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 trasmitir 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].

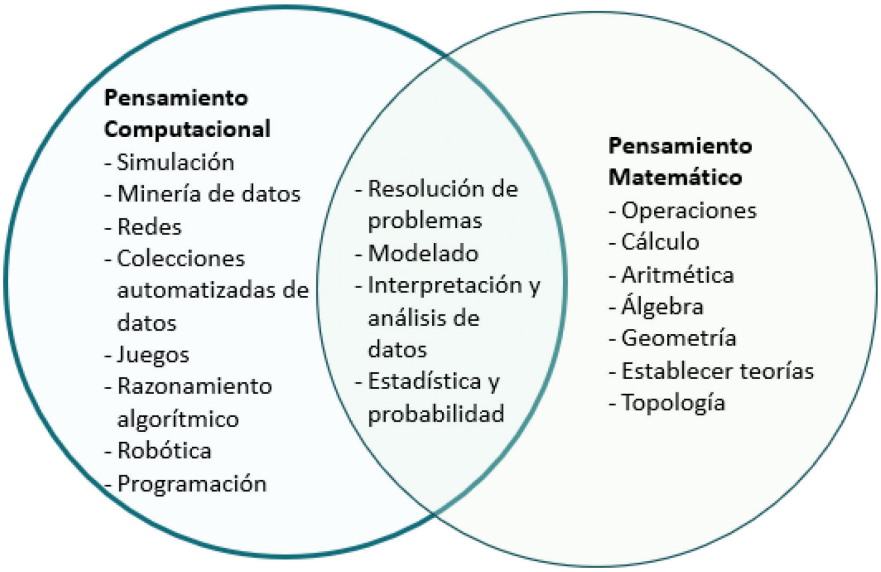


Figura 6. Relación dual entre el Pensamiento Computacional y el Pensamiento de Diseño según Kelly et al. [36].

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 propondrán 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; procesado 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 educacional 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 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, cinestesia 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.

Tabla 1. Resumen de las principales descomposiciones del PC y propuesta.

Wing	National Research Council	Barr y Stephenson	ISTE y CSTA	Grover y Pea	Bers el al.	Atmatzidou *et al.*	Anderson	Shute el al.	TESIS
2006 / 2008	2010	2011	2011	2013	2014	2016	2016	2017	**2021**
Abstracción	Abstracción	Abstracción	Abstracción	Abstracción / patrones	Abstracción	Abstracción	Abstracción	Abstracción	**Abstracción**
							Reconocimiento de patrones		
				Simbolismo / represent.					
	Gestión de datos	Organización / análisis de datos	Organización / análisis de datos	Procesado de la información					**Análisis de datos**
Descomp. de problemas			Descomp. estructurada de problemas / algoritmia	Descomp. estructurada		Descomp.	Descomp. de problemas	Descomp.	**Descomposición**
Reform. de problemas		Reformulación de problemas	Reformulación de problemas						
		Generalización	Generalización		Generalización	Generalización		Generalización	**Generalización**

Wing	National Research Council	Barr y Stephenson	ISTE y CSTA	Grover y Pea	Bers el al.	Atmatzidou *et al.*	Anderson	Shute el 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 platea 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 (CODDII) 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 necesa-

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 desarrolla 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 el 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 complemente 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 puzle, 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 puzles, 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.

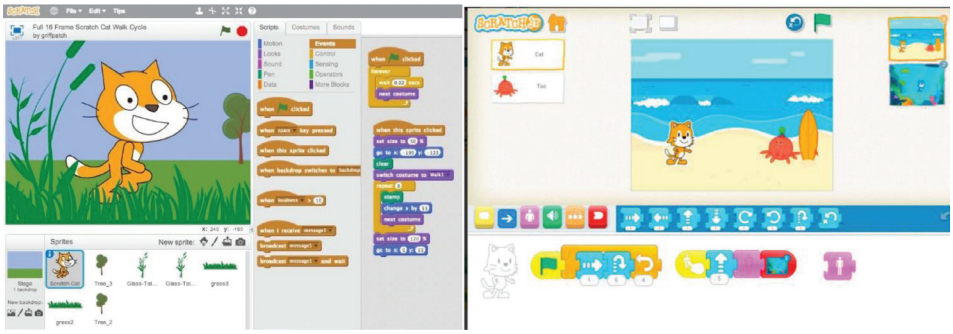
Los entornos de software educativo basados en bloques visuales de programación son muy populares actualmente para el desarrollo del PC en etapas tempranas (véase la Tabla 5). Los estudiantes emplean tablets u ordenadores y utilizan elementos visuales (bloques) que se arrastran y sueltan para construir 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 puzles 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 Techology 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].

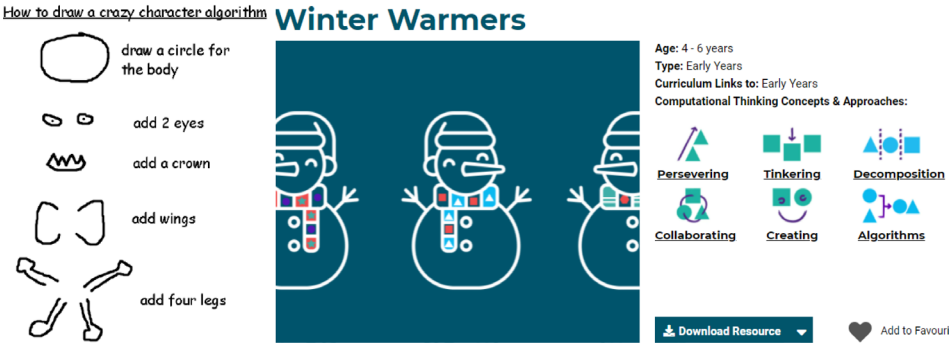
Otra posibilidad es la enseñanza de la programación a etapas tempranas a través de cuentos, como Hello Ruby, en el que cada lenguaje de programación está representado por un personaje. Algunos investigadores proponen crear comics [130] para aprender programación ya que consideran que el lenguaje visual 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 relativas 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 varios 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.				
	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.		

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.					