Aplicación de privacidad diferencial con Python

Daniel López Álamo

Resumen– Este proyecto final recoge una investigación de conceptos relacionados con la privacidad de datos. Se profundiza en la recientemente conocida técnica de privacidad de datos y sus conceptos clave. Además se ponen en práctica el mecanismo de Laplace y el mecanismo exponencial como métodos para aplicar la privacidad diferencial, que garantiza la seguridad de los datos a la vez que preserva su utilidad. También, se expone un análisis práctico de los mecanismos aplicados y sus parámetros, mediante una API que simula consultas a una base de datos. Por último, se muestra un análisis de la seguridad del sistema presentado y se propone un método de prevención contra ataques a un sistema que aplica privacidad diferencial.

Palabras clave– Privacidad de datos, privacidad diferencial, mecanismo de Laplace, mecanismo exponencial, Python, bases de datos, teorema de la composición, epsilon, sensibilidad global, API.

Abstract– This final project is an investigation of concepts related to data privacy. It delves into the recently known differential data privacy and its key concepts. In addition, the Laplace mechanism and the exponential mechanism are implemented as methods to apply differential privacy, which guarantees data security while preserving its usefulness. Also, a practical analysis of the applied mechanisms and their parameters is presented, using an API that simulates queries to a database. Finally, a security analysis of the presented system is shown and a prevention method against differential privacy attacks is proposed.

Keywords– Data privacy, differential privacy, Laplace mechanism, exponential mechanism, Python, databases, composition theorem, epsilon, global sensibility, API.

1 Introducción

La privacidad puede ser definida como el ámbito personal de un individuo que se quiere mantener confidencial, a menudo las bases de datos de las aplicaciones pueden guardar datos de carácter personal o privado. Para protegerlas, existen técnicas de privacidad y alteración de datos.

Todas estas técnicas tienen como objetivo principal la anonimización y reidentificación de los datos, pero dada la complejidad de garantizar la privacidad absoluta de un sistema, se ha asumido la reidentificación como un riesgo asumido y gestionado. Debido a ello se establecen 3 vectores de riesgo de reidentificación, definidos por el Dictamen 05/2014[1] sobre técnicas de anonimización:

- **Singularización:** riesgo de extraer atributos que permitan identificar a un individuo.
- **Vinculabilidad:** riesgo de vincular al menos dos atributos al mismo individuo o grupo, en uno o varios conjuntos de datos.
- **Inferencia:** riesgo de deducir el valor de un atributo crítico a partir de otros atributos.

Como se puede observar en la Tabla 1, no todas las técnicas asumen el coste que tiene eliminar razonablemente la reidentificación. Técnicas como la privacidad diferencial ofrecen un riesgo de reidentificación muy disminuido, no obstante, esto no indica que sea el mejor método en todas las ocasiones.

La privacidad diferencial se interpreta como una técnica que permite realizar consultas en un sistema de datos,

• E-mail de contacto: daniel.lopezala@e-campus.uab.cat
• Mención realizada: Tecnologías de la Información
• Trabajo tutoritzado por: Guillermo Navarro Arias (DEIC)
• Curso 2021/22

	¿Existe riesgo de singularización?	¿Existe riesgo de vinculabilidad?	¿Existe riesgo de inferencia?
Seudonimización	Sí	Sí	Sí
Adición de ruido	Sí	Puede que no	Puede que no
Sustitución	Sí	Sí	Puede que no
Agregación y anonimato K	No	Sí	Sí
Diversidad l	No	Sí	Puede que no
Privacidad diferencial	Puede que no	Puede que no	Puede que no
Hash/Tokens	Sí	Sí	Puede que no

Fig. 1: Técnicas de privacidad[2]

garantizando tanto la utilidad como la seguridad de los mismos.

Es decir, los datos sensibles del sujeto quedan protegidos a la adición de ruido generado por mecanismos que aplican distribuciones matemáticas. Estos mecanismos han sido estudiados, mencionados y fundados en varios artículos e investigaciones desde sus orígenes.

2 Estado del arte

La privacidad diferencial comienza a aparecer como concepto por primera vez en un estudio realizado por Cynthia Dwork, Frank McSherry, Kobbi Nissim, Adam Smith, llamado "Calibrating Noise to Sensitivity in Private Data Analysis" en el año 2006 [3]. Durante casi una década, Cynthia Dwork, científica informática e investigadora de Microsoft, continúa realizando y participando en estudios que dan vida a la técnica de privacidad.

Más adelante en 2007, Frank McSherry y Kunal Talwar, redactaron un artículo que profundiza en los algoritmos de preservación de la privacidad y el diseño de mecanismos que aplican la privacidad diferencial. En este artículo, aparece por primera vez el concepto del mecanismo exponencial [4].

En 2014, al fin se exponen las bases y fundamentos de este método de privacidad, dando significado a la privacidad diferencial y mostrando el conjunto de mecanismos a aplicar que ofrecen una garantía matemática de la seguridad de los datos de un sistema [5].

Alrededor de 2020, la Oficina del Censo de EE.UU publicó un comunicado confirmando el uso de privacidad diferencial en sus sistemas de datos[6]. El objetivo principal del censo es aportar información sobre los habitantes y la economía de los EE.UU, afirmando que la privacidad diferencial es una técnica moderna que proporciona la protección y la utilidad de los datos adecuada para su labor .

A día de hoy, grandes empresas tecnológicas utilizan esta técnica para procesar una gran cantidad de datos sobre sus usuarios sin revelar de forma alguna la identidad de los mismos, lo que les permite identificar patrones de comportamiento, tendencias y realizar estudios que podrían impulsar el crecimiento de la empresa. Recientemente empresas como Apple y Google han confirmado el uso de esta técnica y también han revelado su funcionamiento compartiendo librerías de código y casos de test [7][8].

3 Objetivos

El objetivo principal de la propuesta de este trabajo de fin grado consistía en desarrollar métodos de aleatorización de datos en Python, inspirados en los métodos de adición de ruido de la librería sdcMicro en R.

Tras una reunión con el tutor, surgió otra propuesta más interesante y el objetivo principal se convierte en la implementación de métodos que apliquen privacidad diferencial a un sistema de datos.

A partir de este objetivo el trabajo se divide en dos partes, por un lado la recopilación de información sobre la privacidad diferencial, que conlleva la comprensión de la definición de privacidad diferencial, su funcionamiento, los parámetros necesarios para llevarla a cabo y los resultados que aporta .

Y por otro lado, el diseño e implementación de métodos que apliquen privacidad diferencial. Para ello, necesitamos comprender cómo funcionan los mecanismos de Laplace y exponencial, que se utilizan para calcular el ruido que se añade tanto a datos numéricos como a datos categóricos.

Por último, con el objetivo de analizar y evaluar los métodos aplicados, se ha diseñado una API que gestiona el acceso a la base de datos, que en este caso viene simulada por un archivo .csv.

4 Metodologia

Para alcanzar el objetivo establecido para este proyecto, se ha llevado a cabo una planificación compuesta por una metodología ágil Kanban. Esta metodología consiste en la gestión de tareas en una vista Tablero, donde se observa el estado actual de cada actividad. Para completar una actividad, esta debe experimentar todos los estados, que han sido previamente creados por el gestor del proyecto.

Para complementar esta metodología se ha escogido una estrategia de planificación de trabajo semanal, que se muestra en un diagrama de Gantt, donde se puede observar la evolución que debía seguir el proyecto con el paso de las semanas y las dependencias que entre las tareas del mismo.

Las herramientas que se han utilizado son:

- **ClickUp**, una aplicación de gestión de proyectos que permite la creación de un tablero donde se visualizan todas las tareas y sus estados actuales. Además contiene una vista calendario que permite controlar las fechas límite de las tareas [9].
- **LucidChart**, una herramienta online que permite diseñar cualquier tipo de diagrama [10].

5 Desarrollo

En este apartado se explica fase por fase el desarrollo que ha seguido el proyecto, la información recogida, las herra-

mientas utilizadas, los mecanismos y parámetros utilizados y por último la solución final.

5.1. Qué es la privacidad diferencial y cómo funciona

La privacidad diferencial se define formalmente de la siguiente manera [**?**]:

Una función K_f para una consulta f, proporciona ϵ-privacidad diferencial si para todas las bases de datos D_1 y D_2 que difieren en un solo elemento, y para todo $S \subseteq Rango(\mathrm{K}_f)$.

Uno de los métodos para proporcionar privacidad diferencial a un sistema es la aplicación de mecanismos aleatorios, basados en distribuciones matemáticas, que consisten en la adición de ruido al valor que se devuelve en la consulta realizada.

Para facilitar la comprensión del concepto vamos a exponer un caso teórico [**?**], en el que se dispone de una base de datos D, compuesta por datos de distintos individuos. Para acceder a esta información los usuarios pueden hacer consultas con los filtros que determinen necesarios, si un usuario quiere saber cuáles de los usuarios que tienen un salario superior a 250 se debe se realizará una consulta con salario mayor que 250. Estas consultas se pueden considerar como funciones que se aplican a la base de datos f, el resultado de aplicar la función D_f

La privacidad diferencial entra en juego cuando el número de registros entre las versiones de la base de datos es distinto en 1, supongamos que ahora existen dos versiones de la base de datos: D_1 y D_2.

Tabla 1 D_1

nombre	edad	salario
Ataúlfo	43	100
Sigerico	**15**	**300**
Walia	50	200
Teodoredo	51	100
Turismundo	16	500
Tedorico II	40	300
Eurico	14	700
Alarico II	49	300
Gesaleico	69	500
Amalarico	17	200

Tabla 2 D_2

nombre	edad	salario
Ataúlfo	43	100
Walia	50	200
Teodoredo	51	100
Turismundo	16	500
Tedorico II	40	300
Eurico	14	700
Alarico II	49	300
Gesaleico	69	500
Amalarico	17	200

Fig. 2: Versiones de la base de datos D

En este caso, al ejecutar la misma consulta en D_1 y D_2, obtendremos que $f(D_1) = 6$ y $f(D_2) = 5$. Aparentemente estos resultados no son ninguna amenaza y se puede pensar que no dan ningún tipo de información privada al usuario, pero realmente si se puede identificar a este usuario que ha sido desvinculado de la base de datos gracias a información externa.

Como hemos mencionado previamente, para conservar la utilidad de los datos y obtener un nivel de privacidad aceptable, se devuelve el valor original de la consulta con ruido añadido. De esta forma, al realizar la misma consulta en D1 y D2, obtenemos casi la misma información, ya que los datos han sido aleatorizados

Hasta ahora, podemos definir como mecanismo a la función que recibe la base de datos como parámetro y que devuelve un resultado (f). Por otro lado, un mecanismo aleatorio añade ruido aleatorio a la respuesta para preservar la privacidad.

5.2. Gestión de una base de datos con Pandas

Para gestionar y procesar los datos del sistema se ha hecho uso de la librería pandas de Python [12].

Esta librería permite cargar los datos de un archivo .csv en un Dataframe, una estructura de datos bidimensional similar a una tabla. Gracias a esta estructura se ha podido acceder a la información y modificarla de forma sencilla y eficaz. Además, esta librería también incluye funciones que permiten hacer cálculos masivos por columnas p.ej, sumatorios, medias, conteos, etc.

Para poner en práctica estos conocimientos, se ha desarrollado una base de datos de estudiantes de distintos grados en ingeniería. Se han generado 100 registros compuestos por nombre, edad, nota media y el grado que cursan, los datos de estos registros son completamente aleatorios. Como se ha mencionado antes, es importante que la segunda versión de la base de datos difiera en un registro y también hay que tener en cuenta que trabajar con datos acotados facilita la implementación de los mecanismos, más adelante se explica por qué.

Las funciones diseñadas para aplicar los mecanismos aleatorios son:

- ***get approved***: Esta función devuelve el número de estudiantes que tiene una nota media igual o superior a 5.
- ***get average qualification***: Esta función devuelve la media de notas de los estudiantes.
- ***get average age approved***: Esta función devuelve la media de edad de los estudiantes aprobados.
- ***get most approved career***: Esta función devuelve la carrera con mayor número de estudiantes aprobados.

5.3. Implementación de privacidad diferncial. Mecanismo de Laplace

El mecanismo de Laplace es uno de los más aplicados y más estudiados, que se aplica en datos de tipo numérico.

Consideramos D1 y D2 como dos versiones de la misma base de datos, mostradas a continuación. Se puede observar que la diferencia entre ambas bases de datos es de un registro, este diseño tan sencillo nos ayudará a entender la sensibilidad global de una función [13].

La sensibilidad global viene definida formalmente de la siguiente forma [14]: Siendo d un valor entero positivo, D una colección de conjuntos de datos formada por números reales y f una funcion que se aplica en D. La sensibilidad global de una función, es definida por:

$$\Delta f = max \| f(D_1) - f(D_2) \|$$

Fig. 3: Fórmula de la sensibilidad global

donde el máximo es sobre todos los pares de conjuntos de datos de D_1 y D_2 en D que se difieren como máximo en un elemento.

Para facilitar la comprensión de esta definición, supongamos que ahora existen dos versiones de la base de datos: D_1 y D_2.

Nombre	Nota
Maria Anders	0
Francisco Chang	10

Tabla de datos D_1

Nombre	Nota
Maria Anders	0

Tabla de datos D_2

Fig. 4: Versiones de la base de datos D

Para averiguar la sensibilidad global de la función es necesario calcular la máxima diferencia realizando la misma consulta en ambas versiones. Por ejemplo, para la consulta get approved, debemos considerar que el registro que falta en la segunda versión tiene una media igual o superior a 5, en este caso la diferencia seria $f(D_1)$ - $f(D_2)$ = 3 - 2 = 1. De lo contrario, si tuviera una nota inferior a 5 el cálculo de la diferencia sería igual a $f(D_1)$ - $f(D_2)$ = 2 - 2 = 0. Por lo tanto, la sensibilidad global de la función get approved sería $max(0, 1)$ = 1.

Para el caso de la función get average qualification el cálculo de la sensibilidad sería el siguiente, suponiendo una base de datos de dos registros para simplificar el cálculo, para calcular la máxima diferencia entre consultas es necesario el máximo valor de media que podamos obtener.

Por ello, como se ha comentado previamente, es importante utilizar datos acotados ya que facilitan mucho los cálculos de la sensibilidad, la notas de los estudiantes están dentro del intervalo [0,10].

Dadas las dos anteriores bases de datos vemos que la diferencia entre notas es máxima, si realizamos la consulta get average qualification para D1 y D2 la diferencia entre ellos es igual a $f(D_1)$ - $f(D_2)$ = 5 - 0 = 5.

Una vez comprendido esto, pasamos al parámetro epsilon, de ahora en adelante ϵ , que permite indicar el nivel de privacidad que se desea aplicar a la consulta. Este valor es escogido por el usuario que realiza la consulta, cuanto menor es el valor más privada es la consulta, y viceversa.

Una vez calculados estos parámetros, utilizamos la distribución de Laplace, considerada una densidad de probabilidad continua, para calcular el ruido que se añadirá al resultado de f. La densidad de probabilidad, simplemente describe la probabilidad de que una variable aleatoria tome determinado valor [15] y viene determinada por la siguiente función:

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b} \right)$$

Fig. 5: Cálculo de la densidad de probabilidad (Laplace)

Para la aplicación de esta función, se utiliza el valor predeterminado de μ (0), y el valor de la escala viene dado por b = sensibilidad global / ϵ.

Por último, el mecanismo devolverá el valor alterado de la conuslta, compuesto por el dato original + ruido.

5.4. Implementación de privacidad diferncial. Mecanismo Exponencial

El mecanismo aleatorio exponencial es utilizado para aplicar privacidad diferencial a consultas de datos categóricos, ya que estos no permiten realizar cálculos sobre ellos. Como la distribución de Laplace, la exponencial también es una distribución continua y su función de densidad es:

$$f_X(x) = \lambda e^{-\lambda x}$$

Fig. 6: Cálculo de la densidad de probabilidad (Exponencial)

El funcionamiento de este mecanismo consiste en la elección del "mejor" valor de un conjunto preservando la privacidad [16]. El analista se encarga de definir qué elemento es "mejor" y especifica el cálculo para determinar las puntuaciones de cada elemento del conjunto. Una vez calculadas las puntuaciones, la sensibilidad global y se haya determinado la ϵ, aplicamos la distribución exponencial para calcular las probabilidades que tiene cada elemento de ser escogido como resultado. Por último, es necesario normalizar los valores de las probabilidades para que estén entre 0 y 1.

Para poner en práctica este proceso, se ha implementado la función *get most approved career*. Los valores del conjunto están formados por los siguientes grados de ingeniería: Informática, Mecánica, Electrónica, Datos, Química, Telecomunicaciones y Aeronáutica.

En esta función, el "mejor" elemento se considera el que tenga una frecuencia mayor, por lo tanto, cuantas más veces aparezca en el conjunto "mejor" se considera el elemento.

Una vez completado el proceso y obtenida la lista de probabilidades, una función escoge de manera aleatoria un elemento considerando las ponderaciones calculadas previamente y se devuelve el valor escogido como respuesta de la consulta.

5.5. Diseño de una API para gestionar consultas a una base de datos

Con el objetivo de mostrar los resultados de una forma más visual, simulando una consulta real a una base de datos, se ha diseñado una API que nos permite hacer llamadas a un servidor local que, como se ha mencionado al principio del proyecto, permite gestionar las consultas y los datos (almacenados en un archivo .csv).

Este sistema permite solo la consulta de datos, no permite actualizarlos ni eliminarlos.

Para la creación de este sistema se ha utilizado el framework Flask [17], que permite crear una aplicación que y ejecutarla en un servidor local, que escuchará peticiones al puerto indicado, analizando el nombre de la ruta y devolviendo la respuesta.

El funcionamiento de la aplicación es el siguiente:

- Las funciones que se ejecutarán con las llamadas API son los mecanismos aleatorios de privacidad diferencial implementados y explicados previamente (Laplace y Exponencial).
- Para ejecutar dichas funciones es necesario crear rutas. Para declarar una es necesario añadir @app.route("nombre de la ruta") sobre la función que debe ejecutar la llamada API. A través de la ruta, también es posible pasar los parámetros necesarios para la ejecución de la función, esto hace posible que el usuario pueda escoger la que desee.
- Una vez realizada la llamada se ejecuta la función y se devuelve el resultado en formato JSON, permitiendo al usuario visualizarla.

5.6. Funcionamiento del sistema

Hasta ahora se ha visto como se reciben las llamadas API, como son procesadas y que respuesta devuelven. Unicamente falta averiguar como realizar las llamadas al servidor y donde recibir las respuestas. Para ello, se ha hecho uso de la aplicación Postman [18], que satisface las necesidades descritas previamente. Esta aplicación dispone de una sencilla interfaz, que nos permite describir el nombre de la ruta de la llamada que se quiere realizar y establecer los parametros necesarios para obtener su respuesta, una vez ejecutada la llamada el resultado se muestra por consola en formato JSON.

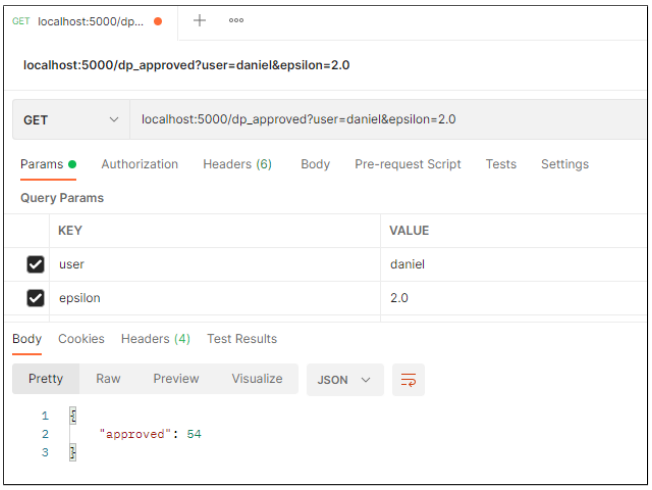

Fig. 7: Ejecución de la consulta dp approved

Este sistema de gestión de datos nos permite implementar funciones de control de usuarios, pudiendo controlar las consultas que hace cada usuario y que ϵ utilizan. Con esto sería posible, si se deseara, crear un sistema de control de acceso por usuario y contraseña.

6 RESULTADOS

En este último apartado se presentan los resultados obtenidos del desarrollo y la ejecución del proyecto.

Tras implementar los mecanismos aleatorizados, se han realizado una serie de pruebas y análisis de los resultados:

- La ejecución de las consultas devuelven resultados correctos y se puede confirmar que a mayor ϵ más se aproxima el resultado alterado al valor original, no es lo mismo realizar una consulta con una igual a 0.1 que con una ϵ igual a 10. En la figura anterior pode-

```
----Resultado original, get_approved()------
56
----Resultado dp_approved(), epsilon = 0.1----
64
----Resultado dp_approved(), epsilon = 10----
55
```

Fig. 8: Consulta get approved vs. dp approved

mos ver la diferencia entre la consulta sin privacidad diferencial (get approved) y la consulta con privacidad diferencial, tanto para ϵ igual a 0.1 como para ϵ igual a 10.

- Realizando un número muy elevado de consultas podemos observar que la media de los resultados se aproxima cada vez más al valor original, este hecho viene determinado por el teorema de la composición. Este teorema sostiene que si se consulta un mecanismo de privacidad epsilon-diferencial x veces, con una aleatorización independiente del mecanismo en cada consulta, el resultado sería ϵx-diferencialmente privado. Suponiendo que hay n mecanismos M_1, M_2, ..., M_n con sus correspondientes ϵ_1, ϵ_1, ..., ϵ_n privacidad diferencial, cualquier función $g(M_1, M_2, ..., M_n)$ tiene una ϵ igual al sumatorio de todas las ϵ de g.

  Resumidamente, al ejecutar la misma consulta n veces y realizando la media aritmetica de los resultados, obtenemos un valor ϵ-diferencialmente privado igual a una única ejecución de la consulta, con epsilon igual a la suma de las n epsilon utilizadas previamente.

  Este hecho puede resultar una amenaza a nuestro sistema, ya que ejecutar consultas de forma ilimitada puede vulnerar la privacidad de los datos que lo componen.

  Para eliminar esta vulnerabilidad, primero hay que entender que ϵ es un valor acumulativo, es decir, es prácticamente lo mismo hacer un consulta con $\epsilon = 5$ que cinco consultas con $\epsilon = 1$.

  Como antes hemos mencionado, el uso de una API para gestionar el sistema de datos nos facilita el control de las llamadas que realizan los usuarios sobre él. Para solucionar este problema se ha propuesto e implementado un archivo que incorpora un control de la ϵ acumulada por cada usuario en cada consulta, de esta manera un usuario nunca superará el valor global de ϵ establecido por el administrador del sistema, evitando así, perder la privacidad que se ha adquirido con los mecanismos aplicados.

  Para encontrar un buen valor de ϵ se ha ejecutado cada consulta con valores diferentes para ϵ, más concretamente 10 consultas por cada ϵ. Con el resultado de estas consultas se ha calculado la media de la diferencia entre el dato original y los generados por las consultas, con el objetivo de obtener información sobre qué ϵ otorga una diferencia mayor al resultado original y así conseguir la máxima privacidad.

  Este método se ha utilizado para cada consulta que aplica privacidad diferencial y el resultado de la ϵ global se ha implementado en una función de control de

consultas de la API, de manera que si un usuario supera o va superar este valor de ϵ global, se restringe la consulta.

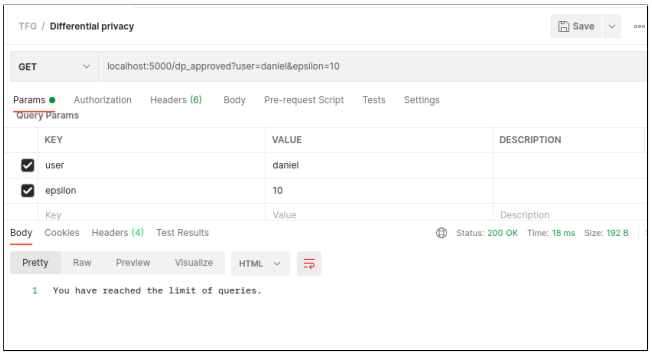

Fig. 9: Restricción de la ejecución de la consulta

7 Conclusión

Una vez finalizado el proyecto y analizados los resultados, se puede decir que los objetivos de este trabajo de Fin de Grado se han completado en el tiempo establecido y los resultados son los esperados.

La planificación se ha seguido semanalmente y no ha habido inconvenientes ni retrasos, las frecuentes reuniones con el tutor han sido determinantes para el desarrollo del proyecto. En general, considero haber aprendido mucho sobre un tema tan interesante como es la privacidad y además conocer una idea emergente como la privacidad diferencial.

En cuanto a dificultad, he considerado más difícil el hecho de comprender el concepto de este proyecto, la privacidad diferencial, que el hecho de implementarla. La mayoría de términos y propiedades han sido completamente nuevos para mí, las fuentes de información eran bastantes técnicas y he tenido que desglosar las definiciones en conceptos para facilitar su comprensión, además del ámbito matemático que tiene este proyecto del cual no soy especialista.

También es cierto que el sistema de datos sobre el que se ha trabajado no es real, al inicio del proyecto se consideró la idea de crear un sistema de datos pero tras unas reuniones con el tutor consideramos que se alejaba del objetivo principal del trabajo.

Aun así los resultados obtenidos han sido todo un éxito y me siento plenamente satisfecho.

Agradecimientos

Quiero agradecer este trabajo especialmente a mi tutor Guillermo Navarro, por toda la ayuda y motivación aportadas en todo momento durante el proyecto. También a mi familía y amigos por el soporte que me han brindado durante esta etapa.

Referencias

[1] Dictamen 05/2014 sobre técnicas de anonimización. Recuperado 21 de enero de 2022,de https://www.aepd.es/sites/default/files/2019-12/wp216-es.pdf

[2] datos.gob.es. (2021, 28 octubre). La importancia de la anonimización y la privacidad de datos. Recuperado 21 de enero de 2022, de https://datos.gob.es/es/blog/la-importancia-de-la-anonimizacion-y-la-privacidad-de-datos

[3] Dwork, C. (2006, marzo). Calibrating Noise to Sensitivity in Private Data Analysis. Microsoft Research. Recuperado 26 de enero de 2022, de https://www.microsoft.com/en-us/research/publication/calibrating-noise-to-sensitivity-in-private-data-analysis/

[4] McSherry, F. (2007, octubre). Mechanism Design via Differential Privacy. Microsoft Research. Recuperado 26 de enero de 2022, de https://www.microsoft.com/en-us/research/publication/mechanism-design-via-differential-privacy/

[5] Dwork, C. (2014, agosto). The Algorithmic Foundations of Differential Privacy. Microsoft Research. Recuperado 26 de enero de 2022, de https://www.microsoft.com/en-us/research/publication/algorithmic-foundations-differential-privacy/

[6] U.S. Census Bureau.(2021, julio). Differential Privacy and the 2020 Census. Recuperado 26 de enero de 2022, de https://www.census.gov/content/dam/Census/library/factsheets/2021/differential-privacy-and-the-2020-census.pdf

[7] Apple. (s. f.). Privacidad - Control. Apple (España). Recuperado 17 de enero de 2022, de https://www.apple.com/es/privacy/control/

[8] Google. (s. f.). Como Google anonimiza los datos - Privacidad y Condiciones - Google. Privacy Terms - Google. Recuperado 17 de enero de 2022, de https://policies.google.com/technologies/anonymization?hl=es

[9] Click Up. (s. f.). ClickUpTM — One app to replace them all. ClickUp. Recuperado 4 de octubre de 2021, de https://clickup.com/

[10] LucidChart. (s. f.). Intelligent Diagramming. Recuperado 6 de octubre de 2021, de https://www.lucidchart.com/pages/

[11] Navarro, G. (2021). Privacidad diferencial. Introducción. Recuperado 4 de octubre de 2021, de https://deic.uab.cat/ %7Egnavarro/gna-differential-privacy.html

[12] Pandas. (s. f.). pandas. Python Data Analysis Library. Recuperado 20 de septiembre de 2021, de https://pandas.pydata.org/

[13] Navarro, G. (2021). Privacidad diferencial. Mecanismo de Laplace. Recuperado 22 de septiembre de 2021, de https://deic.uab.cat/ %7Egnavarro/gna-dp-laplace.html

[14] Wikipedia Contributors. (2021, 22 diciembre). Differential privacy - Sensitivity. Wikipedia. Recuperado 30 de enero de 2022, de https://en.wikipedia.org/wiki/Differential_privacy Definition_of_ %CE %B5-differential_privacy

[15] Wikipedia contributors. (2021a, enero 9). Distribución de Laplace. Wikipedia. Recuperado 30 de enero de 2022, de https://es.wikipedia.org/wiki/Distribuci %C3 %B3n _de_Laplace

[16] P. Near, J. (2021). The Exponential Mechanism. Programming Differential Privacy. Recuperado 15 de enero de 2022, de https://programming-dp.com/notebooks/ch9.html: %7E:text=The % 20mechanism %20provides %20differential %20 privacy,not %20have %20the %20highest %20score.

[17] Flask. (s. f.). Flask. Flask Documentation (2.0.x). Recuperado 23 de diciembre de 2021, de https://flask.palletsprojects.com/en/2.0.x/

[18] Postman. (s. f.). Introduction. Postman Learning Center. Recuperado 28 de enero de 2021, de https://learning.postman.com/docs/getting-started/introduction/