

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Desarrollo de una aplicación para realizar exámenes del
corazón en tiempo real de manera remota**

Trabajo de graduación en modalidad de Megaproyecto presentado por
Diego de Jesús Arredondo Turcios para optar al grado académico de
Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de
la Información

Guatemala,

2024

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




**Desarrollo de una aplicación para realizar exámenes del
corazón en tiempo real de manera remota**

Trabajo de graduación en modalidad de Megaproyecto presentado por
Diego de Jesús Arredondo Turcios para optar al grado académico de
Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de
la Información

Guatemala,


2024

Vo.Bo.:

(f) 

Ing. Oscar Iván Robles


Tribunal Examinador:

(f) 

Ing. Oscar Iván Robles

(f) 

Ing. Douglas Leonel Barrios González

(f) 

Ing. Leonel Guillén del Valle

Fecha de aprobación: Guatemala, 17 de junio de 2024.

Lista de figuras	VI
Resumen	VII
Abstract	VIII
I. Introducción	1
II. Justificación	2
III. Objetivos	5
A. Objetivo general	5
B. Objetivos específicos	5
IV. Marco teórico	6
A. Software médico y sus tipos	6
B. Inteligencia artificial responsable [<i>Responsible AI</i>]	7
C. Inteligencia artificial explicable [<i>Explainable AI</i>]	8
D. Inteligencia artificial	8
1. Tipos de inteligencia artificial	8
2. Machine learning	9
3. ¿Cómo Funciona?	10
E. Servicios en la nube	10
F. Websockets	10
G. Ley de responsabilidad y portabilidad del seguro médico [<i>Health Insurance Portability and Accountability Act (HIPAA)</i>]	11
1. Reglas del HIPAA	11
V. Metodología	13
A. Investigación sobre la arquitectura del sistema	13
1. Desarrollo del API	13
2. Desarrollo de la aplicación visual	14
3. Desarrollo de Base de datos	14

4.	Escalabilidad	14
5.	Seguridad	15
B.	Desarrollo del Modelo de Machine Learning	15
C.	Recolección y análisis de datos	22
D.	Reclutamiento de colaboradores	22
E.	Definición de escenarios de prueba	22
F.	Pruebas piloto y recolección de datos	22
G.	Recopilación de retroalimentación	22
H.	Análisis de resultados	22
VI.	Resultados	23
A.	Pruebas con usuarios	23
B.	Escalabilidad y pruebas de estrés	25
C.	Pruebas de seguridad	25
D.	Machine Learning	26
1.	Métricas de rendimiento con un corazón sano	26
2.	Interpretación de los resultados	27
3.	Visualización de resultados	28
4.	Métricas de rendimiento con un corazón no sano (más anomalías)	29
5.	Análisis de los resultados	29
6.	Responsabilidad de la inteligencia artificial y aplicaciones médicas en la detección de anomalías	30
E.	Usabilidad	31
F.	Consideraciones HIIPA	33
1.	Confidencialidad	33
2.	Integridad	34
3.	Disponibilidad	34
4.	Notificación y monitoreo	35
VII.	Conclusiones	36
VIII.	Recomendaciones	37
IX.	Bibliografía	38
X.	Anexos	40
A.	Diagrama y código fuente de la base de datos	40
B.	Capturas de pantalla de la pantalla principal de la página	42
C.	Documentación del API desarrollado	42
D.	Programa de machine learning para detectar anomalías en los ECG	46

Lista de figuras

1.	Enfermedades cardiovasculares, tasas de mortalidad estimadas, 2000 y 2017	4
2.	Causas de mortalidad cardiovascular, Latinoamérica, 2012	4
3.	Proceso Completo del Sistema	14
4.	Criterio que los SVM intentan optimizar	17
5.	Gráfica de funcionamiento de One-Class SVM.	19
6.	Datos para el modelo a entrenar	20
7.	Visualización de la detección de anomalías en las señales de ECG. Los puntos rojos representan las anomalías identificadas por el modelo One-Class SVM.	21
8.	Pruebas con Usuario en el módulo de ingreso	23
9.	Pruebas con Usuario en el módulo de perfil del doctor	24
10.	Pruebas con Usuario en el módulo de exámenes	24
11.	Prueba de Estrés para el servidor Web	25
12.	Test de Seguridad para el Front-End	26
13.	Test de Seguridad para el Back-End	26
14.	Resultados de las curvas ROC y PR para corazón sano	28
15.	Resultados de las curvas ROC y PR	29
16.	Resultados de la encuesta acerca de lo intuitivo del software	31
17.	Resultados de la encuesta acerca de lo que proporciona el Software	32
18.	Resultados de la encuesta acerca de la cantidad de clics	32
19.	Resultados de la encuesta acerca de la utilidad del software	33
20.	Inicio de Sesión de la Aplicación	34
21.	Monitoreo en AWS	35
22.	Diagrama de la base de datos del software de monitoreo cardíaco	40
23.	Código de la base de datos del software de monitoreo cardíaco	41
24.	Diagrama de la base de datos no Relacional del software de monitoreo cardíaco	42
25.	Código de la base de datos no Relacional del software de monitoreo cardíaco	43
26.	Captura de pantalla de la pantalla principal del software	44
27.	Captura de pantalla de la pantalla de pacientes del software	44
28.	Captura de pantalla del código implementado para los controladores del API	45
29.	Captura de pantalla del código implementado para las rutas del API	45

30.	Captura de pantalla del código implementado para la conexión del API	46
31.	Captura de pantalla de la configuración de la IA para detección de anomalías en el ECG	47
32.	Captura de pantalla de la impresión en pantalla de los resultados de la IA para detección de anomalías en el ECG	48
33.	Cumplimiento AWS HIPAA (<i>AWS HIPAA Compliance</i>)	49

En la actualidad, en muchos países de Latinoamérica, el acceso a atención médica especializada en cardiología enfrenta serias limitaciones. Estas limitaciones incluyen no solo la dificultad de acceso a consultas con cardiólogos, sino también una deficiente continuidad en el seguimiento de los tratamientos. A pesar de que los exámenes cardiológicos disponibles suelen ser precisos, presentan desafíos significativos, ya que en muchos casos deben realizarse en centros médicos específicos, y la información obtenida no puede ser analizada en tiempo real, lo que retrasa la toma de decisiones críticas.

En respuesta a estas necesidades apremiantes, se ha desarrollado una innovadora plataforma de software médico diseñada para el monitoreo remoto en tiempo real de exámenes cardíacos. Este desarrollo tiene como principal objetivo mejorar la accesibilidad, precisión y continuidad de la atención cardiológica. La plataforma permite la visualización y análisis de datos cardíacos generados por hardware especializado, integrando tecnologías avanzadas para que tanto los pacientes como los profesionales de la salud puedan acceder a resultados de manera inmediata desde cualquier ubicación geográfica.

Palabras clave: Cardiología, Exámenes cardiológicos, Software médico, Monitoreo remoto en tiempo real, Análisis de datos cardíacos

Currently, in many Latin American countries, access to specialized cardiology care faces serious limitations. These limitations include not only the difficulty of accessing consultations with cardiologists, but also poor continuity in the follow-up of treatments. Although the available cardiology tests are usually accurate, they present significant challenges, since in many cases they must be performed in specific medical centers, and the information obtained cannot be analyzed in real time, which delays critical decision-making.

In response to these pressing needs, an innovative medical software platform has been developed designed for remote real-time monitoring of cardiac examinations. This development has the main objective of improving the accessibility, accuracy and continuity of cardiology care. The platform allows the visualization and analysis of cardiac data generated by specialized hardware, integrating advanced technologies so that both patients and health professionals can access results immediately from any geographic location.

Keywords: Cardiology, Cardiac examinations, Medical software, Real-time remote monitoring, Cardiac data analysis

En la era actual de avances tecnológicos y transformación digital, la integración de la tecnología en el ámbito médico ha generado impactos significativos en la mejora de la atención sanitaria. Uno de los campos pioneros en esta convergencia es el desarrollo de sistemas de software que permiten la visualización en tiempo real de datos cardíacos provenientes de hardware especializado. Este proyecto de desarrollo de software se centra en la creación de una plataforma que posibilita la monitorización en tiempo real de exámenes cardíacos, permitiendo tanto al paciente como al profesional de la salud acceder a los resultados de manera remota. La base de este sistema radica en la recopilación de datos cardíacos a través de un hardware específicamente diseñado para tal fin. Este dispositivo, portátil y de fácil utilización, recolecta datos vitales del corazón del paciente, estableciendo así una conexión crucial entre la atención médica y la tecnología avanzada. La capacidad de transmitir estos datos de forma remota no solo facilita el monitoreo continuo, sino que también abre la puerta a una atención médica más accesible y flexible. El componente fundamental que distingue este proyecto es la implementación de un módulo de *machine learning*. Este elemento de inteligencia artificial no solo facilita la interpretación de los datos cardíacos, sino que también ofrece un apoyo valioso al profesional de la salud en la detección temprana y el diagnóstico preciso de posibles enfermedades cardíacas. La combinación de la tecnología de punta con la capacidad predictiva del *machine learning* brinda una herramienta integral para la toma de decisiones médicas informadas y eficientes.

El desarrollo de este proyecto surge de la necesidad imperante de mejorar y optimizar la monitorización cardíaca, aprovechando las capacidades que ofrece la tecnología actual. A continuación, se exponen las principales razones y motivaciones que respaldan la realización de este trabajo:

1. Accesibilidad y Continuidad de la Atención Médica:

En la actualidad, el acceso constante a la atención médica es esencial. Este proyecto busca abordar la demanda creciente de soluciones que permitan a los pacientes recibir atención cardíaca de manera remota, sin comprometer la calidad de la asistencia. La posibilidad de visualizar en tiempo real los exámenes cardíacos desde cualquier ubicación proporciona una continuidad en la atención médica, especialmente valiosa para pacientes con condiciones cardíacas crónicas o en situaciones de emergencia. Incluso en tiempos de pandemia, como la sucedida en el 2020, en la cual la telemedicina fue útil para poder atender pacientes con problemas cardíacos (Jaramillo et al., 2022).

2. Diagnóstico Temprano y Preciso:

La detección temprana de enfermedades cardíacas es crucial para mejorar las tasas de supervivencia y calidad de vida de los pacientes. La implementación de un módulo de machine learning en el sistema ofrece la capacidad de analizar grandes conjuntos de datos de manera eficiente, identificando patrones y signos de posibles anomalías cardíacas. Esto no solo facilita el diagnóstico rápido, sino que también permite a los profesionales de la salud intervenir de manera oportuna, brindando tratamientos preventivos y personalizados.

La insuficiencia cardíaca es una de las enfermedades más comunes del corazón, que para personas con poco acceso al sistema de salud, la telemedicina es una alternativa muy viable para estas personas ya que les ahorra tiempo de traslado y los gastos son menores (Roldán et al., 2019).

3. Flexibilidad y Comodidad para Pacientes y Profesionales de la Salud:

La posibilidad de realizar exámenes cardíacos de forma remota no solo alivia la carga de los centros de atención médica, sino que también ofrece una mayor comodidad a los pacientes. La flexibilidad en la programación de las pruebas cardíacas y la capacidad de revisar resultados desde cualquier ubicación benefician tanto a los pacientes como a los profesionales de la salud, permitiendo una atención más eficiente y personalizada.

4. Aprovechamiento de la Tecnología Avanzada:

En un mundo impulsado por la innovación tecnológica, la combinación de hardware especializado y machine learning representa una oportunidad única para mejorar la atención cardíaca. Este proyecto busca aprovechar al máximo estas tecnologías avanzadas para proporcionar soluciones prácticas y efectivas en el ámbito médico, contribuyendo así al progreso de la atención cardíaca moderna.

5. ¿Porque los objetivos hacen que la propuesta sea viable para resolver el problema?

- a) La implementación de una aplicación para el uso remoto de un Sistema de Procesamiento (SP) garantiza la accesibilidad y la continuidad del monitoreo cardíaco, abordando la necesidad de atención médica a distancia. Esta característica es esencial para situaciones en las que la presencia física en un centro de atención médica puede ser limitada o inconveniente.
- b) La integración de inteligencia artificial para el apoyo en el diagnóstico de enfermedades cardíacas potencia la capacidad de la plataforma para analizar datos de manera eficiente y proporcionar a los profesionales de la salud herramientas de apoyo valiosas. La IA contribuye a la detección temprana y precisa de posibles patologías, mejorando la calidad de la atención médica y permitiendo intervenciones oportunas.
- c) Una interfaz visual intuitiva no solo facilita la interacción entre el paciente y la aplicación, sino que también optimiza la experiencia del profesional de la salud al revisar los resultados. La usabilidad adecuada es crucial para garantizar la adopción efectiva del sistema por parte de los usuarios y maximizar su eficacia en situaciones clínicas.
- d) La elección de utilizar servidores en la nube para alojar el API y las bases de datos garantiza una infraestructura tecnológica sólida y escalable. Esto no solo optimiza el rendimiento y la seguridad del sistema, sino que también permite adaptarse eficientemente a cambios en la carga de trabajo y escalar recursos según sea necesario. Además, el uso de bases de datos tanto relacionales como no relacionales ofrece flexibilidad para manejar diversos tipos de datos de manera eficaz.

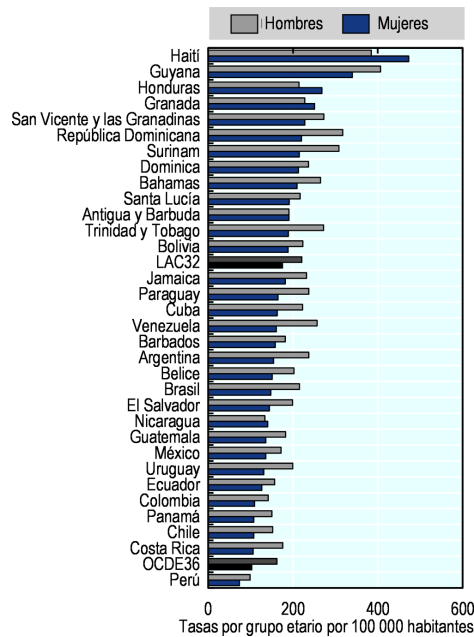
6. Estadísticas y gráficas

Durante el año 2021, aproximadamente un millón 100 mil personas perdieron la vida debido a diversas causas, de las cuales 220 mil fallecieron a causa de enfermedades cardíacas. Dentro de este grupo, el 78 % correspondió a casos de infarto del corazón o cardiopatía isquémica. Además, se registraron alrededor de 30 mil defunciones relacionadas con la hipertensión arterial. Es importante destacar que las enfermedades coronarias isquémicas afectan tanto a hombres como a mujeres por igual. (Secretaría de Salud de México, 2022.)

Anualmente, un mayor número de personas fallece debido a enfermedades cardiovasculares (ECV) en comparación con cualquier otra causa de mortalidad. Más del 75 % de los decesos relacionados con enfermedades cardíacas y accidentes cerebrovasculares se registran en naciones con ingresos medios y bajos. El riesgo de desarrollar una enfermedad cardiovascular (ECV) se incrementa como resultado de una alimentación poco saludable, caracterizada por un bajo consumo de frutas y verduras, así como por un elevado consumo de sal, azúcares y grasas. La adopción de hábitos alimenticios no saludables contribuye al desarrollo de la obesidad y el sobrepeso, ambos reconocidos como factores de riesgo significativos para las enfermedades cardiovasculares (“Enfermedades cardiovasculares”, s.f.).

Figura 1

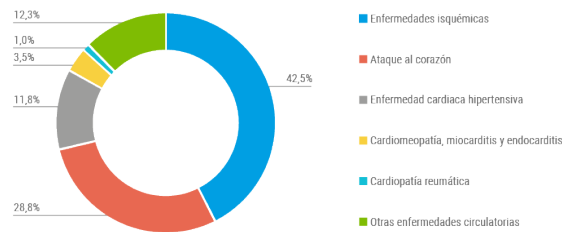
Enfermedades cardiovasculares, tasas de mortalidad estimadas, 2000 y 2017



Nota Adaptado de Tasa de mortalidad por enf. cardiovasculares, 2000 - 2017, Observatorio Global de Salud, OMS, 2019.

Figura 2

Causas de mortalidad cardiovascular, Latinoamérica, 2012



Nota Adaptado de Causas de mortalidad cardiovascular, Latinoamérica, 2012, Lanasa, 2014.

A. Objetivo general

Desarrollar una aplicación para el uso remoto de un SP para el monitoreo y visualización de un examen cardiaco en tiempo real.

B. Objetivos específicos

- Desarrollar una IA que apoye al diagnóstico de una enfermedad cardiaca e incorporar Explainable AI para la correcta interpretación de los datos.
- Desarrollar una interfaz visual intuitiva y agradable para el usuario.
- Hacer uso de servidores en la nube para el alojamiento del API y las bases de datos tanto relacionales como no relacionales.

A. Software médico y sus tipos

Un software médico es una solución integral que aborda las necesidades específicas de la gestión en un entorno médico. Desde la programación de citas y la gestión de expedientes médicos hasta el seguimiento de inventario y la facturación, estas plataformas están diseñadas para agilizar y mejorar todas las facetas de la operación de un centro de salud, como también la digitalización de los exámenes o el historial de los mismos (Dugarte et al., 2007).

Al automatizar tareas administrativas como el registro de pacientes, la programación de citas y el procesamiento de pagos, el software médico libera tiempo para que el personal médico se enfoque en brindar una atención de calidad a los pacientes. Además, al facilitar la comunicación entre médicos y pacientes a través de mensajes seguros y recordatorios automáticos, se promueve una mejor atención al paciente y se reduce el riesgo de errores en la gestión de la información médica.

Estos sistemas deben tener una buena gestión de riesgos y calidad en el contexto del software médico, cobertura, aprendizaje automático, estadística, ciberseguridad, ingeniería de software y atención médica (Papademetris et al., 2022).

Además, estos sistemas están diseñados para cumplir con los estándares de privacidad y seguridad de datos, asegurando que la información confidencial del paciente esté protegida de manera adecuada (“¿Qué es un software médico?”, s.f.).

Tipos de software médico

- Software de registro de salud electrónico (EHR):

El sistema de registro médico electrónico puede ser visto como un tipo de archivo digital que registra todos los procedimientos médicos realizados y el progreso de recuperación de los pacientes en un departamento de hospital específico.

- Software de registro de salud electrónico (Software de diagnóstico médico):
 Facilita la transferencia automatizada de datos en tiempo real entre distintas áreas de la medicina. Esta visión posibilita abordar el conjunto de síntomas médicos de manera integral, lo que garantiza un diagnóstico rápido y preciso.
- Software para imágenes y visualización:
 Aplicaciones diseñadas para la planificación y ejecución de cirugías, incluyendo la opción de telecirugía, son útiles especialmente para procedimientos laparoscópicos, ya que permiten visualizar y evaluar el cuerpo en tiempo real.
- Software de base de datos médicos:
 La principal distinción entre un EHR y una base de datos médica es que los EHR clasifican los casos según el diagnóstico de la enfermedad, lo que significa que hay una base de datos separada para cada enfermedad.
- Software de prescripción electrónica:
 Clínicas y hospitales emplean esta técnica para mejorar la distribución de medicamentos y reducir los errores en la dosificación, lo que genera mayor confianza y seguridad en los pacientes.
- Software de programación de citas:
 Un sistema digital para la programación de citas en clínicas y hospitales aumenta la productividad del personal al disminuir la cantidad de trabajo asociado con la programación telefónica de citas.
- Software de gestión de equipos médicos:
 El software permite seguir y supervisar el funcionamiento y cuidado de dispositivos médicos, como marcapasos o prótesis. Sus características incluyen la organización de los equipos por categorías (Morales, 2023).

B. Inteligencia artificial responsable [*Responsible AI*]

Los software de inteligencia artificial (IA) ya forman parte de nuestras vidas, pero para confiar plenamente en su funcionamiento y resultados, los sistemas de aprendizaje automático no pueden ser percibidos como injustos, inexactos o dañinos. El Responsable IA Implica desarrollar y usar sistemas de inteligencia artificial de forma ética y justa, asegurando que sean implementados de manera adecuada (Morales, 2023).

El machine learning ya se utiliza en sector médico, educativo, financiero, legal y de marketing, por poner algunos ejemplos. Encontramos herramientas machine learning enfocadas en predicciones para estos sectores empresariales (como los software de IA financiera), el software RPA que automatiza procesos digitales, el software industrial inteligente, avanzados ERP con aprendizaje automático y sistemas de marketing automatizado que predicen tendencias (García, 2023).

El Responsable IA tiene 2 características importantes:

- Su objetivo es minimizar los riesgos asociados al uso de la inteligencia artificial.
- Se basa en 4 principios: transparencia, equidad, privacidad y responsabilidad.

C. Inteligencia artificial explicable [*Explainable AI*]

Explicar la IA Explicable es presentar un conjunto de herramientas y marcos que permiten comprender y explicar las predicciones hechas por los modelos de aprendizaje automático. Estas herramientas están integradas en varios productos y servicios de Google y ayudan a depurar y mejorar el rendimiento de los modelos, así como a facilitar la comprensión de su comportamiento por parte de los usuarios. Además, permiten generar atribuciones de características para analizar visualmente el comportamiento de los modelos en diferentes plataformas como AutoML Tables, BigQuery ML y Vertex AI con la herramienta de hipótesis (“Explainable AI”, s.f.).

D. Inteligencia artificial

La inteligencia artificial (IA) ha pasado a ser un término amplio que describe aplicaciones que llevan a cabo tareas complejas que anteriormente requerían intervención humana, como la comunicación en línea con clientes o jugar ajedrez. A menudo se utiliza junto con subcampos como el aprendizaje automático y el aprendizaje profundo. Sin embargo, hay diferencias entre ellos. Por ejemplo, el aprendizaje automático se enfoca en crear sistemas que aprenden o mejoran su rendimiento basándose en datos. Es importante notar que, aunque todo aprendizaje automático es IA, no toda IA es aprendizaje automático. Para sacar el máximo provecho de la IA, muchas empresas están invirtiendo en equipos de ciencia de datos. La ciencia de datos combina estadísticas, informática y conocimiento empresarial para extraer valor de diversas fuentes de datos (“¿Qué es la inteligencia artificial (IA)?”, s.f.).

1. Tipos de inteligencia artificial

En el mundo de la inteligencia artificial hay muchos tipos de esta, pero las más comunes y utilizadas son:

- Redes neuronales artificiales [*Artificial neural networks*]
Las redes neuronales artificiales son algoritmos de inteligencia computacional que abordan algunas limitaciones de los sistemas expertos, como su incapacidad para resolver casos nuevos o aquellos que están fuera de sus reglas de programación. Funcionan de manera similar al cerebro humano, con unidades de información que colaboran entre sí para procesar datos.
- Aprendizaje Profundo [*Deep learning*]
El aprendizaje profundo, también llamado deep learning, se refiere a un tipo de algoritmo de redes neuronales artificiales que son altamente complejos y están

diseñados para realizar tareas computacionalmente intensivas. Estos algoritmos son conocidos por su profundidad, lo que significa que tienen múltiples capas que les permiten aprender y procesar información de manera más sofisticada. El deep learning se utiliza principalmente en conjunción con la minería de datos, una disciplina estadística que busca descubrir patrones en grandes conjuntos de datos. Esto implica el uso de códigos complejos y la alimentación de extensas bases de datos para entrenar los modelos de aprendizaje profundo.

- **Sistemas expertos**

Los sistemas expertos representan una forma elemental de inteligencia artificial que está presente en numerosos aspectos de nuestra vida cotidiana, aunque a menudo pasan desapercibidos. Estas tecnologías están diseñadas para llevar a cabo tareas específicas de manera similar al razonamiento humano, utilizando un nivel de conocimiento especializado en el área correspondiente. En resumen, son herramientas que imitan el comportamiento y la toma de decisiones de un experto humano en un campo particular.

- **Robótica**

La robótica se ha convertido en una disciplina autónoma dentro del ámbito de la inteligencia artificial, y ha experimentado un crecimiento significativo gracias al uso de recursos digitales. Esto ha propiciado que muchos avances en robótica se basen en el empleo de redes neuronales artificiales. En la actualidad, se han desarrollado robots con capacidades avanzadas, como locomoción, ejecución de tareas complejas e incluso habilidades para jugar al ajedrez, todo ello gracias a sistemas de visión por computadora, algoritmos de aprendizaje y código que les permite tomar decisiones. La inteligencia artificial es fundamental en estos avances.

2. Machine learning

Machine learning es una disciplina dentro del ámbito de la inteligencia artificial y la informática que se enfoca en emplear datos y algoritmos para simular el proceso de aprendizaje humano, con el objetivo de mejorar su precisión de manera progresiva.

En el aprendizaje automático, los algoritmos utilizan datos de entrada para entrenar modelos matemáticos que pueden realizar tareas específicas, como clasificación, regresión, clustering o detección de anomalías. Estos modelos aprenden de los datos pasados y pueden aplicarse a nuevas situaciones para hacer predicciones o tomar decisiones basadas en la información disponible, en el caso de la cardiología, poder detectar enfermedades en las primeras etapas de las mismas (Mappangara et al., 2020).

El proceso de aprendizaje automático implica varias etapas, que incluyen la recopilación y preparación de datos, la selección y entrenamiento de modelos, la evaluación del rendimiento y la optimización de los resultados. Además, existen diferentes enfoques y técnicas de aprendizaje automático, como el aprendizaje supervisado, no supervisado y por refuerzo, cada uno de los cuales se adapta a distintos tipos de problemas y datos (“¿Qué es machine learning?”, s.f.).

3. ¿Cómo Funciona?

El Machine Learning descompone el proceso de aprendizaje de un algoritmo en tres componentes principales:

- Un proceso de toma de decisiones: típicamente, los algoritmos de Machine Learning se utilizan para realizar predicciones o clasificaciones. Basándose en los datos de entrada, ya sea etiquetados o no, el algoritmo genera una estimación sobre un patrón presente en los datos.
- Una función de error: esta función evalúa qué tan precisa es la predicción del modelo. Si se dispone de ejemplos conocidos, la función de error puede comparar la predicción del modelo con estos ejemplos para evaluar su precisión.
- Un proceso de optimización del modelo: si el modelo puede ajustarse mejor para ajustarse a los datos de entrenamiento, las ponderaciones se ajustan para reducir la discrepancia entre los ejemplos conocidos y las estimaciones del modelo. Este proceso se repite, actualizando las ponderaciones de forma iterativa hasta alcanzar un umbral de precisión deseado (“¿Qué es machine learning?”, s.f.).

E. Servicios en la nube

Los servicios de nube son recursos informáticos que se proporcionan a través de Internet, incluyendo infraestructuras, plataformas y software. Estos servicios permiten a los usuarios enviar y recibir datos desde cualquier dispositivo conectado a Internet, facilitando el desarrollo y la implementación de aplicaciones en la nube. Para acceder a ellos, solo se requiere una conexión a Internet y un dispositivo compatible (“¿Qué son los servicios de nube?”, s.f.).

Al emplear servicios de computación en la nube, los usuarios pueden acceder a recursos en línea desde diferentes dispositivos como estaciones de trabajo, laptops, tablets y smartphones. Estos dispositivos están configurados para garantizar la seguridad de los datos y activos alojados en la nube. Con un modelo de pago por consumo, los servicios de nube ofrecen una forma rentable de manejar los picos de demanda de manera más eficiente que los servicios de computación internos (“¿Qué son los servicios de nube?”, s.f.).

F. Websockets

En API Gateway, es posible crear una API de WebSocket que actúa como una interfaz de usuario con estado para un servicio de AWS, como Lambda o DynamoDB, o para un punto de enlace HTTP. Esta API de WebSocket se encarga de invocar al backend en función del contenido de los mensajes que recibe de las aplicaciones cliente.

A diferencia de una API de REST, que simplemente recibe y responde a solicitudes, una API de WebSocket permite una comunicación bidireccional entre las aplicaciones cliente y el

backend. Esto significa que el backend puede enviar mensajes de devolución de llamada a los clientes conectados (“Acerca de las API de WebSocket en API Gateway”, s.f.).

Los websockets permiten establecer una conexión bidireccional entre el navegador del usuario y un servidor, lo que facilita una comunicación interactiva en tiempo real. Con esta tecnología, es posible enviar mensajes al servidor y recibir respuestas de forma controlada por eventos, sin necesidad de realizar consultas repetidas al servidor para obtener una respuesta. Esto proporciona una experiencia más dinámica y eficiente para los usuarios, ya que pueden recibir actualizaciones instantáneas sin necesidad de recargar la página constantemente (“Acerca de las API de WebSocket en API Gateway”, s.f.).

G. Ley de responsabilidad y portabilidad del seguro médico [*Health Insurance Portability and Accountability Act (HIPAA)*]

Se trata de una ley federal que estableció normas nacionales para salvaguardar la confidencialidad de la información médica de los pacientes, prohibiendo su divulgación sin el consentimiento o conocimiento del paciente. El Departamento de Salud y Servicios Humanos de EE. UU. (HHS) emitió la Regla de Privacidad de la Ley HIPAA para implementar dichos requisitos. La Regla de Seguridad de la Ley HIPAA, por su parte, protege un subconjunto específico de la información contemplada en la Regla de Privacidad.

1. Reglas del HIPAA

Las normativas de la Regla de Privacidad se centran en el manejo y la revelación de la información de salud de las personas, conocida como Información de Salud Protegida (PHI, por sus siglas en inglés), por parte de entidades sujetas a dicha regla, denominadas "entidades cubiertas". La Regla de Privacidad también establece estándares para garantizar los derechos de las personas en comprender y controlar el uso de su información médica. Un objetivo fundamental de esta regla es asegurar la adecuada protección de la información de salud de las personas, al mismo tiempo que facilita el flujo necesario de esta información para brindar y promover atención médica de alta calidad, así como salvaguardar la salud y el bienestar del público. La Regla de Privacidad permite usos significativos de la información, manteniendo la privacidad de aquellos que buscan atención y tratamiento.

El Reglamento de Privacidad de HIPAA protege la información de salud protegida (PHI), mientras que la Regla de Seguridad protege un subconjunto de la información cubierta por la Regla de Privacidad. Este subconjunto es toda la información de salud identificable individualmente que una entidad cubierta crea, recibe, mantiene o transmite en formato electrónico. Esta información se llama información de salud protegida electrónicamente, o e-PHI. La Regla de Seguridad no se aplica a la PHI transmitida oralmente o por escrito (“Health Insurance Portability and Accountability Act of 1996 (HIPAA)”, s.f.).

Para cumplir con la Regla de Seguridad de HIPAA, todas las entidades cubiertas deben:

- Asegurar la confidencialidad, integridad y disponibilidad de toda la e-PHI
- Detectar y protegerse contra amenazas anticipadas a la seguridad de la información
- Protegerse contra usos o divulgaciones anticipadas improcedentes que no están permitidas por la regla
- Certificar el cumplimiento por parte de su personal
- Las entidades cubiertas deben confiar en la ética profesional y el mejor juicio al considerar solicitudes para estos usos y divulgaciones permitidos. La Oficina de Derechos Civiles de HHS hace cumplir las reglas de HIPAA y todas las quejas deben ser reportadas a esa oficina. Las violaciones de HIPAA pueden resultar en sanciones monetarias civiles o penales.

El proyecto se dividirá en varios módulos; la aplicación web que conecte con el hardware de detección cardiaca, un modelo de inteligencia artificial para el apoyo de diagnósticos y los servicios en la nube con un API. Los servicios en la nube deben ser capaces de registrar la información necesaria para poder llevar un control sobre los pacientes y los exámenes que se realizan. Este software fue puesto a prueba por especialistas en el campo de la cardiología, dichas pruebas tuvieron 6 etapas

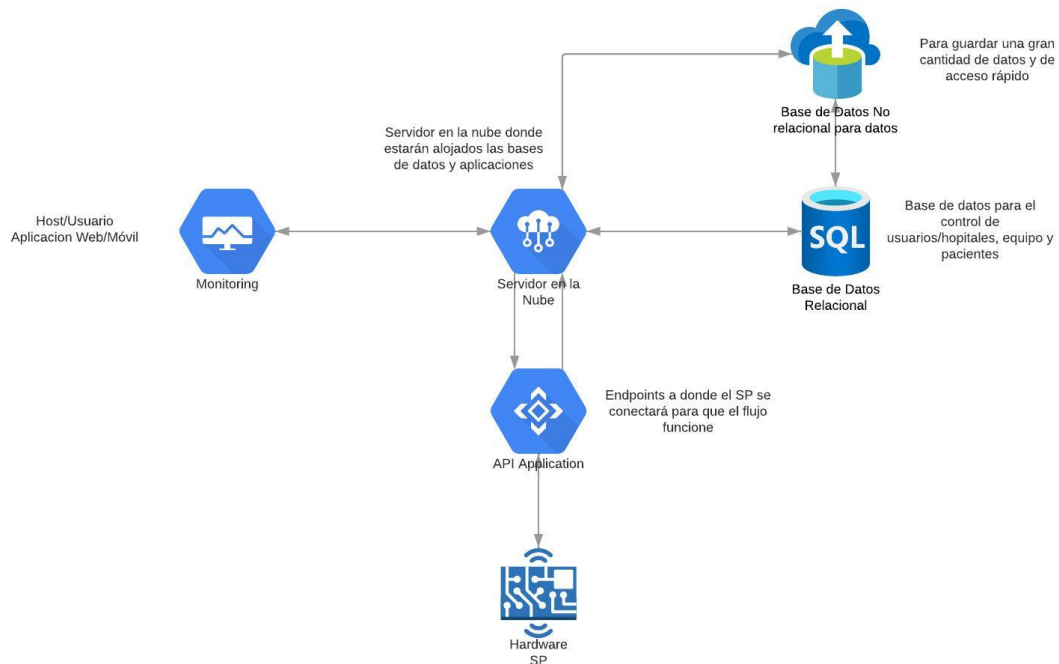
1. Investigación sobre la arquitectura del sistema
2. Reclutamiento de colaboradores
3. Definición de escenarios de prueba
4. Pruebas piloto y recolección de datos
5. Recopilación de retroalimentación
6. Análisis de resultados

A. Investigación sobre la arquitectura del sistema

1. Desarrollo del API

Para el desarrollo del API, se utilizarán tecnologías de vanguardia como lo es Node JS, ya que es una tecnología fácil de utilizar y bastante versátil para el proyecto que se quiere implementar. Aparte de esto, el API implementará las convenciones más acordes para el desarrollo de un API funcional. Este API contará con seguridad en todos sus endpoint como también estará alojada en un servidor de AWS.

Figura 3
Proceso Completo del Sistema



Nota Elaboración propia.

2. Desarrollo de la aplicación visual

Para la aplicación visual, se utilizará el framework de Javascript llamado React, ya que es un framework de mucha utilidad por su estructura basada en componentes. Esta aplicación igual estará alojada en un servidor de AWS y también se aplicarán las medidas de ciberseguridad necesarias para que sea una aplicación segura.

3. Desarrollo de Base de datos

Para la base de datos, se estarán utilizando una base de datos relacional y otra no relacional. La base de datos relacional se utilizará para el control de usuarios y de hospitales. La base de datos no relacional se utilizará para guardar los datos de los exámenes y para el control de conexión del servidor con el hardware. Además, se estará tomando en cuenta el Health Insurance Portability and Accountability Act para poder guardar de manera correcta la información de los pacientes.

4. Escalabilidad

Para asegurar la escalabilidad del proyecto, se adoptó un enfoque modular en el desarrollo de los algoritmos de detección de anomalías. Los modelos de IA se diseñaron para ser altamente adaptables y capaces de manejar grandes volúmenes de datos médicos. Se utilizaron técnicas

de aprendizaje automático escalables, como el aprendizaje profundo, que pueden procesar eficientemente grandes conjuntos de datos y adaptarse a diferentes dominios médicos y tipos de datos.

Además, el proyecto se implementó en una infraestructura de nube escalable, lo que permitió escalar vertical y horizontalmente según fuera necesario para manejar aumentos en la carga de trabajo y el volumen de datos. Esto permitió realizar análisis de datos en tiempo real y proporcionar resultados rápidos y precisos, incluso en entornos clínicos de alta demanda.

Para probar la escalabilidad, se realizaron pruebas con Postman haciendo uso de funciones internas del software como solicitudes al servidor de manera simultánea y pruebas con usuarios virtuales.

5. Seguridad

La seguridad de los datos médicos fue una prioridad en el proyecto. Se implementaron medidas de seguridad robustas en todas las etapas del proceso, desde la recopilación hasta el análisis y el almacenamiento. Se utilizaron técnicas de encriptación para proteger los datos durante la transmisión y el almacenamiento, y se implementaron controles de acceso y autenticación para garantizar que solo personal autorizado tuviera acceso a la información confidencial del paciente.

Además, se adhirió algunas regulaciones y estándares de privacidad de datos, como la Ley de Portabilidad y Responsabilidad del Seguro Médico (HIPAA) en los Estados Unidos. Esto garantizó que el proyecto cumpliera con las normativas legales y éticas aplicables en el manejo de datos médicos sensibles y protegiera la privacidad y confidencialidad de los pacientes en todo momento.

B. Desarrollo del Modelo de Machine Learning

Se estará realizando un modelo de machine learning para poder realizar el apoyo en el diagnóstico de enfermedades del corazón. Se tendrán bases de datos de enfermedades del corazón para que la IA pueda entrenarse y dar un apoyo en el diagnóstico de la persona que está siendo examinada. También se usará las herramientas de Explainable IA y Responsible IA para poder interpretar y presentar de manera correcta los resultados.

Modelo a Utilizar

El modelo utilizado para la detección de anomalías en el ECG es el One-Class SVM (Support Vector Machine de una sola clase). Este modelo es una técnica de aprendizaje no supervisado que es adecuada para detectar anomalías en conjuntos de datos no etiquetados. Utiliza un enfoque de aprendizaje basado en la idea de encontrar la región en la que se encuentran la mayoría de los datos y considerar como anomalías aquellos puntos que están significativamente lejos de esta región.

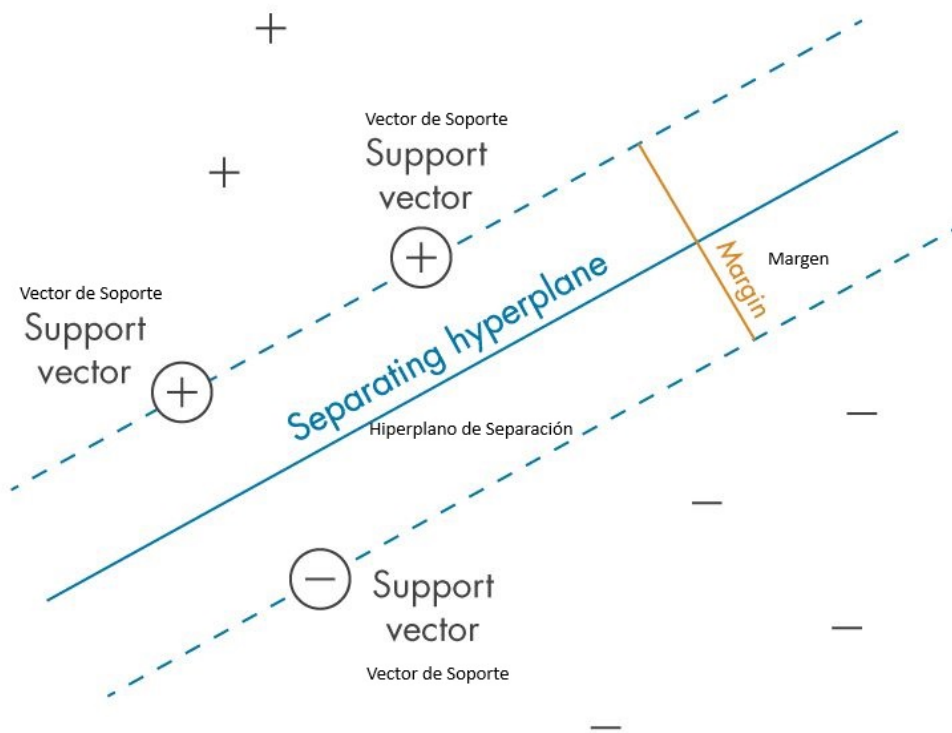
¿Porqué se escogió este modelo?

La Máquina de Vectores de Soporte de una Sola Clase (One-Class SVM) es un algoritmo de aprendizaje automático utilizado para la detección de anomalías en conjuntos de datos no etiquetados. A diferencia de las SVM tradicionales que se utilizan para clasificar datos en dos o más clases, One-Class SVM está diseñada específicamente para identificar patrones anómalos en datos unimodales, es decir, datos que pertenecen a una sola clase.

El funcionamiento de One-Class SVM se basa en encontrar la región de mayor densidad en el espacio de características que contiene la mayoría de los datos normales. Para ello, el algoritmo busca un hiperplano en el espacio de características que separa los datos normales de los datos anómalos. Este hiperplano se coloca de manera que maximiza la distancia entre el hiperplano y los puntos de datos más cercanos, conocidos como vectores de soporte. on Computer Science, 2024

El objetivo del algoritmo SVM es determinar un hiperplano que separe de manera óptima dos clases diferentes de puntos de datos. "De manera óptima" se refiere al hiperplano que tiene el margen más amplio entre las dos clases, representado por los signos más y menos en la figura siguiente. El margen se define como la mayor anchura de la región paralela al hiperplano que no contiene puntos de datos. El algoritmo puede encontrar este hiperplano en problemas que permiten una separación lineal; sin embargo, en la mayoría de los problemas prácticos, el algoritmo maximiza el margen de forma flexible permitiendo un pequeño número de clasificaciones incorrectas.

Figura 4
Criterio que los SVM intentan optimizar



Nota Definición del “margen” entre clases: el criterio que los SVM intentan optimizar, Tomado de Mathworks, sf.

Específicamente, las máquinas de vectores de soporte son parte de una clase de algoritmos de aprendizaje automático conocidos como métodos kernel. Estos métodos utilizan una función de kernel para transformar las características de los datos. Las funciones de kernel proyectan los datos a un espacio de mayor dimensión, con la intención de que sea más sencillo separar las clases después de esta transformación. Esto puede convertir límites de decisión complejos y no lineales en límites lineales en el nuevo espacio de características. En este proceso, no es necesario transformar los datos explícitamente, lo cual reduciría la carga computacional. Este método es conocido como el truco de kernel.

Otros Modelos considerados

Se tomaron en cuenta otros modelos para realización de la detección de anomalías, tomando en cuenta el como funcionan, si son con datos no etiquetados y sean eficientes con los datos adquiridos.

Bosque de aislamiento [*Isolation Forest*]: Este es un algoritmo de detección de anomalías basado en árboles de decisión. Utiliza la idea de que las anomalías son excepciones y, por lo tanto, tienen menos conexiones con otras instancias en el conjunto de datos. Isolation Forest funciona dividiendo aleatoriamente los datos en subconjuntos más pequeños y más pequeños, lo que permite detectar anomalías más fácilmente.

Factor de valores atípicos locales [*Local Outlier Factor (LOF)*]: LOF es un algoritmo que calcula la desviación local de densidad de un punto de datos con respecto a sus vecinos. Funciona comparando la densidad local de un punto con la densidad local de sus vecinos. Si un punto tiene una densidad local significativamente menor que la de sus vecinos, se considera una anomalía.

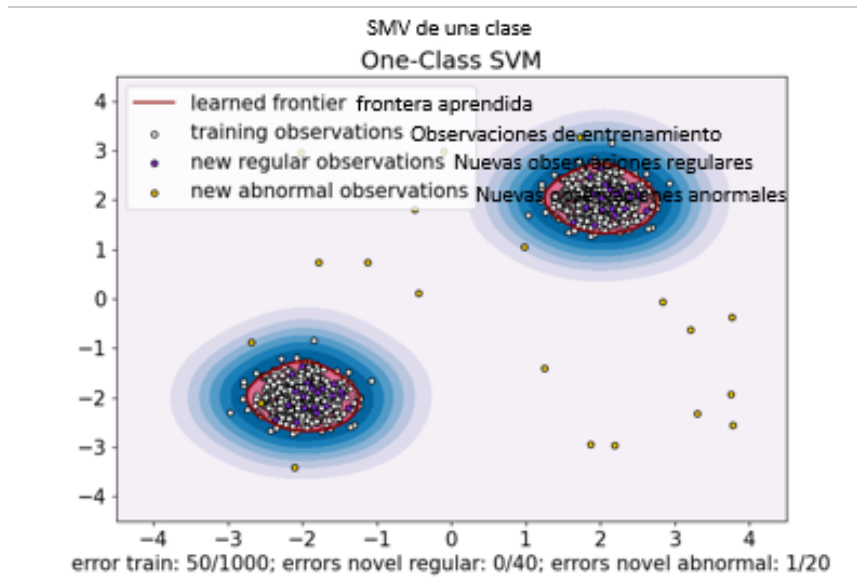
Codificadores automáticos [*Autoencoders*]: Los autoencoders son una forma de red neuronal que se utiliza para aprender representaciones eficientes de los datos de entrada. En el contexto de la detección de anomalías, puedes entrenar un autoencoder en datos normales y luego utilizarlo para reconstruir los datos de entrada. Si la reconstrucción es pobre para ciertas instancias, es probable que esas instancias sean anomalías.

Detección de anomalías basada en densidad [*Density-based anomaly detection*]: Este enfoque se basa en la idea de que las anomalías son instancias que tienen una densidad de datos mucho más baja que las instancias normales. Algunos métodos de detección de anomalías basados en densidad incluyen el algoritmo de estimación de densidad kernel (KDE) y el algoritmo de vecinos más cercanos basado en densidad (DBSCAN).

Configuración del modelo

Para este estudio, hemos configurado el One-Class SVM con un hiperparámetro ν de 0.1. El parámetro ν controla la cantidad de puntos atípicos aceptados como anomalías. Un valor más alto de ν permite más puntos atípicos, mientras que un valor más bajo es más restrictivo en la detección de anomalías.

Figura 5
Gráfica de funcionamiento de One-Class SVM.



Nota Adaptado de One-Class SVM, Liu, Shuchao and Xu, 2022.

Datos a utilizar

Los datos a utilizar son datos realistas del hardware del cual se está elaborando el software para su uso conjunto. Actualmente, el hardware cuenta con solamente la lectura de 1 electrodo que representa 2 derivaciones. La cantidad de información que detecta el hardware son de 100 datos por segundo. El dataset contiene 2800 datos de un corazón sano. Estos datos incluyen el tiempo de la señal y el valor analógico correspondiente en cada punto temporal. El conjunto de datos se preprocesó para eliminar cualquier información identificable y se dividió en un conjunto de entrenamiento y un conjunto de prueba para el entrenamiento y la evaluación del modelo, respectivamente.

Figura 6
Datos para el modelo a entrenar

Time	AnalogicValue	Userld
0.01	-0.02	1
0.02	-0.02	1
0.03	-0.02	1
0.04	-0.02	1
0.05	0	1
0.06	0.02	1
0.07	0.03	1
0.08	0.03	1
0.09	0.02	1
0.1	0.01	1
0.11	0	1
0.12	0	1
0.13	0	1
0.14	0	1
0.15	0	1
0.16	-0.01	1
0.17	0	1
0.18	0	1
0.19	0	1
0.2	0.57	1
0.21	0.57	1
0.22	0.57	1
0.23	0.24	1
0.24	-0.03	1
0.25	0.02	1
0.26	-0.11	1
0.27	-0.2	1

Nota Elaboración propia.

Modo de entrenamiento

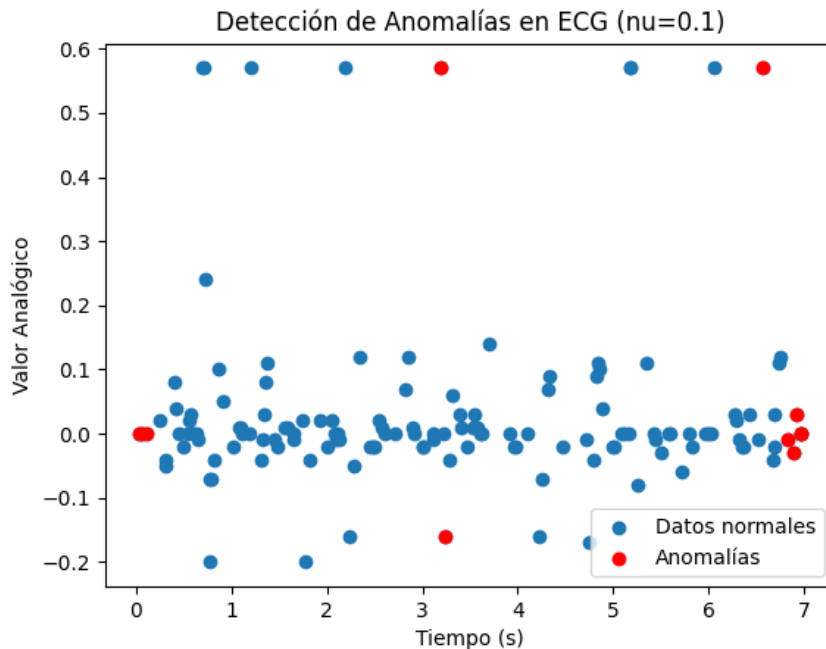
El conjunto de datos se dividió en un conjunto de entrenamiento y un conjunto de prueba utilizando una proporción de 80:20. Esto significa que el 80 % de los datos se utilizaron para entrenar el modelo, mientras que el 20 % restante se reservaron para evaluar su rendimiento. Durante el entrenamiento, el modelo One-Class SVM ajustó sus parámetros para identificar patrones normales en los datos de entrenamiento.

Cálculo de la eficiencia

Para evaluar la eficiencia del modelo en la detección de anomalías, se utilizaron dos métricas principales: el área bajo la curva ROC (AUC-ROC) y el área bajo la curva PR (AUC-PR). Estas métricas proporcionan una medida cuantitativa del rendimiento del modelo en la clasificación de los datos de prueba como normales o anómalos. Además, se realizaron análisis cualitativos de los resultados para comprender mejor el rendimiento del modelo en diferentes situaciones. Esta precisión ayuda al cardiólogo a poder detectar enfermedades cardíacas en tempranas etapas de la misma.

Figura 7

Visualización de la detección de anomalías en las señales de ECG. Los puntos rojos representan las anomalías identificadas por el modelo One-Class SVM.



Nota Elaboración propia.

C. Recolección y análisis de datos

Para llevar a cabo este proyecto de investigación y experimentación, se realizaron cinco etapas. Estas etapas comprendieron el proceso de aprendizaje, aplicación, y traslado de conocimiento a los estudiantes.

D. Reclutamiento de colaboradores

Se reclutaron dos cardiólogos altamente calificados, ambos con una amplia experiencia en cardiología clínica y un profundo conocimiento en el diagnóstico y tratamiento de enfermedades cardiovasculares.

E. Definición de escenarios de prueba

En colaboración con los cardiólogos, se definieron escenarios de prueba realistas que abarcan una variedad de casos clínicos, como la detección de arritmias o alguna anomalía en el ECG.

F. Pruebas piloto y recolección de datos

Se realizó una serie de pruebas piloto con los cardiólogos, durante las cuales utilizaron el software en entornos clínicos simulados. Ambos cardiólogos proporcionaron valiosos comentarios sobre la usabilidad y la funcionalidad del software.

G. Recopilación de retroalimentación

Se recopiló retroalimentación detallada de ambos cardiólogos sobre su experiencia utilizando el software. Los Cardiólogos destacaron los puntos mas importantes que vieron en el programa, que falló y que se puede mejorar para un mejor uso para ellos en su campo de trabajo.

H. Análisis de resultados

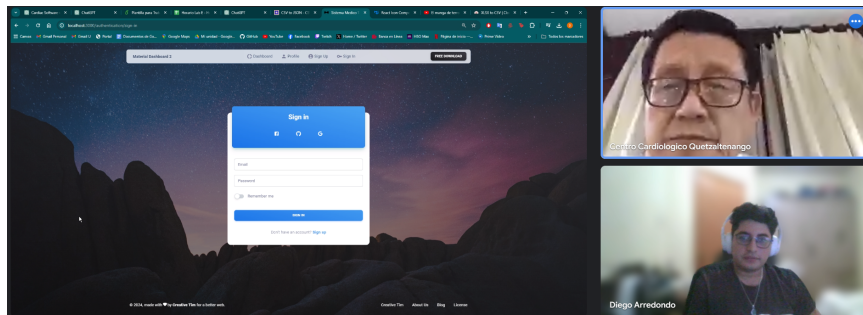
Con las respuestas de la encuesta enviada a los cardiólogos se procedió a analizar las respuestas. Estas respuestas fueron tabuladas y encontradas en un archivo de Excel que el propio Google Forms generó. Con esta hoja de trabajo, se procedió a realizar cálculos de proporcionalidad y gráficas representativas de las respuestas para identificar tendencias sobre el software utilizado.

A. Pruebas con usuarios

Se hicieron pruebas con los cardiólogos para saber su opinión profesional sobre el tema de los ECG. En estas pruebas se fue navegando por la aplicación pidiendo la opinión de los cardiólogos.

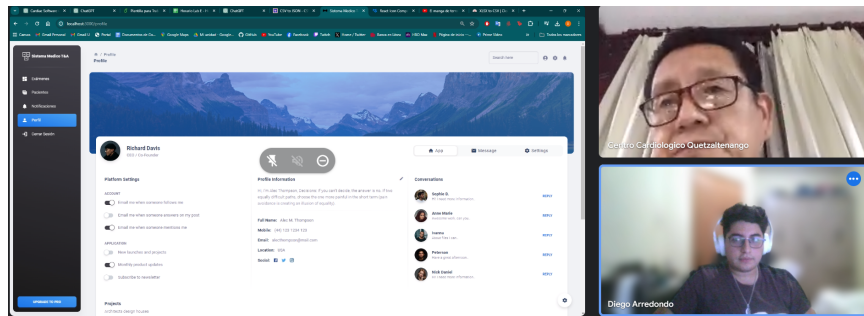
Figura 8

Pruebas con Usuario en el módulo de ingreso



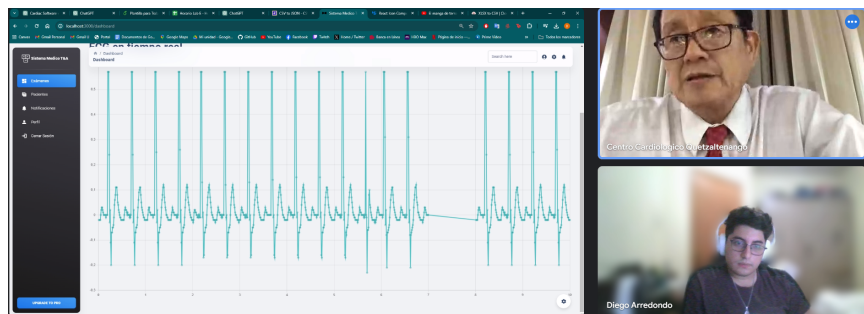
Nota Elaboración propia.

Figura 9
Pruebas con Usuario en el módulo de perfil del doctor



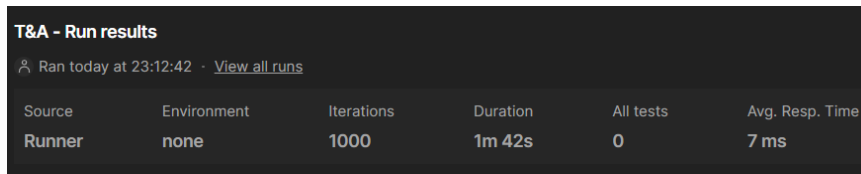
Nota Elaboración propia.

Figura 10
Pruebas con Usuario en el módulo de exámenes



Nota Elaboración propia.

Figura 11
Prueba de Estrés para el servidor Web



T&A - Run results
Ran today at 23:12:42 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1000	1m 42s	0	7 ms

Nota Elaboración propia.

Los cardiólogos hicieron observaciones acerca de la gráfica de ECG, dando recomendaciones sobre la calibración del mismo y el tiempo de captura de información a mostrar. Estas observaciones son mejoras para que ellos puedan hacer lecturas de corazón con mayor precisión.

Los cardiólogos mencionaron su conformidad con el diseño de la aplicación, explicando que lo que el software ofrece está bien y todo puesto de una manera fácil de acceder. Los usuarios también expresaron lo intuitivo de la aplicación para ver los exámenes y los pacientes a los que están a cargo.

Otra observación dada por los cardiólogos fue el tema de la competencia. Ellos comentaron que hay software parecido o productos parecidos en el mercado. Recalaron que el producto realizado puede mejorar los que ya existen en el mercado y ofrecer una mejor solución para los cardiólogos.

B. Escalabilidad y pruebas de estrés

Se hicieron pruebas de estrés en el servidor, tomando como parámetros 1000 solicitudes con un retraso de 1 milisegundo entre cada solicitud.

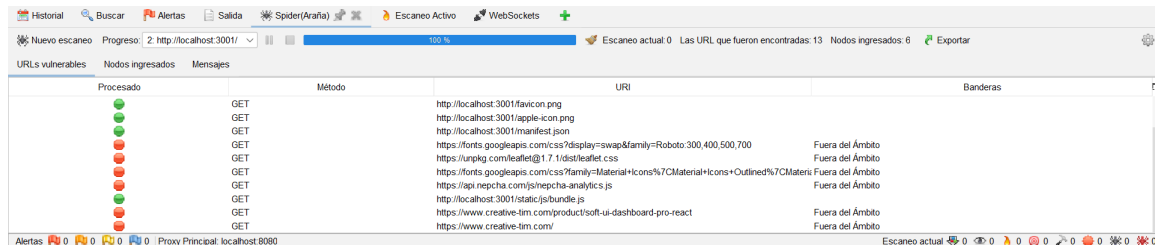
El resultado obtenido es positivo, dando un resultado de un 100 % de solicitudes respondidas por el servidor, con un promedio de 7 milisegundos por cada solicitud. Esto demuestra que la aplicación puede soportar una gran cantidad de solicitudes sin verse comprometido el rendimiento del mismo.

C. Pruebas de seguridad

Para este paso, se realizaron pruebas en el servidor local, ya que OWASP, el cual es el software que se utilizó para poder ver la seguridad de las aplicaciones, no puede conectarse a un servidor de AWS (Amazon Web Service) por políticas de OWASP y las pruebas de seguridad dentro de AWS son muy costosas o requieren mucha configuración.

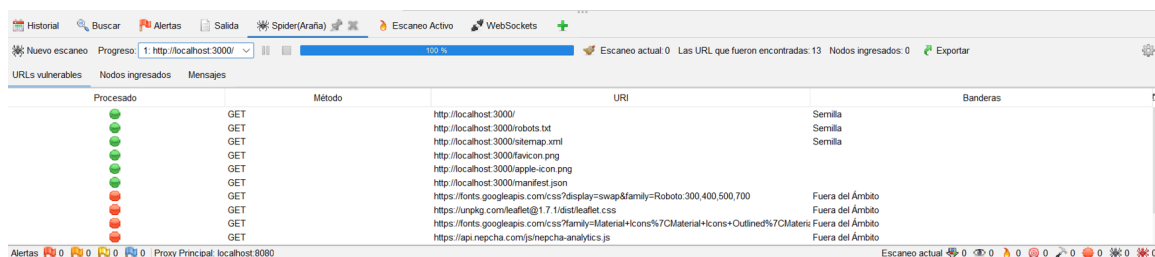
Las pruebas con las aplicaciones mostraron resultados positivos, ya que no se presentaron alertas o alguna brecha por la cual la aplicación sería comprometida por algún ciberatacante. Los resultados que se muestran en rojo, son URL's las cuales OWASP no puede acceder como las de Google Fonts u otras librerías o API's de Google.

Figura 12
Test de Seguridad para el Front-End



Nota Elaboración propia.

Figura 13
Test de Seguridad para el Back-End



Nota Elaboración propia.

D. Machine Learning

En esta sección se presenta la implementación de un modelo de clasificación para la detección de anomalías cardíacas, utilizando el algoritmo One-Class SVM. El enfoque se centra en evaluar el rendimiento del modelo en un conjunto de datos de pacientes con corazones sanos, proporcionando una base para la detección temprana de posibles irregularidades en el funcionamiento cardíaco.

1. Métricas de rendimiento con un corazón sano

El análisis del modelo One-Class SVM arrojó los siguientes resultados:

- Porcentaje de anomalías en los datos de prueba: 1.43 %
- Anomalías detectadas en los datos de prueba: 2 anomalías de 2800 datos
- AUC-ROC: 0.99
- AUC-PR: 0.25

Estas métricas proporcionan una visión detallada del rendimiento del modelo. El porcentaje de anomalías en los datos de prueba indica la proporción de datos que se identificaron

como anomalías en comparación con el total de datos de prueba. En este caso, el modelo identificó correctamente el 10.00 % de los datos como anomalías.

El AUC-ROC y el AUC-PR son métricas comunes para evaluar el rendimiento de los modelos de clasificación. El AUC-ROC mide la capacidad del modelo para distinguir entre clases positivas y negativas, mientras que el AUC-PR evalúa la precisión de las predicciones positivas. En nuestro caso, el modelo One-Class SVM obtuvo un AUC-ROC de 0.99 y un AUC-PR de 0.25, lo que indica un rendimiento decente pero con margen de mejora.

2. Interpretación de los resultados

Los resultados muestran que el modelo One-Class SVM logra detectar una parte significativa de las anomalías presentes en los datos de prueba, como se refleja en el valor de recall del 60 %. Esto sugiere que el modelo es eficaz para identificar los casos anómalos de ECG, lo que podría ser crucial en aplicaciones médicas donde la detección temprana de anomalías es fundamental para el diagnóstico y tratamiento.

Sin embargo, también se observa que el modelo tiene margen de mejora, especialmente en términos de precisión y AUC. Esto indica que el modelo podría estar identificando algunas observaciones regulares como anomalías (falsos positivos) o que podría ser más preciso en sus predicciones.

Curva ROC (Receiver Operating Characteristic)

La Curva ROC se utiliza para visualizar la relación entre la tasa de verdaderos positivos (True Positive Rate, TPR) y la tasa de falsos positivos (False Positive Rate, FPR) a diferentes umbrales de decisión. En la Figura 14 se muestra la Curva ROC obtenida para el modelo One-Class SVM, con un Área Bajo la Curva (AUC) de 0.99.

Interpretación del AUC

Un AUC de 0.99 indica un rendimiento excelente del clasificador en términos de discriminación entre las clases positiva y negativa. Este valor sugiere que el modelo es altamente efectivo para identificar correctamente la mayoría de los ejemplos positivos y negativos. La curva se acerca a la esquina superior izquierda del gráfico, lo cual es indicativo de un alto rendimiento.

Curva de precisión-recall

La Curva de Precisión-Recall se utiliza para evaluar la relación entre la precisión (Precision) y la recuperación (Recall) del clasificador. En la Figura 14 se presenta esta curva para el modelo One-Class SVM, con un Área Bajo la Curva (AUC) de 0.25.

Interpretación del AUC:

Un AUC de 0.25 es bajo y sugiere un rendimiento mejorable en términos de precisión y recuperación del clasificador. Este bajo valor indica que el modelo tiene dificultades significativas para mantener una alta precisión mientras captura la mayoría de los verdaderos positivos. La curva es una línea diagonal que va desde (1, 1) a (0, 0), lo cual evidencia una falta de equilibrio entre precisión y recuperación.

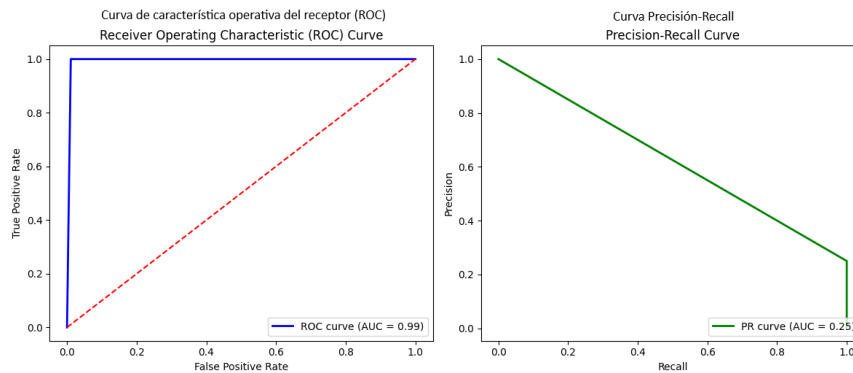
3. Visualización de resultados

Además de las métricas, las curvas ROC y PR proporcionan una visualización adicional del rendimiento del modelo. La curva ROC muestra cómo varía la tasa de verdaderos positivos frente a la tasa de falsos positivos, mientras que la curva PR muestra cómo varía la precisión frente al recall.

En la Figura 14, se puede observar que las curvas ROC se sitúa por encima de la línea de referencia aleatoria (línea punteada roja), lo que indica que el modelo tiene un rendimiento significativamente mejor que el azar en la detección de anomalías.

Figura 14

Resultados de las curvas ROC y PR para corazón sano



Nota Elaboración propia.

4. Métricas de rendimiento con un corazón no sano (más anomalías)

El análisis del modelo One-Class SVM arrojó los siguientes resultados:

- Porcentaje de anomalías en los datos de prueba: 7.14 %
- Anomalías detectadas en los datos de prueba: 200 anomalías de 2800 datos
- AUC-ROC: 0.97
- AUC-PR: 0.05

Estos resultados indican un mayor porcentaje de anomalías en los datos de prueba en comparación con el escenario anterior. Se observa un aumento en el número de anomalías detectadas y en el rendimiento del modelo en términos de recall, lo que sugiere una capacidad mejorada para identificar las anomalías presentes en los datos.

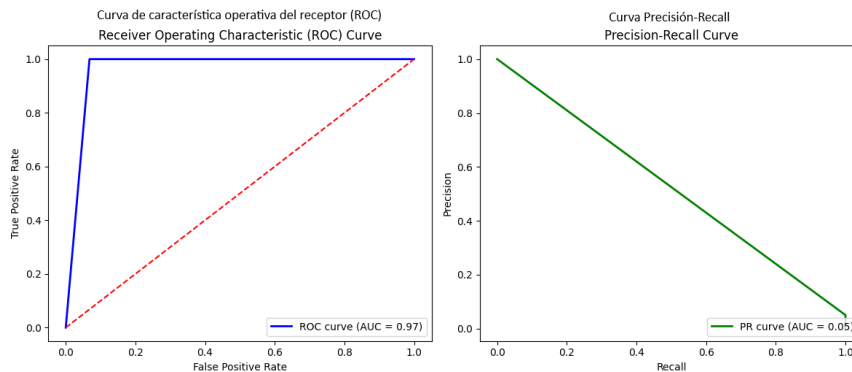
El AUC-ROC sigue siendo alto, lo que indica una buena capacidad del modelo para distinguir entre clases positivas y negativas. Sin embargo, el AUC-PR en comparación con el escenario anterior, muestra un valor menor, lo que sugiere que el modelo todavía tiene margen de mejora en términos de precisión.

5. Análisis de los resultados

El aumento en el porcentaje de anomalías y en el rendimiento del modelo en términos de recall indica una mejor capacidad del modelo para identificar casos anómalos en los datos de electrocardiogramas. La discrepancia entre el alto AUC de la Curva ROC y el bajo AUC de la Curva de Precisión-Recall es notable. Aunque el modelo muestra una capacidad de discriminación elevada, reflejada en la Curva ROC, su efectividad en términos de precisión y recuperación es baja, según lo mostrado por la Curva de Precisión-Recall.

Figura 15

Resultados de las curvas ROC y PR



Nota Elaboración propia.

6. Responsabilidad de la inteligencia artificial y aplicaciones médicas en la detección de anomalías

Se presentaron los resultados de la investigación sobre la aplicación de la Inteligencia Artificial (IA) en el apoyo en la detección de anomalías en datos médicos, específicamente en el análisis de electrocardiogramas (ECG). Se tomó en cuenta la responsabilidad de la IA en el desarrollo y aplicación de estas tecnologías, así como su factibilidad y aplicaciones prácticas en software médico.

- Responsabilidad de la IA: Durante el desarrollo y aplicación de algoritmos de IA para la detección de anomalías en datos médicos, se consideró la responsabilidad de la IA en términos de transparencia, interpretabilidad y justicia. Con este estudio, se aseguró de que los modelos de IA sean transparentes y fácilmente interpretables por los profesionales de la salud. Además, se probaron rigurosamente los algoritmos en con diversas configuraciones y escenarios clínicos para evaluar su rendimiento y generalización.
- Aplicaciones en la detección de anomalías: Los resultados demuestran el potencial de la IA para mejorar la detección temprana y precisa de anomalías en datos médicos, como los ECG. Mediante el análisis de grandes volúmenes de datos, los algoritmos de IA pueden identificar patrones sutiles que pueden indicar la presencia de enfermedades cardíacas u otros trastornos. Esta capacidad permite a los profesionales de la salud intervenir pro activamente y proporcionar un tratamiento oportuno a los pacientes, lo que puede mejorar significativamente los resultados clínicos.
- Revisión con cardiólogos: Al revisar el modelo con los cardiólogos, se les explicó el modelo, cómo funcionaba y los resultados obtenidos. Los cardiólogos explicaron que el uso de la IA y que tan útil es para su área. Los resultados para ellos les demuestra que hay una alta posibilidad que la IA les ayude en sus diagnósticos, y que la precisión al ser relativamente baja, se deja mayormente a interpretación del cardiólogo, pues debe ser quien determine lo que pasa en el ECG, pero la ayuda que da la IA es útil para diferentes casos y poder detectar enfermedades en etapas tempranas de una enfermedad.
- Recomendaciones: Los cardiólogos recomendaron revisar el Código Minnesota" para los resultados de un ECG, el cual es utilizado para categorizar diferentes tipos de patrones anormales encontrados en un ECG. Este sistema fue desarrollado por investigadores en el estado de Minnesota en los Estados Unidos. El código consta de una serie de letras y números que representan distintas características del ECG, como la forma de las ondas, la duración de los intervalos, etc. Estos códigos ayudan a los médicos a comunicar de manera concisa y precisa las anomalías observadas en el ECG. Sin embargo, se destacó que el uso del código Minnesota puede variar según el contexto clínico y la preferencia del médico o centro médico.

E. Usabilidad

Los resultados de este proyecto se pueden extraer de las respuestas a la encuesta elaborada y enviada a cardiólogos expertos en su área de la medicina. Estas respuestas contienen información acerca de cuán usable es el software, qué impacto tendría este en su día a día y comentarios acerca del mismo. Luego de ver e interactuar con el software se obtuvieron dos respuestas a la encuesta.

Figura 16

Resultados de la encuesta acerca de lo intuitivo del software



Nota Elaboración propia.

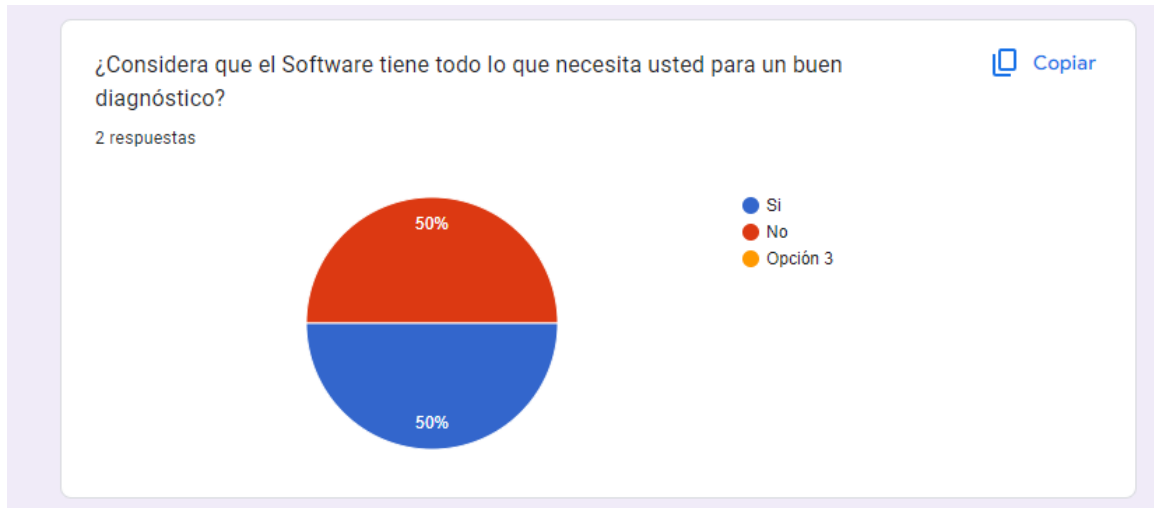
Los resultados de la encuesta acerca de la intuitividad del software muestran una alta satisfacción entre los encuestados. De un total de dos personas encuestadas, el 100 % expresó que consideran el software como intuitivo. Este hallazgo sugiere que el diseño y la usabilidad del software son efectivos en términos de facilitar la interacción y comprensión por parte de los usuarios.

Los resultados de la encuesta sobre las funcionalidades proporcionadas por el software revelan una percepción mixta entre los encuestados. Uno de los dos participantes indicó que el software proporciona las características necesarias, mientras que el otro expresó que le gustaría ver una función adicional específica. En concreto, mencionó la necesidad de contar con una sección que permita visualizar el historial de los exámenes, con la capacidad de filtrarlos por fecha o por alguna anomalía cardíaca detectada. Esta retroalimentación proporciona una dirección clara para posibles mejoras o desarrollos futuros del software, con el objetivo de satisfacer mejor las necesidades y expectativas de los usuarios.

Los resultados de la encuesta sobre la cantidad de clics necesarios para visualizar los exámenes reflejan una opinión unánime entre los participantes. Ambos encuestados coincidieron en que la cantidad de clics requeridos es mínima, lo que sugiere que el software presenta una experiencia de usuario eficiente en este aspecto. Esta percepción positiva indica que el diseño de la interfaz de usuario logra simplificar el proceso de acceso a los exámenes cardíacos, lo que potencialmente mejora la productividad y la experiencia general del usuario.

Figura 17

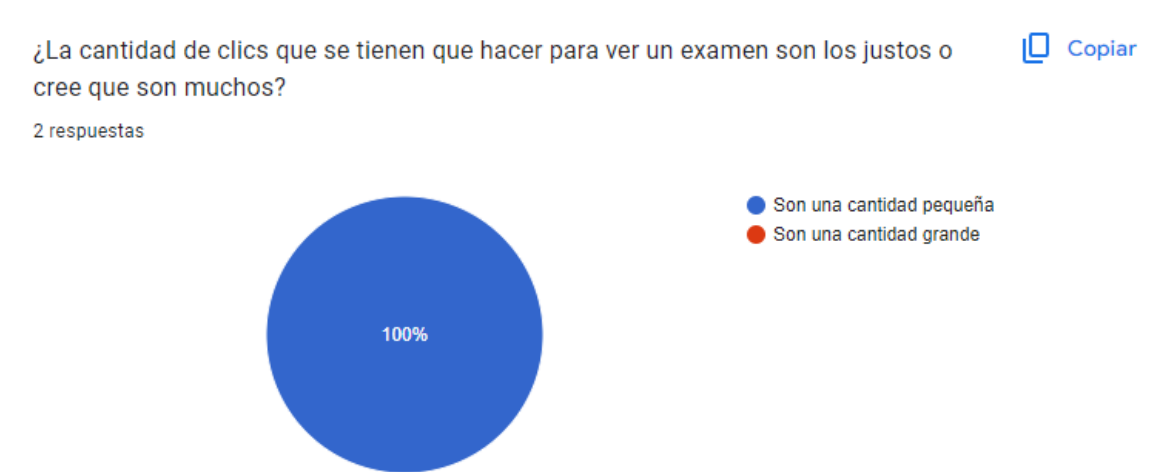
Resultados de la encuesta acerca de lo que proporciona el Software



Nota Elaboración propia.

Figura 18

Resultados de la encuesta acerca de la cantidad de clics



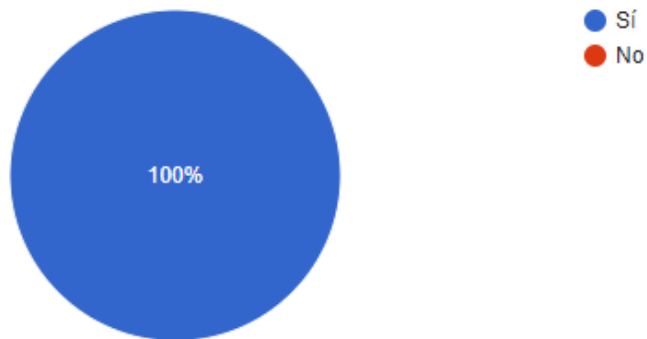
Nota Elaboración propia.

Figura 19

Resultados de la encuesta acerca de la utilidad del software

¿Diría que esta aplicación sería útil para usted?

2 respuestas



Nota Elaboración propia.

Los resultados de la encuesta sobre la utilidad del software revelan una percepción favorable por parte de los encuestados. Ambos participantes indicaron que consideran que el software sería útil para su día a día. Esta respuesta sugiere que los usuarios perciben que el software ofrece beneficios significativos que pueden integrarse de manera efectiva en su práctica diaria. La valoración positiva de la utilidad del software respalda la relevancia y la pertinencia de su desarrollo, destacando su potencial para satisfacer las necesidades y expectativas de los usuarios en su contexto profesional.

F. Consideraciones HIIPA

El cumplimiento con la normativa HIPAA (Health Insurance Portability and Accountability Act) es fundamental para garantizar la seguridad y privacidad de la información médica sensible. En esta sección se describen las medidas implementadas para proteger la confidencialidad de los datos en el sistema, asegurando que solo personal autorizado pueda acceder a la información y que los datos se mantengan seguros mediante cifrado robusto.

1. Confidencialidad

Acceso controlado

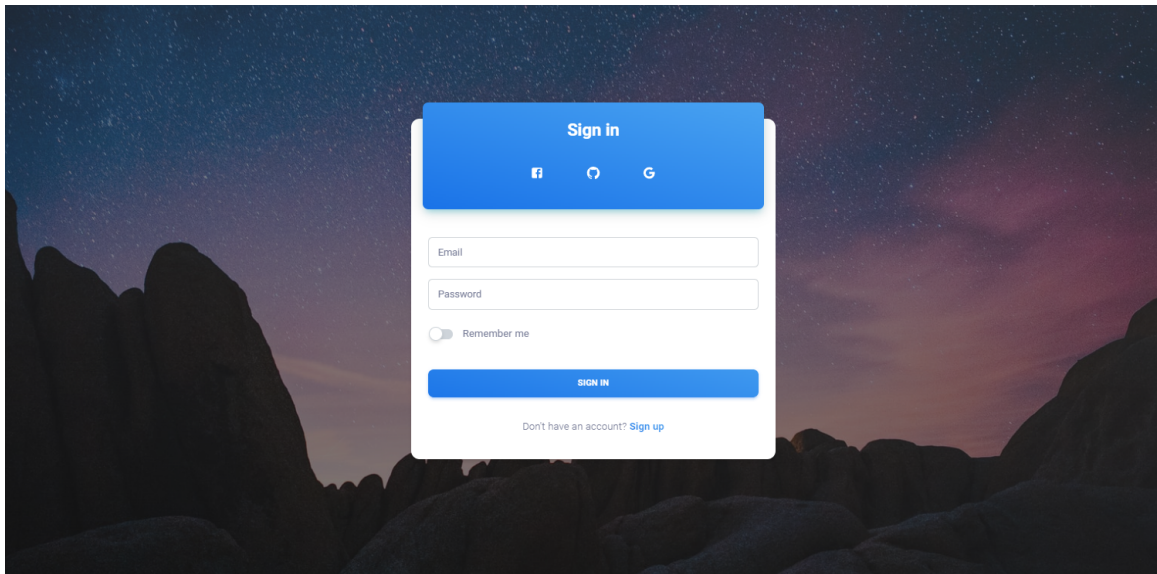
El software cuenta con un login en el cual solo las personas autorizadas pueden acceder al mismo.

Cifrado de datos

Todos los datos de inicio de sesión y el ingreso de datos a la base de datos se encriptaron con librerías de código abierto las cuales hicieron que el procesamiento de datos sea un poco más pesado, pero es necesaria para la protección de los datos.

Figura 20

Inicio de Sesión de la Aplicación



Nota Elaboración propia.

2. Integridad

Control de versiones

Se implementó un control de versiones con la tecnología de Git, la cual fue muy versátil y ayuda a que nada del código sea modificado y todo sea autorizado por un administrador.

Backup y recuperación

La base datos se construyó en SQL, la cual se tuvo opciones para hacer backups de manera eficiente, continua y versátil, como puede verse en la Figura 23.

3. Disponibilidad

Mantenimiento regular

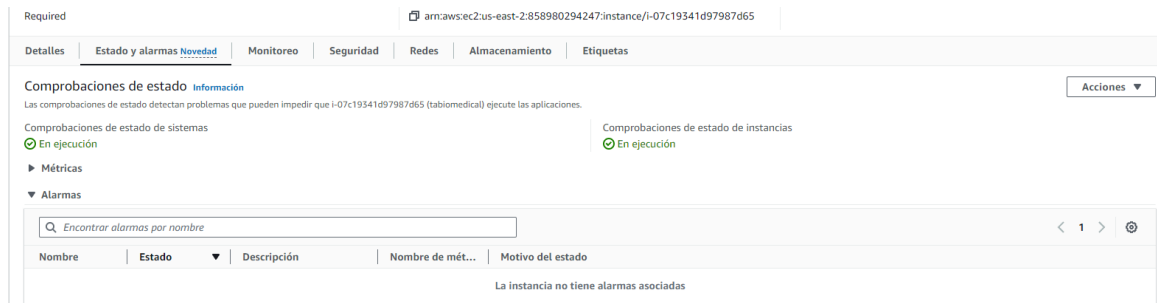
El mantenimiento fue posible ya que el software está montado en un servidor en la nube, específicamente en AWS, la cual proporciona una infraestructura segura y servicios

que pueden ser utilizados para crear aplicaciones que cumplen con HIPAA (AWS HIPAA Compliance), como puede verse en la Figura 33.

4. Notificación y monitoreo

El monitoreo continuo se logró gracias a que AWS tiene servicios de monitoreo continuo para saber el estado de la aplicación en todo momento o alertas por si hay algún inconveniente con la arquitectura implementada.

Figura 21
Monitoreo en AWS



Nota Elaboración propia.

- Se logró realizar un software para que los cardiólogos estuvieran cómodos y conformes para poder atender a sus pacientes de manera remota y en tiempo real.
- Se logró demostrar que la interfaz es intuitiva y sencilla para los cardiólogos y puedan identificar cada una de las partes del software.
- Basado en los resultados de las pruebas, los cardiólogos afirmaron que utilizarían este software para la atención de sus pacientes.
- Se concluye que el apoyo a la integración del Machine Learning para el apoyo en el diagnóstico del paciente sería útil para ellos.
- Los cardiólogos identificaron puntos de mejora y oportunidades para poder agregar al software para el mejoramiento del mismo.

.. Recomendaciones

- Se recomienda seguir evaluando y mejorando el sistema de manera regular para mantenerlo actualizado con los avances tecnológicos y las necesidades del usuario/cardiólogos.
- Realizar pruebas adicionales con cardiólogos y especialistas médicos para validar la precisión y eficacia del sistema en la interpretación de datos cardíacos, incorporando sus comentarios y recomendaciones para mejorar la calidad del diagnóstico.
- Organizar sesiones de capacitación periódicas para usuarios y profesionales de la salud, asegurando que estén familiarizados con todas las funcionalidades y características del sistema.
- Garantizar la seguridad y confidencialidad de los datos del paciente mediante el uso de prácticas de seguridad de la información robustas y el cumplimiento de regulaciones de protección de datos.
- Investigar la posibilidad de integrar el sistema con otras plataformas o dispositivos médicos para ampliar su funcionalidad y utilidad en diferentes entornos clínicos.
- Realizar estudios y análisis para evaluar el impacto del sistema en la detección temprana y el manejo de enfermedades cardiovasculares, midiendo indicadores de salud y calidad de vida de los pacientes.
- Iteraciones adicionales en el diseño de la interfaz de usuario para mejorar la usabilidad y la experiencia del usuario, asegurando que sea intuitiva y fácil de navegar tanto para pacientes como para profesionales de la salud.
- Explorar la adición de funcionalidades avanzadas de acceso remoto, como la capacidad de realizar consultas virtuales con cardiólogos en tiempo real y compartir fácilmente resultados y datos con otros profesionales de la salud.
- Permitir la personalización del software para adaptarse a las preferencias y necesidades específicas de cada cardiólogo, como la configuración de alertas y notificaciones personalizadas, y la visualización de datos según las preferencias individuales.

-
- ¿Qué es la inteligencia artificial (IA)?* (s.f.). Oracle México. <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>
- ¿Qué es machine learning?* (s.f.). IBM. <https://www.ibm.com/mx-es/topics/machine-learning>
- ¿Qué es un software médico?* (s.f.). Medilink. <https://www.softwaremedilink.com/blog/que-es-un-software-medico>
- ¿Qué son los servicios de nube?* (s.f.). Red Hat. <https://www.redhat.com/es/topics/cloud-computing/what-are-cloud-services>
- Acerca de las API de WebSocket en API Gateway.* (s.f.). AWS. https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/apigateway-websocket-api-overview.html
- Dugarte, N., Jugo, D., & Raimondi, V. (2007). Sistema para digitalización del electrocardiograma e historia médica computarizada aplicada a cardiología. http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0798-04772007000100004
- Enfermedades cardiovasculares.* (s.f.). Organización Panamericana de la Salud. Consultado el 5 de mayo de 2024, desde <https://www.paho.org/es/temas/enfermedades-cardiovasculares>
- Explainable AI.* (s.f.). Google. <https://cloud.google.com/explainable-ai?hl=es>
- García, M. R. (2023, julio). *¿Cómo garantizar un uso responsable de la inteligencia artificial (IA)?* <https://blogs.manageengine.com/espanol/2023/07/14/como-garantizar-uso-responsable-inteligencia-artificial.html#:~:text=%C2%BFQu%C3%A9%20es%20una%20Inteligencia%20Artificial,%2C%20inclusi%C3%B3n%2C%20transparencia%20y%20responsabilidad>
- Health Insurance Portability and Accountability Act of 1996 (HIPAA).* (s.f.). CDC. <https://www.cdc.gov/phlp/publications/topic/hipaa.html>
- Jaramillo, N., Malkov, D., Nikakis, J., Arora, U., & Cohen, T. (2022). At-home ECG monitoring with a real-time outpatient cardiac telemetry system during the COVID-19 pandemic. <https://pubmed.ncbi.nlm.nih.gov/35946871/>
- Mappangara, I., Qanitha, A., Uiterwaal, C., & Henriques, J. (2020). Tele-ECG consulting and outcomes on primary care patients in a low-to-middle income population: the

- first experience from Makassar telemedicine program, Indonesia. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7702690/>
- Morales, E. S. (2023, noviembre). *IA responsable: claves para comprender el futuro de la inteligencia artificial*. <https://www.softwareoit.es/guias-software/ia-responsable-futuro-inteligencia-artificial.html>
- on Computer Science, B. (2024, marzo). *What Is One Class SVM and How Does It Work? / Baeldung on Computer Science*. Baeldung On Computer Science. <https://www.baeldung.com/cs/one-class-svm>
- Papademetris, X., Quraishi, A., & Licholai, G. (2022). *Introduction to Medical Software: Foundations for Digital Health, Devices, and Diagnostics*. Cambridge University Press. <https://www.amazon.com/Introduction-Medical-Software-Foundations-Diagnostics/dp/1316514994>
- Roldán, P., Tarraga, L., Madrona, F., Sadeh, I., Celada, C., Panisello, J., & Tárraga, P. (2019). La telemedicina en la insuficiencia cardíaca, el camino pendiente. *Universidad Complutense de Madrid*. <https://revistas.proeditio.com/jonnpr/article/view/3201/PDF3201>

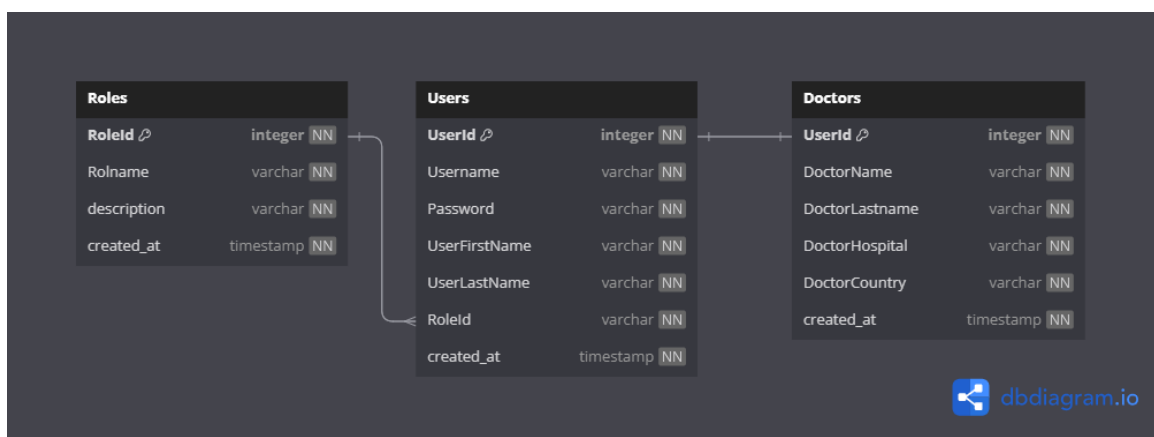
En este capítulo se presentan los anexos correspondientes al desarrollo del software de monitoreo de exámenes cardíacos en tiempo real.

A. Diagrama y código fuente de la base de datos

Se adjunta un diagrama que representa la estructura y relaciones de la base de datos tanto de la relacional como la no relacional utilizada en el desarrollo del software. El diagrama proporciona una visión general de cómo se organizan y relacionan entre sí las entidades y tablas de ambas bases de datos.

Figura 22

Diagrama de la base de datos del software de monitoreo cardíaco



Nota Elaboración propia.

Figura 23

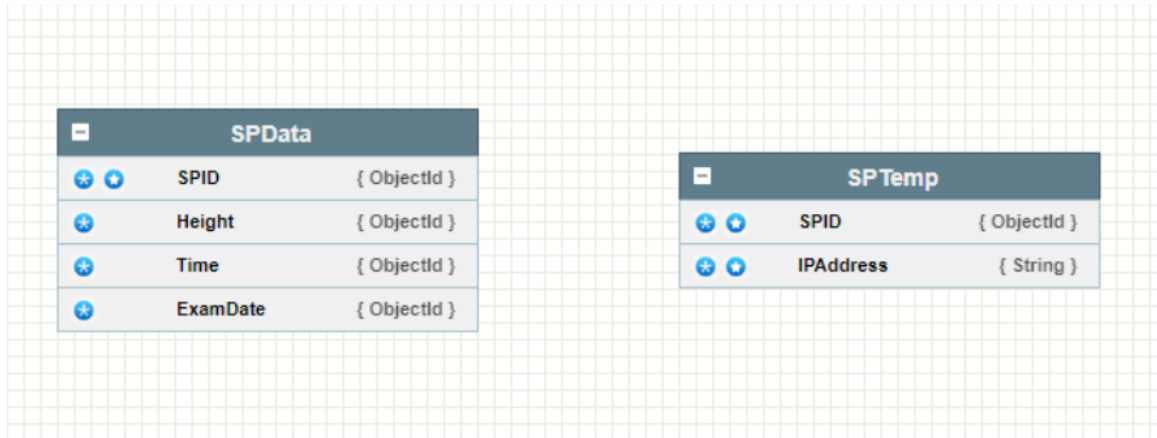
Código de la base de datos del software de monitoreo cardíaco

```
CREATE TABLE [Users] (  
  [UserId] integer PRIMARY KEY NOT NULL IDENTITY(1, 1),  
  [Username] nvarchar(255) NOT NULL,  
  [Password] nvarchar(255) NOT NULL,  
  [UserFirstName] nvarchar(255) NOT NULL,  
  [UserLastName] nvarchar(255) NOT NULL,  
  [RoleId] nvarchar(255) NOT NULL,  
  [created_at] timestamp NOT NULL  
)  
GO  
  
▷ Execute | JSON | Copy  
CREATE TABLE [Roles] (  
  [RoleId] integer PRIMARY KEY NOT NULL IDENTITY(1, 1),  
  [Rolname] nvarchar(255) NOT NULL,  
  [description] nvarchar(255) NOT NULL,  
  [created_at] timestamp NOT NULL  
)  
GO  
  
▷ Execute | JSON | Copy  
CREATE TABLE [Doctors] (  
  [UserId] integer PRIMARY KEY NOT NULL IDENTITY(1, 1),  
  [DoctorName] nvarchar(255) NOT NULL,  
  [DoctorLastname] nvarchar(255) NOT NULL,  
  [DoctorHospital] nvarchar(255) NOT NULL,  
  [DoctorCountry] nvarchar(255) NOT NULL,  
  [created_at] timestamp NOT NULL  
)  
GO  
  
▷ Execute | JSON  
ALTER TABLE [Users] ADD FOREIGN KEY ([RoleId]) REFERENCES [Roles] ([RoleId])  
GO  
  
▷ Execute | JSON  
ALTER TABLE [Doctors] ADD FOREIGN KEY ([UserId]) REFERENCES [Users] ([UserId])
```

Nota Elaboración propia.

Figura 24

Diagrama de la base de datos no Relacional del software de monitoreo cardíaco



Nota Elaboración propia.

B. Capturas de pantalla de la pantalla principal de la página

En las figuras 26 y 27 se incluyen capturas de pantalla que muestran la interfaz de usuario de la página principal del software. Estas imágenes permiten visualizar el diseño y las funcionalidades principales del software desarrollado.

C. Documentación del API desarrollado

Se proporciona la documentación técnica del API desarrollado para el intercambio de datos entre el hardware de monitoreo y el software de visualización de exámenes cardíacos. La documentación detalla los endpoints disponibles, los métodos de autenticación y los formatos de datos admitidos.

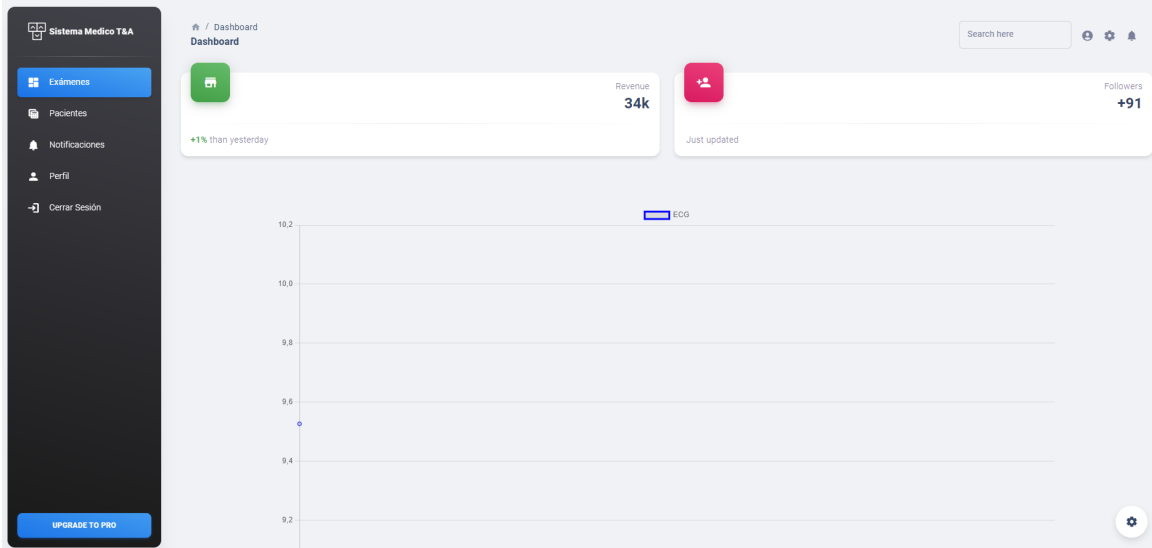
Figura 25

Código de la base de datos no Relacional del software de monitoreo cardíaco

```
[
  {
    "schemaType": "Collection",
    "name": "SPTemp",
    "type": "ObjectId",
    "required": true,
    "unique": true,
    "defaultValue": "",
    "description": "",
    "index": 0,
    "customProps": [],
    "fields": [
      {
        "schemaType": "Field",
        "name": "SPID",
        "type": "ObjectId",
        "required": true,
        "unique": true,
        "defaultValue": "",
        "description": "",
        "index": 0,
        "customProps": []
      },
      {
        "schemaType": "Field",
        "name": "IPAddress",
        "type": "String",
        "required": true,
        "unique": true,
        "defaultValue": "",
        "description": "",
        "index": 0,
        "customProps": []
      }
    ]
  },
  {
    "schemaType": "Collection",
    "name": "SPData",
    "type": "ObjectId",
    "required": true,
    "unique": true,
    "defaultValue": "",
    "description": "In this table all the data will be stored",
    "index": 0,
    "customProps": [],
    "fields": [
      {
        "schemaType": "Field",
        "name": "SPID",
        "type": "ObjectId",
        "required": true,
        "unique": true,
        "defaultValue": "",
        "description": "",
        "index": 0,
        "customProps": []
      },
      {
        "schemaType": "Field",
        "name": "Height",
        "type": "ObjectId",
        "required": true,
        "unique": false,
        "defaultValue": "",
        "description": "",
        "index": 0,
        "customProps": []
      },
      {
        "schemaType": "Field",
        "name": "Time",
        "type": "ObjectId",
        "required": true,
        "unique": 0,
        "defaultValue": "",
        "description": "",
        "index": 0
      },
      {
        "schemaType": "Field",
        "name": "ExamDate"
      }
    ]
  }
]
```

Nota Elaboración propia.

Figura 26
Captura de pantalla de la pantalla principal del software



Nota Elaboración propia.

Figura 27
Captura de pantalla de la pantalla de pacientes del software

AUTHOR	FUNCTION	STATUS	EMPLOYED	ACTION
John Michael john@creative-tim.com	Manager Organization	ONLINE	23/04/18	Edit
Alexa Liras alex@creative-tim.com	Programator Developer	OFFLINE	11/01/19	Edit
Laurent Perrier laurent@creative-tim.com	Executive Projects	ONLINE	19/09/17	Edit
Michael Levi michael@creative-tim.com	Programator Developer	ONLINE	24/12/08	Edit
Richard Gran richard@creative-tim.com	Manager Executive	OFFLINE	04/10/21	Edit
Miriam Eric miriam@creative-tim.com	Programator Developer	OFFLINE	14/09/20	Edit

Nota Elaboración propia.

Figura 28

Captura de pantalla del código implementado para los controladores del API

```
'use strict'

const connection = require('../database/connection')

async function getResults(req, res) {
  const result = await connection.getConnections("SELECT * FROM Results")
  res.status(200).json(result)
}

function registerUser(req, res) {
  const { name, password, country, role } = req.body
  config.pool.query("INSERT INTO Results (name, password, country, role) VALUES ($1, $2, $3, $4)",
    [name, password, country, role], (err, results) => {
      if (err) {
        throw err
      }
      console.log(results);
      res.status(201).json("Usuario Agregado")
    })
}

function updateUser(req, res) {
  const { name, password, country, role } = req.body
  config.pool.query("INSERT INTO Results (name, password, country, role) VALUES ($1, $2, $3, $4)",
    [name, password, country, role], (err, results) => {
      if (err) {
        throw err
      }
      console.log(results);
      res.status(201).json("Usuario Agregado")
    })
}
```

Nota Elaboración propia.

Figura 29

Captura de pantalla del código implementado para las rutas del API

```
const express = require('express')
const api = express.Router()

const DoctorController = require('../controllers/DoctorController')
const UserController = require('../controllers/UserController')
const RoleController = require('../controllers/RoleController')
const ResultsController = require('../controllers/ResultsController')

api.get('/', (request, response) => {
  console.log('llega')
  response.json({ info: 'Node.js, Express, and Postgres API' })
})

//User
api.post('/user/login/', UserController.loginUser)
api.post('/user/register/', UserController.registerUser)
api.get('/user/getAllUsers/', UserController.getAllUsers)

//Travel
api.get('/docto/get/', DoctorController.getDoctors)

api.get('/role/get/', RoleController.getRoles)

api.get('/results/getAllResults/', ResultsController.getAllResults)

module.exports = api
```

Nota Elaboración propia.

Figura 30

Captura de pantalla del código implementado para la conexión del API

```
const sql = require("mysql");
const config = require("../config");

async function getConnections(query) {
  try {
    await sql.connect(config.sqlConfig);
    const result = await sql.query(query)
    return result;
  } catch (err) {
    console.log(err);
  }
}

module.exports = {
  getConnections
}
```

Nota Elaboración propia.

D. Programa de machine learning para detectar anomalías en los ECG

Se adjunta información sobre el programa de machine learning utilizado para detectar anomalías en los electrocardiogramas (ECG). Se incluye una descripción de la estructura del programa, los algoritmos empleados y los resultados obtenidos en pruebas de validación.

Figura 31

Captura de pantalla de la configuración de la IA para detección de anomalías en el ECG

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import OneClassSVM
from sklearn.metrics import roc_auc_score, average_precision_score, roc_curve, precision_recall_curve
import matplotlib.pyplot as plt

# Cargar los datos desde el archivo CSV
data = pd.read_csv("../grafico_analogico 3.csv")

# Seleccionar la característica (AnalogicValue)
X = data["AnalogicValue"].values

# Etiquetar los datos de prueba
y_test = np.zeros(len(X)) # Todos los datos se etiquetan como normales (0)
y_test[-10:] = 1 # Los últimos 50 ejemplos se etiquetan como anomalías (1)

# Ajustar el modelo OneClassSVM con nu=0.5
nu = 0.02
model = OneClassSVM(nu=nu)
model.fit(X.reshape(-1, 1))

# Predecir las anomalías (valores atípicos)
predicciones = model.predict(X.reshape(-1, 1))

# Calcular el AUC-ROC y el AUC-PR
auc_roc = roc_auc_score(y_test, -predicciones)
auc_pr = average_precision_score(y_test, -predicciones)

# Calcular la curva ROC
fpr, tpr, _ = roc_curve(y_test, -predicciones)

# Calcular la curva PR
precision, recall, _ = precision_recall_curve(y_test, -predicciones)

# Imprimir los resultados de las métricas
print("Resultados de las métricas:")
print("-----")
print("Porcentaje de anomalías en los datos: {:.2f}%".format((sum(predicciones == -1) / len(X)) * 100))
print("Anomalías en los datos: {:.2f} anomalías de {:.2f} datos".format(sum(predicciones == -1), len(X)))
print("AUC-ROC: {:.2f}".format(auc_roc))
print("AUC-PR: {:.2f}".format(auc_pr))
print("-----")

# Análisis de los resultados
if auc_roc >= 0.9:
    print("El AUC-ROC indica un excelente rendimiento en la detección de anomalías.")
elif auc_roc >= 0.8:
```

Nota Elaboración propia.

Figura 32

Captura de pantalla de la impresión en pantalla de los resultados de la IA para detección de anomalías en el ECG

```
# Análisis de los resultados
if auc_roc >= 0.9:
    print("El AUC-ROC indica un excelente rendimiento en la detección de anomalías.")
elif auc_roc >= 0.8:
    print("El AUC-ROC indica un buen rendimiento en la detección de anomalías.")
elif auc_roc >= 0.7:
    print("El AUC-ROC indica un rendimiento razonable en la detección de anomalías.")
else:
    print("El AUC-ROC sugiere que se podría mejorar el rendimiento en la detección de anomalías.")

if auc_pr >= 0.9:
    print("El AUC-PR indica un excelente rendimiento en la detección de anomalías.")
elif auc_pr >= 0.8:
    print("El AUC-PR indica un buen rendimiento en la detección de anomalías.")
elif auc_pr >= 0.7:
    print("El AUC-PR indica un rendimiento razonable en la detección de anomalías.")
else:
    print("El AUC-PR sugiere que se podría mejorar el rendimiento en la detección de anomalías.")

# Visualizar los resultados
plt.figure(figsize=(12, 5))

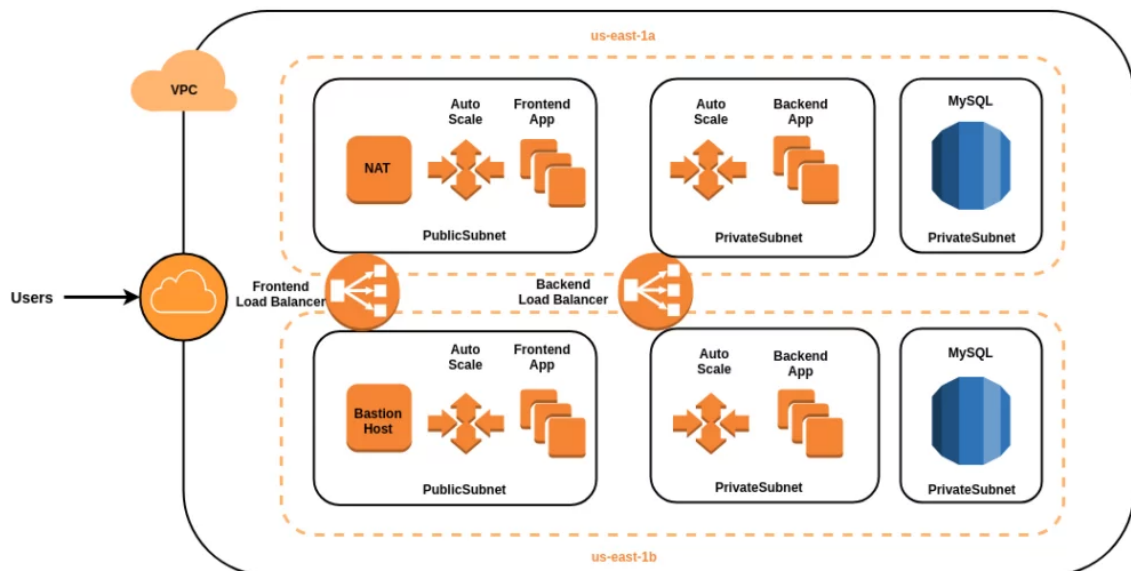
# Graficar la curva ROC
plt.subplot(1, 2, 1)
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (AUC = {:.2f})'.format(auc_roc))
plt.plot([0, 1], [0, 1], color='red', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")

# Graficar la curva PR
plt.subplot(1, 2, 2)
plt.plot(recall, precision, color='green', lw=2, label='PR curve (AUC = {:.2f})'.format(auc_pr))
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.legend(loc="lower right")

plt.tight_layout()
plt.show()
```

Nota Elaboración propia.

Figura 33
Cumplimiento AWS HIPAA (*AWS HIPAA Compliance*)



Nota Adaptado de Creating HIPAA Compliance Architecture, Tudip Technologies, sf.