# Neuralizador: Patrón de Seguridad para el Derecho al Olvido en Ecosistemas Big Data

Julio Moreno[1], Eduardo B. Fernandez[2], Manuel A. Serrano[3],
Eduardo Fernández-Medina[1]

[1] Grupo de investigación GSyA, Universidad de Castilla la Mancha, Ciudad Real, España
[2] Dept. of Computer and Elec. Eng. and Computer Science, Florida Atlantic University, USA
[3] Grupo de investigación Alarcos, Universidad de Castilla la Mancha, Ciudad Real, España
{julio.moreno | manuel.serrano | eduardo.fdezmedina}@uclm.es
ed@cse.fau.edu

**Resumen.** Los ecosistemas Big Data son cada vez más usados por compañías de cualquier ámbito. Big Data permite la obtención de información valiosa a partir del análisis de grandes cantitades de datos. Normalmente, este tipo de entornos suele tener una alta complejidad lo que hace que sean difíciles de gestionar. Además, en los últimos años han surgido diferentes legislaciones que tratan de controlar el uso y análisis de los datos, lo cual, puede afectar de forma directa a este tipo de ecosistemas. Una de la normativas que más debate está generando es el derecho al olvido, gracias a la cual, se intenta que los usuarios tengan un mayor control sobre dónde se encuentran sus datos y cómo se utilizan. Por ello, sin una correcta adaptación de los entornos Big Data a las nuevas normativas, las empresas pueden no solo recibir graves sanciones económicas sino que les puede ocasionar una pérdida de reputación entre sus clientes. En este artículo proponemos un patrón de seguridad específico para ayudar a los administradores de Big Data a implementr el derecho al olvido en sus ecosistemas Big Data definiendo diferentes escenarios y los elementos que lo conforman.

**Palabras clave:** Big Data, Derecho al olvido, Seguridad de la información, Patrones de Seguridad.

## 1 Introducción

En los últimos años, los sistemas Big Data han experimentado un incremento en popularidad en cualquier ámbito [1]. Para todos ellos, los datos son esenciales para llevar a cabo sus actividades diarias, para ayudar a la alta dirección a alcanzar sus objetivos de negocio y, como resultado, tomar mejores decisiones basadas en la información extraída de dichos datos [2], una de las formas de sacar partido de estos datos es usar un ecosistema Big Data. Un ecosistema Big Data puede definirse como el conjunto de diferentes componentes que permiten almacenar, procesar, visualizar y proporcionar información útil para las aplicaciones destino. Normalmente estos componentes son muy complejos y necesitan trabajar juntos para obtener información valiosa [3]. Sin una correcta gestión de los datos para obtener esta información útil, Big Data carece de sentido. Además, cuando se menciona Big Data generalmente se trata de un ecosistema

heterogéneo donde diferentes tecnologías trabajan juntas: desde bases de datos no relacionales, pasando por recursos de computación en nube o hasta los algoritmos para realizar el análisis de los datos. Esta diversidad hace que su manejo sea muy difícil de tratar.

En estos sistemas Big Data, el contexto puede afectar significativamente la forma en que se implementará el ecosistema. Por ello, una de las cosas que hay que tener en cuenta son las diferentes normativas que pueden limitar los datos que se procesarán y los resultados que se obtendrán. En los últimos tiempos, han surgido diferentes leyes que se centran en la privacidad en general, una de las más importantes en el Reglamento General de Protección de Datos (GDPR, por sus siglas en inglés) impulsado por la Unión Europea. Uno de los artículos más polémicos que incluye esta normativa es el referente al Derecho al Olvido (RTBF, por sus siglas en inglés) [4]. La RTBF afirma que cualquier usuario puede solicitar que sus datos sean borrados de todos sus sistemas. Por supuesto, hay algunos casos en los que los datos deben permanecer en el sistema; por ejemplo, si la finalidad de los datos recogidos todavía no se ha alcanzado y el usuario ha dado permiso para el uso de esos datos. Además, existen algunos escenarios en los que los datos deben permanecer en el sistema por motivos legales, por ejemplo, durante la investigación policial de algún delito.

El principal problema es que los legisladores tienden a pensar en un ordenador como si fuera una memoria humana, con lo que, borrar un dato sería similar a olvidar un recuerdo. Sin embargo, en un escenario altamente distribuido donde los datos se almacenan y se replican en diferentes nodos, incluso en diferentes clusters y con distintos sistemas de almacenamiento, el borrado adecuado (u “olvido”) de todos los datos personales de un usuario específico se complica. Existen algunas técnicas generales que intentan resolver este problema como la anonimización (proceso mediante el cual es imposible relacionar un dato con su usuario) o la seudonimización (proceso mediante el cual la información sensible se sustituye por un seudónimo, al contrario que la anonimización este proceso es reversible). Sin embargo, ninguna de estas técnicas ha demostrado ser lo suficientemente eficiente como para no afectar a los resultados que se obtienen del procedimiento de análisis.

En este trabajo, definimos la creación de un patrón específico para abordar este problema. El patrón se llama "Neuralizador", en referencia a la película "Men in black" en la que usaban este dispositivo para hacer que la gente olvidase los eventos "inapropiados". Un patrón es un mecanismo genérico que permite solucionar problemas recurrentes haciendo uso de la abstracción de sus causas principales. Así, un patrón de seguridad permite a los desarrolladores aplicar medidas de seguridad en sus sistemas, sin necesidad de que sean expertos en la materia [5]. Desde su creación, los patrones de diseño se han convertido en un mecanismo muy popular en el área de ingeniería de software a la hora de proteger sistemas contra amenazas y situaciones de mal uso. Cuando se define un patrón es común utilizar la siguiente estructura: intención (el problema que se desea resolver con dicho patrón), contexto (la situación general, en la cual, se aplica el patrón), problema (pequeño enunciado en el que se expresa el problema a resolver), fuerzas (las dificultades o preocupaciones a considerarse a la hora de definir el patrón),

solución (la forma recomendada de resolver el problema en un contexto dado), implementación (pistas o ayudas para implementar el patrón), usos conocidos (mínimo tres usos en sistemas reales), consecuencias (ventajas y desventajas de la solución) y patrones relacionados (patrones que resuelven un problema similar o son complementarios al patrón definido) [6].

El público objetivo de este patrón de seguridad es el responsable de datos (Chief Data Officer, CDO) de la empresa que se encarga de crear la estrategia de gestión de los datos y de la información del sistema, incluyendo su gobierno, control y desarrollo de diferentes políticas y controles para el correcto aprovechamiento de los datos. Por lo tanto, las decisiones de implementación deben tomarse teniendo en cuenta los requisitos y el contexto del ecosistema de Big Data.

El resto del trabajo se estructura de la siguiente forma: primero, una sección de trabajo relacionado, en la cual, se explican los conceptos de patrón de seguridad y las principales características e implicaciones de la aplicación de la normativa RTBF. A continuación, se define nuestro patrón para la aplicación del RTBF en ecosistemas Big Data incluyendo su objetivo, su contexto, la problemática que resuelve, las restricciones que tiene y ayudas para su implementación. Finalmente, se incluye una sección con las conclusiones y trabajo futuro.

## 2 Conocimiento previo

En esta sección se explica la problemática derivada de la aplicación de la normativa GDPR y más específicamente de las implicaciones del derecho al olvido. Por otro lado, se especifica cómo se suelen almacenar los datos en un ecosistema Big Data.

En mayo del año 2018 entró en vigor la Regulación General de Protección de Datos 2016/679 (más conocida por sus siglas en inglés como GDPR). El objetivo de esta normativa es mejorar la protección de datos de carácter personal de los ciudadanos de los estados miembros de la Unión Europea. Esta norma surge a consecuencia del creciente análisis de datos realizado por parte de organizaciones debido al auge de la informática y más específicamente de Big Data. Esta normativa se encuentra compuesta por una serie de artículos, de los cuales, probablemente el más debatido sea el referido al artículo 17 sobre el Derecho al Olvido (RTBF, por sus siglas en inglés) [7]. Esta normativa tiene su origen en el caso de Google Spain, en el cual, la justicia dio la razón a un particular que pidió que se eliminase información suya vinculada con embargos, puesto que vulneraba su honor [8]. Por ello, el RTBF contenido en la GDPR se puede considerar una evolución de dicha sentencia. El RTFB contempla la posibilidad de realizar un borrado retroactivo de los datos personales de una persona a petición de esta. Este borrado se realizará en todos los lugares disponibles en los que puedan haber sido difundidos. Sin embargo, esta normativa genera una gran controversia debido a que choca con otros derechos fundamentales como la libertad de expresión y de información. A esto se le suma la dificultad técnica para ser llevada a cabo, la cual, es incluso mayor en entornos distribuidos y descentralizados como un entorno Big Data.

Para el almacenamiento de los datos, dentro de un ecosistema Big Data existe un componente que suele tener una gran importancia: el Data Lake. Un Data Lake es un repositorio de almacenamiento que contiene una gran cantidad de datos en brutos tal y como fueron generados mientras no sea necesario procesarlos. En general, estos Data Lake almacenan datos no estructurados, pero pueden combinar diferentes tipos de datos en función de los requisitos del sistema. En algunas ocasiones, este "lago de datos" puede convertirse en un caos, en el cual, se dispone de datos repetidos y con poco valor mezclados con aquellos que sí aportan información de utilidad a la compañía. Por tanto, una buena gestión de sus metadatos es crucial para un buen funcionamiento del entorno Big Data [9].

# 3 Patrón Neuralizador

El patrón Neuralizador pretende ayudar en la implementación de la normativa del derecho al olvido dentro de ecosistemas Big Data. En esta sección, se van a definir los diferentes elementos que lo forman, incluyendo la solución, para ello se seguirá la estructura típica de definición de patrones.

## 3.1 Intención

El objetivo principal del patrón es describir cómo gestionar la eliminación de datos personales en entornos Big Data para cumplir con la normativa de derecho al olvido. Normalmente, este derecho es solicitado por un usuario, que es la persona que quiere que el sistema "olvide" sus datos. Por otro lado, también hay algunos escenarios en los que el sistema debería realizar esta acción automáticamente. Por ejemplo, en algunos contextos puede existir la política de borrar los datos de un usuario que no ha accedido a su cuenta en un periodo largo de tiempo.

## 3.2 Contexto

El contexto objetivo en el que puede ser aplicado este patrón es el de entornos distribuidos con grandes cantidades de datos personales y sensibles, almacenados en diferentes bases de datos y sistemas de almacenamiento, los cuales, se encuentran gestionados por diferentes herramientas software.

## 3.3 Problema

El derecho al olvido, también conocido como el derecho al borrado, es una característica que sugiere que es necesario encontrar una manera de eliminar efectivamente los datos de los sistemas de almacenamiento. Borrar los datos de un individuo implica encontrar y borrar todos sus datos; posiblemente dispersos en varios lugares. Esta funcionalidad se complica aún más en contextos de Big Data donde no sólo se almacenan datos personales, sino también información inferida de ellos debido al uso de técnicas

de análisis como la inteligencia de negocio o el aprendizaje automático. Además, esta información suele ser utilizada por muchas herramientas software, con lo cual, seguir la traza de dónde se encuentra cualquier dato personal, se complica.

### 3.4 Fuerzas

Esta solución está restringida por las siguientes fuerzas:

- *La brecha entre los reguladores y la tecnología.* Existe una enorme brecha entre las buenas intenciones de los reguladores y la complejidad de los entornos reales de Big Data. Es necesario que nuestro patrón satisfaga ambas necesidades.
- *Flexibilidad.* Los ecosistemas de Big Data pueden ser utilizados en diferentes escenarios, desde un hospital hasta una fábrica. Diferentes tipos de contextos requieren diferentes soluciones.
- *Valor.* Probablemente, una de las principales características de un sistema Big Data es el valor que se puede obtener de estos datos. Debido a la aplicación del RTBF, esta característica puede verse comprometida; el borrado de registros puede afectar el valor obtenido de los análisis.
- *Control de acceso.* Debido a la importancia de la operación, cualquier borrado o anonimato debe ser realizado por orden del responsable de datos (CDO), quien gestionará todas las solicitudes realizadas por los sujetos de los datos, ya que, es el único autorizado para realizar los cambios en los datos.

### 3.5 Solución

Nuestra solución se basa en la definición de un patrón arquitectural. Por lo tanto, define una arquitectura donde el sujeto de los datos exige la eliminación de sus datos de todas las bases de datos del ecosistema Big Data. Es importante destacar que la eliminación de los datos debe ser autorizada por el CDO. Entonces, una entidad llamada Neuralyzer borrará todos los datos relacionados con el sujeto de los datos. Para ello, utilizará diferentes técnicas, por ejemplo, el enmascaramiento de los datos. Estas decisiones deben tomarse teniendo en cuenta el contexto del sistema y cómo puede afectar al rendimiento de la analítica. Diferentes escenarios y organizaciones necesitan diferentes soluciones, por lo que la decisión sobre qué datos pueden ser borrados del sistema es un proceso complejo en el que el CDO, junto con la alta dirección de la empresa, debe considerar todas las posibilidades y, por lo tanto, tomar decisiones que deberá introducir en el sistema en forma de reglas. En las siguientes subsecciones se explicarán más detalles de la solución.

#### *3.5.1 Estructura*

La Figura 1 muestra el modelo de clase para este patrón. Las clases “Sujeto” y “CDO” representan dos interfaces donde los usuarios que representan son autenticados

por el "Servidor" utilizando un "Servicio de Autenticación". Una vez autenticados, tienen objetivos diferentes. Por un lado, el sujeto solicita la eliminación de todos sus datos personales del sistema. Por otro lado, el responsable de los datos se encarga de introducir reglas de negocio para llevar a cabo el proceso de cumplimiento del RTBF. Estas reglas son introducidas al inicializar el servicio, pero puede ser necesario introducir nuevas reglas para adaptarse a nuevas situaciones. El Olvidador es la clase principal en este patrón, su objetivo principal es decidir qué técnica debe ser usada dependiendo de las reglas introducidas, el contexto del sistema y de los datos almacenados por el usuario.

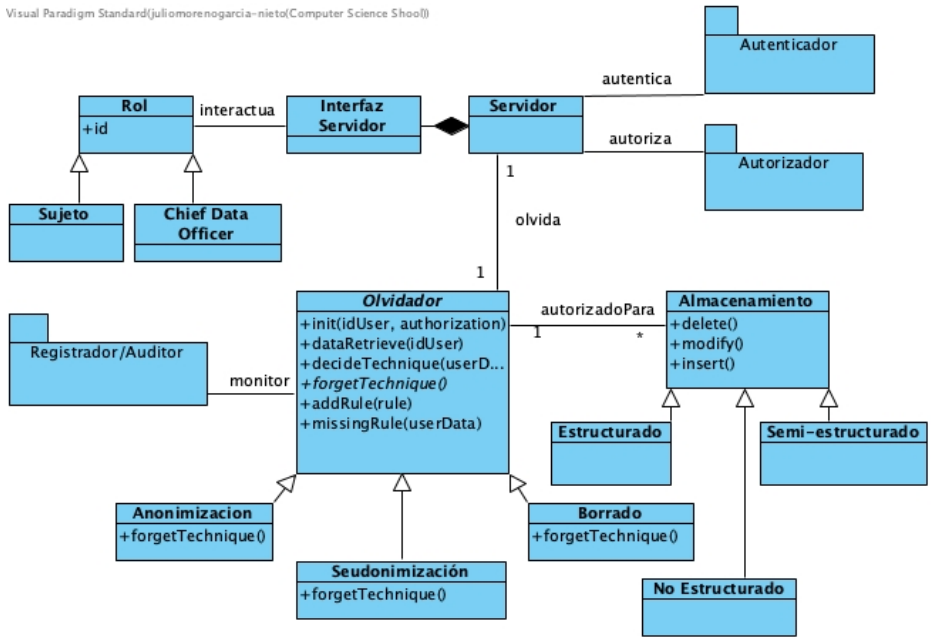


**Figura 1.** Diagrama de clases para el patrón Neuralizador.

Hay tres categorías principales de técnicas representadas por tres subclases diferentes: anonimización, seudonimización y borrado de los datos. Una vez que el "Almacenamiento" se haya modificado adecuadamente para cumplir con la normativa, el sistema debe notificar al sujeto que se ha completado el borrado. Todas las operaciones realizadas sobre los datos deben registrarse en un archivo de log utilizando el patrón registrador/auditor de seguridad [5]. En la Tabla I se resumen los principales componentes de la pauta y se da una breve explicación de su propósito.

**Tabla I**. Componentes del patron Neuralizador

| Nombre | Tipo | Descripción |
|---|---|---|
| Rol | Meta-sujeto | Generalización de los roles que pueden realizar acciones en el patrón. |
| Sujeto | Rol | Sujeto que solicita que sus datos sean borrados del sistema. Ellos inician el uso del patrón. |

| | | |
|---|---|---|
| Chief Data Officer (CDO) | Rol | CDO es responsable de los datos de una empresa al más alto nivel, tanto desde el punto de vista tecnológico como empresarial, incluida la seguridad. Este rol añade las reglas de olvido al sistema basadas en el contexto de la empresa. |
| Interfaz del servidor | Software | Interfaz que muestra todas las acciones proporcionadas por el servidor. |
| Servidor | Software | El servidor proporciona las acciones relacionadas con el RTBF en función de la función a la que se accede. Actúa como un puente entre los roles y el patrón Neuralyzer. |
| Autenticador | Patrón de seguridad | Patrón de seguridad auxiliar utilizado para proporcionar autenticación a los usuarios. |
| Autorizador | Patrón de seguridad | Patrón de seguridad auxiliar utilizado para dar permisos específicos a cada rol. |
| Olvidador | Software | La clase principal del patrón neuralizador. Basado en las reglas de olvido implementadas por el CDO y los datos que deben ser olvidados, aplica las técnicas de olvido en el sistema de almacenamiento. Algunos casos requieren el uso de diferentes técnicas al mismo tiempo. |
| Anonimización | Software | Técnica usada por el Neuralizador. La anonimización de datos oculta la identidad y los datos confidenciales de los propietarios de los registros de datos [10]. |
| Seudonimización | Software | Técnica usada por el Neuralizador. La seudonimización de datos sustituye la identidad del usuario de tal manera que se necesitan datos adicionales para volver a identificar al usuario de los datos. |
| Borrado | Software | Técnica usada por el Neuralizador. Eliminación de los datos del sujeto de todas las diferentes fuentes de datos a lo largo del ecosistema de Big Data. |
| Almacenamiento | Software | Colección de datos que permite acceder a ellos, administrarlos y actualizarlos. En los ecosistemas Big Data, existen normalmente tres formas de almacenamiento de datos: datos estructurados, semiestructurados y no estructurados. |
| Estructurados | Software | Las bases de datos relacionales tradicionales, por ejemplo, MySQL o PostgreSQL; normalmente, en este tipo de almacenamiento se utilizan un lenguaje similar a SQL para realizar consultas a los datos. |
| No estructurados | Software | Son ampliamente utilizados en los ecosistemas de Big Data. En este tipo de almacenamiento, hay cuatro subtipos diferentes: basados en grafos (normalmente utilizados para representar datos de redes sociales; por ejemplo, neo4j), columnar (en estos almacenes cada clave está asociada a uno o más atributos, a diferencia de las bases de datos relacionales; por ejemplo, HBase o Cassandra), documental (los datos se almacenan con un formulario de documento, su principal ventaja es la escalabilidad; por ejemplo, MongoDB o CouchDB), y key-value (utilizan un estilo de tabla de hash similar en el que cada clave está asociada a un conjunto de valores; por ejemplo, Apache Accumulo o Riak). |

| | | |
|---|---|---|
| Semi-estructu-rados | Software | Una forma de almacenar datos que no son ni datos en bruto, ni un sistema de base de datos relacional muy estricto; por ejemplo, el formato JSON puede considerarse como un formato de datos semi-estructurado. |
| Patrón registra-dor/Auditor | Patrón de seguridad | Un mecanismo para realizar el Neuralizador. La mayoría de las veces, la aplicación del RTBF implica la consulta de datos sensibles, por lo que es importante realizar un seguimiento de todas las operaciones realizadas en este proceso. |

### *3.5.2 Dinámicas*

Para mejorar la comprensión del patrón de seguridad en esta subsección se explican dos escenarios de ejecución del mismo. La Figura 2 muestra el diagrama de secuencia del caso de uso “Olvidar los datos de un usuario”, cuyos pasos son:

1. El sujeto solicita que se olviden sus datos del sistema del sistema.
2. Se solicita autenticación al sujeto.
3. El sujeto se autentica en el sistema.
4. El servidor recibe una prueba de autenticación.
5. El servidor inicia el componente “olvidador”.
6. El olvidador consulta los sistemas de almacenamiento para recolectar todos los datos del usuario.
7. El olvidador recibe los datos relacionados con el solicitante.
8. El olvidador decide qué técnica utilizará en función de los datos recuperados y de las reglas creadas por el CDO.
9. En este caso de uso, se decide utilizar una técnica de anonimización para olvidar los datos del usuario.
10. La técnica de anonimización se utiliza en los sistemas de almacenamiento.
11. El servidor notifica al usuario.

Este escenario puede considerarse como el "caso ideal" en el que los datos se encuentran fácilmente, y las reglas del olvido están perfectamente diseñadas. Aunque idealmente este patrón funciona automáticamente sin interacción humana, en algunos casos se le pedirá a la CDO que introduzca nuevas reglas que permitan solucionar nuevos conflictos. Estas nuevas reglas se añadirán al sistema para que puedan ser reutilizadas en el futuro. Este escenario se representa en la Figura 3.

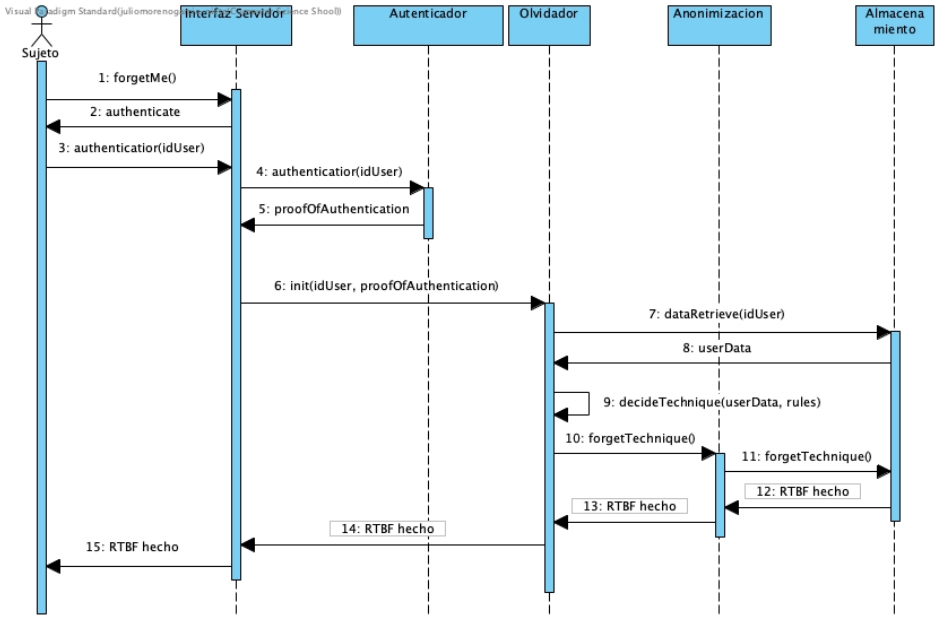


**Figura 2.** Diagrama de secuencia para "Olvidar los datos de un usuario"

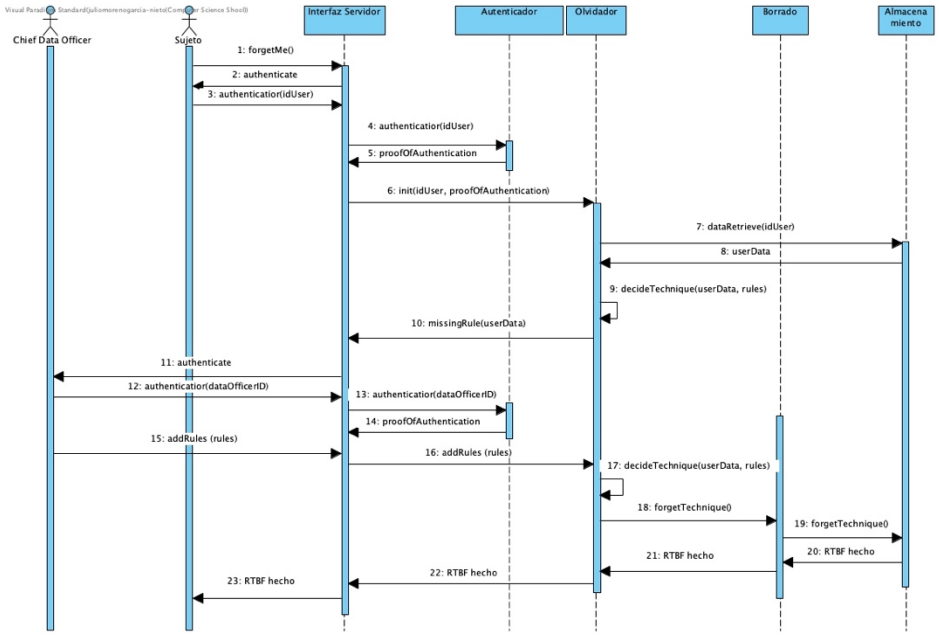


**Figura 3.** Diagrama de secuencia alternativo

### 3.6 Implementación

En la Figura 4 se muestra una posible implementación. En este diagrama de objetos, tenemos un escenario con las dos funciones implicadas: el sujeto (que inicia la solicitud de eliminación de sus datos) y el CDO (que es la persona encargada de gestionar la eliminación de los datos). También hemos añadido el papel del científico de datos, que normalmente está presente en este tipo de sistemas, para demostrar que en este caso particular no tiene ningún tipo de derecho sobre los datos. Además, en este escenario, tenemos un sistema Big Data que tiene un sistema de almacenamiento basado en HDFS (Hadoop Distributed File System).

En este caso, no es necesario eliminar completamente todos los datos; en lugar de hacerlo, basta con aplicar técnicas de anonimización (por ejemplo, k-anonimización) o seudonimización (por ejemplo, enmascaramiento de datos) sobre los datos. Este ejemplo trata de resaltar que el patrón depende del contexto y de las regulaciones que pueden afectar al entorno de Big Data. También se ve afectado por las características de implementación del sistema, por ejemplo, el uso de HDFS.

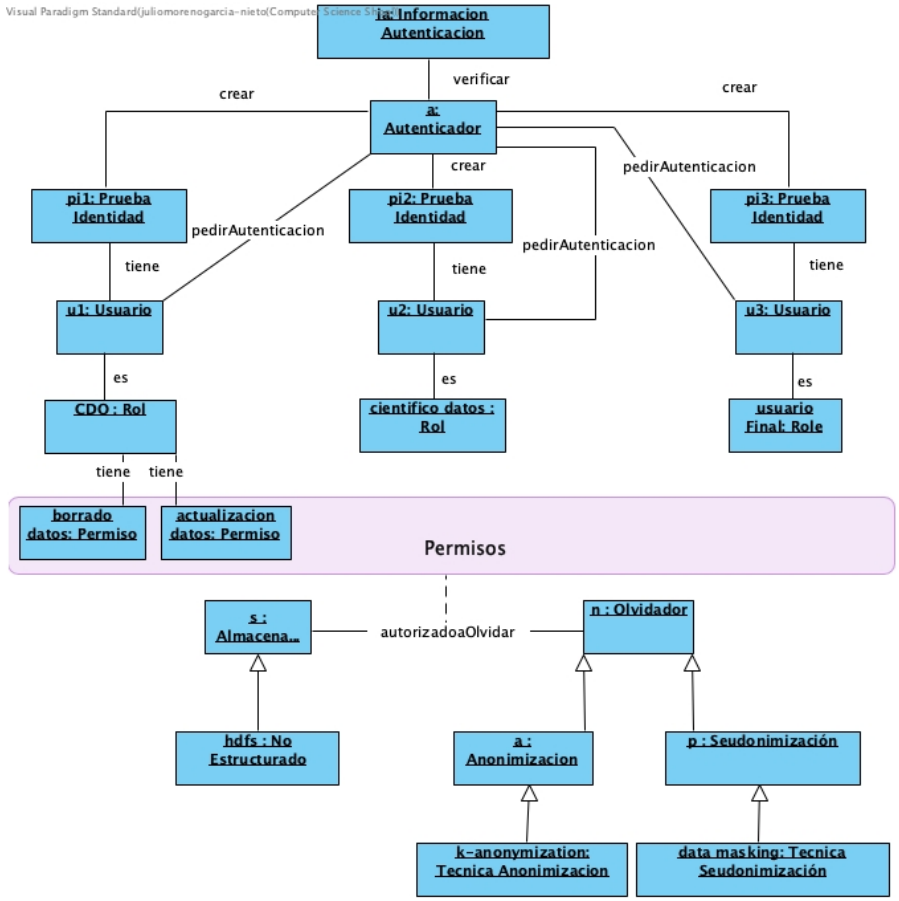


**Figura 4**. Ejemplo de uso del patrón Neuralizador

### 3.7 Usos conocidos

En esta subsección se describen algunos casos de uso que implementan una solución tecnológica similar a nuestra propuesta de patrón de seguridad:

- En [4] los autores explican los principales enfoques para abordar el derecho al olvido en entornos de Inteligencia Artificial. Estas soluciones se pueden clasificar en las tres categorías que consideramos para nuestro patrón. También concluyen que todas estas soluciones tienen un impacto relevante en la calidad del análisis. Además, en [11] el autor también destaca que la mejor manera de cumplir con el derecho al olvido es la anonimización de los datos. Para ello, muestra algunos ejemplos relacionados con el reglamento GDPR.
- *BankingHub* [12]. Explica un escenario específico en el que el derecho al olvido se aplica en el contexto bancario. En este caso, se etiquetan los diferentes tipos de datos almacenados y, dependiendo de ello, asignan un período de retención de los datos; por ejemplo, la probabilidad de migración de su cuenta y las aficiones del sujeto deben ser eliminados inmediatamente, mientras que las transacciones de la cuenta deben permanecer en el sistema 10 años después de que la cuenta quede inactiva. Este es un ejemplo de cómo los datos no pueden ser borrados sólo por demanda, sino también, debido a los requerimientos del contexto en el que el sistema Big Data realiza sus operaciones.

### 3.8 Consecuencias

En esta sección se describen las diferentes ventajas e inconvenientes derivadas de la implementación de nuestro patrón. Así, en nuestro patrón se pueden identificar las siguientes ventajas:

- *La brecha entre los reguladores y la tecnología*. El uso apropiado de diferentes técnicas puede reducir esta brecha. Sin embargo, sigue siendo importante mejorar la formación sobre estos temas para los reguladores y el personal de TI.
- *Flexibilidad*. El patrón incluye tres maneras de enfrentar el problema del derecho a ser olvidado. Por lo tanto, puede adaptarse a diferentes contextos.
- *Control de acceso*. Al realizar una instanciación del patrón RBAC para controlar las acciones en las bases de datos y restringir el acceso sólo a la función CDO se consigue una mejor seguridad en el sistema.

Por otro lado, de la implementación de nuestro patrón también se pueden derivar una desventaja que puede limitar su implantación:

- *Valor*. Cada vez que se borra cualquier registro de los datos que se están analizando, es muy probable que los resultados se vean afectados y varíen con respecto a la ejecución anterior. Si la solicitud de aplicar el derecho al

olvido la realizan muchos usuarios, el valor aportado por el ecosistema Big Data puede verse reducido drásticamente.

### 3.9 Patrones relacionados

Finalmente, en esta subsección se definen diferentes patrones de seguridad que cubren aspectos similares al nuestro o que se encuentran incorporados de alguna forma en nuestra propuesta:

- *Autenticador [5].* Cuando una entidad activa como un usuario o un sistema (sujeto) se identifica con el sistema, ¿cómo se verifica que el sujeto que intenta acceder al sistema es quien dice ser? Presentando información que es reconocida por el sistema e identifica al sujeto. Después de ser reconocido, el sujeto recibe algún tipo de prueba de que ha sido autenticado. Este patrón se utiliza para autenticar a los usuarios en el sistema.
- *Acceso basado en roles [5].* Describe cómo asignar permisos en función de las funciones o tareas de los usuarios en un entorno en el que se requiere el control del acceso a los recursos informáticos. El patrón del RBAC es necesario para establecer los derechos que las diferentes funciones tienen sobre los datos.
- *Registro de seguridad/Auditor [5].* Este patrón busca llevar una traza de las acciones del usuario para determinar quién hizo qué y cuándo lo hizo. Registrar todas las acciones de seguridad, que pueden incurrir en incumplimiento de reglas para datos sensibles, ayuda a realizar un mejor control del acceso a los recursos que pueden ser útiles para procesos de auditoría. La eliminación de datos individuales es una operación muy delicada que debe registrarse en un archivo de log.

## 4 Conclusiones y trabajo futuro

Este patrón tiene como objetivo ayudar en la implementación de la normativa sobre el derecho al olvido en entornos Big Data. Diferentes regulaciones, como el GDPR de la Unión Europea, incluyen esta normativa que puede disuadir a las empresas de utilizar estos sistemas debido a su dificultad para ser implementado. Esta dificultad es consecuencia de la heterogeneidad de tecnologías que suelen utilizarse en ecosistemas de Big Data. La finalidad principal de este tipo de ecosistemas es recolectar y analizar datos, por lo que resulta especialmente difícil cumplir con esta regulación. Este patrón considera los diferentes sistemas de almacenamiento que puede tener un sistema Big Data y también las diferentes técnicas que se pueden realizar para cumplir con esta limitación. Aún así, es importante investigar más a fondo cómo se pueden implementar estas técnicas en diferentes sistemas de almacenamiento antes de que este patrón se utilice correctamente en escenarios del mundo real. Así, como trabajo futuro se realizará un estudio de cómo localizar y etiquetar los datos de cada usuario, además de la trazabilidad de estos al ser modificados por diferentes aplicaciones.

## Agradecimientos

Este trabajo ha sido financiado por el proyecto ECLIPSE (Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional FEDER) y el proyecto GENESIS (Consejería de Educación, Cultura y Deportes de la Dirección General de Universidades, Investigación e Innovación de la JCCM, SBPLY-17-180501-000202).

## Referencias

1. Akoka, J., Comyn-Wattiau, I., Laoufi, N.: Research on Big Data – A systematic mapping study. Comput. Stand. Interfaces. 54, 105–115 (2017). https://doi.org/10.1016/j.csi.2017.01.004.
2. Mayer-Schönberger, V., Cukier, K.: Big Data: A Revolution that Will Transform how We Live, Work, and Think. Houghton Mifflin Harcourt (2013).
3. Demchenko, Y., De Laat, C., Membrey, P.: Defining architecture components of the Big Data Ecosystem. In: Collaboration Technologies and Systems (CTS), 2014 International Conference on. pp. 104–112. IEEE (2014).
4. Villaronga, E.F., Kieseberg, P., Li, T.: Humans forget, machines remember: Artificial intelligence and the Right to Be Forgotten. Comput. Law Secur. Rev. 34, 304–313 (2018). https://doi.org/10.1016/j.clsr.2017.08.007.
5. Fernandez, E.B.: Security patterns in practice: designing secure architectures using software patterns. John Wiley & Sons (2013).
6. Bunke, M.: Software-security Patterns: Degree of Maturity. In: Proceedings of the 20th European Conference on Pattern Languages of Programs. pp. 42:1–42:17. ACM, New York, NY, USA (2015). https://doi.org/10.1145/2855321.2855364.
7. Politou, E., Michota, A., Alepis, E., Pocs, M., Patsakis, C.: Backups and the right to be forgotten in the GDPR: An uneasy relationship. Comput. Law Secur. Rev. 34, 1247–1257 (2018). https://doi.org/10.1016/j.clsr.2018.08.006.
8. Post, R.C.: Data privacy and dignitary privacy: Google spain, the right to be forgotten, and the construction of the public sphere. Duke Law J. 67, 981–1072 (2018).
9. Diamantini, C., Giudice, P.L., Musarella, L., Potena, D., Storti, E., Ursino, D.: A new metadata model to uniformly handle heterogeneous data lake sources. Commun. Comput. Inf. Sci. 909, 165–177 (2018). https://doi.org/10.1007/978-3-030-00063-9_17.
10. Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., Shen, X.S.: Security and Privacy in Smart City Applications: Challenges and Solutions. IEEE Commun. Mag. 55, 122–129 (2017).
11. Mohammed J. Khan, C.: Big Data Deidentification, Reidentification and Anonymization. (2018).
12. GDPR deep dive—how to implement the 'right to be forgotten,' https://www.bankinghub.eu/banking/finance-risk/gdpr-deep-dive-implement-right-forgotten, (2017).