fundar

Guía práctica para la protección de datos

Área de Datos

Junio 2022

Paula Luvini

Área de
Datos

Junio 2022

Guía práctica para la protección de datos

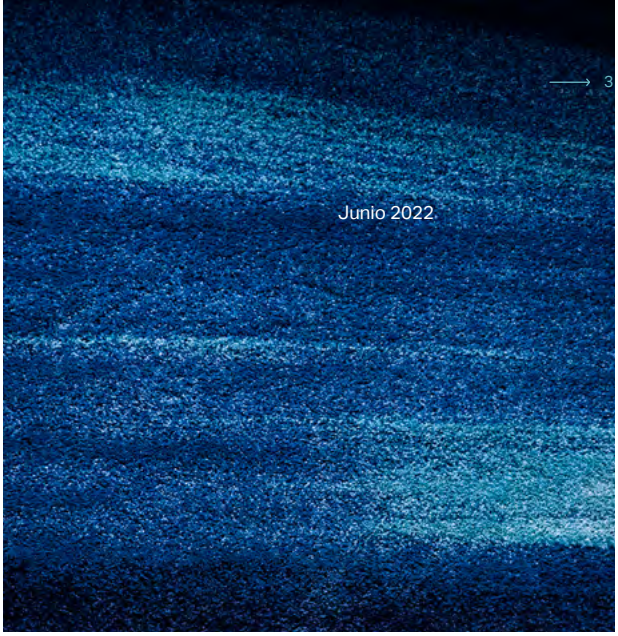
El objetivo de este documento es revisar algunas consideraciones prácticas a tener en cuenta al realizar intercambios de datos o publicar datos que contienen información de carácter sensible. Es una guía práctica de aplicación de técnicas presentadas conceptualmente con anterioridad, y propone el uso de herramientas ejemplificadas en el lenguaje Python. La presente guía no intenta ser una solución completamente abarcativa a problemas de anonimización, sino presentar instrumentos específicos que pueden ser útiles a quien necesite publicar o compartir datos, y desee usar herramientas de licencia libre y programables capaces de adaptarse a necesidades concretas del caso a resolver. Están destinadas tanto a aquellas personas que trabajen con datos personales —registros administrativos de alguna dependencia del Estado, por ejemplo— como para quienes quieran enriquecerse con el intercambio de datos entre organizaciones.

En la actualidad, recopilar información personal y de usuarios es algo común que atraviesa tanto a empresas privadas como a organismos del Estado. Más allá de la información específica que cada base de datos provea, la recolección a nivel de individuo suele estar acompañada de muchos datos privados o personales. Por ello, para que la población, terceros, o miembros de la organización accedan a estos datos, es necesario llevar a cabo un proceso de anonimización, de manera de preservar la identidad y privacidad del individuo.

Sentencias anónimas

Existen varios ejemplos de aplicación de estas metodologías que redundan en disponibilizar información que contribuya al bienestar ciudadano. Tal es el caso del algoritmo[1] que desarrollaron el Juzgado Penal Contravencional y de Faltas N°10 (Juzgado PCyF N°10) de la Ciudad Autónoma de Buenos Aires (CABA) y la cooperativa Cambá. Allí desarrollaron una herramienta para anonimizar de manera automática las sentencias del juzgado bajo la supervisión humana, con el fin de hacerlas públicas y disponibles virtualmente. El desarrollo de ese anonimizador también derivó en que los autores se dieran cuenta que podían utilizar la metodología para incorporar la perspectiva de género, convirtiendo la información relativa a violencia de género en un conjunto de datos tabulados, ordenados y sistematizados; de esta manera, se facilitó su extracción y posterior análisis.

1 Hassel Fallas y Claudia A. Contreras, Algoritmos para integrar la perspectiva de Género en la administración de la Justicia (2022).



1 →

Consideraciones a la hora de anonimizar

Preservar la identidad de las personas que conforman una base de datos es una práctica que se debe tener siempre presente, porque la preservación de la identidad es un derecho del ciudadano y una obligación de quien dispone de la información. Ciertamente no todos los intercambios de información conllevan el mismo riesgo: no es lo mismo hacer pública una base de datos que compartirla entre Secretarías de un mismo Ministerio.

A la hora de realizar un trabajo de protección de datos nos encontramos siempre con una elección: lograr un mayor grado de anonimización en los datos o llevar a cabo análisis más enriquecedores con datos de mayor granularidad. Es fundamental encontrar un equilibrio entre ambas alternativas. Si vamos a realizar un análisis donde las características personales del individuo son relevantes, y estas son eliminadas de la base de datos, enmascaradas o agrupadas de alguna manera, puede que lo que se quería analizar en primera instancia ya no esté disponible. De esta manera, lo que ganamos en seguridad lo estamos resignando en calidad de análisis.

Por otro lado, sabemos que en el caso general no es posible anonimizar una base real para hacerla completamente inmune si existen datos externos que puedan consultarse y cruzarse[2]. Una alternativa posible es anonimizar por agrupación, es decir "juntar" individuos en grupos y considerar solo datos agregados para compartir. Esto no siempre es una buena solución porque hay muchos casos en que la agregación por sí sola no logra un buen nivel de privacidad; y, por otro lado, puede implicar la pérdida de mucha información, produciéndose un desequilibrio en la elección entre anonimización y análisis.

Es esperable que miembros de la propia organización o terceros bajo ciertos acuerdos no lleven a cabo malos usos de los datos. Sin embargo, aun en estos casos debe haber un proceso mínimo de anonimización para preservar la vida privada de las personas; no importa si los datos van a tener un uso interno o no van a ver la luz pública. De nuevo, esto último se debe hacer en pos de buscar un equilibrio entre la seguridad de los datos personales y la riqueza de contar con algunas variables para incluir en modelos o procesos internos.

A continuación, se detallan algunos pasos prácticos para identificar riesgos de identificación o de acceso a datos sensibles y algunas herramientas que se pueden utilizar si se necesita protegerlos. También se puede acceder a los elementos de código ejemplificados en la guía a través del siguiente enlace a Github.

2 Como por ejemplo el caso de la base de Taxis de Nueva York

2 →

¿Cómo lograr una anonimización efectiva?

Una metodología posible para lograr una anonimización efectiva consta de 4 pasos:

1 → Identificar datos

¿Con qué otras bases se puede cruzar la información que se intenta proteger? Se deben analizar todas las variables que tienen en la base y pensar si estas pueden cruzarse con alguna otra fuente de datos externa que permita la reidentificación de los individuos. Si alguna de estas variables es sensible de ser cruzada entonces tenemos que, de alguna manera, enmascararla o encriptarla.

Por ejemplo, si contamos con una base de datos con individuos beneficiarios de algún programa del Estado y quitamos datos personales tales como el nombre, DNI y CUIT, pero dejamos otros como los de geolocalización, en algunos casos con baja densidad poblacional será muy fácil identificar a esos ciudadanos cruzando con otras bases de datos geográficas. Lo mismo puede pasar si enmascaramos parcialmente alguna variable como el DNI o la fecha de nacimiento: hay una mayor propensión a que haya otros datos disponibles que nos permitan reconstruir a esas personas.

2 → Identificar riesgos

Esto es importante para priorizar o no la encriptación de los datos y va a depender de los escenarios de uso. Si el riesgo de que reidentifiquen a los ciudadanos es alto y con ello se vulnerará su identidad o tendrá alguna implicancia en su vida, entonces se tiene que evitar que esto suceda modificando y anonimizando la base de datos. Este riesgo va a depender de cómo se utilicen los datos y si estos van a ser públicos o no. Si tomamos el ejemplo anterior, una base de datos con individuos que reciben algún tipo de beneficio del Estado, debemos pensar qué otras bases podrían cruzarse con esta; y, si eso efectivamente sucede, debemos considerar las consecuencias para estas personas y cómo se podrían ver perjudicadas

→

3 → Identificar soluciones

Además de eliminar columnas innecesarias, aquellas incluidas pueden atravesar un proceso de *hashing* y encriptado o de remoción de datos personales. Por ejemplo, remover todos los datos personales del listado de beneficiarios es mandatorio, pero tratar los datos geográficos de alguna manera también lo es. Una buena alternativa podría ser encriptar el código postal o mostrar esta información en agrupaciones mayores de manera de evitar una granularidad que permita la identificación.

4 → Identificar ataques y problemas

Teniendo en cuenta que los ataques pueden cambiar con el paso del tiempo es necesario revisar las soluciones pensando en estos riesgos. Esto involucra pensar en el trabajo de los ejemplos anteriores y reiterarlo con el paso del tiempo, considerando nuevos datos publicados que constituyan un nuevo riesgo de reidentificación.

En Anónimos pero no tanto: cómo hacer una gestión de datos eficiente sin poner en riesgo la privacidad (Yankelevich, 2021) se muestra una versión más completa de este método. Asimismo, se presentan algunos ejemplos de casos donde la anonimización realizada no fue suficiente para proteger los datos.

Antes de poner en práctica cualquier metodología de anonimización, es fundamental que nos hagamos las preguntas correctas: ¿necesitamos todas las variables que estamos almacenando? ¿Las variables que guardamos son todas necesarias para llevar a cabo el posterior análisis o tenemos algunas que son redundantes? En aquellos casos en que ciertas variables no aporten información relevante para el análisis que se va a hacer entonces no debemos recolectarlas ni intercambiarlas. Como se puede ver claramente, no son preguntas que requieran un saber científico o técnico sobre datos: son preguntas que apuntan a consideraciones morales y éticas. Son preguntas que cualquier persona debería estar en condiciones de responder, no solo la ciencia de datos. Para más información relevante sobre este tema, recomendamos leer Data Minimization: Key To Protecting Privacy And Reducing Harm (Acces Now, 2021).

3

Librerías para aplicar en Python

Una vez que se han identificado conceptualmente qué variables son las que se deben enmascarar o proteger, **es hora de procesar la base de datos** con estos lineamientos para publicarla o intercambiarla. En Python hay algunos paquetes específicos que son útiles para estos casos. **Mencionamos dos a continuación:**

Scrubadub

Esta librería permite quitar información personal de textos. Esto incluye nombres, direcciones de correo electrónico, urls, usuarios de Twitter o Instagram, fechas de nacimiento, entre otros. Es útil para tratar aquellos campos donde tengamos un texto con partes a enmascarar.

Con la función de Scrubber se van a identificar dentro del texto ingresado aquellas partes que contienen información sensible, y con el denominado *detector* se va a clasificar qué tipo de objeto es: un nombre, un usuario de una red social, un e-mail, etc. Por último, estos objetos identificados por el detector van a ser reemplazados por un *token* y el texto final va a quedar limpio de cualquier información personal que podía encontrarse previamente.

Son varios los elementos que pueden cambiarse dentro del Scrubber y que van a alterar los resultados. Por ejemplo, hay tres opciones de detectores para utilizar que no están incluidos por *default*, pero que podemos cambiar para mejorar la performance.

Con el Spacy v3 detector podemos utilizar modelos preentrenados en español: hay varias opciones dependiendo del modelo, el tamaño que tienen y cuánto tiempo de procesamiento conllevan. Hay cuatro opciones listadas aquí que varían en el modelo utilizado y en su tamaño. Hay tres opciones distintas correspondientes a la capa preentrenada para español de Tok2Vec cuya diferencia está en el tamaño, siendo que cuanto mayor sea más tiempo va a llevar en correr pero con mejores resultados. También hay un modelo que utiliza un transformer en español (BERT) con muy buenas métricas.

Respecto a los otros dos detectores que tenemos, uno de ellos es el de Stanford. Según la web de Scrubadub, este es el mejor pero tiene bastantes complicaciones a la hora de instalarlo por la cantidad de requerimientos. Por otro lado, Spacy v3 es casi tan bueno como el Stanford y mucho más fácil de instalar. El otro detector es el TextBlob, que tiene una tasa de falsos positivos muy alta según el sitio web.

Tomando como ejemplo el detector Spacy y el modelo de Tok2Vec mediano, para enmascarar los textos debemos hacer lo siguiente:

```
### Importar librerías y paquetes
necesarios para usar scrubadub y spacy

import scrubadub, scrubadub_spacy
import spacy_transformers

### Se inicializa el scrubber con
la localización del país para que
reconozca características específicas,
como números de teléfono.
scrubber = scrubadub.
Scrubber(locale="es_AR")

### Se agrega el detector elegido

scrubber.add_detector(scrubadub_spacy.
detectors.SpacyEntityDetector(model =
"es_core_news_md"))

### Se limpia el texto elegido
scrubber.clean("Mi nombre es Juan
González")

---

'Mi nombre es {{NAME}}'
```

Otras herramientas

Así como contamos con librerías de código libre para anonimizar bases de datos, también existen software pagos para anonimizar bases de datos. Cada uno tiene una funcionalidad específica y distintas prestaciones que pueden ser evaluadas para conocer si se ajustan mejor o peor a las necesidades de enmascaramiento. Algunos ejemplos de posibles herramientas de software en este cuadro comparativo y en esta nota.

Hashlib

Es una librería de Python que permite encriptar datos con una serie de distintos algoritmos, detallados en el link. Los algoritmos de hashing son funciones matemáticas usadas para transformar datos en valores o códigos *hash* y que sea imposible identificar el contenido original de ese dato. El *hashing* es una operación unidireccional en el que los valores ingresados son transformados a un código hash de longitud finita que no puede revertirse.

El *hashing* se utiliza en muchos procedimientos que requieran de niveles altos de seguridad, como firmas digitales, autenticaciones y protección de contraseñas. Por ejemplo, las contraseñas de Google no son guardadas directamente sino que se guardan los códigos *hash* de las mismas para evitar que otras personas puedan leer las contraseñas de los usuarios o de alguna manera desencriptar las mismas. Este ejemplo es uno puntual y familiar para muchos, en verdad son muchas las aplicaciones que utilizan *hashing* para resguardar datos, documentos y distintos tipos de archivos.

En la librería Hashlib contamos con varios modelos distintos a utilizar. Están incluidos los Secure Hash Algorithm (SHA): SHA1, SHA224, SHA256, SHA384, and SHA512, y por último BLAKE2. BLAKE2 es más rápido que los SHA e incluso que MD5 (este último está obsoleto y se considera menos fuerte ante ataques por lo que no es recomendable). Respecto a los SHA, al parecer SHA-2 es bastante seguro y más rápido que SHA-3, por lo que en muchos casos el SHA-2 es aceptable. Al respecto hay algunos posts interesantes para tener en cuenta:

1. How to Use Hashing Algorithms in Python using hashlib.
2. hashlib – Cifrar con los algoritmos MD5 y SHA
3. hashlib – Cryptographic hashes and message digests

A continuación se ejemplifica el *hashing* de un conjunto de datos ficticios.

```
### Se crea un dataframe de ejemplo con las jurisdicciones
dic = {'Jurisdicción':['Buenos Aires', 'Córdoba', 'San Juan', 'Buenos Aires', 'Buenos Aires', 'Córdoba', 'Entre Ríos']}
df = pd.DataFrame(data = dic)
df
```

```
  Jurisdicción
0      Buenos Aires
1      Córdoba
2      San Juan
3      Buenos Aires
4      Buenos Aires
5      Córdoba
6      Entre Ríos
```

```
### Se importa la librería de hashing
import hashlib

### Como ejemplo se utiliza un hashing con sha256.
df['Jurisdicción'].apply(lambda x: x.encode()).apply(hashlib.sha256).apply(lambda x: x.hexdigest())
```

```
0    abae768240354770b732e87de24a76aa-3123c0abbe7f4ebe6110c090d30eae98
1    263bcb39796e48285ae952fb25ddb-5550242fc8d3bafae8cc985e539f8053f6e
2    0c754c271376cd8bdaf8e93edba2c6e-034afa14925d617aab4f4a3be48f4525d
3    abae768240354770b732e87de24a76aa-3123c0abbe7f4ebe6110c090d30eae98
4    abae768240354770b732e87de24a76aa-3123c0abbe7f4ebe6110c090d30eae98
5    263bcb39796e48285ae952fb25ddb-5550242fc8d3bafae8cc985e539f8053f6e
6    b50525cee17c74a8416f580e965be-92058b7792a3ebf4614477bdd358b3cd9d4

Name: Jurisdicción, dtype: object
```

Conclusiones

El presente documento constituye una guía práctica para anonimizar un conjunto de datos sin que pierdan su granularidad y evitando posibles riesgos de identificación. Para tal propósito, se identificaron pasos a seguir para anonimizar un conjunto de datos y se presentaron una serie de herramientas con el software libre de Python.

Como se menciona, siempre es importante tener en cuenta el escenario en que van a ser utilizados los datos: para poder evaluar los riesgos de que sean compartidos y con ello la rigurosidad del proceso. De todas maneras, aun cuando la base no sea compartida públicamente, también se debe llevar a cabo este ejercicio de preservación de datos sensibles: es una práctica deseable sin importar el nivel de confidencialidad del ámbito. En esos casos particulares, deben considerarse las ganancias y pérdidas por las cuales anonimizar de manera más estricta trae como consecuencia la pérdida de información y con ello perjudica el análisis posterior. Esto mismo no quita importancia al punto de que, más allá de que vamos a tratar la información confidencialmente, debemos no vulnerar la identidad de los involucrados.

Por último, insistimos en la necesidad de focalizarnos en tener la información que necesitamos para el análisis y *no acumular información de más*. Es importante que se reflexione en el por qué estamos guardando esta información: ¿son necesarias las variables que se están guardando? ¿Van a ser utilizadas de alguna manera? Estas preguntas son fundamentales y se deben incluir desde el diseño inicial de la recolección.

Acerca de la autora

Paula Luvini

Científica de datos del Área de Datos. Licenciada en Economía por la UBA y maestranda en Ciencia de Datos en la UdeSA. Trabajó en el sector público y en el privado y como docente.

Dirección ejecutiva: Martín Reydó

Coordinación editorial: Gonzalo Fernández Rozas

Diseño: Jimena Zeitune

Fundar es un centro de estudios y diseño de políticas públicas que promueve una agenda de desarrollo sustentable e inclusivo para la Argentina. Para enriquecer el debate público es necesario tener un debate interno: por ello lo promovemos en el proceso de elaboración de cualquiera de nuestros documentos. Confiamos en que cada trabajo que publicamos expresa algo de lo que deseamos proyectar y construir para nuestro país. Fundar no es un logo: es una firma.

Esta obra se encuentra sujeta a una licencia Creative Commons 4.0 Atribución-NoComercial-SinDerivadas Licencia Pública Internacional (CC-BY-NC-ND 4.0). Queremos que nuestros trabajos lleguen a la mayor cantidad de personas en cualquier medio o formato, por eso celebramos su uso y difusión sin fines comerciales.

En Fundar creemos que el lenguaje es un territorio de disputa política y cultural. Por ello, sugerimos que se tengan en cuenta algunos recursos para evitar sesgos excluyentes en el discurso. No imponemos ningún uso en particular ni establecemos ninguna actitud normativa. Entendemos que el lenguaje inclusivo es una forma de ampliar el repertorio lingüístico, es decir una herramienta para que cada persona encuentre la forma más adecuada de expresar sus ideas.

Modo de citar

Luvini, Paula (2022). Guía práctica para la protección de datos. Buenos Aires: Fundar.

Disponible en https://www.fund.ar

fundar