

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**LUDOLAB CONNECTSYS: SISTEMA INFORMÁTICO PARA LA  
GESTIÓN DE EVENTOS DE FORMACIÓN Y SERVICIO A LA  
COMUNIDAD DEL LABORATORIO LUDOLAB**

**COMPONENTE DE SEGURIDAD**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO/A EN  
SOFTWARE**

**GINNO SEBASTIAN TAIMAL VILLARROEL**

ginno.taimal@epn.edu.ec

**DIRECTOR: VICTOR VICENTE VELEPUCHA BONETT**

victor.velepucha@epn.edu.ec

**DQM, JULIO 2024**

## **CERTIFICACIONES**

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

---

**GINNO SEBASTIAN TAIMAL VILLARROEL**

Certifico que el presente trabajo de integración curricular fue desarrollado por GINNO SEBASTIAN TAIMAL VILLARROEL , bajo mi supervisión.

---

**VICTOR VICENTE VELEPUCHA BONETT**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

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

GINNO SEBASTIAN TAIMAL VILLARROEL

VICTOR VICENTE VELEPUCHA BONETT

LUIS GONZALO CHILIGUANO SUAREZ

MIGUEL ANGEL MUENALA QUINGA

## DEDICATORIA

Dedico este trabajo a mis padres, Ángel Taimal y Ruth Villarroel, quienes con su amor incondicional y apoyo constante han sido la base de todo lo que he logrado. Su ejemplo de perseverancia y esfuerzo ha sido una guía en cada paso de mi camino académico y personal. Les agradezco profundamente por su sacrificio y por siempre creer en mí.

A mi hermano, Kevin Taimal, por darme la oportunidad de equivocarme, por ser mi compañero de caídas, peleas y apoyo desde pequeño. Has sido una fuente de inspiración y fortaleza, y nuestros momentos compartidos me han enseñado el valor de la resiliencia y la importancia de tener a alguien en quien confiar.

Me dedico este logro a mí mismo, por la perseverancia y el esfuerzo dedicados a lo largo de este proceso. He superado desafíos y aprendido valiosas lecciones que han moldeado mi carácter y habilidades. Este trabajo es un reflejo de mi compromiso y dedicación para alcanzar mis metas.

Finalmente, dedico este trabajo a todas las personas que caminaron a mi lado, quienes con su apoyo y compañía hicieron posible este logro. A mis amigos, compañeros y mentores, gracias por sus palabras de aliento y por estar presentes en cada etapa de este viaje. Su presencia ha sido fundamental para llegar hasta aquí.

## **AGRADECIMIENTO**

A mi familia, especialmente a mis padres y mi hermano, quienes con su amor y apoyo incondicional han sido el pilar fundamental en mi vida. Su guía y confianza me han motivado a superar cada desafío y a alcanzar mis metas con determinación y esfuerzo.

A mis compañeros de trabajo de integración curricular, Luis Chiliguano y Miguel Muenala. Su colaboración y compromiso fueron esenciales para el éxito de este proyecto. Gracias por su dedicación y trabajo en equipo, que hicieron posible este logro compartido.

A mis mejores amigos de Ibarra, mi ciudad de origen. Diego Chacón, por ser mi amigo durante 19 años, Alexander Paspuel, por siempre cuidarme, Nayely Rosero, por alimentarme en HappyPizza, y Josue Julio, por creer en mí. Su amistad y apoyo han sido un refugio y una fuente constante de ánimo y fortaleza.

A mis mejores amigos de Quito. Leonardo Asitimbaya, por ser el apoyo de mis ideas más ambiciosas, Karla Vivas, por ser mi confidente, Fausto Román, por ser mi apoyo, Santiago León, por su constante respaldo, y Lizbeth Freire, por ser mi gym sister. Su amistad ha enriquecido mi vida y me ha dado la fuerza para seguir adelante.

A mi amiga Madelyn Fernandez, por ser la persona que me ha ayudado a tomar riesgos y a crecer profesionalmente con sus proyectos y trabajos. Su apoyo y guía han sido invaluable en mi desarrollo personal y profesional.

A mi grupo Polisoftware, mis amigos más apreciados, que han sido mi mayor apoyo durante mi carrera. Su camaradería y apoyo me han dado la confianza y el ánimo necesarios para enfrentar cada reto académico.

A mi director MSc. Víctor Velepucha, quien fue nuestro apoyo fundamental para el desarrollo de este trabajo, y al Dr. Marco Santorum, que me acogió en su laboratorio LudoLab junto a la Dra. Mayra Carrión, ayudándome a conceptualizar la idea del proyecto. Su orientación y respaldo fueron clave para la realización de este trabajo de integración curricular.

# Índice general

<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Objetivos . . . . .	3
1.1.1. Objetivo general . . . . .	3
1.1.2. Objetivos específicos . . . . .	3
1.2. Alcance . . . . .	4
1.3. Marco teórico . . . . .	5
1.3.1. Sistema de información estudiantil . . . . .	5
1.3.2. Seguridad en aplicaciones web . . . . .	5
1.3.3. Evaluación de riesgos . . . . .	6
1.3.4. Modelado de amenazas (Threat Modeling) . . . . .	7
1.3.5. Gestión de identidades y accesos . . . . .	7
<b>2. METODOLOGÍA</b>	<b>9</b>
2.1. Metodología . . . . .	9
2.1.1. Metodología Kanban . . . . .	9
2.1.2. Marco de trabajo OWASP SAMM . . . . .	11
2.1.3. Integración de OWASP SAMM con Kanban . . . . .	12
2.2. Herramientas . . . . .	12
2.3. Desarrollo . . . . .	15
2.3.1. Planificación . . . . .	16
2.3.2. Diseño . . . . .	22
2.3.3. Implementación . . . . .	29
<b>3. EVALUACIÓN Y RESULTADOS</b>	<b>58</b>
3.1. Endpoints . . . . .	58
3.2. Pruebas de funcionalidad . . . . .	60

<b>4. CONCLUSIONES Y RECOMENDACIONES</b>	<b>68</b>
4.1. Conclusiones . . . . .	68
4.2. Recomendaciones . . . . .	69
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>71</b>
<b>ANEXOS</b>	<b>75</b>
A. <a href="#">Historias de usuario</a> . . . . .	75
B. <a href="#">Análisis de riesgos</a> . . . . .	75
C. <a href="#">Diseño de base de datos</a> . . . . .	75
D. <a href="#">Arquitectura</a> . . . . .	75
E. <a href="#">Reporte del modelo de amenazas</a> . . . . .	75
F. <a href="#">Prototipo</a> . . . . .	75
G. <a href="#">GitHub</a> . . . . .	75

# Índice de figuras

2.1. Diagrama de base de datos. . . . .	23
2.2. Diagrama de contexto. . . . .	24
2.3. Diagrama de aplicación. . . . .	24
2.4. Diagrama de componentes. . . . .	25
2.5. Modelo de amenazas. . . . .	27
2.6. Pantalla de bienvenida. . . . .	28
2.7. Pantalla de ingreso. . . . .	29
2.8. Pantalla de registro. . . . .	29
2.9. Estructura del frontend. . . . .	30
2.10. Estructura del backend. . . . .	31
2.11. Estructura de la biblioteca de clases,. . . . .	32
2.12. Microservicios. . . . .	33
2.13. Estructura del microservicio AuthService. . . . .	34
2.14. Tablero Kanban en estado Pendiente para el Submódulo de acceso y registro. . . . .	35
2.15. Tablero Kanban en estado En Progreso para el Submódulo de acceso y registro. . . . .	36
2.16. Tablero Kanban en estado En Progreso para el Submódulo de acceso y registro. . . . .	37
2.17. Submódulo de acceso y registro - Pantalla de bienvenida. . . . .	38
2.18. Submódulo de acceso y registro - Pantalla de ingreso. . . . .	38
2.19. Submódulo de acceso y registro - Pantalla de registro. . . . .	39
2.20. Función HandleValidSubmit() para el Login. . . . .	40
2.21. Función GetAuthenticationStateAsync() para token. . . . .	41
2.22. Tablero Kanban en estado Completado para el Submódulo de acceso y registro. . . . .	42
2.23. Tablero Kanban en estado Pruebas para el Submódulo de acceso y registro. . . . .	42
2.24. Tablero Kanban en estado Integración para el Submódulo de acceso y registro. . . . .	43
2.25. Tablero Kanban en estado Finalizado para el Submódulo de acceso y registro. . . . .	44



2.26. Tablero Kanban en estado Pendiente para el Submódulo de aprobación de usuarios. . . . .	45
2.27. Pantalla de aprobación de usuarios. . . . .	46
2.28. Tablero Kanban en estado Finalizado para el Submódulo de aprobación de usuarios. . . . .	47
2.29. Tablero Kanban en estado Pendiente para el Submódulo de autenticación y autorización. . . . .	48
2.30. Políticas de autorización en Backend. . . . .	49
2.31. Atributo de autorización en Frontend. . . . .	49
2.32. Configurar la autorización en Frontend. . . . .	49
2.33. Submodulo autenticación y autorización - Pantalla credenciales incorrectas. .	50
2.34. Submodulo autenticación y autorización - Pantalla no Autorizado. . . . .	50
2.35. Tablero Kanban en estado Finalizado para el Submódulo de autenticación y autorización. . . . .	51
2.36. Tablero Kanban en estado Pendiente para el Submódulo de gestión de contraseñas. . . . .	52
2.37. Pantalla para generar una nueva contraseña. . . . .	53
2.38. Tablero Kanban en estado Pendiente para el Submódulo de auditoría. . . . .	54
2.39. Pantalla de auditoría del sistema. . . . .	55
2.40. Código relevante de AuditService. . . . .	56
2.41. Tablero Kanban en estado Finalizado para el Submódulo de auditoría. . . . .	57
3.1. Prueba POST /auth/login. . . . .	61
3.2. Respuesta del JWT. . . . .	61
3.3. Prueba GET /password/lastlogin/{userId} incluye la fecha y hora del último inicio de sesión. . . . .	62
3.4. Prueba GET /password/lastlogin/{userId} cuando no tiene fecha y hora del último inicio de sesión. . . . .	62
3.5. Prueba POST /password/changepassword exitosa. . . . .	63
3.6. Prueba POST /password/changepassword no exitosa. . . . .	63
3.7. Prueba POST /account/approve. . . . .	64
3.8. Prueba POST /account/deny. . . . .	64
3.9. Prueba GET /audit/. . . . .	65
3.10. Prueba GET /audit/{id}. . . . .	66

3.11.Prueba POST /register/ . . . . .	66
---------------------------------------	----

# Índice de Tablas

2.1. Historias de usuario. . . . .	18
2.2. Requisitos no funcionales . . . . .	19
2.3. Análisis de riesgos . . . . .	20
3.1. Endpoints . . . . .	59

## RESUMEN

El presente trabajo de integración curricular se centra en el desarrollo y aseguramiento de la aplicación LudoLab ConnectSys, un sistema de administración estudiantil destinado a gestionar proyectos educativos en el laboratorio LudoLab de la Facultad de Sistemas. La aplicación automatiza procesos manuales, mejorando la eficiencia y reduciendo la carga administrativa. El enfoque principal de este trabajo es la seguridad, abarcando submódulos como acceso y registro, aprobación de usuarios, autenticación y autorización, gestión de contraseñas y auditoría.

Se utilizó la metodología Kanban, complementada con el marco de trabajo Open Web Application Security Project Software Assurance Maturity Model (OWASP SAMM), para integrar prácticas de seguridad en todo el ciclo de desarrollo del software. En la fase de planificación, se documentaron los requisitos funcionales y no funcionales, se realizó un análisis de riesgos y se definieron especificaciones de seguridad. En la fase de diseño, se elaboraron diagramas de la base de datos y la arquitectura del sistema, detallando tanto el frontend desarrollado en Blazor .NET 8, como el backend basado en microservicios con Web API .NET 8 y una base de datos SQL Server 2022.

Durante la implementación, se desarrollaron y probaron los distintos submódulos, asegurando que cada componente cumpliera con los requisitos de seguridad. Los resultados demuestran que la integración de Kanban y OWASP SAMM es efectiva para gestionar y mejorar la seguridad en el desarrollo de software, garantizando así la protección de datos sensibles y la integridad del sistema.

**PALABRAS CLAVE** - Kanban, OWASP SAMM, seguridad en aplicaciones web, microservicios, sistema de administración estudiantil

## ABSTRACT

The present curricular integration work focuses on the development and assurance of the LudoLab ConnectSys application, a student management system intended to manage educational projects in the LudoLab laboratory of the Faculty of Systems. The application automates manual processes, improving efficiency and reducing administrative workload. The main focus of this work is security, covering submodules such as access and registration, user approval, authentication and authorization, password management, and auditing.

The Kanban methodology, complemented by the Open Web Application Security Project Software Assurance Maturity Model (OWASP SAMM) framework, was used to integrate security practices throughout the software development lifecycle. In the planning phase, functional and non-functional requirements were documented, a risk analysis was performed, and security specifications were defined. In the design phase, database and system architecture diagrams were developed, detailing both the frontend developed in Blazor .NET 8 and the backend based on microservices with Web API .NET 8 and a SQL Server 2022 database.

During implementation, the various submodules were developed and tested, ensuring that each component met the security requirements. The results demonstrate that the integration of Kanban and OWASP SAMM is effective in managing and improving security in software development, thus ensuring the protection of sensitive data and the integrity of the system.

**KEYWORDS** - Kanban, OWASP SAMM, web application security, microservices, student administration system

# Capítulo 1

## INTRODUCCIÓN

A lo largo de estos años la Escuela Politécnica Nacional (EPN), ha ido integrando significativamente las Tecnologías de la Información y la Comunicación (TIC), en sus procesos académicos, administrativos y de investigación. Estas herramientas han ayudado al desarrollo de la EPN y a las funciones esenciales de la institución, promoviendo la creación de unidades y departamentos dedicados a impulsar distintos dominios científicos y tecnológicos.

Dentro de estas unidades y departamentos, entra la creación del Laboratorio de Investigación de Sistemas de Información e Inclusión Digital LudoLab fundado en el año 2019 por la Escuela Politécnica Nacional. Este laboratorio ha sido de suma importancia debido a que juega un papel fundamental en la disminución de la brecha digital por medio del proyecto “Inclusión Digital”, el cual se distingue porque estudiantes voluntarios de la EPN actúan como instructores, brindando clases personalizadas a todo tipo de personas. [1]

Podemos destacar que el objetivo de este proyecto es brindar habilidades digitales fundamentalmente a grupos subrepresentados, como maestros de escuelas rurales y adultos mayores con el fin de mejorar su acceso a las oportunidades disponibles en el mundo digital actual. [1]

Para el laboratorio LudoLab, la administración del proyecto ha representado un área compleja y con un desafío constante debido a la gran cantidad de gente que se inscribe al mismo año tras año, lo que ha generado una gran cantidad de datos, lo que incluyen las inscripciones, creación de cursos, calificaciones, encuestas de satisfacción, entre otros. Estos datos no han sido trabajados de la mejor manera sino han sido manejados con herramientas no especializadas o de manera manual, lo cual dificulta la extracción de información relevante para mejorar los programas académicos.

Actualmente, la gestión de los procesos administrativos y de operación del proyecto de Inclusión Digital se realiza de forma manual o con herramientas no especializadas, lo cual requiere la participación de colaboradores y es propenso a errores. Este proceso no solo es ineficiente, sino que también consume una gran cantidad de tiempo y recursos, lo que limita la capacidad del laboratorio para gestionar eficazmente sus operaciones y brindar un servicio de alta calidad a sus estudiantes y beneficiarios.

Para abordar estos problemas, se ha propuesto el desarrollo de LudoLab ConnectSys, un sistema de administración estudiantil que automatiza diversas tareas administrativas. Este sistema está diseñado para facilitar la gestión de inscripciones, la organización de cursos, la asignación de instructores y estudiantes, la generación de certificados y manejar las evidencias de las clases impartidas. Al automatizar estos procesos, se espera mejorar la eficiencia operativa, reducir la carga de trabajo manual y minimizar el riesgo de errores.

En el contexto del desarrollo de LudoLab ConnectSys, el sistema se compone de tres componentes principales: Administrativo, Automatización de Procesos y Seguridad. El presente componente se enfoca específicamente en la seguridad de la aplicación. La seguridad es un aspecto crítico en cualquier sistema de gestión de información, ya que implica la protección de datos sensibles y la garantía de la integridad y confidencialidad de la información. Según Niño Benitez y Silega Martínez, gestionar la seguridad informática desde el inicio del desarrollo de software es esencial para evitar vulnerabilidades y minimizar los riesgos asociados al acceso no autorizado y malintencionado. [2] Asimismo, la importancia de la seguridad en aplicaciones de administración estudiantil no puede subestimarse, ya que estos sistemas manejan datos personales y académicos de los estudiantes, lo que los convierte en objetivos atractivos para ataques.

Para empezar, se contempló el desarrollo de pantallas de bienvenida, login y registro que constituye el Submódulo de Acceso y Registro. Estas pantallas son el primer punto de interacción del usuario con el sistema, por lo que es crucial garantizar que sean seguras y amigables.

Por otro lado, la gestión de la aprobación o rechazo de nuevos usuarios se maneja en el Submódulo de Aprobación de Usuarios. Este proceso incluye el envío de un correo electrónico con un número de cédula y una contraseña temporal en caso de aprobación, o un correo de rechazo con las razones de la decisión. Esta parte permite controlar el acceso, asegurando que solo personas que realmente van a utilizar el aplicativo sean aprobadas, evitando así la acumulación de usuarios innecesarios en la base de datos y garantizando una gestión más eficiente y segura de los recursos del sistema.

En cuanto al control de acceso, se ha implementado el Submódulo de Autenticación y Autorización para mantener sesiones seguras y controlar el acceso a funcionalidades específicas. Utilizando tokens JWT (JSON Web Tokens) y un sistema de autorización basado en roles, este submódulo garantiza que solo los usuarios con los permisos adecuados puedan acceder tanto al frontend como al backend, protegiendo así los datos sensibles y las operaciones del sistema.

De igual forma, el Submódulo de Gestión de Contraseñas requiere que los usuarios creen una nueva contraseña en su primer inicio de sesión. Este proceso incluye recomendaciones para la creación de contraseñas seguras, como el uso de caracteres especiales, números y letras, mayúsculas y minúsculas. Además, incluye una interfaz para la recuperación de contraseñas en caso de olvido, siguiendo las mismas políticas de seguridad. Esta medida asegura que las cuentas de los usuarios estén protegidas desde el primer uso y facilita la gestión de contraseñas de manera segura.

Para concluir, todas las actividades importantes dentro del sistema se registran en el Submódulo de Auditoría de Seguridad. Este servicio permite un monitoreo continuo y facilita la detección y respuesta a incidentes de seguridad. Los registros de auditoría incluyen eventos como intentos de inicio de sesión, cambios en la configuración del sistema y accesos a recursos, asegurando así la integridad y confidencialidad de los datos.

## **1.1. Objetivos**

### **1.1.1. Objetivo general**

Desarrollar e implementar un componente de seguridad para la aplicación LudoLab ConnectSys, asegurando la protección de los datos y la integridad del sistema mediante el uso de metodologías ágiles y buenas prácticas de seguridad.

### **1.1.2. Objetivos específicos**

1. Aplicar una metodología ágil para el desarrollo junto al marco de trabajo de OWASP SAMM, integrando prácticas de seguridad a lo largo del ciclo de vida del desarrollo del software.
2. Implementar un submódulo de autenticación y autorización robusto que garantice que solo los usuarios autorizados puedan acceder a los recursos y funcionalidades de la



aplicación.

3. Implementar un submódulo de auditoría de actividades que registre todas las operaciones críticas del sistema, permitiendo un seguimiento detallado y la detección de actividades sospechosas.
4. Implementar un submódulo de políticas de contraseñas que aseguren la creación de contraseñas seguras y la protección de las cuentas de usuario.
5. Implementar un submódulo de aprobación de usuarios que verifique y valide las solicitudes de registro, asegurando que solo los usuarios autorizados tengan acceso al sistema.
6. Ejecutar pruebas a cada funcionalidad de la aplicación, cumpliendo con las normas de seguridad, para garantizar que todas las funcionalidades operen de manera segura y eficiente.

## **1.2. Alcance**

El alcance abarca el desarrollo integral de la aplicación LudoLab ConnectSys, centrándose específicamente en la implementación y aseguramiento del componente de seguridad. Este componente incluye submódulos clave como el acceso y registro, la aprobación de usuarios, la autenticación y autorización, la gestión de contraseñas y la auditoría. Utilizando la metodología ágil Kanban, se gestionaron todas las etapas del desarrollo, desde la planificación y diseño hasta la implementación y pruebas. Asimismo, se integró el marco de trabajo OWASP SAMM para asegurar que las mejores prácticas de seguridad se aplicaran a lo largo de todo el ciclo de vida del desarrollo del software. El componente también contempló la ejecución de pruebas funcionales para garantizar la robustez y la seguridad del sistema, logrando un sistema que no solo automatiza procesos administrativos, sino que también protege de manera eficaz los datos sensibles y asegura la integridad del sistema frente a posibles amenazas.

## **1.3. Marco teórico**

### **1.3.1. Sistema de información estudiantil**

Un Sistema de Información Estudiantil o Student Information System (SIS) es una plataforma tecnológica diseñada para gestionar toda la información relacionada con los estudiantes en instituciones educativas. Estos sistemas integran diversas funciones administrativas y académicas, permitiendo un manejo eficiente de datos como inscripciones, calificaciones, horarios de clases, asistencia y más. Los SIS son fundamentales para mejorar la gestión administrativa y académica, ofreciendo una visión integral y centralizada de la información estudiantil [3]. Además, estos sistemas facilitan la toma de decisiones al proporcionar datos precisos y actualizados a administradores, profesores y estudiantes.

Los sistemas de información estudiantil no solo agilizan los procesos administrativos, sino que también mejoran la experiencia educativa. Al centralizar la información y automatizar tareas repetitivas, los SIS permiten a los profesores y administradores enfocarse en aspectos más importantes de la educación, como la calidad del aprendizaje y la atención personalizada a los estudiantes [4]. La integración de funcionalidades como la gestión de inscripciones, el seguimiento del progreso académico y la comunicación entre estudiantes y profesores convierte a los SIS en herramientas indispensables para las instituciones educativas modernas.

En el contexto de la educación superior, los sistemas de información estudiantil son aún más críticos. Estas plataformas soportan una gran cantidad de datos y transacciones, asegurando que la información esté siempre disponible y segura. Los SIS también facilitan la conformidad con regulaciones educativas y estándares de calidad, lo que es crucial para la acreditación y el financiamiento institucional [5]. Además, la capacidad de generar informes y análisis avanzados permite a las instituciones identificar tendencias, prever necesidades futuras y mejorar continuamente sus programas educativos.

### **1.3.2. Seguridad en aplicaciones web**

La seguridad en aplicaciones web es un aspecto crítico del desarrollo de software, ya que estas aplicaciones son continuamente objetivo de ataques cibernéticos que buscan explotar vulnerabilidades para acceder a datos sensibles o interrumpir servicios. La seguridad en aplicaciones web implica la implementación de diversas medidas y prácticas que pro-

tegen contra amenazas como el cross-site scripting (XSS), inyección de SQL, y cross-site request forgery (CSRF), entre otras. OWASP (Open Web Application Security Project) es una organización que ofrece una guía exhaustiva sobre las mejores prácticas para asegurar aplicaciones web, promoviendo la creación de software seguro y la educación sobre los riesgos y métodos de mitigación [6].

El enfoque en la seguridad de aplicaciones web debe comenzar desde las primeras etapas del desarrollo e incluir pruebas continuas a lo largo de todo el ciclo de vida del software. Entre las prácticas recomendadas se encuentran el uso de conexiones seguras a través de HTTPS (Hypertext Transfer Protocol Secure), la validación y sanitización de entradas de usuario, y la implementación de autenticación y autorización robustas. Además, es crucial mantenerse actualizado con las últimas vulnerabilidades y parches de seguridad, utilizando herramientas y servicios especializados para realizar auditorías de seguridad y pruebas de penetración [7]. Estos enfoques ayudan a identificar y corregir vulnerabilidades antes de que puedan ser explotadas.

Para las pequeñas y medianas empresas (pymes) y startups, mejorar la seguridad de las aplicaciones web puede ser un desafío debido a recursos limitados. Sin embargo, la adopción de prácticas de seguridad recomendadas por OWASP y la utilización de servicios de seguridad gestionados puede significativamente mejorar su postura de seguridad. Invertir en la formación de desarrolladores sobre las mejores prácticas de seguridad y realizar revisiones regulares de código también contribuye a la creación de aplicaciones más seguras. Las empresas deben considerar la seguridad como una inversión esencial para proteger tanto sus activos como la confianza de sus usuarios [8].

### **1.3.3. Evaluación de riesgos**

La evaluación de riesgos es esencial para detectar, analizar y mitigar posibles amenazas que puedan poner en peligro la seguridad de la información en una organización. Este proceso comienza con la identificación de los riesgos específicos, considerando tanto amenazas internas, como el acceso no autorizado por parte de empleados, como externas, como los ataques cibernéticos por parte de hackers. La identificación precisa de estos riesgos es fundamental para una gestión de seguridad efectiva, permitiendo a las organizaciones tomar decisiones informadas sobre cómo proteger sus activos más importantes [9].

Posteriormente, se evalúan los riesgos para determinar su probabilidad de ocurrencia

y el impacto potencial en la organización. Este análisis se realiza mediante metodologías de evaluación de riesgos que combinan estos dos factores para priorizar las amenazas. La evaluación puede incluir la clasificación de riesgos en categorías como alto, medio y bajo, lo que permite a la organización dirigir sus recursos hacia la mitigación de los riesgos más críticos [10]. Una evaluación de riesgos bien realizada proporciona una visión clara de las áreas que requieren mayor atención y recursos para fortalecer la seguridad de la información.

#### **1.3.4. Modelado de amenazas (Threat Modeling)**

El modelado de amenazas es un proceso esencial para identificar y mitigar riesgos de seguridad durante el desarrollo de software. Este enfoque permite a los equipos anticipar posibles vulnerabilidades y tomar medidas preventivas antes del despliegue. Herrera y Quispe explican que el modelado de amenazas involucra la identificación de atacantes potenciales y sus métodos, así como la evaluación de los impactos que estas amenazas podrían tener [11]. Se utilizan herramientas como diagramas de flujo de datos y metodologías como STRIDE para categorizar las amenazas.

Además, el modelado de amenazas incluye la evaluación de riesgos, priorizando las vulnerabilidades según su probabilidad e impacto [12]. Este enfoque facilita la implementación de controles de seguridad efectivos para mitigar los riesgos más críticos, como el cifrado de datos y la autenticación multifactor. Al mismo tiempo, promueve una cultura de seguridad dentro del equipo de desarrollo, asegurando que todos los miembros estén al tanto de las mejores prácticas y los riesgos potenciales.

La colaboración entre los equipos de desarrollo, seguridad y operaciones es crucial para la implementación exitosa del modelado de amenazas. Esta cooperación asegura que las medidas de seguridad se integren adecuadamente en todas las fases del ciclo de vida del software [11]. Herramientas como Microsoft Threat Modeling Tool ayudan a estructurar el proceso, facilitando la identificación y evaluación de amenazas de manera eficiente.

#### **1.3.5. Gestión de identidades y accesos**

La gestión de identidades y accesos, o Identity and Access Management (IAM), es un marco de políticas y tecnologías que aseguran que las personas adecuadas tengan el acceso apropiado a los recursos tecnológicos dentro de una organización. IAM abarca la administración de identidades digitales y la definición y aplicación de políticas de acceso. Este

sistema incluye procesos para la creación, mantenimiento y eliminación de identidades de usuarios, así como la asignación de roles y permisos. IAM también implementa mecanismos de autenticación y autorización para garantizar que solo los usuarios autorizados puedan acceder a recursos específicos. Según Ghosh, la gestión de identidades y accesos es vital para proteger los datos y sistemas empresariales al minimizar los riesgos asociados con el acceso no autorizado [13]. Además, OSTEC-Push destaca que IAM es crucial para cumplir con normativas y estándares de seguridad, asegurando que las organizaciones mantengan un control riguroso sobre quién puede acceder a qué recursos [14].

## Capítulo 2

# METODOLOGÍA

En el desarrollo de esta aplicación web, se utilizarán dos enfoques metodológicos complementarios: Kanban y OWASP SAMM. Kanban se centrará en la gestión eficiente del flujo de trabajo y la mejora continua, mientras que OWASP SAMM proporcionará un marco estructurado para implementar y evaluar prácticas de seguridad a lo largo del ciclo de vida del desarrollo del software. La combinación de estas metodologías garantizará un desarrollo ágil y seguro, permitiendo abordar eficazmente los desafíos de seguridad y asegurar la calidad del producto final.

### 2.1. Metodología

#### 2.1.1. Metodología Kanban

Kanban ha emergido como una herramienta esencial en la gestión de proyectos de desarrollo de software debido a su simplicidad y efectividad. Originada en el sistema de producción de Toyota, Kanban se ha adaptado para su uso en el desarrollo de software, permitiendo a los equipos visualizar y gestionar su trabajo de manera más eficiente [15]. Su enfoque principal se basa en varios principios fundamentales: visualizar el trabajo, limitar el trabajo en progreso, gestionar el flujo de trabajo, implementar ciclos de retroalimentación y fomentar la mejora colaborativa. Estos principios facilitan una gestión más transparente y eficaz del trabajo [16].

El primer principio, la visualización del trabajo, se realiza mediante un tablero Kanban. Este tablero está dividido en columnas que representan las diferentes etapas del proceso de desarrollo, tales como “Pendiente”, “En Progreso” y “Completado”. Las tareas se representan mediante tarjetas que se mueven a través de estas columnas a medida que avanzan

en el proceso. Esta visualización permite a los equipos identificar rápidamente el estado del trabajo y cualquier cuello de botella que pueda surgir [17].

Otro principio crucial es la limitación del trabajo en progreso. Este enfoque asegura que solo una cantidad limitada de tareas puede estar en progreso al mismo tiempo, lo que ayuda a prevenir la sobrecarga de trabajo, reduce el tiempo de ciclo y mejora la calidad del trabajo [18]. Al limitar el trabajo en progreso, los equipos se enfocan en completar las tareas antes de comenzar nuevas, lo que incrementa la eficiencia y asegura una entrega continua de valor.

La gestión del flujo se centra en la capacidad de un equipo para mover tareas a través del sistema de manera constante y rápida [16]. Medir el flujo ayuda a identificar áreas donde el proceso puede ser mejorado. Las métricas comunes incluyen el tiempo de ciclo (el tiempo que una tarea toma desde el inicio hasta su finalización) y el tiempo de entrega (el tiempo desde que una tarea es solicitada hasta que se completa). Estas métricas proporcionan información valiosa para identificar y eliminar ineficiencias en el proceso.

Finalmente, los ciclos de retroalimentación son esenciales para la mejora continua en Kanban. Estos ciclos se implementan a través de reuniones regulares, como las reuniones diarias de pie, revisiones de retrospectiva y revisiones de entrega. Estas reuniones permiten a los equipos reflexionar sobre su desempeño, identificar áreas de mejora y planificar acciones para abordar esos aspectos [15]. La retroalimentación continua fomenta un ambiente de aprendizaje y adaptación constante.

## **Beneficios de Kanban en el Desarrollo de Software**

La implementación de Kanban en el desarrollo de software ofrece varios beneficios clave. Entre ellos se encuentran una mayor transparencia, una mejor gestión del flujo de trabajo, una reducción en el tiempo de entrega y una mejora en la calidad del producto final [17]. Al limitar el trabajo en progreso, los equipos pueden enfocarse en completar las tareas antes de iniciar nuevas, lo que disminuye el tiempo de espera y acelera la entrega de valor.

Otro beneficio significativo es la capacidad de adaptarse rápidamente a los cambios en los requisitos y prioridades del proyecto [16]. En un entorno de desarrollo de software dinámico, donde las necesidades del cliente pueden cambiar con frecuencia, Kanban proporciona la flexibilidad necesaria para ajustar el enfoque sin interrumpir el flujo de trabajo. Esta adaptabilidad es crucial para mantener la satisfacción del cliente y cumplir con las expectativas en términos de entrega y calidad del producto [18].

## **2.1.2. Marco de trabajo OWASP SAMM**

El Modelo de Madurez de Aseguramiento de Software de OWASP (Open Web Application Security Project Software Assurance Maturity Model, OWASP SAMM) es un marco diseñado para ayudar a las organizaciones a formular una estrategia de seguridad para el desarrollo de software. Este modelo ofrece un conjunto de herramientas, guías y metodologías para evaluar y mejorar la postura de seguridad de una organización. La principal ventaja de OWASP SAMM es su flexibilidad, permitiendo adaptarse a las necesidades específicas de diferentes organizaciones [19].

OWASP SAMM está estructurado en torno a cinco funciones principales: gobernanza, diseño, implementación, verificación y operaciones. Cada una de estas funciones incluye varias prácticas de seguridad que deben ser implementadas para asegurar que la seguridad se integre a lo largo del ciclo de vida del desarrollo de software. Por ejemplo, en la fase de gobernanza, se enfatiza la importancia de la estrategia y las métricas, la política y el cumplimiento, y la educación y guía. Estas prácticas aseguran que la seguridad sea una prioridad desde el nivel más alto de la organización [20].

En la fase de diseño, OWASP SAMM recomienda prácticas como la evaluación de amenazas, los requisitos de seguridad y la arquitectura segura. Estas prácticas son esenciales para identificar y mitigar posibles riesgos antes de que se conviertan en problemas en las etapas posteriores del desarrollo. La evaluación de amenazas, por ejemplo, implica identificar y analizar las posibles amenazas que podrían afectar al software, permitiendo a los equipos de desarrollo diseñar soluciones que prevengan estas amenazas desde el inicio [21].

La fase de implementación se centra en la codificación segura y las pruebas de seguridad. OWASP SAMM sugiere que se realicen revisiones de código y pruebas de penetración para asegurar que el código sea resistente a ataques. Estas prácticas no solo ayudan a identificar y corregir vulnerabilidades en el código, sino que también fomentan una cultura de seguridad entre los desarrolladores. La revisión de código, en particular, es una práctica que permite detectar errores y vulnerabilidades que pueden no ser evidentes durante el desarrollo [22].

En la fase de verificación, OWASP SAMM aboga por la revisión de diseño, la revisión de código y las pruebas de seguridad. Estas actividades aseguran que todas las fases del desarrollo del software han considerado la seguridad y que cualquier problema potencial se haya identificado y resuelto antes de la implementación. Las pruebas de seguridad incluyen



tanto pruebas automatizadas como manuales, lo que permite una evaluación exhaustiva de la seguridad del software [23].

Finalmente, en la fase de operaciones, OWASP SAMM enfatiza la gestión de vulnerabilidades, el endurecimiento del entorno y la habilitación de operaciones. Estas prácticas aseguran que el software se mantenga seguro una vez que se ha desplegado. La gestión de vulnerabilidades, por ejemplo, implica monitorear continuamente el software en busca de nuevas amenazas y vulnerabilidades y responder de manera efectiva para mitigar cualquier riesgo identificado [24].

### **2.1.3. Integración de OWASP SAMM con Kanban**

La integración de OWASP SAMM con Kanban permite a los equipos de desarrollo incorporar prácticas de seguridad directamente en su flujo de trabajo ágil. Al utilizar Kanban para gestionar las tareas y OWASP SAMM para guiar las actividades de seguridad, los equipos pueden asegurarse de que las consideraciones de seguridad se aborden en cada etapa del ciclo de vida del desarrollo de software. Esta combinación facilita la identificación temprana de riesgos y la implementación oportuna de controles de seguridad, mejorando así la calidad y la seguridad del producto final [19].

Las cuatro fases del proyecto serán planificación, diseño, implementación y pruebas. Durante estas fases, Kanban permitirá organizar y priorizar las tareas, gestionando el flujo de trabajo de manera efectiva [15]. OWASP SAMM asegurará que se mantengan altos estándares de seguridad en cada etapa del desarrollo, aportando prácticas esenciales como la evaluación de amenazas, la codificación segura y las revisiones de código [21]. La integración de Kanban y OWASP SAMM será clave para lograr un equilibrio entre eficiencia operativa y seguridad, elementos críticos en el desarrollo de software moderno.

## **2.2. Herramientas**

### **Microsoft .NET Framework**

Este es un entorno de desarrollo que proporciona una plataforma para ejecutar aplicaciones en Windows. Ofrece una extensa biblioteca de clases que abarca desde interfaces de usuario hasta criptografía, incluyendo también conectividad a bases de datos, algoritmos numéricos y comunicaciones en red. Su ventaja principal radica en la capacidad de desarrollar aplicaciones sólidas y de alto rendimiento, facilitando tanto la gestión de la memoria

como la seguridad. Además, .NET Framework es compatible con múltiples lenguajes de programación, permitiendo a los desarrolladores trabajar con el lenguaje que prefieran [25].

### **ASP.NET Core Web API**

Es una herramienta que simplifica la creación de servicios HTTP que pueden ser utilizados por una amplia variedad de clientes, incluyendo navegadores y dispositivos móviles. Permite a los desarrolladores construir aplicaciones RESTful, destacándose por su alto rendimiento y flexibilidad. ASP.NET Core Web API soporta la inyección de dependencias, el middleware y la configuración basada en entornos, lo que facilita un desarrollo modular y adaptable a diversas situaciones [26].

### **Microsoft Blazor**

Este framework de Microsoft está diseñado para construir aplicaciones web interactivas del lado del cliente utilizando C# en lugar de JavaScript. Facilita a los desarrolladores la posibilidad de compartir código y bibliotecas entre el cliente y el servidor, aumentando la productividad y reduciendo la complejidad. Blazor utiliza WebAssembly para ejecutar código C# directamente en el navegador, ofreciendo una experiencia de usuario rápida y eficiente [27].

### **JetBrains Rider**

Es un entorno de desarrollo integrado (IDE) para .NET que combina la inteligencia de ReSharper con las funcionalidades de la plataforma IntelliJ. Rider soporta .NET Framework, .NET Core y Mono, ofreciendo herramientas avanzadas de depuración, refactorización y navegación de código. Su interfaz amigable y su capacidad para integrarse con diversas herramientas de control de versiones lo hacen ideal para equipos de desarrollo que buscan mejorar su eficiencia y calidad de código [28].

### **Microsoft SQL Server**

Es un sistema de gestión de bases de datos relacional desarrollado por Microsoft, utilizado para almacenar y recuperar datos según las necesidades de otras aplicaciones de software, ya sea en la misma computadora o en otra a través de una red. SQL Server 2022 ofrece características avanzadas de seguridad, rendimiento y disponibilidad, siendo una

opción popular para aplicaciones empresariales críticas que requieren alta confiabilidad y escalabilidad [29].

## **ClickUp**

Esta plataforma de gestión de proyectos y productividad proporciona herramientas para la planificación, seguimiento y colaboración en tareas. Ofrece una amplia variedad de vistas como listas, tableros Kanban, calendarios y diagramas de Gantt, permitiendo a los equipos organizar su trabajo de manera que mejor se adapte a sus necesidades. ClickUp integra características como la asignación de tareas, seguimiento del tiempo y comunicación en tiempo real, mejorando la coordinación y eficiencia del equipo [30].

## **Figma**

Es una herramienta de diseño colaborativo en línea que permite a los equipos de diseño trabajar juntos en tiempo real. Facilita la creación de interfaces de usuario, prototipos interactivos y la colaboración en tiempo real. Figma permite la integración con otros flujos de trabajo de diseño y desarrollo, permitiendo a los equipos iterar rápidamente y asegurando que el diseño y la funcionalidad estén alineados desde el principio [31].

## **Microsoft Teams**

Es una plataforma de comunicación y colaboración que integra chat, videoconferencias, almacenamiento de archivos y aplicaciones en el flujo de trabajo del equipo. Se utiliza para coordinar el trabajo en equipo y facilitar la comunicación en tiempo real, tanto en reuniones programadas como en discusiones espontáneas. Teams también se integra con otras aplicaciones de Microsoft 365, lo que permite un flujo de trabajo más cohesivo y eficiente [32].

## **Microsoft Threat Modeling Tool**

Es una herramienta que ayuda a identificar y mitigar posibles amenazas en el diseño de software. Permite a los desarrolladores y arquitectos de software crear modelos de amenaza y generar informes automáticamente que destacan las vulnerabilidades y recomiendan contramedidas. Esta herramienta es crucial en la fase de diseño de la seguridad en el desa-

rollo de software, ayudando a asegurar que los sistemas sean diseñados con una postura de seguridad robusta desde el principio [33].

## **GitHub**

Es una plataforma de alojamiento de código que permite a los desarrolladores colaborar en proyectos de software de manera eficiente. Proporciona herramientas para el control de versiones, la revisión de código, y la gestión de proyectos, facilitando la coordinación entre equipos de desarrollo y la integración continua. GitHub también permite a los desarrolladores compartir su trabajo con la comunidad global, lo que promueve la colaboración abierta y el desarrollo de software de alta calidad. [34]

## **Git**

Es un sistema de control de versiones distribuido que permite a los desarrolladores rastrear los cambios en el código fuente durante el desarrollo de software. Es utilizado para coordinar el trabajo entre múltiples programadores, administrar versiones de código y mantener un historial de cambios detallado. Git facilita la fusión de diferentes ramas de desarrollo y resuelve conflictos de manera eficiente, lo que lo convierte en una herramienta esencial para el desarrollo colaborativo de software. [35]

### **2.3. Desarrollo**

En esta sección, se detalla el desarrollo del sistema LudoLab ConnectSys con un enfoque específico en el componente de seguridad de la aplicación. Cabe recalcar, que la seguridad es una pieza fundamental en la creación de cualquier sistema de gestión de información, especialmente cuando se manejan datos sensibles de múltiples usuarios. Por ello, el componente de seguridad está diseñado para proteger la integridad, confidencialidad y disponibilidad de los datos, garantizando que solo usuarios autorizados puedan acceder a las funcionalidades y datos necesarios.

Antes de entrar a las fases del desarrollo vamos a explicar brevemente sobre el sistema, LudoLab ConnectSys es una aplicación web diseñada para gestionar los proyectos y cursos de capacitación que se desarrollan en el laboratorio LudoLab de la Facultad de Sistemas. Uno de los proyectos más conocidos de este laboratorio es el proyecto de “Inclusión Digital”, el cual ha alcanzado su 20ava edición, a la fecha del documento. Este proyecto se ha venido

desarrollando tanto de manera virtual como presencial, pero siempre de forma manual. Por este motivo, esta aplicación web tiene como objetivo automatizar la gestión administrativa de estos proyectos, facilitando la creación y administración de los cursos de capacitación impartidos en LudoLab. Para ello, el sistema se dividió en tres componentes principales: Administrativo, Automatización de procesos y Seguridad. En las siguientes secciones, se detallará el desarrollo y las consideraciones de seguridad que se tomaron en cuenta.

### 2.3.1. Planificación

La seguridad es un aspecto crucial en el desarrollo de cualquier sistema de software y debe integrarse desde la etapa de planificación. Esta integración temprana permite identificar y mitigar riesgos antes de que se conviertan en problemas mayores. En esta sección, se detallan los submódulos desarrollados, los requisitos funcionales y no funcionales, el análisis de riesgos y las especificaciones de seguridad implementadas para mitigar dichos riesgos.

#### Submódulos

A continuación, se detalla cada uno de los submódulos desarrollados para el sistema LudoLab ConnectSys, explicando sus objetivos y funcionalidades específicas.

1. **Submódulo de acceso y registro:** En primer lugar, este submódulo es fundamental para gestionar las primeras interacciones de los usuarios con el sistema. Incluye la implementación de pantallas de bienvenida, login y registro, asegurando que los usuarios puedan autenticarse y registrarse de manera segura, con validaciones adecuadas de los datos ingresados.
2. **Submódulo de aprobación de usuarios:** Por otro lado, este submódulo garantiza que solo personas verificadas puedan acceder al sistema. Permite a los administradores revisar y aprobar o rechazar nuevas solicitudes de usuarios, evitando la proliferación de usuarios no autorizados y manteniendo la base de datos libre de registros innecesarios.
3. **Submódulo de autenticación y autorización:** Asimismo, este submódulo se encarga de gestionar el acceso a las funcionalidades del sistema basado en los roles de los usuarios. Utilizando el Control de Acceso Basado en Roles (RBAC), asegura que

cada usuario solo pueda acceder a las áreas del sistema que le corresponden según su rol. Además, incluye límites de intentos de inicio de sesión para prevenir ataques de fuerza bruta, bloqueando la IP del usuario después de múltiples intentos fallidos.

4. **Submódulo de gestión de contraseñas:** Del mismo modo, este submódulo se encarga de la creación y recuperación de contraseñas, siguiendo políticas de seguridad estrictas. Permite a los usuarios crear una nueva contraseña en su primer inicio de sesión, así como recuperar su contraseña en caso de olvido. Las contraseñas se hashean utilizando un algoritmo seguro antes de ser almacenadas en la base de datos, aplicando políticas que requieren el uso de caracteres especiales, longitud mínima y longitud máxima.
5. **Submódulo de auditoría:** Finalmente, se enfoca en detectar y registrar todos los cambios significativos en la base de datos, incluyendo operaciones de creación, actualización y eliminación. Además, registra el usuario que realiza cada una de estas operaciones, lo que es crucial para monitorear la integridad de los datos y mantener un registro detallado de las actividades dentro del sistema.

## **Historias de usuario y requisitos no funcionales**

Estas historias de usuario y requisitos no funcionales, se derivan de las necesidades identificadas durante entrevistas y análisis de las actividades que se desarrollan en Ludo-Lab. A continuación, se presentan las historias de usuario y requisitos no funcionales que guiarán el desarrollo del componente:

### **Historias de usuario**

Los requisitos funcionales definen las funcionalidades que el sistema debe cumplir para satisfacer las necesidades de los usuarios. Estas funcionalidades se derivan de las historias de usuario y se detallan en la siguiente tabla:

Tabla 2.1: Historias de usuario.

<b>Nº</b>	<b>Código</b>	<b>Descripción</b>	<b>Módulo</b>
1	HU-001	Como usuario, quiero ver una pantalla de bienvenida al iniciar la aplicación, para orientarme sobre el sistema y acceder rápidamente a las funcionalidades más importantes.	Submódulo de Acceso y Registro
2	HU-002	Como usuario, quiero tener una pantalla de login, para poder acceder al sistema de manera segura y empezar a utilizar sus funcionalidades.	Submódulo de Acceso y Registro
3	HU-003	Como usuario, quiero tener una pantalla de registro, para poder enviar mis datos y crear una cuenta en el sistema para empezar a utilizar sus funcionalidades.	Submódulo de Acceso y Registro
4	HU-004	Como usuario, quiero crear una nueva contraseña en mi primer inicio de sesión, con recomendaciones de seguridad, para asegurar que mi cuenta esté protegida.	Submódulo de Gestión de Contraseñas
5	HU-005	Como usuario, quiero poder recuperar mi contraseña en caso de olvido, para poder acceder nuevamente al sistema.	Submódulo de Gestión de Contraseñas
6	HU-006	Como administrador, quiero verificar y aprobar o rechazar nuevos usuarios mediante el envío de correos electrónicos, para asegurar que solo personas autorizadas tengan acceso al sistema.	Submódulo de Aprobación de Usuarios
7	HU-007	Como administrador, quiero ver la auditoría del sistema, para monitorear todas las actividades importantes dentro del sistema.	Submódulo de Auditoría

Como se observa en la Tabla 2.1, estos requisitos cubren aspectos clave del sistema, incluyendo la autenticación, la gestión de usuarios y la auditoría. Cada uno de estas historias está diseñada para asegurar que el sistema brinde todas las funcionalidades necesarias

para un manejo eficiente y seguro de las actividades en LudoLab.

### Requisitos no funcionales

Los requisitos no funcionales establecen los criterios de comportamiento del sistema con un enfoque en la seguridad. La tabla siguiente presenta una descripción detallada de estos requisitos:

Tabla 2.2: Requisitos no funcionales

Nº	Código	Descripción	Módulo
1	RNF-001	Como administrador, quiero que las contraseñas se hasheen usando un algoritmo de hashing seguro en la base de datos, para asegurar que las contraseñas de los usuarios estén protegidas.	Submódulo de Gestión de Contraseñas
2	RNF-002	Como administrador, quiero limitar los intentos de inicio de sesión y bloquear la IP por 30 minutos después de múltiples intentos fallidos, para prevenir ataques de fuerza bruta.	Submódulo de Autenticación y Autorización
3	RNF-003	Como administrador, quiero gestionar y auditar todas las actividades del sistema, para asegurar que se puedan detectar y responder rápidamente a incidentes de seguridad.	Submódulo de Auditoría de Seguridad
4	RNF-004	Como usuario, quiero que la sesión expire automáticamente después de 30 minutos de inactividad, para asegurar la seguridad de mi cuenta.	Submódulo de Autenticación y Autorización
5	RNF-005	Como usuario, quiero ver una pantalla cuando intento acceder a un path que no existe, para saber que la página no está disponible.	Todos
6	RNF-006	Como usuario, quiero ver una pantalla cuando no estoy autorizado para acceder a ciertos contenidos, para saber que no tengo permisos suficientes.	Todos

*Continúa en la siguiente página*



Nº	Código	Descripción	Módulo
7	RNF-007	Como administrador, quiero que se validen todos los datos de entrada para prevenir inyecciones y otros ataques, para asegurar la integridad del sistema.	Todos

Tal como se muestra en la Tabla 2.2 estos requisitos aseguran que el sistema tenga un comportamiento adecuado en términos de protección de datos, prevención de accesos no autorizados, y manejo seguro de las sesiones y contraseñas.

## Análisis de riesgos

La identificación y evaluación de riesgos potenciales permite implementar estrategias de mitigación efectivas que reducen la probabilidad de que ocurran estos riesgos. En relación con LudoLab ConnectSys, el análisis de riesgos es particularmente importante debido a la sensibilidad de los datos procesados y las numerosas interacciones entre los distintos componentes del sistema.

A continuación, se presenta un análisis detallado de los riesgos identificados y las medidas de mitigación propuestas:

Tabla 2.3: Análisis de riesgos

Número de Riesgo	Fecha	Descripción del Riesgo	Fuente del Riesgo	Consecuencias del Riesgo	Calificación de Consecuencias	Calificación de Probabilidad	Calificación de Riesgo Inherente	Controles Planeados	Fecha de Revisión del Riesgo
1	20-may.-24	Fallos en la autenticación	Interna	Acceso no autorizado, posible fuga de datos	Alto	Alta	Alto	Autenticación robusta, revisión regular de logs	15-jul.-24
2	23-may.-24	Fuga de datos sensibles	Externa	Exposición de datos personales y sensibles	Muy Alto	Media	Muy Alto	Encriptación de datos en tránsito y en reposo, auditorías de seguridad periódicas	15-jul.-24
3	26-may.-24	Ataques de fuerza bruta	Externa	Acceso no autorizado, posible bloqueo del sistema	Medio	Alta	Alto	Limitar intentos de inicio de sesión, bloqueo de IP después de múltiples intentos fallidos	15-jul.-24
4	29-may.-24	Errores en la aprobación de usuarios	Interna	Acceso no autorizado, posible inclusión de usuarios falsos	Alto	Baja	Medio	Sistema de notificación para alertar sobre errores en la aprobación de usuarios	15-jul.-24
5	1-jun.-24	Acceso no autorizado a datos	Interna	Exposición de datos sensibles, posibles sanciones legales	Muy Alto	Alta	Muy Alto	Control de acceso basado en roles (RBAC), auditorías periódicas de acceso	15-jul.-24
6	4-jun.-24	Inyección de código malicioso	Externa	Compromiso del sistema, posible manipulación de datos	Alto	Media	Alto	Validación de todos los datos de entrada, uso de parámetros preparados para las consultas SQL	15-jul.-24
7	7-jun.-24	Pérdida de datos debido a fallos en el almacenamiento	Interna	Pérdida permanente de datos importantes	Muy Alto	Media	Muy Alto	Copias de seguridad regulares, procedimientos de recuperación de datos	15-jul.-24

Para mitigar los riesgos identificados en la Tabla 2.3, se han desarrollado varias especi-

ficaciones de seguridad que se integrarán en el sistema. Estas especificaciones aseguran que los riesgos sean abordados de manera efectiva, protegiendo así el sistema y sus datos.

1. **Fallos en la autenticación:** Para abordar los fallos en la autenticación, se implementará JWT para la gestión de sesiones y autenticación. Además, se aplicarán políticas estrictas para mejorar la calidad de las contraseñas, asegurando que contengan una combinación de caracteres especiales, números y letras mayúsculas y minúsculas. Las contraseñas se hashearán utilizando algoritmos seguros antes de ser almacenadas en la base de datos.
2. **Fuga de datos sensibles:** La protección contra la fuga de datos sensibles incluye la encriptación de la cédula utilizando AES (Advanced Encryption Standard), el hashing de contraseñas con algoritmos seguros, y el uso de HTTPS para asegurar los datos en tránsito. Estas medidas garantizan que los datos estén protegidos tanto durante su almacenamiento como durante su transmisión.
3. **Ataques de fuerza bruta:** Para prevenir ataques de fuerza bruta, se limitarán los intentos de inicio de sesión. Si se detectan múltiples intentos fallidos, la IP del usuario será bloqueada temporalmente. Este enfoque reduce significativamente el riesgo de acceso no autorizado por fuerza bruta.
4. **Errores en la aprobación de usuarios:** Para minimizar los errores en la aprobación de usuarios, se lanzará una pantalla de notificación si el proceso de aprobación falla.
5. **Acceso no autorizado a datos:** El control de acceso basado en roles (RBAC) será esencial para asegurar que solo los usuarios con los permisos adecuados puedan acceder a datos sensibles. Este control se asegura que cada usuario acceda únicamente a las áreas del sistema que le corresponden según su rol.
6. **Inyección de código malicioso:** Para proteger contra la inyección de código malicioso, se validarán todas las entradas del usuario utilizando parámetros preparados para las consultas SQL y técnicas de sanitización de datos. Esto previene que código malicioso sea ejecutado en el sistema, protegiendo la integridad y la seguridad de los datos.
7. **Pérdida de datos debido a fallos en el almacenamiento:** Finalmente, se recomienda que el cliente implemente un programa de copias de seguridad regulares. Aunque

esta tarea no será gestionada directamente, es crucial que el cliente programe y realice copias de seguridad periódicas para asegurar la recuperación de datos en caso de fallos en el almacenamiento.

En conjunto, estas especificaciones de seguridad forman una defensa integral y multifacética contra los riesgos identificados, asegurando que el sistema LudoLab ConnectSys opere de manera segura y confiable. Al implementar estas medidas, se fortalece la protección de los datos sensibles, se previene el acceso no autorizado y se garantiza la integridad y disponibilidad del sistema en todo momento.

### **2.3.2. Diseño**

El diseño de la solución es una fase crucial en el desarrollo de LudoLab ConnectSys, donde se define la estructura y la arquitectura que soportarán el sistema. Esta sección proporciona una visión detallada de la base de datos, la arquitectura de software y el modelo de amenazas, así como las interfaces de usuario que se implementarán.

#### **Base de datos**

Para empezar, el diseño de la base de datos es fundamental para asegurar la integridad y seguridad de la información. El siguiente diagrama ilustra la estructura de la base de datos utilizada en LudoLab ConnectSys:

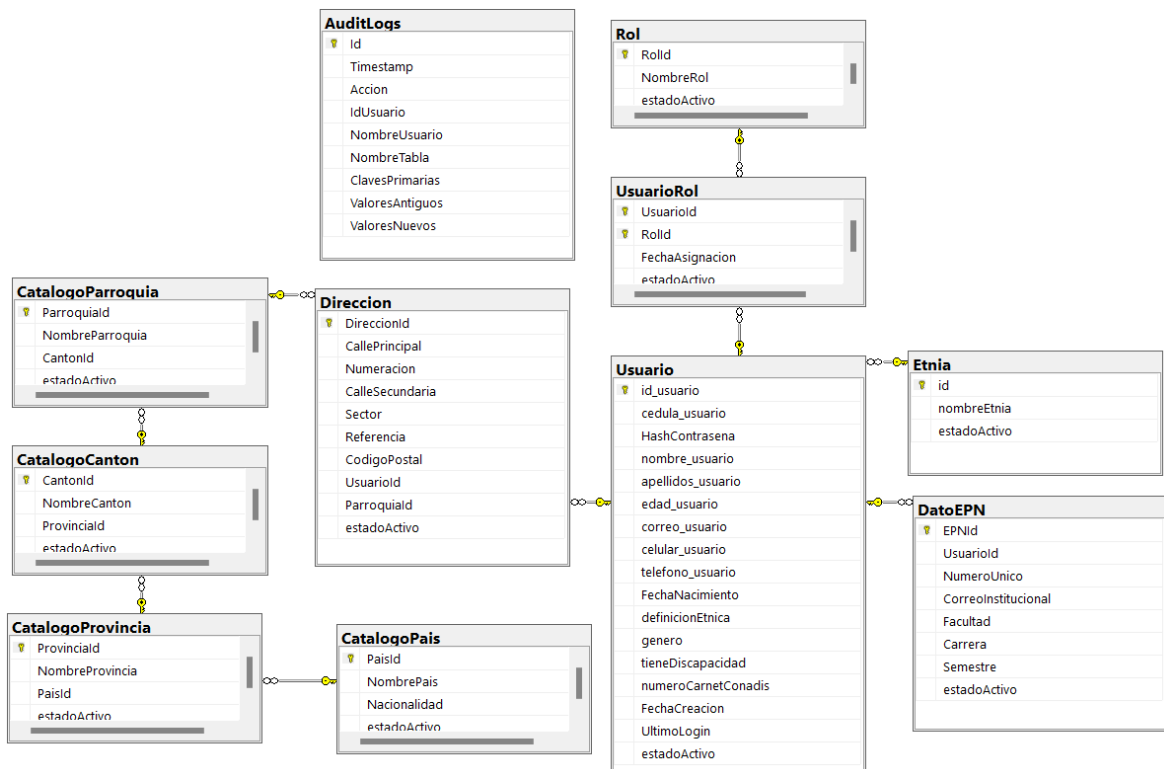


Figura 2.1: Diagrama de base de datos.

Aunque la base de datos completa es extensa (Anexo C), el componente de seguridad se centra específicamente en manejar las siguientes tablas:

- Usuario
- UsuarioRol
- Rol
- Dirección
- Catalogo País
- Catalogo Provincia
- Catalogo Cantón
- Catalogo Parroquia
- Datos EPN
- Etnia

- AuditLogs

Estas tablas son esenciales para gestionar la información de los usuarios y asegurar la correcta auditoría de sus acciones dentro del sistema.

## Arquitectura

La arquitectura de LudoLab ConnectSys está diseñada para ser escalable, segura y eficiente, asegurando que cada componente del sistema cumpla con sus responsabilidades de manera óptima. A continuación se presenta un diagrama C4 que muestra el contexto y la aplicación del sistema:

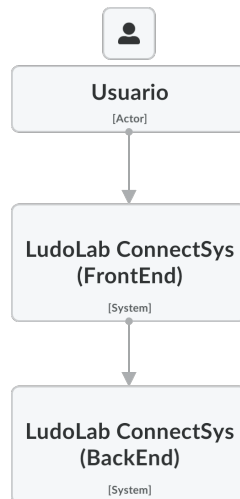


Figura 2.2: Diagrama de contexto.

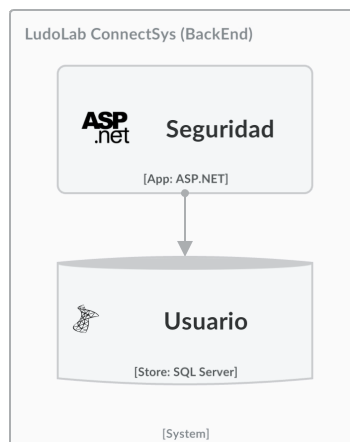


Figura 2.3: Diagrama de aplicación.

El diseño arquitectónico se divide en tres componentes principales: frontend, backend y

base de datos, como se ve en la Figura 2.2 y Figura 2.3. Cada uno de estos componentes juega un papel crucial en la funcionalidad y seguridad general del sistema.

## Frontend

Está desarrollado utilizando Blazor .NET 8, una tecnología que permite crear interfaces de usuario interactivas y modernas. Blazor facilita la construcción de aplicaciones web utilizando C#, lo que permite una integración más fluida con el backend y proporciona una experiencia de usuario coherente. [27]

## Backend

Está compuesto por varios microservicios implementados con Web API .NET 8. Esta estructura modular permite que cada microservicio maneje una parte específica de la funcionalidad del sistema, asegurando que las tareas se realicen de manera eficiente y escalable.

El siguiente diagrama C4 muestra los componentes de la arquitectura:

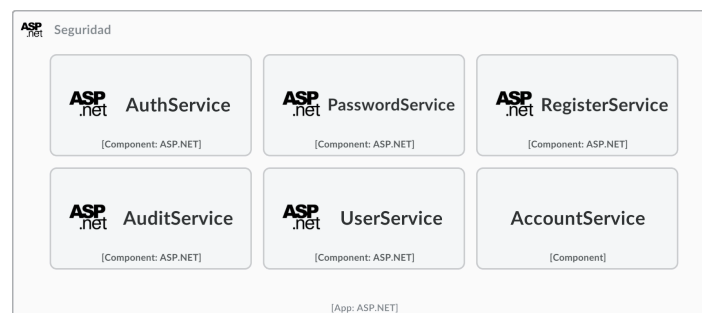


Figura 2.4: Diagrama de componentes.

Los microservicios creados son esenciales para gestionar diferentes aspectos del sistema. Cada microservicio tiene una función específica y trabaja en conjunto con los demás para proporcionar una solución completa y segura. A continuación se describen brevemente los microservicios vistos en la Figura 2.4:

1. **AuthService:** Recibe la cédula, contraseña y el rol para generar el JWT que permite controlar la sesión del usuario y autorizar en las páginas correspondientes.
2. **PasswordService:** Se encarga de generar y almacenar las contraseñas siguiendo las políticas establecidas.
3. **RegisterService:** Maneja el registro de nuevos usuarios en el sistema a través de la página de registro.

4. **AuditService:** Actúa como un interceptor para guardar los registros de creación, actualización y eliminación que se desarrollen en el sistema principal.
5. **UserService:** Se encarga de todas las operaciones CRUD que involucren directamente al usuario.
6. **AccountService:** Gestiona la aprobación del registro de usuarios y envía notificaciones por correo electrónico.

## **Base de datos**

La base de datos utilizada es SQL Server y la versión que se usó para el desarrollo es Microsoft SQL Server 2022 (RTM-GDR) (KB5040936) - 16.0.1121.4 (X64), en este caso se usó la Developer Edition (64-bit).

## **Modelo de amenazas**

El análisis del modelo de amenazas es un paso crucial en el desarrollo de aplicaciones seguras, ya que permite identificar y mitigar posibles riesgos antes de que se conviertan en vulnerabilidades explotables. Este proceso involucra una evaluación detallada de las posibles amenazas que podrían afectar la seguridad del sistema, considerando factores como la arquitectura, los flujos de datos y las posibles interacciones con el entorno externo. A continuación, se presenta un modelo de amenazas obtenido mediante el uso de la herramienta Microsoft Threat Modeling Tool que se vio en el Sección 1.3. Esta herramienta nos proporciona una lista de amenazas, algunas de las cuales abordaremos mientras que otras no podrán ser resueltas en su totalidad. Además, es importante destacar que las amenazas de seguridad son dinámicas y siempre habrá nuevas vulnerabilidades que surgirán con el tiempo.

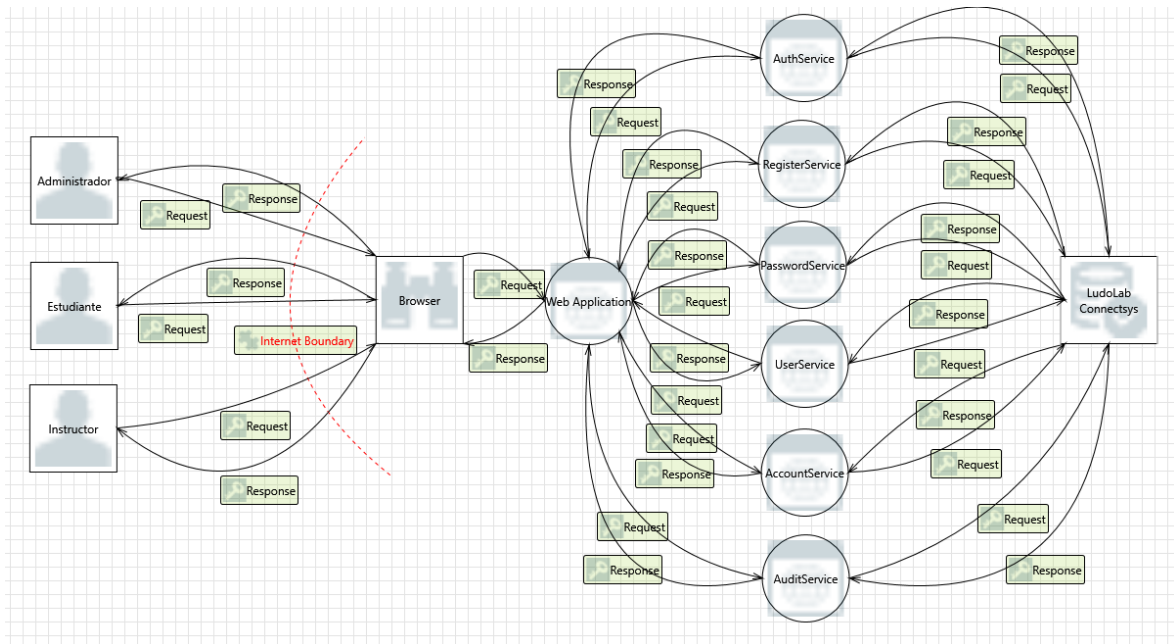


Figura 2.5: Modelo de amenazas.

El modelo de amenazas generado resalta varios aspectos críticos a considerar para mantener la seguridad del sistema. Primero, se identificaron las amenazas relacionadas con la autenticación y autorización de usuarios. Estas amenazas pueden surgir de intentos de acceso no autorizado, lo que podría comprometer datos sensibles. Para mitigar estos riesgos, se implementaron políticas de contraseñas robustas, la utilización de tokens de autenticación y la gestión de sesiones para prevenir accesos no autorizados.

Además, se destacaron amenazas vinculadas a la integridad de los datos, especialmente en las operaciones CRUD (Create, Read, Update, Delete). Estas operaciones son puntos críticos donde un atacante podría intentar manipular datos importantes. La implementación de registros de auditoría y controles de acceso estrictos son medidas esenciales para asegurar que solo usuarios autorizados puedan realizar modificaciones en los datos.

El modelo también subraya la importancia de proteger los datos en tránsito y en reposo. La encriptación de datos utilizando AES para información sensible, como números de cédula, y el uso de HTTPS para todas las comunicaciones, aseguran que los datos no sean interceptados ni alterados durante su transmisión o almacenamiento.

Otra área clave es la detección y respuesta a incidentes. El modelo de amenazas incluye la capacidad de registrar y monitorear todas las actividades críticas dentro del sistema, permitiendo una respuesta rápida a cualquier actividad sospechosa o no autorizada. Esto



se logra mediante la implementación de un microservicio de auditoría que registra todas las acciones relevantes y las asocia con los usuarios correspondientes.

En última instancia, el análisis del modelo de amenazas ayuda a identificar posibles puntos débiles en la infraestructura de red y su configuración. Asegurar que todos los componentes del sistema estén correctamente configurados y protegidos contra accesos no autorizados es fundamental para mantener la integridad y disponibilidad del sistema.

El reporte completo generado por la herramienta de modelado de amenazas puede ser consultado en el Anexo E, donde se detallan todas las amenazas identificadas y las medidas de mitigación propuestas.

## Mockups

Las interfaces de usuario son la parte del sistema con la que interactúan directamente los usuarios. A continuación, se presentan los mockups que se plantearon usando la herramienta Figma, que incluyen las pantallas de bienvenida, login y registro.



Figura 2.6: Pantalla de bienvenida.

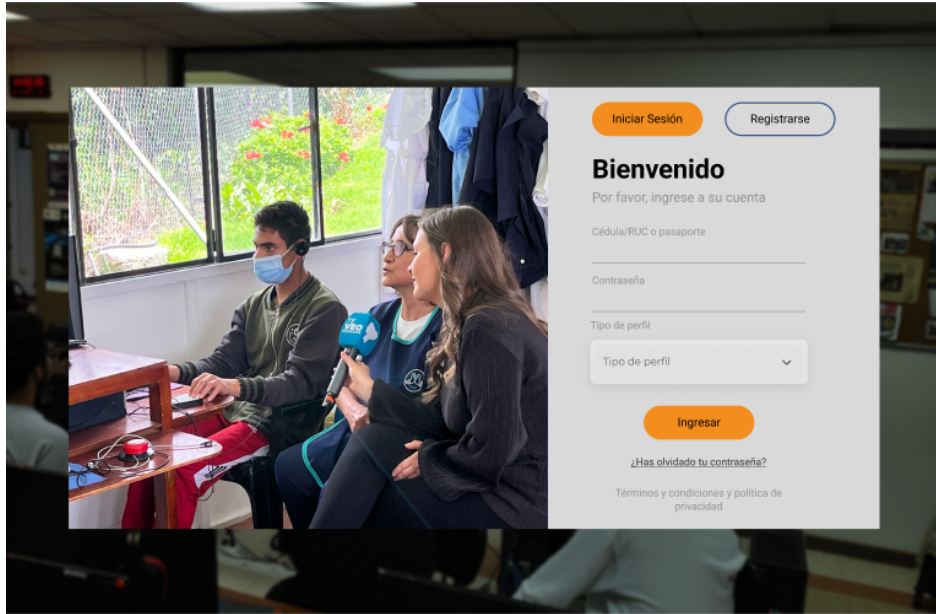


Figura 2.7: Pantalla de ingreso.

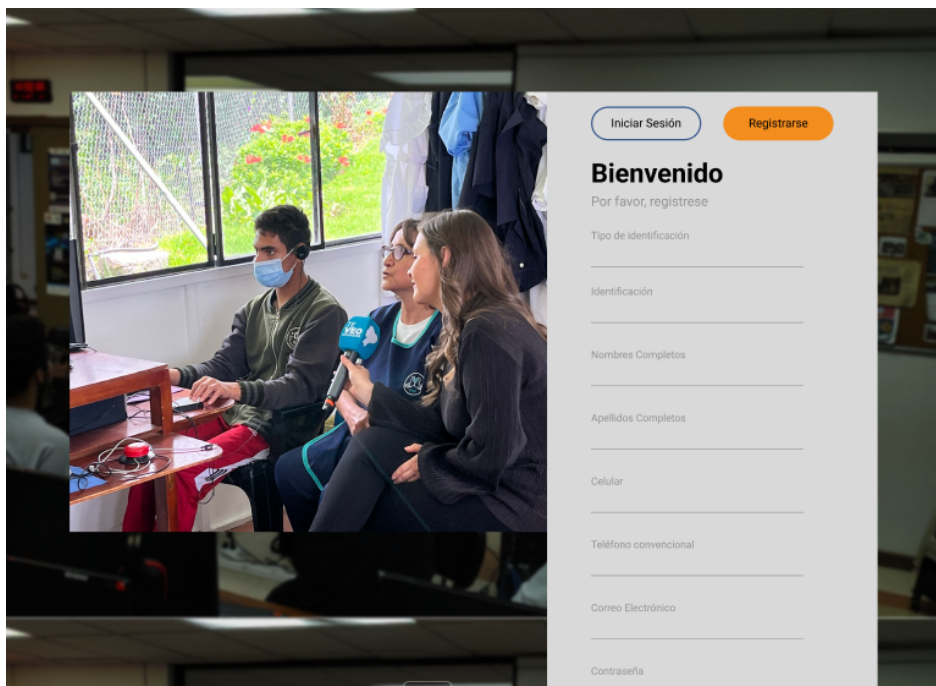


Figura 2.8: Pantalla de registro.

### 2.3.3. Implementación

La fase de implementación es crucial en el desarrollo de un sistema, ya que es cuando las ideas y diseños se convierten en un producto funcional. En esta sección, se describe cómo se implementaron los diferentes componentes del sistema

## Estructura del Proyecto

### Frontend

El frontend del proyecto se desarrolló utilizando Blazor. En la Figura 2.9 se puede ver cómo se estructura en carpetas según el rol de usuario (Administración, Instructor y Estudiante). Mi parte se centró en la carpeta Principal, que contiene las primeras pantallas que verá el usuario, como la pantalla de bienvenida, la pantalla de login, pantalla de registro, entre otras.

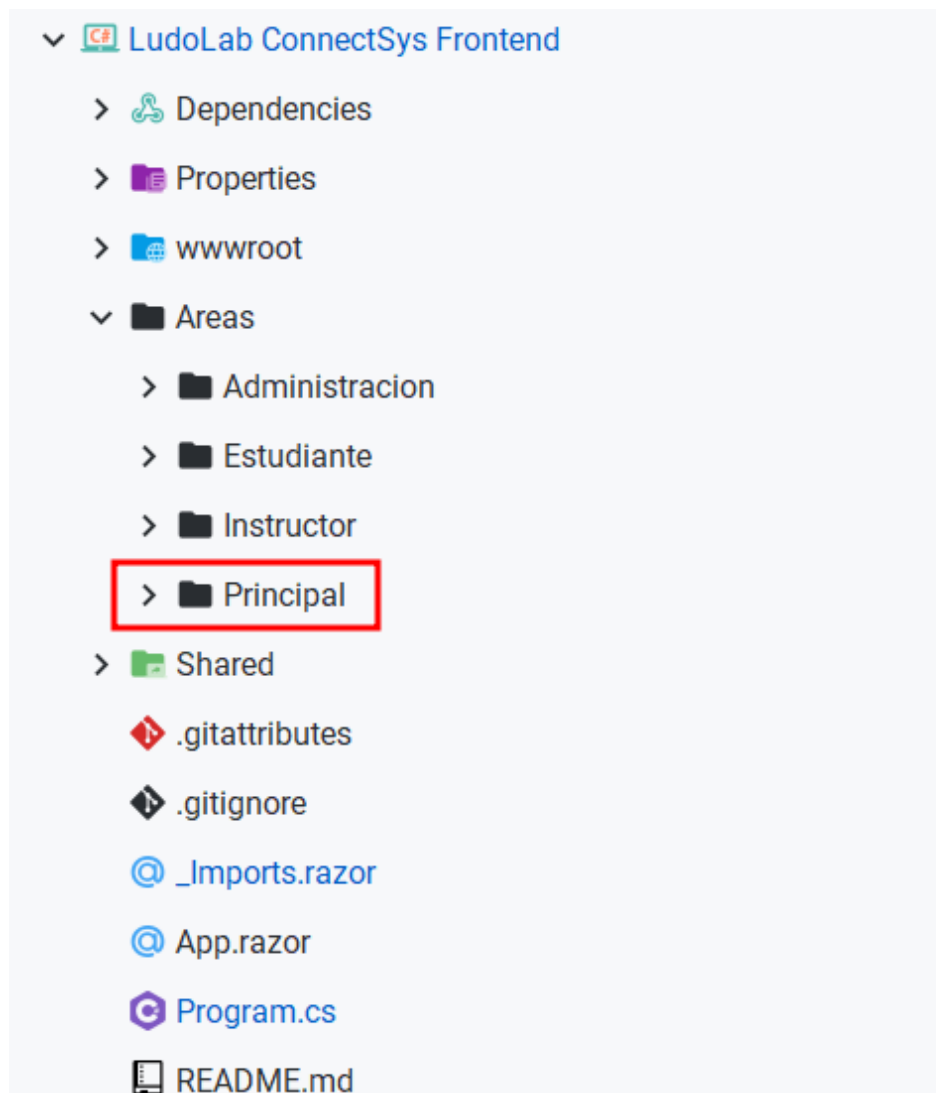


Figura 2.9: Estructura del frontend.

## Backend

El componente backend fue desarrollado principalmente por mis compañeros. Sin embargo, realicé modificaciones para agregar la autorización y un interceptor que registra los cambios en la base de datos, asegurando que todas las operaciones crear, actualizar y eliminar sean monitoreadas y registradas adecuadamente.

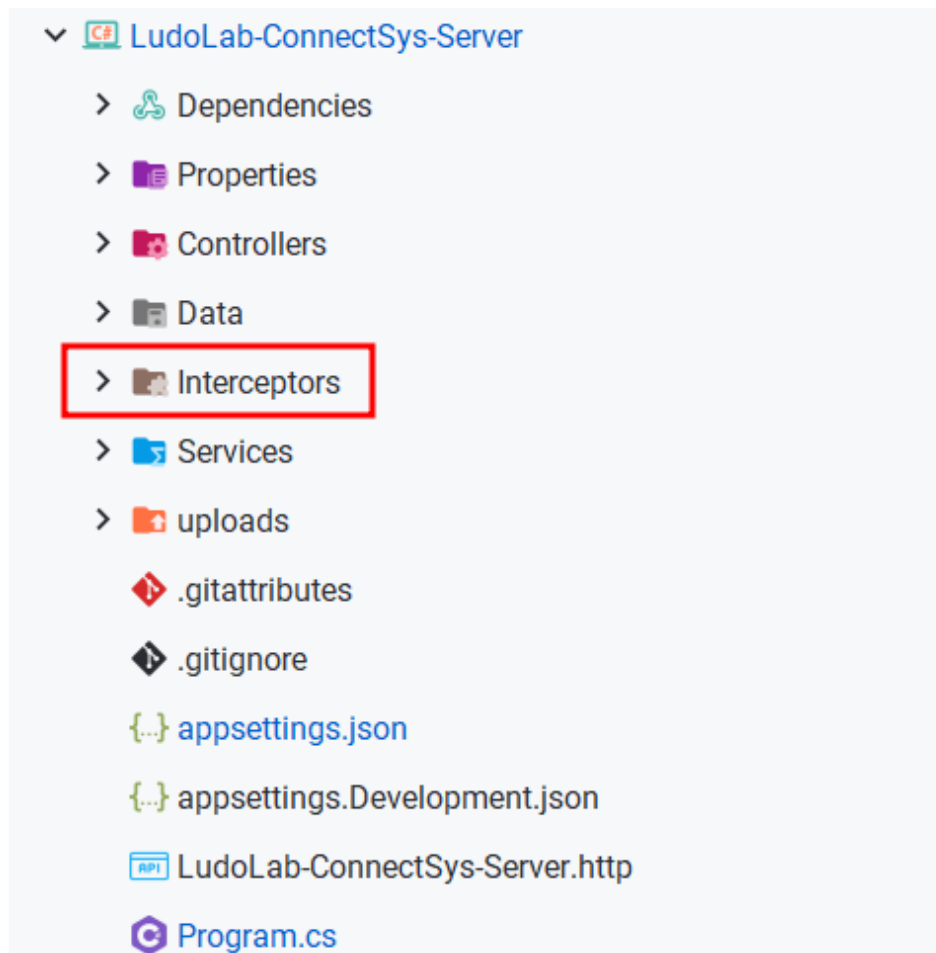


Figura 2.10: Estructura del backend.

## Biblioteca de clases

La biblioteca de clases fue desarrollada por mis compañeros y contiene los modelos de la base de datos y los modelos DTO (Data Transfer Objects). Esta biblioteca es compartida entre el frontend y el backend, proporcionando una estructura unificada y coherente para manejar los datos a lo largo de todo el sistema.

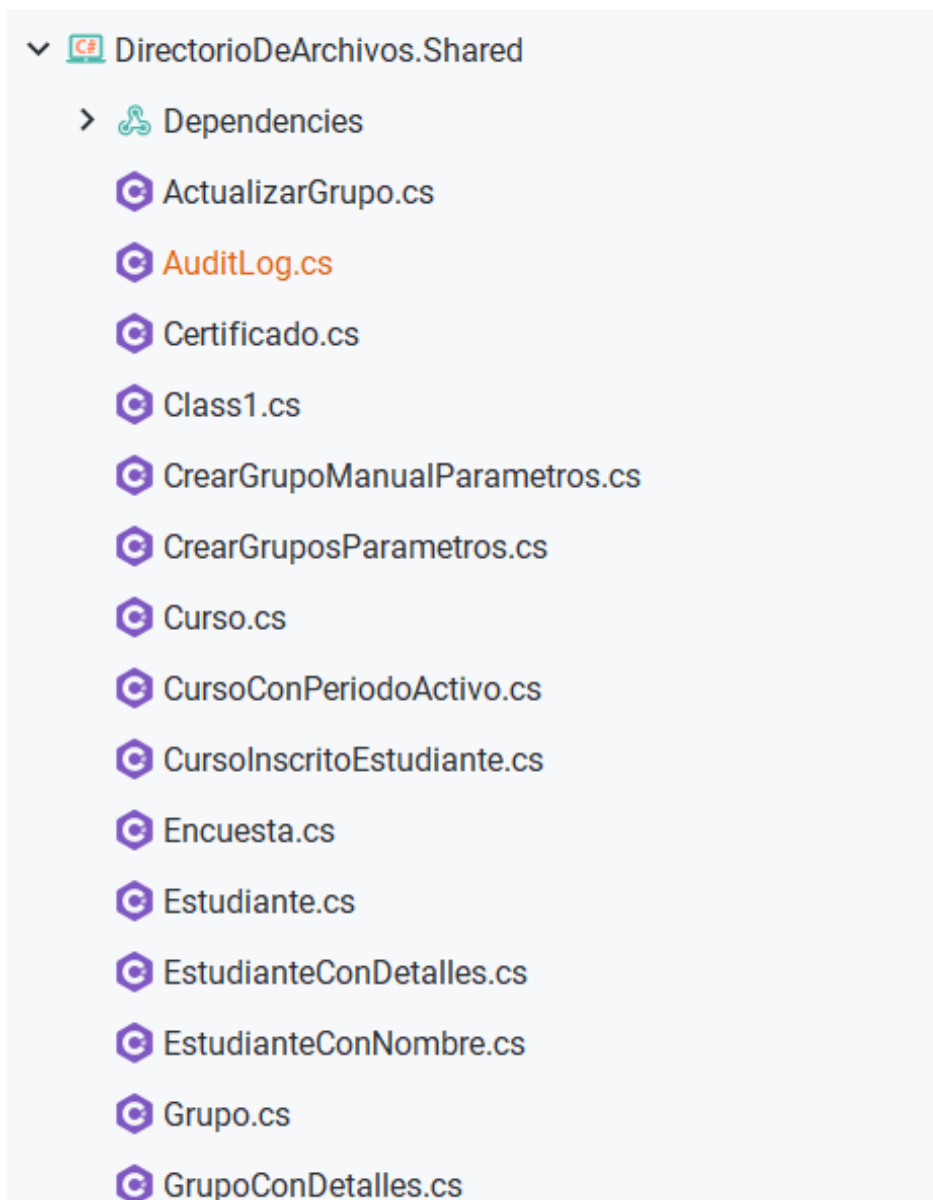


Figura 2.11: Estructura de la biblioteca de clases,.

## Microservicios

Los microservicios son componentes esenciales del backend, cada uno diseñado para manejar tareas específicas de manera eficiente y segura. A continuación, se muestra todos los microservicios desarrollados:

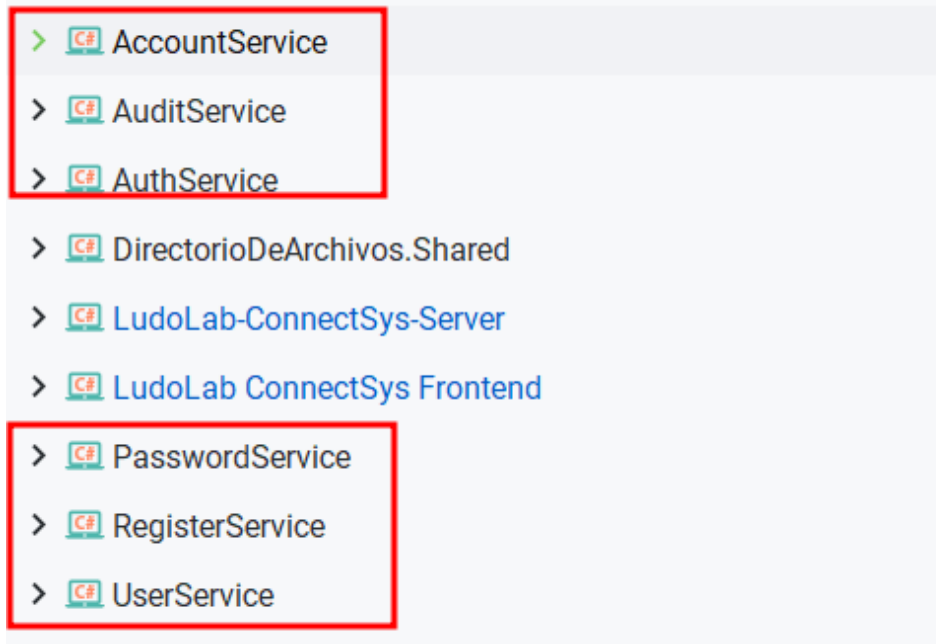


Figura 2.12: Microservicios.

Todos los microservicios comparten una misma estructura, lo que facilita su desarrollo, mantenimiento y escalabilidad. Esta estructura común incluye componentes esenciales como controladores, data, servicios y modelos, como se ve en la Figura 2.13.

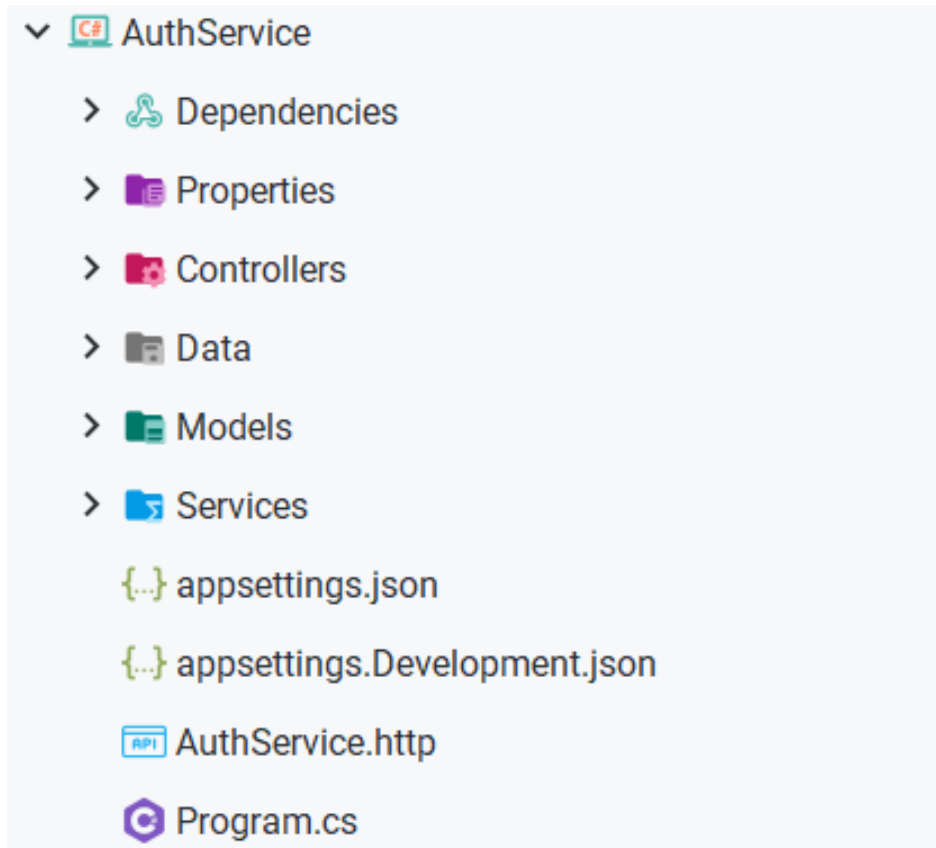


Figura 2.13: Estructura del microservicio AuthService.

Estos microservicios contienen modelos propios que solo toman los atributos necesarios para su uso específico. Además, se siguieron las buenas prácticas de seguridad para WEB API's .NET 8, tal como lo describió Sukhpinder Singh. [36] Esta independencia y adherencia a las prácticas de seguridad aseguran que cada microservicio pueda operar de manera segura y eficiente.

### **Submódulo de Acceso y Registro**

El desarrollo del Submódulo de Acceso y Registro fue el primer paso en la implementación del sistema, debido a su importancia fundamental para garantizar que los usuarios puedan interactuar con la aplicación de manera segura y eficiente desde el inicio. Este submódulo incluye las interfaces de bienvenida, login y registro, que son esenciales para cualquier sistema de administración.

## Historias de Usuario Involucradas

- **HU-001:** Como usuario, quiero ver una pantalla de bienvenida al iniciar la aplicación, para orientarme sobre el sistema y acceder rápidamente a las funcionalidades más importantes.
- **HU-002:** Como usuario, quiero tener una pantalla de login, para poder acceder al sistema de manera segura y empezar a utilizar sus funcionalidades.
- **HU-003:** Como usuario, quiero tener una pantalla de registro, para poder enviar mis datos y crear una cuenta en el sistema para empezar a utilizar sus funcionalidades.
- **RNF-007:** Como administrador, quiero que se validen todos los datos de entrada para prevenir inyecciones y otros ataques, para asegurar la integridad del sistema.

Estas historias de usuario iniciaron desde el estado **Backlog** del tablero de Kanban, utilizando la herramienta ClickUp para gestionar el flujo de trabajo. El primer paso fue mover estas historias al estado **Pendiente**, preparándolas para su desarrollo inmediato.

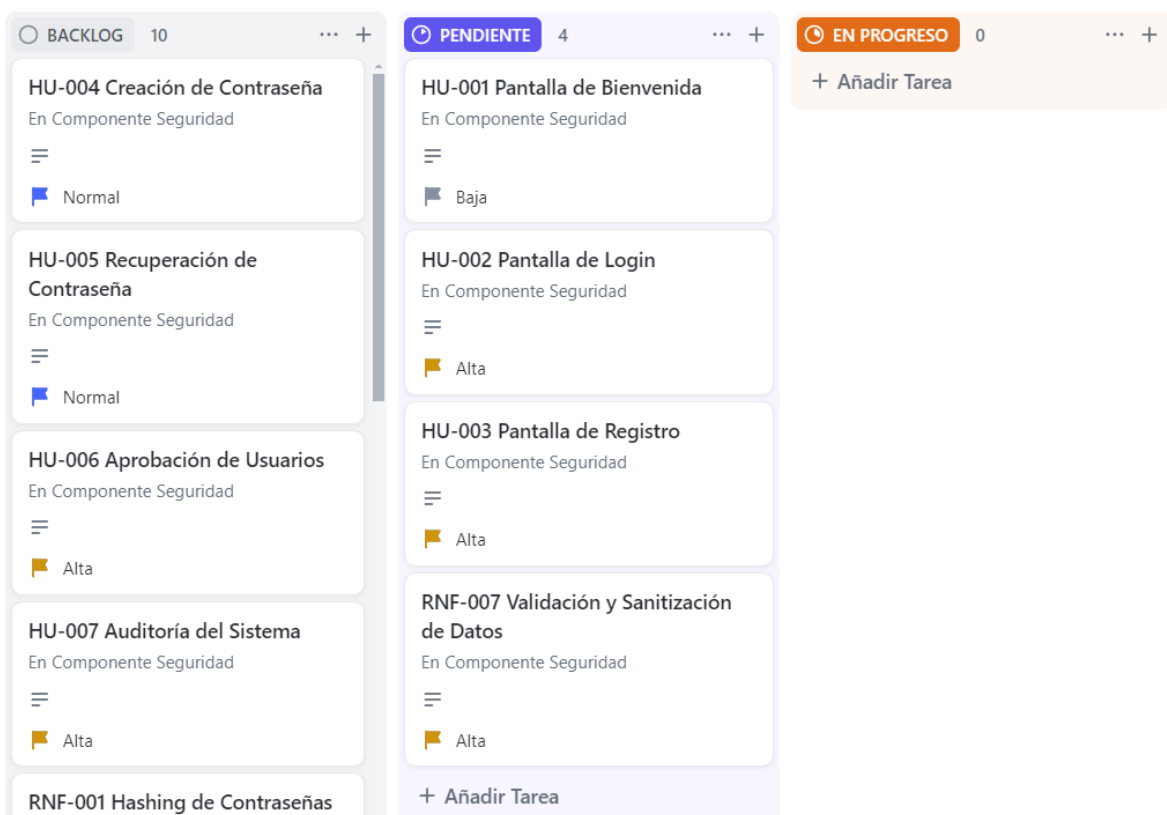


Figura 2.14: Tablero Kanban en estado Pendiente para el Submódulo de acceso y registro.



Una vez identificadas las historias de usuario de mayor prioridad, específicamente las de la pantalla de login y registro, se procedió a cambiarlas al estado **En Progreso**. Estas historias fueron seleccionadas debido a su alta prioridad en la secuencia de uso del sistema, ya que los usuarios deben poder autenticarse y registrarse antes de acceder a otras funcionalidades.

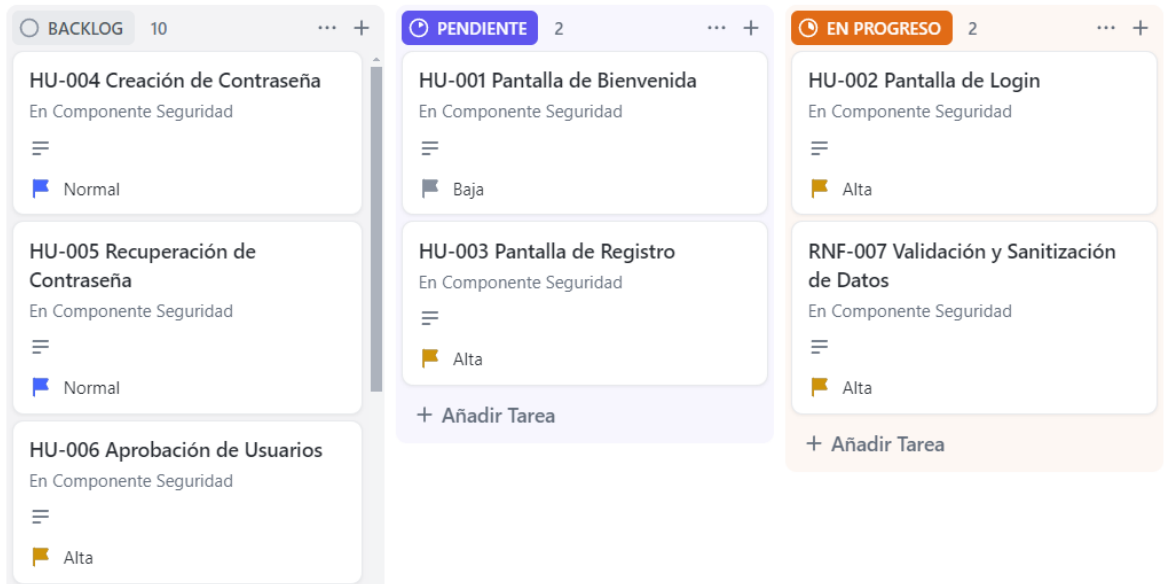


Figura 2.15: Tablero Kanban en estado En Progreso para el Submódulo de acceso y registro.

Posteriormente, se incluyó el desarrollo de la pantalla de bienvenida, la cual se abordó una vez que las otras dos historias de usuario fueron completadas. Esto permitió tener una transición suave y una experiencia de usuario coherente desde el primer contacto con la aplicación.

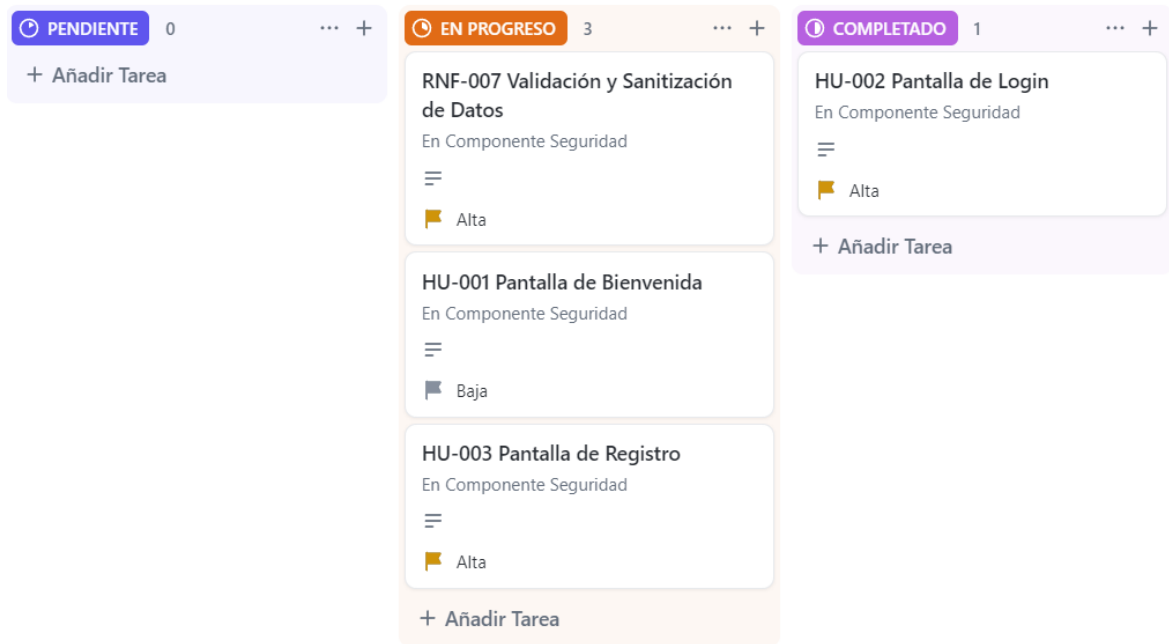


Figura 2.16: Tablero Kanban en estado En Progreso para el Submódulo de acceso y registro.

## Microservicios

Para este submódulo, se desarrollaron dos microservicios esenciales:

- **AuthService:** En una etapa temprana, este microservicio se enfocaba únicamente en permitir a los usuarios logearse en el sistema. La funcionalidad inicial se limitaba a verificar que los datos ingresados coincidieran con los de la base de datos, sin implementar aún JWT, autenticación ni autorización avanzadas.
- **RegisterService:** Este microservicio maneja el registro de nuevos usuarios en el sistema a través de la página de registro, validando y almacenando los datos de los usuarios de manera segura.

A continuación, se presentan las interfaces desarrolladas para estas historias de usuario. La pantalla de bienvenida orienta al usuario, la pantalla de login permite una autenticación segura y la pantalla de registro facilita la creación de nuevas cuentas.



Figura 2.17: Submódulo de acceso y registro - Pantalla de bienvenida.



Figura 2.18: Submódulo de acceso y registro - Pantalla de ingreso.

The image shows a web form for user registration. The title is "Registro de Usuario". Below the title is a paragraph of text: "Toma en cuenta que su registro nos permitirá facilitar la toma de datos, mas no tendrá su cuenta activa de forma inmediata. El administrador del sistema verificará los datos y después de ello se le enviarán las credenciales para que pueda ingresar al sistema. Ahora para su registro es importante que seleccione el tipo de usuario que va a ser dentro del sistema." Below this text are two dropdown menus: "Tipo de Usuario" with "Instructor" selected, and "¿Es estudiante de la EPN?" with "Si" selected. An orange "Enviar" button is positioned below these dropdowns. Below the registration section is a "Datos Personales" section with several input fields: "Tipo de Identificación" (with "Cédula" selected), "Identificación" (text input), "Nombres" (text input), "Apellidos" (text input), "Autodefinición Étnica" (text input), and "Género" (text input).

Figura 2.19: Submódulo de acceso y registro - Pantalla de registro.

El código más relevante de estas pantallas, es del funcionamiento de Login. El siguiente código muestra cómo se guarda el token en el localStorage del navegador y toma el rol seleccionado en el formulario para la redirección de las páginas según el usuario, siempre y cuando el AuthService devuelva una respuesta positiva.

```

private async Task HandleValidSubmit()
{
    var response = await Http.PostAsJsonAsync( requestUri: "https://localhost:7000/auth/login", _loginRequest);

    if (response.IsSuccessStatusCode)
    {
        var loginResponse = await response.Content.ReadFromJsonAsync<LoginResponse>();

        // Guardar en localStorage
        await Js.InvokeVoidAsync( identifier: "localStorage.setItem", "authToken", loginResponse?.Token);

        // Redirigir al usuario a la página correspondiente según el rol
        switch (_loginRequest.Rol)
        {
            case "Administrador":
                Navigation.NavigateTo(uri: "/adminDashboard");
                break;
            case "Estudiante":
                Navigation.NavigateTo(uri: "/studentDashboard");
                break;
            case "Instructor":
                Navigation.NavigateTo(uri: "/instructorDashboard");
                break;
            default:
                await SweetAlertNotify( texto: "Rol no válido", tipo: "error");
                break;
        }
    }
    else
    {
        var error:string = await response.Content.ReadAsStringAsync();
        Console.WriteLine(error);
    }
}

```

Figura 2.20: Función HandleValidSubmit() para el Login.

Luego, se crea un servicio para la lectura del token almacenado en localStorage, el cual debe ser extraído y leído para manejar la autorización o el acceso a las páginas de la aplicación.

```

public override async Task<AuthenticationState> GetAuthenticationStateAsync()
{
    var token = await _localStorageService.GetItemAsync<string>("authToken");

    var identity = string.IsNullOrEmpty(token)
        ? new ClaimsIdentity()
        : new ClaimsIdentity(ParseClaimsFromJwt(token), "jwt");

    var user = new ClaimsPrincipal(identity);

    _httpClient.DefaultRequestHeaders.Authorization = token != null
        ? new AuthenticationHeaderValue("Bearer", token)
        : null;

    return new AuthenticationState(user);
}

```

Figura 2.21: Función GetAuthenticationStateAsync() para token.

Al finalizar el desarrollo de estas historias de usuario, se las pasó al estado Completado. Cabe recalcar que el requerimiento no funcional RNF-007, relacionado con la validación y sanitización de datos, se mantuvo en el estado **En Progreso** hasta que todas las historias de usuario fueran completadas. Esto se debe a que la validación y sanitización de datos es una tarea transversal que debe aplicarse a todas las entradas de usuario en el sistema.

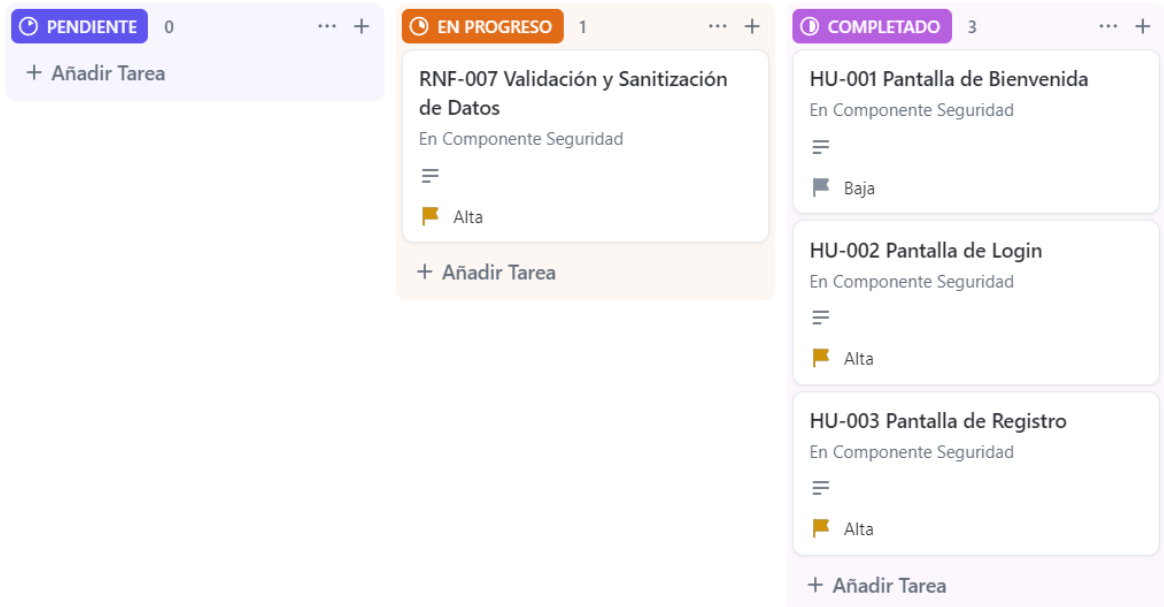


Figura 2.22: Tablero Kanban en estado Completado para el Submódulo de acceso y registro.

Después de completar el desarrollo, las historias de usuario se movieron al estado de **Pruebas**. Esta fase es crucial para asegurar que las funcionalidades implementadas cumplan con los criterios de aceptación definidos y funcionen correctamente sin errores. La sección de Evaluación y Resultados 3 detallará este proceso de manera más exhaustiva.

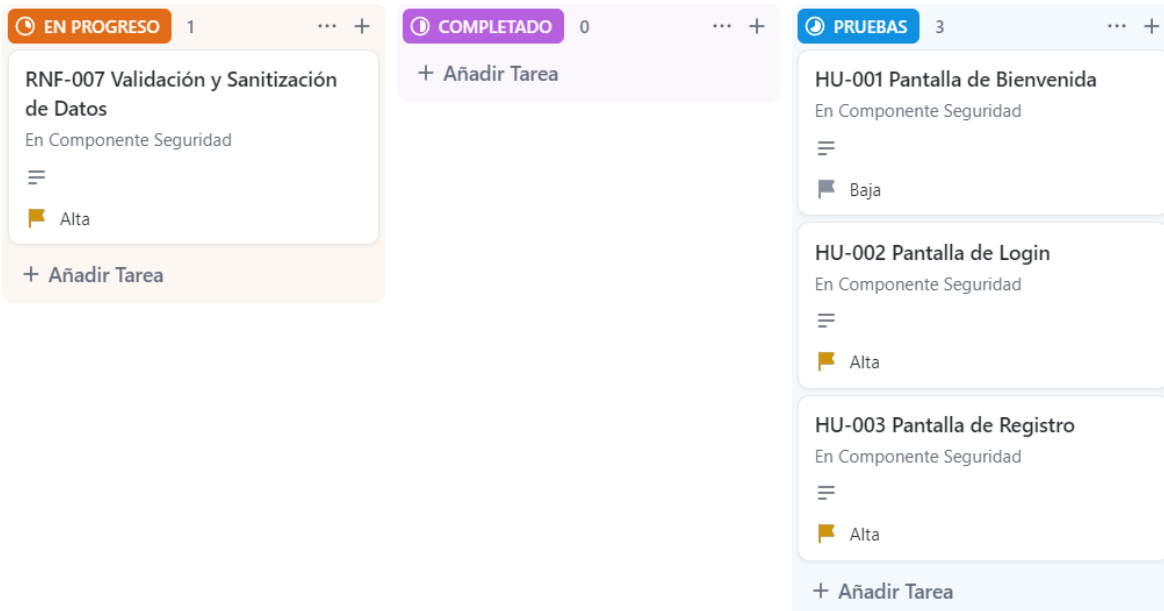


Figura 2.23: Tablero Kanban en estado Pruebas para el Submódulo de acceso y registro.

Una vez que las pruebas fueron satisfactorias, las historias de usuario pasaron al estado de **Integración**. En esta etapa, se integraron las primeras pantallas del sistema con los componentes desarrollados por mis compañeros, asegurando una coherencia y funcionalidad completas en toda la aplicación.

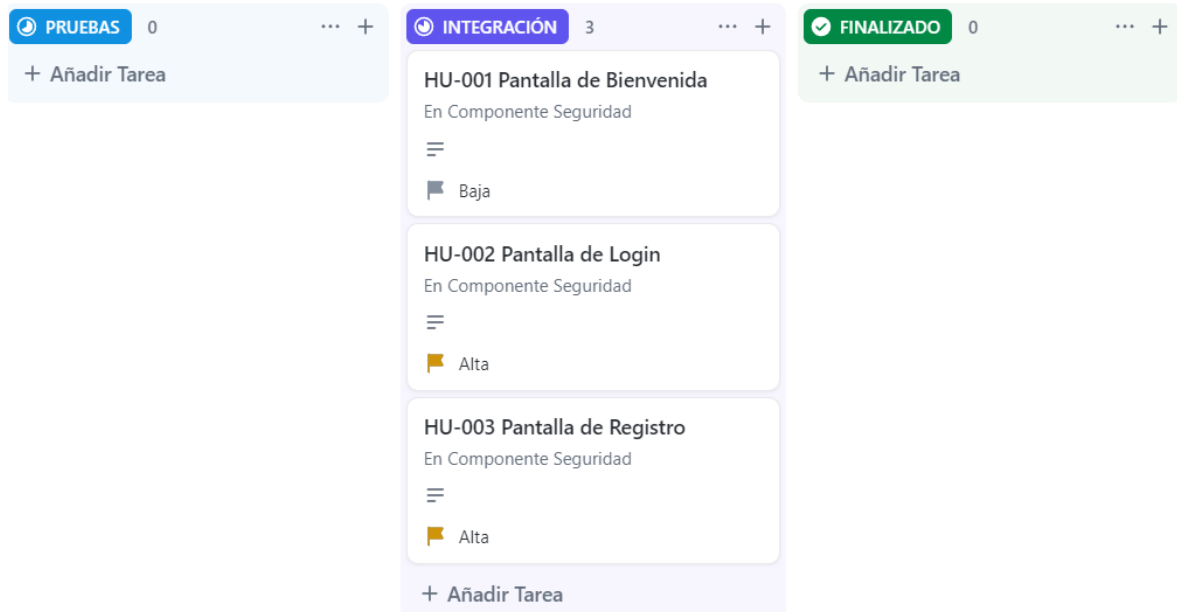


Figura 2.24: Tablero Kanban en estado Integración para el Submódulo de acceso y registro.

Finalmente, cuando todo el proceso de las historias de usuario fue culminado con éxito, se movieron al último estado del tablero de Kanban, que es **Finalizado**. Este estado indica que las funcionalidades están completamente desarrolladas, probadas e integradas en el sistema.



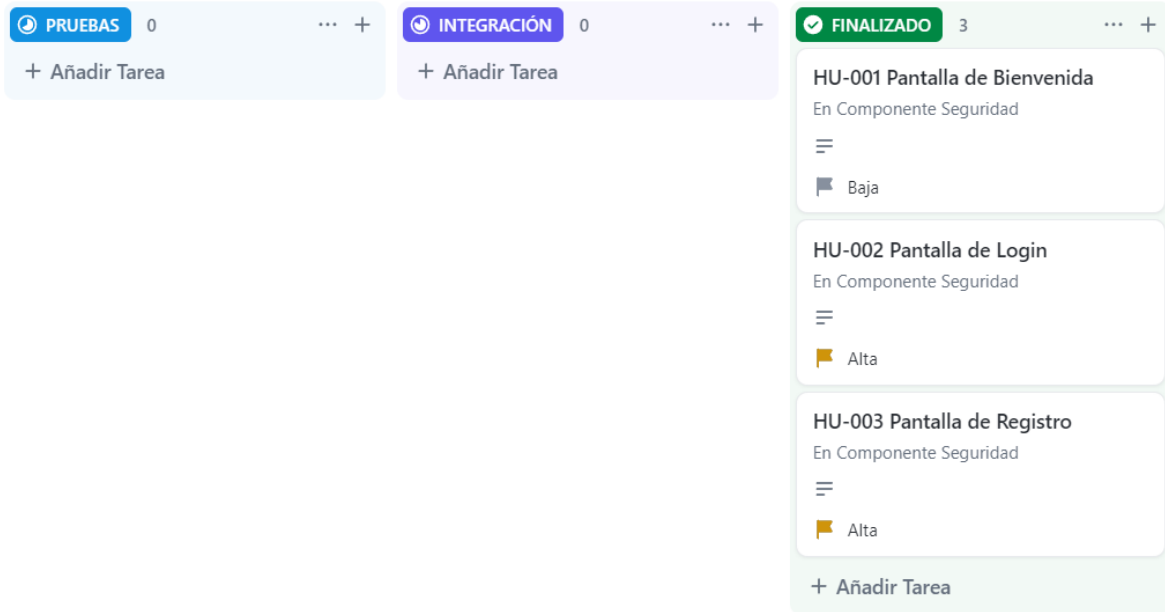


Figura 2.25: Tablero Kanban en estado Finalizado para el Submódulo de acceso y registro.

## Submódulo de Aprobación de Usuarios

El Submódulo de Aprobación de Usuarios es fundamental para asegurar que solo personas verificadas puedan acceder al sistema. Este submódulo permite a los administradores revisar y aprobar o rechazar nuevas solicitudes de usuarios, manteniendo así la integridad y seguridad del sistema. La implementación de este submódulo garantiza que la base de datos esté libre de registros innecesarios y que los usuarios tengan los permisos adecuados.

### Historias de usuario involucradas

- **HU-006:** Como administrador, quiero verificar y aprobar o rechazar nuevos usuarios mediante el envío de correos electrónicos, para asegurar que solo personas autorizadas tengan acceso al sistema.

Este submódulo se inicia con la siguiente configuración en el tablero Kanban, en el estado **Pendiente**:

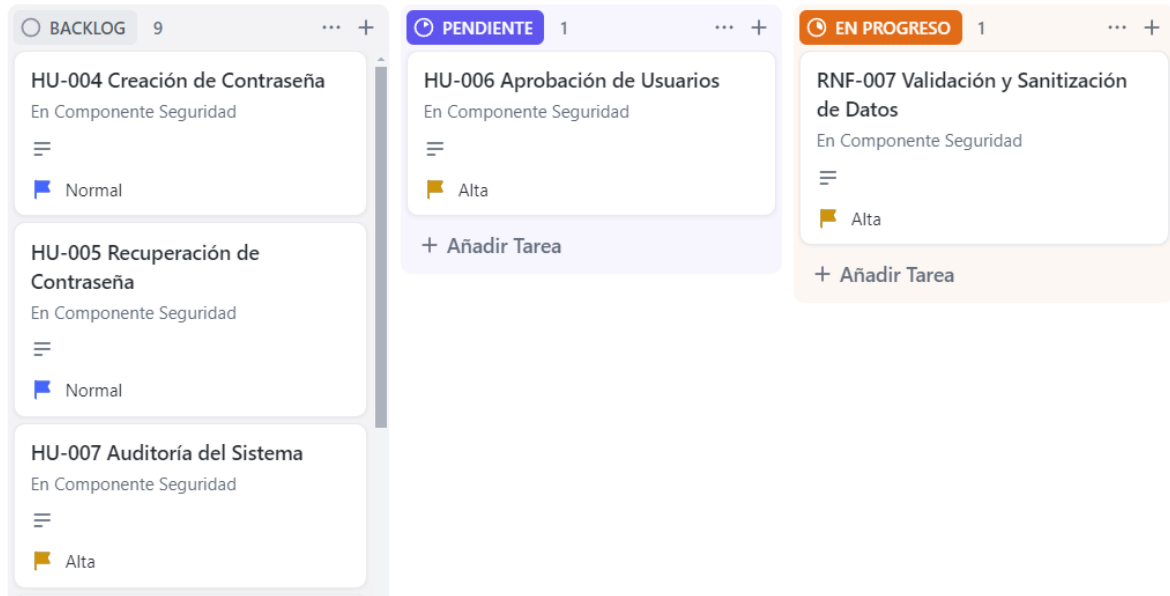


Figura 2.26: Tablero Kanban en estado Pendiente para el Submódulo de aprobación de usuarios.

Una vez que las historias de usuario han sido priorizadas, se mueven al estado En Progreso. Para este submódulo, se desarrollaron los siguientes componentes clave:

### Microservicios

- **AccountService:** Este microservicio es responsable de gestionar la aprobación de los registros de usuarios. Permite a los administradores aprobar o rechazar nuevas solicitudes y enviar correos electrónicos de notificación a los usuarios.
- **UserService:** Este microservicio se encarga de manejar todas las operaciones CRUD relacionadas con los usuarios. Esto incluye la creación, lectura, actualización y eliminación de registros de usuarios, asegurando que toda la información esté actualizada y se gestione de manera segura.

Se desarrolló una pantalla donde los administradores pueden visualizar una lista de usuarios pendientes de aprobación. Esta pantalla incluye opciones para aprobar, denegar y ver más detalles de cada usuario.

Aprobar usuarios					
Id	Nombre	Apellido	Edad	Correo	Acciones
1	Juan	Pérez	30	juan.perez@example.com	<a href="#">Aprobar</a> <a href="#">Rechazar</a> <a href="#">Ver usuario</a>
2	Ana	García	25	ana.garcia@example.com	<a href="#">Aprobar</a> <a href="#">Rechazar</a> <a href="#">Ver usuario</a>
3	Luis	Martínez	28	luis.martinez@example.com	<a href="#">Aprobar</a> <a href="#">Rechazar</a> <a href="#">Ver usuario</a>

Figura 2.27: Pantalla de aprobación de usuarios.

Después de completar el desarrollo, las historias de usuario pasan por varias fases antes de ser finalizadas. Primero, se someten a pruebas para asegurar que todas las funcionalidades implementadas funcionen correctamente y cumplan con los criterios de aceptación definidos.

Una vez que las pruebas son satisfactorias, las historias de usuario se integran con los otros componentes del sistema. Esta fase de integración asegura que el Submódulo de Aprobación de Usuarios funcione correctamente dentro del sistema completo.

Finalmente, después de la integración y una última ronda de pruebas, las historias de usuario se mueven al estado **Finalizado**. Este estado indica que el submódulo está completamente desarrollado, probado e integrado.

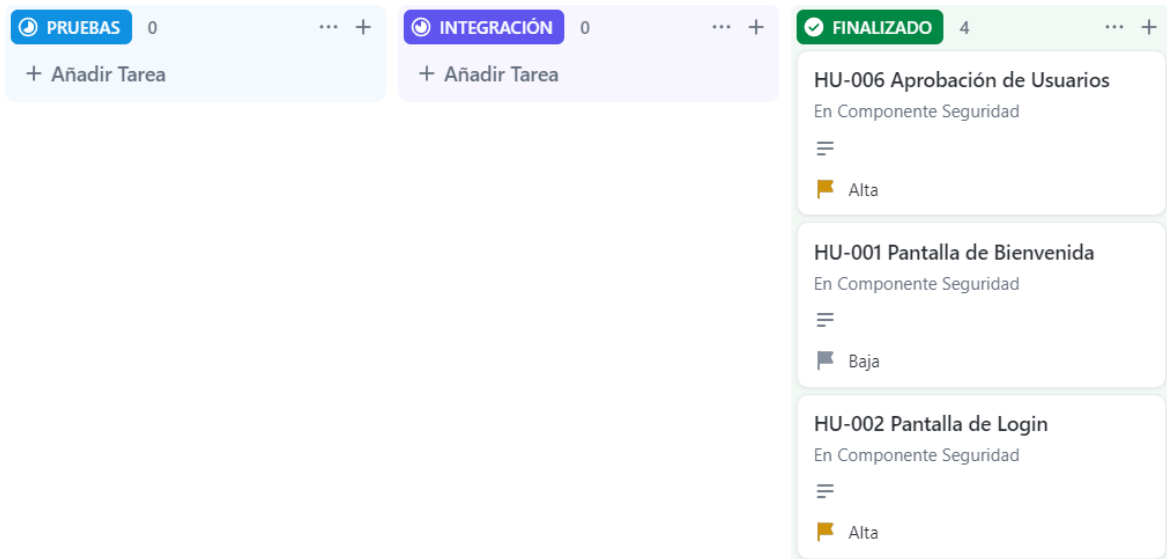


Figura 2.28: Tablero Kanban en estado Finalizado para el Submódulo de aprobación de usuarios.

## Submódulo de Autenticación y Autorización

El Submódulo de Autenticación y Autorización es fundamental para asegurar que solo usuarios verificados y con los permisos adecuados puedan acceder a las funcionalidades del sistema. Este submódulo se centra en implementar mecanismos de autenticación y políticas de autorización para proteger los datos y recursos del sistema.

### Historias de usuario involucradas

- **RNF-002:** Como administrador, quiero limitar los intentos de inicio de sesión y bloquear la IP por 30 minutos después de múltiples intentos fallidos, para prevenir ataques de fuerza bruta.
- **RNF-004:** Como usuario, quiero que la sesión expire automáticamente después de 30 minutos de inactividad, para asegurar la seguridad de mi cuenta.
- **RNF-006:** Como usuario, quiero ver una pantalla cuando no estoy autorizado para acceder a ciertos contenidos, para saber que no tengo permisos suficientes.

Este submódulo se inicia con la siguiente configuración en el tablero Kanban, en el estado Pendiente:

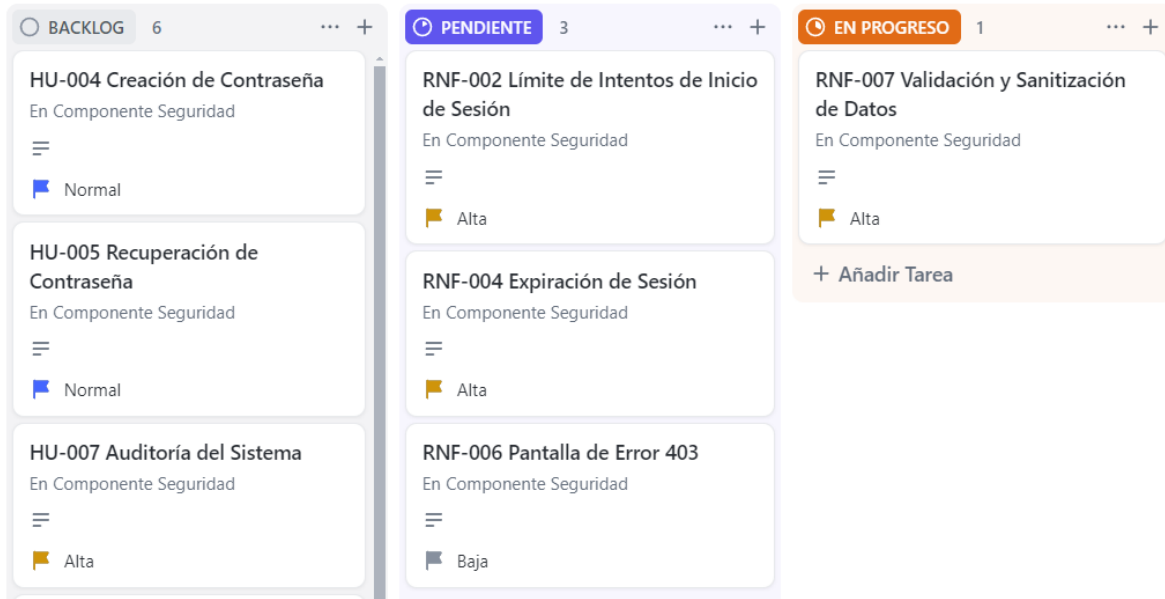


Figura 2.29: Tablero Kanban en estado Pendiente para el Submódulo de autenticación y autorización.

Una vez que las historias de usuario han sido priorizadas, se mueven al estado En Progreso. Para este submódulo, se desarrolló y completó el microservicio AuthService.

## Microservicio

- **AuthService:** Este microservicio ahora se encuentra completo y es responsable de manejar la autenticación de los usuarios. Utiliza JWT (JSON Web Tokens) para gestionar las sesiones de usuario de manera segura y eficiente.

Para asegurar que solo los usuarios autenticados puedan acceder a las diferentes partes del sistema, se realizaron configuraciones específicas tanto en el frontend como en el backend:

**Backend:** En el backend, se crearon políticas de autorización que se aplicaron a los controladores y, en algunos casos, a funciones específicas dentro de los controladores. Estas políticas aseguran que solo los usuarios con los permisos necesarios puedan ejecutar ciertas acciones.

```
// Configurar autorización basada en roles
builder.Services.AddAuthorization(options =>
{
    options.AddPolicy(name: "AdminPolicy", configurePolicy: policy => policy.RequireRole("Administrador"));
    options.AddPolicy(name: "InstructorPolicy", configurePolicy: policy => policy.RequireRole("Instructor"));
    options.AddPolicy(name: "EstudiantePolicy", configurePolicy: policy => policy.RequireRole("Estudiante"));
});
```

Figura 2.30: Políticas de autorización en Backend.

**Frontend:** En el frontend, se configuró cada pantalla para que requiriera autenticación antes de permitir el acceso. Esto se logró mediante el uso de guardias de ruta y otros mecanismos de seguridad que verifican el estado de autenticación del usuario antes de cargar la página.

```
@page "/crear-curso"
@layout AdminLayout
@attribute [Authorize(Roles = "Administrador")]
@using Microsoft.AspNetCore.Authorization
@inject HttpClient Http
@inject NavigationManager NavigationManager
```

Figura 2.31: Atributo de autorización en Frontend.

```
<CascadingAuthenticationState>
  <Router AppAssembly="@typeof(App).Assembly">
    <Found Context="routeData">
      <AuthorizeRouteView RouteData="@routeData" DefaultLayout="@typeof(MainLayout)" />
      <FocusOnNavigate RouteData="@routeData" Selector="h1" />
    </Found>
    <NotFound>
      <LayoutView Layout="@typeof(MainLayout)">
        <p>Sorry, there's nothing at this address.</p>
      </LayoutView>
    </NotFound>
  </Router>
</CascadingAuthenticationState>
```

Figura 2.32: Configurar la autorización en Frontend.

Cuando las credenciales ingresadas son incorrectas, el sistema muestra mensajes de

error claros y específicos para guiar al usuario.

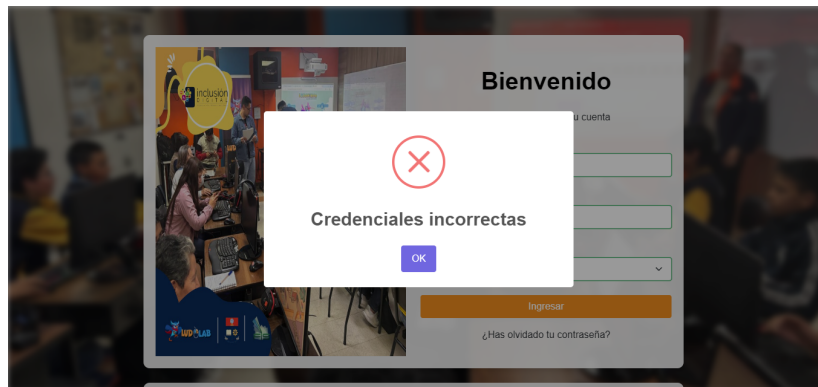


Figura 2.33: Submodulo autenticación y autorización - Pantalla credenciales incorrectas.

Además, si un usuario intenta acceder a una parte del sistema para la cual no tiene autorización, se muestra una pantalla informando sobre la falta de permisos.

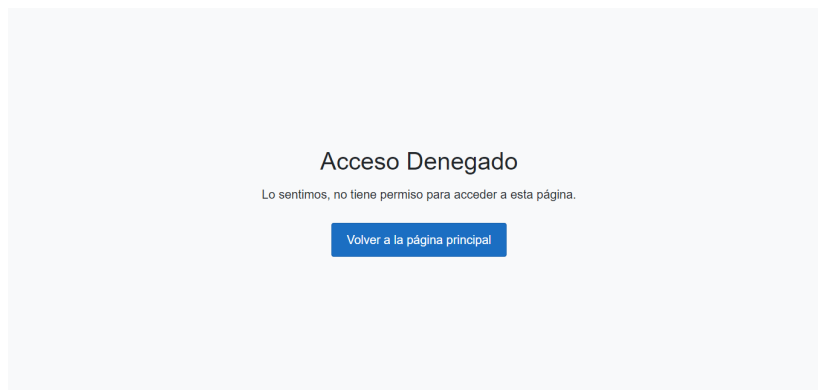


Figura 2.34: Submodulo autenticación y autorización - Pantalla no Autorizado.

Finalmente, después de la integración y una última ronda de pruebas, las historias de usuario se mueven al estado Finalizado. Este estado indica que el submódulo está completamente desarrollado, probado e integrado, y está listo para ser utilizado en el entorno de producción.

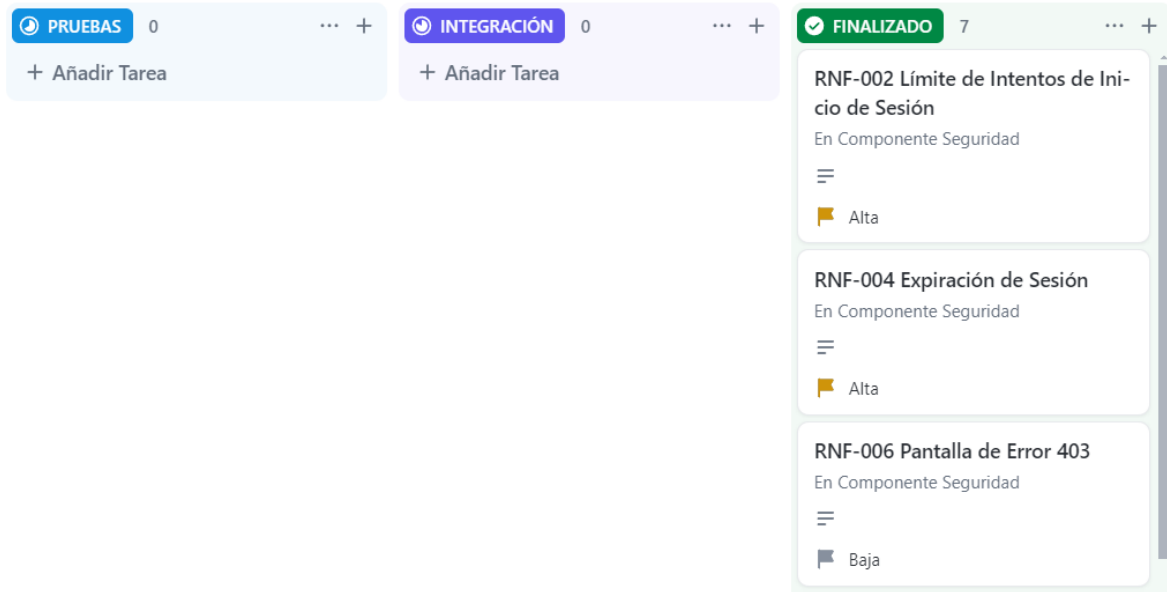


Figura 2.35: Tablero Kanban en estado Finalizado para el Submódulo de autenticación y autorización.

## Submódulo de Gestión de Contraseñas

El Submódulo de Gestión de Contraseñas es crucial para asegurar que las contraseñas de los usuarios sean robustas y cumplan con los estándares de seguridad establecidos. Este submódulo se encarga de la creación, recuperación y validación de contraseñas, asegurando que los usuarios puedan gestionar sus credenciales de manera segura.

### Historias de usuario involucradas

- **HU-004:** Como usuario, quiero crear una nueva contraseña en mi primer inicio de sesión, con recomendaciones de seguridad, para asegurar que mi cuenta esté protegida.
- **HU-005:** Como usuario, quiero poder recuperar mi contraseña en caso de olvido, para poder acceder nuevamente al sistema.
- **RNF-001:** Las contraseñas deben ser hasheadas usando un algoritmo de hashing seguro en la base de datos, para asegurar que las contraseñas de los usuarios estén protegidas.

Este submódulo se inicia con la siguiente configuración en el tablero Kanban, en el



estado Pendiente:

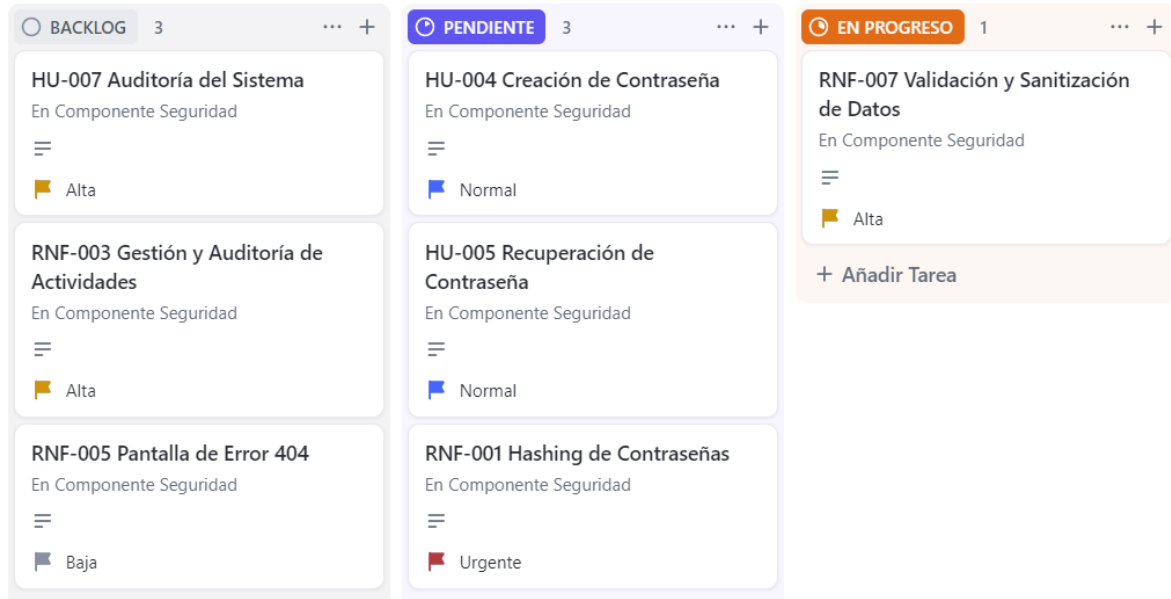


Figura 2.36: Tablero Kanban en estado Pendiente para el Submódulo de gestión de contraseñas.

Una vez que las historias de usuario han sido priorizadas, se mueven al estado En Progreso. Para este submódulo, se desarrolló el microservicio PasswordService.

## Microservicios

- **PasswordService:** Este microservicio se encarga de generar y almacenar las contraseñas siguiendo las políticas establecidas.

Las políticas de contraseña implementadas son las siguientes:

1. **Longitud mínima:** Debe tener al menos 8 caracteres.
2. **Mayúsculas:** Debe contener al menos una letra mayúscula.
3. **Minúsculas:** Debe contener al menos una letra minúscula.
4. **Números:** Debe contener al menos un número.
5. **Caracteres especiales:** Debe usar al menos un carácter especial.

Estas políticas aseguran que las contraseñas sean suficientemente complejas y difíciles de adivinar, protegiendo así las cuentas de los usuarios contra ataques de fuerza bruta y otras amenazas.

A continuación, se muestra una pantalla del sistema donde los usuarios pueden establecer una nueva contraseña cumpliendo con las políticas de seguridad mencionadas.

**Generar nueva contraseña**

Como es la primera vez usando el sistema o está cambiando su contraseña, deberá generar una contraseña exclusiva para su uso dentro del sistema.

Nueva contraseña

\*\*\*\*\*

La contraseña debe tener al menos 8 caracteres.  
La contraseña debe contener al menos una letra mayúscula, una letra minúscula, un número y un carácter especial.

Confirmar nueva contraseña

\*\*\*\*\*

Las contraseñas no coinciden.

Mostrar contraseñas

Políticas para la contraseña:

- Debe tener al menos 8 caracteres
- Debe contener al menos una letra mayúscula
- Debe contener al menos una letra minúscula
- Debe contener al menos un número
- Debe contener al menos un carácter especial (e.g., @, #, \$, %)

Enviar

Figura 2.37: Pantalla para generar una nueva contraseña.

## Submódulo de Auditoría

El Submódulo de Auditoría de Seguridad es esencial para monitorear y registrar las actividades dentro del sistema, asegurando que todas las acciones críticas sean documentadas para futuras revisiones y auditorías. Este submódulo permite a los administradores ver un historial detallado de las operaciones realizadas, proporcionando transparencia y responsabilidad en la gestión del sistema.

### Historias de usuario involucradas

- **HU-007:** Como administrador, quiero ver la auditoría del sistema, para monitorear todas las actividades importantes dentro del sistema.
- **RNF-003:** Como administrador, quiero gestionar y auditar todas las actividades del sistema, para asegurar que se puedan detectar y responder rápidamente a incidentes de seguridad.

Este submódulo se inicia con la siguiente configuración en el tablero Kanban, en el estado Pendiente:

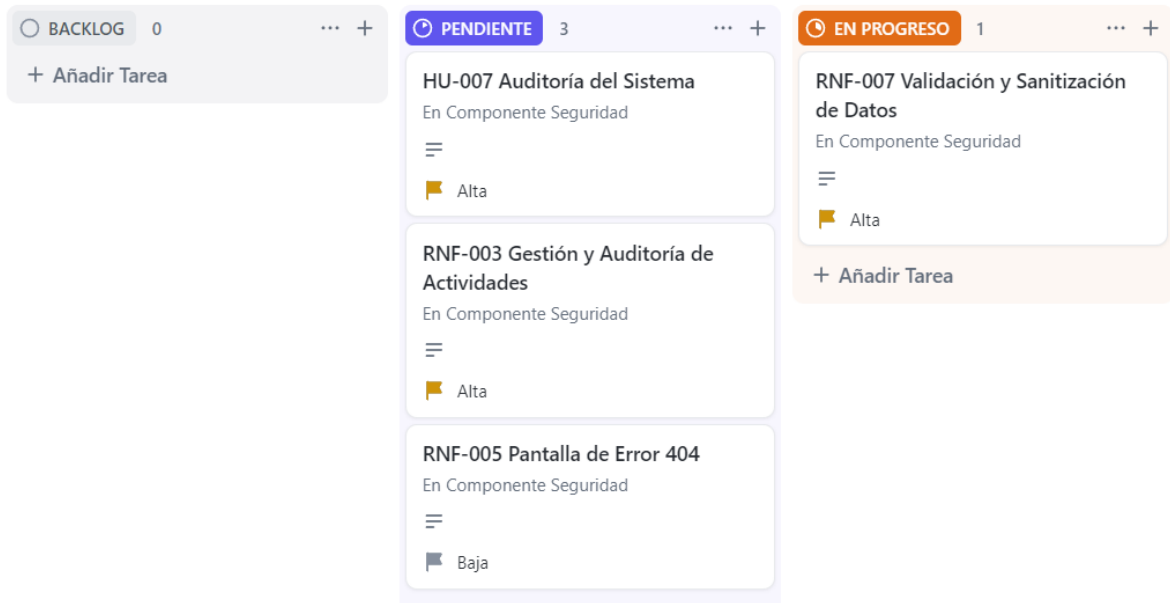


Figura 2.38: Tablero Kanban en estado Pendiente para el Submódulo de auditoría.

Una vez que las historias de usuario han sido priorizadas, se mueven al estado En Progreso. Para este submódulo, se desarrolló el microservicio AuditService.

## Microservicios

- **AuditService:** Este microservicio actúa como un interceptor para registrar las operaciones crear, actualizar y eliminar realizadas en el sistema. Cada vez que se realiza una operación, se registra un log con detalles específicos sobre la acción realizada.

Se desarrolló una interfaz en la pestaña de administrador que muestra una tabla detallada de las actividades auditadas. Los datos registrados en esta tabla incluyen:

- **ID:** Identificador único del registro de auditoría.
- **Fecha y Hora:** Marca temporal de cuándo se realizó la acción.
- **Acción:** Tipo de operación realizada (Crear, Actualizar, Eliminar).
- **ID Usuario:** Identificador del usuario que realizó la acción.
- **Nombre Usuario:** Nombre del usuario que realizó la acción.
- **Nombre Tabla:** Nombre de la tabla afectada por la acción.
- **Claves Primarias:** Claves primarias de los registros afectados.

- **Valores Antiguos:** Valores de los registros antes de la acción.
- **Valores Nuevos:** Valores de los registros después de la acción.

Registros de Auditoría

Id	Timestamp	Acción	Id Usuario	Nombre Usuario	Nombre Tabla	Claves Primarias	Valores Antiguos	Valores Nuevos
1	22/7/2024 11:47:30	Added	1	admin	Curso	{"id_curso":- 2147482647 }		{"id_curso":- 2147482647, "Conjunto_h oras_id_conj unto_horas": 0,"Modalidad _id_modalid ad":2,"horas" :50,"nombre _curso": "Pro gramacion Javascript", "t ipo_curso": " Programacio n"}
2	22/7/2024 11:48:38	Added	1	admin	Curso	{"id_curso":- 2147482646		{"id_curso":- 2147482646,

Figura 2.39: Pantalla de auditoría del sistema.

Para que dicho interceptor funcione fue necesario crear una clase dentro del backend general Figura 2.10, y las funciones más relevantes del código son SavingChanges, SavingChangesAsync y LogChangesAsync. Estas funciones interceptan las operaciones de guardado de Entity Framework, registran los cambios en las entidades y envían estos registros a un servicio externo. El método SavingChanges es sincrónico, mientras que SavingChangesAsync es asíncrono, ambos llamando a LogChangesAsync para realizar el registro de auditoría. LogChangesAsync recopila las entradas de auditoría y envía las solicitudes HTTPS necesarias para registrar los cambios.

```

public override InterceptionResult<int> SavingChanges(DbContextEventData eventData, InterceptionResult<int> result)
{
    LogChangesAsync(eventData.Context).GetAwaiter().GetResult();
    return base.SavingChanges(eventData, result);
}

public override async ValueTask<InterceptionResult<int>> SavingChangesAsync(DbContextEventData eventData,
    InterceptionResult<int> result, CancellationToken cancellationToken = default)
{
    await LogChangesAsync(eventData.Context);
    return await base.SavingChangesAsync(eventData, result, cancellationToken);
}

2 usages
private async Task LogChangesAsync(DbContext context)
{
    if (context == null) return;

    var auditEntries :List<AuditLog> = context.ChangeTracker.Entries()
        .Where(e :EntityEntry => e.State == EntityState.Added || e.State == EntityState.Modified ||
            e.State == EntityState.Deleted) // IEnumerable<EntityEntry>
        .Select(e :EntityEntry => CreateAuditEntry(e)) // IEnumerable<AuditLog>
        .ToList();

    foreach (var auditEntry :AuditLog in auditEntries)
    {
        var auditEntryJson :string = JsonConvert.SerializeObject(auditEntry, _jsonSettings);
        _logger.LogInformation($"Sending audit entry: {auditEntryJson}");

        var request = new HttpRequestMessage(HttpMethod.Post, requestUri: "https://localhost:7500/audit")
        {
            Content = JsonConvert.Create(auditEntry)
        }
    }
}

```

Figura 2.40: Código relevante de AuditService.

Después de completar el desarrollo del microservicio AuditService y la interfaz de usuario para la auditoría, se pasó por el resto de las etapas del proceso Kanban. Esto incluyó la fase de pruebas, donde se realizaron verificaciones exhaustivas para asegurar la correcta funcionalidad y exactitud de los registros de auditoría. Posteriormente, se integró el submódulo con los demás componentes del sistema para garantizar una experiencia coherente y completa. Finalmente, las historias de usuario se movieron al estado Finalizado.

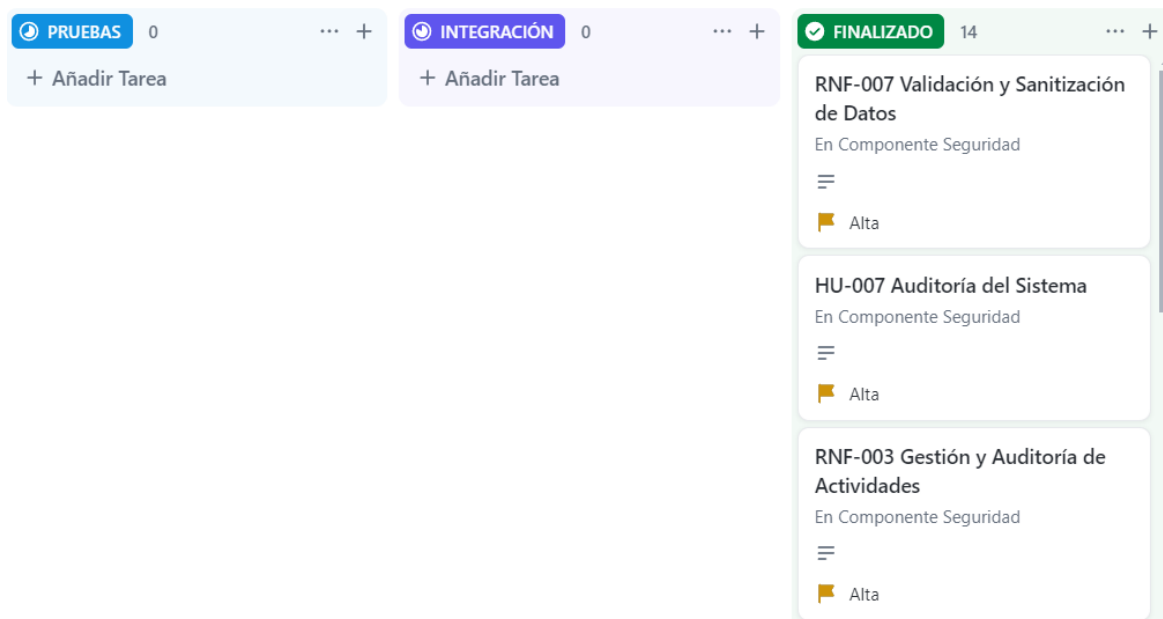


Figura 2.41: Tablero Kanban en estado Finalizado para el Submódulo de auditoría.

Además, en este último desarrollo se completaron dos historias de usuario adicionales: RNF-005, que proporciona una pantalla cuando el usuario intenta acceder a un path que no existe, y RNF-007, que asegura la validación y sanitización de todos los datos de entrada para prevenir inyecciones y otros ataques.

## Capítulo 3

# EVALUACIÓN Y RESULTADOS

En este capítulo se presentan los resultados obtenidos tras la implementación del sistema, haciendo énfasis en la evaluación de los diferentes componentes desarrollados. Se analizarán los endpoints generados, así como los resultados de las pruebas unitarias y las pruebas de penetración realizadas para asegurar la robustez y seguridad del sistema.

### 3.1. Endpoints

Los endpoints son las puertas de acceso a las funcionalidades del sistema a través de las API. La correcta implementación y documentación de estos endpoints es crucial para asegurar que el sistema sea accesible y utilizable por los desarrolladores y usuarios finales. A continuación, se presenta una tabla con la lista de todos los endpoints generados y una breve descripción de cada uno.

Tabla 3.1: Endpoints

Microservicio	Método HTTP	Endpoint	Descripción
AuthService	POST	/auth/login	Ingresa la cédula, contraseña y rol del usuario. Devuelve un JWT que contiene el ID del usuario, nombre del usuario, rol del usuario y tiempo de expiración.
PasswordService	GET	/password/lastlogin/{userId}	Consulta la fecha del último inicio de sesión del usuario. Útil para la funcionalidad de creación de una nueva contraseña.
PasswordService	POST	/password/changepassword	Crea una nueva contraseña y la hashea para almacenarla en la base de datos.
UserService	GET	/user/	Lista todos los usuarios de la base de datos.
UserService	GET	/user/{id}	Lista un único usuario basado en su ID.
UserService	POST	/user/	Crea un nuevo usuario.
UserService	PUT	/user/{id}	Actualiza la información de un usuario basado en su ID.
UserService	DELETE	/user/{id}	Elimina un usuario basado en su ID.

*Continúa en la siguiente página*



Microservicio	Método HTTP	Endpoint	Descripción
AccountService	POST	/account/approve	Aprueba el registro de la cuenta del usuario y envía un correo electrónico de aprobación junto con la cédula y la contraseña temporal.
AccountService	POST	/account/deny	Niega el registro de la cuenta del usuario y envía un correo electrónico de negación.
AuditService	POST	/audit/	Crea una nueva auditoría.
AuditService	GET	/audit/	Lista todas las auditorías.
AuditService	GET	/audit/{id}	Lista una única auditoría basada en su ID.
RegisterService	POST	/register/	Crea un nuevo usuario con todos sus datos, tales como etnia, datos de EPN, dirección, entre otros.

Este conjunto de endpoints se desarrolló gracias a la creación de seis microservicios, cada uno encargado de manejar una parte específica de la funcionalidad del sistema. Estos microservicios son AuthService, PasswordService, RegisterService, UserService, AccountService y AuditService.

### 3.2. Pruebas de funcionalidad

Las pruebas de funcionalidad son esenciales para garantizar que todos los componentes del sistema funcionen correctamente según lo especificado. En este proyecto, se desarrollaron varios microservicios, cada uno con sus propios endpoints para manejar diversas



dula, contraseña y rol. Al ingresar la información correcta, el sistema devuelve un JSON Web Token (JWT) que contiene el ID del usuario, nombre, rol y tiempo de expiración. Esta respuesta confirma que el proceso de autenticación se ha realizado correctamente y que el usuario tiene acceso al sistema.

## PasswordService

**Endpoint** GET /password/lastlogin/{userId}



Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7027/password/lastlogin/1' \
  -H 'accept: */*'
```

Request URL

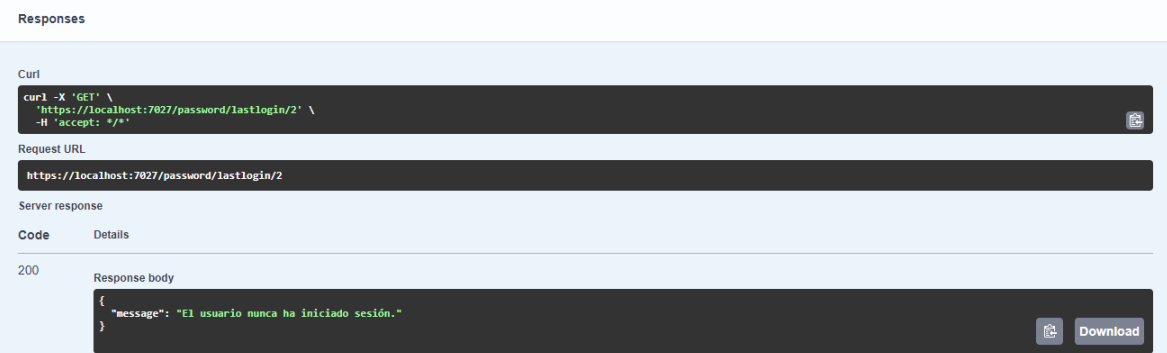
```
https://localhost:7027/password/lastlogin/1
```

Server response

Code	Details
200	Response body

```
{
  "lastLogin": "2024-07-25T14:31:59.16"
}
```

Figura 3.3: Prueba GET /password/lastlogin/{userId} incluye la fecha y hora del último inicio de sesión.



Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7027/password/lastlogin/2' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7027/password/lastlogin/2
```

Server response

Code	Details
200	Response body

```
{
  "message": "El usuario nunca ha iniciado sesión."
}
```

Figura 3.4: Prueba GET /password/lastlogin/{userId} cuando no tiene fecha y hora del último inicio de sesión.

Este endpoint permite consultar la fecha del último inicio de sesión del usuario, lo cual es útil para determinar si el usuario necesita crear una nueva contraseña. La respuesta incluye la fecha y hora del último inicio de sesión, permitiendo al sistema notificar al usuario si es necesario actualizar su contraseña.

## Endpoint POST /password/changepassword

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7027/password/changepassword?userId=2' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{"password": "123"}'
```

Request URL

```
https://localhost:7027/password/changepassword?userId=2
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "message": "Contraseña cambiada exitosamente." }</pre>

Figura 3.5: Prueba POST /password/changepassword exitosa.

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7027/password/changepassword?userId=2' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{"string"}'
```

Request URL

```
https://localhost:7027/password/changepassword?userId=2
```

Server response

Code	Details
400	<p><i>Undocumented</i> Error: response status is 400</p> <p>Response body</p> <pre>{   "error": "La contraseña debe tener al menos 8 caracteres. La contraseña debe contener al menos una letra mayúscula. La contraseña debe contener al menos un número. La contraseña debe contener al menos un carácter especial." }</pre>

Figura 3.6: Prueba POST /password/changepassword no exitosa.

Este endpoint se utiliza para crear una nueva contraseña para el usuario. La contraseña se hashlea antes de almacenarse en la base de datos, asegurando que se mantenga segura. La respuesta del endpoint confirma que la contraseña ha sido actualizada exitosamente.

## AccountService

### Endpoint POST /account/approve

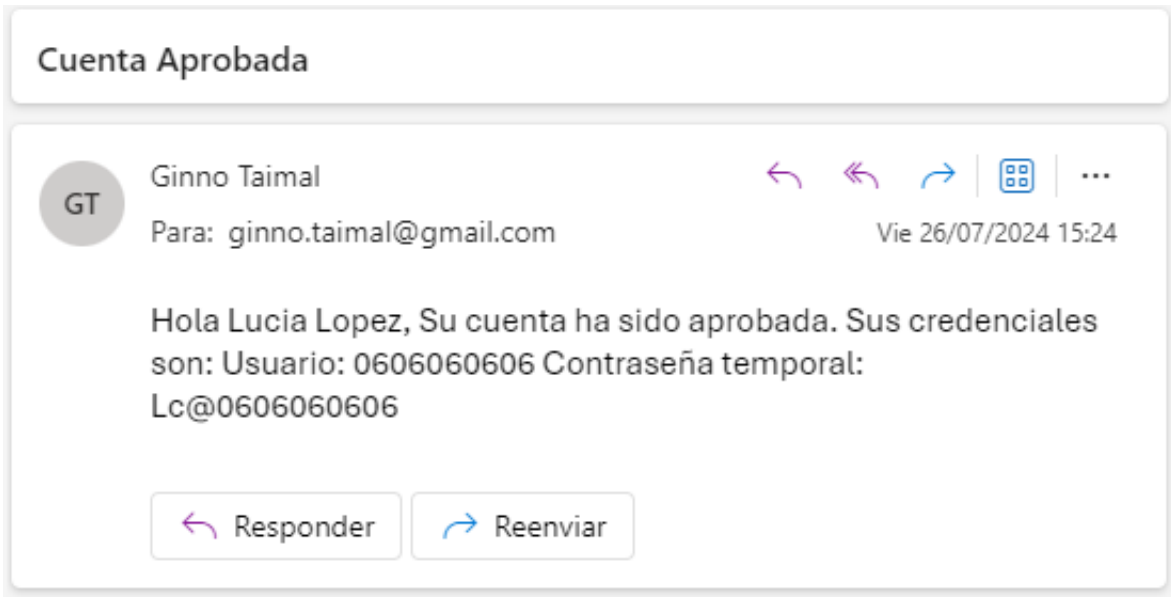


Figura 3.7: Prueba POST /account/approve.

Este endpoint aprueba el registro de la cuenta del usuario y envía un correo electrónico de aprobación junto con la cédula y la contraseña temporal. La respuesta confirma que el usuario ha sido aprobado y que el correo ha sido enviado.

**Endpoint** POST /account/deny

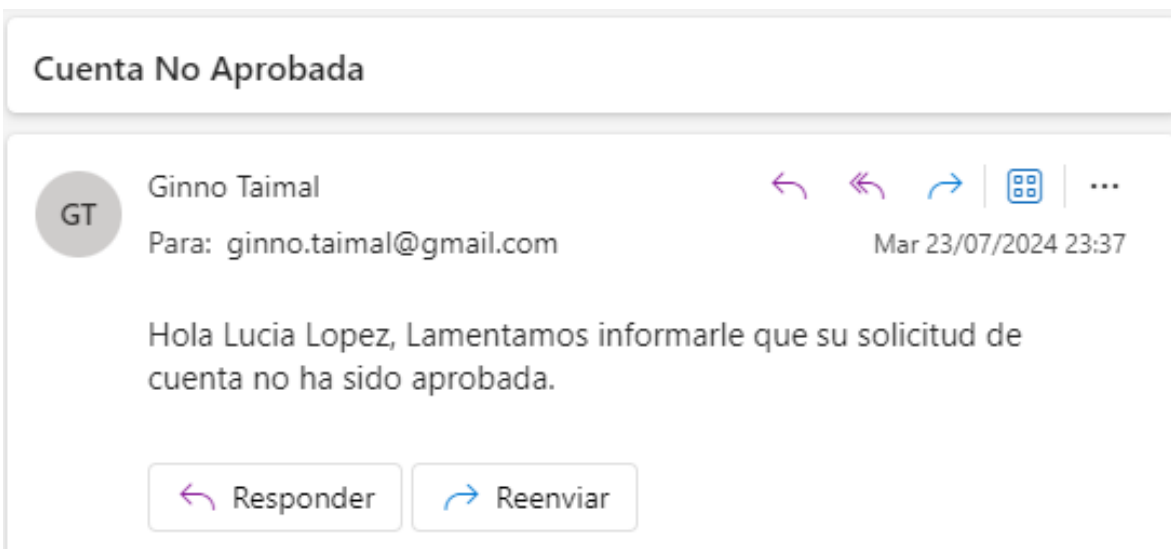


Figura 3.8: Prueba POST /account/deny.

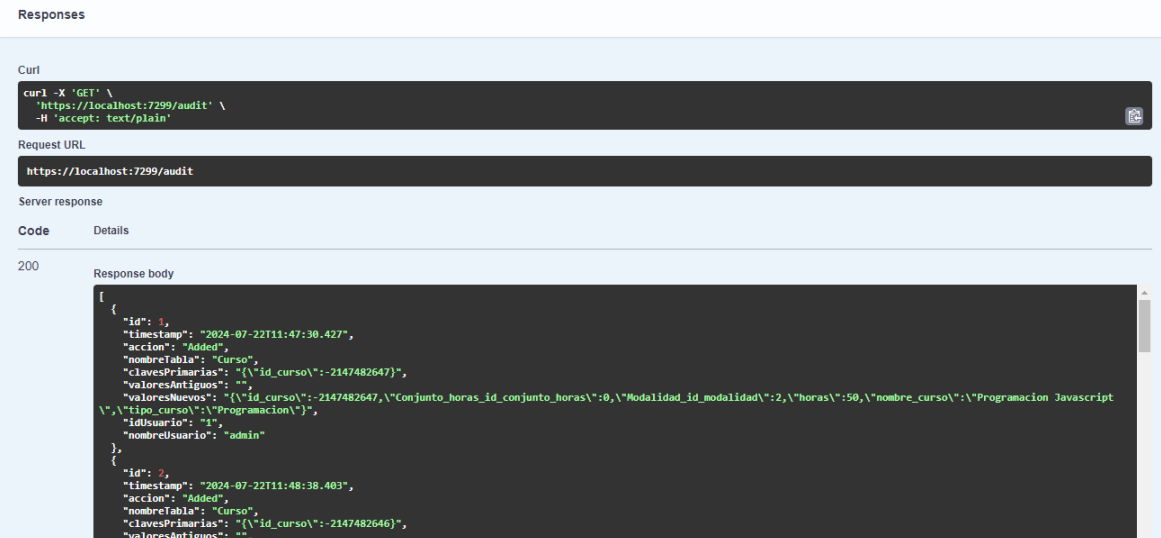
Este endpoint niega el registro de la cuenta del usuario y envía un correo electrónico de negación. La respuesta confirma que el registro ha sido denegado y que el correo ha sido enviado.

## AuditService

### Endpoint POST /audit/

Este endpoint actúa como un interceptor, registrando las operaciones de creación, actualización y eliminación en la base de datos. Las pruebas de este servicio se realizan desde el cliente o desde el backend de los otros componentes, administrativo y automatización de procesos. Cuando se detecta un cambio en la base de datos, el interceptor se activa automáticamente, registrando la operación correspondiente. Además, el cliente debe proporcionar el 'idUsuario' y el 'nombreUsuario' para que se pueda realizar el registro adecuado de la auditoría. Esta funcionalidad asegura que todas las modificaciones importantes en el sistema sean monitorizadas y registradas, proporcionando una capa adicional de seguridad y trazabilidad.

### Endpoint GET /audit/



The screenshot displays a REST client interface with the following details:

- Request:** curl -X 'GET' 'https://localhost:7299/audit/' -H 'accept: text/plain'
- Request URL:** https://localhost:7299/audit
- Server response:** 200
- Response body:** A JSON array of two audit records. The first record has an id of 1, timestamp of "2024-07-22T11:47:30.427", action of "Added", and nombreTabla of "Curso". The second record has an id of 2, timestamp of "2024-07-22T11:48:38.483", action of "Added", and nombreTabla of "Curso".

Figura 3.9: Prueba GET /audit/.

Este endpoint lista todas las auditorías registradas en el sistema, permitiendo a los administradores revisar todas las actividades importantes. La respuesta incluye detalles como la fecha, la acción realizada y el usuario involucrado.

## Endpoint GET /audit/{id}



The screenshot shows a REST client interface with the following details:

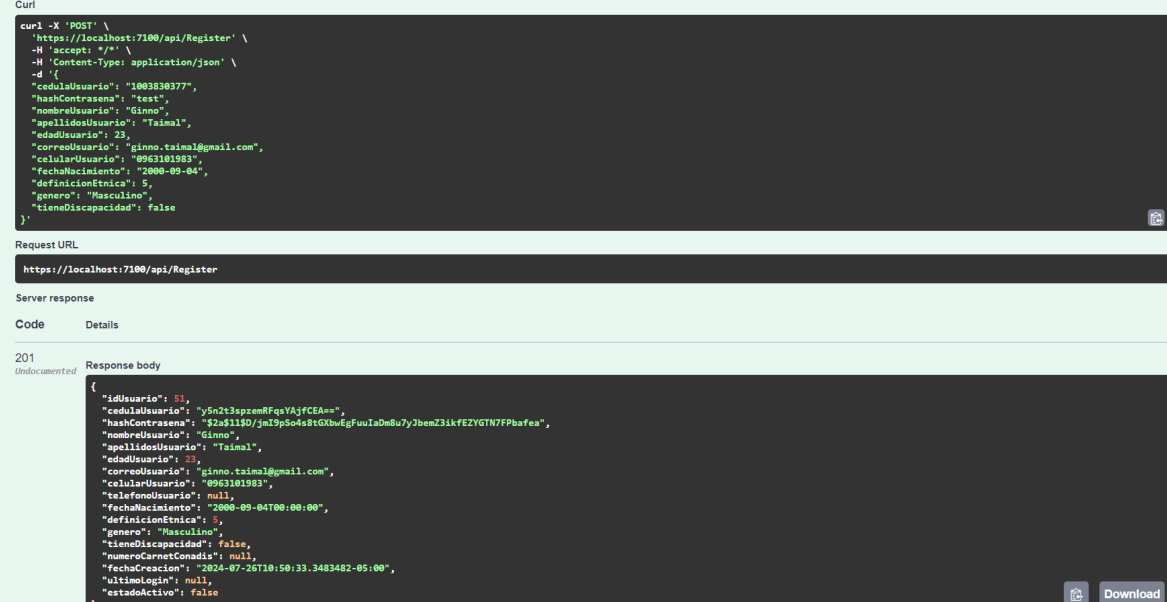
- Request:** curl -X 'GET' \ 'https://localhost:7299/audit/1' \ -H 'accept: text/plain'
- Request URL:** https://localhost:7299/audit/1
- Server response:** Code 200
- Response body:** A JSON object containing audit details for ID 1, including timestamp, action, course name, and user information.

Figura 3.10: Prueba GET /audit/{id}.

Este endpoint permite obtener los detalles de una única auditoría basada en su ID. La respuesta proporciona información específica sobre la auditoría seleccionada.

## RegisterService

### Endpoint POST /register/



The screenshot shows a REST client interface with the following details:

- Request:** curl -X 'POST' \ 'https://localhost:7100/api/Register' \ -H 'accept: \*/\*' \ -H 'Content-Type: application/json' \ -d '{ ... }'
- Request URL:** https://localhost:7100/api/Register
- Server response:** Code 201
- Response body:** A JSON object representing a newly created user with fields like idUsuario, cedulaUsuario, hashContraseña, nombreUsuario, and fechaNacimiento.

Figura 3.11: Prueba POST /register/.

Este endpoint permite crear un nuevo usuario con todos sus datos, tales como etnia,

datos de EPN, dirección, entre otros. Como se puede visualizar en la imagen adjunta, la operación se realiza con éxito. En la imagen, se observa que la cédula aparece encriptada y la contraseña hasheada, dado que son los datos más sensibles de la aplicación. Esta medida asegura que la información crítica esté protegida desde el momento del registro, cumpliendo con las políticas de seguridad establecidas.

## **UserService**

Cada uno de los endpoints del UserService hace referencia a las operaciones CRUD (Create, Read, Update, Delete) en la base de datos de usuarios. Estos endpoints permiten gestionar los usuarios del sistema de manera eficiente, proporcionando funcionalidad para listar todos los usuarios, obtener detalles de un usuario específico, crear nuevos usuarios, actualizar información existente y eliminar usuarios. Debido a la naturaleza estándar de estas operaciones y su funcionamiento bien comprendido, no se consideró necesario incluir imágenes de las pruebas, ya que no se consideraron de mucha relevancia visual.



## Capítulo 4

# CONCLUSIONES Y RECOMENDACIONES

### 4.1. Conclusiones

- El componente de seguridad fue implementado de forma satisfactoria siguiendo las especificaciones de seguridad derivadas del análisis de riesgos. Esta implementación aseguró que las medidas de seguridad fueran adecuadamente integradas en el sistema, protegiendo los datos y la integridad del mismo frente a diversas amenazas identificadas durante el análisis inicial.
- La implementación de submódulos específicos para la autenticación y autorización, así como para la aprobación de usuarios, ha sido fundamental para garantizar que solo los usuarios autorizados puedan acceder a los recursos del sistema. Estos submódulos no solo refuerzan la seguridad al controlar el acceso, sino que también aseguran que las cuentas de usuario sean verificadas y aprobadas de manera adecuada antes de permitir su acceso al sistema. Esta doble capa de seguridad protege la integridad de la aplicación y la confidencialidad de los datos.
- La auditoría ha avanzado significativamente, monitoreando actualmente los cambios realizados en las tablas por los usuarios del sistema. Este submódulo puede seguir mejorando al agregar la capacidad de registrar accesos a la aplicación, intentos de acceso a otras páginas, entre otros eventos relevantes, para alcanzar una supervisión de seguridad aún más exhaustiva.
- La implementación del submódulo de gestión de contraseñas tomando en cuenta las

políticas discutidas con el cliente. Estas políticas aseguran que las contraseñas cumplan con estándares de complejidad adecuados, dificultando los intentos de acceso no autorizado. Además, el uso de técnicas de hashing para almacenar contraseñas garantiza que incluso si los datos son comprometidos, las contraseñas permanezcan protegidas.

- En este caso, se realizaron pruebas de funcionalidad a cada microservicio, evaluando los resultados en base a las consideraciones de seguridad establecidas desde el inicio. Estas pruebas permitieron verificar que cada microservicio opera conforme a los requisitos de seguridad, asegurando la protección de los datos y la robustez del sistema frente a posibles ataques.

## 4.2. Recomendaciones

- Para empezar, se pudieron considerar otros tipos de autenticación en vez de JWT, tales como claves SSH, tokens OAuth, certificados SSL y credenciales tradicionales. Evaluar y seleccionar el método de autenticación adecuado según las necesidades del proyecto puede mejorar la seguridad y eficiencia del sistema.
- Cuando se van a agregar más métodos de autenticación, como fue en nuestro caso con Microsoft SharePoint mediante Azure Active Directory y nuestro microservicio AuthService, es mejor abordarlo con más tiempo y planificación para evitar conflictos. Esto nos sucedió a nosotros, dado que el sistema no sabía qué método usar, y tuvimos que optar por usar uno u otro, pero no ambos en conjunto. Desafortunadamente, nos dimos cuenta de este problema al final del desarrollo, lo que nos impidió solucionarlo adecuadamente.
- Es recomendable, es preferible realizar pruebas de seguridad con frecuencia para detectar nuevos riesgos en el sistema. Implementar pruebas periódicas permite identificar vulnerabilidades emergentes, lo que facilita la aplicación de medidas correctivas de manera oportuna y mejora la resiliencia del sistema frente a posibles ataques.
- Es aconsejable desarrollar un módulo para manejar el perfil del usuario. Esta medida facilita el registro interno por parte del administrador, permitiéndole centrarse en los datos más relevantes, mientras que el usuario completa el resto de la información. Este enfoque no solo mejora la experiencia del usuario, sino que también aumenta la eficiencia del proceso de registro.

- Finalmente, es crucial conocer más a profundidad el sistema para identificar componentes que se pueden desarrollar de manera más efectiva. En nuestro caso, faltó dividir mejor los componentes, lo que resultó en situaciones más complejas o en la necesidad de crear módulos no planeados originalmente. Una comprensión exhaustiva del sistema desde el principio ayuda a evitar estos problemas, permitiendo diseñar soluciones más efectivas y coherentes.

# REFERENCIAS BIBLIOGRÁFICAS

- [1] EPN. «La EPN graduó a 164 beneficiarios del Proyecto de Inclusión Digital Democratización de Accesos a las TICS.» (10 de mar. de 2023), dirección: <https://www.epn.edu.ec/graduacion-proyecto-inclusion/>.
- [2] Y. Niño Benitez y N. Silega Martínez, «Requisitos de Seguridad para aplicaciones web.,» *Revista Cubana de Ciencias Informáticas*, vol. 12, págs. 205-221, 2018.
- [3] P. Magazine, *EdTech: la nueva forma de aprender y enseñar en el mundo*, P. Magazine, ed., 1 de jun. de 2023. dirección: [https://issuu.com/promagmty/docs/pro\\_magazine\\_ed.\\_84\\_edtech/s/25631075](https://issuu.com/promagmty/docs/pro_magazine_ed._84_edtech/s/25631075).
- [4] C. T. Duque. «SIS y LMS, ¿Cuál es el mejor Sistema de Información Estudiantil?» *eLearnMagazine*, ed. (1 de mayo de 2019), dirección: <https://www.elearnmagazine.com/espanol/sis-y-lms-cual-es-el-mejor-sistema-de-informacion-estudiantil/>.
- [5] Ellucian. «¿Qué es un Sistema de Información Estudiantil para Educación Superior?» (2024), dirección: <https://www.ellucian.com/es/ideas/que-es-un-sistema-de-informacion-estudiantil-para-educacion-superior>.
- [6] C. Cilleruelo. «OWASP: Protege tus aplicaciones web con las mejores prácticas.» *KeepCoding*, ed. (19 de abr. de 2024), dirección: <https://keepcoding.io/ciberseguridad/owasp-protege-tus-aplicaciones-web-con-las-mejores-practicas/#:~:text=OWASP%20es%20una%20organizaci%C3%B3n%20dedicada,functiones%20de%20seguridad%20en%20aplicaciones..>
- [7] D. Herrera. «Seguridad en aplicaciones web: Qué es, cómo funciona y los mejores servicios.» *Hostinger*, ed. (13 de feb. de 2023), dirección: <https://www.hostinger.es/tutoriales/seguridad-en-aplicaciones-web>.

- [8] J. Santos. «Web Application Security: ¿Qué es y cómo pueden mejorarla las pymes y startups?» (14 de mayo de 2024), dirección: <https://www.deltaprotect.com/blog/que-es-web-application-security>.
- [9] R. Toro-García, «Evaluación de riesgos de seguridad y planificación de la actividad preventiva en una empresa de desarrollo de Software,» Tesis de mtría., 2021.
- [10] P. V. Chicaiza Chabla, «Evaluación de riesgos de seguridad de la información y generación del plan de gestión de incidentes. Caso de estudio fondo para la protección del agua (FONAG).,» Tesis de mtría., Quito: EPN, 2023., 2023.
- [11] G. G. V. Herrera y J. A. H. Quispe, «Un Marco de Trabajo para el Desarrollo de Software Web Seguro con Metodologías Ágiles,» *Revista peruana de computación y sistemas*, vol. 6, n.º 1, págs. 47-60, 2024.
- [12] A. F. Melo Hernández, «Implementación del modelado de amenazas para la optimización de procesos de aseguramiento de aplicaciones,» Tesis doct., Universidad Santo Tomás.
- [13] A. Ghosh. «¿Qué es la gestión de identidades y accesos (IAM)?» (13 de mayo de 2024), dirección: <https://blog.scalefusion.com/es/what-is-identity-and-access-management/>.
- [14] OSTECH-Push. «¿Qué es la gestión de identidades y accesos (IAM)?» (21 de dic. de 2022), dirección: <https://ostec.blog/es/seguridad/que-es-la-gestion-de-identidades-y-accesos-iam/>.
- [15] Y. L. Erika Dayana y A. G. Kleber Fabián, «Aplicación de la metodología kanban en el desarrollo del software para generación, validación y actualización de reactivos, integrado al sistema informático de control académico UNACH,» B.S. thesis, Universidad Nacional de Chimborazo, 2020, 2020.
- [16] P. E. Colla, «Uso de opciones reales para evaluar la contribución de metodologías KANBAN en desarrollo de software,» en *Simposio Argentino de Ingeniería de Software (ASSE 2016)-JAIIO 45 (Tres de Febrero, 2016).*, 2016.
- [17] W. Zayat y O. Senvar, «Framework study for agile software development via scrum and Kanban,» *International journal of innovation and technology management*, vol. 17, n.º 04, pág. 2 030 002, 2020.

- [18] M. C. Anzules y E. J. G. Moya, «Kanban: Una metodología ágil para la gestión eficiente del flujo de trabajo en el desarrollo de software, una revisión sistemática,» *Revista Ingenio global*, vol. 3, n.º 1, págs. 17-28, 2024.
- [19] D. Fucci, E. Alégroth, M. Felderer y C. Johannesson, «Evaluating software security maturity using OWASP SAMM: Different approaches and stakeholders perceptions,» *Journal of Systems and Software*, vol. 214, pág. 112 062, 2024.
- [20] L. Williams, «Secure software lifecycle,» *The Cyber Security Body of Knowledge*, 2019.
- [21] N. Teodoro y C. Serrao, «Web application security: Improving critical web-based applications quality through in-depth security analysis,» en *International Conference on Information Society (i-Society 2011)*, IEEE, 2011, págs. 457-462.
- [22] M. Gutfleisch, M. Schöps, S. A. Horstmann, D. Wichmann y M. A. Sasse, «Security Champions Without Support: Results from a Case Study with OWASP SAMM in a Large-Scale E-Commerce Enterprise,» en *Proceedings of the 2023 European Symposium on Usable Security*, 2023, págs. 260-276.
- [23] E. Carrera i Aguiar, «SecDevOps modeling for web services and applications,» Tesis de mtría., Universitat Politècnica de Catalunya, 2022.
- [24] J. Hellström y A. Moberg, *A Lightweight Secure Development Process for Developers*, 2019.
- [25] Microsoft. «.NET Overview.» (10 de ene. de 2024), dirección: <https://learn.microsoft.com/en-us/dotnet/core/introduction>.
- [26] Microsoft. «Tutorial: Create a web API with ASP.NET Core.» (15 de jul. de 2024), dirección: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-8.0&tabs=visual-studio#overview>.
- [27] Microsoft. «ASP.NET Core Blazor.» (9 de feb. de 2024), dirección: <https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-8.0>.
- [28] JetBrains. «Rider.» (10 de ago. de 2023), dirección: <https://www.jetbrains.com/rider/>.
- [29] Microsoft. «SQL Server 2022.» (18 de jul. de 2024), dirección: <https://learn.microsoft.com/es-es/sql/sql-server/what-s-new-in-sql-server-2022?view=sql-server-ver16>.

- [30] ClickUp. «ClickUp Features.» (), dirección: <https://clickup.com/features>.
- [31] Figma. «What is Figma?» (), dirección: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>.
- [32] Microsoft. «Video Conferencing, Meetings, Calling | Microsoft Teams.» (), dirección: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>.
- [33] Microsoft. «Microsoft Threat Modeling Tool.» (25 de ago. de 2022), dirección: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool>.
- [34] GitHub. «About GitHub.» (2024), dirección: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git#about-github>.
- [35] S. Chacon y B. Straub, *Pro Git*, 2nd. Berkeley, CA, USA: Apress, 2023, ISBN: 978-1484200773.
- [36] S. Singh. «Mastering .NET 8 Web API: From Setup to Security — 50 Tips Guide for Developers.» (31 de mar. de 2024), dirección: <https://medium.com/c-sharp-programming/mastering-net-8-web-api-from-setup-to-security-a-comprehensive-guide-for-developers-a72887d04848>.

# ANEXOS

**ANEXO A:** [Historias de usuario](#)

**ANEXO B:** [Análisis de riesgos](#)

**ANEXO C:** [Diseño de base de datos](#)

**ANEXO D:** [Arquitectura](#)

**ANEXO E:** [Reporte del modelo de amenazas](#)

**ANEXO F:** [Prototipo](#)

**ANEXO G:** [GitHub](#)