UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería


# Diseño y creación de una página web con despliegue en la nube para seguridad y protección de la información en áreas de salud

Trabajo de graduación presentado por Rebecca Maria Smith Martínez para optar al grado académico de Licenciada en Ingeniería en Ciencia de la Computación y Tecnologías de la Información

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería


# Diseño y creación de una página web con despliegue en la nube para seguridad y protección de la información en áreas de salud

Trabajo de graduación presentado por Rebecca Maria Smith Martínez para optar al grado académico de Licenciada en Ingeniería en Ciencia de la Computación y Tecnologías de la Información

Guatemala,

2025

Vo. Bo.:

(F)_____
Ing. Gabriel Brolo

Tribunal:

(F)_____
MSc. Douglas Lenel Barrios

Fecha de aprobación:

Guatemala, 6 de junio de 2025

# Prefacio

En la actualidad, la ciberseguridad de la información se ha vuelto cada vez más importante debido al creciente volumen de datos sensibles que se generan, transmiten y almacenan digitalemente, especialmente en sectores críticos como la salud, educación y finanzas. Estos avances han traído una mejora significativa en la eficiencia de los procesos, mayor conectividad entre personas e instituciones, y un acceso más ágil a la información. Sin embargo, también han expuesto a individuos y organizaciones a nuevos riesgos relacionados con la protección de los datos, la privacidad y la integridad de los sistemas digitales.

Particularmente en el área de la salud, la información que se maneja es altamente sensible, ya que incluye datos personales, médicos y administrativos de los pacientes. Garantizar la seguridad de esta información no solo es una necesidad técnica, sino también ética y legal, ya que el mal manejo de datos sensibles puede comprometer la privacidad de los pacientes, pérdida de confianza y consecuencias legales. Las instituciones deben estar preparadas para proteger los datos contra accesos no autorizados, filtraciones y manipulaciones, todo mientras mantienen la disponibilidad y precisión de la información clínica ya que si no lo hacen, esta información puede ser comprometida.

Este trabajo nace por el deseo de aplicar conocimientos de desarrollo web y ciberseguridad para poder aportar soluciones a los sectores más críticos: el manejo de la información médica. Esta problemática es relevante en el contexto local, donde muchas clínicas pequeñas y medianas aún carecen de sistemas digitalizados seguros, lo que incrementa el riesgo a la pérdida de datos, accesos no autorizados y exposición de información sensible. A través de la investigación, implementación y validación de una plataforma web segura y funcional, se busca ofrecer una alternativa accesible para clínicas pequeñas y medianas que necesiten digitalizar sus procesos de forma responsable.

La realización de este trabajo representa un reto técnico y académico en el cuál hubo mucho aprendizaje además de crecimiento personal y profesional. Quisiera mostrar mi agradecimiento principalmente a papás quienes me apoyaron y mostraron su confianza y amor en todo momento. Su confianza en mi ayudó a la motivación para siempre seguir adelante y siempre dar lo mejor de mí. A mi asesor, **Ing. Gabriel Brolo** por su apoyo, guía y dedicación a lo largo de este proyecto. Finalmente a mis amigos por estar presentes en los momentos más importantes, por su compañía incondicional.

# Índice

# Lista de figuras

# Resumen

El presente trabajo describe el diseño y desarrollo de una plataforma web para el manejo de información médica en entornos clínicos. La plataforma fue implementada usando React para el frontend, Node.js en el backend y PostgreSQL como sistema de gestión de base de datos, alojado en Amazon RDS. Esta arquitectura asegura escalabilidad, modularidad y compatibilidad con servicios en la nube. Permitiendo un entorno accesible y seguro para la administración de datos clínicos, implementando lineamientos basados en los frameworks de seguridad: HIPAA, NIST, CSA, CMMC y SLSA. Con ello se garantiza confidencialidad, integridad y disponibilidad de los datos en la plataforma. Con ello también se aseguró protección frente a amenazas externas. Se integraron mecanismos de autenticación segura por medio de tokens JWT.

La aplicación fue sometida a pruebas de seguridad usando las siguientes herramientas: SonarQube (análisis estático) y OWASP ZAP (análisis dinámico). No se obtuvieron vulnerabilidades críticas. Se obtuvieron 1 de riesgo bajo y 3 informativas. Asimismo, se evaluó la experiencia de usuario por medio del método System Usability Scale (SUS): se obtuvo una puntuación de 98.5, por lo que se considera que la plataforma es usable. Estos resultados demuestran que la plataforma es segura, es fácil de usar y con despliegue en la nube el cual puede adaptarse a las necesidades del sector de salud y es accesible para las clínicas medianas y pequeñas.

# CAPÍTULO 1

## Introducción

Desde hace muchos años la tecnología ha venido a ayudar y facilitar muchos trabajos, desde tareas hogareñas hasta construcciones, pero aún más importante, la manera de guardar información. Se ha pasado de tener que usar espacio físico y labor manual a usar dispositivos electrónicos para guardar y encontrar información. Este es un enfoque práctico, pero puede presentar problemas con relación a la seguridad de la información ante potenciales brechas de seguridad, las cuales pueden provocar considerables pérdidas monetarias estimadas en millones de dólares [1]. En algunas ocasiones, las instituciones públicas y privadas también pueden ser víctimas de ransomware, promediando pérdidas de millones de dólares y pérdida de reputación [2].

En este contexto, es importante comprender que hay mucho riesgo y amenazas, como el robo de información privada de los pacientes, que enfrentan los hospitales y clínicas médicas, por lo que es necesario aplicar seguridad en los sistemas computacionales que alojan y procesan la información personal de los pacientes en el área de salud, así como en la infraestructura y arquitecturas de software que sostienen estos procesos de alojamiento y procesamiento de datos. Dado que la información alojada por estas instituciones es confidencial, puede ser un objetivo para actores maliciosos, los cuales no solo pueden venderla a terceros, sino también utilizarla para realizar fraudes, extorsiones o suplantación de identidad. Por ejemplo, al obtener historiales médicos, datos personales o información financiera vinculada, los atacantes pueden acceder a seguros médicos de forma fraudulenta, manipular recetas electrónicas o chantajear a individuos o instituciones. Además, estos incidentes pueden provocar interrupciones operativas graves, pérdida de confianza por parte de los pacientes y sanciones legales significativas si se incumplen normativas como HIPAA. Por ello, proteger esta información no solo es una cuestión técnica, sino también ética, legal y social. Por ello, es importante establecer controles para la seguridad de la información en estos sistemas administrados por las instituciones de salud [3]. El correcto manejo y seguridad de esta información asegura la protección de la privacidad y confidencialidad de los datos de los pacientes, y también contribuye a una mejor disponibilidad de la información para los trabajadores del área de salud, quienes requieren un acceso óptimo y seguro a esta información [4]. Es responsabilidad de estas instituciones proteger la información personal identificable y confidencial de los

pacientes en áreas de salud, así como la privacidad de esta información. Esto aplica también para el personal de salud que atiende a dichos pacientes y a todo el ecosistema administrativo detrás de los procesos asociados con el manejo de la información en general dentro de las instituciones de salud [5].

CAPÍTULO 2

# Antecedentes

## 2.1. Antecedentes

El manejo de la información médica, la cual hace referencia a todos los datos relacionados con la salud del paciente, representa un desafío constante, especialmente en clínicas pequeñas y medianas donde los recursos pueden ser limitados. Estas instituciones enfrentan dificultades para implementar sistemas digitales seguros, mantener la integridad y disponibilidad de los datos, y cumplir con normativas de privacidad como HIPAA o leyes locales de protección de datos. La falta de personal técnico, infraestructura tecnológica adecuada o protocolos de ciberseguridad incrementa el riesgo de pérdida, filtración o acceso no autorizado a información crítica, lo que puede comprometer tanto la salud del paciente como la reputación de la institución.

Para solucionar estos problemas, han surgido diversas soluciones de sistemas de historia clínica (EHR), como OpenEMP, Epic o MediTracker. Estos buscan digitalizar la gestión de información de los pacientes. Sin embargo, estas soluciones presentan barreras de entrada como costos elevados, requerimientos técnicos o falta de flexibilidad, en especial para clínicas pequeñas por sus recursos limitados.

Con el avance de la tecnología y servicios en la nube, ha sido posible diseñar plataformas accesibles, seguras y específicas. Las soluciones basadas en la nube eliminan la necesidad de infraestructura local costosa lo cual hace que las clínicas pequeñas puedan acceder a ellas sin necesidad de inversiones significativas en servidores o mantenimiento de estos. Esta accesibilidad facilita la adopción tecnológica en entornos con recursos limitados, promoviendo la digitalización en sectores que antes dependían completamente de procesos manuales. Según la Cloud Security Alliance, el uso de soluciones en la nube también mejora la disponibilidad, escalabilidad y cumplimiento normativo en sistemas que manejan información médica confidencial [6].

# CAPÍTULO 3

## Justificación

Es importante establecer controles de seguridad de la información sobre toda la arquitectura de software que sostiene a los sistemas de ingestión y visualización de los datos, el backend que procesa la información, las bases de datos que alojan la disponibilidad e integridad de la información y la infraestructura en la nube que sostiene estos servicios donde esta información es alojada, entre otros, para poder proteger la confidencialidad, disponibilidad e integridad de la información [7] de los pacientes en entornos médicos. Las repercusiones de no hacerlo pueden ser negativas y expresarse en diversas formas, tales como la pérdida de reputación de la institución de salud, pérdida de dinero, pérdida de contratos, impacto negativo sobre la información personal de las personas, perjudicando su seguridad en general y exponiéndolos a potenciales problemas cómo pérdidas económicas o perjudicar la seguridad de los pacientes de manera que sean extorsionados o divulgan información privada sin consentimiento.

Dado que el manejo y la seguridad de la información en el área de salud representa un entorno vulnerable, en 2005 se aprobó una regla para HIPAA (Health Insurance Portability and Accountability Act) que asegura que se esté cumpliendo la confidencialidad de la información alojada por las instituciones de salud, y permite investigar quejas y aplicar sanciones en caso de incumplimientos [4]. Aunque esta norma aplica al contexto estadounidense, constituye una base sólida para la implementación de un marco de seguridad de la información en salud que puede adaptarse a otros contextos, como el guatemalteco, con los ajustes pertinentes.

Por ello, se decidió trabajar con diversos frameworks de seguridad específicos, los cuales permiten asegurar la plataforma y cumplir con lineamientos establecidos por HIPAA.

Los frameworks utilizados son:

- CSA (Cloud Security Alliance)
- CCM (Cloud Controls Matrix)
- CMMC (Cybersecurity Maturity Model Certification)

- HITRUST CSF (HITRUST Cybersecurity Framework)
- SLSA (Supply-chain Levels for Software Artifacts)
- NIST (National Institute of Standards and Technology)

Estos marcos permiten establecer una infraestructura segura para el almacenamiento de información médica. En particular, el marco CMMC evalúa que se sigan buenas prácticas a través de cinco niveles de madurez, que contemplan el manejo, almacenamiento y transmisión segura de datos por parte de las organizaciones. HITRUST CSF facilita el cumplimiento de HIPAA para instituciones de salud. SLSA, por su parte, mejora la integridad y protege la cadena de suministro del software, mientras que el marco NIST proporciona lineamientos técnicos especializados para proteger la información médica electrónica. La incorporación de estos frameworks permite a las instituciones garantizar que la información esté protegida contra pérdidas, accesos no autorizados o robo de datos [8].

# CAPÍTULO 4

# Objetivos

## 4.1. General

Desarrollar una plataforma web basada en Software as a Service (SAAS) para el manejo de información médica con despliegue en la nube para áreas de salud cumpliendo con estándares y marcos de seguridad reconocidos como HIPAA (Health Insurance Portability and Accountability Act), NIST (National Institute of Standards and Technology), CSA (Cloud Security Alliance) y otros frameworks complementarios como CMMC, HITRUST CSF y SLSA, que aseguren la confidencialidad, integridad y disponibilidad de la información.

## 4.2. Específicos

- Seleccionar e implementar controles de seguridad adecuados según los frameworks elegidos, estableciendo un patrón de seguridad para la arquitectura e infraestructura de la plataforma y garantizando el cumplimiento de estándares y mejores prácticas.
- Construir el código necesario para el front-end, back-end, infraestructura y bases de datos , asegurando una plataforma amigable con el usuario.
- Diseñar e implementar una arquitectura escalable y modular para el despliegue en la nube, compatible con al menos un proveedor de la nube.

# CAPÍTULO 5

# Marco teórico

## 5.1. Ciberseguridad

La triada CIA (Confidencialidad, Integridad y Disponibilidad) es el núcleo de la seguridad de la información y ciberseguridad. Estos tres principios permiten estructurar políticas y controles para proteger la información y los sistemas que la manejan. La Confidencialidad se enfoca en proteger los datos contra accesos no autorizados, la integridad garantiza que los datos no sean modificados de manera no autorizada y la disponibilidad asegura que los datos y sistemas sean accesibles para los usuarios autorizados cuando los necesiten [7].

En este contexto, la ciberseguridad se consolida como un componente esencial de la protección digital. Según la Unión Internacional de Telecomunicaciones (ITU), abarca todas las medidas y prácticas destinadas a salvaguardar redes, sistemas y datos ante amenazas emergentes. De este modo, las estrategias de defensa digital deben contemplar tanto amenazas internas como externas, desarrollando entornos seguros para la información.

José Cano (2011) destaca la necesidad de aplicar estas prácticas de manera que no solo protejan los datos, sino que otorguen sentido a la seguridad digital en su totalidad. Para lograrlo, marcos como el de NIST ofrecen un enfoque integral que permite a las organizaciones identificar, proteger, detectar, responder y recuperarse frente a amenazas, facilitando una gestión más efectiva de la ciberseguridad [9].

## 5.2. Vulnerabilidades

Una vulnerabilidad en el contexto de ciberseguridad es una debilidad o falla en el diseño, implementación, operación o gestión de un sistema de información que puede ser explotada por ciberdelincuentes para obtener acceso no autorizado [3], interrumpir servicios o robar datos sensibles. Según IBM, una vulnerabilidad puede residir tanto en el software como en el hardware y, si no se gestiona adecuadamente, puede ser el punto de entrada para ataques

que causen daños significativos a una organización [2].

La falta de una gestión efectiva de vulnerabilidades puede llevar a incidentes graves, como la pérdida de información confidencial, interrupciones en el servicio y daños a la reputación de la organización. Además, en muchas regiones, las leyes de ciberseguridad pueden imponer multas significativas a las compañías que no protegen adecuadamente sus sistemas.

Una gestión efectiva de vulnerabilidades implica un proceso continuo y sistemático para identificar, tratar y monitorear las vulnerabilidades en los sistemas y redes de una organización. Este proceso permite reducir el riesgo de que amenazas exploten esas vulnerabilidades, lo que ayuda a proteger la integridad, confidencialidad y disponibilidad de los datos y sistemas. La gestión de vulnerabilidades es parte integral de la ciberseguridad y tiene como objetivo prevenir incidentes graves, como pérdida de datos, interrupciones del servicio o daños a la reputación.

## 5.3. Riesgo

En el contexto de la seguridad informática, un **riesgo** se define como la posibilidad de que una amenaza explote una vulnerabilidad en un sistema, provocando un impacto negativo sobre la confidencialidad, integridad o disponibilidad de la información [10]. El riesgo combina tres elementos clave: la existencia de una amenaza, una vulnerabilidad explotable y la probabilidad de que ocurra un evento no deseado. Evaluar los riesgos permite priorizar acciones de mitigación y tomar decisiones informadas para fortalecer la postura de seguridad de una organización.

## 5.4. Ataques cibernéticos

Un ataque cibernético es cualquier esfuerzo intencional para robar, alterar, deshabilitar o destruir información o aplicaciones a través de acceso no autorizado a una red, sistema o hardware [2]. Estos ataques pueden variar en sofisticación y alcance, y pueden dirigirse a cualquier componente del sistema de información.

Estos ataques pueden afectar a empresas de cualquier tamaño, tanto grandes como pequeñas. Es fundamental conocer las estrategias de prevención para proteger a la organización, sus empleados y clientes, evitando pérdidas que puedan impactarlos negativamente. Por esto hay diferentes frameworks de seguridad que se pueden implementar a la hora de crear alguna aplicación que sea utilizada por varias personas y guarde información que es confidencial.

## 5.5. Frameworks de seguridad

Los frameworks de seguiridad son esenciales en todo tipo de aplicación donde el usuario ingrese cualquier tipo de información, ya que permiten que la información esté protegida frente a amenazas, vulnerabilidades y accesos no autorizaods. CSA define un framework de

seguridad como un set de controles de seguiridad, asi como políticas y procedimientos para proteger la data [6].

## 5.6. MITRE ATT&CK

Este marco es una base de conocimientos a la cual se puede acceder para informarse sobre actualizaciones que permiten modelar, detectar, prevenir y combatir amenazas de ciberseguridad, basándose en el análisis del comportamiento de los ciberdelincuentes [11].

El marco se especializa en catalogar todas las etapas de un ciberataque, desde la recopilación inicial de información hasta la ejecución del ataque. Esto permite a las empresas simular y comprender los distintos tipos de ataques a los que pueden estar expuestas.

Implementar este framework es crucial para estudiar y prevenir ataques cibernéticos pasados y potenciales, asegurando así que la aplicación cumpla con los estándares de seguridad definidos.

## 5.7. HIPAA

Las organizaciones deben cumplir con diversas leyes y estándares que regulan la ciberseguridad para proteger datos sensibles y garantizar la privacidad de los usuarios. HIPAA (Health Insurance Portability and Accountability Act) es una ley en EE. UU. la cual establece requisitos de privacidad y seguridad de los datos médicos de las personas y los trabajadores de las empresas que se ajustan a la definición de "socios empresariales" según HIPAA [5]. Los principales objetivos de HIPAA incluyen:

- Privacidad: garantizar que la información médica personal sea utilizada y divulgada de manera adecuada.
- Seguridad: implementar medidas administrativas, físicas y técnicas para proteger la información electrónica de salud (ePHI).
- Notificación de brechas: requerir que las organizaciones informen a los afectados y a las autoridades pertinentes en caso de violaciones de datos.

El incumplimiento de HIPAA puede resultar en sanciones financieras considerables y daños significativos a la reputación de la organización. La implementación de este marco de seguridad en el generador de páginas web garantizará que la información, tanto de los empleados como de los pacientes, se mantenga protegida en un entorno seguro, minimizando el riesgo de exposición o robo de datos sensibles. Esto no solo cumple con las regulaciones, sino que también refuerza la confianza de los usuarios en la protección de su privacidad.

## 5.8. HITECH

En el año 2009, Estados Unidos pasó el acto llamado HITECH (Health Information Technology for Economic and Clinical Health) el cual expande en la ley de HIPAA de 1996. HITECH cambió cómo funcionaban las relaciones entre profesionales de salud y pacientes al mostrar la implementación y uso de tecnología en la información sanitaria [12].

HITECH cambió la naturaleza de las relaciones entre los profesionales de la salud, las organizaciones, los pacientes y los pagadores al centrarse en la implementación y el uso de la tecnología de la información sanitaria. Pone especial énfasis en la privacidad y la seguridad, incluida una aplicación y aplicación ampliadas. HITECH también ofrece incentivos y subsidios para intercambios de información y educación sobre salud, que están fuera del alcance de este artículo.

## 5.9. CSA Cloud Security Alliance

CSA es una organización que promueve el uso de las mejores prácticas para ofrecer garantías de seguridad en Cloud Computing. Además, proporcionan educación sobre los usos de la computación en la nube para asegurar otras formas de informática [13]. CSA tiene un programa de seguridad, confianza y registro de garantías (STAR) para ayudar a que las personas encuentren un proveedor de servicios de la nube usando la autoevaluación, auditoría de terceros y supervisión continua.

## 5.10. CM Cloud Controls Matrix

La Cloud Controls Matrix proporciona fundamentos de seguridad para que los vendedores de la nube puedan asistir a sus clientes. Ellos proveen frameworks para poder entender detalladamente los conceptos de seguridad y principios que están alineados con el CSA [14].

Sus características principales incluyen:

- Fundamentos de seguridad: define controles de seguridad específicos para abordar riesgos en entornos de nube.
- Alineación con estándares: se alinea con otros marcos y normativas, facilitando la integración y el cumplimiento.
- Soporte para proveedores y clientes: ayuda a los proveedores de servicios en la nube a demostrar su compromiso con la seguridad y a los clientes a evaluar la seguridad de los servicios ofrecidos.

## 5.11. CMMC Cybersecurity Maturity Model Certification

CMMC es un framework el cual provee al Departamento de defensa de los Estados Unidos, con verificación que los miembros de la base industrial de defensa, pueden proteger la información federal de contrato e información sin clasificar desde el cliente a contractores primarios o subcontratistas [15].

## 5.12. HITRUST CSF HITRUST cyber security framework

HITRUST CSF (Health Information Trust Alliance cybersecurity framework) fue diseñado para áreas de salud en el 2007 para protegerlos de amenazas cibernéticas. Su estructura está diseñada para ser flexible y escalable, permitiendo a las organizaciones adaptar el marco según sus necesidades específicas.

## 5.13. SLSA Supply-chain Levels for Software artifacts

"Salsa" es un framework de seguridad el cual ayuda a prevenir manipulación y mejorar la integridad para asegurar la infraestructura [16]. Al proporcionar un conjunto estructurado de directrices, mejores prácticas y herramientas, Salsa ayuda a las empresas a fortalecer sus defensas contra amenazas cibernéticas, garantizar la confiabilidad de sus sistemas y cumplir con las normativas de seguridad.

## 5.14. NIST National Institute of Standards And Technology

El National Institute of Standards and Technology (NIST), conocido en español como el Instituto Nacional de Estándares y Tecnología, es una agencia del Departamento de Comercio de los Estados Unidos. Su principal objetivo es promover la innovación y la competitividad industrial mediante el avance de la ciencia y la tecnología. NIST desempeña un papel crucial en el establecimiento de estándares que garantizan la calidad, la seguridad y la eficiencia en diversos sectores industriales y tecnológicos [9]. A través de sus estándares, marcos de referencia y publicaciones especializadas, NIST proporciona a las organizaciones las herramientas necesarias para enfrentar los desafíos de la ciberseguridad en un entorno digital cada vez más complejo y dinámico. La adopción de los estándares del NIST no solo mejora la seguridad y la eficiencia operativa, sino que también facilita el cumplimiento normativo y fortalece la confianza de los stakeholders, posicionando a las organizaciones para un éxito sostenido en la era digital.

## 5.15. Servicios en la nube

Los servicios en la nube transformaron la manera en la que las aplicaciónes modernas se desarrollan, despliegan y mantienen. Por su escalabilidad y disponibilidad, estos servicios permiten que organizaciones de diferentes tamaños puedan implementar soluciones tecnológicas sin tener que invertir en infraestructura física.

El uso de servicios en la nube ofrece múltiples ventajas, entre ellas:

- **Alta disponibilidad y continuidad del servicio**, lo cual es crucial para sistemas en el área de salud.
- **Escalabilidad automática**, permitiendo ajustar recursos según la demanda sin interrupciones.
- **Seguridad integrada**, incluyendo cifrado en tránsito y en reposo, control de accesos y monitoreo continuo.
- **Reducción de costos operativos**, al eliminar la necesidad de mantener servidores físicos y personal técnico especializado en infraestructura.

Al integrar estos servicios, se garantiza una base sólida, confiable y segura para el manejo de la información médica, contribuyendo al cumplimiento de estándares como HIPAA y NIST, y facilitando la adopción tecnológica en instituciones de salud con recursos limitados.

## 5.16. Amazon Web Services (AWS)

Amazon Web Services (AWS) es una plataforma de servicios en la nube ofrecida por Amazon, que proporciona recursos tecnológicos bajo demanda a través de internet. Entre sus principales servicios se encuentran el cómputo, almacenamiento, redes, bases de datos, inteligencia artificial, entre otros. AWS se caracteriza por su escalabilidad, flexibilidad y seguridad, permitiendo a organizaciones de distintos tamaños implementar soluciones digitales sin necesidad de infraestructura física local.

## 5.17. Amazon RDS

Dentro de sus servicios de bases de datos, destaca **Amazon Relational Database Service (RDS)**, un sistema gestionado para bases de datos relacionales. RDS permite configurar, operar y escalar bases de datos como PostgreSQL, MySQL, MariaDB, Oracle y SQL Server, sin necesidad de encargarse directamente del mantenimiento del hardware o del software subyacente. Este servicio automatiza tareas administrativas como la creación de copias de seguridad, la replicación entre zonas de disponibilidad, la recuperación ante fallos y la aplicación de parches de seguridad, brindando así un entorno fiable y altamente disponible para el almacenamiento de datos estructurados [17].

## 5.18. Diseño de interfaz de usuario con Figma

La herramienda Figma permite diseñar una interfaz visual clara, ordenada y centrada en el usuario. Esta herrmaienta permitela creación de prototipos, interfaces de usuario y flujos de navegación. Su funcionalidad de trabajo en tiempo real facilita la retroalimentación y mejora la coherencia visual en el desarrollo del frontend [18].

## 5.19. Desarrollo del frontend con React

React es una biblioteca de JavaScript desarrollada por Meta, la cual fue diseñada para construir interfaces de usuario basadas en componentes reutilizables. Su arquitectura permite el desarrollo eficiente de aplicaciones web dinámicas e interactivas [19].

## 5.20. Gestión de autenticación y control de acceso

La autenticación es el procesos en el cual un sistema valida la identidad del usuario y es uno de los elementos fundamentales en cualquier sistema que maneje información sensible. Existen diversos estándares y mecanismos para llevar a cabo este proceso, como SAML (Security Assertion Markup Language), LDAP (Lightweight Directory Access Protocol), OAuth 2.0, y JWT (JSON Web Token), cada uno con ventajas específicas dependiendo del entorno y la arquitectura del sistema.

JWT es un estándar que permite transmitir información entre partes de manera segura. Cuando el usuario se autentica exitosamente, el servidor genera un token JWT que contiene información codificada y lo devuelve al cliente. Este token se envia cada vez que se hace una solicitud y así permite que se tenga una sesión sin necesidad de almacenar el estado en el servidor [20].

A diferencia de mecanismos como SAML, que está orientado a entornos empresariales con servicios federados [21], o LDAP, que se usa para autenticación centralizada en redes internas[22], JWT es ideal para aplicaciones modernas distribuidas basadas en API. JWT ofrece una implementación más sencilla en contextos donde no se requiere delegación de permisos a terceros, sino autenticación directa del usuario con el sistema.

## 5.21. Backend y lógica del servidor

El backend es la parte de la aplicación que se encarga del procesamiento interno, comunicación con bases de datos, autenticación y respuesta a las solicitudes del cliente. El backend se ocupa de ejecutar operaciones que no son visibles directamente [23].

Usualmente, este se desarrolla usando lenguajes como JavaScript (Node.js), Python, Java o PHP y se usan frameworks de apoyo como Express o Django. Para que un backend esté

bien diseñado, debe ser modular, escalable y seguro, ya que es una base importante de la lógica y funcionamiento de una aplicación moderna.

## 5.22. Pruebas de penetración tipo caja negra

Las pruebas de tipo caja negra (black box) consisten en simular ataques desde el punto de vista de un usuario que no posee conocimiento previo de la estructura interna del sistema. Este tipo de prueba se enfoca en evaluar el sistema desde su interfaz pública, tal como lo haría un atacante real que busca encontrar vulnerabilidades sin acceso al código fuente ni a la configuración del servidor [24]. Las herramientas comúnmente utilizadas para este tipo de pruebas incluyen OWASP ZAP, Burp Suite, Nikto y Nmap.

Las pruebas de caja negra son esenciales dentro de una estrategia de defensa ya que permiten descubrir fallos que no se detectan con un análisis estatico o revisiones de código.

## 5.23. Pruebas de penetración tipo caja gris

En las pruebas de tipo caja gris (gray box), el evaluador dispone de un conocimiento parcial del sistema, como credenciales de usuario o documentación funcional, pero no del código fuente completo. Este enfoque permite simular ataques desde el punto de vista de un usuario autenticado o un usuario con acceso limitado, lo cual es útil para evaluar controles de acceso y rutas protegidas. Según Moyo et al. **moyo2020pentesting**, las pruebas gray box ofrecen un equilibrio entre realismo y profundidad, al permitir descubrir vulnerabilidades tanto externas como internas sin requerir acceso completo al sistema.

## 5.24. Pruebas de penetración tipo caja blanca

Las pruebas de penetraciónde tipo caja blanca (white box) se caracterizan por el acceso total del evaluador al sistema, incluyendo el código fuente, estructuras de base de datos, configuraciones del servidor y lógica de negocio. Este tipo de análisis es útil para identificar vulnerabilidades profundas y errores lógicos que no son visibles desde el exterior. Como señalan Engebretson **engebretson2013metasploit**, el white box testing permite una evaluación exhaustiva de seguridad, ya que abarca todas las capas de la aplicación, desde el backend hasta la interfaz de usuario.

## 5.25. Análisis estático de código

El análisis estático de código es una técnica que ayuda a evaluar la calidad y seguridad del software sin tener que ejecutarlo. Por medio de esta técnica, el código fuente se revisa para identificar errores, vulnerabilidades, duplicaciones, malas prácticas a la hora de programar y oportunidades de mejora en la mantenibilidad [25].

Para llevar a cabo este análisis, existen herramientas especializadas como SonarQube, ESLint, PMD, y FindBugs. Estas herramientas automatizan el proceso y generan reportes detallados que permiten tomar decisiones informadas para mejorar el código.

Este tipo de análisis es especialmente útil en etapas tempranas del desarrollo, ya que permite detectar fallos antes de que el software sea desplegado. Entre los aspectos más evaluados se encuentran: la presencia de código inseguro, funciones obsoletas, condiciones lógicas inconsistentes, y cobertura de pruebas unitarias.

## 5.26. Evaluación de usabilidad con el método SUS

Es necesario evaluar la usabilidad de la aplicación en donde el usuario puede tener distintos niveles de experiencia técnica. Para evaluar la percepción de usabilidad de la aplicación desarrollada, se empleó el método **System Usability Scale (SUS)**, una herramienta estandarizada ampliamente utilizada en la evaluación de interfaces de usuario [26].

SUS consiste en un cuestionario de 10 afirmaciones que los participantes responden en una escala de tipo Likert de 5 puntos, que varía desde "totalmente en desacuerdo"hasta "totalmente de acuerdo". Al finalizar, las puntuaciones se procesan y normalizan para obtener un puntaje total que va de 0 a 100, donde valores superiores a 68 suelen considerarse como buena usabilidad.

Este método destaca por su simplicidad, rapidez y fiabilidad, lo cual lo convierte en una opción ideal para pruebas con usuarios reales en fases intermedias o finales del desarrollo. En particular, SUS permite cuantificar la experiencia subjetiva de los usuarios en torno a aspectos como facilidad de uso, consistencia, aprendizaje y confianza al utilizar el sistema.

# CAPÍTULO 6

## Alcance

Este trabajo comprende el desarrollo de una plataforma web para el manejo de información médica, dirigida a clínicas médicas. La solución permite el registro y autenticación de usuarios segmentados por clínica así como la gestión de pacientes, visualización de datos clínicos básicos y resguardo de dicha información en una base de datos alojada en la nube mediante Amazon RDS.

El proyecto abarca el desarrollo del frontend con React así como el backend con Node.js, incluyendo la implementación de un sistema de autenticación con JSON Web Tokens (JWT). Se incorporaron buenas prácticas de desarrollo seguro y se integraron frameworks de seguridad como HIPAA, NIST, CSA, CMM y SLSA para así garantizar la protección de la información de los pacientes en la aplicación.

Además, se incluye la realización de pruebas de seguridad con las herramientas SonarQube y OWASP ZAP y la evaluación de usabilidad con el método System Usability Scale (SUS). Todo esto se enmarca en una arquitectura modular, escalable y diseñada para facilitar su adopción en entornos reales.

Quedan fuera del alcance del presente trabajo funcionalidades avanzadas como interoperabilidad con sistemas hospitalarios existentes, procesamiento de imágenes médicas, almacenamiento de archivos clínicos o integración con servicios de terceros, aunque podrían considerarse en etapas futuras.

CAPÍTULO 7

# Metodología

## 7.1. Metodología

La página web fue desarrollada enfocándose en la seguridad de acceso (IAM) y manejo de la información para asegurarse que los accesos estén correctamente autenticados y autorizados, evitando así el uso no autorizado de los recursos del sistema. Para ello, se implementaron módulos de autenticación y registro de usuarios, los cuales permiten que se controle el acceso a la clínica que les corresponde.

El desarrollo de la página contiene una arquitectura cliente-servidor el cual da prioridad a la seguridad, escalabilidad y correcta segmentación de la información.

### 7.1.1. Formulario de análisis de necesidades clínicas

Como parte del proceso de diseño centrado en el usuario, se implementó un formulario dirigido a profesionales del área de salud, con el fin de recopilar información clave sobre sus necesidades informativas al atender pacientes. Este formulario permite comprender qué tipo de datos deben estar disponibles en distintos contextos clínicos, tanto antes, durante y después de una consulta médica.

El formulario incluía las siguientes preguntas:

- **Área de especialización:** para identificar el campo médico del profesional y poder contextualizar sus respuestas.
- **Al estar atendiendo a un paciente, ¿qué información considera necesaria tener a mano?** Esta pregunta busca establecer cuáles son los datos prioritarios que deben mostrarse de forma inmediata durante la atención (como antecedentes, historial clínico, alergias o resultados recientes).

- **Si desea visualizar la información de su paciente luego de la cita, ¿qué información quisiera tener disponible?** El objetivo de esta sección es comprender qué elementos son relevantes para un seguimiento posterior, como evolución del tratamiento, prescripciones o notas de la consulta.
- **¿Qué otra información considera necesaria tener disponible de su paciente fuera de una consulta?** Esta última pregunta permite detectar necesidades adicionales como datos de contacto, seguimiento a largo plazo, o alertas importantes, que deben ser accesibles incluso fuera del entorno clínico inmediato.

Las respuestas recopiladas fueron consideradas para definir qué campos y módulos debían estar presentes en el sistema, permitiendo así construir una plataforma la cual estos profesionales pudieran usar.

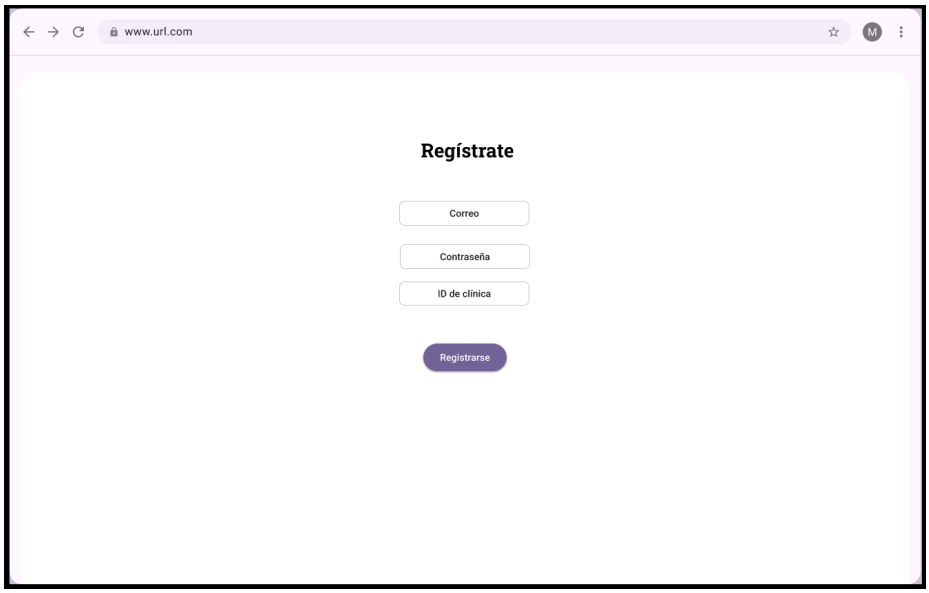
### 7.1.2. Estructura y diseño de las vistas principales

La aplicación fue estructurada en distintas vistas principales que guían al usuario desde el acceso hasta la gestión de información clínica. Estas pantallas fueron diseñadas inicialmente en Figma y posteriormente implementadas en React.

#### Página de registro

La vista de registro permite a nuevos usuarios crear una cuenta ingresando tres campos principales: correo electrónico, contraseña y el ID de la clínica a la que pertenecen. Este identificador es esencial para asegurar que cada profesional acceda únicamente a la información correspondiente a su clínica. La interfaz valida los campos antes de enviarlos al backend, y notifica cualquier error de manera clara al usuario.

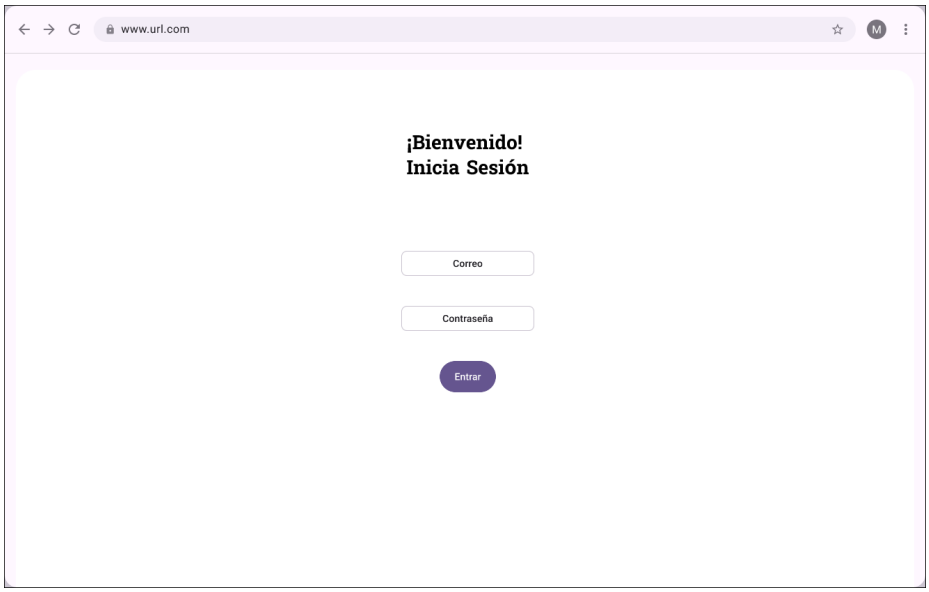
Figura 1: Vista de registro donde se solicita correo, contraseña e ID de clínica



### Página de inicio de sesión (login)

La vista del inicio de sesión está destinada a usuarios ya registrados. En ella se solicita el correo y la contraseña, los cuales son verificados en el servidor. En caso de éxito, se retorna un token JWT que es almacenado en el cliente para autorizar futuras acciones. En caso de error, se informa al usuario.

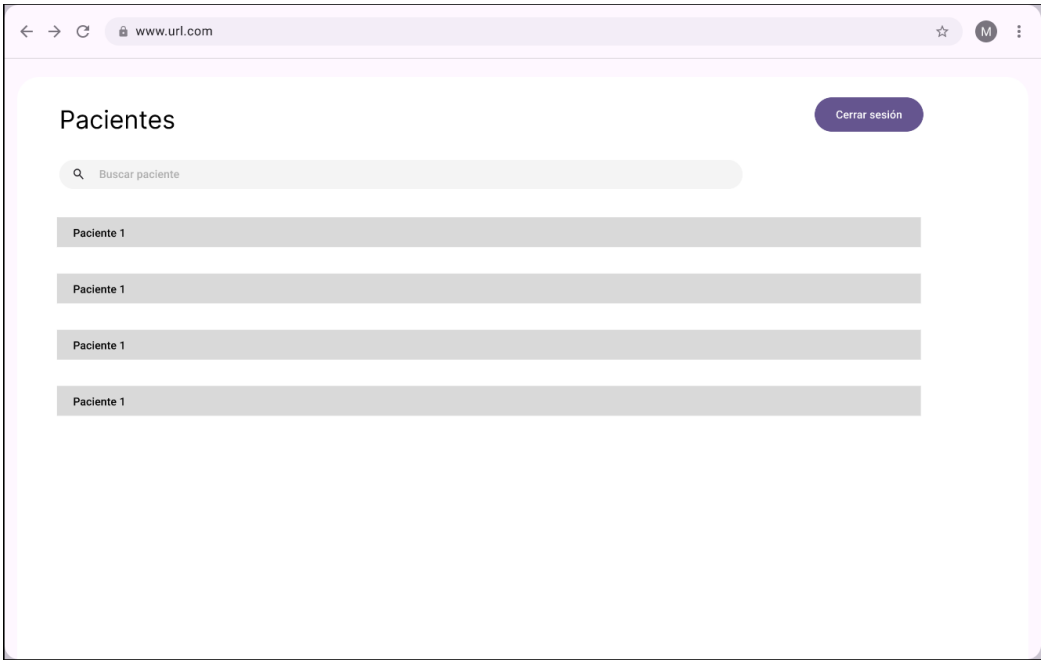
Figura 2: Vista de inicio de sesión con validación de credenciales




### Dashboard

Una vez autenticado, el usuario es redirigido al *dashboard*, donde se muestra una vista general de la plataforma. Desde este panel, el usuario puede navegar hacia distintas secciones, como el listado de pacientes o el gestor de archivos. El diseño está pensado para mostrar la información de forma clara y accesible, priorizando la eficiencia en el acceso a datos relevantes.

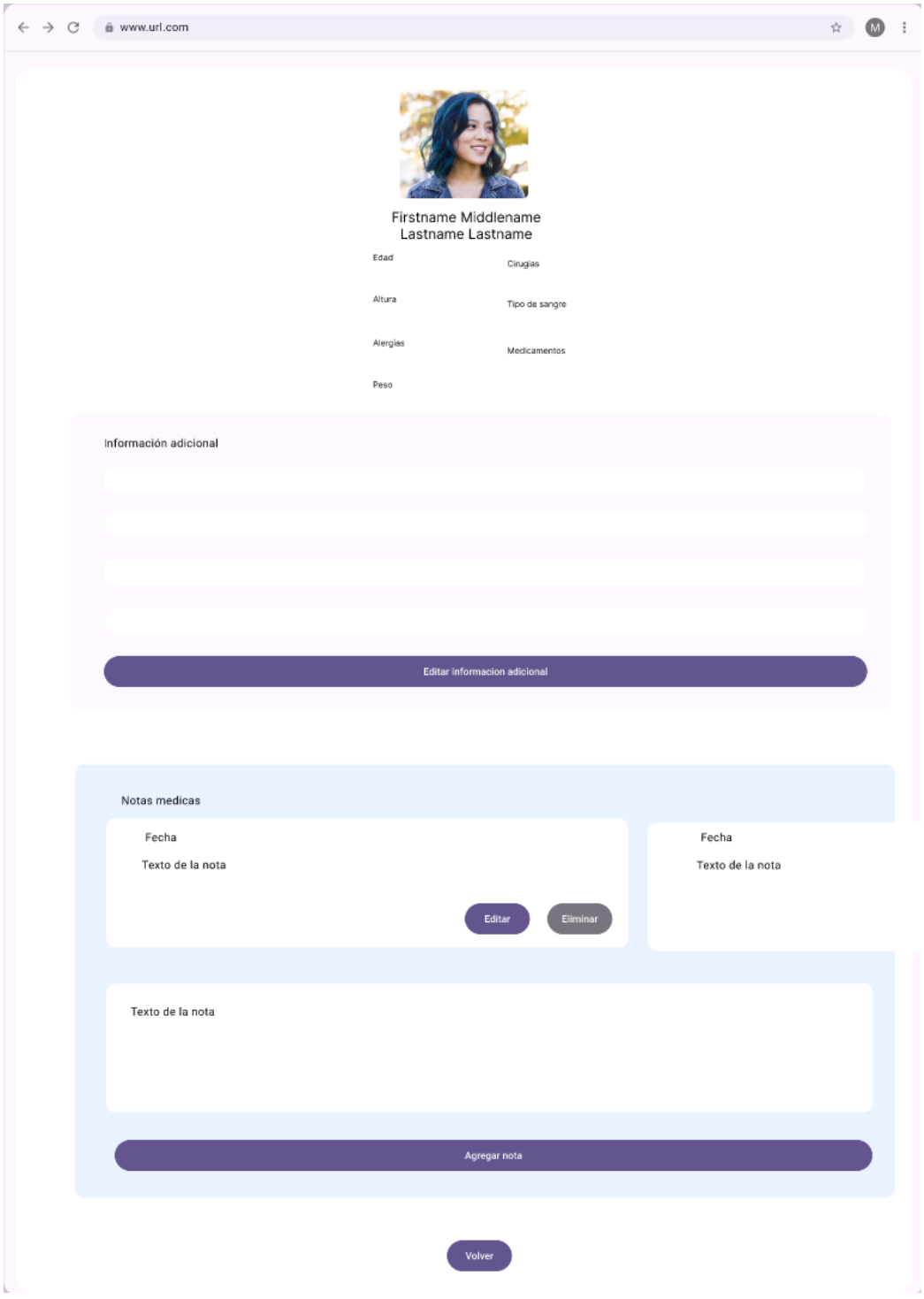
Figura 3: Dashboard principal luego del inicio de sesión




### Vista de pacientes

En la sección de pacientes se presenta un listado filtrado según el ID de clínica del usuario. Desde esta vista es posible consultar información básica del paciente, y acceder a su ficha clínica completa si fuera necesario. Esta funcionalidad está diseñada para facilitar tanto la consulta en tiempo real como el seguimiento posterior a las citas.

Figura 4: Vista con el listado de pacientes filtrados por clínica



### 7.1.3. Frontend y experiencia de usuario

El frontend fue desarrollado con **React**. Desde esta interfaz, los usuarios pueden registrarse, iniciar sesión y acceder a vistas personalizadas según su clínica dependiendo el ID que tengan asignado a su usuario.

Las solicitudes al backend se realizan mediante la API `fetch`, y los datos se muestran al usuario asegurando que sean las correctas.

### 7.1.4. Autenticación y manejo de sesiones

Para la gestión segura de usuarios, se implementó un sistema de autenticación basado en **JSON Web Tokens (JWT)**. Al iniciar sesión correctamente, el servidor genera un token que incluye información codificada sobre el usuario, como su correo electrónico. Esta información se codifica para garantizar su integridad y evitar manipulaciones del token por parte de terceros.

Este token es almacenado en el frontend utilizando `localStorage`, y se incluye en cada solicitud posterior mediante el encabezado `Authorization`. De esta manera el servidor puede validar la identidad del usuario sin necesidad de mantener sesiones activas lo cual es ideal para este tipo de aplicación web. Se garantiza que los usuarios autenticados puedan acceder a rutas protegidas, cómo las que los permite ver la información de sus pacientes. Cuando se recibe una solicitud, el servidor extrae y verifica el token antes de autorizar el acceso para poder asegurar la integridad del flujo de autenticación. El uso de `localStorage` permite un control explícito sobre cuándo y cómo se envían los tokens, evitando el envío automático en cada solicitud como ocurre con las cookies.

Es importante mencionar que `localStorage` es vulnerable si no se aplican otras medidas de seguridad por lo que el uso de las cookies con las banderas `HttpOnly` y `Secure` son las opciones principales. En este caso no se aplicaron porque no son accesibles mediante JavaScript [27]. Por esta razón, en esta implementación se adoptaron controles compensatorios como encabezados HTTP seguros y políticas de CORS restrictivas.

De esta manera, se garantiza que solo los usuarios autenticados puedan acceder a rutas protegidas, como aquellas que permiten visualizar la información de los pacientes usando Control de Acceso por Atributo (ABAC simplificado). Cuando se recibe una solicitud, el servidor extrae y verifica el token antes de autorizar el acceso, asegurando la integridad y validez del flujo de autenticación.

### 7.1.5. Segmentación por clínica

Durante el registro, se solicita al usuario un **ID de clínica**, el cual se almacena junto con su información personal. Este identificador se utiliza para filtrar automáticamente los datos mostrados en la aplicación, permitiendo que cada clínica acceda únicamente a sus propios pacientes. Esta funcionalidad fue implementada tanto en el backend (mediante consultas filtradas en PostgreSQL) como en el frontend, mostrando los datos correspondientes a cada sesión activa.

### 7.1.6. Backend y lógica del servidor

El backend fue desarrollado con **Node.js** utilizando el framework **Express** para la construcción de una API. Este servidor se encarga de gestionar las operaciones relacionadas con el registro, autenticación y consulta de datos, así como de servir de intermediario entre el frontend y la base de datos.

Cada solicitud enviada desde el cliente es procesada en rutas específicas, las cuales validan los datos, aplican lógica de negocio y se comunican con la base de datos. Además, se incorporó la biblioteca `bcrypt` para el cifrado de contraseñas, garantizando así que la información sensible no se almacene en texto plano.

### 7.1.7. Base de datos y almacenamiento en AWS RDS

Los datos de la aplicación se almacenan en una base de datos **PostgreSQL** desplegada en **Amazon Web Services (AWS RDS)**, lo cual permite una alta disponibilidad, recuperación ante desastres y cumplimiento con estándares de seguridad.

La base de datos contiene las tablas necesarias para manejar la información de los usuarios y sus respectivos pacientes, con una estructura que permite segmentar la información según el *ID de clínica*. Este diseño facilita que cada usuario acceda únicamente a los datos correspondientes a su clínica, cumpliendo así con principios de aislamiento y confidencialidad.

### 7.1.8. Pruebas de seguridad para asegurar la aplicación

Tener la base de datos en una nube, como AWS RDS, que protege la información no es sufuicente. Para asegurar la aplicación, es necesario proteger el código fuente, tanto en el funcionamiento como en la resistencia a posibles vulnerabilidades.

Para ello, se realizaron pruebas utilizando dos herramientas **SonarQube** y **OWASP ZAP**. SonarQube permite realizar un análisis estático del código, identificando vulnerabilidades, errores de seguridad, y malas prácticas de programación antes de que el software sea desplegado. Esto ayuda a fortalecer la calidad del código fuente y mitigar riesgos desde las etapas tempranas del desarrollo.

OWASP ZAP (Zed Attack Proxy) se utilizó para llevar a cabo pruebas dinámicas de seguridad en la aplicación desplegada. Esta herramienta simula ataques como inyecciones SQL, exposición de datos o fallos en la autenticación. Con el escaneo activo que provee, es posible detectar los puntos donde se comprometen la confidencialidad o integridad de la información.

Al combinar ambas herramientas, se puede alcanzar una validación estática y dinámica para garantizar que la plicaión cumpla con estándares modernos de seguridad y que los datos de los usuarios se mantengan protegidos frente a amenazas comunes.

CAPÍTULO 8

# Resultados

## 8.1. Resultados

Los resultados para el desarrollo se separan en dos: pruebas con usuarios y pruebas de seguridad en la aplicación.

### 8.1.1. Interfaz final

A continuación se presentan capturas de pantalla que ilustran el funcionamiento de la aplicación desarrollada:

Figura 5: Página de registro en la aplicación web desarrollada

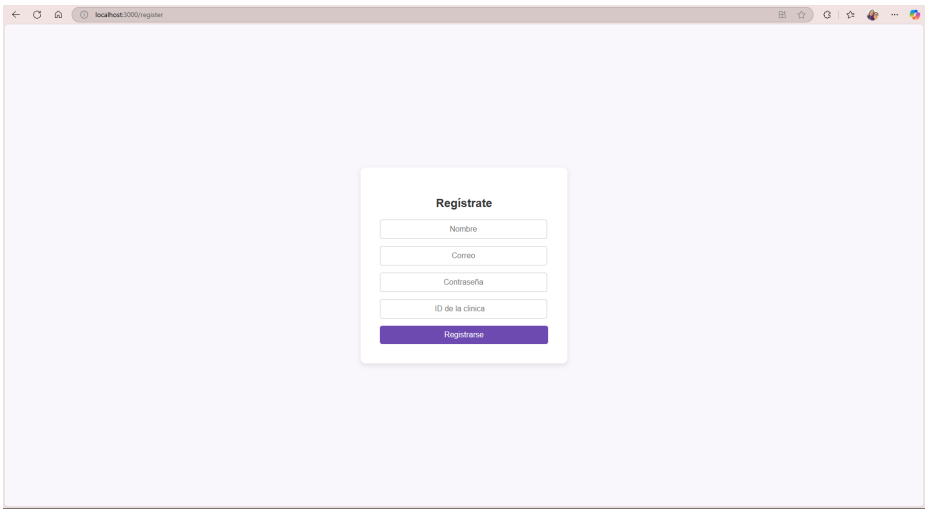


Figura 6: Página de inicio de sesión en la aplicación web desarrollada

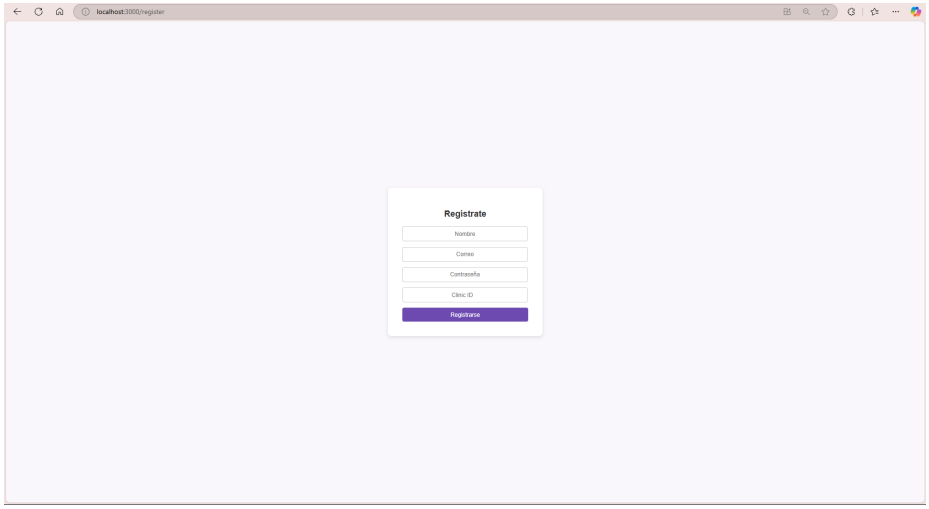



Figura 7: Dashboard en la aplicación web desarrollada.

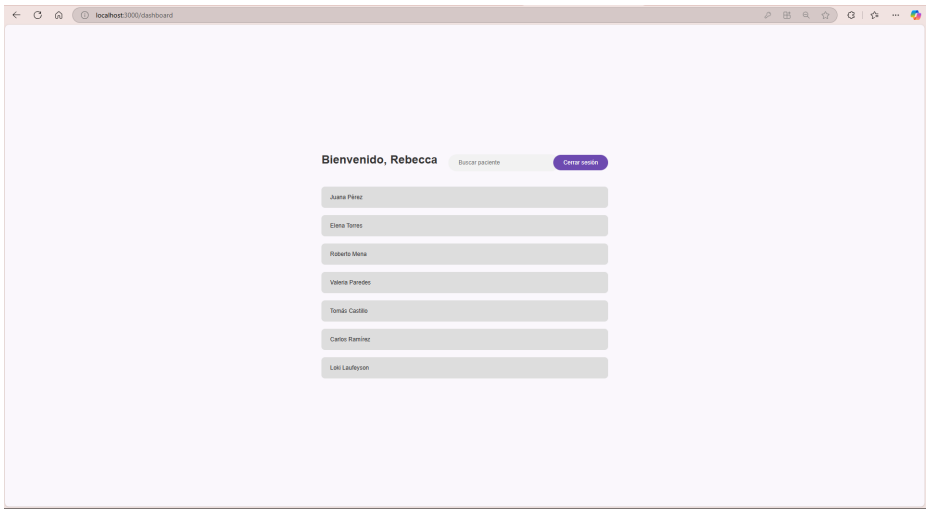


Figura 8: Página de paciente individual en la aplicación web desarrollada

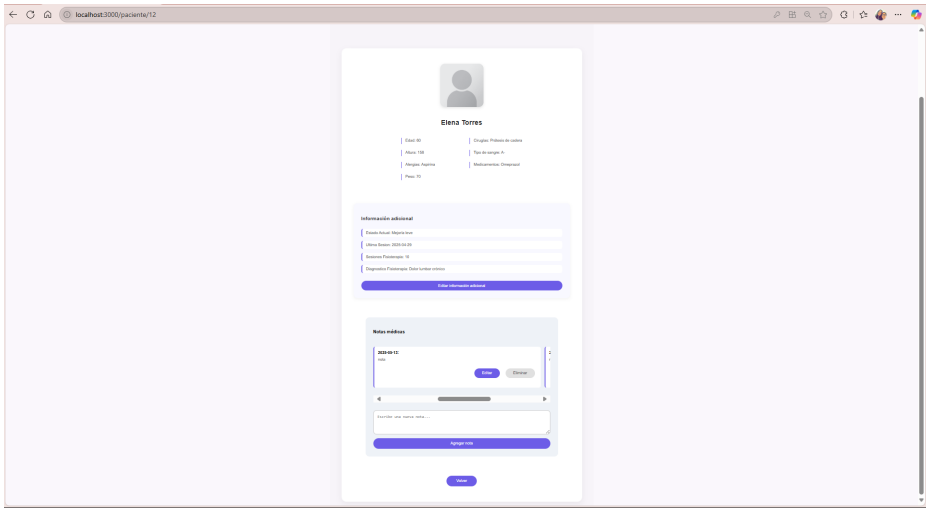
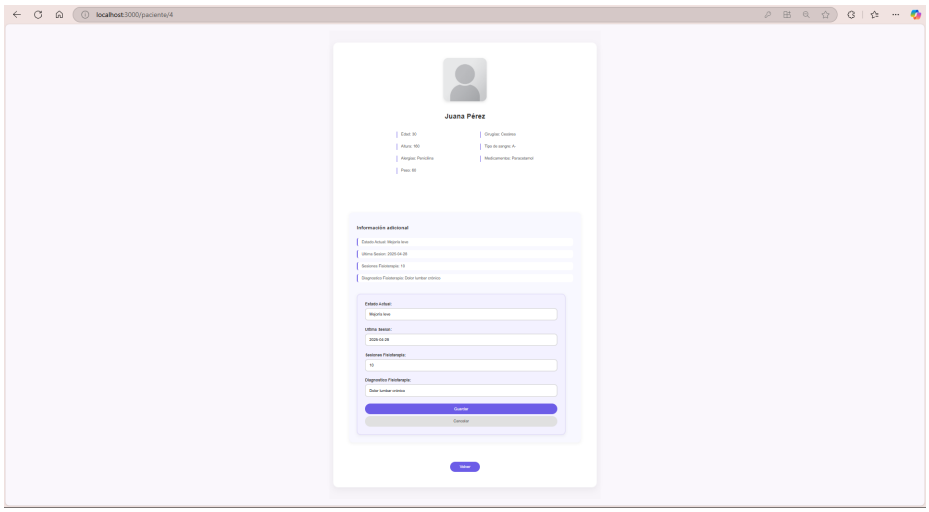



Figura 9: Página de paciente individual en la aplicación web desarrollada a la hora de editar la información




### 8.1.2. Pruebas con usuarios

A los usuarios se les presentó la aplicación, permitiéndoles interactuar de manera autónoma, sin intervención externa. Para evaluar su experiencia de uso, se aplicó el método System Usability Scale (SUS), el cual consiste en una serie de preguntas estructuradas que recogen la percepción del usuario sobre distintos aspectos de la aplicación, tales como facilidad de uso, eficiencia y satisfacción general. Aunque la muestra es reducida, esto es aceptable para estudios de usabilidad exploratoria. Según Brooke [26], el creador del SUS, una muestra mínima de cinco participantes puede ofrecer resultados estadísticamente confiables para detectar problemas de usabilidad y obtener una primera aproximación del desempeño percibido de una interfaz.

**Resultados de las pruebas de usabilidad (SUS)**

Se aplicó el cuestionario a cinco usuarios que trabajan en el área de salud. Estos usuarios trabajan en diferentes obteniendo los siguientes resultados:

- **Puntaje promedio:** 98.5
- **Calificación promedio:** A

Figura 10: Resultado de las pruebas SUS hecha a usuarios

| Average SCORE | Average GRADE |
|---|---|
| 98.5 | A |

| | QUESTION 1 | QUESTION 2 | QUESTION 3 | QUESTION 4 | QUESTION 5 | QUESTION 6 | QUESTION 7 | QUESTION 8 | QUESTION 9 | QUESTION 10 | SUS Score | SUS Grade |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Usaría esta página web de nuevo | La página web era compleja | Esta página web era fácil de navegar | Necesito la ayuda de una persona tecnológica para usar esta página web | Las funciones de la página web estaban bien integradas | La página web era muy inconsistente | La mayoría de las personas aprendería a usar la página web rapidamente | La página web era muy incómoda de usar | Me sentí comodo usando la página web | Tengo que aprender muchas cosas para usar la página web | | |
| **Participant 1** | 5 | 0 | 5 | 0 | 5 | 0 | 5 | 5 | 5 | 0 | 100 | A |
| **Participant 2** | 5 | 0 | 5 | 0 | 4 | 0 | 5 | 5 | 5 | 0 | 97.5 | A |
| **Participant 3** | 5 | 0 | 5 | 0 | 5 | 0 | 5 | 5 | 5 | 0 | 100 | A |
| **Participant 4** | 5 | 0 | 5 | 0 | 5 | 0 | 5 | 5 | 5 | 0 | 100 | A |
| **Participant 5** | 5 | 0 | 5 | 0 | 4 | 0 | 5 | 5 | 4 | 0 | 95 | A |

Cada uno de los participantes alcanzó una calificación individual dentro del rango A (de 95 a 100 puntos), lo cual refleja un alto nivel de satisfacción con la aplicación. Los resultados indican que los usuarios percibieron la aplicación como:

- Fácil de usar.
- Bien integrada en sus funciones.
- Con bajo nivel de complejidad innecesaria.
- Cómoda y accesible para un uso rápido sin necesidad de asistencia técnica.

Los resultados demuestran que según los estándares de interpretación se considera una calificación de excelencia. Este tipo de puntuaciones suelen estar asociadas a sistemas con interfaces intuitivas, bajo nivel de complejidad, y un diseño visual y funcional bien integrado [26].

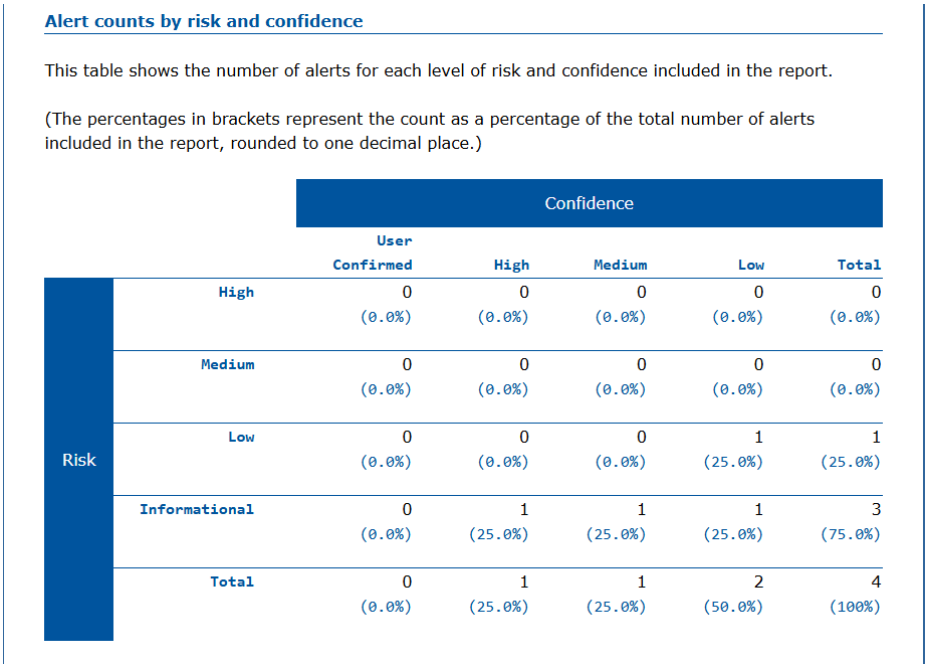
Los participantes demuestran que la aplicación es fácil de usar, tiene funciones coherentes y no les genera confución, además, es accesible sin necesidad de asistencia técnica. Esto nos dice que la aplicaión se percibe como altamente usable, eficiente y agradable para el usuario final [28].

### 8.1.3. Pruebas de seguridad

### 8.1.4. Análisis de seguridad con OWASP ZAP

Se realizó un escaneo de seguridad automatizado utilizando **OWASP ZAP** sobre la versión compilada de la aplicación desplegada en entorno local.

Figura 11: Resultado de la prueba con OWASP ZAP



**Alert counts by risk and confidence**

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

| | | Confidence | | | | |
|---|---|---|---|---|---|---|
| | | User Confirmed | High | Medium | Low | Total |
| Risk | High | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) |
| | Medium | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) |
| | Low | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 1 (25.0%) | 1 (25.0%) |
| | Informational | 0 (0.0%) | 1 (25.0%) | 1 (25.0%) | 1 (25.0%) | 3 (75.0%) |
| | Total | 0 (0.0%) | 1 (25.0%) | 1 (25.0%) | 2 (50.0%) | 4 (100%) |

### Alertas pendientes

- **Timestamp Disclosure - Unix (Riesgo Bajo):** se detectaron encabezados o campos que revelan marcas de tiempo en formato Unix, tales como fechas de creación o modificación de recursos. Sin embargo, esta información no incluye datos sensibles ni contribuye directamente a la explotación de vulnerabilidades. Desde una perspectiva de análisis de riesgo, la divulgación de timestamps se clasifica como un riesgo bajo ya que su aprovechamiento requiere combinarse con otras debilidades, como errores de lógica o accesos no controlados. Por lo tanto, en el contexto actual de la aplicación, no representa una amenaza directa a la confidencialidad, integridad o disponibilidad del sistema.
- **Authentication Request Identified (Informativo):** se identificó una ruta que solicita credenciales (`/api/login`), lo cual es esperado en una aplicación que incluye autenticación de usuarios.
- **Suspicious Comments (Informativo):** se encontraron comentarios en archivos enviados al cliente. Ya que no contienen información sensible, no es necesario eliminarlos.
- **Modern Web Application (Informativo):** OwaspZap identificó la aplicación como moderna basada en JavaScript (React). Esta observación no representa un riesgo, sino una característica.

Todas las alertas detectadas en esta fase corresponden a riesgos bajos o informativos. Ninguna representa una vulnerabilidad crítica o explotable en el entorno actual. Se considera que la aplicación está en cumplimiento con buenas prácticas de seguridad para su fase actual

de desarrollo, y se han documentado los hallazgos restantes como parte del proceso de mejora continua.

### Resultados del análisis de calidad de código (SonarQube)

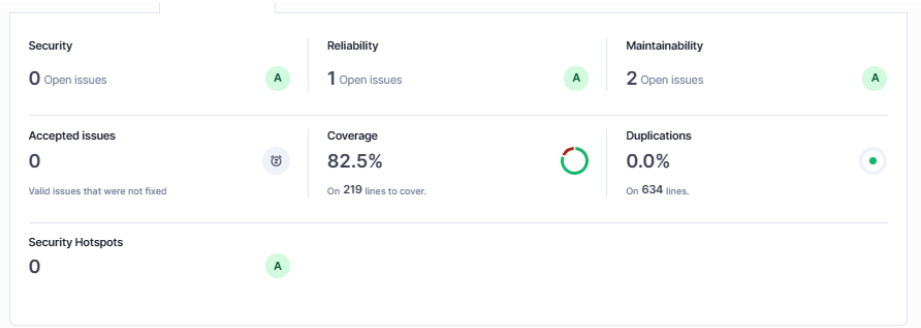
Para evaluar la calidad del código fuente de la aplicación, se utilizó la herramienta **SonarQube**, que permite identificar problemas relacionados con seguridad, mantenibilidad, fiabilidad y cobertura de pruebas.

El análisis arrojó los siguientes resultados:

- **Seguridad:** sin problemas detectados (calificación A).
- **Fiabilidad:** un único problema leve detectado (calificación A).
- **Mantenibilidad:** dos problemas menores detectados (calificación A).
- **Cobertura de pruebas unitarias:** 82.5 % de las líneas de código cubiertas.
- **Duplicación de código:** 0 % de duplicación en 634 líneas analizadas.
- **Hotspots de seguridad:** no se detectaron puntos críticos de seguridad.

- **Puntaje promedio:** 98.5
- **Calificación promedio:** A

Figura 12: Resultado de SonarQube.




Estos resultados reflejan que la calidad del código es bastante alta. Gracias a la ausencia de problemas de seguridad y baja duplicación indica que el proyecto ha seguido buenas prácticas de programación segura y eficiente. Además, la cobertura de las pruebas unitarias muestra la garantía, robustéz y confiabilidad del software por medio de pruebas automatizadas.

# CAPÍTULO 9

## Conclusiones

- El cumplimiento de los objetivos de seguridad, usabilidad y despliegue en la nube se logró por medio de la integración de buenas prácticas de desarrollo y el uso de frameworks de seguridad. La alta puntuación obtenida en la evaluación SUS, refleja la facilidad de uso de la plataforma y el enfoque centrado en el usuario aplicado desde la fase de diseño.
- La arquitectura implementada demostró ser robusta y segura, ya que los resultados de las pruebas de seguridad respaldan su eficacia. El uso de SonarQube permitió detectar y corregir problemas en el código, mientras que OWASP ZAP no identificó vulnerabilidades críticas. Estos hallazgos demuestran que la integración de mecanismos como autenticación con JWT, encabezados seguros y políticas de control de acceso contribuyen significativamente a mitigar riesgos.
- Se comprobó que una plataforma segmentada por clínicas, basada en buenas prácticas de desarrollo seguro y desplegada en un entorno en la nube, puede ofrecer una solución eficiente y viable para el manejo de información médica, especialmente en contextos con recursos limitados. Esto se debe a que la segmentación por clínica permitió una organización clara y controlada de los datos, mientras que la nube redujo la necesidad de infraestructura local. Asimismo, la incorporación de principios de seguridad y autenticación adecuados permitió proteger los datos sin necesidad de sistemas complejos o costosos, lo que ayuda a aplicar este tipo de solución en entornos con limitaciones técnicas y presupuestarias.

# CAPÍTULO 10

## Recomendaciones

- Se recomienda ampliar el tamaño y la diversidad de la muestra de usuarios en futuras evaluaciones, para obtener resultados más representativos y estadísticamente sólidos. Una muestra más grande permitiría confirmar tendencias observadas y aumentar la confianza en los niveles de usabilidad percibidos.
- Se recomienda diseñar la plataforma de forma que requiera una capacitación mínima, especialmente en entornos donde los usuarios no cuentan con formación técnica. Esto responde al resultado obtenido con el método SUS, donde una puntuación de 98.5 sugiere que el sistema es altamente comprensible incluso sin asistencia.
- Se recomienda utilizar hooks personalizados y patrones reutilizables en React para mejorar la consistencia, reducir redundancia y facilitar la mantenibilidad del código a largo plazo.
- Se recomienda dividir de manera lógica los contextos y manejadores de estado en React, siguiendo el principio de separación de responsabilidades, lo cual mejora la escalabilidad del sistema y su claridad estructural.
- Se recomienda realizar una comparación más exhaustiva entre proveedores de infraestructura, considerando métricas de rendimiento, costos, escalabilidad y disponibilidad. Esta comparación debe basarse en indicadores concretos, como tiempos de respuesta, uptime garantizado y latencia bajo carga.
- Se recomienda evaluar el uso de infraestructura en la nube como alternativa viable en proyectos con restricciones presupuestarias o recursos técnicos limitados, ya que permite escalar sin inversión inicial en servidores físicos.
- Se recomienda mantener actualizados los frameworks, bibliotecas y lenguajes de desarrollo para prevenir vulnerabilidades y asegurar compatibilidad con herramientas modernas.
- Se recomienda programar auditorías internas y externas de seguridad en ciclos periódicos, incluyendo análisis estático, dinámico y revisión manual, para identificar y corregir vulnerabilidades antes de que puedan ser explotadas.

- Se recomienda evaluar la aplicación con base en guías como el OWASP Testing Guide (OTG), aplicando correcciones según el nivel de riesgo y los recursos disponibles de la organización.
- Se recomienda documentar, clasificar y priorizar las vulnerabilidades detectadas en función de su severidad e impacto, estableciendo planes de remediación que equilibren riesgo, tiempo y costo.
- Se recomienda incluir funciones de exportación de datos clínicos en formatos estándar como PDF y CSV, para facilitar la interoperabilidad con otros sistemas y el respaldo de la información por parte del personal médico.
- Se recomienda implementar autenticación multifactor (MFA) en futuras versiones del sistema, con el objetivo de fortalecer la seguridad del acceso ante escenarios de suplantación de identidad.

# CAPÍTULO 11

## Bibliografía


[1] IBM, *Coste de la vulneración de datos 2023*, n.d. dirección: https://www.ibm.com/es-es/reports/databreach.

[2] IBM, *¿Qué es un ciberataque?* n.d. dirección: https://www.ibm.com/es-es/topics/cyber-attack.

[3] World Health Organization, *Ciberataques contra las infraestructuras de salud críticas*, n.d. dirección: https://www.who.int/es/news-room/questions-and-answers/item/cyber-attacks-on-critical-health-infrastructure.

[4] J. Fruhlinger, "HIPAA explained: definition, compliance, and violations," *CSO Online*, ene. de 2021. dirección: https://www.csoonline.com/article/570241/hipaa-explained-definition-compliance-and-violations.html.

[5] Office for Civil Rights, *Summary of the HIPAA Privacy Rule*, oct. de 2022. dirección: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html.

[6] Cloud Security Alliance, *Your Ultimate Guide to Security Frameworks*, abr. de 2024. dirección: https://cloudsecurityalliance.org/blog/2024/04/29/your-ultimate-guide-to-security-frameworks.

[7] Washington University in St. Louis, *Confidentiality, Integrity, and Availability: The CIA Triad*, n.d. dirección: https://informationsecurity.wustl.edu/items/confidentiality-integrity-and-availability-the-cia-triad/.

[8] SecurityScorecard, *Top 25 Cybersecurity Frameworks to Consider*, mayo de 2024. dirección: https://securityscorecard.com/blog/top-cybersecurity-frameworks-to-consider.

[9] NIST, *About NIST | NIST*, ene. de 2022. dirección: https://www.nist.gov/about-nist.

[10] National Institute of Standards and Technology, *Guide for Conducting Risk Assessments (NIST SP 800-30 Rev. 1)*, Consultado en 2025, 2012. dirección: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf.

[11] IBM, *¿Qué es el marco MITRE ATT&CK?* n.d. dirección: https://www.ibm.com/docs/es/detect/1.0.0?topic=framework-mitre-attck.

[12] H. Burde, *HITECH and HIPAA: A Survival Guide for the Modern Medical Practice*. American Medical Association, 2011.

[13] Cloud Security Alliance, *CSA STAR Program Overview*, Consultado en 2025, n.d. dirección: https://www.atlassian.com/trust/compliance/resources/csa.

[14] Cloud Security Alliance, *Cloud Controls Matrix (CCM)*, Consultado en 2025, n.d. dirección: https://cloudsecurityalliance.org/research/cloud-controls-matrix.

[15] H. Strohmier, G. Stoker, M. Vanajakumari, U. Clark, J. Cummings y M. Modaresnezhad, "Cybersecurity Maturity Model Certification Initial Impact on the Defense Industrial Base," *Journal of Information Systems Applied Research*, vol. 15, n.º 2, págs. 17-29, 2022. dirección: https://www.jisara.org/2022-15/n2/JISARv15n2p17.html.

[16] SLSA, *Supply-chain Levels for Software Artifacts*, n.d. dirección: https://slsa.dev/#:~:text=SLSA%20is%20a%20security%20framework.%20It%20is%20a,as%20possible%2C%20at%20any%20link%20in%20the%20chain.

[17] Amazon Web Services, *What is AWS?* 2025. dirección: https://aws.amazon.com/what-is-aws/.

[18] Figma, *Figma: Collaborative interface design tool*, Consultado en 2025, 2024. dirección: https://www.figma.com.

[19] Meta Platforms Inc., *React – A JavaScript library for building user interfaces*, Consultado en 2025, 2024. dirección: https://react.dev.

[20] Auth0, *Introduction to JSON Web Tokens*, Consultado en 2025, 2024. dirección: https://auth0.com/learn/json-web-tokens.

[21] OASIS Standard, *Security Assertion Markup Language (SAML) V2.0*, Consultado en 2025, 2024. dirección: https://wiki.oasis-open.org/security/.

[22] LDAP.com, *LDAP: The Lightweight Directory Access Protocol*, Consultado en 2025, 2024. dirección: https://ldap.com/.

[23] IBM Cloud Education, *What is back-end development?* Consultado en 2025, 2023. dirección: https://www.ibm.com/cloud/learn/back-end-development.

[24] OWASP Foundation, *Penetration Testing Methodologies - Black Box Testing*, Consultado en 2025, 2024. dirección: https://owasp.org/www-community/Testing.

[25] SonarSource, *SonarQube Documentation*, Consultado en 2025, 2024. dirección: https://docs.sonarsource.com/sonarqube/latest/.

[26] J. Brooke, "SUS: A "Quick and DirtyÜsability Scale," Digital Equipment Corporation, inf. téc., 1996, Consultado en 2025. dirección: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[27] OWASP Foundation, *HttpOnly*, Consultado en 2025, 2024. dirección: https://owasp.org/www-community/HttpOnly.

[28] A. Bangor, P. T. Kortum y J. T. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *Journal of Usability Studies*, vol. 4, n.º 3, págs. 114-123, 2009.