UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ciencias Empresariales
Departamento de Sistemas de Información

Software para la aplicación de algoritmos que preservan la privacidad sobre las bases de datos estadísticas.

Diego Fernández Cáceres

David Quijón Flores

Profesor Guía: Patricio Galdames Sepúlveda

CONCEPCIÓN, 2018

Memoria presentada para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío para optar al Título de Ingeniero Civil en Informática.

Resumen

El presente proyecto de título consiste en la descripción de las primeras técnicas y/o algoritmos de anonimato desarrollados con el fin de proporcionar una solución al problema existente de la vulnerabilidad de la privacidad de individuos al liberar bases de datos con fines estadísticos. Para lograr esto, se estudian detalladamente las técnicas de k-anonimato, ℓ-diversidad y t-cercanía, proporcionando definiciones, características y comparaciones teóricas respecto a ellas.

Luego, se definen métricas para analizar las distintas técnicas respecto a la utilidad de la información que proporcionan al ser ejecutadas, teniendo en cuenta conceptos relacionados a la perdida de información y riesgo de re-identificación definidos en el desarrollo del documento.

Basado en las métricas definidas y teniendo en cuenta distintos parámetros de configuración, se procede a la ejecución de las diversas técnicas sobre una base de datos que contiene información real acerca del CENSO realizado en Estados Unidos en 1994.

Finalmente, se analizan los resultados obtenidos para comprender en qué medida las técnicas afectan a la información considerando la utilidad de la misma. Dichos resultados, se presentan en forma de gráficos, facilitando la generación de conclusiones asociadas al estudio de las técnicas de anonimato. Paralelamente, se desarrolla una aplicación web diseñada para docentes y alumnos de la Universidad del Bío-Bío, con el fin de presentar el trabajo realizado y permitiendo la interacción, manipulación y ejecución de pruebas con los distintos algoritmos.

Índice General

Índice Figuras

Índice Tablas

1. Introducción

Agencias u organizaciones usualmente necesitan liberar información, tales como datos médicos, electorales y/o relacionados con el censo del país, para fines estadísticos o de investigación. Dicha información generalmente es liberada en forma de tablas, donde cada registro o fila de ella representa a un individuo en particular el cual se caracteriza por una serie de atributos, algunos de los cuales pueden contener información sensible, que es la forma por la cual se denomina a los datos personales de algún individuo tales como enfermedades y/o sueldo; así como también información que permite identificar directa o indirectamente a la persona en cuestión, como es el caso del Rol Único Nacional (RUN), nombres y apellidos, entre otros. Además, en la actualidad, los servicios informáticos que recopilan información de sus usuarios a través de la internet ha aumentado considerablemente, un claro ejemplo es el caso de los navegadores web, tales como Google Chrome, Firefox o Safari, los cuales recopilan información de uso de los usuarios, incluyendo el historial de navegación. Otro ejemplo son los servicios de streaming multimedia como "Netflix" que posee una base de datos enorme con información de sus usuarios.

Una correcta utilización de estos datos, como se mencionó anteriormente, es ser analizada con fines estadísticos o de investigación, e incluso para fines comerciales o de marketing; pero si la información es liberada de una forma pública puede caer en manos de sujetos malintencionados, quienes pueden utilizar esto para crear perfiles de los individuos afectados, con lo cual podrían generar un ser ficticio en redes sociales simulando ser una real, estudiar el comportamiento de ciertas personas, ver qué lugares frecuenta, cuáles son sus gustos o intereses, conocer enfermedades, remuneraciones o sueldos, profesión y otras

características de los usuarios, lo cual, claramente puede resultar perjudicial para los individuos que forman parte de los datos liberados.

Debido a este riesgo se hace más fuerte la noción de la privacidad de los datos públicos, la cual tiene como objetivo resguardar la identidad de los usuarios y/o evitar que, en el momento en que son liberadas estas bases de datos estadísticas, las personas puedan ser re-identificadas o se pueda conocer información sensible referente a ellas.

Bajo este contexto, surgen varias técnicas o algoritmos que buscan anonimizar las tablas a la hora de ser liberadas. Estas técnicas, inclusive en la actualidad, son extremadamente poco conocidas incluso por personas que tienen estricta relación con el área de la informática, ya que se requiere un cierto nivel de especialización en temas de seguridad de datos e información para adentrarse en estos conceptos, lo que implica la realización de un estudio del arte exhaustivo. Además, dichas técnicas son tratadas mayormente en ambientes teóricos más que de forma práctica, por lo cual resultan ser algo abstractas o difíciles de comprender para los alumnos a la hora de ser explicadas en entornos académicos por un docente, debido a la falta de ejemplos prácticos en los cuales se pueda apreciar claramente el actuar de los mecanismos tratados, y apreciar las diferencias entre las distintas técnicas de privacidad de acuerdo a las características que cada una de ellas posee y el escenario en el cual resulta más conveniente preferir un algoritmo sobre otro.

Por todos estos motivos surge la idea de este proyecto, donde se desea desarrollar un sistema en el cual se apliquen las diversas técnicas, donde se permita apreciar sus características principales y sus respectivas comparaciones para facilitar así su aprendizaje.

1.1. Definición del problema

Hoy en día, considerando la gran cantidad de información perteneciente a usuarios y personas que se almacena y se divulga públicamente, por parte de organizaciones o empresas, sin tener en cuenta las consecuencias que esto puede ocasionar, se hace indispensable un proceso de anonimización de datos que permita asegurar la privacidad de estos, sin modificarlos más de lo necesario. Sin embargo, en la actualidad, este es un tema que recién cobra importancia (algunos medios de comunicación y noticiarios han informado a la comunidad acerca de los riesgos de entregar datos a distintos tipos de entidades) y no se cuenta con suficientes ejemplos aplicados de forma práctica, lo cual implica que la enseñanza, comprensión y desarrollo sea difícil. Por lo tanto, es de vital importancia la creación de un punto de partida directamente en la formación de alumnos con el fin capacitados en este tema y que les permita obtener una noción completa sobre algoritmos de privacidad de datos y cómo aplicarlos.

1.2. Objetivos del proyecto de titulo

A continuación se describen los objetivos definidos para el desarrollo de este proyecto, los cuales son definidos en conjunto con el docente Patricio Galdames S.

Objetivo General:

Desarrollar una plataforma que implemente y muestre la efectividad de las diversas técnicas de privacidad de datos públicos desde el punto de vista de su utilidad.

Objetivos Específicos:

1. Realizar estudio del arte en técnicas que permiten proteger la privacidad de datos que se hacen públicos.
2. Comparar los diversos mecanismos de protección de datos desde el punto de vista de su efectividad en la anonimización de los datos.
3. Implementar un sistema web que permita a un estudiante experimentar con las diferencias entre las distintas técnicas de privacidad consideradas.
4. Generar, comparar y analizar resultados obtenidos al aplicar distintas técnicas de privacidad sobre la información de la base de datos considerando la privacidad versus la utilidad de la información anonimizada.

2. Estudio del arte

La sociedad ha experimentado un crecimiento exponencial en lo que se refiere a la cantidad, variedad y disponibilidad de conjuntos de datos, los cuales contienen información específica de las personas que incluyen datos personales, gustos, lugares que frecuenta, historiales de búsqueda, entre otros. Esto se debe a que la tecnología informática, la conectividad a la red de internet y los medios de almacenamiento se vuelven cada vez más accesibles. Empresas como supermercados, tiendas de retail, bancos; instituciones como universidades, hospitales, municipalidades: u otras organizaciones, se ven obligadas a almacenar grandes volúmenes de información en estructuras como bases de datos que son administradas por personas con gran conocimiento de las tecnologías de la información, quienes cumplen la función de ordenar los datos y velar por la disponibilidad de estos.

Es muy común que las organizaciones lleven a la práctica un proceso de publicación de datos propios de los usuarios o personas, ya sea para compartir, intercambiar y/o vender la información con el fin de realizar estudios estadísticos o análisis de datos. Sin embargo, los administradores de las bases de datos tienen una gran dificultad a la hora de liberar información de tal forma que no comprometa la privacidad y/o confidencialidad. Cabe mencionar que entre la información que contienen las bases de datos o tablas (a las cuales se les desea hacer un tratamiento para no arriesgar la seguridad de los usuarios) se encuentran identificadores explícitos, como el RUN (Rol único nacional) y el nombre completo del individuo; además de otros datos como la fecha de nacimiento, sexo, código postal, dirección, estado civil, etcétera. La información que identifica explícitamente a una persona es eliminada o encriptada en su totalidad, de modo que el proceso de protección de datos se aplique al resto de los valores.

¿Por qué es necesario realizar un tratamiento a los datos antes de ser liberados, si la información que identifica a cada una de las personas está suprimida? Existe una creencia común que afirma lo siguiente: “si los datos se ven anónimos son anónimos”. Pensar esto es incorrecto y se debe eliminar del pensamiento de las personas ligadas a la protección de la información, ya que la des-identificación de los datos (eliminación de los identificadores) no garantiza el anonimato. El resto de la información divulgada contiene otros datos que si son combinados pueden vincularse a otra fuente de datos que esté disponible de forma pública, lo que permite realizar un proceso de re-identificación de los individuos. Este proceso consiste en vincular directamente los atributos que están contenidos en la información liberada con otros datos disponibles públicamente o que pueden ser conseguidos a través de un pago, y así, obtener conocimiento extra sobre algún individuo que esté presente en ambas tablas.

Como se puede apreciar en la Figura 1, en la Tabla Privada, la cual puede ser una tabla de datos médicos, se han eliminado los identificadores explícitos con el fin de ocultar la identidad de los individuos, como lo son el nombre o el RUN, quedando solo los registros como el estado civil, la enfermedad, el sueldo y la previsión de salud que posee una determinada persona; mientras que en la Tabla Externa, la cual puede representar una tabla del Censo que puede ser encontrada de forma pública, que contiene información como el nombre completo de una persona, la dirección y ciudad donde vive, no se ha realizado ningún proceso de protección de la información. Además, se puede apreciar que ambas tablas comparten ciertos atributos, como lo son el código postal, la fecha de nacimiento y el sexo de la persona, y a partir de estos datos se puede identificar a un individuo siguiendo los siguientes pasos (ver Tabla 1):

Primero, en el caso de querer saber la enfermedad de algún individuo en particular, que forme parte de ambas tablas, es posible identificar su tupla directamente en la Tabla Externa, con el nombre, ciudad u otros datos ya conocidos. Luego, buscando los tres datos que se encuentran en ambas tablas (mencionados anteriormente) y comparándolos con los datos almacenados en la Tabla Privada, y si se cumple que dichos datos sean únicos para una tupla en ella, se podrá identificar los datos del individuo cuya información se quiere obtener y conocer así su enfermedad, además de otros datos como su estado civil, previsión de salud y su sueldo.

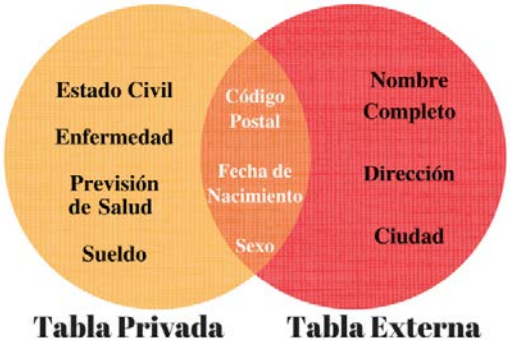


Figura 1: Atributos compartidos entre dos fuentes de información

RUN	Nombre Completo	Estado Civil	Previsión de Salud	Sexo	Fecha de Nacimiento	Código Postal	Enfermedad	Sueldo
***	***	Soltero/a						
***	***	Soltero/a						
***	***	Casado/a						
***	***	Casado/a						
***	***	*Divorciado/a*	*Fonasa*	*V*	*15/03/1961*	*1300000*	*Cáncer*	*$628.030*
***	***	Viudo/a						
***	***	Viudo/a						

Nombre Completo	Sexo	Fecha de nacimiento	Código Postal	Dirección	Ciudad
…	…	…	…	…	…
…	…	…	…	…	…
ACOSTA PIZARRO OMAR DEMETRIO	*V*	*15/03/1961*	*1300000*	*A PRATT 888*	*TALTAL*
…	…	…	…	…	…
…	…	…	…	…	…

Tabla 1: Proceso de re-identificación

Debido a este proceso de re-identificación se encuentra la necesidad de proporcionar algún tipo de protección a la información de las personas que se encuentran contenidas en bases de datos que se quieren liberar. Es por esto que se han desarrollado diversos procesos de protección utilizando técnicas como el intercambio de valores, aleatorización de datos y agregar ruido a los mismos. Sin embargo, estas técnicas incurren en un problema de manipulación excesiva de los registros, ya que al ser aplicadas se pierde información precisa que es requerida por los procesos de extracción de datos, investigación o análisis. Además, no proporcionan el anonimato que se requiere para proteger la información de los usuarios.

Es así como se da paso a las técnicas que son el objeto de estudio de este trabajo, que intentan amparar la información teniendo en cuenta la posterior utilización de los datos. El primer concepto que se revisará es el k-anonimato (en inglés, k-anonymity) dado que es uno de los primeros acercamientos en lo que respecta a la protección de la información. Es aquí donde se verán los conceptos de “anonimato”, “anonimización” y “cuasi-identificadores”; además, dos técnicas para lograr el k-anonimato; y, sus falencias. Luego, teniendo en cuenta las debilidades acerca de la técnica mencionada anteriormente, surge la idea de incluir el estudio de una técnica llamada ℓ-diversidad (en inglés, ℓ-diversity) que tiene como fin lograr la privacidad de los datos que deseen ser liberados teniendo en cuenta ciertos atributos que poseen características particulares denominados “atributos sensibles”. Finalmente, se dedicará un estudio acerca de un procedimiento que proporciona una mayor robustez al arte de la protección de los datos personales presentes en distintos almacenes de datos, el cual fue nombrado t-cercanía (en inglés, t-closeness).

2.1. k-anonimato

Previo a comentar, estudiar y desarrollar esta técnica, se definirán algunos conceptos que se deben tener en cuenta al momento de comprender los distintos mecanismos existentes que pretenden llevar una infinidad de datos a un estado de seguridad, para que el usuario tenga certeza de que su información estará protegida a la hora de ser liberada.

El primer concepto a definir es “anonimato” ya que, en el marco de algoritmos de protección de información, se utiliza para referirse al estado en el cual se encuentra un individuo dentro de un grupo finito de tuplas. En otras palabras, se encuentra en anonimato aquella persona que no se puede distinguir fácilmente dentro de un contenedor de datos. Además, es relevante saber que una “tabla privada” hace referencia a una tabla de datos donde cada tupla es una entidad o un individuo único. Es bueno recordar que, en una tabla privada, los identificadores explícitos están encriptados o han sido suprimidos. Teniendo en cuenta estos conceptos básicos, se procede a adentrarse en lo que es el k-anonimato.

El enfoque o técnica llamada k-anonimato es una de las primeras técnicas desarrolladas para aplicar una debida protección a la información de los usuarios contenida en algún tipo de almacenamiento de datos, siendo la más común, una tabla perteneciente a una base de datos relacional. Dicha técnica se desarrolla para otorgar una solución a la gran problemática, presentada anteriormente, llamada re-identificación, donde los individuos se pueden vincular teniendo dos o más fuentes de datos y enlazando conjuntos de atributos que los identifican de forma única. Esta solución se realiza mediante un algoritmo que pretende llevar la información contenida en una tabla de datos a un estado de anonimato, con fundamentos matemáticos sólidos, los cuales se muestran a continuación.

En primer lugar se debe tener presente el concepto de cuasi identificadores, los cuales se definen como una combinación de características sobre las que se pueden aplicar vinculantes.

Definición Cuasi identificadores (QI_T): Sea $T\ (A_1, \dots, A_n)$ una tabla con atributos. Un cuasi identificador de T es un conjunto de atributos $\{A_i, \dots, A_j\} \subseteq \{A_1, \dots, A_n\}$ cuya liberación debe ser controlada (Sweeney, 2002).

Además, se debe tener en cuenta el requisito de k-anonimato para luego definir como tal esta técnica. El requisito de k-anonimato dice que cada versión de los datos debe ser tal que cada combinación de los valores de los cuasi identificadores pueda ser igualada indistintamente con al menos k entidades o individuos (Samarati & Sweeney, 1998). Es en esta definición donde recae la importancia de los cuasi identificadores, ya que una buena elección de estos permitirá que la información externa existente no sirva (o sea dificultosa) para vincular e identificar a algún individuo en específico, así se protegerían los datos de cada una de las entidades contenidas en la tabla, asegurando su anonimato. Por el contrario, una elección errónea de los cuasi identificadores no serviría y sería muy perjudicial para los datos de los usuarios. Una recomendación para la elección de los cuasi identificadores es que deben ser seleccionados teniendo en cuenta la información que, más comúnmente, esté disponible de forma externa como por ejemplo: $QI_T = \{sexo, \text{fecha de nacimiento y código postal}\}$ (ver Tabla 2).

RUN	Nombre Completo	Estado Civil	Previsión de Salud	Sexo	Fecha de Nacimiento	Código Postal	Enfermedad	Sueldo
***	***	Soltero/a						
***	***	Soltero/a						
***	***	Casado/a						
***	***	Casado/a						
***	***	*Divorciado/a*	*Fonasa*	*V*	*15/03/1961*	*1300000*	*Cáncer*	*$628.030*
***	***	Viudo/a						
***	***	Viudo/a						

Tabla 2: Elección de un posible conjunto cuasi identificador

Con todas estas aclaraciones, se puede definir formalmente la técnica denominada k-anonimato, donde se sentencia que una tabla proporciona k-anonimato si los intentos que puede realizar una persona con el fin de vincular explícitamente los datos de algún individuo son en vano, ya que la información se mapea ambiguamente con al menos k entidades. Definitivamente la definición de k-anonimato es la siguiente (Sweeney, 2002):

Definición k-anonimato: Sea $T\ (A_1, \dots, A_n)$ una tabla y QI_T sean los cuasi-identificadores asociados con ella. Se dice que la tabla T satisface k-anonimato si, para cada cuasi-identificador $QI \in QI_T$, cada secuencia de valores en $T[QI]$ aparece al menos con k ocurrencias en $T[QI]$.

Si la tabla satisface la definición de k-anonimato para un determinado k, quiere decir que satisface el requisito de k-anonimato para tal k. Cabe mencionar que el valor de "k" representa la cantidad de tuplas a las que debe ser igual un determinado registro. Esta variable debe ser asignada por el administrador de los datos, teniendo en cuenta que mientras mayor sea este número, mayor será la seguridad hacia los usuarios contenidos en esta tabla.

Para conseguir este objetivo, se deben aplicar distintos procesos de transformación a los datos como son la generalización y la supresión de la información, lo cual se explicara a continuación, comenzando por el método de generalización de datos, donde se explicara todo lo necesario para entender en que consiste; luego, se estudiara el método de supresión de datos, donde se redefinirán algunos conceptos anteriores.

2.1.1. Método de Generalización para alcanzar el k-anonimato

La aplicación de un proceso de generalización de datos es de vital importancia para alcanzar el objetivo del k-anonimato, ya que consiste en hacer que el conjunto de datos en forma de tabla sea más anónimo o más general, entregando menos información para quien lo desee visualizar, consiguiendo así un mayor número de tuplas iguales, lo que es muy necesario para proteger los datos utilizando esta técnica denominada k-anonimato.

Para comprender como es el proceso de la generalización de información contenida en una tabla, se debe comprender un concepto muy importante, que corresponde al "dominio" en un sistema de base de datos, que se define como un conjunto de valores que puede asumir un determinado atributo (Samarati & Sweeney, 1998). Este conjunto de valores dependerá del tipo de atributo que se esté tratando, como por ejemplo: atributos de tipo numérico, fecha, cadena, código postal, entre otros.

Con el fin de ilustrar gráficamente lo que es la generalización del valor de un atributo, visualice la Figura 2, donde se toma como ejemplo un código postal:

Dominio Base		Dominio Generalizado (1)
Código Postal	$\rightarrow$	Código Postal
Z_0	$\rightarrow$	Z_1
05349	$\rightarrow$	05340

Figura 2: Proceso de generalización de un atributo

Se puede apreciar, que en una base de datos original, donde no se ha aplicado algún proceso de protección de la información, cada uno de los valores que poseen los atributos son lo más

completos y específicos posibles, a esto se le conoce como atributos en dominio base, el cual se denota como Z_0 (en el caso de que el atributo sea un código postal). Ahora, una de las formas para alcanzar el k-anonimato es realizando un proceso para que los valores sean menos informativos, haciendo un cambio en el dominio actual por otro más general y menos especifico, con el cual se puede describir un código postal, denotado como Z_1. En este dominio se ha reemplazado el último digito del código postal por un cero ($05349 \rightarrow 05340$). Dicho proceso de generalización del valor de un atributo se representa por un orden parcial ($\leq_D$) en el conjunto de dominios llamado Dom, y es necesario para cumplir las siguientes condiciones:

i. Cada dominio D_i tiene como máximo un solo dominio generalizado directo en el conjunto Dom.
ii. Cada uno de los elementos máximos de Dom son únicos.

Estas dos condiciones traen como consecuencia la existencia de una jerarquía de generalización de dominio denominada DGH_D (por sus siglas en inglés, Domain Generalization Hierarchy), para cada uno de los dominios $D \in Dom$. Así mismo, la definición de una relación de generalización de valores de orden parcial ($\leq_V$), la cual determina una asociación de cada valor v_i en un dominio D_i con un único valor en un dominio D_j (el cual es una generalización directa de D_i). Debido a esto, se crea una jerarquía de generalización de valores VGH_D (por sus siglas en inglés, Values Generalization Hierarchy) para cada dominio D. Con el fin de ilustrar estas definiciones, se toma como ejemplo los atributos código postal y etnia, creando sus jerarquías de generalización de dominio junto con sus jerarquías de generalización de valores, tal como se muestra en la Figura 3.

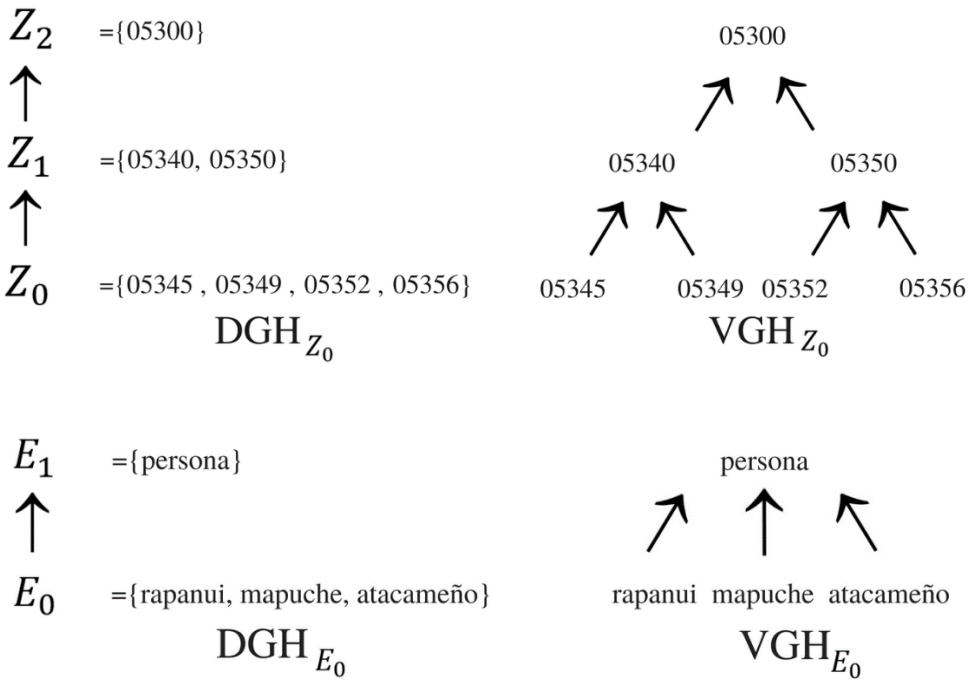


Figura 3: Jerarquías de generalización de dominio y de valores

Con el fin de llevar la teoría a una forma más práctica, se debe tener en cuenta que los cuasi identificadores son varios atributos, por lo tanto se redefine el concepto de "jerarquía de generalización de dominio" teniendo en cuenta los conjuntos de atributos que determinan un cuasi identificador en términos de tuplas para obtener las diferentes formas en que DT se puede generalizar.

Dada una tupla $DT = \langle D_1, \dots, D_n \rangle$ tal que $D_i \in Dom, i = 1, \dots, n,$ se define la jerarquía de generalización de dominio de DT como $DGH_{DT} = DGH_{D1} \times \dots \times DGH_{D_n}$, donde DGH_{DT} es el resultado de una matriz que posee un elemento mínimo DT (Samarati & Sweeney, 1998). En la Figura 4 se puede apreciar de forma gráfica esta definición, donde cada ruta desde el elemento mínimo DT hasta alcanzar un elemento máximo único de DGH_{DT} describe una posible alternativa de generalización de atributos. Debido a esto se define el concepto de "estrategia de generalización para DGH_{DT}" (en inglés, Generalization Strategy) como el

conjunto de todos los nodos con sus respectivas relaciones de generalización que abarca cada alternativa o ruta de generalización (Samarati & Sweeney, 1998). En resumen, la Figura 4 representa la jerarquía de generalización de dominio de DT junto con sus estrategias de generalización para dicha jerarquía, ejemplificándolo con los atributos cuasi identificadores "Etnia" y "Código postal".

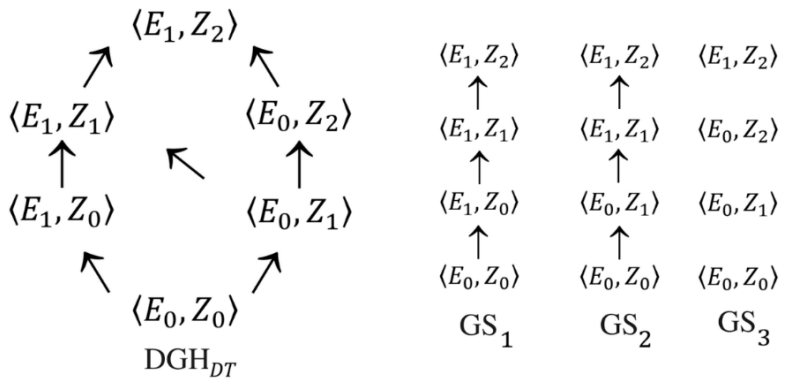


Figura 4: Jerarquía de generalización de dominio de una tupla y sus respectivas estrategias de generalización

Gracias a este gran proceso es posible generalizar cada uno de los valores almacenados en una tabla de datos, provocando un crecimiento en el tamaño de los clústeres (tuplas que tienen valores iguales) debido a la disminución de la cantidad de tuplas distintas. Además, hace posible que todos y cada uno de los valores contenidos en la tabla pertenezcan al mismo dominio, ya que la generalización es llevada a cabo a nivel de atributo, y cada uno de los valores son reemplazados por algún valor correspondiente a un dominio más general y menos informativo. Esto se denota de la siguiente forma: $D_i = dom(A_i, PT)$ donde D_i es el dominio asociado con el atributo A_i en la tabla privada PT.

Ya definido y desarrollado el proceso de generalización de datos, es posible realizar una definición formal para representar una tabla generalizada (Samarati & Sweeney, 1998):

Sean $T_i(A_1, \dots, A_n)$ y $T_j(A_1, \dots, A_n)$ dos tablas de datos definidas en un conjunto de atributos idéntico. Se dice que la tabla T_j es una generalización de la tabla T_i, ($T_i \leq T_j$), si y solo si cumple las siguientes condiciones:

i. $|T_i| = |T_j|$. T_i tiene el mismo número de tuplas que T_j.

ii. $\forall z = 1, \dots, n : dom(A_z, T_i) \leq_D dom(A_z, T_j)$. Para cualquier z entre 1 y n, el dominio de cada atributo en la tabla T_j es el mismo o una generalización del dominio del mismo atributo en T_i.

iii. Es posible definir una función biyectiva entre las dos tablas (T_i y T_j) que asocie cada una de las tuplas t_i y t_j tal que $t_i[A_z] \leq_V t_j[A_z]$. Cada tupla t_i perteneciente a T_i tiene una tupla t_j correspondiente en T_j tal que el valor para cada atributo en t_j es igual o una generalización del valor del atributo correspondiente a t_i.

Con la última definición desarrollada, se proporciona un ejemplo (Tabla 3) donde es posible apreciar el proceso de generalización de una forma más práctica y real, aplicándolo a una tabla privada PT de datos reales que consiste en dos atributos cuasi identificadores Etnia (E_0) y Código postal (Z_0) y un conjunto de 12 tuplas. Dicha tabla PT se generaliza aplicando la jerarquía de generalización de dominio y valores para E_0 y Z_0 que se aprecian en la Figura 4, dando origen a las otras 4 tablas restantes, donde cada una de ellas representa una posible generalización de la tabla PT, indicando en la parte superior de cada una el dominio de cada atributo.

Etnia: E_0	C. Postal: Z_0
rapanui	05345
rapanui	05349
rapanui	05352
rapanui	05356
mapuche	05345
mapuche	05349
mapuche	05352
mapuche	05356
atacameño	05345
atacameño	05349
atacameño	05352
atacameño	05356

PT

Etnia: E_1	C. Postal: Z_0
persona	05345
persona	05349
persona	05352
persona	05356
persona	05345
persona	05349
persona	05352
persona	05356
persona	05345
persona	05349
persona	05352
persona	05356

$GT_{[1,0]}$

Etnia: E_1	C. Postal: Z_1
persona	05340
persona	05340
persona	05350
persona	05350
persona	05340
persona	05340
persona	05350
persona	05350
persona	05340
persona	05340
persona	05350
persona	05350

$GT_{[1,1]}$

Etnia: E_0	C. Postal: Z_2
rapanui	05300
rapanui	05300
rapanui	05300
rapanui	05300
mapuche	05300
mapuche	05300
mapuche	05300
mapuche	05300
atacameño	05300
atacameño	05300
atacameño	05300
atacameño	05300

$GT_{[0,2]}$

Etnia: E_0	C. Postal: Z_1
rapanui	05340
rapanui	05340
rapanui	05350
rapanui	05350
mapuche	05340
mapuche	05340
mapuche	05350
mapuche	05350
atacameño	05340
atacameño	05340
atacameño	05350
atacameño	05350

$GT_{[0,1]}$

Tabla 3: Tabla privada junto con sus posibles generalizaciones

De las posibles generalizaciones de la tabla PT, se extrae la siguiente información referente a la satisfacción o cumplimiento del objetivo de k-anonimato:

a. $GT_{[1,0]}$ satisface el k-anonimato para valores de $k = 2,3$.

b. $GT_{[1,1]}$ satisface el k-anonimato para valores de $k = 2,3,4,5,6$.

c. $GT_{[0,2]}$ satisface el k-anonimato para valores de $k = 2,3,4$.

d. $GT_{[0,1]}$ satisface el k-anonimato para valores de $k = 2$.

Así mismo, se infiere que no todas las posibles generalizaciones se pueden considerar aceptables o satisfactorias, ya que algunas pueden ser consideradas “extremas” por el hecho de alcanzar el nivel más alto de generalización, teniendo como resultado una tabla con todas sus tuplas idénticas. Llegar a la obtención de una tabla con esas cualidades es innecesario si

existe alguna posible tabla generalizada que contenga valores más específicos y menos manipulados que satisfaga el mismo grado de k-anonimato. Es por este motivo que se introduce un nuevo concepto, el cual se definirá a continuación, que tiene completa relación con la restricción de la generalización innecesaria:

Vector de distancia: Sean $T_i(A_1, \dots, A_n)$ y $T_j(A_1, \dots, A_n)$ dos tablas tal que $T_i \leq T_j$. El vector de distancia desde la tabla T_i hasta la tabla T_j es $DV_{i,j} = [d_1, \dots, d_n]$, donde cada d_z representa la longitud de la ruta única entre $D = dom(A_z, T_i)$ y $dom(A_z, T_j)$ en la jerarquía de generalización de dominios DGH_D (Samarati & Sweeney, 1998). De esta forma se puede obtener el vector de distancia entre dos tablas con el objetivo de apreciar que tanto se ha manipulado la información en relación al número de veces en que un atributo cuasi identificador se ha sometido a un proceso de generalización. Por otro lado, los vectores de distancia $DV = [d_1, \dots, d_n]$ y $DV' = [d'_1, \dots, d'_n]$ se pueden relacionar de dos formas:

a. $DV \leq DV'$ si y solo si $d_i \leq d'_i$ para todo $i = 1, \dots, n$, es decir que d'_i es el mismo o una generalización de d_i para todo $i = 1, \dots, n$.
b. $DV < DV'$ si y solo si $DV \leq DV'$ y $DV \neq DV'$.

Con el objetivo de ilustrar esto, se presenta el siguiente ejemplo, donde se considera una tabla privada PT junto con sus respectivas tablas generalizadas (como las de la Tabla 3), obteniendo los vectores de distancia entre dicha tabla privada y sus posibles generalizaciones en forma de matriz (Figura 5).

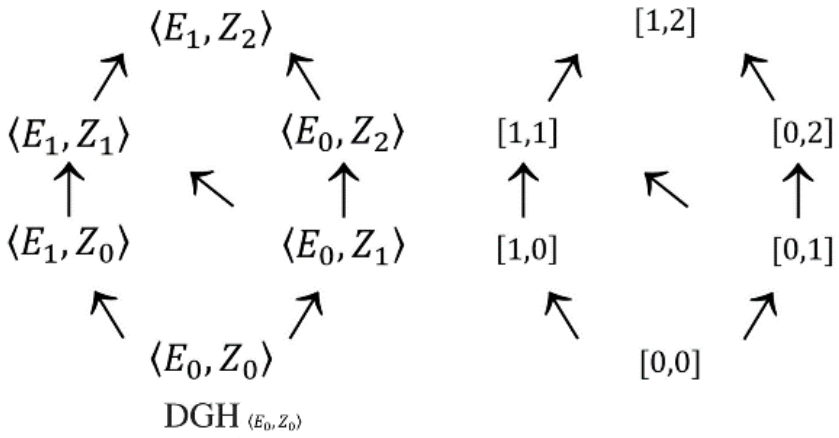


Figura 5: Jerarquía de generalización de dominio y su matriz de vectores de distancia

Para finalizar el desarrollo del proceso de generalización, se definirá un concepto de mucha importancia, que permite seleccionar una o más tabla(s) generalizada(s) de las que se generan a partir de la tabla privada. A continuación se define el concepto de generalización k-minima, el cual utiliza la definición de "vector de distancia" mencionada anteriormente.

Generalización k-mínima: Sean T_i y T_j dos tablas tal que $T_i \leq T_j$. Se dice que T_j es una generalización k-minima de T_i si y solo si (Samarati & Sweeney, 1998):

i. T_j satisface el k-anonimato para un valor determinado de k.

ii. $\nexists T_z: T_i \leq T_z$, T_z satisface el k-anonimato, y $DV_{i,z} < DV_{i,j}$. Es decir que no exista una tabla T_z que sea una generalización de T_i que satisfaga el k-anonimato y que el vector de distancia entre la tabla T_z y T_i sea menor que el vector de distancia entre T_j y T_i.

Por otro lado, es bueno mencionar que la técnica de k-anonimato requiere la existencia de k-tuplas iguales dentro de la tabla, pero esto aplica solo para los atributos cuasi identificadores, por lo tanto, para cada generalización k-minima, el vector de distancia en cualquier atributo que no pertenece al conjunto cuasi identificador es igual a cero.

Para ejemplificar el concepto (Tabla 4), nuevamente es considerada la tabla privada PT con sus respectivas posibles generalizaciones de la Tabla 3. Cabe recordar que el cuasi identificador escogido para los ejemplos consiste en $QI = (Etnia, Código\ postal)$. Teniendo en cuenta esto, se genera una tabla que consistente en las generalizaciones k-minimas considerando dos parámetros de anonimato, donde $k = 2$ y $k = 3$. Además se agrega una columna que muestra la(s) generalización(es) no k-minima(s), las cuales corresponden a las nuevas generalizaciones a partir de las k-minima(s).

Parámetro de k-anonimato	Generalización(es) k-minima(s)	Generalización(es) no k-minima(s)
$k = 2$	$GT_{[1,0]}$	$GT_{[1,1]}$, $GT_{[1,2]}$
	$GT_{[0,1]}$	$GT_{[1,1]}$, $GT_{[0,2]}$, $GT_{[1,2]}$
$k = 3$	$GT_{[1,0]}$	$GT_{[1,1]}$, $GT_{[1,2]}$
	$GT_{[0,2]}$	$GT_{[1,2]}$

Tabla 4: Generalizaciones k-mínimas y no k-mínimas a partir de una tabla privada

2.1.2. Método de Supresión

Como se explicó anteriormente, el proceso de generalización es de vital importancia para alcanzar el objetivo que propone el k-anonimato. Ahora que el objetivo ya se alcanzó, siempre es bueno mejorar el proceso, en este caso, con un enfoque complementario a la generalización, el cual es conocido como supresión de datos o información. Dicho método consiste en la supresión o eliminación de datos de alguna tabla, con el propósito de que estos no sean divulgados. Su principal función es actuar como moderador o controlador del proceso de generalización, ya que en ciertos casos un número limitado y pequeño de valores atópicos (tuplas con menos de k ocurrencias o apariciones) forzarían una o más generalizaciones adicionales para alcanzar el k-anonimato. En otras palabras, la supresión de datos se encarga de eliminar los valores atópicos para no generalizar la información más de lo necesario.

El proceso de supresión de registros es aplicado a nivel de tupla, ya que si existe un valor atópico en una determinada fila, solo se puede eliminar dicha fila en su totalidad, permitiendo que la tabla siga siendo consistente y estructurada.

Con el fin de ejemplificar y hacer practico el proceso de supresión de datos, se realizara una tabla que ilustra una posible proyección de atributos cuasi identificadores (etnia, fecha de nacimiento, sexo, código postal y estado civil) junto con dos generalizaciones mínimas (Tabla 5). Además, se proponen sus respectivas jerarquías de generalización de dominios y valores (Figura 6), complementando la información entregada en la Figura 3. Cabe mencionar que en el siguiente ejemplo, se tiene como objetivo lograr el k-anonimato con un valor de $k = 2$.

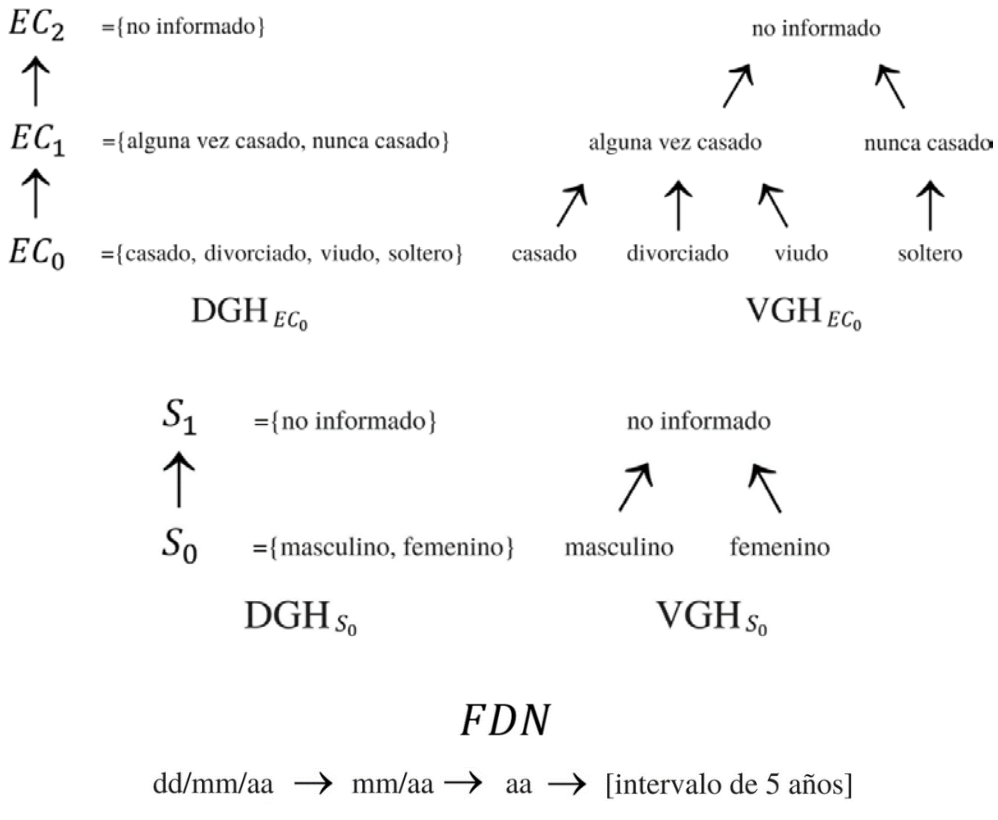


Figura 6: Jerarquías de generalización de dominio y valores (Estado civil, Sexo) y generalización del atributo Fecha de nacimiento

Etnia	Fecha de nacimiento	Sexo	Código postal	Estado civil
rapanui	19/08/63	femenino	05349	divorciado
rapanui	25/08/63	femenino	05349	divorciado
rapanui	15/03/63	masculino	05349	casado
rapanui	03/03/63	masculino	05349	casado
mapuche	07/02/62	masculino	05345	casado
mapuche	12/02/62	masculino	05345	casado
mapuche	08/08/63	femenino	05352	casado
mapuche	05/08/63	femenino	05352	casado
atacameño	10/06/60	masculino	05345	soltero
atacameño	01/06/60	masculino	05345	soltero
atacameño	13/06/60	femenino	05356	viudo

PT

Etnia	Fecha de nacimiento	Sexo	Código postal	Estado civil
rapanui	63	no informado	05300	no informado
rapanui	63	no informado	05300	no informado
rapanui	63	no informado	05300	no informado
rapanui	63	no informado	05300	no informado
mapuche	62	no informado	05300	no informado
mapuche	62	no informado	05300	no informado
mapuche	63	no informado	05300	no informado
mapuche	63	no informado	05300	no informado
atacameño	60	no informado	05300	no informado
atacameño	60	no informado	05300	no informado
atacameño	60	no informado	05300	no informado

$GT_{[0,2,1,2,2]}$

Etnia	Fecha de nacimiento	Sexo	Código postal	Estado civil
persona	[60-65]	femenino	05340	alguna vez casado
persona	[60-65]	femenino	05340	alguna vez casado
persona	[60-65]	masculino	05340	alguna vez casado
persona	[60-65]	masculino	05340	alguna vez casado
persona	[60-65]	masculino	05340	alguna vez casado
persona	[60-65]	masculino	05340	alguna vez casado
persona	[60-65]	femenino	05350	alguna vez casado
persona	[60-65]	femenino	05350	alguna vez casado
persona	[60-65]	masculino	05340	nunca casado
persona	[60-65]	masculino	05340	nunca casado
persona	[60-65]	femenino	05350	alguna vez casado

$GT_{[1,3,0,1,1]}$

Tabla 5: Ejemplo de una tabla PT junto con sus generalizaciones mínimas

Es posible verificar que la presencia de la última tupla de la tabla PT, $\{atacameño, 13/06/60, femenino, 05356, viudo\}$, requiere los siguientes pasos de generalización para cumplir este requisito:

a. 2 pasos de generalización en el atributo fecha de nacimiento, 1 en sexo, 2 en código postal y 2 en estado civil.
b. 1 paso de generalización en el atributo etnia, 3 en fecha de nacimiento, 1 en código postal y 1 en estado civil.

Así mismo, se puede constatar que, en el caso de que la última tupla de la tabla PT no hubiera estado presente (ver Tabla 6), el objetivo del k-anonimato con un valor de $k = 2$ pudiera haberse logrado mediante 2 pasos del proceso de generalización en el atributo cuasi identificador fecha de nacimiento, provocando una menor manipulación de los datos (menos generalizaciones) cumpliendo el objetivo de k-anonimato.

Etnia	Fecha de nacimiento	Sexo	Código postal	Estado civil
rapanui	19/08/63	femenino	05349	divorciado
rapanui	25/08/63	femenino	05349	divorciado
rapanui	15/03/63	masculino	05349	casado
rapanui	03/03/63	masculino	05349	casado
mapuche	07/02/62	masculino	05345	casado
mapuche	12/02/62	masculino	05345	casado
mapuche	08/08/63	femenino	05352	casado
mapuche	05/08/63	femenino	05352	casado
atacameño	10/06/60	masculino	05345	soltero
atacameño	01/06/60	masculino	05345	soltero

PT

Etnia	Fecha de nacimiento	Sexo	Código postal	Estado civil
rapanui	63	femenino	05349	divorciado
rapanui	63	femenino	05349	divorciado
rapanui	63	masculino	05349	casado
rapanui	63	masculino	05349	casado
mapuche	62	masculino	05345	casado
mapuche	62	masculino	05345	casado
mapuche	63	femenino	05352	casado
mapuche	63	femenino	05352	casado
atacameño	60	masculino	05345	soltero
atacameño	60	masculino	05345	soltero

$GT_{[0,2,0,0,0]}$

Tabla 6: Ejemplo de una tabla PT con supresión junto con su generalización mínima

Con este proceso complementario, que es la supresión de datos, se hace indispensable la redefinición de ciertos conceptos que se han definido anteriormente. En este caso se replanteara la definición de una tabla generalizada, agregando las características del proceso de supresión.

Tabla generalizada con supresión: Sean $T_i(A_1, \dots, A_n)$ y $T_j(A_1, \dots, A_n)$ dos tablas de datos definidas en un conjunto de atributos iguales. Se dice que la tabla T_j es una generalización de la tabla T_i, escrito $T_i \leq T_j$, si y solo si cumple las siguientes condiciones (Samarati & Sweeney, 1998):

i. $|T_i| \geq |T_j|$. T_i tiene un número menor o el mismo número de tuplas que T_j.

ii. $\forall z = 1, \ldots, n : dom(A_z, T_i) \leq_D dom(A_z, T_j)$. Para cualquier z entre 1 y n, el dominio de cada atributo en la tabla T_j es el mismo o una generalización del dominio del mismo atributo en T_i.

iii. Es posible definir un mapeo inyectivo entre las dos tablas (T_i y T_j) que asocie cada una de las tuplas t_i y t_j tal que $t_i[A_z] \leq_V t[A_z]$. Cada tupla t_i perteneciente a T_i tiene una tupla t_j correspondiente en T_j tal que el valor para cada atributo en t_j es igual o una generalización del valor del atributo correspondiente a t_i, o bien, puede ser el caso que alguna tupla que aparezca en la tabla T_i, no tenga una tupla correspondiente en T_j (propiedad de la función inyectiva), ya que puede haber sido suprimida.

Además, al aplicar la supresión de datos, se debe tener en cuenta la cantidad de tuplas suprimidas por cada una de las tablas generalizadas. En el caso de tener, por ejemplo, dos tablas generadas a partir de una tabla privada con vectores de distancia iguales, se debe seleccionar la tabla con menor supresión posible. Esto se contempla en la siguiente definición:

Supresión requerida mínima: Sean $T_i \; y \; T_j$ dos tablas tal que $T_i \leq T_j$, y T_j satisface el k-anonimato. Se dice que la tabla T_j impone una supresión mínima requerida si y solo si no existe una tabla T_z tal que (Samarati & Sweeney, 1998):

i. $T_i \leq T_z$. T_z sea una generalización de T_i.

ii. $DV_{i,z} = DV_{i,j}$. El vector de distancia desde la tabla T_i a la T_z es igual al vector de distancia desde la tabla T_i a la T_j.

iii. $|T_j| < |T_z|$. La cantidad de tuplas en la tabla T_z es mayor a la cantidad de tuplas que contiene T_j.

iv. T_z satisfaga el k-anonimato.

La definición anterior se puede ejemplificar considerando una tabla PT junto con sus posibles generalizaciones, que son ilustradas en la Tabla 7. En este ejemplo se puede observar que ciertas tuplas quieren ser suprimidas (filas escritas en cursiva y sombreadas) con el objetivo de alcanzar el k-anonimato con $k = 2$. Además, se puede inferir que la tabla más adecuada que cumple con la restricción de k-anonimato es la $GT_{[0,1]}$, ya que tiene una mayor cantidad de tuplas (comparada con la tabla que tiene el mismo vector de distancia y considerando que el vector de distancia es el menor de todos).

Etnia: E_0	C. Postal: Z_0
rapanui	05345
rapanui	05345
rapanui	05352
rapanui	05352
mapuche	*05345*
mapuche	*05351*
mapuche	*05352*
atacameño	*05345*

PT

Etnia: E_1	C. Postal: Z_0
persona	05345
persona	05345
persona	05352
persona	05352
persona	05345
persona	*05351*
persona	05352
persona	05345

$GT_{[1,0]}$

Etnia: E_0	C. Postal: Z_1
rapanui	05340
rapanui	05340
rapanui	05350
rapanui	05350
mapuche	*05340*
mapuche	05350
mapuche	05350
atacameño	*05340*

$GT_{[0,1]}$

Etnia: E_0	C. Postal: Z_2
rapanui	05300
rapanui	05300
rapanui	05300
rapanui	05300
mapuche	05300
mapuche	05300
mapuche	05300
atacameño	*05300*

$GT_{[0,2]}$

Etnia: E_1	C. Postal: Z_1
persona	05340
persona	05340
persona	05350
persona	05350
persona	05340
persona	05350
persona	05350
persona	05340

$GT_{[1,1]}$

Tabla 7: Tabla PT con sus respectivas generalizaciones aplicando posible supresión

Al permitir que se realice el proceso de supresión de filas, puede ocurrir que se generen más tablas por cada nivel de generalización, pero normalmente se cumple que para cada vector de distancia exista una única tabla generalizada que imponga una supresión mínima, ya que

el proceso la forma de aplicar estos procesos comienza con la generalización de vector de distancia y luego se eliminan las tuplas que estén presentes con menos ocurrencias.

Cabe mencionar que ambos enfoques (generalización y supresión de datos) producen los mejores resultados cuando son aplicados en conjunto, ya que de una u otra forma, si se utiliza la generalización de forma individual se puede producir una pérdida de información considerable, ya que ciertos atributos pueden generalizarse más, con el fin de alcanzar el anonimato, obteniendo tuplas sin información concreta e imprecisas. Por otro lado, si se aplica solamente una supresión de datos, puede incurrir en la eliminación de la totalidad de las tuplas, quedando una tabla vacía o sin información. Es así como se determina que la aplicación de estos enfoques debe ser de forma conjunta, ya que así se obtienen mejores resultados, más precisos (menos supresión y menos generalización) y cumpliendo con la restricción de k-anonimato.

Las definiciones anteriores dejan abierta la posibilidad de suprimir tuplas sin pensar en las repercusiones que esto puede ocasionar en temas de perdida de información, ya que si el proceso de supresión no tiene límites, se podría obtener una tabla sin registros o sin la utilidad requerida, como se mencionó anteriormente. Por otro lado, nace una interrogante: ¿es mejor generalizar a costa de una menor precisión o suprimir teniendo consecuencias de perdida de integridad? El concepto que se definirá a continuación, busca dar una solución a esta problemática:

Supresión máxima (MaxSup): es un umbral de supresión que se considera aceptable, el cual establece un número máximo de tuplas suprimidas. Bajo esta medida, la supresión se considera mejor que la generalización, ya que la primera afecta a tuplas de forma individual,

en cambio la generalización manipula todos los registros de la tabla. Es por este motivo que se procede a redefinir la generalización k-mínima agregándole el proceso de supresión.

Generalización k-mínima con supresión: Sean T_i y T_j dos tablas tal que $T_i \leq T_j$ y $MaxSup$ el umbral de supresión aceptable. Se dice que T_j es una generalización k-minima de T_i si y solo si (Samarati & Sweeney, 1998):

i. T_j satisface el k-anonimato para un valor determinado de k.

ii. $|T_i| - |T_j| \leq MaxSup$. La diferencia entre la cantidad de tuplas presentes en T_j respecto a las que contiene T_i debe ser menor al umbral $MaxSup$, es decir, no se debe suprimir más que la variable $MaxSup$.

iii. $\nexists T_z: T_i \leq T_z$, T_z satisface las condiciones (i) y (ii), y $DV_{i,z} < DV_{i,j}$. Es decir que no exista una tabla T_z que sea una generalización de T_i que satisfaga el k-anonimato, que se ajuste al umbral de supresión $MaxSup$ y que el vector de distancia entre la tabla T_z y T_i sea menor que el vector de distancia entre T_j y T_i.

A continuación, como en todas las definiciones, se proporciona un ejemplo para hacer posible la comprensión del concepto de umbral máximo de supresión, teniendo en cuenta las generalizaciones provenientes de una tabla privada y considerando una restricción de k-anonimato con un $k = 2$.

Etnia: E_0	C. Postal: Z_0
rapanui	05345
rapanui	05345
rapanui	05352
rapanui	05352
mapuche	05345
mapuche	05351
mapuche	05352
atacameño	05345

PT

Etnia: E_1	C. Postal: Z_0
persona	05345
persona	05345
persona	05352
persona	05352
persona	05345
persona	05352
persona	05345

$GT_{[1,0]}$

Etnia: E_0	C. Postal: Z_1
rapanui	05340
rapanui	05340
rapanui	05350
rapanui	05350
mapuche	05350
mapuche	05350

$GT_{[0,1]}$

Etnia: E_0	C. Postal: Z_2
rapanui	05300
rapanui	05300
rapanui	05300
rapanui	05300
mapuche	05300
mapuche	05300
mapuche	05300

$GT_{[0,2]}$

Etnia: E_1	C. Postal: Z_1
persona	05340
persona	05340
persona	05350
persona	05350
persona	05340
persona	05350
persona	05350
persona	05340

$GT_{[1,1]}$

Etnia: E_1	C. Postal: Z_2
persona	05300
persona	05300
persona	05300
persona	05300
persona	05300
persona	05300
persona	05300
persona	05300

$GT_{[1,2]}$

Tabla 8: Tabla PT con sus respectivas generalizaciones aplicando supresión

Umbral de supresión máxima	Generalización(es) k-minima(s)	Superan umbral de supresión	Generalización(es) no k-minima(s)
$MaxSup = 0$	$GT_{[1,1]}$	$GT_{[1,0]}, GT_{[0,1]}$ y $GT_{[0,2]}$	$GT_{[1,2]}$
$MaxSup = 1$	$GT_{[1,0]}$ y $GT_{[0,2]}$	$GT_{[0,1]}$	$GT_{[1,1]}$ y $GT_{[1,2]}$
$MaxSup \geq 2$	$GT_{[1,0]}$ y $GT_{[0,1]}$	-	$GT_{[0,2]}, GT_{[1,1]}$ y $GT_{[1,2]}$

Tabla 9: Generalizaciones k-mínimas y no k-mínimas a partir de una tabla privada considerando umbral de supresión

Complementando el ejemplo, se puede observar que es posible la generación de más de una generalización mínima para una tabla PT, un umbral de supresión y una restricción de k-anonimato. Sea cual sea la solución que se escoja, va a depender de medidas subjetivas y de

ciertas preferencias del administrador de datos, quien puede decidir una sobre otra ajustándose a la posterior utilización de los datos.

2.1.3. Falencias del k-anonimato y ataques

a) Falencia del orden de tuplas:

Como se ha expresado en el documento, el k-anonimato tiene como finalidad liberar tablas de datos cumpliendo una restricción de anonimato. Esto trae un problema si las tablas liberadas están igualmente ordenadas, es decir, el ataque consiste en comparar dos tablas que han sido lanzadas y ver si están ordenadas de tal forma que se pueda inferir o encontrar información adicional a la que está permitida. Un ejemplo simple, se puede ilustrar con una tabla privada PT junto a sus dos tablas generalizadas que se liberan. La tabla GT_1 es liberada en un momento determinado, luego se lanza una versión posterior (GT_2) de la misma tabla PT. Ambas tablas están en el mismo orden y si se realiza una relación entre cada una de las tuplas que contienen las tablas, se puede llegar a la misma información que contiene la tabla privada PT.

Etnia: E_0	C. Postal: Z_0
rapanui	05345
rapanui	05346
rapanui	05351
rapanui	05352
mapuche	05345
mapuche	05346
mapuche	05351
mapuche	05352
atacameño	05345
atacameño	05346
atacameño	05351
atacameño	05352

PT

Etnia: E_1	C. Postal: Z_0
persona	05345
persona	05346
persona	05351
persona	05352
persona	05345
persona	05346
persona	05351
persona	05352
persona	05345
persona	05346
persona	05351
persona	05352

GT_1

Etnia: E_0	C. Postal: Z_1
rapanui	05340
rapanui	05340
rapanui	05350
rapanui	05350
mapuche	05340
mapuche	05340
mapuche	05350
mapuche	05350
atacameño	05340
atacameño	05340
atacameño	05350
atacameño	05350

GT_2

Tabla 10: Ejemplo de falencia del orden de tuplas

Una solución a este ataque puede ser simple, y se trata de realizar un tratamiento de orden aleatorio a los datos. Así, se hace posible liberar distintas versiones privadas de la tabla PT, cada una con un orden diferente de las tuplas.

b) Ataque de atributos no-cuasi identificadores:

Como se pudo observar en el ejemplo anterior, ambos atributos pertenecían al conjunto de cuasi identificadores, pero esto no siempre es así. Muchos atributos que están contenidos en alguna tabla que se desee liberar no son considerados cuasi identificadores, por lo que los cuasi identificadores solo son un subconjunto de todos los atributos de la tabla.

Una forma de ejemplificar este problema se puede realizar con una tabla privada PT que quiere ser liberada cumpliendo el k-anonimato con $k = 2$ y considerando como cuasi identificador $QI_{PT} = \{Etnia, Fecha\ de\ nacimiento, Sexo, Código\ postal\}$. Aplicando un proceso de generalización se libera una tabla llamada GT_1, y después de un determinado periodo se libera otra generalización llamada GT_2 (ambas nacen de PT). Al momento de liberar GT_2 se pierde la protección de k-anonimato (aunque el orden de las tuplas sea aleatoria) ya que se puede realizar un proceso de vinculación en el atributo "Problema", generando otra tabla muy parecida a PT llamada "tabla de vinculación" que no cumple con la restricción de k-anonimato. Esto se determina ya que se puede observar dos tuplas que son únicas:

- $\langle rapanui, 1964, masculino, 05345, falta\ de\ aliento \rangle$
- $\langle rapanui, 1965, femenino, 05345, hipertención \rangle$

Este ataque no se podría realizar si se considerara alguna medida como: la inclusión del atributo "Problema" dentro del conjunto cuasi identificador QI_{PT}, o bien, si GT_2 tomará como

base la tabla GT_1 en lugar de PT, así en ningún caso existiría un valor más específico que los que contiene la tabla GT_1.

Etnia	Fecha de nacimiento	Sexo	C. Postal	Problema
mapuche	20/09/1965	masculino	05351	falta de aliento
mapuche	14/02/1965	masculino	05351	dolor de pecho
mapuche	23/10/1965	femenino	05345	dolor de ojo
mapuche	24/08/1965	femenino	05345	respiración sibilante
mapuche	07/11/1964	femenino	05345	obesidad
mapuche	01/12/1964	femenino	05345	dolor de pecho
rapanui	23/10/1964	masculino	05345	falta de aliento
rapanui	15/03/1965	femenino	05346	hipertensión
rapanui	13/08/1964	masculino	05346	obesidad
rapanui	05/05/1964	masculino	05346	fiebre
rapanui	13/02/1967	masculino	05345	vómitos
rapanui	21/03/1967	masculino	05345	dolor de espalda

PT

Etnia	Fecha de nacimiento	Sexo	C. Postal	Problema
persona	1965	masculino	05350	falta de aliento
persona	1965	masculino	05350	dolor de pecho
persona	1965	femenino	05340	dolor de ojo
persona	1965	femenino	05340	respiración sibilante
persona	1964	femenino	05340	obesidad
persona	1964	femenino	05340	dolor de pecho
persona	1964	masculino	05340	falta de aliento
persona	1965	femenino	05340	hipertensión
persona	1964	masculino	05340	obesidad
persona	1964	masculino	05340	fiebre
persona	1967	masculino	05340	vómitos
persona	1967	masculino	05340	dolor de espalda

GT_1

Etnia	Fecha de nacimiento	Sexo	C. Postal	Problema
mapuche	[1960-1969]	no informado	05351	falta de aliento
mapuche	[1960-1969]	no informado	05351	dolor de pecho
mapuche	[1960-1969]	no informado	05345	dolor de ojo
mapuche	[1960-1969]	no informado	05345	respiración sibilante
mapuche	[1960-1969]	no informado	05345	obesidad
mapuche	[1960-1969]	no informado	05345	dolor de pecho
rapanui	[1960-1969]	no informado	05345	falta de aliento
rapanui	[1960-1969]	no informado	05346	hipertensión
rapanui	[1960-1969]	no informado	05346	obesidad
rapanui	[1960-1969]	no informado	05346	fiebre
rapanui	[1960-1969]	no informado	05345	vómitos
rapanui	[1960-1969]	no informado	05345	dolor de espalda

GT_2

Etnia	C. Postal	Sexo	C. Postal	Problema
mapuche	1965	masculino	05351	falta de aliento
mapuche	1965	masculino	05351	dolor de pecho
mapuche	1965	femenino	05345	dolor de ojo
mapuche	1965	femenino	05345	respiración sibilante
mapuche	1964	femenino	05345	obesidad
mapuche	1964	femenino	05345	dolor de pecho
rapanui	*1964*	*masculino*	*05345*	*falta de aliento*
rapanui	*1965*	*femenino*	*05346*	*hipertensión*
rapanui	1964	masculino	05346	obesidad
rapanui	1964	masculino	05346	fiebre
rapanui	1967	masculino	05345	vómitos
rapanui	1967	masculino	05345	dolor de espalda

Tabla de vinculación

Tabla 11: Ataque de atributos no-cuasi identificadores

c) Ataque de homogeneidad.

Uno de los ataques más conocidos que se puede aplicar en el k-anonimato es el llamado "ataque de homogeneidad". Para ejemplificar este acontecimiento se puede observar la Tabla 12, la cual proviene de un tabla privada que se protegió utilizando la técnica de k-anonimato con un valor de $k = 3$, donde el cuasi identificador se compone por los atributos "Edad" y "Código Postal", ambos generalizados. El atributo restante corresponde a la "Problema" al cual no se le aplica ningún tipo de transformación.

El ataque se puede llevar a cabo si el individuo malintencionado conoce la edad de la persona que busca (o bien, conoce si es menor o mayor a los 60 años), y también sabe el código postal completo o solo los tres primeros dígitos. En este caso, se sabe que la persona buscada tiene una edad menor a los 50 años y su código postal comienza con los dígitos "054" (ya que víctima y atacante son vecinos).

Edad	Código Postal	Problema
< 50	053**	gripe
< 50	053**	dolor de pecho
< 50	053**	dolor de espalda
≥ 50	053**	obesidad
≥ 50	053**	vómitos
≥ 50	053**	vómitos
< 50	054**	gripe
< 50	054**	gripe
< 50	054**	gripe
≥ 50	054**	vómitos
≥ 50	054**	dolor de espalda
≥ 50	054**	dolor de espalda

Tabla 12: Ataque de homogeneidad

Con esta información se puede saber que el individuo buscado tiene “gripe”. Además, se aprecia una falta de diversidad de atributos sensibles (problemas) en un grupo generalizado, lo que puede provocar una divulgación de información involuntaria. Por lo tanto se concluye que la tabla cumple con 3-anonimato, pero no asegura la privacidad de las personas.

d) Ataque de conocimiento previo.

Es muy común que los atacantes posean conocimientos previos acerca del individuo que quieren encontrar, los cuales pueden utilizarse para descartar información que no se relaciona con la persona en cuestión. Por ejemplo, se supone que el adversario quiere encontrar a una persona, la cual tiene asociada una edad mayor a 50 años y un código postal “05485”.

Edad	Código Postal	Problema
< 50	053**	gripe
< 50	053**	dolor de pecho
< 50	053**	dolor de espalda
≥ 50	053**	obesidad
≥ 50	053**	vómitos
≥ 50	053**	vómitos
< 50	054**	gripe
< 50	054**	gripe
< 50	054**	gripe
≥ 50	054**	vómitos
≥ 50	054**	dolor de espalda
≥ 50	054**	dolor de espalda

Tabla 13: Ataque de conocimiento previo

Con estos datos, el atacante no puede saber si el problema de la persona es “vómitos” o “dolor de espalda”, pero, el individuo posee un conocimiento previo, y sabe que la persona no fue al médico por un problema estomacal, por lo tanto, puede inferir que tiene un “dolor de espalda”. Es así como se demuestra que el k-anonimato no tiene en cuenta el conocimiento previo del o los adversarios.

2.2. ℓ-diversidad

En la sección 2.1.3 de este documento se presentan los ataques y falencias más comunes que se pueden aplicar sobre la técnica de k-anonimato, de los cuales sobresalen dos, el "ataque de homogeneidad" y "ataque de conocimiento previo", ambos muestran la necesidad de la creación de una nueva técnica para proteger la información de las personas. Dichos ataques se pueden mitigar o eliminar aplicando una noción de "diversidad" de datos, haciendo aún más difícil la tarea de los atacantes para identificar a un determinado individuo o conocer información sensible sobre este.

Antes de adentrarse en la definición de una nueva técnica que proporciona anonimato y diversidad a los individuos contenidos en una estructura de datos, se procede a introducir una noción ideal de privacidad, la cual es denominada "Bayes-Optimal Privacy" (Machanavajjhala, Venkitasubramaniam, Kife, & Gehrke, 2006), la que se puede traducir como *Privacidad Óptima de Bayes* que hace alusión al *Teorema de Bayes* (proposición estadística planteada por Thomas Bayes). Esta noción es aplicada para el caso en que tanto el administrador de los datos como el atacante tenga un conocimiento de fondo completo e idéntico de la información, teniendo en cuenta que el conocimiento de fondo es un conocimiento externo, adquirido en el mundo real, que se posee sobre algún individuo, por ejemplo, el adversario sabe que una persona en particular es propensa a tener enfermedades cardiacas. Cabe mencionar que la Privacidad Óptima de Bayes no es práctica, ya que dicha privacidad supone el conocimiento completo de ambas partes, lo que no es probable. Por lo tanto, se utilizara esta definición como introducción a un nuevo algoritmo que proporciona privacidad.

Bayes-Optimal Privacy supone el peor escenario, donde el adversario tiene un conocimiento de fondo completo de la distribución de los atributos sensibles S y no sensibles Q. Además sabe que la persona que busca está en un registro t que pertenece a la tabla privada T, el cual se ha generalizado a un registro t^* en la tabla publicada T^* y el valor de los atributos no sensibles $t[Q] = q$. Con esta información el atacante pretende identificar el valor del dato sensible de un individuo ($t[S]$). El éxito que puede tener el adversario se define en base a dos tipos de creencias:

a) El primero es el llamado "Prior belief" o creencia previa, que se define como la creencia que tiene el atacante antes de ver la tabla anónima liberada.

$$\alpha_{(q,s)} = P_f(t[S] = s | t[Q] = q)$$

b) Por otro lado está el "Posterior belief" o creencia posterior, el que surge luego de que el atacante observa la tabla publicada T^*.

$$\beta_{(q,s,T^*)} = P_f\left(t[S] = s \middle| t[Q] = q \wedge \exists t^* \in T^*, t \overset{*}{\rightarrow} t^*\right)$$

A partir de esta definición de la creencia posterior surge la siguiente fórmula matemática:

$$\beta_{(q,s,T^*)} = \frac{n_{(q^*,s)} \frac{f(s|q)}{f(s|q^*)}}{\sum_{s' \in S} n_{(q^*,s')} \frac{f(s'|q)}{f(s'|q^*)}}$$

Ecuación 1: Bayes-Optimal Privacy

Donde q es el valor de los atributos no sensibles Q en la tabla T; q^* el valor generalizado de q en la tabla T^*; s un posible valor del atributo sensible; $n_{(q^*,s')}$ el número de tuplas t^* de la

tabla T^*, donde $t^*[Q] = q^*$ y $t^*[S] = s'$; y $f(s'|q^*)$ la probabilidad condicional del atributo sensible condicionado al hecho de que el atributos no sensible se pueda generalizar a q^*.

Es importante notar de esta fórmula que se considera la cantidad de apariciones del valor de un atributo sensible proporcional a todos los valores sensibles en un "bloque q^*" (este término corresponde a una "clase de equivalencia", la que se definirá más adelante), y la frecuencia de distribución f de un valor sensible respecto a todos los posibles valores sensibles de una población de datos. Además se debe recordar que esta fórmula supone un conocimiento de fondo completo.

Debido al conocimiento de fondo del adversario, una tabla T^* puede divulgar información importante de dos formas:

a) Revelación positiva: si el adversario puede identificar correctamente el valor de un atributo sensible.
b) Revelación negativa: si el adversario puede eliminar correctamente algunos valores posibles del atributo sensible, con alta probabilidad.

De ambas definiciones se desprende el "Principio Desinformativo", el cual es la base de la definición ideal de privacidad. Este principio expresa lo siguiente:

"La tabla publicada no debe proporcionar información adicional a la que ya conocía, en otras palabras, la diferencia entre la creencia previa y la creencia posterior debe ser mínima".

Los problemas que aparecen gracias a la definición de Bayes-Optimal Privacy, como son las formas de revelación (positiva y negativa), se deben a la falta de diversidad en la información sensible. Es así como se da paso a la definición de la nueva técnica para proteger los datos llamada ℓ-diversity.

La ℓ-diversity o ℓ-diversidad es la segunda técnica que se presenta en este documento. Su origen recae en las falencias y ataques que presenta la técnica anterior (k-anonimato) y en la definición de la Privacidad Óptima de Bayes, las que se pueden mejorar aplicando diversidad a los registros. Una definición simple para el algoritmo de ℓ-diversidad es la siguiente (Machanavajjhala, Venkitasubramaniam, Kife, & Gehrke, 2006):

Definición ℓ-diversidad: Se dice que una clase de equivalencia tiene ℓ-diversidad si hay al menos ℓ valores bien representados para el atributo sensible. Una tabla tiene ℓ-diversidad si cada clase de equivalencia en la tabla tiene ℓ-diversidad, donde una "clase de equivalencia" se puede definir como un conjunto de tuplas que poseen los mismos valores en sus atributos cuasi identificadores.

2.2.1. Instancias de la ℓ-diversidad

2.2.1.1. ℓ-diversidad distintiva

La forma más simple de interpretar la frase "bien representados", extraída desde la definición de ℓ-diversidad, sería asegurar que existan al menos ℓ valores distintos para el atributo sensible en cada clase de equivalencia que contiene una estructura de datos. Esta instancia de la técnica de protección de datos se puede aprecian en la Tabla 14, la cual representa una tabla T que está compuesta por tres atributos, de los cuales "Edad" y "Código Postal" son atributos no sensibles (cuasi identificadores) y "Problema" es un atributo sensible; además de cuatro clases de equivalencia, cada una formada por cuatro tuplas. Dicha tabla cumple la definición de ℓ-diversidad distintiva con un valor de ℓ igual a 3.

Atributos no sensibles (Q)		Atributo sensible (S)
Edad	Código Postal	Problema
< 50	053**	gripe
< 50	053**	dolor de pecho
< 50	053**	dolor de pecho
< 50	053**	dolor de espalda
≥ 50	053**	obesidad
≥ 50	053**	fiebre
≥ 50	053**	vómitos
≥ 50	053**	vómitos
< 50	054**	fiebre
< 50	054**	fiebre
< 50	054**	dolor de ojo
< 50	054**	hipertensión
≥ 50	054**	vómitos
≥ 50	054**	gripe
≥ 50	054**	gripe
≥ 50	054**	dolor de espalda

Tabla 14: Tabla 3-diversa distintiva

Sin embargo, esta definición es propensa al ataque de inferencia probabilística, el cual consiste en que un adversario pueda aprovechar que en una clase de equivalencia aparezca un valor sensible con una frecuencia mucho mayor a la de los demás valores. Permitiendo así que el adversario pueda concluir que una entidad tiene altas probabilidades de tener ese valor.

2.2.1.2. Entropía ℓ-diversidad

Considerando las debilidades de la definición anterior de ℓ-diversidad distintiva, se presenta la segunda instancia de esta técnica, la cual está basada principalmente en el cálculo de una medida de incertidumbre llamada entropía. La llamada "Entropía ℓ-diversidad" tiene su origen en la proposición de Bayes-Optimal Privacy, lo que será tratado más adelante.

La definición formal para describir la Entropía ℓ-diversidad es la siguiente: "Una tabla T cumple el requisito de Entropía ℓ-diversidad si para cada clase de equivalencia se cumple que (Machanavajjhala, Venkitasubramaniam, Kife, & Gehrke, 2006):

$$-\sum_{s\in S} p(q^*, s)\log\left(p_{(q^*,s\prime)}\right) \geq \log(\ell)$$

Donde $p_{(q^*,s)} = \frac{n_{(q^*,s)}}{\sum_{s\prime\in S} n_{(q^*,s\prime)}}$ es la fracción de tuplas en la clase de equivalencia con un atributo sensible igual a s. Cabe mencionar, que esta fórmula proviene de la Ecuación 1: Bayes-Optimal Privacy, desestimando el conocimiento de fondo completo que era considerado en el teorema de Bayes, permitiendo que esta definición sea práctica, y no solo posible en un ambiente teórico.

Una forma más sencilla de representar la entropía de una clase de equivalencia E, es la siguiente (Stammler, Katzenbeisser, & Hamacher, 2016):

$$Entropía(E) = -\sum_{s\in S} p(E, s)\log\ p(E, s) \geq \log(\ell)$$

Donde S es el valor del atributo sensible y $p(E, s)$ es el grupo de registros dentro de la clase de equivalencia E que tiene el valor del atributo sensible s.

De lo anterior, se desprende que para hacer cumplir el requisito de Entropía ℓ-diversidad, el valor de la entropía de la tabla completa debe ser al menos $\log(\ell)$, donde la entropía de la tabla completa corresponde a los valores de entropía de cada clase de equivalencia, los cuales deben ser (cada uno de ellas) al menos $\log(\ell)$. Esto queda demostrado en el siguiente ejemplo, donde se muestra una tabla T que contiene tres clases de equivalencia, cada una con el mismo valor de entropía, la cual es calculada de la siguiente forma:

$$Entropia\ (E) = -\left[\frac{1}{4}\log\left(\frac{1}{4}\right) + \frac{1}{4}\log\left(\frac{1}{4}\right) + \frac{2}{4}\log\left(\frac{2}{4}\right)\right] = 0.4515$$

Por lo tanto la entropía de la tabla completa es igual a 0.4515, y con esto se puede concluir que dicha tabla cumple Entropía *2.8*-diversidad (para encontrar este valor se debe calcular $10^{0.4515} = 2.8$).

Atributos no sensibles (Q)		Atributo sensible (S)
Edad	Código Postal	Problema
< 50	0538*	gripe
< 50	0538*	dolor de pecho
< 50	0538*	dolor de espalda
< 50	0538*	dolor de espalda
≥ 50	0647*	dolor de espalda
≥ 50	0647*	gripe
≥ 50	0647*	dolor de pecho
≥ 50	0647*	dolor de pecho
< 50	0539*	gripe
< 50	0539*	dolor de pecho
< 50	0539*	dolor de espalda
< 50	0539*	dolor de espalda

Tabla 15: Tabla que cumple Entropía 2.8-diversidad

Esta noción de ℓ-diversidad es más fuerte que la anterior, sin embargo, un problema que presenta es que en algunos casos puede resultar ser muy restrictiva, ya que la entropía de la tabla completa puede ser muy baja si unos pocos valores son muy comunes, considerando que la definición fuerza a que la entropía de la tabla entera debe ser al menos $\log(\ell)$.

2.2.1.3. (c, ℓ)-diversidad recursiva

Debido a la característica restrictiva de la definición de Entropía ℓ-diversidad, se posibilita la creación de una noción menos conservativa y más ambiciosa de esta técnica, la llamada (c, ℓ)-diversidad recursiva. Esta instancia de ℓ-diversidad permite asegurar que la aparición del valor del atributo sensible menos frecuente no sea tan escasa y que la presencia del valor más frecuente no sea tan numerosa. Con el fin de conseguir esto, se definirá un

conjunto de los valores sensibles diferentes en una clase de equivalencia como "S" y el número de veces en que el i-ésimo valor sensible más frecuente aparece en la clase de equivalencia E como "n_i", tal que $1 \leq i \leq |S|$, considerando que el conjunto de frecuencias de aparición de los valores sensibles se debe ordenar descendentemente.

Así se puede pensar acerca de la ℓ-diversidad de la siguiente manera (Machanavajjhala, Venkitasubramaniam, Kife, & Gehrke, 2006): un adversario necesita eliminar al menos $\ell - 1$ valores posibles de S para poder inferir la revelación positiva. Por lo tanto, se dice que una clase de equivalencia E es (c, ℓ)-diversa si se cumple que $n_1 < c(n_\ell + n_{\ell+1} + \ldots + n_s)$ para algún valor de la constante c definida por el administrador de los datos, y si se puede eliminar un valor de S y esta clase de equivalencia aun cumpla con $(c, \ell - 1)$-diversidad recursiva (debido a esto surge la propiedad de recursividad de esta instancia de ℓ-diversidad. Por lo tanto, una tabla cumple el requisito de (c, ℓ)-diversidad recursiva si cada clase de equivalencia E cumple con $n_1 < c(n_\ell + n_{\ell+1} + \ldots + n_s)$.

Para explicar esta definición, se presentará el siguiente ejemplo:

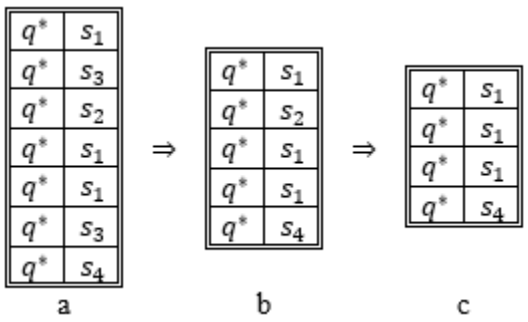


Figura 7: Clase de equivalencia (c,l)-diversa

Se muestra una clase de equivalencia E que consta de siete tuplas. Desde ella se extrae el conjunto de los diferentes valores sensibles $S = \{s_1, s_2, s_3, s_4\}$, teniendo en cuenta que los

valores del cuasi identificador han sido generalizados a q^*. Entonces, el conjunto de las frecuencias de todos los valores sensibles (en la clase de equivalencia) ordenados descendentemente es $\{n_1 = 3, n_2 = 2, n_3 = 1, n_4 = 1\}$ (el subíndice de cada elemento en este conjunto no tiene relación con el subíndice de los valores sensibles en la clase de equivalencia).

Teniendo estos datos, y definiendo una constante $c = 2$, se puede proseguir con la ejecución del algoritmo, el cual lo describiremos en forma de iteraciones:

i. Se tiene una clase de equivalencia que es $(2,3)$-diversa (ver Figura 7.a), en la cual $n_1 = 3$ (mayor número de repeticiones de un atributo sensible) debe ser menor que $c(n_\ell + n_{\ell+1} + \ldots + n_s)$, para $\ell = 3$ en este caso.

   Por lo tanto, $3 < 2(n_3 + n_4) \Rightarrow 3 < 2(1 + 1) \Rightarrow 3 < 4$ cumple la condición.

ii. En la siguiente iteración se procede a eliminar el segundo valor más frecuente del conjunto S (s_3), produciendo la tabla que se puede apreciar en la Figura 7.b, y se vuelve a definir el conjunto de frecuencias, obteniendo $\{n_1 = 3, n_2 = 1, n_3 = 1\}$. Esta nueva clase de equivalencia cumple con el requisito de $(2,2)$-diversidad recursiva, ya que el valor de ℓ vario a 2 (se tienen 3 valores diferentes, por lo tanto se calcula $\ell - 1 \Rightarrow 3 - 1 \Rightarrow 2$).

   Es así como, $3 < 2(n_2 + n_3) \Rightarrow 3 < 2(1 + 1) \Rightarrow 3 < 4$ cumple la condición.

iii. Se procede a eliminar el tercer valor sensible más frecuente del conjunto S (s_2), obteniendo la tabla que se muestra en la Figura 7.c; además, se redefine el conjunto de frecuencias obteniendo: $\{n_1 = 3, n_2 = 1\}$. Esta clase de equivalencia cumple con el requisito de $(2,1)$-diversidad recursiva.

   Entonces, $3 < 2(n_1 + n_2) \Rightarrow 3 < 2(3 + 1) \Rightarrow 3 < 8$ cumple la condición.

De esta forma concluye la ejecución del algoritmo, ya que no es posible eliminar más valores sensibles, ya que si $\ell = 1$ se tendría un solo valor distinto y no cumpliría la definición básica de ℓ-diversidad. Así mismo, se demuestra el éxito de esta clase de equivalencia para la instancia de (c, ℓ)-diversidad recursiva.

2.2.2. Limitaciones de la ℓ-diversidad y ataques

La ℓ-diversidad puede ser difícil e innecesaria de conseguir en algunos casos, por ejemplo, si se tiene una tabla compuesta por 1.000 registros donde el único atributo sensible es el resultado de un test de alguna enfermedad, el cual puede tomar solo los valores de negativo o positivo. Supongamos que el 99% de los resultados fueron negativos y solo el 1% fue positivo. Claramente, el administrador de los datos desearía ocultar el hecho de que pocos valores fueron positivos, ya que la mayoría fue negativo. En este caso, *2*-diversidad es innecesaria para una clase de equivalencia que solo contiene valores negativos. Para conseguir *2*-diversidad distintiva puede haber máximo 10 clases de equivalencia (1% de 1.000) y la perdida de información seria extremadamente grande. Además, se debe notar que la entropía de la tabla completa es muy pequeña, lo que ocasionaría que el valor de ℓ deba ser muy pequeño, en el caso de que el administrador quiera utilizar Entropía ℓ-diversidad.

Además, la técnica ℓ-diversidad puede ser el blanco de variados ataques, como los que se presentan a continuación:

a) Ataque de la falta de simetría (oblicuidad):

Este ataque puede ocurrir cuando la distribución total esta sesgada, en tal caso, alcanzar el requisito de la ℓ-diversidad no previene la revelación de atributos.

Para ejemplificar este problema se piensa en una tabla perteneciente a una base de datos que está compuesta por los resultados de un test de enfermedad, que al igual que el anterior, solo puede tomar valores “positivo” o “negativo”. Si se hace un supuesto en donde una clase de equivalencia tiene igual número de registros positivos y negativos, la tabla satisface *2*-diversidad distintiva, entropía *2*-diversidad y cualquier requerimiento de $(c, 2)$-diversidad recursiva que pueda ser impuesto.

Sin embargo, esto presenta un serio riesgo de la privacidad, ya que cada persona de la clase de equivalencia puede tener 50% de posibilidades de ser positivo, en comparación del 1% de la población total del ejemplo anterior. Por otro lado, supongamos que una clase de equivalencia tiene 49 registros positivos y solo un valor negativo, cumple *2*-diversidad distintiva y tiene mayor entropía que la tabla completa. Aunque cada persona en la clase de equivalencia puede tener un 98% de posibilidades de ser positiva, en lugar de 1%. De hecho, esta clase de equivalencia tiene la misma diversidad que una clase donde se encuentre 1 registro positivo y 49 negativos, aunque las dos clases de equivalencia tienen niveles de privacidad totalmente diferentes.

b) Ataque de la semejanza:

El problema se da cuando los valores del atributo sensible en una clase de equivalencia son diferentes, pero semánticamente similares. Debido a esto, un adversario puede conocer información importante y sensible acerca de algún individuo. Un ejemplo de este ataque se ve en la Tabla 16, que muestra una tabla que cumple el requisito de ℓ-diversidad con un valor de $\ell = 3$, compuesta por dos atributos sensibles: “Salario” y “Enfermedad”.

Edad	Código Postal	Problema
40	0538*	úlcera gástrica
40	0538*	gastritis
40	0538*	cáncer de estomago
≥ 50	0647*	gastritis
≥ 50	0647*	gripe
≥ 50	0647*	bronquitis
30	0539*	bronquitis
30	0539*	neumonía
30	0539*	cáncer de estomago

Tabla 16: Tabla propensa al ataque de la semejanza

Supongamos que el atacante sabe que el registro de un individuo está en la primera clase de equivalencia, por lo tanto sabe que el salario de la persona está en el rango de $[3K - 5k]$ y se puede inferir que su salario es relativamente bajo en comparación al resto de los individuos.

Por otro lado, este ataque también se puede realizar en los atributos categóricos, como es el caso de "Enfermedad". Al conocer que el individuo está en la primera clase de equivalencia, se puede inferir que tiene un problema al estómago, ya que las tres enfermedades de dicha clase de equivalencia corresponden a algún tipo de problema al estómago.

Estos problemas ocurren porque la ℓ-diversidad asegura "diversidad" de los valores sensibles que están dentro de cada clase de equivalencia, pero no toma en cuenta la cercanía semántica de esos valores.

Así es como se concluye el desarrollo de esta nueva técnica, teniendo en cuenta que distribuciones que tienen el mismo grado de diversidad pueden proporcionar niveles muy diferentes de privacidad, ya que existen relaciones semánticas entre cada uno de los valores del atributo sensible, porque distintos valores tienen diferentes niveles de sensibilidad y porque la privacidad también es afectada por la relación con la distribución general.

2.3. *t*-cercanía.

Recordando la noción de "Bayes-Optimal Privacy", la privacidad puede ser medida por la información que gana un adversario luego de ver una tabla liberada, es decir, por la diferencia entre la creencia previa y la creencia posterior del atacante. En base a esto, *t*-closeness (que a lo largo de este informe se traducirá como *t*-cercanía) agrega una nueva etapa en el proceso de ganancia u obtención extra de información del adversario, ya que no solo considera dicha ganancia sobre un individuo en específico, sino que también la obtención extra de información acerca de toda la población de datos que contiene la tabla.

Con el objetivo de explicar lo que se definió anteriormente, se propone un pequeño ejemplo (Li, Li, & Venkatasubramanian, 2007): en primer lugar, el atacante tiene una creencia previa sobre el atributo sensible de algún individuo que forma parte de los registros de la tabla, la que se denomina B_0, luego, en una situación hipotética, se le proporciona una tabla cuyos cuasi identificadores se encuentran totalmente generalizados (o completamente suprimidos), lo que significa que dicha tabla solo contendrá información en los campos de los atributos sensibles, esto permite que el atacante aprenda la distribución del valor del atributo sensible con respecto a toda la tabla, denominada por Q, obteniendo así la creencia denominada B_1. Posteriormente, se le hace entrega de la tabla liberada, la cual contiene los valores de los cuasi identificadores generalizados de acuerdo a los requerimientos de privacidad planteados para ese caso. De este modo, el adversario, teniendo en cuenta B_0, es capaz de identificar la clase de equivalencia en la cual se encuentra el registro que busca, aprendiendo así, la distribución P del atributo sensible en dicha clase de equivalencia. Es así como la creencia del adversario cambia a B_2, o creencia posterior.

Recordando, en la técnica de ℓ-diversidad, se limitaba la diferencia entre B_0 y B_2. Por el contrario, en la *t*-cercanía, se escogió cambiar este procedimiento y optar por la limitación de la diferencia entre B_1 y B_2, ya que se desea limitar la ganancia de información sobre individuos en particular, y no sobre toda la población contenida en la tabla.

Con el propósito de motivar esta decisión, se plantea que la distribución del atributo sensible en toda la tabla, Q, debe ser publica, ya que de alguna u otra forma, la distribución Q será liberada independientemente de la generalización que se le apliquen a los datos, siendo esta distribución, la que hace útil a la información. Además, un gran cambio entre B_0 y B_1 significa que la tabla contenía mucha información nueva, en otras palabras, corrige alguna creencia previa que estaba equivocada. Es así como, de cierta forma, mientras más amplia sea esta diferencia, más valiosa será la información, y como la ganancia de información entre B_0 y B_1 es respecto a toda la población, no es necesario limitarla. Entonces, lo que se desea limitar es la divergencia entre B_1 y B_2, lo cual se puede conseguir limitando la distancia entre las distribuciones P y Q. De acuerdo a esto, si las distribuciones P y Q son iguales, no hay aprendizaje adicional sobre la información. Basándose en esta proposición se puede inferir que mientras mayor sea la diferencia entre ambas distribuciones, menor será la privacidad de la información y mayor la utilidad de la misma. En cambio, si P y Q no son distantes, menor será la utilidad de la información y mayor la privacidad de los datos.

De esta forma se obtiene el principio de la *t*-cercanía (Li, Li, & Venkatasubramanian, 2007), el cual propone que una clase de equivalencia cumple con el requisito de *t*-cercanía si la distancia entre la distribución de un atributo sensible en la clase de equivalencia y la distribución del mismo atributo en toda la tabla, no supera un umbral *t*. Además, una tabla posee *t*-cercanía si todas las clases de equivalencia cumplen la *t*-cercanía. Por ende, se puede

concluir que el valor del parámetro "*t*" es el que permite variar los niveles de privacidad y utilidad de los datos.

Gracias a las definiciones anteriores, se procede a explicar el proceso de calcular la distancia entre dos distribuciones.

2.3.1. ¿Cómo calcular la distancia entre las distribuciones?

Para alcanzar el objetivo que propone el principio de *t*-cercanía, es indispensable realizar una medición de la distancia entre ambas distribuciones probabilísticas (P y Q). Para ello, dentro de las numerosas opciones disponibles, se escogió el uso del método llamado "Earth Mover's Distance", o simplemente, EMD, el cual es básicamente un problema de transporte de Monge-Kantorovich (Rubner, Tomasi, & Guibas, 2000). La elección de este método se debe a que refleja la distancia semántica entre los valores, es decir, considera que tan similares o cercanos son los valores (en atributos numéricos) o que tan parecidos son los significados (en el caso de atributos categóricos) entre sí. Además, EMD cuenta con una propiedad interesante, la cual dice que si la distancia entre cualquiera de los elementos del dominio está normalizada, es decir, entre 0 y 1, entonces al calcular el EMD entre dos distribuciones este resultado siempre estará entre 0 y 1, obteniendo así un rango entre el cual se puede elegir el valor de T (Li, Li, & Venkatasubramanian, 2007).

Con el propósito de llevar a cabo este método, se procederá a explicar el procedimiento para calcular la EMD, comenzando por los atributos de tipo numérico y luego para los atributos categóricos:

a) Cálculo de EMD para atributos numéricos:

Para este tipo de atributos se utiliza la denominada “Distancia ordenada”, la cual, en primera instancia, requiere que los valores de este atributo estén ordenados de forma descendente, tanto en la distribución de los valores del atributo sensible de la clase de equivalencia P como en la distribución de los valores sensibles de la tabla completa Q. A partir de esto, la fórmula para calcular la distancia entre dos distribuciones es:

$$D[P,Q] = \frac{1}{m-1}(|r_1| + |r_1 + r_2| + \ldots + |r_1 + r_2 + \ldots r_{m-1}|)$$

Donde, m es la cantidad de valores en Q; $r_i = p_i - q_i$, con $0 < i < m$.

A continuación, se proporciona un ejemplo sobre el cálculo de la distancia ordenada con EMD para la tabla de la Tabla 17, compuesta por los atributos cuasi identificadores “Código postal”, “Edad”; y los atributos sensibles “Problema” y “Sueldo”. Este último es el atributo numérico al cual se calculara su distancia ordenada con EMD:

Edad	Código Postal	Sueldo	Problema
40	0538*	3K	úlcera gástrica
40	0538*	5K	cáncer de estomago
40	0538*	9K	neumonía
≥ 50	0647*	6K	gastritis
≥ 50	0647*	11K	gripe
≥ 50	0647*	8K	bronquitis
40	0539*	4K	gastritis
40	0539*	7K	bronquitis
40	0539*	10K	cáncer de estomago

Tabla 17: Tabla para cálculo de EMD

Se tiene que $P_1 = \{3, 5, 9\}$, $P_2 = \{6, 8, 11\}$, $P_3 = \{4, 7, 10\}$ y $Q = \{3, 4, 5, 6, 7, 8, 9, 10, 11\}$. Con estos datos, se calcula cada una de las distancias de las distribuciones de las clases de equivalencia con respecto a Q, $D_{[P,Q]}$, considerando que un flujo óptimo para calcular dichas

distancias es mover el $\frac{1}{9}$ de la masa de probabilidad entre esos valores y que sus posiciones (i) comienzan en 1. El proceso es el siguiente:

i. $P_1 = \{3, 5, 9\}$, entonces la distancia entre esta distribución y Q es:

$$D_{[P_1,Q]} = \frac{1}{9} \cdot \frac{|r_1| + |r_1 + r_2| + \ldots + |r_1 + r_2 + \ldots r_{m-1}|}{m-1}$$

$$= \frac{|p_1 - q_1| + |p_1 - q_2| + |p_1 - q_3| + |p_3 - q_4| + |p_3 - q_5| + |p_3 - q_6| + |p_7 - q_7| + |p_7 - q_8| + |p_7 - q_9|}{9(m-1)}$$

$$= \frac{|1-1| + |1-2| + |1-3| + |3-4| + |3-5| + |3-6| + |7-7| + |7-8| + |7-9|}{9(9-1)}$$

$$D_{[P_1,Q]} = \frac{12}{72} \approx 0.167$$

ii. Luego, la distancia entre $P_2 = \{6, 8, 11\}$ y Q es:

$$= \frac{|p_4 - q_1| + |p_4 - q_2| + |p_4 - q_3| + |p_6 - q_4| + |p_6 - q_5| + |p_6 - q_6| + |p_9 - q_7| + |p_9 - q_8| + |p_9 - q_9|}{9(m-1)}$$

$$= \frac{|4-1| + |4-2| + |4-3| + |6-4| + |6-5| + |6-6| + |9-7| + |9-8| + |9-9|}{9(9-1)}$$

$$D_{[P_2,Q]} = \frac{12}{72} \approx 0.167$$

iii. Finalmente, $D_{[P_3,Q]}$, con $P_2 = \{4, 7, 10\}$, es igual a:

$$= \frac{|p_2 - q_1| + |p_2 - q_2| + |p_2 - q_3| + |p_5 - q_4| + |p_5 - q_5| + |p_5 - q_6| + |p_8 - q_7| + |p_8 - q_8| + |p_8 - q_9|}{9(m-1)}$$

$$= \frac{|2-1| + |2-2| + |2-3| + |5-4| + |5-5| + |5-6| + |8-7| + |8-8| + |8-9|}{9(9-1)}$$

$$D_{[P_3,Q]} = \frac{6}{72} \approx 0.083$$

Es así como, se dice que esta tabla cumple con el requisito de *0.167*-cercania con respecto al atributo “Salario”, ya que el parámetro “*t*” define el umbral de distancia entre las distribuciones, por lo tanto, se escoge la mayor distancia entre las obtenidas.

b) Cálculo de EMD para atributos categóricos:

Los atributos categóricos poseen una característica cualitativa, y pueden ser representados de acuerdo a una jerarquía. Dicha jerarquía es generada teniendo en cuenta la similitud, de tipo semántica, entre los valores de un atributo sensible. Es así como, para calcular la EMD de este tipo de atributos, se utiliza la llamada “Distancia jerárquica”, la cual está basada en el nivel mínimo en el cual, dos valores de un atributo, son generalizados al mismo valor, de acuerdo a la jerarquía del dominio de dicho atributo.

Con el fin de ejemplificar este método, se propone el siguiente caso, donde se considera una jerarquía (Figura 8), la cual está compuesta por cuatro niveles en el dominio de generalización, comenzando por el nivel 0 que contiene 11 nodos hoja, que representan los valores que puede tomar el atributo sensible “Problema”.

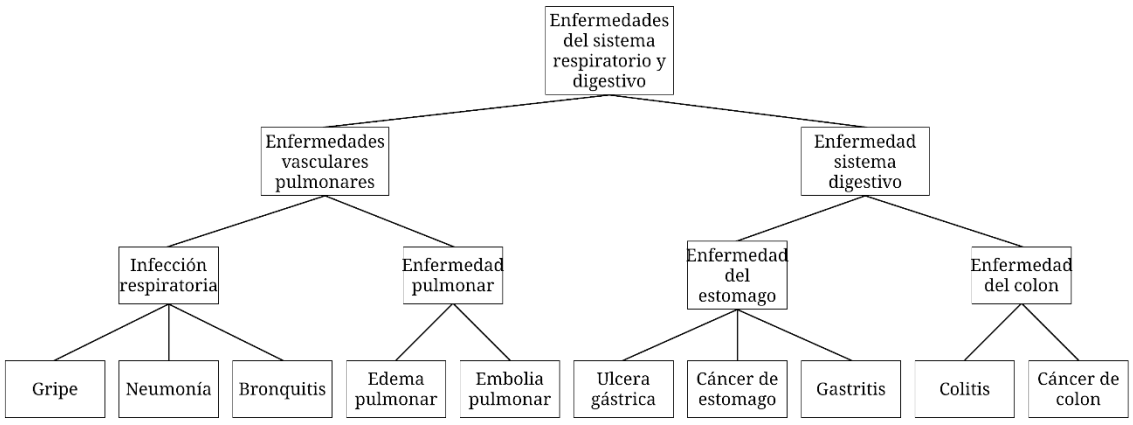


Figura 8: Jerarquía de generalización del atributo "Problema"

Dada la altura H del árbol de jerarquía de generalización, la distancia entre dos valores (v_1 y v_2) se obtiene con la siguiente formula:

$$D_{[v_1, v_2]} = \frac{nivel\ (v_1, v_2)}{H}$$

Donde $nivel\ (v_1, v_2)$ es el nivel mínimo donde las generalizaciones de dichos valores coinciden en el mismo valor. Es así como, la distancia entre Gripe y Edema pulmonar, considerando que el nodo en que coinciden sus generalizaciones corresponde a "Enfermedades vasculares pulmonares", es:

$$D_{[Gripe,\ \ Edema\ pulmonar]} = \frac{nivel\ (v_1, v_2)}{H} = \frac{2}{3}$$

2.3.2. Limitaciones de la *t*-cercanía

La ultima técnica desarrollada en este documento, denominada *t*-closeness, protege la información y/o datos contenidos en una estructura (tabla) contra la revelación de atributos, en otras palabras, se puede decir que la *t*-cercanía distribuye la información de tal manera que hace más difícil el trabajo de encontrar un dato sensible sobre un individuo por parte de un adversario. Sin embargo, no protege los datos contra la revelación de identidad por si sola. Es por este motivo, se recomienda la utilización de este algoritmo en conjunto con la técnica de k-anonimato.

2.4. Comparación de las técnicas de anonimato

La primera comparación está basada en la revelación de información donde se debe considerar los siguientes tipos:

a) Revelación de identidad: ocurre cuando un individuo es vinculado a un registro en particular de una tabla liberada. Un ejemplo seria cuando a partir del nombre y el RUN se puede identificar directamente la tupla de un individuo en la tabla.
b) Revelación de atributos: ocurre cuando se puede obtener información nueva acerca de algún individuo perteneciente a la tabla liberada. En otras palabras, la información liberada hace posible inferir características del individuo de forma más precisa de lo que sería sin observar esta tabla.

	Revelación de información	
	Protege Revelación de identidad	Protege Revelación de atributos
$\boldsymbol{k}$-anonimato	✓	
$\boldsymbol{\ell}$-diversidad		✓
t-cercanía		✓

Tabla 18: Revelación de información

Debido al resultado de esta comparación, se puede concluir que en todos los casos es recomendable utilizar técnicas en conjunto, es decir, si se quiere proteger la información utilizando el algoritmo de ℓ-diversad, se recomienda hacerlo en conjunto de k-anonimato, así mismo para la técnica de t-cercanía. De esta forma se previenen ambos tipos de revelación de información y los problemas que esto conlleva.

Por otro lado, se realiza una comparación acerca de los ataques o falencias que poseen dichas técnicas que protegen la información. Los ataques que se mencionarán han sido creados a lo

largo del tiempo, es decir, a medida que surge un nuevo algoritmo, los ataques no tardan en ser creados. La siguiente tabla hace alusión a esta temática:

	k-anonimato	ℓ-diversidad	t-cercanía
Falencia del orden de tuplas	×		
Ataque de atributos no-cuasi identificadores	×		
Ataque de homogeneidad	×		
Ataque de conocimiento previo	×		
Ataque de la falta de simetría (oblicuidad)		×	
Ataque de la semejanza		×	

Tabla 19: Ataques y/o falencias que poseen las técnicas que protegen la información

3. Análisis y diseño del ambiente de simulación

3.1. Especificación de hardware y software

En esta parte del documento se detallara el hardware y software que se utilizara, realizando una diferencia en cuanto a la utilización de software para análisis y comparación sobre las técnicas de anonimato (k-anonimato, ℓ-diversidad y t-cercanía) y para el desarrollo de la aplicación web que permitirá a estudiantes interactuar con dichas técnicas.

Cabe mencionar que durante el desarrollo del proyecto se descubre la librería ARX, la cual implementa las técnicas de anonimato que son objeto de este estudio, por lo que se decide utilizar esta herramienta por la utilidad que proporciona. Además, se busca validar su funcionamiento, realizando pruebas que se pueden observar en el Anexo 01: Desarrollo, utilización y ejecución de ARX.

El hardware utilizado corresponde a dos laptops con diferentes especificaciones, las cuales se detallaran a continuación:

i. Notebook HP 14-BP003LA, con un procesador Intel Core i5 7200U (2500 MHz - 3100 MHz) y 8GB DDR4 de RAM.
ii. Notebook ASUS X55LB, con un procesador Intel Core i5 5200U (2200 MHz - 2700 MHz) con 8GB DDR3L (1600 MHz) de RAM.

Cabe mencionar que ambos equipos cuentan con el sistema operativo Windows, el primero con la última versión (Windows 10) y el segundo con la versión 8.1.

Por otro lado, el software que se utilizara para el proceso de análisis y comparación de las técnicas de anonimato es:

- IDE utilizado para el desarrollo: NetBeans en su versión 8.2
- API utilizada: ARX - Data Anonymization Tool versión 3.7.0
- Lenguaje de programación: JAVA 1.8

Así mismo, el conjunto de software utilizado para el desarrollo de la aplicación web es el siguiente:

- IDE para el desarrollo: IntelliJ IDEA 2018.1.5 PRO con licencia JetBrains Product Pack for Students, con Apache Tomcat como servidor local.
- Frameworks utilizados:
    - Spring Boot, en su versión 2.0.3
    - Bootstrap 4.1
    - JQuery 3.2.1 y Modernizr 2.8.3
- Template Front End CoolAdmin 1.0.0 de ColorLib
- Thymeleaf como motor de plantillas HTML
- Lenguaje de programación: Java, version 1.8
- Control de versiones en plataforma Github.
- Cliente git: GitKraken Pro 3.6.6
- API utilizada: ARX - Data Anonymization Tool versión 3.7.0

Además, es bueno insistir que cierto software fue utilizado con licencia “PRO” obtenidas por ser estudiantes del área informática, lo cual permite la utilización de funcionalidades extra.

3.2. Aplicación desarrollada en lenguaje Java

Para efectos de la ejecución de los algoritmos y la obtención de los datos necesarios para aplicar las métricas, se desarrolló una aplicación en lenguaje Java, cuya estructura es similar a la que se explica en el Anexo 01: Desarrollo, utilización y ejecución de ARX, con la particularidad de que la base de datos y las jerarquías se cargan desde un archivo .CSV, a diferencia de lo que se realiza en el anexo, donde los datos son cargados a partir de una base de datos MySql y las jerarquías son generadas utilizando el "builder de ARX". A continuación, se detallaran los principales métodos implementados para la obtención de datos que serán analizados posteriormente.

Métodos	Parámetros de entrada	Descripción
private String getTime( )	Null	Retorna el tiempo necesario para anonimizar en milisegundos.
private String getLoss( )	ARXResult result	Retorna el porcentaje de perdida de información del resultado.
private void getOptimumGenLevel( )	Data data, ARXResult result	Muestra en pantalla los niveles de generalización de los cuasi-identificadores en el resultado.
private String getEquivalenceClassStatistics( )	Null	Muestra en pantalla las estadísticas asociadas a las clases de equivalencia.
private void getRisk( )	DataHandle handle	Muestra en pantalla el riesgo promedio y máximo de re-identificación, así como las tuplas en riesgo.
private int getAvgInputAge( )	DataHandle handle	Retorna el promedio de edad de los datos reales.
private int getAvgAnonAgeInf( )	DataHandle handle, int supressedRows	Retorna el promedio de edad de la tabla anonimizada con el límite inferior del intervalo.
private int getAvgAnonAgeAvg( )	DataHandle handle, int supressedRows	Retorna el promedio de edad de la tabla anonimizada con el promedio del intervalo.

Private int getMaxAgeInput( )	DataHandle handle	Retorna el valor máximo de edad en los datos reales.
Private int getMaxAgeAnon( )	DataHandle handle	Retorna el valor máximo de edad en los datos anonimizados.
Private int getMinAgeInput( )	DataHandle handle	Retorna el valor mínimo de edad en los datos reales.
Private int getMinAgeAnon( )	DataHandle handle	Retorna el valor mínimo de edad en los datos anonimizados.
Private int[] getModeAgeInput( )	DataHandle handle	Retorna un array con el valor de moda y la cantidad de veces que aparece en los datos reales.
Private int[] getModeAgeAnon( )	DataHandle handle	Retorna un array con el valor de moda y la cantidad de veces que aparece en la tabla anonimizada.
Private String getPrecent( )	Double value	Retorna el valor ingresado en forma de porcentaje.

Tabla 20: Métodos principales de la aplicación

3.2.1. Utilización de la aplicación

El programa no consta de una interfaz gráfica, por lo que su ejecución se realiza a través de la línea de comandos. En este caso se ejecutó en la terminal de Windows (cmd) con las siguientes instrucciones:

Desde la terminal, se debe dirigir al directorio en el cual estén almacenados los archivos libarx-3.7.0.jar, Pruebas.java y la carpeta “data”. Para compilar los códigos fuente y la librería externa en formato .jar se debe utilizar el siguiente comando:

$$\$\, javac - cp *; .Pruebas.java$$

Luego de compilar el programa, es posible ejecutar con:

$$\$\, java - cp *; .Pruebas$$

Al ejecutar, se desplegará un menú, el cual contendrá las tres opciones disponibles, cada una haciendo referencia a una de las técnicas implementadas. Se deberá ingresar la opción "1" para ejecutar el k-anonimato; la opción "2" para ejecutar la k-anonimato, (c, ℓ)-diversidad recursiva; y por último, la opción "3" para ejecutar la *t*-cercanía.

Luego de haber ingresado la opción que se requiera, se solicitara ingresar los parámetros necesarios para el funcionamiento de cada técnica, los cuales son:

a) Para ejecutar el k-anonimato: valor de k.
b) Para ejecutar la (c, ℓ)-diversidad recursiva: valor de $k, c\ y\ \ell$.
c) Para ejecutar la *t*-cercanía: valor de $k\ y\ t$.

Además, cada uno de estos valores tienen sus respectivas restricciones especificadas al momento de la ejecución. Finalmente, se presenta la opción de visualizar la tabla anonimizada resultante por pantalla, para lo cual se deberá ingresar "si" o "no", de acuerdo a lo que sea solicitado por el usuario.

Realizado este procedimiento, comenzara el proceso de anonimización y se mostraran los resultados obtenidos, los cuales serán mostrados por pantalla.

3.3. Aplicación web desarrollada

Con el propósito de generar una instancia en la cual puedan interactuar, tanto alumnos como docentes interesados en el estudio sobre la seguridad y privacidad de datos e información, de una forma más directa con las técnicas que han sido objeto de este estudio, se implementa un sistema web que permite conocer las principales características de las

técnicas de anonimato, observar ejemplos de cada una de ellas y ejecutarlas generando resultados y estadísticas.

Ya que se dispone de la librería ARX, la cual esta implementada en lenguaje Java, es requerido un "web framework" que acepte este lenguaje en el lado del servidor, y que permite la utilización de esta librería en un proyecto. Luego de investigar este tipo de frameworks, se selecciona Spring Boot para el desarrollo de la aplicación web.

En una primera instancia, se procede a interactuar con Spring Boot para conocer y comprender su funcionamiento, realizando diversas configuraciones y pruebas. Posteriormente, se consigue la inclusión de la librería ARX como dependencia de un proyecto, confirmando así la viabilidad de la utilización de este framework en conjunto con la API.

Luego de realizar las configuraciones y desarrollo en la parte del servidor, se procede a investigar la gestión del diseño con el cual el usuario debe interactuar, es así como se decide utilizar un motor de plantillas Java denominado "Thymeleaf", que posee módulos para Spring Framework facilitando el desarrollo front-end.

3.3.1. Spring Boot MVC (model-view-controller)

Para el desarrollo de la aplicación web se utiliza un patrón modelo-vista-controlador (MVC), donde el modelo es una representación de los datos que serán manejados; la vista es la interfaz gráfica con la que interactúa un usuario; y el controlador tiene la función de manejar los datos, de acuerdo a las acciones que realice el usuario considerando los modelos

definidos. En la aplicación con la cual se permite la interacción con las distintas técnicas de anonimato, la implementación de cada elemento de este patrón es la siguiente:

a) Modelos: este componente representa la estructura de la información que será enviada desde el controlador a la vista. Cada modelo se define como una clase de Java, la cual contiene determinados atributos junto a sus constructores correspondientes. Para propósito de esta aplicación, son creados los siguientes modelos:
    i. Persona_kanonimato, Persona_ldiversidad y Persona_tcercania: son utilizados para estructurar las tuplas de la base de datos de entrada y de la base de datos anonimizada, para ser enviadas a la vista de ejemplos correspondiente (k-anonimato, ℓ-diversidad o t-cercanía).
    ii. Jerarquías: es utilizado para estructurar los datos correspondientes a la jerarquía de generalización de cada atributo.
    iii. Adult_dataset: se usa para aplicar un formato a las tuplas de la base de datos que será utilizada para llevar a cabo los experimentos desde la aplicación.
    iv. Chart_data: será utilizado para estructurar la información sobre las estadísticas obtenidas al ejecutar cualquiera de los algoritmos de anonimato, como es la perdida de información, riesgo de re-identificación, tuplas suprimidas, entre otros.
b) Vistas: conjunto de archivos HTML gestionados por el motor de plantillas Thymeleaf, a través de los cuales el usuario interactúa con la aplicación, la cual consta de las siguientes:
    i. Inicio: vista inicial, la cual introduce la aplicación de forma general.

ii. Algoritmos: se desarrolla esta vista con el propósito de introducir el tema acerca de las distintas técnicas existentes de anonimato, a través de una línea de tiempo. Desde esta sección se desprenden tres vistas específicas, cada una de ellas asociada a una técnica de anonimato, donde se proporciona una definición formal, características y ejemplos. Las vistas son: k-anonimato, ℓ-diversidad y t-cercanía.

iii. Ejecución de pruebas: vista con la cual el usuario puede interactuar con las distintas técnicas, aplicándolas sobre una base de datos, seleccionando el algoritmo a ejecutar con sus respectivos parámetros. Además, se muestra información asociada a la tabla resultante junto a sus respectivas estadísticas, algunas de las cuales serán representadas en forma de gráfico.

iv. Estadísticas: se presenta una serie de gráficos para una amplia gama de datos estadísticos, con el propósito de visualizar los resultados que se obtendrán en la etapa de experimentos.

c) Controladores: son un conjunto de clases en lenguaje Java, las cuales contienen diversos métodos ejecutados de acuerdo a la dirección (URL) que sea solicitada por el usuario. Su función principal es la ejecución de los algoritmos asociados a cada técnica de anonimato, haciendo uso de la librería ARX. Además, se encargan de retornar la vista junto a su información correspondiente.

3.3.2. Adquisición y configuración del servidor

Con el propósito de adquirir conocimiento sobre la implementación del proyecto realizado en Spring Boot en un servidor, se adquiere uno de ellos a través de la plataforma “DigitalOcean” sin costos asociados, haciendo uso del beneficio entregado por “GitHub”

hacia estudiantes (education.guthub.com/pack). Dicho servidor cuenta con las siguientes características: 1GB de RAM, 25 GB de disco, sistema operativo Ubuntu 16.04.4 x32, ubicado en Estados Unidos.

Luego de obtener el servidor, se revisan los requerimientos para la implementación del proyecto. Es por esto que se realiza la instalación de Java y ApacheTomcat, siendo esto todo lo necesario para la ejecución del proyecto en el servidor.

4. Experimentos

Luego del desarrollo y presentación del ambiente de simulación, se definen ciertas características que presentarán los experimentos, antes de ser ejecutados. Entre los ítems que se desarrollaran se encuentran los temas relacionados a la preparación de los experimentos, destacando aspectos propios de la base de datos que se utilizara, como su procedencia, contenido, tamaño, entre otros. Además, se realizara una descripción de los métodos e instancias que se escogieron para alcanzar los requisitos que impone cada técnica de protección de datos, así como los parámetros que serán aplicados a cada una de ellas, explicando los fundamentos y las bases que conllevan a dichas elecciones.

Por otro lado se presentaran las métricas que serán consideradas al momento de evaluar los resultados que se obtendrán con la ejecución de ciertos experimentos, las cuales serán categorizadas en dos grupos: el primero contendrá las métricas que se obtendrán a partir de la ejecución de métodos que retornan datos acerca de la perdida de información, riesgo de re-identificación, entre otros, las cuales consideran la tabla resultante en su totalidad; el segundo grupo consiste en métricas asociadas a un solo atributo de la tabla resultante (Edad), las que están estrechamente relacionadas con consultas de agregación como es el mínimo, máximo, promedio de valores, entre otros.

Finalmente, con la intención de detallar el proceso que realiza la aplicación para proveer los datos solicitados, se presentaran un conjunto de funciones que han sido implementadas, añadiendo los parámetros de entrada de cada una de ellas y una breve descripción. Además se proporcionan ciertas consideraciones que se deben tener en cuenta al momento de ejecutar dicha aplicación.

4.1. Preparación

El componente principal para la aplicación de cualquier técnica de protección de información es la base de datos. Es por esto que se proporcionara la información necesaria para comprender el tipo de información que buscara ser anonimizada.

4.1.1. Elección y composición de la base de datos

Cada uno de los algoritmos buscan proteger la información contenida en una base de datos, por lo tanto es necesario adquirir una de ellas y asegurar que cumpla ciertos requisitos, como por ejemplo, que posea una cantidad considerable de registros, que la información almacenada en ella este alineada al problema que se presentará, y que los atributos y/o valores de los mismos sean aptos para ser anonimizados. Basándose en estos requisitos, se opta por utilizar una base de datos facilitada por los mismos creadores de la librería ARX, ya que, al ser utilizada para el desarrollo y para la aplicación de prueba de la librería mencionada, es seguro que cumpla con los requisitos planteados.

La base de datos seleccionada está compuesta por información del Censo en Estados Unidos realizado en el año 1994, la cual contiene 30.162 registros y está almacenada en una archivo con extensión .CSV (Comma Separated Values) con un tamaño de 2.52 MB. Cada registro perteneciente a esta base de datos se caracteriza por nueve atributos o columnas, los cuales son: Género (sex), Edad (age), Raza (race), Estado marital (marital-status), Educación (education), País de nacimiento (native-country), Clase de trabajo (workclass), Ocupación (occupation) y Rango de sueldo (salary-class). De este conjunto, el atributo “Edad” es el único de tipo numérico, ya que los demás son atributos categóricos.

Además, se indica que la tabla consta de un atributo que es considerado sensible, la columna “Ocupación”, y de un conjunto de ocho cuasi identificadores. Por otro lado, cabe mencionar

que, se dispone de cada una de las jerarquías para dichos atributos, las cuales serán detalladas más adelante.

	sex	age	race	marital-satatus	education	native-country	workclass	salary-class	occupation
Tipo del valor del atributo	categórico	numérico	categórico	categórico	categórico	categórico	categórico	categórico	categórico
Tipo de atributo	Cuasi identificador	Cuasi identificador	Cuasi identificador	Cuasi identificador	Cuasi identificador	Cuasi identificador	Cuasi identificador	Cuasi identificador	Sensible
Altura de jerarquía	2	5	2	3	4	3	3	2	3

Tabla 21: Resumen de la composición de la base de datos

4.1.2. Técnicas, atributos y parámetros que se utilizaran durante la ejecución

Es necesario detallar el proceso o instancia que se utilizara de cada una de las técnicas, ya que en el estudio del arte se desarrollaron varias formas de alcanzar el requisito de anonimato que impone cada una de ellas. Además, en esta sección se explicaran algunos de los componentes necesarios para llevar a cabo dichas instancias o procesos, como son los tipos de atributos con sus jerarquías respectivas y los conjuntos cuasi identificadores. Por último se presentaran una serie de parámetros que se utilizaran para posterior ejecución de los algoritmos.

4.1.2.1. Definición de técnicas y tipos de atributos

Para comenzar, la generalización y supresión de datos son los procesos que serán utilizados con el fin de alcanzar el requisito de k-anonimato, considerando que cada uno de los atributos de la base de datos posee su respectiva jerarquía de generalización previamente definida. A continuación, se presentaran las jerarquías de generalización asociadas a cada atributo, las cuales serán utilizadas de acuerdo a la técnica que se ejecutará:

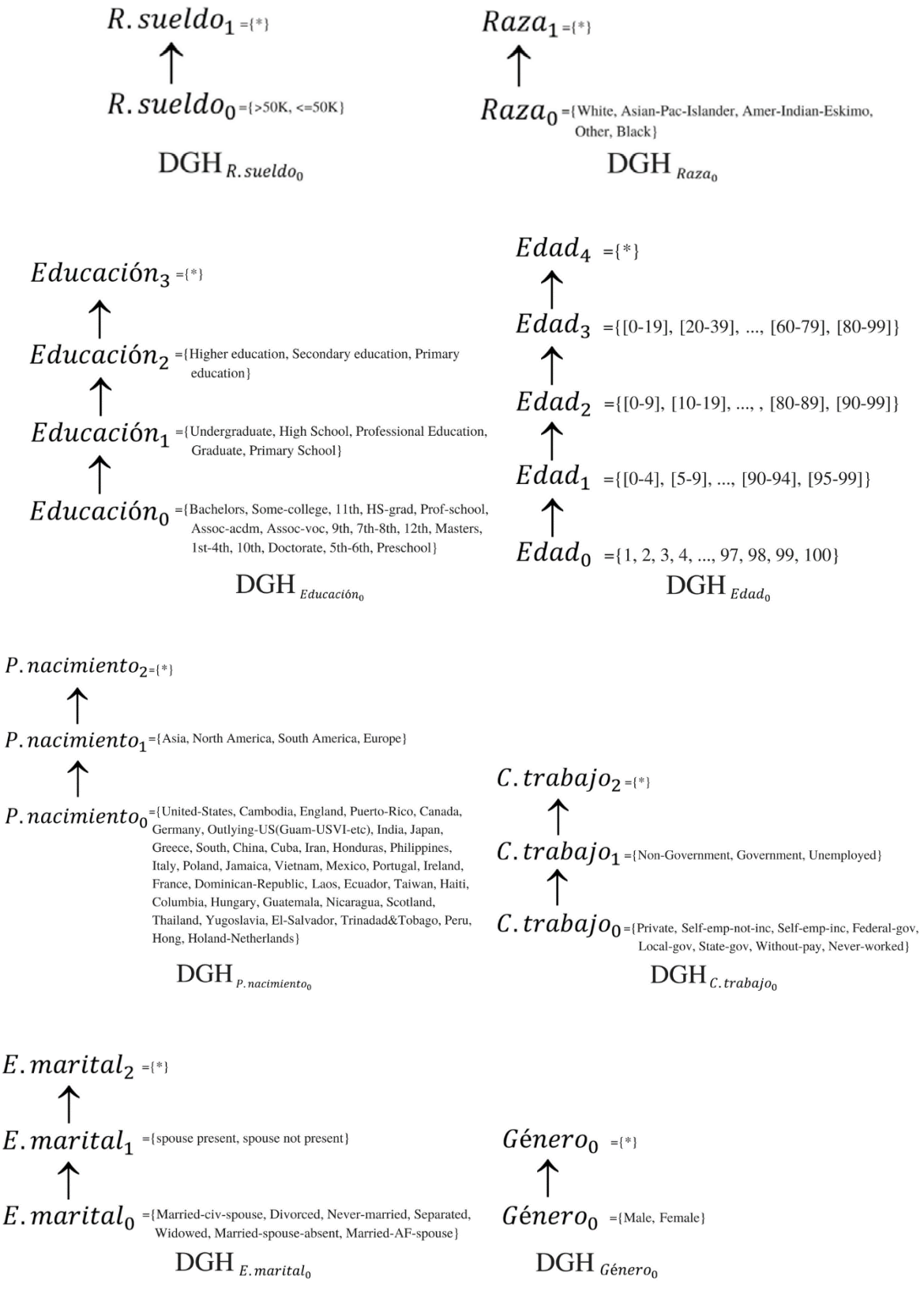


Figura 9: Jerarquías de generalización de dominio (conjunto cuasi identificador)

Además, para la ejecución del algoritmo de k-anonimato se define el siguiente conjunto de atributos cuasi identificadores: $QI_T = \{edad, sueldo, raza, estado\ marital, educación, país\ de\ nacimiento, clase\ de\ trabajo, rango\ de\ sueldo\}$, y también un atributo que contiene la información que se desea proteger, el cual se definirá como un atributo de tipo "Insensible" (según los distintos tipos de atributos definidos por la herramienta ARX) (Prasser & Kohlmayer, 2015), $Insensible = \{Ocupación\}$, el que no se considerara para el proceso de anonimización de esta técnica.

Por otro lado, en el caso de la ℓ-diversad y *t*-cercanía, donde existen más de una instancia para alcanzar sus respectivos requisitos de anonimato, se escogió un método en particular para ejecutar los algoritmos. Así es como, para la técnica de ℓ-diversad se opta por utilizar la instancia llamada (c, ℓ)-diversidad recursiva, ya que es la que se utiliza comúnmente (Prasser, Kohlmayer, & Kuhn, 2014), y al tratarse de la última de las definiciones de dicha técnica, se estima que entrega los mejores resultados al corregir algunas falencias de las instancias anteriormente creadas. Por otra parte, en la técnica de *t*-cercanía, se utilizara el método de la distancia jerárquica de EMD, ya que los valores del atributo sensible de la base de datos son del tipo categórico, y es necesario considerar la relación semántica que existe entre ellos.

Al igual que para la ejecución del algoritmo de k-anonimato, en ℓ-diversad se presenta el mismo conjunto de atributos cuasi identificadores, sin embargo, en esta técnica se define el siguiente atributo sensible: $Sensible = \{Ocupación\}$, el cual será considerado con la finalidad de obtener la diversidad en la distribución del mismo. Finalmente, para aplicar la técnica de *t*-cercanía, se define el conjunto de atributos cuasi identificadores (el mismo que para las técnicas anteriores) y, al igual que en ℓ-diversad, el mismo atributo sensible. La

diferencia de esta última técnica respecto a las demás, es que se requiere la definición de una jerarquía de generalización para dicho atributo sensible, la cual se presenta a continuación:

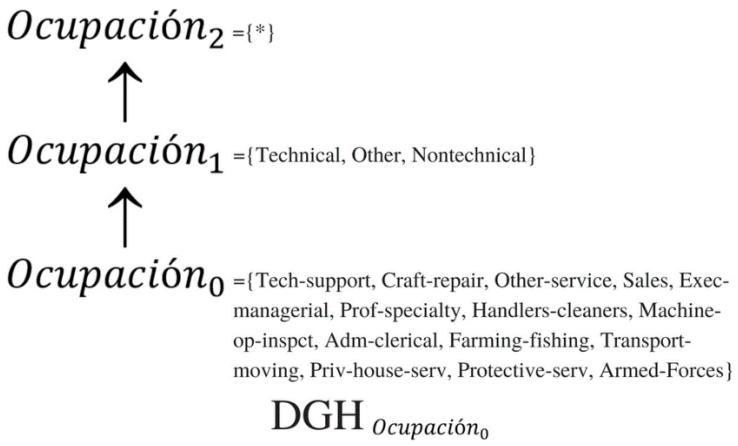


Figura 10: Jerarquía de generalización de dominio (atributo sensible)

4.1.2.2. Definición de parámetros de configuración para las técnicas

Antes de definir concretamente los distintos valores para cada parámetro que utilizan las técnicas, es bueno mencionar que se agruparan en dos categorías: la primera contendrá valores comúnmente utilizados en la ejecución de estos algoritmos; y la otra categoría presenta una mayor cantidad de valores con el fin de obtener resultados considerables.

Para una primera instancia de pruebas, los parámetros que serán utilizados son los más comunes en la práctica (Prasser, Kohlmayer, & Kuhn, 2014), lo cuales son: $k = 5$, para k-anonimato; $c = 4$ y $\ell = 3$, para ℓ-diversad; y, $t = 0.2$, para t-cercanía.

Posteriormente, para obtener un mayor espectro de resultados, se realizarán pruebas con distintos valores para cada uno de parámetros anteriormente mencionados. Es así como los valores que se asignaran para el parámetro "k" van desde el 2 hasta el 25, siendo 2 el mínimo

valor posible por definición de la técnica y 25 un valor considerablemente alto. Para ℓ-diversad, el valor del parámetro "ℓ" podrá variar entre 2 hasta 12, asimismo el parámetro "c". Por último, en la técnica de t-cercanía, el valor de su parámetro t podrá ser cualquiera de los que se presentan en el siguiente conjunto: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$.

Además, las pruebas serán realizadas sin establecer un límite de supresión de tuplas, ya que dicho proceso afecta la utilidad del resultado que se busca obtener, considerando que con un menor límite de supresión, el algoritmo deberá realizar más generalizaciones, provocando así, una gran pérdida de información. Por otro lado, de acuerdo a lo que se indica en la Tabla 18: Revelación de información, se decide aplicar tanto la técnica de ℓ-diversad como la de t-cercanía en conjunto con k-anonimato, debido a que al ser aplicadas individualmente, no protegen la información de la revelación de identidad, como si lo hace el k-anonimato. A continuación, se proporciona una tabla que resume la elección de todos los valores que podrán tomar los parámetros de cada técnica:

	k	c	ℓ	t
Valor utilizado comúnmente	5	4	3	0.2
Rango total de valores	{2, 5, 8, 12, 15, 17, 20, 25}	{2, 4, 6, 8, 10, 12}	{2, 4, 6, 8, 10, 12}	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

Tabla 22: Resumen de elección de parámetros de cada técnica

4.1.3. Métricas

Luego de haber realizado la ejecución de cada una de las técnicas con sus respectivas configuraciones y parámetros, se debe considerar que los resultados serán analizados respecto a una serie de métricas, las cuales se pueden categorizar en dos grupos: el primero corresponde a las métricas que son implementadas en la librería ARX, las cuales realizan el

proceso teniendo en cuenta la tabla resultante en su totalidad, considerando todas las columnas y tuplas que esta contiene. Estas métricas serán útiles para observar cómo afectan los parámetros de las técnicas (k, c, ℓ y t) al resultado anonimizado, principalmente en términos de su utilidad, considerando también otros aspectos, tales como el riesgo, la cantidad de tuplas suprimidas, perdida de información, entre otros. La segunda categoría abarca las métricas propuestas por nosotros en conjunto con el profesor guía, las cuales se aplicaran sobre los resultados de ciertas consultas de agregación considerando un atributo numérico, y serán de útiles para observar como varía el resultado anonimizado con respecto a los valores reales de la tabla original. El atributo en cuestión es "Edad", ya que como se mencionó anteriormente, es el único campo de tipo numérico que posee la base de datos.

Ambos grupos de métricas se detallan de la siguiente forma:

i. Métricas de ARX
    a. Perdida de información: analiza la pérdida de información total de la tabla anonimizada, entregando un resultado en términos de porcentaje. Se considera como pérdida la cantidad de generalizaciones de cada atributo perteneciente a la tabla, junto con la cantidad de tuplas suprimidas.
    b. Riesgo de re-identificación: entrega información sobre el porcentaje de riesgo de re-identificación promedio en la tabla resultante, considerando el escenario de que un atacante conoce que cierto individuo forma parte de los registros de la base de datos.
    c. Cantidad de tuplas suprimidas: total de tuplas que han sido eliminadas producto de la anonimización, con el fin de controlar el proceso de generalización.

d. Numero de clases de equivalencia: total de clases de equivalencia generadas producto de la ejecución de cada algoritmo de anonimato.
e. Tamaño promedio de las clases de equivalencia: cantidad promedio de tuplas que contiene cada una de las clases de equivalencias presentes en la tabla.

ii. Métricas aplicadas al atributo “Edad”

a. Mínimo: valor mínimo que presenta la tabla en el atributo edad.
b. Máximo: valor máximo que presenta la tabla en el atributo edad.
c. Promedio con límite inferior: representa el promedio de todos los valores de la columna edad. Considerando que el resultado anonimizado es entregado en términos de intervalos (por ejemplo: [15-25]), se utilizara el límite inferior de cada intervalo para calcular el promedio.
d. Promedio con promedio del intervalo: promedio de todos los valores de la columna edad. Considerando la acotación anterior, se utilizara el valor intermedio del intervalo, que se obtiene de la forma $(\lim inferior + \lim superior)/2$, para conseguir el valor promedio de la columna.
e. Valor de moda: representa el valor más frecuente de la columna edad.
f. Cantidad del valor de moda: informa la cantidad de veces que aparece el valor más frecuente de la columna edad.

4.2. Ejecución de técnicas de anonimato

Utilizando la aplicación en Java explicada anteriormente, se realizaron una serie de pruebas con la ejecución de distintas técnicas considerando los parámetro definidos en el

punto 4.1.2, obteniendo una amplia gama de resultados (ver Anexo 02: Datos de métricas) basados en las métricas anteriormente descritas (ver punto 4.1.3), de las cuales han sido seleccionadas (para el desarrollo de esta sección) las más representativas en cuanto a los efectos de estas técnicas sobre la utilidad y privacidad de los datos.

Dichos resultados serán presentados a continuación, en forma de gráficos, considerando que en cada uno de ellos el eje de las abscisas corresponde al valor que toma el parámetro de cada técnica; y el eje de las ordenadas muestra el valor, porcentaje o intervalo de cada métrica que se presentará.

i. Métrica asociada a la perdida de información

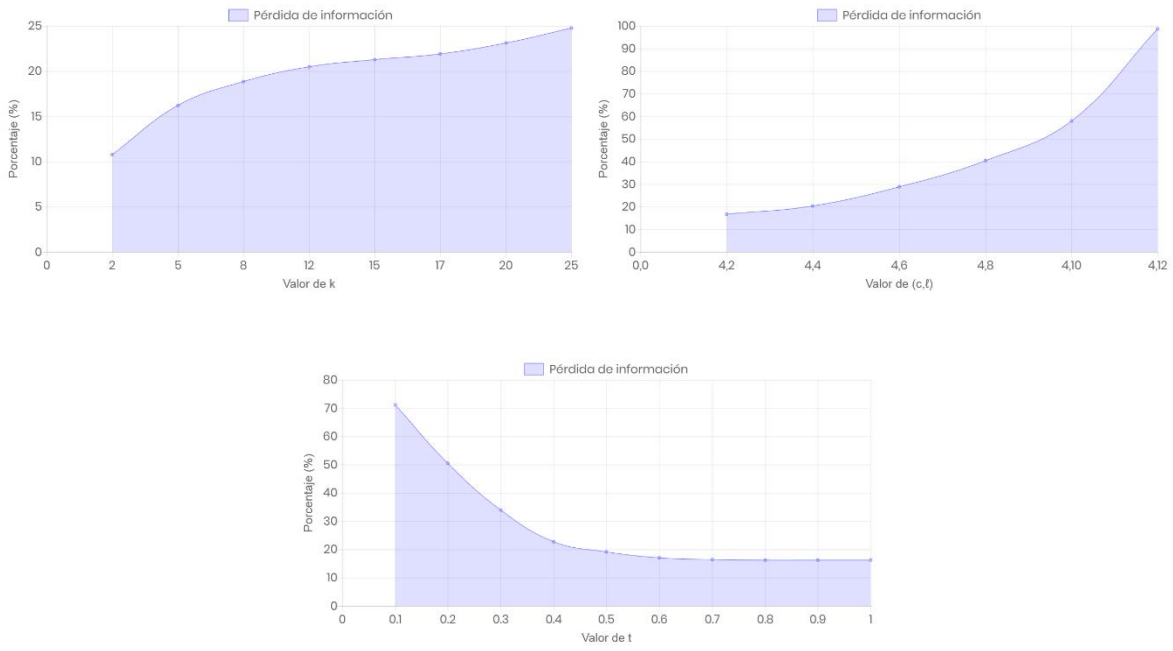


Figura 11: Pérdida de información v/s k, l, t

ii. Métrica asociada al riesgo de re-identificación

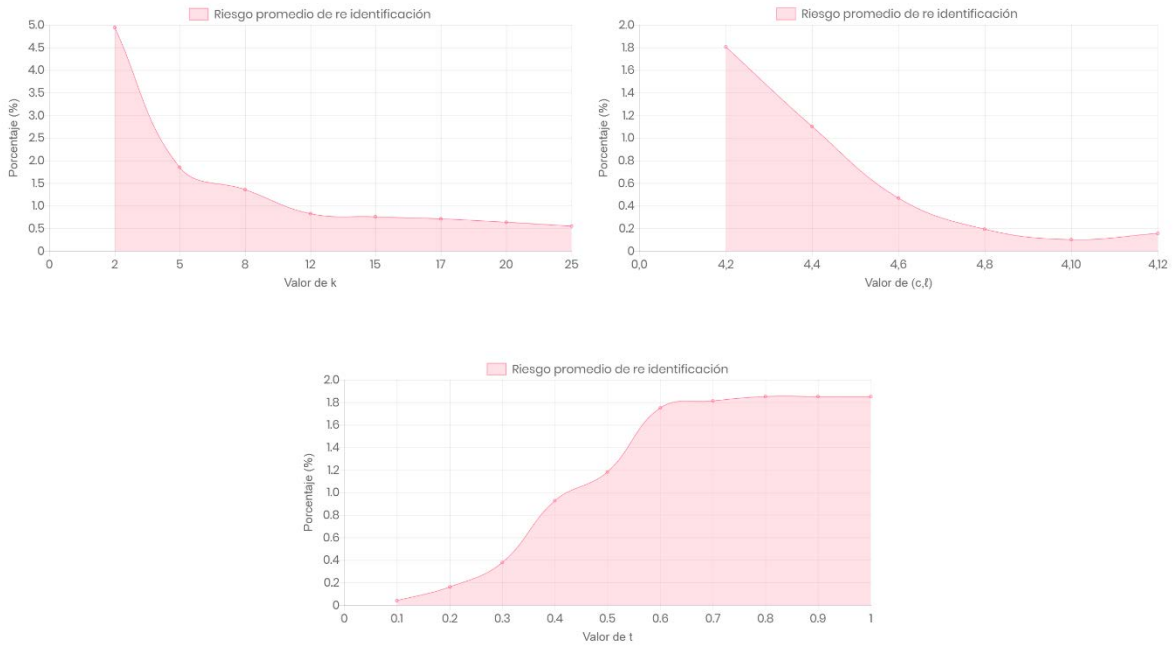


Figura 12: Riesgo de re-identificación v/s k, l, t

4.3. Comparaciones y conclusiones a partir de las métricas

Para comenzar, se realiza una breve verificación a las definiciones o inferencias que se pueden obtener de cada técnica desarrollada.

En primer lugar, la definición de la técnica de k-anonimato plantea que el valor del parámetro k determina la cantidad de tuplas con la misma combinación de valores en su conjunto cuasi identificador, de lo cual se desprende que si se aumenta el valor de este parámetro, aumenta la privacidad para los individuos contenidos en la tabla privada. Este comportamiento se logra verificar, ya que a medida que el valor del parámetro aumenta, existe una mayor pérdida de información y un menor riesgo de re-identificación, lo que implica una tabla resultante con mayor índice de privacidad. Así mismo, en base a los gráficos de ℓ-diversidad, se logra inferir el comportamiento del parámetro ℓ, el cual genera una tabla con un mayor índice de

privacidad a medida que dicho parámetro aumenta. Este comportamiento es el esperado, ya que ℓ define la diversidad de los valores del atributo sensible en cada una de las clases de equivalencia.

A diferencia del comportamiento de los parámetros en ambas técnicas mencionadas anteriormente, en el caso de la *t*-cercanía ocurre lo inverso. En esta técnica se observa que al aumentar el valor de *t* se obtiene un resultado con un menor índice de privacidad, es decir, que aumenta el riesgo de re-identificación ocasionado por una menor perdida de información.

Con el propósito de resumir las observaciones, se proporciona la siguiente tabla:

	Privacidad	Perdida de información	Riesgo de re-identificación
A mayor valor de k	Aumenta	Aumenta	Disminuye
A mayor valor de ℓ	Aumenta	Aumenta	Disminuye
A mayor valor de *t*	Disminuye	Disminuye	Aumenta

Tabla 23: Resumen de comportamiento de parámetros

A partir de los resultados estadísticos y observando las representaciones graficas es posible realizar una serie de comparaciones y conclusiones relacionadas a la utilidad de la información resultante (luego de aplicar una técnica de anonimización), basadas en las métricas de riesgo de re-identificación y perdida de información.

Una de las primeras observaciones se puede realizar a partir de los gráficos de riesgo obtenidos para cada técnica aplicada, gracias a los cuales se logra inferir que el k-anonimato es la técnica que presenta un mayor riesgo de re-identificación, la cual posee un riesgo máximo cercano al 5%, en comparación al 1.8% de riesgo obtenido desde las otras dos técnicas. Es así como se comprueba lo que plantea la literatura acerca de las técnicas de anonimato, que califican al k-anonimato como una técnica propensa a la re-identificación de individuos, y se refuerza la recomendación de aplicar dicho algoritmo en conjunto con ℓ-

diversidad o *t*-cercanía para obtener un mayor índice de privacidad (menor riesgo). Por otro lado, la técnica que entrego los números más bajos de riesgo fue la *t*-cercanía, con un valor mínimo cercano al 0%, sin embargo, se debe considerar que la perdida de información que se obtiene al conseguir este mínimo riesgo es notoriamente alta, llegando a un valor cercano al 70%, lo cual no resulta aceptable para ser aplicado de forma práctica.

En lo que se refiere a la perdida de información, es posible determinar que la ℓ-diversidad llega a un valor máximo de pérdida, alcanzando un índice muy cercano al 100% cuando $\ell = 12$, y aun así, presenta riesgo de re-identificación.

Teniendo en cuanta los parámetros que son utilizados comúnmente en los estudios prácticos aplicando dichas técnicas de anonimato (Prasser, Kohlmayer, & Kuhn, 2014), se logra corroborar la razón del uso de estos valores, ya que se obtiene una relación aceptable entre las métricas de perdida de información y riesgo de re-identificación, es por esto que se respalda la elección de dichos valores para las técnicas de k-anonimato y ℓ-diversidad. Sin embargo, se observa que en la técnica de *t*-cercanía el valor con la mejor relación entre estas dos métricas es $t = 0.3$, ya que la perdida de informacion disminuye considerablemente desde un 51% (con $t = 0.2$) a un 35%, mientras que el siego solo aumenta de un 0.18% (con $t = 0.2$) a un 0.4%, siendo este valor menor en comparación a las otras dos técnicas.

5. Conclusiones

En el desarrollo del presente proyecto se logró conocer a fondo las técnicas de k-anonimato, ℓ-diversidad y t-cercanía, de las cuales conseguimos aprender sobre sus características, funcionamiento de algoritmos, falencias y ataques, además de la motivación por la cual surgieron, lo que liga estrechamente a estas técnicas de protección de la privacidad de información.

En cuanto a la efectividad, cada una de las técnicas cumple con el propósito de anonimizar datos, lo que las convierte en procesos que pueden ser aplicados de forma práctica por parte de organizaciones reales que buscan alcanzar este objetivo. Sin embargo, se debe considerar que se obtiene un mejor resultado cuando el k-anonimato es utilizado en conjunto con alguna de las técnicas restantes, protegiendo la información tanto contra la revelación de atributos como contra la revelación de identidad.

Al garantizar la utilización practica de los algoritmos y considerando la complejidad de su enseñanza actualmente en la carrera Ingeniería Civil en Informática que imparte la Universidad de Bío-Bío, se desarrolla una aplicación web con el fin de facilitar la comprensión y aprendizaje, proporcionando ejemplos claros y permitiendo la ejecución de estas técnicas, obteniendo resultados concretos y estadísticas.

Con la ejecución de los algoritmos y habiendo definido ciertas métricas para el posterior análisis, tales como la perdida de información y el riesgo de re-identificación, se logró determinar cuánto afecta la aplicación de estas técnicas con respecto a la utilidad de la información, concluyendo que el k-anonimato es el algoritmo que presenta menor utilidad, ya que presenta índices elevados de perdida de información y de riesgo de re-identificación;

en contraste con la técnica de *t*-cercanía, la que presenta los mejores resultados de esta métrica, ocasionando una pérdida de información relativamente baja y un riesgo de re-identificación cercano al 0% para valores de $t = [0.3 - 0.4]$.

Finalmente, pese a nuestra creencia previa sobre el bajo desarrollo de aplicaciones prácticas de los algoritmos de anonimato, se descubre la existencia de una herramienta de código abierto, como es ARX, la cual presenta numerosos métodos para aplicar estas y otras técnicas. Esto posibilita la exploración de nuevas técnicas, algunas de ellas más actuales y con distintos enfoques, junto con la comparación de ellas teniendo en cuenta una gama más amplia de métricas disponibles en esta herramienta, las cuales no fueron objeto de este estudio, sin embargo, se considera interesante el aprendizaje y aplicación de las mismas.

6. Bibliografía

Bayardo , R., & Agrawal, R. (2005). Data privacy through optimal k-anonymization. *21st International Conference on Data Engineering (ICDE'05)*, 217-228. Obtenido de http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1410124&isnumber=30564

Emam, K., Dankar, F., Issa, R., Jonker, E., Amyot, D., Cogo, E., . . . Bottomley, J. (2009). A Globally Optimal k-Anonymity Method for the De-Identification of Health Data. *Journal of the American Medical Informatics Association, 16*, 670–682. Obtenido de https://doi.org/10.1197/jamia.M3144

LeFevre, K., DeWitt, D., & Ramakrishnan, R. (2005). Incognito: efficient full-domain K-anonymity. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 49-60. Obtenido de https://dl.acm.org/citation.cfm?id=1066164

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. *2007 IEEE 23rd International Conference on Data Engineering*, 106-115. Obtenido de http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4221659&isnumber=4221635

Machanavajjhala, A., Venkitasubramaniam, M., Kife, D., & Gehrke, J. (2006). ℓ-diversity: Privacy beyond k-anonymity. *22nd International Conference on Data Engineering (ICDE'06)(ICDE)*, 24. Obtenido de https://www.computer.org/csdl/proceedings/icde/2006/2570/00/01617392-abs.html

Prasser, F., & Kohlmayer, F. (Noviembre de 2015). *API*. Obtenido de Putting Statistical Disclosure Control Into Practice: The ARX Data Anonymization Tool: https://arx.deidentifier.org/development/api/

Prasser, F., Kohlmayer, F., & Kuhn, K. (2014). A Benchmark of Globally-Optimal Anonymization Methods for Biomedical Data. *2014 IEEE 27th International Symposium on Computer-Based Medical Systems*, 66-71. Obtenido de http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6881850&isnumber=6881826

Rubner, Y., Tomasi, C., & Guibas, L. (2000). The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision, 40*, 99–121. Obtenido de https://link.springer.com/article/10.1023/A:1026543900054

Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Technical Report SRI-CSL-98-04 SRI Computer Science Laboratory*. Obtenido de https://epic.org/privacy/reidentification/Samarati_Sweeney_paper.pdf

Stammler, S., Katzenbeisser, S., & Hamacher, K. (2016). Correcting Finite Sampling Issues in Entropy l-diversity. *Lecture Notes in Computer Science, 9867*, 135-146. Obtenido de https://doi.org/10.1007/978-3-319-45381-1_11

Sweeney, L. (2002). k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10*, 557-570. Obtenido de https://www.worldscientific.com/doi/abs/10.1142/S0218488502001648

7. Anexos

7.1. Anexo 01: Desarrollo, utilización y ejecución de ARX

En el presente apartado se listaran los pasos a seguir necesarios para utilizar la librería ARX en un proyecto propio en Java, utilizando el IDE NetBeans. Para propósito del estudio de estas funciones, se tuvo como base los ejemplos disponibles en el repositorio oficial de ARX en GitHub (github.com/arx-deidentifier), así como su documentación (Prasser & Kohlmayer, 2015). Además, para la validación del correcto funcionamiento de los métodos que proporciona esta librería, se optó por realizar algunos ejemplos de los que son mencionados y desarrollados en diversos artículos estudiados, en este caso, el siguiente ejemplo fue extraído desde el articulo llamado "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity" (Li, Li, & Venkatasubramanian, 2007):

	ZIP Code	Age	Disease
1	47677	29	Heart Disease
2	47602	22	Heart Disease
3	47678	27	Heart Disease
4	47905	43	Flu
5	47909	52	Heart Disease
6	47906	47	Cancer
7	47605	30	Heart Disease
8	47673	36	Cancer
9	47607	32	Cancer

Tabla 24: Tabla privada "Pacientes" (Anexo)

	ZIP Code	Age	Disease
1	476**	2*	Heart Disease
2	476**	2*	Heart Disease
3	476**	2*	Heart Disease
4	4790*	≥40	Flu
5	4790*	≥40	Heart Disease
6	4790*	≥40	Cancer
7	476**	3*	Heart Disease
8	476**	3*	Cancer
9	476**	3*	Cancer

Tabla 25: Versión 3-anonima de la tabla privada (anexo)

Luego se procede a implementar este ejemplo en el IDE, para ello se realiza la creación de una base de datos MySql en un servidor Ubuntu 16.04, implementando una tabla con los nueve registros que se aprecian en la Tabla 24 y otorgando los permisos para su acceso remoto.

ZIP CODE	AGE	DISEASE
47677	29	Heart Disease
47602	22	Heart Disease
47678	27	Heart Disease
47905	43	Flu
47909	52	Heart Disease
47607	32	Cancer
47673	36	Cancer
47605	30	Heart Disease
47906	47	Cancer

Figura 13: Base de datos MySql (anexo)

Realizado lo anterior, los pasos para la implementación de la librería ARX en NetBeans son:

i. Crear el proyecto: es una etapa auto explicativa, simplemente se refiere a crear un proyecto en NetBeans y seleccionar JavaApplication.

ii. Descargar librería ARX: la librería se obtiene desde la web oficial de ARX Anonymization Tool (arx.deidentifier.org/downloads/), el fichero se llama “libarx-3.7.0.jar”.

iii. Incluir el archivo Example.java: para asegurar un correcto funcionamiento de este ejemplo, se requiere tener el archivo Example.java, disponible en el repositorio de ARX. Este archivo se encuentra en la ruta arx/src/example del proyecto. Se debe copiar el contenido, crear una nueva Java Class en el package de su proyecto con el nombre Example.java y pegue ahí.

iv. Incluir la librería ARX: se debe dirigir a la pestaña Projects y hacer clic derecho sobre la carpeta “Libraries” y seleccionar “Add Jar/Folder”, se desplegara un explorador de archivos y solo resta seleccionar el archivo .jar descargado anteriormente desde la web oficial de ARX. Al realizar esto, su proyecto debe tener la siguiente estructura:

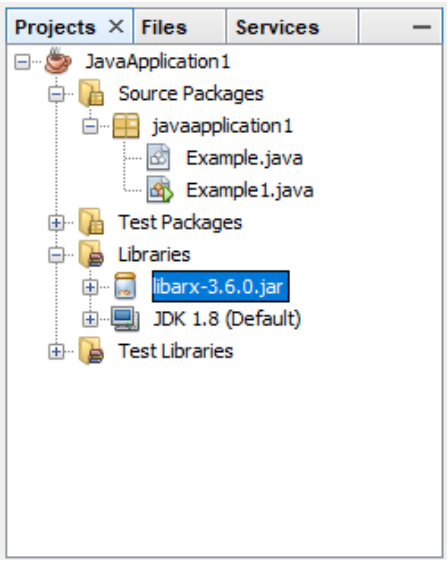


Figura 14: Estructura del proyecto

v. Importar datos: primero es necesario cargar el driver JDBC para el manejo de bases de datos Mysql en Java. Para esto, se debe incluir la siguiente instrucción en la clase “Main” del archivo principal, en este caso es Example1.java:

- `Class.forName("com.mysql.cj.jdbc.Driver");`

Luego, se debe configurar la Fuente de datos con la información del servidor donde se encuentra alojada la base de datos.

- `DataSource source = DataSource.createJDBCSource("jdbc:mysql://159.89.186.236/tcercaniapaper", " root", " 2708", " PACIENTE");`

Al momento de la creación de este documento, dicha base de datos se encuentra active en el servidor, y se espera mantenerla así. Su uso está permitido.

Posteriormente se procede a la inclusión de las columnas, una por una con la instrucción:

- `source.addColumn(0, DataType.INTEGER);`
- `source.addColumn(1, DataType.INTEGER);`
- `source.addColumn(2, DataType.STRING);`

Donde el primer parámetro es el índice de la columna de acuerdo a su posición, según el ejemplo de la Tabla 24, el índice "0" hace referencia a la columna "ZIP Code", "1" a la columna "AGE" y "2" a la columna "DISEASE".

Finalmente, se crea el objeto de datos:

- `Data data = Data.create(source);`

vi. Definición de las jerarquías de los atributos cuasi identificadores: para definir las jerarquías se utilizó la clase "HierarchyBuilder", la cual crea las jerarquías de forma automática, requiriendo solo los parámetros de configuración. Se utilizaron dos tipos de "builders", uno basado en intervalos y otro basado en redacción (o enmascaramiento). El builder de la columna AGE basado en intervalos se creó de la siguiente forma:

- ```
  HierarchyBuilderIntervalBased<Long> ageBuilder =
  HierarchyBuilderIntervalBased.create(
  DataType.INTEGER, new Range<Long>(20l,20l,20l), new
  Range<Long>(60l,60l,60l));
  ```

El primer rango (Range) hace alusión al límite inferior, mientras que el segundo hace referencia al límite superior del rango total permitido para los intervalos. Luego son definidos los intervalos:

- `ageBuilder.setAggregateFunction(DataType.INTEGER.createAggregate().createIntervalFunction(true, false));`
  ```

```
ageBuilder.addInterval(20l, 30l);
ageBuilder.addInterval(30l, 40l);
ageBuilder.addInterval(40l, 60l);
```

El builder necesita leer los registros de la columna, en este caso, AGE, para asignar un intervalo a cada valor. Para ello se crea un "Handle" con el cual se obtienen todos los valores de dicha columna, y son almacenados en el array de String "ageRow".

- ```
  DataHandle inHandle = data.getHandle();
  String[] ageRow = new String[9];
  for (int i=0; i< ageRow.length; i++)
  ageRow[i] = inHandle.getValue(i, 1);
  ```

Por último, se ejecuta el builder, obteniendo la siguiente jerarquía de valores:

- ```
  ageBuilder.prepare(ageRow);
  ```

```
{{"29","[20, 30[","*"},
{"22","[20, 30[","*"},
{"27","[20, 30[","*"},
{"43","[40, 60[","*"},
{"52","[40, 60[","*"},
{"32","[30, 40[","*"},
{"36","[30, 40[","*"},
{"30","[30, 40[","*"},
{"47","[40, 60[","*"}}
```

Figura 15: Jerarquía de generalización del atributo "AGE" (anexo)

El builder de la columna ZIP Code, basado en redacción, se crea de la siguiente forma:

- ```
  HierarchyBuilderRedactionBased<?> zipBuilder =
  HierarchyBuilderRedactionBased.create(Order.RIGHT_TO_LEFT,
  Order.RIGHT_TO_LEFT, ' ', '*');
  ```
- ```
  DataHandle zipCodeHandler = data.getHandle();
  ```
- ```
  String[] zipcode = new String[9];
  ```
- ```
  for (int i=0; i< zipcode.length; i++)
  ```
- ```
  zipcode[i] = zipCodeHandler.getValue(i, 0);
  ```
  ```

- `zipBuilder.prepare(zipcode);`

Este builder reemplaza, de derecha a izquierda, los caracteres de los registros de esta columna por un "*", un caracter por cada nivel de generalización, resultando la siguiente jerarquía:

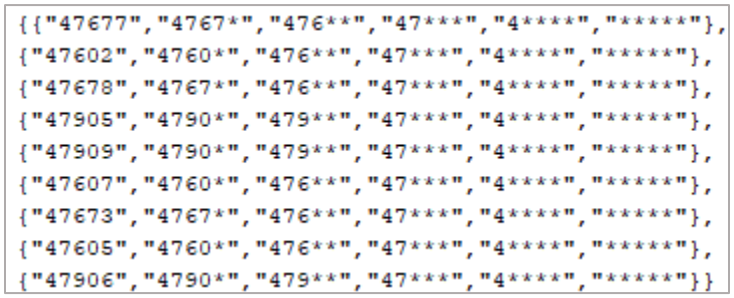


Figura 16: Jerarquía de generalización del atributo "ZIP Code" (anexo)

vii. Definición del tipo de cada atributo (cuasi identificador, sensible, entre otros):

- `data.getDefinition().setAttributeType("AGE", ageBuilder);`
- `data.getDefinition().setAttributeType("ZIP CODE", zipBuilder);`
- `data.getDefinition().setAttributeType("DISEASE", AttributeType.INSENSITIVE_ATTRIBUTE);`

En el caso de los atributos cuasi identificadores, los parámetros de la función "setAtributeType( )" son: el nombre (debe ser el mismo al nombre de la columna correspondiente en la base de datos MySql) y el builder con su jerarquía. El atributo "DISEASE" se considera "Insensitive" o no sensible, ya que nos requiere realizar ninguna acción sobre él.

viii. Definición del modelo de privacidad a utilizar y sus parámetros: se debe crear la instancia del anonimizado:

- `ARXAnonymizer anonymizer = new ARXAnonymizer();`
- `ARXConfiguration config = ARXConfiguration.create();`
- `config.setSuppressionLimit(0d);`

Además, definir el parámetro de k-anonimato con $k = 3$:

- `config.addPrivacyModel(new KAnonymity(3));`

ix. Anonimizar: se realiza el proceso de anonimización.

- `ARXResult result = anonymizer.anonymize(data, config);`

x. Ordenar el resultado (opcional): para obtener una mejor visualización del resultado, se ordena ascendentemente (parámetro "true") según el resultado de la columna ZIP Code (segundo parámetro, "0", índice de la columna ZIP Code) con la función sort( ).

- `DataHandle outHandle = result.getOutput(false);`
- `outHandle.sort(true, 0);`

xi. Mostrar resultado: presentar el resultado en pantalla.

- `System.out.println(" - Transformed data:");`
- `Iterator<String[]> transformed = result.getOutput(false).iterator();`
- `while (transformed.hasNext()){ System.out.print("   "); System.out.println(Arrays.toString(transformed.next())); }`

Se obtiene el siguiente resultado, el cual se compara con la Tabla 25. Es así como se puede corroborar el funcionamiento adecuado de la librería ARX, ya que los resultados entregados son idénticos a los presentados en la literatura:

```
- Transformed data:
  [ZIP CODE, AGE, DISEASE]
  [476**, [20, 30[, Heart Disease]
  [476**, [20, 30[, Heart Disease]
  [476**, [20, 30[, Heart Disease]
  [476**, [30, 40[, Cancer]
  [476**, [30, 40[, Cancer]
  [476**, [30, 40[, Heart Disease]
  [479**, [40, 60[, Flu]
  [479**, [40, 60[, Heart Disease]
  [479**, [40, 60[, Cancer]
```

Figura 17: Tabla resultante 3-anonima (anexo)

7.2. Anexo 02: Datos de métricas

En este anexo, se detallan los datos obtenidos desde la ejecución de los algoritmos que buscan proteger la privacidad, considerando diversas métricas. A continuación, se muestran los resultados en forma de tablas, las cuales son clasificadas teniendo en cuenta la técnica aplicada y parámetros utilizados.

i. k-anonimato

Valor del parámetro $k = 2$	
Tiempo para anonimizar:	0.84 segundos
Perdida de información:	10.755107672663633 %
Tamaño promedio de clases de equivalencia	20.22309899569584
Tamaño máximo de la clase de equivalencia	1250
Tamaño mínimo de la clase de equivalencia	2
Numero de clases de equivalencia	1394
Numero de tuplas	28191
Numero de tuplas suprimidas	1971
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14940
Valor moda	[20-39]
Riesgo promedio de re-identificación	4.9448405519492034 %
Riesgo máximo de re-identificación	50 %
Tuplas en riesgo	13.107019970912703 %

Valor del parámetro $k = 5$	
Tiempo para anonimizar:	1.21 segundos
Perdida de información:	16.21326131666998 %
Tamaño promedio de clases de equivalencia	54.011583011583014
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	518
Numero de tuplas	27978
Numero de tuplas suprimidas	2184
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0

Cantidad del valor moda	14931
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.8514547144184716 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.5821717063406965 %

Valor del parámetro $k = 8$	
Tiempo para anonimizar:	1.32 segundos
Perdida de información:	18.8452728927623 %
Tamaño promedio de clases de equivalencia	73.67391304347827
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	8
Numero de clases de equivalencia	368
Numero de tuplas	27112
Numero de tuplas suprimidas	3050
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	79
Mínimo de edad	0
Cantidad del valor moda	14648
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.3573325464738861 %
Riesgo máximo de re-identificación	12.5 %
Tuplas en riesgo	1.5343759221009147 %

Valor del parámetro $k = 12$	
Tiempo para anonimizar:	1.37 segundos
Perdida de información:	20.483907217969088 %
Tamaño promedio de clases de equivalencia	121.19736842105263
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	12
Numero de clases de equivalencia	228
Numero de tuplas	27633
Numero de tuplas suprimidas	2529
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	79
Mínimo de edad	0
Cantidad del valor moda	14780
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.8251004234067962 %
Riesgo máximo de re-identificación	8.333333333333333 %
Tuplas en riesgo	0.0 %

Valor del parámetro $k = 15$	
Tiempo para anonimizar:	1.63 segundos
Perdida de información:	21.276112589866925 %
Tamaño promedio de clases de equivalencia	132.18357487922705
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	15
Numero de clases de equivalencia	207
Numero de tuplas	27362
Numero de tuplas suprimidas	2800
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	79
Mínimo de edad	0
Cantidad del valor moda	14676
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.7565236459323149 %
Riesgo máximo de re-identificación	6.666666666666667 %
Tuplas en riesgo	0.0 %

Valor del parámetro $k = 17$	
Tiempo para anonimizar:	1.64 segundos
Perdida de información:	21.909891188744024 %
Tamaño promedio de clases de equivalencia	140.65284974093265
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	17
Numero de clases de equivalencia	193
Numero de tuplas	27146
Numero de tuplas suprimidas	3016
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	79
Mínimo de edad	0
Cantidad del valor moda	14553
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.7109703087010978 %
Riesgo máximo de re-identificación	5.8823529411764705 %
Tuplas en riesgo	0.0 %

Valor del parámetro $k = 20$	
Tiempo para anonimizar:	1.77 segundos
Perdida de información:	23.118892328526375 %
Tamaño promedio de clases de equivalencia	157.24705882352941
Tamaño máximo de la clase de equivalencia	1685

Tamaño mínimo de la clase de equivalencia	20
Numero de clases de equivalencia	170
Numero de tuplas	26732
Numero de tuplas suprimidas	3430
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	79
Mínimo de edad	0
Cantidad del valor moda	14445
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.6359419422415083 %
Riesgo máximo de re-identificación	5 %
Tuplas en riesgo	0.0 %

Valor del parámetro $k = 25$	
Tiempo para anonimizar:	1.82 segundos
Perdida de información:	24.76722 %
Tamaño promedio de clases de equivalencia	181.70833333333334
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	25
Numero de clases de equivalencia	144
Numero de tuplas	26166
Numero de tuplas suprimidas	3996
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	79
Mínimo de edad	0
Cantidad del valor moda	14179
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.5503324925475809 %
Riesgo máximo de re-identificación	4 %
Tuplas en riesgo	0.0 %

ii. (c, l)- diversidad recursiva

Valor del parámetro $c = 4;\ \ell = 2$	
Tiempo para anonimizar:	2.31 segundos
Perdida de información:	16.737465 %
Tamaño promedio de clases de equivalencia	55.38844621513944
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	502
Numero de tuplas	27805
Numero de tuplas suprimidas	2357

Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14879
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.805430677935623 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.3912965294011865 %

Valor del parámetro $c = 4;\ \ell = 4$	
Tiempo para anonimizar:	2.90 segundos
Perdida de información:	20.390491 %
Tamaño promedio de clases de equivalencia	90.69180327868852
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	305
Numero de tuplas	27661
Numero de tuplas suprimidas	2501
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14839
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.1026354795560536 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	2.104045406890568 %

Valor del parámetro $c = 4;\ \ell = 6$	
Tiempo para anonimizar:	4.09 segundos
Perdida de información:	28.83921 %
Tamaño promedio de clases de equivalencia	213.93548387096774
Tamaño máximo de la clase de equivalencia	2751
Tamaño mínimo de la clase de equivalencia	7
Numero de clases de equivalencia	124
Numero de tuplas	26528
Numero de tuplas suprimidas	3634
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14236

Valor moda	[20-39]
Riesgo promedio de re-identificación	0.46743063932448736 %
Riesgo máximo de re-identificación	14.285714285714285 %
Tuplas en riesgo	0.20732810615199035 %

Valor del parámetro $c = 4;\ \ell = 8$	
Tiempo para anonimizar:	4.44 segundos
Perdida de información:	40.41359 %
Tamaño promedio de clases de equivalencia	518.0625
Tamaño máximo de la clase de equivalencia	5292
Tamaño mínimo de la clase de equivalencia	9
Numero de clases de equivalencia	48
Numero de tuplas	24867
Numero de tuplas suprimidas	5295
Promedio de edad (con promedio del intervalo)	0
Promedio de edad (con límite inferior del intervalo)	0
Máximo de edad	0
Mínimo de edad	0
Cantidad del valor moda	30162
Valor moda	*
Riesgo promedio de re-identificación	0.19302690312462298 %
Riesgo máximo de re-identificación	11.11111111111111 %
Tuplas en riesgo	0.036192544335866814 %

Valor del parámetro $c = 4;\ \ell = 10$	
Tiempo para anonimizar:	5.23 segundos
Perdida de información:	58.006668 %
Tamaño promedio de clases de equivalencia	990.125
Tamaño máximo de la clase de equivalencia	5495
Tamaño mínimo de la clase de equivalencia	34
Numero de clases de equivalencia	24
Numero de tuplas	23763
Numero de tuplas suprimidas	6399
Promedio de edad (con promedio del intervalo)	38
Promedio de edad (con límite inferior del intervalo)	29
Máximo de edad	79
Mínimo de edad	20
Cantidad del valor moda	13733
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.10099734881959348%
Riesgo máximo de re-identificación	2.941176470588235 %
Tuplas en riesgo	0.0 %

Valor del parámetro $c = 4;\ \ell = 12$	
Tiempo para anonimizar:	4.88 segundos
Perdida de información:	98.71623 %
Tamaño promedio de clases de equivalencia	643.0
Tamaño máximo de la clase de equivalencia	643
Tamaño mínimo de la clase de equivalencia	643
Numero de clases de equivalencia	1
Numero de tuplas	643
Numero de tuplas suprimidas	29519
Promedio de edad (con promedio del intervalo)	69
Promedio de edad (con límite inferior del intervalo)	60
Máximo de edad	79
Mínimo de edad	60
Cantidad del valor moda	29519
Valor moda	*
Riesgo promedio de re-identificación	0.15552099533437014 %
Riesgo máximo de re-identificación	0.15552099533437014 %
Tuplas en riesgo	0.0 %

iii. *t*-cercanía

Valor del parámetro $t = 0.1$	
Tiempo para anonimizar:	5.22 segundos
Perdida de información:	71.18766 %
Tamaño promedio de clases de equivalencia	2531.6666666666665
Tamaño máximo de la clase de equivalencia	3422
Tamaño mínimo de la clase de equivalencia	843
Numero de clases de equivalencia	9
Numero de tuplas	22785
Numero de tuplas suprimidas	7377
Promedio de edad (con promedio del intervalo)	38
Promedio de edad (con límite inferior del intervalo)	36
Máximo de edad	64
Mínimo de edad	20
Cantidad del valor moda	7377
Valor moda	*
Riesgo promedio de re-identificación	0.03949967083607637 %
Riesgo máximo de re-identificación	0.11862396204033215 %
Tuplas en riesgo	0.0 %

Valor del parámetro $t = 0.2$	
Tiempo para anonimizar:	5.09 segundos
Perdida de información:	50.498463 %
Tamaño promedio de clases de equivalencia	618.6756756756756

Tamaño máximo de la clase de equivalencia	5047
Tamaño mínimo de la clase de equivalencia	9
Numero de clases de equivalencia	37
Numero de tuplas	22891
Numero de tuplas suprimidas	7271
Promedio de edad (con promedio del intervalo)	38
Promedio de edad (con límite inferior del intervalo)	29
Máximo de edad	79
Mínimo de edad	20
Cantidad del valor moda	13004
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.1616355773011227 %
Riesgo máximo de re-identificación	11.11111111111111 %
Tuplas en riesgo	0.03931676204621904 %

Valor del parámetro $t = 0.3$	
Tiempo para anonimizar:	4,47 segundos
Perdida de información:	33.960964 %
Tamaño promedio de clases de equivalencia	264.12745098039215
Tamaño máximo de la clase de equivalencia	3066
Tamaño mínimo de la clase de equivalencia	7
Numero de clases de equivalencia	102
Numero de tuplas	26941
Numero de tuplas suprimidas	3221
Promedio de edad (con promedio del intervalo)	37
Promedio de edad (con límite inferior del intervalo)	28
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14958
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.37860510003340636 %
Riesgo máximo de re-identificación	14.285714285714285 %
Tuplas en riesgo	0.1187780705987157 %

Valor del parámetro $t = 0.4$	
Tiempo para anonimizar:	3.25 segundos
Perdida de información:	22.734863 %
Tamaño promedio de clases de equivalencia	107.88353413654619
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	249
Numero de tuplas	26863
Numero de tuplas suprimidas	3299

Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14508
Valor moda	[20-39]
Riesgo promedio de re-identificación	0.9269255109258088 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	1.3885269701820346 %

Valor del parámetro $t = 0.5$	
Tiempo para anonimizar:	2.62 segundos
Perdida de información:	19.153414 %
Tamaño promedio de clases de equivalencia	84.59939759036145
Tamaño máximo de la clase de equivalencia	1685
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	332
Numero de tuplas	28087
Numero de tuplas suprimidas	2075
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14972
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.1820415138676255 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	2.456652543881511 %

Valor del parámetro $t = 0.6$	
Tiempo para anonimizar:	2.23 segundos
Perdida de información:	17.063976 %
Tamaño promedio de clases de equivalencia	57.10927835051547
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	485
Numero de tuplas	27698
Numero de tuplas suprimidas	2464
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14816

Valor moda	[20-39]
Riesgo promedio de re-identificación	1.7510289551592173 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.0472236262546035 %

Valor del parámetro $t = 0.7$	
Tiempo para anonimizar:	2.23 segundos
Perdida de información:	16.404898 %
Tamaño promedio de clases de equivalencia	55.16798418972332
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	506
Numero de tuplas	27915
Numero de tuplas suprimidas	2247
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14911
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.8126455310764824 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.366827870320616 %

Valor del parámetro $t = 0.8$	
Tiempo para anonimizar:	2.52 segundos
Perdida de información:	16.21326 %
Tamaño promedio de clases de equivalencia	54.011583011583014
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	518
Numero de tuplas	27978
Numero de tuplas suprimidas	2184
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14931
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.8514547144184716 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.582171706340697 %

Valor del parámetro $t = 0.9$	
Tiempo para anonimizar:	2.12 segundos
Perdida de información:	16.21326 %
Tamaño promedio de clases de equivalencia	54.011583011583014
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	518
Numero de tuplas	27978
Numero de tuplas suprimidas	2184
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14931
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.8514547144184716 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.582171706340697 %

Valor del parámetro $t = 1$	
Tiempo para anonimizar:	2.20 segundos
Perdida de información:	16.21326 %
Tamaño promedio de clases de equivalencia	54.011583011583014
Tamaño máximo de la clase de equivalencia	1340
Tamaño mínimo de la clase de equivalencia	5
Numero de clases de equivalencia	518
Numero de tuplas	27978
Numero de tuplas suprimidas	2184
Promedio de edad (con promedio del intervalo)	36
Promedio de edad (con límite inferior del intervalo)	27
Máximo de edad	99
Mínimo de edad	0
Cantidad del valor moda	14931
Valor moda	[20-39]
Riesgo promedio de re-identificación	1.8514547144184716 %
Riesgo máximo de re-identificación	20 %
Tuplas en riesgo	4.582171706340697 %

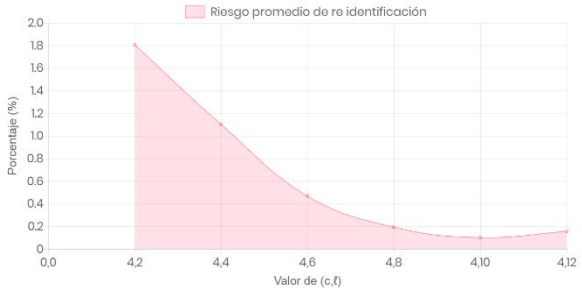
7.3. Anexo 03: Gráficos de métricas

En esta sección serán mostrados los resultados de los experimentos, obtenidos al ejecutar cada una de las técnicas de anonimato, en forma de gráficos. Esto tiene como objetivo generar una representación gráfica de los datos obtenidos a partir de as métricas definidas.

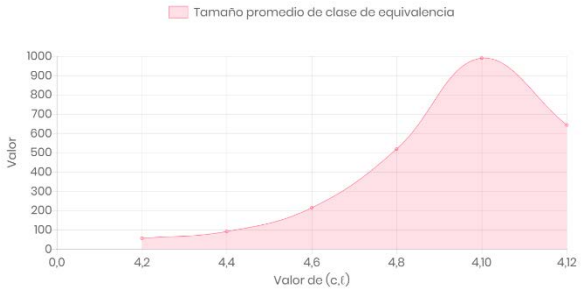
Para la mejor comprensión, estos gráficos serán agrupados de acuerdo a la métrica que representan.

i. Perdida de información

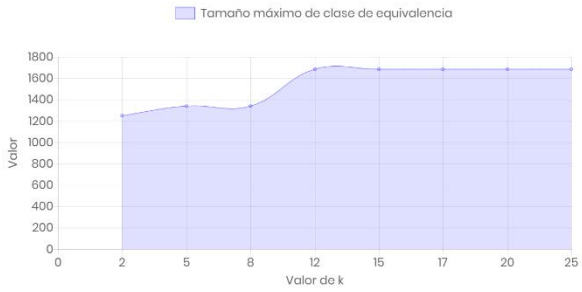
ii. Riesgo de re-identificación



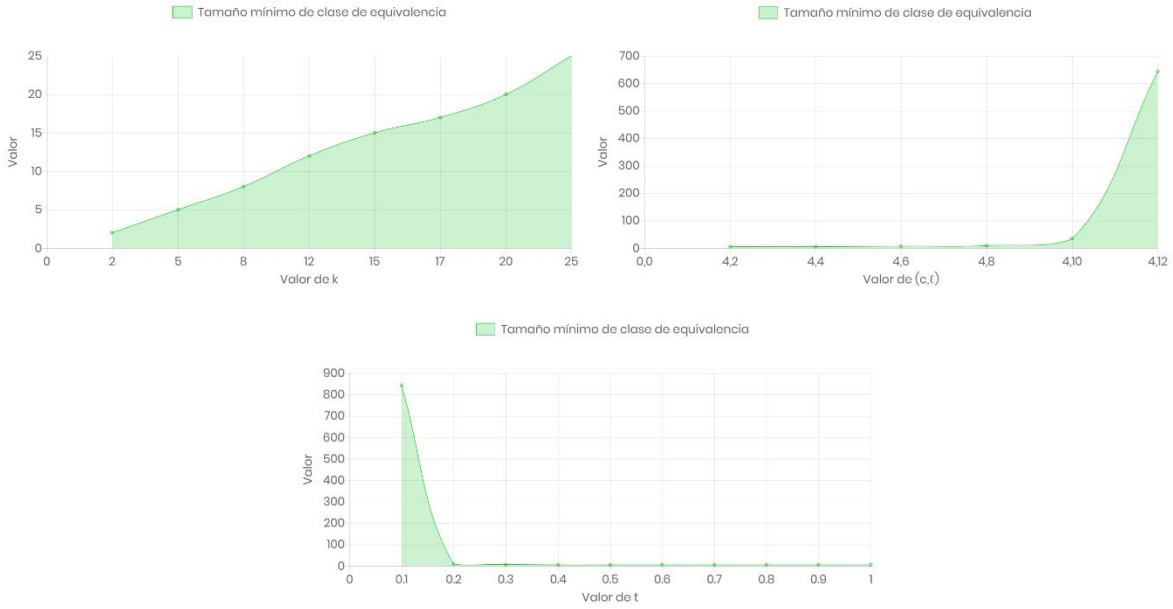
iii. Tamaño promedio de clase de equivalencia



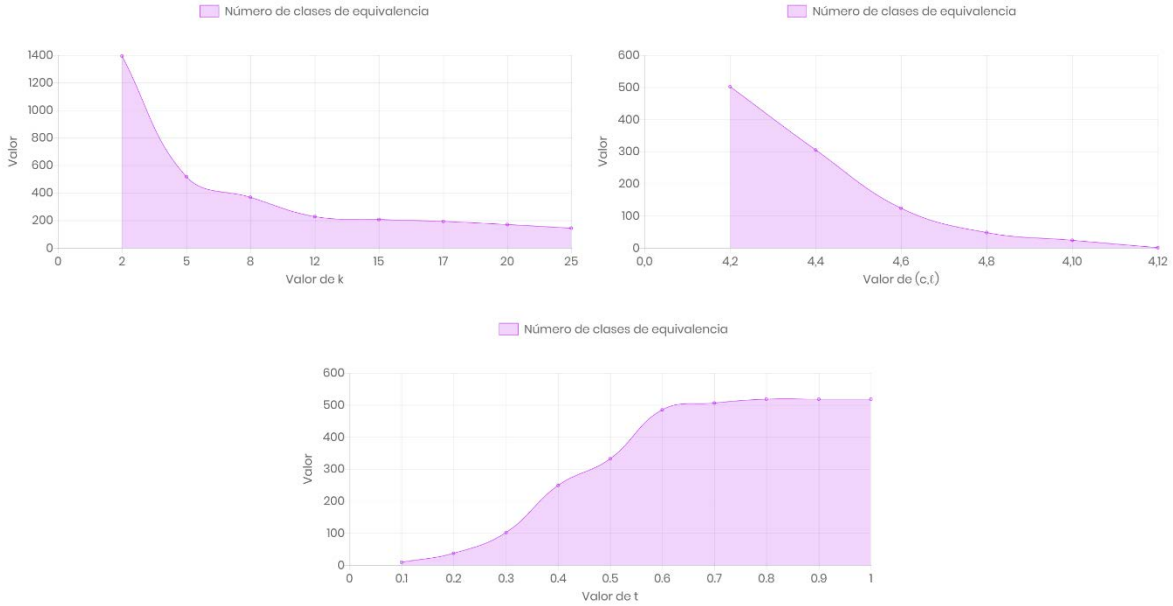
iv. Tamaño máximo de clase de equivalencia



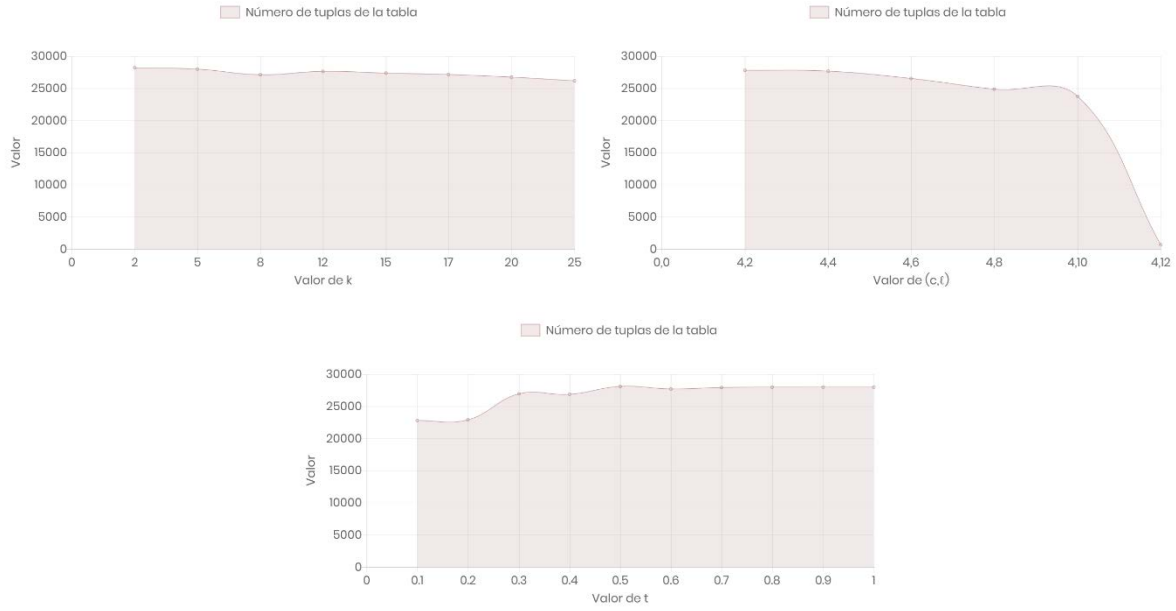
v. Tamaño mínimo de clase de equivalencia



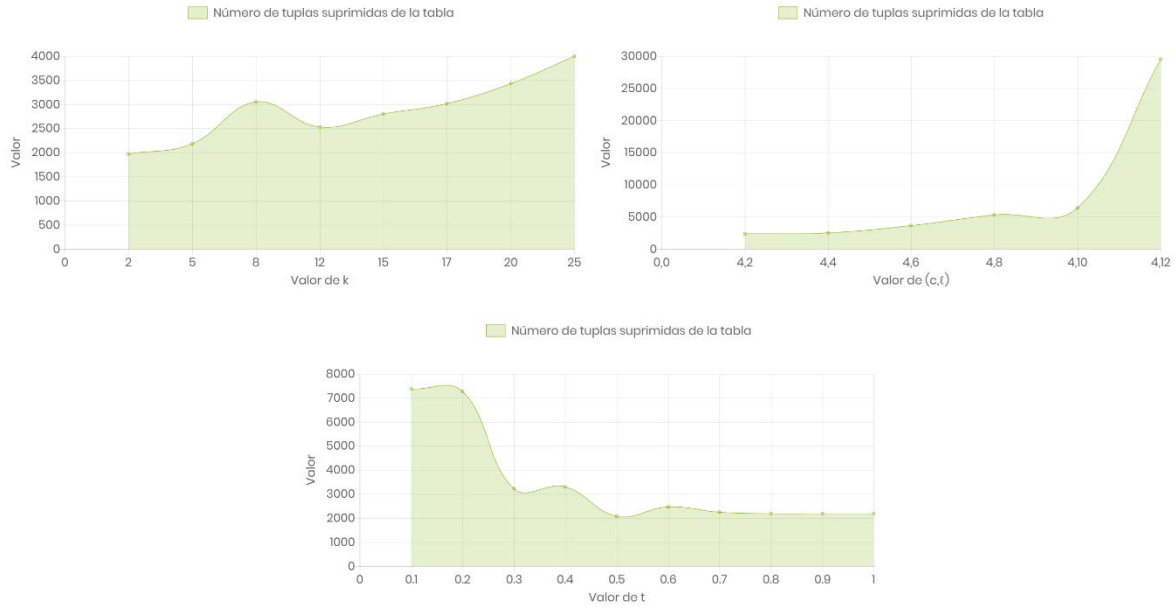
vi. Número de clases de equivalencia



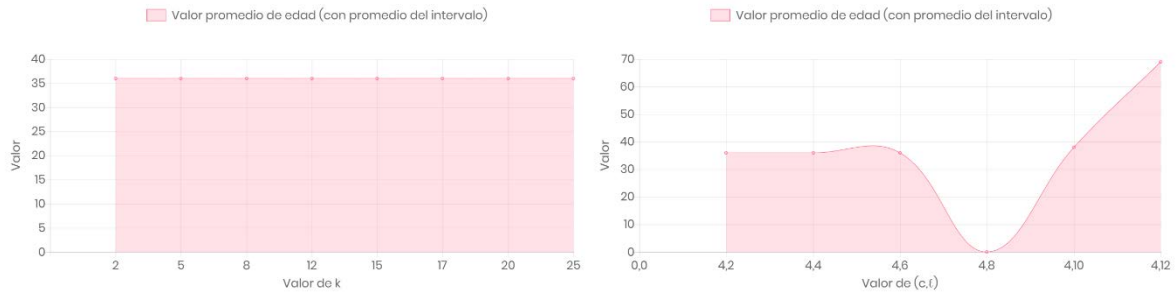
vii. Número de tuplas de la tabla



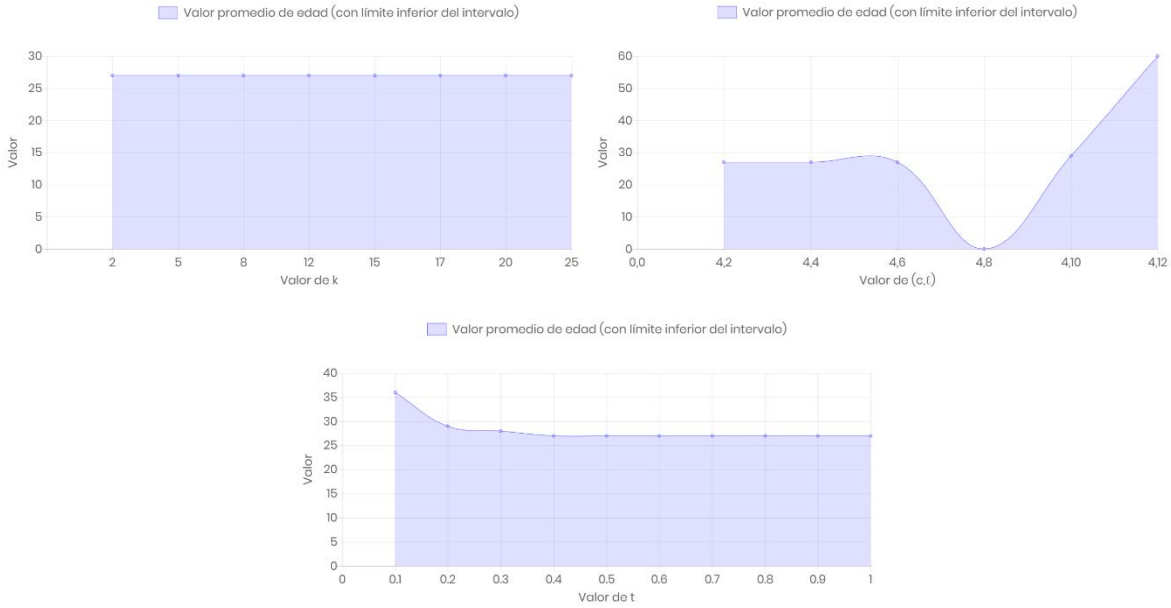
viii. Número de tuplas suprimidas de la tabla



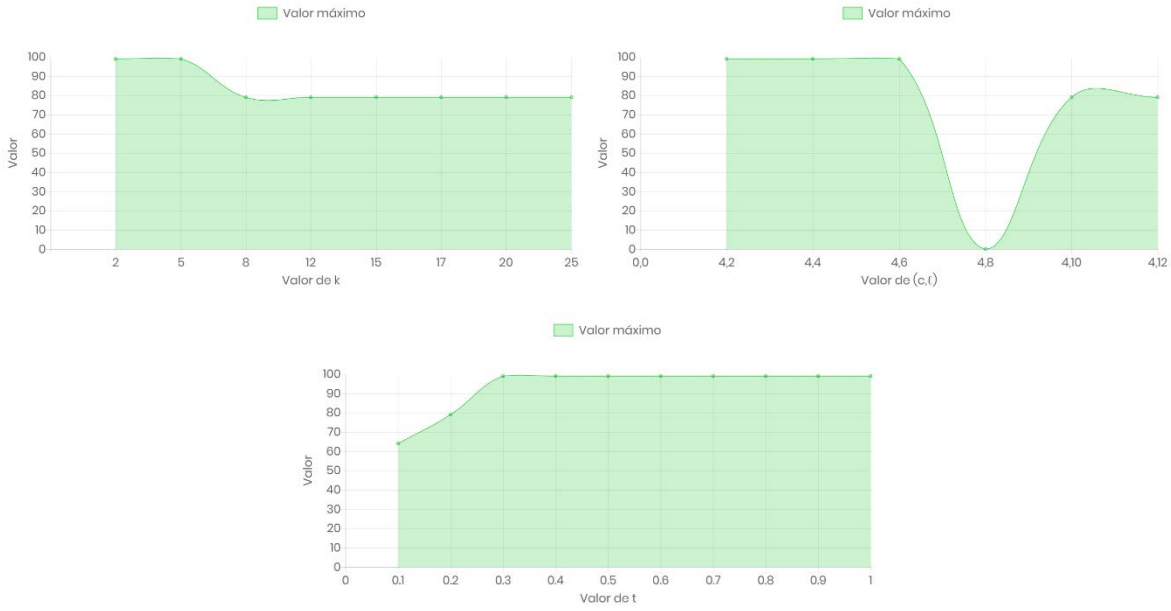
ix. Valor promedio de edad (con promedio del intervalo)



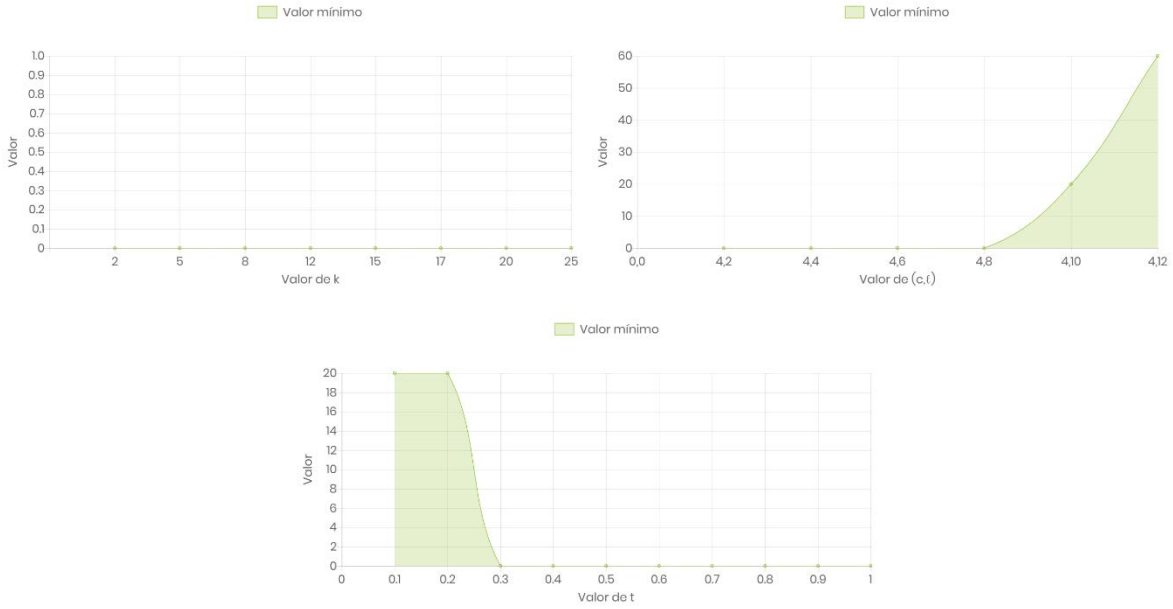
x. Valor promedio de edad (con límite inferior del intervalo)



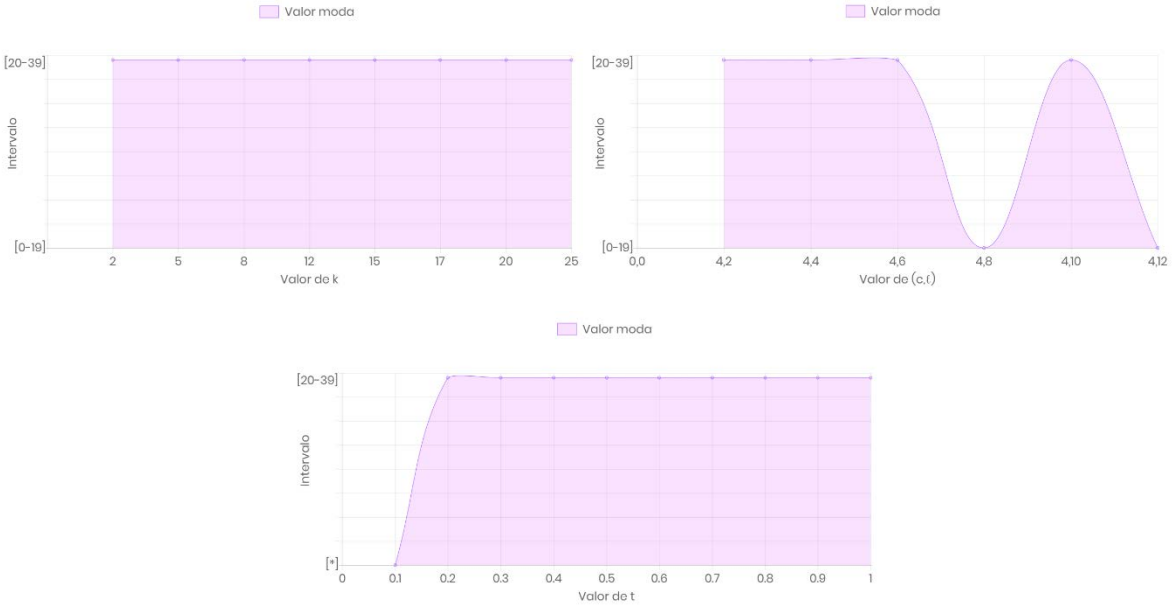
xi. Valor máximo



xii. Valor mínimo



xiii. Valor moda



xiv. Cantidad moda