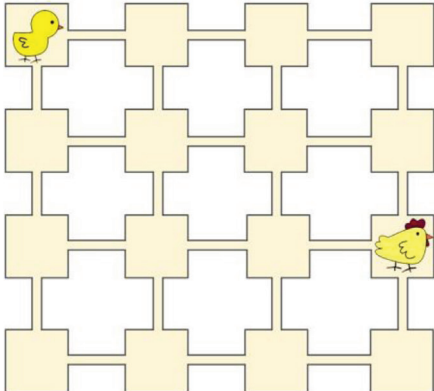

1x



EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



Lleva al pollito con su mamá:



Marca la secuencia correcta:

A

2x




B

1x




1x



C

2x




1x



D

2x



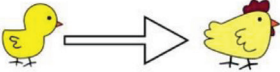

1x



OPCIÓN INCORRECTA

348

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



Lleva al pollito con su mamá:

Marca la secuencia correcta:

OPCIÓN INCORRECTA

A

2x



B

1x



1x



C

2x



1x



D

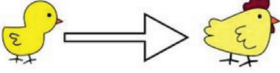
2x



1x



EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



Lleva al pollito con su mamá:

Marca la secuencia correcta:

OPCIÓN INCORRECTA

A

2x



B

1x



1x



C

2x



1x



D

2x



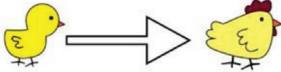
1x





349

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



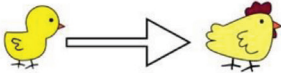
Lleva al pollito con su mamá:

OPCIÓN CORRECTA

Marca la secuencia correcta:

A		B		C		D	
2x		1x 1x	2x 1x	2x 1x			

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



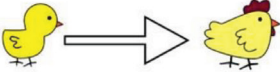
Lleva al pollito con su mamá:

OPCIÓN CORRECTA

Marca la secuencia correcta:

A		B		C		D	
2x		1x 1x	2x 1x	2x 1x			

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



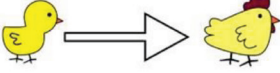
Lleva al pollito con su mamá:

OPCIÓN CORRECTA

Marca la secuencia correcta:

A		B		C		D	
2x		1x		2x		2x	
		1x		1x		1x	

EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



Lleva al pollito con su mamá:

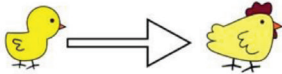
OPCIÓN CORRECTA

Marca la secuencia correcta:

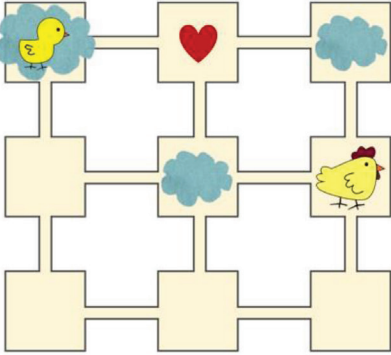
A		B		C		D	
2x		1x		2x		2x	
		1x		1x		1x	

Ejemplo pregunta bloque 4: IF-THEN

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:



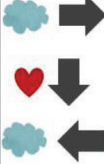
Ejemplo de significado:



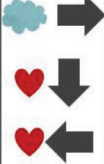
Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

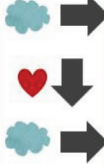
A



B



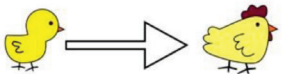
C



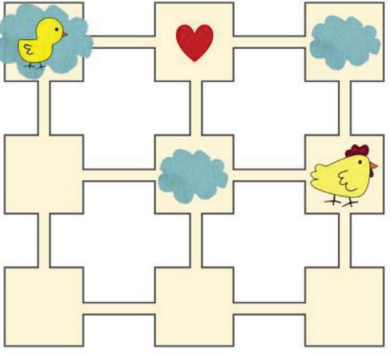
D



EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:



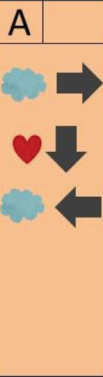
Ejemplo de significado:



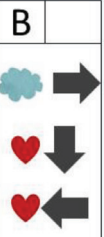
Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

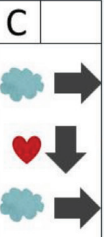
A



B



C



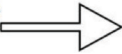
D



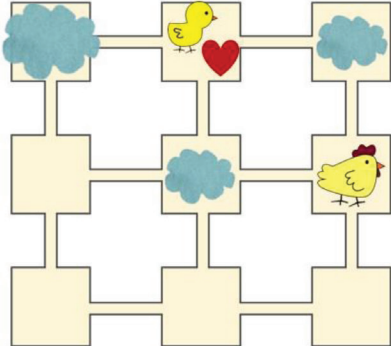
OPCIÓN INCORRECTA

352

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:



Ejemplo de significado:
 → Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

A

B

C

D





















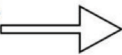




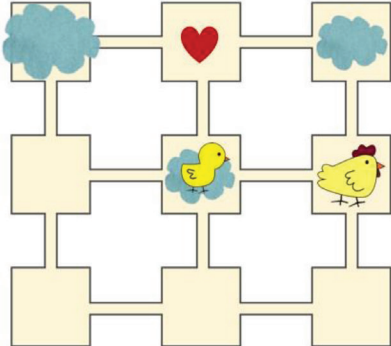


OPCIÓN
INCORRECTA

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:



Ejemplo de significado:
 → Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

A

B

C

D





















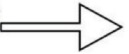






OPCIÓN
INCORRECTA

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:

Ejemplo de significado:



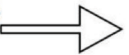
Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

OPCIÓN INCORRECTA

A		B		C		D	
							
							
							
							

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:

Ejemplo de significado:



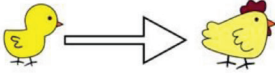
Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

OPCIÓN CORRECTA

A		B		C		D	
							
							
							

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



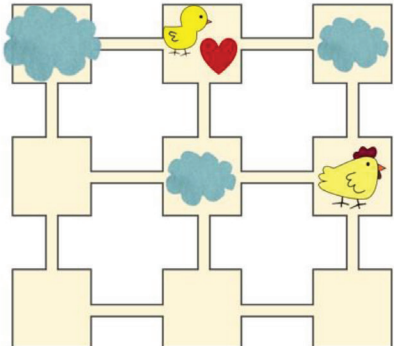
Lleva al pollito con su mamá:

Ejemplo de significado:

 Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

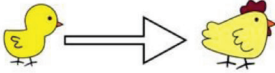
Marca la secuencia correcta:

A	B	C	D
			
			
			



OPCIÓN CORRECTA

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



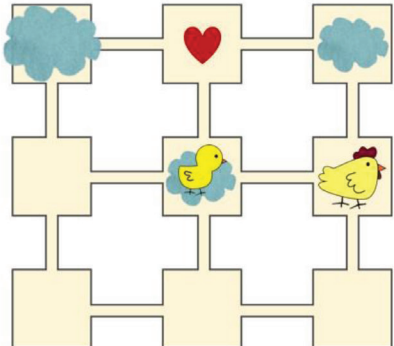
Lleva al pollito con su mamá:

Ejemplo de significado:

 Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

A	B	C	D
			
			
			

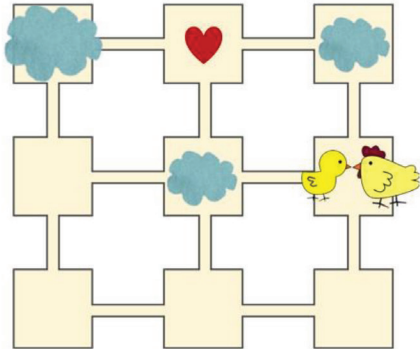


OPCIÓN CORRECTA

EJEMPLO PREGUNTA BLOQUE 4: IF-THEN



Lleva al pollito con su mamá:



Ejemplo de significado:



Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:

A







B







C







D



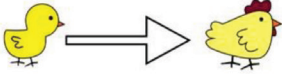
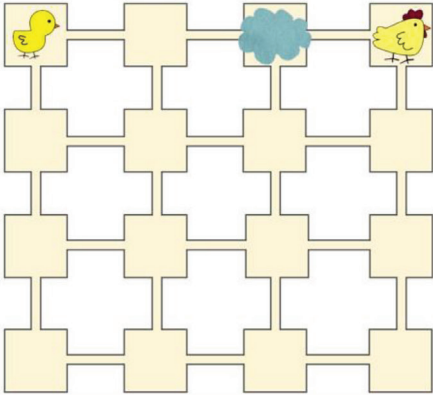
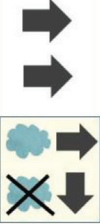
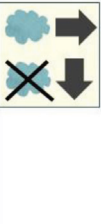
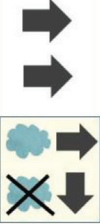
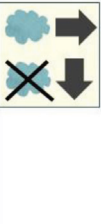
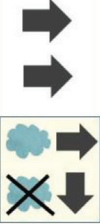
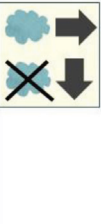


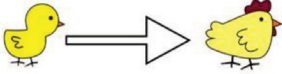
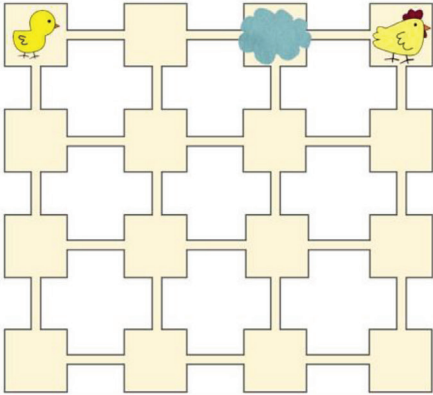
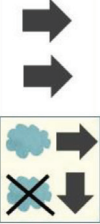
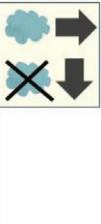
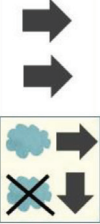
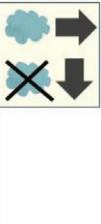
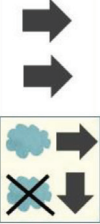
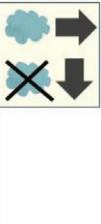




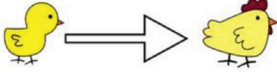
OPCIÓN CORRECTA

Ejemplo pregunta bloque 5: IF-THEN-ELSE

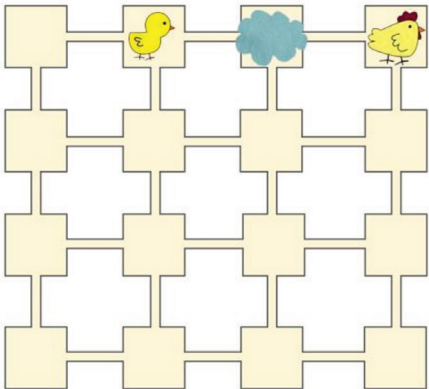
EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE									
	Ejemplo de significado:  Si el pollito está en una casilla con nube, avanza una casilla a la derecha, si no hay nube, avanza una casilla a la izquierda								
Lleva al pollito con su mamá:	Marca la secuencia correcta:								
	<table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	A	B	C	D				
A	B	C	D						
									

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE									
	Ejemplo de significado:  Si el pollito está en una casilla con nube, avanza una casilla a la derecha, si no hay nube, avanza una casilla a la izquierda								
Lleva al pollito con su mamá:	Marca la secuencia correcta:								
	<table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	A	B	C	D				
A	B	C	D						
									

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE



Lleva al pollito con su mamá:



A









B






C








D








Ejemplo de significado:



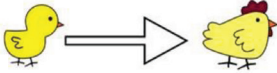



Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

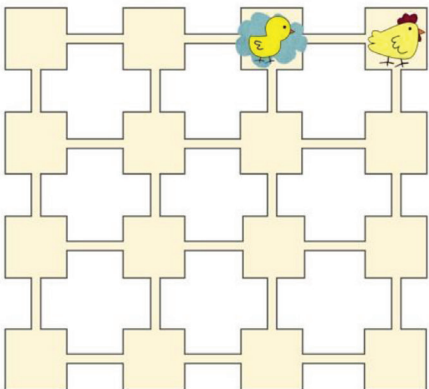
Marca la secuencia correcta:

OPCIÓN CORRECTA

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE



Lleva al pollito con su mamá:



A









B






C








D








Ejemplo de significado:





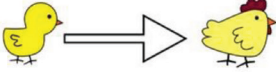

Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

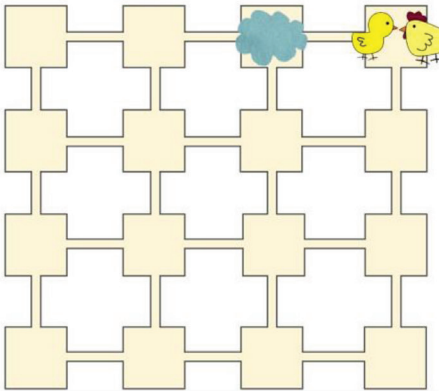
OPCIÓN CORRECTA

358

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE



Lleva al pollito con su mamá:



OPCIÓN CORRECTA

Ejemplo de significado:

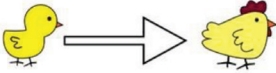


Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

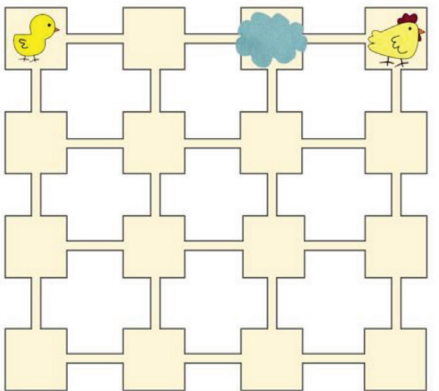
Marca la secuencia correcta:

A	B	C	D
			
			
			

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE



Lleva al pollito con su mamá:



OPCIÓN INCORRECTA

Ejemplo de significado:

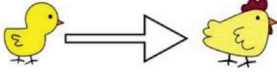


Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

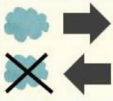
A	B	C	D
			
			

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE



Lleva al pollito con su mamá:

Ejemplo de significado:



Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

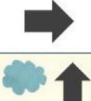
A



B



C

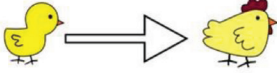


D



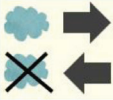
OPCIÓN INCORRECTA

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE



Lleva al pollito con su mamá:

Ejemplo de significado:



Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

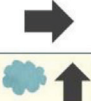
A



B



C



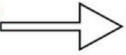
D



OPCIÓN INCORRECTA

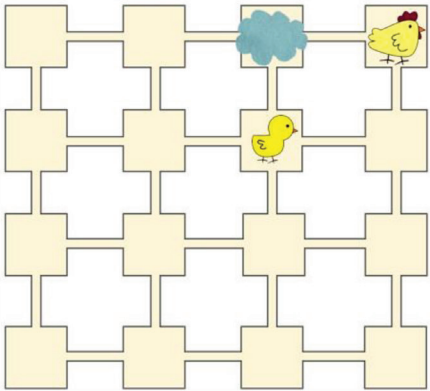
360

EJEMPLO PREGUNTA BLOQUE 5: IF-THEN-ELSE





Lleva al pollito con su mamá:



Ejemplo de significado:





Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha, si no hay nube, avanza una casilla a la izquierda

Marca la secuencia correcta:

A



B



C



D



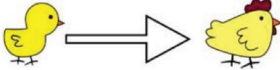
OPCIÓN INCORRECTA



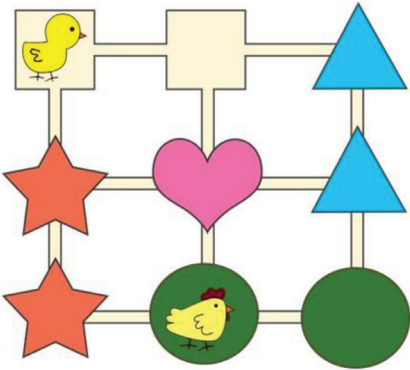
361

Ejemplo pregunta bloque 6: WHILE

EJEMPLO PREGUNTA BLOQUE 6: WHILE



Lleva al pollito con su mamá:



Ejemplo de significado:

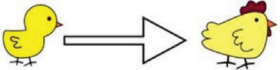


Mientras que el pollito esté en casillas triángulo, siempre avanza a la derecha

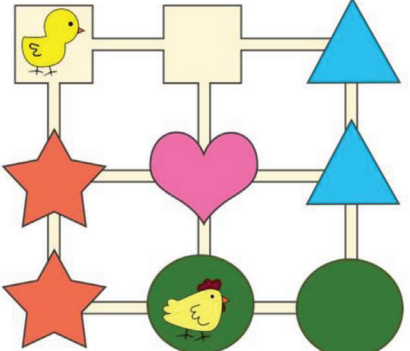
Marca la secuencia correcta:

A		B		C		D	

EJEMPLO PREGUNTA BLOQUE 6: WHILE



Lleva al pollito con su mamá:



Ejemplo de significado:



Mientras que el pollito esté en casillas triángulo, siempre avanza a la derecha

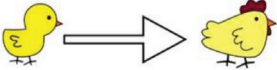
Marca la secuencia correcta:

A		B		C		D	

OPCIÓN CORRECTA

362

EJEMPLO PREGUNTA BLOQUE 6: WHILE



Lleva al pollito con su mamá:

Ejemplo de significado:

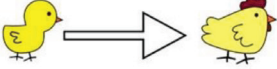
 **Mientras que** el pollito esté en casillas triángulo, **siempre** avanza a la derecha

Marca la secuencia correcta:

OPCIÓN CORRECTA

A	B	C	D
			
			
			

EJEMPLO PREGUNTA BLOQUE 6: WHILE



Lleva al pollito con su mamá:

Ejemplo de significado:

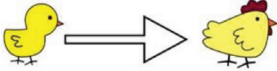
 **Mientras que** el pollito esté en casillas triángulo, **siempre** avanza a la derecha

Marca la secuencia correcta:

OPCIÓN CORRECTA

A	B	C	D
			
			
			

EJEMPLO PREGUNTA BLOQUE 6: WHILE



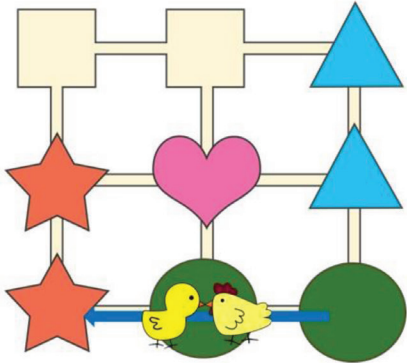
Lleva al pollito con su mamá:

Ejemplo de significado:

 **Mientras que** el pollito esté en casillas triángulo, **siempre** avanza a la derecha

Marca la secuencia correcta:

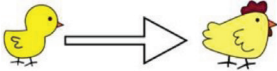
OPCIÓN CORRECTA



A		B		C		D	
							
							
							



EJEMPLO PREGUNTA BLOQUE 6: WHILE



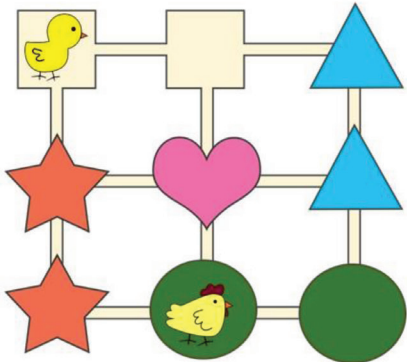
Lleva al pollito con su mamá:

Ejemplo de significado:

 **Mientras que** el pollito esté en casillas triángulo, **siempre** avanza a la derecha

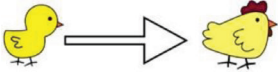
Marca la secuencia correcta:

OPCIÓN INCORRECTA



A		B		C		D	
							
							
							

EJEMPLO PREGUNTA BLOQUE 6: WHILE



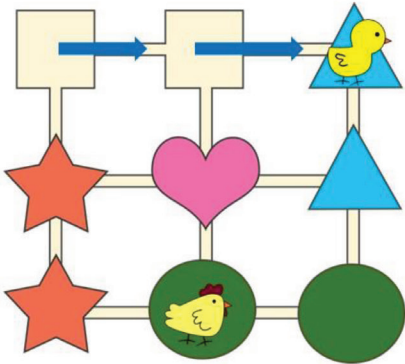
Lleva al pollito con su mamá:

Ejemplo de significado:

 **Mientras que** el pollito esté en casillas triángulo, **siempre** avanza a la derecha

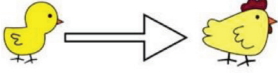
Marca la secuencia correcta:

OPCIÓN INCORRECTA



A	B	C	D
			
			
			

EJEMPLO PREGUNTA BLOQUE 6: WHILE



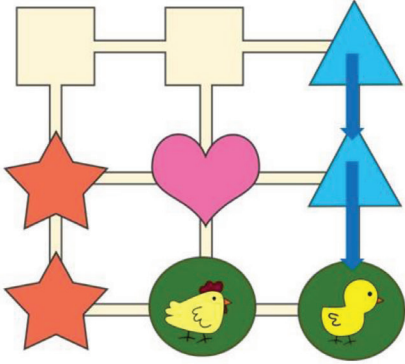
Lleva al pollito con su mamá:

Ejemplo de significado:

 **Mientras que** el pollito esté en casillas triángulo, **siempre** avanza a la derecha

Marca la secuencia correcta:

OPCIÓN INCORRECTA



A	B	C	D
			
			
			

B. BAC

En los siguientes apartados se incluye el protocolo de actuación para una experiencia en entornos escolares con BAC y el esquema de la base de datos asociada a BAC.

B.1. Protocolo de actuación

Objetivo de la experiencia

Oportunidad y expectativas. ¿Por qué?

Se asume, como conclusión del estudio del arte, que debe realizarse un esfuerzo para incluir el aprendizaje de la programación en las aulas de educación primaria y, con ello, contribuir al desarrollo del pensamiento computacional. Son muchos los beneficios que tiene aprender programación informática y cada vez es más sencillo e intuitivo su aprendizaje, por lo que sería muy interesante que se integrara como asignatura con entidad propia dentro del currículo educativo desde primaria, al igual que ya están haciendo otros países. El proyecto se basa en la necesidad de enseñar a los estudiantes de primaria las bases para el aprendizaje de lenguajes de programación informática.

Se detectan carencias en la actualidad para la enseñanza de esta competencia. Las dos carencias en las que se enfoca este proyecto son las siguientes:

- No hay guías concretas que estén respaldadas por la suficiente investigación, para determinar, según la edad del alumno, cuáles son los conceptos que deben enseñarse, son relevantes o pueden ser alcanzables.
- No hay investigación suficiente para determinar cuál es la metodología adecuada para este tipo de aprendizaje.

Se proponen dos objetivos:

OBJETIVO A

Se plantea la aplicación y evaluación de metodologías que engloban el uso de la tecnología y la aplicación de técnicas de juego libre, mediante el uso de videojuegos como herramienta. Se incluyen algunas mejoras de detalle sobre las aplicaciones que ya existen al respecto y se pretende evaluar su relevancia. En concreto, se pretende evaluar si el aprendizaje colaborativo supone una mejora frente al aprendizaje individual utilizando este tipo de herramientas de enseñanza. También se pretende evaluar si la incorpora-

ción de transiciones entre casillas de juego mejora la comprensión de los problemas.

Hipótesis 1: En el modo colaborativo los alumnos aprenden por las interacciones de sus compañeros (efecto motivador), porque cuando están jugando están enseñándose entre iguales y porque resuelven los problemas que le plantea el compañero entre un aprendizaje entre iguales. El modo colaborativo lleva asociado un aprendizaje más significativo que cuando se trabaja con una metodología individual.

Hipótesis 2: La incorporación de transiciones entre las casillas de un puzle mejora la comprensión de los problemas ya que se entiende mejor cuándo, desde dónde y hasta dónde se realiza un paso. Además, se realiza una correspondencia con el concepto de “diagrama de estados” ampliamente utilizado en el lenguaje computacional.

OBJETIVO B

De forma lateral y como producto de esta experiencia, se obtendrán datos sobre los niveles que los niños han realizado y los resultados obtenidos en los mismos. Estos resultados servirán para establecer dependiendo de cada uno de los niveles los mínimos que los niños supieron realizar y cuáles son los conceptos que pueden aprender en una determinada etapa. Sin embargo, estos datos no aseguran que no puedan llegar más allá y abordar conceptos más complejos. Esto se debe a que como los niños escogen el nivel de juego, puede darse la situación que se queden en niveles más cómodos y fáciles para poder ganar más puntos. Al menos, los datos obtenidos en esta experiencia servirán para posteriormente poder hacer una calibración en los objetivos de enseñanza en cuanto a la programación en Educación Primaria (cuáles son los límites de edad para el aprendizaje de cada concepto) y establecer una guía sólida de enseñanza de la programación clasificada por concepto y edad recomendada para la adquisición de ese concepto.

Procedimiento ¿Cómo?

Para la consecución de los objetivos, se han desarrollado diferentes herramientas. Varios cuestionarios, que se detallan más adelante, un test de pensamiento computacional validado por expertos y un juego educativo (*antCode*).

Esta experiencia se centra en el objetivo A, prestando especial atención a la hipótesis 1 mediante la evaluación del aprendizaje y las diferencias existentes en el uso de una metodología colaborativa vs una metodología individual. Para poder medir el aprendizaje de los alumnos, el instrumento de medida principal es el test de pensamiento computacional que se pasa antes y después de

la intervención. Además, se pasará de nuevo el test de pensamiento computacional después de un mes de finalización de la experiencia para poder comprobar si el aprendizaje es duradero en el tiempo.

En cuanto a la hipótesis 2 del objetivo A, se realiza una versión diferente del test de pensamiento computacional en la que se eliminan las transiciones para evaluar cuál de los dos es más sencillo de comprender por los alumnos.

Adicionalmente se recopila más información en distintos cuestionarios para poder obtener información de cada sujeto y relacionarla con los resultados del experimento. Se obtienen también datos, en segundo plano, significativos para el objetivo B. Para ello, se pretende realizar un experimento local controlado como se describe en la siguiente sección.

El juego tiene dos modos de publicación, uno local y otro masivo. En el modo masivo, se subirá a la tienda online de forma que pueda ser jugado por un número elevado de personas que aportarán su edad. Con ello se pretenden conseguir estadísticas masivas para el posterior tratamiento de datos que se espera que aporten luz para el objetivo B. En el modo local, se pueden recoger datos orientados tanto al objetivo A como al objetivo B. Por otro lado, *antCode* tiene un modo de juego colaborativo y un modo de juego de retos fijos. Además, cuenta con seis niveles de juego que corresponden a diferentes conceptos computacionales.

A continuación, se incluye una tabla de correspondencia entre cada nivel del juego y el concepto computacional asociado:

Nivel	Concepto computacional					
	Secuencias		Bucles		Condicional (while)	
	Simple	Complejo	Simple	Complejo (agrupado)	Simple	Complejo
1	x					
2		x				
3			x			
4				x		
5					x	
6						x

Experimento local:

El experimento se centra en las dos hipótesis del objetivo A.

Se plantea la realización de experiencias a lo largo de tres etapas educativas en primaria: 1º-2º primaria; 3º-4º primaria y 5º-6º primaria.

Experiencia para contrastar la hipótesis 1:

En cada etapa y experiencia, se realizará:

1. Un test de pensamiento computacional (cuestionario 12 con transiciones), orientado a conceptos de programación en el que se pretende evaluar el estado inicial de los alumnos respecto a esta competencia. También se realizarán varios test más de motivación y otros de aspectos generales (se detallan más adelante).
2. A continuación, algunos los grupos de alumnos jugarán con *antCode* (en el modo que se haya establecido para la experiencia en particular) que recogerá todos los datos concretos y detalles de juego para su valoración posterior. La primera ocasión que se juegue, se deberán explicar los niveles al niño en orden. Sin embargo, una vez que ya se conozca, se debe dejar a los niños seleccionar a qué juego quieren jugar mientras estén en las sesiones de juego.
3. Se vuelven a realizar test de pensamiento computacional y cuestionarios para determinar si ha habido cambios o mejoras en los objetivos de estudio.
4. Al cabo de un mes de la finalización de todas las experiencias, se vuelve a pasar el post-test de nuevo con el objetivo de medir si el aprendizaje es duradero.

Se pretende la realización de dos experiencias principales en cada etapa educativa: **experiencia colaborativa** en la que juegan dos niños al mismo tiempo, uno de ellos propone el problema y otro resuelve; y **experiencia de retos fijos**, en la cual sólo se resuelven problemas (no se proponen – el videojuego plantea retos al niño que está interactuando con él). De esta forma se pretende evaluar el aprendizaje en función de la metodología usada y ver qué metodología funciona mejor.

Experiencia para contrastar la hipótesis 2:

Se pretende la realización del test de pensamiento computacional en sus dos versiones (con y sin transiciones) en dos grupos distintos, para valorar si se produce una mejora incorporando transiciones.

Resultados ¿Qué?

Se pretende obtener los siguientes resultados principales:

Resultados principales esperados para el objetivo A:

- En cuanto a los resultados sobre la **adecuación de los juegos como herramienta educativa**, se pretende obtener respuestas a las siguientes preguntas: ¿ayudan en la adquisición de conceptos computacionales y ayudan en el desarrollo del pensamiento computacional? ¿Mejora la motivación para el aprendizaje de la competencia con esta metodología?, etc.
- En cuanto a los resultados sobre la **metodología usada**, se pretende obtener respuestas a las siguientes preguntas: ¿es más efectivo el aprendizaje colaborativo en cuanto a resultados y/o motivación frente a la metodología individual? ¿Qué bondades y desventajas tienen cada una de estas metodologías en las distintas etapas?
- En cuanto a otros detalles metodológicos de este tipo de herramientas, se pretende analizar los datos y extraer conclusiones sobre la adecuación de los diferentes detalles de implementación basándose en dificultades que puedan encontrar los alumnos. Para ello, la herramienta guarda estadísticas en cuanto a tiempos y en cuanto a estrategias, para la propuesta y resolución de los diferentes problemas, que se analizarán posteriormente.
- En cuanto a los detalles de enseñanza implementados: ¿es adecuada la incorporación de transiciones como ayuda para la resolución de problemas computacionales?, etc. Para ello, se propone también la realización de test diferentes en sendos grupos, de forma que pueda medirse la adecuación de este detalle de implementación.

Resultados principales esperados para el objetivo B:

En cuanto a los resultados sobre la dificultad de cada concepto, se pretenden obtener datos que ayuden a aclarar las siguientes preguntas: ¿este concepto computacional debe enseñarse en esta edad concreta? ¿Cómo es la curva de aprendizaje de este concepto para cada edad? ¿Se recomienda la enseñanza de este concepto de forma temprana?, etc.

Metodología

Roles

Se definen a continuación los diferentes roles que tomarán los participantes de la experiencia:

- Participante: cualquier persona que participe de alguna forma en la experiencia
- Responsable: Es el encargado de aplicar la experiencia de forma práctica, interactuando con los alumnos. Será el encargado de repartir y explicar el test y los cuestionarios a los alumnos, explicando cada tipo de pregunta con un ejemplo gráfico y resolviendo dudas si fuera necesario. Llevará a cabo la experiencia de juego con los alumnos, aportando explicaciones si fuera necesario en cada nivel de juego. En algunos casos, si los alumnos son muy pequeños, es posible que no puedan rellenar el cuestionario y deba hacerlo esta persona por ellos.
- Profesor-tutor: es el profesor que ejerce como tutor de la clase de cada alumno en particular, en el centro en el que tiene lugar la experiencia.
- Alumno o jugador: se refiere al niño o niña que es objeto de la experiencia.

Pasos a realizar

A continuación, se detallan cuáles serían los pasos a realizar para una experiencia local en una sola etapa educativa, llevada a cabo por el responsable de la experiencia, es decir, el encargado de llevar a cabo la experiencia con los alumnos. En algunos casos, si los alumnos son muy pequeños (primero, segundo y tercero de primaria), es posible que no puedan rellenar el cuestionario y deba hacerlo esta persona por ellos:

Contraste de hipótesis 1 (experiencia principal, evaluación de modo de juego):

- **Realización de cuestionarios para el alumno y el profesor-tutor:**
- Cuestionario 1: Datos generales. Rellenar por **el alumno o responsable** ANTES de la experiencia.
- Cuestionario 2: Datos motivacionales previos. Rellenar por el **alumno o responsable** ANTES de la experiencia.

- Cuestionario 6: Rellenar por el **profesor-tutor** de los alumnos ANTES de la experiencia.
- Cuestionario 4: Datos objetivos sobre el alumno. Rellenar por el **responsable** ANTES (algunos datos se rellenan DESPUÉS).
- Cuestionario 5: Datos objetivos sobre el centro escolar. Rellenar por el **responsable**. ANTES (algunos datos se rellenan DESPUÉS).

- Realización del test de Pensamiento Computacional (CT):

Cuestionario 12 (TEST de CT): Se realiza el mismo test de CT una vez ANTES de la experiencia y otra vez DESPUÉS de la experiencia. A rellenas por **niños**.

- Experiencia con el juego:

Se plantean dos grupos de alumnos divididos de forma aleatoria. Uno de los grupos realizará la experiencia con el juego en modo colaborativo y otro de los grupos en modo fijo. Los dos grupos deben ser homogéneos. Es decir, tienen que tener el mismo número de alumnos de nivel alto, nivel medio y nivel bajo. Ambos grupos tienen que tener el mismo número de alumnos. A modo de ejemplo, si tenemos dos clases de 26 alumnos y NO existen sesgos (ni por género, ni por rendimiento de los alumnos – se comprobará con los pre-test), una de las clases seguirá la metodología individual y otra clase seguirá la metodología colaborativa. Si existen alumnos de necesidades educativas especiales solo en una clase, se recomienda que vayan en la metodología colaborativa.

GRUPO 1: es el grupo que hará la experiencia individual (modo de retos fijos):

- Realización de cuestionarios para el alumno y el profesor-tutor:

- Cuestionario 3: Datos motivacionales posteriores. Rellenar por el **alumno o responsable** DESPUÉS de la experiencia
- Cuestionario 7: Rellenar por el **profesor-tutor** de los alumnos DESPUÉS de la experiencia.
- Rellenar los datos de los cuestionarios 4 y 5 que no pudieron rellenas en el paso A.

- Realización del test de Pensamiento Computacional (CT):

- Cuestionario 12 (TEST de CT): Se realiza el mismo test de CT una vez ANTES de la experiencia y otra vez DESPUÉS de la experiencia. A rellenas por **niños**.

- Realización del test de Pensamiento Computacional (CT) – Mes después de la experiencia

- Cuestionario 12 (TEST de CT): Se realiza el mismo test de CT aproximadamente UN MES DESPUÉS DE LA EXPERIENCIA para comprobar la durabilidad del aprendizaje. A rellenar por **niños**.

Contraste de hipótesis 2 (experiencia secundaria, evaluación de transiciones):

MUY IMPORTANTE: Para la realización de esta experiencia, los grupos y alumnos deben ser otros, distintos a los que han realizado la experiencia principal (contraste de hipótesis 1).

Hay dos casos posibles, según el centro educativo:

CASO 1: Se realiza la experiencia con dos clases de la misma etapa educativa de la experiencia principal, pero de diferente curso:

Si existe la posibilidad de realizar la experiencia secundaria en otros dos cursos de la misma etapa educativa pero no el mismo curso (por ejemplo, se realiza la experiencia principal en dos clases de 1º de primaria, pero se tiene la posibilidad de interactuar también con dos clases de 2º de primaria), estas dos clases pueden realizar la experiencia secundaria para el contraste de la hipótesis 2, de forma que cada clase sea un grupo.

En esta experiencia, solamente se comprueba si la incorporación de transiciones en un problema mejora la comprensión del mismo por los alumnos, para ello se utilizan dos test de pensamiento computacional, uno con transiciones y otro sin ellas para evaluar, en función de los resultados del test, si esta mejora es significativa.

Pasos a realizar:

- Realización de cuestionarios para el alumno y el profesor-tutor:

- Cuestionario 1: Datos generales. Rellenar por **el alumno o responsable**.

- Realización del test de Pensamiento Computacional (CT):

- Cuestionario 12 (TEST de CT con transiciones): Se realiza el test de CT. A rellenar por **el GRUPO 1**.
- Cuestionario 13 (TEST de CT sin transiciones): Se realiza el test de CT. A rellenar por **el GRUPO 2**.

CASO 2: Se realiza la experiencia con dos clases de la misma etapa educativa y el mismo curso que en la experiencia principal:

En el caso de contar con cuatro clases (aulas) del mismo curso (por ejemplo, se dispone de cuatro clases de 5º curso: dos clases realizarán la experiencia principal detallada anteriormente. Las otras dos clases formarán los dos grupos de control y realizarán la experiencia secundaria.

En esta experiencia, solamente se comprueba si la incorporación de transiciones en un problema mejora la comprensión del mismo por los alumnos, para ello se utilizan dos test de pensamiento computacional, uno con transiciones y otro sin ellas para evaluar, en función de los resultados del test, si esta mejora es significativa. En este caso, además, estos dos grupos serán grupos de control frente a los grupos que realizan la experiencia principal.

Pasos a realizar:

- Realización de cuestionarios para el alumno y el profesor-tutor:

- Cuestionario 1: Datos generales. Rellenar por **el alumno o responsable**.

- Realización del test de Pensamiento Computacional (CT):

Se realiza el test de CT en el mismo tiempo (mismo día o semana) en el que se realiza este mismo test a los alumnos que realizan la experiencia principal (paso B de la experiencia principal):

- Cuestionario 12 (TEST de CT con transiciones): Se realiza el test de CT. A rellenar por **el GRUPO 1**.

- Cuestionario 13 (TEST de CT sin transiciones): Se realiza el test de CT. A rellenar por **el GRUPO 2**.

- Realización del test de Pensamiento Computacional (CT):

Se vuelve a realizar el test de CT en el mismo tiempo (mismo día o semana) en el que se vuelve a realizar este mismo test a los alumnos que realizan la experiencia principal (paso E de la experiencia principal):

- Cuestionario 12 (TEST de CT con transiciones): Se realiza el test de CT. A rellenar por **el GRUPO 1**.

- Cuestionario 13 (TEST de CT sin transiciones): Se realiza el test de CT. A rellenar por **el GRUPO 2**.

Instrucciones e indicaciones para la ejecución del juego

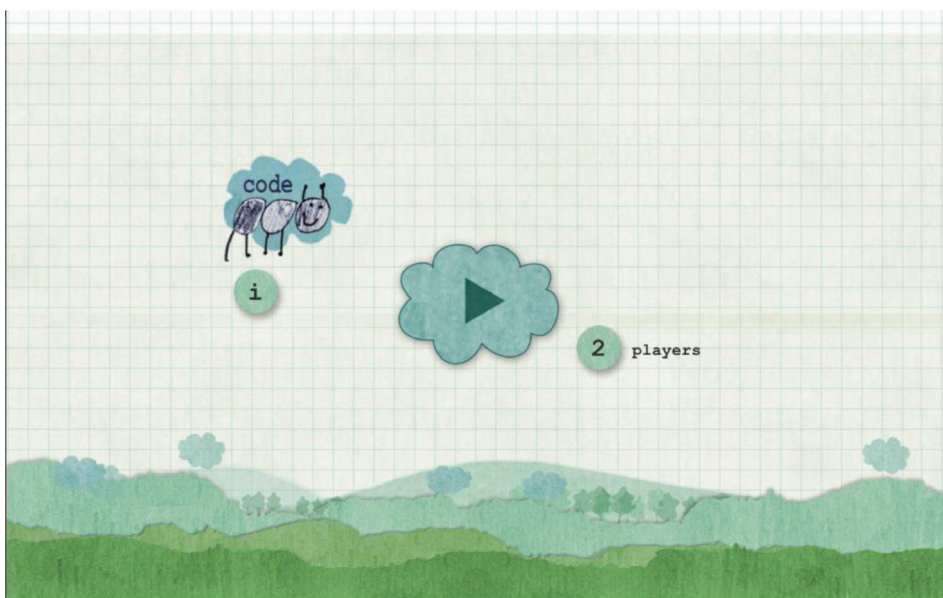
Primeros pasos

Hay dos versiones del juego para el modo local:

- Modo individual o de retos fijos
- Modo colaborativo

Para ejecutar el juego, debe hacerse doble clic en el archivo: "*antCode.exe*" si estamos en versión de PC o sobre la aplicación en caso de tabletas Android.

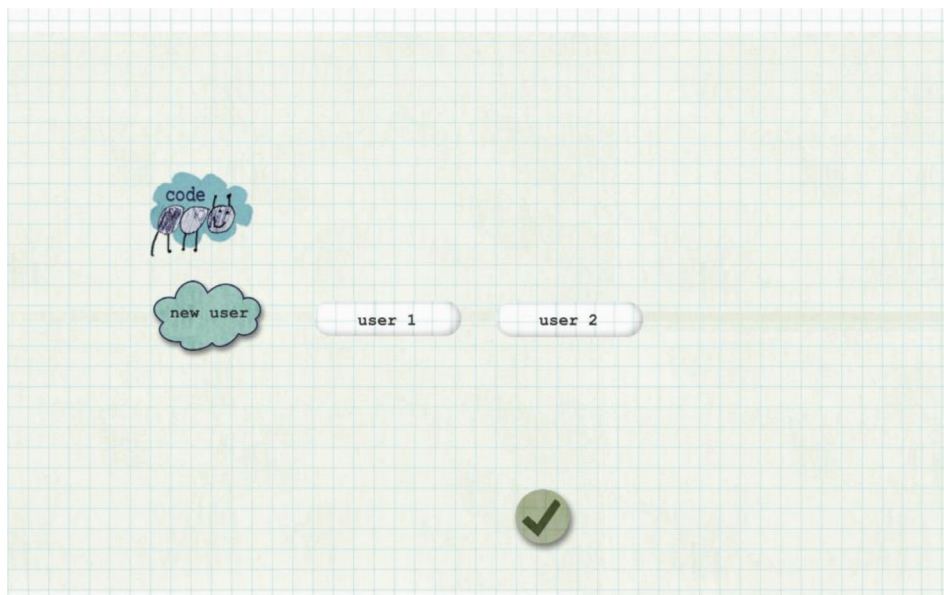
Se podrá cambiar el modo de juego en la pantalla de inicio del juego, en el botón de cambio de número de jugadores:



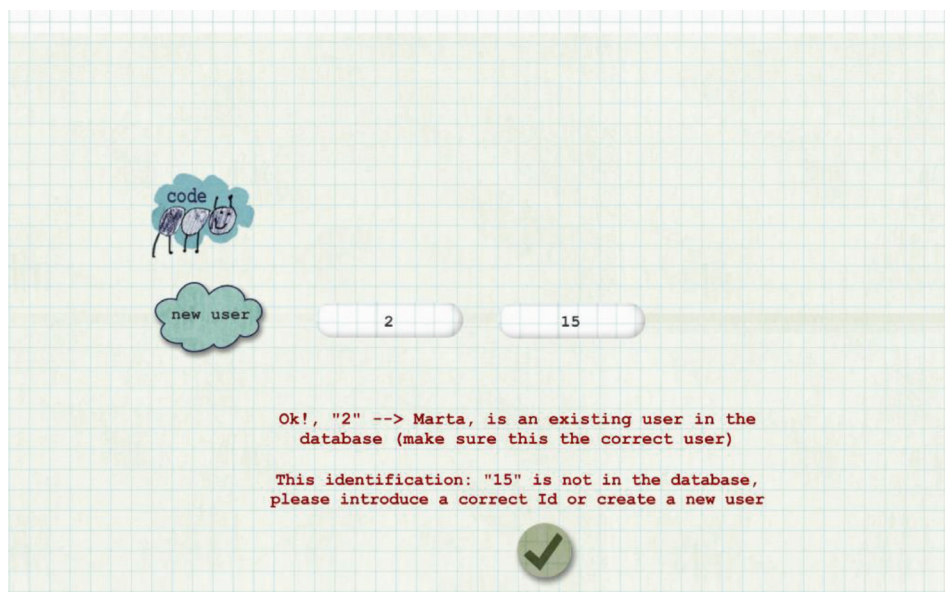
2 jugadores se refiere al modo colaborativo y 1 jugador se refiere al modo individual

Modo colaborativo (2 usuarios):

Cuando ya se haya seleccionado el modo, se pasa a la pantalla de usuarios:



Hay que introducir el identificador de los jugadores en su casilla correspondiente que lo provee de forma automática el juego cuando se registran a los usuarios. **Este punto es crítico** en todo el proceso, dado que, si el identificador del jugador está mal, grabaremos toda la información en otro usuario creado. Hay que asegurarse muy bien que los identificadores incluidos aquí son correctos. Una vez introducidos los identificadores, se hace clic en el botón (o bien se presionará ENTER), y la aplicación nos informará de la existencia de ese jugador (la primera vez esto puede tardar un poco).



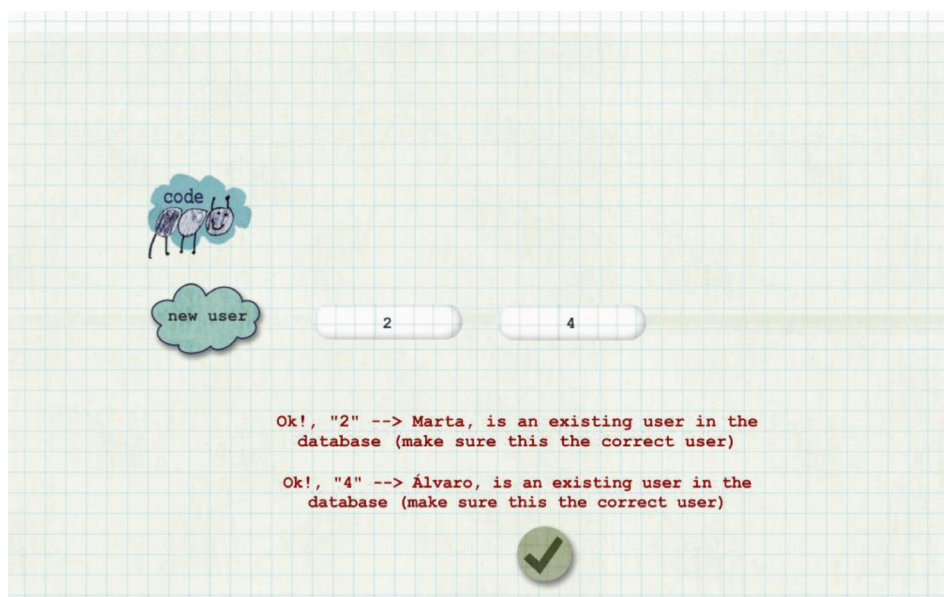
En este ejemplo, hemos probado con los identificadores 2 y 15. El 2 existe y es: Marta, se puede jugar con él porque tengo sus datos (asegurándome de que es mi usuario), pero el usuario 15 no existe, tal vez me he equivocado de identificador o bien tengo que CREAR UN NUEVO USUARIO. La aplicación informará del nombre ficticio del jugador asociado al identificador o bien de que el identificador introducido no se encuentra en la base de datos y, por tanto, no está registrado.

Para crear un nuevo usuario para Álvaro, hago clic en el botón "new user" y se cumplimentará el formulario de nuevo usuario. En la tabla de correspondencia, debe apuntarse el identificador de usuario que se genera, junto al nombre real y al ficticio, puesto que es el identificador a usar en la pantalla de usuarios. Además, se puede anotar también si el niño jugará en modo individual o colaborativo.

Pueden insertarse tantos usuarios como se quiera en una sola sesión. De hecho, se recomienda que se introduzcan todos al principio. Así, quedarán guardados en la base de datos y se podrán introducir los identificadores de usuario en la pantalla de usuarios sin más pasos cuando se quiera que los niños jueguen a la aplicación.

IMPORTANTE: los usuarios solo se inscriben en esta pantalla una vez. No hace falta registrarlos todas las veces. Una vez están en la base de datos, estarán para siempre, se pueden jugar todas las partidas que se quiera en días distintos con ese identificador de usuario.

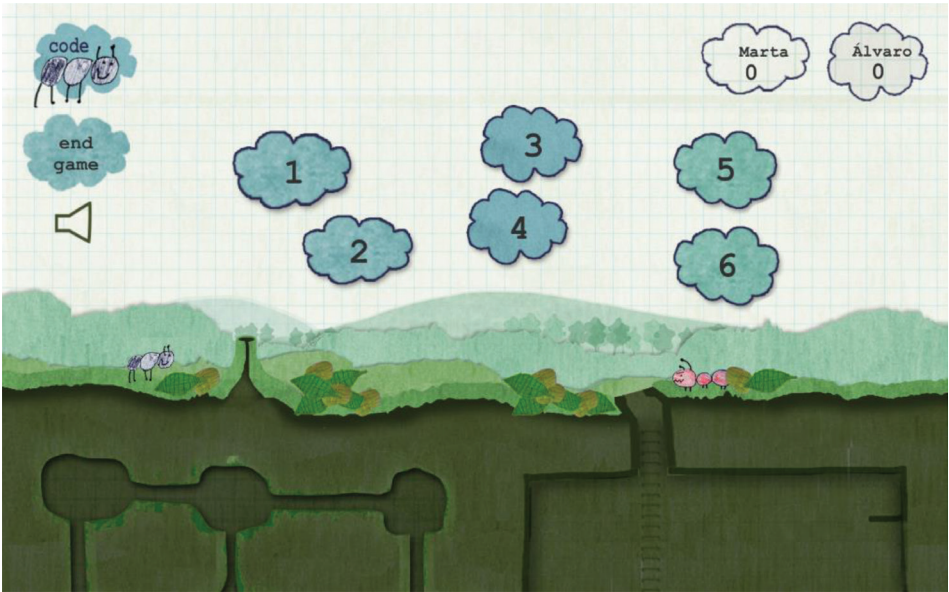
Siguiendo con el ejemplo, finalmente, ya tengo a Marta y a Álvaro en la base de datos, ya puedo jugar las partidas que quiera, en días distintos con ellos:



Ahora puedo pasar a la pantalla de motivación, donde cada jugador indicará como se siente en ese momento. Dado que las palabras están en inglés, se les debe indicar a los niños cuál es su significado:



A continuación, se muestra la pantalla del menú de la aplicación:



En esta pantalla, también puedo quitar el sonido ambiental pulsando sobre el altavoz.

En esta pantalla selecciono el nivel al que quiero jugar. Comenzaré jugando POR ORDEN dos veces a cada nivel, en cada sesión, hasta que se acabe el tiempo. No importa que no se llegue más allá de un determinado nivel. Tal vez en la siguiente sesión de tiempo a más juegos en menos tiempo.

Al hacer clic en cualquiera de las nubes se accede al nivel indicado. Por ejemplo, el nivel 3 con los jugadores Marta y Álvaro.

Al acceder a un nivel por primera vez, deberá realizarse una breve explicación a los niños del objetivo y mecánica del juego. A continuación, se detalla la explicación por cada nivel (las reglas de puntuación se detallan más adelante):

- Nivel 1:

o Objetivo del juego: La hormiga gris debe llevar la hoja hasta el montón de hojas sin pasar por la casilla de la hormiga roja

o Mecánica: el jugador 1 coloca los elementos en el puzle. Cuánto más difícil sea el puzle más puntos obtendrá. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado. Cuánto mejor sea el recorrido, más puntos obtendrá.

- Nivel 2:

o Objetivo del juego: La hormiga gris debe llevar las hojas hasta el montón de hojas sin pasar por la casilla de la hormiga roja

o Mecánica: el jugador 1 coloca los elementos en el puzle. Cuánto más difícil sea el puzle más puntos obtendrá. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado. Cuánto mejor sea el recorrido, más puntos obtendrá.

- Nivel 3:

o Objetivo del juego: La hormiga gris debe llevar las hojas hasta el montón de hojas sin pasar por la casilla de la hormiga roja

o Mecánica: el jugador 1 coloca los elementos en el puzle. Cuánto más difícil sea el puzle más puntos obtendrá. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado, incorporando el número de veces que se repite cada una. Cuánto mejor sea el recorrido, más puntos obtendrá.

- Nivel 4:

- o Objetivo del juego: La hormiga gris debe llevar las hojas hasta el montón de hojas
- o Mecánica: el jugador 1 coloca los elementos en el puzle. Cuánto más difícil sea el puzle más puntos obtendrá. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha" en el orden adecuado, incorporando el número de veces que se repiten cada dos instrucciones agrupadas. Cuánto mejor sea el recorrido, más puntos obtendrá. Este nivel entraña bastante dificultad, ya que las instrucciones agrupadas se repiten las veces indicadas de forma secuencial pero en grupos de dos. Por ejemplo, si un grupo en el que se han incorporado las instrucciones "arriba" y "abajo" y se ha indicado que se repita tres veces, la ejecución será la siguiente: "arriba", "abajo", "arriba", "abajo", "arriba", "abajo". Es posible que este nivel requiera más tiempo de explicación a los alumnos.

- Nivel 5:

- o Objetivo del juego: La hormiga gris debe llegar hasta el montón de hojas
- o Mecánica: el jugador 1 coloca los elementos en el puzle. Cuánto más difícil sea el puzle más puntos obtendrá. El jugador 2 debe conseguir el objetivo del juego colocando las instrucciones "arriba", "abajo", "izquierda", "derecha", incorporando un color que determinará en qué casillas se realizará la acción. Por ejemplo, si se incorpora "amarillo" y "derecha", todas las veces en las que la hormiga esté en una casilla amarilla, avanzará hacia la derecha. Cuánto mejor sea el recorrido, más puntos obtendrá. Este nivel entraña bastante dificultad sobre todo la primera vez por lo que es posible que este nivel requiera más tiempo de explicación a los alumnos.

o Nivel 6:

- o Objetivo del juego: La hormiga gris debe llegar hasta el montón de hojas
- o Mecánica: es la misma que en el nivel 5 pero el laberinto contiene más casillas.

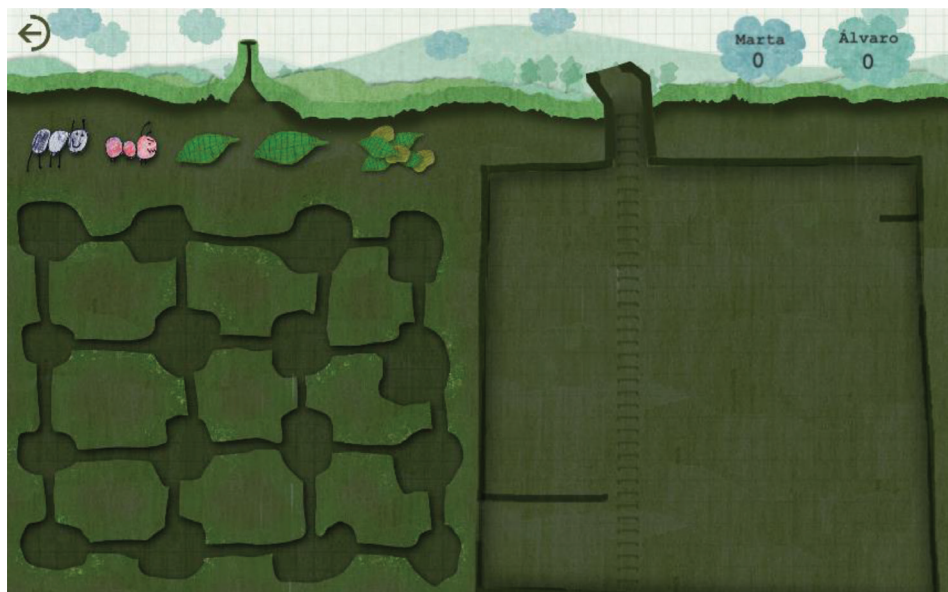
En primer lugar, el jugador con el rol de colocar el puzle (Marta), coloca las piezas donde desee. Cuanto más difícil se lo ponga a su adversario, más puntos conseguirá. Se obtienen más puntos cuando el adversario tenga que dar más pasos para conseguir alcanzar el objetivo. Esto es algo que se debe explicar al alumno la primera vez que se llegue a este escenario colaborativo.

Una vez se coloque el puzle, el otro jugador (Álvaro), tendrá que resolverlo, colocando las piezas como proceda. Habrá que indicar que cuantos menos pasos de la hormiga más puntuación ganará Álvaro:

Reglas de puntuación:

- El jugador 1 recibe:
 - 1 punto por cada paso que da el jugador 2 (si consigue configurar un puzle muy complicado, ganará muchos puntos)
 - 10 puntos si el jugador 2 encuentra a la hormiga roja y pierde
- El jugador 2 recibe:
 - 2 puntos por cada hoja recogida
 - 10 puntos si llega al montón de hojas

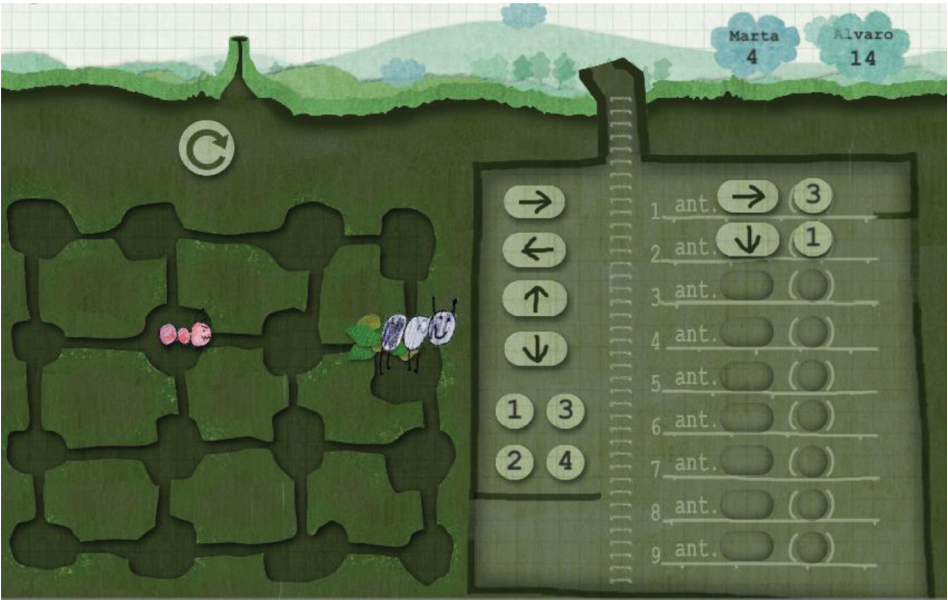
En este caso, juegan Marta y Álvaro. Primero Marta colocará las piezas del puzle y pulsará el botón "tick".



Después, Álvaro resolverá el puzle colocando las instrucciones según el nivel:

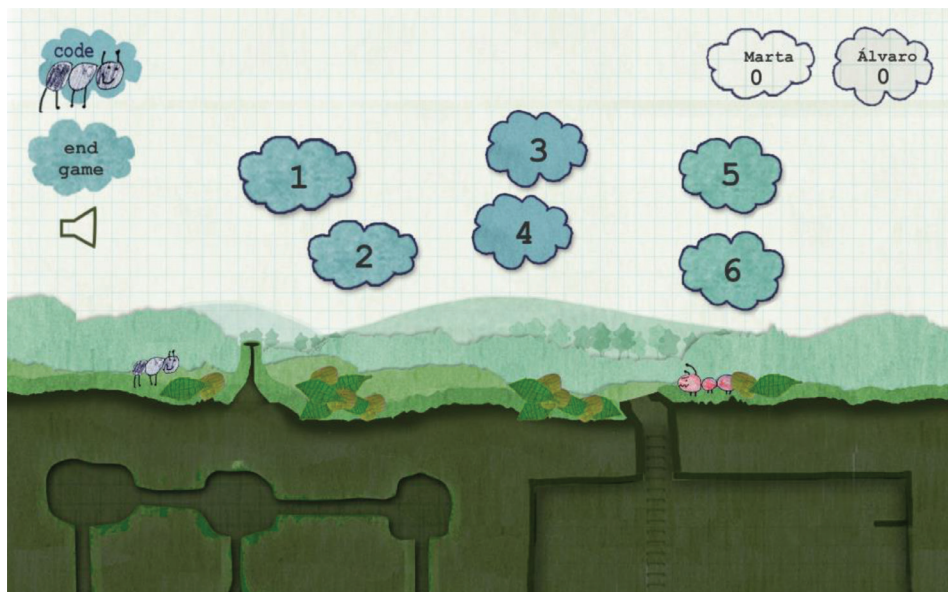


Cuando Álvaro de al botón "Play", se ejecutará su código y la hormiga realizará los pasos propuestos



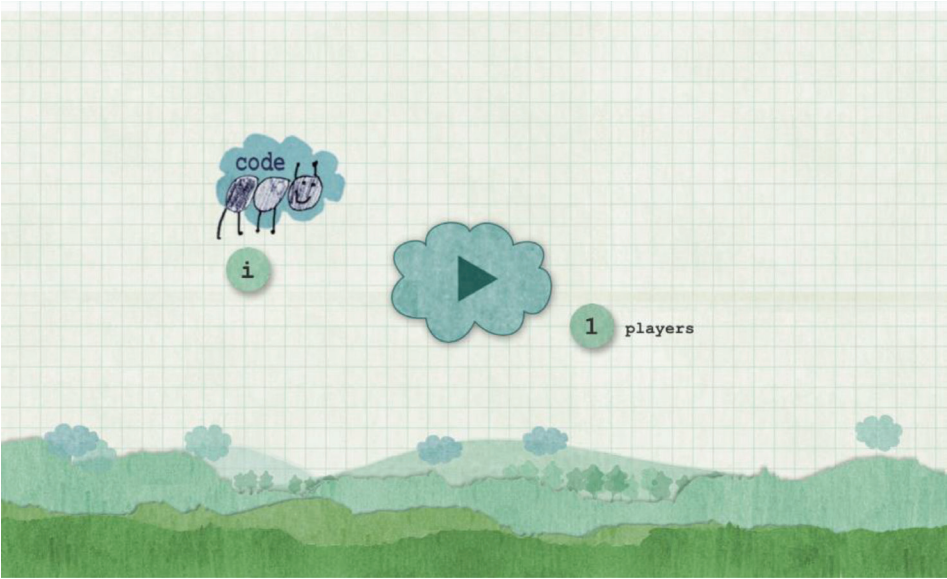
Cuando termine, se da al botón para volver al menú y jugar una nueva sub-partida en el mismo o en otro nivel.

Cuando se cumpla el tiempo por sesión de estos jugadores, ES MUY IMPORTANTE DAR AL BOTÓN "end game" del menú (de esta forma, se guardará la motivación posterior a la prueba y se accederá al menú de usuarios para poder jugar con nuevos usuarios):



Modo individual o de retos fijos (1 usuario):

Se selecciona un jugador en la pantalla inicial, se introduce el código del jugador y se procede exactamente igual, pero la colocación se realiza de forma automática y solo habrá un usuario resolviendo los puzles.



Paso final:

Finalmente, cuando ya se hayan realizado las experiencias de esa sesión, los datos habrán quedado guardados para siempre en la base de datos. Adicionalmente, se crea un archivo de datos con la partida, además de un archivo de datos de bajo nivel muy útil para sacar conclusiones después.

Por cada partida de 2 usuarios (pueden haber jugado a muchos niveles varias veces), se crean dos archivos que contienen el día y la hora de la experiencia en el nombre del archivo. Si ha habido dos partidas con usuarios diferentes, habrá 4 archivos (2 por cada partida con usuarios diferentes).

ES MUY IMPORTANTE, recoger y guardar en un pen-drive, en la nube o en cualquier almacenamiento que se desee los archivos generados, además de enviarlos a maria.zapata@urjc.es y estefania.martin@urjc.es. Preferiblemente, se subirán a la carpeta compartida del proyecto en Google Drive.

Los archivos estarán en la siguiente dirección:

En un dispositivo Android:

Entro en "Mis archivos" y después en "Memoria interna" y sigo esta ruta:
 \Android\data\com.artbabygame.blueAnt\files\

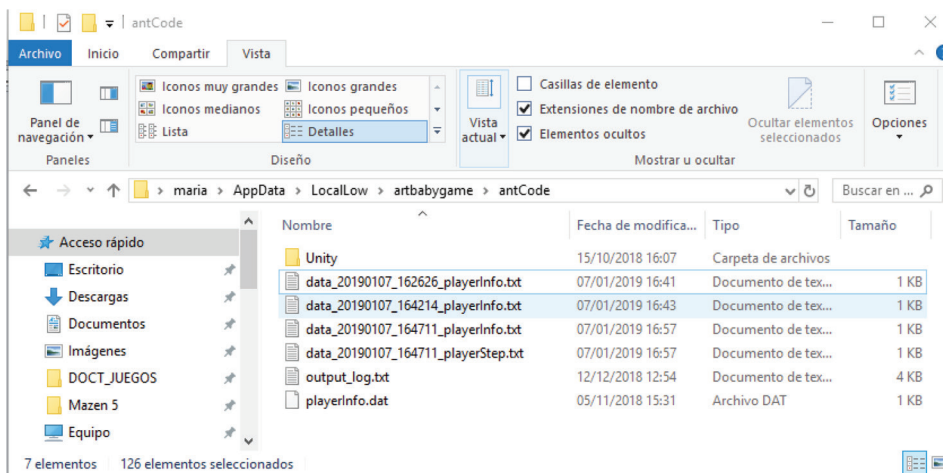
En un PC:

Sigo esta ruta:

C:\Usuarios*nombre de usuario del PC*\AppData\LocalLow\artbabygame\antCode

Ejemplo de ruta: C:\Users\maria\AppData\LocalLow\artbabygame\antCode

NOTA: es posible que algunas de las carpetas de la ruta estén ocultas. Para que se vean es importante tener activado "Elementos ocultos" en la pestaña "Vista" de Windows 10 (para otros sistemas operativos consultar a maria.zapata@urjc.es):



Ejemplo de archivos para una partida con dos usuarios: data_20190107_164711_playerInfo.txt data_20190107_164711_playerStep.txt

RECORDATORIO DE PASOS IMPORTANTES:

- Asegurarse de guardar bien la tabla de correspondencia entre usuarios reales y ficticios (los reales son los nombres reales de los niños y los ficticios son los usuarios del juego) y el identificador de cada usuario (que es el que se mete en la pantalla de usuarios).
- Asegurarse de iniciar una partida con usuarios que existan en la base de datos.
- Al terminar la partida, asegurarse de dar al botón "finalizar partida".
- Asegurarse de guardar en alguna parte los dos archivos que se crean por cada partida entre dos jugadores distintos o entre un solo jugador, y enviarlos a maria.zapata@urjc.es y estefania.martin@urjc.es.

Significado de los archivos

Los datos del archivo: "data_fecha y hora de la prueba_playerInfo.txt", son los mismos que se guardan en la base de datos. El archivo "data_fecha y hora de la prueba_playerStep.txt", ofrece datos extra importantes de bajo nivel para su análisis posterior.

Archivos de tipo: "data_ fecha y hora de la prueba _playerInfo.txt"

A continuación, se describe cada ítem de los archivos de tipo: "data_ *fecha y hora de la prueba* _playerInfo.txt":

Cada archivo que se guarda corresponde a una partida, el nombre del archivo contiene la fecha y hora de creación y puede contener:

- **Cero, uno o varios** usuarios nuevos con sus datos, bajo: @NewUserData
- Datos de una nueva partida bajo: @PartidaData
- Datos de **una o varias** subpartidas dentro de esta misma partida bajo: @SubPartidaData
- Datos finales de **una partida** bajo: @PartidaFinalData

IMPORTANTE: No debe utilizarse el símbolo ";" en el nombre de usuario ni en ningún campo en el que se inserte texto escrito, ya que es el separador de ítems.

Archivos de tipo: "data_ fecha y hora de la prueba _playerStep.txt"

Id;IdPartida;IdSubpartida;Timestamp;IdUsuario;Nivel;Accion;Param1;Param2

- Id: auto-numérico.
- IdPartida: Id de la partida a la que pertenecen estos datos.
- IdSubpartida: Id de la subpartida a la que pertenecen estos datos.
- Timestamp: marca de tiempo estilo UNIX que indica los milisegundos transcurridos desde el inicio de la subpartida en milisegundos.
- IdUsuario: Usuario que realiza la acción.
- Nivel: nivel jugado.
- Accion: identificador de la acción realizada por el usuario. Define todas las acciones posibles.
- Param1: indica el identificador de casilla, único para cada nivel.
- Param2: indica el objeto

Índice de cuestionarios

Los cuestionarios se aportan en documentos aparte. A continuación, se muestra el índice de los mismos:

CUESTIONARIOS PARA EL ALUMNO - EXPERIENCIA:

- Cuestionario 1: Datos generales. Rellenar por **el alumno o responsable*** ANTES de la experiencia.
- Cuestionario 2: Datos motivacionales previos. Rellenar por el **alumno o responsable*** ANTES de la experiencia.

- Cuestionario 3: Datos motivacionales posteriores. Rellenar por el **alumno o responsable*** DESPUÉS de la experiencia.
- Cuestionario 4: Datos objetivos sobre el alumno. Rellenar por el **responsable*** ANTES o DESPUÉS.
- Cuestionario 5: Datos objetivos sobre el centro escolar. Rellenar por el **responsable*** ANTES o DESPUÉS.

CUESTIONARIOS PARA EL PROFESORADO - EXPERIENCIA:

- Cuestionario 6: Rellenar por el **profesor-tutor** de los alumnos ANTES de la experiencia.
- Cuestionario 7: Rellenar por el **profesor-tutor** de los alumnos DESPUÉS de la experiencia.

TEST DE CT

- Cuestionario 12 (TEST de CT): Se realiza el mismo test de CT una vez ANTES de la experiencia y otra vez DESPUÉS de la experiencia. A rellenar por **niños**.
- Cuestionario 13 (TEST de CT)**: Se realiza el mismo test de CT una vez ANTES de la experiencia y otra vez DESPUÉS de la experiencia. A rellenar por **niños**.

*Con responsable nos referimos al responsable encargado de llevar a cabo la experiencia con los alumnos.

**Sólo para la experiencia de contraste de la hipótesis 2.

B.2. Base de datos

Cada archivo que se guarda corresponde a una partida, el nombre del archivo es la fecha y hora de creación y puede contener:

- **Cero, uno o varios** usuarios nuevos con sus datos para la tabla USUARIO (@NewUserData → fila nueva de la tabla USUARIO)
- Datos de una nueva partida, datos para la tabla PARTIDA (sólo **una partida por archivo**) (@PartidaData → fila nueva de la tabla PARTIDA). En este caso, algunos campos estarán vacíos, pero se crea por seguridad, para guardar los datos más importantes de la partida.
- Datos de **una o varias** subpartidas dentro de esta misma partida, datos para la tabla ESTADISTICA (@SubPartidaData → fila nueva de la tabla ESTADISTICA)
- Datos finales de **una partida**, son datos para la tabla PARTIDA (@Partida-FinalData → del mismo registro creado en @PartidaData, pero con datos actualizados). Sólo se guardan si se pulsa el botón "*Finalizar partida*" para volver al menú de login e iniciar una nueva partida con nuevos usuarios. Los datos actualizados que se incluyen son: *Puntuacion1*, *Puntuacion2*, *AnimoPos1*, *AnimoPos2*, *Ganadas1* y *Ganadas2*. Si existe esta información en el archivo, se utilizarán estos datos para la tabla. Si no existe, se utilizarán los de @PartidaData.
- Datos a bajo nivel (sólo en los LOGS): tabla ACCIONES (@NewLog). Se guarda cada acción del jugador a bajo nivel

@NewUserData

Id;Nombre;Centro;Clase;Numero;FechaAlta;EdadMeses;Genero;NecEsp;UsoProg;UsoDis p;UsoPlat;Dispositivo;Exp;Otros

@PartidaData

Id;Dispositivo;FechaHora;IdUsuario2;IdUsuario2;Puntuacion1;Puntuacion2;AnimoPre1;AnimoPre2;AnimoPos1;AnimoPos2;GanadorTotal

@SubPartidaData

Id;IdPartida;Nivel;TiempoInicio;Tiempo1;Tiempo2;PasosProp;PasosDados;PasosOptimo; VectorPosiciones;VectorProp;VectorEjecutado;VectorOptimo;-Ganador

@PartidaFinalData

Id;Dispositivo;FechaHora;IdUsuario2;IdUsuario2;Puntuacion1;Puntuacion2;AnimoPre1;AnimoPre2;AnimoPos1;AnimoPos2;GanadorTotal

@NewLog

Id;IdPartida;IdSubpartida;Timestamp;IdUsuario;Nivel;Accion;Param1;Param2

Debajo de cada epígrafe (por ejemplo, debajo de @NewUserData), se guardan los datos del nuevo registro de la tabla con los atributos necesarios separados por punto y coma: ";". Por ello, no debe utilizarse este símbolo en el nombre de usuario ni en ningún campo en el que se inserte texto escrito

A continuación, se describen las tablas de la base de datos:

USUARIO

Atributo	Id
Tipo	Int KEY Identificador de usuario ÚNICO que se genera automáticamente Es un identificador
Atributo	Publicacion
Tipo	string Descripción de la publicación. Por ejemplo: local android, local pc masiva android, masiva iOs, etc.
Atributo	Nombre
Tipo	string Es el apodo que, por proteccion de datos no corresponde con el nombre real del niño.
Atributo	Exp
Tipo	int Identificador de la experiencia
Atributo	FechaAlta
Tipo	date Fecha en la que el usuario se registra por primera vez

Atributo	Centro
Tipo	string Identificador del centro (colegio)
Atributo	ClaseNumero
Tipo	int Identificador de la clase (numero de aula)
Atributo	ClaseLetra
Tipo	string Identificador de la clase (letra de aula)
Atributo	EdadMeses
Tipo	int Edad del usuario en meses de vida
Atributo	Género
Tipo	int (0 / 1) 0: masculino, 1:femenino
Atributo	NecEsp
Tipo	int (0 / 1) 0: no, 1:si. Indica si el niño tiene necesidades especiales
Atributo	UsoProg
Tipo	int (0 / 1) 0: no, 1:si. Indica si el niño ha utilizado previamente aplicaciones como Scratch Jr, que enseñan a programar
Atributo	UsoDisp
Tipo	int (0 / 1) 0: no, 1:si. Indica si el niño está habituado a utilizar dispositivos tipo tablet o móviles
Atributo	UsoPlat
Tipo	int (0 / 1) 0: no, 1:si. Indica si el niño está habituado a utilizar plataformas tipo youtube

Atributo	Otros
Tipo	string
	Otros datos significativos que se quieran introducir

PARTIDA

Atributo	Id
Tipo	Int KEY
	Es el identificador de partida que se genera automáticamente
Atributo	Dispositivo
Tipo	string
	Describe el tipo de dispositivo en el que se realiza la experiencia y modelo concreto
Atributo	FechaHora
Tipo	date
	Fecha y hora e la que se inicia la partida
Atributo	IdUsuario1
Tipo	int
	FOREIGN KEY Identificador del usuario 1. Corresponde con el atributo Id de la tabla USUARIO del usuario 1. Un jugador puede jugar varias partidas
Atributo	IdUsuario2
Tipo	int
	FOREIGN KEY Identificador del usuario 2. Corresponde con el atributo Id de la tabla USUARIO del usuario 2. Un jugador puede jugar varias partidas
Atributo	Puntuacion1
Tipo	int
	Es la puntuación total de todas las subpartidas de esta partida, de usuario 1, según las reglas de puntuación del juego

Atributo	Puntuacion2
Tipo	int Es la puntuación total de todas las subpartidas de esta partida, de usuario 2, según las reglas de puntuación del juego
Atributo	AnimoPre1
Tipo	int [0,3] Indica cómo se siente el usuario 1 antes de la experiencia en una escala de 0 a 3 (0 no opina). 1:bad, 2:so-so, 3:good
Atributo	AnimoPre2
Tipo	int [0,3] Indica cómo se siente el usuario 2 antes de la experiencia en una escala de 0 a 3 (0 no opina) 1:bad, 2:so-so, 3:good
Atributo	AnimoPos1
Tipo	int [0,3] Indica cómo se siente el usuario 1 después de la experiencia en una escala de 0 a 4 (0 no opina) 1:bad, 2:so-so, 3:good
Atributo	AnimoPos2
Tipo	int [0,3] Indica cómo se siente el usuario 2 después de la experiencia en una escala de 0 a 4 (0 no opina) 1:bad, 2:so-so, 3:good
Atributo	Ganadas1
Tipo	int Número de partidas ganadas por el jugador 1
Atributo	Ganadas2
Tipo	int Número de partidas ganadas por el jugador 2

ESTADÍSTICA

Atributo	Id
Tipo	int
	KEY
	Identificador de subpartida. Una partida tiene varias subpartidas
Atributo	IdPartida
Tipo	int
	FOREIGN KEY
	Identificador de la partida, apunta al atributo Id de la tabla PARTIDA. Una partida tiene varias subpartidas
Atributo	Nivel
Tipo	int
	Nivel de la subpartida (van del 1 al 6, pero en el futuro puede haber más)
Atributo	TiempoInicio
Tipo	date
	Fecha y hora de inicio de la subpartida
Atributo	Tiempo1
Tipo	int
	Tiempo en el que el primer jugador (el que propone el puzle) tarda en colocar las piezas
Atributo	Tiempo2
Tipo	int
	Tiempo en el que el segundo jugador (el que soluciona el puzle) tarda en pensar la solución (no incluye el tiempo de ejecución)
Atributo	PasosProp
Tipo	int
	Numero de pasos que el jugador 2 propone para solucionar el puzle

Atributo	PasosDados
Tipo	int Numero de pasos que se ejecutan realmente en el juego según la solución propuesta por jugador 2
Atributo	PasosOptimo
Tipo	int Numero de pasos que supondrían la solución óptima al problema (no lo calcula la app, se calculará en postproceso, "NA": default)
Atributo	VectorPosiciones
Tipo	string Vector que indica la configuración del puzle (dónde ha colocado las piezas el jugador 1) Es un vector tan largo como el número de casillas del puzle. Cada posición del vector corresponde con una casilla ordenadas por filas. El número que se encuentra en la posición que indica la casilla, indica los siguiente: 0:nada, 1:inicio, 3:Hoja, 2: Hormiga roja, 4:Fin, 5:hoja2
Atributo	VectorProp
Tipo	string Vector que indica la propuesta del jugador 2. Se indica el código propuesto según el siguiente esquema general (varía en cada nivel): [(dirección, número de repeticiones)(dirección, número de repeticiones)]
Atributo	VectorEjecutado
Tipo	string Vector que indica la ejecución de la propuesta del jugador 2 El vector indica las casillas por las que va pasando la hormiga comenzando por la casilla de inicio y sigue según vaya moviéndose. Por ejemplo un vector podría ser: [0,3,4,7,8]. Si se trata de un puzle matriz de 3×3, el recorrido sería el siguiente: [(0,0)(1,0)(1,1)(2,1)(2,2)]

Atributo	VectorOptimo
Tipo	string Vector que indica un recorrido óptimo de los posibles Se calculará en postproceso
Atributo	Ganador
Tipo	int [0,2] Ganador de esta subpartida → 1: ha ganado el jugador 1; 2: ha ganado el jugador 2; 0: default

NEWLOG

Id
IdPartida
IdSubpartida
Timestamp
IdUsuario
Nivel
Accion
Param1
Param2



Tesis doctoral

María Zapata Cáceres

Tesis doctoral