NOVIEMBRE 2022

INTRODUCCIÓN A LA ANONIMIZACIÓN DE DATOS

Técnicas y casos prácticos

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
red.es
Oficina del Dato
Iniciativa aporta
datos.gob.es

Contenido

Contenido elaborado por José Barranquero, experto en Ciencia de Datos y Computación Cuántica.

1. INTRODUCCIÓN

Nos encontramos en un momento histórico, donde **los datos se han convertido en un activo clave** para casi cualquier proceso de nuestra vida cotidiana. Cada vez hay **más formas de recoger datos y más capacidad para procesarlos y compartirlos**, donde juegan un papel crucial nuevas tecnologías como Internet de las Cosas (IoT), Blockchain, Inteligencia Artificial, Big Data o Linked Data.

El uso de estos datos no solo afecta a los resultados de empresas privadas, sino que también se aplica en la mejora de los servicios públicos y la toma de decisiones, así como para la **elaboración de estudios de carácter social, científico o económico.**

Tanto cuando hablamos de datos abiertos, como de datos en general, es crítico poder garantizar la privacidad de los usuarios y la protección de sus datos personales, entendidos como derechos fundamentales. Un aspecto que en ocasiones no recibe especial atención a pesar de las rigurosas normativas existentes, como el Reglamento General de Protección de Datos (RGPD).

En este informe vamos a explicar los **conceptos clave de un proceso de anonimización de datos**, incluyendo definiciones, principios metodológicos y tipos de riesgos existentes. Finalmente se detallarán las técnicas esenciales que se aplican en la actualidad, presentado ejemplos prácticos de las más relevantes. Cabe destacar que existen técnicas más avanzadas no recogidas en este informe, siendo **un campo con mucha actividad a nivel académico e investigador**.

El objetivo del informe es ofrecer una introducción suficiente y concisa, principalmente orientada a **publicadores de datos que necesiten garantizar la privacidad** de estos. Por cuestiones de alcance, no se trata de una guía exhaustiva, sino una primera toma de contacto para entender los riesgos y técnicas disponibles, así como la complejidad inherente a cualquier proceso de anonimización de datos. Dado que los ejemplos planteados como casos prácticos están escritos en código Python, es recomendable tener unos conocimientos mínimos del lenguaje para poder entenderlos adecuadamente.

"Es crítico poder garantizar la privacidad de los usuarios y la protección de sus datos personales."

1.1 DEFINICIONES Y TÉRMINOS RELEVANTES

Antes de empezar, es necesario definir brevemente los conceptos sobre los que vamos a hablar en el informe. Para estas definiciones, se ha tomado como referencia el RGPD:

• **Persona física identificable:** *persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un identificador como, por ejemplo, un nombre, un número de identificación, datos de localización, un identificador en línea o uno o varios elementos propios de la identidad física, fisiológica, genética, psíquica, económica, cultural o social de dicha persona (RGPD, art. 4.1).*

• **Datos personales:** *toda información relacionada con una persona física identificada o identificable, "el interesado" (RGPD, art. 4.1).*

• **Información de identificación personal:** *en inglés Personally Identifiable Information o PII, es cualquier dato personal que podría identificar de forma única a una persona física identificable. Existen tres niveles de identificación de personas: microdatos, datos de identificación indirecta y datos sensibles.*

• **Microdatos:** *son datos únicos para cada individuo, que pueden permitir su identificación directa (DNI, código de historia clínica, número de cuenta, perfil de redes sociales, etc).*

• **Datos de identificación indirecta:** *son aquellos datos que al combinarse con la misma o diferentes fuentes de datos pueden permitir identificar a un individuo (datos sociodemográficos, configuración del navegador, ubicación geográfica, etc). También se conocen como quasi-identificadores.*

• **Datos sensibles:** *son todos aquellos datos personales referidos en el artículo 9 del RGPD (en especial datos financieros y médicos), los cuales suelen ser útiles para la realización de estudios, por lo que es importante que estén presentes. Pero a su vez es crítico que no se pueda identificar al propietario de estos, por tener importantes implicaciones para su privacidad al tratarse de información confidencial del individuo.*

• **Tratamiento:** *cualquier operación o conjunto de operaciones realizadas sobre datos personales o conjuntos de datos personales, ya sea por procedimientos automatizados o no, como la recogida, registro, organización, estructuración, conservación, adaptación o modificación, extracción, consulta, utilización, comunicación por transmisión, difusión o cualquier otra forma de habilitación de acceso, cotejo o interconexión, limitación, supresión o destrucción (RGPD, art. 4.2).*

• **Ofuscación de datos:** *tratamiento para cambiar o alterar datos sensibles o que identifican a una persona, con el objetivo de proteger la información confidencial.*

• **Anonimización de datos:** *define la metodología y el conjunto de buenas prácticas y técnicas que reducen el riesgo de identificación de personas, la irreversibilidad del proceso de anonimización y la auditoría de la explotación de los datos anonimizados, monitorizando quién, cuándo y para qué se usan. Es decir, cubre tanto el objetivo de anonimización, como el de mitigación del riesgo de reidentificación, siendo este último un aspecto clave.*

• **Reidentificación:** *identificar a las personas específicas a las que pertenecen los datos a partir de ellos. Es uno de los riesgos clave a mitigar en un proceso de anonimización de datos.*

• **Cadena de confidencialidad:** *término que incluye el análisis de riesgos específicos para la finalidad del tratamiento a realizar. La rotura de esta cadena implica la posibilidad de reidentificación.*

• **Datos anónimos:** *datos que en ningún momento han contenido información personal de un individuo, por lo que no son datos personales, ni están afectados por el RGPD.*

• **Datos anonimizados:** *son datos que permitían identificar a una persona física o jurídica en su forma original, pero que han pasado por un proceso de anonimización que imposibilita la reidentificación del propietario, por lo que ya no son datos personales, ni están afectados por el RGPD.*

• **Datos seudonimizados:** *tratamiento realizado sobre datos personales de manera que ya no puedan atribuirse a un individuo (interesado), a menos que se emplee información adicional, y siempre que dicha información adicional figure por separado y esté sujeta a medidas técnicas y organizativas destinadas a garantizar que los datos personales no se atribuyan a una persona física identificada o identificable (RGPD, art 4.5). Cabe destacar que esta tipología de datos tiene la consideración de datos personales, por lo que es necesario cumplir con la normativa RGPD, del mismo modo que con los datos personales originales.*

1.2 PRINCIPIOS BÁSICOS DE ANONIMIZACIÓN

La anonimización de datos debe regirse por el concepto de **privacidad desde el diseño y por defecto** (RGPD, art. 25), teniendo en cuenta 7 principios (ver **FIGURA 1**):

Figura 1. *Principios básicos de anonimización de datos*

• **Proactivo:** el diseño debe plantearse desde las etapas iniciales de conceptualización, identificando microdatos, datos de identificación indirecta y datos sensibles, estableciendo escalas de sensibilidad que sean informadas a todos los implicados en el proceso de anonimización.

• **Privacidad por defecto:** es necesario establecer el grado de detalle o granularidad de los datos anonimizados con el objetivo de preservar la confidencialidad, eliminando variables no esenciales para el estudio a realizar, teniendo en cuenta factores de riesgo y beneficio.

• **Objetivo:** dada la imposibilidad de una anonimización absoluta, es crítico evaluar el nivel de riesgo de reidentificación asumido y establecer las políticas adecuadas de contingencia.

• **Funcional:** para garantizar la utilidad del conjunto de datos anonimizado, es necesario definir claramente la finalidad del análisis que se va a realizar sobre los datos una vez anonimizados e informar a los usuarios de los procesos de distorsión empleados para que sean tenidos en cuenta durante su explotación.

• **Integral:** el proceso de anonimización va más allá de la generación del conjunto de datos, siendo aplicable también durante el estudio de estos, a través de contratos de confidencialidad y uso limitado, validados mediante las auditorías pertinentes durante todo el ciclo de vida del proceso de anonimización.

• **Informativo:** este es un principio clave, siendo necesario que todos los participantes en el ciclo de vida del proceso de anonimización sean debidamente capacitados e informados respecto a su responsabilidad y los riegos asociados.

• **Atómico:** es recomendable, en la medida de lo posible, que el equipo de trabajo se defina con personas independientes para cada función dentro del proceso.

2. FASES DE LA ANONIMIZACIÓN

La legislación europea no prescribe ninguna norma concreta (Dictamen 05/2014, apartado 2.2), por lo que la decisión final sobre qué técnica o conjunto de técnicas es más adecuada **depende de cada caso particular**:

> **"conocer las principales fortalezas y debilidades de cada técnica ayuda a comprender cómo diseñar un proceso de anonimización adecuado en un contexto dado".**

La **FIGURA 2** muestra las fases clave del proceso, desarrollada a partir de la Guía de orientaciones y garantías en los procedimientos de anonimización de datos personales de la AEPD:

Figura 2. *Fases del proceso de anonimización de datos*

Como puede observarse, se trata de un proceso extremo a extremo (**principio de integridad**), donde una vez obtenido el conjunto anonimizado, se establece como fundamental **instaurar garantías para proteger los derechos** de los interesados **y realizar auditorías periódicas**, tanto del uso que se hace de los datos anonimizados, como de las propias políticas de anonimización, que deben estar documentadas apropiadamente.

El primer paso consiste en la **definición del equipo de trabajo**, detallando las funciones de cada perfil, y garantizando, en la medida de lo posible, que cada miembro desempeñe sus tareas de forma independiente del resto (**principio de atomicidad**).

Dado el continuo avance de la tecnología, es especialmente **complejo poder garantizar la anonimización absoluta**, por lo que el riesgo de reidentificación se aborda como un riesgo residual, asumido y gestionado, y no como un incumplimiento de la normativa. Es decir, se rige por el **principio de objetividad**, siendo necesario establecer políticas de contingencia. Estas políticas deben plantearse en términos de coste frente a beneficio, haciendo que **el esfuerzo necesario para la reidentificación no sea asumible o sea razonablemente imposible**.

Otro factor importante antes de diseñar un proceso de anonimización, es la **calidad de los datos** resultantes para un fin determinado, también denominado utilidad, dado que en ocasiones es necesario sacrificar parte de la información (**principio de privacidad por defecto**). Esto conlleva un riesgo inherente para el que es necesario identificar y plantear medidas de mitigación para evitar la **pérdida de potencial informativo** del conjunto de datos anonimizado, enfocado a los objetivos concretos de cada caso de uso o estudio (**principio de funcionalidad**).

Una vez definidos los objetivos y riesgos, así como la viabilidad del proceso, una tarea esencial del equipo es definir un esquema basado en los tres niveles de identificación de personas: microdatos, identificadores indirectos y datos sensibles (**principio de proactividad**), donde se asigne un valor cuantitativo a cada una de las variables. Esta escala debe ser conocida por todo el personal implicado (**principio de información**) y es crítico para la Evaluación de Impacto en la Protección de los Datos Personales (EIPD).

En definitiva, el reto reside en conseguir que **el análisis de los datos anonimizados no difiera significativamente** con respecto al mismo análisis realizado sobre el conjunto de datos original, consiguiendo **minimizar el riesgo de reidentificación mediante la combinación de varias técnicas de anonimización y la monitorización de todo el proceso**; desde la anonimización a la explotación con una finalidad concreta.

2.1 TIPOS DE TÉCNICAS DISPONIBLES

De cara a implementar el proceso de anonimización, pueden considerarse varias técnicas, que buscan principalmente **garantizar el avance de la sociedad de la información mediante la compartición de datos entre entidades y la publicación de datos abiertos**, sin menoscabar el derecho de las personas al respeto a la protección de sus datos personales. Actualmente existen **tres enfoques generales**, cada uno de ellos integrado a su vez por varias técnicas (Dictamen 05/2014):

1. **Aleatorización:** tratamiento de datos, eliminando la correlación con el individuo, mediante la adición de ruido, la permutación o la Privacidad Diferencial.

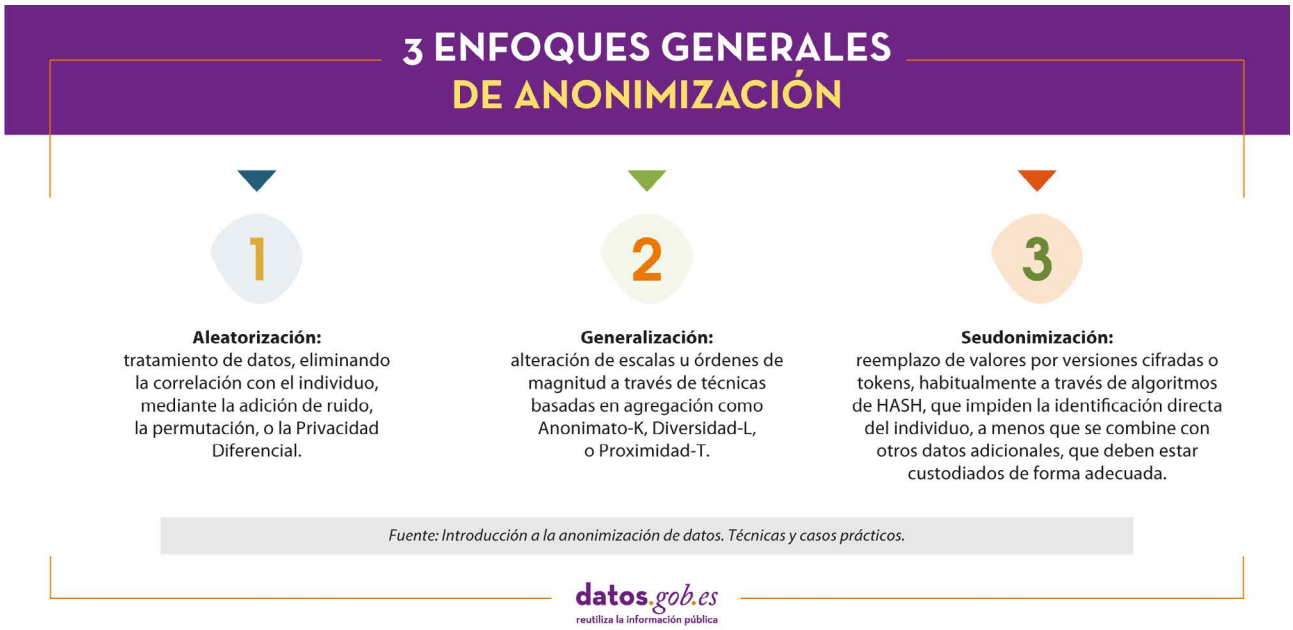
2. **Generalización:** alteración de escalas u órdenes de magnitud a través de técnicas basadas en agregación como Anonimato-K, Diversidad-L o Proximidad-T.

3. **Seudonimización:** reemplazo de valores por versiones cifradas o tokens, habitualmente a través de algoritmos de HASH, que impiden la identificación directa del individuo, a menos que se combine con otros datos adicionales, que deben estar custodiados de forma adecuada.

Se incluye dentro de la categoría de seudonimización la ofuscación de datos mediante cifrado, con o sin borrado de cave, y el procesado directo de datos cifrados a través de cifrado homomórfico. Ambas técnicas pueden ser complementadas con sellos de tiempo o firma digital.

Normalmente la seudonimización y el cifrado son técnicas íntimamente relacionadas y es muy complicado establecer una línea clara entre ambas. Además, **ninguna de las técnicas enmarcadas en la categoría de seudonimización está considerada una técnica válida de anonimización sin ser combinada adecuadamente con técnicas de las otras dos categorías**. Es importante tener en cuenta que un conjunto de datos seudonimizado **sigue considerándose como datos de carácter personal a efectos del RGPD**, porque es factible la reidentificación a través de claves custodiadas, entre otros riegos que veremos más adelante.

En concreto, la utilidad principal de la seudonimización y el cifrado es precisamente **poder reidentificar los datos si se necesita tras el tratamiento**. Por ejemplo, al entrenar un modelo predictivo en un conjunto de datos de pacientes, se podría revertir la seudonimización en aquellos registros que el modelo detecte con riesgo de padecer una patología. Esta reidentificación podría **realizarse internamente por el responsable del tratamiento**, sin necesidad de compartir dicha información con el equipo que construye el modelo predictivo.



2.2 IDENTIFICACIÓN Y TIPOS DE RIESGOS

La etapa de análisis de riesgos dentro del ciclo de vida del proceso de anonimización es especialmente crítica y pocas veces se enfoca de forma adecuada en la práctica, existiendo ejemplos graves de reidentificación de individuos en conjuntos de datos anonimizados, por una aproximación al problema inadecuada.

Es importante señalar que **el riesgo de reidentificación aumenta con el paso del tiempo**, debido a la posible aparición de nuevos datos o el desarrollo de nuevas técnicas, como los futuros avances en computación cuántica, que podrían conllevar la ruptura de claves de cifrado, conocido como Q-day.

Se establecen tres vectores de riesgo concretos asociados a la reidentificación (Dictamen 05/2014):

1. Singularización (singling out): ¿se puede singularizar a una persona? Mide la posibilidad de extraer de un conjunto de datos algunos registros (o todos los registros) que identifican a una persona, es decir, contempla el riesgo de extraer atributos que permitan identificar a uno o varios individuos.

2. Vinculabilidad (linkability): ¿se pueden vincular registros relativos a una persona? Mide la capacidad de vincular como mínimo dos registros de un único interesado o de un grupo de interesados, ya sea en el mismo conjunto de datos o en dos conjuntos de datos distintos. Si el atacante puede determinar (p. ej., mediante un análisis de correlación) que dos registros están asignados al mismo grupo de personas, pero no puede singularizar a las personas en este grupo, entonces la técnica es resistente a la singularización, pero no a la vinculabilidad.

3. Inferencia (inference): ¿se puede inferir información relativa a una persona? Mide la posibilidad de deducir con una probabilidad significativa el valor de un atributo a partir de los valores de un conjunto de otros atributos.

3. TÉCNICAS DE ANONIMIZACIÓN

A continuación, se analizan cada una de las técnicas mencionadas en el apartado anterior en detalle. Si se desea profundizar en los detalles de cada técnica y consultar ejemplos concretos se recomienda revisar los apartados correspondientes del Dictamen 05/2014 y la guía de orientaciones de la AEPD.

Adicionalmente se incluyen **varios casos prácticos en el siguiente capítulo, donde se analizan con ejemplos algunas de estas técnicas en mayor profundidad.**

3.1 ALEATORIZACIÓN

Este tipo de técnicas se basan en **modificar o alterar la veracidad de los datos a nivel individual, respetando la distribución global de éstos**, consiguiendo reducir así la vinculabilidad y la inferencia. La aleatorización empleada de forma aislada no es efectiva frente a la singularización.

Siempre deben combinarse, al menos, con un proceso de filtrado explícito de atributos obvios o identificadores indirectos (principio de privacidad por defecto), o indirectamente mediante técnicas de generalización.

3.1.1 Adición de ruido

El principio básico es perturbar los datos de cada individuo a nivel local, **respetando la distribución global de la muestra para que no se pierda su utilidad.**

Se trata de una de las técnicas más básicas, pero que aplicada de forma adecuada **permite reducir los riesgos, en especial si se combina con otras técnicas**.

RECOMENDACIONES

- Agregar ruido de forma consistente, para que un atacante no sea capaz de filtrar dicho ruido
- Validar que el tamaño del conjunto sea suficientemente grande para evitar la vinculabilidad a partir de otras fuentes de datos
- Definir un nivel de ruido suficiente para obtener el grado de privacidad deseado.

3.1.2 Permutación

Al contrario que con la adición de ruido, en este caso **no se modifican los valores ni afecta a la distribución y rango de los valores**, sino que se intercambian entre diferentes individuos para reducir su vinculabilidad.

Obviamente, al permutar valores, **no se mantienen las correlaciones con los individuos**. Además, si dos o más atributos tienen una relación lógica o una correlación estadística y se permutan independientemente, dicha relación se pierde.

RECOMENDACIONES

- Respetar las correlaciones entre atributos, aplicando los movimientos de datos por grupos, para **evitar que un atacante emplee dichas relaciones para revertir la permutación**.

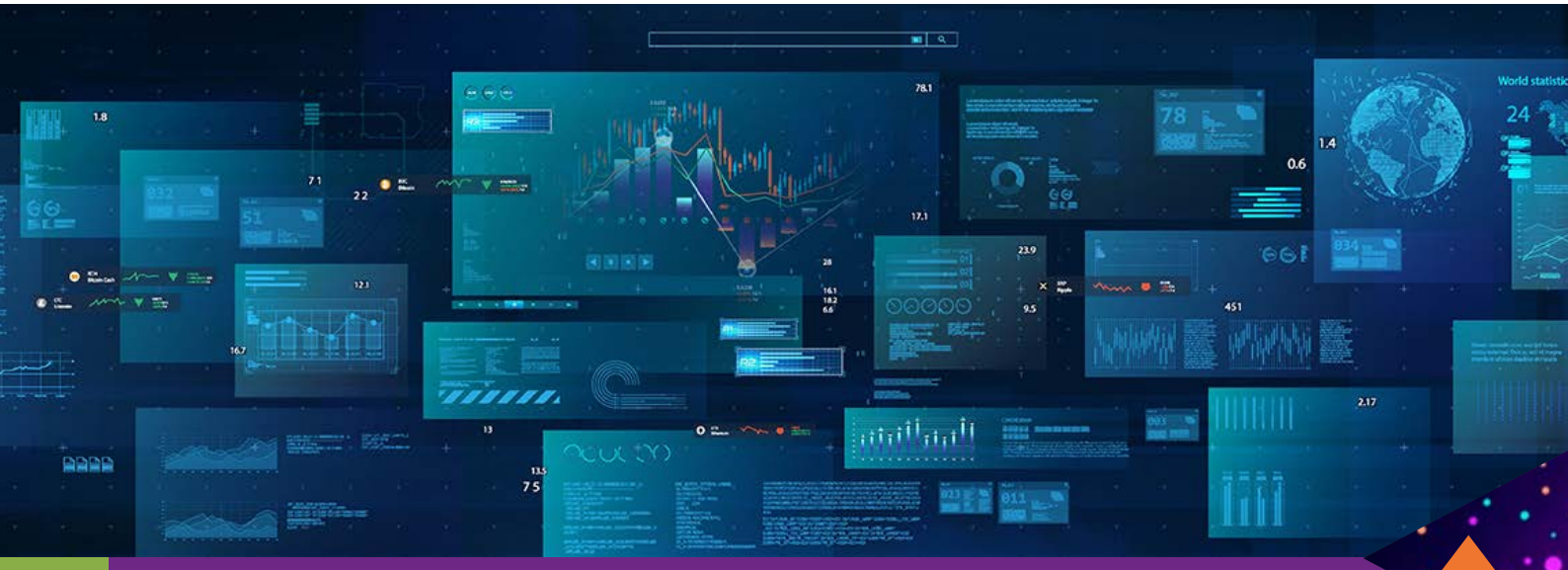
3.1.3 Privacidad Diferencial

Es un caso especial de aleatorización, donde el proceso **se aplica a cada consulta realizada** por un tercero, y es gestionado mediante **vistas anonimizadas** por el responsable del tratamiento.

Es importante señalar que **el conjunto de datos no se publica de forma abierta**, sino que se preserva custodiado por el responsable y **no se modifican los datos originales almacenados**. Por tanto, **los resultados obtenidos deben considerarse datos personales** en términos de privacidad, dado que el responsable del tratamiento mantiene la capacidad de identificación de los individuos en el conjunto de datos original.

RECOMENDACIONES

- **Realizar una trazabilidad exhaustiva de todas las consultas realizadas**, en especial aquellas solicitadas por un usuario o empresa concretos. Este sigue siendo un **campo activo de investigación**, donde el objetivo es conseguir un equilibrio adecuado entre la utilidad de los resultados y las garantías de privacidad ofrecidas.
- Evitar el uso de motores de búsqueda abiertos, dado que es muy complejo realizar la trazabilidad de las consultas, con el fin de **ajustar las respuestas obtenidas en función del histórico de consultas**. Para garantizar la **protección ante ataques de inferencia y vinculación**, es necesario realizar un **estudio personalizado de cada consulta**.
- Generar únicamente resultados estadísticos agregados. Si los resultados ofrecidos están correctamente procesados, esta técnica es **muy útil ante ataques de singularización**.



3.2 GENERALIZACIÓN

Este segundo conjunto de técnicas tiene por objetivo **generalizar algunos atributos críticos** de forma que se evite la singularización, por ejemplo, modificando las escalas u órdenes de magnitud, **reemplazando valores por categorías superiores en una jerarquía**.

Es necesario aplicar el proceso de forma adecuada y **aplicar otras técnicas de forma conjunta para garantizar la protección ante ataques de inferencia o vinculabilidad**.

3.2.1 Anonimato-K

Estas técnicas son especialmente útiles en escenarios donde la relación existente entre algunos conjuntos de atributos permite **generar identificadores mediante vinculabilidad o inferencia a partir de identificadores indirectos**. La principal garantía que ofrecen radica en **no permitir la singularización de individuos en un grupo de al menos K miembros**.

El concepto básico que aplica es la generalización de atributos mediante el **reemplazo de valores por otros superiores dentro de una escala o jerarquía**, como reemplazar días por semanas o meses, o reemplazar ciudades por regiones o provincias. En el caso concreto de valores numéricos continuos, normalmente se aplican **técnicas de agrupación por rangos (discretización)**.

En el capítulo 4 veremos un caso práctico en detalle de cómo aplicar algunas de estas técnicas. A modo ilustrativo en la **FIGURA 3** se muestra cómo al generalizar las variables del conjunto original (izquierda), se obtiene un nuevo conjunto (derecha), donde algunos ejemplos quedan agrupados en categorías, reduciendo considerablemente el riesgo de singularización a partir de las variables transformadas (sin tener en cuenta la tarjeta de crédito, que requerirá tratamiento por cifrado):

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55335	58	Male	8700
1	5418686973265201	55255	36	Female	9700
2	5527060358825468	55559	32	Female	6800
3	5312916958971375	55700	58	Male	4700
4	5541858987662877	55925	52	Male	5700

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55000-55499	senior	Male	high
1	5418686973265201	55000-55499	medium	Female	high
2	5527060358825468	55500-55999	medium	Female	high
3	5312916958971375	55500-55999	senior	Male	high
4	5541858987662877	55500-55999	senior	Male	high

Figura 3. *Ejemplo de aplicación de tratamientos para mejorar el Anonimato-K*

La vinculabilidad se reduce considerablemente a título individual dentro de un grupo de K miembros, **pero el grupo aún tiene una probabilidad 1/K de ser vinculado**. Si bien, el principal riesgo de esta técnica reside en que **no evita la inferencia cuando se conoce el grupo concreto al que pertenece un individuo**.

Un ejemplo interesante de aplicación es Amnesia, un proyecto europeo muy conocido que ofrece una herramienta software de uso muy intuitivo para la **anonimización de conjuntos de datos mediante Anonimato-K**.

RECOMENDACIONES

Uno de los aspectos críticos de esta técnica es la **definición del valor K**:

- El uso de **valores de K insuficientes** asigna mayor peso a cada individuo y hace que los **ataques por inferencia sean más sencillos**.
- Al contrario, el empleo de **valores de K excesivamente grandes** reduce las garantías frente a la singularización, dado que se aumenta el **riesgo de incluir identificadores indirectos**.
- Es necesario encontrar un punto de equilibrio para definir un **valor de K suficientemente grande que evite que algunos individuos tengan más peso dentro del grupo, sin dejar de considerar atributos críticos durante el proceso**.

3.2.2 Diversidad-L y Proximidad-T

Ambas técnicas son versiones evolucionadas del Anonimato-K, donde se busca **mejorar las garantías frente a ataques de inferencia directa**, aunque siguen siendo vulnerables a ataques por inferencia probabilística, es decir, reducen significativamente la confianza de las inferencias. Al ser técnicas complementarias al Anonimato-K ofrecen **el mismo nivel de protección frente a ataques de singularización y vinculabilidad**.

En el caso de la Diversidad-L, el proceso garantiza que existen **al menos L valores diferentes para cada atributo, dentro de un mismo clúster** de al menos K individuos.

La Proximidad-T garantiza de forma adicional que **la distribución de los atributos dentro de cada grupo refleja la misma distribución del conjunto original,** dificultando aún más los ataques por inferencia.

RECOMENDACIONES

- Validar que los valores sigan una **distribución uniforme dentro de cada grupo.**

3.3 SEUDONIMIZACIÓN

La seudonimización **no se considera un método de anonimización** (Dictamen 05/2014), ni tampoco los conjuntos de datos resultantes pueden considerar conjuntos de datos anónimos. Si bien, resultan medidas útiles para mejorar la seguridad, **reduciendo la vinculabilidad del conjunto de datos obtenido**.

El problema principal de estas técnicas es que los individuos del conjunto siguen siendo **vulnerables a ataques por singularización**, dado que son identificables por los pseudónimos y/o tokens. Además, **tampoco protegen frente a vinculabilidad o inferencia**, especialmente en aquellos casos donde se reutilicen atributos seudonimizados en diferentes conjuntos de datos, por ejemplo, por reutilización de claves. También **podría realizarse vinculación a partir de otros atributos del conjunto**.

3.3.1 Cifrado y funciones HASH

El problema principal del cifrado con clave es el **riesgo de que un atacante consiga la clave** y pueda revertir el proceso. Aún sin conocer la clave, un atacante podría hacer uso de un algoritmo para intentar descubrirla, normalmente por fuerza bruta (prueba masiva de combinaciones).

En la práctica, si se emplean algoritmos actualizados de cifrado es muy complejo que un atacante pueda descifrar los valores, pero **las garantías se reducen con el paso del tiempo**, principalmente debido a los **avances tecnológicos**, como el ejemplo ya expuesto sobre computación cuántica.

De hecho, cualquier conjunto de datos cifrado con SHA256 está en **riesgo de ser descifrado en un futuro relativamente cercano**, por ejemplo cuando sea factible aplicar el algoritmo de Shor a escala mediante computadores cuánticos. Es por eso que muchos sistemas críticos financieros, y en especial las criptomonedas, ya están comenzando a aplicar algoritmos más avanzadas que no se basen en RSA, como el cifrado por curva elíptica y otros tipos de cifrado post-cuántico.

Es habitual que el cifrado se complemente con una función HASH o función resumen, como es el caso del ya mencionado algoritmo SHA256. El objetivo principal de una función HASH es **generar un valor de longitud determinada** a partir de otro valor o conjunto de valores. Normalmente son funciones con **bajo coste computacional**, para que sean eficientes, lo cual hace que sean especialmente vulnerables a ataques por fuerza bruta.

Estas funciones **no sólo se usan en el ámbito de la criptografía**, sino que son empleadas también para crear índices de búsqueda en bases de datos, sumas de verificación (checksum), pruebas de integridad, compresión de datos, etc.

RECOMENDACIONES

- En muchas ocasiones se añade un **valor especial denominado salt (sal) para añadir garantías adicionales** a los procesos de HASH. Es muy habitual, por ejemplo, a la hora de almacenar contraseñas. El uso de un **valor aleatorio sin custodia evita que el proceso sea reversible**.
- Si el proceso de cifrado incluye el **borrado de la clave secreta**, el conjunto resultante ofrece **mejores garantías frente a vinculabilidad a partir de otros conjuntos de datos**.

3.3.2 Descomposición en tokens

Se aplica de forma habitual en entornos financieros. Es una técnica que se basa en los mismos principios vistos en el apartado anterior, donde normalmente se aplica un **proceso unidireccional o irreversible**.

RECOMENDACIONES

- El principio básico consiste en usar **seudónimos o tokens diferentes en cada conjunto de datos, para reducir la vinculabilidad**. El coste computacional de un posible ataque se incrementa de forma significativa, haciendo que resulte inviable dada la enorme cantidad de combinaciones posibles.

3.3.3 Cifrado homomórfico

El cifrado homomórfico es un concepto relacionado con el cifrado ordenable. El objetivo de esta técnica es permitir **realizar operaciones sobre datos cifrados que resulten equivalentes a otras operaciones aplicadas sobre el conjunto original**.

Gracias a esta técnica es posible compartir datos cifrados, **aplicar operaciones sin tener acceso a la información sensible** y posteriormente descifrar el resultado final.

La aplicación práctica de estos algoritmos es limitada en la actualidad, aunque se esperan avances en los próximos años relacionado con el concepto de cifrado totalmente homomórfico o FHE (*fully homomorphic encryption*), es decir, sistemas de cifrado homomórfico que soportan tanto operaciones de suma como de producto.

RECOMENDACIONES

- Sólo es factible el uso de aplicaciones de cifrado homomórfico en **casos de uso muy limitados** y normalmente enmarcados en el ámbito de la investigación de nuevos algoritmos.
- En la actualidad ya existen diseños de sistemas FHE, pero todavía **no son lo suficientemente rápidos para poder ser usados en aplicaciones reales prácticas**.

3.4 GARANTÍAS

En la **TABLA 1**, adaptada del Dictamen 05/2014, se resume, a modo indicativo, el posible **nivel de garantías de cada tipo de técnica:**

CATEGORÍA	TÉCNICA	RIESGO DE SINGULARIZACIÓN	RIESGO DE VINCULABILIDAD	RIESGO DE INFERENCIA
ALEATORIZACIÓN	Adición de ruido	Sí	A veces	A veces
	Permutación	Sí	A veces	A veces
	Privacidad diferencial	A veces	A veces	A veces
GENERALIZACIÓN	Anonimato-K	No	Sí	Sí
	Diversidad-L	No	Sí	A veces
	Proximidad-T	No	Sí	A veces
SEUDONIMIZACIÓN	Cifrado por hash	Sí	Sí	A veces
	Tokenización	Sí	Sí	Sí

Tabla 1. *Nivel de garantías aportado por cada técnica de anonimización*

3.4.1 Singularización

Algunas técnicas como la adición de ruido o la permutación pueden reducir el riesgo de singularización de forma parcial, aunque **no evitan que se puedan singularizar los registros de una persona**, a veces de manera no identificable, haciendo simplemente que los datos filtrados sean menos fiables.

Las **técnicas basadas en seudonimización** no ofrecen prácticamente garantías en este sentido, siendo habitualmente **empleadas de forma errónea sin combinar con otras técnicas**.

Únicamente una **combinación adecuada de técnicas** que incluyan agregación por Anonimato-K, Diversidad-L y/o Privacidad Diferencial pueden aportar **garantías respecto a singularización**.

3.4.2 Vinculabilidad

La vinculabilidad es **uno de los riesgos más complejos de minimizar**, aumentando con el paso del tiempo o a partir del acceso a nuevos datos, especialmente cuando se aplica Privacidad Diferencial.

En algunos casos, especialmente en la adición de ruido y permutación, se podría vincular a un individuo por error con datos incorrectos o artificiales, que incluso podría generar un **impacto incluso mayor para el individuo por una atribución incorrecta** (como ser asociado con una enfermedad que no padece).

3.4.3 Inferencia

La mayoría de las técnicas expuestas en este informe, salvo el Anonimato-K o la Seudonimización, **reducen significativamente el riesgo de inferencia**, aumentando el volumen de falsos positivos y/o falsos negativos cuando un actor malicioso intenta identificar los individuos asociados a un conjunto de datos anonimizado, haciendo mucho más complejo realizar ataques de este tipo.

4. CASO PRÁCTICO

En esta sección vamos a ver un ejemplo práctico sencillo de cómo se pueden aplicar algunas técnicas descritas en el apartado anterior, concretamente **Anonimato-K y seudonimización mediante cifrado con borrado de clave**.

4.1 METODOLOGÍA Y OBJETIVOS

El objetivo del caso práctico es presentar una serie de **ejemplos didácticos a modo exploratorio mediante un enfoque iterativo de prueba y error**. Es importante remarcar que se trata de ejemplos sencillos y acotados, diseñados para afianzar los conceptos explicados en apartados anteriores.

Los ejemplos presentados no son casos reales, por lo que es relativamente complejo argumentar las decisiones desde un punto de vista de negocio, dado que **sería necesario plantear los pasos previos del proceso** recogidos en el capítulo 2, y que quedan fuera de alcance en este caso práctico (definición del equipo, evaluación de riesgos, definición de objetivos y finalidad del conjunto anonimizado, etc.).

A nivel metodológico, es habitual que un proceso de anonimización implique igualmente un proceso manual de exploración de datos y aplicaciones de técnicas de anonimización mediante prueba y error. Veremos varios ejemplos que nos permiten mejorar el nivel de Anonimato-K para valores de K cada vez más altos, **en algunos casos el ejemplo presentará un proceso que no nos permite mejorar o no es suficiente**, pero se incluyen asimismo para dar una visión práctica y cercana a la realidad de cómo suele ser un proceso de anonimización.

Por otro lado, es importante también tener en cuenta que existe una gran variedad de estrategias, herramientas y algoritmos que permiten mejorar el nivel de anonimización de un conjunto de datos, siendo **un proceso complejo, vinculado a la finalidad del caso de uso del análisis y el nivel de utilidad y garantías requerido**.

En este caso práctico:

1) **Presentamos primero el proceso exploratorio, aplicando varios tratamientos de forma iterativa y comprobando el nivel de garantías obtenido después de cada paso**, por ejemplo, mediante varias generalizaciones y discretizaciones de salarios, edades o códigos postales. Al ir añadiendo estos pasos se consigue mejorar el nivel de garantías ofrecido (aumentando el grado de Anonimato-K obtenido), minimizando los riesgos de singularización, vinculación e inferencia vistos anteriormente, siendo una práctica habitual.

2) Una vez presentados los ejemplos, **se presentan dos resultados finales a modo de posibles escenarios de procesos completos de anonimización**, ofreciendo un conjunto de datos diferente en cada caso, **con diferentes niveles de utilidad y garantías**.

En un caso real, sería **el responsable de la anonimización quien debe decidir si un nivel específico para el Anonimato-K es suficiente**, **o si por el contrario es necesario agrupar o generalizar de forma más intensiva.** Normalmente estas cuestiones se basan en el nivel de riesgo asumible, el tamaño del conjunto o la utilidad de los datos resultantes, entre otros criterios a valorar y que **se definen a modo de requisitos en las fases iniciales de diseño del proceso**.

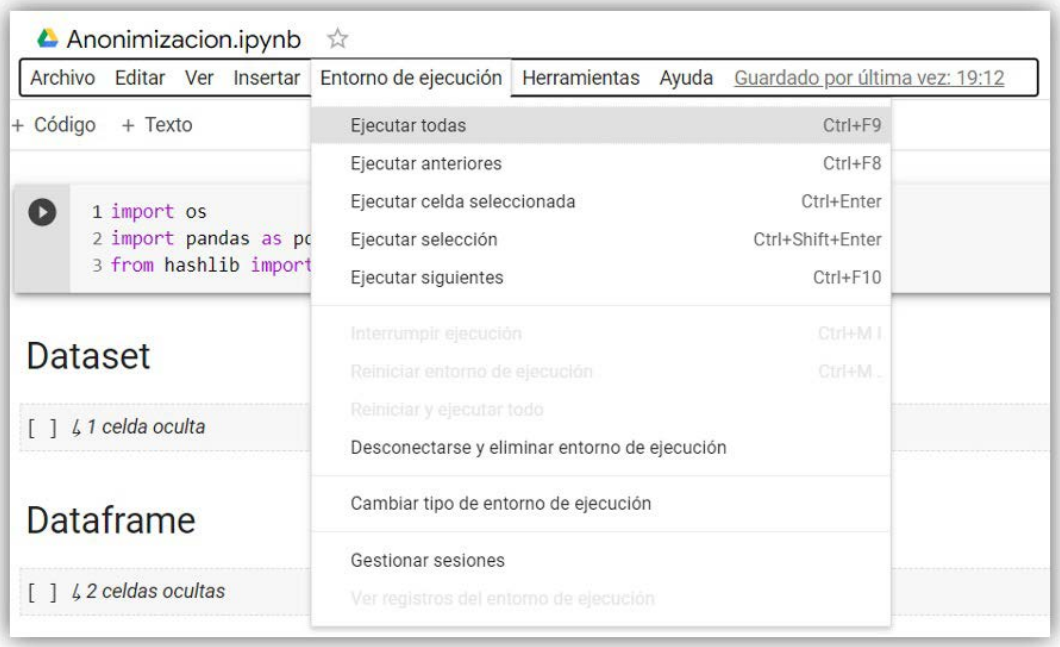
4.2 ENTORNO DE EJECUCIÓN

Para simplificar la ejecución del caso se provee un fichero Python en formato Jupyter Notebook, que se puede cargar en cualquier entorno compatible. El fichero ha sido probado sobre Google Colab, que ofrece un entorno de ejecución gratuito para este tipo de código fuente interactivo. La versión de Python empleada durante el diseño del caso fue la 3.7.

Para ejecutar el código fuente, basta con acceder al enlace mostrado a continuación

- Link al código ejecutable cargado desde Google Colab.
- El siguiente repositorio de Github, contiene el código y los datos utilizados en este caso práctico, los cuales se cargan automáticamente desde Google Colab:

Una vez cargado el fichero se pueden ejecutar todas las celdas mediante el menú "Entorno de ejecución", seleccionando la opción "Ejecutar todas" o bien ejecutar las celdas una a una de forma interactiva:



4.3 CONFIGURACIÓN DEL CASO

Antes de nada, es necesario indicar qué librerías vamos a necesitar para desarrollar el caso práctico, lo cual se define en la primera celda del Notebook:

```
import os
import pandas as pd
from hashlib import sha256, blake2b, algorithms_available
```

En concreto, la librería hashlib es una pieza clave del caso, dado que se va a emplear para aplicar **técnicas de cifrado sobre los identificadores de las tarjetas de crédito**, como veremos más adelante.

Esta librería dispone de algoritmos alternativos, que dependen del entorno sobre el que se ejecuta. Para comprobar los algoritmos disponibles se puede consultar directamente:

```
for name in algorithms_available:
  print(name)

sha3_256
md5
sha3_384
sha512
blake2s
sha3_512
sha224
shake_128
sha384
shake_256
blake2b
sha256
sha3_224
sha1
```

4.4 CONJUNTO DE DATOS

El conjunto de datos empleado está generado de forma sintética a partir de los datos proporcionados por el proyecto Amnesia. En concreto es una variante del ejemplo "Simple Table-Disk based simple table", adaptado al caso que se va a exponer. No se emplean fuentes con datos abiertos reales porque normalmente estos datos ya están anonimizados y resultaría más complejo de exponer el caso.

La estructura del conjunto incluye 5 variables:

1. tarjeta de crédito (**creditcard, string**)
2. código postal (**zipcode, string**)
3. edad en años (**age, integer**)
4. género (**gender, string**)
5. salario bruto mensual (**salary, integer**)

Los datos se cargan directamente en el Notebook desde la carpeta "*Datos*" del repositorio de Github. La matriz completa incluye 999 filas y 5 columnas, y se construye como un DataFrame de Pandas, que es una **estructura de datos habitual para poder manipular datos con formato de tabla en Python**:

```
url = https://raw.githubusercontent.com/datosgobes/Laboratorio-de-Datos/main/Data%20Science/Aplicaci%C3%B3n%20pr%C3%A1ctica%20de%20t%C3%A9cnicas%20de%20anonimizaci%C3%B3n/Datos/data.csv

df = pd.read_csv(url, dtype={
    'creditcard': str,
    'zipcode': str,
    'age': int,
    'gender': str,
    'salary': int
})
```

A continuación, se muestra un extracto de los diez primeros registros a modo de ejemplo:

```
df.head(10)
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55335	58	Male	8700
1	5418686973265201	55255	36	Female	9700
2	5527060358825468	55559	32	Female	6800
3	5312916958971375	55700	58	Male	4700
4	5541858987662877	55925	52	Male	5700
5	5155271703366251	55338	38	Female	7100
6	5485337334153888	55840	38	Male	6000
7	5293804792403628	55772	32	Female	7000
8	5275938856549264	55641	19	Male	100
9	5303041772852809	55861	82	Male	4000

Procedemos también a realizar una copia de respaldo o backup, para poder deshacer alguno de los pasos sin necesidad de recomenzar desde cero:

```
df_backup = df.copy()
```

4.5 COMPROBACIÓN DEL ANONIMATO-K

Para poder validar los diferentes pasos del caso práctico, vamos a definir **tres funciones sencillas que nos permitan saber si el conjunto cumple con un valor de K específico**. Inicialmente el conjunto cumple de forma trivial con K=1, dado que cualquier individuo forma un grupo consigo mismo, de tamaño 1 o mayor, pero **no cumple para K=2 o superior**. Estas funciones se inspiran en el ejemplo disponible en el segundo capítulo del libro Programming Differential Privacy.

4.5.1 Consulta de búsqueda

Para la validación es necesario **definir una consulta de búsqueda de registros** por el grupo de variables que se hayan definido para el proceso de Anonimato-K, en este caso concreto emplearemos todas las variables menos la tarjeta de crédito, puesto que se trata de un identificador único y la trataremos de forma independiente más adelante:

```
def queryKAnonymized(row):
  return f'zipcode == \'{row.zipcode}\'' \
         f' & gender == \'{row.gender}\'' \
         f' & age == {row.age}' \
         f' & salary == {row.salary}'
```

Como puede observarse, la consulta simplemente define **qué campos se van a comprobar a partir de los datos de una fila concreta del conjunto**, y dependiendo del tipo de datos se rodean de comillas (textos: *zipcode y gender*) o no (números: *age y salary*). Para incluir una comilla simple se usa un carácter de escape.

Esta función únicamente **define las variables y los tipos de datos que intervienen en el proceso** de Anonimato-K y depende de cada caso concreto. A lo largo de los siguientes ejemplos se definirán algunas funciones complementarias que modifican tanto la lista de variables como sus tipos.

4.5.2 Búsqueda de grupos

Una vez definida la consulta específica para el caso concreto, se puede pasar como argumento a una función genérica que hemos definido para **buscar grupos dentro de un conjunto de datos** (se define la consulta *queryKAnonymized* como valor por defecto para evitar tener que indicarla). El objetivo de esta función es **validar que todos los ejemplos pertenecen a un grupo de al menos K individuos y por tanto el conjunto cumpliría los requisitos del Anonimato-K para el valor de K indicado** (y todos los valores de K inferiores). Es decir, esta nueva función permite comprobar para cada fila si pertenece a un grupo de al menos K individuos. Si todas las filas cumplen la condición devuelve el valor *True* (verdadero), si encuentra al menos 1 fila que no la cumpla devuelve el valor *False* (falso):

```
def isKAnonymized(df, k, queryFunction = queryKAnonymized):
  for index, row in df.iterrows():
    if df.query(queryFunction(row)).shape[0] < k: return False
  return True
```

Si aplicamos la función isKAnonymized al conjunto original vemos que el resultado es válido para K=1, pero no para K=2 o superior, es decir, **el conjunto original sólo cumple el Anonimato-K hasta K=1**:

```
isKAnonymized(df, 1)
True
isKAnonymized(df, 2)
False
```

4.5.3 Búsqueda de registros no agrupados

El siguiente paso es definir una función complementaria para poder **encontrar qué registros concretos no cumplen con la condición K definida** (se especifica también la consulta queryKAnonymized como valor por defecto), es decir, el objetivo es poder **depurar o identificar los registros que no cumplen las condiciones indicadas** para analizarlos y tratar de corregirlos mediante una nueva iteración:

```
def getNotKAnonymized(df, k, queryFunction = queryKAnonymized):
  rowsNotKAnonymized = pd.DataFrame()
  for index, row in df.iterrows():
    group = df.query(queryFunction(row))
    if group.shape[0] < k:
      rowsNotKAnonymized = pd.concat([rowsNotKAnonymized, group])
  return rowsNotKAnonymized.drop_duplicates()
```

Al aplicar esta función sobre el conjunto original **con K=1 devuelve un conjunto vacío, porque todas las filas cumplen la condición**. Por el contrario, si la aplicamos **con K=2 nos devuelve el conjunto completo, dado que ninguna fila cumple Anonimato-K para K=2 o superior**.

```
getNotKAnonymized(df, 2)
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55335	58	Male	8700
1	5418686973265201	55255	36	Female	9700
2	5527060358825468	55559	32	Female	6800
3	5312916958971375	55700	58	Male	4700
4	5541858987662877	55925	52	Male	5700
...	...	...	...	...	...
994	5395287779118434	55640	82	Female	1700
995	5564301173493387	55067	21	Female	7400
996	5193534712511173	55901	23	Female	8100
997	5164269869571382	55601	52	Male	3500
998	5587367312759585	55547	47	Male	9900

999 rows × 5 columns

Este es el caso habitual antes de aplicar un proceso de anonimización, **de forma trivial todos los registros cumplen para K=1 (forman un grupo de 1 individuo) y normalmente ninguno (o la mayoría) no cumple para valores de K superiores**. En otros ejemplos que veremos a continuación, según se van aplicando diferentes técnicas, **el número de registros que no cumple se reduce, hasta conseguir que el conjunto de datos anonimizado cumpla el Anonimato-K hasta un cierto valor de K**.

4.6 GENERALIZACIÓN POR REDONDEO

Una forma sencilla de generalizar una variable es redondear o enmascarar los valores menos significativos de un número o un código.

4.6.1 Generalización de números enteros por redondeo

En el caso de variables numéricas de tipo entero el proceso consiste en **redondear unidades en decenas, decenas en centenas y así sucesivamente**. A continuación, se muestra una función para aplicar esta generalización sobre una columna concreta de un DataFrame, el parámetro level indica cuantos niveles se desea generalizar:

```
def generalizeInt(df, column, level):
  return df[column].apply(
    lambda x: round(x / (10**level)) * (10**level)
  )
```

En este caso vamos a generalizar la edad a decenas y el salario a millares, a partir del conjunto original:

```
df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55335	58	Male	8700
1	5418686973265201	55255	36	Female	9700
2	5527060358825468	55559	32	Female	6800
3	5312916958971375	55700	58	Male	4700
4	5541858987662877	55925	52	Male	5700

```
df.salary = generalizeInt(df, 'salary', 3)

df.age = generalizeInt(df, 'age', 1)
```

Podemos comprobar el resultado mostrando las primeras cinco filas:

```
df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55335	60	Male	9000
1	5418686973265201	55255	40	Female	10000
2	5527060358825468	55559	30	Female	7000
3	5312916958971375	55700	60	Male	5000
4	5541858987662877	55925	50	Male	6000

Tras la generalización, comprobamos si ha mejorado para K=2, pero vemos que no es suficiente, dado que aún **existen registros que no pertenecen a un grupo de al menos 2 individuos**:

```
isKAnonymized(df, 2)
False
```

Si comprobamos la longitud del resultado de la función que identifica los registros que no cumplen para K=2, vemos que existen 997 filas que aún no cumplen, es decir, sólo hemos conseguido agrupar dos registros, de los 999 del conjunto, por lo que **es necesario aplicar procesos adicionales aún si queremos mejorar el Anonimato-K para K=2 o superior**:

```
len(getNotKAnonymized(df, 2))
997
```

4.6.2 Generalización de códigos alfanuméricos por redondeo

Vamos a probar a generalizar también el código postal con un procedimiento similar, salvo que en este caso al tratarse de un código almacenado como texto empleamos **una nueva función que cambia caracteres desde la derecha por asteriscos**. El parámetro level permite indicar cuántos:

```
def generalizeStringCode(df, column, level):
  return df[column].apply(
    lambda x: x[:-level] + ('*' * level)
  )
```

Probamos a generalizar el código postal enmascarando los dos últimos valores:

```
df.zipcode = generalizeStringCode(df, 'zipcode', 2)
```

Volvemos a comprobar sin éxito si el nuevo conjunto cumple para K=2:

```
isKAnonymized(df, 2)
False
```

Y si buscamos qué registros no cumplen, vemos que **aún quedan 603 filas que no cumplen la condición para K=2, porque no pertenecen a un grupo de al menos dos registros**:

```
getNotKAnonymized(df, 2)
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	553**	60	Male	9000
1	5418686973265201	552**	40	Female	10000
2	5527060358825468	555**	30	Female	7000
4	5541858987662877	559**	50	Male	6000
7	5293804792403628	557**	30	Female	7000
...	...	...	...	...	...
990	5190197578253687	554**	100	Male	4000
993	5342057613343975	556**	50	Male	9000
995	5564301173493387	550**	20	Female	7000
996	5193534712511173	559**	20	Female	8000
997	5164269869571382	556**	50	Male	4000

603 rows × 5 columns

4.7 GENERALIZACIÓN POR AGRUPACIÓN

Otra forma habitual de generalización es mediante jerarquías y/o discretización de valores. Es decir, el objetivo es **agrupar rangos de valores en categorías predefinidas mediante reglas**.

Normalmente estas jerarquías tienen varios niveles, pero por simplicidad vamos a emplear un único nivel en este ejemplo. En concreto, el objetivo que buscamos es poder **transformar los salarios por rangos, en 3 grupos (bajo: 0-1.500, medio: 1.500-3.000 y alto: 3.000-10.000)**.

Definimos dos funciones, la primera simplemente **aplica las reglas de transformación sobre un valor concreto de una variable**, modificándolo por el valor de una categoría de la jerarquía, en caso de no encontrar ninguna regla se define el valor como "outlier":

```
def applyRules(x, rules):
  for key in rules:
    if (x >= rules[key]['min'] and x <= rules[key]['max']): return key
  return "outlier"
```

La segunda función sigue un proceso similar a las funciones para generalizar números que vimos anteriormente, **modificando el valor de la columna especificada para cada fila del *DataFrame*, aplicando las reglas que se indiquen**:

```
def groupDiscretization(df, column, rules):
  return df[column].apply(lambda x: applyRules(x, rules))
```

4.7.1 Discretización del salario en 3 grupos

Antes de nada, vamos a **restaurar la variable original para poder aplicar la discretización sobre la variable sin redondear**:

```
df.salary = df_backup.salary

df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	553**	60	Male	8700
1	5418686973265201	552**	40	Female	9700
2	5527060358825468	555**	30	Female	6800
3	5312916958971375	557**	60	Male	4700
4	5541858987662877	559**	50	Male	5700

Una vez restaurada, definimos las reglas de generalización, las cuales **dependen de la finalidad del conjunto, la distribución de los datos y el nivel de anonimización que se espera conseguir**. En este caso vamos a probar a generalizar el salario en 3 grupos (0-1.500, 1.500-3.000 y 3.000-10.000):

```
salaryRules = {
    'low': {'min': 0, 'max': 1500},
    'medium': {'min': 1500, 'max': 3000},
    'high': {'min': 3000, 'max': 10000}
}
```

Con las reglas definidas, simplemente tenemos que aplicar la función de generalización y comprobar el resultado:

```
df.salary = groupDiscretization(df, 'salary', salaryRules)

df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	553**	60	Male	high
1	5418686973265201	552**	40	Female	high
2	5527060358825468	555**	30	Female	high
3	5312916958971375	557**	60	Male	high
4	5541858987662877	559**	50	Male	high

Como hemos modificado el tipo de datos del salario, es necesario redefinir la consulta de búsqueda para la validación del Anonimato-K:

```
def queryKAnonymized_salaryGrouped(row):
  return f'zipcode == \'{row.zipcode}\'' \
         f' & gender == \'{row.gender}\'' \
         f' & age == {row.age}' \
         f' & salary == \'{row.salary}\''
```

Procedemos a comprobar el Anonimato-K para K=2, pero aún existen 155 registros que no cumplen la condición:

```
isKAnonymized(df, 2, queryKAnonymized_salaryGrouped)
False

len(getNotKAnonymized(df, 2, queryKAnonymized_salaryGrouped))
155
```

4.7.2 Discretización de edad en 3 grupos

Vamos a intentar conseguir un nivel superior de anonimato mediante la agregación de las edades en tres grupos (junior: 0-30, medium: 30-50 y senior: 50-101). Para ello primero restauramos los valores originales y comprobamos la distribución:

```
df.age = df_backup.age

df.describe()
```

	age
count	999.000000
mean	57.364364
std	24.116729
min	18.000000
25%	35.000000
50%	57.000000
75%	79.000000
max	100.000000

```
df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	553**	58	Male	8700
1	5418686973265201	552**	36	Female	9700
2	5527060358825468	555**	32	Female	6800
3	5312916958971375	557**	58	Male	4700
4	5541858987662877	559**	52	Male	5700

Las reglas se definen para 3 grupos (junior: 0-30, medium: 30-50 y senior: 50-101):

```
ageRules = {
    'junior': {'min': 0, 'max': 30},
    'medium': {'min': 30, 'max': 50},
    'senior': {'min': 50, 'max': 101}
}

df.age = groupDiscretization(df, 'age', ageRules)

df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	553**	senior	Male	high
1	5418686973265201	552**	medium	Female	high
2	5527060358825468	555**	medium	Female	high
3	5312916958971375	557**	senior	Male	high
4	5541858987662877	559**	senior	Male	high

Redefinimos una vez más la consulta de búsqueda:

```
def queryKAnonymized_salaryAgeGrouped(row):
  return f'zipcode == \'{row.zipcode}\'' \
         f' & gender == \'{row.gender}\'' \
         f' & age == \'{row.age}\'' \
         f' & salary == \'{row.salary}\''
```

Y comprobamos que para para K=2 aún quedan 30 filas que no cumplen la condición, es decir, esas 30 filas aún no se han podido agrupar en un grupo de 2 o más registros:

```
len(getNotKAnonymized(df, 2, queryKAnonymized_salaryAgeGrouped))
30
```

4.7.3 Discretización de código postal en 3 grupos

A modo de ejemplo, intentamos ahora mejorar la anonimización mediante la agregación de códigos postales en 3 grupos (550**-552**, 553**-556** y 557**-559**):

```
zipcodeRules = {
    '550**-552**': {'min': '550**', 'max': '552**'},
    '553**-556**': {'min': '553**', 'max': '556**'},
    '557**-559**': {'min': '557**', 'max': '559**'}
}
df.zipcode = groupDiscretization(df, 'zipcode', zipcodeRules)
df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	553**-556**	senior	Male	high
1	5418686973265201	550**-552**	medium	Female	high
2	5527060358825468	553**-556**	medium	Female	high
3	5312916958971375	557**-559**	senior	Male	high
4	5541858987662877	557**-559**	senior	Male	high

Volvemos a comprobar para K=2 y en este caso si se consigue con éxito:

```
len(getNotKAnonymized(df, 2, queryKAnonymized_salaryAgeGrouped))
0
```

Si bien para K=3 aún existen dos grupos de 2 registros cada uno que no cumplen la condición de pertenecer a un grupo de al menos 3 registros:

```
getNotKAnonymized(df, 3, queryKAnonymized_salaryAgeGrouped)
```

	creditcard	zipcode	age	gender	salary
233	5148234499441506	550**-552**	medium	Female	low
561	5434614557694547	550**-552**	medium	Female	low
636	5184789207964060	557**-559**	junior	Female	medium
718	5494158926215755	557**-559**	junior	Female	medium

4.7.4 Discretización de código postal en 2 grupos

Volvemos a probar, ahora con un ejemplo de discretización en 2 grupos del código postal. Para lo cual primero restauramos la variable original.

```
df.zipcode = df_backup.zipcode
```

No es necesario volver a aplicar el paso de generalización, porque la discretización que vamos a definir ya cubre esos segmentos, pero en un caso real podría tener sentido aplicar ambos pasos.

A continuación, definimos 2 grupos (55000-55499 y 55500-55999):

```
zipcodeRulesTwoGroups = {
    '55000-55499': {'min': '55000', 'max': '55499'},
    '55500-55999': {'min': '55500', 'max': '55999'}
}
df.zipcode = groupDiscretization(df, 'zipcode', zipcodeRulesTwoGroups)
df.head()
```

	creditcard	zipcode	age	gender	salary
0	5557783527541459	55000-55499	senior	Male	high
1	5418686973265201	55000-55499	medium	Female	high
2	5527060358825468	55500-55999	medium	Female	high
3	5312916958971375	55500-55999	senior	Male	high
4	5541858987662877	55500-55999	senior	Male	high

Vemos que con este cambio conseguimos Anonimato-K para K=5 (todos los registros pertenecen a un grupo de al menos 5):

```
len(getNotKAnonymized(df, 5, queryKAnonymized_salaryAgeGrouped))
0
```

Al comprobar K=6, todavía existen 5 grupos de 5 registros que no cumplen la condición:

```
getNotKAnonymized(df, 6, queryKAnonymized_salaryAgeGrouped)
```

	creditcard	zipcode	age	gender	salary
195	5282876440236496	55000-55499	junior	Male	low
709	5338783612672926	55000-55499	junior	Male	low
790	5590375584035431	55000-55499	junior	Male	low
831	5165553732142557	55000-55499	junior	Male	low
926	5467404471699503	55000-55499	junior	Male	low
227	5243495311011488	55000-55499	medium	Female	low
233	5148234499441506	55000-55499	medium	Female	low
405	5342140519111603	55000-55499	medium	Female	low
561	5434614557694547	55000-55499	medium	Female	low
587	5550229459841456	55000-55499	medium	Female	low
229	5263998877517553	55500-55999	medium	Male	medium
250	5409283975842483	55500-55999	medium	Male	medium
569	5484842165970285	55500-55999	medium	Male	medium
742	5475789913613298	55500-55999	medium	Male	medium
785	5396233745178507	55500-55999	medium	Male	medium
325	5246266944629831	55000-55499	junior	Female	medium
394	5104006338696280	55000-55499	junior	Female	medium
410	5260801451077338	55000-55499	junior	Female	medium
503	5133669132778680	55000-55499	junior	Female	medium
988	5159593977095175	55000-55499	junior	Female	medium
373	5492256342135698	55000-55499	junior	Female	low
465	5407119200524540	55000-55499	junior	Female	low
555	5574301016330760	55000-55499	junior	Female	low
701	5512074870840795	55000-55499	junior	Female	low
949	5338877878438061	55000-55499	junior	Female	low

En este punto el responsable de la anonimización debería decidir si un nivel K=5 para el Anonimato-K es suficiente, o si por el contrario es necesario agrupar o generalizar de forma más intensiva. Normalmente estas cuestiones se basan en el nivel de riesgo asumible, el tamaño del conjunto o la utilidad de los datos resultantes, entre otros criterios a valorar y que se definen a modo de requisitos en las fases iniciales de diseño del proceso.

4.8 FILTRADO DE VARIABLES

Una posibilidad dada la limitación que implica el incluir el código postal a la hora de aumentar el nivel de Anonimato-K por encima de K=5 es descartar dicha variable. Como veremos esta decisión **implica una reducción importante de la utilidad del conjunto** si el caso de uso requiere analizar la distribución geográfica, por lo que es una decisión que dependerá de la finalidad del análisis.

En este ejemplo vamos a probar el nivel de anonimización que se podría conseguir, simplemente descartando la variable y manteniendo el resto de las transformaciones para el resto:

```
df_nozip = df.copy()
df_nozip.drop(columns=['zipcode'], inplace=True)
df_nozip.head()
```

	creditcard	age	gender	salary
0	5557783527541459	senior	Male	high
1	5418686973265201	medium	Female	high
2	5527060358825468	medium	Female	high
3	5312916958971375	senior	Male	high
4	5541858987662877	senior	Male	high

Volvemos a definir la consulta de búsqueda sin tener en cuenta el código postal, dado que ya no existe:

```
def queryKAnonymized_noZip(row):
  return f'gender == \'{row.gender}\'' \
         f' & age == \'{row.age}\'' \
         f' & salary == \'{row.salary}\''
```

Si probamos con niveles progresivos de K, podemos comprobar que mejoramos el Anonimato-K hasta el nivel K=11 (todos los ejemplos pertenecen a un grupo de al menos 11 registros):

```
len(getNotKAnonymized(df_nozip, 11, queryKAnonymized_noZip))
0
```

Si bien para K=12 encontramos dos grupos de 11 registros que no cumplirían:

```
getNotKAnonymized(df_nozip, 12, queryKAnonymized_noZip)
```

	creditcard	age	gender	salary
17	5217036561413487	junior	Female	low
53	5431264098786171	junior	Female	low
373	5492256342135698	junior	Female	low
465	5407119200524540	junior	Female	low
555	5574301016330760	junior	Female	low
630	5308576867572221	junior	Female	low
634	5438231572216713	junior	Female	low
701	5512074870840795	junior	Female	low
802	5416287053945374	junior	Female	low
902	5209381246715967	junior	Female	low
949	5338877878438061	junior	Female	low
325	5246266944629831	junior	Female	medium
394	5104006338696280	junior	Female	medium
410	5260801451077338	junior	Female	medium
414	5583319892315662	junior	Female	medium
503	5133669132778680	junior	Female	medium
560	5418961461593616	junior	Female	medium
636	5184789207964060	junior	Female	medium
643	5324724299778970	junior	Female	medium
718	5494158926215755	junior	Female	medium
904	5563358313867293	junior	Female	medium
988	5159593977095175	junior	Female	medium

4.9 CIFRADO Y SEUDONIMIZACIÓN

Hasta ahora hemos abordado el problema de anonimización desde la aplicación del Anonimato-K, **consiguiendo diferentes niveles de K en función del grado de generalización aplicado**, donde la decisión final depende de la finalidad del análisis y el nivel de garantías deseado.

Si bien, no hemos abordado el **riesgo de re-identificación que existe al compartir el número de tarjeta de crédito de cada individuo**. Podríamos generalizar los números, pero por la naturaleza de la variable (identificador único) tiene más sentido ofuscar sus valores para reducir los riesgos asociados.

4.9.1 Cifrado SHA256

Como ya comentamos anteriormente, el cifrado SHA256 es una técnica bastante habitual de cifrado basado en RSA. Su aplicación mediante la librería hashlib de Python es muy sencilla:

```
def encodeSHA256(df, column):
  return df[column].apply(
    lambda x: sha256(x.encode('utf-8')).hexdigest()
  )
encodeSHA256(df, 'creditcard')
```

```
0      8d184c9660fb4fb4020b668ae72331c058ff2ecee27339...
1      71c24d0c3e9886856df731e8b7f79f8c84549b9199fd83...
2      4be30506958bf04b22543e3854e3b3252956719c0a10d8...
3      4fac46c6a4cec927a80f3fad07befd4008cfc1b409294f...
4      0b403cbfc500b119e1b5eab465a0e0c1cb5bdb9c76e16a...
                           ...
994    324e5152c96e530d3a07c783c57d327df37d4750c3ca36...
995    8e9f2919c79ecf9f62642ff4138e99aa63ef95174b9706...
996    7b53f2d60091e7d8fcd953364a1583cfcda26c3c63d856...
997    caf6a4dda3f2c1060b2594e209f0911dc914b0eb0a9c41...
998    64541d50ad0c31425e7f46b609d0ee979a18e4c59bb754...
Name: creditcard, Length: 999, dtype: object
```

Este algoritmo es determinista, es decir, **siempre devuelve el mismo valor cifrado (*hash*) para los mismos valores de entrada**. Lo cual es ventajoso a la hora de descifrar, pero también es vulnerable a ataques por fuerza bruta, por ejemplo, si comprobamos los cifrados generados en dos llamadas diferentes vemos que devuelve la longitud del del conjunto (todos son iguales):

```
sum(encodeSHA256(df, 'creditcard') == encodeSHA256(df, 'creditcard'))
999
```

4.9.2 Cifrado Blake2b

El algoritmo Blake2b es una versión mejorada de SHA, con garantías similares, pero más eficiente y flexible en algunos aspectos. Podemos acceder a una implementación desde la librería hashlib.

Lo primero que vamos a definir es una función auxiliar para codificar un texto. Los parámetros permiten definir la longitud del valor cifrado en bytes (*size*), el dominio (*domain*) sobre el que se ejecuta y el valor de sal (*salt*).

```
def encodeStringBLAKE2B(x, size=64, domain=b'', salt=b''):
  h = blake2b(
      digest_size = size,
      person = domain,
      salt = salt
  )
  h.update(x.encode('utf8'))
  return h.hexdigest()
```

A partir de dicha función podemos definir una función que aplique dicha transformación a todas las filas de una columna de un DataFrame:

```
def encodeBLAKE2B(df, column, size=64, domain=b'', salt=b''):
  return df[column].apply(
    lambda x: encodeStringBLAKE2B(x, size, domain, salt)
  )
```

Probamos con los parámetros por defecto y tamaño 10 (se escoge este tamaño para mejorar la legibilidad, en entornos reales es mejor emplear la longitud máxima permitida):

```
encodeBLAKE2B(df, 'creditcard', size=10)
```

```
0      174f11dd2cc3672d9510
1      36c6c74ac8b337b06745
2      228fce22fb2e00a8a54f
3      734e901f4319cc18f393
4      5f5131cc88aaefa23a43
              ...
994    30c725768c8584a56197
995    1d964c8df977df307bbb
996    21e72784fdd0d6b9c1c8
997    ede715899bbf4b2dc553
998    ead99cce8f346bbabe78
Name: creditcard, Length: 999, dtype: object
```

Al igual que con SHA256, este tipo de configuración es determinista y por tanto vulnerable a ataques por fuerza bruta. Vamos a ver a continuación cómo mejorar la protección de esta variable.

4.9.3 Cifrado Blake2b con gestión de dominio

Una de las opciones que permite este algoritmo es la definición de un dominio (*domain*), que evita que se generen dos valores cifrados iguales en dos contextos diferentes para el mismo valor, haciendo más complejo poder realizar ataques por vinculabilidad en conjuntos de datos generados a partir de los mismos datos y que compartan el mismo mecanismo de cifrado. En este caso concreto definimos el dominio 'APP1', como ejemplo de un nombre de una posible aplicación:

```
encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP1')
```

```
0      27fae45b3ce8a1b105d0
1      fe528d1da3b60a2bea4a
2      2efaadf53748c20e7325
3      7c4e863ae6dec29b79be
4      cb2e7c6661dac2228b4d
              ...
994    af545c82dd1ccd46ab9d
995    8c5acfb28674725164a2
996    84911b85ca1504816131
997    998035e804420ef27724
998    3183dd9039ba610d72b2
Name: creditcard, Length: 999, dtype: object
```

Podemos comprobar que al añadir el dominio los cifrados son diferentes para todos los registros, simplemente contando el número de filas que comparten el mismo código cifrado (0 registros):

```
sum(
    encodeBLAKE2B(df, 'creditcard', size=10) ==
    encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP1')
)
0
```

Aunque seguimos ante un caso de cifrado reversible determinista siempre que se reutilice el domino, es decir, al igual que en el ejemplo con SHA256, si comprobamos los cifrados generados en dos llamadas diferentes vemos que devuelve la longitud del del conjunto (todos son iguales):

```
sum(
    encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP1') ==
    encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP1')
)
999
```

Si bien nos permite tener tantos dominios como casos de uso deseemos definir, por ejemplo, se podría generar un nuevo conjunto para una segunda aplicación llamada 'APP2'. Al añadir el nuevo dominio los cifrados son diferentes para todos los registros en ambos casos de uso.

```
sum(
    encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP1') ==
    encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP2')
)
0
```

No es necesario que el valor del dominio sea privado, aunque es recomendable para mejorar la resistencia frente a ataques de fuerza bruta. Estos nombres únicamente tienen por objetivo definir un **caso de uso para un proceso de anonimización**, dado que los objetivos de cada caso son diferentes, así como los requisitos de garantías, utilidad, variables esenciales, etc.

4.9.4 Cifrado Blake2b con sal (*salt*)

Para evitar el problema del determinismo de los resultados, y por tanto posibles ataques por fuerza bruta, es recomendable aplicar un valor de sal (salt) que **normalmente es aleatorio y se destruye después de su aplicación**, aunque depende del caso y del nivel de garantías requerido.

El valor de salt se puede generar fácilmente con una función aleatoria estándar:

```
rndSalt = os.urandom(blake2b.SALT_SIZE)

encodeBLAKE2B(df, 'creditcard', size=10, domain=b'APP1', salt=rndSalt)
```

```
0      e8c7f1ffcd75dc4afa5f
1      0032f88cefa21c0a6b27
2      3e8a5ecd5c432bb3f694
3      58f262ab4d61d1e26749
4      969a5e346730aea5a8b7
              ...
994    a9c6aad6a0f810ec3a52
995    6e549322708e834d59d7
996    37d75cac29891586df89
997    a15c3225e01f5f5be626
998    7604d6799a04ed9f932b
Name: creditcard, Length: 999, dtype: object
```

Podemos comprobar que al añadir el valor de salt los cifrados son diferentes para todos los registros, simplemente contando el número de filas que comparten el mismo código cifrado (o registros):

```
sum(
    encodeBLAKE2B(df, 'creditcard', size=10) ==
    encodeBLAKE2B(df, 'creditcard', size=10, salt=rndSalt)
)
0
```

Si se almacena la sal y un atacante consigue dicho valor, podría aplicar nuevamente ataques por fuerza bruta, por lo que es recomendable destruir el valor o simplemente no almacenarlo después de su uso (variable temporal). Si comprobamos los cifrados generados en dos llamadas diferentes con el mismo valor de salt vemos que devuelve la longitud del del conjunto (todos son iguales):

```
sum(
    encodeBLAKE2B(df, 'creditcard', size=10, salt=rndSalt) ==
    encodeBLAKE2B(df, 'creditcard', size=10, salt=rndSalt)
)
999
```

4.10 RESULTADO FINAL DE ANONIMIZACIÓN

Como hemos visto, existen diferentes estrategias y algoritmos que permiten conseguir la anonimización de un conjunto de datos, siendo un **proceso complejo y muy vinculado a la finalidad del caso de uso del análisis y el nivel de utilidad y garantías requerido**.

En este caso práctico hemos visto algunos procesos sencillos, sin buscar ser un análisis exhaustivo, siendo más un conjunto de ejemplos iterativos de aplicación de diferentes técnicas. Existen multitud de librerías para diferentes entornos que permiten **realizar un análisis automático e incluso definir reglas en función de los requerimientos que definamos**, si bien quedan fuera del alcance de este informe por su complejidad y extensión.

A modo de resumen vamos a presentar a continuación el resultado de dos variantes que hemos ido desarrollando en detalle en los apartados anteriores.

4.10.1 Conjunto anonimizado con Anonimato K=5 y código postal en 2 grupos

La primera variante incluye todas las variables, agrupando el código postal en 2 secciones y definiendo un dominio específico y un valor salt aleatorio sin custodia para la tarjeta de crédito:

```
df.creditcard = encodeBLAKE2B(
df, 'creditcard', size=10,
domain = b'Dataset con ZIP',
salt = os.urandom(blake2b.SALT_SIZE)
)
```

	creditcard	zipcode	age	gender	salary
0	eed20f5a07191ca9176d	55000-55499	senior	Male	high
1	753321061a16b87373b3	55000-55499	medium	Female	high
2	4f8b68cd254c19d5bb7c	55500-55999	medium	Female	high
3	6be63a3cdc9c874c89d9	55500-55999	senior	Male	high
4	61d230518d57f97363d5	55500-55999	senior	Male	high
...	...	...	...	...	...
994	2f150589486f019c7ad1	55500-55999	senior	Female	medium
995	eacc1fba049a3838d000	55000-55499	junior	Female	high
996	37d283de0c8cd3ee1e65	55500-55999	junior	Female	high
997	9dd35709d738f6d2445c	55500-55999	senior	Male	high
998	48ad384275a2b87fbd54	55500-55999	medium	Male	high

999 rows × 5 columns

En este caso se puede comprobar que cumple Anonimato-K para un valor máximo de K=5, aumentando progresivamente el valor de K, siendo este el máximo nivel alcanzado mediante los pasos definidos:

```
isKAnonymized(df, 5, queryKAnonymized_salaryAgeGrouped)
True
```

4.10.2 Conjunto anonimizado con Anonimato K=11, sin código postal

En la segunda versión, mejoramos significativamente el anonimato hasta K=11, pero el conjunto de datos pierde bastante utilidad al eliminar el código postal. Si bien, comparativamente, tampoco implica una gran diferencia con respecto a la anterior solución, dado que ya estaba agrupado en 2 secciones.

A nivel de cifrado se aplica el mismo criterio, pero definiendo un dominio diferente, de forma que, si un atacante tuviera acceso a ambas versiones, este tendría más dificultades para realizar ataques por vinculación para descubrir el código postal mediante la comparación de los códigos cifrados de la tarjeta entre ambas versiones del conjunto.

```
df_nozip.creditcard = encodeBLAKE2B(
df, 'creditcard', size=10,
domain = b'Dataset sin ZIP',
salt = os.urandom(blake2b.SALT_SIZE)
)
```

	creditcard	age	gender	salary
0	9262d1d0943f385ccc02	senior	Male	high
1	1fb282844a530f1b45bc	medium	Female	high
2	74fcac442a3bfd0fc1fc	medium	Female	high
3	1d38608a7a679e8ebf15	senior	Male	high
4	541f5c75f15cef37eecb	senior	Male	high
...	...	...	...	...
994	3767a4e8f27445b2b7e5	senior	Female	medium
995	bdbf873c087f1c8b285d	junior	Female	high
996	41dd7e84e6fc7055fd8d	junior	Female	high
997	b4a0340681e2ccd32f81	senior	Male	high
998	74ff15f379b61726d5e4	medium	Male	high

999 rows × 4 columns

Comprobamos con éxito que se puede llegar hasta Anonimato-K con K=11, aumentando progresivamente el valor de K, siendo este el máximo nivel alcanzado mediante los pasos definidos:

```
isKAnonymized(df, 11, queryKAnonymized_noZip)
True
```

"Si quieres profundizar más en el campo de la anonimización, la Agencia Española de Protección de Datos (AEPD) ha traducido la Guía básica de Anonimización de la Autoridad de Protección de Datos de Singapur. La guía se complementa con una herramienta gratuita de anonimización de datos, que la AEPD pone a disposición de las organizaciones".

5. CONCLUSIONES

Las soluciones de anonimización de datos están en constante evolución, siendo un problema especialmente complejo de abordar, dado que no es posible garantizar una anonimización absoluta. En este sentido, la anonimización se gestiona como un **proceso de análisis de riesgos**, donde se busca ofrecer un **equilibrio entre las garantías de privacidad del conjunto anonimizado y la utilidad que tiene** para una tarea concreta, donde el grado de anonimato se puede medir en una escala, definida en función de las técnicas aplicadas y las garantías que ofrecen en cada caso concreto.

Un aspecto crítico es que **ninguna técnica aplicada de forma aislada es suficiente para aportar garantías frente a los tres tipos de ataques principales**: singularización (identificar a un individuo concreto), vinculación (asociar uno o varios registros de un individuo en uno o varios conjuntos de datos) e inferencia (probabilidad de deducir atributos de un individuo a partir de otros atributos).

En general, existe un **alto grado de desconocimiento sobre las técnicas disponibles y las garantías que ofrecen**, lo cual hace que existan múltiples fallos comunes en estos procesos. El **problema más habitual es considerar que el cifrado o la seudonimización son técnicas adecuadas**, cuando ni siquiera se podrían considerar técnicas de anonimización, ni los conjuntos resultantes estarían exentos de la aplicación del RGPD, dado que siguen siendo considerados datos de carácter personal. Si bien estas técnicas sí **mejoran la protección frente a la vinculabilidad**, por la ofuscación de datos sensibles.

Dado que además **no existe ninguna prescripción oficial** con respecto al uso de ninguna técnica concreta, **lo más recomendable es aplicar una combinación adecuada**, que incluya al menos técnicas de aleatorización, para enmascarar la correlación de valores con individuos concretos, y de generalización, para alterar escalas u órdenes de magnitud. **La aleatorización es muy útil para minimizar los riesgos de vinculabilidad e inferencia**, que al ser **combinada con técnicas de generalización reduce además el riesgo de singularización drásticamente**.

Aun así, es muy importante abordar cualquier proceso de anonimización de forma adecuada, entendiendo correctamente las **implicaciones y limitaciones** que tiene. En concreto, existen siete principios básicos que es recomendable tener en cuenta: **proactivo** (diseñar el proceso desde sus etapas iniciales), **privacidad por defecto** (descartar todos los atributos que no sean relevantes), **objetivo** (gestión de riesgos inherentes), **funcional** (acotada a un caso de uso concreto), integral (monitorización y auditorías de uso), **informativo** (capacitación de involucrados) y **atómico** (división de responsabilidades).

La principal tarea es definir un esquema adecuado basado en los tres niveles de identificación de personas: microdatos, identificadores indirectos y datos sensibles (**principio de proactividad**), donde se asigne un valor cuantitativo a cada una de las variables. Esta escala debe ser conocida por todo el personal implicado (**principio de información**) y es crítico para la Evaluación de Impacto en la Protección de los Datos Personales (EIPD).

Para aquellos casos donde las garantías no son suficientes o el proceso no es viable, existen iniciativas alternativas a la anonimización, como son las salas seguras o ***Safe Reading Rooms***, que permiten acceder a datos sensibles mediante un control de acceso muy restrictivo, sin posibilidad de uso de dispositivos externos y sin conexión a Internet.

6. REFERENCIAS

- Unión Europea. Directiva (UE) 2019/1024 del Parlamento Europeo y del Consejo del 20 de junio de 2019 relativa a los datos abiertos y la reutilización de la información del sector público. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32019L1024&from=ES

- Unión Europea. Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos) [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04?locale=es

- Grupo de trabajo sobre protección de datos del artículo 29. Dictamen 05/2014 sobre técnicas de anonimización adoptado el 10 de abril de 2014 por el Grupo de Trabajo creado en el artículo 29 de la Directiva 95/46/CE. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.aepd.es/sites/default/files/2019-12/wp216-es.pdf

- European Data Protection board. Guidelines 03/2020 on the processing of data concerning health for the purpose of scientific research in the context of the COVID-19 outbreak [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-032020-processing-data-concerning-health-purpose_en

- Gobierno de España. Ley Orgánica 3/2018, del 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673

- Gobierno de España. Carta de derechos digitales. [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.lamoncloa.gob.es/presidente/actividades/Documents/2021/140721-Carta_Derechos_Digitales_RedEs.pdf

- Agencia Española de Protección de Datos. Orientaciones sobre la protección de datos en la reutilización de la información del sector público [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://datos.gob.es/es/documentacion/orientaciones-sobre-la-proteccion-de-datos-en-la-reutilizacion-de-la-informacion-del

- Agencia Española de Protección de Datos. Guía de Privacidad desde el Diseño. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.aepd.es/sites/default/files/2019-11/guia-privacidad-desde-diseno.pdf

- Agencia Española de Protección de Datos. Guía de orientaciones y garantías en los procedimientos de anonimización de datos personales de la AEPD del 13 de octubre de 2016. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://datos.gob.es/es/documentacion/orientaciones-y-garantias-en-los-procedimientos-de-anonimizacion-de-datos-personales

- Agencia Española de Protección de Datos. Gestión del riesgo y evaluación de impacto en tratamientos de datos personales. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.aepd.es/sites/default/files/2019-09/guia-evaluaciones-de-impacto-rgpd.pdf

- Agencia Española de Protección de Datos. Cifrado y Privacidad III: Cifrado Homomórfico. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.aepd.es/es/prensa-y-comunicacion/blog/cifrado-privacidad-iii-cifrado-homomorfico

- Real Casa de la Moneda. Sellado de tiempo Cualificado. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.cert.fnmt.es/catalogo-de-servicios/sellado-de-tiempo

- Agencia Española de Protección de Datos. Malentendidos relacionados con la anonimización. [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://www.aepd.es/es/documento/10-malentendidos-anonimizacion.pdf

- Massachusetts Institute of Technology (MIT). The beginning of the end for encryption schemes? [en línea] [fecha de consulta: 16 septiembre 2022] Disponible en: https://news.mit.edu/2016/quantum-computer-end-encryption-schemes-0303

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

red.es

Oficina del Dato

Iniciativa aporta

datos.gob.es