**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Título del trabajo:

# Desarrollo de un Sistema de Comunicación Segura para la Transmisión de Datos Médicos Sensibles

**PROYECTO DE TITULACIÓN**

Previo la obtención del Título de:

**Magíster en ingeniería biomédica**

Presentado por:

Jairo Patricio Huaraca Samaniego

GUAYAQUIL - ECUADOR

Año: 2025

# DEDICATORIA

El presente proyecto lo dedico a mi familia, en especial a mis padres ya que fueron el soporte y apoyo incondicional para poder cumplir con una meta más en mi carrera profesional.

# AGRADECIMIENTOS

Mi más sincero agradecimiento a la Escuela Superior Politécnica del Litoral, por darme la apertura y seguir avanzando con mi carrera profesional, a los docentes de la MIB por los conocimientos impartidos, a mi tutor Washington Velásquez PhD por sus guías y ayuda en el trabajo de titulación, y a mi familia por su apoyo incondicional a lo largo de este proceso de aprendizaje.

# DECLARACIÓN EXPRESA

Yo Jairo Patricio Huaraca Samaniego acuerdo y reconozco que: La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. El o los estudiantes deberán procurar en cualquier caso de cesión de sus derechos patrimoniales incluir una cláusula en la cesión que proteja la vigencia de la licencia aquí concedida a la ESPOL.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, secreto empresarial, derechos patrimoniales de autor sobre software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al autor que existe una innovación potencialmente patentable sobre los

resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 11 de marzo del 2025.


Firmado electrónicamente por:
JAIRO PATRICIO
HUARACA SAMANIEGO

Autor

# EVALUADORES

Firmado electrónicamente por:
WASHINGTON ADRIAN VELASQUEZ VARGAS

**Washington Velásquez PhD**

PROFESOR TUTOR

Firmado electrónicamente por:
FEDERICO XAVIER DOMINGUEZ BONINI

**Federico Domínguez PhD**

PROFESOR EVALUADOR

# RESUMEN

En los sistemas sanitarios, la transmisión de datos médicos sensibles como historias clínicas electrónicas requieren de altos niveles de seguridad, porque se guarda información confidencial y por su naturaleza pueden ser víctima de ataques o manipulación de datos, poniendo en riesgo la vida del paciente. Por lo tanto, el siguiente trabajo tiene como objetivo desarrollar un sistema de comunicación segura para la transmisión de datos médicos, mediante la implementación de mecanismos de seguridad, cumpliendo las recomendaciones de las normativas de protección de datos HIPAA y GDPR.

La arquitectura empleada se basa en contenedores FIWARE, que cuenta con mecanismos de seguridad como el cifrado AES, autenticación de dos factores a través de una contraseña más un token, y para la autorización se usó el protocolo OAuth2 que se basa en la gestión de roles. Las pruebas realizadas para verificar el rendimiento y la seguridad dieron como resultado que el sistema funciona de manera óptima, ya que los tiempos de respuesta son mínimos, demostró robustes ante ataques de fuerza bruta, e igualmente brindó confiabilidad ya que puede procesar una gran cantidad de peticiones sin llegar a saturarse, optimizando el uso de recursos.

El sistema de comunicación es una solución segura ya que cumple con las principales recomendaciones de la HIPAA y GDPR, como implementación de cifrado, métodos de autenticación, autorización, y pruebas de seguridad para garantizar la integridad, disponibilidad y confiabilidad de los datos médicos sensibles.

**Palabras Clave:** Mecanismos de seguridad, históricas clínicas electrónicas, contendores FIWARE, HIPAA, GDPR.

# *ABSTRACT*

*In healthcare systems, the transmission of sensitive medical data, such as electronic health records, requires high levels of security because they store confidential information and, due to their nature, can be targeted for attacks or data manipulation, putting patients' lives at risk. Therefore, this work aims to develop a secure communication system for the transmission of medical data by implementing security mechanisms that comply with the recommendations of HIPAA and GDPR data protection regulations.*

*The architecture used is based on FIWARE containers, which include security mechanisms such as AES encryption, two-factor authentication through a password plus a token, and OAuth2 protocol for authorization, which is based on role management. Performance and security tests showed that the system operates optimally, as response times are minimal, it demonstrated robustness against brute-force attacks, and it provided reliability by processing a large number of requests without becoming overloaded, optimizing resource usage.*

*The communication system is a secure solution as it meets the key recommendations of HIPAA and GDPR, such as the implementation of encryption, authentication methods, authorization, and security testing to ensure the integrity, availability, and reliability of sensitive medical data.*



***Keywords**: security mechanisms, electronic health records, FIWARE containers, HIPAA, GDPR.*

# ÍNDICE GENERAL

RESUMEN ..... I

*ABSTRACT* ..... II

ÍNDICE GENERAL ..... III

ABREVIATURAS ..... VI

SIMBOLOGÍA ..... VII

ÍNDICE DE FIGURAS ..... VIII

ÍNDICE DE TABLAS ..... IX

ÍNDICE DE ALGORITMOS ..... X

INTRODUCCIÓN ..... XI

CAPÍTULO 1 ..... 1

1. Descripción del problema ..... 1

1.1 Identificación del problema ..... 1

1.2 Justificación del problema ..... 2

1.3 Objetivos ..... 5

1.3.1 Objetivo General ..... 5

1.3.2 Objetivos Específicos ..... 5

1.4 Estado del arte ..... 5

1.4.1 Plataforma FIWARE ..... 7

1.4.2 Seguridad en los sistemas de comunicación ..... 8

1.5 Resultados esperados ..... 10

1.6 Alcance y limitaciones del proyecto ..... 11

1.7 Impacto social y tecnológico ..... 12

CAPÍTULO 2 ..... 13

2. Marco teórico ..... 13

2.1 Sistemas de comunicación ..... 13
2.2 Seguridad en los sistemas de comunicación ..... 13
2.2.1 Vulnerabilidades ..... 14
2.2.2 Ataques ..... 14
2.3 Protocolos de encriptación ..... 15
2.4 Métodos de autenticación ..... 16
2.5 Datos médicos sensibles ..... 17
2.6 Historias clínicas electrónicas ..... 17
2.7 Normativas para la seguridad de datos personales ..... 18
2.7.1 HIPAA ..... 18
2.7.2 GDPR ..... 18
CAPÍTULO 3 ..... 20
3. Metodología ..... 20
3.1 Diseño del sistema de comunicación ..... 20
3.1.1 Arquitectura del sistema de comunicación ..... 21
3.1.2 Usuarios o actores ..... 22
3.1.3 Políticas de usos y servicios ..... 22
3.2 Infraestructura tecnológica ..... 23
3.2.1 Fiware Orion (Context Brocker) ..... 23
3.2.2 Fiware Keyrock ..... 24
3.2.3 Fiware Wilma pep proxy ..... 24
3.2.4 Mongo DB ..... 25
3.2.5 MYSQL ..... 26
3.3 Implementación de protocolo de encriptación ..... 27
3.4 Aplicación de métodos de autenticación y autorización ..... 28
3.5 Desarrollo del sistema de comunicación ..... 29

3.5.1 Entidad para el sistema de comunicación ..... 29
3.5.2 Flujo del sistema de comunicación ..... 30
3.5.3 Plataforma de interacción con el usuario ..... 33
3.5.4 Configuración del entorno de prueba ..... 33
3.6 Métricas de evaluación ..... 34
CAPÍTULO 4 ..... 35
4. Resultados ..... 35
4.1 Funcionamiento del sistema de comunicación ..... 35
4.2 Escenarios de evaluación ..... 40
4.3 Pruebas en base a métricas de evaluación ..... 41
4.4 Ejecución de pruebas de seguridad ..... 44
4.4.1 Pruebas en el cifrado AES ..... 44
4.4.2 Pruebas de autenticación y autorización ..... 45
4.4.3 Sobrecargo del sistema ..... 46
4.5 Evaluación en base a las normativas de protección de datos ..... 48
4.6 Análisis de resultados ..... 50
Conclusiones ..... 52
Recomendaciones ..... 54
Bibliografía ..... 55
Anexos ..... 60
Anexo 1: Archivo docker-compose ..... 60
Anexo 2: Comprobación de contenedores en la misma red ..... 63
Anexo 3: Modelo de HCE en formato JSON ..... 64
Anexo 4: Algoritmo de ataque de fuerza bruta ..... 65
Anexo 5: Repositorio ..... 66

# ABREVIATURAS

| | |
|---|---|
| TICs | Tecnologías de la información y comunicación |
| HCE | Historia clínica electrónica |
| JSON | Notación de objetos JavaScript |
| AES | Estándar de encriptación Avanzado |
| 2FA | Autenticación de dos factores |
| HTTP | Protocolo de transferencia de hipertexto |
| DoS | Denegación de servicio |
| URL | Localizador uniforme de recursos |
| HIPAA | Ley de Portabilidad y Responsabilidad de Seguros de Salud |
| GDPR | Reglamento General de Protección de Datos |

# SIMBOLOGÍA

| | |
|---|---|
| Ms | Milisegundo |
| P/s | Paquetes por segundo |
| CPU | Unidad de procesamiento central |
| MEM | Memoria RAM |
| ID | Identificación |
| Min | Mínimo |
| Máx | Máximo |
| Prom | Promedio |

# ÍNDICE DE FIGURAS

Figura 3.1 Arquitectura del sistema de comunicación ..... 21
Figura 3.2 Fiware orion ..... 23
Figura 3.3 Flujo de Keyrock ..... 24
Figura 3.4 Combinación de pep-proxy y keyrock ..... 25
Figura 3.5 Almacenamiento de datos MySQL ..... 27
Figura 3.6 Credenciales OAuth2 ..... 28
Figura 3.7 Entidades de Orion ..... 30
Figura 3.8 Autenticación de usuarios por Keyrock ..... 31
Figura 3.9 Validación del token por Wilma pep-proxy ..... 31
Figura 3.10 Flujo de comunicación de cada usuario con el sistema ..... 32
Figura 4.1 Contenedores en Docker ..... 35
Figura 4.2 Servicio FIWARE Orion ..... 36
Figura 4.3 Componente Keyrock ..... 36
Figura 4.4 Inicialización de la aplicación por Visual Studio Code ..... 37
Figura 4.5 Menú del administrador del sistema ..... 37
Figura 4.6 Menú del médico ..... 38
Figura 4.7 Menú del paciente ..... 38
Figura 4.8 Almacenamiento de tokens ..... 39
Figura 4.9 Creación de una HCE ..... 39
Figura 4.10 HCE almacenadas y cifradas en MongoDB ..... 40
Figura 4.11 Tiempos de respuesta del escenario 1 ..... 42
Figura 4.12 Recursos empleados en el escenario 1 ..... 42
Figura 4.13 Tiempos de respuesta del escenario 2 ..... 43
Figura 4.14 Recursos empleados en el escenario 2 ..... 44
Figura 4.15 Ataque de fuerza bruta ..... 45
Figura 4.16 Autenticación y autorización Keyrock ..... 46
Figura 4.17 Autenticación y autorización Wilma-pep-proxy ..... 46
Figura 4.18 Recursos empleados en la prueba DoS ..... 47
Figura 4.19 Tiempos de respuesta en la prueba DoS ..... 48

# ÍNDICE DE TABLAS

Tabla 3.1 Acciones de cada usuario según su rol ..... 32
Tabla 4.1 Parámetros del escenario 1 ..... 40
Tabla 4.2 Parámetros del escenario 2 ..... 41
Tabla 4.3 Resultados del escenario 1 ..... 41
Tabla 4.4 Resultados del escenario 2 ..... 43
Tabla 4.5 Tiempos de cifrado y descifrado del sistema ..... 44
Tabla 4.6 Resultados prueba DoS ..... 47

# ÍNDICE DE ALGORITMOS

Algoritmo 3.1 Configuración de credenciales en Wilma pep-proxy ..... 29

# INTRODUCCIÓN

En los últimos años, el avance de las tecnologías de la información y comunicación (TICs), ha dado paso a una evolución en diversos sectores estratégicos como educación, transporte, finanzas y en especial el campo de la salud [1]. En este contexto, la transmisión de datos médicos sensibles se ha convertido en una necesidad crítica debido a que el medio o canal por el que se vaya a transmitir debe garantizar la confidencialidad e integridad de los datos del paciente, cumpliendo normativas que rigen sobre el cuidado de los datos personales [2].

De la misma forma los datos médicos sensibles son transmitidos por historias clínicas electrónicas (HCE) [3], ya que tienen la facilidad de almacenar y administrar una gran cantidad de información, además de que pueden ser intercambiables entre sistemas de diferentes centros de salud [4].

A pesar de los avances tecnológicos en el campo médico, la seguridad en la transmisión de datos médicos sigue siendo una preocupación constante, tanto para el personal médico y los pacientes, ya que según el Cyber Readiness Institute el índice de ataques dirigidos por infecciones de Malware ha aumentado en un 117% [5]. La falta de técnicas de seguridad que brinden un cuidado optimo y robusto de los datos, ha dejado vulnerables muchos sistemas de salud, lo que ha generado la exposición a ciberataques y vulnerabilidades del sistema [6].

Por lo tanto, el presente trabajo de tesis tiene como objetivo diseñar un sistema de comunicación segura para la transmisión de datos médicos sensibles, utilizando protocolos de encriptación robusto y métodos avanzados de autenticación y autorización para el acceso a la información médica, de manera que se busca integrar soluciones para garantizar la confidencialidad e integridad de los datos médicos sensibles, conforme a las normativas de protección de datos como la GDPR [7] o HIPAA [8].

El proyecto no solo aportará con una solución en el campo de la seguridad de la información en sistemas de salud para la transmisión de datos médicos sensibles, sino que también puede ayudar a mejorar a que los pacientes tengan una mejor aceptación a los sistemas de salud, lo que facilitaría la adopción e implementación de los sistemas de telemedicina en los centros de salud.

La presente tesis está compuesta por cuatro capítulos. En el primer capítulo se detalla la identificación del problema, justificación y los objetivos que se va a cumplir en el trabajo, principalmente. El segundo capítulo ofrece una revisión teórica de los conceptos claves para el diseño de un sistema de comunicación seguro para transmitir datos médicos sensibles. El tercer capítulo aborda la metodología utilizada para el diseño. En el cuarto capitulo se muestran los resultados obtenidos en el diseño en base a pruebas de funcionamiento y evaluaciones. Y finalmente se presentan las conclusiones que demuestra la factibilidad del sistema de comunicación en cuanto a la seguridad de datos mediante mecanismos de cifrado, autenticación y autorización, bajo las recomendaciones de las normativas HIPAA o GDPR.

# CAPÍTULO 1

## 1. DESCRIPCIÓN DEL PROBLEMA

### 1.1 Identificación del problema

En el contexto de la telemedicina, la transmisión de datos médicos sensibles, como historiales clínicos, diagnósticos, imágenes médicas y otros datos personales, se realiza a través de redes digitales [6]. Esta transmisión debe garantizar la confidencialidad, integridad y disponibilidad de los datos para asegurar la confianza de los pacientes y cumplir con las normativas de protección de datos, como la Ley General de Protección de Datos (LGPD), el Reglamento General de Protección de Datos (GDPR)[1] y la Health Insurance Portability and Accountability Act (HIPAA)[2], sin embargo, en la actualidad los datos médicos son un objetivo para los ciberdelincuentes ya que esta clase de datos son valiosos con los cuales se puede cometer extorsiones [2], fraudes [9], llevando a consecuencias devastadores a tal punto de poner en riesgo la vida del paciente, según un estudio de la agencia de ciberseguridad de la Unión Europea (Enisa por sus siglas en inglés) [10], en 2023 se evidenció que el sector de la salud es uno de los más afectados por los ciberataques, registrando un 8% de los incidentes de ciberseguridad, donde las principales vulnerabilidades fueron por malas configuraciones con un 68%, seguido de errores humanos con un 16%; en cuanto a los ciberataques, ransomware fue el más utilizado para perpetrar los sistemas de salud con un 54% de los incidentes registrados, seguido de robo de datos con un 46%, por lo que la implementación de sistemas de telemedicina a menudo enfrenta varios desafíos de seguridad y privacidad, tales como:

- **Ciberataques y Vulnerabilidades:** Como indican las estadísticas, los sistemas de telemedicina son susceptibles a ciberataques como intercepciones de datos, ataques Man-in-the-middle, Phishing y Ransomware. La falta de mecanismos de encriptación robustos y protocolos de seguridad puede exponer datos sensibles a actores malintencionados.

[1] https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679

[2] https://hhs.gov/hipaa/for-professionals/security/index.html

- **Protección de Datos en Tránsito y en Reposo:** Asegurar que los datos médicos estén protegidos tanto durante su transmisión a través de redes públicas o privadas como cuando se almacenan en servidores remotos o dispositivos locales es crucial. Sin una encriptación adecuada y controles de acceso, los datos pueden ser vulnerables a accesos no autorizados y brechas de seguridad.
- **Autenticación y Autorización:** La autenticación de usuarios y la autorización de accesos son aspectos críticos en telemedicina. Sistemas débiles de autenticación pueden permitir accesos no autorizados, lo que compromete la privacidad del paciente y la integridad de los datos.
- **Cumplimiento Normativo:** Los sistemas de telemedicina deben cumplir con estrictas regulaciones de protección de datos. La falta de cumplimiento puede resultar en sanciones legales y pérdida de confianza por parte de los usuarios.
- **Interoperabilidad y Compatibilidad:** Asegurar que los sistemas de comunicación sean compatibles e interoperables con otros sistemas de salud es un desafío técnico. Además, estos sistemas deben mantener altos estándares de seguridad sin comprometer la funcionalidad.

## 1.2 Justificación del problema

Los avances de las TICs [11], permitieron que los centros de salud tanto públicos como privados estén adoptando el uso de Historias Clínicas Electrónicas (HCE por sus siglas en inglés) las cuales pueden almacenar una gran cantidad de datos como diagnósticos, pruebas de rutinas, imágenes médicas, ayudando a manejar de una manera más eficiente los datos de los pacientes, sin embargo, a la par existieron desafíos en cuanto al resguardo de la seguridad de la información de los pacientes ya que en el sector de la salud los ciberataques son más susceptibles debido a que la información de un paciente puede ser muy valiosa para los ciberdelincuentes ya que de las mismas se pueden generar robos de identidades, fraudes o peor aún atentar con la vida del paciente [2].

Una de las ventajas que permite el uso de la telemedicina son los controles o monitoreos remotos, y para lo cual se debe contar con un sistema de comunicación que garantice la seguridad de los datos. En ciertos casos este hecho no se cumple tal y como aconteció en un estudio donde se analiza la debilidad que pueden tener la seguridad de los datos

en el diseño de un equipo médico destinado al monitoreo de profundidad de anestesia, aquí los autores evaluaron escenarios de ataques que pueden comprometer la dosificación de anestesia y la privacidad del paciente, concluyendo que la mejor opción para la seguridad del paciente es blindar el sistema minimizando las vías de conectividad, lo que significa suprimir la aplicación de acceso remoto [12]. Así como paso en el estudio anterior pueden existir vulnerabilidades en los sistemas de comunicación que utilizan los centros de salud operacionales como fue el caso de algunos hospitales públicos del Reino Unido que debido a que contaban con sistemas obsoletos y una escasa seguridad, sufrieron un ataque de Ransomware mejor conocido como WannaCry, dejando inoperables todos los sistemas, a tal punto que se tuvo que cancelar cirugías programadas y el personal sanitario ni siquiera pudo acceder a las historias clínicas de los pacientes [13].

Al incrementarse los sistemas de comunicaciones en el ámbito sanitario, ha dado paso a nuevas políticas al momento de la aplicación en la consulta médica, en donde el problema latente es la gran cantidad de información a tratar que es sensible y confidencial, y precisamente por esto las vulnerabilidades que presentan estos sistemas son aprovechadas por los ciberdelincuentes. A medida que la necesidad del uso de sistemas de comunicaciones ha aumentado, el índice de ataques dirigidos por infecciones de malware creció en un 117% según Cyber Readiness Institute [5]. Por lo que recomiendan prácticas como la familiarización con la telemedicina en cuanto a protocolos, comunicación, obtención de licencias y normativas de la ley HIPAA; ser proactivo con la creación de planes de respuesta ante incidentes; ser coherente con el uso de estas plataformas ya que las mismas son compatibles con la ley HIPAA e informarse sobre conceptos básicos de ciberseguridad como el uso de contraseñas, actualizaciones de software, vulnerabilidades e intercambio de datos a través de dispositivos [5].

Como ya se dijo antes en sistemas de comunicaciones para la transmisión de datos médicos se maneja una gran cantidad de información, y hay ocasiones en donde los desarrolladores de estos sistemas desconocen sobre la privacidad de datos y carecen de conocimiento para desarrollar el sistema y así garantizar la confidencialidad y privacidad de los datos [14]. Por esta razón para el procesamiento y privacidad de datos médicos, los sistemas pueden regirse a las siguientes leyes:

- LGPD: su objetivo principal es el procesamiento de datos a través de controladores y procesadores [14].
- GDPR: brinda amparo legal para un mayor control y seguridad sobre los datos personales digitales en ambientes de telemedicina. GDPR se lo aplica de carácter obligatorio en la Unión Europea [7].
- HIPAA: es una ley estadounidense que regula y protege los datos personales en ambientes sanitarios para su transmisión electrónica [8].

En el Ecuador para la protección de los datos en el ámbito de salud rige la Ley Orgánica de Protección de Datos Personales (LOPDP) cuyo fin es garantizar el ejercicio del derecho a la protección de los datos personales, estos incluyen el acceso y la decisión a tomar sobre los datos, así como su protección, para ello se estipularon principios, derechos, obligaciones y mecanismos de tutela [15].

Los sistemas de comunicación en entornos sanitarios deben incorporar herramientas y tecnologías para prevenir vulnerabilidades en sus sistemas, así como encriptación tanto para la transmisión y recepción de datos con el fin de asegurar la integridad y confidencialidad de los datos, procesos de autenticación y autorización para el acceso de personal autorizado previamente identificado con su usuario y contraseña, para que los mismos tengan acceso a la información según su jerarquía, y todo esto se podría complementar con medidas preventivas evaluando los riesgos del sistema a través de pruebas de penetración para detectar y corregir vulnerabilidades [2].

Por lo tanto, la finalidad de este proyecto es diseñar y desarrollar un sistema de comunicación que permita la transmisión de datos médicos sensibles, el diseño se lo realizara con el propósito de aportar con una posible solución a la falta de manejo de infraestructura en TICs en los centros de salud del Ecuador para el manejo de esta clase de datos, lo que implicaría que la información sanitaria de algún paciente puede ser accedida por personas no autorizadas [16]. El diseño garantizaría la confidencialidad e integridad de datos médicos mediante la integración de herramientas o técnicas del área tecnológica las cuales trabajaran en conjunto con leyes o reglamentos del área legislativa, ya que la información a transmitir va a ser utilizada y tratada por profesionales de la salud, y de esa forma se ofrecerá a los pacientes la confianza necesaria sobre la seguridad de sus datos personales.

Con el diseño del sistema se pretende enviar datos médicos a través de un sistema de comunicaciones, para lo cual previo al envío los datos deben contar con un protocolo de encriptación para que estos no sean vulnerados, así también se utilizaran métodos para la autenticación de las personas que vayan a utilizar el sistema y definir el tipo de acceso que tendrán sobre los datos médicos.
Para garantizar la fiabilidad del sistema se realizarán pruebas preventivas para detectar posibles vulnerabilidades y corregirlas basándose en leyes o normativas para la protección de datos como la LGPD, HIPAA, GDPR, LOPDP.

## 1.3 Objetivos

### 1.3.1 Objetivo General

Desarrollar un sistema de comunicación segura mediante la implementación de mecanismos de seguridad cumpliendo las normativas internacionales de protección de datos, para que garantice la confidencialidad, integridad y disponibilidad de los datos médicos sensibles transmitidos.

### 1.3.2 Objetivos Específicos

- Implementar un protocolo de encriptación robusto con la aplicación de algoritmos de cifrado para la protección de los datos médicos tanto en tránsito como en reposo.
- Emplear métodos avanzados de autenticación y autorización por medio de la implementación de mecanismos de verificación y control de acceso asegurando que solo los usuarios autorizados tengan acceso controlado y seguro a los datos médicos.
- Evaluar la seguridad y rendimiento del sistema mediante métricas de evaluación para un análisis de efectividad en la protección de datos sensibles en conformidad con las normativas de protección de datos como la HIPAA y el GDPR.

## 1.4 Estado del arte

Los sistemas de comunicación que transmiten HCE presentan varios desafíos entre los cuales destacan la confidencialidad y privacidad de los datos clínicos [6].

En [3], aborda la implementación de HCE en todos los establecimientos públicos de salud bajo el amparo de la ley de protección de datos personales donde la confidencialidad y privacidad de los datos son esenciales en estos entornos, para lo cual se han ajustado normativas considerando a las HCE como datos sensibles para garantizar la seguridad e integridad de los mismos, por lo que los centros de salud deberán establecer los mecanismos de seguridad y accesibilidad de las HCE, así mismo en [17], detallan el proceso de evaluación de HCE vs historias clínicas de papel para la prescripción de medicamentos en pacientes con neumonía, considerando una muestra de 149 pacientes con historias clínicas de papel y 149 pacientes con HCE, en donde la evaluación se la realizo con el cuadrado de Pearson, dando como resultado que las HCE mejoraron las prescripciones de medicamentos reduciendo de forma significativa el riesgo de errores en cuanto a la legibilidad, registros, detalles de medicamentos, y constancia de firmas y sellos; demostrando de esa manera la importancia que hoy en día conlleva el uso de HCE en los ambientes sanitarios.

Para transmitir información de HCE es necesario contar con un canal de comunicación el cual soporte el flujo de información y además que garantice la confidencialidad y privacidad de los datos, para ello se analizará algunos artículos que involucren el diseño de un canal de comunicación seguro para la transmisión de datos médicos, basados en este contexto, Huang [18] propuso un canal de comunicación a través del diseño de un sistema de atención sanitaria que integra tecnologías como IoT, computación en la nube y redes inalámbricas corporales (WBAN por sus siglas en inglés), para mejorar la atención sanitaria mediante la recolección, transmisión y análisis de datos médicos en tiempo real. Para ello empleo una arquitectura WBAN con sensores que son colocados en el cuerpo del paciente para el monitoreo. Para la transmisión y recolección de datos se utilizó redes de área personal inalámbrica (WPAN) y a través de un Gateway conecta los datos recolectados a la red de salud. A su vez, Facco et. al. [19] propuso un modelo publish/subscribe (PubSub) basado en la nube llamado PS2DICOM, el cual emplea elasticidad y compresión de recursos para un mejor rendimiento en la transmisión de imágenes DICOM. Al adoptar un modelo de comunicación PubSub consideraron la coexistencia de dos tipos de usuarios: productores que publican los datos, y los consumidores los cuales se suscriben para recibir información sobre los datos. Con esto se demostró que PS2DlCOM utilizando el modelo PubSub se asegura que la

infraestructura diseñada tenga disponibilidad para el especialista médico desde cualquier parte del mundo, además de que puede mejorar significativamente la transmisión, almacenamiento y recuperación de imágenes DICOM.

### 1.4.1 Plataforma FIWARE

FIWARE es una plataforma de middleware que se utiliza para recopilar e intercambiar datos a través de un estándar público lo que facilita la interacción entre distintos sistemas, esta característica es aprovechada para el diseño de sistemas sanitarios, como es el caso del acoplamiento de dispositivos inteligentes a una aplicación de sistema de vida activa asistido (AAL por sus siglas en inglés) utilizando un middleware, donde los autores muestran cómo se puede acoplar FIWARE con un sistema AAL llamado AYUDO el cual sirve para monitorizar la salud en personas mayores de edad [20]. Para el acople AYUDO cuenta con un adaptador predeterminado REST-JSON para conectarse a FIWARE por medio de la API NGSI v2 de orion context broker, permitiendo la transferencia de datos en formato JSON, y los datos recolectados por los dispositivos inteligentes conectados a FIWARE son transformados en base a las reglas del modelo AYUDO-ML, mismo que ya viene integrado en este sistema AAL para la interpretación y análisis de datos en base a las características propias de AYUDO.

La investigación realizada por Tsiouris et. al. [21], muestra el diseño de plataformas de salud interoperables las cuales utilizan dispositivos IoT con infraestructura en la nube. Para la arquitectura consideraron el diseño de la plataforma en base a las guías del estándar IEEE 1471–2000, un módulo de comunicación FIWARE-orion para la interoperabilidad y compatibilidad entre distintos componentes asegurando la conectividad entre dispositivos IoT y la nube, módulos de plataformas para Edge computing y la nube, y HOLOBALANCE que es una plataforma de tele rehabilitación empleado en fisioterapias holográficas, aquí los pacientes pueden realizar ejercicios de rehabilitación en sus hogares y ser monitorizados en tiempo real por el sistema. El sistema brindo una interoperabilidad y escalabilidad permitiendo la incorporación de nuevos módulos o dispositivos con facilidad cuando se tenga alguna necesidad, además de que se puede adaptar a diferentes dominios de salud, permitiendo que tanto pacientes y doctores puedan acceder a la información.

Con esto se demuestra que FIWARE es una opción ideal para la integración y acoplamiento de varios sistemas sanitarios distintos.

### 1.4.2 Seguridad en los sistemas de comunicación

Son varios los ataques o vulnerabilidades que pueden sufrir los sistemas de comunicación para transmisión de datos, por ello Al Osail et. al. [22], sugieren que los sistemas sanitarios tengan requisitos similares a los sistemas generales de tecnología de la información como autenticación, integridad, confidencialidad, disponibilidad, control de acceso. Por lo tanto, recomiendan el uso de técnicas como [22]: encriptación, marcas de agua, código de autenticación de mensaje (MAC), firmas digitales, auditoria, modelo de confianza cero; para evitar riesgos a futuro en los sistemas sanitarios y asegurar la privacidad de los datos médicos. A su vez en [23] los autores consideran que el uso de técnicas como algoritmos de curva elíptica (ECC), estándar de encriptación avanzada (AES) y las funciones hash criptográficas, ayudarán a fortalecer los sistemas sanitarios ofreciendo una mayor seguridad en la transmisión de datos sin comprometer la eficiencia del sistema.

A continuación, se analizará algunos estudios sobre las técnicas de seguridad que fueron implementados en diseños de sistemas de comunicación.

En [18] el canal de comunicación con redes WBAN, para la seguridad de los datos emplearon grupos de modelo de envío y recepción (GSMR) y un esquema de cifrado homomórfico basado en matrices (HEBM), con esto el sistema permite el análisis automático de los datos médicos que ya fueron cifrados para luego procesarlos y realizar diagnósticos preliminares sin comprometer la privacidad del paciente.

En el estudio propuesto por El Zouka y Hosni [24], sobre comunicaciones seguras en ambientes IoT para un sistema de monitoreo médico inteligente, proponen una arquitectura compuesta por varios sensores para medir parámetros médicos y que los mismos pasaran por un microprocesador que estará conectado a la nube para luego emitir un diagnóstico en base a los datos obtenidos y que los médicos y pacientes puedan observarlos a través de una plataforma. Para salvaguardar los datos médicos utilizaron una autenticación ligera y segura basada en kerberos y un protocolo de acuerdo de claves en base al intercambio de claves Diffie-Helman, en donde el protocolo consta de una fase de registro del paciente, una fase de inicio de sesión del paciente y una fase de autenticación del paciente. De esta forma el sistema brinda una plataforma confiable para el monitoreo de un paciente en tiempo real haciendo énfasis en la seguridad de los datos.

En el diseño de un sistema de telemedicina mediante una arquitectura orientada a servicios (SOP) para pacientes de COVID 19 [25], los autores diseñaron una arquitectura de seguridad basadas en dos interfaces de comunicación (web y móvil), se comunican a través de la tecnología del sistema global de comunicaciones (GSM) y están protegidas con técnicas de cifrado y descifrado SSK. La autenticación consta de un token que se envía en cada mensaje. El token se inicia usando datos del paciente como la identificación, si los datos son incorrectos, el token caducará y se generará uno nuevo, mientras que cuando los datos son correctos, el token permite cifrar el mensaje, convirtiendo el texto normal en texto cifrado. Este texto cifrado se puede descifrar posteriormente usando la información contenida en el token. Con esto el sistema diseñado facilita la comunicación segura entre distintas plataformas de telemedicina permitiendo el intercambio de datos médicos sin vulnerar la seguridad de los pacientes.

Chen et.al. [26], presentan un sistema de comunicación para salvaguardar datos biométricos como electrocardiograma (EKG) en redes WBAN. Para la autenticación y acuerdo de llaves utilizan un extractor difuso mejorado reutilizable Juels-Sudan (RIJS) que utiliza la técnica biogrupo aleatorio compartido (SRB por sus siglas en inglés) que mejora la seguridad, eliminado la correlación que se da cuando se extrae múltiples datos biométricos, además RIJS utiliza un modelo biométrico dinámico (DBM) para generar señales sintéticas de EKG permitiendo la autenticación y acuerdo de llaves de forma segura y eficiente. Para el cifrado y la transmisión de datos propusieron una estrategia de implementación de migración de tareas (TMDS), para reducir de forma significativa la cantidad de datos que se deben encriptar y transmitir, reduciendo el consume de energía en la red.

En los trabajos [27] y [28] se diseñaron modelos híbridos de cifrado para la transmisión de datos en sistemas médicos basados en la nube. En ambos diseños utilizan la transformada discreta de Wavelet en 2d (2S-DWT) para cifrar imágenes. En este caso se hará énfasis en las técnicas de cifrado que utilizaron para la transmisión de datos. En [27] utilizan dos algoritmos de cifrado simétrico: el primero es Blowfish que utiliza un tamaño de bloque de 64 bits y una longitud de clave que puede variar entre 32 a 448 bits. El segundo es Twofish que funciona con bloques de 128 bits y que admite claves hasta de 256 bits. En [28] utilizan un cifrado AES-256 por su eficiencia y RSA-64 que

proporciona una capa adicional de seguridad sin afectar al rendimiento del sistema, dando como resultado un cifrado más robusto y resistente a ataques.

Como se pudo observar en los trabajos analizados, las técnicas de seguridad en un sistema de comunicación dependerán del tipo de dato que se va a transmitir, además que empleando técnicas de cifrado y autenticación eficientes se puede mejorar la seguridad y eficiencia en recursos de un sistema de comunicación.

Finalmente, para proteger los datos médicos, a más de la implementación de técnicas de seguridad que fueron analizadas en cada artículo, se debe tomar en cuenta las regulaciones jurídicas vigentes como la HIPAA, o la ley de protección de datos personales de cada país, como se da el caso de estudio de Sánchez et. al. [29], en donde su propuesta para garantizar la privacidad en una red de sensores inalámbrica (WSN) es el trabajo en conjunto entre las áreas tecnológicas, empresariales y jurídicas, haciendo énfasis que la cobertura de seguridad va a depender de las necesidades específicas que cada sistema requiera y de los datos médicos que se va a transmitir, sin comprometer la seguridad de los datos personales de los usuarios, afianzando su confianza.

## 1.5 Resultados esperados

Los resultados que se esperan con el desarrollo de la presente tesis son:

A. Un sistema de comunicación basada en infraestructura de contenedores que permita la transmisión de datos médicos sensibles a través de HCE, la misma que contará con mecanismos de seguridad y una base de datos no relacional para su almacenamiento.
B. La ejecución de un protocolo de encriptación el cual sea lo suficientemente robusto para salvaguardar la confidencialidad e integridad de los datos médicos de las HCE que se van a transmitir a través del sistema de comunicación, el protocolo será ejecutado tanto en los datos en reposo, es decir antes de ser transmitidos, y al momento que se ejecute la transmisión.
C. La aplicación de métodos de autenticación mediante el uso de contraseña o token,

así como un protocolo de autorización que dará acceso a los usuarios según el rol que estos tengan para el manejo de la información en el sistema de comunicación.

D. Validación del sistema de comunicación bajo las recomendaciones de las normativas HIPAA y GDPR, y ejecución de pruebas de seguridad en base a las métricas de evaluación planteadas para comprobar el rendimiento del sistema.

## 1.6 Alcance y limitaciones del proyecto

En cuanto al alcance el proyecto tiene la intención de brindar los aspectos necesarios que se deben tomar en cuenta para salvaguardar la seguridad de los datos médicos en sistemas de comunicación, mediante la protección de la confidencialidad, integridad y disponibilidad de los datos, en base al cumplimiento de normativas de seguridad de datos médicos como la HIPAA y el GDPR.

Con el uso de técnicas de cifrado en la transmisión de datos, métodos de autenticación y control de acceso se garantizará que solo el personal de salud autorizado tenga acceso a los datos médicos del sistema, y que la integridad y disponibilidad de los datos no sean vulnerados.

Mediante un enfoque tecnológico y legal, el proyecto busca integrar a los pacientes en el sistema de comunicación, con el objetivo de generar confianza en ellos, asegurándoles que sus datos estarán protegidos una vez ingresen al sistema, cumpliendo con las normativas vigentes sobre la seguridad de los datos personales médicos.

Las limitaciones del proyecto serían la ubicación geográfica de los centros de salud, ya que por su inaccesibilidad no contarían con una infraestructura tecnológica necesaria para dar soporte a sistemas de telemedicina, esto recae en que puede existir fallas en la conectividad, latencias altas, sobrecarga de procesos y tiempos de respuesta largos.

El sistema puede presentar problemas de interoperabilidad si se desea acoplar con diferentes sistemas de telemedicina, y finalmente la falta de capacitación al personal de salud puede provocar que el sistema genere vulnerabilidades, afectando la seguridad de los datos médicos.

## 1.7 Impacto social y tecnológico

El aporte que tendrá este trabajo es que ayudará a reducir la brecha en calidad de atención a través del manejo de historias clínicas electrónicas, ya que se podrá gestionar de una manera más eficiente los datos médicos del paciente, evitando posibles fallas o errores que se solían cometer cuando se manejaba una historia clínica en papel. Contribuirá a la eficiencia y escalabilidad del canal de comunicación, utilizando técnicas de seguridad acordes a los requerimientos necesarios para una gestión de recursos óptima. Además, el trabajo ayudará a crear conciencia que para la ejecución de proyectos sanitarios es necesario la colaboración de varias ramas profesionales, en este caso específico el trabajo en conjunto del área tecnológica con el área legislativa, para velar por el cumplimento normativo en cuanto a la ley de protección de datos personales en el sistema de comunicación, con el fin de precautelar los datos del paciente, mejorando su confianza.

# CAPÍTULO 2

## 2. MARCO TEÓRICO

A continuación, se dará a conocer algunos aspectos importantes que se debe considerar para el diseño de un sistema de comunicación segura para la transmisión de datos médicos sensibles:

### 2.1 Sistemas de comunicación

Un sistema de comunicación de datos médicos es un conjunto de tecnologías y protocolos que permite transmitir información médica como HCE, imágenes o videoconferencias, entre el personal de salud y los pacientes. Estos sistemas están en la capacidad de realizar consultas, diagnósticos, recolección y manejo de datos médicos, desde cualquier parte que se encuentre el médico o el paciente [30].

Los sistemas de comunicación para transmitir datos médicos incluyen [31]:

- **Infraestructura de red**: es el medio por el que se establecerá la transmisión de datos médicos a través del internet, garantizando que la comunicación sea fluida y confiable, para ello se utiliza diferentes medios de conectividad como fibra óptica, redes móviles 4G/5G, Wifi de alta velocidad, redes privadas virtuales (VPN), servicios en la nube.
- **Software o plataformas:** son herramientas que facilitan la gestión de los datos médicos, mediante las plataformas se pueden incluir aplicaciones, sitios web, registros de HCE, entre otros.
- **Protocolos de seguridad:** son mecanismos que ayudan a la privacidad y confidencialidad de los datos médicos a transmitir, cumpliendo con normativas como la HIPAA o GDPR.

### 2.2 Seguridad en los sistemas de comunicación

Para garantizar la confidencialidad e integrad de los datos, los sistemas de comunicaciones deben contar mecanismos de seguridad como [31]: encriptación de

datos para cambiar el formato de los datos a transmitir asegurando la confidencialidad de estos; seguridad en el acceso al sistema empleando políticas que regulen el acceso a los usuarios o dispositivos al sistema; autenticación para verificar la identidad de un usuario antes de que ingrese al sistema; seguridad en la capa de transporte, monitoreo y registro de eventos para detectar anomalías en la red del sistema. Todos estos mecanismos ayudaran a prevenir vulnerabilidades o ataques que puedan afectar el funcionamiento del sistema.

### 2.2.1 Vulnerabilidades

Las vulnerabilidades son debilidades o fallos en el sistema de comunicación, en donde la seguridad es insuficiente o defectuosa, lo que es aprovechado por los atacantes para poder acceder al sistema. Dentro de las vulnerabilidades más recurrentes en los sistemas de comunicación para transmitir datos médicos son [32]:

- **Cifrado débil:** los datos médicos al momento de la transmisión no están cifrados, son un blanco fácil para los atacantes.
- **Contraseñas débiles:** si se utilizan contraseñas débiles, los atacantes podrán acceder al sistema con facilidad.
- **Malas configuraciones de seguridad:** errores en la configuración del sistema pueden permitir accesos no autorizados.
- **Errores humanos:** la falta de educación para el uso del sistema de comunicación puede generar aperturas que los atacantes aprovechen con facilidad.

### 2.2.2 Ataques

Los ataques son acciones maliciosas que los atacantes ejecutan para explotar las vulnerabilidades de un sistema, dentro de los cuales destacan los siguientes:

- **Hacking:** se trata de una serie de actividades que exploran debilidades a nivel de software y sistemas informáticos. Las personas que ejecutan estos ataques suelen intentar robar o alterar datos de forma ilegal [33].
- **Malware:** o también conocido como código malicioso, es un pequeño software

escrito para atacar computadoras. Se trata de elementos como virus o troyanos [32].

- **Phishing:** ataques que están diseñados específicamente para atacar a individuos mediante engaños para que revelen datos específicos y acceder a los sistemas de comunicación [33].
- **Ransomware:** la atacante cifra los datos médicos que se transmiten en un sistema de comunicación, para pedir un pago y así liberar los datos [32].
- **Denegación de servicios (DoS/DDoS):** se trata de una inundación de ataques a los sistemas de comunicación para que el sistema deje de funcionar, afectando la disponibilidad de este [33].
- **Man in the middle (MitM):** el atacante intercepta los datos médicos que se transmiten por el sistema de comunicación para espiar o modificar la información [33].
- **Ataque de fuerza bruta:** el atacante prueba varias combinaciones posibles para obtener claves de cifrado, contraseñas de inicio de sesión u otro tipo de información se seguridad [33].

## 2.3 Protocolos de encriptación

Un protocolo de encriptación es un conjunto de reglas y procesos que están diseñados para cifrar y proteger los datos cuando estos van a ser transmitidos. El propósito es garantizar la confidencialidad, integridad, autenticidad y no repudio de los datos al momento de la transmisión por un sistema de comunicación, para evitar que personas ajenas al sistema no puedan acceder, modificar o interceptar los datos [34].

Uno de los protocolos más utilizados para sistemas de comunicación es el TLS, que es un protocolo de seguridad de capa de transporte que asegura las comunicaciones entre un emisor y su receptor, proporcionando confidencialidad en los datos [22].

Para encriptar los datos se pueden utilizar los siguientes métodos de cifrado [33] [34]:

- **Cifrado simétrico:** utiliza una sola clave secretea para cifrar y descifrar los datos, la clave debe ser compartida entre el emisor y el receptor. Dentro de sus características es que posee un rendimiento eficiente para cifrar un volumen de datos grande. Un ejemplo de este cifrado es el estándar de encriptación avanzada

AES, que es un algoritmo que utiliza claves de 128, 192 o 256 bits, lo que permite que la longitud de su clave sea larga y difícil de vulnerar.

- **Cifrado asimétrico:** utiliza dos claves, una pública para cifrar los datos y otra privada para descifrarlos. La clave publica se comparte de manera libre, mientras que la calve privada se mantiene en secreto. Dentro de las características principales es que brinda una mayor seguridad en el intercambio de claves porque no se comparte la clave privada como en el cifrado simétrico, pero en cuanto a funcionalidad es lento en comparación con el cifrado simétrico, ya que requiere de más recursos computacionales. Como ejemplos se encuentran el algoritmo RSA o el algoritmo de cifrado de curva elíptica ECC.

## 2.4 Métodos de autenticación

La autenticación es un proceso que garantiza la identidad de un usuario, garantizando su acceso de manera segura al sistema, en el contexto de sistemas de comunicación para la transmisión de datos médicos se requiere que estos métodos de autenticación sean lo suficientemente robusto para asegurar la privacidad y confidencialidad de los datos, a la vez que sean flexibles para cubrir las necesidades de los usuarios [35].

En los sistemas de comunicación para el manejo de datos médicos se recomienda el uso de métodos de autenticación por dos factores (2FA) o multi-factor (MFA), ya que combina al menos 2 tipos de credenciales o métodos de autenticación para ofrecer un nivel de seguridad elevado y proteger el acceso no autorizado al sistema [36], estos métodos pueden ser [36]:

- **Contraseñas:** comúnmente utilizada donde se solicita un código o PIN, donde solo el usuario la conoce y que está conformada por letras, palabra o secuencia numérica.
- **Tokens**: son generados por el sistema como una clave temporal para verificar la identidad del usuario, pueden ser basados en software, hardware o enviados por SMS o correo electrónico.
- **Biometrías:** utiliza características físicas de un usuario para verificar su identidad y permitirle el acceso al sistema. Los datos biométricos que se utilizan suelen ser huellas dactilares, reconocimiento fácil, reconocimiento de voz o escaneo de iris.

- **Certificados digitales:** archivos electrónicos que vinculan la clave de un usuario a través de una autoridad de certificación confiable.

## 2.5 Datos médicos sensibles

Los datos médicos sensibles son aquellos que poseen información personal de un paciente, y por su naturaleza requieren de un nivel de protección alta. Debido a la naturaleza de estos potencialmente pueden causar daño al paciente si se llegan a divulgar sin el consentimiento adecuado. Estos datos sensibles pueden incluir información acerca del historial clínico del paciente en donde se detalle diagnósticos, tratamientos o pruebas médicas; también contiene información genética sobre enfermedades de alto riesgo, información relacionada sobre la salud mental, tratamientos médicos, entre otros [37].

## 2.6 Historias clínicas electrónicas

Las HCE son registros digitales de información médica de un paciente, en donde se puede almacenar, gestionar y compartir datos médicos de un paciente de forma segura y eficiente [38]. Hoy en día las HCE son de gran importancia en la práctica médica por las siguientes razones [39]:

- Brinda al personal de salud un acceso centralizado para ingresar de forma rápida y sencilla a la información de un paciente, independientemente de su ubicación geográfica.
- Almacenamiento y recuperación de información y datos sobre la salud.
- Mejora la calidad de atención facilitando la coordinación entre distintos proveedores de salud, lo que resultaría en diagnósticos y tratamientos más precisos.
- Están diseñadas para cumplir con normas de privacidad y seguridad de datos para proteger la información sensible del paciente mediante mecanismos de cifrado y control de acceso.
- Eficiencia en la gestión de datos, generando procesos automáticos para la generación de citas o gestión de recetas médicas.

## 2.7 Normativas para la seguridad de datos personales

Las normativas principales que son utilizadas para el diseño de sistemas de comunicación de datos médicos son [7] [8]:

### 2.7.1 HIPAA

La HIPAA establece una serie de reglas y recomendaciones para que los sistemas de comunicación garanticen que la transmisión de datos médicos sea segura. Las recomendaciones están diseñadas para proteger la integridad, confidencialidad y disponibilidad de los datos médicos durante la transmisión, y estas recomendaciones son[3]:

- **Cifrado de datos:** para en caso de que los datos médicos sean interceptados en la transmisión por el canal de comunicación, los datos no serán legibles para personas no autorizadas.
- **Autenticación de usuarios:** implementar métodos de autenticación robustos para verificar la identidad de los usuarios antes de ingresar al sistema y manipular los datos médicos.
- **Control de acceso:** limitar el acceso a los datos médicos solo a personas autorizadas, controlar quien y cuando accede a los datos.
- **Auditorias y seguimiento:** realizar registros donde se detalle las interacciones que se realicen con los datos médicos en el sistema.
- **Integridad de datos:** asegurar que los datos médicos no sean alterados o eliminados sin previa autorización.
- **Evaluación continua de riesgos:** ejecutar evaluaciones de riesgo periódicas para identificar vulnerabilidades en el sistema de comunicación y corregirlas.

### 2.7.2 GDPR

De la misma forma el GDPR establece estrictas normas a seguir para garantizar la protección de los datos médicos de una persona, durante su transmisión por el sistema

[3] https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164

de comunicación. Las principales recomendaciones para una comunicación segura de datos médicos son[4]:

- **Cifrado de datos:** se exige que los datos médicos estén cifrados cuando están almacenados o reposo, y cuando estos se estén transmitiendo por el sistema. Consideran al cifrado una medida técnica adecuada para mitigar posibles violaciones de los datos.
- **Autenticación y control de acceso:** se requiere un control estricto de que personas acceden al sistema, además solo el personal autorizado podrá acceder a los datos médicos a través de métodos de autenticación robustos.
- **Registro de acceso y auditoria**: los sistemas deberán contar con mecanismos de auditoría y registros de acceso para monitorear quien accede a los datos médicos, para detectar posibles violaciones de seguridad o accesos no autorizados.
- **Evaluación de impacto relativa a la protección de datos:** antes de implementar algún sistema de comunicación para transmitir datos médicos, se debe realizar una evaluación de impacto en la protección de datos para identificar y eliminar riesgos potenciales.

[4] https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=celex%3A32016R0679

# CAPÍTULO 3

## 3. METODOLOGÍA

Los sistemas de comunicación son de gran importancia al momento de transmitir, gestionar y almacenar datos médicos, a su vez una de las fuentes principales de datos en sistemas sanitarios son las historias clínicas electrónicas (HCE por sus siglas en inglés), ya que recoge información delicada sobre el paciente como, datos personales, enfermedades, tratamientos, entre otros. En base a esta información se va a desarrollar un sistema de comunicación el cual permita transmitir de manera segura HCE.

Para esto se utilizará una metodología experimental, ya que el sistema se va a enfocar en el uso de infraestructura basada en contenedores con componentes FIWARE, donde la infraestructura está orientada a salvaguardar la información sensible que se transmita por el sistema, mediante un mecanismo de cifrado, métodos de autenticación y autorización, a su vez se efectuará pruebas de funcionamiento a través de métricas de evaluación para medir el rendimiento del sistema y que estas cumplan con las normativas de la protección de datos personales como la HIPAA o GDPR.

### 3.1 Diseño del sistema de comunicación

Para desarrollar el sistema de comunicación se priorizó la gestión y ahorro de recursos, por lo que se va a emplear contenedores. Los contenedores son paquetes de software que ya incluyen la aplicación y todas sus dependencias y que comparten el kernel con otros contenedores, lo que permite ejecutarse como un proceso aislado en el espacio del sistema operativo del usuario[5].

Para administrar el manejo de los contenedores se va a usar Docker, que es un software de código abierto que ayuda a la migración entre distintos sistemas, evitando posibles

[5]https://www.hpe.com/lamerica/es/whatis/containers.html#:~:text=Los%20contenedores%20son%20tecnolog%C3%ADa%20que,sistema%20operativo%20en%20cualquier%20contexto.

inconvenientes de dependencia ente versiones de software de los contenedores que se vaya a utilizar[6].

Dentro de los contenedores se utilizan componentes o tecnologías FIWARE, la cual se trata de una plataforma de código abierto para desarrollar soluciones inteligentes, donde el componente principal es Orion context brocker[7], para administrar la información de contexto que se añadirá al sistema de comunicación.

Además, dentro de los datos médicos que se transmiten por el sistema de comunicación son HCE, ya que en [14] [15], tiene un alto grado de importancia en el ambiente sanitario, porque registran información médica de un paciente como antecedentes médicos, diagnósticos, resultados de exámenes de manera digital, para facilitar el acceso, almacenamiento y gestión de los datos médicos [31].

### 3.1.1 Arquitectura del sistema de comunicación

En la figura 3.1 se muestra la arquitectura utilizada para desarrollar el sistema de comunicación segura para la transmisión de HCE, donde se puede observar los componentes FIWARE y bases de datos que se utilizaron, además de cómo es el flujo de comunicación entre componentes a través de los puertos activos y como estos van a interactuar con el usuario y las HCE.

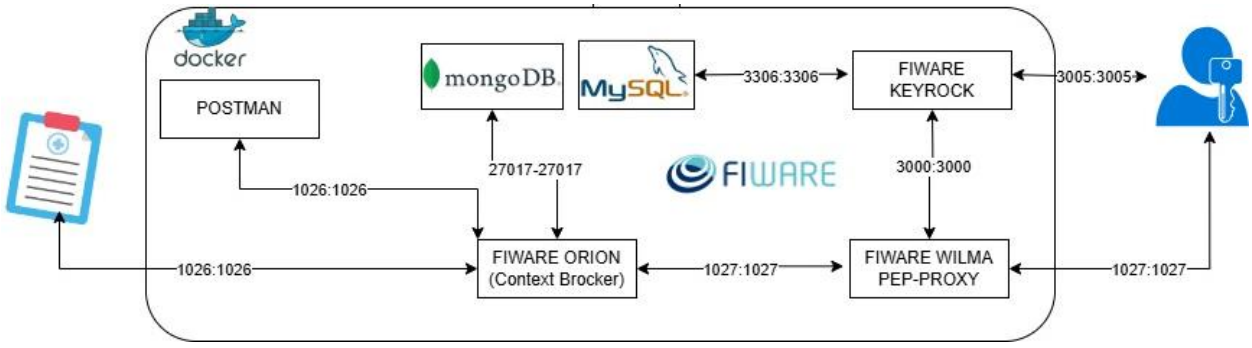



**Figura 3.1 Arquitectura del sistema de comunicación**

[6]https://www.servidoresadmin.com/que-es-docker-y-como-funciona-introduccion-adocker/

[7]https://fiwaretraining.readthedocs.io/esmx/latest/ecosistemaFIWARE/plataformaFIWARE/

### 3.1.2 Usuarios o actores

En [3], destaca la importancia de incorporar tecnologías a los procesos sanitarios, para que médicos y pacientes sean beneficiados en la implementación de sistemas de HCE, por lo tanto, los actores o usuarios que van a ser parte del sistema de comunicación son:

- **Administrador:** es el usuario con más privilegios para administrar el sistema de comunicación, ya que será el encargado de velar por el buen funcionamiento del sistema al momento de la transmisión de HCE, adicional el podrá crear o eliminar las HCE para que médicos y pacientes puedan acceder a ellas.
- **Médico:** su función como usuario es leer y actualizar ciertos atributos de las HCE de los pacientes registrados en el sistema de comunicación.
- **Paciente:** como usuario solamente podrá acceder a su propia HCE, donde podrá observar y verificar sus datos médicos.

### 3.1.3 Políticas de usos y servicios

El sistema de comunicación estará desarrollado de tal manera que los datos de las HCE que se vaya a transmitir este resguardada por un método de cifrado para que la información sea ilegible en cada de ser interceptada, y la ejecución de métodos de autenticación para que solo los usuarios previamente registrados puedan acceder al sistema. A continuación, se detallan algunas políticas y servicios que se deben considerar para el desarrollo del sistema:

- Para la transmisión de datos médicos a través de HCE, el sistema asegurará la confidencialidad e integridad de los datos sensibles cuando ya se encuentren en uso.
- El usuario será el responsable de velar por sus datos de acceso, como correo electrónico y contraseña para poder acceder al sistema.
- Los datos médicos que se registran en las HCE serán almacenados en una base de datos, así como los datos de los usuarios, roles y permisos de acceso al sistema.
- A través del sistema se pueden efectuar operaciones de creación, lectura, actualización y eliminación de HCE, según el rol o tipo de usuario.

- El sistema se desarrollará en base a las recomendaciones como la HIPAA o GDPR para garantizar que los datos médicos estén seguros dentro del sistema.

## 3.2 Infraestructura tecnológica

La infraestructura tecnológica que se plasmó en este trabajo de titulación está basada en componentes FIWARE, ya que al tratarse de una plataforma de código abierto brinda característica como escalabilidad, confiabilidad, tolerancia a fallos; además que permite la integración de diferentes componentes de software, sensores IoT, aplicaciones web, entre otros.

### 3.2.1 Fiware Orion (Context Brocker)

El componente orion es el núcleo de la plataforma FIWARE[8], actúa como un punto de conexión donde los usuarios a través de una entidad (v2/entities) pueden ejecutar operaciones como crear (*POST*), leer (*GET*), actualizar (*PUT*) o eliminar (*DELETE*) HCE según sea el caso.

El broker permite manejar las solicitudes REST utilizando la API NGSIv2 que ayuda a la interoperabilidad por medio del formato JSON y finalmente utiliza el puerto 1026 para entablar comunicación con otros componentes y manejar la información, tal y como se puede ver en la figura 3.2.

En [21] se utilizó Orion como el módulo principal de comunicación en el diseño de una plataforma de telemedicina, en donde se encargó de la gestión del flujo de datos entre diferentes módulos, incluyendo infraestructura en la nube, dispositivos IoT e interacción con el usuario. En este sistema Fiware Orion es el encargado de gestionar los datos de las HCE, y posteriormente almacenarlos en una base de datos.

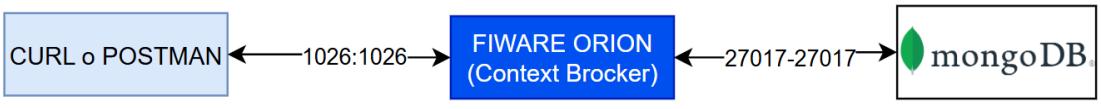



**Figura 3.2 Fiware orion**

[8] https://fiware-orion.readthedocs.io/en/master/

### 3.2.2 Fiware Keyrock

Keyrock es el componente encargado de la gestión y administración de identidades[9], el uso de keyrock con otros componentes como en este caso wilma-pep-proxy, permiten agregar seguridad para la autenticación y autorización basada en OAuth2 para acceder al sistema de comunicación y opera en el puerto 3005.

Mediante keyrock[10] se crean los usuarios o actores, la aplicación, roles y permisos según el flujo de la figura 3.4, donde los usuarios forman parte de la aplicación siempre y cuando estén autorizados con su rol respectivo. Cada usuario accederá al sistema por medio de su email y contraseña.

En [40] por ejemplo utilizaron keyrock como gestor de identidades, para la autenticación de usuarios, dispositivos y gestión de permisos, en un sistema de gestión de vacunación basado en tecnología Blockchain.

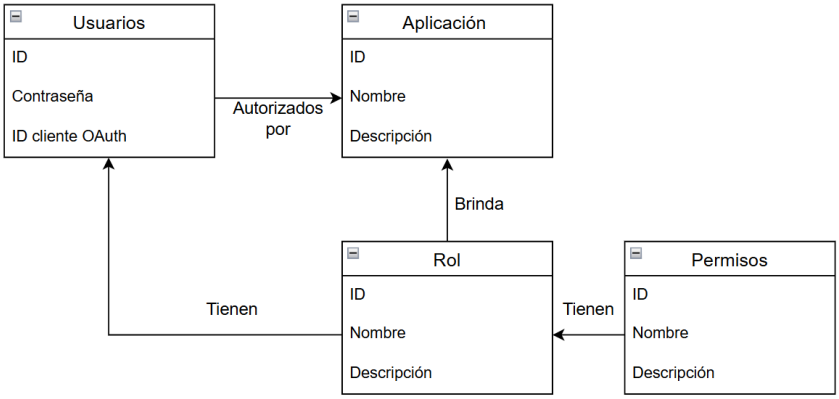



**Figura 3.3 Flujo de Keyrock**

### 3.2.3 Fiware Wilma pep proxy

Wilma es un componente de FIWARE que actúa como PEP proxy en el puerto 1027[11], por lo tanto, al combinarlo con el componente keyrock se logra aplicar un control de acceso a Orion que es el componente principal del sistema de comunicación, con esto

[9] https://fiware-idm.readthedocs.io/en/latest/

[10] https://fiware-tutorials.readthedocs.io/en/latest/roles-permissions.html

[11] https://fiware-pep-proxy.readthedocs.io/en/latest/

solo los usuarios previamente registrados y autorizados en keyrock podrán acceder al sistema. Gracias a la gestión de roles se puede administrar los permisos para el ingreso a los datos de las HCE[12], lo que permite brindar diferentes niveles de acceso a los usuarios según el rol que estos tengan.

Mediante keyrock, se puede enlazar con Wilma Pep-Proxy, a través de credenciales como la identificación de la aplicación, el nombre y la contraseña del Pep Proxy, tal como se muestra en la figura 3.5.

En [40] se utilizó pep-proxy como un proxy de seguridad entre los usuarios y los servicios internos de un sistema de vacunación basado en blockchain, donde pe-proxy validaba las solicitudes de los usuarios usando tokens OAuth2 generados por keyrock.

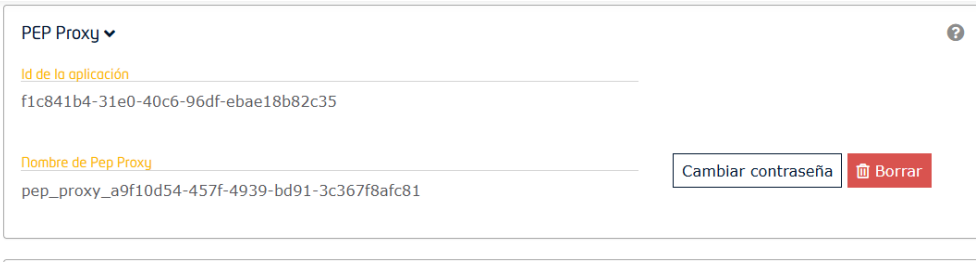



**Figura 3.4 Combinación de pep-proxy y keyrock**

### 3.2.4 Mongo DB

Mongo es una base de datos no relacional para gestionar de una manera más dinámica cargas de trabajo[13], además que Orion está basada en la tecnología de código abierto de mongo DB para la persistencia de datos de contexto que contiene[14], razón por la cual se utiliza esta base de datos para que almacene la información de las HCE en formato JSON que se gestiona a través de Orion, y su comunicación se da mediante el puerto 27017.

---

[12] https://fiware-tutorials.readthedocs.io/en/latest/roles-permissions.html

[13] https://phoenixnap.com/kb/docker-mongodb

[14] https://fiware-zone.readthedocs.io/es/latest/getting-started.html

### 3.2.5 MYSQL

Para almacenar los datos generados en keyrock como permisos, roles, usuarios, entre otros, se requiere de una base de datos relacional como MySQL[15], ya que keyrock sigue un modelo relacional debido a que los datos que genera están ligados unos con otros[16]. El uso de MySQL permite que la información se almacene de una forma segura y organizada en una base de datos llamada IDM, de tal forma que se pueda consultar los datos mediante tablas normalizadas como se muestra en la figura 3.5, de las cuales las más utilizada son[17]:

- **Pep-proxy:** almacena información de las credenciales OAuth2 para trabajar con keyrock.
- **Rol:** almacena los roles utilizados para el sistema de comunicación.
- **Permisos:** almacena los permisos o acciones de cada rol.
- **Usuario:** almacena información como ID, nombre de usuario, email, contraseña.
- **Token de acceso OAuth:** almacena los tokens generados por Keyrock para acceder al sistema.

Adicional para acceder a esta base de datos se lo hará por el puerto 33060.

[15]https://biblus.us.es/bibing/proyectos/abreproy/92801/fichero/TFG-2801+CARMONA-PALOMARES%2C+%C3%81LVARO.pdf

[16] https://fiware-idm.readthedocs.io/en/7.0.1/admin_guide/#system-administration

[17] https://fiware-idm.readthedocs.io/en/7.0.1/admin_guide/

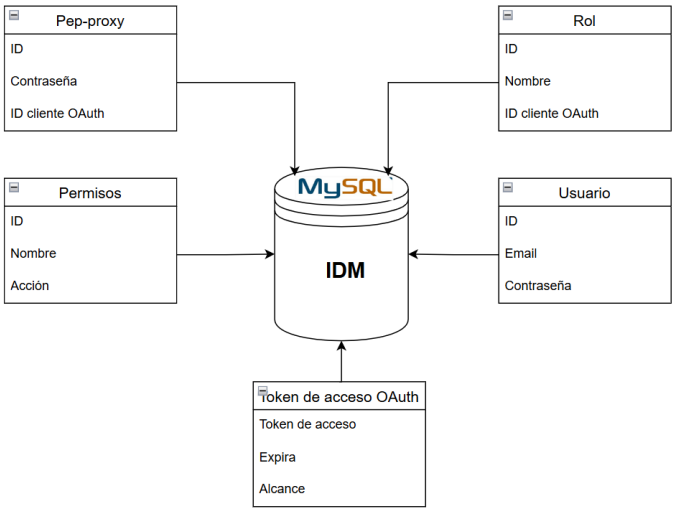



**Figura 3.5 Almacenamiento de datos MySQL**

## 3.3 Implementación de protocolo de encriptación

Para este punto se va a implementar AES, ya que es un cifrado simétrico implementado en varios sistemas de comunicación para proteger datos sensibles y que es recomendado por el instituto nacional de estándares y tecnología (NIST por sus siglas en inglés) [41]. El uso de este mecanismo de cifrado permite que los datos de las HCE estén cifrados tanto en la transmisión y al momento de almacenamiento en la base de datos, ya que ejecuta operaciones de división y expansión del texto, sustitución, desplazamiento, mezcla, ronda de clave (round key)[18], según el tamaño de la clave la cual puede ser de 128 o 256 bits, razón por la cual en caso de que la información sea interceptada el atacante deberá intentar probar las claves $2^{\text{bits}}$ para poder descifrar los datos, lo que recae en que se vuelva una tarea computacionalmente intratable [42].
Además, brinda mayor rapidez para cifrar y descifrar los datos ya que el tamaño máximo de su clave puede llegar hasta los 256 bits y por ende no ocuparía demasiado espacio de almacenamiento en el sistema, lo cual es primordial para sistema de comunicación en ambientes sanitarios donde se requiere que una alta rapidez y eficacia del sistema sin descuidar la seguridad, integridad y confidencialidad de los datos [28].

[18] https://www.pandasecurity.com/es/mediacenter/cifrado-aes-guia/

## 3.4 Aplicación de métodos de autenticación y autorización

Se va a aplicar una autenticación de dos factores (2FA por sus siglas en inglés), gracias a que Keyrock ya viene con esta capa adicional de seguridad para autenticar a un usuario que desea ingresar al sistema, en este caso los factores que se utilizan para la autenticación es la combinación de nombre de usuario con su contraseña y un token de acceso[19].

Mediante Keyrock se va a utilizar el protocolo OAuth2 para la autorización, el cual proporciona flujos de autorización específicos para que los usuarios accedan al sistema de comunicación según el rol asignado[20].

Para que un usuario acceda al sistema deberá contar con las credenciales generadas por OAuth2, las cuales son ID del cliente y el secreto del cliente[21], tal y como se muestra en la figura 3.6.


**Figura 3.6 Credenciales OAuth2**

Adicional al proceso de autorización, se añade un control de acceso al sistema de comunicación por medio del componente Wilma pep-proxy, el cual estará encargado de autorizar o denegar el acceso al sistema de comunicación en base al token generado por Keyrock. Las credenciales que se muestra en la figura 3.6, sirven para combinar

---

[19] https://fiware-idm.readthedocs.io/en/latest/user_and_programmers_guide/user_guide.html

[20] https://oauth.net/2/

[21] https://aaronparecki.com/oauth-2-simplified/

Keyrock con Wilma pep-proxy, y deben ser configuradas en el archivo config.js del componente Wilma, junto con el token (JWT) que se obtiene del protocolo de autorización OAuth2, en el algoritmo 3.1 se muestra la configuración del archivo config.js.

**Algoritmo 3.1 Configuración de credenciales en Wilma pep-proxy**

```
// Credentials obtained when registering PEP Proxy in app_id in Account Portal
config.pep = {
    app_id: (process.env.PEP_PROXY_APP_ID ||  'f1c841b4-31e0-40c6-96df-
ebae18b82c35'),
    username: (process.env.PEP_PROXY_USERNAME || 'pep_proxy_a9f10d54-457f-4939-
bd91-3c367f8afc81'),
    password: (process.env.PEP_PASSWORD ||'pep_proxy_6d01bfb2-95d4-4d8d-955e-
88f4e877915e'),
    token: {
        secret: (process.env.PEP_TOKEN_SECRET ||  '2ba93033c9130886')  // Secret
must be configured in order validate a jwt
    },
    trusted_apps : []
}
```

## 3.5 Desarrollo del sistema de comunicación

Una vez que se diseña el sistema en base a una arquitectura hecha por contenedores con componentes FIWARE y sus bases de datos, así como la implementación del cifrado AES, y el empleo de los métodos de autenticación y autorización, el sistema de comunicación para la transmisión de HCE se desarrolla de la siguiente forma:

### 3.5.1 Entidad para el sistema de comunicación

Al utilizar FIWARE Orion como componente principal del sistema, se puede utilizar entidades las cuales se representan como elemento de contexto con los campos identificación (Id) y tipo (type), a su vez las entidades cuentan con atributos que pertenecen al contexto, y cuenta con campos como nombre (name), tipo (type) y valor (value)[22], tal y como se muestra en la figura 3.7.

[22] https://fiware-training.readthedocs.io/es-mx/latest/ecosistemaFIWARE/ocb/#orion-context-broker

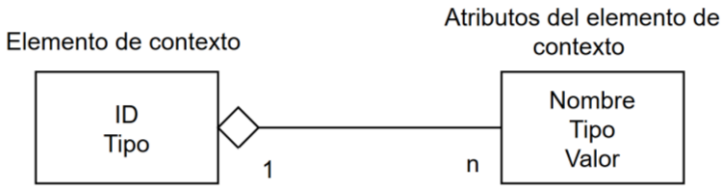



**Figura 3.7 Entidades de Orion**

En base a esto, se determina el uso de una entidad llamada Hce para el uso de las historias clínicas electrónicas, la cual cuenta con su identificación o ID y los siguientes atributos:

- **Nombre del paciente:** de tipo texto, su valor será los nombres y apellidos del paciente
- **Cédula:** de tipo estructura de valor, su valor será los datos de la cédula de identidad.
- **Fecha de nacimiento:** de tipo fecha, su valor será el día, mes y año de nacimiento.
- **Género:** de tipo texto, su valor será el género del paciente.
- **Contacto:** de tipo texto, su valor será el celular, dirección y email del paciente.
- **Historial médico:** de tipo texto, su valor será el historial médico del paciente.
- **Medicación:** de tipo texto, su valor será la receta asignada por el médico.
- **Alergias:** de tipo texto, su valor será las alergias del paciente.
- **Último ingreso:** de tipo fecha, su valor será la fecha del último ingreso del paciente.

### 3.5.2 Flujo del sistema de comunicación

Los actores o usuarios que van a interactuar con el sistema de comunicación van a ser los médicos, pacientes y el administrador del sistema.

En primer lugar, todos los usuarios deben ingresar su correo y contraseña para obtener un token que le autoriza el acceso al sistema, como se muestra en la figura 3.8.

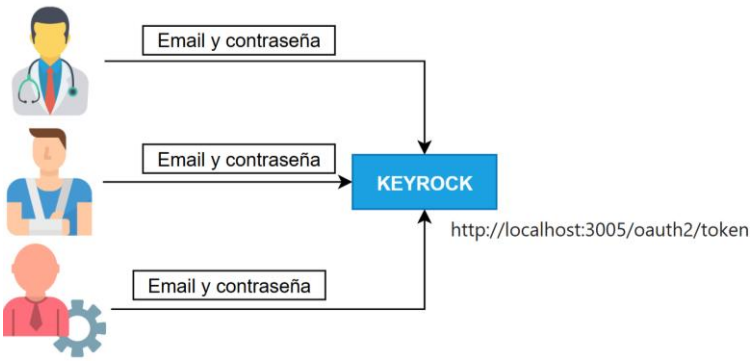



**Figura 3.8 Autenticación de usuarios por Keyrock**

Cuando se obtiene el token, Wilma pep-proxy se encarga de verificar la validez del token obtenido por Keyrock y dar acceso al sistema de comunicación por la URL **http://localhost:1027/v2/entities,** para que el usuario pueda realizar sus peticiones por Orion, tal y como se muestra en la figura 3.9.

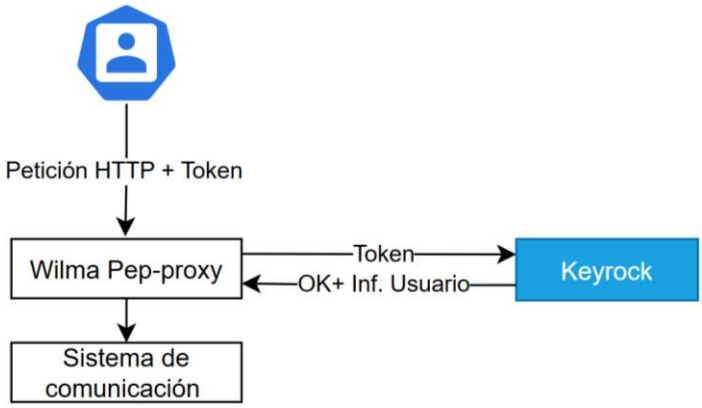



**Figura 3.9 Validación del token por Wilma pep-proxy**

Según los roles y permisos que se configuraron el Keyrock, las operaciones y acciones de cada usuario se muestra en la tabla 3.1, la cuales se las realizará de manera local a través de la URL http://localhost:1027/v2/entities:

**Tabla 3.1 Acciones de cada usuario según su rol**

| Usuario | Operación HTTP | Acción sobre las HCE |
|---|---|---|
| Administrador | POST | Creación de HCE de nuevos pacientes |
| | GET | Leer sobre todas las HCE de los pacientes |
| | PUT | Actualizar o modificar todos los atributos de las HCE de los pacientes. |
| | DELETE | Eliminar las HCE. |
| Médico | GET | Leer sobre todas las HCE de los pacientes |
| | PUT | Actualizar o modificar solo los atributos: historial médico, mediación y alergia, de las HCE de los pacientes. |
| Paciente | GET | Leer su propia HCE |

En la figura 3.10 se muestra el flujo de comunicación de cada usuario con el sistema de comunicación en base a las acciones que cada uno de ellos puede hacer en las HCE.

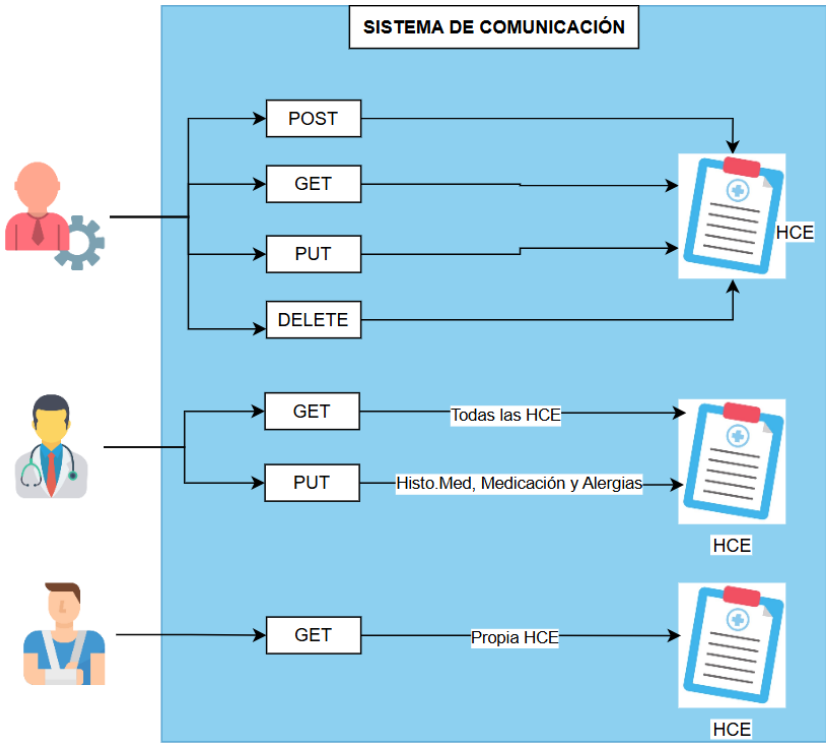



**Figura 3.10 Flujo de comunicación de cada usuario con el sistema**

### 3.5.3 Plataforma de interacción con el usuario

El flujo de comunicación será más sencillo para los usuarios o actores mediante una aplicación en lenguaje PHP, a través del entorno de desarrollo visual studio code, con el fin de mejorar la experiencia del usuario al interactuar con el sistema de comunicación.
PHP[23] es compatible con sistemas operativos como Windows o Linux, soporta varias bases de datos como en este caso MySQL que trabaja con Keyrock, o MongoDB que trabaja con Orion, a su vez que se puede trabajar en conjunto tanto el frontend como el backend, consume servicios RESTful que ayudan a comunicar la aplicación con los componentes FIWARE, y trabajar en torno a las configuraciones realizadas en Keyorck y Wilma pep-proxy.
Su uso se basa en que es un lenguaje muy utilizado para realizar plataformas de sistemas de comunicación, como es el caso de [43] en donde se diseña una aplicación de acceso al historial clínico para el sistema sanitario en Argentina, y para la configuración de herramientas dentro de la aplicación para el personal de salud se utiliza principalmente PHP, o en [44] que utilizan PHP para desarrollar un aplicativo y una base de datos con MySQL, para un sistema de comunicación inalámbrico que se encarga de la recolección de datos de una turbina eólica que operaba a 400W.

### 3.5.4 Configuración del entorno de prueba

Se deben considerar las siguientes configuraciones para que el sistema de comunicación segura para la transmisión de HCE inicie con las pruebas de funcionamiento:

- Para la ejecución de Docker en el Windows es necesario habilitar la virtualización desde la BIOS.
- Los contenedores se crean desde un archivo docker-compose.yml, que se puede visualizar en el anexo 1, en dicho archivo se encuentra las configuraciones de cada componente empleado en el sistema, así como su puerto de conexión. Dentro de los contenedores esta FIWARE Orion 3.4.0; Keyrock 8.0.0, Wilma-pep-proxy 7.5.1, MySQL 5.7 y mongoDB 6.0.

[23] https://www.php.net/manual/en/

- Verificar que todos los contenedores estén dentro de la misma red, en este caso la red se llama fiware2_my_network y en el anexo 2 se puede comprobar que todos los contenedores están dentro de la misma red.
- El modelo de HCE en formato JSON con el que va a trabajar el sistema de comunicación, en especial Orion se muestra en el anexo 3, este modelo será creado o añadido por el administrador a través de la URL http://localhost:1027/entities.

### 3.6 Métricas de evaluación

Las métricas que se consideraron para evaluar el rendimiento del sistema fueron:

- **Tiempo de respuesta:** tiempo en que el usuario hace una petición hasta que recibe una respuesta, se lo mide en milisegundos.
- **Paquetes por segundo:** paquetes que se envían en un periodo de tiempo, se los mide en paquetes/segundo.
- **Uso de espacio en base de datos:** el espacio que ocupara la información una vez sea almacenada en la base de datos del sistema, se lo mide en porcentaje de consumo de memoria del CPU y de memoria RAM.
- **Tiempo de cifrado:** tiempo que se tardara la información en ser cifrada y descifrada para que el usuario pueda visualizar la información, se mide en milisegundos.
- **Exactitud de autorización:** accesos correctamente autorizados en base a los roles de Keyrock, se lo mide en tamaño en bytes de la petición.
- **Tasa de intentos de acceso denegados:** solicitudes bloqueadas por Wilma pep-proxy en caso de que una persona ajena al sistema trate de acceder, se lo mide en tamaño en bytes de la petición.
- **Uso de recursos**: monitoreo del uso de la de recursos en la computadora cuando se esté ejecutando el sistema, así como la memoria ocupada del disco, se lo mide en porcentaje de consumo de memoria del CPU y de memoria RAM.

# CAPÍTULO 4

## 4. RESULTADOS

En el siguiente capitulo se detalla el funcionamiento, escenarios planteados, pruebas y resultados del sistema de comunicación para la transmisión segura de historias clínicas electrónicas (HCE por sus siglas en inglés). Se evaluó el sistema basado con infraestructura FIWARE en función de las métricas de evaluación planteadas con el fin de sustentar que las mismas cumplen con las recomendaciones de las normativas HIPAA y GDPR para salvaguardar la integridad y confidencialidad de los datos médicos en las HCE.

### 4.1 Funcionamiento del sistema de comunicación

Se corroboró que tanto la infraestructura y aplicación del sistema de comunicación fueran inicializados de forma correcta.

En primer lugar, se verificó que los contenedores empleados se ejecutaran de manera correcta en Docker o al menos de una forma saludable (healthy) mediante el comando *docker ps*, tal y como se muestra en la figura 4.1.

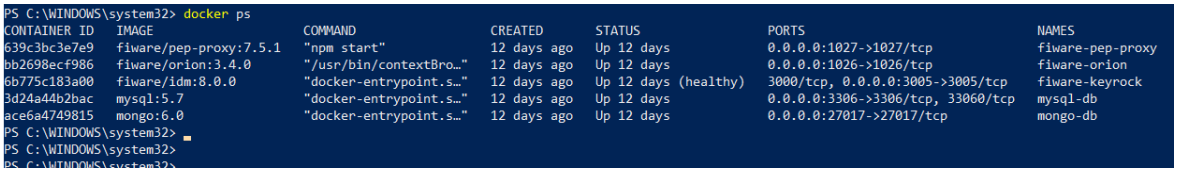



**Figura 4.1 Contenedores en Docker**

Se comprobó el correcto funcionamiento del contenedor FIWARE orion a través de la URL http://localhost:1026/version, tal y como se muestra en la figura 4.2.

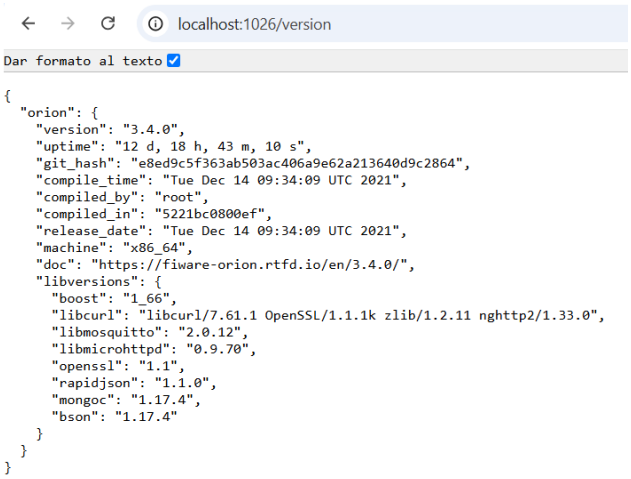



**Figura 4.2 Servicio FIWARE Orion**

En la figura 4.3 se muestra el funcionamiento del componente keyrock que a través de su interfaz gráfica se pudo configurar los roles, permisos y usuarios del sistema a través de la URL http://localhost:3005, así como la obtención de credenciales para la autorización OAuth2.

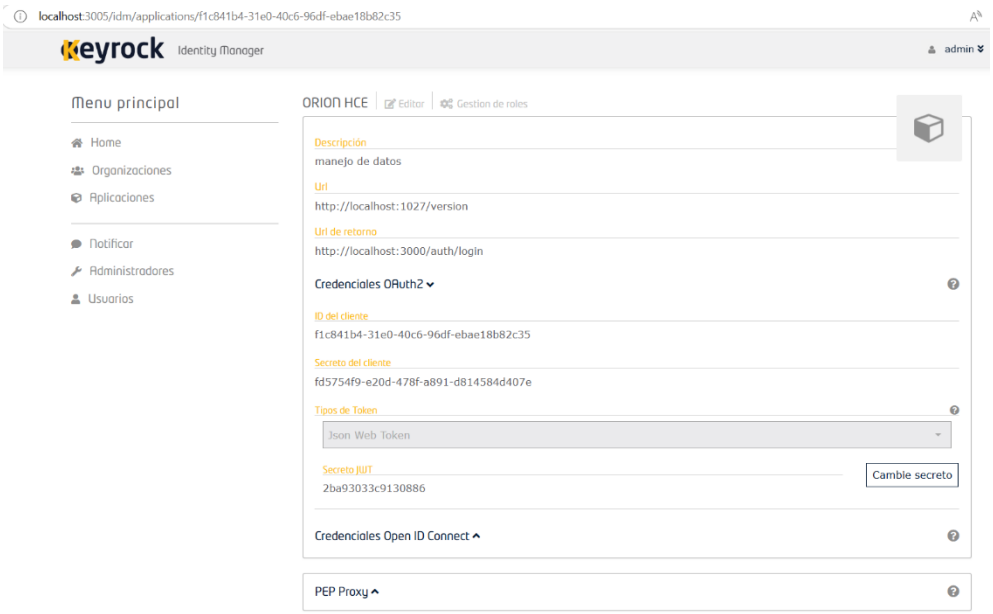



**Figura 4.3 Componente Keyrock**

Para poner en marcha la aplicación del sistema de comunicación se utilizó *visual studio code,* donde se ejecutó el programa a través del servidor de PHP, en la figura 4.4 se muestra la inicialización del aplicativo de forma local en el puerto 3000.

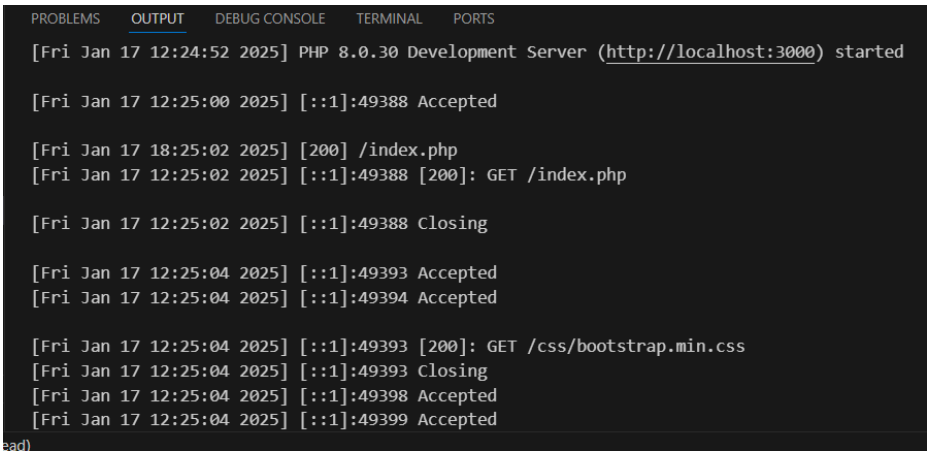



**Figura 4.4 Inicialización de la aplicación por Visual Studio Code**

Una vez ejecutada la aplicación, en la figura 4.5 se observa el menú del administrador, en la figura 4.6 el menú del médico y en la figura 4.7 el menú del paciente.


**Figura 4.5 Menú del administrador del sistema**


**Figura 4.6 Menú del médico**

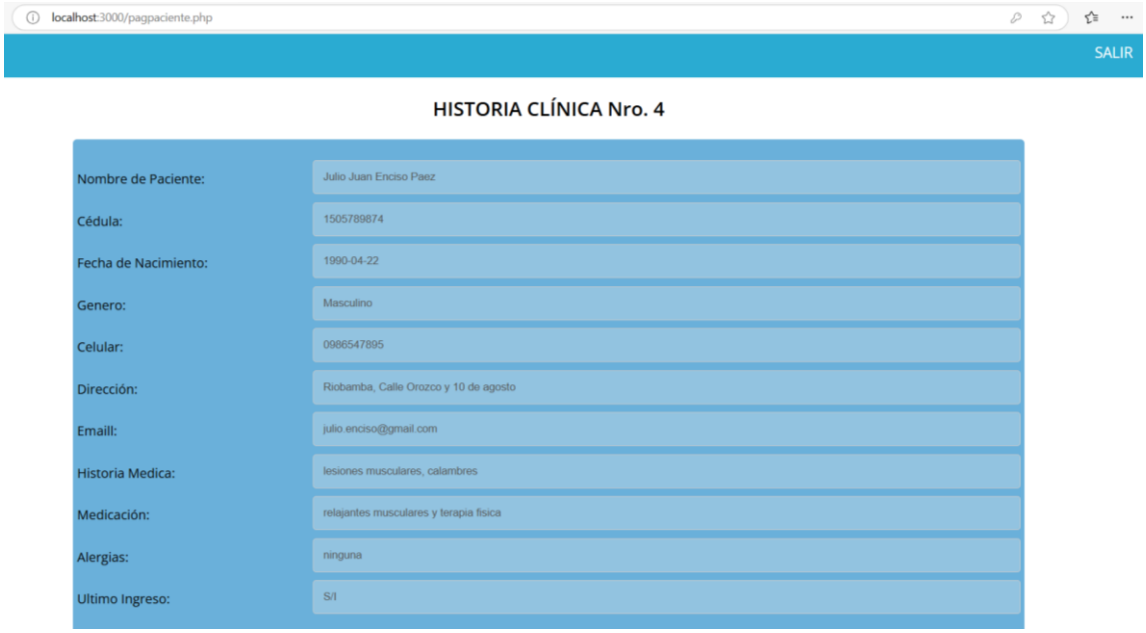



**Figura 4.7 Menú del paciente**

El token generado por cada usuario al momento de la autenticación se irá registrando de forma automática y se almacena en la base de datos IDM de MYSQL en la tabla *oauth_access_token*, tal y como se muestra en la figura 4.8.

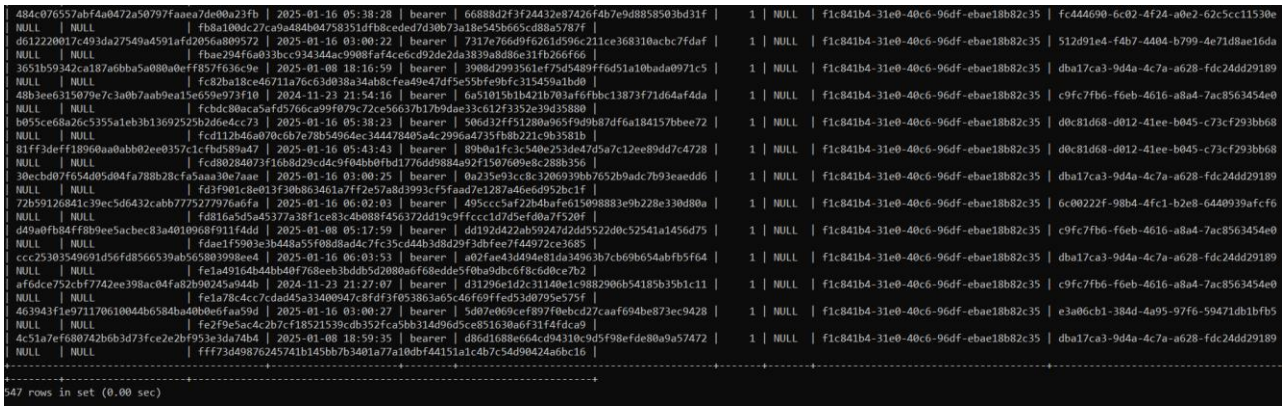



**Figura 4.8 Almacenamiento de tokens**

Finalmente, en la figura 4.9 se puede observar cómo se añade una nueva HCE al sistema mediante la aplicación por medio del menú del administrador, y la misma es almacenada en la base de datos MongoDB con el cifrado AES, tal y como se muestra en la figura 4.10.

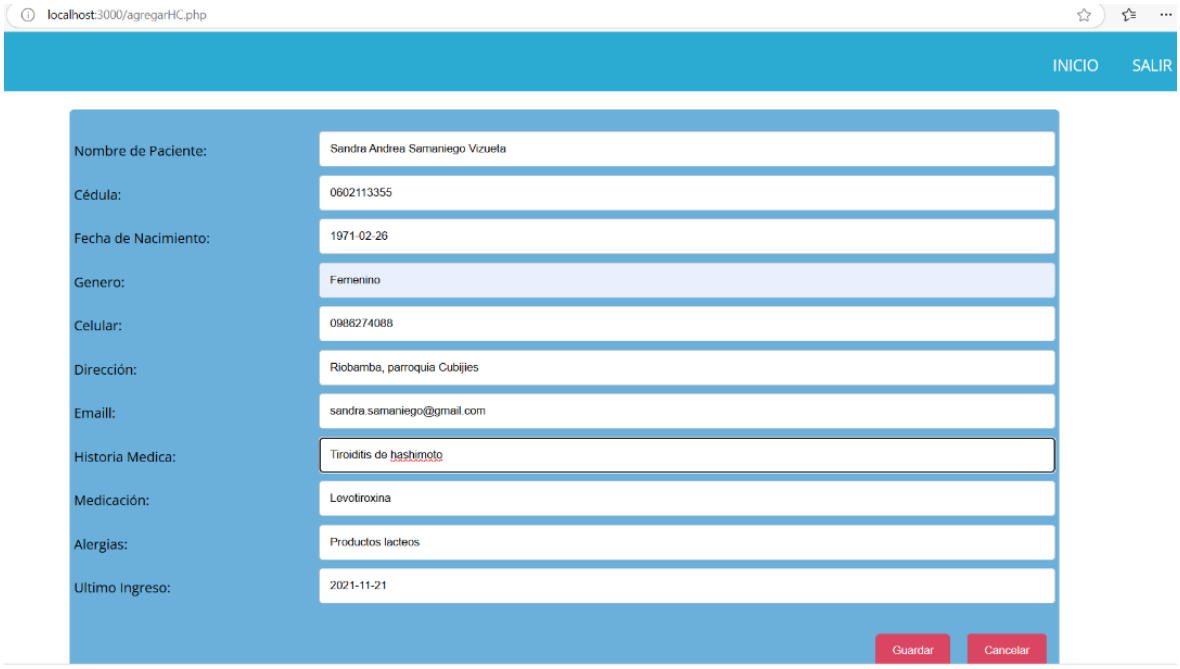



**Figura 4.9 Creación de una HCE**


**Figura 4.10 HCE almacenadas y cifradas en MongoDB**

## 4.2 Escenarios de evaluación

La evaluación del sistema de comunicación se los hizo a través de escenarios controlados para verificar su correcto funcionamiento, para ello se utilizó la herramienta *Apache Jmeter,* ya que al ser una herramienta de código abierto, permitió realizar pruebas de rendimiento en el sistema de comunicación en base a las métricas de evaluación que se plantearon anteriormente.

A continuación, se presentan los dos escenarios planteados para las pruebas de rendimiento:

- **Escenario 1:** en base a la tabla 4.1, se evaluó 60 peticiones por minuto para la actualización de atributos a los cuales los médicos tienen acceso, se obtuvo de multiplicar 20 médicos que ingresarán al sistema de forma simultánea, por 3 solicitudes que realizarán en ese instante de tiempo.

**Tabla 4.1 Parámetros del escenario 1**

| Parámetros | Valores |
|---|---|
| Médicos | 20 |
| Solicitudes de acceso | 3 |
| Tiempo de simulación | 1 minuto |

- **Escenario 2:** en base a la tabla 4.2, se evaluó 140 peticiones para la visualización o lectura de datos, en este punto existirá 70 usuarios entre pacientes y médicos conectados de forma simultánea, que harán 2 solicitudes de acceso en ese periodo de tiempo.

**Tabla 4.2 Parámetros del escenario 2**

| Parámetros | Valores |
|---|---|
| Usuarios | 70 |
| Solicitudes de acceso | 2 |
| Tiempo de simulación | 1 minuto |

## 4.3 Pruebas en base a métricas de evaluación

Mediante *Jmeter* se pudo simular diferentes niveles de carga para verificar el rendimiento del sistema en los escenarios planteados, por lo tanto a continuación se muestran los resultados obtenidos en cada uno de los escenarios:

- **Escenario 1**

En la tabla 4.3 se muestran los resultados obtenidos para el escenario 1, donde se simuló peticiones PUT para actualización de datos las cuales todas fueron aceptadas, y se corroboró el comportamiento del sistema y los tiempos de respuesta generados en las solicitudes.

**Tabla 4.3 Resultados del escenario 1**

| Solicitudes | # Peticiones | Tiempos de respuesta (ms) | | | Throughput (rendimiento) |
|---|---|---|---|---|---|
| | | Prom. | Máx. | Min. | Paquetes /s |
| Petición HTTP | 60 | 54 | 187 | 30 | 3,1 |

En base a los resultados, se muestra que el sistema al procesar 60 peticiones PUT tiene un rendimiento optimo, ya que los tiempos de respuesta mostrados en cada una de las peticiones de la figura 4.11 son mínimos siendo 187 ms el tiempo más elevado en esas peticiones. El throughput de 3,1 p/s es acorde a los parámetros del escenario, ya que cada usuario solo envió 3 solicitudes al sistema.

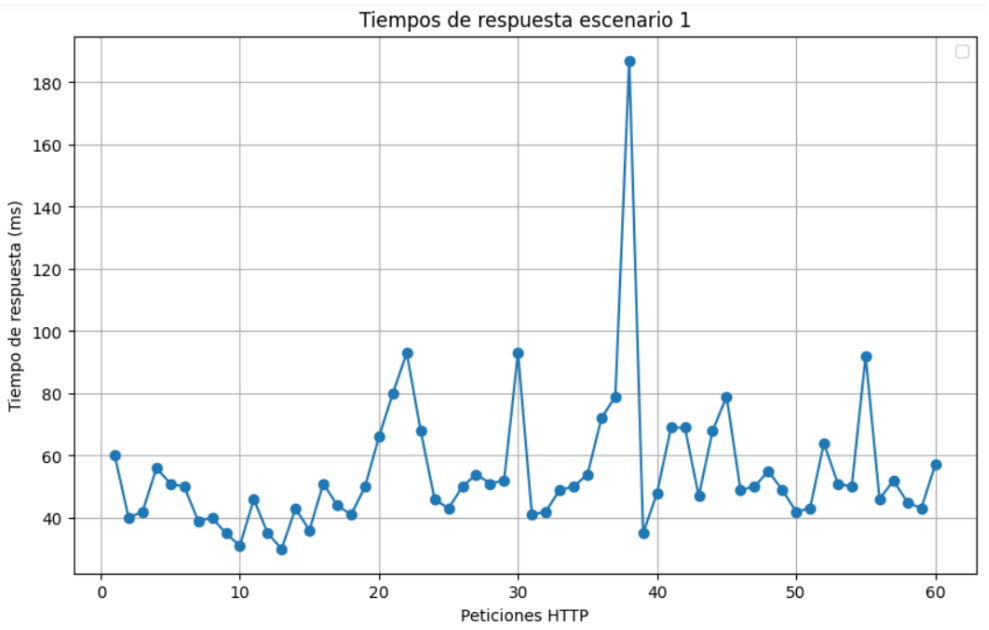



**Figura 4.11 Tiempos de respuesta del escenario 1**

En la figura 4.12 se muestra el comportamiento del sistema en el escenario 1, donde el contenedor Keyrock es el que más recursos de CPU consume debido a que se encargó de verificar el rol y el permiso de cada usuario que ingresa al sistema, así como de generar el token; seguido de Wilma-pep-proxy que se encargó de verificar la autenticidad del token junto con keyrock para dar acceso al usuario, mientras que las bases de datos tienen un consumo de recursos de CPU bajo, pero consumen más memoria RAM debido a que almacenan datos en cache para acelerar las consultas y disminuir la carga en el almacenamiento de disco, haciendo así más eficiente al sistema de comunicación

| CONTAINER ID | NAME | CPU % | MEM USAGE / LIMIT | MEM % | NET I/O | BLOCK I/O | PIDS |
|---|---|---|---|---|---|---|---|
| 539c3bc3e7e9 | fiware-pep-proxy | 2.22% | 56.45MiB / 3.785GiB | 1.46% | 283MB / 265MB | 0B / 0B | 21 |
| b2698ecf986 | fiware-orion | 0.69% | 12.83MiB / 3.785GiB | 0.33% | 325MB / 222MB | 0B / 0B | 4 |
| 5b775c183a00 | fiware-keyrock | 8.49% | 84.3MiB / 3.785GiB | 2.18% | 485MB / 395MB | 0B / 0B | 46 |
| 8d24a44b2bac | mysql-db | 1.19% | 224.3MiB / 3.785GiB | 5.79% | 281MB / 435MB | 0B / 0B | 37 |
| ace6a4749815 | mongo-db | 1.37% | 257.7MiB / 3.785GiB | 6.65% | 135MB / 217MB | 0B / 0B | 57 |

**Figura 4.12 Recursos empleados en el escenario 1**

- **Escenario 2**

En la tabla 4.4 se muestran los resultados obtenidos en el escenario 2 luego de realizar peticiones GET en el sistema donde todas fueron aceptadas, para verificar el comportamiento del sistema ante estas peticiones.

**Tabla 4.4 Resultados del escenario 2**

| Solicitudes | # Peticiones | Tiempos de respuesta (ms) | | | Throughput (rendimiento) |
|---|---|---|---|---|---|
| | | Prom. | Máx. | Min. | Paquetes /s |
| Petición HTTP | 140 | 40 | 68 | 28 | 7,1 |

En base a los resultados, se muestra que el sistema al procesar 140 peticiones GET posee un rendimiento optimo, porque los tiempos de respuesta de cada petición son mínimos como se observa en la figura 4.13, siendo 68 ms el tiempo más elevado en este escenario. El throughput de 7,1 p/s es eficiente ya que el número de usuarios aumentó y por ende el número de peticiones.

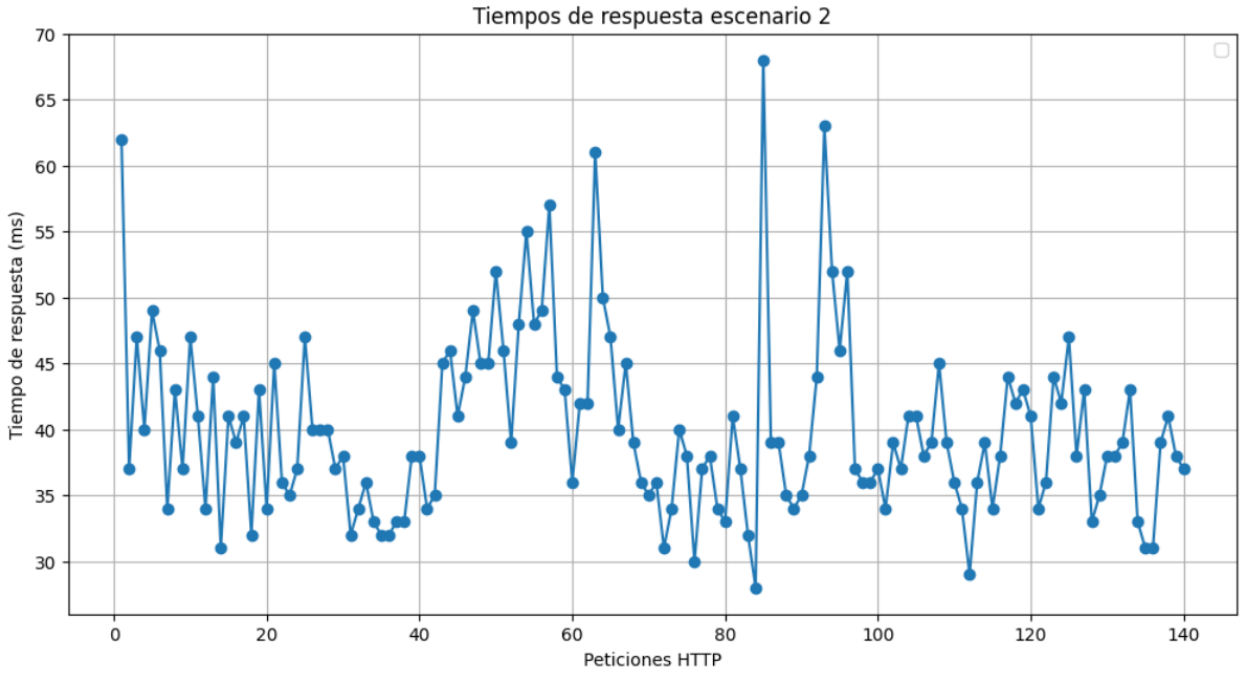
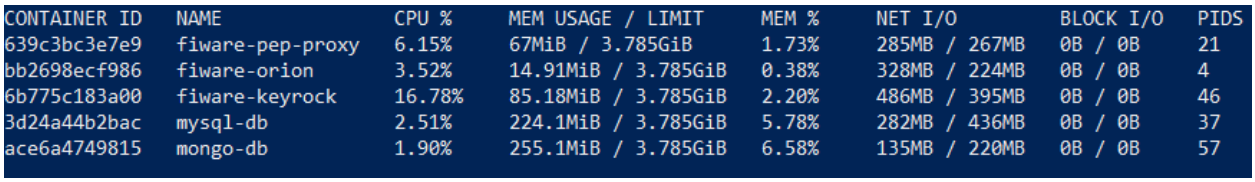



**Figura 4.13 Tiempos de respuesta del escenario 2**

En la figura 4.14 se observa que Keyrock es el contenedor que más recursos de CPU consume debido a que se encargó de verificar el rol y el permiso de cada usuario que ingresa al sistema, así como de generar el token; seguido de Wilma-pep-proxy que se encargó de verificar la autenticidad del token junto con keyrock para dar acceso al usuario, y las bases de datos tienen un consumo de recursos de CPU bajo, pero consumen más memoria RAM debido a que almacenan datos

en cache para acelerar las consultas y disminuir la carga en el almacenamiento de disco, haciendo así más eficiente al sistema de comunicación



| CONTAINER ID | NAME | CPU % | MEM USAGE / LIMIT | MEM % | NET I/O | BLOCK I/O | PIDS |
|---|---|---|---|---|---|---|---|
| 639c3bc3e7e9 | fiware-pep-proxy | 6.15% | 67MiB / 3.785GiB | 1.73% | 285MB / 267MB | 0B / 0B | 21 |
| bb2698ecf986 | fiware-orion | 3.52% | 14.91MiB / 3.785GiB | 0.38% | 328MB / 224MB | 0B / 0B | 4 |
| 6b775c183a00 | fiware-keyrock | 16.78% | 85.18MiB / 3.785GiB | 2.20% | 486MB / 395MB | 0B / 0B | 46 |
| 3d24a44b2bac | mysql-db | 2.51% | 224.1MiB / 3.785GiB | 5.78% | 282MB / 436MB | 0B / 0B | 37 |
| ace6a4749815 | mongo-db | 1.90% | 255.1MiB / 3.785GiB | 6.58% | 135MB / 220MB | 0B / 0B | 57 |

**Figura 4.14 Recursos empleados en el escenario 2**

## 4.4 Ejecución de pruebas de seguridad

### 4.4.1 Pruebas en el cifrado AES

En primer lugar, se registraron los tiempos de cifrado y descifrado cada que un usuario accedía al sistema, dichos tiempos se muestran en la tabla 4.5, y en base a esto se puede verificar que los tiempos de cifrado y descifrado son mínimos en comparación con otros mecanismos de cifrado como RSA, ya que en [42] se observó tiempos de cifrado y descifrado mucho mayores a los tiempos obtenidos para este sistema de comunicación, demostrando la viabilidad del sistema, optimizando la fluidez en los procesos de seguridad a través del mecanismo AES.

**Tabla 4.5 Tiempos de cifrado y descifrado del sistema**

| Acción | Tiempo Prom. | Tiempo Max. | Tiempo Min. |
|---|---|---|---|
| Cifrado | 0.14 ms | 4.69 ms | 0.006 ms |
| Descifrado | 0.019 ms | 4.79 ms | 0.005 ms |

En la figura 4.15 se muestra la ejecución de un algoritmo que simula un ataque de fuerza mostrado en el anexo 4, para verificar si la clave de 32 bytes empleada en el cifrado puede ser detectada y con ello descifrar la información de las HCE, para ello se tomó un dato de muestra ya cifrado de una HCE para ver si el ataque podía descifrarlo, y luego de un tiempo prolongado de ejecución el ataque no pudo encontrar la clave y por lo tanto no pudo descifrar el dato de prueba, ya que probar todas las combinaciones posibles en base al tamaño de la clave resulta una tarea computacionalmente imposible de realizar, demostrando así la robustes y eficiencia del cifrado AES, garantizando la confidencialidad e integridad de los datos en el sistema de comunicación.

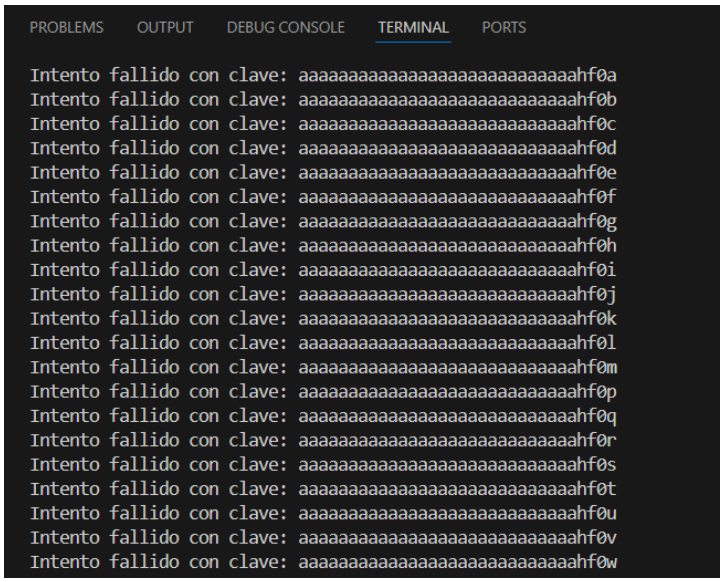



**Figura 4.15 Ataque de fuerza bruta**

### 4.4.2 Pruebas de autenticación y autorización

Se utilizó *Jmeter* para simular el acceso de 50 usuarios, de los cuales 25 estaban registrados en el sistema con su rol asignado, y el resto eran usuarios no autorizados, en las figuras 4.16 y 4.17 se muestran los resultados de las peticiones hechas al mismo tiempo por cada usuario tanto para Keyrock, como para Wilma-pep-proxy, dando como resultado que el sistema de comunicación funciona de forma correcta al momento al momento de la autenticación y autorización, ya que las peticiones con mayor tamaño en bytes tanto en Keyrock y Wilma-pep-proxy, pertenecen a los usuarios registrados en el sistema de comunicación, lo que demuestra que las configuraciones de autenticación, permisos y roles están siendo respetados para que solo usuarios autorizados accedan a la información, bloqueando o negando el acceso a usuarios que no pertenezcan al sistema.

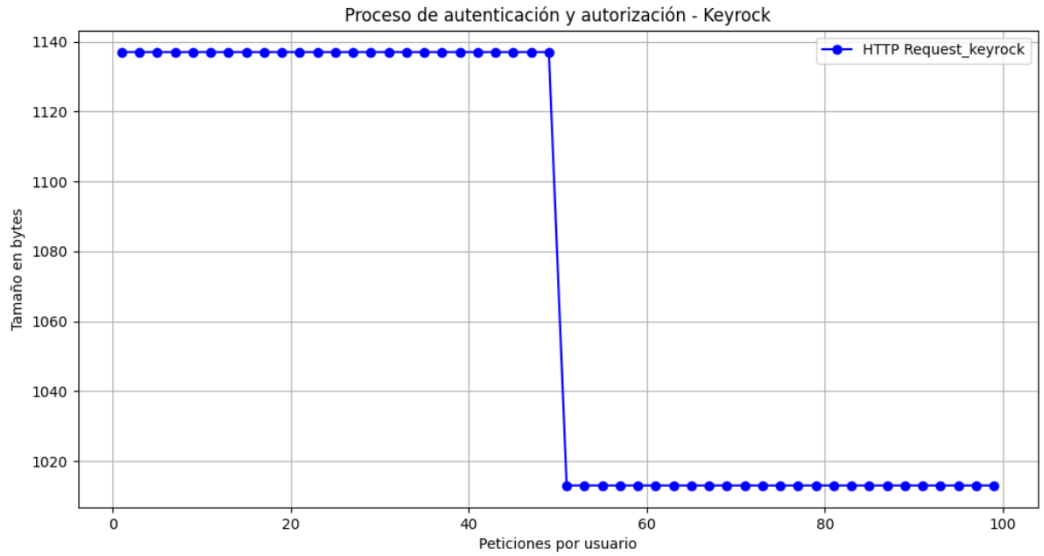



**Figura 4.16 Autenticación y autorización Keyrock**

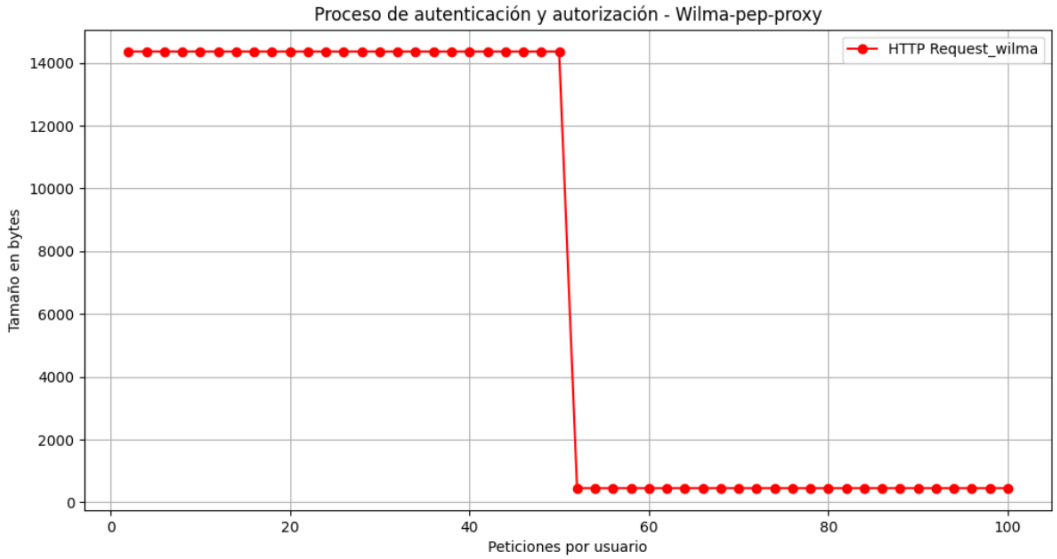



**Figura 4.17 Autenticación y autorización Wilma-pep-proxy**

### 4.4.3 Sobrecargo del sistema

Se utilizó *Jmeter* para simular un ataque de denegación de servicios (DoS por sus siglas en inglés), al punto que se simuló una inundación de peticiones al sistema, para ello se hicieron peticiones PUT de forma constante cada 2 segundos, con 500 usuarios por 1 minuto.

En la tabla 4.6 se muestran los resultados obtenidos, cuando el sistema tiene un exceso de peticiones que procesar.

**Tabla 4.6 Resultados prueba DoS**

| Solicitudes | # Peticiones | Tiempos de respuesta (ms) | | | Throughput (rendimiento) |
|---|---|---|---|---|---|
| | | Prom. | Máx. | Min. | Paquetes /s |
| Petición HTTP | 3552 | 8971 | 12154 | 4463 | 52,4 |

Según los resultados obtenidos, se verificó la eficiencia del sistema de comunicación, ya que no presentó errores al procesar 3552 peticiones, aceptando todas, demostrando que las configuraciones de autenticación y autorización no presentan fallos, y un throughput de 52,4 p/s muestra que el sistema puede manejar peticiones significativamente seguidas sin saturarse.

En la figura 4.18, se evidenció un alto procesamiento de los contenedores, lo cual es normal con una gran cantidad de peticiones por procesar y de forma ininterrumpida en 1 minuto, sin embargo, esto no afectó la capacidad del sistema en procesar todas las peticiones sin fallo alguno, además el consumo de memoria RAM no sobrepasa el 6.65% lo que demuestra la eficiencia del sistema en el uso de recursos.

```
CONTAINER ID   NAME              CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O   PIDS
639c3bc3e7e9   fiware-pep-proxy  84.02%    110.6MiB / 3.785GiB   2.85%     268MB / 251MB   0B / 0B     21
bb2698ecf986   fiware-orion      36.15%    17.46MiB / 3.785GiB   0.45%     308MB / 211MB   0B / 0B     17
6b775c183a00   fiware-keyrock    102.78%   138MiB / 3.785GiB     3.56%     453MB / 368MB   0B / 0B     45
3d24a44b2bac   mysql-db          11.29%    224.2MiB / 3.785GiB   5.78%     262MB / 406MB   0B / 0B     37
ace6a4749815   mongo-db          41.83%    257.9MiB / 3.785GiB   6.65%     126MB / 208MB   0B / 0B     57
```

**Figura 4.18 Recursos empleados en la prueba DoS**

Aunque simular 500 usuarios de forma simultánea con un periodo de envío de solicitudes de 2 segundos es poco común, dieron como resultado tiempos de respuesta elevados como se observa en la figura 4.19, esto es normal cuando se tiene una gran cantidad de peticiones por procesar, en un estudio de rendimiento para una plataforma IoT en escenarios sanitarios [45], el autor ejecutó una simulación en *Jmeter* para procesar 5000 peticiones y en promedio los tiempos de respuesta fueron de 17473 ms, con lo cual se evidencia la confiablidad del sistema de comunicación, ya que mantiene la integridad y funcionalidad en situaciones de alta demanda.

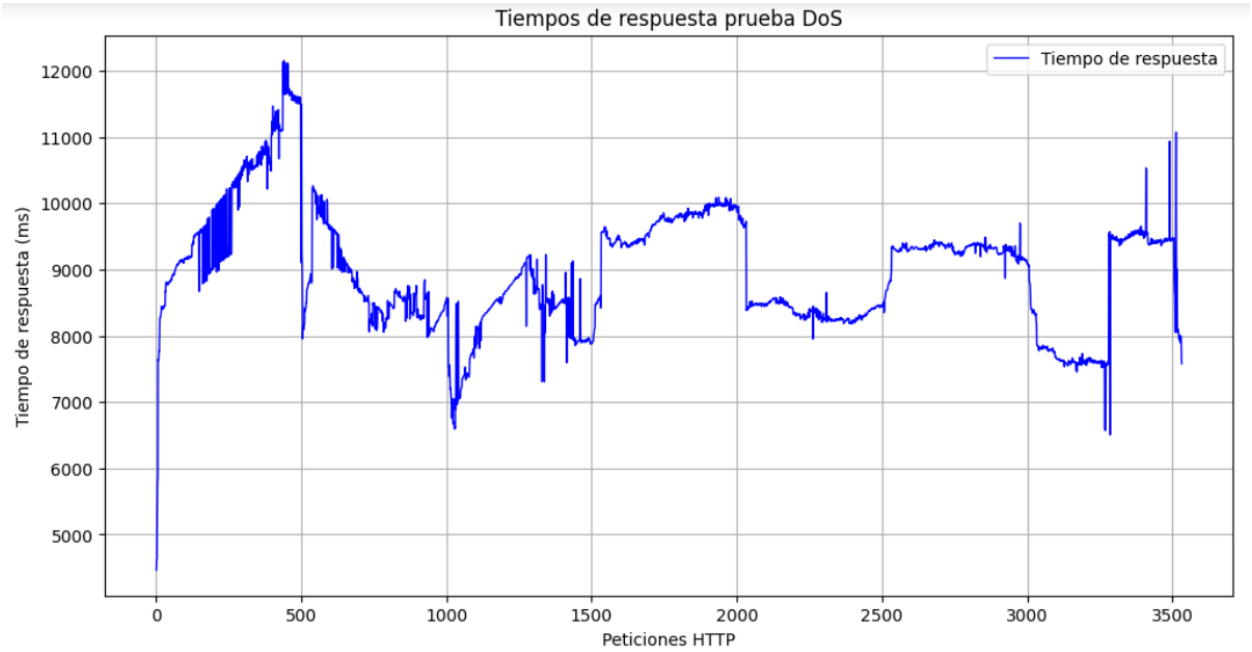



**Figura 4.19 Tiempos de respuesta en la prueba DoS**

## 4.5 Evaluación en base a las normativas de protección de datos

A continuación, se muestra la evaluación del sistema de comunicación en base a las recomendaciones de la HIPAA y GRPR.

En cuanto a la normativa HIPAA el sistema cumple con las siguientes recomendaciones:

**Tabla 4.7 Evaluación del sistema normativa HIPAA**

| Recomendación | Cumplimento |
|---|---|
| Cifrado de datos | Se implementó el mecanismo AES para cifrar los datos de las HCE |
| Autenticación de usuarios | Se empleo autenticación de 2 factores mediante Keyrock, y el protocolo Oauth2 para la autorización. |
| Control de acceso | Se empleo Wilma-pep-proxy para la validación del token y el control de acceso. |
| Auditorias y seguimientos | Los logs de Keyrock y Wilma-pep-proxy contienen información de los accesos al sistema, lo que ayudaría en procesos de auditoría, para esto se ingresaría con el comando *docker logs nombre-del-contenedor*. |

| | |
|---|---|
| Integridad de datos | Solo el administrador puede manipular todos los datos de las HCE. |
| Evaluación continua de riesgos | Se ejecutaron pruebas de seguridad en base a métricas de evaluación para verificar el funcionamiento optimo del sistema. |

Mientras que para la normativa GDPR el sistema cumple con las siguientes recomendaciones:

**Tabla 4.8 Evaluación del sistema normativa GDPR**

| Recomendación | Cumplimento |
|---|---|
| Cifrado de datos | Se implementó el mecanismo AES para cifrar los datos de las HCE |
| Autenticación y control de acceso | Se empleo autenticación de 2 factores mediante Keyrock, protocolo Oauth2 para la autorización y que junto a Wilma-pep-proxy ayudaron a la validación del token y el control de acceso. |
| Registro de acceso y auditoria | Los registros de acceso se los puede ver en los logs de Keyrock y Wilma-pep-proxy por el comando *docker logs nombre-del-contenedor,* para procesos de auditoría. |
| Evaluación de impacto relativa a la protección de datos | Se ejecutaron pruebas de seguridad en base a métricas de evaluación para verificar el funcionamiento optimo del sistema. |

A su vez ciertas recomendaciones de las normativas HIPAA y GDPR no fueron tomadas en cuenta para este proyecto por los siguientes motivos[24] [25]:

- Entrenamiento o capacitación al personal de salud: el sistema está en proceso de desarrollo.
- Controles de acceso a las instalaciones físicas: el sistema se está ejecutando de manera local.

[24] https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164

[25] https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=celex%3A32016R0679

- Plan de contingencia: el trabajo se enfoca más en la parte tecnológica de seguridad, por lo que no cubre planes ante desastres naturales, fuego o vandalismo.
- Consentimiento del paciente: el sistema está en desarrollo por lo que no se requirió de pacientes o datos médicos reales.
- Representante de la Unión Europea (UE): el sistema está en desarrollo y por ende no requiere de representantes para el tratamiento de datos médicos.

Finalmente, en base a las tablas 4.7 y 4.8 y al desarrollo de este trabajo, el sistema de comunicación segura para la transmisión de HCE cumple con las principales recomendaciones de las normativas HIPAA y GDPR, para la protección de datos médicos sensibles.

## 4.6 Análisis de resultados

Basados en los resultados se pudo verificar que la gestión de usuarios está bien configurada, ya que se respetaron los roles y permisos de cada usuario al momento de realizar las pruebas de funcionamiento, especialmente en la parte de autenticación y autorización.

Los resultados obtenidos en cada una de las pruebas realizadas mostraron que, en condiciones normales, los tiempos de respuesta son menores a los 187ms, demostrando la eficacia del sistema en el procesamiento de peticiones. Los contenedores Keyrock y Wilma-pep-proxy son los que más recursos consumen en cuanto al procesamiento de CPU, ya que se encargaron de la parte de autenticación, autorización y control de acceso para el sistema de comunicación, mientras que ambas bases de datos consumen más memoria RAM debido a que almacenan en cache datos que se consultan con bastante frecuencia para agilizar las consultas y evitar una sobrecarga en el almacenamiento de disco, adicional cabe recalcar que en todas las pruebas el consumo de memoria RAM no sobrepasó el 6,65%, mostrando la eficiencia del sistema de comunicación en cuanto al uso de recursos.

Por el uso del cifrado AES, los tiempos de respuesta no se vieron interrumpidos, demostrando la robustes y eficiencia que tiene este cifrado en sistemas donde se requiere tiempos de respuesta rápidos.

Finalmente, el sistema fue desarrollado en torno a las principales recomendaciones de las normativas HIPAA y GDPR para que el sistema cumpla la función de salvaguardar la información contenidas en las HCE.

# Conclusiones

- La implementación del cifrado AES demostró ser eficiente en el sistema de comunicación, ya que los tiempos de cifrado y descifrado al ser mínimos no interfirieron con los tiempos de respuesta en las peticiones que cada usuario hacia al sistema, asegurando la confidencialidad e integridad de las HCE sin descuidar la fluidez del sistema. De la misma forma demostró ser un cifrado robusto ya que al momento de ejecutar el ataque de fuerza bruta sobre el dato cifrado de muestra, el ataque no pudo encontrar la clave de cifrado y con ello resulto imposible que el dato vaya a ser descifrado, indicando que el sistema está protegiendo de forma adecuada los datos médicos de las HCE.
- Se empleó una autenticación de dos factores mediante Keyrock, esto ayudó a que los factores de riesgo asociados a accesos no autorizados se reduzcan de manera significativa, ya que se requiere de una contraseña y token para poder acceder al sistema.
- A través de Keyrock se aplicó el protocolo OAuth2 como mecanismo de autorización y en conjunto con Wilma-pep-proxy brindaron un acceso seguro a los usuarios autorizados para que los mismos puedan crear, modificar, eliminar o leer los datos de las HCE según el rol y los permisos que estos tengan.
- En base a las métricas de evaluación que se plantearon, en los escenarios de simulación se verificó que el sistema de comunicación funciona de manera óptima, ya que los tiempos de respuesta, rendimiento y consumo de recursos fueron acordes al número de peticiones y usuarios que se utilizaron en cada escenario, demostrando que la arquitectura basada en contendores FIWARE es eficiente y a la vez puede ser escalable.
- Las pruebas de seguridad demostraron la efectividad del sistema en prevenir ataques o vulnerabilidades, ya que el mismo mantiene la información cifrada tanto en tránsito y reposo, a su vez pudo prevenir el acceso no autorizado al sistema, asegurando que solo los usuarios autorizados puedan acceder a las HCE según su rol, y se comprobó que ante grandes cargas de trabajo el sistema pudo gestionar correctamente los procesos de autenticación y autorización en cada

petición sin llegar a saturarse, demostrando robustez y garantizando así la confiabilidad del sistema.

- El sistema de comunicación desarrollado cumple con las principales recomendaciones de seguridad de las normativas HIPAA y GDPR, como la implementación de un mecanismo de cifrado robusto, métodos de autenticación, autorización, control de acceso, y la ejecución de pruebas de seguridad, garantizando la confiabilidad, integridad y disponibilidad del sistema en el manejo de los datos médicos de las HCE.
- La arquitectura basada en contenedores FIWARE demostró ser segura, eficiente y escalable, lo que permitiría que el sistema de comunicación segura pueda ser adaptado en entornos con mayor demanda.

# Recomendaciones

- Pese a que el sistema demostró ser capaz de procesar un número considerable de peticiones en muy poco tiempo, es recomendable añadir componentes que puedan limitar la cantidad de peticiones que se envían al sistema para que el mismo no tenga problemas, y un componente podría ser Nginx[26], que es un proxy inverso el cual actuaría como un limitador de peticiones, protegiendo al sistema de una forma más eficiente.
- La arquitectura basada en contenedores FIWARE se permitiría emplear agentes IoT para añadir una capa más de seguridad en el proceso de autenticación, mediante el uso de sensores biométricos que detecten los rasgos físicos del usuario y que Keyrock los procese en conjunto con la contraseña y el token previamente definidos.
- Se recomienda realizar pruebas de estrés con cargas de trabajo mucho mayores a los presentados en las pruebas de seguridad, con el fin de evaluar los límites del sistema de comunicación.

[26] https://docs.nginx.com/nginx-management-suite/acm/how-to/policies/rate-limit/

# Bibliografía


[1] C. Graf, «Tecnologías de información y comunicación (TICs). Primer paso para la implementación de TeleSalud y Telemedicina,» *Rev. parag. reumatol.,* vol. 6, nº 1, pp. 1-4, Jul. 2020.

[2] A. Cervera García y A. Goussens, «Ciberseguridad y uso de las TIC en el Sector Salud,» *Atencion Primaria,* vol. 56, nº 3, pp. 1-7, Mar. 2024.

[3] J. Gil Yacobazzo y M. J. Viega Rodríguez, «Historia clínica electrónica: confidencialidad y privacidad de los datos clínicos,» *Revista Médica del Uruguay,* vol. 34, nº 4, pp. 102-119, Dic. 2018.

[4] R. Aleixandre-Benavent, A. Ferrer-Sapena y F. Peset, «Informatización de la historia clínica en España,» *Información Biomédica y Farmacéutica,* vol. 19, nº 3, pp. 231-239, Jul. 2010.

[5] Cyber Readiness Institute, «Prácticas recomendadas sobre ciberseguridad para la telemedicina,» 17 Nov 2021. [Online]. Available: https://cyberreadinessinstitute.org/wp-content/uploads/CRI-Cybersecurity-Best-Practices-for-Telehealth-es.pdf.

[6] E. Guillen, L. Ramirez y E. Estupiñan, «ANÁLISIS DE SEGURIDAD PARA EL MANEJO DE LA INFORMACIÓN MÉDICA EN TELEMEDICINA,» *CIENCIA E INGENIERÍA NEOGRANADINA,* vol. 21, nº 2, pp. 57-89, Dic. 2011.

[7] J. F. Rodríguez Ayuso, «La disrupción tecnológica En El ámbito Sanitario Europeo: Implicaciones De La Telemedicina Pública En La protección De Datos De Los Pacientes,» *Cuadernos Europeos De Deusto,* nº 69, pp. 29-55, Sep. 2023.

[8] S. Nass, L. Levit y L. Gostin, Beyond th Hipaa privacy rule, Washington: The national academies press, 2009.

[9] ISSA, «Detectar el fraude en la atención de salud mediante las tecnologías emergentes,» 04 Jul. 2022. [Online]. Available: https://www.issa.int/es/analysis/detecting-fraud-health-care-through-emerging-technologies.

[10] J. Díez González, «Ciberseguridad en el sector salud: características, amenazas y recomendaciones,» 25 Ene. 2024. [Online]. Available: https://www.incibe.es/incibe-cert/blog/ciberseguridad-en-el-sector-salud-caracteristicas-amenazas-y-recomendaciones#:~:text=Seg%C3%BAn%20el%20estudio%20(ENISA%3A%20TL 2023,%25)%20o%20energ%C3%ADa%20(4%25)..

[11] N. Tejo-Machado, F. Rodrigues-Martinez-Basile, F. Cezar-Amate y L. Ramírez-López, «Protocolo de informática forense ante ciberincidentes en telemedicina para preservar información como primera respuesta,» *Revista Científica General José María Córdova,* vol. 19, nº 33, p. 181–203, Ago. 2021.

[12] D. Coulombie, A. Reyes y A. Miguens, «Impacto de una debilidad de ciberseguridad en la arquitectura de un sistema electromédico,» *CACIC 2020,* pp. 380-388, Oct. 2020.

[13] Avast Business, «Ransomware en un hospital del Servicio Público de Salud del Reino Unido,» Avast Business, 29 Jun. 2024. [Online]. Available: https://www.avast.com/es-ww/business/resources/what-is-hospital-ransomware#pc.

[14] E. Dias, A. Seidel, I. Bandeira y e. al, «Guidelines adopted by agile teams in privacy requirements elicitation after the Brazilian general data protection law (LGPD) implementation,» *Requirements Engineering,* vol. 27, nº 4, pp. 545-567, Nov. 2022.

[15] ASAMBLEA NACIONAL DE LA REPUBLICA DEL ECUADOR, «LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES,» 26 May. 2021. [Online]. Available: https://www.telecomunicaciones.gob.ec/wp-content/uploads/2021/06/Ley-Organica-de-Datos-Personales.pdf.

[16] R. Albarracín, «Resultados de la implementación de registros médicos,» *INGENIO,* vol. 1, nº 1, pp. 5-14, Mar. 2018.

[17] D. L. Guerra y M. E. López, «Evaluación de la calidad en la prescripción de medicamentos antes y después de la implementación de un módulo de prescripción en historias clínicas electrónicas, de pacientes con diagnóstico de neumonía,» *Tesis de especialidad médica, Fac. de med., Puce, Quito, Ecuador,* 2017.

[18] H. Huang, T. Gong, N. Ye, R. Wang y Y. Dou, «Private and Secured Medical Data Transmission and Analysis for Wireless Sensing Healthcare System,» *IEEE Transactions on Industrial Informatics,* vol. 13, nº 3, pp. 1227-1237, Jun. 2017.

[19] V. Facco Rodrigues, E. Palma Paim, R. Kunst y et al., «Exploring publish/subscribe, multilevel cloud elasticity, and data compression in telemedicine,» *Computer Methods and Programs in Biomedicine,* vol. 191, Jul. 2020.

[20] D. E. Ströckl, E. Oberrauner, J. Plattner y et al., «Smart City Technology meets Smart Health Assistive Systems – on the example of the project AYUDO,» *PETRA '20: Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments,* nº 50, pp. 1-7, Jun. 2020.

[21] K. M. Tsiouris, D. Gatsios, V. Tsakanikas y et al., «Designing interoperable telehealth platforms: bridging IoT devices with cloud infrastructures,» *Enterprise Information Systems,* vol. 14, nº 8, p. 1194–1218, Abr. 2020.

[22] D. AlOsail, N. Amino y N. Mohammad, «Security Issues and Solutions in E-Health and Telemedicine,» *Computer Networks, Big Data and IoT,* vol. 66, pp. 305-318, Jun. 2021.

[23] P. Olivarez, A. Lezcano Gil y A. Mendoza De Los Santos, «Principales técnicas criptográficas aplicadas a la seguridad de la información en IoT: una revisión sistemática,» *Ingenio Tecnologico ,* vol. 5, nº 41, Nov. 2023.

[24] H. El Zoukaa y M. Hosni, «Secure IoT communications for smart healthcare monitoring,» *Internet of Things,* vol. 13, pp. 1-14, Mar. 2021.

[25] A. Shaikh, M. Al Reshan, A. Sulaiman y H. Alshahrani, «Secure Telemedicine System Design for COVID-19 Patients Treatment Using Service Oriented Architecture.,» *Sensors ,* vol. 22, nº 3, p. 952, Ene. 2022.

[26] H. Chen, D. Ding, L. Zhang, C. Zhao y X. Jin, «Secure and resource-efficient communications for telemedicine systems,» *Computers & Electrical Engineering ,* vol. 98, nº 4, p. 107659, Mar. 2022.

[27] B. PUSHPA, «Hybrid Data Encryption Algorithm for Secure Medical Data Transmission in Cloud Environment,» *Fourth International Conference on Computing Methodologies and Communication (ICCMC),* pp. 329-334, May. 2020.

[28] R. Denis y P. Madhubala, «Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems.,» *Multimed Tools Appl,* vol. 80, p. 21165–21202, Mar. 2021.

[29] J. Sánchez, L. López y J. Martinez, «Solución para garantizar la privacidad en internet de las cosas,» *EPI,* vol. 24, nº 1, pp. 62-70, Ene. 2015.

[30] R. Bashshur, G. Shannon y B. Smith, «The Empirical Foundations of Telemedicine Interventions for Chronic Disease Management,» *TELEMEDICINE and e-HEALTH,* vol. 20, nº 9, pp. 769-800, Sep. 2014.

[31] W. Velasquez, «Tecnologías para telemedicina,» de *Telemedicina, Maestría en ingenieria biomedica*, Guayaquil, 2024.

[32] C. Kruse, B. Frederick, T. Jacobson y D. Monticone, «Cybersecurity in healthcare: A systematic review of modern threats and trends,» *Technol Health Care,* vol. 25, nº 1, pp. 1-10, 2017.

[33] Fortinet, «Tipos de ciberataques: ataque DDoS, ransomware y más,» 20 Ene. 2021. [Online]. Available: https://www.fortinet.com/lat/resources/cyberglossary/types-of-cyber-attacks.

[34] J. Katz y Y. Lindell, INTRODUCTION TO MODERN CRYPTOGRAPHY, Boca Raton, FL: CRC Press, 2015.

[35] A. Wangenheim, D. Spagnuelo, T. Idalino y Et.al, «MULTI-FACTOR AUTHENTICATION FOR TELEMEDICINE USING MOBILE DEVICES AND ONE-TIME PASSWORDS,» *CBIS,* vol. 1, pp. 1041-1050, Nov. 2016.

[36] M. Papathanasaki, L. Maglaras y N. Ayres, «Modern Authentication Methods: A Comprehensive Survey,» *IntechOpen,* vol. 0, pp. 1-24, Jun. 2022.

[37] M. Bermeo, «Las buenas prácticas para el tratamiento del dato de salud,» *Revista Ruptura ,* pp. 219-243, Feb. 2023.

[38] NIH, «historia clínica electrónica,» 29 Mar 2021. [Online]. Available: https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/historia-clinica-electronica.

[39] J. Nelson, G. Cafagna y L. Tejerina, «Electronic Health Record Systems: Definitions, Evidence, and Practical Recommendations for Latin America and the Caribbean,» Abr. 2020. [Online]. Available: http://dx.doi.org/10.18235/0002240.

[40] S. Loss, P. S. Har y F. L. Nelio Cacho, «Using FIWARE and Blockchain in Post Pandemic Vaccination Scenario,» *BCCA,* pp. 143-150, 2021.

[41] NIST, Advanced Encryption Standard (AES), Gaithersburg: NIST, May, 2023.

[42] E. Alvarado, «ESTUDIO DE EFICIENCIA Y EFICACIA DE LOS ALGORITMOS CRIPTOGRÁFICOS RSA, AES, IDEA, RC4 EN LA SEGURIDAD INFORMÁTICA,» *Trabajo de especialización, Esc.C.B. Tecnol. e Ing., UNAD, Bogotá, Colombia,* 2020.

[43] M. Montmollin, «SaludAR+: diseño de una aplicación de acceso al historial clínico centralizado para el sistema sanitario argentino,» *Tesis de master, Fac.d'Infor. i Mit. Audiov., Uni. Barcelona, Barcelona, España,* 2021.

[44] A. Marriaga y V. Rangel, «DISEÑO DE SISTEMA DE COMUNICACIÓN WIRELESS PARA LA ADQUISICIÓN DE DATOS DE OPERACIÓN DE UNA TURBINA EÓLICA DE 400W,» *Jóvenes en la ciencia,* vol. 2, nº 1, p. 1817–1822, Ene. 2021.

[45] I. Gomez, «Estudio de rendimiento de una plataforma IoT en escenarios sanitarios,» *Tesis de grado, Dept. Ing. Tel, Uni. Sevilla, España,* 2021.

# Anexos

### Anexo 1: Archivo docker-compose

En el anexo 1 se muestra la configuración de los contenedores FIWARE y las bases de datos que formaron parte de la arquitectura del sistema de comunicación. La configuración de cada contenedor cuenta con la imagen, el nombre del contenedor, el puerto, la red y las dependencias que tienen para su buen funcionamiento.

```
services:
  # ORION CONTEXT BROKER
  orion:
    image: fiware/orion:3.4.0
    hostname: orion
    container_name: fiware-orion
    networks:
    - my_network
    ports:
      - "1026:1026"
    environment:
      - ORION_PORT=1026
    #command: -dbURI mongodb://mongo:27017 #chequear 4.0.0
    command: -dbhost mongo -logLevel DEBUG
    depends_on:
      - mongo

  # MONGODB (base de datos para Orion)
  mongo:
    image: mongo:6.0
    hostname: mongo
    container_name: mongo-db
    networks:
    - my_network
    ports:
      - "27017:27017"
    volumes:
      - mongo-db:/data/db

  # KEYROCK - Identity Manager para la autenticación
  keyrock:
    image: fiware/idm:8.0.0
    hostname: keyrock
    container_name: fiware-keyrock
    networks:
    - my_network
```

```
  depends_on:
    - mysql
  environment:
    - DEBUG=idm:*
    - IDM_DB_HOST=mysql
    - IDM_DB_PASS=password
    - IDM_DB_USER=root
   # - IDM_DB_NAME=keyrock
    - IDM_HOST=http://localhost:3005
    - IDM_PORT=3005
    - IDM_ADMIN_USER=admin
    - IDM_ADMIN_EMAIL=admin@test.com
    - IDM_ADMIN_PASS=1234
  ports:
    - "3005:3005"

# MYSQL - Base de datos para Keyrock
mysql:
  image: mysql:5.7
  hostname: mysql
  container_name: mysql-db
  networks:
  - my_network
  environment:
    - MYSQL_ROOT_PASSWORD=password
    #- MYSQL_DATABASE=idm

  ports:
    - "3306:3306"
  volumes:
    - mysql-db:/var/lib/mysql

# PEP PROXY - Wilma Proxy
pep-proxy:
  image: fiware/pep-proxy:7.5.1
  hostname: pep-proxy
  container_name: fiware-pep-proxy
  networks:
  - my_network
  environment:

    - PEP_PROXY_APP_HOST=orion
    - PEP_PROXY_APP_PORT=1026
    - PEP_PROXY_APP_ID=f1c841b4-31e0-40c6-96df-ebae18b82c35
    - PEP_PROXY_USERNAME=pep_proxy_a9f10d54-457f-4939-bd91-3c367f8afc81
    - PEP_PASSWORD=pep_proxy_6d01bfb2-95d4-4d8d-955e-88f4e877915e
    - PEP_TOKEN_SECRET=2ba93033c9130886
```

```
      - PEP_PROXY_PORT=1027
      - PEP_PROXY_IDM_HOST=keyrock
      - PEP_PROXY_IDM_PORT=3005
      - PEP_PROXY_HTTPS_ENABLED=false
      - PEP_PROXY_AUTH_ENABLED=true
      - PEP_PROXY_IDM_SSL_ENABLED=false
      - PEP_PROXY_PDP=idm
      - PEP_PROXY_MAGIC_KEY=1234
    ports:
      - "1027:1027"
    expose:
      - "1027"
    depends_on:
      - keyrock
      - orion

networks:
  my_network:
    driver: bridge

volumes:
  mongo-db: ~
  mysql-db: ~
```

## Anexo 2: Comprobación de contenedores en la misma red

En el anexo 2 se verifica que los contendores empleados para el sistema de comunicación estén dentro de la misma red para que no exista problemas de funcionamiento, para esto se utiliza el comando *docker network inspect fiware2_my_network*

```
    "3d24a44b2bac01cb43e8cb0d86db5adafa18e6dfdaa6b4cb2dc4127a69d21dbb": {
        "Name": "mysql-db",
        "EndpointID": "d02336ee91fd3c799ad8c7df30ea9d002a7ee41050a04112ff1c0acf691c41c6",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
    },
    "639c3bc3e7e96ab8148e1056eb0143e2f750e572dbb03aec50d889258e9faa36": {
        "Name": "fiware-pep-proxy",
        "EndpointID": "4a804aef4f782c2b0a458d10e373bad89cc055a29a7f75299f2cd7863bfdd419",
        "MacAddress": "02:42:ac:12:00:06",
        "IPv4Address": "172.18.0.6/16",
        "IPv6Address": ""
    },
    "6b775c183a00ee45ec718177f300c058a73313f185b70fd5be9cedfad5764c31": {
        "Name": "fiware-keyrock",
        "EndpointID": "1caa377667e5feb228896dd548d2e89a0ebc585b1356fdebd20aa1da11fc6ed0",
        "MacAddress": "02:42:ac:12:00:05",
        "IPv4Address": "172.18.0.5/16",
        "IPv6Address": ""
    },
    "ace6a47498159e426c84466742118a761d07b4a65ea9e0628392b2591b2bd15c": {
        "Name": "mongo-db",
        "EndpointID": "4517777f09fae3157db3bba42d39343ab88f8f54bbe813ef532030c8b6d0ef72",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
    },
    "bb2698ecf986326af2e5943e2bfb61f0e2f308848122dae4efd6b9abecf39c1c": {
        "Name": "fiware-orion",
        "EndpointID": "85bc183c7aa7a83d7cbee38e690f2431800e3931eb130fd1126faddda67a5eaa",
        "MacAddress": "02:42:ac:12:00:04",
        "IPv4Address": "172.18.0.4/16",
        "IPv6Address": ""
    }
},
"Options": {},
"Labels": {
    "com.docker.compose.network": "my_network",
    "com.docker.compose.project": "fiware2",
    "com.docker.compose.version": "2.29.1"
}
```

### Anexo 3: Modelo de HCE en formato JSON

En el anexo 3 se muestra el modelo utilizado en formato JSON para la historia clínica electrónica, donde la entidad es Hce y los atributos son los que se detallan en la sección 3.5.1.


```
{

        "id": "Hce:012",
        "type": "Hce",
        "1_nombre_paciente": {
            "type": "Text",
            "value": "Juan Luis Diaz Mora"

        },
        "2_ID": {
            "type": "StructuredValue",
            "value": "0665100185"
        },
        "3_fecha_nacimiento": {
            "type": "Date",
            "value": "1980-01-01"
        },
        "4_genero": {
            "type": "Text",
            "value": "Masculino"
        },
        "5_contacto": {
            "type": "Text",
            "value":
                "celular: 0986035577 / direccion: 123 calle sn / email:
juan.diaz@gmail.com"


        },
        "6_historial_med": {
            "type": "Text",
            "value":
                "Hipertension, Diabetes"

        },
        "7_medicacion": {
            "type": "Text",
            "value":
                "Moxicilina e Hibuprofeno"

        },
        "8_alergia": {
            "type": "Text",
```

```
            "value":  "Penicilina"

        },
        "9_ultimo_ingreso": {
            "type": "Date",
            "value": "2024-11-03"
        }

}
```

### Anexo 4: Algoritmo de ataque de fuerza bruta

En este anexo se muestra el código que se empleó para el ataque de fuerza bruta, en el que se observa que el dato de muestra de una de las HCE cifradas y almacenadas en mongoDB es *+Pl2n/pSW6CGwuwE0mrn2Q*, y el resultado en caso de que el ataque lograra encontrar la clave de 32 bytes debería ser *1987541258.*

```python
import base64
from Crypto.Cipher import AES
from hashlib import sha256
import itertools

ciphertext_base64 = "+Pl2n/pSW6CGwuwE0mrn2Q=="  # Reemplazar con el mensaje cifrado desde PHP
expected_plaintext = "1987541258"  # Texto descifrado que se busca
iv = b"\x00" * 16  # IV fijo de ceros como en el script PHP

# Función para derivar la clave

def derive_key(key_candidate):
    return sha256(key_candidate.encode('utf-8')).digest()

# Función para descifrar con una clave
def try_decrypt(ciphertext, key_candidate):
    key = derive_key(key_candidate)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    try:
        decrypted = cipher.decrypt(ciphertext).decode('utf-8').strip()
        return decrypted
    except (ValueError, UnicodeDecodeError):
        return None

# Probar todas las claves posibles
def brute_force_attack():
    ciphertext = base64.b64decode(ciphertext_base64)
```

```
    # Ejemplo: generar claves de longitud variable (alfabético)
    charset = "abcdefghijklmnopqrstuvwxyz0123456789_/*"

    max_length = 32  # Ajustar según el rango de claves posibles

    for length in range(32, max_length + 1):
        for key_candidate in itertools.product(charset, repeat=length):
            key_candidate = ''.join(key_candidate)
            decrypted = try_decrypt(ciphertext, key_candidate)
            if decrypted == expected_plaintext:
                print(f"Clave encontrada: {key_candidate}")
                print(f"Texto descifrado: {decrypted}")
                return
            print(f"Intento fallido con clave: {key_candidate}")

    print("No se encontró la clave.")

# Ejecutar el ataque
brute_force_attack()
```

### Anexo 5: Repositorio

Los códigos utilizados para el desarrollo de la aplicación del sistema de comunicación segura para transmitir HCE se encuentran en el siguiente enlace: https://github.com/jairopato/SYSTEMMEDIC