UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

"DESARROLLO DE SOFTWARE SEGURO MEDIANTE EL USO DE PATRONES DE DISEÑO Y VALIDACIÓN DE UN PROTOTIPO CON OWASP"

Trabajo de titulación previo a la obtención del título de Ingeniero en Software

Línea de investigación: Desarrollo, aplicación de software y cyber security (seguridad cibernética)

AUTOR:

Marlon Brandon Cachimuel Loyo

DIRECTOR:

Ing. Marco Remigio Pusdá Chulde, PhD

Ibarra - Ecuador 2025

AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1005007602		
APELLIDOS Y NOMBRES:	Cachimuel Loyo Marlon Brandon		
DIRECCIÓN:	San Miguel de Ibarra, Sagrario		
EMAIL:	mbcachimuell@utn.edu.ec / marloncachimuel@gmail.com		
TELÉFONO FIJO:		**TELF. MOVIL**	0990248155

DATOS DE LA OBRA	
TÍTULO:	Desarrollo de software seguro mediante el uso de patrones de diseño y validación de un prototipo con Owasp
AUTOR:	Cachimuel Loyo Marlon Brandon
FECHA	31/07/2025
CARRERA/PROGRAMA:	Pregrado
TITULO POR EL QUE OPTA:	Ingeniero en Software
DIRECTOR:	Ing. Marco.R Pusdá, PhD
ASESOR:	Msc. Daisy Imbaquingo, PhD

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 31 días, del mes de Julio de 2025

EL AUTOR:

Cachimuel Loyo Marlon Brandon
C.I: 1005007602

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 31 de Julio de 2025

Ing. Marco R. Pusdá, PhD
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

MARCO REMIGIO PUSDA CHULDE
Firmado digitalmente por MARCO REMIGIO PUSDA CHULDE
Fecha: 2025.07.31 15:26:44 -05'00'

(f) .....
Ing. Marco R. Pusdá, PhD
DIRECTOR DEL TRABAJO DE GRADO

DEDICATORIA

Dedico este trabajo de grado a mis padres, cuyo amor, esfuerzo y enseñanzas han sido el motor que me impulsó a seguir adelante incluso en los momentos más difíciles. A mi familia, por su constante apoyo emocional y por creer en mí cuando más lo necesité. También lo dedico a todas las personas que, de una u otra forma, hicieron parte de este proceso, brindándome palabras de aliento, comprensión y paciencia.
Y, sobre todo, a mí mismo, por no rendirme, por superar cada obstáculo y por confiar en que este proyecto, más allá de un logro académico, es una muestra de compromiso, perseverancia por lo que hago.

Cachimuel Loyo Marlon Brandon

AGRADECIMIENTO

Agradezco a la Universidad Técnica del Norte por brindarme la oportunidad de formarme como profesional y por ser el espacio donde crecí tanto académica como personalmente. Agradezco especialmente a mi tutor, Ing. Marco Pusdá, PhD por su guía, dedicación y confianza durante el desarrollo de este proyecto, así como por compartir su experiencia y conocimientos.

Extiendo también mi agradecimiento a los docentes de la carrera, quienes, con esfuerzo, paciencia han contribuido significativamente a mi formación.

A mi familia, por su apoyo incondicional en cada etapa de este proceso. Su confianza en mí, sus palabras de aliento y su amor han sido el sustento que me ha permitido continuar incluso cuando parecía difícil. Gracias por estar siempre.

Agradezco también a mis amigos, quienes me acompañaron en este camino con su amistad, colaboración y palabras de motivación. Compartir este proceso con ustedes ha sido invaluable y enriquecedor.

Cachimuel Loyo Marlon Brandon

RESUMEN

En el contexto actual las aplicaciones web tienen un problema que son las vulnerabilidades de seguridad que son una debilidad en donde los atacantes informáticos pueden hacer cualquier daño a las aplicaciones web. En este proyecto se centra en el desarrollo de una aplicación web segura mediante el uso de patrones de diseño en conjunto con el Top ten de OWASP. El principal objetivo es diseñar, desarrollar y evaluar software seguro para poder aliviar este problema de seguridad. El primer capítulo proporciona las bases ideales y necesarias para comprender el proyecto. Se afronta las vulnerabilidades en aplicaciones web, los patrones de diseño que se suelen usar para el desarrollo de software, el uso de las tecnologías como Express y React y se menciona el Top Ten de OWASP que se usa para evaluar las buenas prácticas de seguridad en la aplicación. En el segundo capítulo se inició con el levantamiento de los requisitos que se necesita para la el prototipo. Luego se procede a desarrollar la aplicación web que garantice las buenas prácticas de seguridad. Se realizo dos módulos principalmente: Usuarios y publicaciones cada uno con tareas específicas. En el tercer capítulo se evaluó la aplicación desarrollada según las buenas prácticas de seguridad el Top Ten de OWASP. Se realizo un análisis de la aplicación web donde determino las vulnerabilidades de seguridad del aplicativo y se realizó las correcciones necesarias para evitar estos problemas. Los resultados reflejan una mejora en la seguridad de la aplicación cumpliendo con las correcciones.

Palabras Clave: Owasp (Open Web Application Security Project)

ABSTRACT

In the current context web applications have a problem which is security vulnerabilities that are a weakness where computer attackers can do any damage to web applications. This project focuses on the development of a secure web application by using design patterns in conjunction with the OWASP Top Ten. The main objective is to design, develop and evaluate secure software in order to alleviate this security problem. The first chapter provides the ideal and necessary basis for understanding the project. It addresses vulnerabilities in web applications, design patterns that are commonly used for software development, the use of technologies such as Express and React and mentions the OWASP Top Ten, which is used to evaluate good security practices in the application. In the second chapter we started with the survey of the requirements needed for the prototype. Then we proceeded to develop the web application that guarantees the good security practices. Two main modules were developed: Users and publications, each with specific tasks. In the third chapter the developed application was evaluated according to the good security practices of the OWASP Top Ten. An analysis of the web application was carried out to determine the security vulnerabilities of the application and the necessary corrections were made to avoid these problems. The results reflect an improvement in the security of the application by complying with the corrections.

Keywords: Owasp (Open Web Application Security Project)

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Tema

Desarrollo de Software seguro mediante el uso de patrones de diseño y validación de un Prototipo con Owasp.

Planteamiento del problema

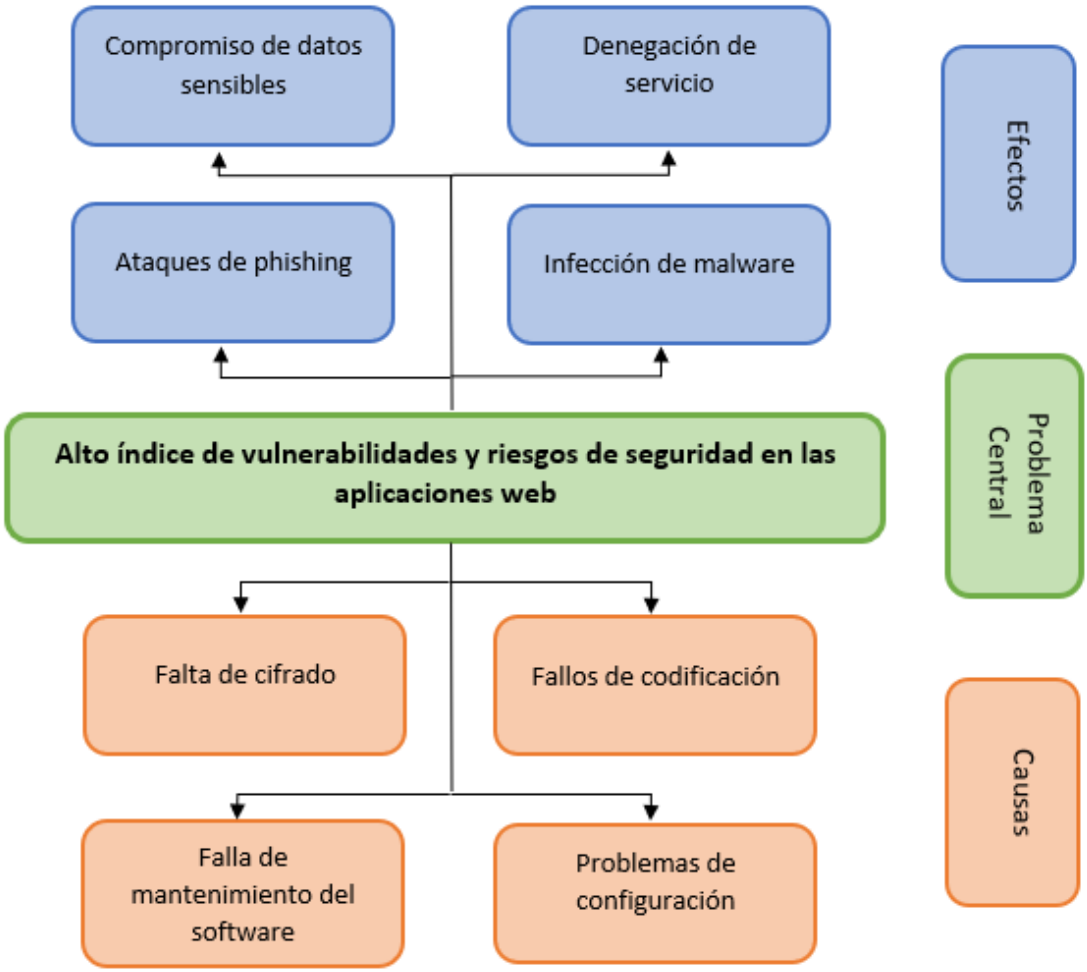
En la actualidad principal problema en las aplicaciones web son las vulnerabilidades. Estas pueden realizar cualquiera daño de distintas maneras en las aplicaciones web y con ello perjudicar el servicio y a los usuarios que la usan [1]. Entre los factores que pueden interferir con este problema es que no se utilicen guías o normas que se puedan seguir para mitigar estas problemáticas y con ello realizar un cambio que brinde una mejor experiencia de seguridad a los usuarios. Los patrones de diseño son una solución general y reutilizable para la resolución de problemas del software [2].

Los ataques informáticos en aplicaciones web ocurren alrededor de 30000 ataques todos los días [3]. Según un informe reciente “Panorama de Amenazas para América Latina” de Kaspersky, la región ha sido testigo de un aumento de actividades cibercriminal como: phishing, ransomware y troyanos [4]. También nos dice Kaspersky que Ecuador es uno de los países que más ataques informáticos recibe en la Latinoamérica, algo muy preocupante entre los ataques que más se realizan son: troyanos, adware, phishing [5]

El problema es especialmente relevante en el contexto de desarrollar software para evitar vulnerabilidades informáticas, donde los patrones de diseño y buenas prácticas de seguridad cooperan para evitar problemas de seguridad. El no seguir guías o buenas prácticas de seguridad generan dificultades a usuarios afectando que la información de los usuarios no se proteja y trate de la mejor manera. Por este motivo se realizó un árbol de problemas mostrado en la Figura 1.

Figura 1

Árbol de problema



Nota. Autoría propia

Objetivos

Objetivo General

Desarrollar una aplicación web mediante el uso de patrones de diseño y validación de la seguridad con Owasp.

Objetivos Específicos

- Realizar la revisión de literatura acerca de patrones de diseño para el desarrollo de aplicaciones seguras.

- Desarrollar una aplicación web utilizando un patrón de diseño y ocupando estándares de seguridad.
- Validar la seguridad de la aplicación web con las buenas prácticas de seguridad el Top Ten de Open Web Application Security Project (Owasp).

Alcance

Este trabajo tiene por finalidad identificar el patrón de diseño más utilizado y seguro en el desarrollo de aplicaciones web y en conjunto con el Top Ten de Owasp para el desarrollo de un prototipo de agenda electrónica. Esto lo logró mediante una investigación en fuentes bibliográficas como: artículos de revistas de alto impacto, tesis, libros, etc. Se llevó a cabo una investigación sobre el uso de patrones de diseño seguros en la bibliografía seleccionada para, posteriormente identificar el patrón de diseño más utilizado en aplicaciones web. Luego se aplicó el patrón seleccionado en el desarrollo de un prototipo de agenda electrónica.

Se desarrolló un prototipo de agenda electrónica personal para organizar actividades e información personal. Esto contribuyó a la eficiencia en las actividades que se tenía que realizar y la distribución correcta del tiempo.

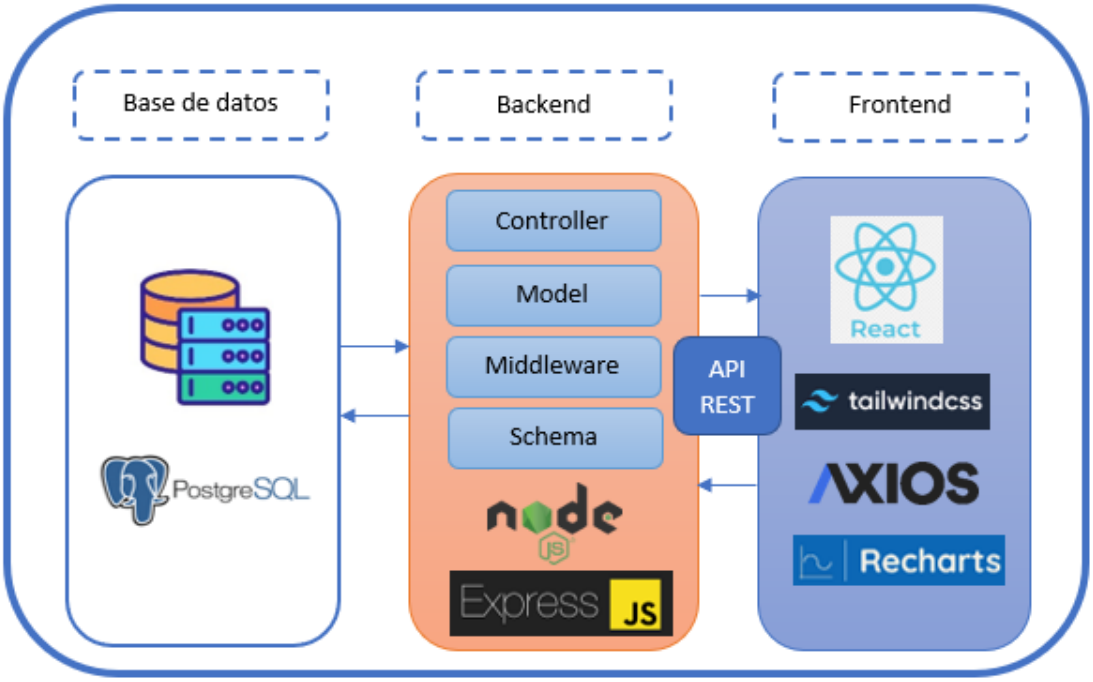
La agenda electrónica consta de los siguientes módulos:

- **Módulo de roles:** Existen dos tipos de usuarios un usuario normal. Este módulo permite crear usuarios, modificarlos, eliminarlos, asignar permisos, cifrado de la información del usuario y autenticación de usuarios.
- **Módulo de publicaciones:** Se permitió crear, editar, eliminar eventos para las actividades.

Se aplicó la metodología Scrum para el desarrollo del proyecto, lo cual ayudó a optimizar la calidad de proyectos, limitar costos y reducir tiempo [6]. Las herramientas utilizadas para el apartado del Back-End es JavaScript con el framework NodeJs y Express, la parte del Front-End se desarrolló con el Framework de React y para el almacenamiento de datos se utilizó PostgreSQL. Toda la solución web fue desplegada localmente.

Figura 2

Diseño inicial de la arquitectura



Nota. En la figura se muestra la disposición de los componentes de arquitectura de la aplicación inicial. Autoría propia

Para la verificación se utilizó OWASP ZAP, el cual es una herramienta que su función es escanear las vulnerabilidades que pueda tener una aplicación web y con ello detectar y resolver estos problemas [7]. Se escaneó el aplicativo web con la herramienta y esta nos proporcionó un informe con las vulnerabilidades que tiene el aplicativo. Con base en ese informe, se corrigieron estas fallas detectadas y posteriormente se volvió a pasarlo para asegurarnos que las correcciones habían resuelto estos problemas detectados.

Metodología

La investigación que se llevó a cabo tuvo como objetivo ser de carácter aplicado, enfocándose en resolver problemas específicos identificados en el desarrollo de software seguro.

Objetivo 1: Realizar la revisión de literatura acerca de patrones de diseño para el desarrollo de aplicaciones seguras.

- **Técnica:** Revisión bibliográfica
- **Instrumentos:** Bases de datos académicos, artículos científicos, tesis, estándares y documentación técnica.
- **Desarrollo:** Se llevó a cabo una revisión de fuentes académicas y técnicas relacionadas con respecto a patrones de diseño, vulnerabilidades informáticas en aplicaciones web. Además, se considera la importancia de usar las buenas prácticas de seguridad de OWASP específicamente el TOP TEN de este mismo.

Objetivo 2: Desarrollar una aplicación web utilizando un patrón de diseño y ocupando estándares de seguridad.

- **Técnica:** Desarrollo de software seguro basado en patrones de diseño. Buenas prácticas de seguridad y metodologías ágiles
- **Instrumentos:** Herramientas de desarrollo JavaScript con el framework Express para el back-end y para el front-end el framework React, herramienta de gestión de proyectos Scrum, base de datos Postgres.
- **Desarrollo:** Para el desarrollo de los dos módulos, se adoptó el patrón de diseño MVC y se utilizará JavaScript junto con el framework Express para el back-end. Además, para el front-end se usará el framework React y para la seguridad se aplicará las buenas prácticas de seguridad de OWASP específicamente el TOP TEN de OWASP. El desarrollo tuvo un enfoque modular e iterativo, dividiéndose en sprints de dos a tres semanas. Al finalizar cada sprint se realizó una revisión y ajustes que sean necesarios. Postgres se usa para la administración de la base de datos.

Objetivo 3: Validar y evaluar la seguridad de la aplicación web con las buenas prácticas de seguridad el Top Ten de Open Web Application Security Project (Owasp).

- **Técnica:** Prueba de seguridad
- **Instrumentos:** OWASP ZAP
- **Desarrollo:** Una vez desarrollado la aplicación web, se procedió a su evaluación con la herramienta OWASP ZAP, con el objetivo de comprobar que no existan problemas de seguridad en la aplicación web. Se tomó como referencia el top ten de Owasp para realizar el análisis de vulnerabilidades y se consideró corregida

una vulnerabilidad después de aplicar medidas que mitiguen esto y cuando la herramienta dejó de detectarla en un nuevo escaneo.

Para el desarrollo del primer objetivo, se elaboró un marco teórico de los patrones de diseño, se realizó una búsqueda de información en diversas fuentes bibliográficas, como artículos de alto impacto, libros, trabajos de tesis, etc. Luego se respondió a las siguientes preguntas: ¿Cuáles son los patrones más frecuentes que aportan mayor valor de seguridad en datos sensibles en las aplicaciones web?, ¿Cuáles son los patrones de diseño más frecuentes en el desarrollo de aplicaciones web? Posteriormente, se seleccionó un patrón de diseño basándose en la revisión literatura y se aplicó en la aplicación web.

Con respecto al segundo objetivo, se utilizó la metodología ágil Scrum la que nos ayudó para optimizar la calidad, limitar costos y reducir tiempo [6]. Para el apartado del Back-End se utilizó JavaScript con el framework NodeJs, la parte del Front-End se desarrolló con el Framework de React y para el almacenamiento de datos se usó PostgreSQL. Toda la solución web será desplegada localmente.

Para llevar a cabo el tercer objetivo, se utilizó las buenas prácticas de seguridad el Top Ten de Open Web Application Security Project (Owasp). Se verificó el cumplimiento de las buenas prácticas recomendadas por Owasp en la aplicación web desarrollada.

Justificación

Este trabajo se enfocó en el Objetivo de Desarrollo Sostenible Nº9: “Industria, Innovación e Infraestructura”. En base a la meta 9.b, la que nos dice que, “Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas [8].

También se tomó en cuenta el plan “Creación de oportunidades 2021 Ecuador”, este trabajo apuntó al “Eje de seguridad integral”, específicamente al “Objetivo 10 Garantizar la soberanía nacional, integridad territorial y seguridad del Estado” pues una de las bases es incrementar el índice de ciberseguridad de 26,6 a 51,3 [9].

El proyecto que se realizó tuvo como objetivo desarrollar software seguro mediante la aplicación de patrones de diseño, utilizando estándares de seguridad para los

riesgos informáticos en un prototipo de agenda electrónica. Los beneficiarios de este estudio fueron para estudiantes o profesionales de software con conocimientos básicos de programación que “Como dice Álvaro Miguel Varela, elegir los patrones de diseño adecuados garantiza una mayor eficiencia y calidad del software” [10].

Justificación Tecnológica: El uso los patrones de diseño, se proporciona soluciones a dificultades que se presentan en el desarrollo de software. Esto, permitió que los desarrolladores pueden crear software más seguro y robusto [11].

Justificación Teórica: En este proyecto se consideraron los conceptos de Patrones de diseño, los cuales se seleccionó uno para la implementación en una aplicación web. La solución propuesta consistió en desarrollar este software seguro utilizando el patrón de diseño más empleado y aplicando estándares de seguridad.

CAPÍTULO I

MARCO TEÓRICO

Las aplicaciones web son herramientas necesarias en la era digital actual, utilizadas para facilitar la interacción entre usuarios y sistemas. Sin embargo, su popularidad también las convierte en un objetivo frecuente de ataques informáticos.

En este capítulo, examinaremos sobre patrones de diseño, vulnerabilidades en aplicaciones web y las buenas prácticas de seguridad de OWASP (TOP TEN).

1.1 Patrones de diseño

1.1.1 ¿Qué son los patrones de diseño?

Los patrones de diseño se conocen como una medida amplia y reutilizable que busca resolver dilemas comunes en el desarrollo del software. Además, permiten que nuestro software llegue a ser más flexible, reutilizable y mantenible [12]. Un patrón de diseño no siempre se lo utiliza en el desarrollo, pero su uso nos evita problemas. Además, lo importante es comprender los patrones de diseño y sus funciones para así elegir el patrón que se adapte a nuestras necesidades y problemas. Como desarrolladores de software, evaluamos el código basándonos en criterios como limpieza, expresividad, eficiencia en el uso del espacio en memoria y rapidez. Lo más importante es poder realizar ajustes en nuestro código sin causar problemas o costos elevados. Los patrones de diseño se presentan como la mejor práctica para abordar estos criterios. Según [13], los patrones constan de seis reglas que los definen:

- **Son soluciones probadas y verdaderas:** Al utilizar los desarrolladores, los patrones de diseño no solo aseguran la funcionalidad, sino que también garantizan que han sido modificados varias veces y optimizados.

- **Son fáciles de reutilizar:** Los patrones de diseño se describen como soluciones reutilizables que se pueden modificar para que se resuelvan en distintas situaciones.

- **Tiene una fuerte personalidad:** Los patrones de diseño pueden describir una solución, pero la podemos modificar para que nos facilite la implementación y nos dé los resultados esperados.
- **Facilitan la comunicación:** El conocimiento que posee cada desarrollador sobre los patrones puede facilitar las posibles soluciones a problemas con mayor eficacia.
- **Eliminar la necesidad de refactorizar el código:** Cuando en el desarrollo de una aplicación se han tomado en cuenta los patrones de diseño, es posible que no se necesite reescribir código más adelante. El patrón de diseño se encarga de este tipo de problemas y si posteriormente se realizan actualizaciones, cualquier desarrollador puede aplicarlas sin esfuerzo y sin complicaciones.
- **Reduce el tamaño del código base:** Los patrones de diseño manejan menos código que otras soluciones porque suelen ser bonitos y óptimos. Aunque algunos desarrolladores suelen aumentar el código extra para mejorar la comprensión.

1.1.2 Tipos de patrones de diseño

Creacionales

Este tipo de patrón se utiliza para facilitar la creación de objetos o instancias de clases. Además, proporcionan la solución a problemas relacionados con la creación de objetos, la reutilización y la flexibilidad [14].

- **Abstract Factory:** Se utiliza para las familias de objetos y que estas se relacionen sin explicar a qué clase pertenece [13].

- **Builder Patterns:** Se emplea para generar diversos elementos siguiendo una secuencia de reglas que ayudarán a la reutilización del código, lo que facilita la elaboración de subclases [15].
- **Factory Method:** Se emplea para generar nuevas instancias de un objeto dentro de una superclase, permitiendo que estas subclases tengan la capacidad de crear diferentes objetos [16].

- **Prototype:** Se emplea para generar instancias adicionales. Mediante la declaración de una interfaz, esto permite la acción de replicarse para los objetos. [15].
- **Singleton:** Se emplea para limitar la generación de nuevos objetos de una clase a un objeto específico [13].

Estructurales

Los patrones estructurales constituyen un conjunto de patrones que se centran en la búsqueda de agrupar los objetos y clases, con el objetivo de crear estructuras más amplias y funcionales. Además, son herramientas que contribuyen a la organización y eficiencia del código. Al mismo tiempo, facilitan la adaptabilidad y el mantenimiento de los sistemas [14].

- **Adapter:** Se emplea para posibilitar la colaboración entre objetos que presentan incompatibles [15].
- **Bridge:** Se emplea para abordar el problema de la herencia entre clases al dividir clases que se relacionan en dos categorías distintas: implementación y abstracción, permitiendo el desarrollo de manera independiente [13].
- **Composite:** Se emplea para agrupar objetos como si fueran un solo conjunto, lo que simplifica que los objetos adopten la configuración de una estructura jerárquica, posibilitando la manipulación de estas estructuras como entidades individuales [16].
- **Decorator:** Se emplea para ampliar las capacidades de un objeto mediante la incorporación de funcionalidades adicionales a través de objetos encapsulados que proporcionan dichas funcionalidades [15].
- **Flyweight:** Disminuye las dimensiones de los objetos al guardar solo su estado y compartiendo el resto de la información entre múltiples objetos semejante [15].
- **Proxy:** Se emplea a fin de generar objetos que representan ocupaciones derivadas de distintas clases y objetos utilizados para acceder a las funcionalidades asociadas [16].

Comportamiento

Estos patrones simplifican la detección y exploración de métodos para la comunicación entre los objetos que interactúan entre sí. Además, son herramientas empleadas para mejorar la colaboración y el flujo de información entre diversos componentes del sistema, lo que posibilita que sea más adaptable [13].

- **Chain of responsibility:** Se emplea para posibilitar la transferencia de solicitudes a lo largo de una cadena de controladores, de manera que, al recibir la solicitud, esta pueda ser procesada y transferida [15].
- **Command:** Se emplea cuando es preciso incluir dentro de un objeto todos los parámetros necesarios para la ejecución de una acción [13].
- **Interpreter:** Utilizando Interpreter, se realiza la evaluación de un lenguaje mediante una interfaz que explique el contexto en el que se lleve la interpretación [15].
- **Iterator:** Se emplea con el fin de ofrecer acceso secuencial a cierto a un conjunto específico de elementos vigentes en un objeto de acopio y no requerir comercio de información [16]
- **Modelo - Vista – Controlador (MVC):** Es crucial ,ya que permite la separación de preocupaciones, divide el código en áreas discretas para permitir actualizar escalar el programa [13].

1.2 Comparativa de patrones de diseño

Tabla 1. Trabajos relacionados de patrones de diseño.

Autor o Autores	Investigación	Patrón de diseño
Gavilánez Oscar, Layedra Natalia, Ramos Vinicio	Estudio comparativo de modelos de diseño de software enfocado en la creación de programas móviles de alta índole: Un análisis metódico de la literatura [17].	- Template Method - Model-View-Controller (MVC) - Front Controller - MVVM
Montenegro Isaac, Rodríguez Luis, Salazar Gabriela	Implementación de modelos de diseño: en un caso práctico [18].	- Polimorfismo - Indirección - Factoría - Singleton - Observador

Maricruz Acosta	Análisis de modelos de diseño en plataforma Java Enterprise versión 6 destinado al diseño de aplicaciones web [19].	- Modelo-Vista-Controlador (MVC) - Front Controller
Aldás Daniel Andrade Maritza	Manual práctico para el empleo en modelos de diseño en el curso de avance de software [20].	Abstract Factory
Corao Francisco Vanega Mariana	Influencia del empleo de modelos de diseño en la industria del software Costarricense [21].	- Modelo-Vista-Controlador (MVC)
Foutse Khomh , YannGael Guéhéneuc	Impacto de los patrones de diseño en la calidad del software: ¿Dónde están las teorías? [22].	- Visitor - Observer
William Flageol, Éloi Menaud, YannGaël Guéhéneuc, Mourad Badri	Un estudio de mapeo de las características del lenguaje que mejora los patrones de diseño orientado a objetos Patrones [23].	- Observer - Visitor - Decorator
Dmitrii Drozdov, Udayanto Atmojo, Cheng Pang, Muhammad Irfan	Utilización de patrones de diseño de software en la fabricación orientada al producto de productos: Un estudio de caso [24].	- Product-driven software - Design pattern
Katzmaier Alexander, Hanneghan Martin	Evaluación de patrones de diseño para dispositivos móviles y web. Basado en Frameworks de aplicaciones [25].	- Modelo-Vista-Controlador (MVC)
Phek Thung, Chu Jian, Swee Thung, Shahida Sulaiman	Optimizar una aplicación web a través de patrones de diseño: Un caso práctico [26].	- Modelo-Vista-Controlador (MVC) - Presentation Abstraction Control (PAC
Ampatzoglou Apostolos , Frantzeskou Georgia , Stamelos Loannis	Un enfoque para medir el efecto de los patrones de diseño en la calidad del software [27].	- Abstract Factory -Visitor - Bridge
Serrato Ricardo, Rodríguez Gustavo, Pérez Julio, Pomares Saul	Diseño de sistemas de software basado en patrones para métodos de tipo Newton tipo Newton [28].	- Bridge - Facade - Adapter - Abstract Factory - Singleton - Template Method

Elshater Yehia, Martin Patrick	Utilización de modelos de diseño con el propósito de optimizar la utilidad de los servicios web [29].	-Singleton
Valencia Juan José	Influencia de los modelos de diseño de software en la protección de programas web [30].	- Chain of Responsibility - Proxy - State (Rate Limiting)
Gonzáles Christian	Una revisión de patrones de diseño dirigido en las aplicaciones web [31].	- Modelo-Vista-Controlador (MVC) -Modelo Vista Presentador (MVP)
Martha Correa, Ivón Beltrán, Fabio Granados	Exploración del impacto del empleo de patrones de diseño en la etapa de mantenimiento [32].	- Singleton
Edson Castañeda	Plan de modelo de diseño de software dirigido a evitar la separación automatizada en materia web [33].	- AntiScraping View - Template View
González Sergio	Manejo de modelos de diseño para la decisión de dilemas de software en la creación de aplicación móvil iOS [34].	- Modelo-Vista-Controlador (MVC) - Object Template
Hernández Eduardo, Rodríguez Roberto, Salinas Jesús	Aplicación Web con módulos de captura automática para la gestión de sueldos en una constructora, empleando modelos de diseño de software [35].	- Modelo-Vista-Controlador (MVC) - Observer - Singleton

Tabla 2. Comparativa de patrones de diseño

Patrón	Cuando usarlo	Ventajas	Desventajas
Singleton	Cuando es necesario hacer una única instancia que controle el acceso.	Control centralizado y ahorro de recursos.	Dificulta pruebas unitarias y mal usado como variable global
MVC	Cuando se quiere separar la lógica de negocio de la visualización y control.	Separación de las responsabilidades y facilita el mantenimiento.	Si la aplicación crece mucho se puede volver complejo.
Abstract Factory	Cuando el sistema debe ser independiente de cómo se origina y componen los objetos.	Facilita la interoperabilidad entre objetos relacionados.	Complejidad elevada si hay muchas variaciones.

Observer	Cuando un cambio en un objeto debe notificar a otro	Acoplamiento flexible entre objetos, escalable.	Difícil de depurar.
Proxy	Cuando necesitas controlar el acceso, registro de operaciones	Control de acceso, carga diferida.	Aumenta la complejidad del sistema.
Templated Method	Cuando múltiples clases comparten lógica, pero tienen diferencias	Reutiliza lógica común, fácil de extender.	Rígido, difícil de cambiar pasos sin herencia.

Se llevó a cabo la investigación en fuentes bibliográficas, donde se han revisado distintos artículos de alto impacto, trabajos de tesis, entre otros. Se observó que los patrones de diseño más utilizados sobre estas investigaciones son el patrón Modelo-Vista-Controlador (MVC) y el patrón Singleton. Los cuales se han empleado 8 veces para el MVC y para el Singleton se han empleado 5 veces. Además, se elaboró una tabla comparativa de patrones donde se presentaron ventajas, desventajas y cuándo usarlos. Con ello, se ha decidido que estos dos patrones serán seleccionados para poder realizar la aplicación web.

1.3 Seguridad en aplicaciones web

La defensa en aplicaciones web comienza desde que el usuario accede a un navegador a través de internet. Sabemos que los programas web no están en las computadoras de los clientes, están en servidores remotos donde se intercambian datos de entrada y salida. La seguridad es un campo preocupante, ya que, la información que se transporta es importante y lo fundamental es proteger de cualquier riesgo que quiera vulnerar esta información, Además, la seguridad no solo es prevenir ataques también detectar y corregir estos, con ello reducir el riesgo de ataques en las aplicaciones web [36].

Las vulnerabilidades pueden encontrarse en cualquier parte puede ir desde la codificación de la aplicación, la transmisión de datos, almacenamiento, control de accesos, hasta la confidencialidad. Por estas razones, es necesario implementar técnicas y mecanismos que nos ayuden a cuidar la información de la aplicación web [37]. Las medidas de seguridad para la protección de aplicaciones web son importantes para la protección de datos, evitar ataques cibernéticos, integridad y cumplir normativas [3].

- **Protección de datos sensibles:** Las aplicaciones tienen datos confidenciales, los cuales deben ser protegidos para evitar robos de ciberdelincuentes.
- **Prevención de ataques:** Las medidas de seguridad ayudan a prevenir malware, robo de datos y al implementar medidas de seguridad, podemos reducir que se exploten este tipo de vulnerabilidades.
- **Mantenimiento de la integridad de las aplicaciones:** Las medidas de seguridad ayudan a que las aplicaciones web tengan un funcionamiento correcto y se eviten errores de codificación, configuraciones y diseño.
- **Cumplimiento normativo:** Países e industrias cuentan con regulaciones que ayudan a la protección de la información y esto ayuda a evitar problemas.

1.4 OWASP

OWASP (Open Web Application Security Project) es una iniciativa mundial abierta; consiste en una comunidad de profesionales y voluntarios en el campo de seguridad informática. Su objetivo principal es mejorar la protección de los programas web mediante concienciación, la educación y promoción de prácticas de seguridad óptimas [38]. Un aspecto importante de OWASP es que todo el material está disponible de forma gratuita y accesible. Dado que las aplicaciones web son blanco común de ataques cibernéticos, las vulnerabilidades de seguridad en estas pueden resultar en problemas como pérdida de datos, robo de identidad, entre otros. La relevancia de OWASP reside en un papel fundamental para la defensa contra las amenazas cibernéticas [39].

1.4.1 Top Ten de OWASP

El Top ten de Owasp se actualizado varios literales de la lista y ha incluido algunos y estos son los siguientes:

Figura 3

Principales riesgos de seguridad según Owasp

Nota. En la figura se detalla las vulnerabilidades clasificadas en el Top Ten de Owasp . Adaptado de Indusface [76].

Pérdida del control de acceso

La gestión de accesos facilita el cumplimiento con políticas de seguridad como autorizaciones y roles. Estas restricciones hacen que los usuarios no accedan a lugares que no tengan permisos y con ello controlan quién accede a los recursos [38].

Fallos criptográficos

Existen datos los cuales deben ser cifrados como credenciales de acceso, documentación sensible, etc. Con esto se evita que ciberdelincuentes generen problemas catastróficos [40].

Inyección

Estas son posibles cuando las entradas del usuario no son desinfectadas adecuadamente y esto se da con el lenguaje SQL donde los usuarios malintencionados pueden ingresar comandos y jugar con los datos que puedan recolectar [40].

Diseño inseguro

Al construir aplicaciones web, la integridad de esta comienza desde la etapa de diseño, y esta es una de las que se ha incluido en esta nueva lista por el hecho de que ha existido una cantidad grande de aplicaciones con defectos en el diseño [38].

Configuración de seguridad defectuosa

La configuración correcta de la seguridad es fundamental porque los ciberdelincuentes buscan la manera de causar daño por medio de defectos, software no actualizado, protección insistente y con ello evitar problemas en el software [40].

Componentes vulnerables y obsoletos

Un atacante informático compromete un software al aprovechar riesgos o debilidades ya identificadas o habituales, las que ya deberían ser cubiertas [38].

Fallos de identidad y autentificación

Suceden cuando no se controlan los puntos de acceso, como la cantidad de intentos de autentificación, la complejidad de las contraseñas o implementar el sistema de doble autentificación [38].

Defecto en el software y en la integridad de los datos

Las aplicaciones tienen a veces mecanismos automáticos para actualizarse, pero cuando estas no son verificadas, los atacantes pueden interceptar estas y modificarlas para su conveniencia [38].

Errores en el registro y la vigilancia de la seguridad

Una ausencia de registros de hechos llamados login, etc. Estos deben estar registrados para detectar amenazas, bloquear amenazas e investigar los problemas que se den en el software [40].

Falsificación de solicitud del lado del servidor

Cuando un agresor informático tiene la ocasión de obligar al servidor a cometer acciones que no están previstas y con ello modificar y realizar peticiones no autorizadas [38].

1.4.2 Relación de Owasp con Normas ISO

El top ten de Owasp se relaciona con las ISO 27001, 27002 y la 27034, específicamente con los siguientes artículos:

ISO 27001

Muchos de sus controles son directamente aplicables a las aplicaciones web y ayudan a cumplir con OWASP Top 10. Algunos de los controles más relevantes incluyen [41]:

- **A.5.1 Control de acceso:** Asegura que solo los usuarios con autorización tengan el acceso a funciones y componentes de la aplicación.
- **A.7.3.1 Seguridad de la entrada:** Válida y filtra las entradas de los usuarios para evitar ataques de inyección.
- **A.7.3.2 Seguridad de la salida:** Codifica de manera correcta la salida para evitar ataques de scripting (XSS).
- **A.7.6 Gestión de parches:** Mantiene actualizados los componentes de software para evitar posibles explotaciones de vulnerabilidades.
- **A.12.1 Gestión de activos:** Identifica y protege la información de la aplicación, esto incluye el código del software y datos.

ISO 27034

Es la que más directamente se alinea con OWASP Top 10, ya que se centra en el ciclo de vida completo del desarrollo de software. Algunos de los artículos de ISO 27034 que más contribuyen a cumplir con OWASP Top 10 son:

- **Diseño seguro:** Aplicar principios de diseño seguro para minimizar la introducción de vulnerabilidades en el código [42].
- **Requisitos de seguridad:** Definir requisitos de seguridad claros y detallados para la aplicación (hashing seguro, almacenamiento de claves).
- **Desarrollo seguro:** Implementar prácticas de desarrollo seguro, como revisiones de código, pruebas de seguridad, gestión de vulnerabilidades e Implementación de MFA (Multi-Factor Authentication) [43].

1.5 Vulnerabilidades en aplicaciones web

Las debilidades son diversas en las plataformas en línea, siendo una debilidad o falla que pueden explotarlos por un ciberdelincuente. Esto permite al atacante acceder, modificar o eliminar información, así como controlar la aplicación, lo que perjudica al usuario [17].

1.5.1 Principales vulnerabilidades

Inyección

El empleo de comandos representa una de las tácticas más frecuentes en aplicaciones web, implica que el atacante aprovecha alguna debilidad del sistema para llevar a cabo comandos en SQL, NoSQL, entre otros, sin el consentimiento de los programadores del sistema. Esto se hace con el fin de obtener datos no autorizados [44].

Autentificación Comprometida

Las aplicaciones vinculadas a la autentificación y dirección de sesiones se ejecutan de manera errónea, lo que facilita a los atacantes que arrebaten el dominio de datos sensibles de los usuarios. Además, pueden aprovechar otras deficiencias para asumir los roles de otros usuarios y mantener un acceso temporal o permanente no autorizado [44].

Exposición de datos confidenciales

Numerosas aplicaciones web no cuentan con una adecuada protección o gestión eficiente de datos sensibles. Los ciberdelincuentes suelen sustraer o modificar esta información, cuya protección resulta insuficiente, para llevar a cabo actividades ilícitas [45].

Entidades externas XML (XXE)

Este tipo de ataque se da apenas se procesa una entrada XML que incluye una mención a una entidad externa mediante un analizador XML por defecto de forma vulnerable. Este tipo de riesgos puede resultar en la exposición de datos sensibles, la

interrupción de servicios, la manipulación de llamar desde el servidor y la exploración de puertos [45].

Configuración de seguridad incorrecto

Errores en las estructuras de seguridad son frecuentes en todas las etapas de la creación de una aplicación. Estos fallos suelen surgir cuando estas configuraciones se establecen, implementan y mantienen con valores por defecto [45].

Malas prácticas de codificación

La adopción de malas prácticas de codificación inadecuadas representa una de las principales vulnerabilidades en aplicaciones. La ausencia de validación de entrada, por ejemplo, puede abrir la puerta a que un atacante aproveche las deficiencias en el código [44].

1.6 Marco de trabajo Scrum

Las metodologías ágiles nos muestran su énfasis en la flexibilidad, colaboración continua y adaptabilidad al cambio. Opuestos a las metodologías tradicionales, que se enfocan en la planificación extensa y en la entrega final del producto. Las metodologías ágiles se fundamentan en la entrega constante y la mejora continua [46]. Las metodologías ágiles, en lugar de seguir un plan rígido, promueven un enfoque flexible y cíclico, donde cada proceso se divide en ciclos más pequeños, permitiendo que cada iteración construya sobre la anterior, permitiendo una evaluación constante y ajustes basados en el feedback recibido. La colaboración estrecha y la comunicación fluida son pilares fundamentales de las metodologías ágiles. Estos enfoques, que han demostrado su eficacia en diversos campos, como el de la arquitectura, permiten al equipo comprender en profundidad las necesidades del cliente y realizar ajustes de manera ágil. Además, la estructura de los equipos ágiles, con roles bien definidos, pero flexibles, fomenta la autonomía y la responsabilidad compartida entre sus miembros [47]. Scrum es una estructura simple creada para la dirección de proyectos y promueve al trabajo en equipo, permitiéndonos resolver problemas complejos que puedan surgir durante el proyecto. Además, nos proporciona herramientas como valores, roles, principios, entre otros, que contribuyen a que el equipo entregue productos eficientes y de alto valor que satisfagan las necesidades del cliente [48].

1.6.1 Scrum team

El Scrum team es un equipo de personas que son responsables para el desarrollo de las diferentes actividades como: investigación, mantenimiento, experimentación, desarrollo, etc. que se presenten en el proyecto.

El equipo o Scrum team se divide en tres roles fundamentales: Scrum Master, Product Owner y Developers. [49].

- **Developers:** Son los encargados del desarrollo del producto con referencia a los requerimientos que se han propuesto, este suele ser un equipo pequeño compuesto de diferentes áreas como: desarrolladores, testers, etc.
- **Producto Owner:** Es el responsable de la toma de decisiones y conoce muy bien las necesidades del cliente y entiende qué busca este y comunica al equipo qué es necesario para el desarrollo del proyecto, además asegura que el equipo cumpla con sus funciones y aprovecha todo lo que tenga para el progreso del proyecto.
- **Scrum Master:** Respaldar que el equipo de trabajo y la metodología continue sin problemas nos referimos a que va a brindar apoyo al equipo en cuestión de dudas, principios y que el equipo esté enfocado en el proyecto.

El sprint es el contenedor para los demás eventos que existen en el Scrum. Cada evento es una oportunidad para inspeccionar y mejorar el desarrollo del proyecto y con esto disminuir las dudas que existan. Estos suelen durar de un mes o menos y dentro del Sprint existen otros eventos.

- **Sprint Planning**: Es el inicio de un Sprint en donde se reúne todo el equipo para conversar sobre lo que se va a realizar en el Sprint [50].
- **Daily Scrum:** Son inspecciones rápidas de 15 minutos que se realizan durante el Sprint para ver cómo va el progreso del proyecto [49].
- **Sprint Review:** Se realiza al fin del Sprint en donde el equipo se reúne a conversar para revisar cómo se han realizado los Sprints y calificar si se cumplieron los objetivos [49].
- **Sprint Retrospective:** Se realiza al fin del Sprint, en este evento, todo el equipo investiga los fallos y aciertos que se tuvieron en el desarrollo de los Sprints [50].

1.6.2 Proceso de Scrum

Figura 4

Dinámica del marco de trabajo SCRUM

Nota. En la figura se muestra el desarrollo del proceso dentro de SCRUM. Adaptado de Tuleap [77].

1.7 Herramientas

1.7.1 Backend

Es responsable de gestionar toda la comunicación que abastece al front-end y conformado: códigos, bases de datos, etc. Con la finalidad de que una aplicación realice sus funciones, es necesaria de una gran cantidad de información, la cual es almacenada en un sistema tecnológico. Contrario al front-end, este tipo de información no está a la vista y vacante para él.

Algunas de las tareas que realiza el back-end incluyen:

- Almacenar y recuperar datos de una base de datos.
- Procesar formularios.
- Autenticación de usuarios.
- Gestión de la seguridad.
- Comunicación con otros servicios

La elección de JavaScript, Node.js y Express para el desarrollo del back-end de una agenda electrónica proporciona una base sólida para construir una aplicación web escalable, eficiente y fácil de mantener. Además, la amplia comunidad y el ecosistema de herramientas disponibles facilitan el desarrollo y la resolución de problemas.

- Aumento de la productividad del desarrollador.
- Mejora el rendimiento de la aplicación.
- Facilidad de mantenimiento y escalabilidad.
- Alineación con las mejores prácticas de desarrollo web.

La combinación de JavaScript, Node.js y Express ofrece una serie de ventajas que la convierten en una excelente elección para el desarrollo de un prototipo de agenda electrónica:

- **Rendimiento:** Ideal para aplicaciones de alta concurrencia y en tiempo real.
- **Desarrollo rápido:** Agiliza el proceso de desarrollo gracias a un único lenguaje y frameworks minimalistas.
- **Gran ecosistema:** Amplia variedad de herramientas y librerías disponibles.
- **Modernidad:** Se alinea con las tendencias actuales de desarrollo web.

JavaScript

Es un lenguaje de codificación que se emplea para el desarrollo web, esta nos ofrece una mayor flexibilidad y está pensado para agregar interacciones y agilidad a las páginas web. Además, se lo utiliza en aplicaciones de servidor, como Node.js y para aplicaciones móviles [51].

Node.js

Es una plataforma de ejecución de código abierto y compatible con varias plataformas que nos facilita la creación de aplicaciones de red y que estas sean rápidas y escalables. Emplea un tipo de entrada y salida sin bloqueo, dirigido por eventos. para mantener su ligereza y eficiencia [52].

Express

Es un framework minimalista, rápido y flexible para back-end en Node.js diseñado para el desarrollo de aplicaciones escalables. Proporciona un conjunto robusto de herramientas y funcionalidades que facilitan la ampliación del framework con componentes más avanzados según las necesidades del proyecto [53].

1.7.2 Frontend

Es la parte del desarrollo web que interactúa con el usuario, es decir, es todo lo que observamos en las pantallas al acceder a una aplicación web, móvil y sitio web. Esto abarca desde cosas sencillas, como tipo de letra, colores, movimientos, efectos visuales, hasta elementos más complejos. Su empleo es crucial para que el cliente disfrute una vivencia positiva, atractiva y lo más importante, funcional [54].

React

Es una biblioteca de código abierto de JavaScript creada por Facebook para reducir la dificultad de codificación. Se buscaba una alternativa para abordar este problema y se lo utiliza para crear interfaces de usuario. Esto permite a los ingenieros de software desarrollar aplicaciones web más complejas, pero de manera sencilla. Además, consta de un orden y jerarquía que ayuda al programador, proporcionándole una guía predefinida para implementar su aplicación [55].

React es una biblioteca JavaScript de código abierto extremadamente popular para construir interfaces de usuario. Su elección como front-end para tu proyecto se basa en las siguientes razones:

- **Componentes Reutilizables:** React fomenta la creación de componentes reutilizables, lo que agiliza el desarrollo y facilita el mantenimiento del código. Esto es especialmente útil en aplicaciones más grandes y complejas como una agenda electrónica.
- **Virtual DOM:** React utiliza un DOM virtual para realizar actualizaciones eficientes en la interfaz de usuario, lo que resulta en un mejor rendimiento y una experiencia de usuario más fluida.
- **Declarativo:** React tiene un enfoque declarativo, lo que significa que describes cómo quieres que se vea la interfaz de usuario en lugar de especificar cómo

cambiar el DOM directamente. Esto hace que el código sea más fácil de razonar y mantener.

- **Gran Ecosistema:** React cuenta con un ecosistema enorme de herramientas, librerías y componentes preconstruidos, lo que acelera el desarrollo y te permite aprovechar soluciones ya probadas.
- **Comunidad Activa:** La comunidad de React es muy grande y activa, lo que significa que encontrarás una gran cantidad de recursos, tutoriales y soporte en línea.

En comparación con otros frameworks, estos pueden ser más difíciles de aprender y tomar más tiempo para su estudio. En caso de problemas o dudas las comunidades de esta son pequeñas y no existe mucha información para poder solventar alguna duda o error . Al elegir React como front-end, estás optando por una tecnología moderna, flexible y con un gran respaldo de la comunidad. Esta elección te permitirá desarrollar una interfaz de usuario intuitiva y performance para tu agenda electrónica.

1.7.3 Base de datos

Postgres

Se trata de una estructura de gestión de base de datos de código accesible que goza de una concreta reputación debido a su confiabilidad, flexibilidad y respaldo. En conjunto con el lenguaje SQL, ofrece diversas funciones que escalan y gestionan las cargas de trabajo de datos. Además, admite tantos tipos de datos relacionales como no relacionales, lo que lo posiciona como una de las combinables, segura y prudente [56]. Se utiliza principalmente para almacenar datos destinados a diversas aplicaciones como móviles, web, análisis, etc. Beneficia a los desarrolladores que buscan entornos altamente escalables, ya que sus infraestructuras son locales y basadas en la nube [57]. Algunas de las ventajas que posee Postgres son que es de código abierto y gratuito. Esto nos permite personalizar y modificar el software según nuestras necesidades, otorgándonos mayor libertad y control sobre el software y datos que utilizamos. Además, es altamente escalable en cuestión de números de usuarios concurrentes y la cantidad de datos que puede gestionar. Es considerado el sistema de base de datos más extenso, destacándose

por sus velocidades rápidas de autentificación de datos, de lectura y escritura, difíciles de superar, convirtiendo a Postgres en una de las bases de datos más eficientes [57].

Tabla 2. Comparativa de Base de datos

Característica	MySQL	PostgreSQL	MongoDB	SQLite
Escalabilidad	Moderada	Alto	Muy alta	Baja
Integridad referencial	Sí (pero menos estricto)	Sí (claves foráneas, constraints)	No (requiere validación manual)	Sí (pero con limitaciones)
Concurrencia	Alta (Sin bloqueos)	Media (menos eficiente)	Alta	Limitada
Seguridad	Media	Alta	Media	Baja
Facilidad de uso	Alto	Media	Media	Alto
Rendimiento	Alto	Muy alto	Alto (excelente en grandes volúmenes)	Alto
Caso de uso ideal	Aplicaciones web simples	Proyectos complejos y escalables	Datos no estructurados	Proyectos pequeños

Después de un análisis de las diferentes opciones de base de datos, se ha visto que la más óptima es PostgreSQL porque esta nos ofrece un equilibrio entre rendimiento, escalabilidad, concurrencia y seguridad.

1.8 Arquitectura de software

La arquitectura de software es un marco importante para el desarrollo sostenible de sistemas informáticos eficientes y efectivos. Este representa el diseño general de un sistema, donde se definen sus componentes, la unión entre ellos y cómo se relacionan dentro del entorno operativo. En el ámbito de la arquitectura de software, se destacan elementos fundamentales como la modularidad, que consiste en dividir el sistema en partes más pequeñas y manejables, lo que facilita su comprensión, desarrollo y mantenimiento. Además, la arquitectura también debe asegurar la solidez y confiabilidad del sistema, garantizando su capacidad para gestionar de manera eficiente, reduciendo al

mínimo las interrupciones y preservando la integridad de los datos. Igualmente, la seguridad constituye un aspecto clave, ya que protege tanto la información como las operaciones del sistema frente a accesos no autorizados y otras posibles amenazas.

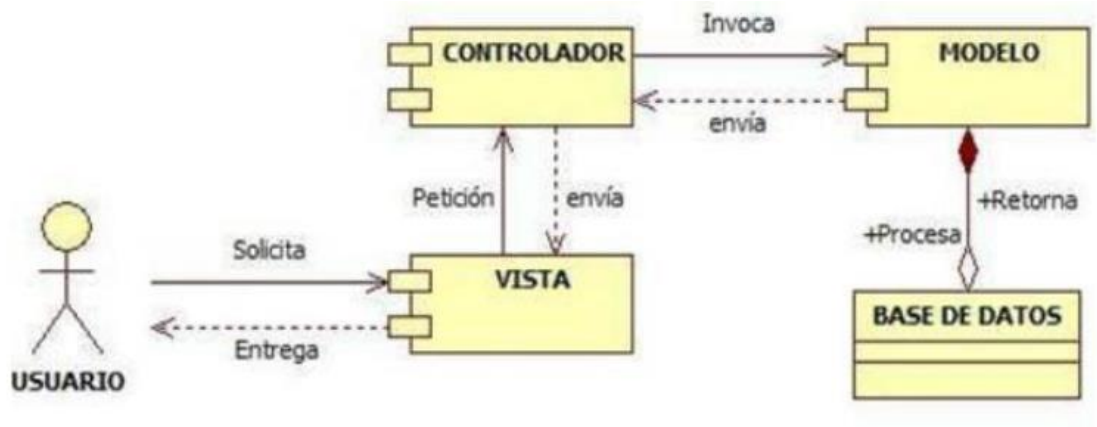
1.8.1 Patrón de Diseño Modelo - Vista - Controlador (MVC)

MVC es un patrón de diseño arquitectónico de software ampliamente reconocido y aplicado en aplicaciones web que sirve para clasificar la información, la lógica del sistema y la interfaz que se le muestra al usuario. En este tipo de arquitecturas existe un sistema central que gestiona las entradas y salidas del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final [58].

- **Modelo:** Este componente se encarga de manipular, gestionar y actualizar los datos. El modelo es responsable de acceder a la base de datos, procesar los datos y definir las reglas del negocio. Además, asegura que la abstracción de la información sea gestionada de manera eficiente y segura.
- **Vista:** Este componente se encarga de presentar al usuario final las interfaces, ventanas, páginas y formularios. Esto resalta la relevancia de mostrar datos de manera clara y efectiva a los usuarios, una función fundamental de la Vista dentro del marco MVC. Además, las Vistas pueden adaptarse según las necesidades que tenga el usuario, pero todas ellas la obtienen su información del Modelo.
- **Controlador:** Este componente desempeña un papel clave al encargarse de gestionar las instrucciones que se reciben, atenderlas y procesarlas. Su principal función es actuar como un puente de comunicación entre el modelo y la vista. Para ello, solicita al modelo los datos necesarios, realiza las operaciones o transformaciones requeridas para obtener los resultados esperados y finalmente entrega estos resultados a la vista. Esto permite que la vista los pueda mostrar de manera comprensible y adecuada para el usuario final, asegurando una interacción fluida y eficiente.

Figura 5

Funcionamiento del patrón de Diseño MVC



Nota. En la figura se muestra la estructura del patrón de diseño. De Sunardi [58].

El uso del patrón MVC en el desarrollo de aplicaciones ofrece numerosas ventajas, entre ellas separar los componentes (interfaz de usuario - lógica de negocio – acceso de datos) y permitir la implementación de cada uno por separado, es decir, si alguno de ellos falla se puede modificar fácilmente. Facilita la realización de pruebas unitarias. Este patrón también soporta una arquitectura escalable y flexible, minimizando el impacto de los cambios en una parte del sistema sobre otras [59].

1.8.2 Patrón de Diseño Singleton

Es un patrón que cuyo objetivo es evitar que se creen más objetos por clase. Esto se logra creando el objeto en una clase y solo tengo una única instancia y proporcioné un acceso global a dicha instancia [13].

1.8.3 Aplicación en el Desarrollo de la aplicación web

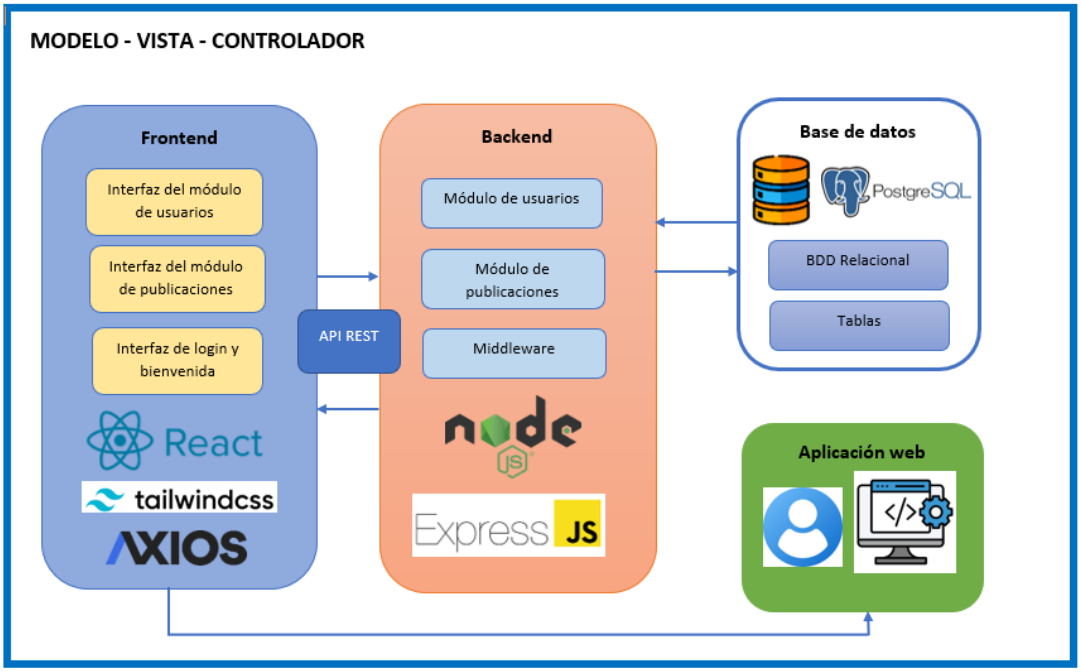
La implementación del patrón MVC en el desarrollo de la aplicación web para el prototipo de la agenda electrónica representa una decisión estratégica que ofrece beneficios. Al aplicar este enfoque, cada componente del prototipo de agenda electrónica se organiza de manera modular, lo que facilita tanto su desarrollo y garantizando una estructura clara y adaptable a futuras mejoras. Para la selección de este patrón se realizó una investigación en 19 fuentes bibliográficas, donde se han revisado distintos artículos de alto impacto, trabajos de tesis, entre otros. Observamos que los patrones de diseño sumamente utilizados sobre estas investigaciones son el patrón Modelo-Vista-Controllador (MVC) y el patrón Singleton.

Los cuales se han empleado 8 veces para el patrón MVC y para Singleton se han empleado 5 veces. Con ello, se han decidido por el patrón MVC será seleccionado para el desarrollo de la aplicación web.

1.8.4Arquitectura de la aplicación web

Figura 6

Arquitectura de la aplicación



Nota. En la figura se muestra la organización general de los componentes que forman la arqutectura la aplicación. Autoría propia

CAPÍTULO II

DESARROLLO

Desarrollo de la Aplicación web "Prototipo de agenda electrónica"

2.1 Iniciación

2.1.1 Adquisición de los requerimientos

En esta ocasión se realizó un análisis previo al problema que se va a solucionar y se asignó un equipo el cual va a formar parte del desarrollo del prototipo de agenda electrónica en todas sus etapas. Se utilizó la metodología agile Scrum dado que esta nos brindará un gran apoyo para abordar proyectos complejos y cambiantes de una forma más flexible.

2.1.2 Definición de Roles

Antes de iniciar el desarrollo del prototipo de agenda electrónica, es fundamental establecer una estructura clara del equipo para el proyecto, identificando a todas las partes interesadas y asignando roles específicos y responsabilidades dentro del proyecto. Este planteamiento asegura que todos los aspectos del proyecto sean manejados por personal calificado y que las interacciones entre los miembros del equipo sean eficientes y efectivas. A continuación, se presenta la Tabla 2 que detalla los roles dentro del marco de trabajo Scrum adoptado para este proyecto.

Tabla 3. Designación de roles

Nro.	Rol	Nombre	Responsabilidad
1	Producto Owner	Ing. Marco R. Pusdá, PhD	Encargado de asegurar que todos los requisitos de la aplicación web sean cumplidos y de priorizar las necesidades del proyecto
2	Scrum Master	Sr. Marlon Cachimuel	Encargado de supervisar el proceso de desarrollo, aplicando las metodologías ágiles.

			Así como de gestionar la resolución de obstáculos y asegurar el progreso.
3	Equipo Scrum	Sr. Marlon Cachimuel	Desarrollador encargado de codificar, diseñar y cumplir con los estándares de funcionalidad del proyecto.
4	Stakeholder	MSc. Daisy Imbaquingo PhD	Contribuye con el feedback durante el desarrollo del proyecto

2.1.3 Product Backlog

El producto backlog es una lista de todo lo que es necesario para desarrollar la aplicación web (prototipo de agenda electrónica). Esta lista constará con historias de usuario, tareas, mejorar, etc. Cada una de estas debe ser estimada para asegurar una planificación adecuada en tiempo y recursos. Para la estima las historias de usuario se empleó el método de los Tres puntos, que brindó una evaluación más justa y realista.

Estimación de Costo y Esfuerzo de desarrollo en proyectos basados en Scrum

La estimación ágil es una de las prácticas utilizadas para estimar el costo y el esfuerzo necesarios para concluir las tareas que se presentan en el Product Backlog, ya que el costo habitualmente se lo mide por dinero y el esfuerzo por el tiempo necesitado para concluir una tarea establecida. Dentro de la planificación de Sprint, estimar es una parte fundamental, ya que evita problemas de sobreestimación y subestimación, el Product Owner y el Scrum Master son los responsables de realizar las estimaciones con mucha precaución y antelación [60].

Método de Estimación T-Shirt

El método T-Shirt o “método talla de camiseta” utiliza las tallas XS (Extra Small), S (Small), M (Medium), L (Large) y XL (Extra Large), para dar estimaciones aproximadas de acumulación de elementos, donde se implica decidir un tamaño relativo

(mediano) mediante una reunión colaborativa en el equipo y llegar a un acuerdo colectivo con respecto a la estimación, asignado valor acordes a los requisitos según el tamaño relativo asignado [61]. Este enfoque se basa en estimaciones de tipo rango y no en números absolutos, facilitando que el equipo de desarrollo realizase acuerdos rápidos sobre estimaciones aproximadas.

Dentro del presente proyecto se utiliza el método antes mencionado con las tallas XS (Extra Small), S (Small), M (Medium), L (Large), XL (Extra Large) los cuales se asignó un rango de horas para presentar una estimación de esfuerzo dentro de una de las historias de usuarios necesarias para el desarrollo de este proyecto. En la Tabla 4 se presentan los rangos de horas de cada una de T-Shirt a utilizar.

Tabla 4. Rango de horas de trabajo

Tamaño de camisa	Rango de horas o semanas	Descripción
XS	1 - 10 horas	Tareas extremadamente sencillas y rápidas de realizar.
S	10 - 20 horas	Tareas pequeñas que requieren poco esfuerzo.
M	20 - 30 hora	Tareas de tamaño moderado y complejidad.
L	30 - 40 horas	Tareas más grandes y complejas.
XL	40 - 50 horas	Tareas muy grandes y complejas, que pueden llevar más tiempo.

Nota: En la tabla 4 se muestra el rango de horas de trabajo por tamaño de camisa. Fuente: Propia

Tabla 5. Product Backlog

Código	Historia de Usuario	Estimación	Prioridad	Actividad
H1	Identificar las necesidades	S	Alta	Identificar las necesidades del stakeholder

H2	Diseño de un boceto para la aplicación web	S	Media	Diseño de un boceto de cómo se va visualizar la aplicación web
H3	Back-end: Administrar los usuarios y roles	XL	Alta	La aplicación debe permitir al administrador la creación, edición, borrar de usuarios y asignación de roles.
H4	Back-end: Administrar publicaciones	L	Media	La aplicación debe permitir al usuario normal la creación, edición y borrar publicaciones del usuario.
H5	Ver historial de acciones de los usuarios	M	Media	La aplicación debe permitir al administrador ver datos del usuario como: tiempo de creación, actualización, reseteo de contraseña.
H6	Búsquedas de publicaciones por fechas y título	M	Baja	La aplicación permitirá al usuario normal realizar búsquedas de sus publicaciones por fechas y título.
H7	Front-end: Realizar pantallas para el usuario administrador	XL	Alta	En las pantallas del administrador se podrán ver, editar, borrar, actualizar los usuarios y roles.
H8	Front-end: Realizar pantallas para el usuario normal	XL	Alta	En las pantallas del usuario normal se podrá ver el perfil del usuario y editar su información y además poder ver, crear, editar y borrar sus publicaciones.
H9	Implementar validaciones de seguridad	L	Alta	La aplicación se le implementará validación de seguridad
H10	Conectar el Frontend y Back-end de la aplicación web	M	Media	Unir las funciones que se desarrollaron en el backend para que las use el Frontend y funciones la aplicación web.
H11	Corrección en Front-end y Back-end	L	Media	Corregir errores o mejoras en ambos campos.

H12	Probar la aplicación web	S	Baja	Examinar que la aplicación si cumple con sus funciones.

2.2 Planificación y estimación

La etapa de planificación y cálculo de recursos desempeña un papel fundamental en el ciclo de vida del desarrollo de software, especialmente en metodologías ágiles, donde la adaptabilidad y la flexibilidad son prioridades. Durante esta fase, se llevó a cabo la identificación de todas las actividades a realizar, junto con la estimación de los recursos y el tiempo necesarios, además de la organización de su ejecución en sprints o iteraciones.

2.2.1 Historias de Usuario

Las historias de usuario son esenciales para definir las funcionalidades que se desarrollaron en la aplicación web. Cada historia describe brevemente una necesidad del usuario y cómo el sistema debe responder a ella.

Tabla 6. Historia de usuario 1

HISTORIA DE USUARIO	
Número: 01	**Usuario:** Desarrollador
Nombre de la historia: Identificar las necesidades	
Prioridad: Alta	**Estimación:** S
Descripción: Como desarrollador de software, necesito conocer y los requisitos específicos del stackeholder con respecto al prototipo de la agenda electrónica. Permitiendo facilitar el diseño de la solución que cumpla con los requerimientos establecidos.	
Criterio de aceptación: • Listado donde conste las necesidades para la aplicación web.	

Tabla 7. Historia de usuario 2

HISTORIA DE USUARIO	
Número: 02	**Usuario:** Desarrollador
Nombre de la historia: Diseño de un boceto para la aplicación web	
Prioridad: Media	**Estimación:** S

Descripción: Como desarrollador de software, necesito diseñar un boceto del aplicativo para que sirva como guía de cómo se mostrará la aplicación en un futuro. En este boceto se verá un login, register y pantallas para el administrador que es de CRUD de usuarios y pantallas para el usuario normal un CRUD de publicaciones.
Criterio de aceptación: • Toda la información y pantallas deben ser claras para no crear confusión. • La funcionalidad debe ser clara y dar una experiencia buena al usuario.

Tabla 8. Historia de usuario 3

HISTORIA DE USUARIO	
Número: 03	**Usuario:** Administrador
Nombre de la historia: Back-end: Administrar los usuarios y roles	
Prioridad: Alta	**Estimación:** XL
Descripción: Como administrador, quiero que la aplicación web me proporcione las funcionalidades de crear, editar, visualizar, borrar usuarios y asignar roles a estos mismos. Además de hacer login y register de usuarios.	
Criterio de aceptación: • La aplicación web deberá mostrar de forma clara toda la información relevante de los usuarios. • La aplicación web permitirá realizar búsquedas de los usuarios por nombre o correo.	

Tabla 9. Historia de usuario 4

HISTORIA DE USUARIO	
Número: 04	**Usuario:** Usuario Normal
Nombre de la historia: Back-end: Administrar publicaciones	
Prioridad: Media	**Estimación:** L
Descripción: Como usuario normal, quiero que la aplicación web me proporcione las funcionalidades de crear, editar, visualizar, borrar publicaciones. Además de poder ver la información personal de sí mismo.	
Criterio de aceptación: • La aplicación web deberá mostrar de forma clara toda la información relevante del usuario.	

• La aplicación web aceptará realizar búsquedas de publicaciones por título. • La aplicación web aceptará realizar ver su propia información.

Tabla 10. Historia de usuario 5

HISTORIA DE USUARIO	
Número: 05	**Usuario:** Administrador
Nombre de la historia: Ver historial de acciones de los usuarios	
Prioridad: Media	**Estimación:** M
Descripción: Como administrador quiero que la aplicación me deba permitir ver datos del usuario como: tiempo de creación, actualización, reseteo de contraseña.	
Criterio de aceptación: • La aplicación web deberá mostrar de forma clara toda la información de los usuarios. • La aplicación web permitirá realizar búsquedas de los usuarios por nombre o correo.	

Tabla 11. Historia de usuario 6

HISTORIA DE USUARIO	
Número: 06	**Usuario:** Usuario Normal
Nombre de la historia: Búsquedas de publicaciones por fechas y título Í	
Prioridad: Baja	**Estimación:** M
Descripción: Como usuario normal, quiero que la aplicación me permita realizar búsquedas de sus publicaciones por fechas y título lo que permitirá más facilidad de encontrar sus publicaciones.	
Criterio de aceptación: • El usuario podrá ingresar una palabra o frase en un campo de texto para buscar publicaciones que contengan dicho texto en el título. • Los resultados de la búsqueda se mostrarán en una lista ordenada por fecha de publicación. • Cada resultado mostrará al menos: título, fecha de publicación y descripción de la publicación.	

Tabla 12. Historia de usuario 7

HISTORIA DE USUARIO

Número: 07	**Usuario:** Administrador
Nombre de la historia: Front-end: Realizar pantallas para el usuario administrador	
Prioridad: Alta	**Estimación:** XL
Descripción: Como administrador, quiero que las pantallas me muestren a todos los usuarios con su información respectiva y me permitan realizar acciones del CRUD en todos los usuarios. Además, también de poder asignar roles a estos usuarios.	
Criterio de aceptación: • El administrador podrá ver una lista de todos los usuarios registrados. • Cada usuario en la lista mostrará información básica, como nombre, correo electrónico, rol asignado. • El administrador podrá cambiar el rol de un usuario a "Administrador" o "Usuario Normal" desde la interfaz. • La pantalla incluirá un campo de búsqueda para localizar usuarios por nombre o correo electrónico.	

Tabla 13. Historia de usuario 8

HISTORIA DE USUARIO	
Número: 08	**Usuario:** Usuario Normal
Nombre de la historia: Frontend: Realizar pantallas para el usuario normal	
Prioridad: Alta	**Estimación:** XL
Descripción: Como usuario normal, quiero que la aplicación me permita ver mi información personal y pueda editarla. Además de poder hacer un CRUD en la cuestión de mis publicaciones.	
Criterio de aceptación: • El usuario puede acceder a una pantalla que muestra su información personal. • El usuario puede crear, visualizar, editar y eliminar publicaciones a través de un formulario. • La pantalla debe tener un diseño limpio y claro.	

Tabla 14. Historia de usuario 9

HISTORIA DE USUARIO	
Número: 09	**Usuario:** Desarrollador
Nombre de la historia: Implementar validaciones de seguridad	
Prioridad: Alta	**Estimación:** L

Descripción: Como desarrollador, quiero que la aplicación sea segura en la cuestión de contraseña esté cifrada, en caso de olvido de contraseña que esta la pueda recuperar de manera segura, al momento de hacer login también pida un código de verificación para poder ingresar a la cuenta, el diseño de la aplicación también debe ser seguro en este caso haciendo uso del patrón de diseño MVC, las api que cumplan con las seguridades, las herramientas que se usen este actuales y garanticen que no exista vulnerabilidades, centro de acceso es decir que solo un usuario pueda entrar a su perfil y nadie más, proteger credenciales importantes como: claves de Apis, conexiones de base de datos, puertos por donde salen tanto el back-end y Front-end, etc.
Criterio de aceptación: • Las contraseñas de los usuarios se almacenarán de forma segura en la base de datos utilizando un algoritmo de cifrado robusto como bcrypt. • La aplicación deberá contar con un flujo seguro para recuperación de contraseñas. • El diseño de la aplicación deberá seguir estrictamente el patrón MVC para garantizar una separación clara entre datos, lógica de negocio y presentación. • Cada usuario solo podrá acceder a su propio perfil y recursos asociados. • Todas las librerías, frameworks y dependencias utilizadas deberán estar actualizadas

Tabla 15. Historia de usuario 10

HISTORIA DE USUARIO	
Número: 10	**Usuario:** Desarrollador
Nombre de la historia: Conectar el Frontend y Backend de la aplicación web	
Prioridad: Media	**Estimación:** M
Descripción: Como desarrollador, quiero que unir ambos campos para poder ejecutar ya la aplicación web y así poder ver cómo va la aplicación y ver si necesita cambios o ajustes.	
Criterio de aceptación: • El frontend debe consumir correctamente las APIs del backend utilizando los endpoints definidos. • Todas las funcionalidades clave, como login, registro, visualización, edición y eliminación de datos, deben operar correctamente tras la conexión. • El frontend deberá manejar de forma adecuada los errores devueltos por el backend, mostrando mensajes claros al usuario.	

Tabla 16. Historia de usuario 11

HISTORIA DE USUARIO	
Número: 11	**Usuario:** Desarrollador
Nombre de la historia: Correcciones en Frontend y Backend	

Prioridad: Media	**Estimación:** L
Descripción: Como desarrollador, quiero corregir los errores o mejorar a ambos casos a que si no se lo hace la aplicación, puede que o cumpla con las funciones que tiene que realizar.	
Criterio de aceptación: • Los errores identificados deberán ser resueltos asegurando que las funciones principales de la aplicación (login, registro, CRUD, etc.) operen correctamente. • Se deberá revisar y mejorar el código tanto del front-end como del back-end para hacerlo más eficiente, legible y mantenible, eliminando redundancias y posibles vulnerabilidades. • Las correcciones en el front-end deberán mantener o mejorar la experiencia de usuario, asegurando que el diseño sea intuitivo.	

Tabla 17. Historia de usuario 12

HISTORIA DE USUARIO	
Número: 12	**Usuario:** Administrador y Usuario normal
Nombre de la historia: Probar la aplicación web	
Prioridad: Baja	**Estimación:** S
Descripción: • Como administrador, quiero comprobar que las funciones que me correspondan funciones como tal y si existe algún percance poder avisar para corregirlo. • Como usuario normal, quiero comprobar que mis funciones también cumplan con las necesidades mías y brindar ayuda si existe algo malo para poder corregirlo.	
Criterio de aceptación: • El administrador deberá verificar que puede realizar las funciones asignadas, como: Crear, leer, actualizar y eliminar usuarios. Asignar roles a los usuarios. • La interfaz deberá ser intuitiva y permitir al administrador navegar y realizar acciones fácilmente. • El usuario normal deberá comprobar que puede: Crear, leer, actualizar y eliminar sus propias publicaciones, editar su información personal.	

2.3 Desarrollo de los Sprints

Una vez definidos roles, requerimientos y el cronograma para la creación del aplicativo, se inicia la etapa de desarrollo. En esta fase se estructura en periodos cortos, “sprints”, donde se colocan las actividades ya planificadas para lograr las metas establecidas. Al finalizar cada sprint, se evalúa el avance y se aplican las mejoras en base de retroalimentación recibida, lo que permite un desarrollo continuo del proyecto.

En la tabla 18 se presenta la información del desarrollo del sprint en relación con las historias de usuario presentes en el producto backlog.

Tabla 18. Planificación de Sprints

ID	Historia de Usuario	Sprint	Fecha Inicio	Fecha Fin	Entregable
01	Identificar las necesidades	1	10-09-2024	22-09-2024	Listado con los requisitos y necesidades del cliente.
02	Diseño de un boceto para la aplicación web	1	10-09-2024	22-09-2024	Boceto que muestra las pantallas de administrador y usuario, login, register.
03	Back-end: Administrar los usuarios y roles	2	23-09-204	11-11-2024	El back-end el usuario administrado podrá hacer su CRUD de usuarios y administrar roles.
04	Back-end: Administrar las publicaciones	2	12-11-2024	04-12-2024	El back-end con el usuario normal puede hacer su CRUD de publicaciones
05	Ver historial de acciones de los usuarios	2	05-12-2024	11-12-2024	El back-end con el usuario administrador me permitiría ver la actividad y datos de los usuarios
06	Búsquedas de publicaciones por título y fecha.	3	12-12-2024	18-12-2024	El back-end con el usuario normal me permite la búsqueda de publicaciones por título y fecha.
07	Front-end: Realizar pantallas para el administrador	3	06-01-2025	13-01-2025	El Front-end con el usuario administrador podrá realizar las acciones de CRUD, pero de manera visual de usuarios.
08	Front-end: Realizar pantallas para el usuario normal	4	14-01-2025	23-01-2025	El Front-end con el usuario normal pude realizar las acciones de CRUD, pero de manera visual de sus publicaciones.
09	Implementar validaciones de seguridad	4	27-01-2025	04-02-2025	La aplicación consta con las seguridades

					para mantener los datos de los usuarios.
10	Conectar el Frontend y Back-end de la aplicación web	4	05-02-2025	06-02-2025	Unión de back-end y front-end para su total funcionamiento.
11	Correcciones en Back-end y Frontend	5	06-02-2025	11-02-2025	Corrección de error encontrados.
12	Probar la aplicación web	5	11-02-2025	13-02-2025	Comprobar que la aplicación funciona correctamente.

Nota: En la tabla 18 se muestra la información del desarrollo de los sprints en concordancia con las historias de usuario.

El desarrollo de la solución se realizó utilizando de una manera ágil e iterativa, fundamentado en la metodología Scrum. Cada sprint tenía como meta terminar las historias de usuario que cumplían con los requisitos de la solución.

Detalle de herramientas

En esta parte se va a hablar más detalladamente sobre las herramientas seleccionadas a utilizar en el Back-end y Front-end.

Backend

Para el desarrollo del back-end, se eligieron las siguientes herramientas que complementan al back-end:

- **Express:** Framework principal a usar para la aplicación web y APIs.
- **Bcryptjs:** Se utilizará para garantizar la seguridad de las contraseñas de los usuarios.
- **Cookie-parser:** Es un middelware de Express que usamos para analizar las cookies de los de las solicitudes HTTP. Permite acceder a las cookies enviadas por el cliente de manera sencilla [62].
- **JWT (jsonwebtoken):** Se utilizará para la autorización y autentificación de los usuarios mediante JSON Web Tokens.
- **Cors:** Facilita la comunicación que se va a tener entre en back-end y el Front-end.
- **Helmet:** Ajusta las cabeceras que estén en HTTP.

- **Nodemailer:** Es un módulo para aplicaciones Node.js que facilita el envío de correos electrónicos [63].
- **Zod:** Esta es una biblioteca que permitirá declaración y validación de esquemas (cualquier tipo de dato).

Frontend

En el Front-end, las siguientes herramientas fueron seleccionadas para que una interfaz sencilla.

- **React:** Librería fundamental para el front-end, ya que permite gestionar el estado de forma eficiente y desarrollar componentes reutilizables.
- **Axios:** Facilita la comunicación con el back-end.
- **React-Router-Dom:** Es esencial para una navegación fluida entre las distintas rutas del gestor documental.
- **Tailwind:** Es un framework CSS de diseño basado en utilidades que permite crear interfaces de usuario de manera rápida y eficiente [64].
- **Vite:** Es una herramienta que tiene como objetivo proporcionar una experiencia de desarrollo más rápida y ágil [65].
- **Recharts:** Es una biblioteca de gráficos para React**,** basada en D3.js**,** que permite crear visualizaciones de datos de manera sencilla y personalizable [66].
- **React-big.calendar:** Es una biblioteca para React que permite integrar un calendario interactivo con opciones avanzadas de visualización y gestión de eventos [67].
- **React-toastify:** Es una potente biblioteca para añadir notificaciones elegantes y personalizables a las aplicaciones React [68].

2.4 Implementación

Después de completar las actividades programadas en los sprints y terminar con el desarrollo de la aplicación web, se confirmó que el aplicativo cumple con los requisitos establecidos. A continuación, se presenta varias imágenes que demuestran el aplicativo terminado exitosamente.

2.5 Codificación segura según Owasp

Validación de entradas

En usuarios normal en la parte de crear publicaciones se valida de la siguiente manera. Con esto evitamos problemas de las vulnerabilidades de inyección sql, cross site scripting (XSS). Para esto usamos en model y controller.

Figura 7

Consulta parametrizada en el model

```
import { query, text } from 'express'
import { conpool } from '../db.js'

//bien
const createPublicacion = async (titulo, descripcion, fecha_inicio, fecha_fin, id_usuario, estado = "Iniciado", prioridad) => {
    const estadosValidos = ["Iniciado", "Pendiente", "Terminado"];
    const prioridadesValidas = ["Alta", "Media", "Baja"];

    if (!estadosValidos.includes(estado)) {
        throw new Error("Estado no válido. Debe ser 'iniciado', 'pendiente' o 'terminado'.");
    }

    if (prioridad && !prioridadesValidas.includes(prioridad)) {
        throw new Error("Prioridad no válida. Debe ser 'Alta', 'Media' o 'Baja'.");
    }

    const query = {
        text: `
            INSERT INTO PUBLICACIONES (titulo, descripcion, fecha_inicio, fecha_fin, id_usuario, estado, prioridad)
            VALUES ($1, $2, $3, $4, $5, $6, $7)
            RETURNING titulo, descripcion, fecha_inicio, fecha_fin, id_usuario, estado, prioridad
        `,
        values: [titulo, descripcion, fecha_inicio, fecha_fin, id_usuario, estado, prioridad]
    };
    const { rows } = await conpool.query(query)
    return rows[0]
}
```

Nota. En la figura se muestra como al hacer consulta sql esta debe estar parametrizada ($1,$2,$3) y esto si cumplimos. Autoría propia

Figura 8

Campos con validaciones en el controller

```
//todo bien
const createPublicacion = async (req, res) => {
    try {
        const { titulo, descripcion, fecha_inicio, fecha_fin, estado, prioridad } = req.body;

        if (!titulo || !descripcion || !fecha_inicio || !fecha_fin) {
            return res.status(400).json({ ok: false, msg: "Error: título, descripción o fechas vacíos" });
        }

        const tituloLimpio = sanitize(titulo, { allowedTags: [], allowedAttributes: {} });
        const descripcionLimpia = sanitize(descripcion, { allowedTags: [], allowedAttributes: {} });

        const estadosValidos = ["Iniciado", "Pendiente", "Terminado"];
        const estadoFinal = estado || "inicio";
        if (!estadosValidos.includes(estadoFinal)) {
            return res.status(400).json({ ok: false, msg: "Estado no válido." });
        }

        const prioridadesValidas = ["Alta", "Media", "Baja"];
        if (prioridad && !prioridadesValidas.includes(prioridad)) {
            return res.status(400).json({ ok: false, msg: "Prioridad no válida." });
        }

        const nuevaPublicacion = await PublicacionModel.createPublicacion(
            tituloLimpio,
            descripcionLimpia,
            fecha_inicio,
            fecha_fin,
            req.id_usuario,
            estadoFinal,
            prioridad
        );

        return res.status(201).json({ ok: true, msg: nuevaPublicacion });

    } catch (error) {
        return res.status(500).json({ ok: false, msg: error.message });
```

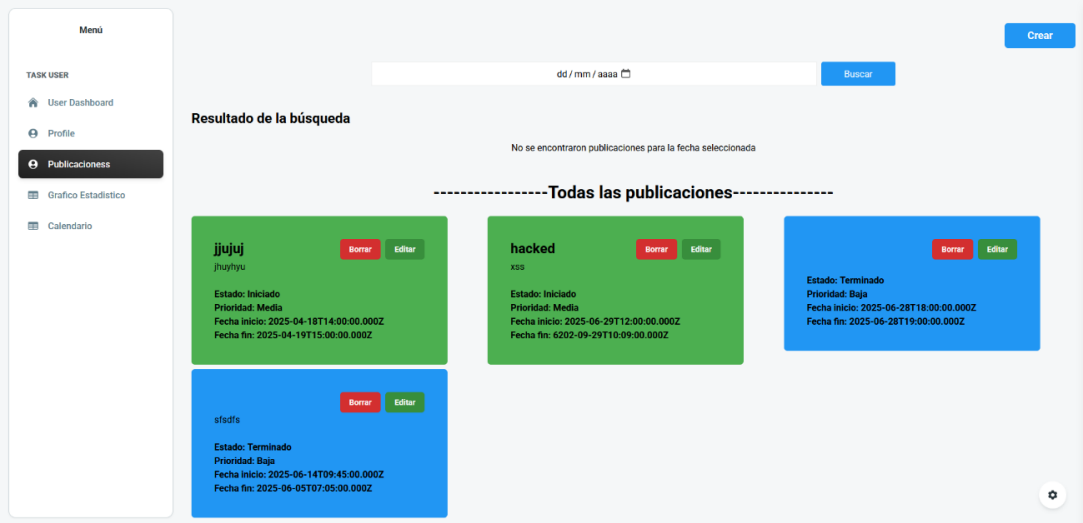
Nota. En la figura se muestra los campos del login controlados con todas las validaciones necesarias en el controller. Autoría propia

Y para evitar la vulnerabilidad XSS.

Se instalo la librería sanitize-hmtl la que nos permite limpiar entradas HTML y no deja que ponga texto que pueda dar alertas de emergencia eso en backe-nd y en Front-end al dar esto “<script>alert("Hacked!")</script>” el texto saldrá vacion que es el objetivo.

Figura 9

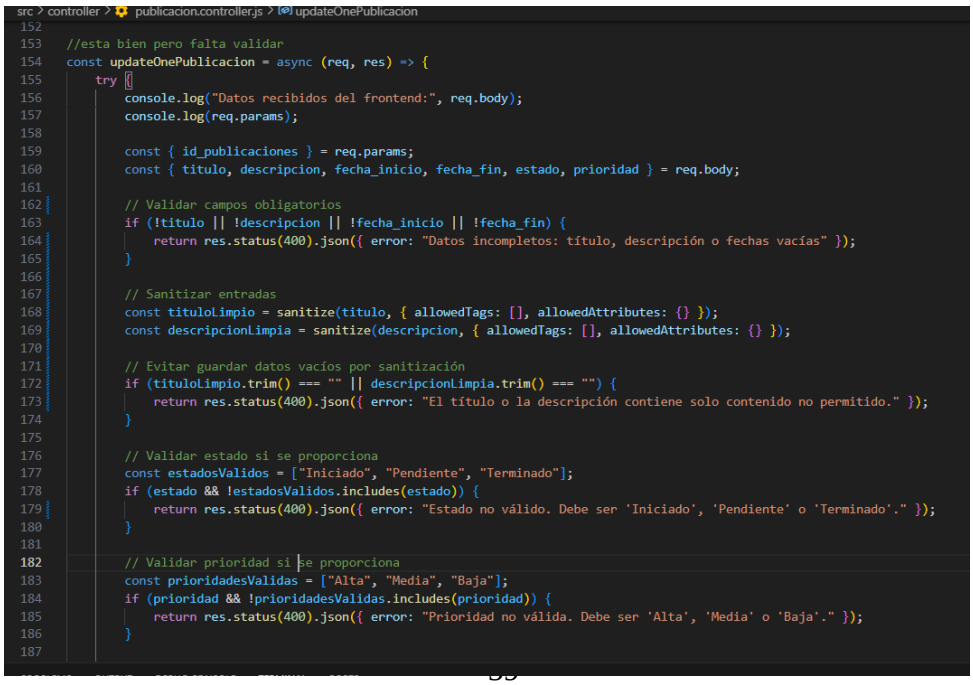
Pantalla de cómo se visualiza el control de entradas html



Nota. En la figura se muestra la pantalla de inicio del usuario normal donde se muestra las publicaciones del usuario. Autoría propia

Figura 10

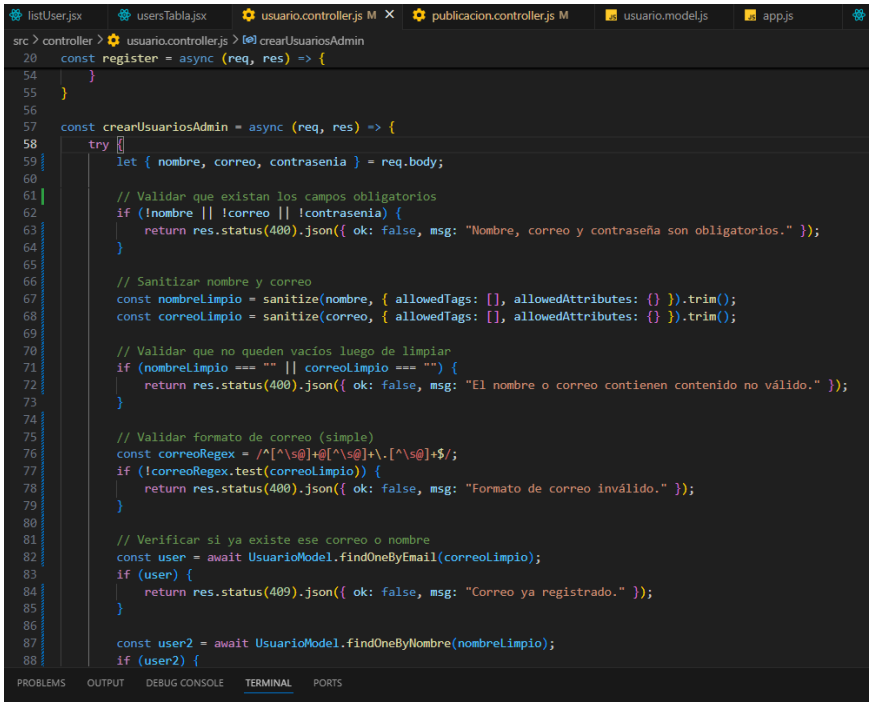
Control de entrada la función de editar publicaciones



Nota. En la figura se muestra la función de editar publicaciones en el controller. Autoría propia

Figura 11

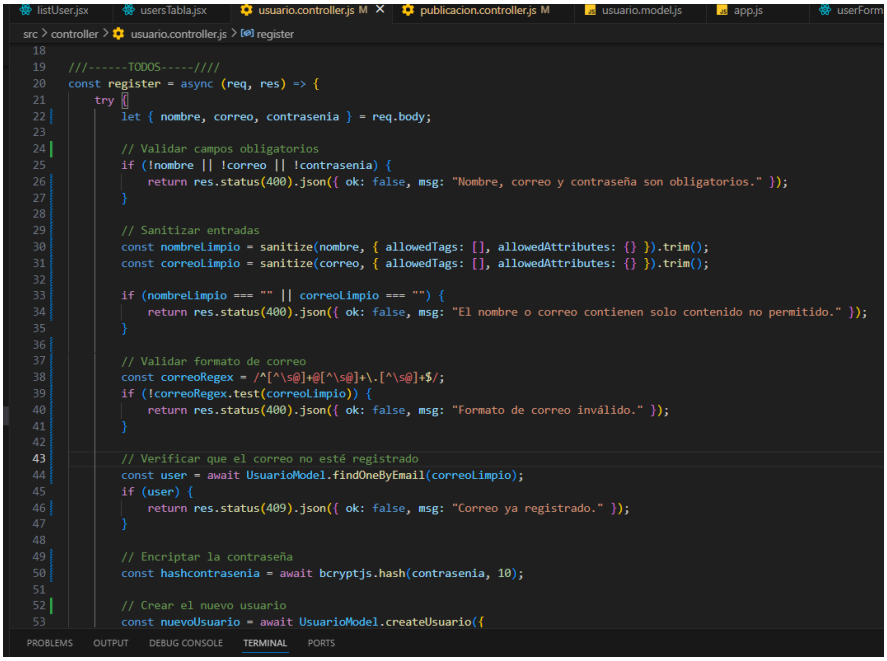
Control de entradas en la creación de usuarios



Nota. En la figura se muestra la función de creación de usuarios nuevos. Autoría propia

Figura 12

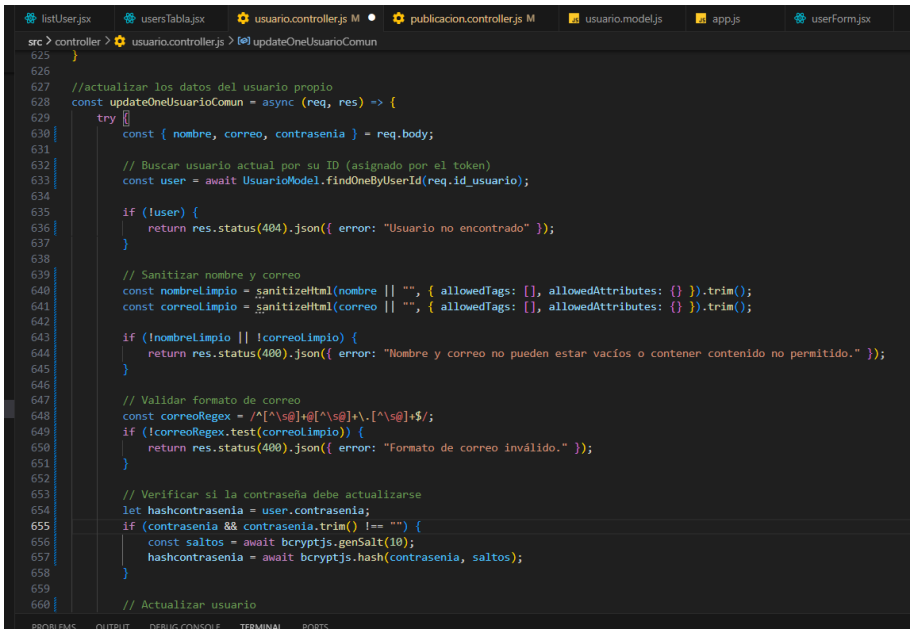
Control de entradas en la función registro de nuevos usuarios



Nota. En la figura se muestra la función de nuevos usuarios cuando no se ha registrado. Autoría propia

Figura 13

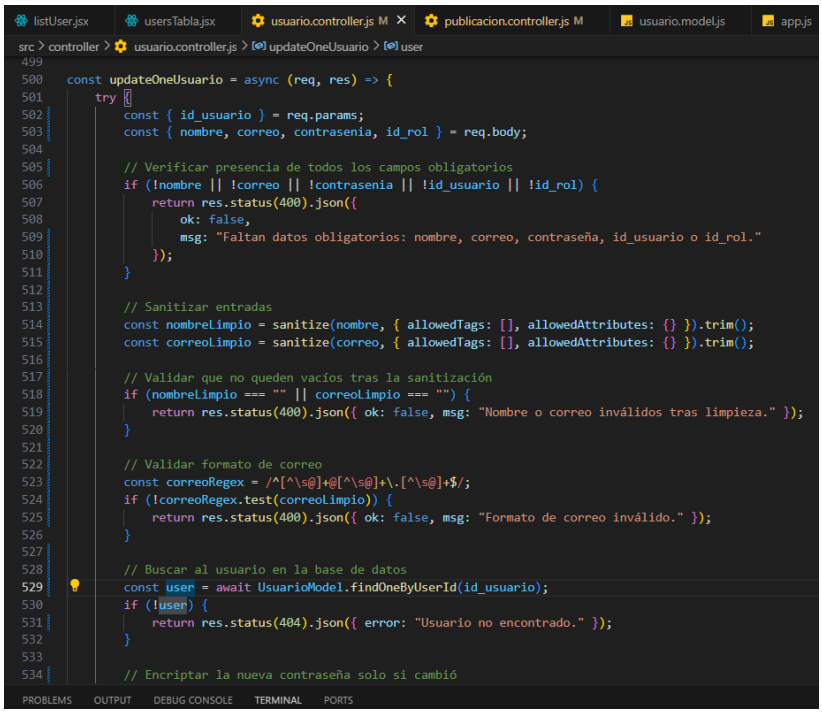
Control de entrada en la función de actualización de datos



Nota. En la figura se muestra la función de actualización de datos propios del usuario donde cambia los datos que sean necesarios . Autoría propia

Figura 14

Control de entrada en la función de actualización de datos por el administrador

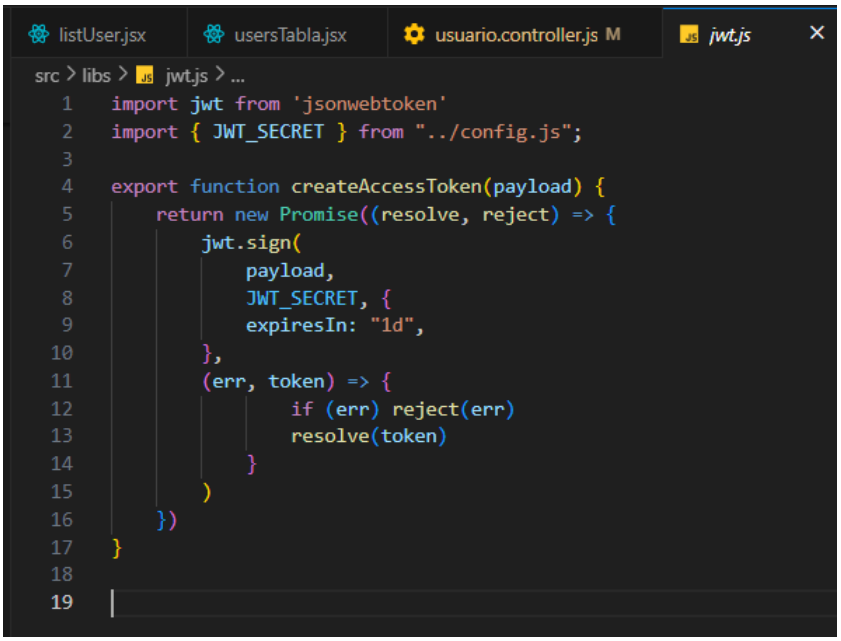


Nota. En la figura se muestra la función de actualizar el usuario por el administrador. Autoría propia

Gestión de sesiones

Figura 15

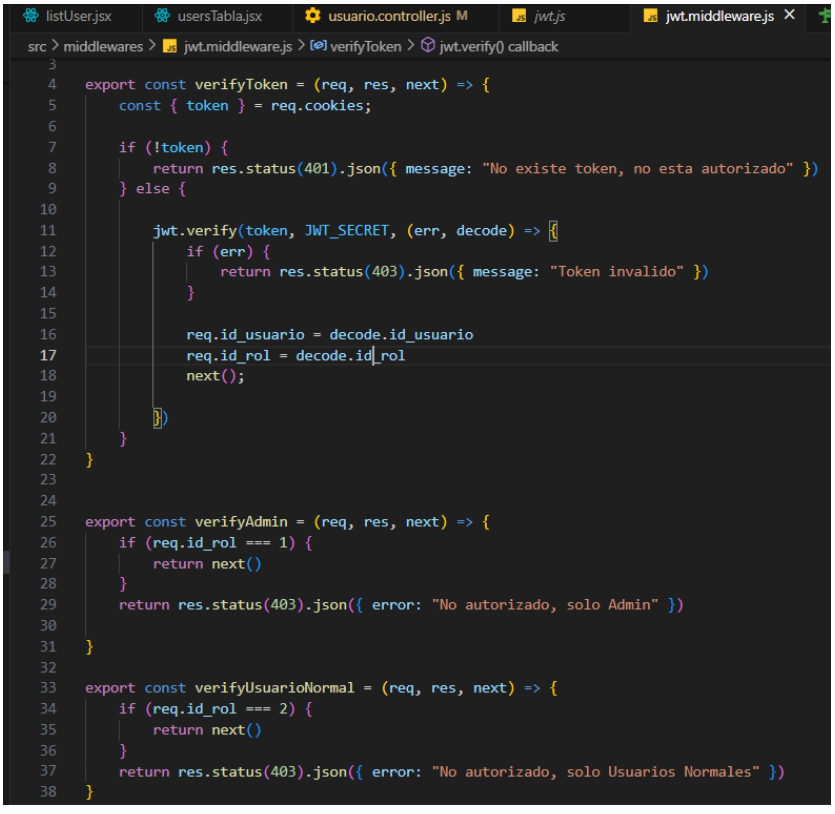
Función de creación de token



Nota. En la figura se muestra el uso jwt para la creación del token para las cuentas de cada usuario. Autoría propia

Figura 16

Middleware para verificar token y roles de los usuarios

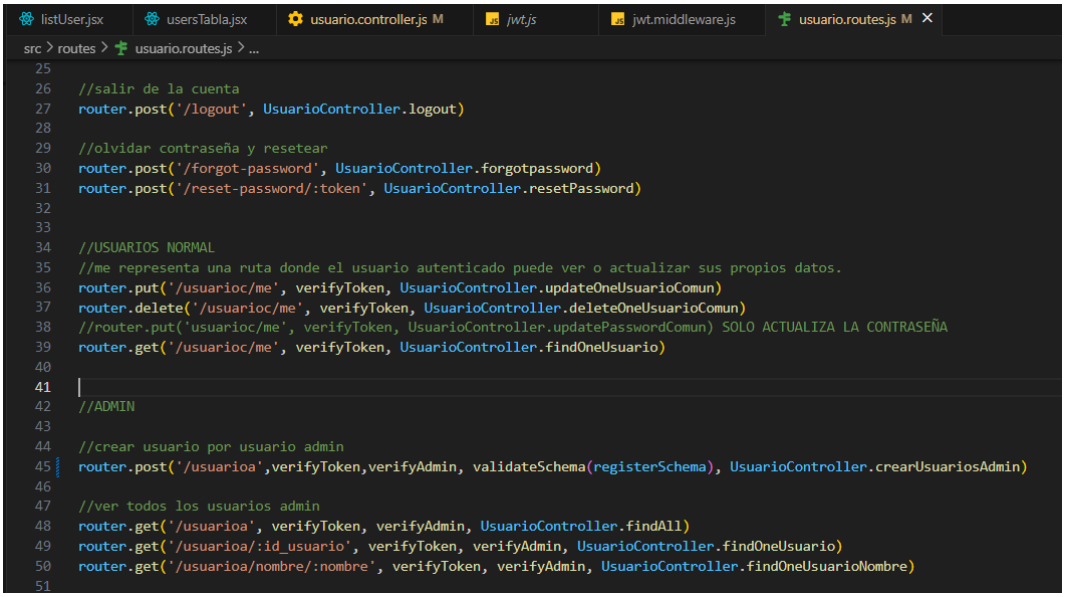


Nota. En la figura se muestra el middleware para verificar el token y verificar los roles para que cada rol tenga sus rutas y accesos necesarios sin afectar otros usuarios. Autoría propia

Control de acceso

Figura 17

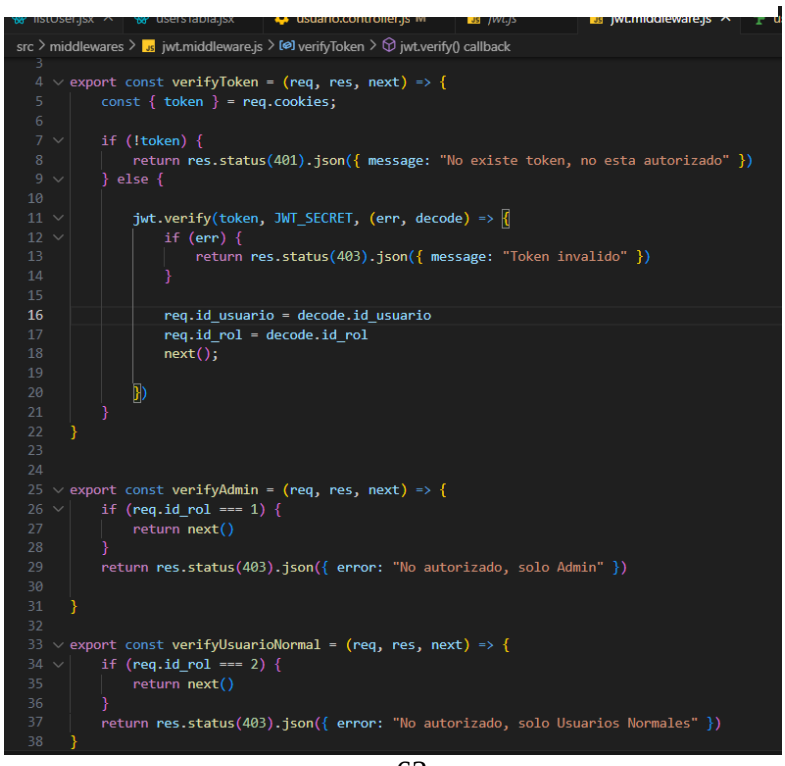
Rutas donde se coloca el middleware para verificar los usuarios y sus respectivas funciones



Nota. En la figura se muestra como el punto anterior en el middlewara, ya que con él se restringe los roles y funciones que tiene cada uno como se muestra. Autoría propia

Figura 18

Middleware de verificación de token roles de usuarios

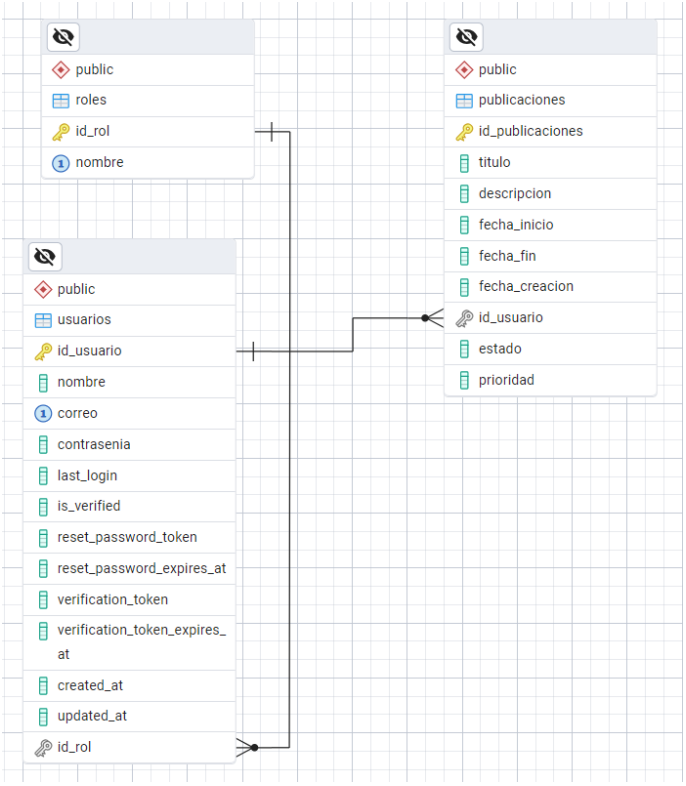


Nota. En la figura se muestra cómo se comprueba el token y el rol del usuario y evitar que se acceda información de otros usuarios. Autoría propia

Base de datos

Figura 19

Modelo de Base de Datos



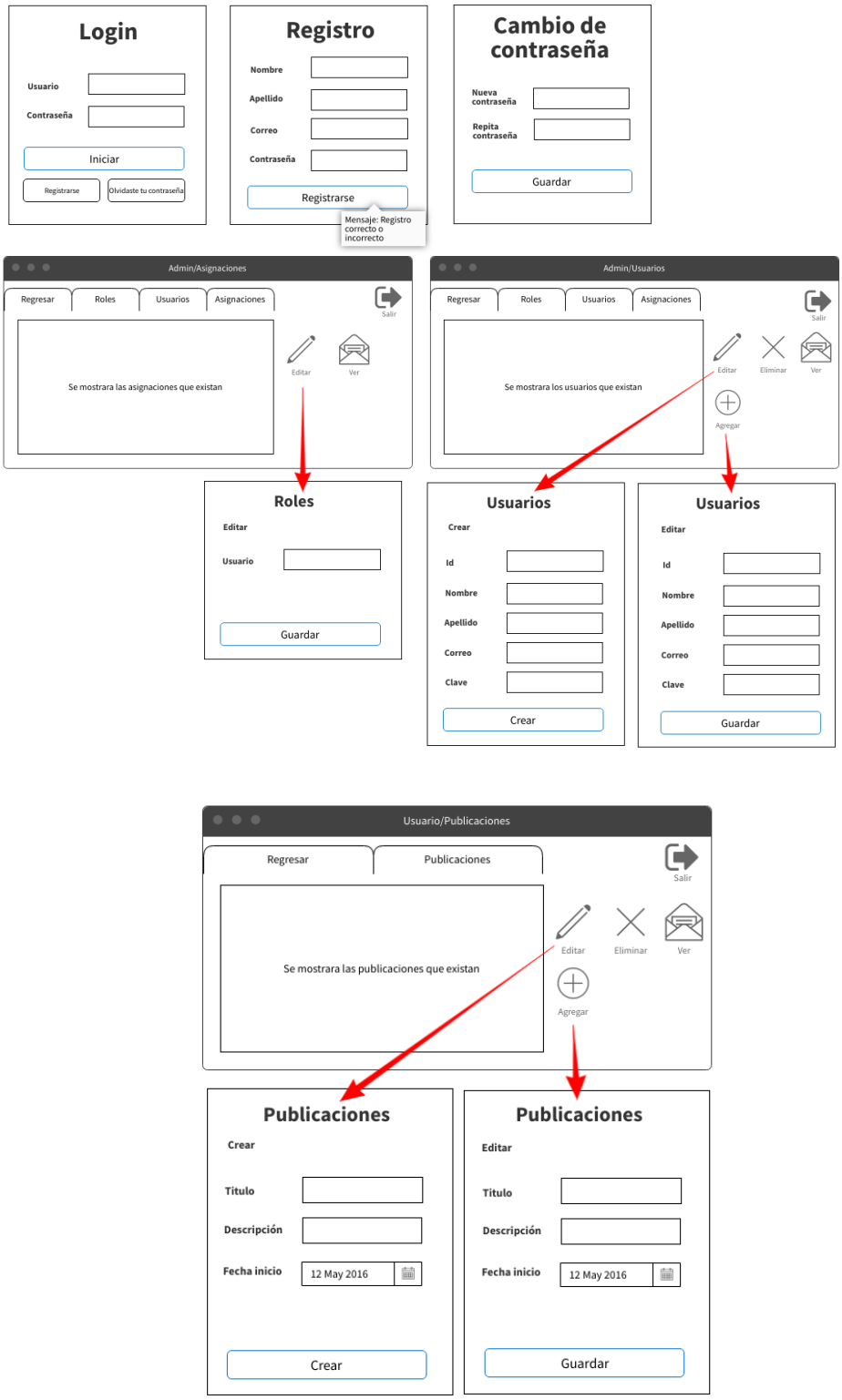
Nota. En la figura se muestra la base de datos que se va a usar. Autoría propia

En el desarrollo de la aplicación web se buscó organizar actividades y ayudar en la eficiencia en las actividades que se tengan que realizar. Además, se demostró que el uso de patrones de diseño y la aplicación de normas de seguridad permitieron realizar software seguro. Se establecieron dos perfiles de usuario que interactúan con el aplicativo. Estos perfiles incluyen: el usuario administrador encargado de gestionar a los usuarios y el usuario común que realizar actividades de elaborar publicaciones que necesite para gestionar sus actividades y su información personal. Además, cada usuario contó con interfaces específicas que le permitieron realizar sus tareas de manera eficiente.

Boceto

Figura 20

Boceto de Aplicación web

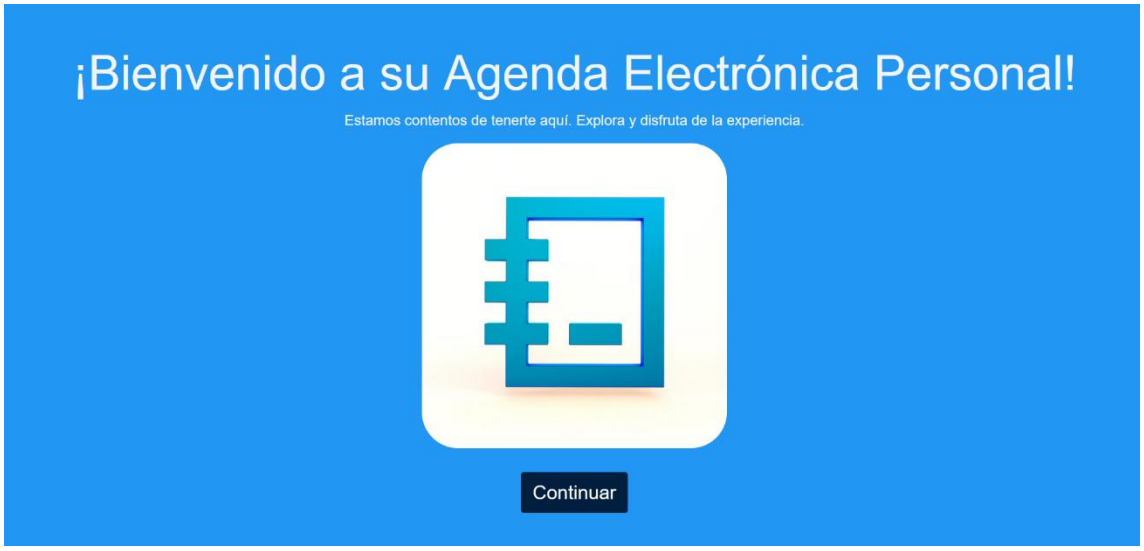


Nota. En esta figura se muestra el boceto de cómo es la aplicación web. Autoría propia

Usuarios

Figura 21

Bienvenida para los usuarios

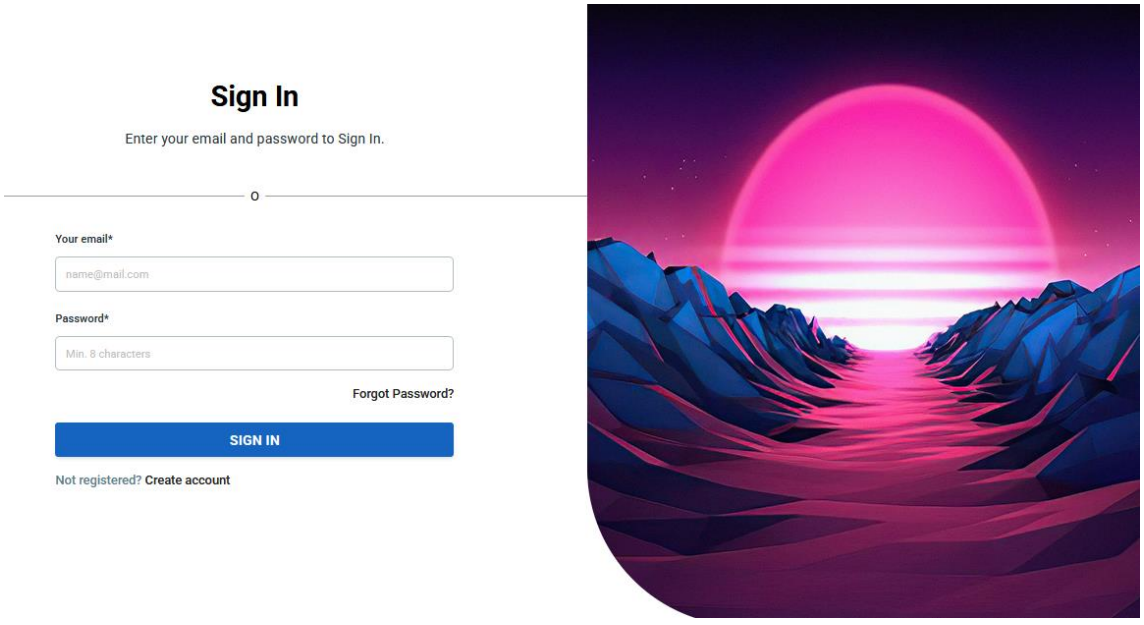


Nota. En esta figura se muestra la pantalla de bienvenida para todos los usuarios de la aplicación.

Autoría propia

Figura 22

Login de Usuarios

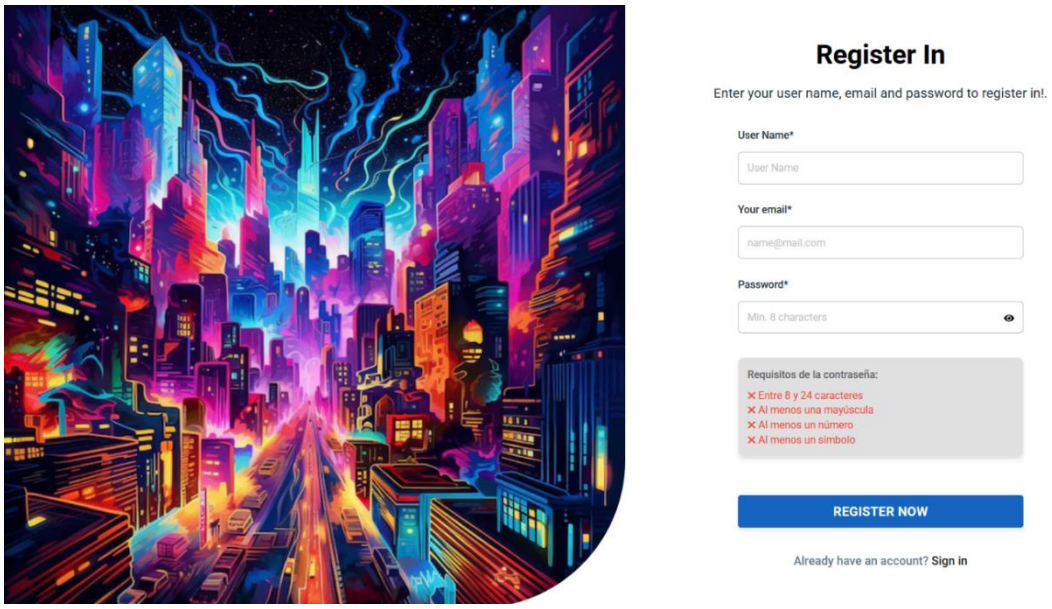


Nota. En esta figura se muestra la pantalla de login de usuarios para poder entrar a la aplicación .

Autoría propia

Figura 23

Registro de Usuarios



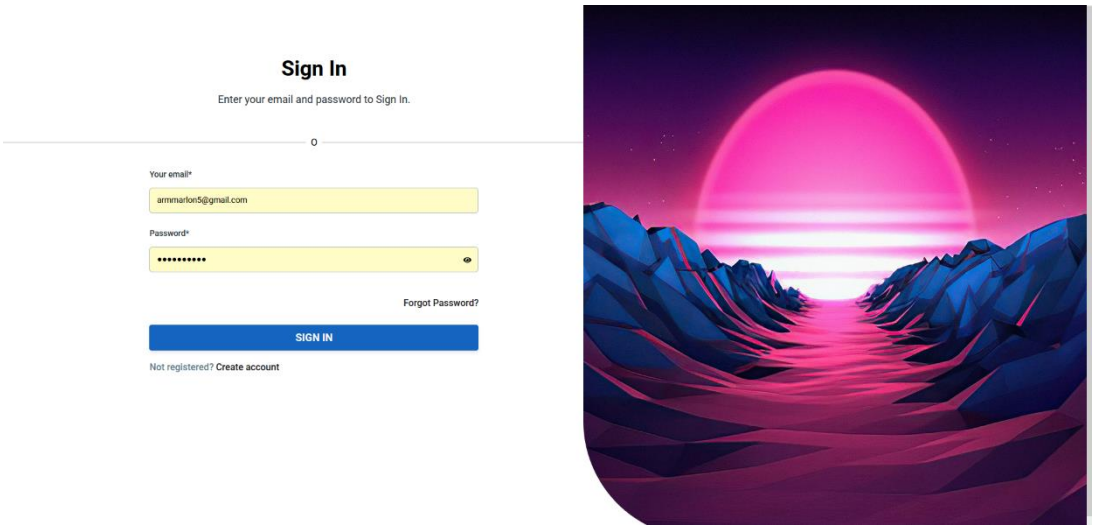
Nota. En esta figura se muestra la pantalla de registro de usuarios nuevos. Autoría propia

Usuario Administrador

El usuario administrador, como se puede observar en las siguientes imágenes, es el encargado de gestionar a los usuarios desde (crear, editar, ver, eliminar).

Figura 24

Ingreso de datos del Usuario Administrador



Nota. En esta figura se muestra el login para el usuario administrador. Autoría propia

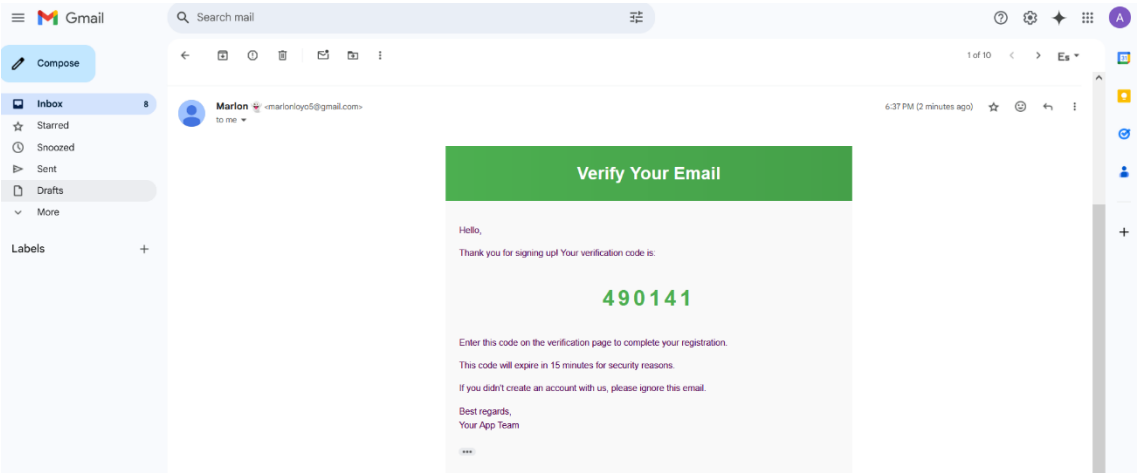
Figura 25

Validación de código para ingresar a la cuenta

Nota. En esta figura se muestra la validación de un código para el ingreso a la cuenta del usuario al momento de ingresar a su cuenta. Autoría propia

Figura 26

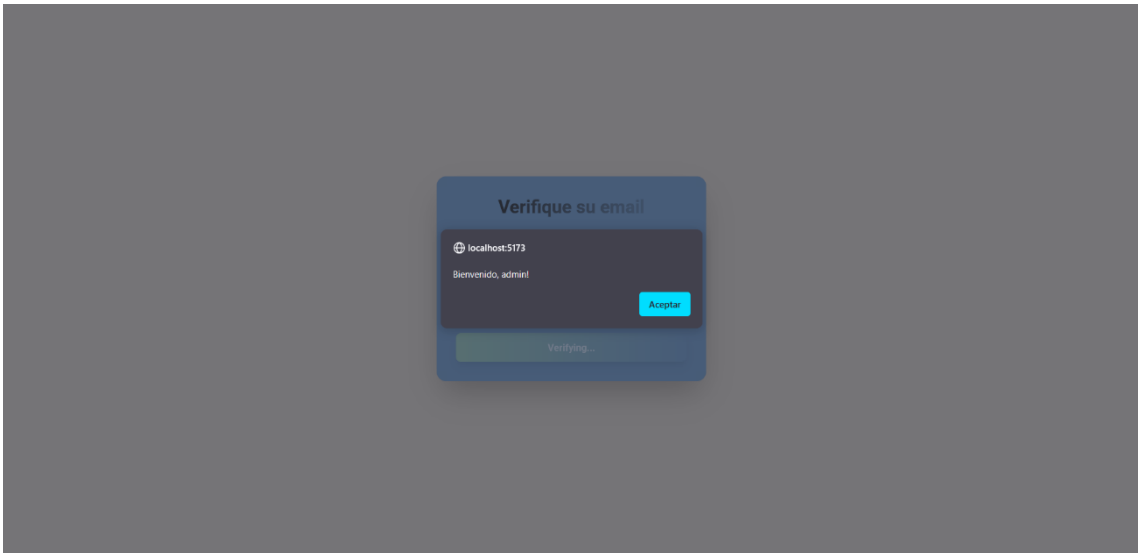
Código para validar que se envía al correo (Gmail)



Nota. En esta figura se muestra el código para la validación de la cuenta para iniciar sesión que se envía al correo (Gmail) de usuario y este puede ingresar a su cuenta. Autoría propia

Figura 27

El código es correcto y puede ingresar a la cuenta



Nota. En esta figura se muestra que la validación es correcta y deja ingresar al usuario a su cuenta. Autoría propia

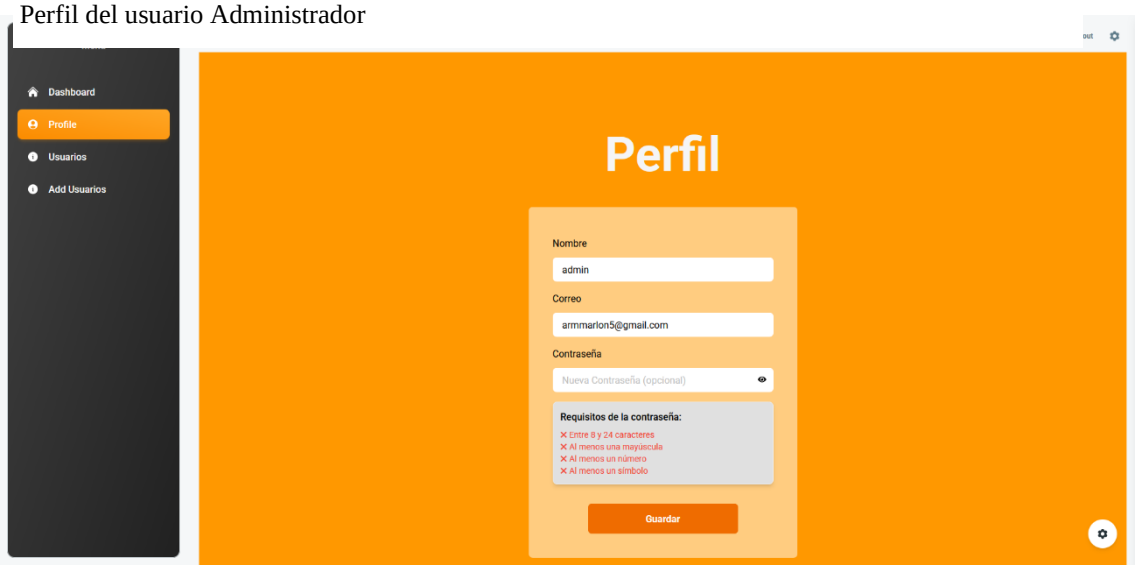
Figura 28

Pantalla de inicio del administrador

Nota. En esta figura se muestra la pantalla de inicio para el usuario administrador donde se da la bienvenida y se ve el gráfico estadístico que muestra las publicaciones en general de todos los usuarios. Autoría propia

Figura 29

Perfil del usuario Administrador



Nota. En esta figura se muestra la pantalla para editar el perfil del usuario administrador. Autoría propia

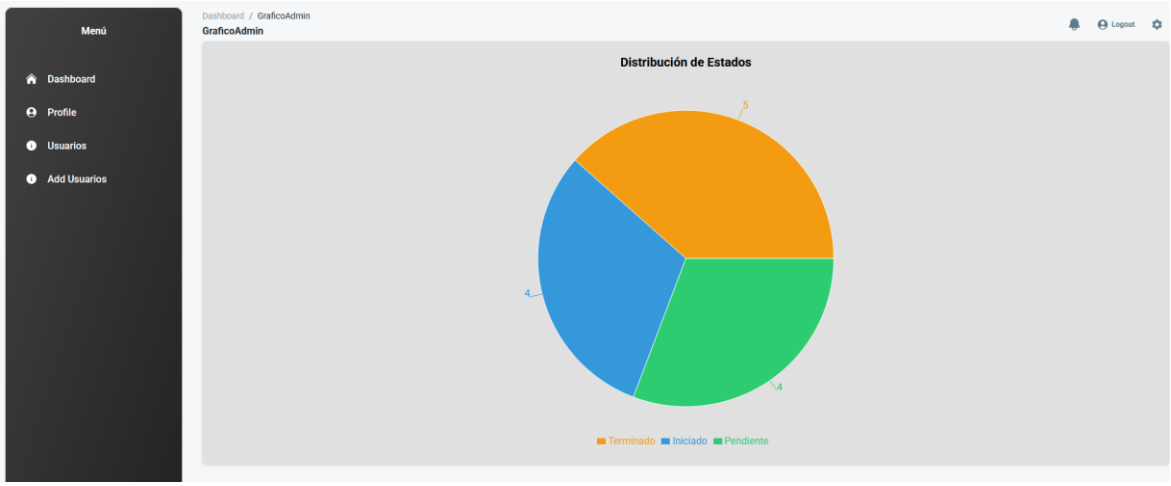
Figura 30

Vista de todos los usuarios.

Nota. En esta figura se muestra la pantalla de usuarios donde se pueden realizar las acciones de crear, editar y eliminar usuarios y búsqueda de estos mismos y además de ver un gráfico estadístico de cada usuario sobre las publicaciones. Autoría propia

Figura 31

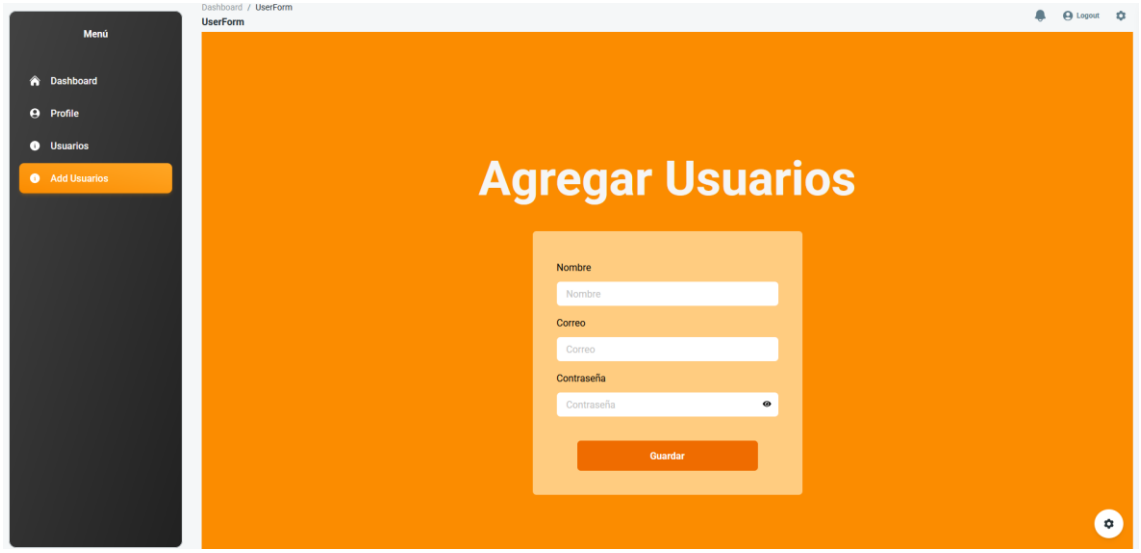
Gráfico estadístico de las publicaciones de un usuario



Nota. En esta figura se muestra el complemento de la anterior donde el usuario administrado puede ver un gráfico estadístico de las publicaciones que tiene el usuario y estas se dividen en publicaciones: iniciado, pendiente y terminado. Autoría propia

Figura 32

Vista para crear un nuevo usuario por el administrador



Nota. En esta figura se muestra la pantalla para crear el usuario nuevo como usuario administrador con los campos de nombre, correo, contraseña y rol. Autoría propia

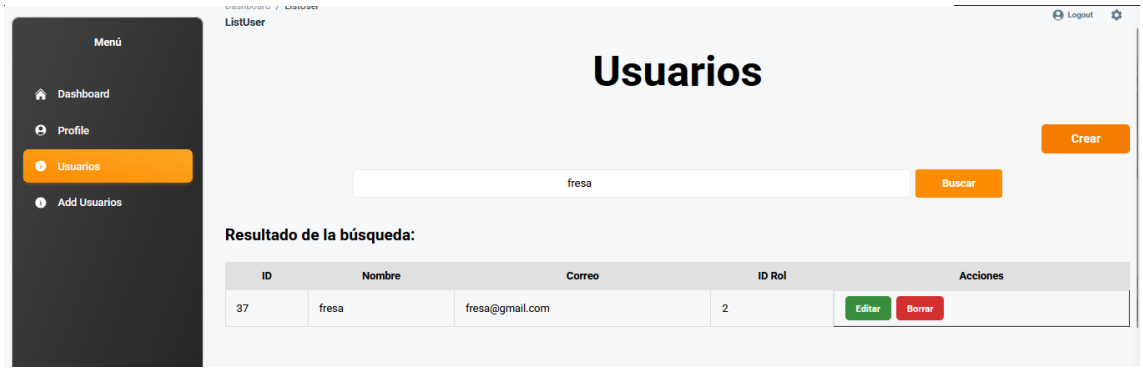
Figura 33

Vista para editar un usuario por el administrador

Nota. En esta figura se muestra la pantalla para editar un usuario seleccionado. Autoría propia

Figura 34

Búsqueda de un usuario por el nombre



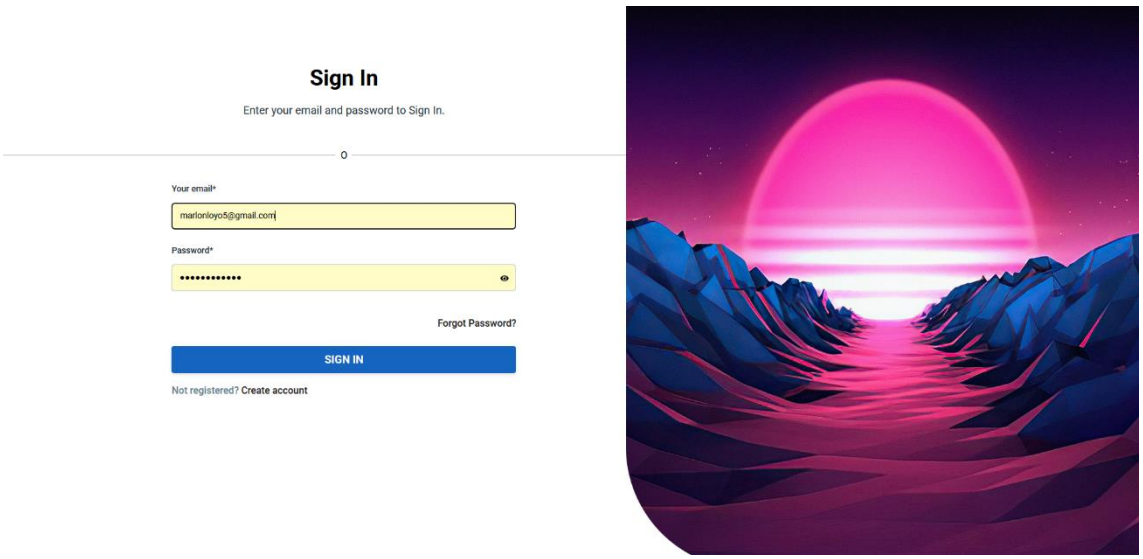
Nota. En esta figura se muestra la opción de búsqueda del usuario por el nombre. Autoría propia

Usuario común

El usuario común, como se puede observar en las siguientes imágenes, es el encargado de gestionar sus publicaciones e información personal.

Figura 35

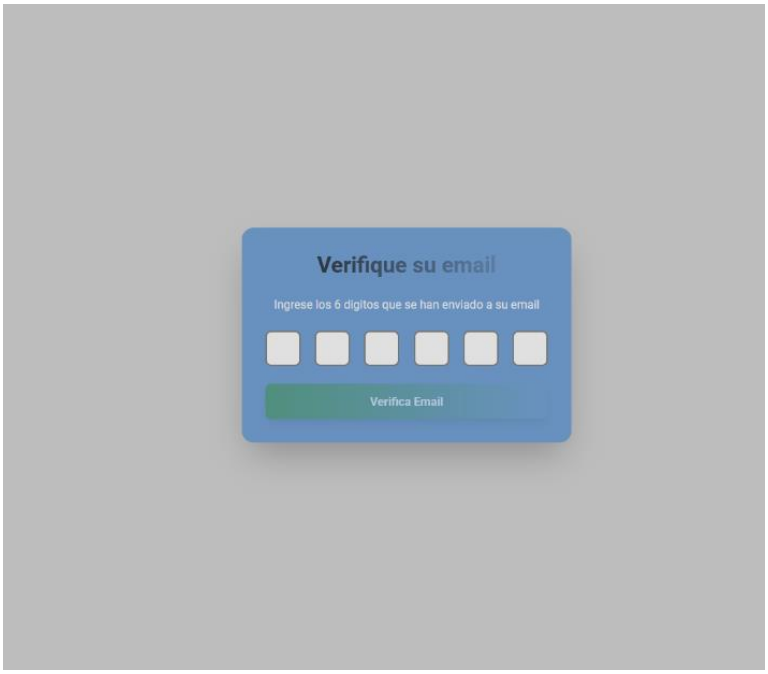
Vista del Login para usuario normal



Nota. En esta figura se muestra el login para el usuario normal. Autoría propia

Figura 36

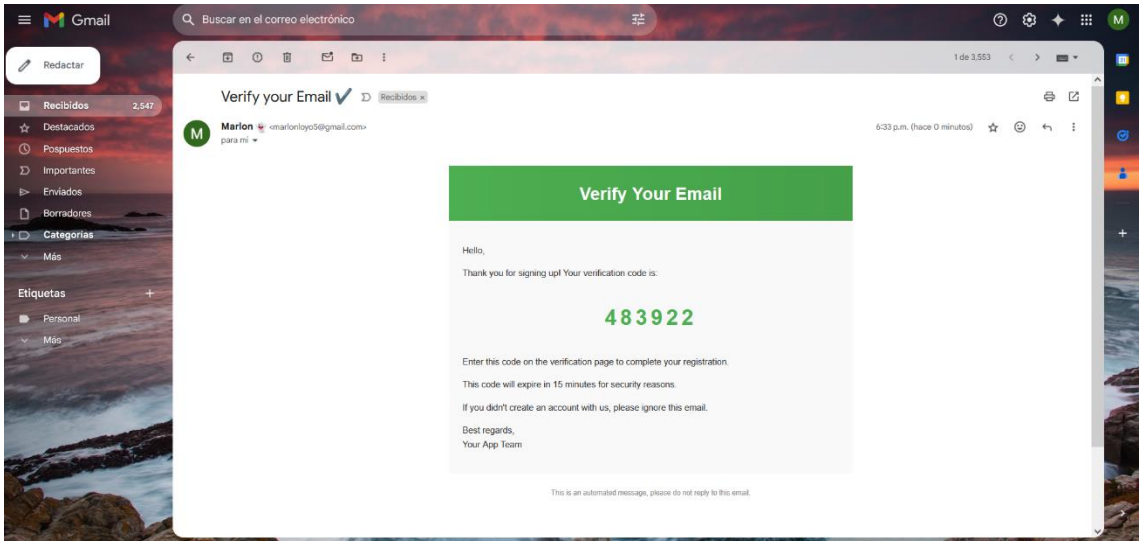
Validación de código para ingresar a la cuenta del usuario normal



Nota. En esta figura se muestra la validación de un código para el ingreso a la cuenta del usuario. Autoría propia

Figura 37

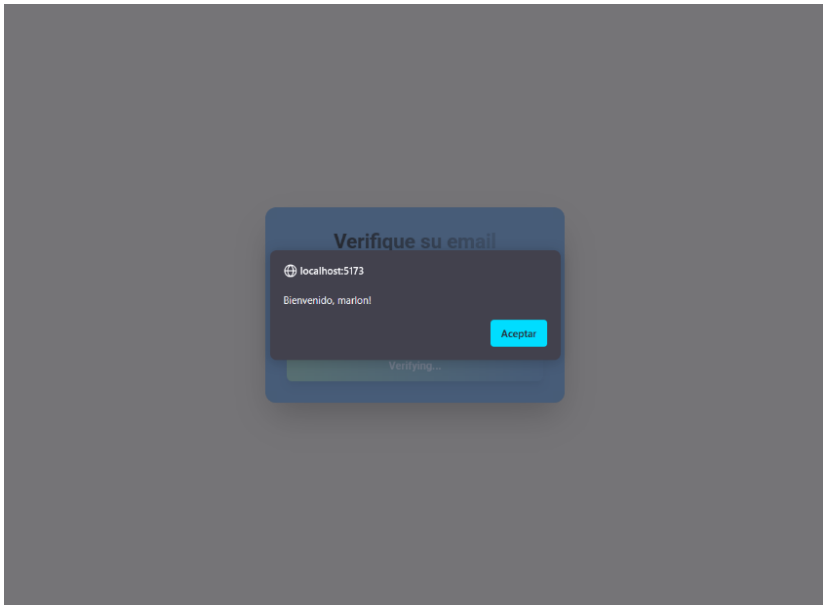
Código para validar que se envía al correo (Gmail) usuario normal



Nota. En esta figura se muestra el código para la validación de la cuenta para iniciar sesión que se envía al correo de usuario y este puede ingresar. Autoría propia

Figura 38

El código es correcto y puede ingresar a la cuenta del usuario normal



Nota. En esta figura se muestra que la validación es correcta o no y deja ingresar al usuario a su cuenta. Autoría propia

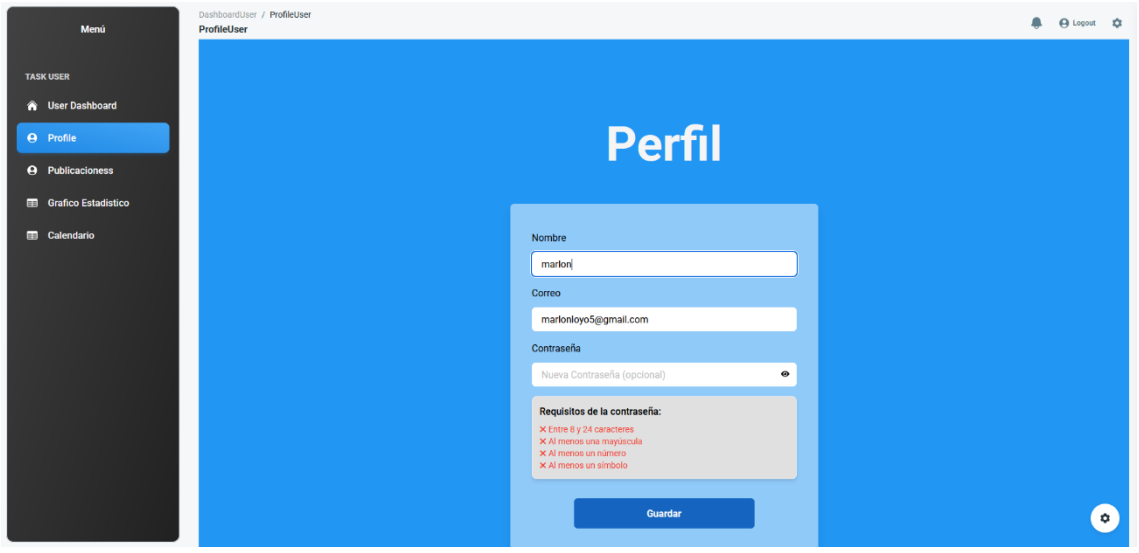
Figura 39

Pantalla de inicio para el usuario normal

Nota. En esta figura se muestra la pantalla de bienvenida al usuario común y un gráfico general de las publicaciones propias del usuario. Autoría propia

Figura 40

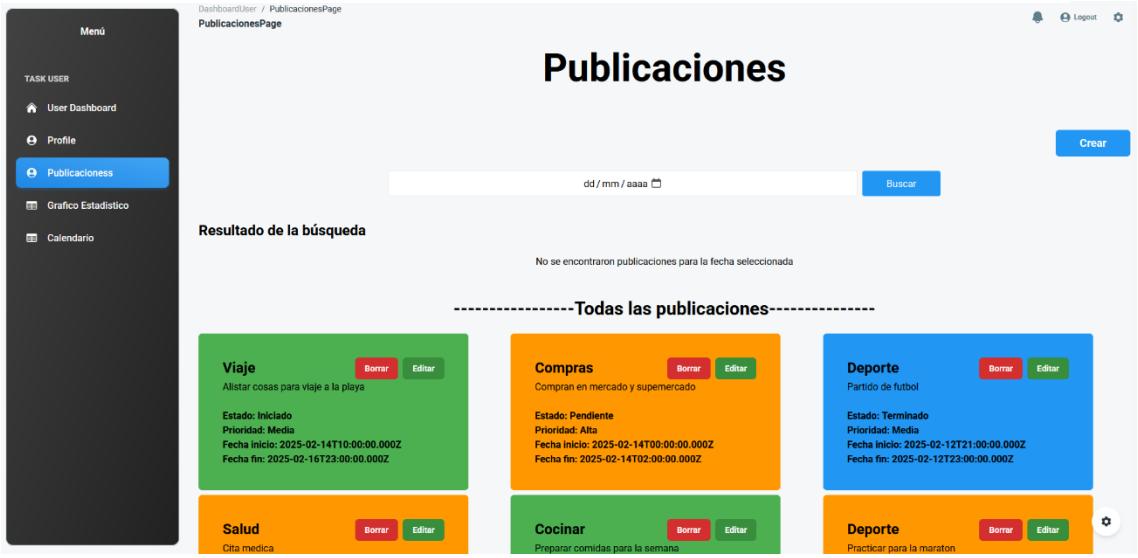
Pantalla para ver el perfil del usuario normal



Nota. En esta figura se muestra el perfil del usuario y la edición de los datos de este mismo. Autoría propia

Figura 41

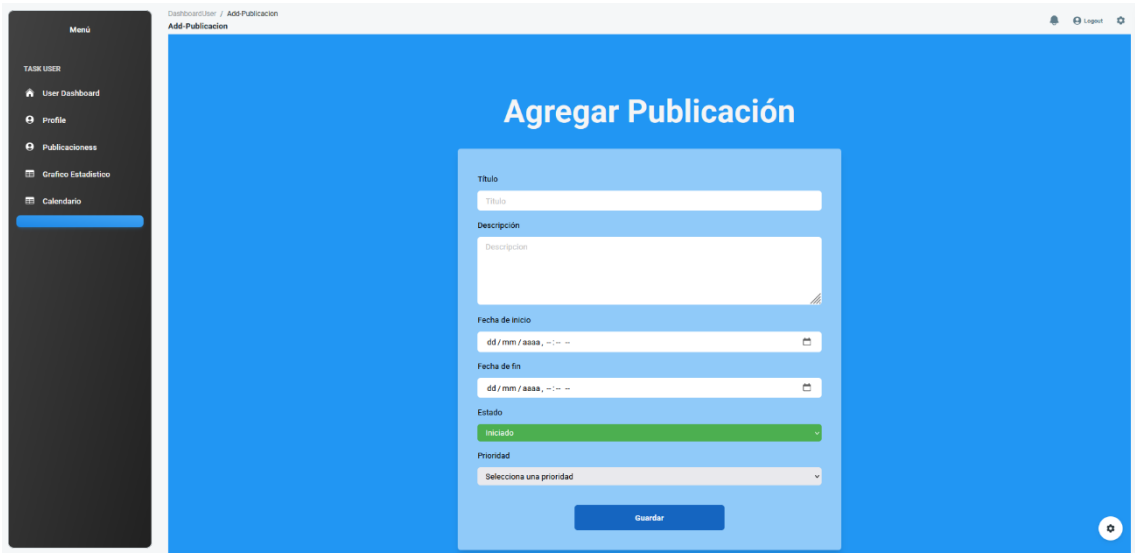
Pantalla donde se ve todas las publicaciones del usuario normal



Nota. En esta figura se muestra la página principal para visualizar todas las publicaciones del usuario y estas están puestas por color verde: publicaciones iniciadas, naranja: publicaciones pendientes y azul: publicaciones terminadas. Autoría propia

Figura 42

Pantalla para crear una publicación nueva



Nota. En esta figura se muestra la opción de crear una publicación con los campos de título, descripción, fecha inicio , fecha fin, estado(iniciado, pendiente y terminado) y prioridad(alta, media, bajo). Autoría propia

Figura 43

Búsqueda de publicaciones por la fecha

Nota. En esta figura se muestra la opción de búsqueda de una publicación mediante la fecha que se necesite. Autoría propia

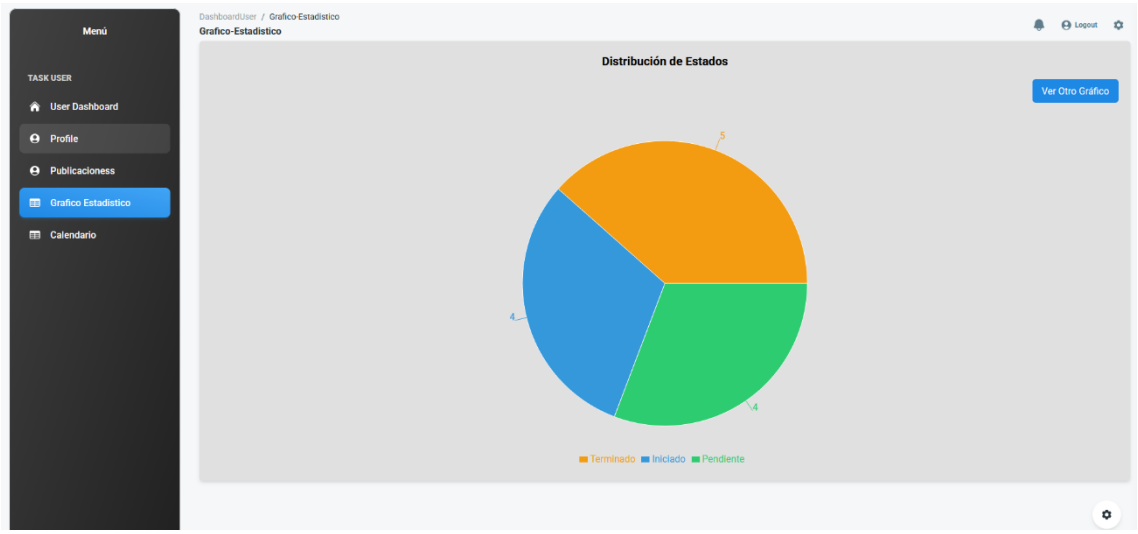
Figura 44

Pantalla para editar una publicación

Nota. En esta figura se muestra la opción de editar la publicación que se ha seleccionado, en donde carga la información de la publicación y se puede cambiar la información. Autoría propia

Figura 45

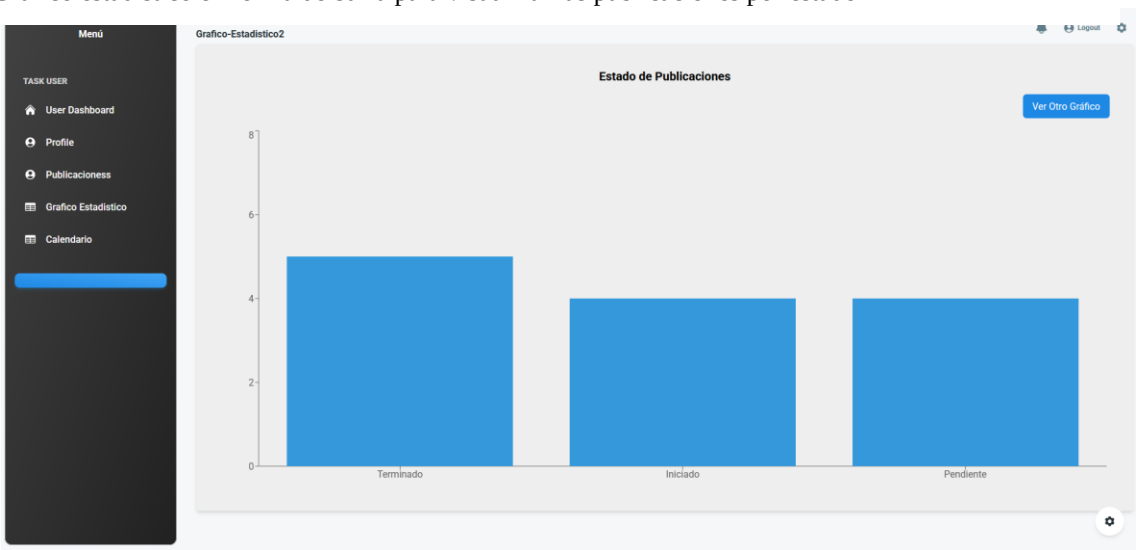
Gráfico estadístico en forma de pastel para visualizar las publicaciones por estado



Nota. En esta figura se muestran las publicaciones del usuario según su estado de una manera de un gráfico estadístico de forma de pastel. Autoría propia

Figura 46

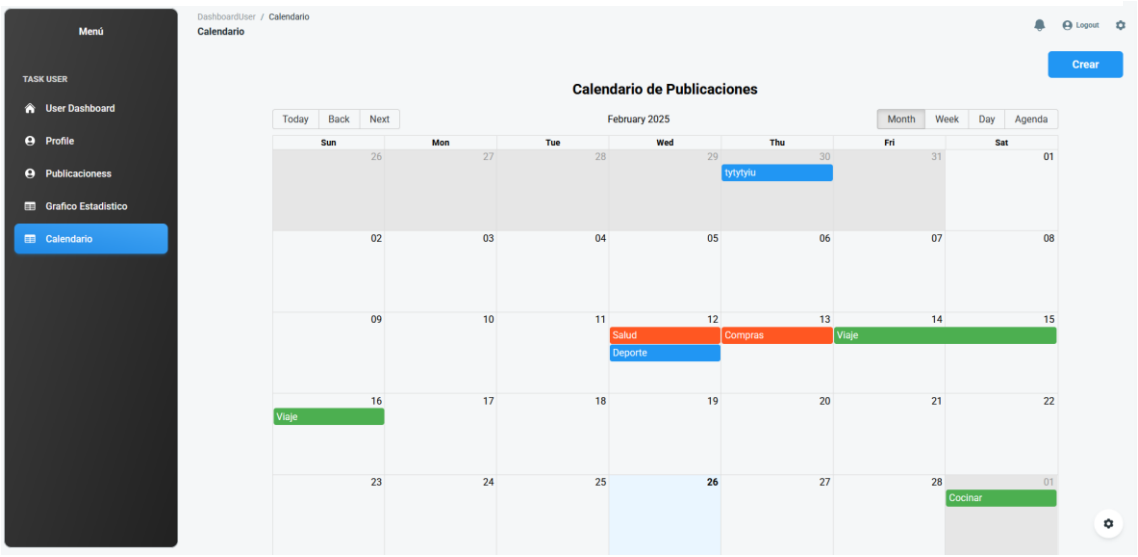
Gráfico estadístico en forma de barra para visualizar las publicaciones por estado



Nota. En esta figura se muestran las publicaciones del usuario según su estado de una manera de un gráfico estadístico de forma de barras. Autoría propia

Figura 47

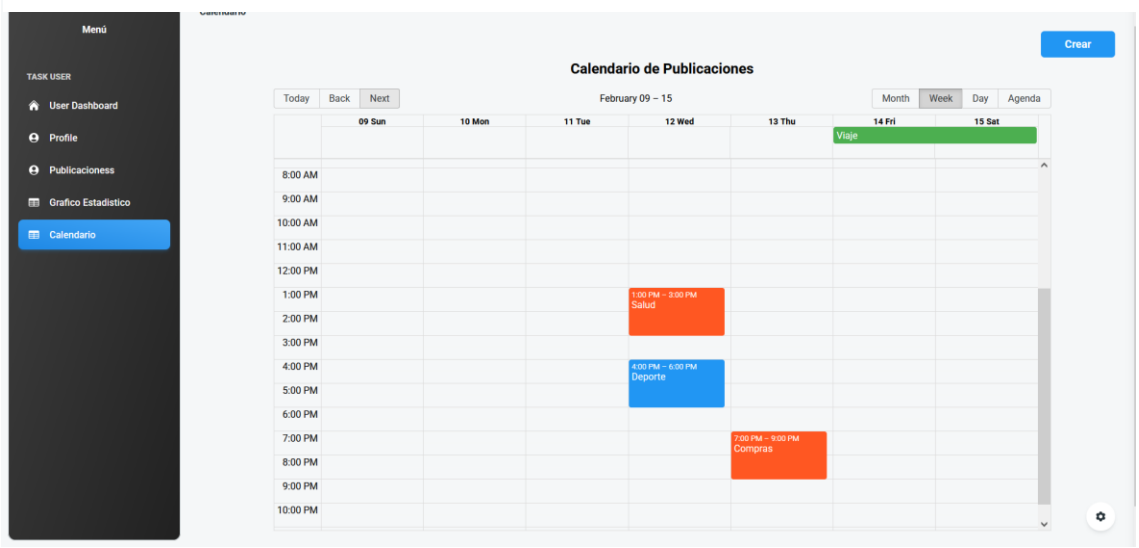
Calendario donde se ven las todas las publicaciones



Nota. En esta figura se muestra la opción al usuario común de ver sus actividades en un calendario para el usuario. Autoría propia

Figura 48

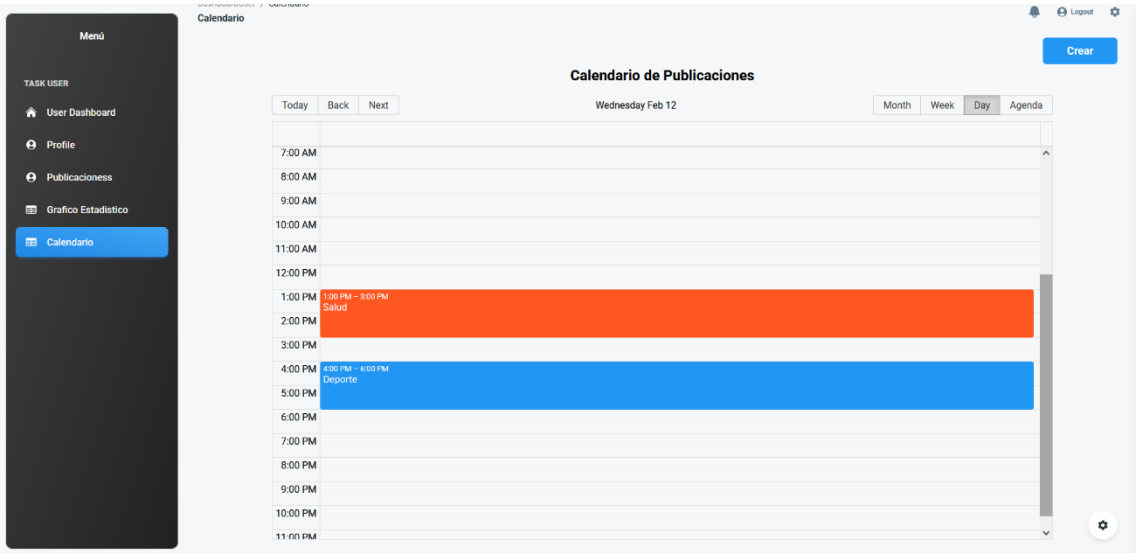
Calendario en vista por semanas para ver las publicaciones



Nota. En esta figura es muestra la opción al usuario común de ver sus actividades en un calendario en forma de semana. Autoría propia

Figura 49

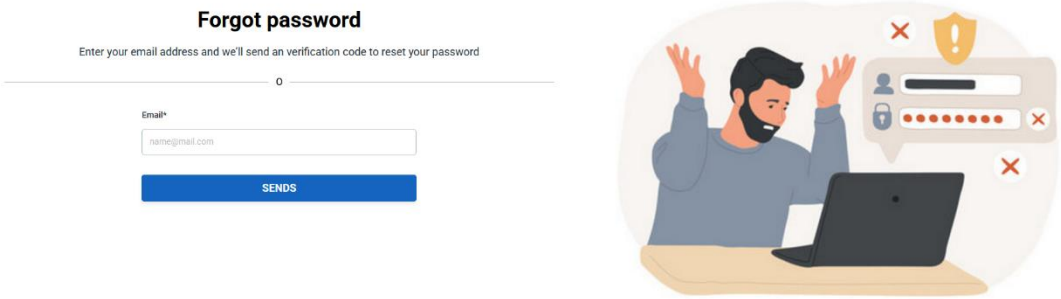
Calendario en vista por días para ver las publicaciones



Nota. En esta figura se muestra la opción al usuario común de ver sus actividades en un calendario en forma de días. Autoría propia

Figura 50

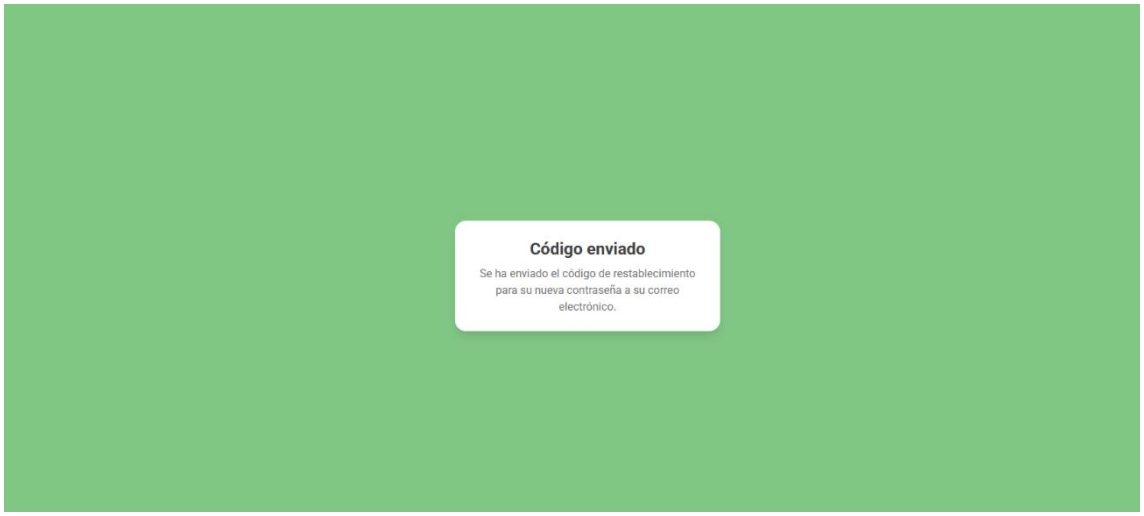
Pantalla para pedir el correo para el cambio de contraseña



Nota. En esta figura se muestra la pantalla de olvido de contraseña donde se debe poner el correo electrónico para enviar la opción de cambiar de contraseña. Autoría propia

Figura 51

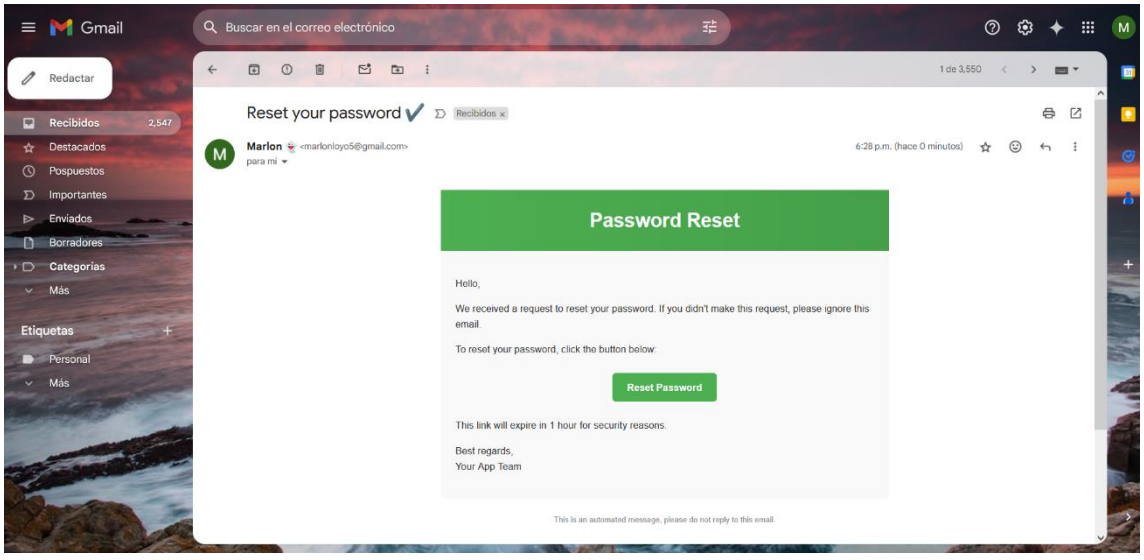
Mensaje que dice que se ha enviado al correo para el cambio de contraseña



Nota. En esta figura se muestra que se ha enviado al correo del usuario la opción de cambiar la contraseña. Autoría propia

Figura 52

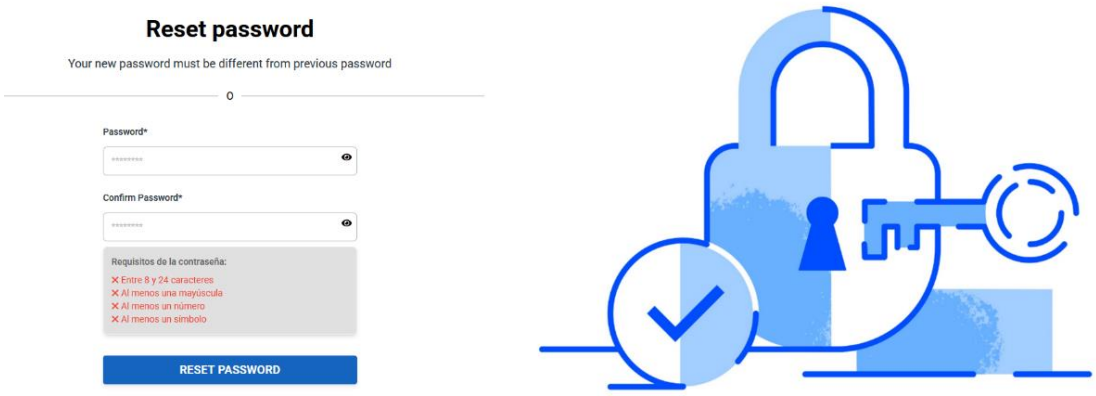
Correo donde se podrá cambiar la contraseña



Nota. En esta figura se muestra el correo del usuario mostrando que se envió un correo con la opción de cambiar la contraseña del usuario. Autoría propia

Figura 53

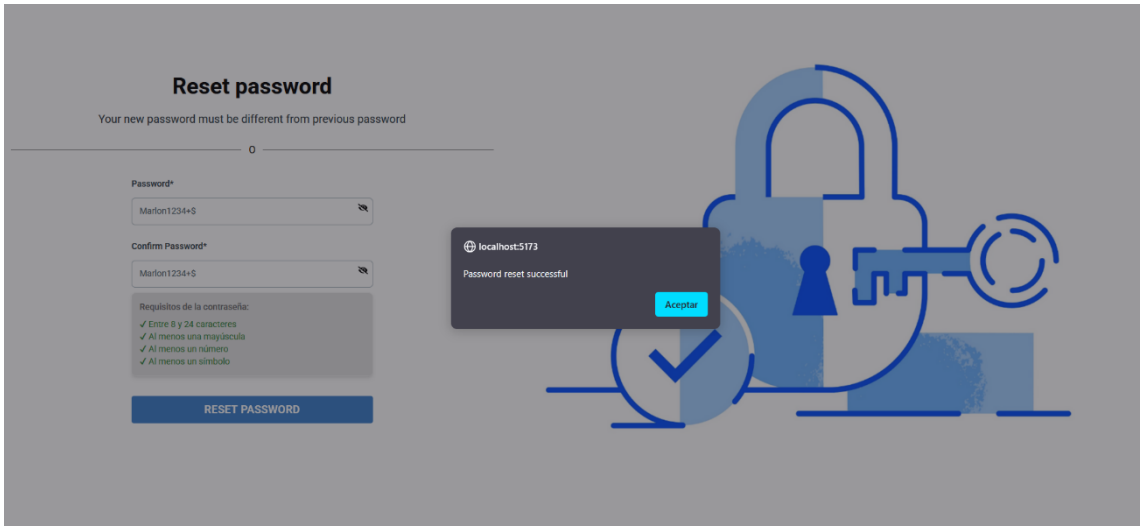
Pantalla donde se pondrá la nueva contraseña



Nota. En esta figura se muestra la pantalla para actualizar la contraseña, donde está debe cumplir con ciertos parámetros de seguridad. Autoría propia

Figura 54

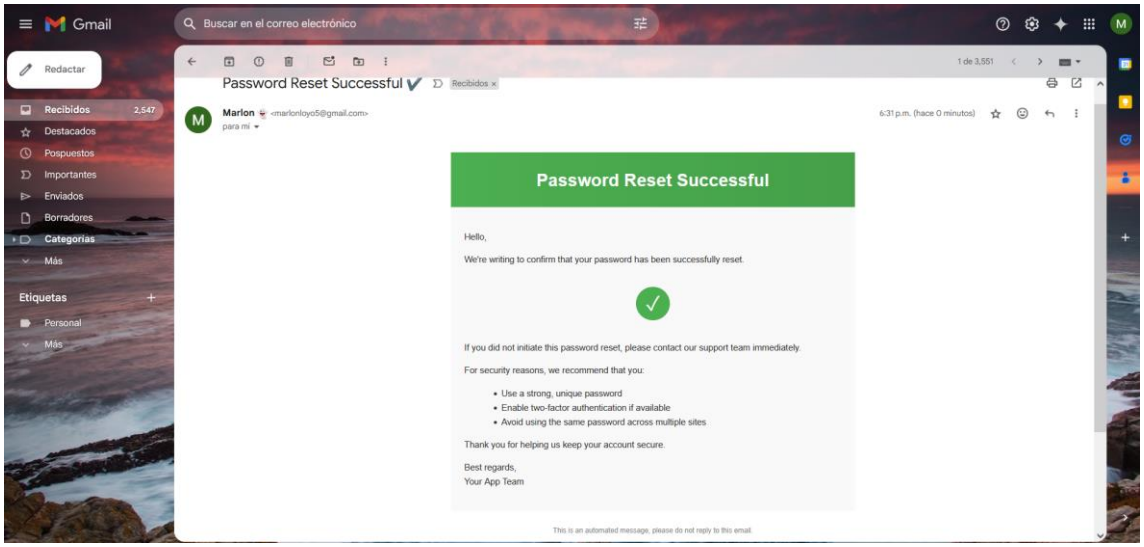
Pantalla donde la nueva contraseña si cumple con las condiciones y da por confirmado que se ha actualizado



Nota. En esta figura se muestra que la contraseña sí se ha actualizado y se puede seguir para ingresar nuevamente. Autoría propia

Figura 55

Correo que se envía el Gmail donde se confirma la contraseña actualizada correctamente



Nota. En esta figura muestra el correo del usuario donde se envía un mensaje de que la contraseña se ha actualizado correctamente. Autoría propia

CAPÍTULO III

RESULTADOS

En este capítulo se describe el proceso de validación del TOP Ten de Owasp en el aplicativo web. La validación se realizó utilizando la herramienta Owasp Zap, ejecutando el análisis de vulnerabilidades que la herramienta nos provee y discutiremos los resultados y posibles recomendaciones que se puedan aplicar para mejorar la seguridad del sistema.

El propósito de esta sección es garantizar que el aplicativo cumpla con las buenas prácticas de Owasp. Para ello, se llevó a cabo un análisis detallado de las vulnerabilidades identificadas, evaluando su impacto y riesgo en el sistema.

3.1 Análisis de resultados de Owasp Zap

En este apartado se describe el proceso de análisis de las vulnerabilidades en el aplicativo web, con el propósito de identificar posibles fallos de seguridad que puedan comprometer la integridad y estabilidad del sistema. Para ello, se empleó la herramienta OWASP ZAP, que permitirá detectar riesgos asociados a las vulnerabilidades más comunes según el OWASP Top Ten**.** Se realizaron pruebas detalladas con OWASP ZAP, lo que permitió detectar diversas áreas críticas que podrían afectar la seguridad de la aplicación. A continuación, se presentan los hallazgos organizados por categorías para facilitar su análisis y comprensión.

Figura 56

Análisis de vulnerabilidades OWASP

		Confidence				
		Confirmado por Usuario	Alta	Media	Baja	Total
Risk	Alto	0 (0,0 %)	1 (11,1 %)	0 (0,0 %)	0 (0,0 %)	1 (11,1 %)
	Medio	0 (0,0 %)	1 (11,1 %)	1 (11,1 %)	0 (0,0 %)	2 (22,2 %)
	Bajo	0 (0,0 %)	0 (0,0 %)	2 (22,2 %)	0 (0,0 %)	2 (22,2 %)
	Informativo	0 (0,0 %)	0 (0,0 %)	3 (33,3 %)	1 (11,1 %)	4 (44,4 %)
	Total	0 (0,0 %)	2 (22,2 %)	6 (66,7 %)	1 (11,1 %)	9 (100%)

Nota. En la figura se muestra el resultado del análisis hecho en Owasp Zap. Autoría propia

3.2 Tipos de alertas de seguridad detectadas

La tabla 19 nos muestra el detalle del tipo de alerta identificado que se encontró durante la evaluación del aplicativo, indicando su nivel de riesgo del tipo y la cantidad encontrada.

Tabla 19. Tipos de alertas de seguridad identificadas

Tipo de Alerta	Riesgo	Cantidad
Divulgación de hash – BCrypt	Alto	6
Cabecera Content Security Policy (CSP) no configurada	Medio	1
Falta de cabecera Anti-Clickjacking	Medio	1
Inclusión de archivos fuente JavaScript entre dominios	Bajo	1
Revelación de IP privada	Bajo	1
Aplicación Web Moderna	Informativo	1
Divulgación de Información - Información sensible en URL	Informativo	1
Divulgación de información - Comentarios sospechosos	Informativo	11
Respuesta de Gestión de Sesión Identificada	Informativo	3

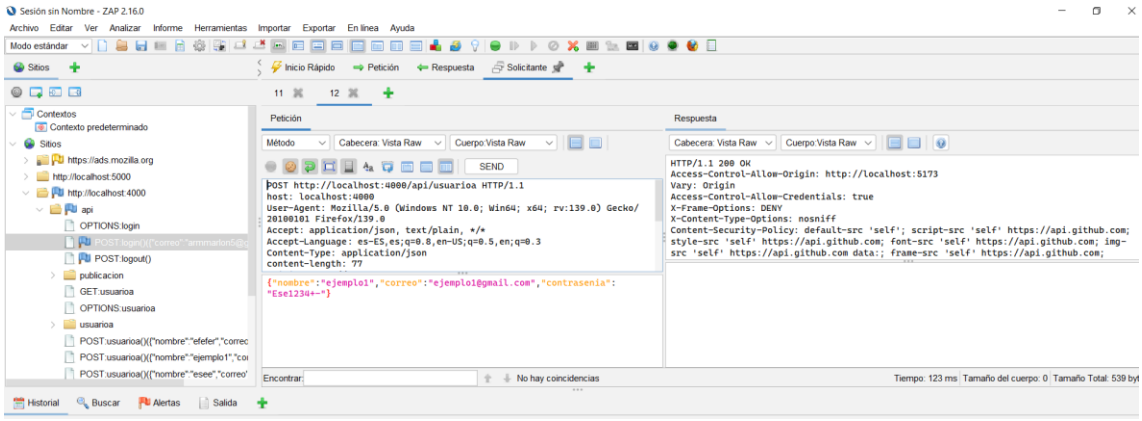
El análisis de vulnerabilidad reveló otros problemas de seguridad que, si bien no se consideran críticos o de alta importancia, deben ser atendidos para mejorar la seguridad general de la aplicación.

3.3 Evaluación del TOP Ten de Owasp

A01: Control de acceso roto

Para comprobar este apartado en el usuario administrador, al ingresar en este se verificó que si al cambiar el token de cookie deja creer otro usuario o no.

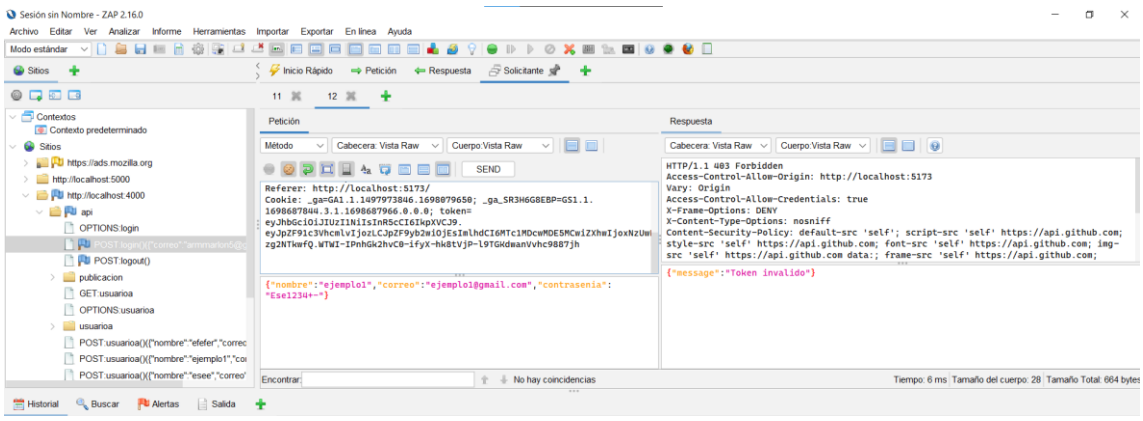
Figura 57
Pantalla del token del administrador para ingresar al login



Nota. En la figura se muestra el login del usuario administrador con el token válido. Autoría propia

En nuestro caso, el token válido deja crear en caso contrario, no lo deja crear a continuación se lo demuestra. El token se cambió algo al final y no deja crear el usuario nuevo.

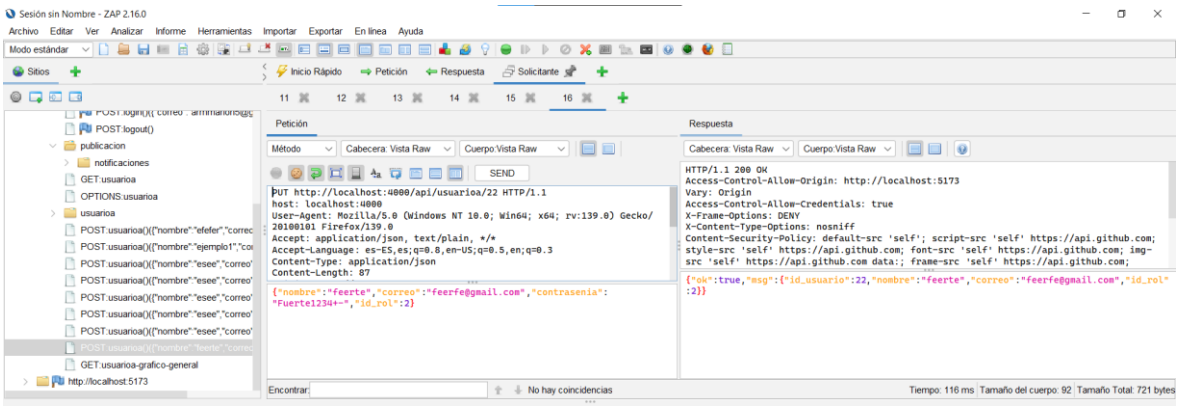
Figura 58
Pantalla donde el token es diferente y no deja entrar a la cuenta



Nota. En la figura se muestra un token no válido y por ende no deja entrar a la cuenta. Autoría propia

Figura 59

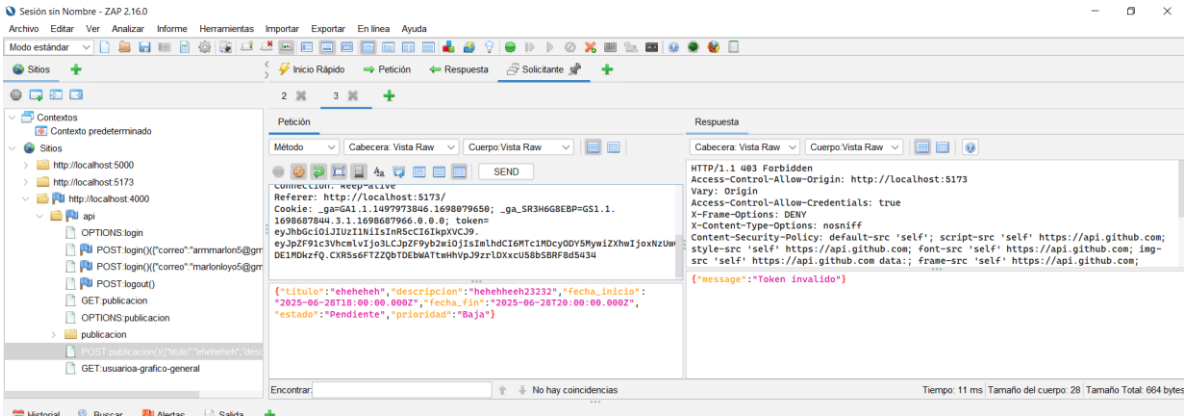
Token diferente y ver si funciona en la petición de actualizar un usuario



Nota. En la figura se muestra la petición de actualizar la información de un usuario con el Token válido. Autoría propia

Figura 60

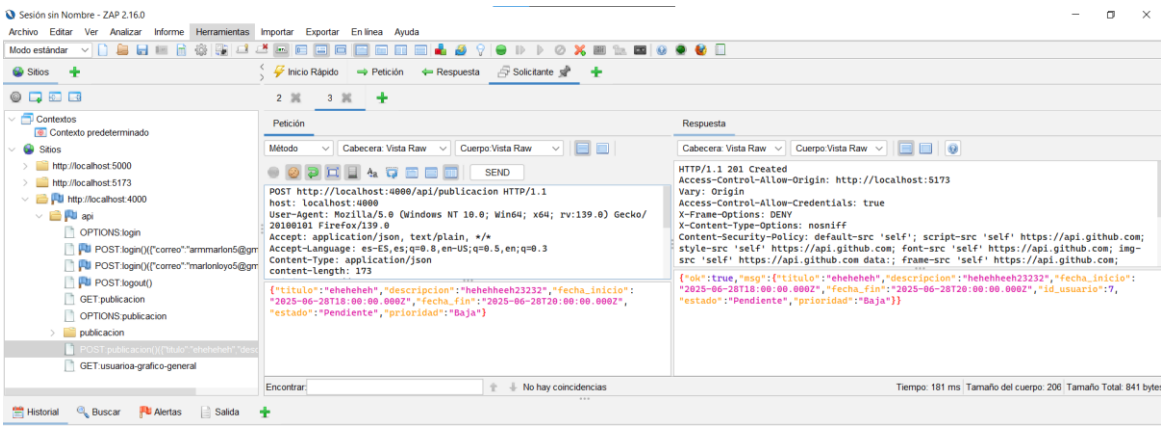
Token invalido y no deja seguir con la acción



Nota. En la figura se muestra el token no es válido y con ello no deja seguir con la acción. Autoría propia

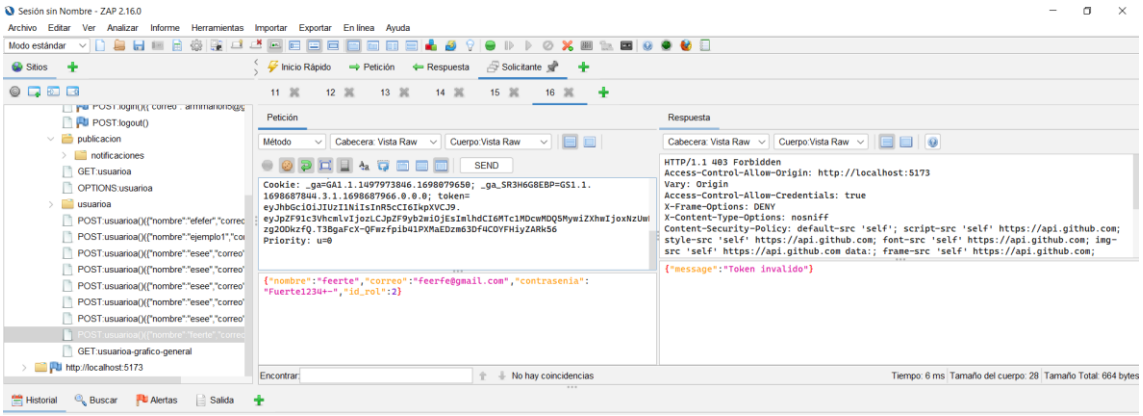
Figura 61

Estamos como usuario normal y el token es valido



Nota. En la figura se muestra en el usuario normal para crear publicaciones con el token válido. Autoría propia

Figura 62
EL token no es válido y no deja continuar la acción

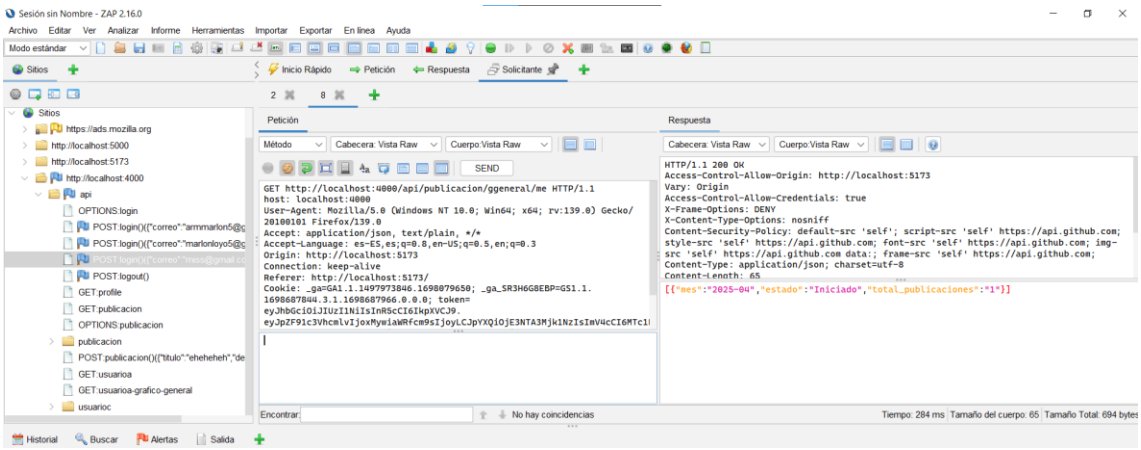


Nota. En la figura se muestra si el token cambia no deja crear la publicación que se desea. Autoría propia

02. Fallos criptográficos

Revisamos que al momento de hacer login se ven datos del usuario y en nuestro caso no muestra esa información.

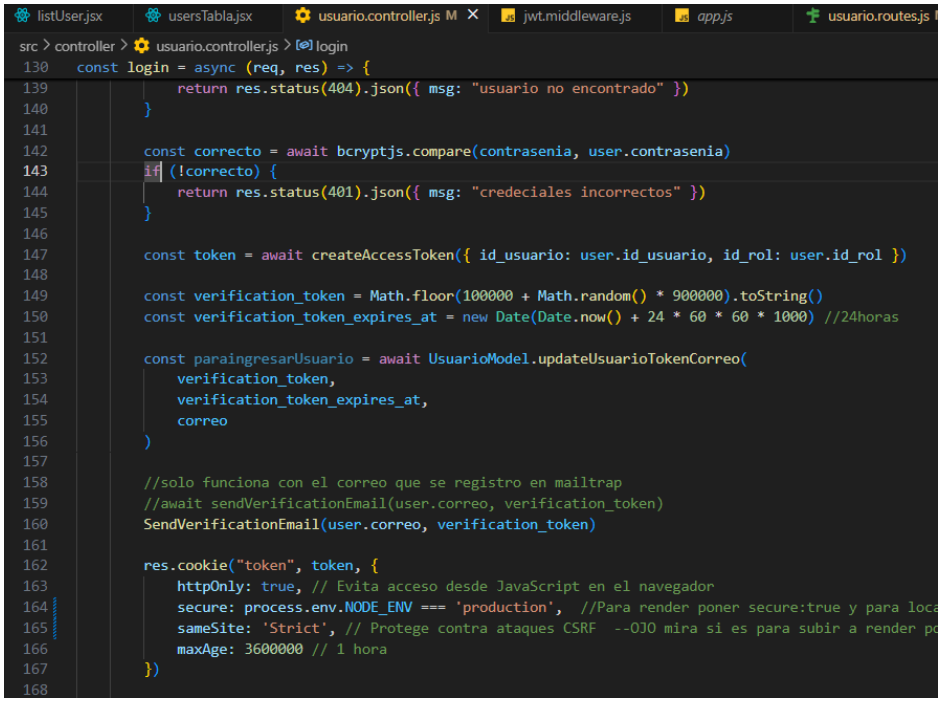
Figura 63
Login y ver si muestra información



Nota. En la figura se muestra el login para ver si la información es visible. Autoría propia

Y revisar que la cookie esté bien configurada con Secure, httpOnly, samesite esto nos permite que sea seguro solo en Https y que evitar que se filtre la cookie y robo del token por XSS y esto lo realizamos en el controller.

Figura 64
Configurar en el controller para evitar posibles ataques

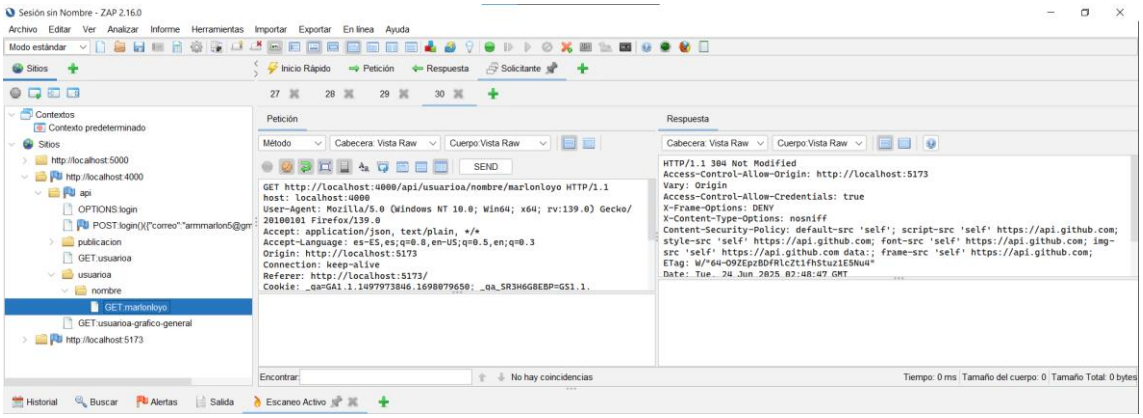


Nota. En la figura se muestra el controller con las configuraciones para evitar problemas de seguridad. Autoría propia

03: Inyecciones SQL

Para probar, se hizo la petición con el usuario administrador y se hizo la búsqueda de usuario y se hace la prueba de inyección sql y otras y muestra que no hay vulnerabilidad.

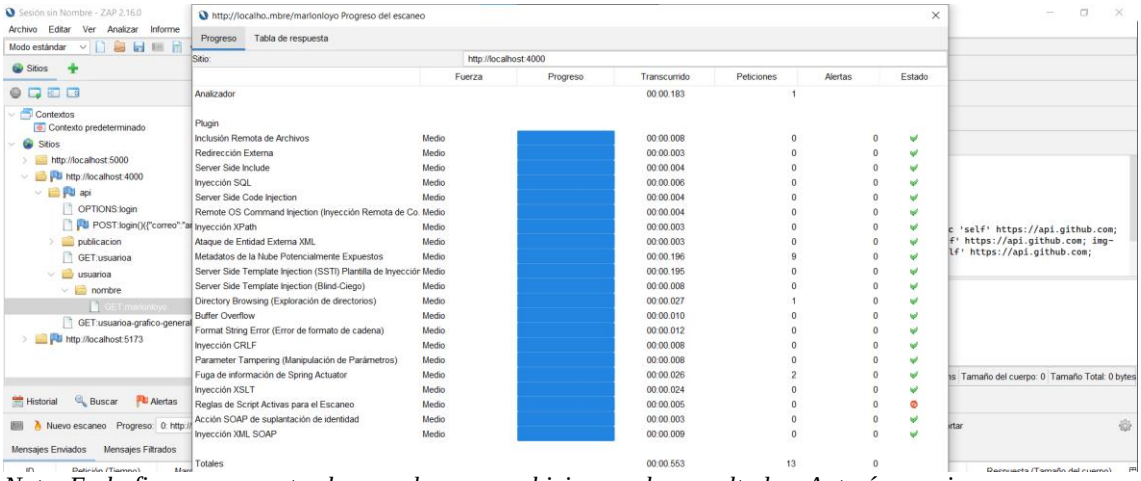
Figura 65
En la función de búsqueda de usuario hacer la prueba de inyección sql



Nota. En la figura se muestra en la función de búsqueda de usuarios y se hace la prueba de inyección. Autoría propia

Figura 66

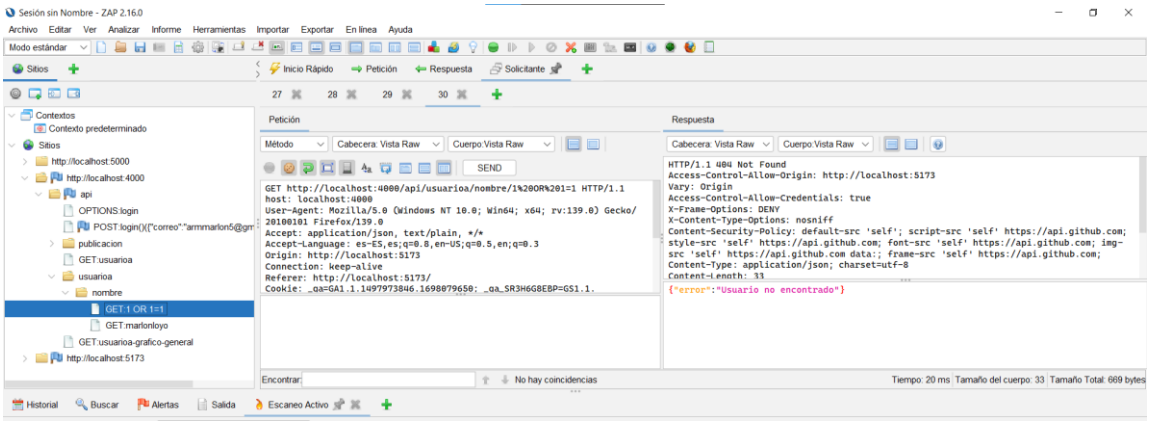
Pantalla de owasp zap donde se ven las pruebas realizadas



Nota. En la figura se muestra las pruebas que se hicieron y los resultados. Autoría propia

Figura 67

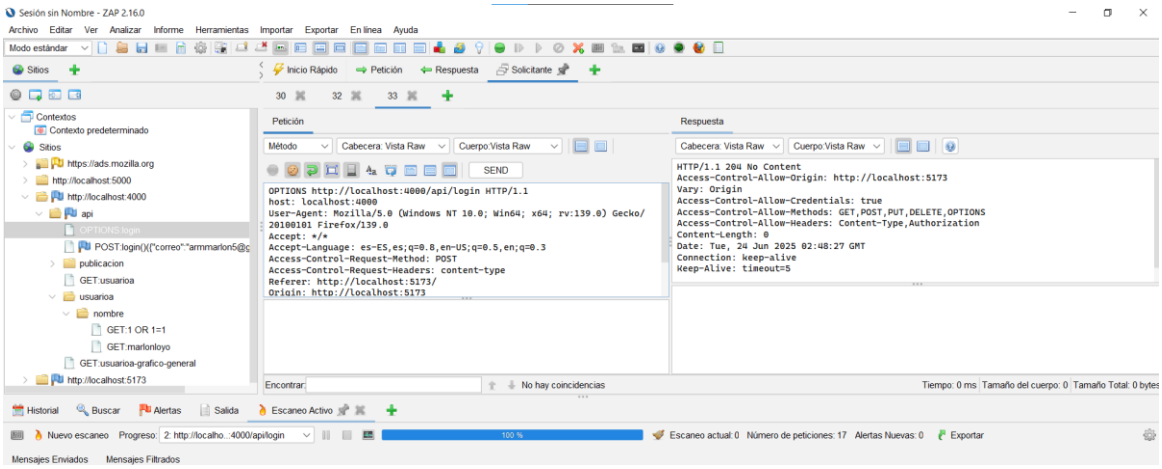
Ataque manual de inyección y no da resultados



Nota. En la figura se muestran la prueba manual y no da resultado alguno de inyección sql. Autoría propia

Figura 68

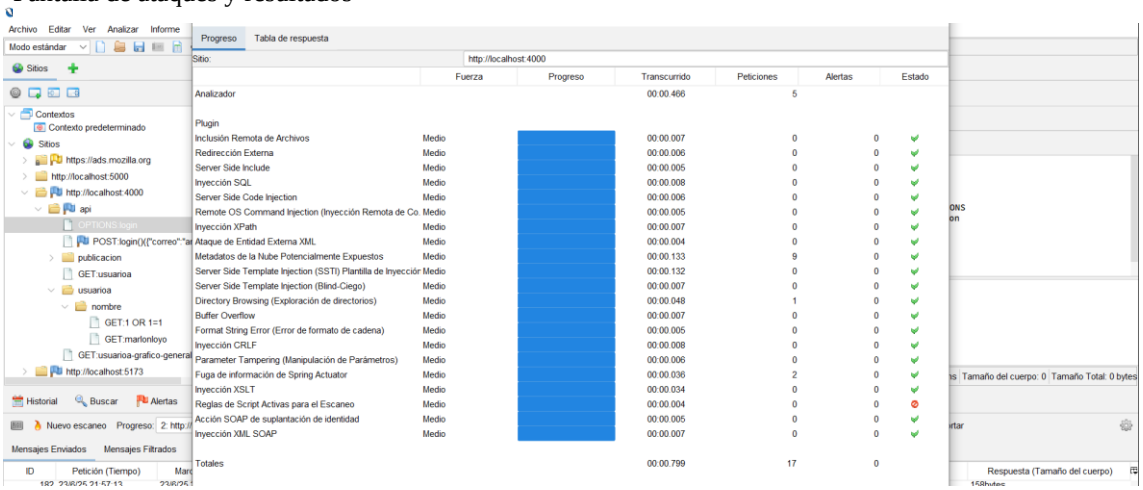
Pantalla en login y se hizo la prueba y no da resultados



Nota. La figura se muestra en el login y no da resultados de que exista inyección sql Autoría propia

Figura 69

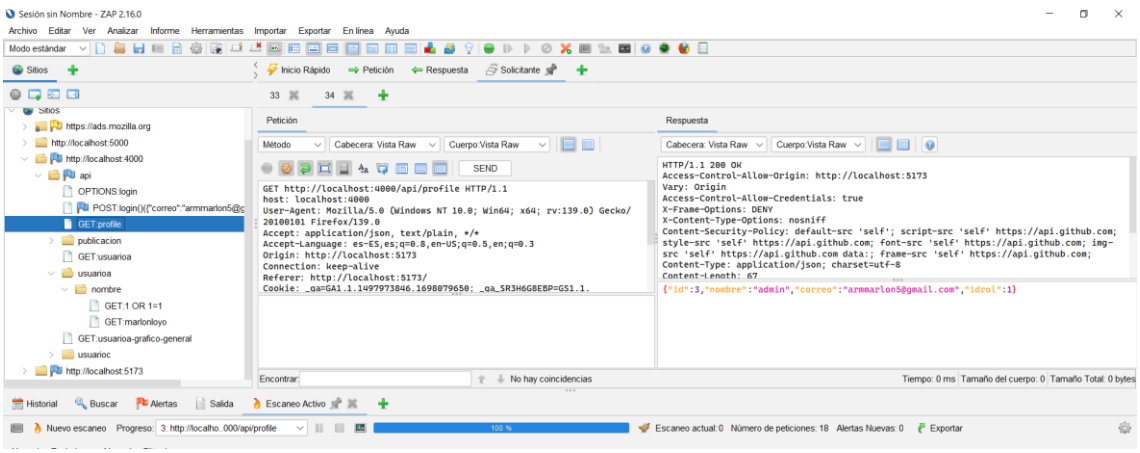
Pantalla de ataques y resultados



Nota. En la figura se muestra los ataques y resultados que se hicieron y ver si existe alguna vulnerabilidad. Autoría propia

Figura 70

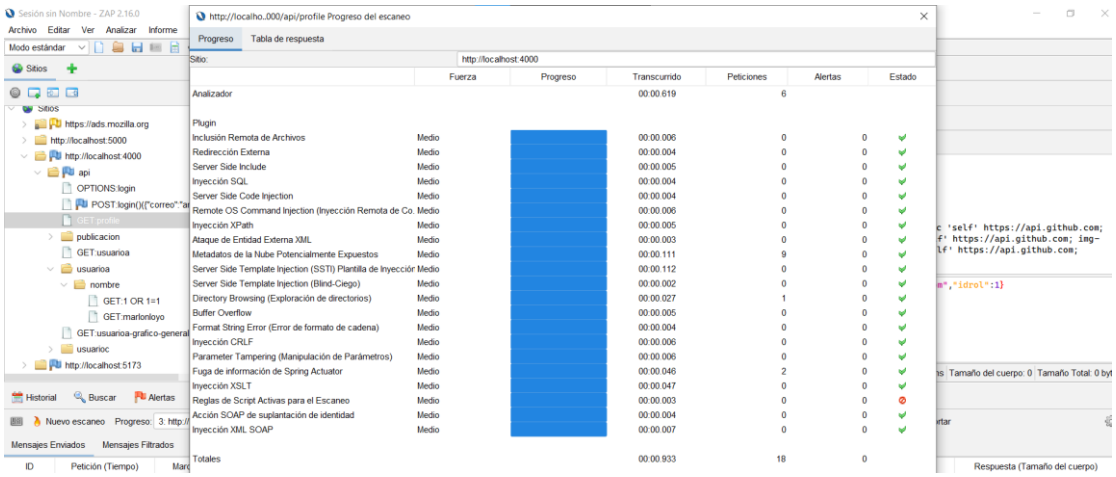
Pruebas en el perfil de usuario



Nota. La figura se muestra pruebas realizadas en el perfil del usuario normal y administrador. Autoría propia

Figura 71

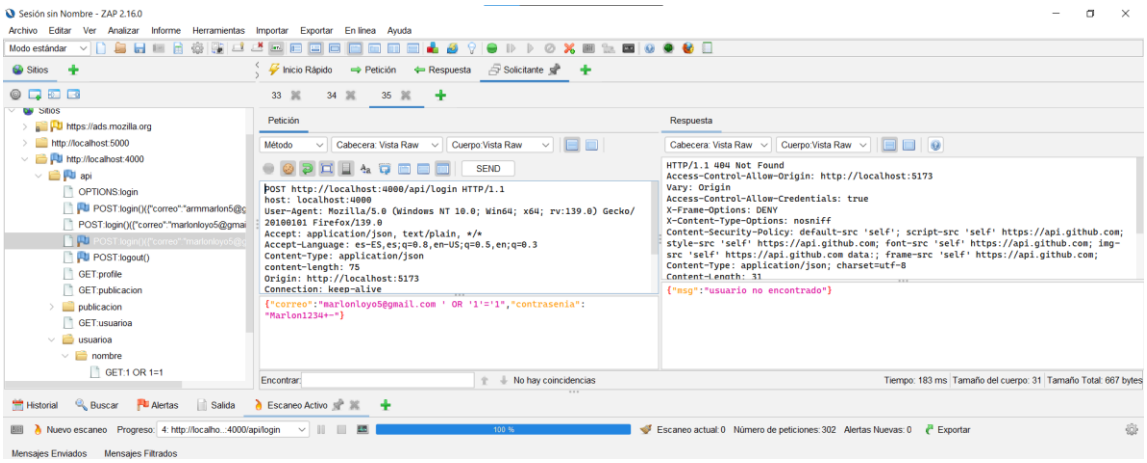
Pantalla de pruebas y resultados en el perfil de usuario



Nota. La figura se muestra las pruebas y resultados que se hicieron en el perfil del usuario y ver si existe alguna vulnerabilidad. Autoría propia

Figura 72

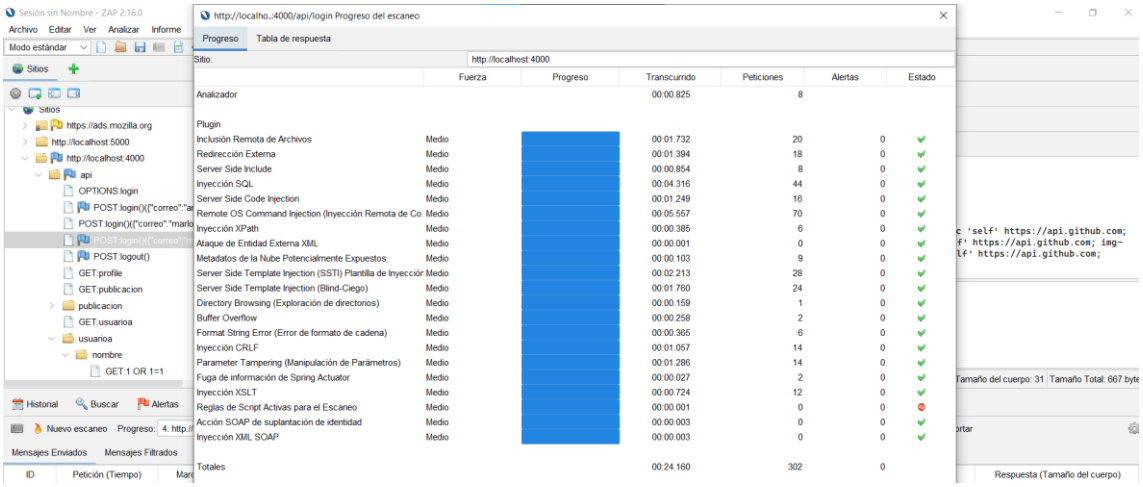
Hacer pruebas en login del usuario normal



Nota. La figura se muestra que sé probó también el usuario normal y no da resultados de sql inyección.

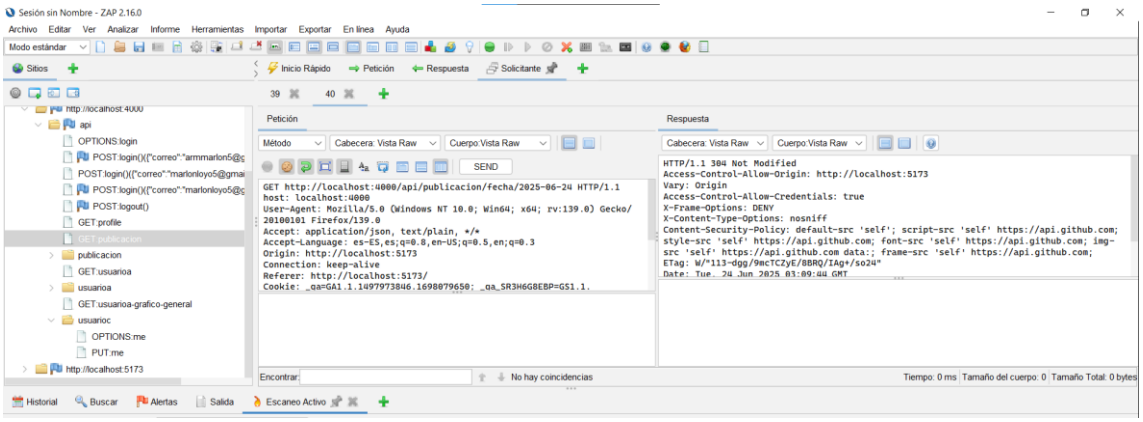
Figura 73

Pantalla de pruebas y resultados



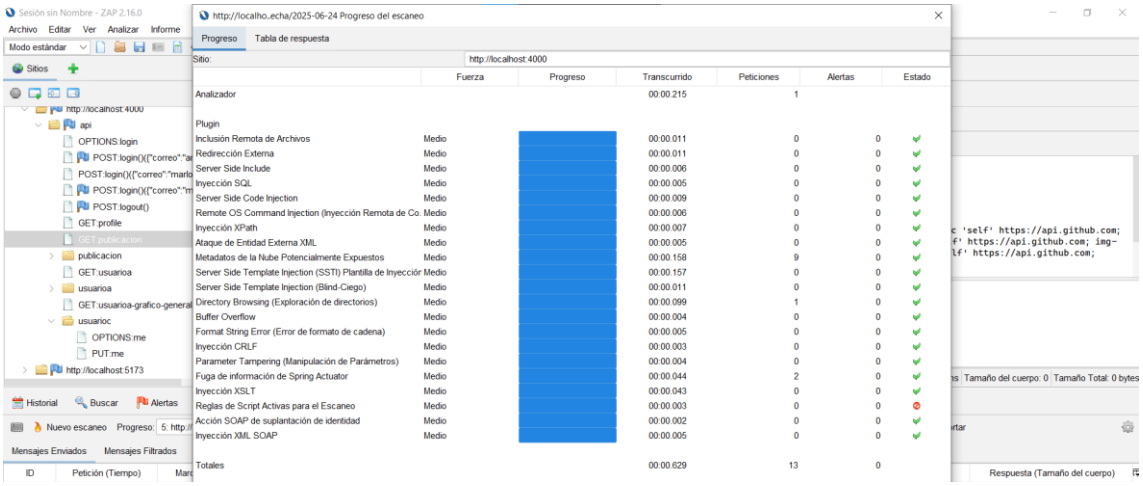
Nota. La figura muestra las pruebas del intento y resultados. Autoría propia

Figura 74
Pruebas en el filtro de búsqueda de publicación por fecha



Nota. La figura se muestra en la función de filtro de búsqueda de fechas de las publicaciones. Autoría propia

Figura 75
Pantalla de pruebas y resultados de las pruebas



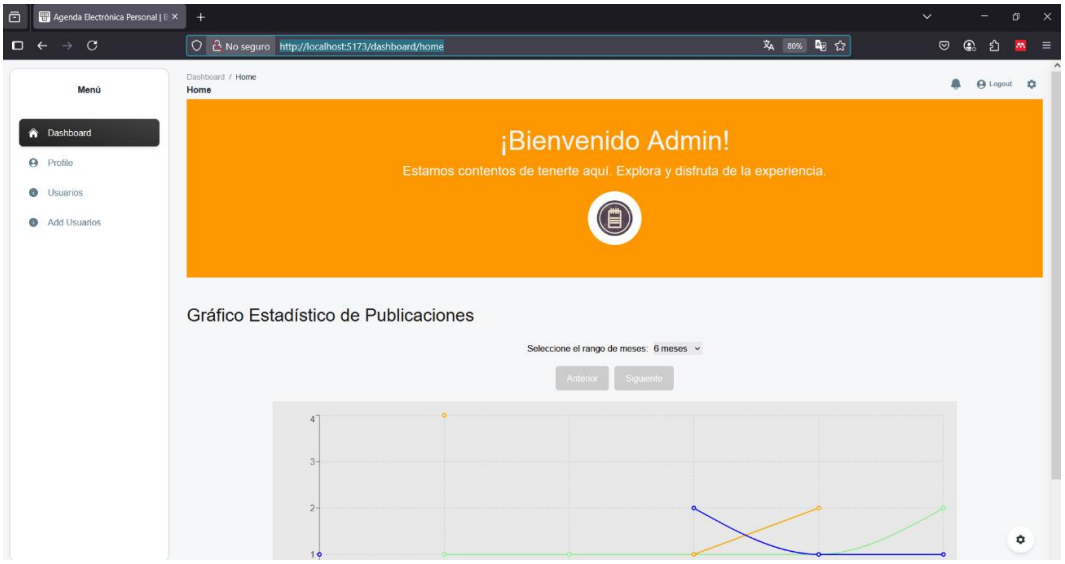
Nota. La figura se muestra los resultados de las pruebas de inyección sql y no dan resultados. Autoría propia

A04: Diseño inseguro

Comprobar si el diseño está bien en este caso vamos a revisar.

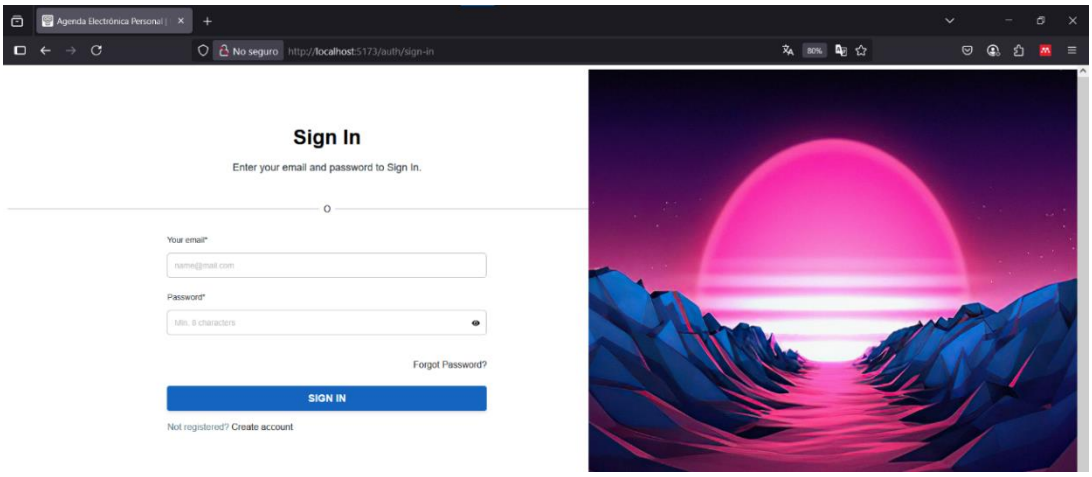
Ingresamos como administrador y vamos a la url y cambiamos la ruta del administrador que es http://localhost:5173/dashboard/home para la pantalla de bienvenida del administrador y cambió eso por el usuario normal http://localhost:5173/dashboardUser/HomeUser y muestra que no deja cambiar.

Figura 76
Pantalla de inicio de administrador



Nota. En la figura se muestra la pantalla del administrador. Autoría propia

Figura 77
Cambió la url de administrador o normal pero no deja

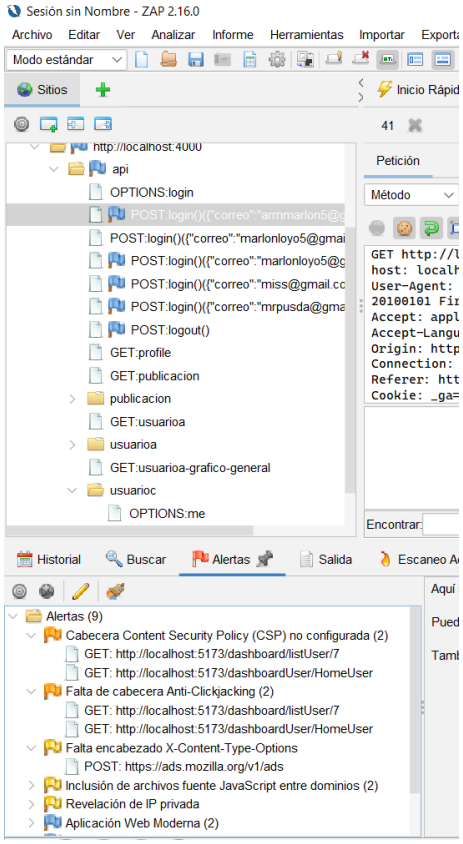


Nota. En la figura se muestra el intento de cambio de url para ir a otro lugar y no nos deja. Autoría propia

A05: Configuración incorrecta de seguridad

Figura 78

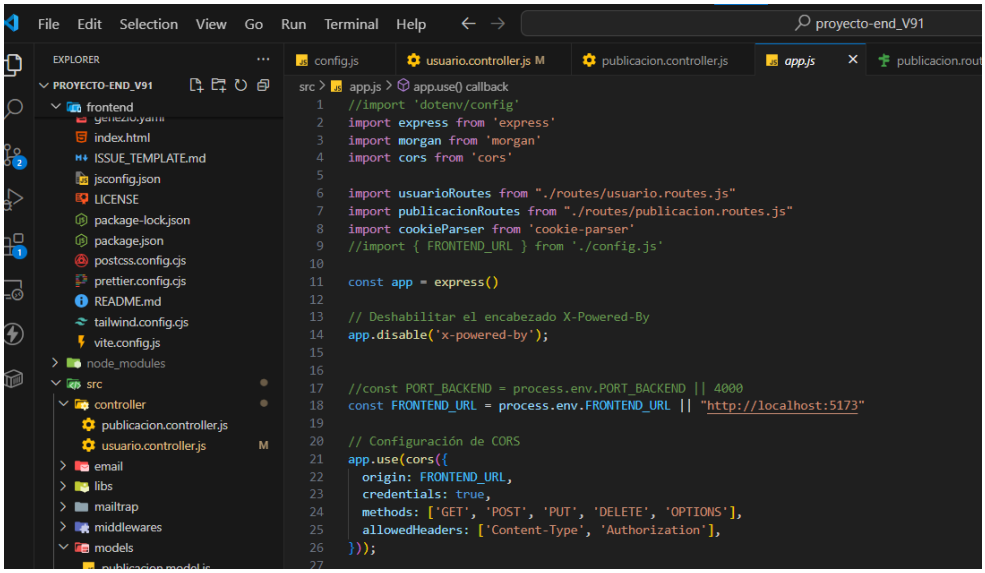
Alertas de ciertas vulnerabilidades



Nota. En la figura se muestra vulnerabilidades cabeceras content security policy – x content type options – anti clickjacking. Autoría propia

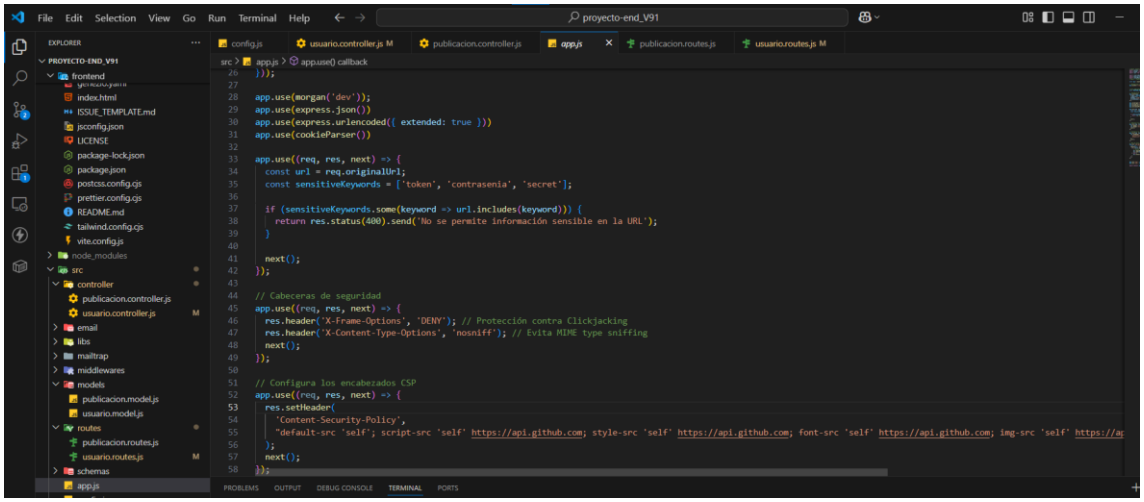
Figura 79

Configuración en el servidor para evitar ataques



Nota. EN la figura muestra la configuración para evitar estos errores. Se lo hizo en servidor donde configuramos para evitar problemas de Clickjacking, encabezado CSP. Autoría propia

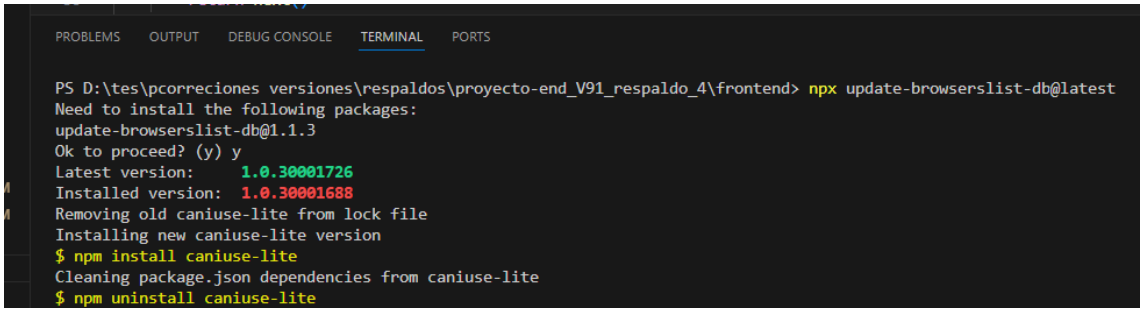
Figura 80
Configuración en el servidor para evitar ataques



Nota. La figura muestra la configuración en el servidor y tiene algunas configuraciones para evitar vulnerabilidades. Autoría propia

A06: Componentes vulnerables o desactualizados

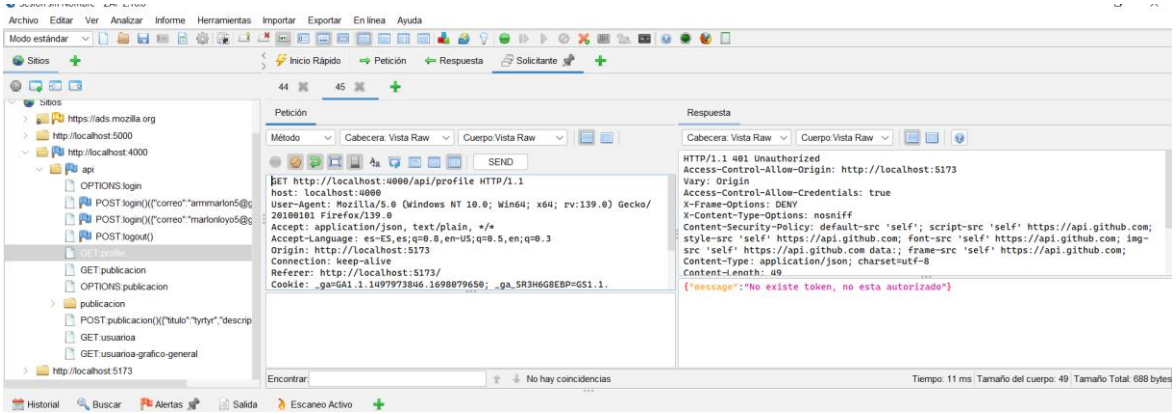
Figura 81
Actualización de librerías por antigüedad



Nota. En la figura muestra a Owasp zap no lo detecta directamente, pero es sencillo ya que ,sería tener actualizadas todas las librerías, frameworks actualizados para que no sean vulnerables por ser ya versiones no tan actuales. Autoría propia

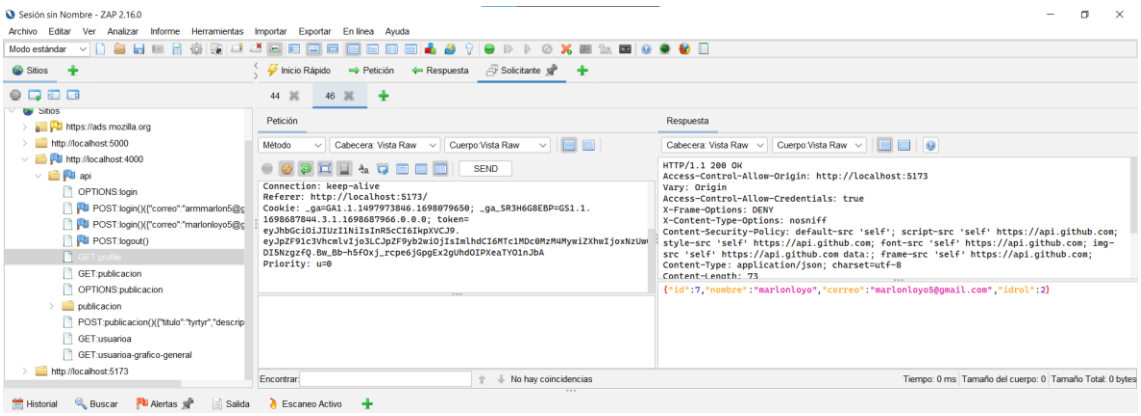
A07: Fallos de identificación y autenticación

Figura 82
Ingreso al usuario normal con la cookie válida



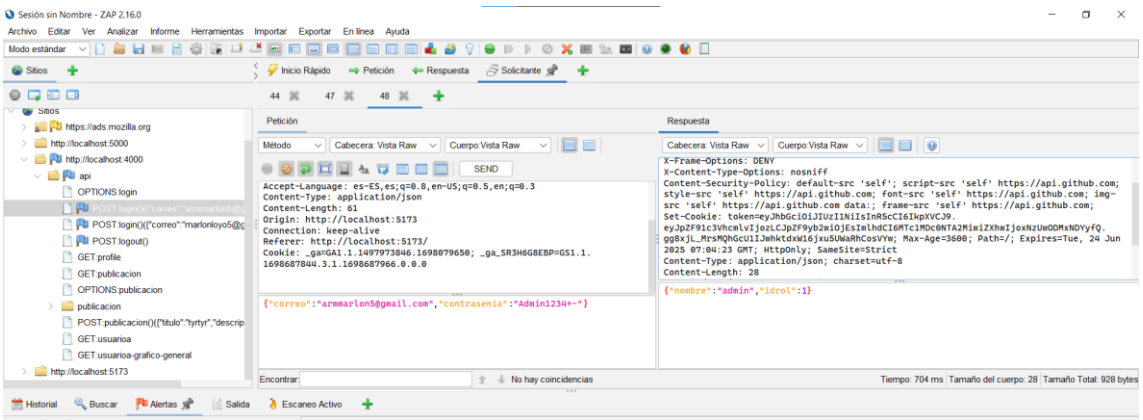
Nota. En la figura muestra si la aplicación permite acceder sin autentificación o fallos de sesiones. En este caso, vamos a ver si accede el perfil del usuario normal con la cookie. Autoría propia

Figura 83
Ingreso correcto por la cookie válida



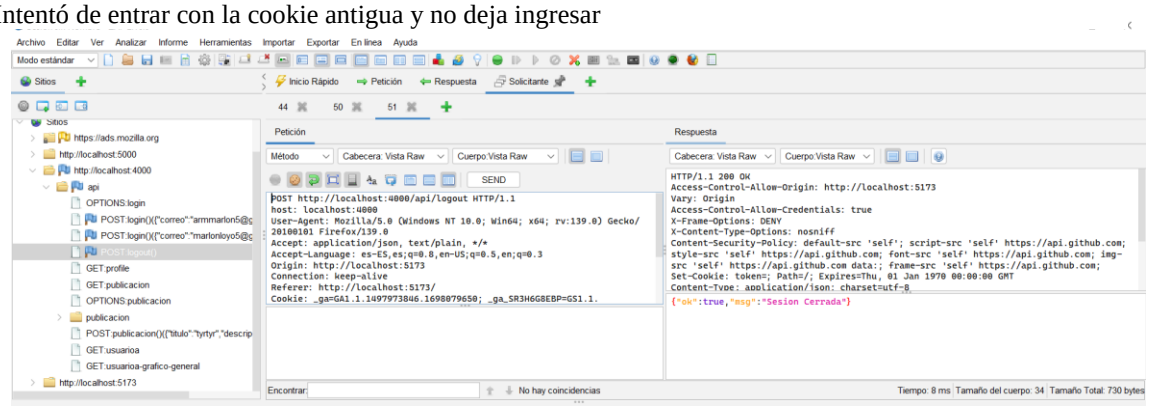
Nota. En la figura se muestra el token válido deja ver el usuario. Autoría propia

Figura 84
Cerrar la cuenta y guardar la cookie



Nota. En la figura muestra el cerrar sesión ya no deja entrar si se coge el token antiguo. Autoría propia

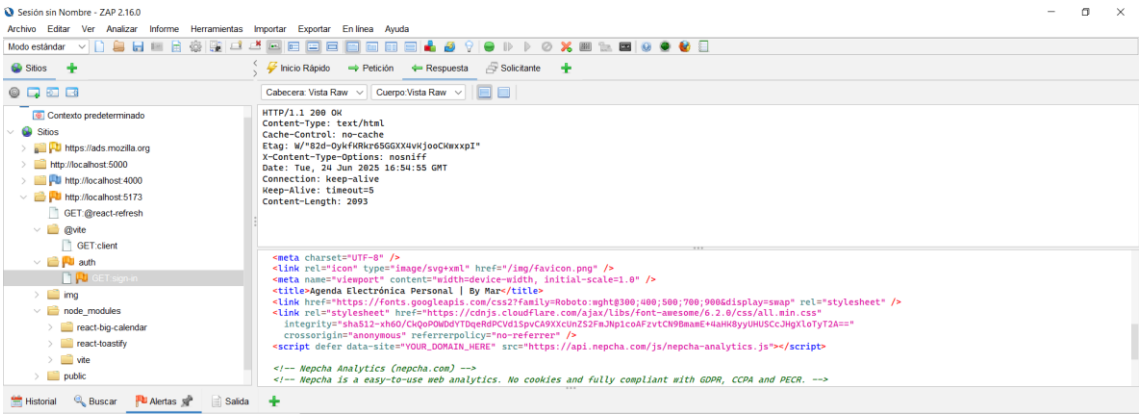
Figura 85
Intentó de entrar con la cookie antigua y no deja ingresar



Nota. En la figura muestra el intentó de entrar y no deja realizar la acción. Autoría propia

A08: Fallos en la integridad del software y los datos

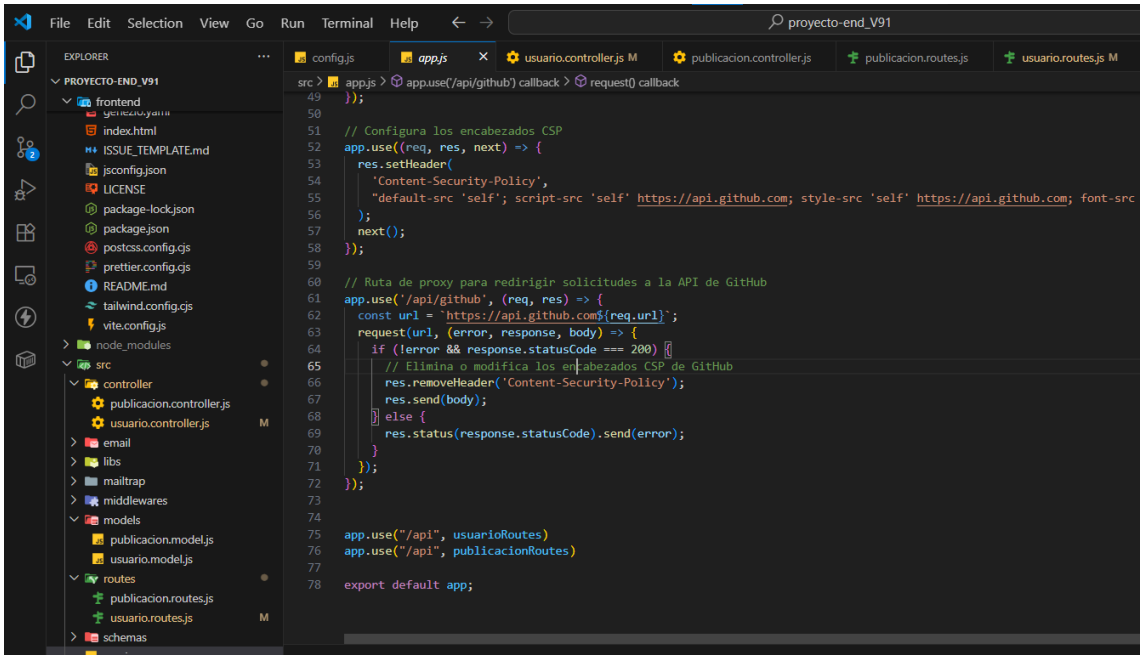
Figura 86
Pantalla de owasp que nos da la alertar de vulnerabilidad



Nota. En la figura muestra el caso, no nos dio la alerta y la fuimos a comprobar en el login donde ya existía una configuración que ayuda a evitar este tipo de problemas y está de esta formar. Autoría propia

La configuración aquí presente es la parte del integrity y crossorigin la cual nos permite que evitar que nos pueda afectar ese componente que se usa ya que si el texto de integrity es diferente, no va a cargar y no funcionará y así evitando que se modifique este componente y pueda generar daños.

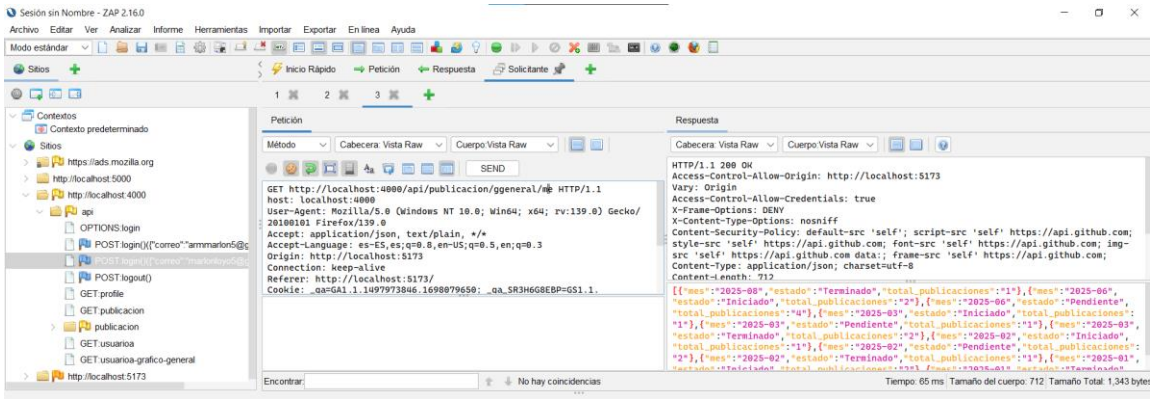
Figura 87
Configurar en el servidor csp para evitar vulnerabilidades



Nota. En la figura se muestra la configuración en el servidor para evitar vulnerabilidad es necesario configurar en el servidor la cabecera Content-Security-Policy Autoría propia

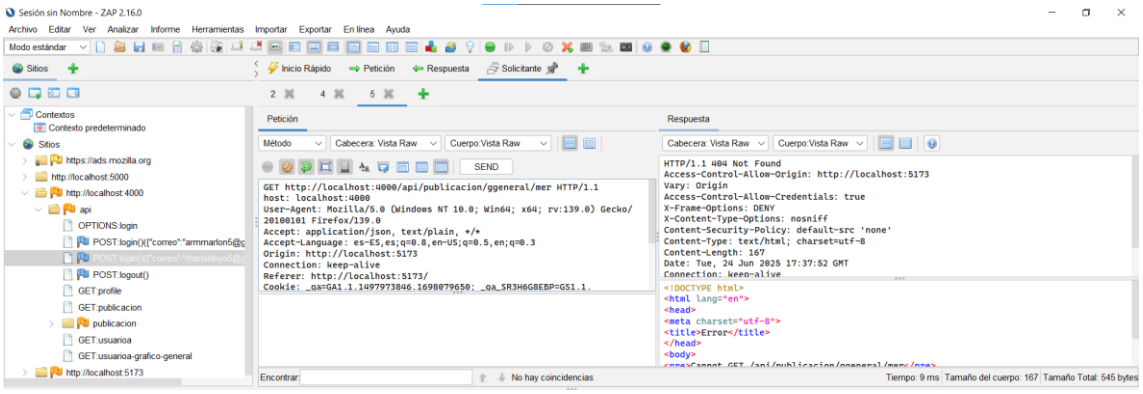
A09: Fallos de registro y monitoreo de seguridad

Figura 88
Pantalla de inicio del usuario normal



Nota. En la figura muestra el comprobar las rutas no permitidas, esto no permite la aplicación, ya que sí está controlado. Autoría propia

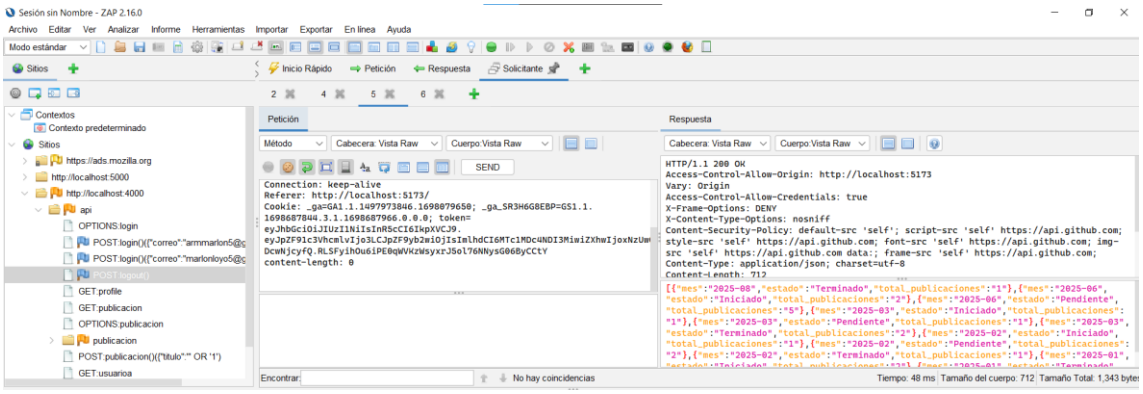
Figura 89
Intento de cambio de ruta



Nota. En la figura se muestra el intento de cambiar la ruta por algo más y no da resultado. Autoría propia

Figura 90

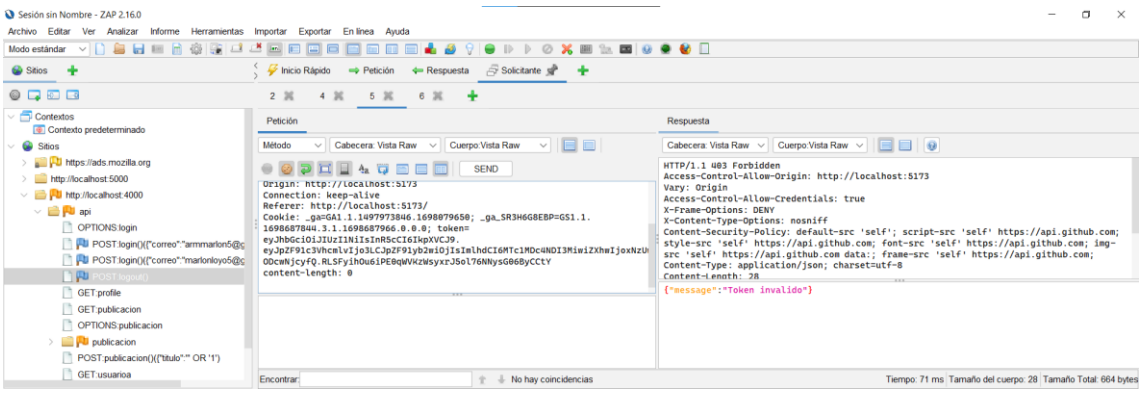
Inicio de la cuenta del usuario normal con el token valido.



Nota. En la figura se muestra el login se intentó con un token antiguo en el usuario normal. Autoría propia

Figura 91

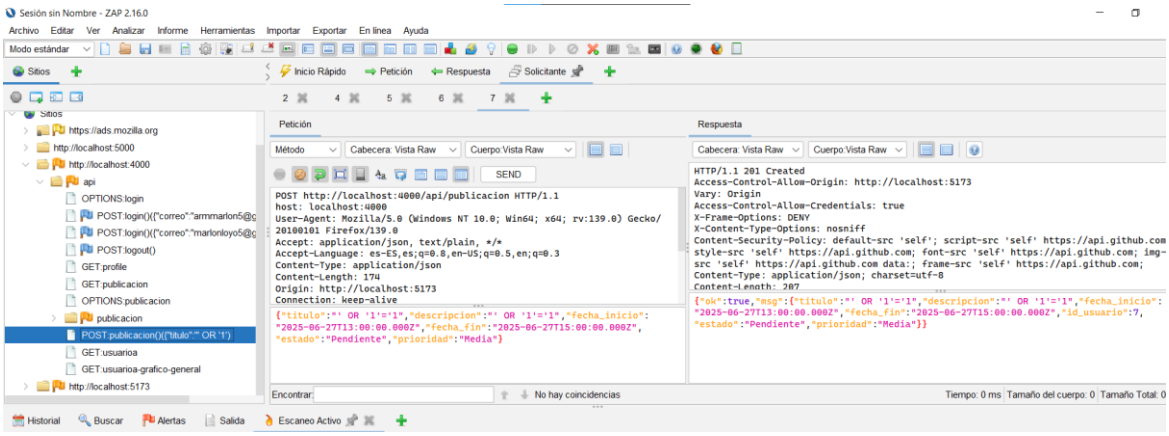
Intento de ingreso con token inválido



Nota. La figura muestra el intento de uso de un Token antiguo ya cuando se cerró la cuenta. Autoría propia

Figura 92

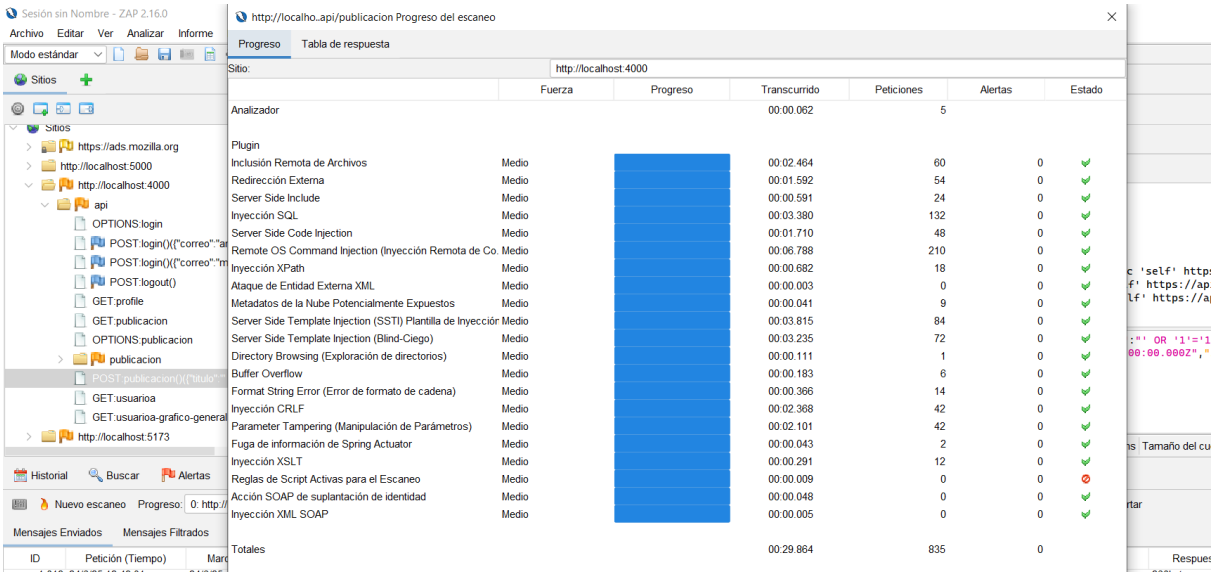
Intento de crear publicación con consultas sql y ver si trae información



Nota. La figura muestra el intento de crear una publicación con lenguaje sql. Autoría propia

Figura 93

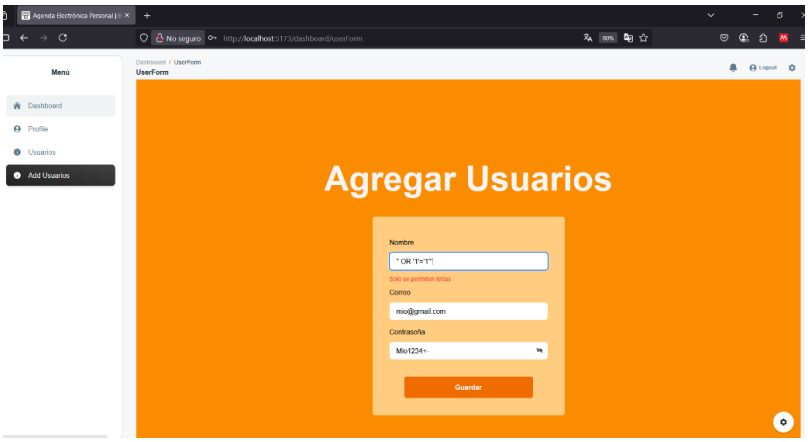
Pantalla de pruebas y resultados



Nota. En la figura muestra que se examinó y tampoco dio rastro de algún inconveniente. Autoría propia

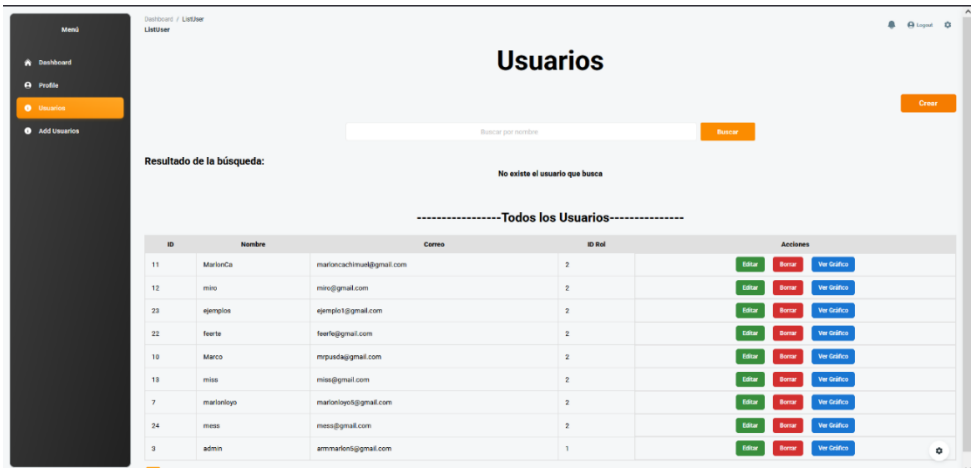
Figura 94

Intento de crear usuarios con sql



Nota. En la figura nos muestra el intentó crear un usuario y tampoco dejó crearlo. Autoría propia

Figura 95
Administración de usuarios por parte del administrador



Nota. En la figura nos muestra que el administrador controla todo sobre la información del usuario que sea necesario. Autoría propia

A10: Falsificación de solicitudes del lado del servidor (SSRF)

En nuestra aplicación no existe esta vulnerabilidad, ya que los datos que se suben solo son texto y no se sube alguna url, por lo que no aplica en este contexto.

3.3 Prototipo de mejora para puntos de acceso de seguridad

Tras identificar las vulnerabilidades presentes en el sistema, se mitiga o contrarresta sus efectos, priorizando aquellas clasificadas como de alto riesgo debido a su

impacto en la seguridad. Se recomienda implementar de inmediato las siguientes acciones para reducir el riesgo en los niveles Alto y Medio.

Tabla 20. Vulnerabilidad 1

Nombre	**Divulgación de hash – BCrypt**
Post	GET http://localhost:4000/api/usuarioa
Alert tags	OWASP_2017_A03
	OWASP_2021_A04
	CWE-200
Alert description	El servidor web reveló un hash. – BCrypt
Request	Request line and header section (546 bytes)
	Request body (0 bytes)
Response	**Status line and header section (632 bytes)**
	Response body (2272 bytes)
	{"ok":true,"msg":[{"id_usuario":3,"nombre":"marlon","correo":"marlonloyo5@gmail.com","contrasenia":"$2a$10$X4PjRaVbLq.yoOjd6iWOseEP5HbY1JD9bQWzY1RvFVJ7S3KqiTUbu","last_login":"2025-02-03T23:24:31.868Z","is_verified":false,"reset_password_token":"29b4933bb752780d3514627d88446adea788b17b","reset_password_expires_at":"2025-03-03T04:40:59.759Z","verification_token":"443755","verification_token_expires_at":"2025-03-04T03:47:24.239Z","created_at":"2025-02-03T23:24:31.868Z","updated_at":"2025-02-03T23:24:31.868Z","id_rol":2},
	{"id_usuario":1,"nombre":"AdminMar","correo":"armmarlon5@gmail.com","contrasenia":"$2a$10$CEoyYuuxTswENTj01ZSZsOtccrDJWtYt/idELH1ZsqCKrKC06UKG6","last_login":"2025-02-03T23:22:10.219Z","is_verified":false,"reset_password_token":null,"reset_password_expires_at":null,"verification_token":"181125","verification_token_expires_at":"2025-03-04T04:04:11.630Z","created_at":"2025-02-03T23:22:10.219Z","updated_at":"2025-02-03T23:22:10.219Z","id_rol":1},
	{"id_usuario":15,"nombre":"pez","correo":"pez@gmail.com","contrasenia":"$2a$10$Qxn12rPbWQ0/yYODQAnRn.E9QFNOxq5r0F4tJfeeZjKXp/lT3ke8W","last_login":"2025-02-24T02:52:51.173Z","is_verified":false,"reset_password_token":null,"reset_password_expires_at":null,"verification_token":null,"verification_token_expires_at":null,"created_at":"2025-02-24T02:52:51.173Z","updated_at":"2025-02-24T02:52:51.173Z","id_rol":2},
	{"id_usuario":2,"nombre":"mesa","correo":"mesa@gmail.com","contrasenia":"$2a$10$lfJTuRL51MAxIMDwSUeZxeJmKpBJv7l5wEbiZnp97HRh9w/IgKGCm","last_login":"2025-02-03T23:23:22.288Z","is_verified":false,"reset_password_token":null,"reset_password_expires_at":null,"verification_token":"779867","verification_token_expires_at":"2025-02-25T02:54:40.747Z","created_at":"2025-02-03T23:23:22.288Z","updated_at":"2025-02-03T23:23:22.288Z","id_rol":2},
	{"id_usuario":14,"nombre":"marlonca","correo":"marloncachimuel@gmail.com","contrasenia":"$2a$10$06HvT22gjHoA999OwoO9seiSC1sHH6byYIvXlDGnhEPqSQcLr/uIK","last_login":"2025-02-09T03:24:40.432Z","is_verified":false,"reset_password_token":null,"reset_password_expires_at":null,"verification_token":"199148","verification_token_expires_at":"2025-02-23T06:39:10.356Z","created_at":"2025-02-09T03:24:40.432Z","updated_at":"2025-02-09T03:24:40.432Z","id_rol":2}]}
Evidence	$2a$10$X4PjRaVbLq.yoOjd6iWOseEP5HbY1JD9bQWzY1RvFVJ7S3KqiTUbu
Solution	Asegurar que los hashes que se usan para proteger credenciales u otros recursos no sean filtrados por el servidor web o la base de datos.
	Normalmente no es necesario que los hashes de las contraseñas sean accesibles para el navegador web.

Explicación

La "Divulgación de Hash esto ocurre cuando una aplicación muestra accidentalmente valores hash en respuestas HTTP, o registros accesibles a un atacante. En nuestro caso Bcrypt donde la aplicación filtra esta información, lo que podría generar que un atacante intente recuperar contraseñas originales mediante ataques de fuerza bruta.

Impacto

El impacto de la divulgación de hashes de BCrypt es significativo, especialmente en sistemas donde la seguridad de las contraseñas es crucial:

1. **Riesgo de ataques de fuerza bruta**
    - Aunque Bcrypt resiste ataques de fuerza bruta debido a su lentitud y capacidad de configuración, pero si un atacante lograra obtener un hash, podría intentar descifrar con técnicas avanzadas.
2. **Compromiso de credenciales**
    - Si los usuarios usan sus credenciales en distintas plataformas, la filtración de un hash el atacante podría descifrarlo y probar en distintos servicios.
3. **Pérdida de confianza y cumplimiento normativo**
    - Filtrar hashes vulnera normativas como GDPR, ISO 27001 o OWASP ASVS, lo que generaría sanciones o pérdida de confianza de los usuarios.

Solución

En el presente desarrollo, la solución fue en el archivo model donde está la consulta sql para traer los usuarios donde se ve que expone los datos de los usuarios y además, este error también está en la función de búsqueda por nombre de un usuario.

Tabla 21. Solución propuesta

Vulnerabilidad
const findAllUsers = async () => {   const query = {   text: `   SELECT * FROM USUARIOS  `,   }

```
    const { rows } = await conpool.query(query)
    return rows
}
```

Corrección

```
const findAllUsers = async () => {
    const query = {
        text: `
        SELECT id_usuario, nombre, correo, id_rol FROM USUARIOS
        `,
    }
    const { rows } = await conpool.query(query)
    return rows
}
```

Vulnerabilidad

```
const findOneByNombre = async (nombre) => {
    const query = {
        text: `
        SELECT * FROM USUARIOS
        WHERE NOMBRE =$1`,
        values: [nombre]
    }
    const { rows } = await conpool.query(query)
    return rows[0]
}
```

Corrección

```
const findOneByNombre = async (nombre) => {
    const query = {
        text: `
        SELECT id_usuario, nombre, correo, id_rol
        FROM USUARIOS
        WHERE nombre = $1
        `,
        values: [nombre]
    };
    const { rows } = await conpool.query(query)
    return rows[0]
}
```

Tabla 22. Vulnerabilidad 2

Nombre	**Cabecera Content Security Policy (CSP) no configurada**
Post	GET http://localhost:5173/auth/sign-in
Alert tags	CWE-693
	OWASP_2021_A05
	OWASP_2017_A06
Alert description	La Política de seguridad de contenido (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques, incluidos Cross Site Scripting (XSS). Estos ataques se utilizan para todo, desde el robo de datos hasta la desfiguración del sitio o la distribución de malware. CSP proporciona un conjunto de encabezados HTTP estándar que permiten a los propietarios de sitios web declarar fuentes de contenido aprobadas que los navegadores deberían poder cargar en esa página; los tipos cubiertos son JavaScript, CSS, marcos HTML, fuentes, imágenes y objetos incrustados como applets de Java, ActiveX, archivos de audio y video.
Request	Request line and header section (366 bytes)
	Request body (0 bytes)
Response	**Status line and header section (251 bytes)**
	Response body (2093 bytes)
	<!DOCTYPE html> <html lang="en"> <head> <script type="module">

	import RefreshRuntime from "/@react-refresh" RefreshRuntime.injectIntoGlobalHook(window) window.$RefreshReg$ = () => {} window.$RefreshSig$ = () => (type) => type window.__vite_plugin_react_preamble_installed__ = true </script> <script type="module" src="/@vite/client"></script> <meta charset="UTF-8" /> <link rel="icon" type="image/svg+xml" href="/img/favicon.png" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <title>Agenda Electrónica Personal | By Mar</title> <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700;900&display=swap" rel="stylesheet" /> <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" integrity="sha512-xh6O/CkQoPOWDdYTDqeRdPCVd1SpvCA9XXcUnZS2FmJNp1coAFzvtCN9BmamE+4aHK8yyUHUSCcJHgXloTyT2A==" crossorigin="anonymous" referrerpolicy="no-referrer" /> <script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script> <!-- Nepcha Analytics (nepcha.com) --> <!-- Nepcha is a easy-to-use web analytics. No cookies and fully compliant with GDPR, CCPA and PECR. --> <script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script> </head> <body> <script type="module" src="/src/main.jsx"></script> </body> </html>
Solution	Asegúrese de que su servidor web, servidor de aplicaciones, balanceador de carga, etc. esté configurado para establecer la cabecera Content-Security-Policy.

Explicación

Política de Seguridad del Contenido o (CSP) - del inglés *Content Security Policy* - Es una capa de seguridad adicional que refuerza y mitiga algunos tipos de ataques como: Cross Site Scripting (XSS) y ataques de inyección de datos[69]. Estos tipos de ataques suelen ser usados para diferentes propósitos, desde robo de información hasta desfiguración de sitios o repartición de malware. Este tipo de ataques consiste en inyectar código de JavaScript en el sitio web del usuario, esto puede producirse de distintas maneras, estos pueden aprovechar brechas de seguridad en la programación del sitio o por librerías que se usen y no se actualicen con frecuentemente [70].

Impacto

El principal objetivo del CSP es mitigar y reportar ataques XSS. Los ataques XSS se aprovechan de la confianza del navegador en el contenido que recibe del servidor. El navegador de la víctima ejecutará los scripts maliciosos porque confía en la fuente del contenido, aun cuando dicho contenido no provenga de donde se supone [71].

- **Prevención de ataques de inyección de código:** CSP ayuda a evitar ataques de inyección de código, al restringir la ejecución de scripts no autorizados. Esto ayudará a que los atacantes se les dificulte insertar y ejecutar código malicioso en los sitios web, lo que mitiga a los usuarios de este tipo de ataques.
- **Integridad de los datos del usuario:** Además, podría mejorar la integridad de los datos del usuario para evitar la carga de recursos de fuentes no confiables. Esto permitirá proteger a los usuarios de phishing o de otros tipos de ataques que pueden mentir al usuario.
- **Mejora del rendimiento:** Ayuda a reducir la carga de recursos innecesarios que restringen los recursos que necesitan de verdad. Esto ayuda a mejorar el tiempo de carga del sitio y experiencia al usuario.
- **Restringir dominios no requeridos:** Restringe dominios donde se puede cargar contenido no necesario, el servidor puede detallar protocolos que especifican que todo el contenido debe cargarse usando HTTPS.

Solución

En el presente desarrollo, la solución fue aumentar ciertas configuraciones para evitar el problema de Cabecera Content Security Policy (CSP) y esto lo configuraremos en el archivo app.js del backend.

Esto nos ayuda en cuestiones como:

- Solo permite cargar recursos desde el mismo origen.
- Solo permite ejecutar scripts desde el mismo origen y desde api.github.com.
- Solo permite estilos del mismo origen y de api.github.com.
- Las fuentes solo pueden cargarse desde el mismo origen y api.github.com.
- Permite imágenes del mismo origen, api.github.com y datos en formato data.
- Solo permite cargar frames del mismo origen y api.github.com

Tabla 23. Solución propuesta

Vulnerabilidad

```
import express from 'express'
import morgan from 'morgan'
import cors from 'cors'
import usuarioRoutes from "./routes/usuario.routes.js"
import publicacionRoutes from "./routes/publicacion.routes.js"
import cookieParser from 'cookie-parser'

const app = express()
const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

app.use(cors({
   origin: FRONTEND_URL,
   credentials: true
}))
app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())

app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)
```

```
export default app;
```

Corrección

```
import express from 'express'
import morgan from 'morgan'
import cors from 'cors'

import usuarioRoutes from "./routes/usuario.routes.js"
import publicacionRoutes from "./routes/publicacion.routes.js"
import cookieParser from 'cookie-parser'

const app = express()

const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

// Configuración de CORS
app.use(cors({
  origin: FRONTEND_URL,
  credentials: true,
  methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS'],
  allowedHeaders: ['Content-Type', 'Authorization'],
}));

app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())

// Configura los encabezados CSP
app.use((req, res, next) => {
  res.setHeader(
    'Content-Security-Policy',
          "default-src 'self'; script-src 'self' https://api.github.com; style-src 'self' https://api.github.com; font-src 'self' https://api.github.com; img-src 'self' https://api.github.com data:; frame-src 'self' https://api.github.com;"
  );
  next();
});

// Ruta de proxy para redirigir solicitudes a la API de GitHub
app.use('/api/github', (req, res) => {
  const url = `https://api.github.com${req.url}`;
  request(url, (error, response, body) => {
    if (!error && response.statusCode === 200) {
      // Elimina o modifica los encabezados CSP de GitHub
      res.removeHeader('Content-Security-Policy');
      res.send(body);
    } else {
      res.status(response.statusCode).send(error);
    }
  });
});
```

```
app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)

export default app;
```

Tabla 24. Vulnerabilidad 3

Nombre	**Falta de cabecera Anti-Clickjacking**
Post	GET http://localhost:5173/auth/sign-in
Alert tags	WSTG-v42-CLNT-09
	OWASP_2021_A05
	OWASP_2017_A06
	CWE-1021
Alert description	La respuesta no protege contra ataques de "ClickJacking". Debes incluir Content-Security-Policy con la directiva "frame-ancestors" o X-Frame-Options.
Request	Request line and header section (366 bytes)
	Request body (0 bytes)
Response	**Status line and header section (251 bytes)**
	Response body (2272 bytes)
	<!DOCTYPE html> <html lang="en"> <head> <script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime.injectIntoGlobalHook(window) window.$RefreshReg$ = () => {} window.$RefreshSig$ = () => (type) => type window.__vite_plugin_react_preamble_installed__ = true </script> <script type="module" src="/@vite/client"></script> <meta charset="UTF-8" /> <link rel="icon" type="image/svg+xml" href="/img/favicon.png" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" />

	<title>Agenda Electrónica Personal | By Mar</title> <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700;900&display=swap" rel="stylesheet" /> <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" integrity="sha512-xh6O/CkQoPOWDdYTDqeRdPCVd1SpvCA9XXcUnZS2FmJNp1coAFzvtCN9BmamE+4aHK8yyUHUSCcJHgXloTyT2A==" crossorigin="anonymous" referrerpolicy="no-referrer" /> <script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script> <!-- Nepcha Analytics (nepcha.com) --> <!-- Nepcha is a easy-to-use web analytics. No cookies and fully compliant with GDPR, CCPA and PECR. --> <script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script> </head> <body> <script type="module" src="/src/main.jsx"></script> </body> </html>
Parameter	x-frame-options Los navegadores web modernos admiten las cabeceras HTTP Content-Security-Policy y X-Frame-Options. Asegúrese de que una de ellas está configurada en todas las páginas web devueltas por su sitio/aplicación.
Soluction	Si espera que la página esté enmarcada solo por páginas en su servidor (por ejemplo, si forma parte de un FRAMESET), utilice SAMEORIGIN; de lo contrario, si no espera que la página esté enmarcada, utilice DENY. Alternativamente, considere implementar la directiva "frame-ancestors" de la Política de Seguridad de Contenidos.

Explicación

EL objetivo de este ataque es engañar a los usuarios del sitio web para realizar acciones normales pero que dentro de estas están ocultas otros sitios web o botones que tienen intenciones maliciosas contra el usuario que lo está usando. Debido a este ataque también se lo conoce como reparación de UI [72].

Impacto

No tener una política Anti-Clickjacking adecuada puede causar varios problemas de seguridad:

1. **Robo de información sensible**
    - Un atacante puede engañar a un usuario para que realice clic en botones de "aceptar" en transacciones bancarias o exponer información personal sin darse cuenta.
2. **Secuestro de sesiones o cuentas**
    - Si la aplicación permite acciones sensibles como cambio de contraseña o confirmar pagos con un clic, un atacante puede explotar este error para obtener acceso.
3. **Compromiso de cuentas de usuario**
    - Los atacantes pueden inducir a los usuarios a activar opciones que les den control sobre sus cuentas sin que se den cuenta [73].

Solución

En esta implementación, la solución fue aumentar ciertas configuraciones para evitar el problema de cabecera Anti-Clickjacking y esto lo configuraremos en el archivo app.js del backend.

Tabla 25. Solución propuesta

Vulnerabilidad
import express from 'express' import morgan from 'morgan'

```
import cors from 'cors'

import usuarioRoutes from "./routes/usuario.routes.js"
import publicacionRoutes from "./routes/publicacion.routes.js"
import cookieParser from 'cookie-parser'

const app = express()
const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

app.use(cors({
   origin: FRONTEND_URL,
   credentials: true
}))
app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())

app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)

export default app;
```

Corrección

```
import express from 'express'
import morgan from 'morgan'
import cors from 'cors'

import usuarioRoutes from "./routes/usuario.routes.js"
import publicacionRoutes from "./routes/publicacion.routes.js"
import cookieParser from 'cookie-parser'

const app = express()
const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

// Configuración de CORS
app.use(cors({
  origin: FRONTEND_URL,
  credentials: true,
  methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS'],
  allowedHeaders: ['Content-Type', 'Authorization'],
}));

app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())

// Cabeceras de seguridad
app.use((req, res, next) => {
  res.header('X-Frame-Options', 'DENY'); // Protección contra Clickjacking
  res.header('X-Content-Type-Options', 'nosniff'); // Evita MIME type sniffing
  next();
});
```

```
app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)
export default app;
```

Tabla 26. Vulnerabilidad 4

Nombre	**Inclusión de archivos fuente JavaScript entre dominios**
Post	GET http://localhost:5173/auth/sign-in
Alert tags	OWASP_2021_A08
	CWE-829
Alert description	La página incluye uno o más archivos de script de un dominio de terceros.
Request	Request line and header section (366 bytes)
	Request body (0 bytes)
Response	**Status line and header section (251 bytes)**
	Response body (2272 bytes)
	<!DOCTYPE html> <html lang="en"> <head> <script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime.injectIntoGlobalHook(window) window.$RefreshReg$ = () => {} window.$RefreshSig$ = () => (type) => type window.__vite_plugin_react_preamble_installed__ = true </script> <script type="module" src="/@vite/client"></script> <meta charset="UTF-8" /> <link rel="icon" type="image/svg+xml" href="/img/favicon.png" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <title>Agenda Electrónica Personal | By Mar</title> <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700;900&display=swap" rel="stylesheet" /> <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" integrity="sha512-xh6O/CkQoPOWDdYTDqeRdPCVd1SpvCA9XXcUnZS2FmJNp1coAFzvtCN9BmamE+4aHK8yyUHUSCcJHgXloTyT2A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

	<script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script> <!-- Nepcha Analytics (nepcha.com) --> <!-- Nepcha is a easy-to-use web analytics. No cookies and fully compliant with GDPR, CCPA and PECR. --> <script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script> </head> <body> <script type="module" src="/src/main.jsx"></script> </body> </html>
Parameter	https://api.nepcha.com/js/nepcha-analytics.js
Evidence	<script defer data-site="YOUR_DOMAIN_HERE" src="https://api.nepcha.com/js/nepcha-analytics.js"></script>
Solution	Si espera que la página esté enmarcada solo por páginas en su servidor (por ejemplo, si forma parte de un FRAMESET), utilice SAMEORIGIN; de lo contrario, si no espera que la página esté enmarcada, utilice DENY. Alternativamente, considere implementar la directiva "frame-ancestors" de la Política de Seguridad de Contenidos.

Explicación

Incluir archivos de JavaScript fuera del dominio es una advertencia de seguridad que puede afectar a una aplicación web que use archivos externos terceros de JavaScript. Si estos externos tienen contenido malicioso o vulnerabilidades intencionales o involuntarias, se puede ejecutar en la aplicación web que los uso y provocando que esta sea víctima a ataques. Esto suelo ocurrir cuando JavaScript externo no está validado [74].

Impacto

No restringir la inclusión de archivos JavaScript entre dominios puede provocar varios problemas de seguridad:

- Posible ejecución de JavaScript malicioso.
- Posible manipulación de datos del usuario y fuga.
- Posible cambio de funcionalidad y redirección de datos.
- Infección por malware.

Solución

En esta implementación, la solución fue aumentar ciertas configuraciones para evitar el problema de cabecera Anti-Clickjacking y esto lo configuraremos en el archivo app.js del backend.

Tabla 27. Solución propuesta

Vulnerabilidad
import express from 'express' import morgan from 'morgan' import cors from 'cors' import usuarioRoutes from "./routes/usuario.routes.js" import publicacionRoutes from "./routes/publicacion.routes.js" import cookieParser from 'cookie-parser' const app = express() const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

```
app.use(cors({
   origin: FRONTEND_URL,
   credentials: true
}))
app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())


app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)


export default app;
```

Corrección

```
import express from 'express'
import morgan from 'morgan'
import cors from 'cors'

import usuarioRoutes from "./routes/usuario.routes.js"
import publicacionRoutes from "./routes/publicacion.routes.js"
import cookieParser from 'cookie-parser'

const app = express()
const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

// Configuración de CORS
app.use(cors({
  origin: FRONTEND_URL,
  credentials: true,
  methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS'],
  allowedHeaders: ['Content-Type', 'Authorization'],
}));

app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())


// Cabeceras de seguridad
app.use((req, res, next) => {
  res.header('X-Frame-Options', 'DENY'); // Protección contra Clickjacking
  res.header('X-Content-Type-Options', 'nosniff'); // Evita MIME type sniffing
  next();
});
```

```
app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)

export default app;
```

Tabla 28. Vulnerabilidad 5

Nombre	**Revelación de IP privada**
Post	GET http://localhost:5173/node_modules/.vite/deps/react-icons_fa.js?v=27fbb789
Alert tags	OWASP_2021_A01
	OWASP_2017_A03
	CWE-497
Alert description	Se ha encontrado una IP privada (como 10.x.x.x, 172.x.x.x, 192.168.x.x) o un nombre de host privado de Amazon EC2 (por ejemplo, ip-10-0-56-78) en el cuerpo de la respuesta HTTP. Esta información podría ser útil para futuros ataques dirigidos a sistemas internos.
Request	Request line and header section (366 bytes)
	Request body (0 bytes)
Response	**Status line and header section (251 bytes)**
	Response body (2272 bytes)

	import {
	  require_react
	} from "/node_modules/.vite/deps/chunk-HS5T2ZWL.js?v=27fbb789";
	import {
	  __toESM
	} from "/node_modules/.vite/deps/chunk-AUZ3RYOM.js?v=27fbb789";
	
	// node_modules/react-icons/lib/iconBase.mjs
	var import_react2 = __toESM(require_react(), 1);
	
	// node_modules/react-icons/lib/iconContext.mjs
	var import_react = __toESM(require_react(), 1);
	var DefaultContext = {
	  color: void 0,
	  size: void 0,
	  className: void 0,
	  style: void 0,
	  attr: void 0

	};// node_modules/react-icons/fa/index.mjs
	function Fa500Px(props) {
	  return GenIcon({ "tag": "svg", "attr": { "viewBox": "0 0 448 512" }, "child": [{ "tag": "path", "attr": { "d": "M103.3 344.3c-6.5-14.2-6.9-18.3 7.4-23.1 25.6-8 8 9.2 43.2 49.2h.3v-93.9c1.2-50.2 44-92.2 97.7-92.2 53.9 0 97.7 43.5 97.7 96.8 0 63.4-60.8 113.2-128.5 93.3-10.5-4.2-2.1-31.7 8.5-28.6 53 0 89.4-10.1 89.4-64.4 0-61-77.1-89.6-116.9-44.6-23.5 26.4-17.6 42.1-17.6 157.6 50.7 31 118.3 22 160.4-20.1 24.8-24.8 38.5-58 38.5-93 0-35.2-13.8-68.2-38.8-93.3-24.8-24.8-57.8-38.5-93.3-38.5s-68.8 13.8-93.5 38.5c-.3.3-16 16.5-21.2 23.9l-.5.6c-3.3 4.7-6.3 9.1-20.1 6.1-6.9-1.7-14.3-5.8-14.3-11.8V20c0-5 3.9-10.5 10.5-10.5h241.3c8.3 0 8.3 11.6 8.3 15.1 0 3.9 0 15.1-8.3 15.1H130.3v132.9h.3c104.2-109.8 282.8-36 282.8 108.9 0 178.1-244.8 220.3-310.1 62.8zm63.3-260.8c-.5 4.2 4.6 24.5 14.6 20.6C306 56.6 384 144.5 390.6 144.5c4.8 0 22.8-15.3 14.3-22.8-93.2-89-234.5-57-238.3-38.2zM393 414.7C283 524.6 94 475.5 61 310.5c0-12.2-30.4-7.4-28.9 3.3 24 173.4 246 256.9 381.6 121.3 6.9-7.8-12.6-28.4-20.7-20.4zM213.6 306.6c0 4 4.3 7.3 5.5 8.5 3 3 6.1 4.4 8.5 4.4 3.8 0 2.6.2 22.3-19.5 19.6 19.3 19.1 19.5 22.3 19.5 5.4 0 18.5-10.4 10.7-18.2L265.6 284l18.2-18.2c6.3-6.8-10.1-21.8-16.2-15.7L249.7 268c-18.6-18.8-18.4-19.5-21.5-19.5-5 0-18 11.7-12.4 17.3L234 284c-18.1 17.9-20.4 19.2-20.4 22.6z" }, "child": [] }] })(props);

Evidence	10.1.9.34
Solucion	Eliminar la dirección IP privada del cuerpo de la respuesta HTTP. Para los comentarios, utilice comentarios JSP/ASP/PHP en lugar de comentarios HTML/JavaScript que pueden ser vistos por los navegadores de los clientes.

Explicación

La revelación de IP privada ocurre cuando una aplicación expone direcciones IP internas en las respuestas HTTP, encabezados, mensajes de error o código fuente accesible públicamente [75].

Impacto

La revelación de direcciones IP privadas puede ser explotada por atacantes para:

1. **Facilitación de ataques internos**
    - Si un atacante compromete un servidor accesible desde internet, conocer las IPs privadas le permite moverse lateralmente dentro de la red.
2. **Explotación de vulnerabilidades internas**
    - Algunas redes permiten conexiones internas sin autenticación estricta, lo que podría facilitar el acceso a servicios no protegidos.
3. **Filtración de información sensible en logs y respuestas**
    - Mostrar direcciones IP internas en errores o encabezados HTTP puede ayudar a atacantes a identificar puntos débiles en la infraestructura.

Solución

En el presente desarrollo, la solución fue aumentar ciertas configuraciones para evitar el problema y esto lo configuraremos en el archivo app.js del backend.

Tabla 29. Solución propuesta

Vulnerabilidad
import express from 'express' import morgan from 'morgan' import cors from 'cors' import usuarioRoutes from "./routes/usuario.routes.js" import publicacionRoutes from "./routes/publicacion.routes.js" import cookieParser from 'cookie-parser'

```
const app = express()
const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"


app.use(cors({
   origin: FRONTEND_URL,
   credentials: true
}))
app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())


app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)


export default app;
```

Corrección

```
import express from 'express'
import morgan from 'morgan'
import cors from 'cors'

import usuarioRoutes from "./routes/usuario.routes.js"
import publicacionRoutes from "./routes/publicacion.routes.js"
import cookieParser from 'cookie-parser'

const app = express()

// Deshabilitar el encabezado X-Powered-By
app.disable('x-powered-by');


const FRONTEND_URL = process.env.FRONTEND_URL || "http://localhost:5173"

// Configuración de CORS
app.use(cors({
  origin: FRONTEND_URL,
  credentials: true,
  methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS'],
  allowedHeaders: ['Content-Type', 'Authorization'],
}));

app.use(morgan('dev'));
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cookieParser())
```

```
// Cabeceras de seguridad
app.use((req, res, next) => {
  res.header('X-Frame-Options', 'DENY'); // Protección contra Clickjacking
  res.header('X-Content-Type-Options', 'nosniff'); // Evita MIME type sniffing
  next();
});
app.use("/api", usuarioRoutes)
app.use("/api", publicacionRoutes)

export default app;
```

3.4 Análisis de resultados luego de las correcciones

Una vez aplicadas las correcciones a ciertos errores en la aplicación web se realizará una nueva evaluación con la que se obtienen los siguientes resultados.

En esta tabla se muestra el número de alertas de cada tipo de alerta, junto con el nivel de riesgos.

Figura 96

Análisis de vulnerabilidades OWASP ZAP después de las correcciones

		Confidence				
		Confirmado por Usuario	Alta	Media	Baja	Total
Risk	Alto	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)
	Medio	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)
	Bajo	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)
	Informativo	0 (0,0 %)	0 (0,0 %)	2 (50,0 %)	2 (50,0 %)	4 (100,0 %)
	Total	0 (0,0 %)	0 (0,0 %)	2 (50,0 %)	2 (50,0 %)	4 (100%)

Nota. En la figura nos muestra el detalle del tipo de alerta identificado que se encontró durante la evaluación del aplicativo, indicando su nivel de riesgo del tipo y la cantidad encontrada. Autoría propia

Tabla 30. Tipos de alertas de seguridad identificadas después de las correcciones

Tipo de Alerta	Riesgo	Cantidad
Divulgación de Información - Información sensible en URL	Informativo	1
Divulgación de información - Comentarios sospechosos	Informativo	5
Loosely Scoped Cookie	Informativo	4
Respuesta de Gestión de Sesión Identificada	Informativo	4

El análisis con OWASP ZAP reveló algunas vulnerabilidades importantes que se deben abordar de manera rápida. La implementación de las recomendaciones proporcionadas fortalecerá la seguridad de la aplicación y reducirá significativamente el riesgo de exposición a amenazas.

A continuación se presentan varias tablas que muestran, para cada fase de Owasp top ten la herramienta utilizada, su estado y una corta observación.

Tabla 31. Análisis de la categoría A01: Control de acceso roto.

OWASP	Herramienta utilizada	Estado	Observación
A01: Control de acceso roto (Broken Access Control)	OWASP ZAP	Aplicado	Ya se habían aplicado restricciones para cada usuario como: token y rol.

Tabla 32. Análisis de la categoría A02: Fallos criptográficos

OWASP	Herramienta utilizada	Estado	Observación
02: Fallos criptográficos	OWASP ZAP	Detectado y corregido	Se aplicó bcryptjs para las contraseñas y los tokens tiene tiempo de expiración y se aumentó httpOnly para evitar accesos de javascript en el navegador y sameSite para evitar ataques CSRF

Tabla 33. Análisis de la categoría A03: Inyección.

OWASP	Herramienta utilizada	Estado	Observación
A03: Inyecciones (Injection)	Owasp ZAP	Detectado y corregido	Se detectó posibilidad de XSS y se usó en campos de entrada sanitize-html

Tabla 34. Análisis de la categoría A04: Diseño inseguro.

OWASP	Herramienta utilizada	Estado	Observación
A04: Diseño inseguro	Manual	Aplicado	Se verificó que no existe acceso a información de un usuario a otro.

Tabla 35. Análisis de la categoría A05: Configuración incorrecta de seguridad.

OWASP	Herramienta utilizada	Estado	Observación
A05: Configuración incorrecta de seguridad	OWASP ZAP	Detectado y corregido	Se configuró encabezados necesarios como: X-Content-Type-Options y anti clickjacking.

Tabla 36. Análisis de la categoría A06: Componentes vulnerables o desactualizados.

OWASP	Herramienta utilizada	Estado	Observación
A06: Componentes vulnerables o desactualizados	OWASP ZAP y Burp Suite	Aplicado	Se actualizó algunos paquetes que eran vulnerables como: sanitize-html, jsonwebtoken, browserslist, etc.

Tabla 37. Análisis de la categoría A07: Fallos de identificación y autentificación.

OWASP	Herramienta utilizada	Estado	Observación
A07: Fallos de identificación y autenticación	OWASP ZAP	Aplicado	Cumple con lo necesario para encriptar la contraseña y validación de los tokens.

Tabla 38. Análisis de la categoría A08: Fallos en la integridad del software y los datos.

OWASP	Herramienta utilizada	Estado	Observación
A08: Fallos en la integridad del software y los datos	OWASP ZAP	Aplicado	Cumple con lo necesario que es que esté configurado bien en recursos externos con integrity y crossorigin

Tabla 39. Análisis de la categoría A09: Fallos de registro y monitoreo de seguridad.

OWASP	Herramienta utilizada	Estado	Observación
A09: Fallos de registro y monitoreo de seguridad	OWASP ZAP	Aplicado	Las rutas de accesos a distintos usuarios están controladas, y para parámetros en formularios está controlado para evitar inyección sql

Tabla 40. Análisis de la categoría A10: Falsificación de solicitudes del lado del servidor.

OWASP	Herramienta utilizada	Estado	Observación
A10: Falsificación de solicitudes del lado del servidor (SSRF)	Manual	No aplica	La aplicación no realiza solicitudes del lado del servidor hacia otros servicios internos.

Estas tablas especificas nos permiten evidenciar y organizar las evidencias de las pruebas de seguridad. Gracias a esto fue posible identificar las categorías con mayor impacto en la aplicación web.

3.5 Análisis de vulnerabilidades detectadas por categoría owasp antes y después de la implementación segura.

Tabla 41. Análisis de vulnerabilidades detectadas antes y después

Categoría OWASP	Vulnerabilidades antes	Vulnerabilidades después
A01: Control de acceso roto	1	0
02: Fallos criptográficos	1	0
A03: Inyecciones (Injection)	4	0
A04: Diseño inseguro	0	0
A05: Configuración incorrecta de seguridad	2	0
A06: Componentes vulnerables o desactualizados	3	0
A07: Fallos de identificación y autenticación	0	0
A08: Fallos en la integridad del software y los datos	1	0
A09: Fallos de registro y monitoreo de seguridad	0	0
A10: Falsificación de solicitudes del lado del servidor (SSRF)	0	0

Los resultados muestran una mejora significativa en la seguridad de la aplicación web tras la implementación de las prácticas seguras. Se encontraron inicialmente algunas vulnerabilidades en las siguientes categorías:

- **Control de acceso roto (A01):** Se mostraba ip privadas de donde venían iconos de react por lo que para solucionarlo se configuró en el servidor para que no mire esta información x-powered-by.
- **Fallos criptográficos (A02):** La falta de configuración de httpOnly y sameSite permitía que ataques CSRF.
- **Inyección (A03):** Se detectaron posibles ataques XSS en campos de formularios como el de publicaciones título o descripción. Se logró mitigar esto con el uso de la librería sanitize-html para limpiar los datos antes de guardar.

- **Configuraciones incorrectas de seguridad (A05):** No se habían configurado correctamente headers de seguridad. Para mitigar esto, se configuró correctamente la cabecera content Security Polity (CSP) y la cabecera de Anti-Clickjacking.
- **Componentes vulnerables o desactualizados (A06):** Algunos componentes tenían nuevas actualizaciones por lo se procedió al actualizar para evitar posibles problemas de ataques.
- **Fallos en la integridad del software y datos (A08):** Se podía por el Frontend hacer la inclusión de archivos fuente JavaScript entre dominios y para mitigar esto se configuró la cabecera Anti-clickjacking.

3.6 Comparativa entre OWASP ZAP y Burp Suite

Owasp Zap

Tabla 42. Resultados del análisis de vulnerabilidad por la herramienta de owasp zap

Tipo de Alerta	Riesgo	Cantidad
Divulgación de hash – BCrypt	Alto	6
Cabecera Content Security Policy (CSP) no configurada	Medio	1
Falta de cabecera Anti-Clickjacking	Medio	1
Inclusión de archivos fuente JavaScript entre dominios	Bajo	1
Revelación de IP privada	Bajo	1
Aplicación Web Moderna	Informativo	1
Divulgación de Información - Información sensible en URL	Informativo	1
Divulgación de información - Comentarios sospechosos	Informativo	11
Respuesta de Gestión de Sesión Identificada	Informativo	3

Tabla 43. Resultados del análisis de vulnerabilidad por la herramienta de burp suite 1

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	1	0	0	1
	Information	6	0	0	6
	False Positive	0	0	0	0

Tabla 44. Resultados del análisis de vulnerabilidad por la herramienta de Burp suite.

Tipo de Alerta	Riesgo	Cantidad
	Alto	0
	Medio	0
	Medio	0
Política de seguridad de contenido (CSP)	Bajo	1
Política de Seguridad de Contenido se basa en una lista de permitidos para controlar la carga de recursos basada en scripts	Informativo	4
Divulgación de información - Comentarios sospechosos	Informativo	2

Tabla 45. Comparativa de herramientas y sus resultados.

Aspecto	OWASP ZAP	Burp Suite
Licencia	Gratuita	Gratuita limitada / Pro de pago
Detección automática	Sencilla	Sencilla
Escaneo automático	Si	Si
Escaneo pasivo/activo	Ambos	Ambos
Interfaz y usabilidad	Fácil	Media
Resultados en tu prueba	6 Graves, 2 media, 2 bajo y 16 informativo	1 bajo, 6 informativo

Ambas herramientas permitieron detectar vulnerabilidades como: Control de acceso, inyección sql, configuraciones incorrectas de seguridad, etc..

CONCLUSIONES

- La revisión de literatura permitió la selección del patrón de diseño MVC para el desarrollo del proyecto debido a su facilidad de organización, mantenimiento y escalabilidad. Su clara separación de responsabilidades facilita la gestión del código, brindando una mejor estructuración y reutilización de componentes. Además, este patrón o variantes de este son muy utilizados en las industrias del desarrollo de software, lo que garantiza compatibilidad con frameworks modernos y un gran soporte por la comunidad. Estas características hacen que el patrón MVC una elección sólida para el desarrollo.

- El uso de un patrón de diseño MVC y aplicando estándares de seguridad garantiza un producto estructurado y confiable. Permitiéndonos una clara separación de responsabilidades, facilitando la escalabilidad de nuestra aplicación. Además, incorporar estándares de seguridad fortaleció la aplicación frente a vulnerabilidades comunes, asegurando la protección de datos.

- La evaluación de la aplicación a través de las herramientas Owasp Zap y Burp Suite permitió identificar y mitigar vulnerabilidades críticas que comprometen la integridad, confidencialidad de los datos. Esto nos demuestra la importancia de aplicar normas de seguridad y herramientas. Esto no solo fortaleció la seguridad de la aplicación, sino que también es necesario incorporar el análisis de vulnerabilidades en el ciclo de desarrollo.

- El empleo de herramientas como Owasp Zap y Burp Suite represento una experiencia valiosa y enriquecedora, ya que nos facilitaron el análisis de vulnerabilidades y permitirá para futuros proyectos usarlo y evitar estos riesgos similares.

- El usar estos dos tipos de herramientas nos proporcionó cierta experiencia y cuándo usarlas, cada una de estas. Owasp Zap nos resultó ser más fácil de usar y es recomendada para un inicio en el análisis de vulnerabilidades y, en cambio, Burp Suite es más compleja, para hacer pruebas con más profundidad y para personas con mayor nivel de conocimiento.

- El desarrollo de la aplicación resultó una experiencia buena, ya que realizar un proyecto siguiendo una arquitectura y proceso me permitió distribuir mejor cómo realizar el software y no cometer tantos errores y además de seguir aspectos de seguridad como el top ten de owasp nos ayuda a evitar problemas de seguridad y con ello desarrollar un software seguro y óptimo.

RECOMENDACIONES

- Para futuras actualizaciones del prototipo de agenda electrónica, se recomienda continuar utilizando el patrón de diseño MVC debido a la facilidad de mantenimiento y estabilidad. Además, se sugiere explorar otras variantes de este patrón como el MVVM o MVP si se da el caso de tener una mayor separación de responsabilidades.

- Se sugiere continuar aplicando el patrón de diseño MVC en futuros proyectos, dada su efectividad en la organización y mantenimiento del código. Además, es fundamental mantenerse actualizado en estándares de seguridad, considerando las nuevas vulnerabilidades y siguiendo a OWASP y otras entidades especializadas. Si el proyecto continuará en desarrollo, se sugiere realizar pruebas de seguridad periódicas para garantizar que la aplicación mantenga robusta y confiable.

- A partir de los resultados obtenidos, es recomendado integrar pruebas de seguridad automatizadas con OWASP ZAP u otras herramientas en el ciclo de desarrollo. Además, es fundamental capacitar continuamente al desarrollador en prácticas de OWASP y promover la cultura de seguridad alineada con estándares como OWASP ASVS, ISO 27001, ISO 27034, entre otras. Esto permitirá mitigar vulnerabilidades de forma proactiva, garantizando la protección continua de la información y fortaleciendo la resistencia de la aplicación ante posibles amenazas.

- Para futuras investigaciones, se recomienda extender la evaluación de aplicaciones web o aplicaciones móviles con herramientas como Owasp Zap, Burp Suite y otras.

- Se recomienda que, al seleccionar las herramientas para el desarrollo de la aplicación, se realicen pruebas básicas, como operaciones (CRUD), para evaluar la facilidad de implementar para el back-end. Del mismo modo, para el Front-end donde seleccionar la plantilla es necesario comprobar su funcionamiento, arquitectura, tamaño del archivo, intuitiva y facilidad de uso para el usuario

común. Estas métricas son esenciales para garantizar que la plantilla seleccionada cumpla con las necesidades requeridas.

REFERENCIAS

[1] D. Herrera, “Seguridad en aplicaciones web: Mejores prácticas y herramientas.” Accessed: Mar. 16, 2025. [Online]. Available: https://www.hostinger.com/es/tutoriales/seguridad-en-aplicaciones-web

[2] Mesías Valencia, Juan José, Cevallos Muñoz, and Fausto Danilo, “Incidencia de los patrones de diseño de software en la seguridad de aplicaciones web,” *MQRInvestigar* , vol. 8, no. 1, pp. 236–259, Jan. 2024, doi: 10.56048/MQR20225.8.1.2024.236-259.

[3] N. Gutiérrez, “Estadísticas de ciberseguridad que debes conocer.” Accessed: Mar. 16, 2025. [Online]. Available: https://preyproject.com/es/blog/estadisticas-seguridad-informatica

[4] S. Valbuena, “Panorama cibernético 2023: América Latina bajo asedio de los criminales por aumento de ataques - Infobae.” Accessed: Mar. 16, 2025. [Online]. Available: https://www.infobae.com/tecno/2023/08/24/panorama-cibernetico-2023-america-latina-bajo-asedio-de-los-criminales-por-aumento-de-ataques/

[5] Kaspersky, “El adware, troyanos bancarios brasileros y el spyware entre las mayores amenazas.” Accessed: Mar. 16, 2025. [Online]. Available: https://latam.kaspersky.com/about/press-releases/brasil-mexico-y-ecuador-los-principales-blancos-de-ataques-a-dispositivos-moviles-en-la-region

[6] M. Scrum, “2020-Scrum-Guide-Spanish-Latin-South-American.”

[7] M. Hernández, “Escaneo de vulnerabilidades automático con OWASP ZAP.” Accessed: Mar. 18, 2025. [Online]. Available: https://academy.seguridadcero.com.pe/blog/escaneo-vulnerabilidades-autom%C3%A1tico-OWASP-ZAP

[8] ONU, “Infraestructura - Desarrollo Sostenible.” Accessed: Nov. 04, 2023. [Online]. Available: https://www.un.org/sustainabledevelopment/es/infrastructure/

[9] Secretaria Nacional de Planificación, “Plan de Creación de Oportunidades 2021 2025.pdf | Enhanced Reader.”

[10] E. Muñoz, “La importancia de la arquitectura de software: cómo los patrones de diseño pueden mejorar la calidad del software.” Accessed: Mar. 18, 2025. [Online]. Available: https://es.linkedin.com/pulse/la-importancia-de-arquitectura-software-c%C3%B3mo-los-del-manr%C3%ADquez-mu%C3%B1oz

[11] M. Martínez, “Qué son los Patrones de Diseño de software / Design Patterns.” Accessed: Mar. 18, 2025. [Online]. Available: https://profile.es/blog/patrones-de-diseno-de-software/#Factory_Method

[12] S. Luján and E. Valarezo, “Aplicaciones Web,” May 2014. Accessed: Jan. 04, 2024. [Online]. Available:

https://rua.ua.es/dspace/bitstream/10045/36735/1/Aplicaciones%20Web%20-%20Patrones%20de%20dise%C3%B1o.pdf

[13] S. Bin Uzayr, "Software Design Patterns," 2022. doi: https://doi.org/10.1201/9781003308461.

[14] D. Spiridione, "Patrones Estructurales - Design Patterns in Go," Gitbook. Accessed: Jan. 25, 2024. [Online]. Available: https://daniel-m-spiridione.gitbook.io/designpatternsingo/parte2/patrones/estructurales

[15] N. Soto, "¿Qué son los patrones de diseño en programación?" Accessed: Jan. 04, 2024. [Online]. Available: https://craft-code.com/que-son-los-patrones-de-diseno/

[16] M. Martínez, "Qué son los Patrones de Diseño de software / Design Patterns," Profile. Accessed: Jan. 25, 2024. [Online]. Available: https://profile.es/blog/patrones-de-diseno-de-software/#Factory_Method

[17] J. Abanto and O. Gonzales, "Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura," *Universidad Peruana Unión*, Dec. 2019, Accessed: Jan. 04, 2024. [Online]. Available: https://repositorio.upeu.edu.pe/bitstream/handle/20.500.12840/2525/Jesus_Trabajo_Bachillerato_2019.pdf?sequence=1&isAllowed=y

[18] G. S. Bermúdez, I. M. Jiménez, and L. R. Rodríguez, "USO DE PATRONES DE DISEÑO: UN CASO PRÁCTICO," *Ingeniería*, vol. 22, no. 2, pp. 45–59, Jun. 2013, doi: 10.15517/RING.V22I2.8220.

[19] M. D. L. Acosta Yerovi, "Estudio de patrones de diseño en plataforma Java Enterpise edition versión 6 para el desarrollo de aplicaciones Web," Jun. 2013, Accessed: Jan. 04, 2024. [Online]. Available: http://repositorio.utn.edu.ec/handle/123456789/3719

[20] D. E. Aldas and M. A. Andrade, "Guía práctica para el uso de patrones de diseño en el desarrollo de software," Jan. 2011, Accessed: Jan. 04, 2024. [Online]. Available: http://bibdigital.epn.edu.ec/handle/15000/2669

[21] F. P. Corao and M. P. Vanegas, "Impacto del uso de patrones de diseño en la industria del software en Costa Rica," *Tecnología Vital*, vol. 2, no. 6, Jul. 2019, Accessed: Jan. 13, 2024. [Online]. Available: https://revistas.ulatina.ac.cr/index.php/tecnologiavital/article/view/249/259

[22] F. Khomh and Y.-G. Gueheneuc, *Design patterns impact on software quality: Where are the theories?*, IEEE. 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2018. doi: https://doi.org/10.1109/SANER.2018.8330193.

[23] W. Flageol, É. Menaud, Y. G. Guéhéneuc, M. Badri, and S. Monnier, "A mapping study of language features improving object-oriented design patterns," Aug. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.infsof.2023.107222.

[24] D. Drozdov *et al.*, "Utilizing software design patterns in product-driven manufacturing system: A case study," in *Studies in Computational Intelligence*, vol. 853, Springer Verlag, 2020, pp. 301–312. doi: 10.1007/978-3-030-27477-1_23.

[25] A. Katzmaier and M. Hanneghan, "Design pattern evaluation of mobile and web based application frameworks," in *Proceedings - 2013 6th International Conference on Developments in eSystems Engineering, DeSE 2013*, Institute of Electrical and Electronics Engineers Inc., Feb. 2013, pp. 157–162. doi: 10.1109/DeSE.2013.36.

[26] L. Thung Phek *et al.*, *Improving a Web Application Using Design Patterns: A Case Study*. 2010.

[27] A. Ampatzoglou, G. Frantzeskou, and I. Stamelos, "A methodology to assess the impact of design patterns on software quality," *Inf Softw Technol*, vol. 54, no. 4, pp. 331–346, Apr. 2012, doi: 10.1016/j.infsof.2011.10.006.

[28] R. Serrato-Barrera, G. Rodríguez-Gómez, J. C. Pérez-Sansalvador, S. Pomares-Hernández, L. Flores-Pulido, and A. Muñoz, "Software system design based on patterns for Newton-type methods," *Computing*, vol. 102, no. 4, pp. 1005–1030, Apr. 2020, doi: 10.1007/s00607-019-00759-8.

[29] Y. Elshater, P. Martin, and E. Hassanein, "Using Design Patterns to Improve Web Service Performance," in *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, Institute of Electrical and Electronics Engineers Inc., Aug. 2015, pp. 746–749. doi: 10.1109/SCC.2015.106.

[30] J. J. Valencia, "Incidencia de los patrones de diseño de software en la seguridad de aplicaciones web," *MQRInvestigar*, vol. 8, no. 1, pp. 236–259, Jan. 2024, doi: 10.56048/MQR20225.8.1.2024.236-259.

[31] C. Gonzales, "UNA REVISIÓN DE LOS PATRONES DE DISEÑO DE SOFTWARE APLICADO A LAS APLICACIONES WEB," 2020. Accessed: Jan. 15, 2024. [Online]. Available: https://repositorio.uss.edu.pe/handle/20.500.12802/6783

[32] M. de L. Correa, I. Hoyos Beltrán, and F. Fabio Quintero, "Análisis del impacto del uso de patrones de diseño en la fase de mantenimiento," 2017, [Online]. Available: http://revistas.udistrital.edu.co/ojs/index.php/tia/issue/archive

[33] E. Castañeda, "PROPUESTA DE PATRÓN DE DISEÑO DE SOFTWARE ORIENTADO A PREVENIR LA EXTRACCIÓN AUTOMATIZADA DE CONTENIDO WEB.," *PUPC*, Nov. 2016, Accessed: Jan. 15, 2024. [Online]. Available: http://hdl.handle.net/20.500.12404/7513

[34] S. González, "Aplicación de patrones de diseño para la resolución de problemas de software en el desarrollo de una aplicación móvil iOS," *Universitat Politécnica de Valéncia*, Jul. 2015, Accessed: Jan. 16, 2024. [Online]. Available: http://hdl.handle.net/10251/69088

[35] E. Hernández, R. Rodríguez, and C. Salinas, "Aplicación Web con módulos de captura automática para la administración de sueldos y salarios de una constructora, utilizando patrones de diseño de software," 2015. Accessed: Jan. 15, 2024. [Online]. Available: https://tesis.ipn.mx/handle/123456789/22577?show=full

[36] B. Vmware, "¿Qué es la seguridad de las aplicaciones? | Glosario de VMware | ES," Broadcom. Accessed: Jan. 04, 2024. [Online]. Available: https://www.vmware.com/es/topics/glossary/content/application-security.html

[37] A. Zambrano, T. Guard, E. Vladimir, V. Haro, and G. Ninahualpa, "Mitigation techniques for security vulnerabilities in web applications," Nov. 2018, Accessed: Jan. 04, 2024. [Online]. Available: https://www.researchgate.net/publication/331178479

[38] F. García, "¿Qué es OWASP? ¿Qué es OWASP TOP 10? | Arsys," Arsys. Accessed: Jan. 25, 2024. [Online]. Available: https://www.arsys.es/blog/owasp

[39] CloudFlare, "¿Qué es OWASP? ¿Qué es el OWASP Top 10? | Cloudflare," Cloudflare. Accessed: Jan. 25, 2024. [Online]. Available: https://www.cloudflare.com/es-es/learning/security/threats/owasp-top-10/

[40] c Ciber 4 All Team, "OWASP: Top 10 de vulnerabilidades en aplicaciones web," Tarlogic. Accessed: Jan. 04, 2024. [Online]. Available: https://www.tarlogic.com/es/blog/owasp-top-10-vulnerabilidades-web/

[41] J. Segovia, "How to use OWASP for ISO 27001 A.14 Secure development." Accessed: Mar. 20, 2025. [Online]. Available: https://advisera.com/27001academy/blog/2018/04/24/how-to-use-open-web-application-security-project-owasp-for-iso-27001/

[42] G. Data, "ISO 27034: Application Security - DataGuard." Accessed: Mar. 27, 2025. [Online]. Available: https://www.dataguard.com/blog/iso-27034/

[43] E. Norma, "TECNOLOGÍAS DE LA INFORMACIÓN – TÉCNICAS DE SEGURIDAD – SEGURIDAD DE LA APLICACIÓN – PARTE 1: DESCRIPCIÓN Y CONCEPTOS (ISO/IEC 27034-1:2011 + Cor 1:2014, IDT)," May 2015.

[44] E. Gutierrez, "Identifica Las Principales Vulnerabilidades En Aplicaciones," Codster. Accessed: Jan. 26, 2024. [Online]. Available: https://codster.io/blog/principales-vulnerabilidades-en-aplicaciones/

[45] D. L. Gamboa, "Vulnerabilidades en aplicaciones web utilizando la metodología de 'proyecto abierto de seguridad de aplicaciones web,'" 2021, Accessed: Jan. 26, 2024. [Online]. Available: https://repositorio.pucesa.edu.ec/handle/123456789/3175

[46] B. M. Montero, H. V. Cevallos, and J. D. Cuesta, "Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software," *Espirales Revista Multidisciplinaria de investigación,* vol. 2, no. 17, Jun. 2018, doi: 10.31876/RE.V2I17.269.

[47] S. Loayan, "Metodología Agile: TODO sobre esta forma de trabajo [2025] • Asana." Accessed: Mar. 20, 2025. [Online]. Available: https://asana.com/es/resources/agile-methodology

[48] J. Martins, "Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2023] • Asana," Asana. Accessed: Jan. 15, 2024. [Online]. Available: https://asana.com/es/resources/what-is-scrum

[49] M. Trigás, "Metodología Scrum," *Universitat Oberta de Catalunya,* Jun. 2012, Accessed: Jan. 15, 2024. [Online]. Available: https://openaccess.uoc.edu/handle/10609/17885

[50] K. Schwaber and J. Sutherland, "La Guía Definitiva de Scrum: Las Reglas del Juego." Accessed: Nov. 04, 2023. [Online]. Available: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf

[51] M. Cappola, "Qué es JavaScript, para qué sirve y cómo funciona," HubSpot. Accessed: Jan. 04, 2024. [Online]. Available: https://blog.hubspot.es/website/que-es-javascript

[52] L. Kisnta, "Qué es Node.js y por qué deberías usarlo." Accessed: Jan. 04, 2024. [Online]. Available: https://kinsta.com/es/base-de-conocimiento/que-es-node-js/

[53] M. Kinsta, "¿Qué es Express.js? Todo lo que Debes Saber." Accessed: Mar. 23, 2025. [Online]. Available: https://kinsta.com/es/base-de-conocimiento/que-es-express/

[54] M. Cappola, “Frontend y backend: qué son, en qué se diferencian y ejemplos.” Accessed: Jan. 04, 2024. [Online]. Available: https://blog.hubspot.es/website/frontend-y-backend

[55] A. Sandeep, “REACT.JS AND FRONT END DEVELOPMENT,” *International Research Journal of Engineering and Technology*, 2020, Accessed: Jan. 04, 2024. [Online]. Available: https://www.geeksforgeeks.org/angular-vs-reactjs-

[56] L. Kinta, “¿Qué es PostgreSQL?” Accessed: Jan. 04, 2024. [Online]. Available: https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/

[57] M. IBM, “¿Qué es PostgreSQL? | IBM.” Accessed: Jan. 04, 2024. [Online]. Available: https://www.ibm.com/mx-es/topics/postgresql

[58] A. Sunardi and Suharjito, “MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based,” *Procedia Comput Sci*, vol. 157, pp. 134–141, Jan. 2019, doi: 10.1016/J.PROCS.2019.08.150.

[59] F. Enríquez, S. Fierro, B. Flores, D. Imbaquingo Esparza, and J. Michelena, “Impacto del patrón modelo vista controlador (MVC) en la seguridad, interoperabilidad y usabilidad de un sistema informático durante su ciclo de vida,” *EASI: Ingeniería y Ciencias Aplicadas en la Industria*, vol. 2, no. 1, pp. 11–16, Jul. 2023, doi: 10.53591/EASI.V2I1.2043.

[60] N. Govil and A. Sharma, “Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors,” *Advances in Engineering Software*, vol. 172, p. 103209, Oct. 2022, doi: 10.1016/J.ADVENGSOFT.2022.103209.

[61] R. K. Mallidi and M. Sharma, “Study on Agile Story Point Estimation Techniques and Challenges,” *Int J Comput Appl*, vol. 174, no. 13, pp. 975–8887, 2021.

[62] M. Express, “Express cookie-parser middleware.” Accessed: Mar. 23, 2025. [Online]. Available: https://expressjs.com/en/resources/middleware/cookie-parser.html

[63] L. Nodemailer, “Nodemailer :: Nodemailer.” Accessed: Mar. 23, 2025. [Online]. Available: https://www.nodemailer.com/

[64] P. Huet, “Qué es Tailwind CSS y por qué deberías usarlo | OpenWebinars.” Accessed: Mar. 23, 2025. [Online]. Available: https://openwebinars.net/blog/que-es-tailwind-css-y-por-que-deberias-usarlo/

[65] R. Vite, “Introducción | Vite.” Accessed: Mar. 23, 2025. [Online]. Available: https://es.vite.dev/guide/

[66] F. McClenahan, “How to use Next.js and Recharts to build an information dashboard.” Accessed: Mar. 23, 2025. [Online]. Available: https://ably.com/blog/informational-dashboard-with-nextjs-and-recharts

[67] React, “React-big-calendar - npm.” Accessed: Mar. 23, 2025. [Online]. Available: https://www.npmjs.com/package/react-big-calendar

[68] CodeParrot, “Getting Started with React Toastify: Enhance Your Notifications.” Accessed: Mar. 23, 2025. [Online]. Available: https://codeparrot.ai/blogs/streamline-your-react-notifications-with-react-toastify

[69] G. Mckeever, “What is Content Security Policy (CSP) | Header Examples | Imperva.” Accessed: Apr. 01, 2025. [Online]. Available: https://www.imperva.com/learn/application-security/content-security-policy-csp-header/

[70] I. Foundeo, “Content-Security-Policy (CSP) Header Quick Reference.” Accessed: Apr. 01, 2025. [Online]. Available: https://content-security-policy.com/

[71] H. ServerBlog, “¿Qué es el ‘Content Security Policy’ (CSP) y cómo se utiliza para mejorar la seguridad y el rendimiento del sitio?” Accessed: Mar. 23, 2025. [Online]. Available: https://servidoresseguros.com/que-es-el-content-security-policy-csp-y-como-se-utiliza-para-mejorar-la-seguridad-y-el-rendimiento-del-sitio/

[72] A. Chiarelli, “Clickjacking Attacks and How to Prevent Them.” Accessed: Mar. 23, 2025. [Online]. Available: https://auth0.com/blog/preventing-clickjacking-attacks/

[73] R. Rehim, “X-Frame options header not implemented.” Accessed: Mar. 23, 2025. [Online]. Available: https://beaglesecurity.com/blog/vulnerability/x-frame-options-header-not-implemented.html

[74] G. Hawk, “StackHawk Docs for Plugin Cross-Domain JavaScript Source File Inclusion.” Accessed: Mar. 23, 2025. [Online]. Available: https://docs.stackhawk.com/vulnerabilities/10017/

[75] PorSwigger, “Private IP addresses disclosed - PortSwigger.” Accessed: Mar. 23, 2025. [Online]. Available: https://portswigger.net/kb/issues/00600300_private-ip-addresses-disclosed

[76] C. Indusface, “What are the OWASP Top 10 Risks 2021 | Indusface Blog,” Indusface Blog. Accessed: Jan. 26, 2024. [Online]. Available: https://www.indusface.com/learning/what-are-the-owasp-top-10-risks-2021/

[77] T. Tuleap, “Understanding Agile Scrum,” Tuleap. Accessed: Jan. 26, 2024. [Online]. Available: https://www.tuleap.org/agile/agile-scrum-in-10-minutes

ANEXOS

Anexo 1

Pruebas hechas en Owasp Zap, enlace Drive.