PA173-1-SAE

Sistema de anonimización de datos estructurados

Andrea Patricia Ortiz Pulido

Jonnathan Silvestre Corredor Merchán

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2017

PA173-1-SAE

Sistema de anonimización de datos estructurados

Autor:

Andrea Patricia Ortiz Pulido

Jonnathan Silvestre Corredor Merchán

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE

MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Directora

Ingeniera Alexandra Pomares Quimbaya

Comité de Evaluación del Trabajo de Grado

María Josefina Curiel Huérfano

Rafael Andrés González Rivera

Página web del Trabajo de Grado

http://pegasus.javeriana.edu.co/~PA173-1-SAE/

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
NOVIEMBRE, 2017

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Rector Magnífico

Jorge Humberto Peláez, S.J.

Decano Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Director Maestría en Ingeniería de Sistemas y Computación

Ingeniera Ángela Cristina Carrillo Ramos

Director Departamento de Ingeniería de Sistemas

Ingeniero Efraín Ortiz Pabón

Artículo 23 de la Resolución No. 1 de Junio de 1946

"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"

AGRADECIMIENTOS

Agradecemos infinitamente a Dios todas las oportunidades que nos ha brindado de formarnos y aprender de diversas disciplinas, forjando profesionales para el servicio. El Espíritu Santo nos ha guiado en este arduo camino que enfrentamos día a día, pero siempre con la recompensa del esfuerzo y el aprendizaje adquirido.

A nuestras familias, gracias por apoyarnos en esta travesía de mucho esfuerzo y dedicación, pues su amor y comprensión nos han permitido concentrarnos en el proyecto, mientras ellos nos cubren la espalda en las demás obligaciones.

Agradecemos a la Alianza CAOBA y en especial a la Ingeniera Alexandra Pomares Quimbaya por confiar en nosotros este importante e interesante proyecto, que esperamos sea de utilidad acá, en Colombia y en el mundo.

Contenido

Lista de Tablas

Lista de Figuras

Lista de Documentos Anexos

Los siguientes documentos anexos pueden ser encontrados en la página web del Trabajo de Grado http://pegasus.javeriana.edu.co/~PA173-1-SAE/.

1. Estado del Arte de Anonimización en *Big Data*
2. Especificación de Requerimientos de Software
3. *Mockup* Anonimización ligera
4. *Mockup* Anonimización completa
5. Documento de Diseño de Software
6. Manual de Instalación
7. Manual de Usuario

ABSTRACT

The most used approaches in the industry to protect private data imply to impair its utility for analytical exercises. For this reason, this work proposes *Anonylitics*, a system for the anonymization of structured data, which is based on the preservation of the distribution of numerical data, at the same time that their privacy is guaranteed. The proposal makes it possible to continue having useful information for business data analytics, which is evidenced through the validation carried out by anonymizing two sets of real data that demonstrate the potential of the system and its algorithms.

RESUMEN

Las aproximaciones más empleadas en la industria para proteger los datos privados implican deteriorar su utilidad para los ejercicios de analítica. Por ello, este trabajo propone *Anonylitics*, un sistema para la anonimización de datos estructurados, que se fundamenta en la preservación de la distribución de los datos numéricos, al mismo tiempo que se garantiza su privacidad. La propuesta realizada permite seguir teniendo información útil para la analítica de datos a nivel empresarial, lo cual es evidenciado a través de la validación efectuada mediante la anonimización de dos conjuntos de datos reales que demuestran el potencial del sistema y sus algoritmos.

RESUMEN EJECUTIVO

La gran cantidad de información que almacenan las compañías, debido a los avances tecnológicos, como por ejemplo el internet de las cosas, es un interesante atractivo para analizar tales datos y poder tomar decisiones informadas. La analítica de datos permite mejorar diferentes aspectos como la calidad de los productos y servicios o la experiencia del cliente, a través del análisis de sus hábitos de consumo, sus gustos, medios de pago, periodicidad de compra, entre otros. Las acciones que pueden tomarse respecto a campañas, promociones, planes, etc., posibilitan que los indicadores de desempeño de las empresas puedan prosperar y redundar en beneficios económicos y reputacionales. Sin embargo, no todas las empresas poseen científicos de datos que puedan generar análisis precisos y útiles, y por ello deciden tercerizar esta tarea para que los expertos puedan ejercer tal labor. De este modo, los datos deben salir de sus empresas, poniendo en riesgo la información privada y confidencial que se almacene de individuos u otras compañías.

La situación planteada supone decisiones complejas respecto a la entrega o no de la información a un tercero. En el primer caso, se decide divulgar los datos para poder enriquecer la actividad de la compañía, pero arriesgando los datos sensibles que puedan estar contenidos. Es importante resaltar que en muchas ocasiones las empresas optan por entregar sus datos encriptados, sin anticipar que esto inhibe la posibilidad de encontrar patrones subyacentes en los datos. En otro caso, se toma una posición conservadora donde se abstiene de entregar los datos, dejando de adquirir nuevos conocimientos que permitirían hacer a la compañía una entidad más competitiva. Estos escenarios representan la situación de diversas empresas que quisieran utilizar los servicios provistos por CAOBA, el centro colombiano de excelencia y apropiación en *Big Data* y *Data Analytics*, pero que no sienten la tranquilidad de entregar sus datos por las diferentes políticas sobre el manejo de información dispuestas en la ley de *Habeas Data* [1] y la ley estatutaria 1581 de 2012 [2], la cual dicta disposiciones generales para la protección de datos personales. Internacionalmente también existe una política denominada Lenguaje de Autorización de Privacidad Empresarial (EPAL) [3], que establece la protección a la privacidad del consumidor por parte de las empresas. Por todo esto, las compañías deben ser conscientes de cómo administran sus datos y así evitar problemas judiciales, que incluso repercutan en su reputación y sostenibilidad económica.

La problemática planteada motivó entonces a la Alianza CAOBA a propiciar un espacio para desarrollar un trabajo de investigación aplicada para la generación de una propuesta que permitiera entregar a sus clientes datos estructurados, sin ponerlos en riesgo, a través de la técnica de *anonimización* [4]. Esta aproximación busca enmascarar los datos mediante diferentes transformaciones, procurando generar una baja pérdida de información, permitiendo que los ejercicios de analítica sobre los datos anonimizados sigan siendo útiles para la toma de decisiones. De este modo, el objetivo del proyecto versó en la construcción de un sistema de anonimización de datos estructurados que permitiera preparar información privada y confidencial para su posterior análisis.

El estado del arte en torno a la temática planteada permitió tener las bases para la generación de una propuesta de anonimización orientada a la analítica de datos, dejando en evidencia que aún existen campos abiertos de investigación para mejorar las aproximaciones actuales. En

primera instancia es importante "desmitificar" la creencia de acuerdo con la cual, se considera suficiente eliminar los datos identificadores, como la cédula, pues la ciencia ha demostrado que a través de otros datos es posible llegar a vulnerar la información [5]. Partiendo de ello y entendiendo que no sólo se trata de eliminar, cifrar o cambiar los datos sin detenerse a analizar si ello podría alterar sus propiedades estadísticas, se evidenció que en general, los algoritmos de anonimización no miden aspectos trascendentales para un ejercicio analítico, como la preservación de la distribución de los datos numéricos [6]. Esto implica un detrimento en la utilidad de los datos, así logre asegurarse su privacidad a través de algún principio de anonimización.

A la luz de los hallazgos realizados, se propuso el algoritmo *Distribution-based Mondrian* el cual se basa en una aproximación previa de LeFevre *et al*. [7], pero se enfoca en el análisis de la función de distribución de probabilidad de los datos numéricos para lograr un mejor balance entre utilidad y privacidad de la información. De acuerdo con ello, se plantearon también los mecanismos necesarios para evaluar estas dos necesidades. Adicionalmente, se analizó un posible escenario de anonimización de un alto volumen de datos en una sola máquina, donde pudiera existir un desbordamiento de memoria. Por lo anterior, se propuso el algoritmo *Rothko-D*, el cual también está inspirado en una aproximación previa de LeFevre *et al*. [8] para estimar el uso de memoria necesaria para anonimizar los datos; el propósito es cargar a la *RAM* un subconjunto que no genere la interrupción de la ejecución. El algoritmo propuesto, de igual forma, se centra en seguir los lineamientos de preservación de la utilidad, a través del análisis de la función de distribución de probabilidad.

Teniendo la propuesta de anonimización orientada a la analítica de datos, se diseñó un sistema que permite a un usuario final transformar sus datos numéricos, procurando encontrar el mejor balance entre utilidad y privacidad. El sistema, *Anonylitics*, está en capacidad de permitir parametrizar las transformaciones que se deseen aplicar para que sea el usuario quien decida cuál es el escenario que le genera los resultados más adecuados para el análisis de información de su organización.

La propuesta fue validada a través de un caso de estudio que contempló dos escenarios de prueba. El primero estuvo orientado a evaluar la utilidad y privacidad de los datos anonimizados mediante *Anonylitics*, a partir de dos conjuntos de datos reales; uno que constituye un marco de referencia para la comparación de nuevas aproximaciones, de acuerdo con el estado del arte, y otro entregado por CAOBA con datos sensibles de uno de sus clientes. Se concluyó que el algoritmo propuesto, a diferencia de la aproximación en la cual se basó, sí logra mantener la distribución univariada de cada uno de los atributos (columnas) numéricos anonimizados, entregando niveles de privacidad adecuados, conforme con el cumplimiento del principio denominado *k-anonymity,* ampliamente referenciado en la literatura.

El segundo escenario de validación permitió evidenciar la pertinencia del algoritmo *Rothko-D*, orientado a la anonimización de grandes volúmenes de datos en un contexto centralizado. Esta aproximación posibilita el manejo apropiado de la información para no permitir el desbordamiento de la memoria de la máquina. Este escenario permite vislumbrar trabajos futuros que se enfoquen en escalar el algoritmo con el objetivo de que funcione en un entorno distribuido, donde se aproveche la capacidad de procesamiento para paralelizar y distribuir las operaciones de transformación de los datos. De igual manera se propone seguir robusteciendo el sistema

construido para que pueda brindar anonimización de datos categóricos, así como nuevas utilidades que permitan al usuario final una mejor experiencia para una exitosa integración de la anonimización en la fase de pre-procesamiento de un ejercicio de analítica de datos.

1. INTRODUCCIÓN

La analítica de datos es una disciplina que ha venido tomando mayor fuerza en Colombia y el mundo gracias al avance tecnológico que ha posibilitado la obtención y almacenamiento de millones de registros, día a día. Sin embargo, esto ha conllevado a mayores desafíos en el manejo responsable de la información pues cada vez se recolectan más datos sensibles que deben ser manejados adecuadamente por las organizaciones, si quieren obtener beneficio de ellos, mediante ejercicios de minería. En la presente sección se establece la motivación del trabajo realizado, así como la propuesta metodológica para abordarlo y los resultados esperados.

1.1.Motivación

El *Centro de Excelencia y Apropiación en Big Data y Data Analytics*, CAOBA, es la alianza colombiana generada entre el sector académico (universidades), la industria (anclas) y empresas tecnológicas (*big players*), quienes han centrado sus esfuerzos en el análisis de datos como fuente principal para la toma de decisiones a diferentes niveles y en diversos ámbitos organizacionales. La Pontificia Universidad Javeriana lidera esta iniciativa con el objetivo de reducir la brecha entre la academia y la industria, de forma que nuevos planteamientos y soluciones generados en un entorno académico sean aplicables de cara a las necesidades reales de la industria. CAOBA se encuentra generando nuevas propuestas para empresas del sector bancario, gubernamental y de *retail*, donde se ha evidenciado la necesidad de aplicar analítica de datos; no obstante, muchas de estas organizaciones no han podido participar abiertamente proponiendo sus proyectos debido a que poseen información privada y/o confidencial que no puede ser divulgada a terceros (proveedores o investigadores).

La necesidad de protección de la información es una responsabilidad que atañe a todas las compañías y dado el auge de los sistemas de información y el aumento de las transacciones electrónicas, se comprende la exigencia de aplicar prácticas adecuadas que protejan la privacidad de los datos, en especial aquellos obtenidos de los individuos y de las propias empresas. Existe, por ejemplo, una política internacional denominada Lenguaje de Autorización de Privacidad Empresarial (EPAL), que establece la protección a la privacidad del consumidor por parte de las empresas, viéndose éstas obligadas a publicar declaraciones de privacidad como promesa a la protección y adecuado manejo de datos personales [3]. El desconocimiento o incumplimiento de estas normas puede acarrear penalidades legislativas, demandas y pérdida de la reputación y/o confianza de las empresas. Esto genera temor, implicando que organizaciones tanto al interior de CAOBA como alrededor del mundo, prefieran abstenerse de analizar sus datos, lo cual implica desaprovechar su potencial para la toma de decisiones estratégicas. Por tal razón, los objetivos analíticos en CAOBA se han visto opacados pues las empresas temen por los datos que tendrían que entregar a las Universidades participantes de la alianza.

CAOBA y muchas empresas que ofrecen servicios de analítica de datos, suelen siguen metodologías que les permiten generar una serie de pasos estructurados y sistemáticos que desembocan en el cumplimiento de los objetivos trazados. Existen diferentes metodologías en este campo, entre ellas CRISP-DM [9], KDD [10] y SEMMA [11]. Cada una tiene como propósito analizar datos con diferentes objetivos de negocio, pero tales datos deben preparase para iniciar

el modelamiento de los mismos. Así como es necesario revisar los valores faltantes, los atípicos, entre otros, es indispensable verificar si los datos deben pasar por una etapa de anonimización para asegurar su privacidad y confidencialidad, y así evitar diferentes problemas legales por un posible inadecuado manejo de la información, así como riesgos reputacionales que afecten la operación normal de la empresa. En este sentido, la anonimización es fundamental en un proyecto de analítica pues hace parte del proceso de preparación, sin el cual, no podrían llegar a hacerse análisis útiles que generen valor para las empresas. Los proyectos de analítica no sólo se limitan a aplicar una técnica de minería, sino que requieren una fase de pre-procesamiento que podría implicar hasta el 60% del esfuerzo total [12], por lo cual es imprescindible darle la importancia que requiere.

De acuerdo con lo anterior, se entiende que se requiere un proceso que permita transformar los datos de forma que no se ponga en peligro la información ni los intereses de las empresas que quieren analizarlos. Sin embargo, no es suficiente sólo la óptica de proteger los datos, sino que se debe seguir propendiendo hacia la conservación de su utilidad para los proyectos de analítica. Las aproximaciones más utilizadas por empresas alrededor del mundo y específicamente en las empresas ancla de CAOBA, son cifrar o hacer sustituciones simples sobre los datos que quieren proteger. En el primer caso de cifrамiento, además de perderse el tipo de dato original en el caso de los numéricos, se entorpece la posibilidad de generar patrones a partir de los ejercicios de analítica aplicados. En el segundo caso, aunque podría mantenerse el tipo de dato cambiándolo por otro, la seguridad es baja pues un atacante podría llegar a obtener los valores de sustitución; además, tampoco se está propendiendo hacia la preservación de las características estadísticas de los datos, por lo cual es poco probable que los subsiguientes ejercicios de analítica sigan siendo útiles.

Adicionalmente, no es suficiente con eliminar los atributos *identificadores* pues esto no garantiza que en realidad los datos estén libres de un ataque, conociendo el conjunto de atributos *cuasi-identificadores*, que puede llevar a conocer mayor información vinculada a un registro, e incluso inferir datos *sensibles*. Un ejemplo de estos tipos de atributos se presenta en la Figura 1, donde podría ser posible inferir los datos sensibles de la tabla, aún cuando se eliminaran los identificadores, en caso de que un atacante tenga conocimiento previo o pueda cruzar estos datos con otras bases disponibles públicamente.

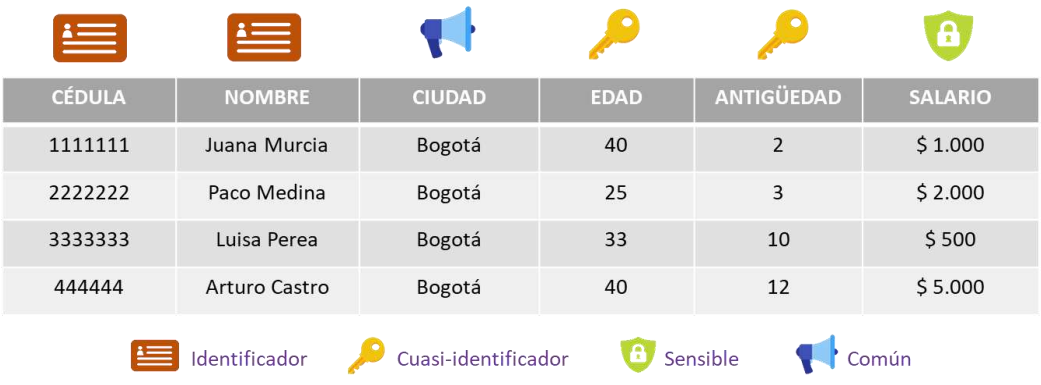


Figura 1. Tipos de atributos en el contexto de la anonimización (Elaboración propia).

Esta situación motiva al centro de excelencia a crear un sistema de anonimización que permita generar un proceso mediante el cual se logre ocultar la información privada y confidencial contenida en un determinado conjunto de datos, sin que ello implique vulnerar los derechos a la protección de datos de las personas y organizaciones, que se puedan referenciar en los mismos [13]; pero también permitiendo su divulgación al procurar minimizar la distorsión de éstos al ser transformados. Esto permitiría que tales organizaciones entreguen con mayor tranquilidad sus datos para realizar proyectos de analítica y puedan enriquecer así su toma de decisiones.

Se entiende así que la anonimización automática o semi-automática de conjuntos de datos es actualmente un desafío importante desde el punto de vista de la ingeniería y la analítica ya que, entre otras cosas, el proceso que se lleva a cabo no se encuentra bien definido, y para su realización se deben combinar diversas operaciones y principios de anonimización [14], [15]. Así mismo, el proceso que se lleve a cabo, debe propender hacia la generación de datos que no permitan re-identificar a la entidad a la que hacen referencia los datos, al mismo tiempo que busque obtener la mínima pérdida de información, con lo cual sea posible seguir generando modelos de analítica útiles [16], [17]. Adicionalmente, se debe evaluar la viabilidad computacional del proceso, dado que garantizar el cumplimiento de ciertos principios de anonimización se convierte en un problema NP-Hard [18] donde sería imposible revisar por completo grandes volúmenes de datos, haciéndose necesario introducir aproximaciones como técnicas de paralelismo para distribuir el procesamiento [19], [20], o valerse de heurísticas y meta-heurísticas [21] que posibiliten encontrar un punto de equilibrio que permita divulgar los datos de una organización, con el objetivo de crear ejercicios analíticos, pero sin poner en peligro su confidencialidad y minimizando la pérdida de información útil.

De acuerdo con la problemática expuesta y los desafíos que le atañen, a lo largo del documento se expone la evaluación de la situación y una propuesta que busca generar una posible solución a través de un trabajo de investigación aplicado que impacta directamente las necesidades de las empresas ancla y que podría extenderse a otros contextos educativos y organizacionales.

1.1.1. Fuentes de datos

Dado que el objetivo es generar un sistema que permita anonimizar conjuntos de datos estructurados, no sería adecuado plantear una única fuente específica para el desarrollo del proyecto, puesto que los datos, per se, son independientes de los principios y algoritmos de anonimización que se deseen emplear para el sistema. No obstante, se planteó un caso de estudio para evaluar un contexto real para CAOBA y lo cual permitió validar la propuesta generada.

Es así, que las fuentes de datos pueden ser diversas, por lo cual aplicaría para contextos de historias clínicas, datos de estudiantes, transacciones bancarias, otorgamiento de créditos, asignación de seguros, información de clientes en empresas de *retail*, entre otros. Siempre que se manejen datos de individuos o de organizaciones, éstos deben ser tratados con especial atención para salvaguardarlos en los múltiples contextos de uso asociados. Es importante tener en cuenta que no todo dato debe ser anonimizado, y por ello las diferentes investigaciones y estudios académicos hechos al respecto han propuesto un marco que permite analizar los tipos de atributos (identificador, cuasi-identificador, sensible y común), con el objetivo de definir cuáles

son vulnerables y las diferentes estrategias requeridas para orientar el proceso de anonimización. Así, es importante entender que los atributos susceptibles de transformación son tanto los identificadores, como los cuasi-identificadores, pues ellos permiten identificar un registro y por tanto inferir un dato sensible. Los datos sensibles son usualmente aquellos que entregan mayor información al momento de realizar proyectos de analítica, pero por sí mismos no representan riesgo, sino lo que de ellos puede saberse a partir de los cuasi-identificadores. Los identificadores tienen aproximaciones simples para ser anonimizados, pues además no representan información indispensable para la analítica, pero aquellos que sí constituyen un riesgo son los cuasi-identificadores, para los cuales se debe generar un mayor esfuerzo de investigación.

De acuerdo con este contexto, se entiende que existe en cualquier base, un conjunto de datos que debe ser analizado para definir un esquema de anonimización antes de ser entregado con cualquier propósito, principalmente la analítica:

- Identificadores como la cédula, el nombre completo de una persona; el NIT de una empresa o su razón social, por ejemplo.
- Cuasi-identificadores como la edad, el género, el estado civil, el nivel de educación, su ubicación de residencia, el número de hijos de una persona, su antigüedad en una empresa; así como los años de constitución de una compañía, datos de su relación con otras empresas, etc.
- Datos sensibles como en valor de un crédito, el salario de una persona, el diagnóstico de enfermedad de un individuo, el valor de una transacción bancaria, entre otros.

Estos ejemplos de datos, que podrían ser vulnerables, están presentes en cualquier empresa pues éstas albergan información tanto de sus empleados como de sus clientes, ya sean personas naturales o jurídicas, así como de sus relaciones con otras empresas proveedoras, aliadas, etc. Tales datos podrían ser usados de forma indebida por algún atacante que tenga conocimiento previo o que pueda cruzarlos con atributos identificadores y/o cuasi-identificadores de bases de datos públicas, como por ejemplo el portal del DANE [22] o el portal de datos abiertos del Gobierno de Colombia [23] que contienen información censal, sobre salud, seguridad, energía, estadísticas, educación, finanzas, entre otros, de la nación.

A pesar de que la anonimización debe permitir trabajar con diferentes fuentes, se comprende que una importante restricción es el formato de los datos el cual debe ser estructurado [24], es decir datos que residen en un campo fijo (columna) dentro de un registro (fila) particular, los cuales cumplen con ser atómicos y se representan mediante los siguientes tipos: numéricos, caracteres (no textos formados por párrafos, sino etiquetas puntuales) y booleanos.

1.1.2. Requerimientos de privacidad y manejo de datos

El proyecto en sí busca la protección de la confidencialidad de datos sensibles de organizaciones y personas, de modo que el tratamiento de los datos entregados por la Pontificia Universidad Javeriana estará sujeto a la ley estatutaria 1581 de 2012, por la cual se dictan disposiciones generales para la protección de datos personales y cuyo objeto es "desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y

garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma" [2]. De este modo, al recibir la autorización para el uso de datos privados y confidenciales provenientes de CAOBA, se velará por salvaguardarlos respetando los requerimientos de privacidad y utilizándolos únicamente con fines investigativos para el desarrollo del presente proyecto.

1.2. Objetivos

A continuación se enuncia el objetivo general del proyecto, así como los cuatro objetivos específicos que están encaminados a lograrlo.

1.2.1. Objetivo general

Construir un sistema de anonimización de datos estructurados que permita preparar información privada y confidencial para su posterior análisis.

1.2.2. Objetivos específicos

1. Caracterizar los modelos y algoritmos para la anonimización de datos estructurados.
2. Diseñar un sistema de anonimización que satisfaga los requerimientos detectados en la alianza CAOBA.
3. Implementar el sistema de anonimización diseñado.
4. Validar el sistema construido a través de un caso de estudio que involucre datos estructurados provistos por la alianza CAOBA.

1.3. Metodología

La estrategia metodológica planteada para la generación del sistema se fundamenta en la investigación basada en el diseño que se cimienta en el rigor y la relevancia, como ejes fundamentales para hacer la construcción de un diseño.

El ciclo de rigor constituye la revisión y análisis en profundidad de la base del conocimiento entendida como fundamentación teórica, estado del arte y su integración, lo cual permitió llevar a cabo la toma de decisiones para realizar el diseño del sistema. Por su parte, el ciclo de relevancia tiene en cuenta el contexto del problema para poder entender el entorno, reflejado en las necesidades de las empresas ancla de CAOBA que desean divulgar sus datos con propósitos analíticos, sin violar la privacidad y confidencialidad de los mismos. En el centro de los dos ciclos se encuentra el objetivo general del proyecto que es el diseño de un sistema de anonimización de datos estructurados, el cual surge como producto de los ciclos de relevancia y rigor.

En la Figura 2 se presenta un esquema de la investigación basada en el diseño. Siguiendo tal modelo, el proyecto se dividió en tres fases principales, asociadas a cada uno de los ciclos de la estrategia metodológica, como se muestra en la Tabla 1.

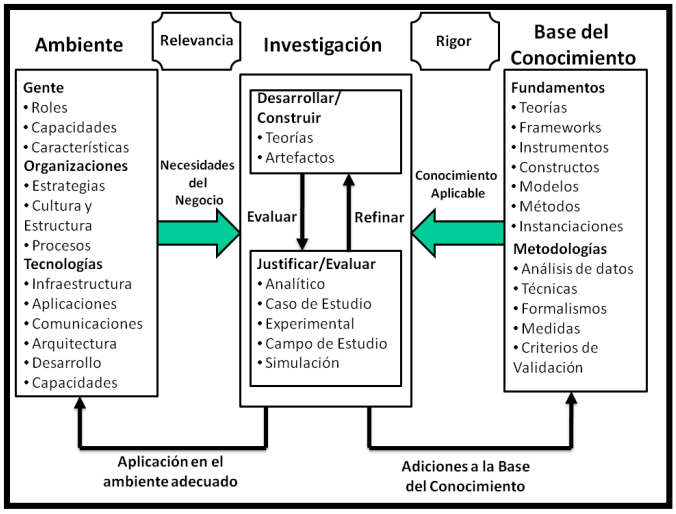


Figura 2. Investigación basada en el diseño, traducido de [25].

Tabla 1. Fases del Trabajo de Grado enmarcadas en la Ciencia basada en el Diseño.

Fase: Trabajo de Grado	Ciclo: Ciencia Basada en el Diseño
Recopilación de Información y Análisis del Ambiente	Rigor y Relevancia
Diseño y Construcción	Diseño
Validación	Diseño

Durante el desarrollo del proyecto, estas fases debieron articularse entre sí, estado en constante retroalimentación. Esto implica que las fases no fueron llevadas a cabo de forma secuencial, sino que en ocasiones fue necesario realizarlas nuevamente con el fin de refinar los insumos y las salidas de cada una para aumentar la calidad en el producto final, siguiendo los lineamientos de la ciencia basada en el diseño.

1.3.1. Fase 1: Recopilación de información y análisis del ambiente

Esta fase comprendió la revisión y análisis del marco teórico y el estado del arte en torno a la anonimización de datos, con el propósito de tener una fundamentación robusta que permitiera sustentar las decisiones tomadas para la construcción del sistema. La investigación se desarrolló en dos frentes: en primera medida se abordó el marco teórico y trabajos relacionados respecto a los contextos "comunes" de anonimización donde no se evalúa la limitación de los recursos (memoria, procesamiento, disco) para las transformaciones de los datos; en la segunda parte se realizó una investigación enfocada en la anonimización en contextos de *Big Data*, generando así un panorama más completo que permitió fortalecer las decisiones de diseño y la implementación. Desde este punto de vista se analizó la *base de conocimiento*; sin embargo, también se hicieron nuevas adiciones a tal base, al construir un artículo científico que enriquece la literatura de anonimización en entornos de *Big Data*, desde la revisión sistemática propuesta.

Esta fase también involucró el entendimiento de negocio, por lo cual se hizo un levantamiento de requerimientos con la dirección de CAOBA y sus empresas ancla, a través de entrevistas y

reuniones, con el objetivo de conocer sus necesidades puntuales de protección de datos. En este caso, se analizó en profundidad el *ambiente*, de acuerdo con la Ciencia Basada en el Diseño.

Esta fase permitió cumplir el objetivo específico No. 1.

Actividades

1. **Revisión del marco teórico y estado del arte en torno a la anonimización de datos estructurados:** Se elaboró una revisión de la literatura sobre el campo de anonimización de datos, incluyendo la necesidad de privacidad de grandes volúmenes de datos, con lo cual se pudo obtener un estado actual del tema, permitiendo apropiar conocimientos relacionados con los algoritmos, operaciones y principios de anonimización para el desarrollo del sistema propuesto. Para la revisión enfocada en *Big Data*, se generó un artículo científico, susceptible de publicación.
2. **Ejecución de entrevistas:** Se realizaron entrevistas presenciales con la dirección de proyectos de CAOBA Javeriana, y entrevistas tanto presenciales como virtuales, con algunas de las empresas ancla de CAOBA, para obtener los requerimientos del sistema.
3. **Especificación de requerimientos del sistema:** Se desarrolló el documento de *Especificación de Requerimientos de Software (SRS)* que reúne sistemáticamente las necesidades del cliente del sistema, es decir la alianza CAOBA y sus empresas ancla.

1.3.2. Fase 2: Diseño y construcción

La segunda fase estuvo orientada a diseñar y construir el sistema, a partir de las necesidades identificadas en la alianza CAOBA. El objetivo del sistema es responder cuestiones del tipo:

1. ¿Cómo lograr la anonimización de los datos a divulgar contemplando información sensible e información que puede actuar como identificador?
2. ¿Cómo garantizar que los procesos de anonimización se puedan integrar fácilmente a los proyectos de analítica?
3. ¿Cómo validar que un conjunto de datos cumple con los principios de anonimización?

En esta etapa se diseñó una propuesta específica de anonimización orientada a la analítica de datos, la cual debió ser soportada a través del diseño del sistema. La implementación fue dirigida bajo un marco de referencia de metodología ágil, denominado SCRUM [26], [27], que es aplicado al desarrollo de productos de software donde es posible entregar de forma incremental fracciones funcionales del producto y obtener retroalimentación temprana por parte del cliente. Se trabajó con JIRA [28] como herramienta de administración y seguimiento de proyectos.

De acuerdo con el SRS, se consignaron todos los requerimientos agrupados en casos de uso en el *backlog* del proyecto, obteniendo una lista con las necesidades del cliente. De la misma forma, de acuerdo con el *Anexo 9.1. Requerimientos implementados por Sprint*, se determinó que se realizarían tres *sprints*, cada uno de los cuales tomó cuatro semanas, y por cada cierre de iteración se acordó una reunión de retrospectiva, donde se mostraron los avances, se recibió la retroalimentación acerca de los requerimientos implementados y se resolvieron dudas que

surgían después de cada iteración. Adicionalmente, se programaron reuniones diarias entre el equipo de desarrollo, al comienzo del día, donde se socializaba los avances e inconvenientes, con el fin de recibir apoyo o recibir sugerencias de cómo abordar el problema encontrado.

Cada iteración se planeó a través de la selección de los requerimientos a desarrollar, tomando como referencia la cohesión entre ellos y su previa priorización, los cuales fueron consignados en el *backlog*. Con la colaboración del cliente, se seleccionaron aquellos requerimientos que le brindaban mayor valor y a su vez permitían realizar entregas progresivas para la duración del proyecto. De esta forma, cada reunión de planeación realizada entre el equipo de desarrollo (Andrea Ortiz y Jonnathan Corredor) y el *Product Owner* (Alexandra Pomares, Gerente de Proyecto de la Alianza CAOBA y Profesora de Planta de la Pontificia Universidad Javeriana), se obtendría una lista de tareas que se debían desarrollar en cada iteración.

Esta fase permitió cumplir los objetivos específicos No. 2 y 3.

Actividades

1. **Diseño de los componentes del sistema y su interacción:** Se desarrolló el documento de *Descripción de Diseño de Software (SDD)* que define los aspectos de diseño del sistema propuesto, para responder a los requerimientos especificados en el documento SRS.
2. **Construcción del sistema:** De acuerdo con los documentos SRS y SDD, se implementó incremental e iterativamente el sistema propuesto, a través del marco de referencia SCRUM de metodologías ágiles, a fin de permitir la preparación de los datos de una compañía, para posteriormente aplicar técnicas de analítica de datos.
3. **Documentación del sistema:** El código fuente generado fue documentado, acorde con los lenguajes de desarrollo seleccionados para la implementación.

1.3.3. Fase 3: Validación

La última fase comprendió la validación de la propuesta a través de un caso de estudio que permitiera comprobar que el sistema cumple los requerimientos identificados. Se propusieron dos escenarios para abordarlo: el primero estuvo orientado a hacer la evaluación de la propuesta de anonimización utilizando el sistema construido para transformar dos conjuntos de datos reales, mientras el segundo se enfocó en revisar la pertinencia del algoritmo planteado para anonimizar un alto volumen de datos en una sola máquina.

Esta última fase permitió cumplir el objetivo específico No. 4.

Actividades

1. **Especificación de los datos con los cuales se generaron los 2 escenarios del caso de estudio:** Se seleccionaron dos conjuntos de datos reales, uno que representa el *benchmark* con el cual se han evaluado las propuestas de anonimización en la literatura, y una base entregada por CAOBA, sobre datos sensibles de uno de sus clientes.

2. **Ejecución de las pruebas del caso de estudio:** A través del sistema construido se ejecutaron las pruebas usando los conjuntos de datos especificados.
3. **Recopilación de resultados del ejercicio de validación:** Se realizó un compendio de la información generada por el sistema para concluir que se cumplieron los requerimientos definidos por la Alianza CAOBA.

1.4. Resultados

A través del desarrollo del trabajo de grado, se propuso generar los siguientes resultados, los cuales están alineados con las actividades descritas en la sección *1.3 Metodología* .

- Investigación realizada sobre el estado del arte en torno a la anonimización de datos en entornos "comunes" y de *Big Data*.
- Especificación de requerimientos del sistema (SRS).
- Descripción del diseño del sistema (SDD).
- Código fuente del sistema construido.
- Manuales de Usuario e Instalación.
- Memoria de Trabajo de Grado que incluye el desarrollo de la propuesta y su validación.

1.5. Propiedad intelectual

El presente Trabajo de Grado se realizó dentro del marco de los proyectos de investigación y consultoría de la Alianza CAOBA (*Centro de Excelencia y Apropiación en Big Data y Data Analytics*) que está bajo la dirección del Departamento de Sistemas de la Facultad de Ingeniería de la Pontificia Universidad Javeriana de Bogotá. Los derechos morales sobre los resultados obtenidos a partir de la presente investigación (artículos, documentos, software, manuales, análisis) son propiedad de los autores. Los derechos patrimoniales sobre tales resultados pertenecen a la Pontificia Universidad Javeriana, en representación de CAOBA, quien podrán disponer de ellos en el ámbito académico y productivo, de acuerdo con las necesidades y requerimientos de los proyectos de investigación y desarrollo que estén adscritos a la Alianza.

1.6. Descripción del documento

El presente documento se organiza de la siguiente manera: En la sección 2 se presenta el marco teórico asociado a los términos y conceptos que permiten comprender el contexto de anonimización. Posteriormente, en la sección 3 se expone la investigación sobre los trabajos relacionados que permiten entender qué se ha hecho hasta el momento y cómo ello aporta al desarrollo del presente trabajo. En la sección siguiente, 4, se refleja la propuesta de anonimización orientada a la analítica de datos. A través de la sección 5 se evidencia el diseño e implementación del sistema que permite soportar la propuesta planteada. La sección 6 presenta la validación de la propuesta, a través del uso del sistema implementado. Finalmente, en la sección 7 se genera un conjunto de conclusiones y planteamiento de trabajos futuros.

2. MARCO TEÓRICO

Los proyectos de analítica suelen estar enmarcados en alguna metodología que permita generar un proceso desde la obtención de los datos hasta la entrega de resultados y su validación. Bajo este contexto existen diferentes metodologías, entre ellas CRISP-DM [9], KDD [10] y SEMMA [11]. Cada una tiene como propósito analizar datos con diferentes objetivos de negocio, para generar resultados que motiven nuevas decisiones estratégicas que redunden en beneficios para una compañía. CRISP-DM es una de las metodologías más conocidas en este ámbito y es utilizada en la Alianza CAOBA para el desarrollo de los proyectos de analítica. Desde esta perspectiva se comprende la necesidad de anonimización como parte de la preparación de los datos que posteriormente serán minados a través de modelos descriptivos o predictivos.

2.1. Metodología CRISP-DM

CRISP-DM es la metodología para el desarrollo de proyectos de Analítica de Datos, donde se manejan seis fases no completamente secuenciales, como modelo de referencia: *Entendimientos del negocio*, *Entendimiento de los Datos*, *Preparación de los Datos*, *Modelamiento*, *Evaluación* y *Despliegue* [9].

- La fase de **Entendimiento del Negocio** se enfoca en la comprensión de los objetivos del proyecto desde la perspectiva de la Organización, de modo que las necesidades reales puedan ser traducidas en un problema de Minería de Datos.
- La fase de **Entendimiento de los Datos** implica su recolección y análisis inicial para identificar problemas de calidad, hacer hallazgos incipientes y determinar el curso a seguir.
- La fase de **Preparación de los Datos** involucra todas las actividades requeridas para generar un conjunto de datos limpio a partir de los datos en bruto que pueden requerir actividades como la anonimización, integración, eliminación de duplicados, transformaciones, análisis de datos atípicos, selección de atributos que se modelarán, entre otros.
- En la fase de **Modelamiento** se seleccionan y aplican varias técnicas de analítica (minería), a fin de calibrar los parámetros para obtener los resultados que mejor se ajusten a las necesidades del negocio.
- La fase de **Evaluación** permite verificar cuál de los modelos construidos se adapta mejor a las necesidades del negocio, además de validar su calidad, obteniendo una propuesta útil para el cliente.
- La fase de **Despliegue** apoya el proceso de implementación del modelo al interior de la organización, lo cual puede ser tan sencillo como entregar un reporte con los resultados obtenidos de la técnica de analítica, pero también podría ser la puesta en producción de un sistema automatizado que permita, por ejemplo, periódicamente generar resultados usando datos que la organización obtiene día a día.

Bajo este panorama, el presente proyecto busca apoyar la fase de *Preparación de los Datos* de cualquier proyecto de analítica, en el contexto de la anonimización de los datos que desean ser transformados para poder ser utilizados con propósitos analíticos. Esto implica, el aseguramiento de la confidencialidad y privacidad de los datos.

2.2. Privacidad y confidencialidad

La ***privacidad*** [29] se refiere al derecho que tienen los individuos de controlar o influenciar qué información de ellos puede ser recolectada o almacenada y qué podría llegarse a divulgar. Por su parte, la ***confidencialidad*** [30] se entiende como la protección en contra del acceso no autorizado, al contenido de personas o entidades; por lo anterior, la confidencialidad incluye la privacidad. De allí, que el interés en estas temáticas radique en cómo generar un balance entre la confidencialidad y/o privacidad de los datos, al mismo tiempo que se siga garantizando su utilidad.

2.3. Riesgos de divulgación y mecanismos de seguridad

Cuando se pasa por alto el aseguramiento de los datos, pueden desencadenarse tres riesgos de divulgación a conocer: divulgación de la *Identidad*, de *Atributo* y de *Inferencia* [31].

- **Identidad (Singularización)**: Un atacante podría asociar a un individuo, mediante información externa, identificándolo plenamente.
- **Atributo (Vinculabilidad)**: Implica que un atacante pueda asociar una nueva característica a un individuo previamente identificado.
- **Inferencia**: Un atacante podría deducir con algún porcentaje de probabilidad, el valor de algún atributo a partir de otros datos externos.

Con el ánimo de generar mecanismos de seguridad, se identifican cuatro medidas [30]:

1. **Control de Acceso**: Se refiere a restringir el acceso a los datos, únicamente a quienes tengan la expresa autorización para ello.
2. **Control de Flujo**: Implica prevenir que información fluya a usuarios no autorizados para conocer los datos.
3. **Ciframiento**: Aproximación para proteger los datos durante su transmisión y almacenamiento.
4. **Control de Inferencia**: Se asocia a impedir la inferencia o deducción de información a partir de los datos privados/confidenciales.

De esas cuatro medidas, la tercera y cuarta proponen transformaciones directamente a los datos, para asegurar que, sin importar la red, las aplicaciones, los puntos de acceso, etc., tal información esté protegida en sí misma. En el caso del ***Ciframiento***, se manejan técnicas que convierten los datos en ilegibles, los cuales sólo se pueden descifrar al conocer la llave asociada al proceso de transformación [32]. En el caso del ***Control de la Inferencia*** se establece la técnica de ***Anonimización***, cuyo propósito es esconder la identidad y/o los datos sensibles de los dueños de los registros, partiendo de que estos datos serán utilizados en posteriores análisis [4].

2.4. Anonimización

En el contexto de anonimización es necesario realizar la distinción entre los tipos de atributos considerados en un conjunto de datos, pues ello permitirá enfocar adecuadamente el esfuerzo

para proteger aquellos que podrían generar vulnerabilidad al ser expuestos a un atacante. Por lo anterior, en un conjunto de datos deben distinguirse los siguientes atributos [33]:

- **Identificadores**: Datos que permitan distinguir inequívocamente a una entidad[1] (cédula, etc.).
- **Pseudo-identificadores**: Datos públicos que en conjunto podrían permitir identificar una entidad (conjunto de: sexo, estado civil, barrio, etc.)
- **Sensibles:** Datos de carácter personal y confidencial que implican la protección en contra del acceso no autorizado debido a que afectan la intimidad del titular y su tratamiento podría generar discriminación [2] (diagnóstico médico, salario, transacciones, etc.).
- **Comunes (No sensibles):** Datos comunes que no hacen parte de los tres tipos previamente descritos.

2.4.1. Operaciones

Con el propósito de anonimizar los datos, existen diferentes operaciones que permiten transformarlos de forma que se procure el aseguramiento de la privacidad y confidencialidad de los mismos. Las operaciones ***Perturbativas*** distorsionan en cierto grado los datos originales, mientras que las ***No Perturbativas*** se limitan a suprimir o reducir el detalle de los datos, pero no los alteran de forma significativa [33]. En el contexto del presente proyecto, y en general para los objetivos de analítica, se centran los esfuerzos en la aplicación de operaciones *No Perturbativas*, con lo cual se pueda continuar asegurando que la calidad de los modelos planteados sigue siendo útil, y los datos válidos, para la toma de decisiones. Entre las operaciones no perturbativas se encuentran las siguientes [33], [34]:

- ***Top-Coding* y *Bottom-Coding***: Consiste en definir un límite superior (*top*) o inferior (*bottom*) para cada atributo protegido. Cuando existe un valor por encima (*top*) o por debajo (*bottom*) del límite, el dato se modifica por una etiqueta que indica que los valores son mayores (*top*) que el límite superior o menores (*bottom*) que el límite inferior.
- **Supresión**: Implica eliminar todos los valores del atributo, es decir, una columna del conjunto de datos.
- **Generalización**: Se enfoca en generar un valor de granularidad menos fino o detallado que el original.

Una primera aproximación para evitar poner en riesgo los datos es suprimir aquellos atributos que se consideran *identificadores* y con los cuales se puede reconocer inequívocamente un registro, por ejemplo, una cédula, un NIT, etc. [35]. Esta aproximación es válida y poco compleja puesto que, además de eliminar datos privados, de antemano se conoce que los datos identificadores no son útiles para un ejercicio de analítica dado que no es posible obtener pa-

[1] Entidad se refiere a personas naturales o jurídicas.

trones a partir de estos datos que son únicos [36]. No obstante, esta aproximación no es suficiente puesto que existen otros atributos, los *cuasi-identificadores*, que no son identificadores, pero que podrían llegar a serlo en caso de que se tenga conocimiento adicional y se pueda inferir a quién corresponde un registro. Lo anterior es conocido como el ataque de singularización [31], en el cual, es posible relacionar un registro con otra fuente de datos, ya sea privada o pública, y asociar los datos a través de los cuasi-identificadores, permitiendo identificar a alguna entidad en particular. Por ejemplo, un hospital puede querer publicar la información de sus pacientes con fines de investigación, como se observa en la parte izquierda de la Figura 3. Diversos atacantes podrían tener acceso a otra fuente de datos, como se observa en la parte derecha de la Figura 3; al asociar los atributos *Job*, *Sex* y *Age* de ambas tablas, podrían identificar a Doug por ser el único hombre abogado de 38 años presente en el conjunto de datos, infiriendo que padece de VIH. Por tal motivo, aunque ninguno de estos tres atributos es considerado un identificador, la combinación de estos cuasi-identificadores sí podría permitir identificarlos e inferir nuevo conocimiento, poniendo en riesgo su privacidad.

(a) Patient table

Job	Sex	Age	Disease
Engineer	Male	35	Hepatitis
Engineer	Male	38	Hepatitis
Lawyer	Male	38	HIV
Writer	Female	30	Flu
Writer	Female	30	HIV
Dancer	Female	30	HIV
Dancer	Female	30	HIV

(b) External table

Name	Job	Sex	Age
Alice	Writer	Female	30
Bob	Engineer	Male	35
Cathy	Writer	Female	30
Doug	Lawyer	Male	38
Emily	Dancer	Female	30
Fred	Engineer	Male	38
Gladys	Dancer	Female	30
Henry	Lawyer	Male	39
Irene	Dancer	Female	32

Figura 3. Ilustración del ataque de vinculabilidad de registros, tomado de [36].

De esta manera se comprende que los atributos que por lo general se anonimizan son los identificadores y los cuasi-identificadores, donde usualmente se eliminan o sustituyen los primeros y se aplican operaciones más robustas sobre los segundos, como la generalización.

2.4.2. Principios

Con el objetivo de prevenir el ataque de *Singularización* por la asociación de registros, Samarati y Sweeney [37] propusieron el principio denominado *k-anonymity* en el cual se asegura que deben existir conjuntos de mínimo *k* número de registros que compartan la misma combinación de valores de atributos cuasi-identificadores. Formalmente, este principio se define así:

"Sea T(A1,. . . ,An) *una tabla y* QI *un conjunto de cuasi-identificadores asociados con ella. Se dice que* T *satisfice k-anonymity con respecto a* QI*, si y solo si cada secuencia de valores en* T[QI] *aparece al menos con k ocurrencias en T[QI]"* [38].

De este modo, no será posible hacer una asociación explícita de registros de la tabla con los datos privados y otras fuentes de datos externas, puesto que siempre existirán conjuntos de registros que posean los mismos valores en sus cuasi-identificadores, eliminando los riesgos de privacidad asociados. Este principio permite validar que los registros de un conjunto de datos quedaron anonimizados y por tanto la probabilidad de asociar una víctima (de un ataque) a un registro, será a lo sumo de $1/k$.

Una posible aproximación para lograr conseguir transformar los datos de manera que se cumpla con el principio *k-anonymity* es aplicar la operación de generalización sobre los datos, con lo cual éstos son llevados de un nivel de granularidad fino a otro menos detallado [33], [34], mediante una jerarquía o taxonomía, como el ejemplo que se propone en la Figura 4.

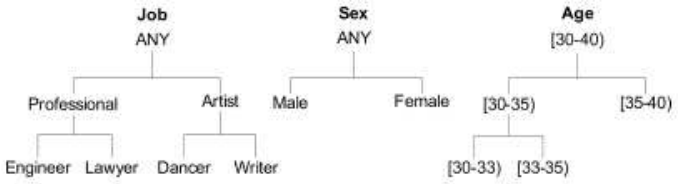


Figura 4. Jerarquías para generalizar los datos propuestos en la Figura 3, tomado de [36].

De tal manera, se podría crear un conjunto de datos anonimizado donde no es posible distinguir registros en particular, pues se pueden generar conjuntos de mínimo *k* registros que comparten la misma combinación de atributos cuasi-identificadores. En la Figura 5 se evidencia un ejemplo de datos anonimizados con k = 3 para el caso presentado en la Figura 3.

Job	Sex	Age	Disease
Professional	Male	[35-40)	Hepatitis
Professional	Male	[35-40)	Hepatitis
Professional	Male	[35-40)	HIV
Artist	Female	[30-35)	Flu
Artist	Female	[30-35)	HIV
Artist	Female	[30-35)	HIV
Artist	Female	[30-35)	HIV

Figura 5. Conjunto de datos anonimizado con k = 3 para los datos propuestos en la Figura 3 y la jerarquía de la Figura 4, tomado de [36].

De igual forma, existen otros principios de anonimización que imponen, de acuerdo con las características de los datos, incrementar la privacidad sobre ellos, lo cual supone un desafío mayor al intentar generar un balance adecuado entre privacidad y utilidad. ***L-diversity*** [39] establece que para cada grupo de *k* registros del principio previo, deben existir al menos *l* valores bien representados para el atributo sensible, es decir que existen al menos *l* valores diferentes para cada bloque de *k* registros y por tanto es posible evitar el riesgo de *Vinculabilidad* e *Inferencia.* ***T-Closeness*** [40], por su parte, propone que la distribución de valores del atributo sensible en cada bloque de *k* registros, de acuerdo con el primer principio, sea cercano a la distribución del atributo en todo el conjunto de datos.

Es así, que con el fin de aplicar las operaciones y evaluar la seguridad de los datos, se requiere de algoritmos específicos que permitan anonimizar, de acuerdo con los principios expuestos. Existen diversas propuestas como *Datafly* [41], *Incognito* [42], *Mondrian* [7], *Allmin* [38], *Mingen* [43], *Top-Down Specialization* [44] , *K-Optimize* [45], entre otros, los cuales evidencian los pasos necesarios para confluir en la protección de los datos. En la siguiente sección, se profundiza en las aproximaciones generadas por diferentes autores para alcanzar los principios de anonimización expuestos.

3. ESTADO DEL ARTE

El enfoque del presente proyecto radica en la generación de un proceso de anonimización que propenda hacia la preservación de la utilidad de los datos para posteriores procesos de analítica. *PPDM* (*Privacy Preserving Data Mining*) [46] es un acrónimo y una ideología que ha tenido una importante acogida dentro del contexto de la preservación de la privacidad en los ejercicios de analítica o minería de datos. Este término hace alusión al proceso de transformación de los datos de forma tal que posteriormente se puedan aplicar algoritmos de minería sin perder la utilidad de los datos y sin comprometer la seguridad de los datos privados y confidenciales. El objetivo es entonces, a partir de la compresión de las operaciones y principios de anonimización existentes, de acuerdo con la sección *2. Marco teórico,* encontrar la ruta adecuada que permita mantener el balance entre privacidad y utilidad de los datos. Para ello, la exploración del estado del arte ha estado enfocada en la búsqueda de algoritmos que logren alcanzar los objetivos trazados tanto en conjuntos de datos "comunes" como en entornos de grandes volúmenes.

El principio de anonimización más reconocido en la literatura es *k-anonymity* [38], y de él se desprenden otros modelos más estrictos, que contemplan situaciones particulares donde es necesario profundizar en los datos para proveer mayor privacidad. Para proyecto se tomó como punto de partida el principio de anonimización mencionado, sobre el cual han versado diferentes algoritmos. A continuación se evidencia el estado del arte en torno a la anonimización de conjuntos de datos convencionales, es decir, que no constituyen un alto volumen de datos y por tanto no evalúan y analizan desafíos desde el punto de vista computacional.

3.1. Algoritmos tradicionales

En las últimas décadas se han desarrollado diversos algoritmos en torno a esta temática; tres con gran trayectoria y ampliamente conocidos dentro del contexto de la anonimización son *Datafly* [41], *Incognito* [42] y *Mondrian* [7], los cuales buscan cumplir el principio de *k-anonymity*. Éstos fueron comparados sistemáticamente por Ayala-Rivera *et al*. [6] en una investigación que comparó sus resultados en términos de eficiencia (tiempo y consumo de memoria) y utilidad de los datos (métricas de *Indistinguibilidad*, *Pérdida generalizada de información* y *Tamaño promedio de la clase de equivalencia*).

Datafly [41] es un algoritmo voraz que lleva a cabo operaciones de generalización sobre cada atributo por separado, para anonimizar los atributos cuasi-identificadores. En este caso, las jerarquías de generalización deben ser entregadas por el usuario al algoritmo. Se cuentan las frecuencias sobre el conjunto de atributos a anonimizar y si aún no se cumple el *k-anonymity*, se generaliza el atributo cuasi-identificador que tenga la mayor cantidad de valores distintos. Esto se hace sucesivamente hasta lograr cumplir el principio. El algoritmo permite asegurar los datos mediante el principio, pero no logra proveer una generalización mínima que conlleve a una baja pérdida de información. De acuerdo con la evaluación realizada en [6], la anterior afirmación se corrobora pues este algoritmo obtuvo resultados menos favorables para las métricas de pérdida de información con respecto a los otros dos algoritmos, *Incognito* y *Mondrian*.

Incognito [42] es otro algoritmo voraz que construye una red de generalización, la cual es recorrida a través de una aproximación *bottom-up*. En este caso, las jerarquías de generalización

también deben ser entregadas al algoritmo. Se define un conjunto de generalizaciones válidas por cada atributo a través de la profundidad de la red. De este modo, se revisan todas las posibles generalizaciones hasta encontrar aquella óptima donde se consiga asegurar los datos, pero exista la menor pérdida de información posible, de acuerdo con la definición de alguna métrica asociada. Aunque se podría llegar a una solución factible, este algoritmo es inviable computacionalmente (para un volumen considerable de datos y en presencia de una importante cantidad de cuasi-identificadores) en términos de recursos de memoria y tiempo, debido a que debe construir todas las posibles generalizaciones para escoger la que permitiría perder menor información. De acuerdo con la comparación realizada en [6], este algoritmo tuvo el mayor tiempo y mayor consumo de memoria en las pruebas realizadas sobre diferentes conjuntos de datos y combinaciones de cuasi-identificadores, pues como es de esperarse, tiene una complejidad exponencial del orden $O(2^n)$, donde *n* corresponde a la cantidad de cuasi-identificadores. De igual forma, dado que debe computar todas las generalizaciones posibles, es intensivo en uso de la memoria, por lo cual podría llegar a ser restrictivo en ambientes con recursos reducidos.

Mondrian [7] también es un algoritmo voraz, pero a diferencia de los anteriores, es multidimensional, es decir que aplica la generalización sobre varios atributos al tiempo, al hacer recursivamente particiones del espacio de domino de los atributos cuasi-identificadores, hasta lograr generar regiones que contienen al menos *k* registros (clases de equivalencia), logrando cumplir el principio de anonimización, como se muestra en la Figura 6 para un *k* igual a 2.

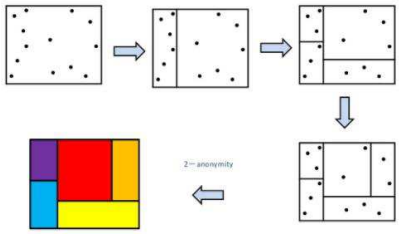


Figura 6. Representación gráfica del algoritmo Mondrian para 2-anonymity, tomado de [7].

El algoritmo inicia seleccionando dentro del conjunto de cuasi-identificadores, uno de los atributos, a través del cual se generará la partición de los datos en dos conjuntos. Se selecciona aquel que tiene el más amplio rango de valores (normalizados) y que permite dividir los registros (filas) sin violar el principio de *k-anonymity*. Es decir, si k = 2 y se tiene un conjunto de 3 registros, no se podría hacer una nueva partición pues quedaría un conjunto de 2 y otro de 1, con lo cual no se cumpliría *2-anonymity*. De este modo, habiendo seleccionado uno de los cuasi-identificadores, *Mondrian* utiliza una aproximación de corte basada en la mediana del conjunto de datos de tal atributo, de modo que, desde el mínimo valor hasta la mediana, se forma una nueva partición sobre la cual se vuelve a aplicar el algoritmo recursivamente; y desde el siguiente valor a la mediana, hasta el máximo, se forma una segunda partición sobre la cual se opera de igual manera. Cuando se han generado todos los grupos de mínimo *k* registros, se reemplaza el valor original en cada uno de los cuasi-identificadores por algún estadístico de centralidad calculado a partir de los datos del atributo que quedaron dentro del conjunto de al menos *k* registros. Al tener un conjunto de, por ejemplo 2 registros, si el valor del cuasi-identificador del primer registro es *A* y el valor del cuasi-identificador para el segundo registro es *B*, entonces el valor anonimizado para ambos registros podrá ser una etiqueta que evidencia un

rango [*A*, *B*]. Sin embargo, dado que esto modificaría el tipo de dato numérico a categórico, el valor anonimizado para ambos registros podría ser cualquier estadístico de centralidad como la media, mediana, u otra, entre *A* y *B*.

Este algoritmo, de forma automática genera el cumplimiento del principio de anonimización, además de no requerir de la definición previa de una jerarquía de generalización, como en los otros dos algoritmos; por lo anterior, esta aproximación de generalización basada en particiones, es útil para los escenarios donde se pretende anonimizar datos numéricos. Además de ello, este algoritmo no generaliza innecesariamente pues lo hace estrictamente para cumplir con el principio de anonimización. De acuerdo con la investigación hecha por Ayala-Rivera *et al.* [6], *Mondrian* obtuvo en general los mejores resultados en comparación con *Incognito* y *Datafly*, respecto a la métrica denominada "*Indistinguibilidad*" (qué tan indistinguible es un registro respecto a otros, al asignar una penalización a cada registro, equivalente al tamaño de la clase de equivalencia) y la métrica denominada "*Tamaño promedio de la clase de equivalencia*" (qué tan bien, la creación de clases de equivalencia, se aproxima al mejor caso, donde cada registro es generalizado en una clase de equivalencia de *k* registros). El estudio, sin embargo, arrojó resultados menos favorables para *Mondrian* respecto a la métrica de "*Pérdida generalizada de información"* (captura de la penalización en que se incurre al generalizar un atributo específico, cuantificando la fracción de los valores del dominio que han sido generalizados), donde la distribución de probabilidad de sus cuasi-identificadores tiene un sesgo y por tanto el criterio de partición a través de la mediana no genera resultados que logren una baja pérdida de información. No obstante, para distribuciones con poco sesgo, el algoritmo genera mejores resultados que los demás algoritmos. Se concluye que la utilidad de los datos después de anonimizar, se ve influenciada por la distribución de los atributos y el criterio de corte utilizado.

3.2. Anonimización en *Big Data*

De acuerdo con las necesidades actuales de la industria, y por supuesto de CAOBA, respecto al manejo de información en un contexto de *Big Data*, se realizó un proceso de investigación adicional que permite complementar los hallazgos obtenidos a través de los algoritmos tradicionales, expuestos en la sección previa. En el documento anexo *Estado del Arte de Anonimización en Big Data* se presenta una versión publicable de la investigación realizada sobre las aproximaciones de anonimización en *Big Data*, donde se evidencia un mayor grado de profundidad.

La evolución en las tecnologías de la información, así como la capacidad de almacenamiento y procesamiento, han permitido en los últimos tiempos entrar en la era del *Big Data*, caracterizada por desafíos que versan en cinco frentes. Debido a la riqueza de los datos y lo que de ellos puede analizarse, se realiza ahora el almacenamiento de un gran ***Volumen*** de datos recolectados desde una ***Variedad*** de fuentes con diversos formatos estructurados y no estructurados, los cuales llegan con gran ***Velocidad***, en forma de flujos continuos e incesantes que esperan ser analizados en el menor tiempo posible para generar ***Valor*** a las organizaciones en el ahora, por lo cual deben lidiar con este entorno rápido y cambiante, pero sin perder de vista la ***Veracidad*** de los datos, sus análisis y su generación de información y nuevo conocimiento [47].

Esta gran cantidad de datos debe ser gestionada de la manera más eficiente con el fin de generar ejercicios de analítica útiles para la toma de decisiones a corto plazo. Sin embargo, no por las características del *Big Data*, deja de ser relevante el proceso de anonimización para asegurar la privacidad de los entes de los cuales se obtiene, almacena y procesa información día a día. Por tal razón se indagó sobre las investigaciones llevadas a cabo en el ámbito de la preservación de la privacidad de los datos, para entornos con las características de *Big Data*. La revisión posibilitó la identificación de cuatro categorías de trabajos relacionados en este ámbito, permitiendo entender en qué se han centrado las investigaciones en los últimos diez años: (i). *Enfoques conceptuales*, donde se presenta una visión completa que permite entender la relevancia, importancia y necesidad del manejo responsable de la información en diferentes etapas de la minería de datos y la inteligencia de negocios; (ii). *Arquitecturas*, donde se definen las características lógicas y de implementación que deben ser tenidas en cuenta en un entorno de protección de datos; (iii). *Herramientas*, las cuales presentan aplicaciones tangibles de sistemas escalables en diferentes contextos donde es necesario entregar un adecuado manejo de la información; y finalmente, (iv). *Algoritmos,* que permiten la anonimización de datos, presentando paso por paso, las aproximaciones necesarias para integrar operaciones y principios.

Entendiendo desde lo más general a lo más particular, en la Figura 7 se presenta gráficamente el hallazgo generado a partir de los trabajos relacionados.

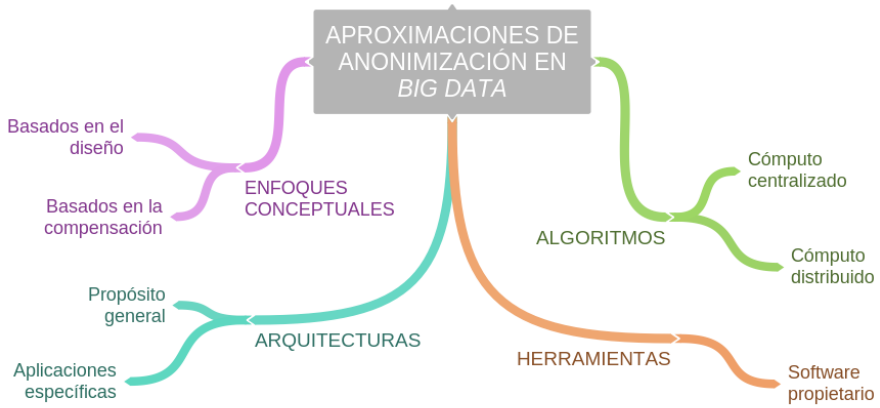


Figura 7. Clasificación de aproximaciones de anonimización Big Data (Elaboración propia).

A continuación se profundiza en cada una de estas categorías con el objetivo de comprender los desarrollos actuales en torno a la anonimización en contextos de *Big Data.*

3.2.1. Enfoques conceptuales

Los ***Enfoques conceptuales*** se refieren a cómo abordar todo el proceso de anonimización de datos. La forma común en la cual los investigadores se aproximan a la anonimización es directamente proponiendo una arquitectura, una herramienta o un algoritmo, sin embargo, existe otro conjunto de investigaciones que colocan esta escena en contexto, permitiendo presentar enfoques conceptuales basados en el proceso. De este modo, se analiza el panorama general de

la aplicación, la vista de desarrollo de la solución, los actores y las características relevantes de cada paso, con el fin de abordar la anonimización desde una perspectiva holística. Dentro de esta clasificación están los enfoques ***basados en el Diseño***, donde los datos son analizados desde el inicio del proceso para evitar problemas de seguridad de la información en etapas tardías. Los enfoques ***basados en la Compensación***, se refieren a la aplicación de la teoría de juegos para analizar la relación entre la pérdida de información y el uso de los datos dado en cada etapa del proceso de *PPDM*.

3.2.1.1. Basados en el diseño

Monreale *et al.* [48] proponen que en vez de aplicar una aproximación reactiva para asegurar la anonimización de los datos, mejor se plantee el análisis desde el inicio del proceso de analítica con el fin de seguir una aproximación proactiva. Los autores presentan una instanciación del paradigma denominado *"Privacy-by-design"*, orientado a la analítica en *Big Data*. De acuerdo con la propuesta, para inscribir el paradigma de privacidad basado en el diseño desde el principio, los procesos de analítica de datos deberían ser diseñados bajo tres supuestos. El primero es identificar los datos personales sensibles que serán sujetos de análisis. Aquí es necesario ahondar en la naturaleza de los datos que serán protegidos porque, por ejemplo, métodos apropiados para el análisis de redes sociales podrían no ser adecuados para otros tipos de técnicas o datos. El segundo supuesto está relacionado con reconocer los posibles ataques y los modelos subyacentes que pueden describir tales vulneraciones en términos de conocimiento y propósito del adversario. Es importante intentar proponer escenarios para diferentes atacantes cuyo conocimiento previo podría permitirle inferir nueva información y violar la privacidad. El último supuesto describe las consultas analíticas que serán respondidas con los datos. Con el objetivo de encontrar un aceptable balance entre privacidad y utilidad, es fundamental considerar el tipo de respuestas que se esperan obtener desde los datos, lo cual permite entender qué propiedades de los datos deben preservarse, a pesar de las transformaciones propuestas. En esta vía, si datos específicos son eliminados pero necesarios para aplicar analítica, los subsiguientes pasos no serán útiles en el diseño.

3.2.1.2. Basados en la compensación

Yassine *et al.* [49] ofrecen una aproximación de teoría de juegos para balancear los beneficios de ganar acceso a los datos y su privacidad en un contexto donde se recoge información de sensores de medición de consumo energético digitales. Estos aparatos logran recoger datos más completos y detallados sobre consumo de gas, agua y electricidad, que los medidores analógicos tradicionales; por ello, son una fuente de información bastante útil para analizar y promover, por ejemplo, diferentes políticas públicas. No obstante, revelar algunos de los datos recolectados puede ser sinónimo de violación de la privacidad de los individuos acreedores de la información. Por ello, los autores proponen un mecanismo de compensación que busca generar un balance de ganancia para un analista de datos desde el punto de vista de obtención de información, mientras se reconoce monetariamente al acreedor de los datos, quien pierde privacidad en su información. El acreedor, no obstante, con el ánimo de no perder completamente la privacidad en sus datos, puede generar una transformación sobre ellos, basada en el concepto denominado "*Differential Privacy"*, donde se agrega ruido a los atributos a través de una función de distribución, usualmente de *Laplace*, para modificar en cierto nivel los datos, pero sin

alterar drásticamente su utilidad. El éxito de la compensación radica entonces en el balance entre la entrega de información útil por parte del acreedor de los datos, respecto al precio que está dispuesto a pagar el analista que desea conocer tal información privada.

3.2.2. Arquitecturas

Las ***Arquitecturas*** exponen la composición lógica de los sistemas destinados a anonimizar datos. Esos componentes son requeridos para la implementación de los artefactos de anonimización en entornos de grandes volúmenes de datos. En este caso fue posible identificar dos tipos de arquitecturas. Las de ***Propósito general***, plantean componentes lógicos y genéricos que permiten agrupar funcionalidades en un contexto generalizable y escalable a diferentes áreas y contextos, por lo cual no están sujetas a una aplicación en particular. Por el contrario, las arquitecturas asociadas a ***Aplicaciones específicas***, son aquellas que sólo están disponibles para un contexto de aplicación específico. Los componentes descritos en estos casos están completamente acoplados a la aplicación subyacente, de modo que su escalabilidad no es transparente. A pesar de ello, las arquitecturas proveen lineamientos generales orientados a la organización de los componentes con el objetivo de generar un sistema que anonimice los datos existentes.

3.2.2.1. Propósito general

Cho *et al.* [50] inicialmente presentan el *framework* comúnmente conocido para entornos de *Big Data*, compuesto por los siguientes componentes: (i). *Recolector de datos*, (ii) *Sistema de archivos de Big Data*, (iii). *Análisis de Datos* y *Módulo de Procesamiento,* y (iv). *Presentación*. Todos estos componentes producen salidas que son las entradas para los componentes subsiguientes, en el orden presentado. Con el objetivo de mejorar la seguridad, dados los recientes escenarios de *Big Data*, los autores proponen una arquitectura lógica de seguridad que incluye dos procesos adicionales al *framework* presentado, para enmascarar y eliminar los datos sensibles: una capa de *pre-filtro* y otra de *post-filtro*. En ambas, el objetivo es identificar los datos que deberían ser enmascarados para prevenir su re-identificación y violación de la privacidad. La primera capa se sitúa después de la fase de recolección, antes del almacenamiento; los posibles datos sensibles son detectados para ser eliminados en este *pre-filtro*. La segunda capa de *post-filtro* se localiza después del análisis de los datos, justo antes de la fase de presentación o visualización; los datos no sensibles, pero posiblemente cuasi-identificadores (información de comportamiento, preferencias, etc.) son enmascarados para prevenir su descubrimiento en la fase de Presentación. En realidad, la propuesta es más lógica que técnica, así que no hay principios de anonimización para alcanzar un nivel mínimo de privacidad. Dado que se menciona enmascaramiento y eliminación, sería imposible aplicar procesos analíticos debido a la completa pérdida de información. Aunque la arquitectura es planteada como parte de un entorno de *Big Data*, no hay evidencia de conceptos que lo diferencien de un entorno común y corriente.

Por su parte, Zhang *et al.* [51] propusieron Sac-FRAPP, un *framework* escalable para la preservación de privacidad sobre *Big Data* en la nube. Este *framework* es presentado como una arquitectura compuesta por cuatro módulos: (i). *Interfaz de especificación de privacidad* (*PSI*, por sus siglas en inglés), (ii). *Anonimización* (*DA*, por sus siglas en inglés), (iii). *Actualización de datos* (*DU*, por sus siglas en inglés) y (iv). *Gestión de conjuntos de datos anónimos* (*ADM*, por sus siglas en inglés). El módulo DA aplica las operaciones de anonimización de acuerdo

con los principios (*k-anonymity, l-diversity* y *t-closeness),* y parámetros como: el tipo de modelo de minería que será aplicado después de anonimizar, los umbrales correspondientes a los principios, y los algoritmos de anonimización. Todos estos parámetros son entregados por el usuario final usando el módulo *PSI.* La anonimización es llevada a cabo mediante operaciones de generalización (*full-domain, sub-tree, multi-dimensional* y *cell*), utilizando tareas *Map-Reduce* sobre el sistema de archivos distribuido de Hadoop. Así, el módulo *DU* es responsable de anonimizar los nuevos registros entrantes y ajustar los existentes para seguir cumpliendo los principios, tratando todos los datos como un solo conjunto que se va alimentando constantemente. El componente *ADM* permite la gestión de los datos, soportando al módulo *DU* para evitar re-cálculos cuando se ha aplicado la anonimización en conjuntos de datos pre-existentes.

El *framework* fue probado mediante una prueba de concepto que fue implementada sobre la nube en *OpenStack* [52] y *HDFS* [53], usando una implementación de código abierto de *Map-Reduce*. Los autores aseguran haber anonimizado un conjunto de datos de gran escala, mas sólo fueron incluidos alrededor de 30 mil registros, con máximo 500 MB de datos, lo cual tomó aproximadamente 2,7 horas. Bajo este escenario no es posible dilucidar los aspectos de actualización y manejo de datos, al trabajar con un conjunto de información en lote. De igual manera no es claro el entendimiento del contexto de aplicación de los algoritmos bajo esta arquitectura orientada a *Big Data*, puesto que no se profundizó en cómo son manejados tales algoritmos respecto a la necesidad de paralelización para anonimizar datos distribuidos.

3.2.2.2. Aplicaciones específicas

Pàmies-Estrems *et al.* [54] propusieron una arquitectura del lado del servidor que permite a los motores de búsqueda web anonimizar archivos *log* (textos) en un ambiente de flujos continuos de datos. La arquitectura provee escalabilidad, velocidad de procesamiento, bajo consumo de recursos, transparencia y modularidad. La arquitectura está compuesta por: (i). *Clasificador*, (ii). *Anonimizador* y (iii). *Perfilador*. El *Clasificador* utiliza los *logs* de búsqueda como principal insumo para clasificarlos en un tópico de tipo categórico. Con este objetivo, el componente aplica Procesamiento de Lenguaje Natural sobre los textos de cada una de las consultas que llegan de forma continua, resultando un conjunto de unidades semánticas, después de la eliminación de palabras que cumplan con ser identificadores o cuasi-identificadores. Posteriormente, una base de datos de recomendaciones es accedida para corregir posibles errores de ortografía, para finalmente calificar las unidades semánticas utilizando una base de datos de entropías donde se encuentra el término que sea más específico por lo que podría proveer mayor información sobre la consulta efectuada por el usuario. Tal unidad semántica es buscada en la base de datos de categorías para etiquetar la consulta con un tópico predefinido. El *Anonimizador*, recibe por cada *log* procesado, un usuario y el texto de la consulta realizada. El componente recibe datos hasta que el umbral *k* es alcanzado en alguna de las bolsas de categorías (lo cual es similar al principio de *k-anonymity*). En ese momento, el componente toma aleatoriamente un usuario y el texto de una consulta y construye un nuevo *log* de consulta anonimizado, haciendo una permutación. Esto significa que la generalidad es preservada mientras la especificidad se elimina. Por último, el *Perfilador*, utiliza los resultados obtenidos para actualizar el perfil de la persona que envió la consulta. Cada perfil es construido al asignar un porcentaje de pertenencia a cada categoría, de acuerdo con el número de consultas ejecutadas por el usuario.

Basándose en la arquitectura propuesta, los autores desarrollaron un sistema usando Python 2.7 y MongoDB 3. Utilizaron un conjunto de datos de AOL Search Data Collection con 36 millones de *logs* de consultas, correspondientes a un período de 3 meses de actividad real. Los resultados demuestran que la privacidad es mantenida pues se utilizan operaciones de permutación que no permiten fácilmente re-identificar los *logs* originales. Aunque la arquitectura plantea elementos evidentes de un contexto de *Big Data*, la prueba no permite identificar claramente los puntos clave de la propuesta para esta nueva era de datos respecto a la velocidad, por ejemplo.

3.2.3. Herramientas

Las ***Herramientas*** se refieren a la aplicación práctica de los principios, operaciones, algoritmos y/o metodologías de anonimización. Una categorización razonable supondría tener software propietario y código abierto, sin embargo, la investigación conducida mostró que las únicas soluciones disponibles en la literatura son pagas. En este caso, el ***Software propietario*** es una herramienta de anonimización paga. Es importante recalcar que una herramienta es usada por usuarios finales, es decir, una solución que está lista para ser instalada y utilizada, sin conocer de fondo el detalle de los algoritmos o el código que lo soporta.

3.2.3.1. Software propietario

La herramienta encontrada en la investigación es NESTGate [55], la cual fue desarrollada por los Laboratorios Fujitsu. Debido a la naturaleza de este software, es comprensible que los autores no entreguen mayor detalle respecto a cómo esta herramienta realmente funciona para entornos de *Big Data*, sin embargo, desde una perspectiva de investigación, sería complejo determinar cómo el software realmente está apoyando a estos nuevos y cambiantes entornos.

NESTGate es definida por los autores a partir de dos concepciones. La primera radica en la aplicación de una "*pseudo-anonimización*" que se refiere a ejecutar operaciones de sustitución. No obstante, los Laboratorios Fujitsu son conscientes de los riesgos inherentes a esta aproximación poco compleja, por lo cual aparece la segunda funcionalidad, a través de la cual se aplican operaciones de generalización y supresión para cumplir el principio *k-anonymity*. Los autores reconocen las limitaciones de memoria en que se podrían incurrir en un contexto de grandes volúmenes de datos, así que aseguran haber desarrollado un novedoso algoritmo que puede tener en cuenta bloques en vez de todo el conjunto al tiempo. Adicionalmente, la herramienta incluye un ciframiento "*homomórfico"* que busca tener mayor control sobre los datos. Sin embargo, esto va en detrimento del concepto de *PPDM*, al inhibir todo proceso analítico.

3.2.4. Algoritmos

Los ***Algoritmos*** de anonimización se asocian a los pasos estructurados que deben llevarse a cabo para asegurar la privacidad y seguridad de la información en un contexto de *Big Data*, donde las operaciones son aplicadas para cumplir los principios de anonimización. En general, éstos están directamente ligados a metodologías basadas en heurísticas, las cuales buscan encontrar un balance entre la complejidad computacional debido a las características de los datos, y el aseguramiento de la privacidad de la información. Bajo este escenario, se identificaron dos enfoques principales que presentan los algoritmos. El primero de ellos se relaciona con el

Cómputo centralizado, donde los datos están localizados en un único almacenamiento y los cálculos son ajustados para ser llevados a cabo teniendo en cuenta el volumen de información que puede ser mantenido en memoria para efectuar las transformaciones de anonimización. Por su parte, los algoritmos de ***Cómputo distribuido*** son diseñados para escenarios de datos distribuidos, usualmente en sistemas de archivos como *HDFS* (*Hadoop Distributed File System*, por sus siglas en inglés), donde los cálculos son modificados para llevarse a cabo de forma distribuida, por ejemplo, utilizando tareas *Map-Reduce*.

3.2.4.1. Cómputo centralizado

LeFevre *et al.* [8] propusieron dos algoritmos voraces bajo el esquema de *Top-Down Specialization*, que se basan en el algoritmo *Mondrian* [7] propuesto unos años antes, el cual consiste en dividir el problema para hacer generalizaciones en pequeños sub-problemas (regiones multidimensionales). El algoritmo inicial y los aquí propuestos utilizan técnicas de recodificación local, lo cual significa que instancias del mismo elemento (valor del atributo), en diferentes clases de equivalencia, pueden ser generalizados en diferentes niveles. Partiendo de estos fundamentos, los autores proponen los algoritmos *Rothko-t* (*Rothko-Tree*) y *Rothko-S* (*Rothko-Sampling*). Ambos algoritmos son una variación de *Mondrian* aplicado a conjuntos de grandes volúmenes de datos en una sola máquina, donde se tiene en cuenta el escenario en que los datos tienen un peso tal que podrían desbordar la memoria de la máquina en la cual se esté haciendo el procesamiento, debido a los espacios recursivos que se van creando y en los cuales se sacan copias de cada partición de los datos originales.

Rothko-t se basa en una estructura de datos de un árbol para dividir el espacio de dominio recursivamente, alojando ciertos datos de resumen sobre el conjunto completo de datos. El objetivo es no cargar todos los datos en memoria, sino revisar algunas medidas sobre un subconjunto de ellos, a fin de determinar sub-divisiones que no provoquen un desbordamiento de memoria. A pesar de ser una importante consideración para un volumen alto de datos, no es claro cómo los autores logran determinar los estadísticos que almacenan en cada nodo del árbol, sin poner en riesgo su utilidad, al sólo cargar pequeñas particiones y sobre ellas tomar decisiones para posteriormente aplicar el *Mondrian* tradicional.

Rothko-S, por su parte, busca mejorar el rendimiento propuesto en el anterior algoritmo al manejar muestras de datos. De esta manera, se selecciona una muestra del conjunto original, de acuerdo con la cual se decide cuál es el atributo por el que se debe iniciar el particionamiento. Así, sólo se carga en memoria una partición de datos a la vez, de forma secuencial. Los datos de la muestra se seleccionan aleatoriamente para conseguir rapidez en el proceso. Desafortunadamente, no tener en cuenta las propiedades subyacentes de los datos al hacer un muestreo aleatorio, no permite generar confianza en los resultados globales obtenidos, yendo en detrimento de la utilidad de los datos para subsiguientes aplicaciones de analítica.

La propuesta hecha por los autores es interesante desde el punto de vista de analizar la viabilidad computacional del algoritmo *Mondrian* debido al consumo de memoria asociado. No obstante, se requiere mayor profundidad en el planteamiento, puesto que las aproximaciones po-

drían sacrificar la utilidad de los datos debido a la ausencia de entendimiento de las características y propiedades originales de los datos, las cuales podrían perderse al tomar en cuenta sólo algunos datos para generar las particiones que son enviadas al algoritmo *Mondrian.*

3.2.4.2. Cómputo distribuido

Zhang *et al.* presentan en [56] una versión de conferencia del trabajo realizado en [57]. El primero propone una aproximación *Top-Down Specialization* (*TDS*), mientras que el segundo presenta una aproximación *Bottom-Up Generalization* (*BUG*) que integra ambos enfoques en [56]. Los autores se centraron en una anonimización *TDS* denominada *sub-tree* para cumplir el principio de *k-anonymity*. Su principal contribución es proponer el *TDS* bajo la arquitectura de trabajos *Map-Reduce,* por lo cual acuñan la sigla *MRTDS* para referirse a *Map-Reduce Top-Down Specialization*. En este algoritmo es posible escalar dentro de ambientes distribuidos de datos. En la primera fase, es decir el *Map*, el objetivo es utilizar computación paralela para anonimizar por primera vez. Posteriormente, se agregan los datos mediante la fase de *Reduce*. Por lo anterior, *MRTDS* anonimiza particiones de datos para generar niveles de anonimización intermedios, pero el cumplimiento del principio *k-anonymity*, sólo se consigue hasta la segunda fase. En cada paso se calculan métricas de utilidad, las cuales se van actualizando en la medida en que el algoritmo genera las especializaciones más adecuadas para maximizar tal utilidad. Finaliza cuando ya no es posible generar transformaciones válidas sobre los datos.

De acuerdo con lo anterior, en [57], los autores presentan una aproximación híbrida entre *TDS* y *BUG*, de igual forma bajo la estructura de trabajos *Map-Reduce* en ambientes altamente distribuidos. Dado un conjunto de datos de entrada *D*, y el principio *k-anonymity*, la propuesta híbrida selecciona el mejor resultado entre la aplicación de *MRTDS* y *MRBUG* (también por introducir el esquema *Map-Reduce*), de acuerdo con una función $f(D, k)$. El objetivo es que la función *f* considere varias métricas sobre *D*, tales como: entropía, ganancia de información por pérdida de privacidad, pérdida de información por ganancia de privacidad, entre otras. Dependiendo del valor *k*, el algoritmo híbrido decide cuál es la mejor aproximación.

Por otra parte, Zhang *et al.* [58] proponen una anonimización basada en recodificación local, utilizando tareas *Map-Reduce*, para escenarios de proximidad de los datos. Los autores establecen que las aproximaciones existentes para preservar la privacidad de los datos fallan cuando se enfrentan a grandes volúmenes de datos por temas de ineficiencia o debido a baja escalabilidad de la solución. Su algoritmo se denomina *Single-Objective Proximity-Aware Clustering* (*SPAC*) y la propuesta se basa en el modelo de proximidad semántica sobre valores categóricos. Aplican un modelo de *clustering* en dos fases: la primera consiste en un algoritmo inspirado en *k-means* llamado *t-ancestors* y el segundo algoritmo hace la validación de la proximidad. La primera fase divide un conjunto de datos originales en particiones que contienen registros con datos cuasi-identificadores similares. En la segunda fase, los datos de las particiones son recodificados (generalizados) localmente con el valor menos común, valiéndose de un árbol jerárquico para los atributos categóricos, mientras que en los numéricos se generan rangos. A continuación se aplica el algoritmo de *clustering* aglomerativo, el cual crea grupos iterativamente mediante el uso de una cola de prioridad, que almacena la distancia entre los *clusters* y evita que terminen siendo mezclados durante el proceso (dada su corta distancia). El criterio de parada es el tamaño del grupo, que al alcanzar el valor *k*, finaliza la ejecución.

Por su parte, Sowmya *et al.* [59] argumentan que debido a la era creciente del *Big Data*, es necesario redefinir las aproximaciones de anonimización para preservar la privacidad, de acuerdo con *PPDM*, pero ahora en entornos de grandes volúmenes de datos distribuidos. Por ello, los autores proponen la redefinición del principio *k-anonymity*, a través del uso de ambientes de computación distribuida con tareas *Map-Reduce* en un ecosistema Hadoop. El algoritmo de paralelización de *k-anonymity*, denominado *Pk-a*, aplica recodificación local para minimizar la distorsión de los datos originales, sin caer en sobre-generalizaciones innecesarias. El algoritmo funciona bajo el esquema *Top-Down*, donde es dividido recursivamente el conjunto de datos en dos particiones que son posteriormente paralelizadas a través de *Map-Reduce*. Adicionalmente, el algoritmo se vale de una estructura de árbol, la cual es construida recursivamente, donde cada nodo es un conjunto de datos resultante de una partición y, posteriormente, cada nivel del árbol es dirigido a cada *mapper* de la tarea. Luego, en la función de reducción, los conjuntos de datos resultantes que cumplen el principio de *k-anonymity* son agrupados para generar el conjunto de datos anonimizados.

De otra parte, Zhang *et al.* [60] también inspirados en el algoritmo multidimensional *Mondrian*, generan una nueva propuesta escalable para grandes volúmenes de datos, basado en tareas *Map-Reduce*. Los autores reconocen que, aunque las aproximaciones previas para no desbordar la memoria al ejecutar el algoritmo recursivo *Mondrian* son escalables en términos de cantidad de datos, requieren espacios prolongados de tiempo debido a que son en esencia secuenciales, además de generar importantes latencias debido al acceso a disco constante, para permitir que un subconjunto de los datos pueda mantenerse en memoria. Los autores proponen entonces *MRMondrian* (debido a las tareas *Map-Reduce* sobre *Mondrian*). El objetivo es particionar los datos, pero ya no de forma recursiva, sino de manera iterativa, en subconjuntos más pequeños, hasta que todos quepan en memoria en cada nodo; posteriormente, es posible ejecutar el algoritmo tradicional en tales nodos, con una cantidad menor de datos. Para realizar la primera parte, se propone utilizar una única tarea *Map-Reduce* que genere las particiones, a través de una estructura de árbol indexado, donde se insertan los datos estadísticos de las nuevas particiones, mediante un recorrido por amplitud. En cada nodo del árbol se almacenan los datos básicos de cada partición, como los estadísticos que caracterizan al subconjunto de datos. Las ventajas de realizarlo de forma iterativa son varias: evitar la sobrecarga por la inicialización y programación de múltiples tareas *Map-Reduce*, calcular una cantidad constante de nodos para la ejecución de las operaciones *Map*, y reordenar los registros de una partición para que queden localizados en un mismo nodo, sin incurrir en altos movimientos de datos distribuidos.

Finalmente, Zhang *et al.* [61] proponen un acercamiento escalable de codificación local llamado *Locality Sensitive Hashing based local-recoding* (*LSH*) el cual, a través de métricas de distancia tanto para datos categóricos (distancia de *Jaccard*) como numéricos (distancia *Euclidiana*), determina la similaridad entre registros. En primera instancia se divide el conjunto original en diferentes particiones. A continuación, se aplica un algoritmo de *clustering* de *k* miembros formando *clusters* bajo una aproximación aglomerativa, en cada una de las particiones, de forma paralela, a través de tareas *Map-Reduce*. En este punto se utiliza una cola de prioridad para determinar qué *clusters* deben fusionarse en orden de la menor distancia entre los mismos. De esta forma, se anonimiza cada cluster de *k* miembros. Los autores hacen énfasis en que este método permite obtener una menor pérdida de información en la medida que se logran construir pequeños *clusters* y posteriormente es posible aplicar operaciones de generalización sobre cada uno de ellos.

3.3. Conclusiones

De acuerdo con la investigación realizada se puede concluir que en la literatura es usual encontrar el término *Big Data*, aplicado al contexto de anonimización de datos. Sin embargo, las aproximaciones presentadas, en su mayoría no están directamente relacionadas con los paradigmas a los cuales se refieren pues no exponen pruebas, validaciones o escenarios donde se evidencie la aplicación de las *Vs* que caracterizan un contexto de *Big Data*.

De igual manera, se suele entender la anonimización como un proceso aislado, centrado únicamente en la transformación de los datos para asegurar la privacidad y confidencialidad para el manejo responsable de la información. Es importante destacar que, a pesar de que en los trabajos relacionados se hace referencia a términos como *PPDM*, en realidad no se demuestra cómo la propuesta específica (algoritmo, arquitectura o herramienta) en realidad sigue posibilitando la aplicación de un modelo de analítica, tras la anonimización de los datos. Es necesario articular mejor los esfuerzos para proponer artefactos de anonimización, teniendo en cuenta todo el proceso de análisis de información en contextos comunes y en escenarios de *Big Data*. En este sentido, es necesario encontrar la conexión entre los procesos de anonimización y los procesos de minería de datos e inteligencia de negocios para generar enfoques holísticos que busquen un garantizar la privacidad y utilidad, incluso en presencia de grandes volúmenes de datos.

En este mismo sentido, las propuestas actuales, por ejemplo, para la operación de generalización no son contundentes frente a cómo manejar los datos para no alterar su tipo y por tanto imposibilitan la construcción de modelos analíticos que saquen mejor provecho de los datos numéricos. Las aproximaciones presentadas se limitan a presentar un rango o etiqueta sin dilucidar que ello afecta la utilidad de los datos para los procesos analíticos subsiguientes.

De acuerdo con este panorama, el presente trabajo busca profundizar en el propósito de preservación de la utilidad, encontrando el mejor balance, y teniendo en cuenta los aspectos no funcionales que inciden en un contexto de grandes volúmenes de datos.

4. ANONYLITICS: UNA PROPUESTA DE ANONIMIZACIÓN ORIENTADA A LA ANALÍTICA DE DATOS

En esta sección se presenta una propuesta de anonimización pensada desde la analítica de datos. Si bien es necesario proteger la información, este proceso debe estar enmarcado en un contexto más amplio que permita evaluar la utilidad de los datos para los procesos de análisis que buscan hallar relaciones y patrones que conlleven a una toma de decisiones informada y acertada.

4.1. Visión global

Se propone la construcción de ***Anonylitics***, un sistema de anonimización de datos estructurados que se centra en la fase de preparación de datos dentro de una metodología de analítica. En tal etapa es necesario asegurar que la información esté disponible para hacer minería, pero salvaguardando los intereses de la compañía sin poner en riesgo su negocio. En este contexto, es necesario entender que, debido al proceso de anonimización, habrá inevitablemente una pérdida de información asociada, la cual se pretende reducir. Desde la perspectiva técnica, se tiene un problema donde se debe generar un balance entre utilidad de la información y aseguramiento de la privacidad; esto implica que, a mayor privacidad, menor utilidad de los datos, y a mayor utilidad de los datos, menor privacidad de los mismos.

Con el objetivo de entender las necesidades de la Alianza CAOBA y sus empresas ancla, se llevaron a cabo entrevistas presenciales y virtuales que buscaron indagar sobre los problemas actuales que se han presentado para iniciar proyectos de analítica de datos. Estas necesidades fueron estructuradas en forma de requerimientos del sistema de anonimización. El documento SRS, producto de este proceso, se encuentra disponible en el documento anexo *Especificación de Requerimientos de Software*. Después de analizar tales requerimientos, fue posible generar agrupaciones de acuerdo con su relación y afinidad, formando los casos de uso del sistema. Se identificaron en total ocho casos de uso y dos actores que interactúan con *Anonylitics*: el usuario final y el sistema de gestión de datos, como se evidencia en la Figura 8. El primero es el dueño de los datos y es representado por una persona que hace parte de la empresa que desea entregar sus datos a un tercero como CAOBA; debe conocer sobre las características de esos datos y los posibles riesgos de vulnerabilidad, pues es quien interactuará con el sistema para definir las operaciones a aplicar sobre la información que desea protegerse. El segundo actor es el Sistema de Gestión de Datos (SGD) que, dependiendo de su índole, puede utilizar las funcionalidades de anonimización. Estos dos actores interactúan con *Anonylitics*, a través de los siguientes casos de uso:

1. ***Administrar proyecto de anonimización:***
   Abarca aquellos requerimientos orientados a permitir a un usuario final definir, guardar y reutilizar las parametrizaciones establecidas sobre un conjunto de datos a anonimizar. La parametrización hace referencia a los datos asociados a las tablas seleccionadas y la configuración de las posibles operaciones sobre cada atributo.
2. ***Administrar acceso a fuentes de datos****:*
   Comprende los requerimientos que hacen referencia a las funcionalidades de establecimiento, conexión y acceso a diferentes fuentes de datos, como archivos planos en formato tabular, bases de datos relacionales y sistemas distribuidos de archivos como HDFS.

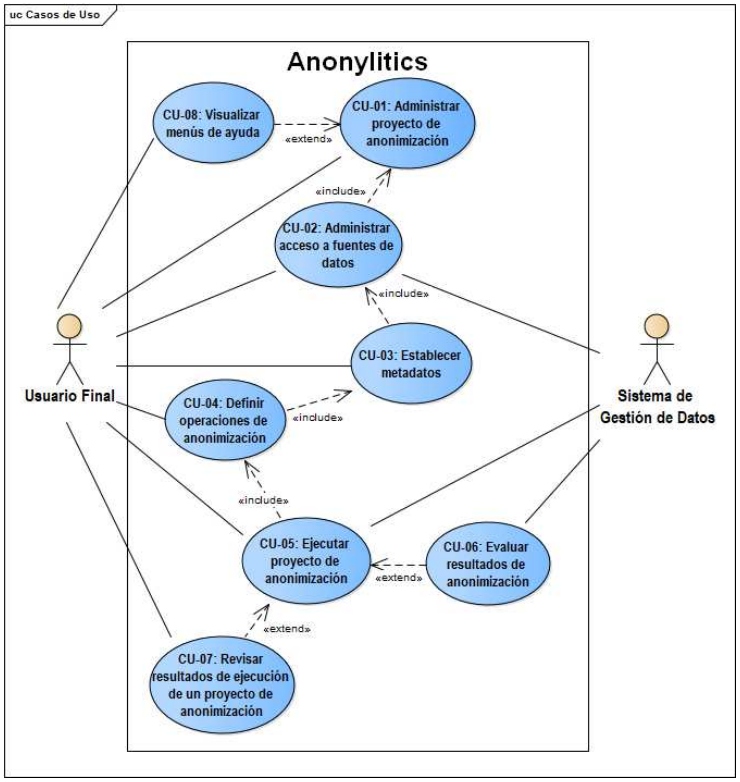


Figura 8. Casos de uso de Anonylitics.

3. ***Establecer metadatos:***
   Incluye los requerimientos referentes a permitirle al usuario final determinar los tipos de atributo: *identificador*, *cuasi-identificador*, *sensible* o *común*, dentro de su conjunto de datos, así como el tipo de anonimización a aplicar (completa o ligera). Se determinó que se deben proveer dos tipos: *i). Ligera*, cuyo propósito es aplicar operaciones de *supresión*, *sustitución* y *ciframiento*, que son las aproximaciones más utilizadas en la actualidad por las compañías, y donde no se realizan validaciones de utilidad o privacidad pues se orienta a ejercicios sencillos donde la criticidad de los datos es baja; y *ii). Completa*, donde se pretende generar un valor agregado al evaluar la utilidad y privacidad de los datos. En este escenario, se proveen algoritmos más robustos que pretenden generar un adecuado balance entre aseguramiento de los datos y preservación de su utilidad para hacer analítica, a través de operaciones de *generalización*, *sustitución* y *supresión*.
4. ***Definir operaciones de anonimización:***
   Integra aquellos requerimientos orientados a permitirle al usuario final determinar la operación que desea hacer sobre los datos y especificar los parámetros necesarios para *suprimir*, *generalizar*, *cifrar* o *sustituir*. Un ejemplo de un parámetro necesario para aplicar una operación de generalización sobre un tipo de dato categórico es usar una jerarquía donde se especifique una relación de ordinalidad sobre valores del atributo.
5. ***Ejecutar proyecto de anonimización:***
   Abarca aquellos requerimientos que hacen referencia a las funcionalidades relacionadas con la ejecución de los dos tipos de anonimización (ligera o completa), la generación de

informes de resumen y notificaciones acerca del estado de la ejecución. En este punto se incluyen los algoritmos necesarios para anonimizar los datos.

6. ***Evaluar resultados de anonimización:***
   Comprende aquellos requerimientos que hacen referencia a las funcionalidades de evaluación del cumplimiento de los principios de anonimización, lo cual sólo cumple para el caso de una anonimización completa. De igual manera, se incluyen los requerimientos de análisis de utilidad de los datos respecto al análisis de preservación de la distribución original de los datos, de forma univariada.
7. ***Revisar resultados de ejecución de un proyecto de anonimización:***
   Integra aquellos requerimientos relacionados con funcionalidades de post-ejecución de un proyecto de anonimización, del lado del usuario final, tales como: generar archivos planos con los datos resultantes después de la anonimización, o almacenarlos en su fuente original, así como visualizar un reporte con el resumen de la ejecución en términos de la evaluación de privacidad y utilidad de los datos.
8. ***Visualizar menús de ayuda:***
   Comprende aquellos requerimientos relacionados con la funcionalidad de disponer de un conjunto de menús de ayuda que permitan explicar las funcionalidades de Anonylitics y conceptos relacionados con el ámbito de la anonimización, usados en el sistema: operaciones, principios, tipos de atributos y tipos de anonimización.

Se propone construir un sistema de anonimización que permita: conectarse a diferentes fuentes de datos estructuradas; construir un proyecto de anonimización que sea parametrizable y reutilizable, y que por tanto sea almacenado incluyendo todos los metadatos asociados; y ejecutar el proyecto, de modo que al final se obtenga un conjunto de datos anonimizados y un reporte que indique la evaluación de privacidad y utilidad de éstos.

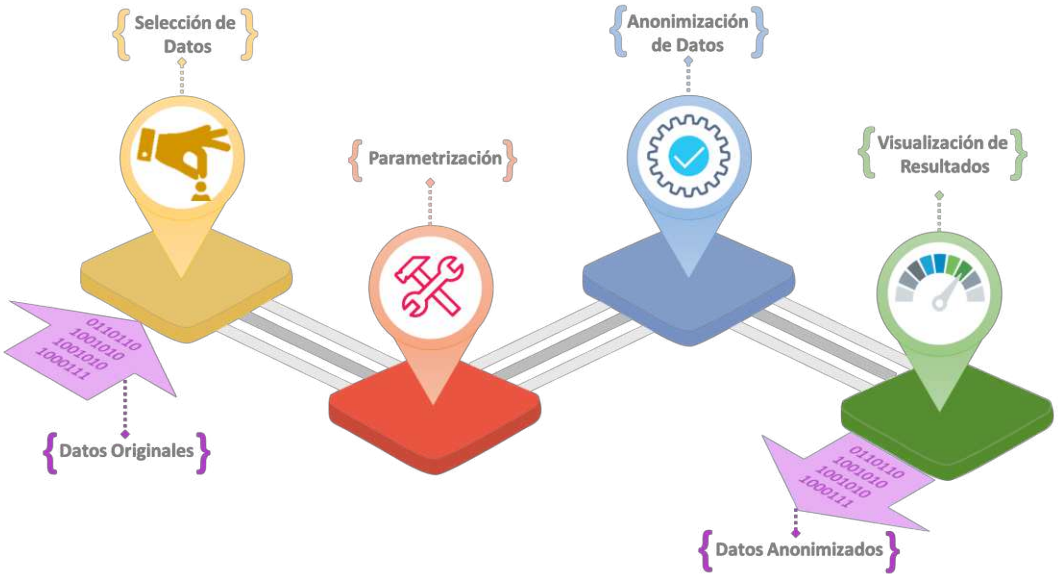


Figura 9. Flujo de anonimización en Anonylitics.

En la Figura 9 se evidencia la visión global del flujo requerido para llevar a cabo la anonimización de un conjunto de datos. En primera instancia se hace una ***Selección de Datos*** que permite definir con qué conjunto se va a trabajar, y desde dónde se va a obtener para empezar el proceso. A continuación, se efectúa la ***Parametrización*** correspondiente a cada atributo (columna) que sea vulnerable y por tanto deba ser anonimizado; esto implica determinar todos los detalles necesarios para saber qué tipo de transformación se desea aplicar, con el fin de entregar al sistema la información necesaria para la aplicación de las transformaciones. Posteriormente, se desencadena el proceso de ***Anonimización de Datos***, con lo cual se obtiene un conjunto de datos anonimizados, sobre los cuales se desprende un análisis de utilidad y privacidad que se expone al usuario final, permitiendo la ***Visualización de Resultados***.

A continuación se detalla el proceso de ***Anonimización de Datos***, evidenciando la propuesta orientada a la analítica de datos para una *anonimización completa* y los mecanismos de evaluación asociados.

4.2.Anonimización de datos

La privacidad y la utilidad son dos aspectos que van en diferentes direcciones puesto que, al incrementar la privacidad de los datos, se deteriora su utilidad para posteriores análisis. La finalidad perseguida es entonces plantear un mecanismo de anonimización que promueva el equilibrio entre ambas necesidades, a la luz de los requerimientos definidos por el cliente. Es fundamental entender que ambas partes de la balanza deben analizarse en conjunto para lograr el objetivo de anonimización. Es decir, no es posible plantear un algoritmo que anonimice y después entrar a revisar cómo garantizar que los datos sigan siendo útiles. Por ello, ese algoritmo debe estar encaminado a responder a ambas necesidades.

La necesidad de **privacidad**, para Anonylitics, se plantea desde el cumplimiento del principio *k-anonymity* pues éste permite plantear un marco a través del cual se puede evaluar en qué forma se asegura que los datos estén siendo protegidos para evitar posibles ataques de Singularización (ver sección *2.3. Riesgos de divulgación y mecanismos de seguridad*). En este sentido, se entenderá que los datos son privados si no existe registro alguno que tenga una combinación única de valores de los cuasi-identificadores dentro de todo el conjunto de datos. Esto inhibirá cualquier posibilidad de ataque basado en conocimiento previo o por asociación con otras bases de datos públicas o privadas de las cuales se disponga para obtener mayor información de las entidades a las cuales hacen referencia los datos.

La necesidad de **utilidad**, para Anonylitics, se fundamenta en uno de los requerimientos del cliente donde se solicita que pueda ser preservada la función de probabilidad de distribución de cada uno de los atributos numéricos vulnerables, es decir, los cuasi-identificadores. En el caso de los identificadores, no es útil perseguir este mismo objetivo dado que cada valor es diferente y no es utilizado para generar modelos analíticos que buscan encontrar patrones que por simple inspección o consulta, no podrían vislumbrarse.

De acuerdo con lo anterior, en las siguientes secciones se presentan las propuestas del presente trabajo, encaminadas a anonimizar atributos cuasi-identificadores numéricos, bajo el esquema de *anonimización completa*, es decir, donde se evalúa la utilidad y privacidad de los datos.

4.2.1. Algoritmo de anonimización

De acuerdo con la investigación realizada para la construcción del Estado del Arte, se encontró que el algoritmo ***Mondrian*** permite cumplir el principio *k-anonymity* y que ha tenido unos buenos resultados en comparación con otros algoritmos en términos de pérdida de información, consumo de memoria y tiempo de procesamiento. Además, permite trabajar con la operación no perturbativa de *Generalización* para no alterar sustancialmente los valores originales. El algoritmo logra trabajar de forma eficiente con datos numéricos debido a su forma de particionamiento que no implica la generación previa de jerarquías predefinidas. Por tal razón, se definió que este algoritmo sería el punto de partida para cumplir con la necesidad de privacidad de los datos. No obstante, fue necesario analizar y proponer variaciones y modificaciones que permitieran también cumplir con la necesidad de utilidad de los datos. De acuerdo con Ayala-Rivera *et al.*[6] este algoritmo se ve fuertemente influenciado por la distribución de los atributos y el criterio de corte utilizado, puesto que, para distribuciones con sesgo, el algoritmo genera mayor pérdida de información, y aunque se cumpla con el principio de anonimización, asegurando la privacidad de los datos, podría estarse comprometiendo su utilidad.

Con el objetivo de alinear las necesidades para generar un balance adecuado entre la privacidad y la utilidad de los atributos numéricos, se diseñó el algoritmo ***Distribution-based Mondrian***, en el cual se propone modificar el criterio de corte para particionar recursivamente el espacio multidimensional de los cuasi-identificadores y el criterio de parada para determinar cuándo no es posible generar nuevas sub-regiones, permitiendo seguir cumpliendo el principio de *k-anonymity*. En la Figura 10 se muestra una representación gráfica del algoritmo propuesto, el cual está basado en el planteamiento original de LeFevre *et al.* [7].

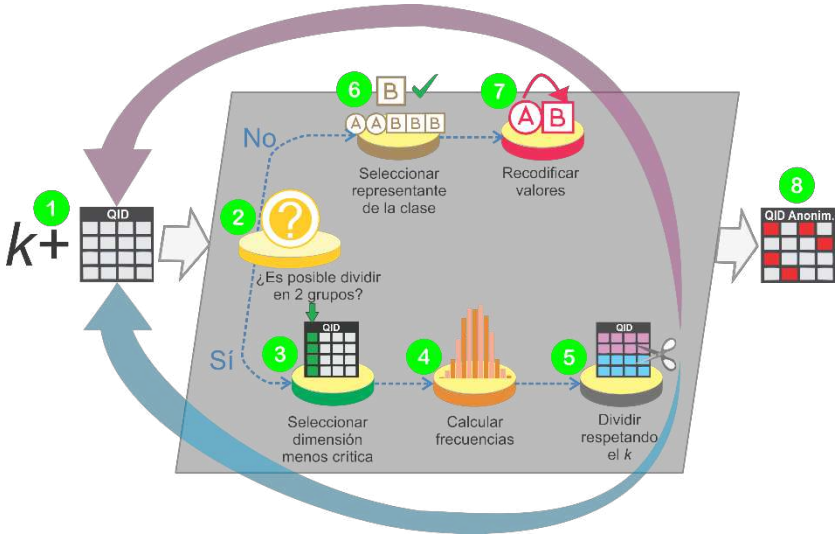


Figura 10. Algoritmo "Distribution-based Mondrian".

La entrada del algoritmo es el valor *k* y el conjunto de datos constituido por los valores cuasi-identificadores numéricos (1), reconocidos previamente. En caso de poder dividir en dos grupos respetando el *k* (2), en primera instancia se selecciona la dimensión menos crítica (3), es

decir, el atributo cuasi-identificador que podría generar menor pérdida de información debido a su amplio conjunto de valores asociados. El cálculo que se realiza está orientado a determinar el atributo que posea el mayor rango, después de haber normalizado todos sus valores. A continuación, de acuerdo con la dimensión seleccionada, se calculan las frecuencias de los valores de ese cuasi-identificador (4), para generar posteriormente el corte del conjunto de datos en dos nuevas particiones que no se solapan (5), basado en la distribución de tal dimensión. Cada región es evaluada nuevamente de forma recursiva hasta que no sea posible generar nuevas particiones puesto que implicaría violar el principio *k-anonymity*. En este caso, cuando no se puede dividir más, se selecciona el representante de la clase (6), es decir el valor anonimizado para la partición resultante. Con el propósito de mantener la distribución de los datos, se selecciona la moda para cada atributo, dentro de cada grupo; si no existe una única moda, se elige la mediana. Con este representante, el último paso es entonces efectuar la recodificación de cada atributo cuasi-identificador dentro de cada clase de equivalencia (7), es decir, dentro de cada conjunto de (mínimo *k*) registros agrupados tras la aplicación del algoritmo. De esta forma, la salida del algoritmo es el conjunto de datos constituido por los valores cuasi-identificadores numéricos anonimizados (8).

La generación de dos nuevas particiones a partir del criterio de corte, está basada en la función de distribución de probabilidad de la dimensión seleccionada. Dado que el objetivo es poder preservar la distribución, es fundamental generar un corte que no altere los datos al dividirlos en dos conjuntos que impliquen una alta pérdida de información. Por lo anterior, se propone el corte denominado *Distribution-based Split* donde se generan dos grupos que procuran respetar la probabilidad de aparición de cada elemento del atributo seleccionado, disminuyendo la pérdida de utilidad asociada. En la Figura 11 se puede apreciar el pseudo-código del criterio de corte propuesto, que corresponde al número (5) de la Figura 10.

Distribution_based_split $(freq, k)$
inicio
  $dis[]$ ← valores distintos en $freq$
  $dis[]$ ← ordenar ascendentemente dis
  $mín$ ← mínimo valor en dis
  $máx$ ← máximo valor en dis
  tam ← tamaño de dis
  desde $(p = 0$ hasta $tam)$
    $conteo\_izq$ ← $\sum_{i=0}^{p}$ frecuencia del valor dis_i en $freq$
    $conteo\_der$ ← $\sum_{j=p+1}^{tam}$ frecuencia del valor dis_j en $freq$
    si $(conteo\_izq \geq k \ \& \ conteo\_der \geq k)$
      $rango\_izq$ ← $[min\ , dis_p]$
      $rango\_der$ ← $[dis_{p+1}\ , máx]$
      retornar $rango\_izq, rango\_der$
    sino
      $p = p + 1$
    fin_si
  fin_desde
fin

Figura 11. Pseudo-código del Criterio de corte "Distribution-based Split".

Para ilustrar el funcionamiento del criterio de corte, en la Figura 12 se presenta un ejemplo para un solo cuasi-identificador, donde se busca cumplir el principio *k-anonymity* con *k=2*. La forma

de definir el corte está ligado a la frecuencia de aparición de cada valor del cuasi-identificador. Si el menor valor de la partición cuenta con una frecuencia igual o mayor al *k* y los demás valores también cuentan con una frecuencia superior, entonces el primer grupo de valores constituirían la partición izquierda, mientras que el segundo grupo, de los restantes, conformarían la partición derecha. En este sentido, aquellos valores que se repiten dentro del conjunto total, siempre estarán organizados para pertenecer a una misma partición. Para aquellos valores menos frecuentes, que por sí mismos no logran cumplir con el principio *k-anonymity*, como por ejemplo el número 1 y el 4 de la Figura 12, se agrupan a los valores (anteriores o posteriores) más cercanos. Al final de cada hoja del árbol que se forma, queda una clase de equivalencia dentro de la cual se debe buscar un representante que contribuya a preservar la distribución original del atributo. El representante es escogido como la moda, que está sustentado por los cortes realizados, con lo cual se busca mantener aproximadamente la probabilidad de aparición de cada valor del atributo cuasi-identificador. En caso de no existir una única moda, se selecciona como representante la mediana de la clase de equivalencia. Es importante aclarar que éste es solo un ejemplo que pretende demostrar el funcionamiento de la propuesta hecha, pero que adquiere mayor valor cuando se está trabajando con más de un cuasi-identificador, puesto que en ese escenario, no siempre las particiones del lado izquierdo estarán listas para ser recodificadas y no resulta obvia la preservación de la distribución, al estar dividiendo por diferentes atributos, de acuerdo con la selección del cuasi-identificador menos crítico.

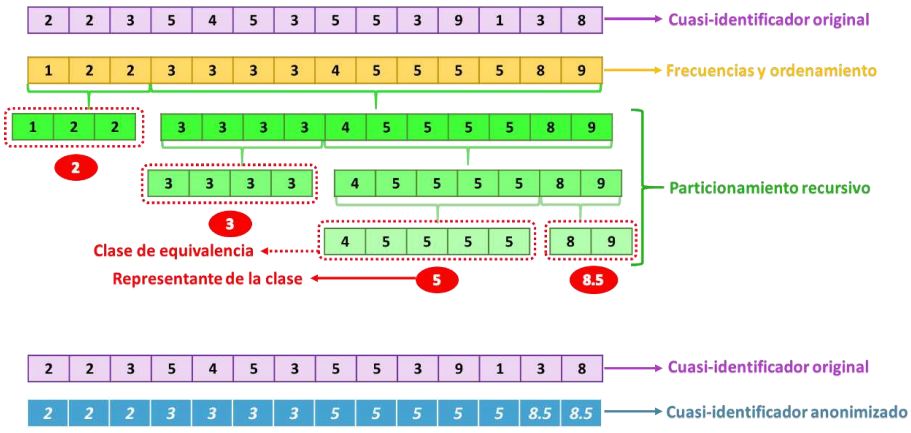


Figura 12. Ejemplo de funcionamiento del criterio de corte "Distribution-based Split" en un cuasi-identificador, para cumplir con un k=2.

Además de realizar el proceso de anonimización, la propuesta de este proyecto es también entregar al usuario los resultados de la evaluación de los criterios de utilidad y privacidad. A continuación se presentan los mecanismos planteados.

4.2.2. Mecanismo de evaluación de utilidad

Debido a la necesidad de evaluación de utilidad, se plantea trabajar con una prueba estadística que permita determinar con un nivel de confianza, si cada atributo cuasi-identificador anonimizado mantiene la distribución original de los datos. Para ello, a continuación se dan las premisas formales para definir la prueba de significancia seleccionada.

De acuerdo con Alvarado y Obaji [62], todo estadístico es una variable aleatoria y por tanto tiene una función de probabilidad asociada que se conoce como distribución muestral. La probabilidad de ocurrencia de un valor dentro de un conjunto de datos se evalúa por medio de números reales denominados pesos o probabilidades que van de 0 a 1 [63]. Cada valor tiene asociada una probabilidad de ocurrencia, así que la suma de todas las probabilidades es 1.

Formalmente, en caso de variables escalares discretas, donde la variable X puede tomar los valores $x_1, x_2, \dots, x_n$, la distribución de probabilidad de X se define como el conjunto de pares (x_i, p_i) que a cada valor x_i le asigna una probabilidad donde $p_i = P(X = x_i)$, tal que la suma de todas las probabilidades es igual a la unidad. En caso de variables escalares continuas, hay infinitos posibles valores de la variable, por lo cual no es factible deducir la probabilidad de un valor puntual de la variable. No obstante, sí es posible calcular la probabilidad acumulada hasta un determinado valor (lo cual se denomina función de distribución) y cómo cambia esa probabilidad acumulada en cada punto (lo cual se denomina densidad de probabilidad). Formalmente, $f(x)$ es la función de densidad no negativa definida sobre la recta real, asociada a la variable X, la cual cumple que, para cualquier intervalo estudiado, se puede verificar que para todo valor perteneciente a X, su probabilidad está dada por $P(X) = \int f(x)dx$.

Dado que el objetivo del proyecto es poder comparar la distribución original de cada atributo cuasi-identificador, en relación con la distribución del atributo anonimizado, se plantea poder comparar las distribuciones a través de la *Prueba de dos muestras de Kolmogorov Smirnov* [64]. Esta prueba estadística evalúa si dos muestras de datos provienen de la misma distribución, sin importar si la distribución original corresponde a una distribución bien conocida (normal, gamma, etc.), por lo cual se considera como una prueba no paramétrica. La prueba compara dos funciones de distribución empíricas, a través de la fórmula (1).

$$D = | E_1(i) - E_2(i) | \quad (1)$$

Donde, E_1 y E_2 son las funciones de distribución empíricas de las dos muestras. A continuación se describen formalmente las hipótesis (2) y (3) en las cuales se basa esta prueba estadística.

$$H_0: Las\ dos\ muestras\ provienen\ de\ una\ distribución\ en\ común \quad (2)$$

$$H_a: Las\ dos\ muestras\ NO\ provienen\ de\ una\ distribución\ en\ común \quad (3)$$

Donde H_0 es la hipótesis nula y H_a es la hipótesis alternativa. El estadístico de prueba es D y el α es el nivel de significancia con el que se validan las hipótesis. Así, la hipótesis nula es rechazada si el estadístico D es mayor que el valor crítico obtenido de aplicar una tabla designada para esta prueba.

Para el caso de Anonylitics, la definición de las hipótesis (4) y (5) se ajusta de la siguiente manera:

$$H_0: El\ atributo\ anonimizado\ mantiene\ la\ distribución\ del\ atributo\ original \quad (4)$$

$$H_a: El\ atributo\ anonimizado\ NO\ mantiene\ la\ distribución\ del\ atributo\ original \quad (5)$$

A través de la aplicación del algoritmo *Distribution-based Mondrian* se busca concluir que no existe evidencia estadística para rechazar la hipótesis nula y por tanto se pueda asegurar con un nivel de confianza, que el atributo anonimizado mantiene la distribución del atributo original.

4.2.3. Mecanismo de evaluación de privacidad

El algoritmo *Distribution-based Mondrian* posibilita de forma automática el cumplimiento del principio *k-anonymity* al generar particiones finales que contengan al menos *k* registros. No obstante, es necesario generar un análisis para determinar cuántos registros podrían re-identificarse en el conjunto de datos original, versus la cantidad de registros que pueden re-identificarse después de la anonimización, el cual debe ser cero. De este modo, la evaluación de la privacidad se genera a partir del conteo de registros que contienen la misma combinación en sus cuasi-identificadores. Si existe una misma combinación de valores en menos de *k* registros, se concluiría que no se cumple el principio de privacidad. Esto se valida a través de un cálculo que determina cuántos registros pueden re-identificarse, con lo cual un atacante podría cruzar con otras bases de datos para inferir información adicional, como la representada por los atributos sensibles presentes en el conjunto de datos. Así, después de anonimizar, el resultado del cálculo debe arrojar que existen cero registros, concluyendo que el conjunto de datos es privado al no poder identificar inequívocamente menos de *k* registros.

Aunque el usuario final es quien determina el valor *k* con el cual se valida el principio de anonimización, es posible que en caso de exigir un valor alto (por ejemplo, por encima de 20), no logre preservarse la utilidad de los datos puesto que, entre más grande es el *k*, mayor cantidad de registros compartirán la misma combinación de valores en sus atributos cuasi-identificadores y por tanto habrá una mayor distorsión en la distribución de probabilidad asociada. Por el contrario, también es posible que el usuario exija un valor de *k* bajo, pero que, por la naturaleza de los datos, sea posible seguir manteniendo un equilibrio al incrementarlo. A continuación se presenta la propuesta de balance entre utilidad y privacidad que busca generar resultados adecuados para la aplicación de ejercicios de analítica de datos.

4.2.4. Balance entre utilidad y privacidad

Se propone que el flujo de anonimización cree automáticamente conjuntos con diferentes valores de *k* (2 hasta 20, de acuerdo con los valores más utilizados en la literatura [7], [17], [21], [59]), con lo cual se pueda determinar qué escenario es el que permite obtener la mayor privacidad, es decir el mayor *k*, al mismo tiempo que se logre mantener la distribución de los cuasi-identificadores implicados en el proceso. En este sentido, se define que el mejor balance está dado cuando se tiene el mayor valor de *k* que logre seguir demostrando que el test de *Kolmogorov-Smirnov* determina que cada cuasi-identificador mantiene la distribución con un nivel de significancia mínimo de 95%.

En la sección *6. Validación de la propuesta* pueden apreciarse los escenarios propuestos para evaluar la privacidad y utilidad de los datos tras hacer el proceso de anonimización utilizando el algoritmo *Distribution-based Mondrian*, en comparación con el algoritmo original basado en la mediana, propuesto por LeFevre *et al.* [7].

4.3. Anonimización centralizada de altos volúmenes de datos

Considerando la naturaleza recursiva de *Mondrian*, es importante analizar el caso en que se está trabajando en una sola máquina y el conjunto de datos a anonimizar pueda superar el tamaño que es posible mantener en memoria, debido a la generación de copias por cada nueva partición. LeFevre *et al.* [8] reconocen este escenario y por ello proponen un ajuste al algoritmo original para sólo permitir cargar en memoria un subconjunto de los datos que no lleguen a desbordarla. Su propuesta, *Rothko-T* se basa en el mismo principio de *Mondrian* de cortar los datos a través de la mediana, para determinar qué particiones se cargan a memoria de forma secuencial, con lo cual se anonimice sin generar un desbordamiento.

Partiendo de esta situación, se planteó una modificación al algoritmo *Rothko-T* para que en vez de determinar las particiones que caben en memoria a través de la mediana de los datos, se haga en base al criterio de corte *Distribution-based Split* propuesto en la sección *4.2.1. Algoritmo de anonimización*. Con ello se pretende seguir respetando la función de distribución de probabilidad de los datos, al mismo tiempo que se asegura que no se provocará un desbordamiento de memoria al anonimizar un alto volumen de datos. En la Figura 13 se muestra una representación gráfica del algoritmo propuesto, ***Rothko-D***, puesto que se inspira en la propuesta inicial de LeFevre *et al.* [8] pero se ajusta para tener en cuenta la Distribución original de los datos.

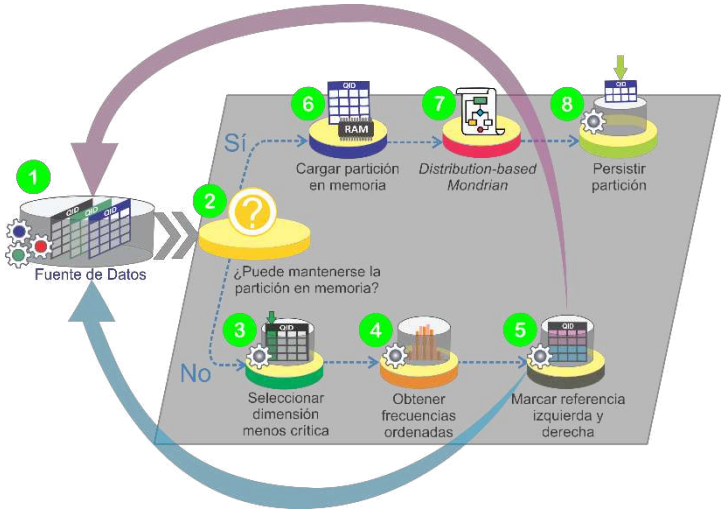


Figura 13. Algoritmo "Rothko-D".

La entrada del algoritmo es una referencia al conjunto de datos constituido por los valores cuasi-identificadores numéricos (1). De acuerdo con lo expuesto anteriormente, algunas de las operaciones realizadas en memoria a través del algoritmo *Distribution-based Mondrian* deberán ahora trasladarse para ser ejecutadas directamente en el Sistema de Gestión de Datos (SGD) que contiene originalmente la información. Por ello, en caso de no poder mantener la partición de datos en memoria (2), la selección de la dimensión menos crítica (3) y la obtención de las frecuencias sobre este atributo (4), son llevadas a cabo en el SGD. Tras obtener las frecuencias de la dimensión seleccionada, el algoritmo se encarga de marcar, a través de una columna adicional en los datos en el SGD, cada una de las dos nuevas particiones (5), de acuerdo con la

generación del corte realizado por el *Distribution-based Split*. Cada nueva región, en términos de la referencia a la partición de los datos, es evaluada de forma recursiva hasta que se estime que ésta no desbordará la memoria. En el caso en que la partición pueda mantenerse en memoria, se obtienen los datos con esa referencia desde el SGD (6) y son enviados al algoritmo *Distribution-based Mondrian* (7). Cada partición, después de ser anonimizada, es persistida el en SGD (8), hasta que todo el conjunto original haya sido analizado y concluya *Rothko-D*.

El algoritmo propuesto está inspirado en una estructura de árbol binario, donde cada nodo alberga una referencia a la partición de datos que se analizará para determinar si cabe en memoria. El árbol es recorrido recursivamente en forma *in-orden*, donde se calcula inicialmente el tamaño del conjunto de datos en el nodo raíz; si no cabe en memoria, se divide el conjunto en dos, donde primero se analiza el nodo izquierdo y luego el derecho [65]. El cálculo necesario para estimar a priori la memoria *RAM* necesaria para anonimizar un determinado conjunto de datos, no se encuentra definido en la propuesta de LeFevre *et al.* [8], por lo cual a continuación se presenta la propuesta generada para *Anonylitics*.

El cálculo de estimación de memoria necesaria para ejecutar el algoritmo *Distribution-based Mondrian*, se basa en la teoría de árboles. El objetivo es que, a partir del conocimiento de cuánto espacio ocupa el conjunto de datos inicial, pueda estimarse el espacio total que podría ocupar, tomando en cuenta el peor de los casos, que sería la frontera representada por el peso máximo que se podría mantener en memoria. Para el contexto del algoritmo, el peor de los casos en que mayor cantidad de veces se replicarían los datos a través de las recursiones, está representado por la situación en la cual todos los valores de los cuasi-identificadores son distintos y, por tanto, todas las clases de equivalencia (*EQ*) tendrán máximo *k* número de registros (para el principio *k-anonymity*). Cuando la cantidad de registros es par, cada *EQ* tendrá máximo *k* registros; para el caso impar, solo una de las *EQ* tendrá máximo *k+1* registros. Tomando la generalidad del caso par, se deduce que habrá a lo sumo tantas *EQ* como el número de registros dividido entre *k*. Por ejemplo, para un conjunto de 4 registros donde se quiere asegurar un *k=2*, el peor de los casos sería que se generaran $4/2$ *EQ*. La cantidad máxima de *EQ* es la cantidad máxima de hojas que podrá tener el árbol. La teoría de árboles indica que es posible calcular la cantidad máxima de niveles a partir de la fórmula (6) [65], [66]:

$$Máximo\ Nivel = Log_2 \left(\frac{\#\ registros}{k}\right) + 1 \qquad (6)$$

Donde el 2 se debe a que el árbol es binario, $\#\ registros/k$ es el número de hojas o cantidad máxima de clases de equivalencia (para valores no enteros se calcula mediante la función piso), y el 1 representa el nodo padre. En la Figura 14 puede observarse gráficamente un ejemplo de este cálculo para *k=2* y una cantidad cualquiera de cuasi-identificadores. El número de hojas es *8/2* pues hay 8 registros y el *k* es 2.

Debido a que el algoritmo *Distribution-based Mondrian* obedece a las propiedades de un árbol binario, donde el conjunto de datos inicial es dividido recursivamente en 2 conjuntos que podrían representar el conjunto total, cada nuevo nivel contendría el mismo número de registros del nodo padre, en el peor caso. Conociendo el número de niveles, y el peso de los datos originales, puede estimarse el tamaño máximo en que incurrirá el algoritmo, con la fórmula (7).

$$Tamaño\ máximo\ estimado = Peso\ del\ conjunto\ original * Máximo\ Nivel \quad (7)$$

Cabe resaltar que, dependiendo de la distribución de los datos, no siempre se obtendrá el peor de los casos, razón por la cual, la estimación de la memoria RAM requerida puede sugerir "reservar" recursos que en realidad no se requerirán. No obstante, la aproximación es válida en cuanto busca que nunca se presente la situación en que se desborde la memoria y se garantice la correcta ejecución de la anonimización.

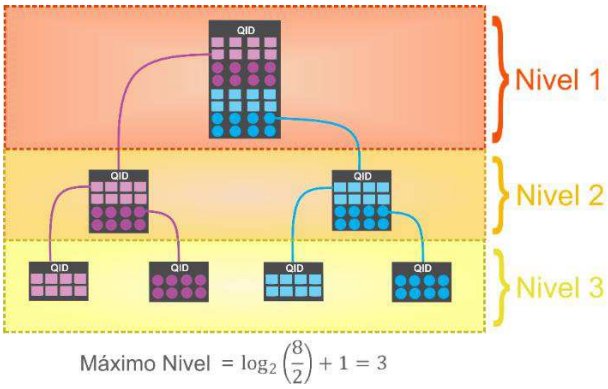


Figura 14. Ejemplo de funcionamiento del cálculo del máximo nivel del árbol para k=2.

En la sección *6. Validación de la propuesta* puede apreciarse el escenario creado para evaluar la pertinencia de la propuesta de anonimización centralizada de un alto volumen de datos.

4.4. Conclusiones

En síntesis, la propuesta de anonimización evidencia los siguientes aportes:

1. Algoritmo de anonimización que busca la preservación de la utilidad y la privacidad de los datos, denominado *Distribution-based Mondrian*.
2. Algoritmo de criterio de corte *Distribution-based Split* que permite cimentar uno de los pasos más relevantes del algoritmo *Distribution-based Mondrian*.
3. Propuesta de mecanismo de evaluación de privacidad a través del cálculo de los registros que se pueden singularizar antes y después de la anonimización.
4. Propuesta de mecanismo de evaluación de utilidad a través de la aplicación de la prueba estadística de dos muestras de Kolmogorov Smirnov para determinar si se mantiene la distribución univariada de los datos después de anonimizar.
5. Algoritmo de anonimización que considera el escenario de la transformación de un alto volumen de datos en una sola máquina, denominado *Rothko-D*.
6. Fórmula de cálculo de estimación del tamaño máximo que podría mantenerse en la memoria *RAM* al utilizar el algoritmo *Distribution-based Mondrian*.

El conjunto de planteamientos expuesto permite enmarcar una propuesta holística para la anonimización de diferentes conjuntos de datos que contengan cuasi-identificadores numéricos.

5. DISEÑO E IMPLEMENTACIÓN DE ANONYLITICS

Teniendo en cuenta la propuesta presentada, aquí se expone el diseño del sistema, el cual abarca el conjunto grueso de requerimientos identificados. En primera instancia se elaboraron unos prototipos no funcionales para acordar con el cliente las funcionalidades requeridas, lo cual puede ser consultado en los documentos anexos *Mockup Anonimización ligera* y *Mockup Anonimización completa*. Posteriormente se realizó la implementación de un subconjunto de los requerimientos, que permiten apoyar específicamente la propuesta de anonimización.

5.1.Diseño del sistema

Se diseñó el sistema de anonimización a través de un conjunto de componentes de software especializados para suplir los requerimientos identificados. A continuación se expone el diseño de componentes, su despliegue y su interacción.

5.1.1. Componentes

En la actualidad, la industria tecnológica ha abandonado las aplicaciones nativas o *standalone* a favor de las aplicaciones web puesto que las primeras presentan algunas desventajas relacionadas con la portabilidad entre sistemas operativos y requieren de un complejo esquema de actualización para solucionar fallos o brindar nuevas características [67]. En contraposición, las aplicaciones web no presentan problemas de portabilidad pues permiten acceder desde cualquier dispositivo que cuente con un navegador y la gestión de actualizaciones se simplifica al solo deber actualizar la versión del servidor. Por esta razón el sistema Anonylitics se diseñó bajo un esquema cliente servidor, donde a través de un navegador, diferentes usuarios desde distintos dispositivos, pueden acceder al sistema de anonimización. De igual forma, este esquema permite manejar de forma centralizada las actualizaciones para agregar nuevas características; en la eventual aparición de errores, es posible aplicar correcciones sin mayores contratiempos, pues se tiene control sobre el servidor web que alberga el sistema.

Bajo el esquema descrito, los componentes diseñados para Anonylitics se dividen en dos grandes grupos: *Front-end* y *Back-end,* cuya separación responde a la delegación de responsabilidades. El *Front-end* corresponde a la interfaz gráfica mediante la cual el usuario final interactúa con Anonylitics, generando los procesos de selección de datos y parametrización, de acuerdo con la Figura 9. De este modo, recibe las solicitudes y especificaciones del usuario final, para ser traducidas al segundo grupo de componentes, el *Back-end*, cuya responsabilidad es procesar tales peticiones, en este caso orientadas a la anonimización de los datos, y generar resultados que son nuevamente entregados al *Front-end* para ser visualizados por el usuario final. Los componentes de *Back-end* están entonces orientados a responder a las solicitudes del usuario final, las cuales se relacionan con tres funcionalidades gruesas: *i. Administración de Fuentes de Datos, ii. Administración de Proyectos de Anonimización* y *iii. Anonimización de Datos*, como se muestra en la Figura 15.

De acuerdo con el análisis hecho, el sistema fue diseñado a través de diferentes componentes de software cuyas responsabilidades están alineadas con las funcionalidades identificadas, a

partir del levantamiento de requerimientos realizado. En la Figura 16, puede apreciarse el diagrama de componentes del sistema en notación UML. El componente de *Front-end* está representado de color verde, mientras que los componentes de *Back-end* se muestran en color azul. Todo lo relacionado con almacenamiento de datos se representa en color morado. Adicionalmente, se evidencian entre recuadros punteados los componentes asociados a cada una de las tres funcionalidades identificadas en la Figura 15.

Figura 15. Funcionalidades de Anonylitics.

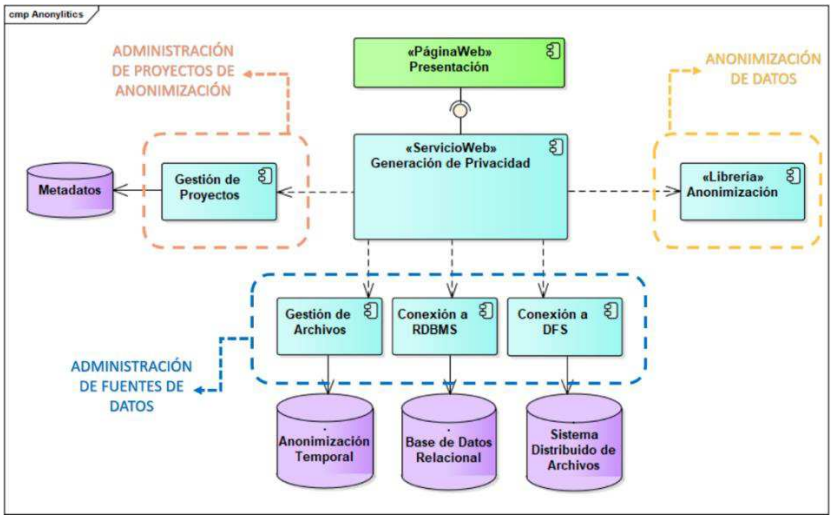


Figura 16. Diagrama de Componentes de Anonylitics.

La ***Presentación*** es el componente de *Front-end*. Este alberga un conjunto de vistas web mediante las cuales el usuario final define los parámetros necesarios para la creación, configuración y ejecución de un proyecto de anonimización. El diseño de cada vista está orientado a permitir la interacción del usuario con el sistema, respondiendo a las necesidades expuestas en los requerimientos asociados la administración de proyectos, el manejo de acceso a las fuentes de datos, el establecimiento de los metadatos, la definición de operaciones y revisión de resultados de ejecución de un proyecto de anonimización. Todo ello está enmarcado también en la

integración de menús de ayuda que permitan al usuario comprender los conceptos utilizados en la interfaz, a fin de generar una curva de aprendizaje corta, con lo cual pueda trabajar sus proyectos de anonimización.

La ***Gestión de Proyectos*** es el componente orientado a suplir la funcionalidad de *Administración de Proyectos de Anonimización*, que, de acuerdo con los requerimientos del caso de uso asociado, se enfoca en las operaciones *CRUD* (por sus siglas en inglés de crear, leer, actualizar y eliminar) de todos los parámetros de configuración de un proyecto, tales como: nombre del proyecto, datos de conexión, parametrización de las transformaciones realizadas sobre los datos y resumen de los resultados del proceso de anonimización. Como se evidencia en la Figura 16, el componente se conecta con un almacenamiento de "Metadatos", en el cual se almacenan los proyectos de anonimización. En la sección *5.1.1.1 Lenguaje de un proyecto de* anonimización se presenta el diseño de persistencia de la información relacionada con un proyecto.

La ***Gestión de Archivos***, ***Conexión a RDBMS*** y ***Conexión a DFS***, son los componentes diseñados para responder a la funcionalidad de *Administración de Fuentes de Datos*, la cual representa los requerimientos del caso de uso asociado, donde se debe poder acceder a diferentes orígenes para obtener los datos que serán sujetos de transformación para generar un conjunto anonimizado resultante. Existen tres componentes para abordar esta funcionalidad dado que cada origen tiene unas particularidades y parámetros de conexión diferentes. Adicionalmente, cada componente permite llevar a cabo en su respectivo Sistemas de Gestión de Datos, algunos procedimientos de anonimización, para aprovechar su poder de procesamiento y evitar el traslado de los datos, situación bastante costosa para altos volúmenes.

Existe un primer escenario donde se requiere obtener los datos para ser anonimizados, a partir de uno o varios archivos planos en formato tabular, para lo cual se diseñó el componente de ***Gestión de Archivos***, el cual se encarga de realizar todos los procedimientos de manejo de los archivos. Tal componente, de acuerdo con la Figura 16, se conecta con un almacenamiento estructurado (debido al formato tabular de los datos) de "Anonimización Temporal", cuyo objetivo es persistir los datos provenientes de los archivos, con el fin de facilitar la ejecución parcial de la anonimización, así como de procedimientos para evaluar los resultados de las transformaciones efectuadas, a la luz de las necesidades de privacidad y utilidad de los datos.

El segundo escenario corresponde al caso en que se quiere acceder a una o varias tablas que residen en un motor de base de datos relacional (*RDBMS*, por sus siglas en inglés), para lo cual se diseñó el componente de ***Conexión a RDBMS***, que se encarga de generar los procedimientos para obtener los datos y sus tipos, así como persistir los registros después del proceso de anonimización. En este caso, el componente se conecta con un almacenamiento estructurado, "Base de Datos Relacional", que corresponde al servidor donde se aloja la información. El componente se encarga de ejecutar los comandos necesarios, al igual que en el caso anterior, para ejecutar parcialmente las transformaciones, así como la evaluación de los resultados.

Finalmente, el tercer escenario se relaciona con el caso en que los datos residen en un conjunto distribuido de archivos, como por ejemplo Hadoop; para ello se diseñó el componente de ***Conexión a DFS***, el cual tiene la responsabilidad de permitir el acceso a un cluster de datos. El componente se conecta con un almacenamiento "Sistema Distribuido de Archivos", que corresponde al cluster de máquinas que alberga un alto volumen de datos a través de archivos en

formato tabular. En este caso, es más evidente que este sistema es el responsable de ejecutar la anonimización, así como la evaluación de los resultados, pues mover los datos hasta Anonylitics sería muy costoso en entornos de Big Data, además de complejizar el hecho de mantener los datos en memoria para hacer las correspondientes transformaciones que permitirán posteriormente aplicar los ejercicios de analítica.

El componente de ***Anonimización***, corresponde a una librería de funciones que permiten transformar los datos a través de las diferentes operaciones provistas en cada uno de los tipos de anonimización existentes en Anonylitics: *Ligera* y *Completa*, los cuales fueron identificados a través de los requerimientos del cliente. Cada algoritmo de anonimización debe ser lo suficientemente modular para permitir la ejecución de algunos procedimientos directamente en el Sistema de Gestión de Datos, y aprovechar las capacidades de procesamiento y la ubicación de los datos en sus respectivas fuentes. En la sección *4.2.1. Algoritmo de anonimización* se presentó el resultado de las iteraciones de diseño llevadas a cabo durante el proyecto, donde se generó una propuesta de algoritmo para anonimizar un conjunto de datos numéricos. De igual forma, el componente está diseñado para proveer los métodos de evaluación de utilidad y privacidad de los datos para generar el resumen de la ejecución del proyecto de anonimización. En la sección *4.2.2. Mecanismo de evaluación de utilidad* y *4.2.3. Mecanismo de evaluación* de privacidad se presentó el resultado de las iteraciones de diseño llevadas a cabo durante el proyecto, donde se generó una propuesta de evaluación de resultados de la anonimización.

Finalmente, está el componente de ***Generación de Privacidad***, el cual, se encarga de estructurar y articular todas las peticiones que se reciben desde la *Presentación* para hacer la delegación de responsabilidades a cada uno de los componentes especializados que permiten abordar las tres funcionalidades descritas en la Figura 15: *Administración de Fuentes de Datos*, *Administración de Proyectos de Anonimización* y *Anonimización de Datos*. Este componente se diseñó como un Servicio Web puesto que, de esta manera, no se limita a que el *Back-end* y el *Front-end* sean desarrollados bajo la misma tecnología, aprovechando las ventajas de las diferentes herramientas que proveen utilidades más apropiadas para uno u otro caso. El Servicio Web permite interoperabilidad debido al uso de estándares abiertos de modo que es posible intercambiar datos entre lenguajes diferentes ejecutados sobre diferentes plataformas [68].

La arquitectura propuesta, permite responder a las necesidades identificadas, sin embargo, es necesario profundizar en el diseño requerido para permitir el manejo de un proyecto de anonimización a través de todo el sistema.

5.1.1.1. Lenguaje de un proyecto de anonimización

De acuerdo con los requerimientos de administración de un proyecto de anonimización (Caso de Uso No.1, Componente de Gestión de Proyectos), se propone construir un conjunto de funcionalidades que permita construir proyectos de anonimización que sean parametrizables y reutilizables, a través de la definición de un "Lenguaje del Proyecto" cuyo propósito es identificar una estructura de almacenamiento que brinde diferentes facilidades como el manejo de los metadatos necesarios para ejecutar la anonimización de los datos, así como la persistencia y reutilización de fracciones o proyectos completos, incrementando la eficiencia en el uso del sistema

y disminuyendo los tiempos asociados a la re-creación y parametrización de conjuntos de datos que un usuario ya ha trabajado con anterioridad.

El Lenguaje del Proyecto está soportado por un esquema flexible que aprovecha las propiedades de las bases de datos no relacionales, para albergar todos los metadatos asociados a un proyecto, como: el nombre, el usuario que lo creó, los tipos de datos, los tipos de atributos, las operaciones realizadas y su posible parametrización (jerarquías, tablas de sustitución, por ejemplo), así como la referencia a la fuente de datos en donde se albergarán los datos originales y anonimizados. La existencia de este lenguaje se logra por medio de un esquema llamado "Proyecto", el cual define los parámetros necesarios que un usuario debe especificar para anonimizar un conjunto de datos. Este esquema consta de un conjunto finito de elementos semiestructurados que permiten definir diferentes metadatos que conforman el conjunto de un proyecto.

A través de la definición de un esquema de almacenamiento de un proyecto, un usuario puede persistir y reutilizar toda la información relevante a la hora de anonimizar un conjunto de datos, a través de la especificación de la información necesaria para establecer la conexión a una fuente de datos, la selección de un conjunto de ellos, el reconocimiento de los tipos de atributos (identificador, cuasi-identificador, sensible o común), los tipos de datos (numérico o categórico), la definición de operaciones sobre cada atributo y su correspondiente parametrización, la definición del tipo de anonimización (ligera o completa) y los principios de anonimización para evaluar los resultados. De esta forma se comprende que un Proyecto de anonimización es equivalente a un conjunto de metadatos que describen todo el proceso llevado a cabo. No obstante, debido a las diferentes necesidades de anonimización de diversos usuarios, los elementos a almacenar no son estáticos, pudiendo o no estar presentes, dependiendo de la configuración establecida. Con el objetivo de manejar esta flexibilidad para tener diferentes campos en diversas situaciones, se estableció trabajar sobre una estructura de datos basada en documentos tipo JSON con el fin de obtener provecho de las propiedades de la definición de un esquema dinámico BSON que es altamente flexible y permite definir estructuras complejas [69].

A continuación, en la Figura 17, se puede observar la definición del esquema, donde es posible apreciar cada uno de los elementos que lo constituyen, junto a una breve descripción, que puede ser: una expresión regular o una expresión matemática, cuyo fin es ilustrar cuáles son los valores esperados en cada uno de ellos.

```
{
  "user": "([a-zA-Z0-9_-])?",
  "project_name": "([a-zA-Z0-9_-])?",
  "anonymization_type": a | a ∈ {"Completa", "Ligera" },
  "connection": {
                    "ip": "(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.
                     (25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.
                     (25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.
                     (25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])$"
                    "port": [0-9]+,
                    "database":"([a-zA-Z0-9_-])?",
                    "username":"([a-zA-Z0-9_-])?",
                    "password":"([a-zA-Z0-9_-])?",
```

```
                    },
  "tables": [
                {
                  "table_name": "([a-zA-Z0-9_-])?",
                  "table_schema": "([a-zA-Z0-9_-])?",
                  "anonymized_table_name": "([a-zA-Z0-9_-])?",
                  "anonymized_table_schema": "([a-zA-Z0-9_-])?",
                  "columns": [
                                {
                                  "column_name":"([a-zA-Z0-9_-])?",
                                  "data_type": dt | dt ∈ {"Numérico", "Categórico },
                                  "attribute_type": at | at ∈
{     "Identificador",Sensible,      },,
{"Cuasi-identificador", "Común" }  ,
                                  "keep_data_type": k | k ∈ {True, False},
                                  "operation": {
                                                  "name": o | o ∈ { "Sustituir", Suprimir, }
                                                                  {"Generalizar", "Cifrar" },
                                                  "parametrization": [
                                                                  {
                                                                    "initial_val":
                                                                    "([a-zA-Z0-9._-])?",
                                                                      "target_val":
                                                                    "([a-zA-Z0-9._-])?"
                                                                  },
                                                                    { }, … ,{ }
                                                                  ]
                                                  },
                                    },
                                    { }, … , { }
                                ]
                  }
                  { }, … , { }
              ],
  "repeated_attributes": [
                                [
                                    {
                                        "column_name": "([a-zA-Z0-9_-])?",
                                        "table_name": "([a-zA-Z0-9_-])?"
                                    },
                                    { }, … , { }
                                ],
                                [ ], … , [ ]
                            ],
  "principles": [
                        {
                            "principle_name": p | p ∈ {"k-anonymity", "t-closeness" },
```

```
                    "principle_value": [0-9]+
                },
                { }, ... , { }
            ]
}
```

Figura 17. Estructura de un proyecto de anonimización.

En la Tabla 2, se describe cada objeto (elemento) JSON que compone un Proyecto.

Tabla 2. Descripción de elementos de un proyecto de anonimización.

Elemento	Descripción
User	Almacena el username correspondiente al usuario que creó el proyecto y obedece al requerimiento que hace referencia a que el sistema debe permitir guardar los cambios hechos en un proyecto de anonimización en el sentido que los cambios realizados se deben asociar a un usuario.
Project _name	En este campo se almacena el nombre del proyecto de anonimización, y obedece al requerimiento de que el sistema debe permitir nombrar un proyecto.
Anonymization _type	Almacena el tipo de anonimización del proyecto, correspondiendo a *Ligera* o *Completa*, que obedece al requerimiento que hace referencia a determinar el tipo de anonimización a aplicar.
Connection	Es una colección de elementos que son necesarios para establecer la conexión con una fuente de datos. La estructura del JSON variará de acuerdo con el tipo de sistema con el cual se desea realizar la conexión. Si la fuente es una base de datos relacional RDBMS o un archivo (caso en que los datos también se almacenan en una base de datos relacional temporal), se tendrían los parámetros de conexión acordes al estándar internacional de acceso y uso de bases de datos SQL CLI [70], donde a través de una conexión ODBC o JDBC, se establecerán los parámetros necesarios para establecer la conexión como: el nombre de la base de datos, el puerto, la IP, el nombre del usuario y la contraseña de acceso. De otra forma, si la fuente de datos correspondiera a un sistema de archivos distribuido, su estructura cambiaría, pues los parámetros serán distintos dependiendo del componente que se esté usando del ecosistema (Hive, Impala, Spark, etc., para el caso de HDFS, por ejemplo). Este campo obedece a los requerimientos relacionados con la conexión con diversos orígenes de datos y a la importación de archivos planos en formato tabular.

Tables Es la colección que almacenan los metadatos asociados a las tablas a anonimizar, donde cada tabla tiene (aplica para todas las fuentes):

- Referencia a los datos de origen: el nombre de la tabla y el esquema al cual pertenece.
- Referencia a los datos anonimizados: el nombre de la tabla y el esquema donde quedarán almacenados los datos anonimizados.
- Una colección de metadatos sobre las columnas asociada a cada tabla.

Cada columna almacena metadatos como: el nombre de la columna, el tipo de dato que soporta (numérico o categórico), si quiere mantener el tipo de dato (esto sólo aplica para cuando el dato es originalmente numérico pues podría querer mantenerse así o podría modificarse a categórico), el tipo de atributo (identificador, cuasi-identificador, sensible o común) y el tipo de operación a aplicar (sustitución, generalización, supresión, ciframiento o N/A). En caso de aplicar alguna operación que requiera una parametrización (sustitución o generalización para los tipos de datos categóricos) es necesario conservar la estructura que se describe en la Figura 18 y en la Figura 19.

La operación de Sustitución requiere especificar el valor original del registro y el valor de "destino" por el cual será reemplazado, de acuerdo con la Figura 18.

```
"operation": {
        "name": "Sustituir",
        "parametrization": [
                {
                  "origin_val": "([a-zA-Z0-9_-])?"
                  "target_val": "([a-zA-Z0-9_-])?"
                },
                {
                  "origin_val": "([a-zA-Z0-9_-])?",
                  "target_val": "([a-zA-Z0-9_-])?"
                }
        ]
  }
```

Figura 18. Estructura del elemento "operation" para una sustitución.

Para las operaciones de generalización es necesario especificar por cada valor categórico existente, una colección con los valores correspondientes a cada nivel de la jerarquía. El campo *level* es el número del nivel de la generalización aplicada, mientras que el campo *value* es valor correspondiente en ese nivel.

```
"operation": {
        "name": "Generalizar",
        "parametrization": [
```

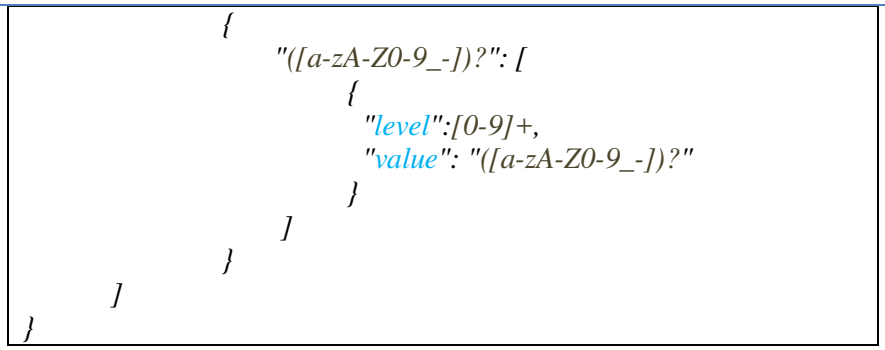


Figura 19. Estructura de la operación de generalización.

Todos los elementos que hacen parte del elemento *Tables* reflejan los requerimientos que hacen referencia a determinar los tipos de atributos, consignados en el caso de uso *Establecer metadatos*. De igual forma se ven reflejados los requerimientos que hacen referencia a la selección de operaciones y el establecimiento de los parámetros necesarios para llevar a cabo la anonimización.

Repeated_attributes	Es la colección donde se almacenan los atributos (columnas) que se repiten entre las tablas a anonimizar, cuyo objetivo es garantizar que sobre estos campos se aplique la misma operación, con el fin de mantener la consistencia, como aparece consignado en los requerimientos que hacen referencia a que puede existir un atributo que se presente en varias, y que se debe asignar la misma operación de anonimización sobre el atributo repetido en diferentes tablas.
Principles	Es la colección donde se almacenan los principios de anonimización que se evaluarán sobre los datos al hacer la transformación. Su definición obedece a la necesidad de almacenar los principios aplicados con el fin de evaluarlos.

En el documento anexo *Documento de Diseño de Software*, se encuentra un ejemplo que ilustra cómo serían almacenados todos los metadatos de un proyecto de anonimización.

5.1.2. Interacción

Los componentes del sistema llevan a cabo diferentes interacciones con el objetivo común de permitir cumplir las funcionalidades que se describen en los requerimientos y casos de uso del sistema. En este sentido, en la Figura 20, se evidencia la representación *BPMN (Business Process Management Notation)* del macro proceso conformado por distintas actividades que se ejecutan de manera secuencial para anonimizar un conjunto de datos. Como objeto del proceso se tiene el lenguaje de un proyecto de anonimización, pues permite albergar todas las propiedades asociadas, que son diligenciadas a través de las actividades. Este proceso es la materialización de la Visión Global representado en el flujo de anonimización plasmado en la Figura 9 de la sección *4. Anonylitics: Una propuesta de anonimización orientada a la analítica de datos*. Cada actividad representa interacciones específicas entre componentes del sistema.

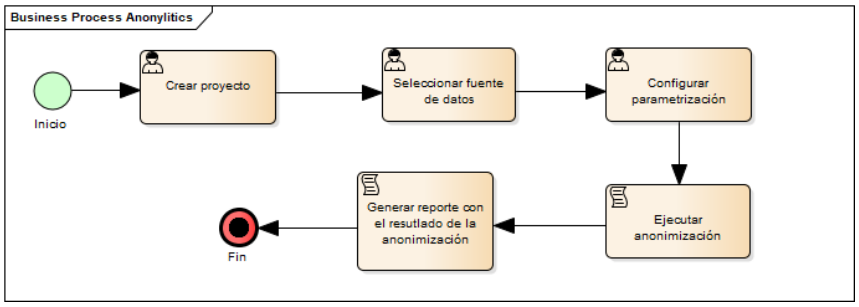


Figura 20. Proceso de anonimización en Anonylitics.

El proceso inicia con la actividad ***Crear proyecto***, donde un usuario especifica desde el componente de *Presentación*, que desea crear un nuevo proyecto de anonimización. Al ingresar los parámetros necesarios, la *Presentación* envía dicha solicitud al componente de *Generación de Privacidad*, el cual crea el proyecto a través del componente de *Gestión de Proyectos*, en el almacenamiento de "Metadatos". Finalmente, el componente de *Generación de Privacidad* retorna un mensaje de aprobación a la *Presentación* para continuar con el proceso.

Luego, el usuario puede realizar la acción de ***Seleccionar fuente de datos***, a través del componente de *Presentación*. Es posible importar archivos en formato tabular, tablas de una base de datos relacional o tablas de un sistema distribuido de archivos. Cada fuente de datos cuenta con un componente especializado de conexión destinado a atender las diferentes necesidades (*Conexión a DFS*, *Conexión a RDBMS* y *Gestión de Archivos*). De manera general, el usuario escoge una fuente de datos, ingresa los parámetros de conexión y posteriormente se hace uso de alguno de los componentes especializados para obtener información de las propiedades de las tablas disponibles. De esta manera el usuario selecciona el conjunto de datos a anonimizar y envía la petición al componente de *Gestión de Proyectos*, el cual orquesta las solicitudes y finalmente genera el almacenamiento de los metadatos del conjunto de datos escogido.

El componente de *Presentación* recibe un mensaje de aprobación y el sistema le permite al usuario dar inicio a la actividad de ***Configurar parametrización*** sobre los conjuntos de datos seleccionados, eligiendo las operaciones de anonimización y configurando lo que se requiera (cargar sustituciones o jerarquías de generalización, por ejemplo), que varían de acuerdo con el tipo de anonimización (ligera o completa), para continuar con el proceso. El componente de *Gestión de Privacidad* recibe los parámetros y delega la responsabilidad al componente de *Gestión de Proyectos* para almacenar tan información en la base de datos de "Metadatos".

Así, *Anonylitics* cuenta con toda la información necesaria para continuar con la actividad de ***Ejecutar anonimización***, dependiendo de si es ligera o completa. El componente de *Gestión de Privacidad* solicita al de *Anonimización* ejecutar un algoritmo, de acuerdo con la parametrización establecida. Esto genera generar un conjunto de datos anonimizados, que son persistidos en la fuente de datos original (*RDBMS* o *DFS*), o son descargados en un archivo plano.

Finalmente, el sistema realiza la actividad de ***Generar reporte con el resultado de la anonimización*** donde, a través del componente de *Anonimización*, se evalúa la utilidad y la privacidad

de los datos para una anonimización completa, para posteriormente mostrarle al usuario a través de la *Presentación*, los resultados de la ejecución.

El detalle de la interacción entre componentes se encuentra consignado en el documento anexo *Documento de Diseño de Software*.

5.1.3. Despliegue

De acuerdo con los componentes expuestos, se genera una posible distribución en máquinas, a través del diagrama de despliegue mostrado en la Figura 21. Allí se pueden distinguir en color naranja los servidores administrados por *Anonylitics*, y en amarillo los servidores externos que almacenan datos que podrían anonimizarse usando el sistema. Entre los servidores de *Anonylitics*, se encuentran dos tipos: *aplicación* y *almacenamiento*. En el ***Servidor de Aplicación*** se encuentran desplegados todos los componentes desarrollados para el sistema: *Presentación*, *Gestión de Privacidad*, *Anonimización*, *Gestión de Proyectos*, *Gestión de Archivos*, *Conexión a DFS* y *Conexión a RDBMS*. En el ***Servidor de Almacenamiento*** están las bases de datos gestionadas por *Anonylitics*: *Metadatos y Anonimización Temporal*. Debido a que el sistema puede conectarse a diferentes fuentes de datos, se puede observar el ***Servidor DFS*** y el ***Servidor RDBMS***. *Anonylitics* tiene acceso a ellos a través de los componentes de conexión.

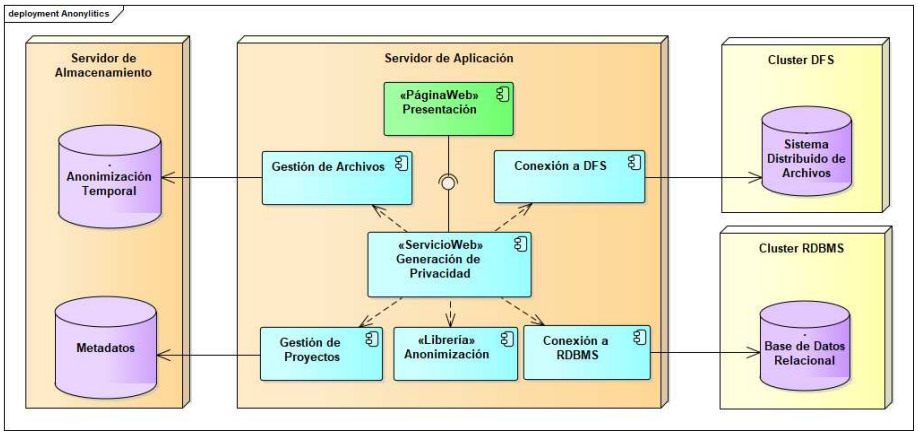


Figura 21. Diagrama de despliegue de Anonylitics.

5.2.Implementación

De acuerdo con el diseño del sistema propuesto, se seleccionó un subconjunto de requerimientos que permiten soportar los planteamientos presentados en la sección *4. Anonylitics: Una propuesta de anonimización orientada a la analítica de datos*, de acuerdo con la priorización realizada durante cada sprint de la fase de construcción del trabajo de grado (*Anexo 9.1. Requerimientos implementados por Sprint*). De acuerdo con tal priorización, la implementación incluyó las funcionalidades relacionadas con la anonimización de archivos planos en formato tabular, a través de la creación de un proyecto, donde se configuran los metadatos y se parametrizan las operaciones a aplicar. Se construyó la anonimización completa, a través del cumplimiento del principio *k-anonymity*, soportando operaciones de supresión y generalización,

usando el algoritmo *Distribution-based Mondrian* y *Rothko-D*. Los componentes implementados son: la página web donde se encuentra la *Presentación*, el servicio web de *Generación de Privacidad*, la librería de *Anonimización*, la *Gestión de Archivos* y la *Gestión de Proyectos*.

La interfaz gráfica de *Anonylitics* ilustra al usuario las distintas opciones y posibles configuraciones que puede realizar con el sistema. Cada acción efectuada sobre la página web envía una petición al grupo de componentes de *Back-end*, orquestados mediante un servicio web *REST*, bajo el protocolo *HTTP*. Se seleccionó *REST* dado que ofrece capacidades distintas a otros servicios web como los *SOAP*, en la medida que cada *URL* permite identificar diferentes recursos y efectuar consultas, usando de forma nativa comandos *HTTP* (GET, POST, DELETE y PUT) [71]; *SOAP* solo utiliza HTTP como protocolo de transporte, por lo cual *REST* resulta ser un protocolo más ligero que permite obtener mejores tiempos de respuesta [72].

El componente de *Front-end* está desarrollado bajo un conjunto de tecnologías JavaScript, que permiten proveer interfaces fluidas al usuario bajo un esquema *SPA* (*Single Page Application*) que delega la lógica de presentación al cliente. Esto permite lograr menores tiempos de repuesta, pues posibilita la distribución de la carga del servidor *web*, con lo cual sólo se encarga de enviar la información necesaria a través de un JSON de acuerdo con la solicitud. De esta forma el *framework* que se ejecuta en el cliente, se dedica únicamente a construir la presentación dinámicamente. El componente fue desarrollado bajo los *frameworks* de *AngularJS* [73] y *Bootstrap* [74] para proveer un diseño basado en plantillas, que permite la reutilización de componentes gráficos y la disminución de la carga asociada al mantenimiento y construcción. Adicionalmente, este escenario permite apoyarse en herramientas de automatización y gestión de dependencias como *Gulp.js* y *Grunt* [75], útil para la implementación.

El conjunto de componentes de *Back-end* está conformado por aquellos que tienen la responsabilidad de procesar las peticiones obtenidas a través del Servicio web *REST*. Estos componentes fueron implementados bajo el lenguaje de programación *Python* 3.0, debido a la naturaleza de los algoritmos requeridos para anonimizar, la necesidad de la aplicación de técnicas estadísticas y la fácil integración de *Python* con aplicaciones *web* [76]. Este lenguaje resulta ser adecuado pues permite administrar todo el flujo de anonimización, desde diferentes fuentes de datos hasta el procesamiento y envío de resultados a la presentación.

El almacén de "Metadatos", donde se persisten los Proyectos de Anonimización se construyó bajo la estructura de un objeto *JSON* complejo, para lo cual se seleccionó *MongoDB* como base de datos ya que es de tipo *NoSQL* y está diseñada para albergar esquemas dinámicos *BSON* en documentos *JSON*. Es una herramienta ampliamente utilizada en la industria [77]. El almacén de "Anonimización Temporal" donde se persisten los datos provenientes de archivos planos en formato tabular, es una base de datos relacional *MySQL*, la cual fue seleccionada por su importante recorrido y reconocimiento como herramienta de código abierto que puede permitir generar procedimientos de forma óptima sobre los datos [78].

En el documento anexo *Manual de Instalación* se da mayor detalle para la configuración del sistema implementado. En el documento anexo *Manual de Usuario* se presenta la interfaz gráfica del sistema, a través de una guía de cómo ser utilizada para anonimizar un conjunto de datos.

6. VALIDACIÓN DE LA PROPUESTA

En la presente sección se evidencia el caso de estudio de validación que abarca dos escenarios. En el primero se realiza la validación del algoritmo propuesto, *Distribution-based Mondrian*, evaluando la utilidad y privacidad, siguiendo lo dispuesto en las secciones *4.2.2. Mecanismo de evaluación de utilidad*, *4.2.3. Mecanismo de evaluación de privacidad* y *4.2.4. Balance entre utilidad y privacidad*. En el segundo escenario se evalúa en qué contexto sería aplicable el algoritmo *Rothko-D* para mostrar su pertinencia en caso de que se requiera anonimizar un alto volumen de datos en una sola máquina.

6.1.Primer escenario

Bajo el primer escenario se trabajó con dos conjuntos de datos para evaluar la privacidad y utilidad, después de aplicar la anonimización completa, utilizando el algoritmo *Distribution-based Mondrian*. Se compararon los resultados obtenidos frente al algoritmo original basado en la mediana, que en adelante se referirá como *Median-based Mondrian*. Para cada caso se presenta el punto de equilibrio encontrado para generar el balance más adecuado entre privacidad y utilidad.

6.1.1. Conjuntos de datos

El primer conjunto de datos es una base llamada *Adult* del repositorio *Machine Learning UCI* [79], el cual se ha convertido en el artefacto común para hacer las validaciones de los algoritmos de anonimización propuestos en la literatura [6], [7], [40], [80]. El conjunto de datos cuenta con 32.561 registros y 13 atributos. Esta base contiene información demográfica y censal de año 1994 de Estados Unidos, así como una variable sobre los ingresos anuales agrupados en dos clases: quienes ganaban más de cincuenta mil dólares y los restantes. En la Tabla 3 se pueden apreciar los metadatos asociados.

Tabla 3. Metadatos del conjunto de datos Adult.

Nombre	Tipo de Dato	Tipo de Atributo	Descripción
age	Numérico	Cuasi-identificador	Edad del censado
workclass	Categórico	Cuasi-identificador	Tipo de empleado
education	Categórico	Cuasi-identificador	Nivel de educación del censado
marital_status	Categórico	Cuasi-identificador	Estado civil del censado
occupation	Categórico	Cuasi-identificador	Ocupación del censado
relationship	Categórico	Cuasi-identificador	Relación dentro del núcleo familiar
race	Categórico	Cuasi-identificador	Raza del censado
sex	Categórico	Cuasi-identificador	Género del censado
capital_gain	Numérico	Cuasi-identificador	Capital ganado en el año
capital_loss	Numérico	Cuasi-identificador	Capital perdido en el año

hours_per_week	Numérico	Cuasi-identificador	Horas de trabajo a la semana
native_country	Categórico	Cuasi-identificador	País de procedencia
annual_income	Categórico	Sensible	Ingresos anuales del censado (>50K, <=50K)

De los 13 atributos disponibles, 12 se consideran cuasi-identificadores puesto que son datos que podrían permitir singularizar o distinguir a una persona, al realizar un ataque asociando la información con una base de datos externa o usando conocimiento previo. De éstos, sólo 4 son numéricos, y por tanto fueron seleccionados para realizar la validación, a través del uso de *Anonylitics*: *age*, *capital_gain, capital_loss* y *hours_per_week*. Existe un 1 atributo sensible, *anual_income*.

El segundo conjunto de datos es una base entregada por la alianza CAOBA, sobre datos reales de uno de sus clientes, el cual cuenta con 493.399 registros y 12 atributos. Esta base contiene información demográfica sobre personas naturales o jurídicas que están vinculadas a una entidad bancaria, así como sus ingresos. En la Tabla 4 se pueden apreciar los metadatos asociados.

Tabla 4. Metadatos del conjunto de datos CAOBA.

Nombre	Tipo de Dato	Tipo de Atributo	Descripción
num_doc	Numérico	Identificador	Número de Documento de la persona
tipo_doc	Categórico	Cuasi-identificador	Tipo de Documento de la persona: Cédula de ciudadanía, de extranjería, etc.
tipo_persona	Categórico	Cuasi-identificador	Tipo de persona: Natural o Jurídica
anhos_v	Numérico	Cuasi-identificador	Años que la persona lleva vinculado a la empresa
num_hijos	Numérico	Cuasi-identificador	Número de hijos de la persona
anho_nacimiento	Numérico	Cuasi-identificador	Año de nacimiento de la persona
valor_ingresos	Numérico	Sensible	Valor de los ingresos de la persona
nivel_academico	Categórico	Cuasi-identificador	Nivel académico de la persona
segmento	Categórico	Cuasi-identificador	Código de segmento de la persona dentro de la empresa
ciudad	Categórico	Cuasi-identificador	Ciudad donde reside la persona
depto	Categórico	Cuasi-identificador	Departamento donde reside la persona
pais	Categórico	Cuasi-identificador	País donde reside la persona

De los 12 atributos disponibles, 10 se consideran cuasi-identificadores puesto que son datos que podrían permitir singularizar o distinguir a una persona, al realizar un ataque asociando la información con una base de datos externa o usando conocimiento previo. De éstos, sólo 3 son numéricos, y por tanto fueron seleccionados para realizar la validación, a través del uso de

Anonylitics: *anhos_v*, *num_hijos* y *anho_nacimiento*. Existe un identificador, *num_doc*, y un atributo sensible, *valor_ingresos*.

6.1.2. Resultados

Se utilizó el sistema construido, con el fin de parametrizar un proyecto de anonimización que permitiese transformar los datos para posteriores ejercicios de analítica. Se definió el *k=2* por ser el valor más utilizado en la literatura. Bajo este escenario, el sistema arrojó los resultados presentados en la Tabla 5, respecto a la evaluación de la privacidad.

Tabla 5. Evaluación de privacidad de los datos Adult y CAOBA (Distribution-based Mondrian).

	Adult	***CAOBA***
Cantidad de registros que se pueden singularizar antes de anonimizar	3.811	6.230
Cantidad de registros que se pueden singularizar después de anonimizar	0	0

Se puede concluir que para ambos conjuntos los datos cumplen con el criterio de privacidad, pues ninguno puede ser singularizado ya que existen por lo menos 2 registros que tienen la misma combinación de valores en los cuasi-identificadores numéricos. Sin hacer la anonimización, podrían singularizarse 3.811 registros de *Adult* y 6.230 de *CAOBA*, dando espacio a un atacante para inferir la información sensible de estas personas, es decir el valor de sus ingresos anuales y algunos cuasi-identificadores.

En la Tabla 6 y Tabla 7 pueden apreciarse los resultados obtenidos respecto a la evaluación de la utilidad, medido en la preservación de la distribución de los atributos anonimizados para ambas bases.

Tabla 6. Evaluación de utilidad de los datos Adult (Distribution-based Mondrian).

Cuasi-identificador	Nivel de significancia del Test de Kolmogorov Smirnov
age	0,999999833
capital_gain	0,99997873
capital_loss	1
hours_per_week	0,999999999986504

Tabla 7. Evaluación de utilidad de los datos CAOBA (Distribution-based Mondrian).

Cuasi-identificador	Nivel de significancia del Test de Kolmogorov Smirnov
anhos_v	1
num_hijos	0,99
anho_nacimiento	1

De acuerdo con la Tabla 6, todos los atributos del conjunto de datos *Adult*, presentan un nivel de significancia superior a 0,95 en la prueba de dos muestras de Kolmogorov Smirnov. Por lo tanto, se puede concluir que, cada uno de los cuatro cuasi-identificadores preserva su utilidad

pues no existe evidencia estadística para rechazar la hipótesis nula bajo la cual el atributo anonimizado y el atributo original proceden de la misma función de distribución de probabilidad. De igual manera sucede con el conjunto de datos *CAOBA*, pues todos los atributos presentan un nivel de significancia superior a 0,95, de acuerdo con la Tabla 7.

Ambas evaluaciones para cada conjunto de datos indican que el algoritmo *Distribution-based Mondrian* logra preservar la utilidad y privacidad de los datos, de acuerdo con la parametrización establecida. Con el ánimo de comparar los resultados respecto a la utilidad de los datos, usando el algoritmo original, *Median-based Mondrian*, se hizo el ejercicio nuevamente en *Anonylitics*, con un *k=2,* pero con el algoritmo original. Se obtuvieron los resultados mostrados en la Tabla 8 y Tabla 9.

Tabla 8. Evaluación de utilidad de los datos CAOBA (Median-based Mondrian).

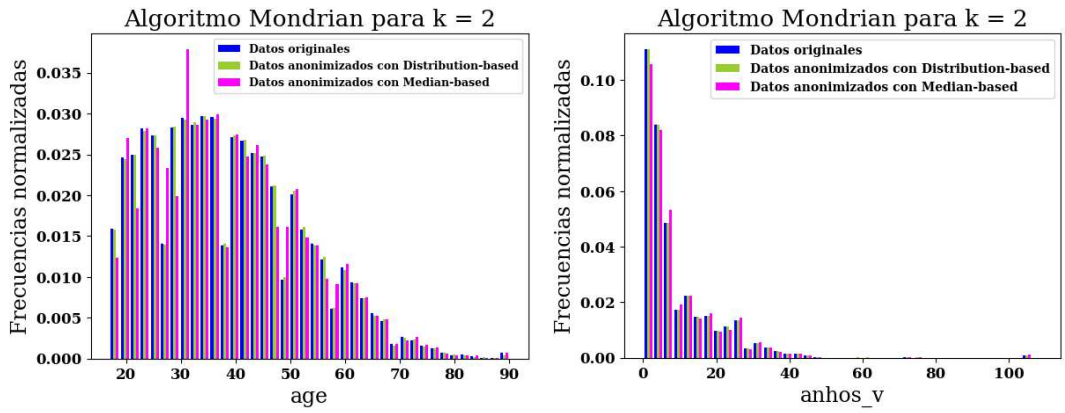
Cuasi-identificador	Nivel de significancia del Test de Kolmogorov Smirnov
anhos_v	0
num_hijos	2,e-228
anho_nacimiento	6,96e-118

Tabla 9. Evaluación de utilidad de los datos Adult (Median-based Mondrian).

Cuasi-identificador	Nivel de significancia del Test de Kolmogorov Smirnov
age	0,000153054032166
capital_gain	0,999998300598
capital_loss	1
hours_per_week	3,75978494175e-185

De acuerdo con la Tabla 8, todos los atributos del conjunto de datos *CAOBA*, presentan un nivel de significancia inferior a 0,95. Por lo tanto, se puede concluir que ninguno de los tres preserva su utilidad, pues existe evidencia estadística para rechazar la hipótesis nula bajo la cual el atributo anonimizado y el atributo original proceden de la misma función de distribución de probabilidad. Sucede una situación similar con el conjunto de datos *Adult*, pues sólo 2 de los 4 atributos presentan un nivel de significancia superior a 0,95, de acuerdo con la Tabla 9. Por lo tanto, se puede concluir que dos de ellos no preservan su utilidad.

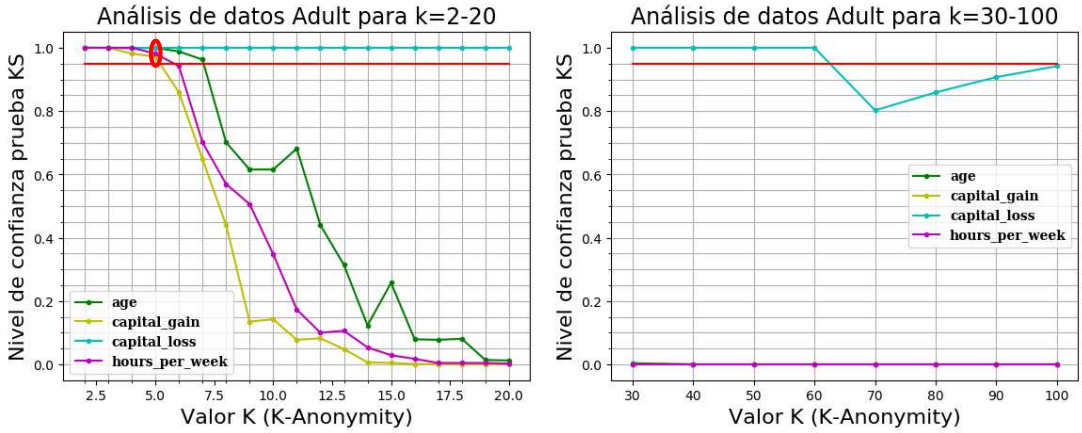
En la Figura 22, a manera de ejemplo, se puede apreciar la distribución de uno de los atributos cuasi-identificadores de cada conjunto de datos. En la Figura 22 *(a)* se presenta el atributo *age* de *Adult*, y en la Figura 22 *(b)* se presenta el atributo *anhos_v* de *CAOBA*, donde en cada cual se observa que la frecuencia normalizada de los datos (para facilitar su visualización) tiene cambios significativos, respecto al atributo original (color fucsia respecto a azul), al emplear el algoritmo original *Median-based Mondrian* para realizar la anonimización. En contraste, la aproximación propuesta en el presente trabajo, *Distribution-based Mondrian*, demuestra ligeras variaciones en la función de distribución de probabilidad de los datos (color verde respecto a azul), por lo cual la prueba de Kolmogorov Smirnov genera resultados favorables concluyendo que la distribución se preserva y que por tanto la utilidad permitirá construir ejercicios de analítica más apropiados para la toma de decisiones.



(a) A dult: age *(b) CAOBA: anhos_v*

Figura 22. Comparación de distribución original, anonimizada con Median-based Mondrian y Distribution-based Mondrian

Para finalizar, a través de *Anonylitics*, se generaron pruebas con diferentes valores de *k*. En la Figura 23 *(a)* puede observarse el nivel de significancia de la prueba de Kolmogorov Smirnov al variar el *k*. De allí se concluye que el mejor balance se logra con un *k=5*, que es el valor máximo que podría proveerse para garantizar tanto la privacidad de los datos como su utilidad. En este punto, cada uno de los cuatro cuasi-identificadores presenta un nivel de significancia por encima de 0,95 (línea roja) y por tanto puede afirmarse que se mantiene la distribución respecto a los atributos originales.

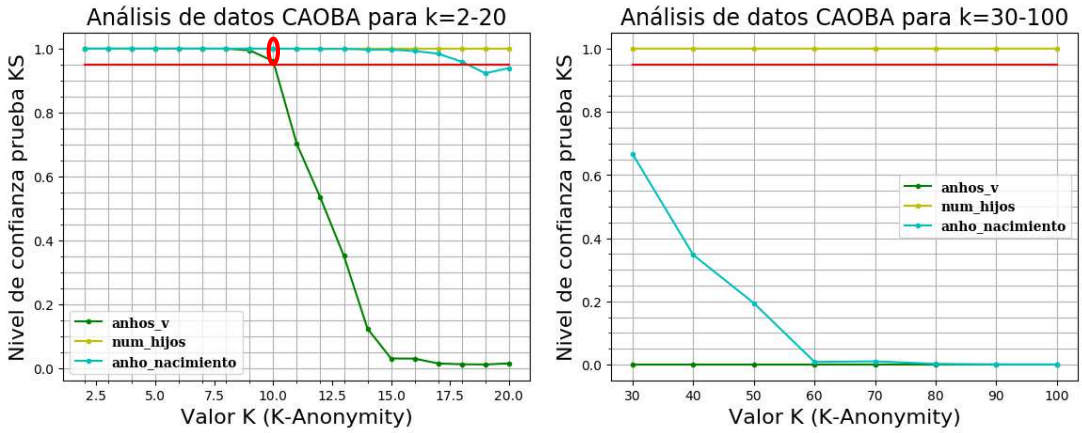


(a) Variando el k entre 2 y 20 *(b) Variando el k entre 30 y 100*

Figura 23. Evaluación de preservación de la privacidad y utilidad de los datos Adult para diferentes valores de k, usando el algoritmo Distribution-based Mondrian.

De igual forma se generaron pruebas con valores de *k* entre 30 y 100 para evidenciar que, a medida que se incrementa este parámetro, la utilidad de los datos se deteriora pues debe modificarse una mayor cantidad de registros para lograr cumplir con el principio *k-anonymity*. En la Figura 23 *(b)* se presentan los resultados para estos valores.

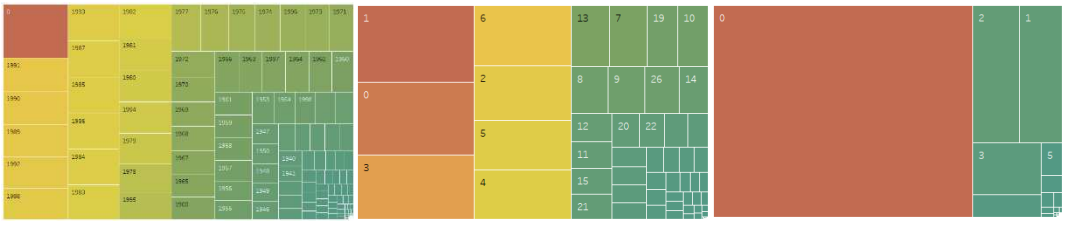
Para el conjunto de datos *CAOBA*, en la Figura 24 *(a)* puede observarse el nivel de significancia de la prueba de Kolmogorov Smirnov al variar el *k*. De allí se concluye que el mejor balance se logra con un *k=10*, que es el valor máximo que podría proveerse para garantizar tanto la privacidad de los datos como su utilidad. En este punto, cada uno de los tres cuasi-identificadores presenta un nivel de significancia por encima de 0,95 (línea roja) y por tanto puede afirmarse que se mantiene la distribución respecto a los atributos originales.



(a) Variando el k entre 2 y 20 *(b) Variando el k entre 30 y 100*

Figura 24. Evaluación de preservación de la privacidad y utilidad de los datos CAOBA para diferentes valores de k, usando el algoritmo Distribution-based Mondrian.

De igual forma se generaron pruebas con valores de *k* entre 30 y 100 para evidenciar que, a medida que se incrementa este parámetro, la utilidad de los datos se deteriora pues debe modificarse una mayor cantidad de registros para lograr cumplir con el principio *k-anonymity*. En la Figura 24 *(b)* se presentan los resultados para estos valores. Se observa una situación particular para el atributo *num_hijos* puesto que, incluso con un *k=100*, éste sigue preservando la distribución original de los datos. Lo anterior se debe a que este cuasi-identificador tiene una distribución bastante diferente a los otros dos, como se puede observar en la Figura 25 *(c)*. Este atributo tiene más del 70% de los datos en valor cero (color rojo), mientras que en la Figura 25 *(a)* y Figura 25 *(b)* se aprecian distribuciones con menor sesgo. El atributo *num_hijos* tiene un rango menor que las otras dos variables, por lo cual, durante la ejecución del algoritmo no es usualmente el candidato a variable menos crítica, para hacer la partición del algoritmo *Distribution-based Mondrian*. Esta situación posibilita que este atributo sea el que menores cambios sufra y por tanto su distribución univariada se logra preservar a pesar de que el valor *k* crezca.



(a) anho_nacimiento *(b) anhos_v* *(c) num_hijos*

Figura 25. Distribución de frecuencias de los cuasi-identificadores de los datos CAOBA.

La preservación de la distribución de un atributo siempre estará supeditada a las características del conjunto de datos: cantidad y distribución de los cuasi-identificadores, tamaño de la base y el valor *k* del principio de anonimización. Esto significa que cada caso es particular y por tanto debe analizarse con detenimiento para tomar las decisiones que mejor se ajusten a la hora de parametrizar la información en *Anonylitics*. El sistema entrega un resumen de ejecución con lo cual el usuario final debe definir qué escenario es el que representa las mejores ganancias en utilidad y privacidad para los posteriores ejercicios de analítica.

6.2. Segundo escenario

En el segundo escenario el objetivo es evaluar la pertinencia del algoritmo propuesto, *Rothko-D* para situaciones en que se necesite anonimizar un alto volumen de datos de forma centralizada, es decir en una sola máquina. En esta vía, se busca evidenciar el problema expuesto por LeFevre *et al.* [7] sobre el desbordamiento de la memoria *RAM* en una máquina, es decir en *commodity hardware*. De acuerdo con ello, se plantea cómo se podría abordar la situación utilizando el algoritmo *Rothko-D*.

Con el propósito de evaluar una situación donde se pudieran llegar a desbordar los datos, se realizó una simulación basada en el conjunto de datos CAOBA, el cual contiene datos reales. La experimentación consistió en duplicar, triplicar, etc., el conjunto de datos original, para evaluar en qué punto podría desbordarse la memoria de la máquina y con ello poner a prueba el algoritmo *Rothko-D*. El experimento fue realizado en una máquina con 8 GB de memoria *RAM*, un procesador Intel Core i7-6500U y un sistema operativo Windows 10 Enterprise, el cual tiene una memoria disponible de 4GB, tomando como referencia el uso promedio que hace el sistema operativo sobre la *RAM* [81]. Para poder generar un escenario controlado y evaluar la situación de la memoria, se controlaron las variables intervinientes del experimento puesto que interfieren con los resultados de las pruebas realizadas. Por ello, se deshabilitó el uso de la *swap*, ya que esta técnica se utiliza cuando la RAM está a punto de agotarse y el sistema operativo reserva un espacio en disco de aproximadamente 1.5 veces el tamaño físico de la memoria *RAM* para continuar ejecutando los procesos en curso [82]. De igual forma se deshabilitaron otros procesos adicionales que no fueran parte de la carga propia del sistema operativo, dejando únicamente en ejecución la anonimización de los datos.

En primera instancia se hizo la ejecución del algoritmo *Distribution-based Mondrian* con un *k=2* y los 3 cuasi-identificadores señalados en el primer escenario de validación, pero en esta

ocasión, incrementando sucesivamente la cantidad de registros y por ende el tamaño del conjunto de datos, para evaluar hasta qué punto se podría llegar a desbordar la memoria *RAM*. En la Figura 26 se observa cómo se incrementa el uso de la *RAM* en función del tamaño de los datos a anonimizar (color verde). Adicionalmente se muestra el cálculo, a través de la fórmula (7) de estimación del tamaño máximo que se debería mantener en memoria (color amarillo), de acuerdo con la sección *4.3. Anonimización centralizada de altos volúmenes de datos*. Se evidencia que, teóricamente, no podría mantenerse en memoria un conjunto de datos *CAOBA* con un tamaño de 0.16 GB o superior. Sin embargo, en realidad fue posible anonimizar hasta un conjunto de 0.2 GB, debido a que el cálculo propuesto realiza la estimación basándose en peor de los casos y por ello restringiría la ejecución del algoritmo *Distribution-based Mondrian* para nunca permitir el desbordamiento de la memoria. Por tanto, se concluye que la propuesta de estimación del tamaño de memoria requerido es viable y útil pues permite proponer una alternativa para que el proceso siempre logre terminarse completamente y pueda generar los resultados esperados por el usuario final.

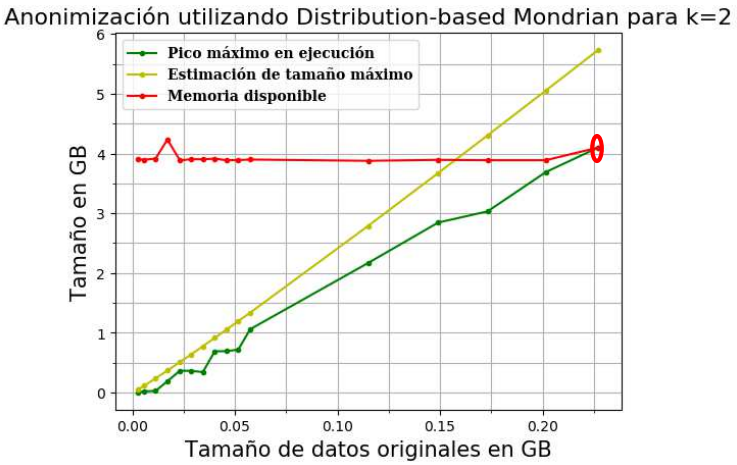


Figura 26. Uso teórico y real de la memoria RAM al anonimizar un conjunto de datos, utilizando el algoritmo Distribution-based Mondrian.

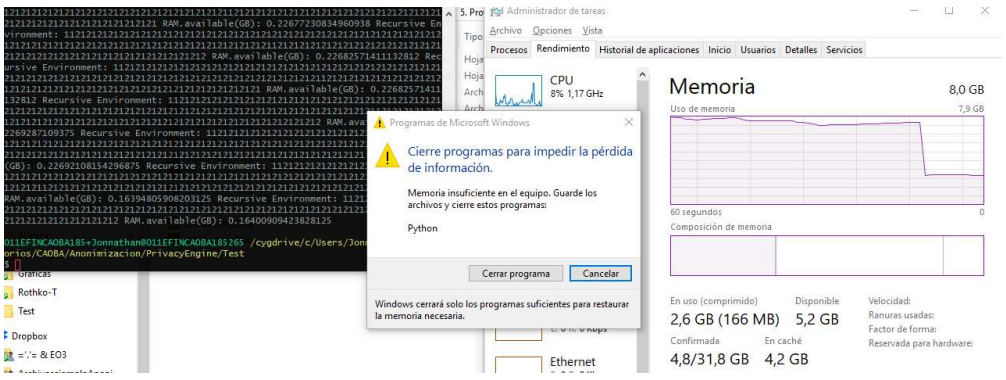


Figura 27. Mensaje de error de la ejecución de anonimización por desbordamiento de la RAM.

En la Figura 27 se evidencia el mensaje de error obtenido en el momento en que ocurrió el desbordamiento de memoria, es decir, al intentar anonimizar un conjunto de datos de 0.22 GB, donde se observa en la Figura 26 que se cruzan la memoria disponible (color rojo) y el pico máximo en ejecución (color verde).

Entendiendo que efectivamente el algoritmo que trabaja en memoria, *Distribution-based Mondrian*, genera un desbordamiento al incrementarse el conjunto de datos, posteriormente se realizaron las mismas pruebas propuestas en la Figura 26, pero empleando el algoritmo *Rothko-D*, el cual fue propuesto para soportar conjuntos con un alto volumen de datos. En la Figura 28 se aprecia el pico máximo de consumo de memoria de *Rothko-D* (color azul), en comparación con el pico máximo del algoritmo *Distribution-based Mondrian* (color anaranjado). Allí se evidencia que a partir de un conjunto de datos de 0.16 G, el algoritmo *Rothko-D* genera un pico máximo de consumo de 3.4 GB de memoria, lo cual es menor al espacio disponible (color rojo), de aproximadamente 4.2 GB. Esta situación se genera debido a la estimación y los pasos que realiza el algoritmo para trabajar directamente en disco algunos cálculos hasta que la partición de los datos pueda mantenerse en la *RAM*, sin poner en riesgo el curso de la ejecución de la anonimización.

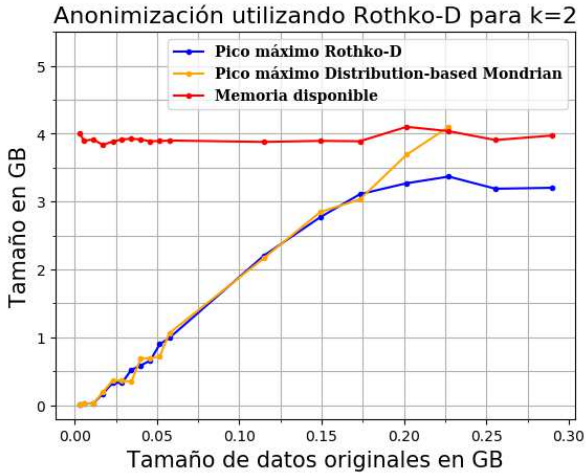


Figura 28. Uso de la memoria RAM al anonimizar un conjunto de datos, utilizando el algoritmo Rothko-D.

De acuerdo con la fórmula de estimación del tamaño máximo que se podría mantener en memoria (7), en la Tabla 10 pueden observarse algunos valores que representan el punto a partir del cual se requeriría el uso del algoritmo *Rothko-D* para solventar el posible desbordamiento de la *RAM*, despendiendo del sistema operativo (los más comunes), capacidad de memoria de la máquina y el tamaño original de datos a anonimizar. Es importante resaltar que, para cada sistema operativo, la memoria disponible para realizar procesamientos es diferente, debido a los procesos que se ejecutan en *background.* Así, para un sistema operativo Windows con 4 GB de *RAM*, aproximadamente se empezará a anonimizar con el algoritmo *Rothko-D*, un conjunto de datos original de 8.000.000 registros con un peso de 0,046 GB.

Tabla 10. Valores estimados a partir de los cuales Anonylitics utilizaría el algoritmo Rothko-D para anonimizar los datos sin desbordar la memoria.

	RAM (GB) de la máquina	4	8	12	16
Windows	Memoria usada (GB)	2,9	2,9	2,9	2,9
	Memoria disponible (GB)	1,1	5,1	9,1	13,1
	No. Registros	8.000.000	35.000.000	60.000.000	86.000.000
	Peso datos (GB)	0,0460	0,2014	0,3453	0,4950
	Estimación (GB) k=2	1,056	5,048	8,922	13,046
Linux Ubuntu	Memoria usada (GB)	1,9	1,9	1,9	1,9
	Memoria disponible (GB)	2,1	6,1	10,1	14,1
	No. Registros	15.000.000	41.000.000	65.600.000	92.000.000
	Peso datos (GB)	0,0863	0,2360	0,3775	0,5295
	Estimación (GB) k=2	2,058	5,967	9,804	14,008

Aunque desde el punto de vista de uso de la memoria, *Rothko-D* es una aproximación adecuada, para manejar grandes volúmenes de datos de forma centralizada, es importante evidenciar las repercusiones que ello tiene en los tiempos de ejecución. En la Figura 29 se pueden apreciar los tiempos en función de los mismos tamaños de los conjuntos de datos originales, evidenciando que el tiempo de *Rothko-D* empieza a aumentar ligeramente después de 0.1 GB respecto al algoritmo *Distribution-based Mondrian*, pero de forma más ágil después de 0.2 GB donde debe hacer más lecturas desde el disco hasta lograr encontrar particiones de datos que no desborden la memoria *RAM*. Esto se debe precisamente a la latencia introducida por tener que acceder al disco duro, lo cual toma mayor tiempo que sólo trabajar con la memoria disponible.

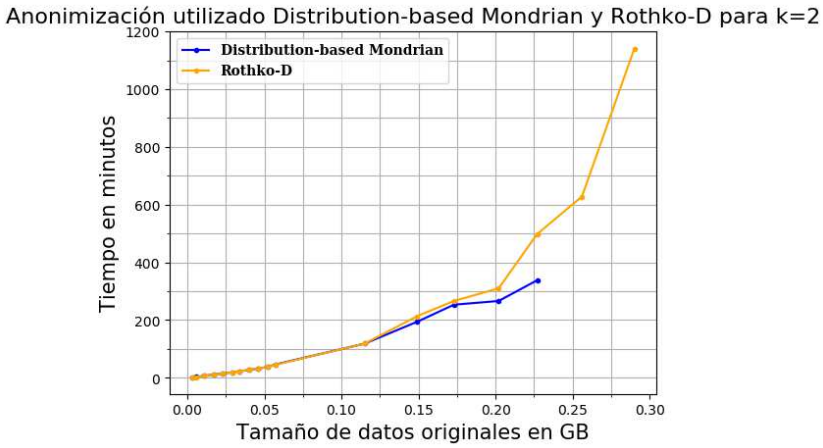


Figura 29. Tiempos de ejecución utilizando los algoritmos Distribution-based Mondrian y Rothko-D.

De esta manera se evidencia que la propuesta realizada en el presente trabajo es propicia para trabajar en volúmenes altos de datos para una anonimización centralizada, es decir en una sola máquina. En caso es que se tuviera un escenario con datos distribuidos o donde se tuvieran requerimientos de tiempo limitado, sería necesario emplear una técnica similar, pero aprovechando las capacidades del cómputo distribuido para mejorar los tiempos de respuesta al dividir parcialmente el trabajo en varios nodos de ejecución.

7. CONCLUSIONES Y TRABAJOS FUTUROS

El trabajo desarrollado permitió aportar a la base de conocimiento relacionada con la anonimización de datos estructurados puesto que, de acuerdo con la investigación sobre el estado del arte en torno a esta temática, no se encontró evidencia de que alguno de los algoritmos propuestos a la fecha, estuviesen orientados a promover la preservación de la distribución univariada de los cuasi-identificadores. Generalmente, las investigaciones se centran en otras medidas como el conteo del número de datos modificados, mediante la métrica de *pérdida generalizada de información* [6], una de las más comunes en la literatura. Partiendo de esta investigación, se propuso un algoritmo de anonimización enfocado en la preservación de la utilidad y la privacidad de los datos, denominado *Distribution-based Mondrian*, el cual analiza la función de distribución de probabilidad de los valores de los cuasi-identificadores numéricos. Adicionalmente, evaluando un escenario de anonimización centralizada de altos volúmenes de datos, se propuso el algoritmo denominado *Rothko-D*, que se basa en los principios del primer algoritmo para seguir proveyendo utilidad y privacidad sin permitir el desbordamiento de la memoria *RAM*. Ambas propuestas fueron validadas a la luz de los escenarios planteados para evaluar su pertinencia en términos de aseguramiento de la privacidad y preservación de la utilidad.

El hecho de abordar la utilidad de los datos numéricos desde el análisis de su distribución univariada es una propuesta planteada en el presente trabajo, como respuesta a los requerimientos de la Alianza CAOBA, lo cual justamente permite evaluar la preservación de una propiedad estadística de los datos. Esto no implica que necesariamente se van a obtener los mismos resultados en un modelo analítico con los datos antes y después de anonimizar, pues es una hipótesis que se debe validar con conjuntos de datos reales, que tengan como propósito un ejercicio de analítica. Debido a que el sistema se diseñó para anonimizar datos sin conocer de antemano el ejercicio que se quiere hacer, es válido trabajar con el análisis de las propiedades estadísticas de los datos, lo cual da cuenta de que no se están alterando los datos en su estructura y que ello puede contribuir a que se sigan obteniendo resultados útiles para la toma de decisiones. De acuerdo con lo anterior, se entiende que existen diferentes métricas para medir la utilidad de los datos después de anonimizar además del análisis de la distribución univariada, como las evidenciadas a través del estado del arte: *Indistinguibilidad*, *Pérdida generalizada de información* y *Tamaño promedio de la clase de equivalencia* [6], entre otras. Dado que éstas podrían no ser suficientes para garantizar la preservación de las propiedades estadísticas de los datos numéricos, por ello en el presente trabajo se hace una nueva propuesta, la cual fue validada.

La propuesta permite fortalecer la fase de anonimización requerida durante el pre-procesamiento de los proyectos de analítica de datos pues, al mismo tiempo que se genera privacidad, se tiene en cuenta la utilidad para no entorpecer las posibles relaciones y patrones que caractericen los datos. Los proyectos de analítica suelen estar enmarcados en alguna metodología que permita realizar, desde la obtención de los datos, hasta la entrega de resultados y su validación. Bajo este contexto existen diferentes metodologías, entre ellas CRISP-DM [9], KDD [10] y SEMMA [11]. Cada una tiene como propósito analizar datos con diferentes objetivos de negocio, para generar resultados que motiven nuevas decisiones estratégicas que redunden en beneficios para una compañía. Cada una de ellas cuenta con una etapa de pre-procesamiento de los datos, donde se necesita no sólo detectar atípicos, imputar valores faltantes, normalizar los datos, etc., sino que también es indispensable llevar a cabo un proceso de anonimización que

permita que las empresas puedan entregar sus datos con tranquilidad a terceros, respetando las normatividades vigentes relacionadas con el manejo adecuado de la información de individuos.

De acuerdo con las propuestas realizadas, se diseñó e implementó *Anonylitics*, un sistema de anonimización de datos estructurados, el cual comprende un subconjunto de requerimientos planteados por la Alianza CAOBA que están orientados a dar soporte a la propuesta de anonimización orientada a la analítica de datos. Para poder administrar, almacenar y reutilizar las parametrizaciones que apoyan el proceso de transformación, se diseñó un lenguaje denominado *Proyecto de Anonimización*, que permite tener una estructura común y flexible para la administración y almacenamiento de los metadatos asociados. Las empresas ancla de CAOBA y muchas otras, pueden beneficiarse de este sistema, logrando que los ejercicios de analítica puedan ser llevados a cabo sin tantas dificultades impuestas por los riesgos de divulgación de la información. Es indispensable tener en cuenta que los sistemas no suelen funcionar como cajas negras y por ello el usuario de *Anonylitics* tiene la responsabilidad de evaluar los resultados obtenidos para determinar si la aproximación generada, usando una determinada parametrización (dada por el usuario), es la que más se ajusta a sus necesidades.

Es importante aclarar que el funcionamiento de *Anonylitics* está limitado a datos numéricos. En este sentido, no podría utilizarse para cualquier conjunto de datos, pues aquellos que contengan datos categóricos, exceden las funcionalidades del sistema. No obstante, desde que se seleccionen sólo atributos numéricos, *Anonylitics*, está en la capacidad de generar resultados que demuestren el mejor balance de anonimización, con lo cual el usuario final podrá tomar decisiones sobre sus necesidades de privacidad y utilidad. Cabe resaltar que *Anonylitics* es una propuesta desde el punto de vista técnico que busca salvaguardar la privacidad de los datos, no obstante, no representa una solución legal.

Todo lo expuesto posibilitó el cumplimiento del objetivo trazado para el proyecto; la construcción del sistema de anonimización orientado a preparar datos para posteriores análisis. De ello se desprenden diferentes trabajos futuros para CAOBA, en pro de seguir complementando *Anonylitics* para robustecerlo, de manera que pueda apoyar otros requerimientos propuestos:

- Integrar algoritmos de anonimización ligera, con lo cual el usuario final pueda decidir si prefiere realizar transformaciones sencillas o si desea ejecutar algoritmos más robustos que evalúen diferentes aspectos de los datos, en base a su intención al momento de anonimizar.
- Desarrollar propuestas de anonimización de datos categóricos que puedan integrarse con las aproximaciones planteadas para numéricos. En este mismo sentido, se deberán desarrollar mecanismos de evaluación acordes al tipo de dato para evidenciar que la privacidad y utilidad están siendo preservadas.
- Extender los algoritmos de anonimización para trabajar en contextos de datos que estén distribuidos o donde se requieren tiempos de respuesta más limitados y que por ende impongan nuevos desafíos en relación a poder paralelizar y distribuir las operaciones necesarias para transformar los datos.
- Implementar nuevas funcionalidades útiles para los usuarios finales, como la reutilización de proyectos de anonimización y los menús de ayuda, lo cual permitiría mejorar la interacción con el sistema y asegurar su éxito en los proyectos de analítica de datos.

8. BIBLIOGRAFÍA

[1] «Sobre el Habeas Data Financiero | Superintendencia de Industria y Comercio». [En línea]. Disponible en: http://www.sic.gov.co/sobre-el-habeas-data-financiero. [Accedido: 11-nov-2017].

[2] Congreso de la República de Colombia, «Ley estatutaria 1581 de 2012: Por la cual se dictan disposiciones generales para la protección de datos personales». 12-oct-2012.

[3] P. Ashley, S. Hada, G. Karjoth, C. Powers, y M. Schunter, «The Enterprise Privacy Authorization Language (EPAL)». [En línea]. Disponible en: https://www.w3.org/2003/p3p-ws/pp/ibm3.html. [Accedido: 15-abr-2017].

[4] H. Nissenbaum, «The Meaning of Anonymity in an Information Age», *Inf. Soc.*, vol. 15, n.º 2, pp. 141-144, may 1999.

[5] L. Sweeney, «Simple Demographics Often Identify People Uniquely», *Health (N. Y.)*, vol. 671, ene. 2000.

[6] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, y L. Murphy, «A Systematic comparison and evaluation of k-Anonymization algorithms for practitioners», *Trans. Data Priv.*, vol. 7, n.º 3, pp. 337-370, 2014.

[7] K. LeFevre, D. J. DeWitt, y R. Ramakrishnan, «Mondrian Multidimensional K-Anonymity», en *22nd International Conference on Data Engineering (ICDE'06)*, 2006, pp. 25-25.

[8] K. LeFevre, D. J. DeWitt, y R. Ramakrishnan, «Workload-aware anonymization techniques for large-scale datasets», *ACM Trans Database Syst*, vol. 33, n.º 3, pp. 1-47, 2008.

[9] *CRISP-DM 1.0: Step-by-step Data Mining Guide*. SPSS, 2000.

[10] U. M. Fayyad, G. Piatetsky-Shapiro, y P. Smyth, «From Data Mining to Knowledge Discovery in Databases», *AI Mag.*, vol. 17, pp. 37-54, mar. 1996.

[11] SAS Institute, «SAS® Help Center: Data Mining and SEMMA». [En línea]. Disponible en: http://documentation.sas.com/?docsetId=emcs&docsetTarget=n0pejm83csbja4n1xueveo2uoujy.htm&docsetVersion=12.3&locale=en. [Accedido: 17-oct-2017].

[12] D. T. Larose y C. D. Larose, *Data Mining and Predictive Analytics*. John Wiley & Sons, 2015.

[13] C. C. Aggarwal y P. S. Yu, «A General Survey of Privacy-Preserving Data Mining Models and Algorithms», en *Privacy-Preserving Data Mining*, C. C. Aggarwal y P. S. Yu, Eds. Springer US, 2008, pp. 11-52.

[14] C. C. Aggarwal, *Social Network Data Analytics*. Springer Science & Business Media, 2011.

[15] B. Raghunathan, *The Complete Book of Data Anonymization: From Planning to Implementation*. CRC Press, 2013.

[16] X. Yang, X. Liu, B. Wang, y G. Yu, «A K-Anonymizing Approach for Preventing Link Attacks in Data Publishing», en *Parallel and Distributed Processing and Applications - ISPA 2005 Workshops*, 2005, pp. 627-636.

[17] L. Sweeney, «k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY», *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, n.º 05, pp. 557-570, oct. 2002.

[18] X. Huang, J. Liu, Z. Han, y J. Yang, «A new anonymity model for privacy-preserving data publishing», *China Commun.*, vol. 11, n.º 9, pp. 47-59, sep. 2014.

[19] M. M. Mano, *Arquitectura de computadoras*. Pearson Educación, 1994.

[20] F. J. Q. Flor y A. G. del Solo, *Computadores paralelos y evaluación de prestaciones*. Univ de Castilla La Mancha, 1996.

[21] G. Wang, I. Ray, J. M. A. Calero, y S. M. Thampi, *Security, Privacy and Anonymity in Computation, Communication and Storage: SpaCCS 2016 International Workshops, TrustData, TSP, NOPE, DependSys, BigDataSPT, and WCSSC, Zhangjiajie, China, November 16-18, 2016, Proceedings*. Springer, 2016.

[22] DANE, «Catálogo Central de Datos». [En línea]. Disponible en: https://formularios.dane.gov.co/Anda_4_1/index.php/catalog/. [Accedido: 01-nov-2017].

[23] «Datos Abiertos Colombia | Datos Abiertos Colombia», *la plataforma de datos abiertos del gobierno colombiano*. [En línea]. Disponible en: https://www.datos.gov.co/. [Accedido: 24-oct-2017].

[24] T. Gartner, *Kernels for Structured Data*. World Scientific, 2008.

[25] A. R. Hevner, S. T. March, J. Park, y S. Ram, «Design Science in Information Systems Research», *MIS Q*, vol. 28, n.º 1, pp. 75–105, mar. 2004.

[26] M. Trigas y A. Domingo, «TFS - Gestión de Proyectos Informáticos - SCRUM». 24-oct-2017.

[27] «What is Scrum?», *Scrum.org*. [En línea]. Disponible en: http://www.scrum.org/resources/what-is-scrum. [Accedido: 25-oct-2017].

[28] Atlassian, «Jira | Issue & Project Tracking Software | Atlassian». [En línea]. Disponible en: https://www.atlassian.com/software/jira. [Accedido: 25-oct-2017].

[29] S. D. Warren y L. D. Brandeis, «The Right to Privacy», *Harv. Law Rev.*, vol. 4, n.º 5, pp. 193-220, 1890.

[30] ITU, *Security in Telecommunications and Information Technology 2006*. Paris: Organisation for Economic Co-operation and Development, 2006.

[31] D. Lambert, «Measures of Disclosure Risk and Harm», *J. Off. Stat.*, vol. 9, n.º 2, pp. 313-331, 1993.

[32] A. J. Menezes, P. C. van Oorschot, y S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[33] V. Ciriani, S. D. C. di Vimercati, S. Foresti, y P. Samarati, «Microdata Protection», en *Secure Data Management in Decentralized Systems*, Springer, Boston, MA, 2007, pp. 291-321.

[34] M. Templ, B. Meindl, A. Kowarik, y S. Chen, «Introduction to Statistical Disclosure Control (SDC)». IHSN Working Paper No 007, 2014.

[35] J. S. Davis y O. A. Osoba, «Privacy Preservation in the Age of Big Data», *SSRN Electron. J.*, ene. 2016.

[36] B. C. M. Fung, K. Wang, R. Chen, y P. S. Yu, «Privacy-preserving data publishing: A survey of recent developments», *ACM Comput Surv*, vol. 42, n.º 4, pp. 1-53, 2010.

[37] P. Samarati y L. Sweeney, «Generalizing data to provide anonymity when disclosing information», *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Princ. Database Syst.*, vol. 98, ene. 1998.

[38] P. Samarati, «Protecting respondents identities in microdata release», *IEEE Trans. Knowl. Data Eng.*, vol. 13, n.º 6, pp. 1010-1027, nov. 2001.

[39] A. Machanavajjhala, D. Kifer, J. Gehrke, y M. Venkitasubramaniam, «L-diversity: Privacy Beyond K-anonymity», *ACM Trans Knowl Discov Data*, vol. 1, n.º 1, mar. 2007.

[40] N. Li, T. Li, y S. Venkatasubramanian, «t-Closeness: Privacy Beyond k-Anonymity and l-Diversity», en *2007 IEEE 23rd International Conference on Data Engineering*, 2007, pp. 106-115.

[41] L. Sweeney, «Datafly: a system for providing anonymity in medical data», en *Database Security XI*, Springer, Boston, MA, 1998, pp. 356-381.

[42] K. LeFevre, D. J. DeWitt, y R. Ramakrishnan, «Incognito: Efficient Full-domain K-anonymity», en *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2005, pp. 49–60.

[43] L. Sweeney, «Achieving k-anonymity privacy protection using generalization and suppression», *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, n.º 05, pp. 571-588, oct. 2002.

[44] B. C. M. Fung, K. Wang, y P. S. Yu, «Top-down specialization for information and privacy preservation», en *21st International Conference on Data Engineering (ICDE'05)*, 2005, pp. 205-216.

[45] R. J. Bayardo y R. Agrawal, «Data privacy through optimal k-anonymization», en *21st International Conference on Data Engineering (ICDE'05)*, 2005, pp. 217-228.

[46] L. Xu, C. Jiang, J. Wang, J. Yuan, y Y. Ren, «Information Security in Big Data: Privacy and Data Mining», *Ieee Access*, vol. 2, pp. 1149-1176, 2014.

[47] F. J. Ohlhorst, *Big Data Analytics: Turning Big Data into Big Money*. John Wiley & Sons, 2012.

[48] A. Monreale, S. Rinzivillo, F. Pratesi, F. Giannotti, y D. Pedreschi, «Privacy-by-design in big data analytics and social mining», *EPJ Data Sci.*, vol. 3, n.º 1, pp. 1-26, 2014.

[49] A. Yassine, A. A. N. Shirehjini, y S. Shirmohammadi, «Smart meters big data: Game theoretic model for fair data sharing in deregulated smart grids», *IEEE Access*, vol. 3, pp. 2743-2754, 2015.

[50] D.-E. Cho, S. J. Kim, y S.-S. Yeo, «Double privacy layer architecture for big data framework», *Int. J. Softw. Eng. Its Appl.*, vol. 10, n.º 2, pp. 271-278, 2016.

[51] X. Zhang, C. Liu, S. Nepal, C. Yang, W. Dou, y J. Chen, «SaC-FRAPP: a scalable and cost-effective framework for privacy preservation over big data on cloud», *Concurr. Comput.-Pract. Exp.*, vol. 25, n.º 18, pp. 2561-2576, dic. 2013.

[52] OpenStack, «OpenStack Open Source Cloud Computing Software», *OpenStack*. [En línea]. Disponible en: https://www.openstack.org/. [Accedido: 01-nov-2017].

[53] S. Achari, *Hadoop Essentials*. Packt Publishing Ltd, 2015.

[54] D. Pàmies-Estrems, J. Castellà-Roca, y A. Viejo, «Working at the web search engine side to generate privacy-preserving user profiles», *Expert Syst. Appl.*, vol. 64, pp. 523-535, 2016.

[55] K. Ito, J. Kogure, T. Shimoyama, y H. Tsuda, «De-identification and Encryption Technologies to Protect Personal Information», *Fujitsu Sci. Tech. J.*, vol. 52, n.º 3, pp. 28-36, jul. 2016.

[56] X. Zhang, L. T. Yang, C. Liu, y J. Chen, «A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud», *Ieee Trans. Parallel Distrib. Syst.*, vol. 25, n.º 2, pp. 363-373, feb. 2014.

[57] X. Zhang, C. Liu, S. Nepal, C. Yang, W. Dou, y J. Chen, «A hybrid approach for scalable sub-tree anonymization over big data using Map Reduce on cloud», *J. Comput. Syst. Sci.*, vol. 80, n.º 5, pp. 1008-1020, ago. 2014.

[58] X. Zhang *et al.*, «Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud», *Ieee Trans. Comput.*, vol. 64, n.º 8, pp. 2293-2307, ago. 2015.

[59] Y. Sowmya y R. Nagaratna, «Parallelizing K-anonymity algorithm for privacy preserving knowledge discovery from big data», *Int. J. Appl. Eng. Res.*, vol. 11, n.º 2, pp. 1314-1321, 2016.

[60] X. Zhang, L. Qi, Q. He, y W. Dou, «Scalable Iterative Implementation of Mondrian for Big Data Multidimensional Anonymisation», en *Security, Privacy and Anonymity in Computation, Communication and Storage*, 2016, pp. 311-320.

[61] X. Zhang, C. Leckie, W. Dou, J. Chen, R. Kotagiri, y Z. Salcic, «Scalable Local-Recoding Anonymization using Locality Sensitive Hashing for Big Data Privacy Preservation», en *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, Indiana, USA, 2016, pp. 1793-1802.

[62] J. A. Alvarado Valencia y J. J. Obagi Araújo, *Fundamentos de inferencia estadística*. Pontificia Universidad Javeriana, 2008.

[63] R. E. Walpole, R. H. Myers, y S. L. Myers, *Probabilidad y estadística para ingenieros*. Pearson Educación, 1999.

[64] J. D. Gibbons y S. Chakraborti, *Nonparametric Statistical Inference*. CRC Press, 2003.

[65] T. H. Cormen, C. E. Leiserson, R. L. Rivest, y C. Stein, *Introduction To Algorithms*. MIT Press, 2001.

[66] «Discrete Structures: Trees (3.2 Properties of special k- ary trees: Binary Trees (i.e. k =2))». [En línea]. Disponible en: http://vle.du.ac.in/mod/book/view.php?id=5857&chapterid=4134. [Accedido: 03-nov-2017].

[67] M. Löfberg, P. Molin, L. Lundberg, y T. Jönsson, *Web vs. Standalone Application*. Jun, 2005.

[68] C. T. I. Reviews, *Web Services , Principles and Technology*. Cram101 Textbook Reviews, 2016.

[69] P. J. Sadalage y M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2012.

[70] «IBM Knowledge Center - General information about ODBC, JDBC, and OLE DB». [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/en/SSULQD_7.1.0/com.ibm.nz.datacon.doc/c_datacon_introduction.html. [Accedido: 09-nov-2017].

[71] «Architectural Styles and the Design of Network-based Software Architectures». [En línea]. Disponible en: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm. [Accedido: 27-oct-2017].

[72] R. van den Broek, «Comparing the performance of SOAP and REST PHP clients», en *14th Twente Student Conference on IT, Enschede, Netherlands*, 2011.

[73] B. Green y S. Seshadri, *AngularJS*. O'Reilly Media, Inc., 2013.

[74] J. Spurlock, *Bootstrap: Responsive Web Development*. O'Reilly Media, Inc., 2013.

[75] M. Waikar, *Data-oriented Development with AngularJS*. Packt Publishing Ltd, 2015.

[76] Y. Hilpisch, *Derivatives Analytics with Python: Data Analysis, Models, Simulation, Calibration and Hedging*. John Wiley & Sons, 2015.

[77] K. Chodorow, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. O'Reilly Media, Inc., 2013.

[78] M. Stucky, *MySQL: Building User Interfaces*. Sams Publishing, 2001.

[79] «UCI Machine Learning Repository». [En línea]. Disponible en: https://archive.ics.uci.edu/ml/index.php. [Accedido: 02-nov-2017].

[80] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, y A. W.-C. Fu, «Utility-based anonymization using local recoding», en *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, Philadelphia, PA, USA, 2006, pp. 785-790.

[81] Microsoft, «Pushing the Limits of Windows: Physical Memory – Mark's Blog». [En línea]. Disponible en: https://blogs.technet.microsoft.com/markrussinovich/2008/07/21/pushing-the-limits-of-windows-physical-memory/. [Accedido: 08-nov-2017].

[82] Microsoft, «RAM, memoria virtual, archivo de paginación y administración de memoria en Windows». [En línea]. Disponible en: https://support.microsoft.com/es-co/help/2160852/ram--virtual-memory--pagefile--and-memory-management-in-windows. [Accedido: 08-nov-2017].

9. ANEXOS

9.1. Requerimientos implementados por *Sprint*

En la Tabla 11, Tabla 12 y Tabla 13 se encuentran los identificadores de los requerimientos implementados por cada uno de los tres *sprints* durante la construcción del sistema. Antes de iniciar cada *sprint*, se hizo la correspondiente priorización, la cual se elaboró junto al cliente y el *Product owner*.

Tabla 11. Requerimientos implementados en el primer sprint.

REQ-001	REQ-003	REQ-004	REQ-008	REQ-013	REQ-014

Tabla 12. Requerimientos implementados en el segundo sprint.

REQ-015	REQ-016	REQ-017	REQ-018	REQ-020	REQ-022	REQ-023	REQ-024	REQ-027

Tabla 13. Requerimientos implementados en el tercer sprint.

REQ-032	REQ-034	REQ-042	REQ-044	REQ-046	REQ-051