CAPÍTULO 2. ESTADO DEL ARTE

A lo largo de este capítulo se abordan los conceptos fundamentales y los trabajos del estado del arte relacionados con la temática del presente trabajo de tesis doctoral. De esta forma, se exponen los conceptos clave abordados con el objetivo de sentar los cimientos de la compresión de los capítulos posteriores en el ámbito de la gestión de identidades y del análisis de comportamiento. También, se analizan los trabajos y la literatura más relevantes en dichos ámbitos con el objetivo de situar el presente trabajo en un contexto y por consiguiente, poder establecer las carencias actualmente existentes y así dar sentido al trabajo de investigación aquí realizado.

En primer lugar, se aborda la gestión de identidades. Para ello, se analiza la evolución de la gestión de identidades, desde los primeros sistemas desarrollados, pasando por los sistemas centralizados, hasta llegar a los sistemas federados. Además, se profundiza en los estándares de gestión de identidades federados más populares hoy en día, OpenID [19], OAuth [20] y *OpenId Connect* (OIDC) [21]. Posteriormente, se exponen los principales trabajos que analizan las amenazas de estos estándares y las soluciones propuestas. En segundo lugar, se analiza la rama del aprendizaje máquina denominada análisis de comportamiento. En esta sección se explican las bases del análisis de comportamiento, los dominios de aplicación, así como los trabajos más relevantes y relacionados con el presente trabajo de investigación. Por último se exponen las limitaciones encontradas en la literatura.

2.1. Gestión de identidades y accesos

La gestión de identidades y accesos es la rama de las ciencias de computación que se encarga de gestionar el ciclo de vida de una identidad en un sistema de información, desde que se registra en dicho sistema, durante la interacción con el mismo y hasta que finalmente es eliminada [22]. Está compuesto principalmente por dos ramas fundamentales: el control de accesos y la gestión de identidades. En las siguientes secciones se profundiza en dichas ramas.

2.1.1. Contexto y conceptos básicos

Toda persona posee una identidad física que utiliza para identificarse frente a los controles de accesos existentes en el mundo físico como, por ejemplo, una aduana en un aeropuerto, o para autorizar una transacción económica en una entidad bancaria, entre otros. Esta identidad física está compuesta por el conjunto de características que identifican a una persona concreta como,

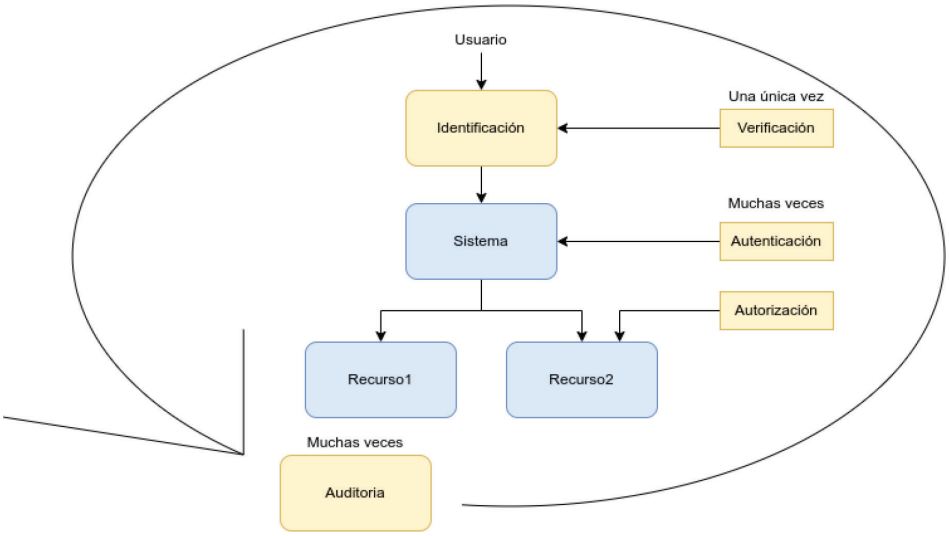
por ejemplo, el nombre y apellidos (características legales), la huella dactilar (características físicas), u otros atributos menos identificativos como el documento nacional de identidad o el conjunto de propiedades a su nombre, etc. En el mundo tecnológico y de los sistemas de información estas identidades físicas son reemplazadas por identidades digitales. Del mismo modo, las identidades digitales se pueden definir como el conjunto de características, preferencias y reputación que identifican a una entidad o usuario concreto dentro de un sistema de información [23]. El concepto de entidad en el ámbito digital no solo hace referencia a los usuarios, sino que también incluye a toda entidad que puede interactuar con dicho sistema como, por ejemplo, empresas, dispositivos o agentes software (servicios web, clientes web, etc), entre otros. De este modo, una entidad física puede corresponderse con una o muchas identidades digitales, pero una identidad digital solo puede corresponderse con una o ninguna identidad física (p. ej. un agente software no se corresponde con ninguna identidad física).

Los sistemas de información alojan un conjunto de activos con los cuales las entidades interactúan. Estos activos son denominados recursos, y al igual que en el mundo físico, han de estar protegidos con el objetivo de garantizar que no se realicen acciones no autorizadas sobre ellos, como accesos indebidos o modificaciones que puedan corromperlos. Estos sistemas que alojan los recursos o que proveen a una entidad o usuario de algún tipo de servicio, son comúnmente denominados, por los estándares de gestión de identidades, como proveedores de servicio (en inglés, *Service Provider* [SP]). Además, hay que hacer mención al agente encargado de gestionar el ciclo de vida de las identidades propiamente dichos, el proveedor de identidades (en inglés, *Identity Provider* [IdP]).

Las identidades digitales son la base de cualquier sistema de información en el que interactúan múltiples entidades o usuarios, pues permiten que se cumplan los tres principios básicos de la seguridad sobre los recursos alojados. Estos principios son: la confidencialidad, la integridad y la disponibilidad [24]. La confidencialidad garantiza que los datos alojados en el sistema de información solo pueden ser accedidos por aquellas entidades autorizadas a ello. La integridad asegura que los recursos no han sido manipulados o corrompidos por un tercero. La disponibilidad hace referencia a que los recursos se encuentren accesibles cuando son requeridos por una entidad. Además, gracias a las identidades digitales también se puede hacer cumplir un cuarto principio de la seguridad, el del no repudio, el cual garantiza que una entidad concreta no puede negar haber realizado una acción concreta en caso de que verdaderamente se haya llevado a cabo.

El éxito de garantizar que los principios básicos de seguridad se cumplan recae sobre los procesos de IAAA. Estos procesos, analizados a continuación, se pueden ver esquematizados en la Figura 2.1.

Figura 2.1. Esquema de los procesos de IAAA



La identificación es el proceso mediante el cual se establece un vinculo de relación entre una entidad y su identidad, dentro del sistema de información, en base a los atributos proporcionas por la misma al sistema. Esta muy relacionado con el proceso de verificación, el cual se encarga de comprobar que una entidad previamente identificada en el sistema se corresponde con otra identidad, por ejemplo, una identidad física. Supóngase que un usuario se da de alta en una aplicación de la agencia tributaria para poder realizar unas gestiones particulares. En este momento, el usuario al darse de alta se ha identificado en el sistema. Sin embargo, es posible que, para realizar ciertas gestiones restringidas, la agencia tributaria deba corroborar que ese usuario es quien dice ser y, por consiguiente, ha de verificar que la identidad digital con la que se ha dado de alta se corresponde con una identidad física que posee los privilegios para poder realizar la gestión restringida. De este modo, puede solicitar al usuario que se presente en las instalaciones físicas de hacienda y presente su DNI, quedando así verificada su identidad digital de manera segura.

La autenticación es el proceso por el cual un sistema de información es capaz de verificar, en un momento determinado, que una entidad es quien dice ser. Los mecanismos de autenticación se basan en proponer distintos retos que solo una entidad o usuario debería ser capaz de responder. Estos retos se pueden estructurar en tres categorías principales de acorde a su naturaleza [23]: algo que se sabe, algo que se tiene, o algo que eres. Para algo que se sabe normalmente se suelen utilizar contraseñas o un código personal (PIN), para algo que se tiene se suelen utilizar tarjetas inteligentes, *tokens* o un teléfono inteligente y

para algo que eres se suelen utilizar rasgos biométricos como la huella dactilar. Estos retos se pueden combinar entre sí, aumentando así la seguridad del sistema que lo incorpora. A este proceso se le denomina Autenticación de Múltiples Factores (AMF) [25]. Cabe destacar que estos retos se suelen evaluar en un único instante, al comienzo de la interacción entre el usuario y el sistema. Con el objetivo de mejorar la seguridad, estos retos se pueden extender a lo largo del tiempo, es decir, a lo largo de toda una sesión de interacciones, pudiendo así verificar la identidad de una entidad tantas veces como sea necesario mientras una sesión esté activa. El proceso de extender la autenticación a lo largo de toda una sesión se denomina autenticación continua [26].

La autorización consiste en permitir o denegar el acceso a un recurso mediante la comprobación de los privilegios y permisos que posee la entidad interesada en un sistema de información. La identificación ocurre una única vez en un sistema, al comienzo de las interacciones, sin embargo, tanto la autenticación como la autorización son procesos que se van a repetir múltiples veces a lo largo del ciclo de vida de una identidad digital.

La auditoría es el proceso que se encarga de registrar todas las interacciones de las entidades con un sistema de información. Gracias a este proceso, todas las interacciones realizadas por las diversas entidades, quedan almacenadas y por lo tanto se pueden utilizar para analizar y verificar si los procesos de identificación, autenticación y autorización, anteriormente descritos, se han realizado de forma satisfactoria.

Finalmente, cabe destacar que, todo sistema de gestión de identidades y accesos ha de garantizar en la medida de lo posible, además de los tres principios básicos de seguridad, los siguientes requisitos de privacidad: revocación, anonimato, libre elección, verificación, antifraude y mínima información. La revocación hace referencia a que un usuario ha de poder solicitar que el sistema deje de almacenar la información asociada a su identidad digital. El anonimato hace referencia a que las identidades, tanto físicas como digitales, no han de poderse asociar con otras identidades para obtener información adicional. La libre elección hace referencia a que cada entidad ha de poder elegir entre múltiples proveedores de identidades. La verificación hace referencia a que un usuario ha de poder contrastar y consultar la información que un proveedor de identidades posee sobre su identidad. El antifraude se refiere a la imposibilidad de realizar determinadas acciones por un agente externo que ha suplantado una identidad del sistema. La mínima información hace referencia a que el sistema no debe almacenar más información sobre la entidad que la estrictamente necesaria.

La Tabla 2.1 recopila, a modo resumen, los conceptos fundamentales abordados en esta sección con el objetivo de mejorar la comprensión global de este documento.

2.1.2. Control de accesos

El control de accesos se encarga de garantizar o restringir el acceso de las entidades y usuarios (bajo una identidad digital) a un servicio o recurso del sistema. Esto garantiza que solo los usuarios legítimos puedan acceder a los recursos o servicios determinados para ello bajo unas condiciones seguras y predefinidas, negándole dicho acceso a los usuarios no autorizados.

Tabla 2.1. Resumen de los conceptos básicos relacionados con la gestión de identidades y accesos

Concepto	Definición
Entidad o usuario	Ente que interactúa con un sistema (usuario, empresa, agente software, etc)
Recurso	Activo de un sistema de información
Identidad digital	Conjunto de atributos, preferencias y reputación que identifican a una entidad o usuario concreto dentro de un sistema de información
Service Provider (SP)	Proveedor de servicio, recursos o aplicación.
Identity Provider (IdP)	Proveedor de identidades. Encargado de gestionar el ciclo de vida de una identidad digital.
Identificación	Proceso mediante el cual se corrobora la identidad digital de una entidad de forma exclusiva en un sistema de información con el objetivo de poder validarla en las siguientes interacciones.
Autenticación	Proceso de validación una identidad digital, garantizando así que una entidad es quien dice ser en un momento determinado.
Autenticación continua	Extensión a lo largo de toda una sesión del proceso de autenticación.
Autorización	Proceso de restringir el acceso de una entidad o usuario a un recurso.
Auditoría	Proceso de registro de todas las interacciones de las entidades o usuarios con un sistema de información.

Figura 2.2. Almacenamiento en directorios. Modelo HRU

	Archivo 1	Archivo 2	Ejecutable 1	....	Objeto M
Usuario 1	Acción1...AcciónK	Acción1...AcciónK	Acción1...AcciónK		
Usuario 2	Acción1...AcciónK	Acción1...AcciónK	Acción1...AcciónK		
....					
Usuario N					

Los primeros sistemas informáticos estaban pensados para ser utilizados por un único operador o usuario. Este único usuario tenía plena disponibilidad sobre los recursos del sistema. Sin embargo, debido a las necesidades intrínsecas de estos sistemas, evolucionaron hasta convertirse en sistemas multiusuario.

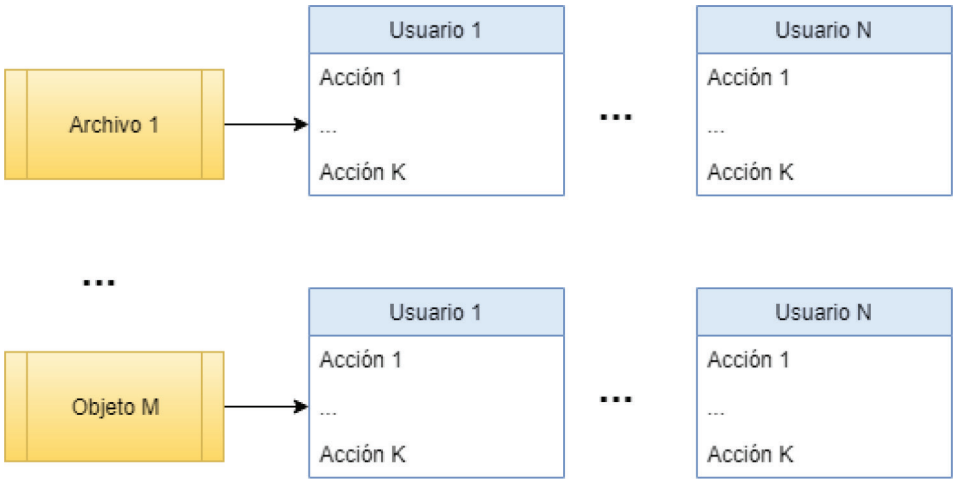
En los sistemas multiusuario, los usuarios compiten por los recursos disponibles tanto en tiempo como en espacio. Debido a esta competencia, surgió la necesidad de elaborar sistemas de control de accesos que restringiesen el acceso de los diferentes usuarios a los recursos, servicios o aplicaciones alojados en el sistema. De esta forma, surgieron las políticas de acceso, las cuales recopilan un conjunto de reglas que determina los privilegios de las entidades y usuarios sobre los recursos del sistema. Por tanto, el control de accesos está muy relacionado con el concepto de autorización.

Existen dos formas muy extendidas, entre otras, de almacenar estas políticas de accesos [27]: directorios y las listas de control de accesos (en inglés, *Access Control Lists* [ACL]). El modelo de directorios, también conocido como modelo Harrison, Ruzzo y Ullman (HRU), viene definido como una matriz en la que cada fila se corresponde un un usuario del sistema y cada columna se corresponde con un recurso del sistema (ver Figura 2.2). Como se puede observar, cada celda posee los privilegios de lectura, escritura, ejecución y propiedad de un usuario para un recurso concreto. Por otro lado, las listas de control de accesos definen un conjunto de listas en el que cada elemento representa un usuario o un recurso (ver Figura 2.3). De este modo, las conexiones entre dos elementos denotan que el primer elemento posee los privilegios indicados sobre el segundo elemento.

Sea cual sea la forma de almacenar las políticas de acceso, existen multitud de formas de gestionar los accesos. Estos mecanismos de gestión son denominados modelos de control de acceso. Hoy en día, existen cuatro modelos fundamentales de control de accesos:

1. Modelos de control de accesos obligatorio (en inglés, *Mandatory Access Control* [MAC]): el administrador del sistema es el encargado de definir

Figura 2.3. Almacenamiento en listas de control de accesos



los privilegios y permisos de los recursos y de los propios usuarios con el objetivo de determinar si un acceso es legítimo. Un claro ejemplo de este sistema de control de accesos es la clasificación de archivos de la Agencia de Seguridad Nacional de Estados Unidos. En este modelo, cada recurso del sistema se categoriza de acorde a su sensibilidad (p. ej. público, privado, confidencial, secreto, etc) y a su categoría (p. ej. departamento, proyecto, rango de mando, etc). De esta forma, solo los usuarios que posean el nivel requerido de sensibilidad y categoría pueden acceder a los recursos de los dichos niveles. Este tipo de sistemas son centralizados, pues es un administrador el que define las políticas de acceso del sistema globalmente.

2. Modelos de control de accesos discrecionales (en inglés, *Discretionary Access Control* [DAC]): definido por *Trusted Computer System Evaluation Criteria*. En este tipo de sistemas, son los propios usuarios los que definen el conjunto de reglas (permisos y privilegios) sobre los recursos de los cuales son propietarios. De esta forma, un usuario puede decidir que recursos, siendo propietario del recurso, son accesibles y de qué forma por terceros. Un claro ejemplo de la implementación de este modelo son los sistemas GNU/Linux. Este tipo de sistemas son descentralizados, pues cada usuario toma decisiones sobre sus recursos.

3. Modelos de control de acceso basados en roles (en inglés, *Role-Based Access Control* [RBAC]): se basa en asignar roles a cada usuario y recurso del sistema. Cada rol recopila los privilegios y permisos asignados sobre los recursos del sistema. De esta forma, la gestión de las identidades dentro

de un sistema se convierte en una tarea más sencilla que en los modelos anteriores, pues se pueden manejar los grupos de permisos de forma conjunta. Por ejemplo, supóngase un rol Director que posee todos los permisos sobre los recursos del sistema, mientras que el rol Programador solo posee acceso a un número limitado de recursos. En caso de que se quiera registrar un nuevo usuario Programador en el sistema, bastará con asignarle el rol ya existente, en vez de tener que modificar los permisos de cada recurso del sistema para otorgarle los accesos correspondientes.

4. Modelos de control de accesos basados en atributos (en inglés, *Attribute-Based Access Control* [ABAC]): se basa en establecer la política de accesos en base a atributos de usuario, recurso y entorno. Estos atributos no son más que características que definen a cada uno de los agentes mencionados anteriormente. De esta forma, el control de accesos se realiza comprobando que cada uno de los atributos necesarios de una petición se cumplen en las reglas recopiladas en la política de accesos. Por ejemplo, supóngase un usuario que posee el atributo Director y que solicita acceso a un recurso con un atributo Confidencial y posee un atributo de entorno 11:00 pm. La política define que Director puede acceder a los recursos Confidencial, sin embargo, se tiene que cumplir una tercera regla para el atributo entorno que sea menor de 7:00pm. En este caso, se denegará el acceso, pues los atributos evaluados no se corresponden al completo con las reglas de la política de accesos. *eXtensible Access Control Markup Language* (XACML) es el estándar más extendido para implementar este tipo de modelo.

Cabe destacar que los modelos aquí expuestos, además de otros existentes, no son excluyentes entre sí y, por consiguiente, un mismo sistema puede implementar varios de ellos simultáneamente. Esta combinación de múltiples modelos resultará, con una buena implementación, en un sistema más seguro. Sin embargo, también aumenta la complejidad del modelo, creando reglas muy complejas que pueden afectar al rendimiento y a la mantenibilidad del mismo. Es por esto que un buen sistema de control de accesos ha de buscar un equilibrio entre seguridad y el resto de los requisitos, buscando siempre obtener un modelo lo más seguro posible siendo lo más sencillo posible. Un ejemplo de esta combinación de modelos, es el módulo de seguridad SELinux [28], el cual permite extender el control de acceso DAC de los sistemas Linux, implementando políticas MAC y RBAC.

2.1.3. Modelos para la gestión de identidades

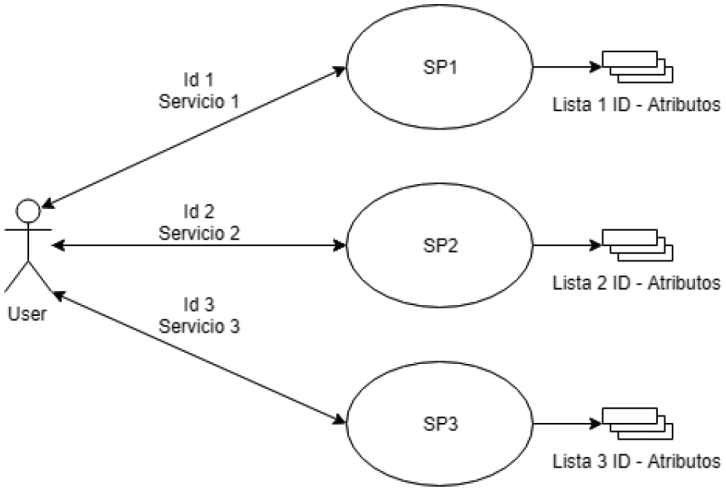
La gestión de identidades hace referencia a los sistemas y arquitecturas que se encargan de almacenar y administrar las identidades digitales dentro de un sistema de información.

El modelo Silo es la primera arquitectura que surgió para gestionar las identidades digitales [29]. En esta arquitectura, el sistema de gestión de identidades actúa tanto de SP como de IdP. En primer lugar, el sistema actúa de SP, ya que provee a los usuarios de un servicio, aplicación o recurso. En segundo lugar, actúa de IdP, ya que tiene como responsabilidad almacenar todas las identidades digitales del sistema, junto a sus credenciales y validar su autenticidad con el objetivo de poder gestionar y completar cualquier flujo de IAAA. De esta forma, en este tipo de sistemas, la figura del IdP y del SP se solapan en un único agente que se encarga de realizar ambas funcionalidades. Su funcionamiento se ve reflejado en la Figura 2.4.

Esta arquitectura para la gestión de identidades es la más antigua. Se implementó cuando el número de servicios o aplicaciones, a los que accedía un mismo usuario, era razonable y limitado y, por lo tanto, el propio usuario podía gestionar todas las identidades que manejaba a lo largo de todos estos servicios o aplicaciones. A medida que el número de servicios o aplicaciones ofertados fue aumentando, la usabilidad real de estos sistemas fue decreciendo. Esto dio paso a los sistemas de gestión de identidades centralizados.

Los sistemas de gestión de identidades centralizados surgieron con el objetivo de suplir las carencias del modelo Silo. De esta forma, tratan de unificar y centralizar en un único punto la gestión de identidades a lo largo de múltiples servicios dentro de un mismo dominio [29]. En esta ocasión, las figuras del IdP y del SP se desvinculan formando agentes totalmente distintos. De este modo, el IdP se encarga de almacenar y gestionar toda la información referente a las identidades digitales. En esta arquitectura, el SP delega todo el flujo de IAAA sobre el IdP. El funcionamiento de este tipo de arquitecturas

Figura 2.4. Modelo Silo para la gestión de identidades



se puede observar en la Figura 2.5. Como se puede observar, cuando un usuario inicia cualquier flujo de IAAA, lo inicia sobre el IdP, y es este último quien verifica y le otorga acceso a todos los SPs bajo los que opera. Esto permite que un mismo usuario pueda realizar procesos de IAAA sobre múltiples servicios, en el mismo dominio, utilizando únicamente una identidad digital. Este tipo de arquitecturas solventa los problemas encontrados para el modelo Silo. Sin embargo, este tipo de sistemas centralizan toda la seguridad en un único punto y, por consiguiente, si un atacante logra sobrepasar dicha barrera, obtendrá acceso no solo a un SP, sino a todos los SP en los que el usuario atacado tiene acceso.

Los sistemas centralizados permiten implementar el proceso de *Single Sign On* (SSO). El SSO unifica todos los puntos de acceso a múltiples SP en un único punto. Esto se consigue ya que el IdPs implementa una base de datos centralizada que contiene toda la información referente a las identidades digitales, incluidas sus credenciales, que recogen todos los SPs.

Existen multitud de estándares, protocolos y sistemas de gestión de identidades centralizados. Los más extendidos y utilizados hoy en día son *Lightweight Directory Access Protocol* (LDAP) [30], Kerberos [31] y *Remote authentication dial in user service* (RADIUS) [32].

La federación de identidades es el conjunto de arquitecturas y estándares que permiten distribuir de forma dinámica las identidades y su información a lo largo de múltiples dominios seguros [33]. Estas arquitecturas extienden a los sistemas centralizados, permitiendo que un usuario pueda realizar procesos

Figura 2.5. Modelo centralizado para la gestión de identidades

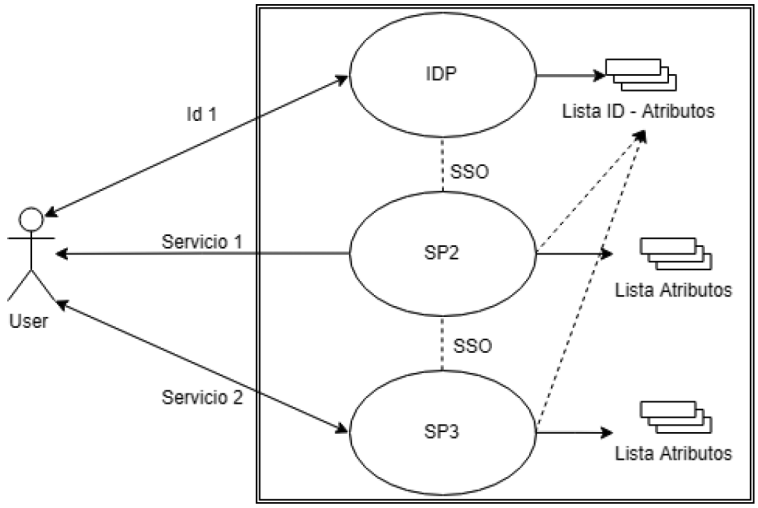
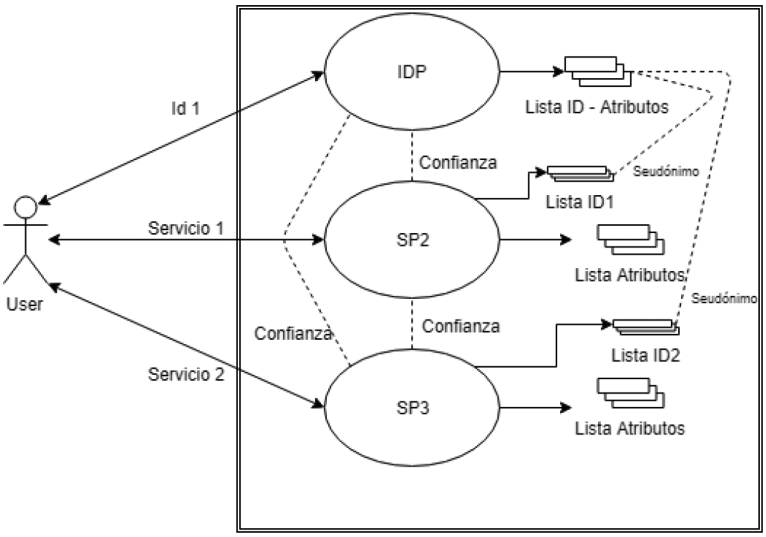


Figura 2.6. Modelo federado para la gestión de identidades



de IAAA sobre múltiples SP alojados en diferentes dominios utilizando un IdP externo y común, en el que se establece una relación de confianza. De esta forma, un SP crea un vínculo de confianza con un IdP externo y delega los procesos de IAAA sobre él. Estos vínculos de confianza forman círculos de confianza (en inglés, *Circle of Trust* [COT]). Una vez creados los COT, el IdP crea seudónimos para identificar y vincular a la identidades digitales que almacena y sus SPs correspondientes. En la Figura 2.6 se puede observar la arquitectura de este tipo de modelos.

Este tipo de modelos vienen siendo utilizados desde el año 2003, con la creación de *Security Assertion Markup Language* (SAML) [34]. Hoy en día, se han extendido y adoptado otros estándares como OpenID [19], OAuth [20] y OIDC [21]. De aquí en adelante, se analizan los estándares federados más utilizados hoy en día, haciendo especial hincapié en sus conceptos fundamentales y sus flujos de información.

2.1.4. Estándares federados para la gestión de identidades

OpenID

OpenID se creó en el año 2005 y se define como un *framework* abierto, descentralizado y gratis para la gestión de identidades [19]. Su desarrollo ha sido apoyado por grandes corporaciones como Google, Microsoft y IBM. Está orientado a solventar la autenticación en escenarios web.

OpenID define tres roles fundamentales que interactúan entre sí para lograr realizar el proceso de autenticación:

- *Relying Party (RP)*: es la parte en la que los otros roles confían. Su función principal es hacer de cliente, es decir, es la aplicación o servicio con la que el usuario final interactúa y por lo tanto con la que necesita autenticarse. El rol de la RP es el de SP.

- *OpenID Provider (OP)*: es el encargado de gestionar el ciclo de vida de una identidad digital. El rol del OP es el de IdP.

- *End User (EU)*: es el usuario final, es decir, es la entidad que posee una identidad digital dentro del OP e interactúa con la RP para realizar la operativa deseada.

El flujo de autenticación que utiliza OpenID se resume a continuación:

(A) El EU accede a la RP e inicia el proceso de autenticación presentando el identificador, previamente registrado, a la RP por medio del agente de usuario (p. ej. un navegador web).

(B) La RP recibe y normaliza el identificador recibido. De esta forma, extrae e identifica el OP que necesita el EU para lograr autenticarse.

(C) La RP y el OP establecen un código secreto compartido que es almacenado por la RP. Este código secreto se utiliza para verificar que el intercambio de mensajes entre ambas partes es correcto.

(D) La RP redirige al agente de usuario del EU al OP, previamente identificado, realizando una petición de autenticación.

(E) El OP valida las credenciales proporcionadas por el EU.

(F) El OP redirige el agente de usuario del EU de vuelta a la RP comunicándole si el proceso de autenticación ha quedado completado o si por el contrario ha fallado.

(G) RP La RP verifica que la información recibida es correcta y por tanto el proceso de autenticación ha quedado finalmente completado.

OAuth

OAuth es un *framework* de código libre para la gestión de identidades de forma federada. Su desarrollo empezó en el año 2006, logrando su primera versión

estable OAuth 1.0 en el año 2010, publicado como RFC 5849 [35]. Actualmente se encuentra en su versión OAuth 2.0 publicado como RFC 6749 [20].

OAuth 2.0 provee todos los mecanismos necesarios para poder realizar autorización en aplicaciones web, aplicaciones de escritorio y dispositivos inteligentes de forma federada. Cuando un usuario realiza una petición de acceso a un recurso protegido alojado en un SP, este le redirige al IdP externo, en el que confían tanto el usuario, como el SP. En este instante, el IdP verifica la identidad del usuario y procede a evaluar la petición, categorizándola como legítima en caso de que el usuario introduzca unas credenciales válidas y por consiguiente otorgándole acceso al recurso solicitado, o como no legítima, en caso contrario y, por consiguiente, negándole dicho acceso. Este proceso se utiliza exclusivamente para garantizar un proceso de autorización y, por lo tanto, OAuth 2.0 asume que el proceso de autenticación se realiza por otra vía, ya sea de forma local, con un modelo centralizado o de forma federada con otro estándar que lo permita. De aquí en adelante se va a detallar el funcionamiento de OAuth 2.0.

OAuth 2.0 define cuatro roles fundamentales que interactúan en cualquier flujo de autorización:

- *Resource Owner*: entidad propietaria de los recursos protegidos. Esta entidad puede ser el propio EU.

- *Resource Server*: es el servidor que almacena los recursos protegidos. Su funcionalidad es recibir y responder correctamente a las peticiones de acceso a los recursos protegidos.

- *Client*: es el cliente, es decir, es una aplicación que solicita acceso a los recursos protegidos. Por ejemplo, un navegador web que utiliza un usuario para acceder a un recurso. Dependiendo de su capacidad para mantener la confidencialidad y las credenciales del usuario pueden ser confidenciales o públicos.

- *Authorization Server*: servidor encargado de verificar los privilegios y roles de un *Client* con el objetivo de garantizar que solo los usuarios legítimos accedan a los recursos protegidos.

El modo en el que estos roles interactúan y se comunican entre sí, para lograr el proceso de autorización es por medio del uso de *tokens*. En OAuth 2.0 se distinguen fundamentalmente dos tipos de *tokens*:

- *Access token*: es el *token* de acceso. Lo utiliza el *Client* para acceder a los recursos protegidos alojados en el *Resource Server*. En otras palabras, este *token* sustituye a las credenciales del usuario en un proceso convencional de autorización. Este *token* es generado por el *Authorization Server*.

- *Refresh token*: es el *token* de refresco. Lo utiliza el *Client* para solicitar un nuevo *access token* cuando el vigente ha sido invalidado.

El flujo de autorización de OAuth 2.0, que se lleva a cabo entre los diferentes roles para lograr el proceso de autorización, se puede ver ilustrado en la Figura 2.7.

(A) El *Client* solicita autorización al propietario del recurso protegido (*Resource Owner*).

(B) El *Client* recibe la concesión de autorización (en inglés, *Authorization Grant*) que le otorga permisos para poder acceder al recurso protegido.

(C) El *Client* utiliza esta concesión para solicitar al *Authorization Server* el *access token*.

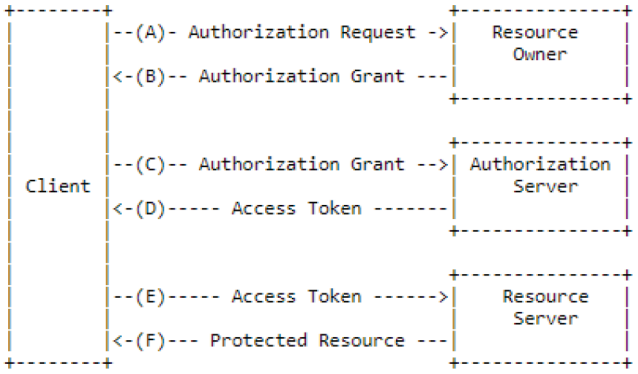
(D) El *Authorization Server* comprueba que la concesión es válida, y en caso afirmativo procede a devolverle el *access token*.

(E) El *Client* utiliza el *access token* para solicitar acceder al recurso protegido alojado en el *Resource Server*.

(F) El *Resource Server* valida el *access token* y devuelve al *Client* el recurso solicitado.

Las concesiones de autorización (paso B y C de la Figura 2.7) definen las tareas e interacciones que se han de realizar, de forma secuencial, entre los diferentes roles, con el objetivo de garantizar que se logre un proceso de autorización. De este modo, dependiendo de las diferentes casuísticas

Figura 2.7. Flujo de autorización de OAuth



Fuente: https://tools.ietf.org/html/rfc6749

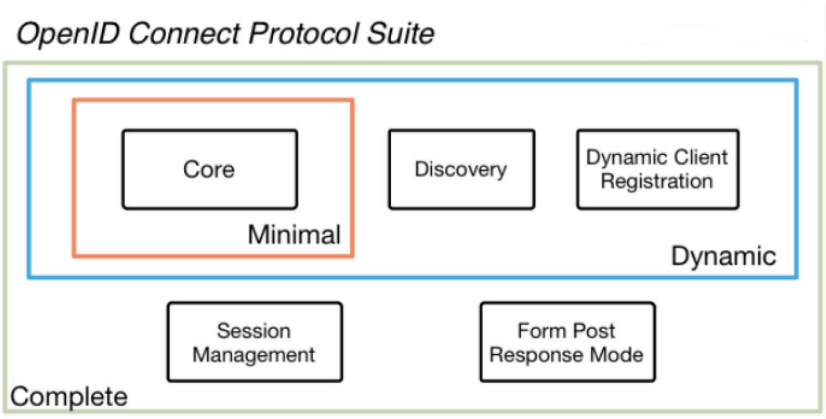
y escenarios posibles, los diferentes roles han de seguir una de las posibles concesiones de autorización disponibles. Dentro del estándar de OAuth, se definen, cuatro tipos de concesiones: *Authorization Code*, *Implicit*, *Resource Owner Password Credentials* y *Client Credentials*.

OpenID Connect

OIDC es una especificación para manejar la gestión de identidades federadas creada en el año 2014 por la OpenID Foundation [21]. Este estándar permite realizar procesos tanto de autenticación como de autorización. Se posiciona como una simbiosis entre OpenID y OAuth 2.0. Toma como punto de partida OAuth 2.0, permitiendo así, realizar procesos de autorización y auditoría. Por otro lado, OIDC integra y extiende a OpenID, con el objetivo de dotar al estándar resultante de los procesos de autenticación. Todos estos procesos pueden ser implementados y accesibles por medio de una APIs, lo cual hace que su uso e implantación sea muy amigable para los desarrolladores.

OIDC se presenta como un protocolo ligero que permite que todo tipo de clientes (p. ej. aplicaciones web y aplicaciones móviles), puedan verificar y consultar información de las identidades digitales. Además, es flexible y extensible, permitiendo incorporar características adicionales como servicios de encriptación de la información, servicios de gestión de sesiones o el descubrimiento de proveedores de OpenID Connect de forma automática. OIDC esta compuesto por tres módulos funcionales que recogen todas estas funcionalidades (ver Figura 2.8):

Figura 2.8. Módulos de OpenID Connect



Fuente: https://openid.net/connect

- *Core*: define la funcionalidad básica y mínima para resolver los procesos de IAAA. Utiliza tres puntos de acceso: *Authorization Server Endpoint*, *token Endpoint* y *UserInfo Endpoint*.

- *Dynamic*: extiende el Core para incluir el servicio de descubrimiento dinámico de registro de clientes (*Discovery Dynamic Client Registration*). Utiliza dos nuevos punto de acceso: *Discovery Endpoint* y *Client Registration Endpoint*.

- *Complete*: añade los servicios de gestión de sesión y el *From Post Responde Mode* para codificar los parámetros de respuesta de autorización como valores de formularios HTML.

En OIDC se utilizan los tres mismos roles que se definen para OpenID. Estos son: EU (el usuario final), RP (la parte confiable o SP) y OP (OpenID Provider o IdP). Hace uso de los *tokens* definidos en OAuth 2.0, esto es, el *access* token, y el *refresh token*. Además, define un tercer *token* llamado *ID token*. El *ID token* contiene los atributos específicos asociados a una entidad (en inglés, *claims*) y que son necesarios durante el proceso de autenticación.

Estos roles interactúen entre sí por medio de peticiones y respuestas a dichas peticiones. Se distinguen seis tipos de ellas:

- Petición de autorización/autenticación: la realiza la RP al *Authroization Server Endpoint*, que se encuentra en el OP. El objetivo es solicitar el *Authorization Code*, *access token* y/o el *ID token* dependiendo del flujo de información a realizar. Se realiza mediante HTTP GET/POST.

- Respuesta de autorización/autenticación: Se envía desde el *Authorization Server Endpoint* a la URI de redirección indicada por la RP en la petición de autorización. Incluye el *Authorization Code* y los *tokens* solicitados, así como el estado y opcionalmente la caducidad de los *tokens*.

- Petición de tokens: Este realizada por la RP al *token Endpoint*, que se encuentra en el OP, para solicitar los *tokens*. Se realiza por medio de HTTP POST.

- Respuesta de Tokens: el *token Endpoint* realiza esta respuesta a la RP devolviendo los *tokens* solicitados. Estos *tokens* son el *access* token, el *ID token* y el *refresh token*.

- Petición de información del usuario: la RP realiza esta petición al *UserInfo Endpoint*, que se encuentra en el OP, para pedir información del EU. Se codifica mediante una petición HTTP GET/POST, e incluye el *access token* del EU con el objetivo de poder identificarlo.

- Respuesta de información del usuario: el *UserInfo Endpoint* devuelve en formato JSON la información solicitada referente al EU identificado por el acess *token* recibido.

OIDC contempla tres flujos de información, muy parecidos a los proporcionados por OAuth 2.0. Estos flujos son: *Authorization Code, Implicit y Hybrid*. En primer lugar, el flujo *Authorization Code* es el análogo al de OAuth 2.0. Este flujo está representado en la Figura 2.9 y consta de los siguientes pasos:

0. El EU comienza a interaccionar con la RP iniciando el flujo de autorización/autenticación.

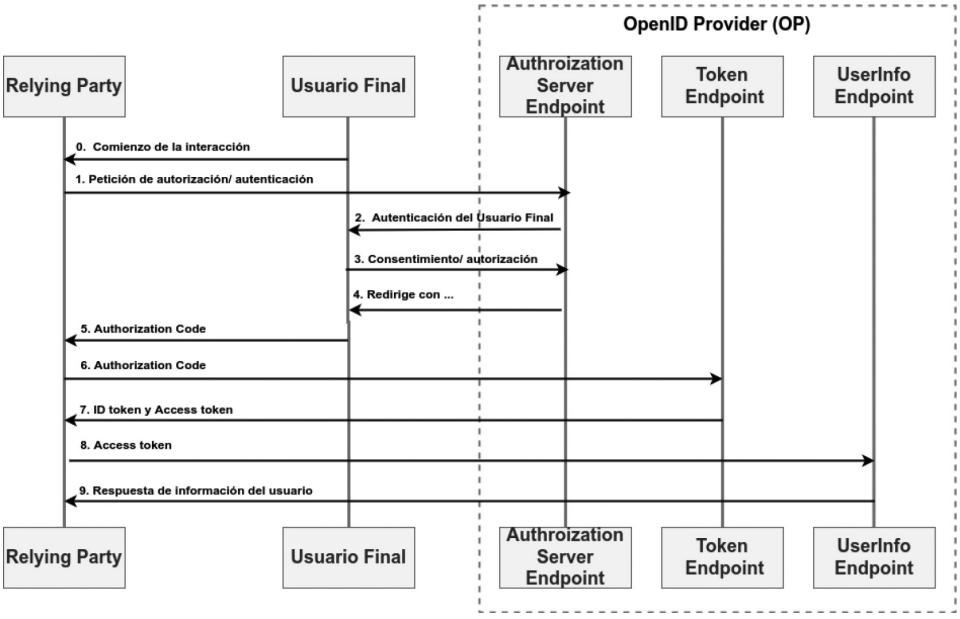
1. La RP envía una petición de autorización/autenticación al *Authorization Server Endpoint*.

2. El *Authorization Server Endpoint* se comunica con el EU para autenticarle.

3. El EU proporciona las credenciales al *Authorization Server Endpoint* o le da el consentimiento necesario.

4. El *Authorization Server Endpoint* valida las credenciales y en caso satisfactorio continua con el flujo y redirige al EU a la RP.

Figura 2.9. Flujo *Authorization Code* de OpenID Connect



5. El *Authorization Server Endpoint* envía la respuesta de autorización/autenticación proporcionando el *Authorization Code* a la RP.

6. La RP utiliza el *Authorization Code* para enviárselo al *token Endpoint* del OP por medio de una petición de tokens.

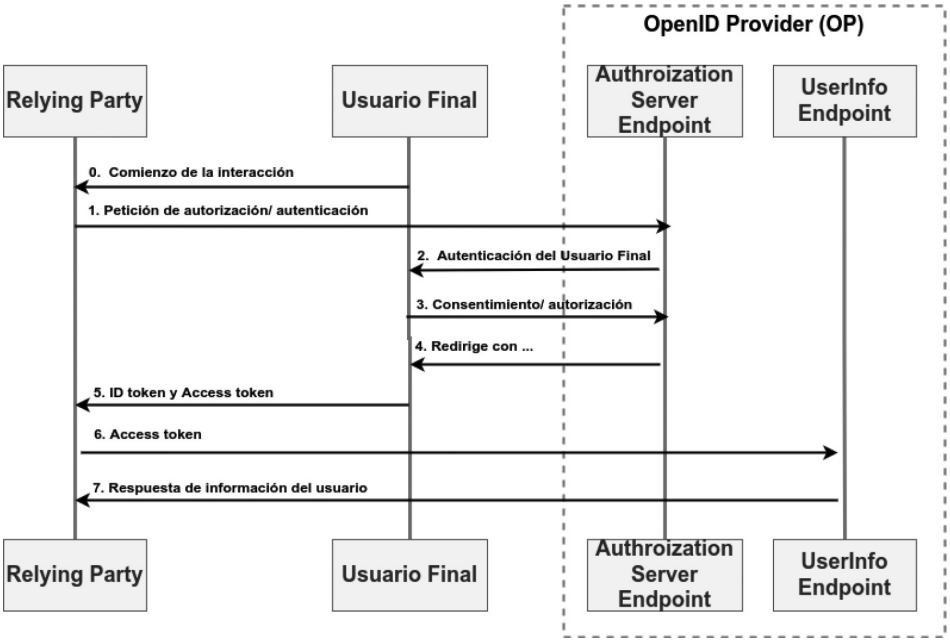
7. El *token Endpoint* del OP valida el *Authorization Code* y en caso satisfactorio proporciona a la RP el *ID token* y el *access token* por medio de la respuesta de tokens.

8. De forma opcional, la RP utiliza el *access token* para enviar una petición de información del usuario.

9. El *UserInfo Endpoint* devuelve la información solicitada.

Por otro lado, el flujo *Implicit* se puede ver detallado en la Figura 2.10. La única diferencia con el flujo anterior viene en el paso 4 y 5, pues la RP recibe directamente el *ID token* y el *access token* ya que el *Authorization Server* puede

Figura 2.10. Flujo *Implicit* de OpenID Connect



determinar que dicho usuario ya estaba autenticado o autorizado previamente. Este flujo se consigue gracias a que el *Authorization Server* es capaz de verificar que el usuario esta previamente autenticado o autorizado por el uso de algún lenguaje de *Scripting* (p ej. *Javascript*). Este proceso se realiza, normalmente, recuperando una *cookie* por medio del agente de usuario. Este flujo es menos seguro que el *Authorization Code* pues un atacante podría obtener el *ID token* y el *access token* del usuario simplemente secuestrando una sesión activa del usuario, sin embargo, permite agilizar mucho el proceso cuando el *Authorization Server* y el EU ya confían en el cliente.

Finalmente, el flujo *Hybrid* es una combinación de los dos flujos anteriores, es decir, del *Authorization Code* y del *Implicit*. En este caso, se proporciona flexibilidad y, por lo tanto, dependiendo del tipo de conexión que se quiera utilizar y de la configuración del IdP, los *tokens* pueden ser proporcionado por el OP o por el *token Endpoint*.

En la Tabla 2.2 se puede observar las principales similitudes y diferencias entre los tres flujos.

Tabla 2.2. Comparativa de flujos de información en OIDC

	Authorization Code	***Implicit***	***Hybrid***
Authorization Endpoint devuelve *tokens*	No	Si	Si
Token Endpoint devuelve *tokens*	Si	No	Si
Los *tokens* pasan por el agente de usuario	No	Si	Si
El cliente se puede autenticar	Si	No	Si
Puede usar *refresh tokens*	Si	No	Si

2.1.5. Amenazas y soluciones actuales en la federación de identidades

Las especificaciones federadas son muy utilizadas hoy en día, sin embargo, como cualquier especificación o tecnología, existen amenazas asociadas tanto a la propia especificación como a las implementaciones de las mismas [36], [37]. Además, existen multitud de vulnerabilidades conocidas que pueden ser explotadas por cualquier atacante.

Las principales líneas de investigación y soluciones planteadas hasta el momento para tratar de solventar estas amenazas suelen estar orientadas al enriquecimiento de las peticiones y de los *tokens* empleados, a realizar una mejora en la gestión de las sesiones de usuario, a la mejora de la propia implementación por medio de las APIs y SDKs ofrecidas, a la mejora de los flujos de información, al uso de criptografía a diferentes niveles o a la creación de políticas o sistemas de reputación [17].

Por ejemplo, profundizando en la especificación de OAuth, en [38] tratan de identificar todas las ambigüedades o aspectos que no quedan claros dentro de la propia especificación. Posteriormente, analizan implementaciones específicas para ver como se han solventado dichos aspectos, llegando a la conclusión de que casi un 60% de las implementaciones no han sido implementadas correctamente y por lo tanto son vulnerables. En [39] demuestran que, por motivos de diseño, OAuth es vulnerable a sufrir suplantación a nivel de aplicación debido a los propios flujos de autorización y tipos de tokens de los que se disponen. En [40] proponen un modelo adaptativo que permite detectar a gran escala vulnerabilidades existentes y nuevas en las implementaciones de OAuth. Además proponen mitigar algunas de las nuevas vulnerabilidades encontradas que permiten materializar ataques como *Cross Site Request Forgery* (CSRF) y ataques de suplantación de identidad por medio de la mejora de los SDKs y de los propios tokens de la especificación. En [41] se propone modificar el estándar de OAuth para unificar todos los clientes externos y los flujos de información en uno común, con el objetivo de simplificar la configuración necesaria para su implementación. Además, se propone utilizar AMF para autenticar clientes externos, y firmas digitales y criptografía para mitigar posibles riesgos y vulnerabilidades asociados al mal uso de los *tokens*. En [42] se analizan las principales APIs proporcionadas por los mayores IdPs y los flujos de autorización de OAuth para analizar las posibles implicaciones a la privacidad de los usuarios finales.

En cuanto a la especificación de OIDC, en la propuesta [36] descubren que multitud de ataques ya existentes para otros protocolos de SSO son igualmente aplicables realizando pequeñas modificaciones de los mismos. Además, proponen dos nuevos ataques para materializar nuevas vulnerabilidades asociadas a los propios flujos de la especificación. Finalmente, proponen soluciones para mitigar estos ataques basadas en la mejora del propio estándar, las cuales han sido actualmente incluidas en el propio estándar, y soluciones de menos impacto basadas en la mejora de los propios *tokens* y en la mejora de las implementaciones. En [16] realizan un análisis formal de la seguridad de la especificación y proponen métodos para solventar las vulnerabilidades encontradas basándose en la mejora de las propias implementaciones modificando los flujos de información y los *tokens*. En [43] y [44] tratan de mejorar la privacidad de OIDC por medio del uso de técnicas criptográficas y la modificación de los propios flujos de información. En [45] proponen una

serie de políticas y técnicas criptográficas en las distintas comunicaciones de los agentes para mejorar la privacidad de los usuarios. En [17] analizan amenazas tanto de la seguridad como de la privacidad y proponen diversas técnicas para mitigarlas basadas en la mejora de los tokens, en la mejora de la implementación, en la mejora de los flujos de información, en aspectos de criptografía y en la creación de políticas y sistemas de reputación. En [46] se propone un método para mejorar la privacidad de los usuarios basado en realizar pequeñas modificaciones a los flujos de información ya existentes. En [47] proponen un servicio novedoso de *tokens* para mantener los *access tokens* durante un mayor periodo de tiempo aumentando la seguridad de los tokens de larga duración (usualmente menos seguro que los que tienen un ciclo de vida corto). En [48] se proponen una serie de buenas prácticas para mejorar la seguridad de OAuth y OIDC en aplicaciones nativas para clientes Android. Además demuestran que la gran mayoría de las implementaciones actuales son vulnerables a todo tipo de ataques como la suplantación de identidad debido a una mala implementación de las buenas prácticas propuestas.

En la Tabla 2.3 se pueden observar las características principales de los trabajos analizados anteriormente.

Por último, cabe destacar que la rama del análisis de comportamiento se ha posicionado como una línea de investigación muy recomendable y adoptada para mejorar los sistemas de control de accesos y gestión de identidades [4]. Sin embargo, tal y como se ha podido ver hasta ahora, los trabajos que tratan de mejorar los niveles de seguridad de los estándares de gestión de identidades federados no integran o implementan estas técnicas. Además, estas técnicas son de especial interés para lograr implementar autenticación continua, un recurso poco utilizado hasta ahora en la federación de identidades.

2.2. Análisis de comportamientos

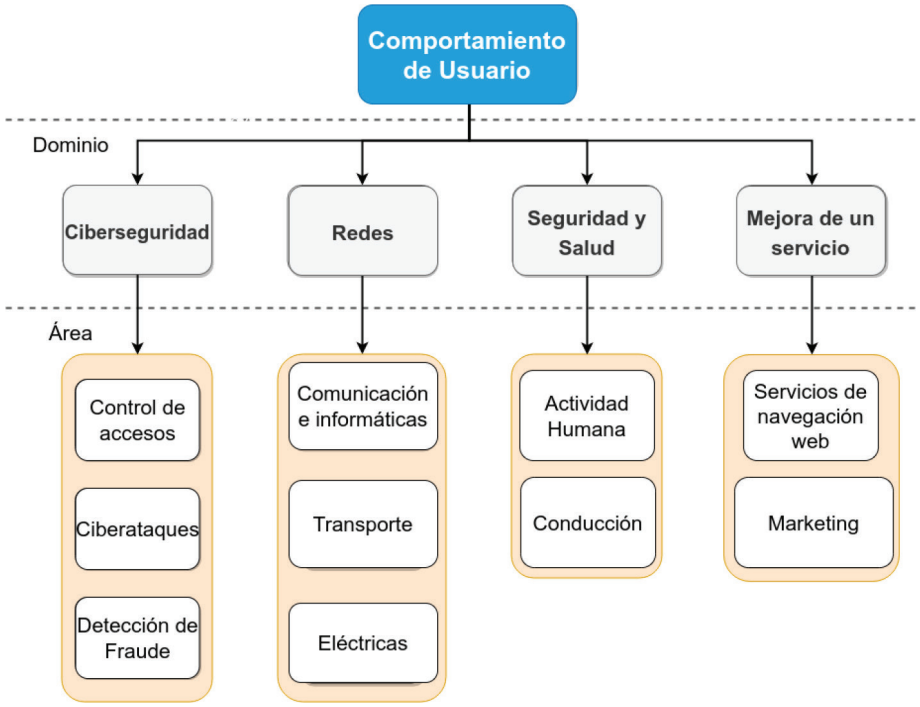
En el ámbito tecnológico, el análisis de comportamientos se centra en analizar, modelar y predecir los comportamientos pasados, presentes y futuros de los usuarios o entes que interactúan con un sistema de información, con el objetivo de obtener un beneficio de negocio [4]. Es comúnmente conocido por sus siglas en inglés, *User and Entity Behavior Analysis* (UEBA). El uso de técnicas de análisis de comportamientos aporta grandes ventajas para conseguir multitud de objetivos, a lo largo de una gran variedad de dominios de aplicación. En la Figura 2.11, se pueden observar los cuatro grandes dominios de aplicación de este área hoy en día, y sus áreas específicas de aplicación. Estos dominios son: redes, seguridad y salud, mejora de un servicio y ciberseguridad.

En el dominio de las redes, el análisis de comportamientos es una herramienta muy efectiva a la hora de mejorar las redes de comunicación, de transporte

Tabla 2.3. Trabajos de la literatura sobre amenazas y mejoras de los esquemas de gestión de identidades federados. La columna Ámbito representa si el trabajo se centra en mejorar la seguridad (*S*) o la privacidad (*P*). Las columnas Token, Criptografía, Implementación, Políticas/Reputación y Flujo representan los elementos donde se centran las soluciones propuestas de los trabajos analizados

Trabajo	Especificación	Ámbito	*Token*	Criptografía	Implementación	Políticas/ Reputación	Flujo
[37]	OAuth	S	-	-	-	-	-
[38]	OAuth	S	-	-	-	-	-
[39]	OAuth	S	-	-	-	-	-
[40]	OAuth	S	✓	-	✓	-	-
[41]	OAuth	S	✓	✓	-	-	✓
[42]	OAuth	P	-	-	-	✓	-
[16]	OIDC	S	✓	-	✓	-	✓
[36]	OIDC	S	✓	-	✓	-	-
[43]	OIDC	P	-	✓	-	-	✓
[44]	OIDC	P	-	✓	-	-	✓
[45]	OIDC	P	-	✓	-	✓	-
[17]	OIDC	S/P	✓	✓	✓	✓	✓
[46]	OIDC	P	-	-	-	-	✓
[47]	OIDC	S	✓	✓	✓	-	✓
[48]	OAuth /OIDC	S	-	-	-	-	-

Figura 2.11. Dominios y áreas específicas de aplicación de los trabajos de análisis de comportamiento



y redes eléctricas. En este ámbito, el objetivo suele ser mejorar la eficiencia de las distintas dichas redes, por ejemplo, extrayendo patrones de uso que permiten establecer distintos perfiles de usuario. Estos patrones se pueden utilizar para detectar posibles congestiones en la red o cuellos de botella, permitiendo desviar el tráfico de forma óptima dependiendo del perfil de usuario, tanto en redes informáticas [49], como en redes de transporte público o privado [50], [51]. Otra aplicación, es la predicción del precio del mercado eléctrico con el objetivo de que los productores eléctricos puedan ganar las subasta energética a sus competidores y obtener un mayor beneficio de ello [52]. En este área, con la incorporación de los contadores inteligentes, también se utilizan estas técnicas para mejorar la eficiencia de la propia red y por lo tanto poder ofrecer precios más competitivos a los consumidores [53].

En el dominio de la seguridad y salud, el análisis de comportamientos es de gran utilidad para la detección temprana de ciertos factores de interés que pueden suponer un riesgo para la salud o la seguridad de un individuo. Por ejemplo, con la incorporación de sensores en una vivienda se puede detectar si una persona ha sufrido un accidente o una caída en el caso de personas mayores [54], [55], o se puede detectar si una persona está perdiendo capacidad

cognitiva debido al cambio de su comportamiento [56]. Más específicamente relacionado con la seguridad, hoy en día multitud de vehículos disponen de sistemas basados en el análisis de comportamiento para detectar situaciones adversas, como un posible accidente [57] o fatiga y cansancio [58].

En el dominio de la mejora de un servicio, el análisis de comportamientos permite generar perfiles de usuario que son utilizados para entender las carencias del servicio y, por consiguiente, poder mejorarlo u obtener un beneficio por ello. Algunos ejemplos son, los sistemas de recomendación que permiten aumentar las ventas de un comercio tanto electrónico como físico [59]. La fidelización de clientes o el lanzamiento de campañas específicas de publicidad son algunos otros ejemplos en los que las técnicas de análisis de comportamientos son efectivas en este dominio [60].

En el dominio de la ciberseguridad, el análisis de comportamientos se utiliza para elaborar sistemas de control de accesos, para detectar ciberataques y para la detección de fraude. Los sistemas de control de accesos basados en técnicas de análisis de comportamientos se fundamentan en analizar los patrones de comportamiento de un usuario o entidad específica con el objetivo de poder realizar los procesos de IAAA frente a un sistema de información [61], [62]. En el campo de la detección de ciberataques, el análisis de comportamientos se utiliza para detectar patrones anómalos, por ejemplo, durante la ejecución de un software que pueden causar daños en el sistema [63] o en las comunicaciones de una red [64]. Por último, el análisis de comportamiento es útil en la detección de fraude. En este ámbito, el análisis de comportamiento en el uso de tarjetas de crédito puede ayudar a prevenir la realización de transacciones fraudulentas [65].

De aquí en adelante, se analizan los trabajos relacionados con el análisis de comportamiento en el dominio de la ciberseguridad y específicamente en el área del control de accesos.

2.2.1. Análisis de comportamientos para el control de accesos

En esta sección se analizan los trabajos del estado del arte que utilizan técnicas de análisis de comportamiento para solventar el control de accesos. Los trabajos, aquí recopilados, se centran mayoritariamente en la autenticación, tanto progresiva como autenticación continua. El objetivo de estas propuestas es tratar de modelar la información de comportamiento para extraer y detectar patrones intrínsecos a cada usuario. Estos patrones se utilizan posteriormente para evaluar nuevas muestras, pudiendo determinar si concuerdan, es decir, pertenecen a un usuario legítimo, o por el contrario si son diferentes, es decir, son anomalías de comportamiento y por lo tanto es probable que no pertenezcan al usuario legítimo y puedan suponer una amenaza

para la seguridad. En este último caso, el sistema debe evaluar los riesgos y proceder a realizar las contramedidas necesarias, como solicitar al usuario nuevamente sus credenciales, o utilizar un segundo factor de autenticación con el objetivo de corroborar que el usuario es legítimo.

Los trabajos del estado del arte en este ámbito se dividen principalmente en dos categorías dependiendo del dispositivo donde se aplican los modelos de análisis de comportamientos: teléfonos inteligentes y ordenador. Los modelos de aprendizaje máquina que se aplican en ambas categorías suelen ser de la misma naturaleza y por lo tanto muy parecidos. Sin embargo, los procesos de recopilación de información, las fuentes de información utilizadas (distintas dinámicas de comportamiento) y su integración final (debido a los recursos limitados de los teléfonos inteligentes) son más dispares.

En el caso de los teléfonos inteligentes, las fuentes de información son principalmente los sensores que están habitualmente integrados en dichos dispositivos. Estos sensores se pueden categorizar en cuatro grandes grupos [62]: movimiento (p. ej. acelerómetro y giroscopio), entorno (p. ej. luz ambiente y temperatura), posición (p. ej. GPS y brújula) y de pantalla (p. ej. presión y capacitivo). El objetivo de los trabajos que se engloban en este ámbito suele ser detectar anomalías en el comportamiento del usuario a partir de la información recogida por estos sensores. Estas anomalías se deben principalmente a la suplantación de identidad, ya sea por un robo de credenciales (robo de los autenticadores que pertenecen al usuario legítimo) o por un secuestro de sesión (obtención de una sesión activa perteneciente a un usuario legítimo para lograr un acceso no autorizado). Además, los trabajos categorizados aquí, se pueden dividir a su vez en los que se centran en la autenticación progresiva y los que se centran en la autenticación continua.

En la autenticación progresiva, las propuestas suelen estar centradas en evaluar una determinada petición de acceso a una aplicación o recurso restringido, y categorizarla en función de los valores recopilados por los sensores. En este ámbito, normalmente se suelen utilizar los sensores de posición como el GPS, y los sensores de pantalla. Un ejemplo de la utilización de sensores de posición se encuentra en [66], donde los usuarios se agrupan de acuerdo a un algoritmo de *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) basándose en sus posiciones. Posteriormente, los grupos y las transiciones entre grupos se modelan como un proceso de Markov, permitiendo utilizar un *Hidden Markov Model* (HMM) para determinar si una petición es legítima o no. Por otro lado, en [67] utilizan las dinámicas de pulsación de teclado recopiladas por los sensores de pantalla. Esta información longitudinal se agrupa en ventanas para poder comparar las dinámicas de comportamiento entre sí. De este modo, las nuevas dinámicas que se quieren evaluar se comparan con el histórico del usuario, pudiendo así obtener la distancia entre ellas y clasificarlas en legítimas o impostoras en función de un umbral.

En la autenticación continua, existe más variabilidad respecto a las fuentes de información. En este caso, los sensores de movimiento, entorno y los sensores de la pantalla suelen ser los más utilizados. Independientemente de la fuente, cabe destacar que la naturaleza de los datos recogidos es la misma, es decir, son datos longitudinales. En primer lugar, se va a considerar los trabajos que utilizan información recogida de los sensores de pantalla. Normalmente, el tratamiento de estos datos comienza con la preparación y limpieza de los mismos, esto es, con los procesos de normalización y estandarización [61], generación de datos artificiales [68] y extracción de descriptores y medidas de centralidad y de dispersión (p. ej. media y desviación típica). Además, existen algunos trabajos que también utilizan la lógica difusa [69] para extraer conocimiento [70]. Normalmente, los trabajos en este ámbito suelen representar la información en forma de secuencias. El siguiente paso que siguen es modelar la información limpia y procesada. Para ello se utilizan múltiples algoritmos de aprendizaje máquina. Cabe destacar el uso de *K-Nearest Neighbors* (KNN) y de *Support Vector Machine*s (SVMs) cuando las secuencias generadas son lo suficientemente largas [61]. Por otro lado, *Naïve Bayes* (NB), *Bayesian Network* (BN) y *Neural Network* (NN) suelen funcionar correctamente para conjuntos de datos pequeños[71].

En el caso de los trabajos que utilizan información recogida del acelerómetro y del giroscopio, los algoritmos de clasificación de clase única son los más utilizados. Por ejemplo, en [68] utilizan técnicas de aumento de datos, extracción exhaustiva de características y el algoritmo de clasificación *One-Class SVM* (OC-SVM) para definir los patrones de uso de cada usuario. Estos patrones extraídos, permiten distinguir entre las dinámicas de comportamiento que se consideran normales para un usuario, y las muestras atípicas, las cuales se clasifican como muestras no pertenecientes al usuario legítimo. Además, existen otros trabajos que tratan de detectar estos mismos patrones utilizando un HMM de clase única en múltiples escenarios como, sostener el dispositivo, coger el dispositivo desde una mesa y sostener el dispositivo mientras se camina [62]. Cabe destacar la propuesta [72], donde modelan información de estos sensores combinados con técnicas de criptografía con el objetivo de garantizar la privacidad de los usuarios, no viéndose afectado significativamente el rendimiento en cuanto a eficacia y eficiencia de los modelos propuestos.

Para los sensores de ambiente, normalmente se utilizan algoritmos como KNN, NB y *Hoeffding Adaptive Trees* (HAT). Estos sensores, además, se suelen utilizar en combinación con otro tipo de sensores e indicadores como, por ejemplo, el uso de la batería con el objetivo de mejorar los modelos previamente desarrollados. Estas propuestas suelen ser las menos invasivas para la privacidad de los usuarios [73].

Por otro lado, en los trabajos que se centran en el dispositivo del ordenador personal, también se analiza tanto la autenticación progresiva, como la auten-

ticación continua. Para el primer caso, las principales fuentes de información son los registros de navegación y la información recogida de los sistemas de gestión de identidades, es decir, registros de peticiones de accesos a recursos. En el caso de la autenticación continua, se suelen utilizar las dinámicas de comportamiento recogidas desde el teclado y las dinámicas de comportamiento recogidas del ratón.

La mayoría de los trabajos que utilizan el ordenador personal, se centran en agrupar las interacciones de los usuarios legítimos, con el objetivo de definir los comportamientos esperados para cada usuario y poder así evaluarlos frente a las nuevas muestras. Por ejemplo, para el caso de los registros de navegación, estos se suelen agrupar en sesiones. Cada una de estas sesiones está formada por una secuencia de acciones, que recoge las dinámicas de comportamiento decada usuario durante un intervalo de tiempo. De esta forma, por ejemplo, se utilizan medidas simples de similitud [74], SVM [75] y el modelo de Markov [76] para poder compararlas y clasificarlas.

En el caso de las propuestas en las cuales los datos provienen de sistemas de gestión de identidades, los datos normalmente consisten en secuencias de peticiones de acceso a aplicaciones, servicios o recursos. El principal objetivo de estos trabajos es determinar la legitimidad de una petición teniendo en cuenta los atributos de la misma, y poder establecer el riesgo asociado. Al igual que en el caso anterior, las aproximaciones en este ámbito suelen utilizar la información legítima para alimentar un algoritmo de aprendizaje máquina. Posteriormente, este algoritmo se utiliza para comparar las nuevas muestras, y categorizarlas en legítimas o no legítimas acorde a un umbral de decisión. De esta forma, cuando una nueva muestra sobrepasa el umbral, el sistema lanza una alerta que puede ser utilizada para avisar a los administradores del mismo, o directamente proceder a tomar una contramedida de forma automática como, por ejemplo, rechazar la petición. Algunos ejemplos representativos de algoritmos que se utilizan en este ámbito son SVM [77], NB [78], técnicas de reducción de la dimensionalidad [79] y OC-SVM [80].

Como se ha podido observar hasta ahora, uno de los factores que más diferencian a las propuestas analizadas es la fuente de información que se utiliza para diseñar el modelo de análisis de comportamiento. Es por esto que es de gran importancia analizar en detalle los trabajos que encajan en cada una de estas fuentes de información específicas.

2.2.2. Fuentes de información específicas y su combinación

Los trabajos aquí presentados se han categorizado en función de la fuente de información utilizada para modelar el comportamiento de los usuarios. De este modo, se presentan tres categorías de acorde a los intereses de la

presente investigación. Estas fuentes de información son: teclado, ratón y combinación de información.

En cuanto a la categoría del teclado, los primeros trabajos en el área empezaron a surgir en 1980 con el primer análisis de las dinámicas de comportamiento asociadas a dicha fuente de información [81]. Posteriormente, los trabajos en este área empezaron a utilizar clasificadores bayesianos [82], NNs [83] o técnicas *clustering* [84] para modelar el comportamiento de los usuarios y poder detectar así anomalías en el comportamiento de los usuarios. Estas anomalías entonces podían ser categorizadas como posibles brechas de seguridad. Estos trabajos, aún siguen siendo de gran utilidad y permiten hoy en día obtener resultados satisfactorios a la hora de detectar comportamientos sospechosos.

Con el paso del tiempo han ido surgiendo nuevos trabajos que han ido utilizando otras técnicas o han probado más clasificadores para solventar el mismo problema. Por ejemplo, en [85] se utilizan 14 clasificadores, incluyendo algunos basados en la distancia de Mahalanobis, la distancia de Manhattan, KNN, NNs, K-means, lógica difusa y SVMs, para comparar y analizar la eficacia de dichos clasificadores a la hora detectar impostores cuando ingresan una cierta contraseña predefinida. En [86] se utiliza una *Ant Colony* (AC) para realizar un primer paso de selección de características, de esta forma, se pueden ordenar por importancia y seleccionar las más representativas, aunque no sean las más convencionales en la literatura. En el trabajo [87] se presenta un método novedoso para seleccionar las características más representativas, de forma independiente para cada usuario en particular, y posteriormente generar un modelo independiente para cada uno de ellos. Esta propuesta se basa en utilizar un modelo de estimación de densidades gausiano, Parzen window density estimation, OC-SVM, KNN y K-means. En [88] se demuestra que, para ciertos casos de uso, las variables no agregadas son más discriminatorias a la hora de comparar dinámicas de comportamiento recogidas del teclado. Utilizando este tipo de variables, se entrenan NB, *Tree Augmented Naïve Bayes* (TANB), KNN y modelos de regresión logística para detectar usuarios impostores. Cabe destacar que los mejores resultados los obtienen cuando combinan vectores de variables tanto agregadas como variables no agregadas de forma simultánea. En [89], se propone una técnica novedosa para transformar los vectores de características en espectrogramas de frecuencias, consiguiendo así transformar los datos longitudinales (es decir, señales) en una imagen. Posteriormente, utilizan una NN basada en la optimización de Gauss-Newton para clasificar estas imágenes y poder determinar así, si el vector de características original es de un impostor o por el contrario pertenece a un usuario genuino. En la propuesta [90], se utilizan NN convolucionales y recurrentes de forma combinada para generar un modelo efectivo para el conjunto de datos, bien conocido en la literatura, de Buffalo [91]. En [92] se considera utilizar *Kernel Density Estimation* (KDE) para compararse contra algoritmos populares en la literatura en este ámbito,

en multitud de conjuntos de datos como el de Clarkson, Torino y Buffalo. Por último, en [93], proponen utilizar una métrica basada en *Instance-based Tail Area Density* (ITAD) para reducir el número de interacciones necesarias para autenticar a un usuario, de esta forma, mejoran la eficiencia de los modelos y reducen latencias, no afectando a la eficacia.

El análisis de las dinámicas de ratón con el objetivo de autenticar usuarios comenzó en los años 2000 [94]. Posteriormente, en el trabajo [95], se empiezan a considerar variables relevantes como la velocidad de movimiento, la dirección, el tipo de acción que se realiza (movimiento, clic, desplazamiento de la rueda), la distancia recorrida y el tiempo transcurrido entre acciones. Este proceso de extracción de características se utiliza para generar un vector de variable que alimenta una NN obteniendo resultados satisfactorios. En el trabajo [96], también utilizan NN, en este caso convolucionales, recurrentes y un modelo híbrido que considera ambos tipos, basándose en la propagación de la relevancia por capas. Estos algoritmos son evaluados utilizando múltiples conjuntos de datos bien conocidos de la literatura. En [97], las variables comúnmente utilizadas en la literatura se categorizan en holísticas y procedimentales. Posteriormente, comparan que tipo de características son más discriminatorias a la hora de autenticar usuarios utilizando una OC-SVM. La investigación propuesta en [98] utiliza el algoritmo de *Progress-Adjusted Dynamic Time Wraping* (PADTW) combinado con un algoritmo de segmentación para transformar las variables originales y alimentar así un clasificador SVM. En [99], convierten las variables temporales en imágenes utilizando una función de mapeo que permite aumentar la dimensión de los datos. A continuación, utilizan una NN convolucional para comparar las imágenes entre sí y poder determinar si los vectores originales de características pertenecen a un usuario genuino o a un usuario impostor.

Finalmente, está surgiendo una nueva línea de investigación que trata de combinar información recogida de múltiples fuentes de datos heterogéneas. Esto es, generar modelos de aprendizaje máquina que permiten utilizar información del teclado y del ratón simultáneamente. Esto se traduce en que los clasificadores generados aumentan su eficacia con respecto a solo los que utilizan una única fuente de información. De forma general, existen dos formas de conseguir combinar la información: a nivel de decisión y a nivel de características. Ambas formas de combinar la información presentan multitud de ventajas y mejoran de forma notable la eficacia en los sistemas de autenticación, aunque añaden cierta complejidad a los modelos finales. Las bases de la combinación de información en el ámbito de las dinámicas de comportamiento se fijaron en [100]. En este trabajo, se aborda la combinación de información, tanto a nivel decisión como a nivel características para el reconocimiento facial, el reconocimiento de huella dactilar y reconocimiento de las texturas y formas de la mano, mejorando considerablemente los resultados de los trabajos de la literatura existentes.

La combinación a nivel de decisión se basa en generar un modelo de aprendizaje máquina de forma independiente para cada una de las fuentes de información. De esta manera, cada modelo se entrena para con un tipo de datos específicos. A la hora de dar una predicción en un instante concreto de tiempo, cada modelo de aprendizaje máquina procesa la información de una fuente de información específica, dando una probabilidad de pertenecer a la clase genuina o impostora hasta ese instante. Finalmente, se combinan las probabilidades de todos los modelos, obteniendo una probabilidad final y por lo tanto permitiendo categorizar muestras de múltiples fuentes de información.

En [101] se implementa un Modelo de Confianza (MC) basado en ajustar de forma dinámica los pesos de los modelos independientes generados para el teclado y el ratón, utilizando algoritmos genéticos. Los modelos propuestos en este trabajo son NNs y Counter-Propagation Artificial NNs y utilizan una SVM para combinar ambos modelos. Además, proponen otro método en el que prueban diferentes métricas de distancias como paso previo al entrenamiento de los modelos, con el objetivo de no utilizar datos de impostores en esta fase. En [102], utilizan una combinación basada en utilizar NB para representar cada fuente de información a el mismo espacio de decisión. Posteriormente, utilizan una SVM para clasificar las muestras. En [103], evalúan los modelos de *Random Forest* (RF), SVM, *Decision Tree*s (DTs) y BN con el mismo objetivo, es decir, combinar la información a nivel de decisión. En [104] se propone combinar información de contexto de la sesión con información de comportamiento recogida del teclado y del ratón para mejorar la eficacia de los sistemas de autenticación continua. Para evaluar su propuesta, utilizan el conjunto de datos *The Wolf of SUTD* (TWOS) [105]. En primer lugar, generan un modelo que utiliza única y exclusivamente la información de contexto. Por otro lado, implementan otro modelo que combina tanto la información de teclado como la de ratón, durante una sesión. La combinación de ambos modelos se evalúa por medio de tres modelos: *Parametric Linear Combination* (PLC), un clasificador de RF y un clasificador de SVM. De esta manera, para cada sesión se evalúa la información proveniente de las tres fuentes de información obteniendo una única predicción.

La combinación a nivel de características se basa en generar un único modelo de aprendizaje máquina que permita evaluar de forma simultánea las características provenientes de múltiples fuentes de información. De esta manera, este modelo puede clasificar una muestra que contenga información de una o múltiples fuentes de información sin necesidad de tener que entrenar un modelo para cada una de las mismas. Algunos ejemplos de trabajos en este ámbito se analizan a continuación.

En [106] se utiliza un *Multi-kernel Learning Method* (MKL) para combinar las características de las fuentes de información de teclado y de ratón. Posteriormente, se evalúan los modelos de DT, RF, NB, OC-SVM y SVM entrenados con

el kernel obtenido, obteniendo resultados prometedores. En [107] se compara tanto la combinación a nivel de decisión como la combinación a nivel de características. En primer lugar, para combinar a nivel de decisión se genera un modelo de NB para las dinámicas de teclado y una SVM para las dinámicas de ratón. Posteriormente, ensamblan ambos modelos utilizando un J48 DT para combinar las predicciones obtenidas para cada modelo de forma independiente. A nivel de características, se utiliza *Principal Component Analysis* (PCA) para realizar la combinación y posteriormente evaluar los modelos BN, J48 y SVM de forma independiente. Cabe destacar, que también utilizan una tercera fuente de información proveniente de las interacciones que realizan los usuarios con la interfaz gráfica. Por último, en [108] proponen utilizar estas técnicas de combinación de la información a nivel de características utilizando fuentes de información provenientes de los sensores de los teléfonos inteligentes. Nuevamente, se confirma que la combinación de información supone una gran ventaja a la hora de aumentar la precisión y eficacia para detectar comportamientos sospechosos, frente a las propuestas que no lo utilizan.

La Tabla 2.4 muestra los trabajos relacionados con el método propuesto. Las propuestas que únicamente utilizan una fuente de información (es decir, única y exclusivamente teclado o ratón), han sido reducidas y seleccionadas de acorde a la relación con la presente propuesta, debido al gran número de trabajos en este ámbito.

2.3. Limitaciones de los trabajos previos

Después de analizar en profundidad el estado del arte, se han encontrado las siguientes limitaciones:

- Las principales líneas de investigación y soluciones para mejorar los niveles de seguridad están orientadas al enriquecimiento de las peticiones y/o de los *tokens*, a realizar una mejora en la gestión de las sesiones de usuario, a la mejora de la propia implementación por medio de las APIs y SDKs ofrecidas, a la mejora de los flujos de información, al uso de criptografía a diferentes niveles o a la creación de políticas o sistemas de reputación. No se conocen metodologías o flujos de trabajo para integrar el análisis de comportamiento en los sistemas de gestión de identidades federados con el objetivo de mejorar la seguridad de los mismos.

- Los trabajos previos que proponen utilizar técnicas de análisis de comportamientos para solventar los procesos de IAAA se suelen centrar principalmente en la autenticación adaptativa, ya sea centralizada o distribuida. En casi ningún caso, estas soluciones se integran dentro de un protocolo, *framework* o estándar de gestión de identidades, y mucho menos en los estándares federados en los que se tiene en cuenta a un IdP externo.

Tabla 2.4. Comparación de trabajos previos relacionados con el análisis de comportamiento. *T* y *R* representan las dinámicas de teclado y de ratón respectivamente. *C* se refiere a información de contexto. La columna datos representa el conjunto de datos utilizado el trabajo específico. La columna interacción libre representa si el conjunto de datos contiene dinámicas de comportamiento recogidas en un entorno en el que el usuario es libre de interactuar con el sistema sin realizar una tarea predeterminada

Trabajo	Dinámica de Comportamiento	Método	Nivel de combinación	Datos	Interacción Libre
[85]	T	14 clasificadores	-	.tie5Roanl	No
[86]	T	DT + SVM + AC	-	Propio	Si
[87]	T	Gauss + Parzen + OC-SVM + k-NN + K-means	-	Propio	Si
[88]	T	NB + TANB + KNN	-	Propio	No
[89]	T	NNs	-	Propio	No
[90]	T	NNs	-	Buffalo	Si
[92]	T	KDE	-	Buffalo + Clarkson + Torino	Si
[93]	T	ITAD	-	Buffalo	Si
[95]	R	NNs	-	Propio	Si

[96]	R	NNs	-	Balabit + TWOS	Si
[97]	R	OC-SVM	-	Propio	No
[98]	R	PADTW	-	Propio+ [97]	No
[99]	R	NNs	-	Balabit	Si
[101]	T + R	MC +NNs +SVM	Decisión	Propio	Si
[102]	T + R	NB + SVM	Decisión	Propio	Si
[109]	T + R	BN + BFS	Decisión	Propio	Si
[103]	T + R	RF+ SVM+ DT+ BN	Decisión	Propio	Si
[104]	T + R+ C	RF+SVM+PLC	Decisión	TWOS	Si
[106]	T + R	MKL + DT + RF + NB+ OC-SVM + SVM	Características	Propio	Si
[107]	T + R	BN + J48 + SVM	Decisión y Características	Propio	Si

- Los trabajos previos para detectar anomalías de comportamientos se suelen centrar en un dominio específico de aplicación. Esto hace que los modelos puedan estar sesgados y no sean capaces de generalizar correctamente.

- El número de trabajos de análisis de comportamiento que combinan información es muy reducido, siendo más notable en el caso de la combinación a nivel de características. Sin embargo, estos trabajos son los que suelen obtener mejores resultados en cuanto a eficacia a la hora de detectar comportamientos anómalos.

- Existen una falta de conjuntos de datos públicamente disponibles que contengan dinámicas de comportamiento. Esto se ve reflejado en que la mayoría de propuestas utilizan conjuntos de datos propios recogidos normalmente en un entorno de laboratorio.

De aquí en adelante se trata de superar todas estas limitaciones encontradas. Para ello se propone un flujo de trabajo para integrar las técnicas de análisis de comportamiento dentro de las especificaciones de gestión de identidades federadas. Posteriormente se propone un modelo novedoso de análisis de comportamientos que permite combinar información a nivel de características. Finalmente se evalúan dichas propuestas y se detalla la creación de un conjunto de datos específico que contiene dinámicas de comportamiento.