ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

IMPLEMENTACIÓN DE MECANISMOS DE CONTROL DE ACCESO A INFORMACIÓN DE IDENTIFICACIÓN PERSONAL (PII) DE ACUERDO CON LA LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES DEL ECUADOR

ALMACENAMIENTO, PRESERVACIÓN, ACCESO SEGURO Y TRANSPARENTE A PII

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

CRISTHIAN FERNANDO TOHASA PASPUEZAN
cristhian.tohasa@epn.edu.ec

DIRECTOR: DENYS ALBERTO FLORES ARMAS
denys.flores@epn.edu.ec

DMQ, marzo 2023

CERTIFICACIONES

Yo, CRISTHIAN FERNANDO TOHASA PASPUEZAN declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Firmado electrónicamente por:
CRISTHIAN FERNANDO
TOHASA PASPUEZAN

CRISTHIAN FERNANDO TOHASA PASPUEZAN

Certifico que el presente trabajo de integración curricular fue desarrollado por CRISTHIAN FERNANDO TOHASA PASPUEZAN, bajo mi supervisión.

Firmado electrónicamente por:
DENYS ALBERTO
FLORES ARMAS

DENYS ALBERTO FLORES ARMAS
DIRECTOR

Certificamos que revisamos el presente trabajo de integración curricular.

NOMBRE_REVISOR1	**NOMBRE_REVISOR2**
REVISOR1 DEL TRABAJO DE INTEGRACIÓN CURRICULAR	**REVISOR2 DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Adicionalmente, declaro que el contenido de este trabajo incluye material de artículos técnicos revisados por pares de mi autoría que han sido publicados durante el desarrollo de este proyecto, y que detallo a continuación:

CRISTHIAN FERNANDO TOHASA PASPUEZAN

DENYS ALBERTO FLORES ARMAS

DEDICATORIA

Quiero dedicar mi tesis a Dios, quien ha sido mi fortaleza y guía en todo momento. Gracias a Él por su amor, su gracia y por nunca abandonarme en los momentos más difíciles. Gracias por iluminar mi camino, por darme la fuerza y el ánimo necesario para continuar en este viaje.

A mi familia, quienes han sido mi apoyo incondicional durante todo este proceso. Gracias por sus palabras de aliento, su paciencia y su increíble comprensión en las situaciones en que necesitaba tiempo y espacio para enfocarme en mi trabajo. Gracias por creer en mí y por siempre estar a mi lado en cada paso que he dado.

A mis amigos, quienes han estado conmigo en las buenas y en las malas. Gracias por ser mi escape en los momentos de estrés, por sus risas y su compañía, y por nunca dejarme sentir solo. Gracias por comprender mi ausencia en ocasiones y por animarme a seguir adelante cuando las cosas se ponían difíciles.

A mis profesores y mentores, quienes me han guiado en el camino de la formación académica y profesional. Gracias por su conocimiento, su paciencia y por compartir conmigo sus experiencias. Gracias por haberme enseñado a mirar más allá de lo evidente y por haberme inspirado a alcanzar mis objetivos.

Por último, a todas aquellas personas que han estado a mi lado en este proceso, gracias por su apoyo incondicional, por sus frases de aliento y por creer en mí. Sin ustedes, esto no habría sido posible.

AGRADECIMIENTO

Queridos lectores, es un placer para mí dedicar un espacio en este proyecto para poder expresar mi profundo agradecimiento quienes me brindaron su apoyo y ayuda en este camino de aprendizaje.

Primeramente, quiero dar gracias a Dios por ayudarme a culminar este proyecto, por darme las fuerzas y el coraje para superar los obstáculos y por ser mi guía y protector en todo momento.

A mi familia, les agradezco por su amor incondicional, paciencia y comprensión durante este tiempo de estudio. Gracias por ser mi pilar y apoyarme en todo momento, por brindarme un hogar donde siempre me sentí seguro y cómodo para estudiar. Les agradezco por su ejemplo de dedicación, esfuerzo y perseverancia, que siempre me inspiraron para seguir adelante.

A mis amigos, les agradezco por su compañía, su ánimo y su apoyo incondicional. Gracias por haber estado ahí para mí, por haberme brindado su ayuda en momentos de necesidad y por ser parte de mi vida. También agradezco a mi profesor Denys Flores y a mi profesora Jenny Torres por su valioso apoyo, enseñanzas y orientación en la realización de esta tesis. Sus conocimientos, experiencia y dedicación fueron fundamentales en este proyecto.

Finalmente, quiero expresarles mis agradecimientos a todas las personas que estuvieron conmigo, que me brindaron su amistad, su cariño y su confianza en todo momento. Les agradezco de corazón por ser parte de mi vida y por compartir conmigo este camino de aprendizaje.

ÍNDICE DE CONTENIDO

RESUMEN

El almacenamiento, preservación, acceso seguro y transparente a la información personal identificable (PII) es un tema que con el paso del tiempo se ha ido tomando más en cuenta en el campo de la seguridad informática. En este sentido, este proyecto está enfocado en desarrollar un componente de almacenamiento seguro y distribuido para la gestión de la PII.

El componente propuesto utiliza una arquitectura distribuida y fragmentada para de esta manera garantizar la triada CIA (confidencialidad, integridad y disponibilidad) de la información almacenada. Se implementan controles de seguridad para poder acceder a la información una de estas es la verificación de firmas digitales las cuales mantienen la integridad de los datos. Además, se utiliza un proceso de replicación para mantener así la disponibilidad a través de la sincronización de los datos en todo momento.

La herramienta de modelado de amenazas se utilizó para identificar y mitigar los riesgos y amenazas potenciales que pueden surgir durante el almacenamiento de datos sensibles. Asimismo, se realizó un análisis de riesgos para identificar y planificar la gestión de contingencias en caso de que se produzcan amenazas.

Finalmente, se desarrolló una aplicación de prueba que integra el componente de almacenamiento con otros dos componentes, la recolección y transferencia de datos, para demostrar la viabilidad y la efectividad del componente de almacenamiento seguro y distribuido.

En resumen, el presente trabajo proporciona un enfoque práctico y efectivo para el almacenamiento, preservación, acceso seguro y transparente a la información personal identificable. La arquitectura distribuida y fragmentada del componente de almacenamiento asegura la triada CIA de los datos almacenados, lo que garantiza la protección de los datos sensibles.

PALABRAS CLAVE: almacenamiento, preservación, acceso seguro, acceso transparente, información de identificación personal, arquitectura distribuida.

ABSTRACT

The storage, preservation, safe and transparent access to personally identifiable information (PII) is an issue that over time has been taking more into account in the field of information security. In this sense, this project is focused on developing a secure and distributed storage component for the management of PII.

The proposed component uses a distributed and fragmented architecture to guarantee the CIA triad (confidentiality, integrity, and availability) of the stored information. Security controls are implemented to be able to access the information, one of these is the verification of digital signatures which maintain the integrity of the data. In addition, a replication process is used to always maintain availability through data synchronization.

The threat modeling tool was used to identify and mitigate potential risks and threats that may arise during the storage of sensitive data. Likewise, a risk analysis was carried out to identify and plan contingency management in case threats occur.

Finally, a test application that integrates the storage component with two other components, data collection and transfer, was developed to illustrate the viability and effectiveness of the secure and distributed storage component.

In summary, the present work provides a practical and effective approach for the storage, preservation, secure and transparent access to personally identifiable information. The distributed and fragmented architecture of the storage component ensures the CIA triad of the stored data, which guarantees the protection of sensitive data.

KEYWORDS: storage, preservation, secure access, transparent access, personally identifiable information, distributed architecture.

1. INTRODUCCIÓN

Este capítulo menciona los aspectos generales del trabajo de integración, tales como una descripción general del componente, el objetivo general propuesto, así también como los objetivos específicos, el alcance del proyecto, conceptos teóricos fundamentales y los diferentes trabajos relacionados al proyecto.

1.1. Descripción del Componente

El desarrollo de este componente lo vamos a separar en dos partes. La primera va a consistir en realizar un análisis y selección de una metodología adecuada para implementar un sistema de almacenamiento que garantice la preservación, acceso seguro y transparencia de la información personal identificable (PII). Este sistema debe obedecer los requisitos fijados por la Ley Orgánica de Protección de Datos Personales (LODPD), el Reglamento General de Protección de Datos (GDPR) y los principios de la inversión socialmente responsable y ambientalmente sostenible (ESGI).

Este sistema de almacenamiento permitirá al usuario ejercer sus derechos de consentimiento, acceso, actualización, corrección y eliminado de su información personal en cualquier momento que lo desee, lo que les dará un mayor control y transparencia sobre sus datos personales.

La primera parte del componente se llevó a cabo a través de una pregunta de investigación, la cual se formuló de la siguiente manera: "¿Cuáles son los diferentes tipos de almacenamiento y cuáles son sus ventajas y desventajas?". A partir de esta pregunta, se realizó una revisión sistemática de la literatura con el objetivo principal de investigar cual son los estudias que se han realizado (estado del arte) sobre de los tipos de almacenamiento, sus ventajas y desventajas, y así determinar cuál de ellos era el más seguro.

Gracias a esta investigación, se pudo tomar una decisión acerca del prototipo de almacenamiento que sería implementado en el sistema. Además, se obtuvieron los conocimientos necesarios para establecer los requerimientos para la implementación de la arquitectura del prototipo que se desarrollaría.

Después de realizar profundamente la revisión sistemática de la literatura, se seleccionó una arquitectura de base de datos distribuida donde se realizará el almacenamiento de la información personal identificable (PII) del sistema. Esta base de datos se implementó en dos nodos y además posee una única base de datos consolidada.

La seguridad y privacidad de los datos se garantiza gracias a la implementación de un proceso de recopilación y transferencia de datos basado en firmas digitales. De esta manera, se asegura que cualquier usuario que acceda al sistema sólo pueda ver la información que le corresponde, evitando así cualquier intento de acceso a la información confidencial por parte de usuarios que no posean los permisos requeridos.

Para la implementación de este sistema de base de datos distribuida, se trazó una arquitectura que permite el acceso seguro y transparente a la información personal identificable (PII) a través de vistas y procedimientos almacenados. En la Figura 1, se puede observar la disposición de los nodos y el mecanismo de acceso que se implementará.

Gracias a esta arquitectura, se logra un acceso eficiente y controlado a la información almacenada, evitando así posibles brechas de seguridad o accesos a la información confidencial sin autorización. Además, la implementación de procedimientos almacenados permite una mayor flexibilidad en la inserción de la información, asegurando la integridad de los datos almacenados y facilitando su gestión.

Con esta arquitectura, el sistema podrá cumplir con los requerimientos establecidos por la Ley Orgánica de Protección de Datos Personales (LODPD), el Reglamento General de Protección de Datos (GDPR) y los principios de la inversión socialmente responsable y ambientalmente sostenible (ESGI). También permitirá a los usuarios del sistema ejercer sus derechos de consentimiento, acceso, corrección, actualización y borrado de su información personal de manera eficiente y transparente.

1.2. Objetivo General

El objetivo general de este proyecto es:

- Asegurar la preservación de información y el acceso transparente y autorizado a la misma, en cumplimiento con la Ley Orgánica de Protección de Datos Personales (LOPDP) del Ecuador.

1.3. Objetivos Específicos

Los objetivos específicos de este proyecto son:

1. Analizar el estado del arte de soluciones para el almacenamiento, preservación y acceso transparente a PII.
2. Identificar requerimientos funcionales y no funcionales relacionados con la seguridad y privacidad de PII durante su almacenamiento.
3. Implementar un prototipo experimental de acuerdo con los lineamientos del Reglamento General de Protección de Datos (GDPR) europeo, la LOPDP ecuatoriana, el Esquema Gubernamental de Seguridad de la Información (EGSI) y la Ley Orgánica de Transparencia y Acceso a la Información Pública (LOTAIP), garantizando una preservación segura y acceso transparente a la misma.
4. Evaluar el prototipo propuesto en un entorno controlado, considerando su resiliencia ante la eventual manipulación externa de PII que se encuentra almacenada en una base de datos.

1.4. Alcance

El objetivo principal de este proyecto es implementar mecanismos de control que garanticen la seguridad y privacidad de la información personal identificable (PII) durante su almacenamiento, preservación y acceso transparente en cumplimiento con las regulaciones del GDPR europeo, la LOPDP ecuatoriana, el EGSI y la LOTAIP. En resumen, el alcance del proyecto incluye el diseño, implementación y evaluación de un sistema de almacenamiento distribuido seguro y transparente que cumpla con los requerimientos legales y los principios de la inversión socialmente responsable y ambientalmente sostenible.

Este proyecto contempla el despliegue de un servicio seguro de almacenamiento de información en entidades interesadas, con el fin de identificar y abordar los requerimientos

funcionales y no funcionales relacionados con la seguridad y privacidad de PII. Los mecanismos de control implementados serán prototipos experimentales y se evaluarán en entornos controlados, sin que esto implique su despliegue en alguna infraestructura productiva, ya sea pública o privada.

El sistema de almacenamiento implementado se encargará de evitar la manipulación de información almacenada por parte de terceros no autorizados, lo que garantizará la privacidad y confidencialidad de la información almacenada.

En resumen, el alcance del proyecto incluye el diseño, implementación y evaluación de un sistema de almacenamiento distribuido seguro y transparente que cumpla con los requerimientos legales y los principios de la inversión socialmente responsable y ambientalmente sostenible.

En la Figura 1 se muestra el proceso de almacenamiento que se implementa en varias entidades.

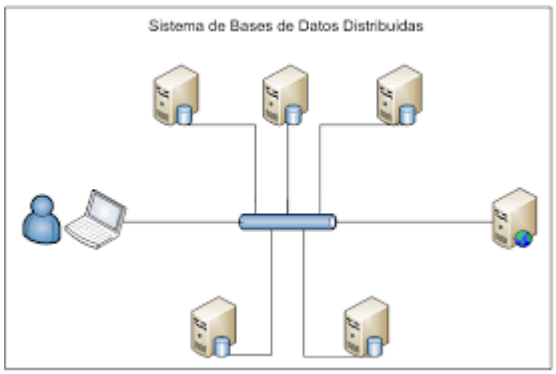


Figura 1 Bases de Datos Distribuidas [1]

1.5. Marco Teórico

En el marco teórico, se abordan los conceptos más importantes que se revisaron en el estado del arte y que se utilizaron para el desarrollo del componente propuesto.

1.5.1. Bases de Datos

Una base de datos es una colección de datos organizados y relacionados entre sí, diseñados para ser utilizados y accedidos de manera eficiente. Estos datos pueden ser de diferentes tipos, como texto, números, imágenes, audio, video, entre otros. Además, la base de datos debe estar diseñada de manera que permita la gestión y control de los datos almacenados [2].

Existen diferentes tipos de bases de datos, cada una diseñada para un propósito específico. Algunos ejemplos son las bases de datos relacionales, las bases de datos NoSQL y las bases de datos en tiempo real. Cada tipo tiene sus propias ventajas y desventajas, por lo que es importante elegir el tipo adecuado para la aplicación en cuestión [3].

La seguridad y privacidad de los datos también son aspectos importantes en el diseño y gestión de bases de datos. Es necesario implementar mecanismos de control de acceso y protección de datos para evitar el acceso no autorizado y la manipulación de los datos almacenados [4].

1.5.2. Bases de Datos Distribuidas

Las bases de datos distribuidas se refieren a un conjunto de bases de datos que están ubicadas en distintos nodos y que están conectados a través de una red de comunicaciones. Esta tecnología se utiliza en entornos donde se requiere un alto rendimiento y escalabilidad, permitiendo el acceso y la gestión de grandes volúmenes de datos distribuidos en diferentes ubicaciones geográficas [1].

La implementación de bases de datos distribuidas presenta numerosos desafíos técnicos en cuanto a la administración y el mantenimiento de los datos distribuidos, como la sincronización de datos, la redundancia y la recuperación ante fallas, la seguridad y la privacidad. En la actualidad, existen diferentes arquitecturas y técnicas para la implementación de bases de datos distribuidas, como la replicación de datos, la fragmentación y la partición de datos [2], [5].

1.5.3. Fragmentación

La fragmentación es un proceso en el cual una base de datos se divide en fragmentos más pequeños para mejorar la eficiencia en la recuperación de datos y la gestión de recursos en sistemas de bases de datos distribuidas. Esta técnica es ampliamente utilizada en entornos empresariales donde se manejan grandes volúmenes de datos. Los fragmentos pueden ser distribuidos en diferentes sitios geográficos para mejorar la disponibilidad de datos y reducir el tiempo de respuesta en consultas [1].

La fragmentación se puede clasificar en horizontal y vertical. La fragmentación horizontal divide una tabla en fragmentos de filas, mientras que la fragmentación vertical divide una tabla en fragmentos de columnas [2]. Además, existen técnicas de fragmentación híbridas que combinan ambas técnicas para lograr una mejor distribución de los datos en sistemas distribuidos [6].

1.5.4. Replicación

La replicación en bases de datos se refiere al proceso de crear y mantener copias de los mismos datos en múltiples servidores para mejorar la disponibilidad y confiabilidad del sistema [1]. La replicación permite que los usuarios accedan a los datos de manera local, en lugar de tener que consultar un servidor centralizado. Además, la replicación puede mejorar el rendimiento del sistema al distribuir la carga de trabajo entre los servidores. Existen diferentes técnicas de replicación, incluyendo replicación completa, replicación parcial y replicación por demanda [7]. La elección de la técnica de replicación depende de los requerimientos del sistema y de la cantidad de datos que se deben replicar.

La replicación puede ser un proceso complejo debido a la necesidad de mantener la consistencia de los datos entre los servidores. La sincronización de los datos es un aspecto crítico de la replicación, y se requieren mecanismos para garantizar que los datos en todos los servidores estén actualizados y sean coherentes [8]. La replicación también puede introducir problemas de confidencialidad y seguridad, ya que se deben tomar medidas para proteger los datos replicados y garantizar que solo sean accesibles para usuarios autorizados.

1.5.5. Modelo Entidad Relación

El Modelo Entidad-Relación (MER) es una herramienta de modelado conceptual de datos que permite representar de manera gráfica y abstracta la estructura de una base de datos. El MER está compuesto por dos elementos principales: entidades y relaciones. Las entidades representan objetos o conceptos del mundo real, mientras que las relaciones establecen vínculos entre estas entidades [2].

El MER es ampliamente utilizado en la etapa de diseño de bases de datos para describir los requerimientos del sistema y definir una estructura clara y coherente. El modelo se basa en un conjunto de reglas y convenciones que permiten la representación de diferentes aspectos de la realidad de forma precisa y concisa [9]. Además, el MER sirve como punto de partida para la generación de esquemas de bases de datos y para la implementación de sistemas de gestión de bases de datos relacionales [10].

1.5.6. Modelo Relacional

El modelo relacional es una forma de representar datos en una base de datos mediante tablas que se relacionan entre sí. Fue introducido por E.F. Codd en 1970 [11]. En este modelo, los datos se organizan en tablas, también llamadas relaciones, donde cada fila representa una entidad y cada columna una propiedad o atributo. Las relaciones entre las

tablas se establecen mediante claves primarias y foráneas, lo que permite relacionar las filas de diferentes tablas y hacer consultas complejas.

El modelo relacional es el modelo de bases de datos más utilizado en la actualidad debido a su simplicidad y facilidad de uso. Una de las ventajas de este modelo es que permite evitar la duplicidad de datos y, por lo tanto, reduce el riesgo de errores y la inconsistencia de datos. Sin embargo, también tiene limitaciones, como la complejidad de algunas consultas que pueden requerir múltiples tablas y la posible degradación del rendimiento en bases de datos muy grandes [12].

En resumen, el modelo relacional es una forma eficiente y útil de organizar datos en una base de datos. A pesar de sus limitaciones, sigue siendo el modelo más utilizado en la actualidad debido a su simplicidad y facilidad de uso.

1.5.7. Grafo Relacional

El grafo relacional es un modelo de datos que representa la estructura de una base de datos como un grafo dirigido etiquetado. En este modelo, los nodos representan entidades y los arcos representan las relaciones entre ellas. El grafo relacional se utiliza como una alternativa al modelo relacional tradicional, ya que permite una representación más flexible y eficiente de la información, especialmente en bases de datos con muchas relaciones complejas [13].

Una de las principales ventajas del grafo relacional es que permite una consulta más rápida y eficiente de los datos. Al utilizar un grafo para representar las relaciones entre las entidades, las consultas pueden ser resueltas utilizando algoritmos de búsqueda en grafos, lo que puede ser más rápido que las consultas basadas en el álgebra relacional [3].

Además, grafo relacional es capaz de representar relaciones de varias cardinalidades, incluyendo relaciones n-a-m, que pueden ser difíciles de representar en el modelo relacional tradicional [14]. Sin embargo, la principal desventaja del grafo relacional es que es menos intuitivo que el modelo relacional, lo que hace que sea más difícil de entender y mantener [15].

En conclusión, el grafo relacional es un modelo de datos eficiente y flexible para representar relaciones complejas en bases de datos. Si bien tiene algunas desventajas, es una alternativa interesante al modelo relacional tradicional y ha sido utilizado con éxito en varias aplicaciones.

1.5.8. Procedimientos Almacenados

Los procedimientos almacenados son un tipo de programa que se ejecuta en una base de datos y se utiliza para realizar operaciones en los datos almacenados en ella. Estos procedimientos son útiles para reducir el tráfico de red y mejorar el rendimiento, ya que permiten realizar operaciones complejas en la base de datos sin la necesidad de transferir grandes cantidades de datos a través de la red [2].

Un procedimiento almacenado se puede escribir en diferentes lenguajes de programación, como SQL, PL/SQL, T-SQL, entre otros [16]. Además, los procedimientos almacenados pueden aceptar parámetros de entrada y devolver valores de salida, lo que los hace muy flexibles y adaptables a diferentes situaciones.

Los procedimientos almacenados también son útiles para garantizar la integridad de los datos en la base de datos, ya que se pueden utilizar para implementar restricciones y validaciones en los datos que se ingresan en la base de datos. Además, los procedimientos almacenados pueden ser utilizados para automatizar tareas de mantenimiento de la base de datos, lo que reduce la necesidad de intervención humana en el proceso [17].

En resumen, los procedimientos almacenados son una herramienta poderosa y versátil para trabajar con bases de datos, ya que permiten realizar operaciones complejas, reducir el tráfico de red y garantizar la integridad de los datos almacenados. Además, su capacidad para aceptar parámetros y devolver valores de salida los hace muy flexibles y adaptables a diferentes situaciones.

1.5.9. Vistas

Las vistas son objetos de base de datos que pueden utilizarse para simplificar la complejidad de una consulta al crear una vista que se construya a partir de tablas relacionadas, estas vistas se definen por una consulta y los datos devueltos por ella. Las vistas no contienen datos, sino que son consultas almacenadas en la base de datos que pueden accederse como una tabla. Al ejecutar una consulta que referencia a una vista, la consulta real se ejecuta utilizando la definición de la vista. Las vistas permiten reducir la complejidad de las consultas y aumentar la seguridad de los datos al limitar el acceso a una tabla solo a los campos necesarios para la vista [18].

Las vistas se utilizan comúnmente para proporcionar una visión personalizada de una base de datos a los usuarios que tienen diferentes necesidades de información. Además, las vistas también se utilizan para simplificar la estructura de la base de datos, reducir la cantidad de código SQL y mejorar la velocidad de las consultas. Las vistas pueden ser

utilizadas en combinación con otros objetos de base de datos, como procedimientos almacenados y funciones, para proporcionar una capa de abstracción de la complejidad de la base de datos [12].

1.5.10. API REST

Un API REST (Representational State Transfer) es un estilo de arquitectura de software que define un conjunto de restricciones y principios para crear servicios web. Se basa en el protocolo HTTP y se utiliza para crear aplicaciones web que sean escalables, flexibles y mantenibles. El API REST utiliza los métodos HTTP como GET, POST, PUT, DELETE, entre otros, para crear una interfaz que permita a las aplicaciones cliente interactuar con los recursos del servidor de manera uniforme y coherente [19], [20].

Los recursos en una API REST se identifican mediante URLs y pueden ser cualquier cosa, desde imágenes hasta documentos, pasando por datos estructurados. El formato de representación de los recursos se establece mediante los tipos MIME (Multipurpose Internet Mail Extensions), como JSON (JavaScript Object Notation) o XML (eXtensible Markup Language) [20], [21].

El API REST ha ganado popularidad en los últimos años debido a su facilidad de uso y flexibilidad en la creación de servicios web. Permite a los desarrolladores crear aplicaciones web complejas utilizando tecnologías y lenguajes de programación diferentes, lo que hace que sea muy fácil de integrar con otras aplicaciones [19], [20].

1.6. Estado del Arte y Trabajo Relacionado

En esta unidad hace referencia a los trabajos relacionados que dieron una ayuda y apoyo a la creación del documento.

1.6.1. Revisión Sistemática de la Literatura

Se realizo un análisis del estado del arte de soluciones para el almacenamiento seguro de PII. El propósito de esta investigación es analizar los estudios existentes y sus hallazgos con el fin de entender la problemática planteada y encontrar todas las referencias posibles para dar solución a la pregunta de investigación planteada:

- ¿Cuál es el mejor mecanismo para un seguro almacenamiento, preservación y control de acceso?

El documento proporcionó contribuciones para futuras investigaciones sobre el mecanismo de almacenamiento seguro.

- Identificamos 35 estudios principales relacionados con seguridad hasta mediados del 2022. Otros investigadores pueden usar esta lista de estudios para avanzar en su trabajo en este campo específico.
- Seleccionamos además 10 estudios primarios que cumplen con los criterios que establecimos para la evaluación de la calidad. Estos estudios pueden proporcionar puntos de referencia adecuados para el análisis comparativo con investigaciones similares.
- Realizamos una revisión exhaustiva de los datos contenidos en el subconjunto de 10 estudios y presentamos los datos para expresar la investigación, las ideas y las consideraciones en los campos de mecanismos de almacenamiento, preservación y control de acceso.

El presente estudio de revisión sistemática de la literatura (SLR) empleó la metodología PRISMA, la cual establece una secuencia de pasos para garantizar la presentación precisa y adecuada del SLR y así poder responder a la pregunta de investigación planteada de manera efectiva.

El método PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) es un enfoque sistemático para la realización de revisiones sistemáticas y metaanálisis en ciencias de la salud [22]. Este método se utiliza para identificar, seleccionar y evaluar críticamente estudios relevantes que aborden una pregunta de investigación específica. PRISMA tiene como objetivo mejorar la transparencia y la calidad de los informes de revisiones sistemáticas y metaanálisis para permitir la evaluación de la validez de los resultados y su aplicabilidad en la práctica clínica [23].

El proceso PRISMA consta de una serie de etapas bien definidas que incluyen la identificación de la pregunta de investigación, la búsqueda sistemática y la selección de estudios, la evaluación de la calidad de los estudios incluidos y la síntesis de los resultados [22]. La implementación adecuada de estas etapas asegura que la revisión sistemática o el metaanálisis sean rigurosos y confiables [24].

El uso del método PRISMA se ha vuelto cada vez más común en la literatura científica, y muchos investigadores lo consideran un estándar de oro para la realización de revisiones sistemáticas y metaanálisis [25]. La aplicación adecuada del método PRISMA puede ayudar a los investigadores a obtener resultados más precisos y confiables, lo que a su vez puede mejorar la toma de decisiones clínicas y mejorar la calidad de la atención al paciente.

En las siguientes secciones, se presentarán los resultados alcanzados tras aplicar la metodología PRISMA en el análisis del estado del arte, el cual sirve como fundamento para la elaboración del documento del proyecto.

La búsqueda de documentos de investigación se hizo en el periodo del 2017 al 2022 (fecha actual), en cual logramos encontrar un documento relevante en 2017, en el año que más se logró encontrar es en el 2021 con 3 documentos. Esto se puede observar el la Figura 2. Así deducimos que mientras más vamos avanzando en los años, hay más documentos para investigar, ya que, el tema se presta para hacer investigación al respecto, no hay que sorprenderse que en el futuro haya más investigaciones sobre este tema, debido a que mucha gente está interesada por saber cómo se utilizan y protegen sus datos personales.

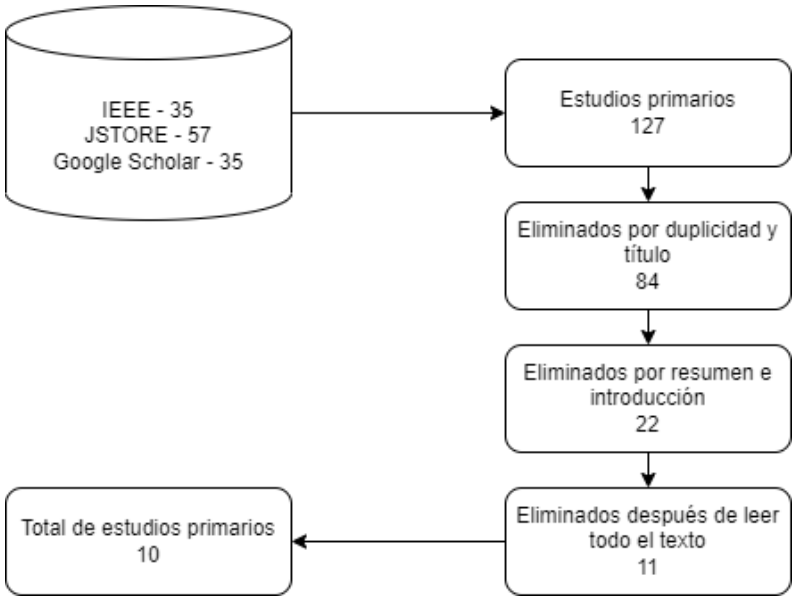


Figura 2 Diagrama de Flujo
[Autor: Cristhian Tohasa]

En la Tabla 1 están los hallazgos más importantes que el documento menciona, se puede ver la tabla completa en el Anexo I.

Tabla 1 Literatura Relevante
[Autor: Cristhian Tohasa]

No.	Estudios Primarios		
	Titulo	**Autores**	**Enfoque**
S1	"An Integrated Privacy…" [26].	Runhua Xu… [26]	Control de acceso de datos
S2	"Blockchain for Giving…" [27].	Mohammad Moussa Madine… [27].	Preservación de datos
S3	"RS-HABE: Revocable-Storage…" [28].	Jianghong Wei… [28].	Almacenamiento y preservación de datos
S4	"Unified Fine-Grained…" [29].	Wei Li… [29].	Control de acceso

No.	Estudios Primarios		
	Titulo	**Autores**	**Enfoque**
S5	“Ciphertext-Policy Attribute-Based…” [30].	Fuhu Deng… [30].	Preservación y control de acceso de datos
S6	“Efficient and Expressive…” [31].	Kennedy Edemacu… [31].	Control de acceso de datos
S7	“Fully Decentralized Multi-Party…” [32].	Mohammad Moussa Madine… [32].	Almacenamiento y preservación de datos
S8	“HTAC: Fine-Grained…” [33].	Qi Li… [33].	Control de acceso de datos
S9	“Secure Decentralized Attribute-Based…” [34].	Leyou Zhang… [34].	Almacenamiento de datos
S10	“A Security Model…” [35].	Al Hamid H… [35].	Almacenamiento de datos

En esta investigación se pudo observar y deducir que cada vez las organizaciones buscan una mejor manera de implementar seguridad para los datos personales que adquieren de sus clientes y muchas personas interesadas en saber cómo se utilizan sus datos. Por ende, los estudios a realizar se podrían resumir en:

- Investigación potencial 1: El almacenamiento seguro en la nube. Se podría seguir avanzando en este campo de la informática con más enfoque en llaves seguras de acceso a la información que se guarda ahí, como lo es en el caso de AWS. Con la incentivación de proponer soluciones a la humanidad y así reducir los actos ilegales y de corrupción.

- Investigación potencial 2: Preservación y transparencia de datos. En esta investigación se podrían hacer instigaciones del cómo desarrollar algoritmos incluso con inteligencia artificial para detectar quién tiene acceso a datos personales, esto partiendo de los permisos del cliente y así poder con un sistema biométrico detectar a quién debe o no tener acceso a esos datos.

2. METODOLOGÍA

En esta sección se describirá la metodología seleccionada y las herramientas empleadas para el diseño y desarrollo del prototipo experimental, el cual busca abordar la problemática planteada de manera efectiva.

2.1. Selección de Herramientas

En esta sección se describirán las herramientas y metodologías utilizadas en la implementación y despliegue de la solución propuesta. Para la construcción del proyecto se utilizó la metodología Design Science Research (DSR), que se enfoca en la creación de

prototipos que permitan solucionar problemas y generar nuevos conocimientos. Asimismo, se empleó la metodología SCRUM para la construcción del prototipo final, ya que proporciona un marco de trabajo estructurado en pasos para el desarrollo ágil de software. A continuación, se detallarán estas y otras herramientas utilizadas en la implementación de la solución.

2.1.1. Design Science Research (DSR)

El diseño de investigación basado en la ciencia (DSR) es una metodología para la creación de soluciones innovadoras en el campo de la tecnología de la información y la comunicación (TIC) [36]. Esta metodología se centra en la creación de conocimiento nuevo y la aplicación de este conocimiento en la práctica a través de la creación de artefactos, como prototipos, modelos y sistemas [37].

El Proceso DSR

El proceso de DSR consta de seis etapas: identificación del problema, definición de los objetivos, diseño y desarrollo del artefacto, demostración del artefacto, evaluación y comunicación [36].Esta metodología se enfoca en la creación de soluciones prácticas que aborden problemas específicos, y se basa en la retroalimentación constante para mejorar el proceso de diseño y desarrollo del artefacto [37].

El DSR se ha utilizado ampliamente en el campo de las TIC para abordar una variedad de problemas, desde la creación de sistemas de información empresarial hasta el diseño de sistemas de seguridad cibernética [38].La metodología DSR ha demostrado ser efectiva para desarrollar soluciones prácticas y significativas en la práctica, así como para producir conocimiento nuevo y teórico [37].

El proceso de Design Science Research (DSR) se basa en la idea de que la construcción y evaluación de artefactos es una forma válida de investigación en las ciencias de la información y la tecnología. Este enfoque se enfoca en la creación de soluciones prácticas y concretas para problemas específicos, mientras se genera nuevo conocimiento a través de la construcción de artefactos. El proceso de DSR consta de seis etapas: identificación del problema, definición de los objetivos del artefacto, diseño y desarrollo, demostración, evaluación y comunicación [36].

Durante la etapa de identificación del problema, se analiza y define el problema en cuestión, se establecen los requisitos y se identifica la oportunidad para la creación de un artefacto [37].

En la etapa de definición de los objetivos del artefacto, se establecen los objetivos del artefacto, las restricciones y las medidas de éxito [37].

En la etapa de diseño y desarrollo, se crea el artefacto y se establecen los criterios de calidad [37].

En la etapa de demostración, se presenta el artefacto y se demuestra su capacidad para resolver el problema en cuestión [37].

En la etapa de evaluación, se evalúa la utilidad, eficacia, eficiencia, satisfacción del usuario y calidad del artefacto [37].

Finalmente, en la etapa de comunicación, se presenta el artefacto y se comunica el conocimiento generado a través de la creación del artefacto [37].

Este enfoque es ampliamente utilizado en la investigación de sistemas de información, y ha demostrado ser útil en la solución de problemas prácticos en una amplia gama de dominios. La aplicación de DSR no solo permite a los investigadores resolver problemas del mundo real, sino que también puede generar conocimiento teórico a través de la construcción de artefactos.

En la Figura 3 se muestra el proceso de un DSR

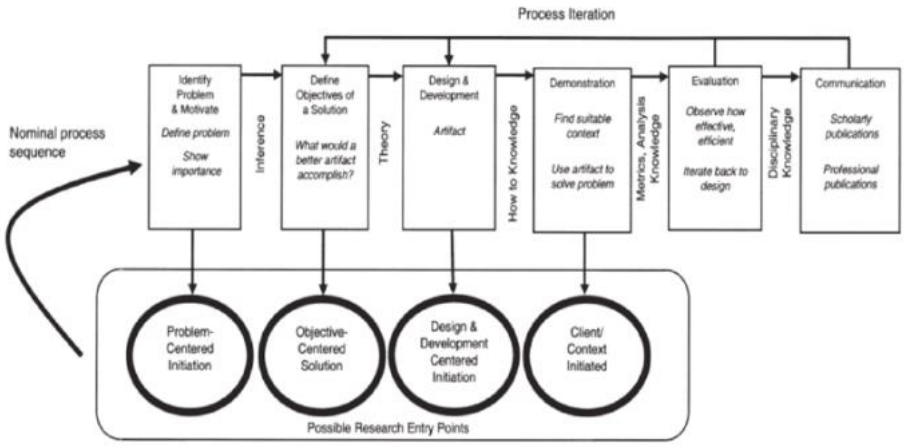


Figura 3 Modelo de Proceso [37]

2.1.2. Scrum

El marco de trabajo Scrum es una metodología ágil que se utiliza para la gestión y el desarrollo de proyectos de software. Se basa en la iteración y la colaboración en equipo, con el objetivo de crear productos de alta calidad en un plazo de tiempo breve. La metodología se enfoca en la entrega incremental y continua de funcionalidades del

producto final, permitiendo a los desarrolladores responder rápidamente a los cambios y adaptarse a las necesidades del cliente [39].

Scrum se compone de tres roles principales: el Product Owner, el Scrum Master y el Equipo de Desarrollo. El Product Owner es responsable de la definición y priorización de los elementos del producto a desarrollar, mientras que el Scrum Master se enfoca en facilitar el proceso y eliminar los obstáculos que puedan impedir el avance del equipo. El Equipo de Desarrollo se encarga de la implementación del producto, trabajando de forma autónoma e interdependiente.

Scrum se organiza en iteraciones llamadas Sprints, que tienen una duración típica de 2 a 4 semanas. Durante cada Sprint, el equipo trabaja en una serie de elementos del producto que han sido previamente seleccionados y priorizados por el Product Owner. Al finalizar cada Sprint, se presenta un incremento del producto que puede ser entregado al cliente si se desea.

Scrum es uno de los marcos de trabajo ágil más populares y utilizado en el desarrollo de software y en otros campos. Su éxito se debe a la flexibilidad y eficiencia que ofrece a los equipos de trabajo, permitiéndoles adaptarse rápidamente a los cambios y a los requerimientos del cliente [40].

2.1.3. Python

Python es un lenguaje de programación de alto nivel, interpretado y general-purpose [41]. Fue creado por Guido van Rossum y lanzado por primera vez en 1991 [41]. Es conocido por su sintaxis clara y concisa, lo que lo hace fácil de leer y escribir, y por su amplia gama de bibliotecas y marcos de trabajo que lo hacen adecuado para una variedad de aplicaciones, desde ciencia de datos hasta desarrollo web y móvil [41], [42].

Python tiene una filosofía de diseño que se enfoca en la legibilidad del código y en la facilidad de uso, lo que lo convierte en un lenguaje popular entre principiantes y expertos en programación [41]. Además, es multiplataforma, lo que significa que puede ejecutarse en diferentes sistemas operativos, como Windows, macOS y Linux [41].

Python también es conocido por su comunidad activa y su gran cantidad de recursos de aprendizaje en línea, lo que lo hace fácilmente accesible para aquellos que desean aprender a programar o mejorar sus habilidades [42].

En resumen, Python es un lenguaje de programación popular debido a su sintaxis clara y concisa, su amplia gama de bibliotecas y marcos de trabajo, su enfoque en la legibilidad del código y la facilidad de uso, su multiplataforma y su comunidad activa [41], [42].

2.1.4. Flask

Flask es un micro-framework para desarrollo de aplicaciones web en Python. Su diseño es simple y flexible, permitiendo a los desarrolladores crear aplicaciones web rápidamente y con pocas líneas de código. Flask no impone una estructura específica para la aplicación, lo que permite a los desarrolladores elegir la estructura que mejor se adapte a sus necesidades. Además, Flask es altamente extensible gracias a su amplia variedad de extensiones disponibles para añadir funcionalidades adicionales a la aplicación [43].

El enfoque minimalista de Flask hace que sea una buena opción para aplicaciones web pequeñas o medianas, así como para prototipos o proyectos de desarrollo rápido. Flask proporciona una amplia variedad de herramientas para la creación de aplicaciones web, incluyendo soporte para plantillas, gestión de sesiones, autenticación, validación de formularios, entre otras funcionalidades [44].

2.1.5. Toad Data Modeler

Toad Data Modeler es una aplicación diseñada para el modelado de datos, la cual brinda a los usuarios la posibilidad de crear, gestionar y actualizar modelos de datos de gran complejidad. Esta herramienta proporciona una interfaz gráfica de usuario para diseñar y construir estructuras de base de datos, incluyendo tablas, relaciones, restricciones y otros elementos clave. Además, Toad Data Modeler admite múltiples plataformas de bases de datos, lo que permite a los usuarios diseñar modelos de datos compatibles con diferentes tipos de sistemas de gestión de bases de datos [45].

Toad Data Modeler es conocido por su capacidad para generar scripts SQL automáticamente a partir de modelos de datos y para permitir a los usuarios validar y optimizar sus diseños de bases de datos. También incluye una variedad de características adicionales, como herramientas de ingeniería inversa, análisis de impacto y generación de documentación, lo que lo convierte en una herramienta de modelado de datos completa [46].

2.1.6. SQL Server

SQL Server es un software desarrollado por Microsoft que pertenece a la categoría de sistemas de gestión de bases de datos relacionales (RDBMS). SQL Server se utiliza para almacenar y recuperar datos solicitados por otras aplicaciones de software. Es una plataforma de base de datos completa que proporciona soporte para la integración de aplicaciones, análisis de datos y análisis de inteligencia empresarial [47].

SQL Server es altamente escalable y puede administrar grandes cantidades de datos y usuarios. La plataforma es compatible con una amplia gama de sistemas operativos, lenguajes de programación y herramientas de desarrollo. SQL Server incluye una variedad de características y herramientas avanzadas, como la replicación de datos, la seguridad avanzada, la inteligencia empresarial y el procesamiento de datos en memoria [48].

En resumen, Microsoft SQL Server es una plataforma de base de datos completa que ofrece características y herramientas avanzadas para la gestión de grandes cantidades de datos, análisis de datos y análisis de inteligencia empresarial. Es ampliamente utilizado en empresas y organizaciones de todo el mundo para almacenar y administrar sus datos [47].

2.1.7. Visual Studio Code

Visual Studio Code es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft. Este editor se ha convertido en una herramienta muy popular para el desarrollo de software debido a su flexibilidad, rapidez y gran cantidad de extensiones que permiten personalizar su funcionalidad según las necesidades del desarrollador [49].

Visual Studio Code soporta múltiples lenguajes de programación y sistemas operativos, lo que lo hace muy versátil y adaptable a diferentes entornos de desarrollo. Además, incluye características útiles para la programación en equipo, como la integración con control de versiones y la posibilidad de compartir sesiones de depuración [50].

2.1.8. Postman

Postman es una plataforma de colaboración para la creación de API y pruebas de software, que permite a los desarrolladores diseñar, compartir, probar, documentar y monitorizar APIs de forma más rápida y sencilla [51]. Es una herramienta de gran utilidad para desarrolladores que trabajan con servicios web RESTful y SOAP, ya que permite hacer solicitudes HTTP a través de una interfaz gráfica de usuario intuitiva, lo que facilita la prueba de servicios y la depuración de errores en el proceso de desarrollo [52].

Además de su función principal, Postman también permite la automatización de pruebas, la integración con sistemas de control de versiones como Git, y la colaboración en equipo, lo que hace que sea una herramienta muy útil en el desarrollo de software en equipo [51].

En resumen, Postman es una herramienta indispensable para desarrolladores que trabajan con APIs, ya que permite diseñar, probar, documentar y colaborar en la creación de servicios web de forma más eficiente y productiva [52].

2.1.9. Microsoft Threat Modeling Tool

Microsoft Threat Modeling Tool es una herramienta que permite a los desarrolladores, arquitectos y analistas de seguridad evaluar la seguridad de sus aplicaciones. La herramienta es capaz de ayudar en la identificación y evaluación de posibles amenazas a través de la modelización de los componentes y la definición de los activos, las entradas y salidas, los puntos de entrada y las amenazas correspondientes [53]. Al utilizar esta herramienta, los equipos de desarrollo pueden anticiparse a los riesgos de seguridad y hacer frente a posibles amenazas desde una fase temprana del desarrollo de software.

El Microsoft Threat Modeling Tool es una herramienta gratuita proporcionada por Microsoft que se puede utilizar con cualquier plataforma y lenguaje de programación [54]. La herramienta ayuda a los equipos de desarrollo a realizar un análisis de amenazas sistemático y detallado mediante la utilización de un enfoque basado en diagramas para modelar la arquitectura de la aplicación y los posibles vectores de ataque. El análisis de amenazas resultante puede utilizarse para identificar los riesgos potenciales y para priorizar las tareas de seguridad necesarias para mejorar la seguridad del sistema.

2.2. Descripción de la Solución

El proceso de almacenamiento se basa en la implementación de una arquitectura de bases de datos distribuidas que consiste en dos nodos fragmentados y una base de datos consolidada. La Figura 4 muestra la arquitectura que se planea desarrollar.

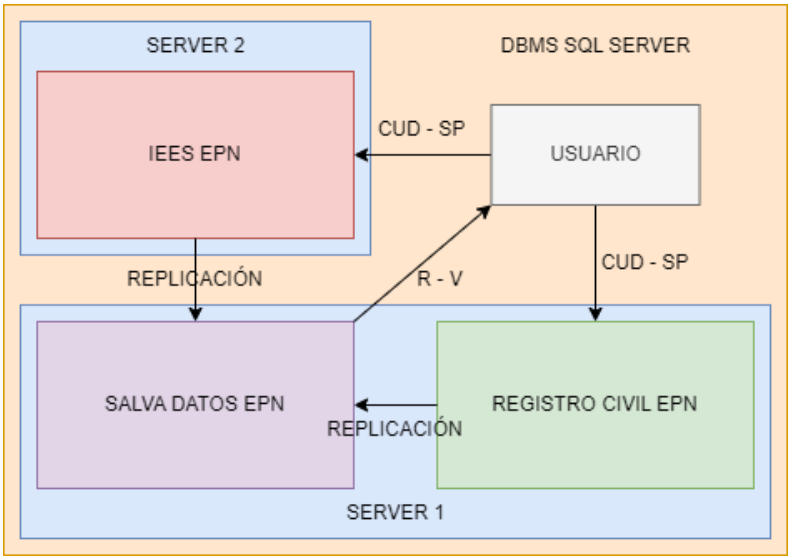


Figura 4 Diagrama de Componentes
[Autor: Cristhian Tohasa]

2.2.1. Descripción del Diagrama de Componentes

En la siguiente Tabla 2 se presenta la descripción de cada componente que se ha propuesto para la gestión de la información.

Tabla 2 Descripción de cada Componente
[Autor: Cristhian Tohasa]

Componente	Descripción
DBMS SQL SERVER	Es la herramienta de base de datos para la gestión de los datos de entrada y salida.
SERVER 1	Es el servidor principal la cual contendrá la base de datos consolidada (SALVA DATOS EPN) y una de las bases de datos fragmentadas (REGISTRO CIVIL EPN).
SERVER 2	Este servidor, por su parte, es el servidor secundario, el cual alojará la otra base de datos fragmentada (IEES EPN).
SALVA DATOS EPN	Es la base de datos consolidada, la cual recibirá los datos provenientes de las bases de datos fragmentadas mediante la replicación.
USUARIO	Es el encargado de realizar las peticiones de: insertar, actualizar y remover (CUD) a las bases de datos fragmentadas a través de un procedimiento almacenado y la petición de consultar (R) a la base de datos consolidada a través de vistas.

2.2.2. Pseudocódico de Diseño de Solución para el Almacenamiento, Preservación, Acceso Seguro y Transparente a PII

En esta sección se presenta el pseudocódigo que describe los pasos del proceso de almacenamiento, preservación, acceso seguro y transparente a PII. A continuación, se en la tabla 3 se muestra el pseudocódigo detalladamente.

Tabla 3 Pseudocódigo del Sistema
[Autor: Cristhian Tohasa]

Pseudocódigo

```
1 Inicio
2 Conexión a las bases de datos fragmentadas y a la base de datos
  consolidada
3 Crear procedimientos almacenados para la inserción, actualización y
  eliminación de datos en las bases de datos fragmentadas
4 Replicar los cambios de las bases de datos fragmentadas a la base de
  datos consolidada
5 Crear vistas en la base de datos consolidada para consultas de PII
6 Configurar los permisos de acceso a los procedimientos almacenados y
  vistas para garantizar la seguridad de los datos
```

```
 7 Esperar solicitudes de inserción, actualización o eliminación de datos
   en las bases de datos fragmentadas
 8 Verificar que la solicitud sea válida y no viole ninguna política de
   privacidad
 9 Ejecutar el procedimiento almacenado correspondiente para insertar,
   actualizar o eliminar los datos en la base de datos fragmentada
10 Replicar los cambios a la base de datos consolidada
11 Esperar solicitudes de consulta de PII
12 Ejecutar las consultas a través de las vistas en la base de datos
   consolidada
13 Enviar los resultados al solicitante de la consulta
14 Fin
```

2.2.3. Solución para el Almacenamiento, Preservación, Acceso Seguro y Transparente a PII

La solución propuesta para el almacenamiento, preservación, acceso seguro y transparente a PII como ya se comentó en la sección 2.2.1 es la siguiente:

- Verificación de firmas digitales.
    - Verificar la firma digital de la petición.
    - Verificar la firma digital del usuario.
    - Si ambas firmas son válidas, continuar con la petición; de lo contrario, mostrar un mensaje de error.
- Procedimientos almacenados para insertar, actualizar y eliminar datos.
    - Los procedimientos almacenados deben estar disponibles en cada una de las bases de datos fragmentadas.
    - Las peticiones de inserción, actualización y eliminación de datos deben realizarse a través de estos procedimientos almacenados.
- Replicación de datos.
    - Configurar la replicación de los datos de las bases de datos fragmentadas a la base de datos consolidada.
    - La replicación debe ser bidireccional para asegurar que los datos se mantengan actualizados en todas las bases de datos.
- Consulta a través de vistas
    - Crear vistas en la base de datos consolidada para cada una de las tablas en las bases de datos fragmentadas.

- Las consultas deben realizarse a través de estas vistas para garantizar que los usuarios solo accedan a la información que se les permite.

Con esta solución, se garantiza que los datos PII se almacenen y preserven de manera segura y transparente, y que solo los usuarios autorizados tengan acceso a ellos.

2.2.4. Requerimientos para el Almacenamiento, Preservación, Acceso Seguro y Transparente a PII

Para lograr que el mecanismo diseñado cumpla con el alcance planteado, es esencial establecer un conjunto de requerimientos que formen parte integral del sistema. A continuación, se presenta una lista de los requerimientos planteados para el sistema en la Tabla 4.

Tabla 4 Requerimientos del Sistema
[Autor: Cristhian Tohasa]

ID	Descripción	Prioridad	Tipo	Riesgo
RQ-01	El sistema debe ser capaz de almacenar PII en dos bases de datos fragmentadas que se replican en una base de datos consolidada.	Alta	Funcional	Alto
RQ-02	El sistema debe permitir la inserción de PII a través de procedimientos almacenados en las bases de datos fragmentadas.	Alta	Funcional	Alto
RQ-03	El sistema debe permitir la actualización de PII a través de procedimientos almacenados en las bases de datos fragmentadas.	Alta	Funcional	Alto
RQ-04	El sistema debe permitir la eliminación de PII a través de procedimientos almacenados en las bases de datos fragmentadas.	Alta	Funcional	Alto
RQ-05	El sistema debe replicar las bases de datos fragmentadas en la base de datos consolidada en tiempo real para asegurar la integridad de los datos.	Alta	No funcional	Alto
RQ-06	El sistema debe verificar la firma digital obtenida durante la recopilación de los datos antes de realizar cualquier petición de inserción, actualización o eliminación.	Alta	Seguridad	Muy alto
RQ-07	El sistema debe verificar la firma digital obtenida durante la transferencia de los datos antes de permitir la consulta de PII.	Alta	Seguridad	Muy alto
RQ-08	El sistema debe permitir el acceso a los datos de forma transparente a través de vistas a la base de datos consolidada.	Alta	Funcional	Alto
RQ-09	El sistema debe garantizar la privacidad de los datos almacenados en el sistema mediante técnicas de encriptación y autenticación de usuarios.	Alta	Seguridad	Muy alto
RQ-10	El sistema debe contar con una copia de seguridad automatizada de los datos	Media	No funcional	Medio

	almacenados en la base de datos consolidada para minimizar el riesgo de pérdida de datos.			

2.2.5. Mapeo de Requerimientos entre Leyes y la Literatura

Una vez planteados y escritos los requerimientos que debería tener el sistema, es conveniente para el desarrollo de nuestro estudio mapear estos mecanismos o principios junto con las leyes y las investigaciones encontradas en la revisión sistemática de la literatura. Es por esto por lo que, se muestra la Tabla 5 con todo el mapeo y el cotejamiento ya mencionado.

Tabla 5 Mapeo entre Requerimientos, Leyes y Literatura
[Autor: Cristhian Tohasa]

Req. No.	Mecanismo	Estudio Primario SLR				Instrumentos		
		Título	Autor(es)	Enfoque	Ref. No.	GDPR	LOPDP	EGSI
RQ. 01 05 06 07 10	Centralización	"An Integrated Privacy Preserving…" [26]	Xu R… [26]	Seguridad	[26]	Art. 5 6 9 17 25	Art. 37 38 39 40 41 42	Control. 5.3.1.2 7.2.1.3 7.2.7 7.2.9
RQ. 01 05 08 10	Blockchain	"Blockchain for Giving…" [27]	Madine M…[27]	Disponibilidad	[27]	Art. 32 35 44	Art. 34 35	Control 14.1.3.1 14.1.3.3 14.1.3.6 14.1.3.8
RQ. 01 05 06 07 09 10	Distribución	"RS-HABE: Revocable-Storage" [28]	Wei J… [28]	Seguridad	[28]	Art. 58 89 91	Art. 43 44 45 46	Control 8.3.1.10 10.1.3.5 10.2.5.6 10.2.6.9
RQ. 02 03 04 06 07	Nube	"Unified Fine-Grained Access…" [29]	Li W… [29]	Acceso	[29]	Art. 17 25 32	Art. 43 44 45 46	Control 14.1.3.1 14.1.3.3 14.1.3.6 14.1.3.8
RQ. 01 06 07 09	Encriptación	"Ciphertext-Policy Attribute-Based…" [30]	Deng F… [30]	Seguridad	[30]	Art. 44 45 46 47 48	Art. 34 35	Control 5.3.1.2 7.2.1.3 7.2.7 7.2.9
RQ. 02 03 04 05 08	Procedimientos Almacenados	"Efficient and Expressive…" [31]	Edemacu K… [31]	Acceso	[31]	Art. 25 Art. 32 Art. 35	Art. 8 Art. 13 Art. 14 Art. 15 Art. 16 Art. 17	Control 7.2.1.3 7.2.7 7.2.9
RQ. 01 05 07 08 10	Vistas	"HTAC: Fine-Grained…" [33]	Li Q… [33]	Acceso	[33]	Art. 58 89 91	Art. 8 13 14 15 16 17	Control 8.3.1.10 10.1.3.5 10.2.5.6 10.2.6.9
RQ. 01 05 06 07 09	Descentralización	"Fully Decentralized Multi-Party…" [32]	Madine M… [32]	Seguridad	[32]	Art. 44 45 46 47 48	Art. 37 38 39	Control 5.3.1.2 7.2.1.3 7.2.7 7.2.9
RQ. 01 05 06 07 09	Blockchain	"Secure Decentralized Attribute-Based…" [34]	Zhang L… [34]	Seguridad	[34]	Art. 25 32 35	Art. 37 38 39 40 41	Control 8.3.1.10 10.1.3.5 10.2.5.6 10.2.6.9

10							42	
RQ. 01 06 07 09	Replicación	"A Security Model…" [35]	Al Hamid H… [35]	Seguridad	[35]	Art. 5 6 9 17	Art. 43 44 45 46	Control 14.1.3.1 14.1.3.3 14.1.3.6 14.1.3.8

2.2.6. Propuesta de Almacenamiento, Preservación, Acceso Seguro y Transparente a PII

En la actualidad, el manejo de información personal es de gran importancia en cualquier organización, sin embargo, la gestión de esta información debe realizarse de manera responsable y segura para proteger los derechos de privacidad de las personas. En este contexto, surge la necesidad de desarrollar un sistema de almacenamiento, preservación, acceso seguro y transparente de la información personal identificable (PII) en cumplimiento con los estándares de protección de datos y la normativa legal aplicable. En este proyecto se propone la implementación de una solución que aborde estas necesidades a través de la fragmentación de la información en dos bases de datos, su replicación en una base consolidada, y la implementación de mecanismos de verificación de firmas digitales en un API REST que permita el acceso a los datos a través de procedimientos almacenados y vistas de consulta. Esta propuesta busca garantizar la integridad, confidencialidad y disponibilidad de la información personal, además de brindar transparencia en su manejo y asegurar el cumplimiento de la normativa aplicable.

Modelo Entidad Relación

El modelo entidad-relación que se ha propuesto para la implementación representa de manera gráfica las entidades, atributos y relaciones que intervienen en el almacenamiento, preservación, acceso seguro y transparente a PII. Este modelo ha sido diseñado con el objetivo de representar de manera clara y concisa la estructura de datos necesaria para el correcto funcionamiento del sistema propuesto. Además, el modelo ha sido construido teniendo en cuenta los requerimientos establecidos y la normativa vigente en cuanto a la protección de datos personales, asegurando así el cumplimiento de las regulaciones y la protección de la información de los usuarios. En la Figura 5 se puede apreciar la interrelación entre las diferentes entidades, lo que permite entender de forma integral el funcionamiento del sistema y facilita su implementación y mantenimiento.

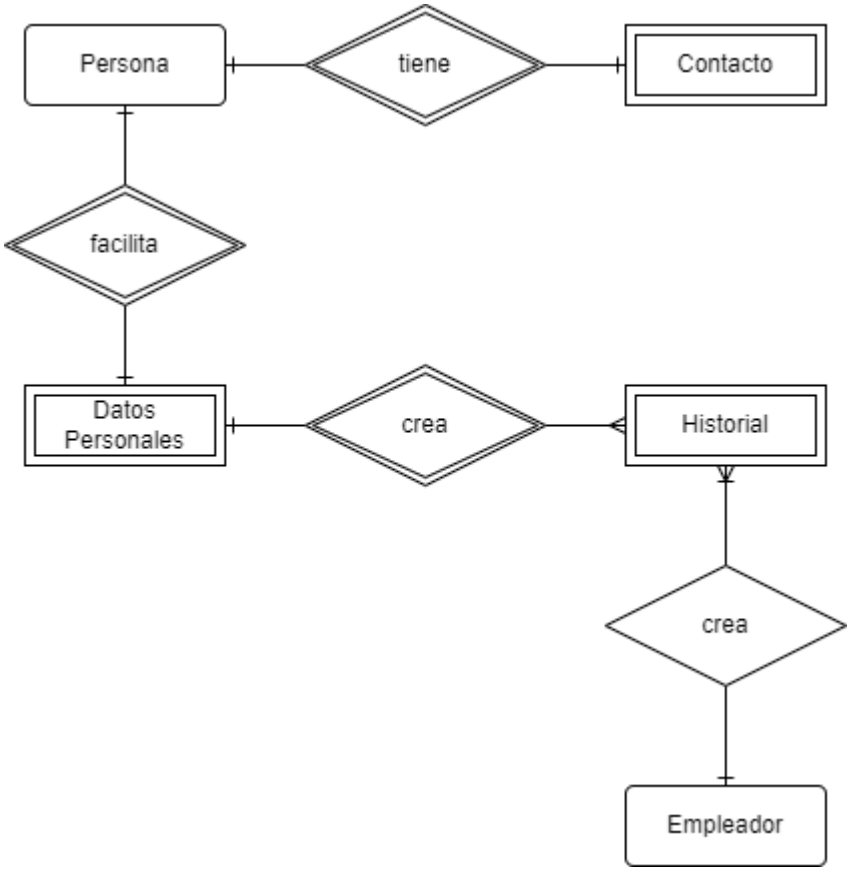


Figura 5 Modelo Entidad Relación
[Autor: Cristhian Tohasa]

Modelo Relacional

El modelo relacional propuesto para la implementación se basa en la conversión del modelo entidad-relación a tablas relacionales, se lo puede observar en la Figura 6. Este modelo se compone de un conjunto de tablas relacionadas que representan entidades, atributos y relaciones entre ellas. Cada tabla representa una entidad y las columnas de la tabla representan los atributos de la entidad. Las relaciones entre entidades se establecen a través de claves foráneas, que permiten la conexión entre las tablas. El modelo relacional facilita la gestión de los datos y asegura la integridad de estos, evitando la redundancia y la inconsistencia de los datos. Además, permite la creación de consultas complejas y la generación de informes precisos y actualizados.

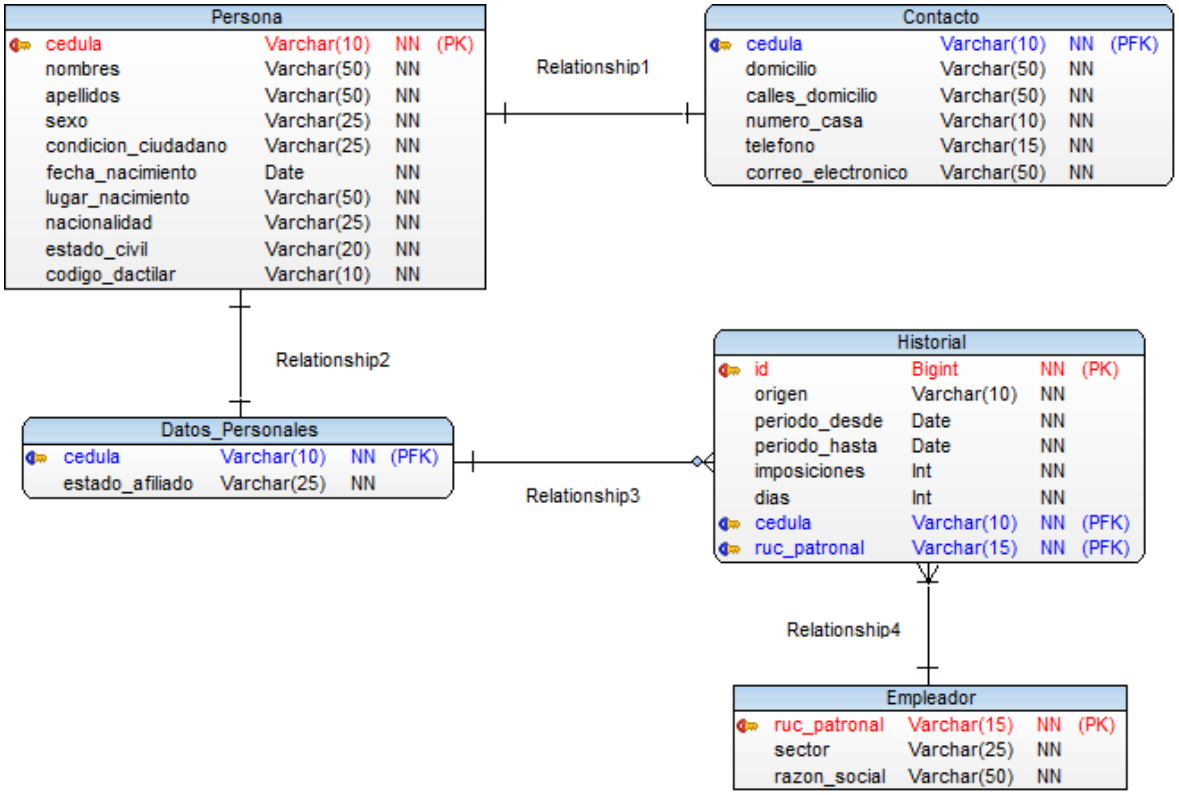


Figura 6 Modelo Relacional
[Autor: Cristhian Tohasa]

Fragmentación

Para la implementación se ha propuesto utilizar dos bases de datos fragmentadas. La fragmentación se ha llevado a cabo por rangos de valores en una columna común a ambas bases. La primera base de datos contendrá los registros cuyo valor en dicha columna esté en un rango determinado, mientras que la segunda base contendrá los registros cuyo valor esté en otro rango. De esta manera, se asegura que los datos estén distribuidos de manera equitativa entre las dos bases y se pueda realizar la replicación de manera efectiva. Además, se ha implementado un mecanismo de sincronización para garantizar que ambas bases estén actualizadas en tiempo real.

Es esquema de asignación se lo puede ver en la Tabla 6.

Tabla 6 Esquema de Asignación
[Autor: Cristhian Tohasa]

	Salva Datos EPN	**Registro Civil EPN**	**IEES EPN**
Persona		Persona	
Contacto		Contacto	
Datos Personales			Datos Personales
Empleador			Empleador
Historial			Historial

También podemos observar el modelo de cada base de datos fragmentada en la Figura 7 y Figura 8.

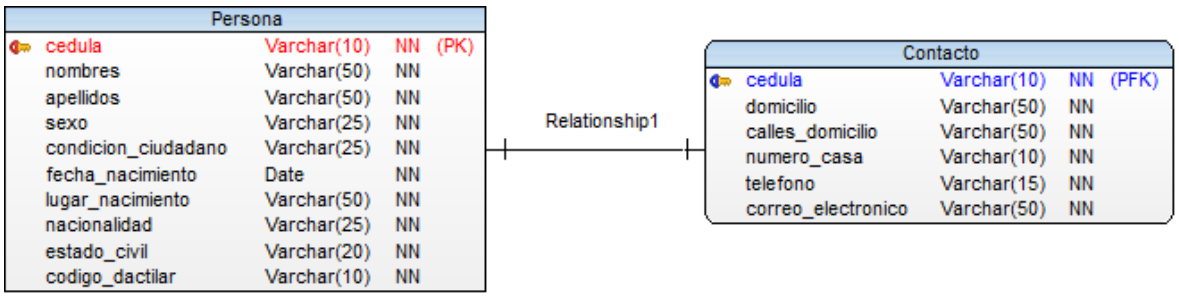


Figura 7 Modelo Relacional
[Autor: Cristhian Tohasa]

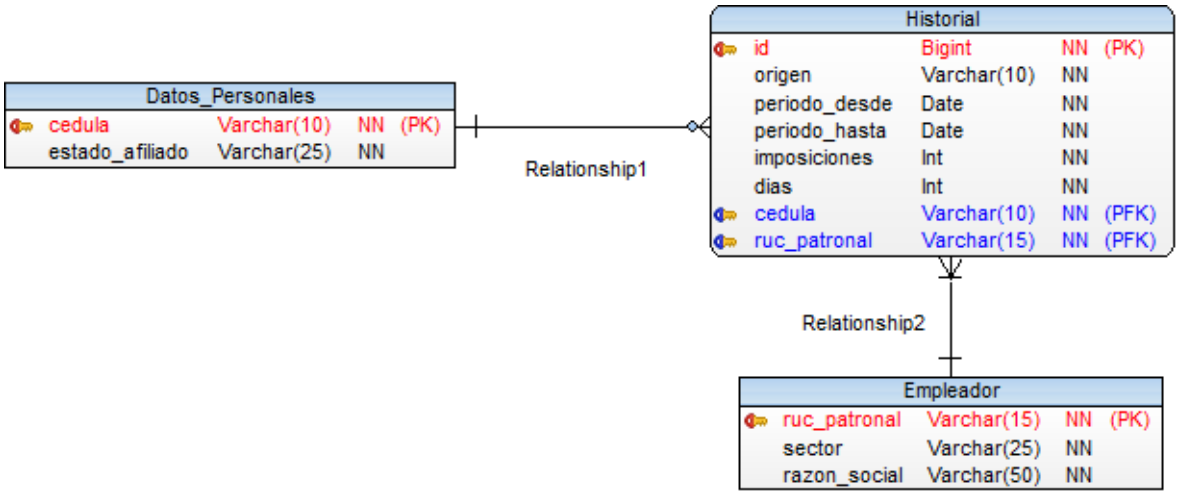


Figura 8 Modelo Relacional IEES EPN
[Autor: Cristhian Tohasa]

Replicación

Para garantizar la preservación y acceso seguro a los datos personales, se ha propuesto replicar las dos bases de datos fragmentadas en una tercera base de datos consolidada. Esta replicación se llevará a cabo mediante un proceso automatizado que garantice la integridad y consistencia de los datos. La base de datos consolidada servirá como la fuente única de consulta para los usuarios autorizados, quienes podrán acceder a los datos a través de vistas creadas en esta base de datos. Este enfoque de replicación también permite una mayor disponibilidad y confiabilidad de los datos, ya que cualquier interrupción en una de las bases fragmentadas no afectará la disponibilidad de la información en la base de datos consolidada.

El esquema de replicación de lo puede observar en la Tabla 7.

Tabla 7 Esquema de Replicación
[Autor: Cristhian Tohasa]

	Salva Datos EPN	**Registro Civil EPN**	**IEES EPN**
Persona	Persona R	Persona	
Contacto	Contacto R	Contacto	
Datos Personales	Datos Personales R		Datos Personales
Empleador	Empleador R		Empleador
Historial	Historial R		Historial

Procedimientos Almacenados

En el marco de la implementación, se ha propuesto el uso de procedimientos almacenados para llevar a cabo la inserción, actualización y eliminación de datos en las dos bases de datos fragmentadas. Estos procedimientos almacenados están diseñados de tal manera que permiten asegurar la consistencia de los datos en ambas bases de datos, evitando la duplicación o la pérdida de información. Asimismo, se han definido políticas de seguridad para garantizar el acceso seguro y transparente a la información. La utilización de procedimientos almacenados brinda además la ventaja de mejorar el rendimiento y la eficiencia en el manejo de grandes volúmenes de información.

En la Tabla 8 se puede ver un ejemplo de procedimiento almacenado

Tabla 8 Ejemplo de Procedimiento Almacenado
[Autor: Cristhian Tohasa]

```
USE [Registro_Civil_EPN]
GO
/****** Object:  StoredProcedure ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[sp_insertarDatosRegistroCivil]
        (@cedula Varchar(10)
        ,@nombres Varchar(50)
        ,@apellidos Varchar(50)
        ,@sexo Varchar(25)
        ,@condicion_ciudadano Varchar(25)
        ,@fecha_nacimiento Date
        ,@lugar_nacimiento Varchar(50)
        ,@nacionalidad Varchar(25)
        ,@estado_civil Varchar(20)
        ,@codigo_dactilar Varchar(10)
        ,@domicilio Varchar(50)
        ,@calles_domicilio Varchar(50)
        ,@numero_casa Varchar(10)
        ,@telefono Varchar(15)
        ,@correo_electronico Varchar(50))
AS
BEGIN
        IF (SELECT [cedula] FROM [dbo].[Persona] WHERE cedula = @cedula) = @cedula
        BEGIN
                RETURN -1
        END
        ELSE
        BEGIN
                INSERT INTO [dbo].[Persona]
                        ([cedula]
```

```
34                          ,[nombres]
35                          ,[apellidos]
36                          ,[sexo]
37                          ,[condicion_ciudadano]
38                          ,[fecha_nacimiento]
39                          ,[lugar_nacimiento]
40                          ,[nacionalidad]
41                          ,[estado_civil]
42                          ,[codigo_dactilar])
43                  VALUES
44                          (@cedula
45                          ,@nombres
46                          ,@apellidos
47                          ,@sexo
48                          ,@condicion_ciudadano
49                          ,@fecha_nacimiento
50                          ,@lugar_nacimiento
51                          ,@nacionalidad
52                          ,@estado_civil
53                          ,@codigo_dactilar)
54
55                  INSERT INTO [dbo].[Contacto]
56                          ([cedula]
57                          ,[domicilio]
58                          ,[calles_domicilio]
59                          ,[numero_casa]
60                          ,[telefono]
61                          ,[correo_electronico])
62                  VALUES
63                          (@cedula
64                          ,@domicilio
65                          ,@calles_domicilio
66                          ,@numero_casa
67                          ,@telefono
68                          ,@correo_electronico)
69          END
70  END
```

Vistas

Para realizar las consultas de datos de manera segura y eficiente, se ha propuesto la utilización de vistas en la base de datos consolidada. Estas vistas permiten acceder a los datos de las dos bases fragmentadas de manera transparente y sin comprometer la seguridad de la información. Además, mediante la creación de vistas específicas, se pueden seleccionar únicamente los datos necesarios para cada consulta, lo que reduce el tiempo de respuesta y mejora el rendimiento del sistema. En resumen, las vistas son una herramienta fundamental para el acceso a la información de manera segura y eficiente en el sistema propuesto.

En la Tabla 9 se puede ver un ejemplo de vista.

Tabla 9 Ejemplo de Vista
[Autor: Cristhian Tohasa]

```
1 USE [Salva_Datos_EPN]
2 GO
3 /****** Object:  View ******/
4 CREATE VIEW v_obtenerDatosNivelCinco
5 AS
6           SELECT      p.cedula
7                       ,p.nombres
8                       ,p.apellidos
```

```
 9                      ,p.sexo
10                      ,p.condicion_ciudadano
11                      ,p.fecha_nacimiento
12                      ,p.lugar_nacimiento
13                      ,p.nacionalidad
14                      ,p.estado_civil
15                      ,p.codigo_dactilar
16                      ,c.domicilio
17                      ,c.calles_domicilio
18                      ,c.numero_casa
19                      ,c.telefono
20                      ,c.correo_electronico
21                      ,d.estado_afiliado
22                      ,e.ruc_patronal
23                      ,e.sector
24                      ,e.razon_social
25                      ,h.origen
26                      ,h.periodo_desde
27                      ,h.periodo_hasta
28                      ,h.imposiciones
29                      ,h.dias
30            FROM Persona p
31                      ,Contacto c
32                      ,Datos_Personales d
33                      ,Empleador e
34                      ,Historial h
35            WHERE p.cedula = c.cedula
36                      AND p.cedula = d.cedula
37                      AND d.cedula = h.cedula
38                      AND h.ruc_patronal = e.ruc_patronal
39 GO
```

Api REST

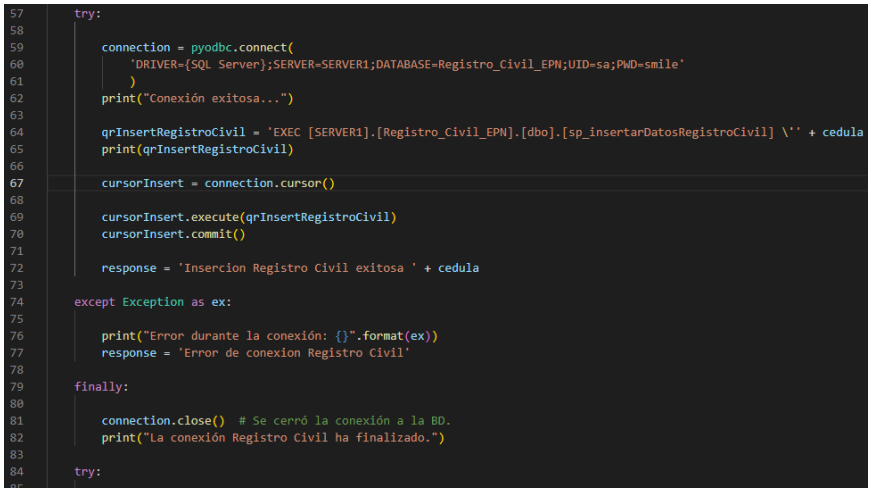
Para la implementación del tema de tesis, se propone el desarrollo de un API REST con Python y Flask que se encargará de la verificación de firmas obtenidas en la recopilación y transferencia de datos, y el llamado a los procedimientos almacenados y vistas de la base de datos consolidada según la petición del usuario. El API REST será el medio por el cual se realizarán las consultas y actualizaciones de la información en la base de datos y se garantizará la seguridad y transparencia en el acceso a los datos personales. El uso de Flask, un framework de Python para aplicaciones web, permitirá la creación de una API REST eficiente, flexible y fácil de implementar. Además, se incluirán mecanismos de autenticación y autorización para garantizar la protección de los datos almacenados.

Como a través de los dos otros componentes se obtuvo las firmas y datos en un xml, el API REST que se implementó obtiene ese xml y obtiene los datos para poder colocarlos

en variables Figura 9. Estas serán concatenadas a la cadena que enviará cuando se ejecute la petición del procedimiento almacenado Figura 10.

```python
@app.route('/salvadatos/insertarDatos', methods=["POST"])
def insertarDatos():

    with open("resources/datos.xml", 'r') as xmlDatos:
        jsonRoot = xmltodict.parse(xmlDatos.read())

    jsonRoot = json.dumps(jsonRoot)
    jsonRoot = json.loads(jsonRoot)

    jsonDatos = jsonRoot['root']

    jsonPersona = jsonDatos['persona']
    jsonContacto = jsonDatos['contacto']
    jsonDatosPersonales = jsonDatos['datosPesonales']
    jsonEmpelador = jsonDatos['empleador']
    jsonHistorial = jsonDatos['historial']

    cedula = jsonPersona['cedula']
    nombres = jsonPersona['nombres']
    apellidos = jsonPersona['apellidos']
    sexo = jsonPersona['sexo']
    condicionCiudadano = jsonPersona['condicionCiudadano']
    fechaNacimiento = jsonPersona['fechaNacimiento']
    lugarNacimiento = jsonPersona['lugarNacimiento']
    nacionalidad = jsonPersona['nacionalidad']
    estadoCivil = jsonPersona['estadoCivil']
    codigoDactilar = jsonPersona['codigoDactilar']

    domicilio = jsonContacto['domicilio']
    callesDomicilio = jsonContacto['callesDomicilio']
    numeroCasa = jsonContacto['numeroCasa']
    telefono = jsonContacto['telefono']
    correoElectronico = jsonContacto['correoElectronico']

    estadoAfiliado = jsonDatosPersonales['estadoAfiliado']

    rucPatronal = jsonEmpelador['rucPatronal']
    sector = jsonEmpelador['sector']
    razonSocial = jsonEmpelador['razonSocial']

    origen = jsonHistorial['origen']
    periodoDesde = jsonHistorial['periodoDesde']
    periodoHasta = jsonHistorial['periodoHasta']
    imposiciones = jsonHistorial['imposiciones']
    dias = jsonHistorial['dias']
```

Figura 9 Obtención de Datos
[Autor: Cristhian Tohasa]



```
    try:

        connection = pyodbc.connect(
            'DRIVER={SQL Server};SERVER=SERVER1;DATABASE=Registro_Civil_EPN;UID=sa;PWD=smile'
            )
        print("Conexión exitosa...")

        qrInsertRegistroCivil = 'EXEC [SERVER1].[Registro_Civil_EPN].[dbo].[sp_insertarDatosRegistroCivil] \'' + cedula
        print(qrInsertRegistroCivil)

        cursorInsert = connection.cursor()

        cursorInsert.execute(qrInsertRegistroCivil)
        cursorInsert.commit()

        response = 'Insercion Registro Civil exitosa ' + cedula

    except Exception as ex:

        print("Error durante la conexión: {}".format(ex))
        response = 'Error de conexion Registro Civil'

    finally:

        connection.close()  # Se cerró la conexión a la BD.
        print("La conexión Registro Civil ha finalizado.")

    try:
```

Figura 10 Llamado al Procedimiento Almacenado
[Autor: Cristhian Tohasa]

La Figura 11 muestra la arquitectura final del componente en el Proyecto de Integración Curricular. Se puede observar cómo se divide en dos partes: La Base de Datos Distribuida, encargado de la inserción, actualización y eliminado de datos; y la Base de Datos Consolidada que es la encargada de las consultas por parte del usuario. Si se desea observar la arquitectura completa del proyecto TIC, se puede consultar el Anexo III.

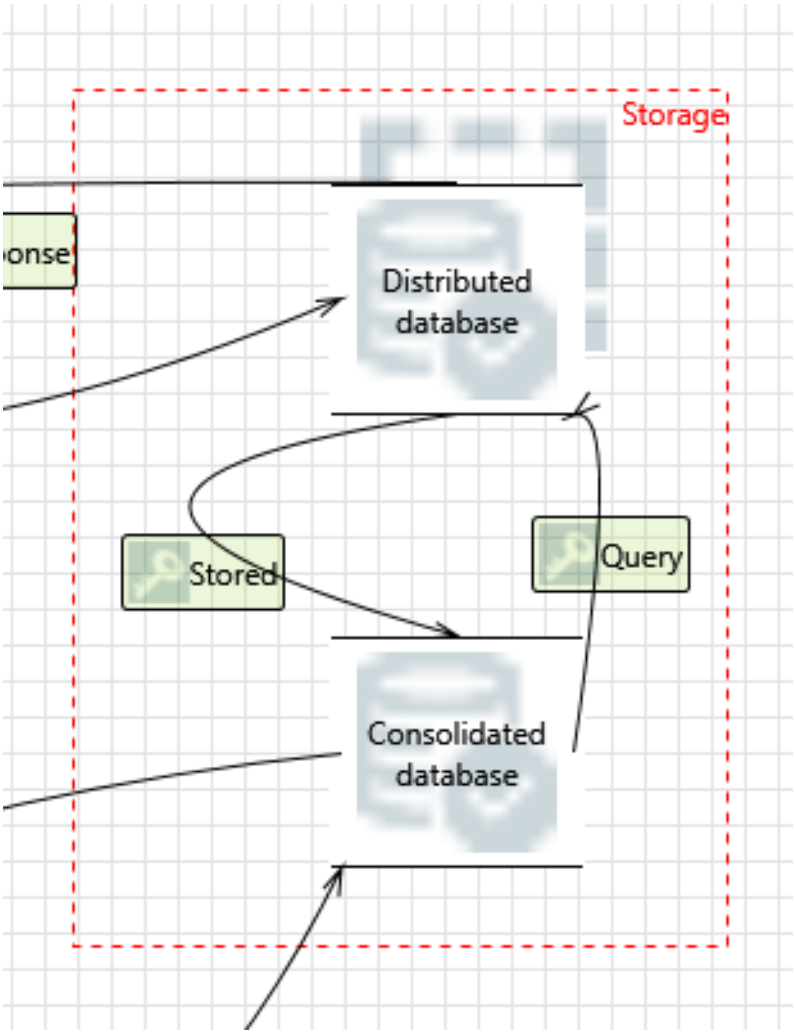


Figura 11 Arquitectura del Componente de Almacenamiento del Proyecto TIC [Autor: Danny Venegas]

3. EVALUACIÓN, CONCLUSIONES Y RECOMENDACIONES

3.1. Pruebas

Se llevaron a cabo dos tipos de pruebas para obtener resultados en el proyecto: las pruebas funcionales y las pruebas de rendimiento. A continuación, se presentarán los resultados obtenidos después de la implementación de ambos tipos de pruebas.

3.1.1. Pruebas Funcionales

Para la prueba funcional del componente se crearon las bases de datos fragmentadas en donde la primera prueba funcional se hará en una inserción.

En la Figura 12 se puede observar el mecanismo de verificación de firmas.

```python
def verificar_xml(datos):
    # Obtiene el número de firmas existentes en el diccionario
    num_signatures = sum([1 for key in datos.keys() if 'signature' in key])

    # Recorre todas las firmas y claves públicas del diccionario y verifica cada una
    for i in range(1, num_signatures+1):
        # Carga la clave pública desde la cadena de texto
        pubkey = rsa.PublicKey.load_pkcs1(base64.b64decode(datos['pubkey{}'.format(i)]))

        # Obtiene la firma digital del cliente
        signature = base64.b64decode(datos['signature{}'.format(i)])

        # Verifica la firma digital del documento XML recibido
        if not verify(str(datos['xml_document']).encode('utf-8'), signature, pubkey):
            print("La firma digital {} no es válida.".format(i))
            return False

    # Si se verificaron todas las firmas sin problemas, devuelve True
    print("Todas las firmas digitales son válidas.")
    return True
```

Figura 12 Verificación de firmas
[Autor: Cristhian Tohasa]

En la Figura 13 se manda a llamar al método POST, en este método se verifican las firmas y se insertan los datos.

```python
@app.route('/recibirCris',methods = ['POST'])
def recibirCris():
    # Recibir el XML como un diccionario
    xml_firmado_danny = request.get_json()
    if verificar_xml(xml_firmado_danny):
        print("SE VERIFICÓ LAS 2 FIRMAS\n----------------\nSe inserta a la base")
        #insert en la base de datos fragmentada
        insertarDatosABase(str(xml_firmado_danny['xml_document']))
    return "Respuesta válida"
```

Figura 13 Método POST
[Autor: Cristhian Tohasa]

Entonces teniendo las bases de datos corriendo en el SERVER 1 y SERVER 2 se manda a hacer una petición en este caso ser hará con Postman Figura 14.

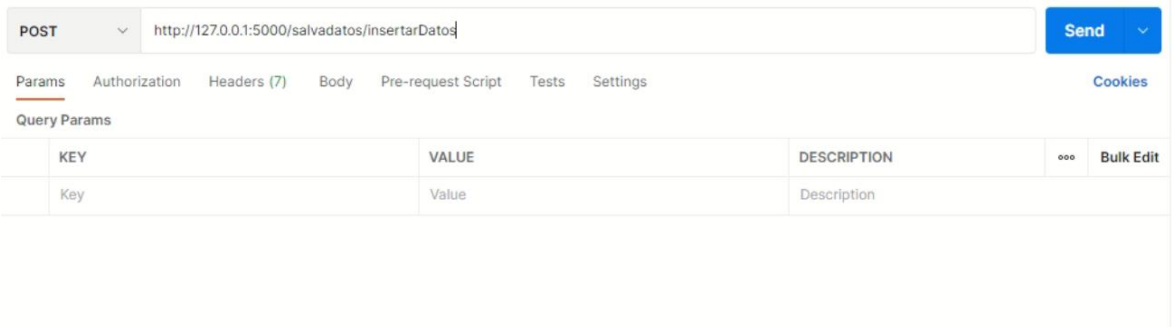


Figura 14 Prueba desde Postman
[Autor: Cristhian Tohasa]

Podemos ver como los datos se insertan exitosamente llamando a un procedimiento almacenado Figura 10, luego de haber verificado las firmas, Figura 12.

En la Figura 15 y en la Figura 16 podemos observar que la petición se realizó exitosamente.

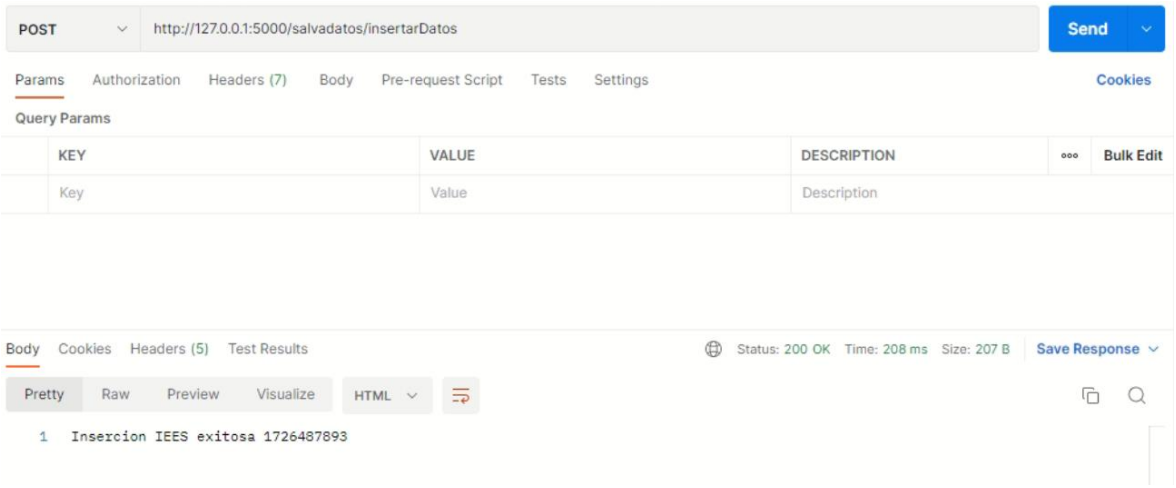


Figura 15 Resultado del Postman
[Autor: Cristhian Tohasa]

Figura 16 Resultado del API REST
[Autor: Cristhian Tohasa]

En la Figura 17 y Figura 18 podemos observar que tanto la base de datos Registro Civil EPN, así como la base de datos IEES EPN tienen los datos insertados.

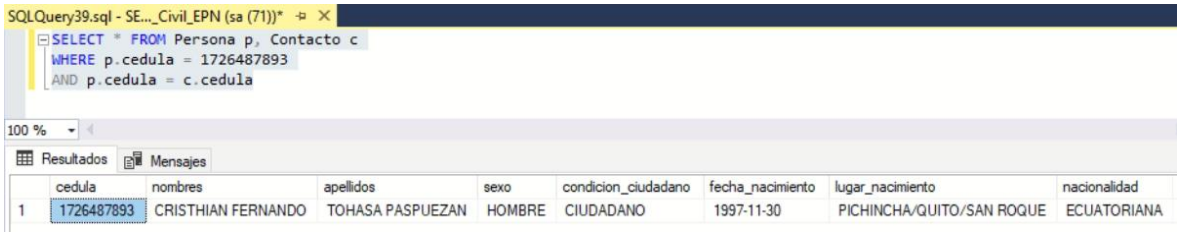


Figura 17 Datos en el SERVER 1 Registro Civil EPN
[Autor: Cristhian Tohasa]

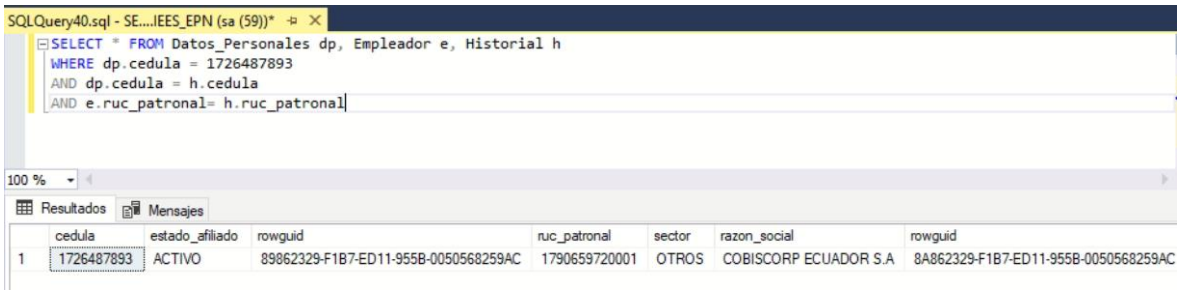


Figura 18 Datos en el SERVER 2 IEES EPN
[Autor: Cristhian Tohasa]

Además, podemos ver como en la base de datos consolidada ya están replicado los datos que se ingresaron Figura 19.

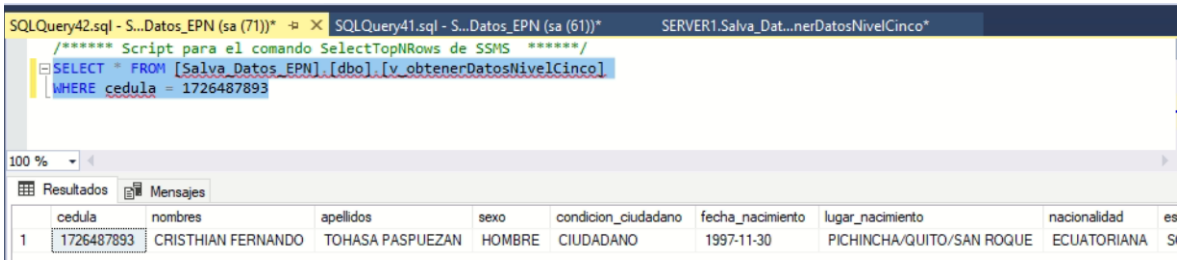


Figura 19 Datos en el SERVER 1 Salva Datos EPN
[Autor: Cristhian Tohasa]

En esta segunda prueba se hará una prueba de consultas que es básicamente el mismo proceso, la diferencia es que se llama a un método GET y esta llama a un vista.

Para este ejemplo llamaremos a la vista de nivel cinco, Figura 20.

```
qrSelectSalvaDatos = 'SELECT * FROM v_obtenerDatosNivelCinco'
print(qrSelectSalvaDatos)

cursorSelect = connection.cursor()

cursorSelect.execute(qrSelectSalvaDatos)

rows = cursorSelect.fetchall()

rows = [tuple(row) for row in rows]
rows = json.dumps(rows)

response = rows
```

Figura 20 Método GET
[Autor: Cristhian Tohasa]

Esta prueba lo haremos a través del Postman Figura 20.

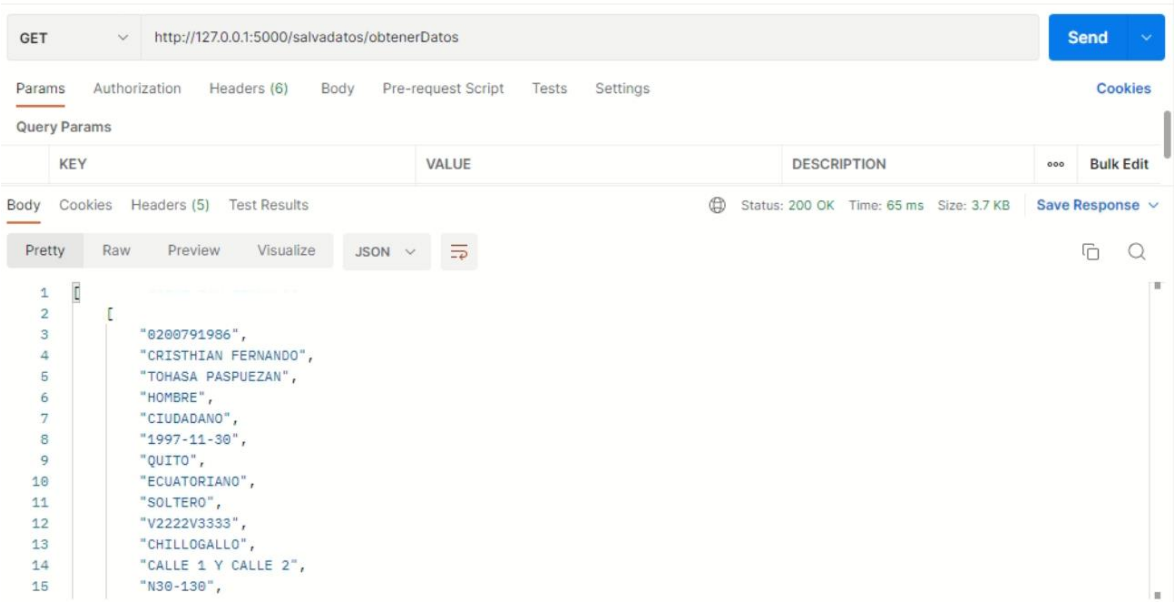


Figura 21 Consulta de Datos a través de Vistas
[Autor: Cristhian Tohasa]

3.1.2. Pruebas de Rendimiento

En esta sección se presentarán los resultados y gráficas obtenidos al realizar las pruebas de rendimiento del sistema. Es importante destacar que la aplicación fue probada de manera local y no fue cargada en un servidor en línea debido a limitaciones económicas del grupo y al hecho de que fue desarrollada en entornos virtualizados llamados VDI proporcionados por la Escuela Politécnica Nacional. Debido a esta limitación, los valores de red como el throughput de red y el ancho de banda, entre otros, no fueron considerados ya que la herramienta de monitoreo de rendimiento de Windows los mostraba como cero o no recolectaba ningún valor.

En la Figura 22 se muestra el proceso para agregar las métricas necesarias con el fin de permitir que la herramienta de monitoreo de rendimiento de Windows recolecte valores durante las pruebas de rendimiento. Este proceso fue crucial para garantizar la recopilación de datos precisos y detallados para el análisis posterior de los resultados de las pruebas.

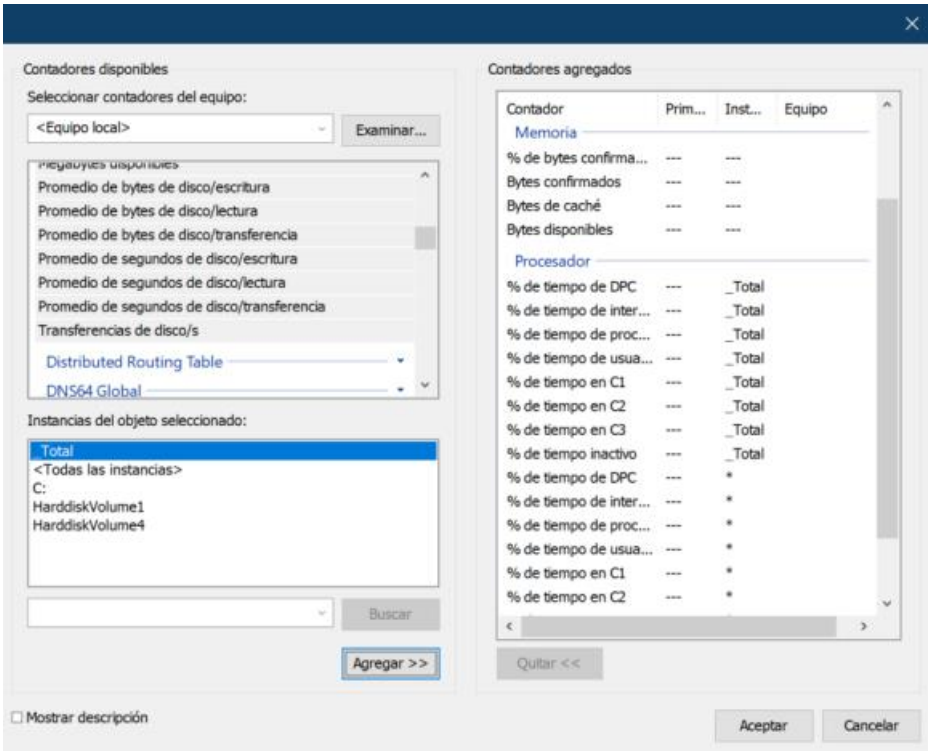


Figura 22 Herramientas de Monitoreo de Windows
[Autor: Cristhian Tohasa]

En la Figura 23 se puede observar el porcentaje de tiempo en el que el disco fisco es utilizado, enfocándonos desde la iteración 29 que es cuando se insertan los datos y desde la iteración 41 que es cuando se hacen las consultas, siendo esta la iteración de más consumo, pero relativamente baja debido al uso de vistas.

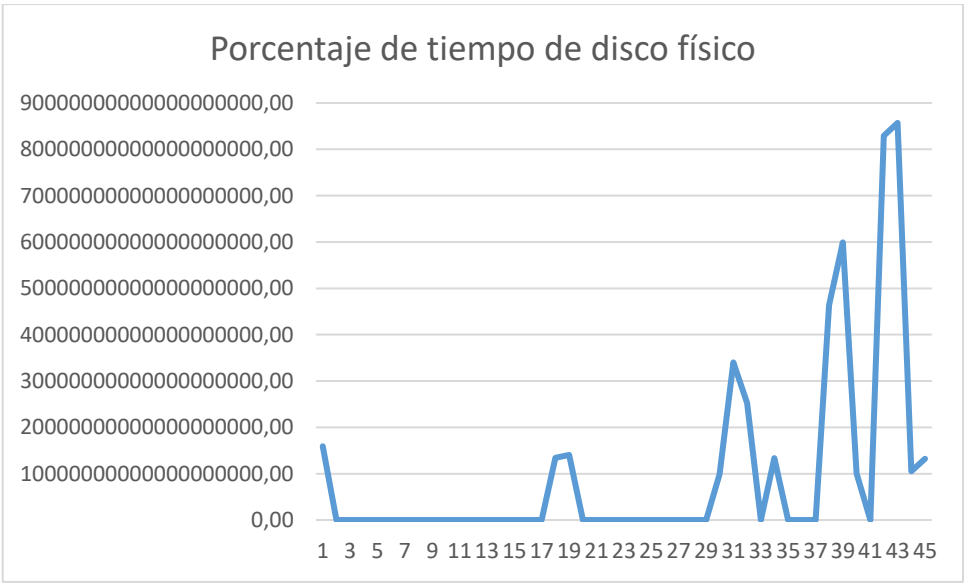


Figura 23 Porcentaje de Tiempo de Disco Físico
[Autor: Cristhian Tohasa]

En la Figura 24 se puede observar el porcentaje de tiempo en el que el disco fisco se usa para ser escrito, en las diferentes iteraciones anteriormente explicadas.

Figura 24 Porcentaje de Tiempo de Escritura
[Autor: Cristhian Tohasa]

En la Figura 25 se puede observar el porcentaje de tiempo en el que el disco fisco se usa para ser leído, en las diferentes iteraciones anteriormente explicadas.

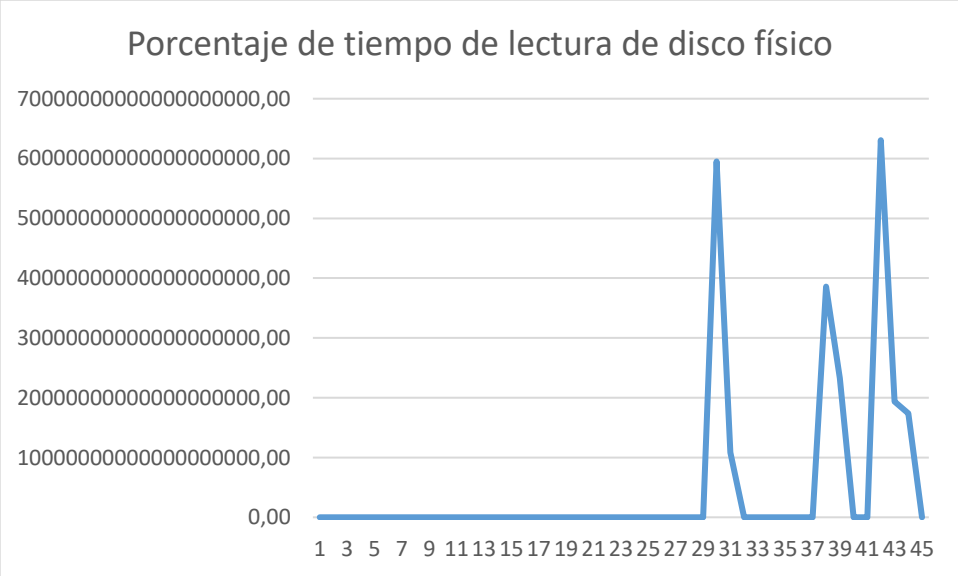


Figura 25 Porcentaje de Tiempo de Lectura
[Autor: Cristhian Tohasa]

En la Figura 26 se puede observar el porcentaje de tiempo en el que el disco lógico se usa para ser escrito, en las diferentes iteraciones anteriormente explicadas. A diferencia del disco físico este no se usa para ser escrito por ende las medidas serás cero.

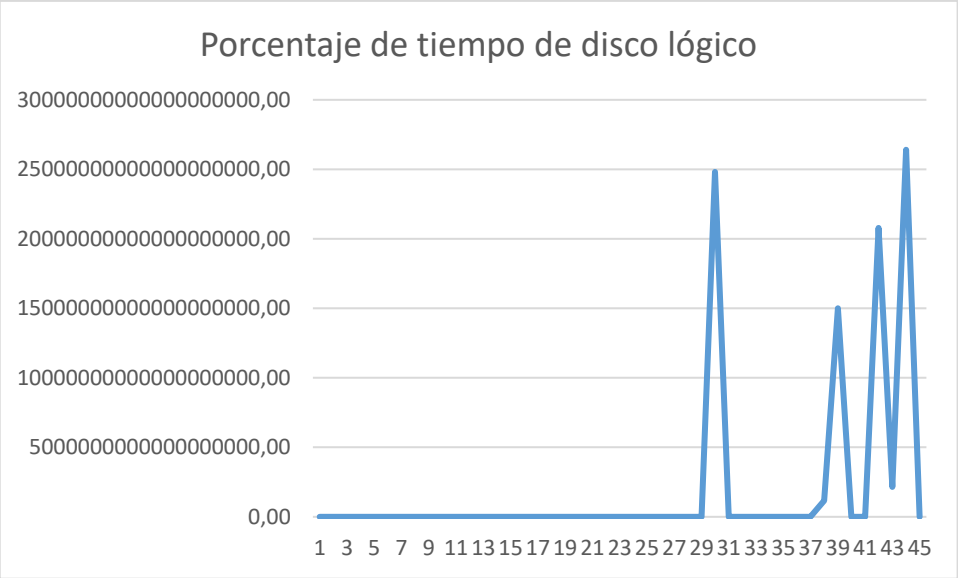


Figura 26 Porcentaje de Tiempo de Disco Lógico
[Autor: Cristhian Tohasa]

En la Figura 27 se puede observar el porcentaje de bytes que se están usando en la memoria durante las iteraciones.

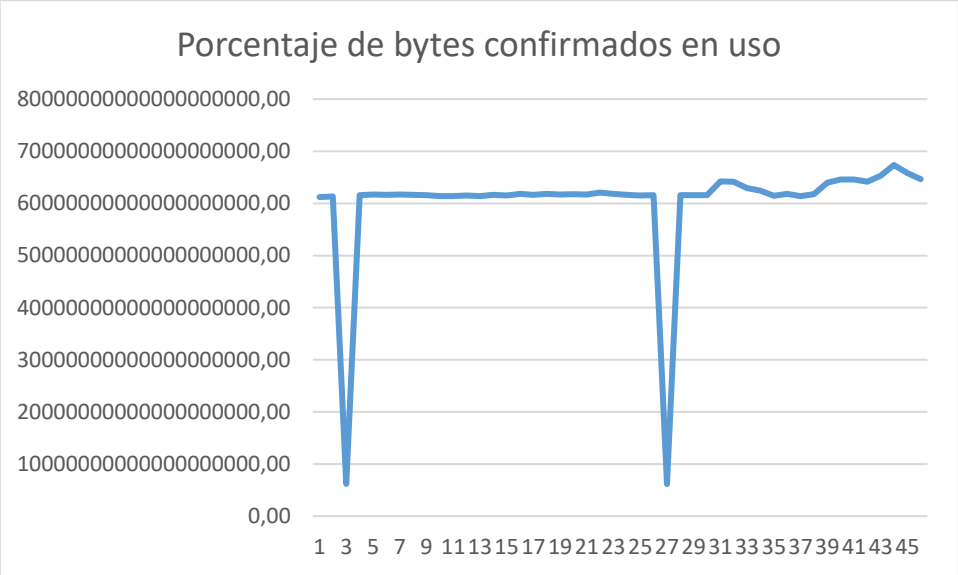


Figura 27 Porcentaje de Bytes en Uso
[Autor: Cristhian Tohasa]

En la Figura 28 se puede observar el porcentaje de bytes disponibles que quedan en la memoria durante las iteraciones.

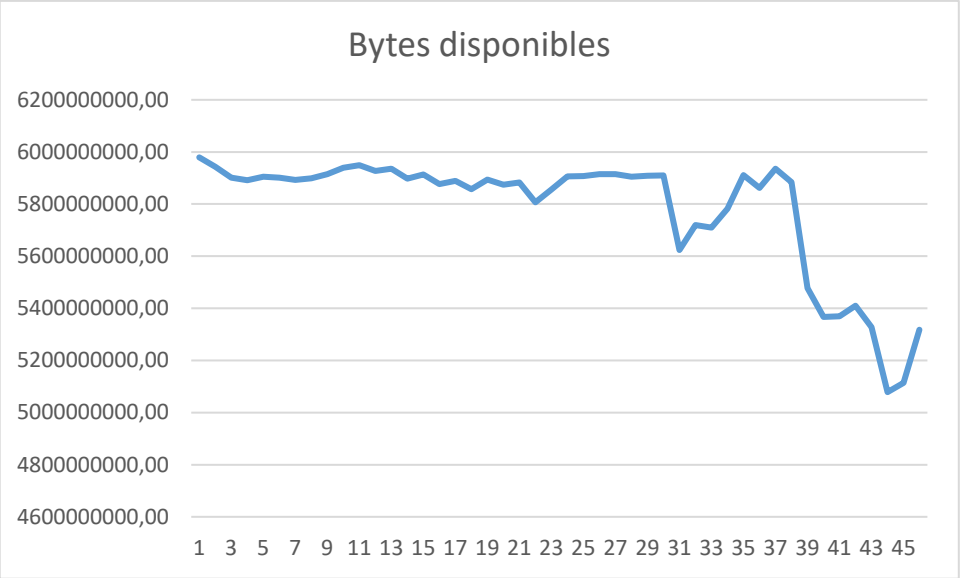


Figura 28 Bytes Disponibles
[Autor: Cristhian Tohasa]

En la Figura 28 se puede observar el porcentaje de tiempo que el procesador está en uso durante las iteraciones del proceso de pruebas.

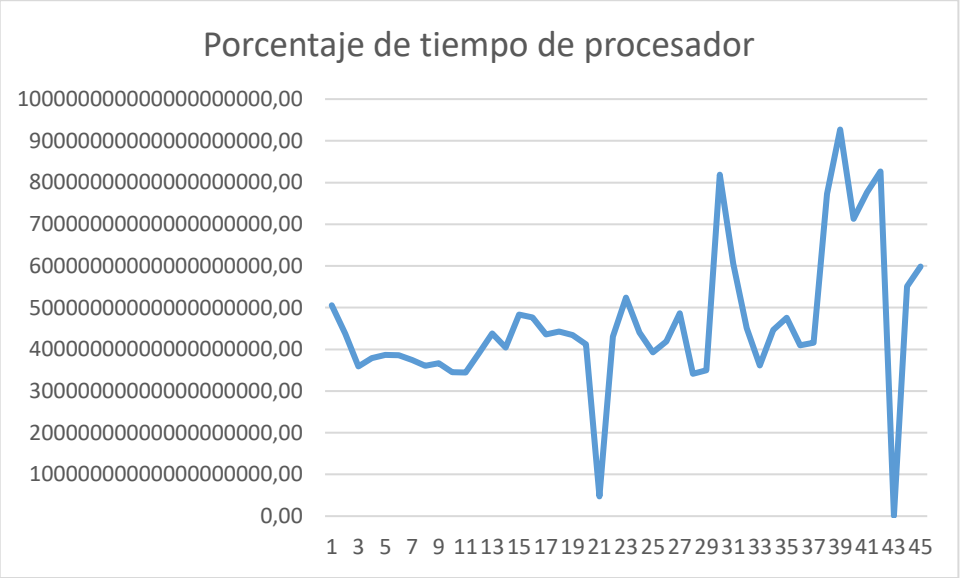


Figura 29 Porcentaje de Tiempo de Procesador
[Autor: Cristhian Tohasa]

En la Figura 29 se puede observar el porcentaje de tiempo que el procesador está en uso por parte del usuario durante las iteraciones del proceso de pruebas.

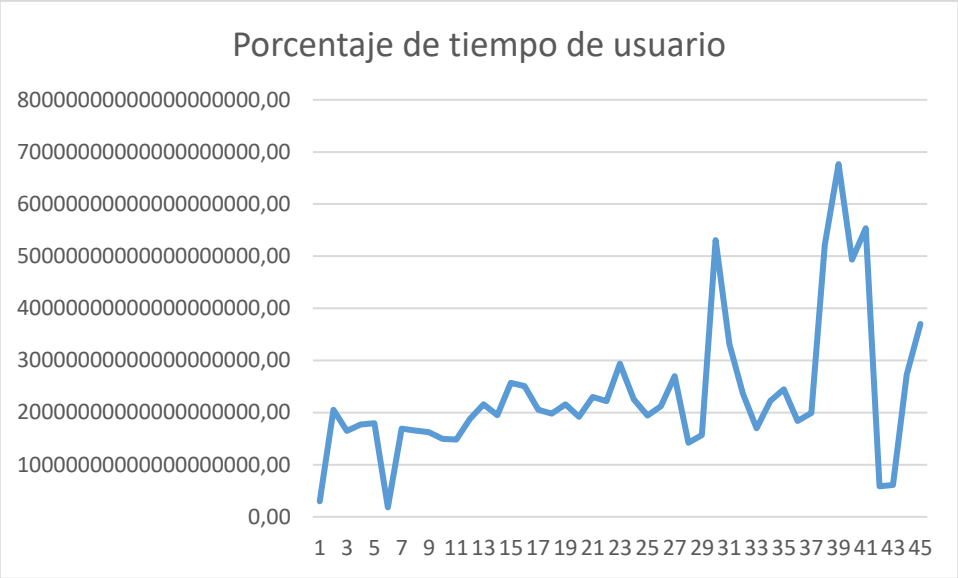


Figura 30 Porcentaje de Tiempo de Usuario
[Autor: Cristhian Tohasa]

En la Figura 30 se puede observar la velocidad del procesador durante las iteraciones del proceso de pruebas.

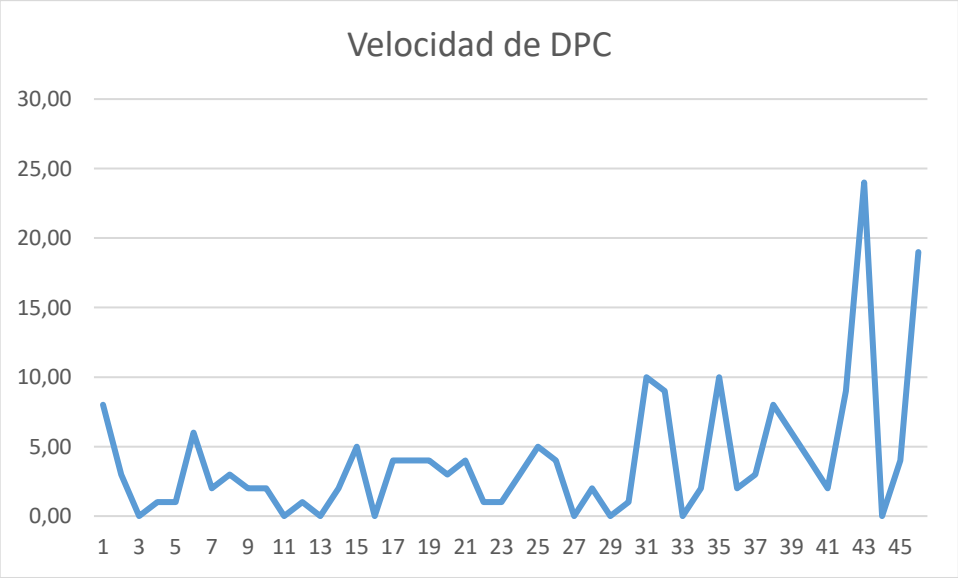


Figura 31 Velocidad de DPC
[Autor: Cristhian Tohasa]

3.1.3. Modelado de Amenazas

Utilizando la herramienta Microsoft Threat Modeling Tool, se realizó una evaluación de la arquitectura presentada en la Figura 11, con el objetivo de identificar posibles amenazas y vulnerabilidades. El resultado de esta evaluación se encuentra detallado en el informe adjunto (ver Anexo III), donde se presentan las principales amenazas identificadas y los controles implementados para mitigarlas.

Estos controles se incorporaron en el proyecto para garantizar un nivel adecuado de seguridad contra amenazas internas y externas, y así lograr una resiliencia adecuada ante estas posibles contingencias.

En la Tabla 10 que se muestra a continuación, se presenta una descripción detallada de las amenazas más relevantes y los respectivos controles implementados para su mitigación.

Se puede encontrar la evaluación completa en el Anexo II.

Tabla 10 Evaluación de Riesgos
[Autor: Cristhian Tohasa]

Nro.	Amenaza	Descripción	Responsable	Riesgo	Opción	Control (Opc = 1)
R9	E.O.Privilege	An attacker may…	Cristhian Tohasa	Alto	1	Limitar el acceso de los usuarios a Query Web Server y validar los datos de entrada para prevenir la ejecución malintencionada de código.
R14	I. Disclosure	Improper data protection…	Cristhian Tohasa	Alto	1	Revisar los permisos de autorización en la base de datos consolidada y aplicar medidas de protección de datos adecuadas para evitar la divulgación no autorizada de información.
R41	D.O. Service	An external agent…	Cristhian Tohasa	1	Alto	Implementar medidas de seguridad para evitar que agentes externos impidan el acceso a la base de datos a través de la frontera de confianza.
R52	Tampering	SQL injection is…	Cristhian Tohasa	1	Muy Alto	Revisar todos los procedimientos que construyen sentencias SQL para detectar y corregir vulnerabilidades de inyección SQL. Validar la entrada de datos y limitar los privilegios de usuario.
R65	Spoofing	Consolidated database may…	Cristhian Tohasa	1	Muy Alto	Utilizar un mecanismo estándar de

						autenticación para identificar
R67	I. Disclosure	mproper data protection…	Cristhian Tohasa	1	Muy Alta	Revisar los permisos de autorización en la base de datos distribuida y aplicar medidas de protección de datos adecuadas para evitar la divulgación no autorizada de información.
R70	E.O. Privilege	Distributed database may…	Cristhian Tohasa	1	Alta	Limitar el acceso a la base de datos distribuida y restringir la ejecución remota de código PKI.

3.2. Discusión de Resultados

Es muy satisfactorio ver que el componente de almacenamiento y acceso a los datos ha sido implementado con éxito. La seguridad de los datos se ha mejorado significativamente debido a la implementación de controles como la verificación de firmas digitales, lo que asegura la integridad de los datos almacenados y protege contra posibles manipulaciones no autorizadas.

Es importante destacar que la seguridad de los datos no depende únicamente de este componente, sino que también se ha mejorado gracias a los otros dos componentes implementados por los otros estudiantes, la recolección y la transferencia de datos. La combinación de estos tres componentes ha proporcionado un sistema completo y robusto para la gestión segura de datos.

Además, la fragmentación de la base de datos en dos nodos ha mejorado la eficiencia del proceso de inserción, actualización y eliminación de datos, ya que se pueden realizar estas operaciones de manera más rápida y eficiente en cada nodo. La sincronización y replicación de la base de datos consolidada también asegura la disponibilidad de los datos en todo momento, lo que es fundamental para garantizar un acceso transparente y autorizado a la información.

En cuanto al uso de procedimientos almacenados y vistas, podemos decir que son herramientas muy útiles en la implementación de soluciones de seguridad para bases de datos. Los procedimientos almacenados permiten encapsular la lógica de negocio en la base de datos, evitando la exposición de dicha lógica a través de consultas directas. Esto ayuda a prevenir ataques como la inyección SQL y elevación de privilegios. Además, al

tener una interfaz definida para acceder a los datos, se puede controlar de manera más efectiva el acceso a la base de datos.

Por otro lado, las vistas pueden utilizarse para presentar una vista parcial de los datos a ciertos usuarios o aplicaciones, lo que ayuda a mantener la privacidad y confidencialidad de la información sensible. Además, las vistas pueden ser utilizadas para simplificar la consulta de datos complejos o para proporcionar una vista personalizada de los datos.

En cuestión de rendimiento, se analizó el desempeño con una herramienta de monitoreo de rendimiento de Windows. Para ello, se llevaron a cabo pruebas de rendimiento en un entorno de prueba controlado, utilizando diferentes herramientas y configuraciones.

Se puede observar que el porcentaje donde se usa el disco es más en la escritura que en la lectura, siendo entendible debido a que cuando se escribe en el disco, su uso es mayor que cuando se leen datos. Además, los bytes utilizados durante la escritura y lectura en la base de datos llegan a sus picos más altos en el proceso. De la misma forma el procesador es más utilizado cuando empiezan estas iteraciones de escritura y lectura en la base de datos. Finalmente, la velocidad del DPC también llega a sus picos más alto cuando estas iteraciones están en proceso.

En cuanto a la interpretación de los resultados obtenidos, podemos decir que el uso del disco es mayor durante las operaciones de escritura en la base de datos, lo que es esperado ya que escribir datos en el disco es más costoso en términos de recursos que leerlos. Además, los picos más altos en el uso de los bytes y del procesador también se observan durante el proceso de escritura y lectura en la base de datos.

Es importante mencionar que aunque se observaron estos picos de actividad durante las iteraciones de escritura y lectura en la base de datos, las medidas obtenidas indican que el proceso no consume una cantidad significativa de recursos de los servidores en los que se alojan las bases de datos. Esto sugiere que el sistema está diseñado de manera eficiente y que puede manejar cargas de trabajo razonables sin afectar el rendimiento del servidor.

Los resultados obtenidos sugieren que el sistema es capaz de manejar eficientemente las operaciones de escritura y lectura en la base de datos, y que está diseñado de manera que no consume recursos significativos del servidor. Sin embargo, es importante continuar monitoreando el rendimiento del sistema en diferentes situaciones y cargas de trabajo para garantizar su óptimo funcionamiento.

En cuanto a la evaluación de riesgos y la implementación de controles que se discutió anteriormente, están directamente relacionados con los objetivos de la tesis. En particular, los controles propuestos ayudarán a garantizar la seguridad y privacidad de la información personal identificable (PII) durante su almacenamiento y acceso transparente, lo cual es uno de los requisitos fundamentales para cumplir con la LOPDP y otras leyes y regulaciones aplicables.

Además, la evaluación de riesgos y la implementación de controles también están alineados con el objetivo específico de implementar un prototipo experimental que cumpla con los lineamientos del GDPR europeo, la LOPDP ecuatoriana, el EGSI y la LOTAIP. Estos lineamientos establecen requisitos específicos de seguridad y privacidad que deben ser cumplidos por cualquier solución de almacenamiento y acceso a PII.

Finalmente, la evaluación del prototipo propuesto en un entorno controlado, tal como se describe en el objetivo específico 4, también se beneficiará de la evaluación de riesgos y la implementación de controles, ya que ayudarán a garantizar que el prototipo tenga una alta resiliencia ante cualquier intento de manipulación externa de PII almacenada en la base de datos.

En resumen, la evaluación de riesgos y la implementación de controles son una parte importante del proceso de cumplimiento de los objetivos específicos de la tesis, y contribuirán a garantizar una preservación segura y acceso transparente a la información personal identificable de acuerdo con las leyes y regulaciones aplicables.

3.3. Conclusiones

- La tesis presentó un enfoque novedoso al integrar el aprendizaje para abordar el desafío de garantizar la preservación y acceso seguro a los datos personales en cumplimiento con la legislación ecuatoriana. El trabajo demuestra la importancia de la colaboración y la interdisciplinariedad en la solución de problemas complejos, especialmente en el campo de la seguridad de la información y protección de datos.
- Los objetivos específicos planteados en la tesis fueron alcanzados de manera satisfactoria. Se realizó un análisis del estado del arte de soluciones de almacenamiento y acceso a datos personales, se identificaron los requisitos funcionales y no funcionales relacionados con la seguridad y privacidad de los datos, se implementó un prototipo experimental que cumple con los estándares internacionales y leyes ecuatorianas, y se evaluó en un entorno controlado.

- La evaluación de cumplimiento de los objetivos específicos permitió identificar fortalezas y debilidades en la implementación del sistema propuesto. En aquellos casos en que no se lograron los resultados esperados, se propusieron posibles respuestas para explicar por qué sucedió esto o las falencias de lo planteado. Esto demuestra la importancia de la evaluación continua y la retroalimentación en los procesos de desarrollo de software.
- El uso de procedimientos almacenados y vistas es una buena práctica en la implementación de soluciones de seguridad para bases de datos. Estas herramientas ayudan a prevenir ataques y a mantener la privacidad y confidencialidad de la información sensible.
- Es fundamental que las organizaciones implementen medidas de seguridad adecuadas para mitigar los riesgos y prevenir ataques cibernéticos que podrían comprometer la integridad de sus sistemas y datos.

3.4. Recomendaciones

- Es importante fomentar la integración curricular en proyectos de investigación, ya que permite a los estudiantes aplicar y mejorar sus habilidades en diferentes áreas, así como trabajar en equipo y desarrollar soluciones complejas.
- Se recomienda seguir una metodología adecuada para la gestión de proyectos, incluyendo la definición clara de objetivos y entregables, la asignación de roles y responsabilidades, y la planificación adecuada de tiempos y recursos.
- Para futuros trabajos, se sugiere explorar el uso de tecnologías emergentes como la inteligencia artificial, el blockchain y la computación en la nube, para mejorar la seguridad y eficiencia en la gestión de datos.
- Es importante tener en cuenta la regulación y normativas existentes en el área de protección de datos personales, y garantizar el cumplimiento de estas en cualquier proyecto que involucre el manejo de información sensible.
- Finalmente, se recomienda llevar a cabo pruebas exhaustivas y evaluaciones de rendimiento en cualquier solución implementada, con el objetivo de garantizar su estabilidad, eficiencia y seguridad.

4. REFERENCIAS BIBLIOGRÁFICAS

[1] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*, vol. 42. Springer Science & Business Media, 2011. doi: 10.1007/978-1-4419-8834-8.

[2] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*. Boston, MA, USA: Pearson, 2016.

[3] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2015.

[4] D. Agrawal, S. Das, D. Elizondo, and A. B. Kulkarni, "A Privacy-Preserving Architecture for Integrating Genomic Databases," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 14, no. 3, pp. 673–687, 2017, doi: 10.1109/TCBB.2016.2573135.

[5] S. K. Ghosh, *Distributed systems: An algorithmic approach*. CRC Press, 2012. doi: 10.1201/b11349.

[6] D. Singhal and V. Kumar, *Distributed database systems: concepts and design*. CRC Press, 2012.

[7] J. Gray and A. Reuter, *Transaction processing: concepts and techniques*. Morgan Kaufmann Publishers Inc., 1993. doi: 10.1016/C2009-0-21196-9.

[8] H. Garcia-Molina, J. D. Ullman, and J. Widom, *Database systems: the complete book*. Prentice Hall, 2008. doi: 10.1145/1364782.1364784.

[9] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*. McGraw-Hill, 2010.

[10] T. Connolly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson, 2014.

[11] E. F. Codd, "A relational model of data for large shared data banks," *Commun ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[12] R. Ramakrishnan and J. Gehrke, *Database Management Systems*. McGraw-Hill Education, 2003.

[13] R. Angles and C. Gutierrez, "The expressive power of graph-structured data: A survey," *ACM SIGMOD Record*, vol. 34, no. 4, pp. 39–48, 2005, doi: 10.1145/1142473.1142477.

[14] K.-U. Sattler and A. Schwering, "A survey of graph database models," *ACM Comput Surv*, vol. 48, no. 3, pp. 1–39, 2016, doi: 10.1145/2845077.

[15] P. Cudré-Mauroux and S. Seufert, "Managing big data with graph structures: Current practices and future directions," *IEEE Internet Comput*, vol. 20, no. 5, pp. 44–53, 2016, doi: 10.1109/MIC.2016.86.

[16] Microsoft Corporation, "Create a Stored Procedure." 2022.

[17] Oracle Corporation, "Using Procedures and Functions." 2022.

[18] C. J. Date, H. Darwen, and D. Foundation, *Databases, Types, and the Relational Model: The Third Manifesto*. Pearson Education, 2006.

[19] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine, 2000.

[20] L. Richardson and S. Ruby, *RESTful web services*. O'Reilly Media, Inc., 2007.

[21] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, "Extensible markup language (XML) 1.0 (5th ed.)," 2008.

[22] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and P. Group, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *PLoS Med*, vol. 6, no. 7, p. e1000097, 2009.

[23] M. J. Page *et al.*, "Epidemiology and reporting characteristics of systematic reviews of biomedical research: a cross-sectional study," *Journal of the American Medical Informatics Association*, vol. 28, no. 10, pp. 2062–2070, 2021.

[24] A. Liberati *et al.*, "The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: explanation and elaboration," *PLoS Med*, vol. 6, no. 7, p. e1000100, 2009.

[25] P. Sutcliffe and A. C. Tricco, "Methodological guidance for systematic reviews of observational epidemiological studies reporting prevalence and cumulative incidence data," *Int J Epidemiol*, vol. 47, no. 1, pp. 223–232, 2018.

[26] R. Xu, J. Joshi, and P. Krishnamurthy, "An Integrated Privacy Preserving Attribute-Based Access Control Framework Supporting Secure Deduplication," *IEEE Trans Dependable Secure Comput*, vol. 18, no. 2, pp. 706–721, 2021, doi: 10.1109/TDSC.2019.2946073.

[27] M. M. Madine *et al.*, "Blockchain for Giving Patients Control Over Their Medical Records," *IEEE Access*, vol. 8, pp. 193102–193115, 2020, doi: 10.1109/ACCESS.2020.3032553.

[28] J. Wei, X. Chen, X. Huang, X. Hu, and W. Susilo, "RS-HABE: Revocable-Storage and Hierarchical Attribute-Based Access Scheme for Secure Sharing of e-Health Records in Public Cloud," *IEEE Trans Dependable Secure Comput*, vol. 18, no. 5, pp. 2301–2315, 2021, doi: 10.1109/TDSC.2019.2947920.

[29] W. Li *et al.*, "Unified Fine-Grained Access Control for Personal Health Records in Cloud Computing," *IEEE J Biomed Health Inform*, vol. 23, no. 3, pp. 1278–1289, 2019, doi: 10.1109/JBHI.2018.2850304.

[30] F. Deng, Y. Wang, L. Peng, H. Xiong, J. Geng, and Z. Qin, "Ciphertext-Policy Attribute-Based Signcryption With Verifiable Outsourced Designcryption for Sharing Personal Health Records," *IEEE Access*, vol. 6, pp. 39473–39486, 2018, doi: 10.1109/ACCESS.2018.2843778.

[31] K. Edemacu, B. Jang, and J. W. Kim, "Efficient and Expressive Access Control With Revocation for Privacy of PHR Based on OBDD Access Structure," *IEEE Access*, vol. 8, pp. 18546–18557, 2020, doi: 10.1109/ACCESS.2020.2968078.

[32] M. M. Madine *et al.*, “Fully Decentralized Multi-Party Consent Management for Secure Sharing of Patient Health Records,” *IEEE Access*, vol. 8, pp. 225777–225791, 2020, doi: 10.1109/ACCESS.2020.3045048.

[33] Q. Li, Y. Zhang, T. Zhang, H. Huang, Y. He, and J. Xiong, “HTAC: Fine-Grained Policy-Hiding and Traceable Access Control in mHealth,” *IEEE Access*, vol. 8, pp. 123430–123439, 2020, doi: 10.1109/ACCESS.2020.3004897.

[34] L. Zhang, T. Zhang, Q. Wu, Y. Mu, and F. Rezaeibagha, “Secure Decentralized Attribute-Based Sharing of Personal Health Records With Blockchain,” *IEEE Internet Things J*, vol. 9, no. 14, pp. 12482–12496, 2022, doi: 10.1109/JIOT.2021.3137240.

[35] H. A. al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, “A Security Model for Preserving the Privacy of Medical Big Data in a Healthcare Cloud Using a Fog Computing Facility With Pairing-Based Cryptography,” *IEEE Access*, vol. 5, pp. 22313–22328, 2017, doi: 10.1109/ACCESS.2017.2757844.

[36] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[37] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2008.

[38] S. Gregor, “Design science and the grounded theory approach,” *Handbook of grounded theory*, vol. 1, pp. 1–16, 2013.

[39] K. Schwaber and J. Sutherland, “The Scrum Guide.” Scrum.org, 2017.

[40] J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business, 2014.

[41] P. S. Foundation, “Python 3.10.0 documentation.” 2022.

[42] A. Sweigart, *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press, 2019.

[43] M. Grinberg, *Flask web development: Developing web applications with Python*. O’Reilly Media, Inc., 2018.

[44] Flask, “About Flask.” 2022.

[45] Q. Software, “Toad Data Modeler.” 2022.

[46] J. McCall, *Database Design and Modeling: Using Toad Data Modeler*. Apress, 2019.

[47] Microsoft, “What is SQL Server?” 2022. [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2022

[48] D. Kellenberger, *Microsoft SQL Server 2019: A Beginner’s Guide*, 2nd ed. McGraw-Hill Education, 2020.

[49] Microsoft, “What is Visual Studio Code?” 2022. [Online]. Available: https://code.visualstudio.com/docs/introvideos/basics

[50] K. Goyal, *Visual Studio Code: The Complete Guide to Master Visual Studio Code for Beginners and Experts*. Independently published, 2021.

[51] Postman, “What is Postman?” 2022.

[52] A. Chauhan, *Learning Postman: Build API Test Cases with Ease*. Packt Publishing, 2021.

[53] Microsoft, “Threat modeling tool overview.” 2017.

[54] Microsoft, “Microsoft Threat Modeling Tool.” 2022.

5. ANEXOS

En los anexos de esta tesis se pueden encontrar información adicional y detallada que complementa el contenido presentado en el cuerpo del trabajo. En particular, se incluyen:

ANEXO I. Estado del Arte

ANEXO II. Reporte de Amenazas

ANEXO III. Evaluación de Riesgos

ANEXO IV. Video Demostrativo

ANEXO V. Código Fuente

ANEXO I

En el siguiente enlace se encuentra el documento inicial del proyecto, el cual ha sido elaborado por el autor de la tesis y presenta un análisis detallado del estado del arte. Este documento establece las bases teóricas y prácticas necesarias para la realización de la investigación y el desarrollo del proyecto de tesis.

Enlace: https://epnecuador-my.sharepoint.com/:f:/g/personal/cristhian_tohasa_epn_edu_ec/EsUFcwfDgmRHjPa0VHa3u4UBLnfIN8Uol0lFSEhXmlSDaQ?e=5CzNra

ANEXO II

En el siguiente enlace se encuentra el documento que describe la herramienta de modelado de amenazas y que muestra las 76 amenazas identificadas por medio de la misma. Este documento es esencial para entender las posibles amenazas a la seguridad en el contexto específico de la tesis.

Enlace: https://epnecuador-my.sharepoint.com/:f:/g/personal/cristhian_tohasa_epn_edu_ec/EooKJVjRxetHssdeXrLRe3wBit0CS9FOUr51czB0EpAvsQ?e=4uF4hW

ANEXO III

El siguiente enlace proporciona acceso al documento resultante del análisis de riesgos del componente de Almacenamiento, el cual incluye información detallada sobre las amenazas identificadas y las contingencias asociadas.

Enlace: https://epnecuador-my.sharepoint.com/:f:/g/personal/cristhian_tohasa_epn_edu_ec/Eq6qeeYaXclPlTwN9qIJdVsBK8WtB4QUvd5KnJ5CssOg1w?e=XMq4rO

ANEXO IV

Se puede acceder al video que ilustra el funcionamiento del prototipo en el enlace que se encuentra a continuación. Este video fue obtenido a partir de la realización del proyecto de integración curricular en Tecnologías de la Información y la Comunicación (TIC).

Enlace: https://epnecuador-my.sharepoint.com/:f:/g/personal/cristhian_tohasa_epn_edu_ec/Egj6_hHT8alNv1q5oHe4ZhwB3g_yZrF6VVDUXhWhK3G6jg?e=Ubp1oa

ANEXO V

El enlace que se proporciona a continuación permite acceder al código fuente de la aplicación integrada de los tres componentes utilizados en el proyecto de integración curricular (TIC). Este código fuente es la fuente de la aplicación completa y se encuentra disponible para su descarga y uso.

Enlace: https://epnecuador-my.sharepoint.com/:f:/g/personal/cristhian_tohasa_epn_edu_ec/Esl2GYToBA1FpZznwumm9MkBXqZpy8IK6E5x2_Xb_V8nTg?e=BcQ7bn