

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA EN SISTEMAS**

### **DESARROLLO DE UN PROTOTIPO FUNCIONAL PARA LA GESTIÓN DEL SISTEMA DE ARCHIVOS INSTITUCIONALES EPN UTILIZANDO METODOLOGÍAS ÁGILES.**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
CIENCIAS DE LA COMPUTACIÓN**

**MARLON ISRAEL PACHACAMA GRANDA**  
**marlon.pachacama@epn.edu.ec**

**JHOSUE ALEXANDER MORALES CAZAR**  
**jhosue.morales@epn.edu.ec**

**ERICK ALEXANDER VERA TOSCANO**  
**erick.vera@epn.edu.ec**

**JOSE DAVID PALLO MORALES**  
**jose.pallo01@epn.edu.ec**

**DIRECTOR: RODRIGO FABIAN CHANCUSIG CHUQUILLA**  
**rodrigo.chancusig@epn.edu.ec**

**DMQ, febrero 2024**

## **CERTIFICACIONES**

Nosotros, MARLON ISRAEL PACHACAMA GRANDA, JOSE DAVID PALLO MORALES, ERICK ALEXANDER VERA TOSCANO, JHOSUE ALEXANDER MORALES CAZAR declaramos que el trabajo de integración curricular aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

---

**Marlon Israel Pachacama Granda**

---

**José David Pallo Morales**

---

**Erick Alexander Vera Toscano**

---

**Jhosue Alexander Morales Cazar**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante 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.

MARLON ISRAEL PACHACAMA GRANDA

ERICK ALEXANDER VERA TOSCANO

JOSE DAVID PALLO MORALES

JHOSUE ALEXANDER MORALES CAZAR

Yo RODRIGO FABIAN CHANCUSIG CHUQUILLA, certifico que el presente trabajo de integración curricular fue desarrollado por los estudiantes de la Facultad de Sistemas, bajo mi supervisión.

---

**MCs. Rodrigo Chancusig**  
**DIRECTOR**

## **DEDICATORIA**

A la Escuela Politécnica Nacional, cuna de conocimiento y excelencia académica, agradezco profundamente por brindarme las herramientas necesarias para emprender este viaje intelectual. A mi alma mater, fuente inagotable de inspiración, dedicó este trabajo que refleja el compromiso adquirido durante mi formación. A mis profesores y compañeros, gracias por su apoyo y orientación.

## **AGRADECIMIENTO**

A mis queridos abuelos, Gonzalo Pachacama y Mariana Chanchay, cuyos valores y sabiduría han sido la raíz de nuestra familia, les dedico este logro. Su legado de esfuerzo y dedicación ha sido una inspiración constante.

A mis amados padres, Jesús Pachacama y Mariela Granda, quiero expresar mi más profundo agradecimiento por su inquebrantable apoyo, amor incondicional y constante inspiración a lo largo de mi viaje académico. Sus sacrificios y enseñanzas han sido la base sobre la cual he construido mi camino hacia la culminación de esta tesis.

A mi querida hermana Somaly Pachacama, tu aliento y presencia han sido una fuente constante de motivación. Tus palabras alentadoras y tu apoyo inquebrantable han iluminado mi camino en los momentos más desafiantes. Gracias por ser mi amiga, confidente y compañera de vida.

Este logro no solo es mío, sino también de ustedes, quienes han sido pilares fundamentales en mi desarrollo. Agradezco infinitamente el amor, paciencia y comprensión que han brindado en cada paso de este proceso.

Marlon Pachacama.

## **AGRADECIMIENTO**

Quiero agradecer a Dios por darme las fuerzas y los ánimos a seguir este camino el cual está llegando a su final.

A mis abuelos quienes con su ayuda y apoyo incondicional se hicieron presentes a lo largo de esta carrera, su motivación, su lucha y sus interminables fuerzas inspiraron a que yo no me rinda en el proceso.

A mis padres quienes, con sus consejos, su apoyo y su ejemplo me han sabido guiar no solamente en esta carrera, sino en toda mi vida. Son mi inspiración, mi fuerza, mi ayuda y los que siempre van a estar presentes sin importar lo que yo haga.

Jhosue Morales.

## **AGRADECIMIENTO**

A mis amados padres, Samuel Vera y Carmen Toscano, quienes con su amor incondicional y eterno apoyo me guiaron hasta alcanzar todas mis metas propuestas. Gracias por creer en mí, incluso cuando yo mismo dudaba. Su sacrificio y dedicación me han permitido llegar hasta donde estoy hoy.

A mis hermanos y cuñados, Jonathan, Andrés, Katherine, Mercedes y Joseph; quienes con sus palabras de ánimo y apoyo fueron esenciales en los momentos más desafiantes. Gracias por su constante compañía, aliento y por ser siempre un pilar fundamental en mi vida.

Finalmente, a mi familia, que siempre estuvieron al tanto de mí y se mostraron orgullosos de cumplir todas las metas que me propuse en la vida.

Su presencia en mi vida ha sido un regalo invaluable. Agradezco infinitamente todo el amor y apoyo que me han sabido dar a lo largo de mi vida. Sin ustedes no hubiese podido llegar a este punto tan importante de mi vida

Erick Vera

## **AGRADECIMIENTO**

Deseo expresar mi agradecimiento a las personas que estuvieron en este trayecto importante en todos los aspectos de mi vida.

En primer lugar, a mi madre, cuya persona fue la que hizo un gran esfuerzo para que logre llegar a ser el ser humano que soy ahora, teniendo siempre respeto y consideración a cada persona que me rodea.

A mis hermanas y mi padre, que siempre me llenaron de motivación para que este esfuerzo sea para demostrar que soy capaz de lograrlo.

Finalmente, quiero agradecer a mis amigos, que más que eso llegaron a ser las personas con las cuales aprendimos juntos el valor del esfuerzo y trabajo en equipo que se requiere realizar para lograr conseguir grandes éxitos en cada etapa de nuestras vidas.

José Pallo



## ÍNDICE DE CONTENIDO

Certificaciones.....	I
Declaración de autoría .....	II
Dedicatoria .....	III
Agradecimiento .....	IV
Índice de contenido .....	VIII
Índice de figuras .....	X
Índice de tablas .....	XII
Resumen.....	XIII
Abstract.....	XIV
1 Introducción .....	1
1.1 Objetivo general.....	3
1.2 Objetivos específicos .....	3
1.3 Alcance .....	4
1.4 Marco teórico .....	6
2 Metodología .....	10
2.1 Metodología de desarrollo SCRUM.....	10
2.2 Roles.....	11
Artefactos.....	12
Recopilación de requerimientos.....	13
Historias de usuario .....	13
2.3 Diseño de interfaces .....	16
Modulo administrador .....	16
Módulos de gestión documental .....	20
2.4 Diseño de la arquitectura .....	24
Modelo Vista Controlador .....	24
2.5 Herramientas de desarrollo.....	27
2.6 Recolección de datos.....	28
Documentación.....	28
Cartas de confidencialidad y satisfacción .....	29
2.7 Pruebas de seguridad .....	30
SAST.....	30
DAST.....	32
Rendimiento.....	34

3	Resultados y discusión .....	36
3.1	Sprint 0.....	36
	Creación de entorno de desarrollo.....	37
	Instalación de herramientas de Versionamiento .....	37
	Creación de la base de datos .....	37
	Definición de usuarios.....	37
3.2	Sprint 1.....	37
	Creación de usuarios .....	37
	Creación y clasificación de documentos .....	38
3.3	Sprint 2.....	39
	Inicio de sesión: .....	40
	Gestión de documentos .....	40
3.4	Sprint 3.....	41
	Gestión de usuarios.....	41
	Actualización de documentos .....	42
3.5	Sprint 4.....	43
	Actualización de usuarios .....	43
	Búsqueda de documentos .....	44
3.6	Sprint 5 – Pruebas .....	45
3.7	Sprint 6 – Despliegue.....	50
4	Conclusiones y recomendaciones .....	52
4.1	Conclusiones.....	52
4.2	Recomendaciones .....	54
5	Anexos.....	56
	Anexo I. Historias de usuario .....	56
	Anexo II. Diseño de interfaces Figma .....	56
	Anexo III. Macro de Excel .....	57
	Anexo IV. Cartas de confidencialidad de documentos y constancia de evidencia del prototipo .....	57
	Anexo V. Código fuente del Prototipo .....	57
	Anexo VI. Video demostrativo .....	57

## ÍNDICE DE FIGURAS

<b>Figura 2.1.</b> Interfaz inicio de sesión módulo Administrador .....	17
<b>Figura 2.2.</b> Interfaz de usuario módulo Administrador .....	17
<b>Figura 2.3.</b> Formulario agregar usuario .....	18
<b>Figura 2.4.</b> Formulario actualizar usuario .....	19
<b>Figura 2.5.</b> Historial de archivos .....	19
<b>Figura 2.6.</b> Interfaz de inicio de sesión módulo Decanato .....	21
<b>Figura 2.7.</b> Historial de archivos creados módulo Decanato.....	22
<b>Figura 2.8.</b> Formulario nuevo archivo módulo Decanato .....	23
<b>Figura 2.9.</b> Formulario actualizar archivo módulo Decanato.....	24
<b>Figura 2.10.</b> Funcionamiento de patrón MVC .....	25
<b>Figura 2.11.</b> Arquitectura implementada en el sistema de gestión documental...	26
<b>Figura 2.12.</b> Arquitectura de base de datos.....	27
<b>Figura 2.13.</b> Formato de la codificación de documentos .....	29
<b>Figura 2.14.</b> Interfaz de Audit Workbench para la carga del código fuente para el análisis .....	31
<b>Figura 2.15.</b> Detalles del análisis realizado .....	31
<b>Figura 2.16.</b> Interfaz de resultados del análisis SAST en Audit Workbench de Fortify .....	32
<b>Figura 2.17.</b> Interfaz de Fortify WebInspect para la carga del código fuente para el análisis .....	33
<b>Figura 2.18.</b> Interfaz de resultados del análisis DAST en Fortify WebInspect .....	34
<b>Figura 2.19.</b> Escenario definido para la prueba de carga y estabilidad .....	35
<b>Figura 2.20.</b> Interfaz de grabación del script en LoadRunner Profesional .....	36
<b>Figura 3.1.</b> Formulario de registro de usuario.....	38
<b>Figura 3.2.</b> Formulario de registro de nuevos documentos.....	39
<b>Figura 3.3.</b> Formulario de inicio de sesión.....	40
<b>Figura 3.4.</b> Panel de gestión de documentos - Usuario visualizador/editor .....	41
<b>Figura 3.5.</b> Panel de gestión de documentos - Usuario administrador .....	41
<b>Figura 3.6.</b> Panel de gestión de usuarios - Usuario Administrador.....	42
<b>Figura 3.7.</b> Formulario de actualización de documento .....	43
<b>Figura 3.8.</b> Formulario de actualización de usuario .....	44
<b>Figura 3.9.</b> Cuadro de búsqueda de documentos.....	45
<b>Figura 3.10.</b> Resultados de la ejecución de la prueba de carga .....	46

<b>Figura 3.11.</b> Número total de transacciones realizadas.....	46
<b>Figura 3.12.</b> Vulnerabilidades encontradas en el análisis SAST con la herramienta Fortify .....	48
<b>Figura 3.13.</b> Detalle de la violación de privacidad encontrada en el código .....	48
<b>Figura 3.14.</b> Detalle de la violación de privacidad encontrada en el código .....	49
<b>Figura 3.15.</b> Resumen del análisis DAST con Fortify WebInspect .....	49
<b>Figura 3.16.</b> Dashboard de vulnerabilidades encontradas en el análisis DAST ..	50
<b>Figura 3.17.</b> Especificaciones de servidor para despliegue de aplicativo .....	51
<b>Figura 3.18.</b> VPN de la Escuela Politécnica Nacional.....	51
<b>Figura 3.19.</b> Acceso al aplicativo con la IP del servidor y puerto de acceso.....	52

## ÍNDICE DE TABLAS

<b>Tabla 2.1.</b> Conformación de Scrum Team .....	12
<b>Tabla 2.2.</b> Épicas e Historias de usuario para el prototipo de gestión documental .....	15
<b>Tabla 2.3.</b> Herramientas para el desarrollo del prototipo .....	27
<b>Tabla 2.4.</b> Transacciones definidas para medir tiempos .....	35
<b>Tabla 3.1.</b> Tiempo de respuesta promedio de las transacciones .....	47
<b>Tabla 3.2.</b> Resumen de la ejecución del análisis SAST .....	47

## RESUMEN

El objetivo de este proyecto es crear un prototipo funcional para gestionar el sistema de archivos institucionales de la EPN, aplicando metodologías ágiles para optimizar el proceso de desarrollo. Inicialmente, se llevó a cabo un exhaustivo estudio sobre la implementación de prototipos a nivel de sitio web, identificando los beneficios clave que esta metodología aporta al proceso de desarrollo. Este análisis proporcionó una base sólida para la definición de herramientas y recursos a utilizar, abarcando lenguajes de programación, bases de datos y metodologías ágiles específicas para el desarrollo de software.

En la fase subsiguiente, se procedió con el desarrollo del prototipo, donde se definieron con precisión los requisitos necesarios para alcanzar los objetivos establecidos. Este proceso incluyó la creación detallada de mockups que sirvieron como representaciones visuales del diseño previo a la implementación.

La culminación del proyecto involucró pruebas exhaustivas de funcionamiento y seguridad, diseñadas para evaluar el rendimiento del prototipo. Estas pruebas no solo confirmaron la funcionalidad deseada, sino que también garantizaron la robustez y la seguridad del sistema. Con este enfoque integral, el desarrollo del prototipo busca no solo cumplir con los objetivos específicos del proyecto, sino también establecer una base sólida y confiable para futuras iteraciones y desarrollos en el ámbito de la gestión de archivos institucionales.

**PALABRAS CLAVE:** Prototipo funcional, Desarrollo de software, Lenguajes de programación, Metodologías ágiles, Mockups, Pruebas de funcionamiento

## **ABSTRACT**

The objective of this project is to create a functional prototype to manage the EPN's institutional archives system, applying agile methodologies to optimize the development process. Initially, an exhaustive study was carried out on the implementation of prototypes at the website level, identifying the key benefits that this methodology brings to the development process. This analysis provided a solid basis for the definition of tools and resources to be used, covering programming languages, databases, and specific agile methodologies for software development.

In the subsequent phase, we proceeded with the development of the prototype, where the requirements needed to achieve the established objectives were precisely defined. This process included the detailed creation of mockups that served as visual representations of the design prior to implementation.

The culmination of the project involved extensive functional and security testing designed to evaluate the performance of the prototype. These tests not only confirmed the desired functionality, but also ensured the robustness and security of the system. With this comprehensive approach, the development of the prototype aims not only to meet the specific objectives of the project, but also to establish a solid and reliable basis for future iterations and developments in the field of institutional records management.

**KEYWORDS:** Functional prototype, Software development, Programming languages, Agile methodologies, Mockups, Functional testing.

# 1 INTRODUCCIÓN

La EPN es una institución universitaria, de derecho público, autónoma, con personería jurídica, se rige por la Constitución de la República, la Ley Orgánica de Educación Superior y su Reglamento, otras leyes conexas, por su Estatuto, reglamentos, manuales de funciones, instructivos y resoluciones expedidas por sus propios organismos de gobierno y autoridades; tiene como misión formar académicos y profesionales con conciencia ética, solidarios, críticos, capaces de contribuir al bienestar de la comunidad, así como generar, difundir y transmitir el conocimiento científico con responsabilidad social.

Según la Ley del Sistema Nacional de Archivos, la documentación que actualmente existe o que en adelante se produjere en los archivos de todas las Instituciones de los sectores públicos es patrimonio del Estado, por lo que debe ser administrada de acuerdo con criterios preestablecidos que garanticen su eficiente clasificación y conservación.

La complejidad de la información generada y almacenada en el archivo de la EPN requiere una solución integral que garantice la accesibilidad, seguridad y eficiencia en la gestión de archivos. Dado que implementar un sistema que funcione a nivel institucional puede llegar a ser complejo, se ha diseñado un prototipo orientado a la Facultad de Ingeniería en Sistemas (FIS) de la escuela Politécnica Nacional. Este prototipo busca no solo cumplir con las demandas actuales de la facultad, sino también anticiparse a futuras necesidades, proporcionando una plataforma flexible y escalable.

En la FIS existen tres áreas para gestión documental, estas corresponden a Decanato, Subdecanato y Jefatura de Departamento. Cada área maneja sus subáreas donde se administra un tipo específico de documentos.

En el módulo de Decanato, se encuentran dos subáreas distintas: la Secretaría de Decanato y la Secretaría de Titulación, cada una manejando documentos específicos. La Secretaría de Decanato se encarga de gestionar cuatro documentos esenciales relacionados con el alto nivel administrativo y estratégico de la facultad. Estos abarcan documentos como lo son: políticas institucionales, decisiones clave, actas/resoluciones de consejo de facultad, solicitudes de estudiantes que corresponden a formularios de autorización, procesos electorales que contienen actas y documentos orientados a la planificación estratégica de inventarios de bienes de la FIS. Por otro lado, la subárea de Secretaría de Titulación maneja otros cuatro documentos, centrados en procesos relacionados con la titulación de los estudiantes como lo son: designación de revisores calificadoros, expedientes de grado, a su vez contiene documentos relacionados a



sílabos/mallas, comisiones los cuales hacen referencia a prácticas de vinculación y preprofesionales.

Por otro lado, el módulo de Subdecanato se enfoca en documentos más específicos relacionados con la gestión académica y el apoyo a los estudiantes. En el módulo de Subdecanato se manejará las subáreas “Secretaría de Subdecanato” y “Coordinaciones de Carrera”. En la subárea de secretaría de Subdecanato se manejan los tipos documentales: formularios recibidos extensión de matrícula, de créditos, restitución de estudiante regular, control docente, registro de notas y solicitudes de cambios de carrera o de universidad. La subárea de coordinaciones de carrera abarca los tipos documentales: memorandos enviados y recibidos. Todos los documentos manejados por la secretaría de Subdecanato se clasifican por semestre.

Por último, el módulo de Jefatura de Departamento alberga documentos específicos de cada departamento académico dentro de la facultad. En la Jefatura de Departamento existen las subáreas Secretaría de Departamento, Maestrías profesionalizantes, Maestrías de Investigación, Comité Doctoral, Doctorado y Secretaría de Doctorado, en estas subáreas se administra la gestión documental en base a los periodos académicos. Cada subárea maneja tipos documentales únicos. Con respecto a la Secretaría de Departamento, esta maneja los tipos documentales: Memorandos e informes técnicos. En el caso de las subáreas relacionadas a las maestrías, estas manejan los tipos documentales sobre solicitudes de becas, informes sobre becas, memorandos, reportes de planificación académica y exámenes complexivos. La subárea de doctorado maneja los tipos documentales sobre rúbrica y solicitudes de apoyo económico o de matrícula. Finalmente, la secretaría de doctorado maneja los tipos documentales sobre memorandos recibidos, y oficios enviados o recibidos.

Cada uno de estos módulos, se especializa en la gestión de documentos que son cruciales para las responsabilidades específicas de Decanato, Subdecanato y Jefatura de Departamento, asegurando una organización eficiente y un acceso rápido a la información relevante para cada área funcional. Esta diferenciación permite una administración más efectiva de los recursos y una respuesta ágil a las demandas específicas de cada unidad dentro de la facultad, además, la división en subáreas dentro de cada uno de los módulos permite una organización más detallada y eficaz de los documentos correspondientes a cada función específica dentro de la dirección general de la facultad.

Además, para integrar todos estos módulos se implementará un módulo integrador. Este módulo es clave en el desarrollo de este prototipo, dado que, permite el manejo de la

información basada en roles definidos para el uso del sistema. Los tipos de usuarios son los siguientes: administradores, que pueden administrar a los usuarios, crear usuarios, modificar usuarios y asignar roles a los usuarios según lineamientos establecidos por los clientes; los editores se especializan en carga y visualización de documentos, considerando su clasificación según el área y el rol asignado por el administrador del sistema. En este contexto, los editores tienen acceso al listado de documentos registrados, y aunque su capacidad de modificar la información está limitada, esta restricción contribuye a garantizar la trazabilidad en el proceso de carga y modificación de documentos.

Cabe recalcar que la gestión documental debe ser controlada de manera rigurosa, incorporando procesos de digitalización y centralizando los archivos en un único repositorio. Asegurar la seguridad de los documentos desde su origen, implementar esquemas de indexación y facilitar el acceso automatizado a través de portales personalizados son elementos clave para optimizar la eficiencia en el manejo de la información. Este enfoque integral garantiza una gestión documental moderna y efectiva, acorde con los estándares contemporáneos y las necesidades cambiantes de la sociedad.

## **1.1 Objetivo general**

Desarrollar un Prototipo funcional para la gestión del Sistema de Archivos Institucionales FIS utilizando metodologías ágiles.

## **1.2 Objetivos específicos**

### **1. Definir Requisitos del Prototipo**

Desarrollar un conjunto detallado de requisitos para el prototipo del Sistema de Archivos Institucionales orientado a la Facultad de Ingeniería en Sistemas de la Escuela Politécnica Nacional, identificando las funcionalidades clave y los elementos críticos para la gestión eficiente del sistema. Este objetivo se enfocará en la recopilación y documentación precisa de los requisitos, asegurando una comprensión clara de las necesidades del usuario y los objetivos del sistema.

### **2. Implementar el Prototipo con Metodologías Ágiles**

Aplicar metodologías ágiles, como Scrum, para la planificación, diseño e implementación iterativa del prototipo funcional del Sistema de Archivos Institucionales (FIS). Este objetivo se centrará en la rápida adaptación a cambios en los requisitos, la entrega incremental de funcionalidades y la colaboración efectiva entre el equipo de desarrollo y los stakeholders, garantizando un desarrollo ágil y eficiente.

### 3. Realizar Pruebas y Obtener Retroalimentación

Conducir pruebas exhaustivas del prototipo del Sistema de Archivos Institucionales (FIS) para validar su funcionalidad, usabilidad y rendimiento. Obtener retroalimentación de los usuarios y stakeholders, identificando posibles mejoras y ajustes necesarios. Este objetivo se centrará en asegurar la calidad del prototipo y en la incorporación de comentarios para optimizar la experiencia del usuario y cumplir con los requisitos institucionales establecidos.

## 1.3 Alcance

Desarrollar un Prototipo funcional para la gestión del Sistema de Archivos Institucionales de la EPN utilizando metodologías ágiles permitirá identificar los requerimientos técnicos, funcionales y operativos de un Sistema de Gestión Documental Institucional.

La implementación del prototipo consta tres componentes de gestión documental: Decanato, Subdecanato y Jefatura de Departamento para la Facultad de Ingeniería en Sistemas. Con el fin de asegurar un desarrollo integral, se ha asignado a personas específicas para liderar cada módulo, el módulo de Decanato estará a cargo de Marlon Pachacama, el módulo de Subdecanato estará a cargo de Erick Vera y Jhosue Morales en el módulo de Jefatura de Departamento. Estos tres componentes se centran en la administración de documentos físicos, de las áreas antes mencionadas.

Para poder integrar los módulos de gestión documental, se ha definido un módulo de integración el cual estará a cargo de José Pallo. En este módulo se administrarán los niveles de acceso que tendrán los usuarios para acceder al sistema desarrollado, es decir, gracias a este módulo se controlara si un usuario es administrador o visualizador/editor, permitiendo así el acceso a gestión de usuarios o gestión documental, según corresponda. Además, el módulo integrador controlará el área a la que corresponda un usuario registrado, es decir, si el usuario registrado pertenece a Decanato, Subdecanato o Jefatura de Departamento.

Dado que los módulos de Decanato, Subdecanato y Jefatura de Departamento se diferencian únicamente por los documentos que se manejan, se ha dividido las tareas necesarias para el desarrollo entre los participantes para poder diferenciar el trabajo realizado por cada uno de ellos.

El desarrollo del backend, en la que se incluye la creación de las rutas para acceder a las diferentes vistas o formularios del sistema, junto con las funciones necesarias para poder gestionar las operaciones CRUD. Esta actividad estará a cargo de Marlon Pachacama

El desarrollo del frontend del sistema, haciendo uso de handlebars, en la cual se crearán todas las vistas del aplicativo, así como los formularios para ingreso de datos o visualización de la información registrada. Esta actividad estará a cargo de Erick Vera. A estas vistas se accede haciendo uso de las rutas, por lo que se trabaja junto con el backend.

Para controlar los permisos de los usuarios en el módulo de gestión documental en base a los roles definidos en el módulo integrador, José Pallo se encargará de controlar los permisos dentro de la gestión documental para que los administradores únicamente puedan visualizar los documentos cargados en el sistema, al igual que los usuarios visualizadores. Por otra parte, los editores deberán poder agregar o modificar los documentos cargados en el sistema.

Los documentos que se digitalizarán se clasificarán según los lineamientos remitidos por el área de gestión documental de la EPN. La gestión documental y la respectiva codificación estará a cargo de Jhosue Morales. Esta actividad corresponde a la recopilación necesaria para poder crear la codificación que utilizarán los documentos que se almacenen en el sistema desarrollado.

Dado que la FIS maneja documentos de un amplio periodo de tiempo, se ha definido los periodos que se consideraran para la digitalización de los documentos físicos necesarios para la implementación de este prototipo. Los documentos físicos utilizados serán desde el semestre 2018-A hasta el semestre 2023-A.

El desarrollo del prototipo de gestión de documentos físicos tendrá un enfoque WEB, haciendo uso de Node.js con handlebars y bases de datos no relacionales.

Dado que, los documentos manejados son de administración pública, el manejo de estos será con base a los lineamientos remitidos en los documentos “NORMA DE GESTIÓN DOCUMENTAL PARA ENTIDADES DE ADMINISTRACIÓN PÚBLICA” emitido por el gobierno del Ecuador en 2016 y de “REGLA TÉCNICA NACIONAL PARA LA ORGANIZACIÓN Y MANTENIMIENTO DE LOS ARCHIVOS PÚBLICOS” emitido por la Secretaría General de la Presidencia de la República del Ecuador en 2019.

La digitalización de los documentos utilizados se hará siguiendo los lineamientos del documento “NORMA DE DIGITALIZACION DE DOCUMENTOS” emitido por la DINARDAP en 2016.

El Sistema tendrá el enfoque de Open Source, desarrollado con metodologías ágiles lo que facilitará la modularización y corrección de errores de código y de vulnerabilidades de cada uno de los componentes del sistema automatizado de seguimiento a convenios de la EPN.

## 1.4 Marco teórico

Las instituciones públicas, como las universidades, cuentan con documentos físicos que necesitan ser digitalizados y para ello se debe seguir la normativa vigente establecida por el organismo correspondiente del país. El organismo que se encarga de gestionar la normativa para la digitalización de documentos públicos es la Dirección Nacional de Registros Públicos (DINARDAP), esta entidad establece en su norma los artículos sobre los lineamientos que se deben seguir para la digitalización de documentos de entidades públicas [1]. Algunos de los principales artículos de la DINARDAP, que las instituciones públicas deben seguir, establecen:

- **Artículo 4:** Seguir un formato estándar y que sea abierto.
- **Artículo 7:** Los documentos históricos no podrán ser destruidos, incluso cuando hayan sido digitalizados.
- **Artículo 11:** Los involucrados en la digitalización no podrán difundir los contenidos durante la digitalización, toda esta debe ser entregada a la institución propietaria de los documentos.
- **Artículo 19:** Los documentos digitalizados deben tener garantía de autenticidad, además, se debe tener un plan de preservación de estos.
- **Artículo 21:** Se debe realizar el inventario de los documentos digitalizados, con sus especificaciones.
- **Artículo 22:** Se debe clasificar y ordenar los documentos previo a la digitalización.
- **Artículo 46:** La persona encargada de digitalizar debe contar con la presencia de un supervisor, de inicio a fin del proceso.
- **Artículo 47:** El supervisor se encargará de revisar la fidelidad e integridad de los documentos digitalizados respecto a los documentos físicos, generará las actas de apertura y cierre (inicio y fin del proceso) y expedirá el acta de conformidad con la cual se valida que el proceso se cumplió sin irregularidades.

Además, al ser documentos públicos, deben tratarse según la normativa vigente relacionada con la gestión documental. En la "Norma De Gestión Documental Para

Entidades De Administración Pública” se establecen los artículos necesarios para la gestión documental de entidades públicas [2]. Algunos de los artículos relacionados a la gestión documental mencionan:

- **Artículo 2:** La norma es de aplicación obligatoria en todas las instituciones de administración pública.
- **Artículo 11, 12, 13 y 14:** Las instituciones de administración pública deben contar con un sistema institucional, en las que se designa a los responsables de la gestión de archivos o activo, compuesto por:
  - Archivos de gestión o Activo: Controlado por las unidades administrativas, se genera los expedientes de uso cotidiano y son transferidos al archivo central.
  - Dirección de gestión documental y archivo: Encargada de los procesos y procedimientos del sistema institucional, esta debe elaborar un plan institucional que debe ser aprobado por la máxima autoridad de la institución, así como establecer los parámetros para el control de gestión documental
  - Archivo central: En esta se concentra y custodia los expedientes que terminaron su etapa de conversación en el archivo de gestión o activo.
- **Artículo 18:** El sistema institucional deberá encargarse de todos los procesos de gestión documental, como las transferencias documentales.
- **Artículo 20:** El registro de entrada/salida de correspondencia corresponde al canal de recepción de los documentos oficiales que se envíen a un destino garantizando la seguridad.
- **Artículo 23:** Para realizar la clasificación de archivo, la secretaría general o la dirección de gestión documental y archivo deberá generar el cuadro que usaran las unidades para clasificar los expedientes generados.
- **Artículo 25:** La valoración documental debe hacerse según la tabla generada según el artículo 23, que registra los valores documentales y plazos de conservación y las condiciones de acceso.
- **Artículo 27:** Las dependencias deben contar con una adecuada infraestructura que garantice la preservación de los documentos de archivo, acorde a estándares

internacionales. Además, se debe contar con un plan de prevención de riesgos, programa de inspección y mantenimiento, así como un plan de contingencia ante desastres.

Además, el mismo ente establece lineamientos si los archivos se almacenan de forma electrónica.

- **Artículo 29:** Los documentos electrónicos deben utilizar el mismo sistema de intensificación de los documentos de archivo, así como garantizar la autenticidad, fiabilidad, integridad y disponibilidad de estos. Además, deben contar con la estructura, contenido y contexto en base a las atribuciones de origen.
- **Artículo 30:** Los tiempos de conservación de los documentos electrónicos se contarán desde que se registren como concluidos en el sistema informático integral de gestión documental.
- **Artículo 31:** El sistema informático integral de gestión documental y archivo deberá ser implementado para administrar los procesos de los documentos electrónicos, este debe estar basado en estándares abiertos.
- **Artículo 33:** En caso de digitalización, las entidades deben contar con un cuadro y tabla de clasificación aprobada por la dirección de gestión documental, además, contar con la documentación física ordenada, clasificada, foliada e inventariada.

Las herramientas utilizadas para el desarrollo, pruebas y despliegue del equipo son los siguientes:

- **Azure DevOps:** “Proporciona un conjunto integrado de servicios y herramientas para administrar los proyectos de software, desde el planeamiento y el desarrollo a través de pruebas e implementación” [3]. Adicionalmente, esta herramienta nos proporciona una gestión del ciclo de vida de desarrollo de software.
- **Figma:** Emerge como una herramienta esencial en el desarrollo de aplicaciones web, desempeñando un papel fundamental en la fase de diseño y colaboración. Su plataforma basada en la nube permite a los equipos trabajar de manera simultánea y en tiempo real en el diseño de interfaces de usuario (UI) y experiencias de usuario (UX) [4].
- **MongoDB 1.41.0:** Con su enfoque de base de datos NoSQL, se presenta como una herramienta valiosa para el desarrollo de un sistema de gestión documental eficiente adicionalmente esta es una plataforma que nos permite manejar muchas

cargas de trabajo garantizando “transaccionales, analíticas, de búsqueda de texto completo, de series temporales y más” [5]. La flexibilidad inherente a MongoDB se alinea perfectamente con los requisitos expansibles de un sistema de gestión documental, donde la información puede variar en estructura y contenido. Su modelo de documentos BSON permite almacenar datos de forma nativa en un formato similar a JSON, facilitando la representación y manipulación de documentos complejos. Además, la capacidad de indexación y consulta eficiente de MongoDB agiliza la recuperación de documentos, permitiendo una rápida y eficaz búsqueda y filtrado de la información.

- **Visual Studio Code 1.86.0:** Esta es una herramienta de desarrollo muy utilizada por la mayor parte de programadores, esta herramienta nos permite el desarrollo de diversas aplicaciones y se adapta con cada uno de los frameworks. Adicional a esto, es una herramienta Open Source, la cual facilita cualquier tipo de desarrollo en cada uno de los lenguajes actualmente existentes.
- **GitHub 3.3.8:** Esta herramienta de versionamiento se ha “basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso” [6]. Adicionalmente, es Open Source, por lo que el desarrollo y versionamiento de código es sencillo con GitHub.
- **Citrix Secure Access 23.8.1.11:** La solución de acceso seguro de Citrix ofrece una pila unificada de servicios entregados en la nube que permite a los profesionales de TI proporcionar un entorno de trabajo híbrido productivo con seguridad de confianza cero. Aborda uno de los mayores desafíos de seguridad que enfrentan las empresas hoy en día: a medida que la fuerza laboral se vuelve más distribuida y se entregan más aplicaciones empresariales desde la nube, las arquitecturas de seguridad tradicionales que requieren soluciones basadas en dispositivos, como VPN, SWG, controles de DLP y WAF, resultan ser insuficientes [7]. Esta herramienta para la gestión de conexiones seguras es facilitada por la universidad mediante un licenciamiento por usuario, por lo que esta herramienta es ideal para el desarrollo de aplicaciones web.



## 2 METODOLOGÍA

### 2.1 Metodología de desarrollo SCRUM

El origen de la palabra se encuentra en un ámbito muy alejado del de la gestión de proyectos: en el deporte. En rugby, «scrum» es el término que define a la formación en la que ambos equipos, agazapados y atenzados entre sí, empujan para obtener el balón sin tocarlo con la mano [8].

Scrum se caracteriza por basarse en la agilidad promulgada con equipos autosuficientes y autónomos, estos equipos trabajan juntos para compartir conocimientos y que este sea para el aprendizaje conjunto del equipo. Adicionalmente, uno de los beneficios importantes es el aprendizaje en base a la experiencia, la cual nos permite adaptarnos a la mejora continua de las actividades realizadas o procesos necesarios que nos ayudan a garantizar el éxito en el desarrollo de sistemas.

Scrum se fundamenta en tres pilares: transparencia, inspección y adaptación constante, y estos se caracteriza por sus roles que están bien definidos, eventos estructurados y artefactos esenciales. De forma general, la inspección nos permite determinar los puntos de mejora con base en los procesos ya establecidos; la transparencia nos ayuda a que el equipo conformado tenga acceso al conocimiento adquirido en las actividades o procesos realizados; finalmente, la adaptación nos permite que las experiencias aprendidas se reflejen en el equipo.

Para garantizar la implementación del marco Scrum en el desarrollo del producto se utilizó Azure DevOps, herramienta que se utiliza como una solución integral permitiéndonos gestionar el desarrollo del aplicativo. En el marco Scrum, la planificación del ciclo de vida de desarrollo de software es crucial, por lo que Azure DevOps ayudo a garantizar los siguientes elementos clave:

- Ciclo de vida de Scrum:
  - Sprints: Estos fueron creados como interacciones de tiempo fijo, lo que permite planificar y ejecutar el trabajo en tiempos establecidos. Adicionalmente, permitió definir la duración de los sprints y gestionar las tareas dentro de ellos.
  - Roles: Propietario del producto, en Azure DevOps, el Product Backlog es donde se mantendrá el trabajo pendiente. El propietario del producto puede priorizar y actualizar los elementos del backlog; en el equipo de desarrollo,

los miembros del equipo utilizarán Azure Boards para crear, asignar y realizar un seguimiento de las tareas.

- Planeamiento del sprint: Azure DevOps permitirá al equipo seleccionar elementos del Product Backlog para incluirlos en el Sprint Backlog. Adicionalmente, se tiene la estimación y compromiso, en donde los equipos pueden estimar el esfuerzo y compromiso con las tareas utilizando Azure Boards.
- Azure Repos: Proporcionará el control de versiones y colaboración para el código fuente.

## 2.2 Roles

Un equipo Scrum está compuesto por tres roles: Product Owner, Desarrollador, Scrum Máster. Estos roles mencionados cumplen funciones específicas que aportan al desarrollo de un producto software.

El **Product Owner** o Propietario del Producto es quien transmite las ideas y necesidades a los desarrolladores buscando que este sea un medio de negociación y conversación con el cliente. Este es quien toma las decisiones del cliente y tiene como responsabilidad es garantizar el valor del producto [8]. En algunos casos el Product Owner y el cliente son la misma persona; esto suele ser muy común en aplicaciones internas [9].

Además, es quien está a cargo de los requisitos, decide cómo queda el producto final y así se construye, se adiciona o se debería eliminar según los requerimientos del producto, y se asigna la prioridad de las historias de usuarios definidas en la recopilación. Este también conoce aspectos importantes para el producto como: plan del producto, inversión planificada, alcance del producto, y principalmente como el producto agrega valor a la organización. Finalmente, este también tiene la responsabilidad de que los plazos de entrega se cumplan con las entregas previstas del producto.

**Desarrollador** o equipo de desarrollo es quien construye el producto que solicita el cliente. “El equipo decide a que se compromete y como hacer lo mejor para cumplir con lo prometido” [9]. El equipo de desarrollo es encargado de seleccionar las actividades que pueden manejar y ejecutar de la forma más eficiente posible [10].

El equipo está conformado por 3 o hasta 10 miembros, estos generalmente están conformado por desarrolladores, diseñadores de interfaces, personal de pruebas,

ingenieros de requerimientos, etc. Este equipo se comunica con el Scrum Máster para que la información necesaria para la elaboración del producto sea la más eficaz.

**Scrum Máster**, este es el responsable del cumplimiento de las reglas y que el marco scrum funcione, moderando las reuniones de scrum para mantener el avance del equipo de desarrollo [8]. El rol de Scrum Máster gestiona que cada miembro del equipo de desarrollo este realizando el trabajo acordado y que estos se entreguen en los plazos correspondientes. En esencia su trabajo no consiste en dar órdenes, sino en guiar el equipo en la aplicación correcta de las reglas que el marco scrum propone.

La Tabla 2.1, proporciona una representación detallada del Scrum Team conformado para el desarrollo del presente proyecto:

**Tabla 2.1.** Conformación de Scrum Team

<b>Rol</b>	<b>Personal</b>	<b>Módulo/Responsable</b>
Product Owner	Tnlgo. Enver David Ayala Cabrera	Stakeholder
Scrum Master	MCs. Rodrigo Fabian Chancusig Chuquilla	Director de Tesis
Desarrolladores	Sr. José David Pallo Morales	Administrador/Integrador
	Sr. Marlon Israel Pachacama Granda	Decanato
	Sr. Erick Alexander Vera Toscano	Subdecanato
	Sr. Jhosue Alexander Morales Cazar	Jefatura de Departamento

### **Artefactos**

Según Palacio (2022): “Los artefactos de scrum son sus herramientas, sus bloques de construcción elementales. Ayudan a los «roles» durante los «eventos»”. Este paso ayuda a que el Product Owner tenga una visión clara del producto y la definición de cada artefacto este alineada con la visión del producto. De los artefactos se puede destacar tres que son clave para el correcto funcionamiento del marco Scrum:

El **Product Backlog** tiene como objetivo registrar el conjunto de historias de usuario las cuales representas las necesidades desde el punto de vista del cliente, las mismas que son generalmente generadas por el Product Owner. Es importante la participación del Product Owner, dado que, es el quien ayuda a comprender las historias de usuario al

equipo de desarrollo para que sean debidamente incluidas en los sprints planificados. Además, hay que considerar que el producto empieza con una versión inicial y puede ir cambiando en el tiempo, es decir, que puede crecer en términos de requisitos.

El **Sprint Backlog** se enfoca en las tareas necesarias para construir una parte de las historias de usuarios que ayudan a la construcción parcial del producto que se quiere construir con cada uno de los sprints, los cuales están compuestos por unidades de tamaño adecuadas para que el avance diario contribuya significativamente en la construcción total del producto.

El **incremento** se materializa como producto del trabajo llevado a cabo durante un sprint y se encuentra listo para su entrega al cliente, con el objetivo de obtener retroalimentación. Como fase conclusiva del incremento, se requiere evaluar su conformidad con los estándares de calidad establecidos por normativas y los específicos del cliente.

### **Recopilación de requerimientos**

Para recopilar la información necesaria, se emplea la ingeniería de requisitos, cuya finalidad es la transmisión de estos con todas sus características. Tradicionalmente, existen dos métodos para llevar a cabo este proceso. En primer lugar, se utiliza la comunicación escrita, la cual posibilita obtener un registro permanente, facilita la compartición con todos los miembros involucrados en el desarrollo del producto y permite realizar un análisis de manera general y completa. En segundo lugar, se recurre a la comunicación verbal, que posibilita recibir feedback de manera inmediata. Además, este enfoque se realiza de forma dinámica, permitiendo que la conversación se adapte para maximizar la eficiencia en la recopilación de requerimientos. Un aspecto destacado es que esta modalidad facilita la generación de ideas nuevas y contribuye a lograr un mayor nivel de comprensión y claridad con un menor esfuerzo.

### **Historias de usuario**

De acuerdo con Palacios: "Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal". Esta forma de recopilar los requerimientos nos ayuda a que en una o dos frases se describa una funcionalidad de software desde el punto de vista del usuario, de tal forma que, sea entendido el valor que busca conseguir y este mismo sea transmitido al equipo encargado del desarrollo del producto cumpliendo con las necesidades o problemas que va a solucionar.

La utilización de historias de usuario posibilita a la organización agilizar la recopilación de requisitos al reducir la necesidad de documentos formales y los tiempos asociados a su elaboración. Según Jeffries (2001), “las historias de usuarios se componen por tres partes que son card, conversation y confirmation; también denominadas como las tres C”.

Una **tarjeta** (card), es la reducción del requerimiento hasta hacerlo fácil de memorizar y de comprender sintetizándolo en una o dos frases. La tarjeta como tal es una promesa y recordatorio de una **conversación** (conversation) con las partes interesadas en la elaboración del producto, adicionalmente, scrum se basa en la agilidad, por lo que los cambios se pueden implementar constantemente a lo largo del ciclo de vida del desarrollo del producto. Finalmente, la **confirmación** señala que el Product Owner o los usuarios han manifestado su acuerdo con los requisitos recopilados y han revisado los criterios de aceptación.

En la metodología Scrum, se espera que las historias de usuario y otros artefactos cumplan con estándares de calidad. En el caso específico de las historias de usuario, se emplean atributos destinados a garantizar su calidad, tales como ser independientes, negociables, valiosas, estimables, pequeñas y comprobables. Se utiliza un método conocido como INVEST para incorporar estos atributos, considerándolos como elementos fundamentales. Este enfoque se implementa con el objetivo de asegurar la calidad en la redacción y desarrollo de las historias de usuario dentro del contexto de Scrum.[11].

Una historia de usuario **independiente** se refiere a la capacidad para poder planificar e implementar las mismas de una forma coherente, sin la necesidad que se acuda a otra historia [11]. Este atributo es importante para la estimación y su correcta implementación.

La flexibilidad para que los clientes o usuarios lleguen a un acuerdo de los pormenores de las historias de usuario permiten que una historia de usuario sea **negociable**. Esto significa que no es un contrato explícito en donde se documentan todas las funciones, sino todo lo contrario, son detalles creados conjuntamente por el cliente y el equipo de desarrollo [12].

La habilidad para conferir una elevada importancia a una historia de usuario revela su valor intrínseco, indicando que contribuye significativamente al beneficio del cliente o usuario. En muchos casos, esta valía se logra al incorporar activamente al cliente en la redacción de la historia de usuario, permitiendo así una comprensión más precisa de sus necesidades y preferencias. [11].

Una historia de usuario es **estimable**, si esta se ha evaluado para definir si se va a completar en las iteraciones planificadas [13]. En el proceso de estimación, el equipo de

desarrollo se encarga de llevar a cabo dicha tarea, estableciendo una relación directa con el tamaño de la historia de usuario. Esta conexión sugiere que la presencia de cualquier grado significativo de ambigüedad en la historia de usuario puede dificultar la precisión de la estimación.

Una historia de usuario es estimable, si esta se ha evaluado para definir si se va a completar en las iteraciones planificadas [13]. Para esta estimación el equipo de desarrollo es quien realiza la misma, y relacionada directamente con el tamaño de la historia, esto nos indica que si la historia de usuario tiene algún nivel de ambigüedad mayor para estimarla correctamente.

Por otra parte, se sostiene que las historias de usuario deberían abarcar un periodo de unas pocas semanas de trabajo asignadas al equipo de desarrollo. Esto conlleva la necesidad de que dichas historias sean de dimensiones reducidas. Se argumenta que descripciones concisas y breves aportan un mayor valor al equipo de desarrollo, especialmente cuando este ya posee un conocimiento sólido sobre el producto y su funcionamiento a nivel de negocio. [14].

Las historias deben redactarse de manera que sean comprobables, lo que ayuda a verificar si la historia de usuario cumple con su función específica. Esta práctica facilita al equipo de desarrollo determinar si la implementación de una o varias funciones se considera finalizada. [13].

En base a las reuniones realizadas con el Product Owner junto con los usuarios del sistema, se ha hecho una recopilación de los requisitos de estos para la creación de las historias de usuario para el presente proyecto. En la Tabla 2.2 se muestra las épicas junto con las historias de usuario que se han definido para el proyecto.

**Tabla 2.2.** Épicas e Historias de usuario para el prototipo de gestión documental

Épica		Historia de usuario	
Código	Nombre	Código	Nombre
US001	Gestión Documental	US001-1	Registro de documento
		US001-2	Visualización de historial de documentos
		US001-3	Edición de documentos
		US001-4	Eliminación de documentos
		US001-5	Visualización de archivo de documento guardado

		US001-6	Búsqueda de documentos
US002	Gestión de usuarios	US002-1	Registro de usuario
		US002-2	Inicio de sesión
		US002-3	Actualización de usuario
		US002-4	Visualización de usuarios

En el Anexo I, se puede visualizar todas las historias de usuario que se desarrollaron para el sistema de gestión documental.

## 2.3 Diseño de interfaces

Un mockup es una representación de fidelidad media a alta de la apariencia del producto, en este se muestra la funcionalidad básica que tendrá. En el diseño de los mockups se tiene en cuenta los detalles visuales como colores o tipografía, ya que proporciona una visión anticipada del aspecto final del producto y una aproximación a la funcionalidad de este [15]. En este proceso, se empleó la herramienta Figma, una plataforma colaborativa de diseño de interfaces, que permite la creación y edición eficiente de mockups [16]. Esta elección se basa en la capacidad de Figma para facilitar la colaboración en tiempo real y su interfaz amigable.

### Modulo administrador

Este módulo alberga las interfaces destinadas a la observación del historial de usuarios creados. En este contexto, se posibilita la creación, actualización y búsqueda de usuarios según las necesidades específicas. Asimismo, permite la visualización del historial de archivos registrados por los módulos correspondientes a Decanato, Subdecanato y Jefatura de Departamento.

En la Figura 2.1, se muestra el Login de la aplicación, para el ingreso al módulo de usuarios se requiere un email y una contraseña como credenciales, estas son necesarias para el proceso de autenticación en la base de datos.



**Figura 2.1.** Interfaz inicio de sesión módulo Administrador

En la Figura 2.2, se muestra una lista de los usuarios creados, estos se muestran listados por parámetros de número, nombre, facultad, rol, email. Adicionalmente se muestra un cuadro de búsqueda el cual esta implementado de tal forma que filtra datos en función de cada uno de los datos credos para los usuarios



**Figura 2.2.** Interfaz de usuario módulo Administrador

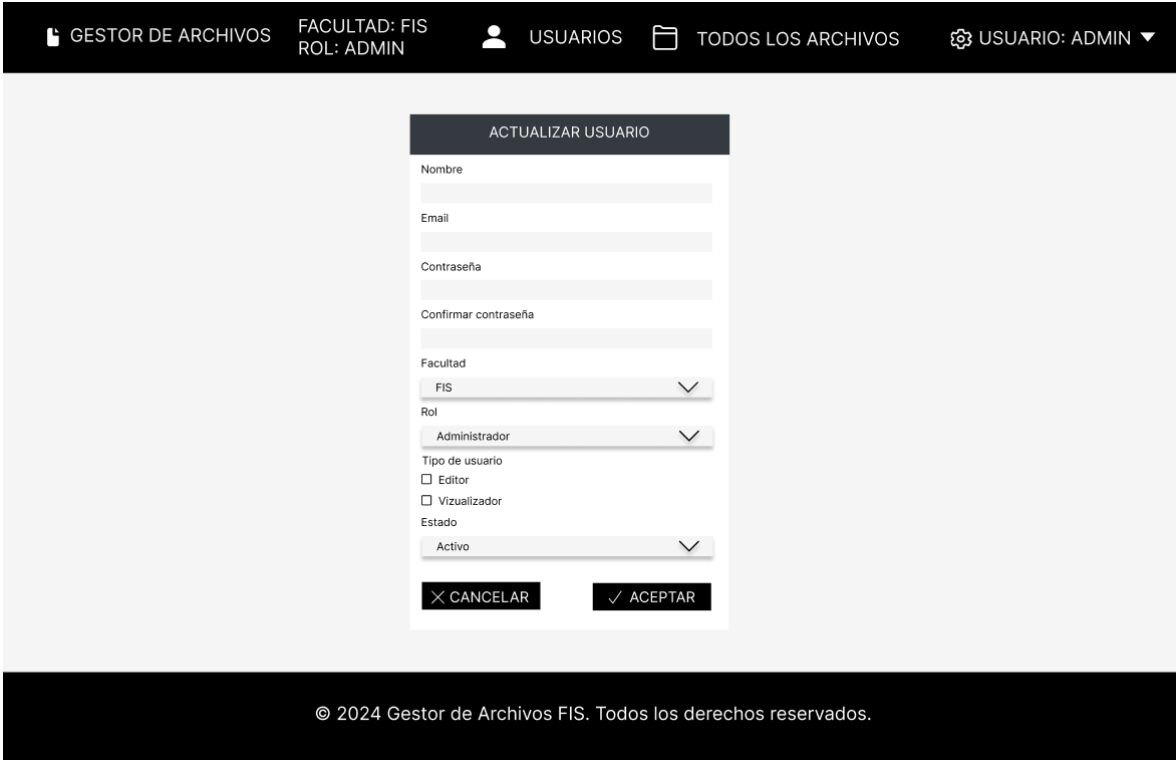


En la Figura 2.3, se presenta el formulario de creación de un usuario, donde se exhiben todos los parámetros necesarios para llevar a cabo dicho proceso. En este módulo, se requiere tener en cuenta el tipo de usuario a ser creado, así como su estado, ya que ambos aspectos son de alta importancia para verificar los privilegios asignados en función del tipo de rol seleccionado para dicho usuario.

The screenshot shows a web application interface with a dark header. The header contains the following elements from left to right: a folder icon and the text 'GESTOR DE ARCHIVOS', 'FACULTAD: FIS' and 'ROL: ADMIN', a user icon and the text 'USUARIOS', a folder icon and the text 'TODOS LOS ARCHIVOS', and a gear icon and the text 'USUARIO: ADMIN' with a dropdown arrow. Below the header is a light gray area containing a white modal form titled 'AGREGAR USUARIO'. The form has the following fields and controls: 'Nombre' (text input), 'Email' (text input), 'Contraseña' (password input), 'Confirmar contraseña' (password input), 'Facultad' (dropdown menu with 'FIS' selected), 'Rol' (dropdown menu with 'Administrador' selected), 'Tipo de usuario' (checkboxes for 'Editor' and 'Visualizador'), and 'Estado' (dropdown menu with 'Activo' selected). At the bottom of the form are two buttons: 'CANCELAR' with a close icon and 'GUARDAR' with a save icon. Below the form is a dark footer with the text '© 2024 Gestor de Archivos FIS. Todos los derechos reservados.'

**Figura 2.3.** Formulario agregar usuario

Continuando con el módulo de Administrador, en la Figura 2.4, se presenta el formulario destinado a la actualización de un usuario. En este contexto, se tienen en cuenta los elementos esenciales para llevar a cabo la actualización del usuario. El formulario exhibe parámetros significativos que están definidos en acuerdos y requisitos no funcionales cruciales, los cuales tienen como finalidad asegurar la integridad de los datos que se modifican.



© 2024 Gestor de Archivos FIS. Todos los derechos reservados.

**Figura 2.4.** Formulario actualizar usuario

Finalmente, en la Figura 2.5, se muestra la vista de todos los archivos, esto sin importar el rol con el que se haya cargado los documentos, con esto garantizamos la integración en un solo módulo para la visualización de los archivos cargados.



© 2024 Gestor de Archivos FIS. Todos los derechos reservados.

**Figura 2.5.** Historial de archivos

## Módulos de gestión documental

Los módulos de Decanato, Subdecanato y Jefatura de Departamento dentro del sistema de gestión documental se distinguen por la naturaleza y tipología de los documentos que contienen, reflejando así sus funciones específicas dentro de la Facultad de Ingeniería en Sistemas.

Estos módulos posibilitan la visualización del historial de archivos registrados, presentándolos de manera estructurada en una tabla. En este contexto, se ofrecen diversas acciones que corresponden al acrónimo CRUD, que representa las operaciones fundamentales aplicables a los datos dentro de una base de datos. A continuación, se detallan las siglas y sus respectivas descripciones.

**Crear (C):** Esta operación implica agregar nuevos registros de datos a la base de datos. En el contexto de una base de datos, generalmente corresponde a insertar una nueva fila en una tabla.

**Leer (R):** La operación de lectura implica recuperar y ver datos existentes de la base de datos. Corresponde a consultar la base de datos para obtener registros específicos o recuperar todos los registros de una tabla.

**Actualizar (U):** La operación de actualización se utiliza para modificar datos existentes en la base de datos. Normalmente implica cambiar los valores de campos específicos dentro de un registro o actualizar el registro completo.

**Eliminar (D):** La operación de eliminación se utiliza para quitar datos de la base de datos. Corresponde a eliminar una fila de una tabla, eliminando efectivamente un registro.

Como referencia visual, se presentan capturas de pantalla del diseño y funcionamiento del módulo de Decanato, ofreciendo una visión práctica de su interfaz.

En la Figura 2.6, se presenta la interfaz de inicio de sesión diseñada para acceder al módulo de Decanato. Para garantizar la seguridad y autenticidad de los usuarios, se solicita la introducción de un correo electrónico y una contraseña, ambas creadas previamente por el administrador del sistema. Estas credenciales actúan como un mecanismo de acceso exclusivo, asegurando que solo usuarios autorizados puedan ingresar al sistema y acceder a las funcionalidades del módulo de Decanato. Para salvaguardar la información sensible y garantizar un acceso seguro y controlado a las herramientas del sistema, se ha implementado este proceso de autenticación.



**Figura 2.6.** Interfaz de inicio de sesión módulo Decanato

En la Figura 2.7, se puede apreciar el historial de archivos que alberga el módulo de Decanato. Este componente no solo ofrece un registro completo de los documentos gestionados, sino que también proporciona diversas opciones relacionadas con CRUD (Create, Read, Update, Delete). Entre estas funciones, se destacan la capacidad de crear nuevos archivos, editar la información existente, eliminar registros no deseados y buscar archivos específicos. La versatilidad de este módulo permite a los usuarios gestionar de manera eficiente la información almacenada, facilitando así la administración y actualización de los archivos en el entorno del Decanato.

GESTOR DE ARCHIVOS FACULTAD: FIS ROL: DECANATO TODOS LOS ARCHIVOS USUARIO: DECANATO

### HISTORIAL DE ARCHIVOS

Mostrar 10 registros Buscar:

N°	FACULTAD	CARRERA	ÁREA	SUBÁREA	SERIE	SUBSERIE	PERIODO/AÑO	CODIFICACIÓN	N° FOLIO	ASUNTO	OBSERVACIONES	ACCIONES
1	FIS	Computación	Decanato	Secretaría de Decanato	Consejo de Facultad	Actas/Resoluciones	2018-A	FIS-DC-2018-0001	5	asunto	observaciones	<a href="#">Ver</a> <a href="#">Eliminar</a> <a href="#">Actualizar</a>
2	FIS	Software	Decanato	Secretaría de Titulación	Pensum	Silabos	2020	FIS-DC-2020-0001	10	asunto	observaciones	<a href="#">Ver</a> <a href="#">Eliminar</a> <a href="#">Actualizar</a>
3	FIS	Computación	Decanato	Secretaría de Decanato	Solicitud de Estudiantes	Formularios de Autorización	2021	FIS-DC-2021-0001	42	asunto	observaciones	<a href="#">Ver</a> <a href="#">Eliminar</a> <a href="#">Actualizar</a>
4	FIS	Computación	Decanato	Secretaría de Titulación	Comisiones	Practicas Pre-Profesionales	2022-B	FIS-DC-2022-0001	4	asunto	observaciones	<a href="#">Ver</a> <a href="#">Eliminar</a> <a href="#">Actualizar</a>

Mostrando registros del 1 al 4 de un total de 4 registros Anterior 1 Siguiente

[+ NUEVO](#)

© 2024 Gestor de Archivos FIS. Todos los derechos reservados.

**Figura 2.7.** Historial de archivos creados módulo Decanato

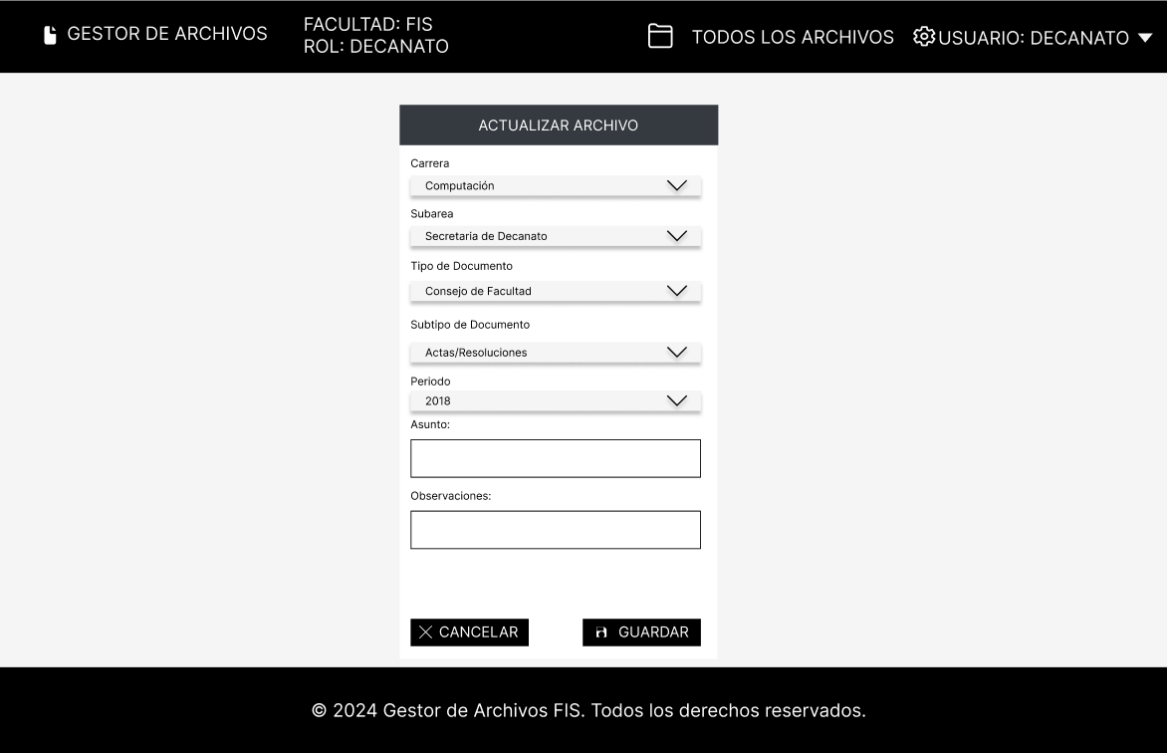
En la Figura 2.8 se exhibe el formulario diseñado para facilitar la creación de un archivo nuevo por parte del Decanato. Este formulario presenta diversos parámetros cuidadosamente establecidos, permitiendo así un registro de archivos de manera flexible y eficaz. La interfaz intuitiva y los campos específicos proporcionan una herramienta eficiente para asegurar la correcta captura de información, garantizando un proceso ágil y adaptado a las necesidades del Decanato en el manejo de archivos.

The image shows a web interface for a document management system. At the top, a dark navigation bar contains the text 'GESTOR DE ARCHIVOS', 'FACULTAD: FIS', 'ROL: DECANATO', 'TODOS LOS ARCHIVOS', and 'USUARIO: DECANATO'. Below this is a central form titled 'NUEVO ARCHIVO'. The form contains several dropdown menus: 'Carrera' (selected: Computación), 'Subarea' (selected: Secretaría de Decanato), 'Tipo de Documento' (selected: Consejo de Facultad), 'Subtipo de Documento' (selected: Actas/Resoluciones), and 'Período' (selected: 2018). There is a file selection field for 'Archivo PDF' with the text 'Seleccionar archivo' and 'Ningun archivo sel...'. Below that are input fields for 'Número de Páginas', 'Asunto', and 'Observaciones'. At the bottom of the form are two buttons: 'CANCELAR' and 'GUARDAR'. A footer at the bottom of the page reads '© 2024 Gestor de Archivos FIS. Todos los derechos reservados.'

**Figura 2.8.** Formulario nuevo archivo módulo Decanato

El formulario de actualizar archivo, representado en la Figura 2.9, constituye una herramienta esencial dentro del sistema, permitiendo la modificación de archivos existentes de manera eficiente. Este formulario opera siguiendo parámetros previamente establecidos, de manera análoga al formulario de nuevo archivo. Al realizar cambios mediante este formulario, se actualizan automáticamente los datos en la base de datos correspondiente, asegurando la coherencia y precisión de la información almacenada. Además, estos ajustes se reflejan de manera inmediata en la tabla de historial de archivos específica para el módulo de decanato, proporcionando un registro detallado de todas las

actualizaciones realizadas en los archivos, lo que facilita el seguimiento y la gestión eficaz de la documentación en el contexto del decanato.



The image shows a web interface for updating an archive. At the top, a dark navigation bar contains the text 'GESTOR DE ARCHIVOS', 'FACULTAD: FIS', 'ROL: DECANATO', 'TODOS LOS ARCHIVOS', and 'USUARIO: DECANATO'. The main content area features a white form titled 'ACTUALIZAR ARCHIVO'. The form contains several dropdown menus: 'Carrera' (selected: Computación), 'Subarea' (selected: Secretaría de Decanato), 'Tipo de Documento' (selected: Consejo de Facultad), and 'Subtipo de Documento' (selected: Actas/Resoluciones). Below these are 'Periodo' (selected: 2018), an 'Asunto:' text input field, and an 'Observaciones:' text input field. At the bottom of the form are two buttons: 'CANCELAR' and 'GUARDAR'. A footer bar at the bottom of the page contains the text '© 2024 Gestor de Archivos FIS. Todos los derechos reservados.'

**Figura 2.9.** Formulario actualizar archivo módulo Decanato

Para más detalles sobre el desarrollo y contenido de los demás módulos, se recomienda consultar el Anexo II, donde se detalla la estructura y funcionalidades específicas de cada módulo. Este anexo complementario ofrece una visión integral del sistema de gestión documental, proporcionando información adicional sobre la organización y acceso a la documentación pertinente para cada módulo.

## 2.4 Diseño de la arquitectura

### Modelo Vista Controlador

Introducido inicialmente en 1979 para Smalltalk, el patrón de arquitectura MVC es ampliamente empleado en la implementación de sistemas de software que implican interacción con clientes o usuarios a través de interfaces. La adopción de esta arquitectura en el desarrollo de productos de software posibilita la obtención de un ciclo de vida más apropiado, facilitando la implementación de funcionalidades, el mantenimiento y la reutilización del código. Además, destaca por ser un modelo altamente escalable. [17].

La arquitectura del Modelo-Vista-Controlador se basa en capas que permiten asignar a cada sección de código funcionalidades y cada capa se trata como entidades separadas, así que cualquier cambio realizado a nivel del modelo se refleje en cada vista configurada.

El modelo se descompone en la capa de presentación y la capa de dominio; los componentes son: el **modelo** en el cual se incluye la implementación de las funcionalidades necesarias y los datos del producto software, la vista que es esencialmente el componente que muestra la información al usuario final y el **controlador** el cual es responsable de la gestión de las interacciones con el usuario [18].

En la capa de presentación se incluyen estos componentes:

- Vista: presentación de la información
- Controlador: mecanismo de interacción entre la vista y el modelo

En la capa de dominio se encuentra el modelo que es el que se encarga de la gestión de datos y la representación lógica del negocio.

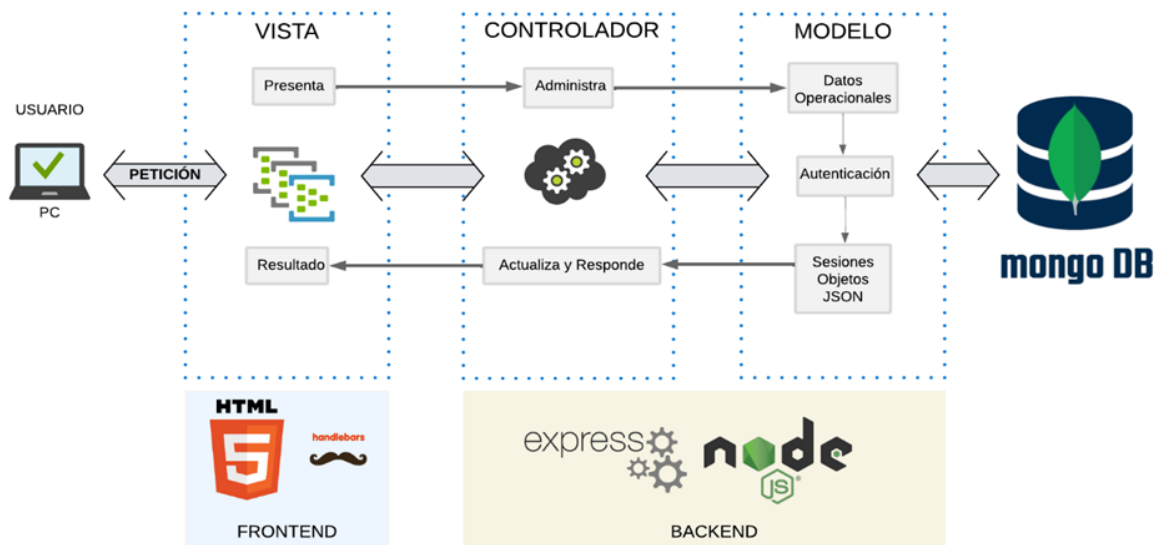


**Figura 2.10.** Funcionamiento de patrón MVC [19]

La elección de la arquitectura Modelo-Vista-Controlador (MVC) para la aplicación web se basó en la necesidad de establecer una estructura organizada y modular que permita un desarrollo eficiente y mantenible. Como se muestra en la Figura 2.11, la adopción de esta arquitectura se traduce en una clara separación de responsabilidades, donde la Vista, representada por tecnologías HTML5 y Handlebars, se encarga de la presentación y la interfaz de usuario, proporcionando una experiencia visual atractiva y dinámica. Por otro lado, el Controlador, implementado con Node.js y Express, gestiona las interacciones entre el usuario y la aplicación, manejando las solicitudes y coordinando las respuestas correspondientes. Esta elección se fundamenta en la capacidad de Node.js para ejecutar operaciones de entrada/salida de manera eficiente y en la flexibilidad proporcionada por



Express para desarrollar rutas y middleware de manera estructurada. Finalmente, el Modelo representado en la Figura 2.12, que utiliza MongoDB, ofrece una base de datos NoSQL que se integra sin problemas con Node.js, permitiendo un almacenamiento eficiente y escalable de los datos de la aplicación. La combinación de estas tecnologías en el contexto de la arquitectura MVC asegura un desarrollo coherente, facilita la escalabilidad y mejora la mantenibilidad de la aplicación web, al mismo tiempo que permite un despliegue más ágil de nuevas funcionalidades.



**Figura 2.11.** Arquitectura implementada en el sistema de gestión documental

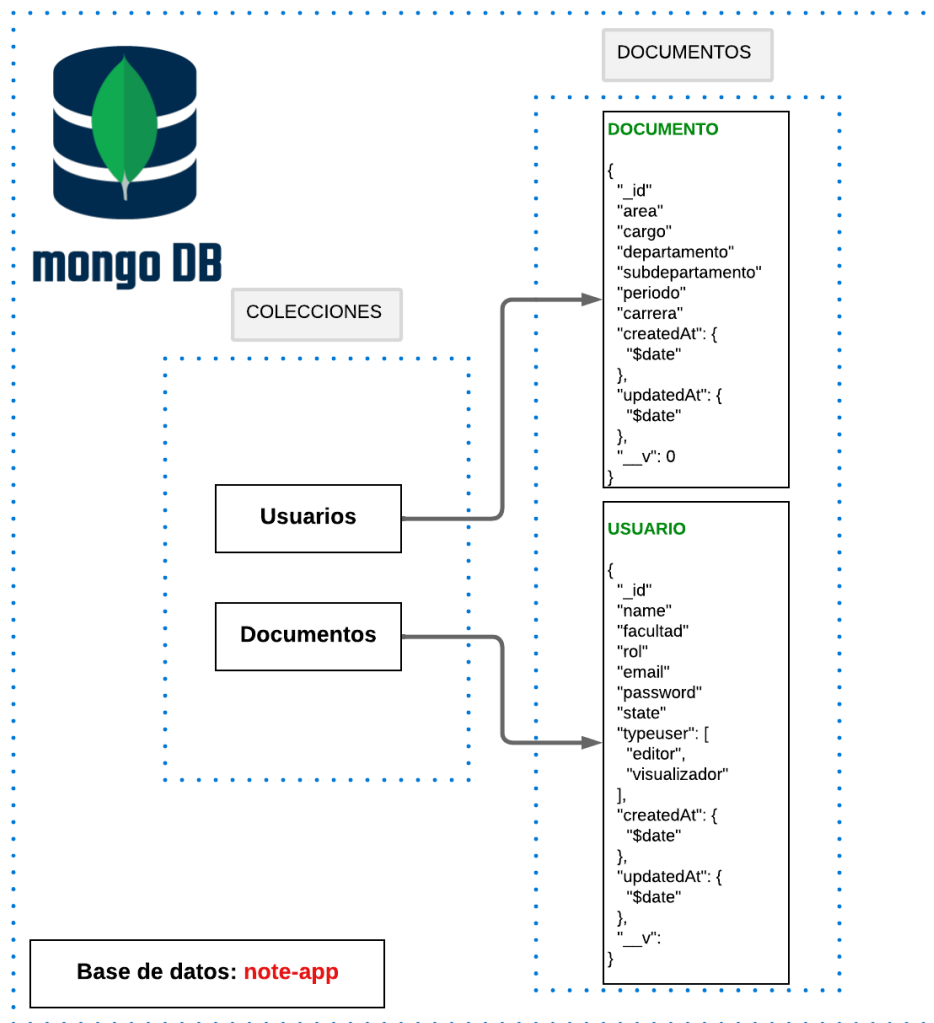


Figura 2.12. Arquitectura de base de datos

## 2.5 Herramientas de desarrollo

En el contexto del desarrollo del prototipo del gestor de archivos para la FIS, se emplearán herramientas y recursos de software de licencias gratuitas destinados a facilitar y mejorar el proceso integral de creación, diseño y prueba. Estas herramientas son esenciales para maximizar la productividad de los desarrolladores, asegurar la calidad y eficiencia del proyecto de desarrollo web. A continuación, se detallan las herramientas implementadas para la creación del prototipo.

Tabla 2.3. Herramientas para el desarrollo del prototipo

Herramienta de desarrollo	Software propuesto
Planificación	Azure DevOps
Diseño de interfaz	Figma
Editor de código	Visual Studio Code 1.86.0

Control de versiones	GitHub 3.3.8
Bases de datos	MongoDB 1.41.0

## 2.6 Recolección de datos

La recolección de los datos para el desarrollo del sistema se hizo mediante:

- Documentación
- Cartas de confidencialidad y satisfacción

### Documentación

Antes de la implementación del prototipo para la Facultad de Ingeniería de Sistemas (FIS), se llevó a cabo un exhaustivo análisis de los requisitos proporcionados por el "Product Owner", quien es el encargado de definir las necesidades y expectativas del proyecto. En este proceso se identificó el uso de una macro de Excel, utilizada en el gestor de archivos de la secretaría general de la Escuela Politécnica Nacional (EPN), como referencia clave para información específica de los campos necesarios para el prototipo.

La macro de Excel proporcionó una estructura organizada que sirvió como base para la creación de campos esenciales en el formulario de registro de documentos, el cual se ve reflejado en una tabla denominada "Historial de archivos". Estos campos incluyen información vital como carrera, subárea, serie, subserie, periodo, codificación, número de folio, asunto y observaciones. La interfaz del formulario se diseñó con la premisa de facilitar el uso a los usuarios, cumpliendo así con uno de los requisitos establecidos en las reuniones de definición de requerimientos.

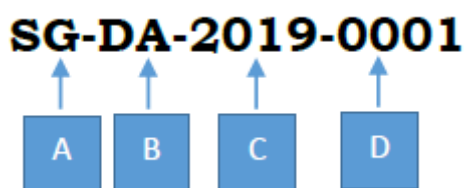
La atención meticulosa a los detalles presentes en la macro de Excel no solo facilitó la transición hacia el nuevo sistema de gestión de archivos para la FIS, sino que también aseguró la continuidad y mejora de los procesos administrativos en la Secretaría General de la EPN.

La macro de Excel proporcionada por el Product Owner y utilizada en el desarrollo del prototipo se la puede encontrar en el Anexo III.

Además, en el contexto de la presente investigación, es esencial destacar que la gestión eficiente de los documentos almacenados en el servidor para la gestión documental se rige por un sistema de codificación estructurado. La Figura 2.13 proporciona una visualización detallada de este proceso. Conforme a las directrices establecidas por la Escuela

Politécnica Nacional, la codificación de los documentos se lleva a cabo mediante la utilización de cuatro partes identificadas como A, B, C y D [20].

- A. Sección documental
- B. Subsección documental (en caso de ser aplicable)
- C. Año de la documentación
- D. Número de expediente con 4 dígitos.



**Figura 2.13.** Formato de la codificación de documentos

Esta metodología brinda una estructura organizada para la clasificación y recuperación de documentos, permitiendo una fácil identificación y acceso a la información almacenada. La implementación de este sistema de codificación no solo facilita la eficiencia en la gestión documental, sino que también asegura la coherencia y uniformidad en la nomenclatura de los archivos, contribuyendo así a una administración integral y ordenada de los recursos documentales.

### **Cartas de confidencialidad y satisfacción**

La Carta de Confidencialidad se refiere a un documento oficial que establece y delimita la naturaleza confidencial de ciertos datos, información o documentos. En el contexto de la Facultad de Ingeniería en Sistemas (FIS), esta carta especifica claramente que los documentos de sus archivos son reservados, destinados exclusivamente al acceso y uso del personal autorizado de la facultad. Además, se destaca que, en ciertas circunstancias específicas, como el desarrollo de prototipos y pruebas, los estudiantes también pueden acceder a estos documentos bajo condiciones controladas y con el único propósito de contribuir al avance y perfeccionamiento de proyectos académicos. Esta medida garantiza la protección de la información sensible y confidencial, preservando la integridad de los datos y asegurando su uso adecuado dentro del marco establecido por la facultad.

Por otro lado, la satisfacción de un cliente es el conjunto de sentimientos de placer o decepción que se genera en una persona como consecuencia de comparar el valor percibido en el uso de un producto en el desarrollo de un producto o servicio [21]. Las

cartas de satisfacción permiten obtener la retroalimentación de los clientes/usuarios sobre un producto o servicio. En este medio los clientes/usuarios hacen constar por escrito que se encuentran conformes con un producto o servicio dado por un tercero.

En el Anexo IV se puede visualizar las cartas de confidencialidad y satisfacción redactadas y a su vez firmadas por los usuarios que harán uso del desarrollo del prototipo.

## 2.7 Pruebas de seguridad

### SAST

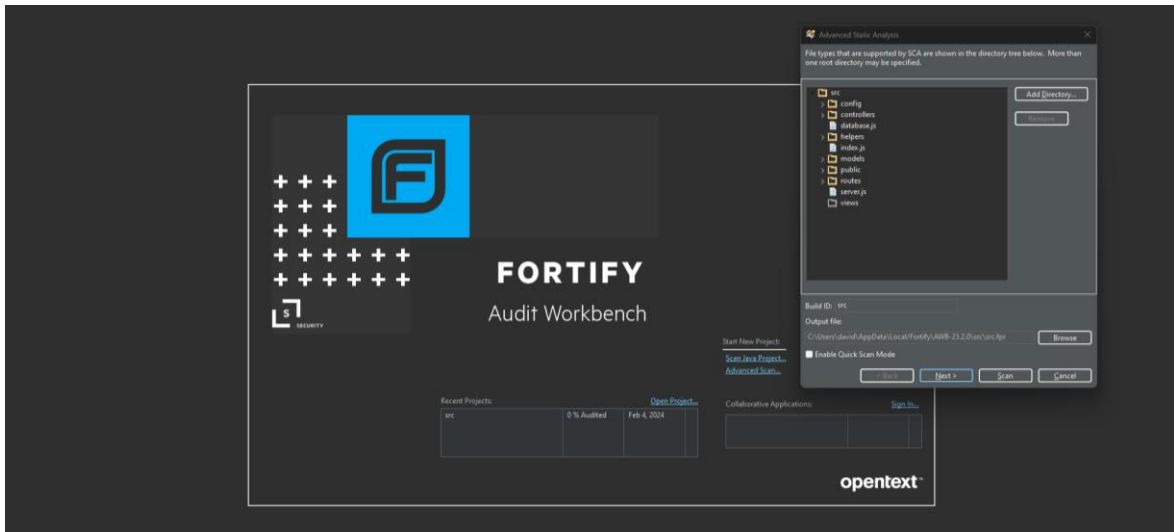
A continuación, se detalla el proceso de ejecución del análisis SAST (Static Application Security Testing), describiendo los pasos para la ejecución de este, mediante el componente AuditWorkbench, que es parte de la solución Fortify.

La solución utilizada para el escaneo del código fuente del “**Sistema de Gestión del Sistema de Archivos**” es **Fortify Static Code Analyzer**, y través del análisis de código estático, se revisa y analiza el código fuente para detectar vulnerabilidades de seguridad que hacen a las aplicaciones de su organización susceptibles a ataques.

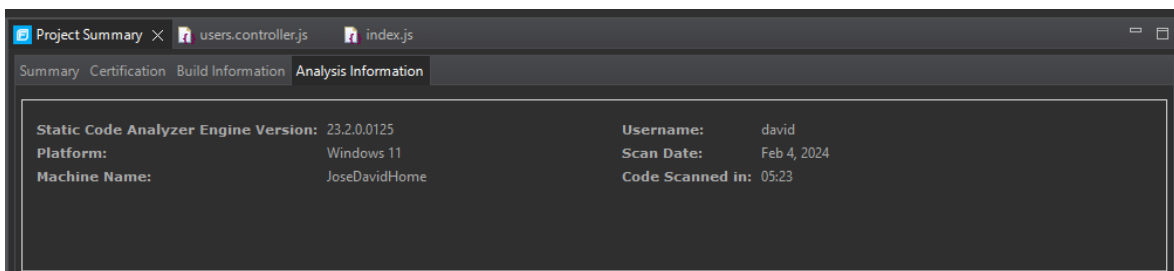
Los principales beneficios que se busca obtener mediante este análisis son los siguientes:

- Detectar las vulnerabilidades en etapas tempranas del ciclo de desarrollo.
- Identificar vulnerabilidades en código fuente, binario o byte.
- Reducir el riesgo al identificar y priorizar qué vulnerabilidades representan una mayor amenaza.
- Reduce el tiempo y el costo de desarrollo.
- Previo al inicio de la fase de producción/post-lanzamiento permite corregir vulnerabilidades en etapas tempranas del desarrollo.
- Minimizar la carga de trabajo del auditor.

En el componente Audit Workbench de Fortify, se carga el código fuente de la aplicación, para lo cual se debe seleccionar el directorio donde se encuentra el repositorio como se observa en la Figura 2.14. Este componente se encarga de ejecutar la fase de **traducción**, en donde se reúne el código fuente mediante una serie de comando y lo traduce a un formato intermedio asociado con un ID de compilación. Seguido de esto, se **escanea** los archivos fuente identificados en la fase de traducción para el cual se genera un archivo de resultados del análisis denominado FRP (Fortify Project Result).

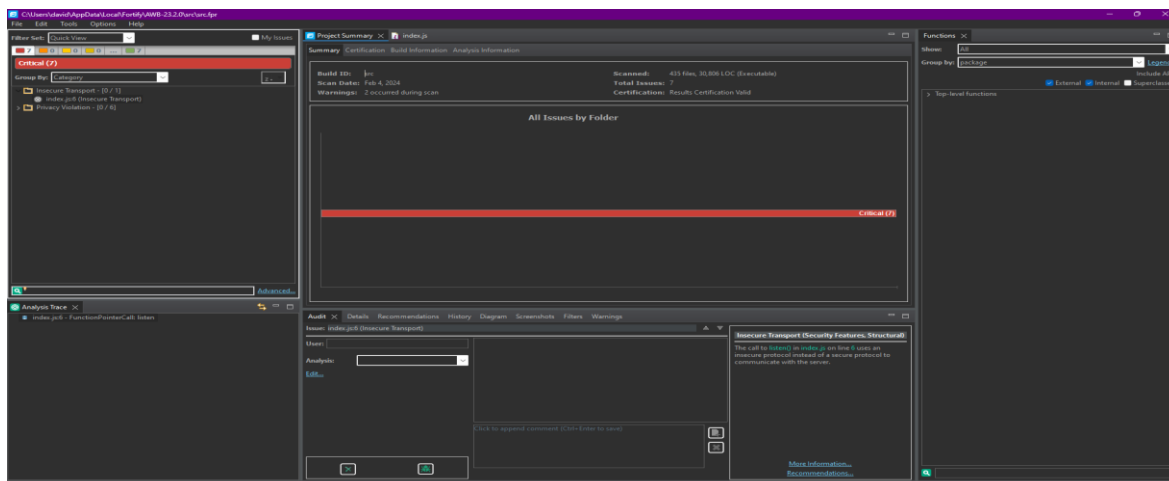


**Figura 2.14.** Interfaz de Audit Workbench para la carga del código fuente para el análisis. Finalmente, se **verifica** los archivos que se utilizaron en el análisis de los paquetes de reglas y se obtiene un resumen posterior al análisis como se observa en la Figura 2.15.



**Figura 2.15.** Detalles del análisis realizado

Una vez realizadas las fases necesarias para la ejecución de la prueba se muestran los resultados de este. En la Figura 2.16, se muestra la interfaz con los resultados obtenidos después de la ejecución de la prueba, en el cual se muestra cada una de las vulnerabilidades encontradas en análisis.



**Figura 2.16.** Interfaz de resultados del análisis SAST en Audit Workbench de Fortify

## DAST

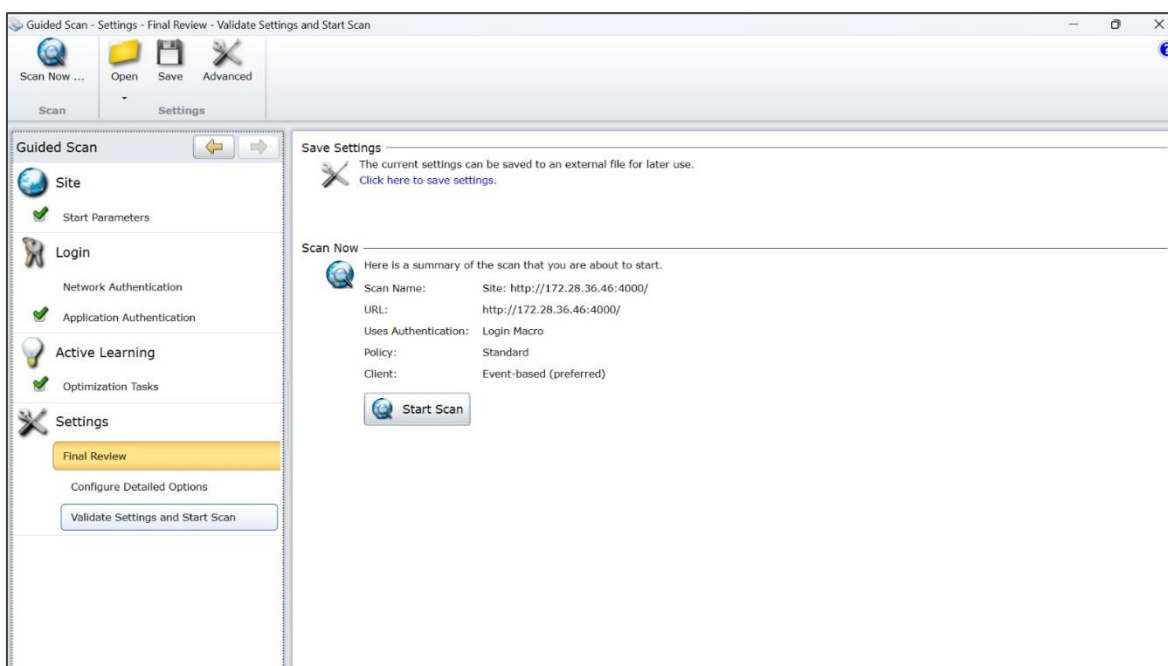
A continuación, se detalla el proceso de ejecución del análisis DAST (Dinamic Application Security Testing), describiendo los pasos para la ejecución de este, mediante el componente Fortify WebInspect, que es parte de la solución Fortify.

La solución utilizada para el escaneo dinámico de vulnerabilidades del “**Sistema de Gestión del Sistema de Archivos**” es **Fortify WebInspect**, y través del análisis dinámico, se realiza un análisis para identificar vulnerabilidades en la aplicación y servicios web implementado para el sistema.

Los principales beneficios que se busca obtener mediante este análisis son los siguientes:

- Detectar las vulnerabilidades que se encuentran el sistema implementado en un entorno de producción.
- Identificar vulnerabilidades en el lado del cliente y los números de versión, y como estos podrían convertirse en vulnerabilidades si estos no se actualizan o reparan.
- Identificar vulnerabilidades a través de los archivos HAR.
- Cumplir con regulaciones de cumplimiento relacionada con seguridades en aplicativos webs, incluidas PCI DSS, DISA STIG, NIST 800-53, ISO 27K, OWASP e HIPAA.
- Antes de iniciar la fase de producción o poslanzamiento, se brinda la oportunidad de corregir vulnerabilidades en etapas tempranas del desarrollo.
- Minimizar la carga de trabajo del auditor.

En el componente Fortify WebInspect, se ingresa la URL donde se encuentra publicada o desplegada la aplicación, estas acciones realizadas se las puede observar en la Figura 2.17. Este componente se encarga de ejecutar las fases necesarias para el análisis; la primera es verificar la conectividad con la aplicación, para lo que se hizo el uso de una VPN proporcionada por la DGIP; seleccionar el tipo de análisis, en este caso se realizó el análisis de toda la aplicación; registrar inicios de sesión, en el caso del aplicativo del Sistema de Gestión del Sistema de Archivos se lo realiza a través de un login inicial; finalmente se realiza el análisis de las vulnerabilidades para la aplicación desplegada.



**Figura 2.17.** Interfaz de Fortify WebInspect para la carga del código fuente para el análisis

Una vez realizadas las fases necesarias para la ejecución de la prueba se muestran los resultados de este. En la Figura 2.18 se muestra la interfaz con los resultados obtenidos después de la ejecución de la prueba, en el cual se muestra cada una de las vulnerabilidades encontradas en análisis.



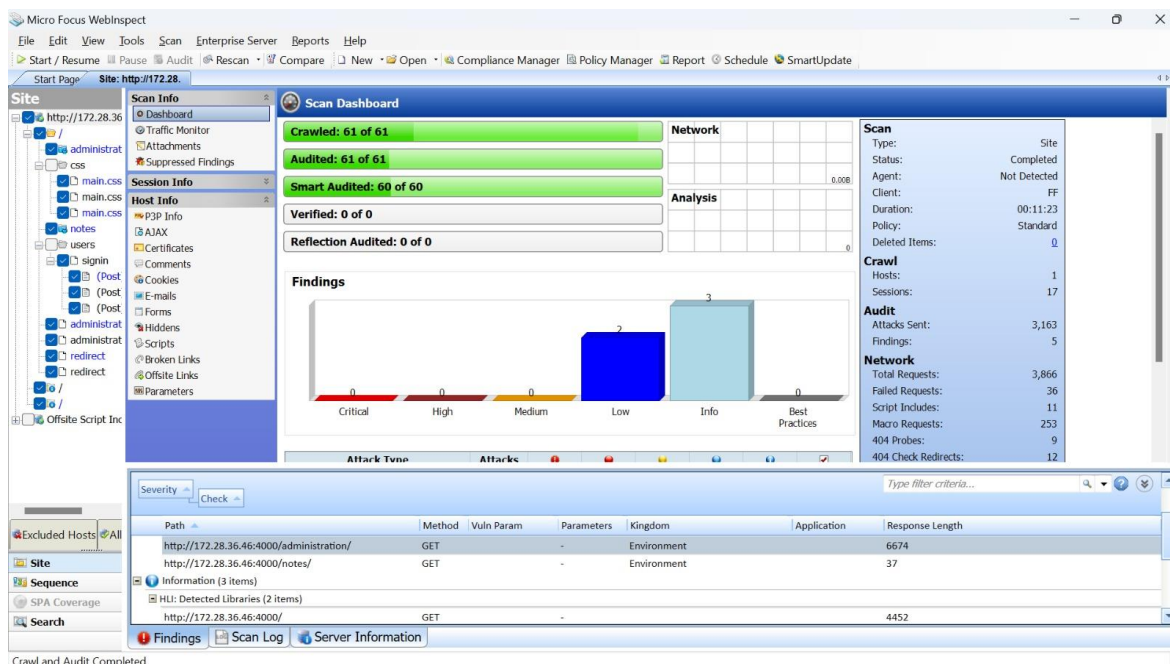


Figura 2.18. Interfaz de resultados del análisis DAST en Fortify WebInspect

## Rendimiento

Esta sección se detalla cómo se realizó las pruebas de carga y estabilidad sobre el aplicativo **Sistema de Gestión del Sistema de Archivos**, en donde se busca determinar el comportamiento del sitio web ante una carga concurrente de usuarios basados en el escenario definido por el equipo de desarrollo, dicha prueba nos permite obtener tiempos de respuesta, número de transacciones, métricas de rendimiento que servirán de insumo para actividades de optimización y mejora continua sobre el sitio.

La herramienta utilizada para la prueba de carga y estabilidad en el Front End de la aplicación es LoadRunner Profesional de Opentext.

Para la ejecución de esta prueba se siguieron las siguientes fases:

1. Se decide llevar a cabo la ejecución de un escenario de prueba de carga y estabilidad en el entorno de desarrollo del aplicativo **Sistema de Gestión del Sistema de Archivos**. En la creación de este escenario, se lleva a cabo una estimación del número máximo de usuarios concurrentes, estableciendo un total de 80 **Vusers** (usuarios virtuales). La determinación de este número se realiza en colaboración con el equipo de desarrollo, basándose en la información recopilada acerca del potencial número máximo de usuarios.

El escenario planificado descrito en la Figura 2.19 tiene en consideración métricas y el análisis de los patrones de uso típico de los usuarios objetivos: personal

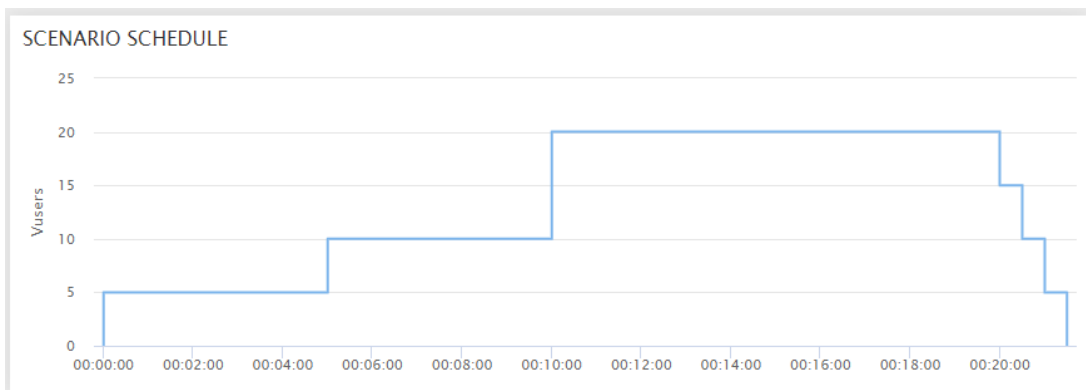
encargado de la carga y visualización de archivos en un tiempo aproximado de 5 minutos de acciones activas en el aplicativo web.

El escenario de carga y estabilidad a ser ejecutado está diseñado con base a la siguiente programación:

5 Vusers se conectan a los 0 minutos

+5 Vusers a los 5 minutos

+10 Vusers a los 10 minutos



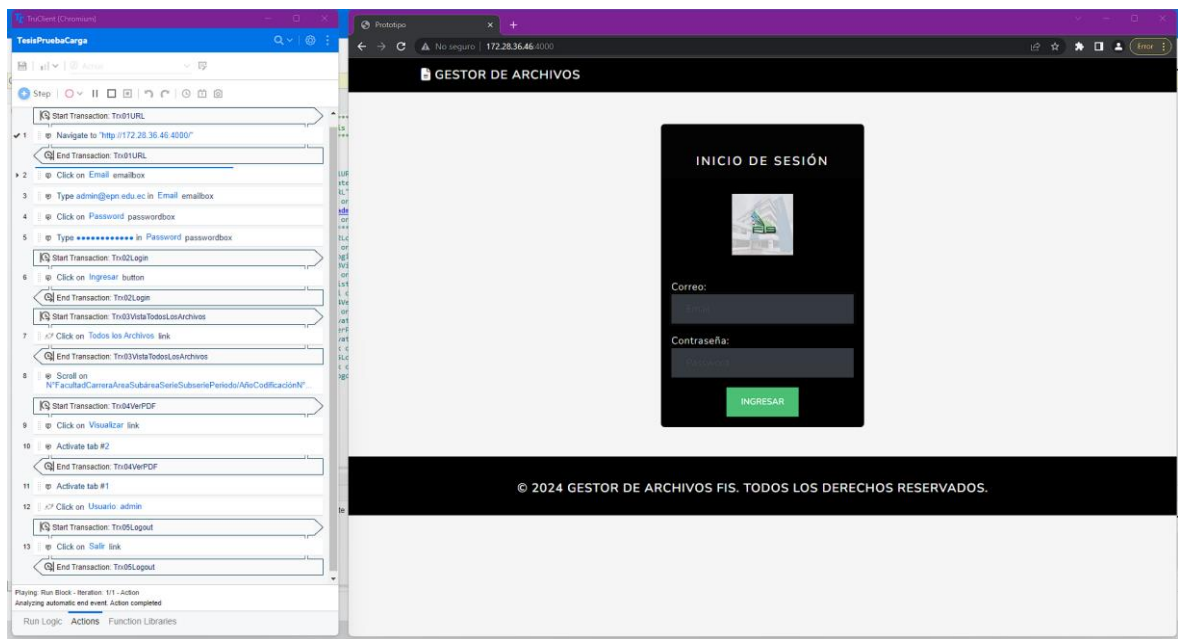
**Figura 2.19.** Escenario definido para la prueba de carga y estabilidad

2. Se realiza el grabado del script que simula las acciones de un usuario, en donde se realizan exactamente cada una de las acciones que realiza desde inicio a fin. El perfil definido para esta prueba es la del usuario encargado de la carga y visualización de los documentos, en donde se busca determinar como la concurrencia puede afectar al rendimiento de cada una de las transacciones definidas como importantes. Estas transacciones están definidas en la Tabla 2.4. Transacciones definidas para medir tiempos, configuradas en el script de simulación, el cual será utilizado en la ejecución del escenario; cada una de estas transacciones se encuentran detalladas para su mejor entendimiento del flujo.

**Tabla 2.4.** Transacciones definidas para medir tiempos

Código de la Transacción	Descripción
Trx01URL	Ingreso a la URL del aplicativo web
Trx02Login	Ingreso al aplicativo con credenciales
Trx03VistaTodosLosArchivos	Vista de todos los archivos
Trx04VerPDF	Carga y visualización del PDF
Trx05Logout	Cierre de sesión del aplicativo web

En LoadRunner Professional, se realizó la grabación del script de prueba con la medición de cada uno de los pasos importantes, de forma que se incluyeron los parámetros que los users, que son los usuarios virtuales en este caso, usarían para hacer un login correcto y una simulación eficaz de cada uno de los pasos definidos en el script. La interfaz de LoadRunner Professional que se usó para establecer las transacciones y los campos necesarios en el proceso se muestra en la Figura 2.20.



**Figura 2.20.** Interfaz de grabación del script en LoadRunner Profesional

### 3 RESULTADOS Y DISCUSIÓN

A continuación, se presenta los resultados obtenidos durante el desarrollo del sistema en el transcurso de los seis sprints propuestos, incluyendo un sprint 0 que sirve como punto de preparación antes de comenzar con el desarrollo. Los módulos de inicio de sesión y gestión documental se desarrollaron paralelamente.

#### 3.1 Sprint 0

El sprint 0 permitió preparar el ambiente para comenzar con el desarrollo del sistema propuesto. Las actividades que corresponden al sprint 0 son:

- Creación de entorno de desarrollo
- Creación de espacio colaborativo y de versionamiento
- Instalación software para la gestión de la base de datos

- Definición de usuarios

### **Creación de entorno de desarrollo**

Para el desarrollo del aplicativo web, se hizo uso de Node.js como entorno de tiempo de ejecución

### **Instalación de herramientas de Versionamiento**

El repositorio GitHub se utilizará para el Versionamiento del desarrollo durante todos los sprints, además, servirá como espacio colaborativo con los desarrolladores para obtener todos los cambios que han realizado los colaboradores.

### **Creación de la base de datos**

El tipo de base de datos escogido para el desarrollo del sistema corresponde a una base de datos no relación, NoSQL, este tipo de base de datos utiliza colecciones en vez de tablas. El sistema de base de datos utilizado es MongoDB, el cual se ha instalado en el servidor en el cual se realizará el deploy del sistema desarrollado.

### **Definición de usuarios**

El presente sistema tendrá tres tipos de usuarios que manejaran el sistema

- **Administrador:** Podrá administrar a los usuarios y visualizar todos los documentos que se hayan subido al servidor.
- **Visualizador:** Podrá visualizar los documentos, pero no podrá realizar modificaciones en los mismos
- **Editor:** Podrá visualizar y editar los documentos subidos

## **3.2 Sprint 1**

Acorde a la planificación, durante el sprint 1 se realizó el desarrollo de:

- Creación de usuarios
- Creación y clasificación de documentos

### **Creación de usuarios**

La Figura 3.1 muestra el formulario de registro de usuario, el cual puede ser accedido únicamente por los administradores del sistema de gestión documental. Para llevar a cabo el registro de un usuario, se requiere disponer de un nombre, correo electrónico y contraseña, además de la elección de la facultad y rol que tendrá el usuario registrado.

Al guardar los valores del formulario, se emplean validaciones de HTML5 para asegurar que cada campo, incluidos el nombre de usuario y el correo, estén debidamente ingresados.

En el caso de la contraseña, se implementa una verificación personalizada, la cual exige que la contraseña tenga más de 8 caracteres, sin restricciones específicas sobre su contenido, ya sea números, letras o signos especiales.

Asimismo, durante el proceso de almacenamiento en la base de datos, se aplica un proceso de encriptación. Esta tarea se lleva a cabo mediante algoritmos específicos proporcionados por bibliotecas especializadas en encriptación y desencriptación. De esta manera, se garantiza la seguridad y confidencialidad de las contraseñas almacenadas en el sistema.

AGREGAR USUARIO

Nombre

Email

Contraseña

Confirmar contraseña

Facultad:

FIS

Rol

Administrador

Tipo de usuario

Editor

Visualizador

Estado

Activo

CANCELAR GUARDAR

**Figura 3.1.** Formulario de registro de usuario

### **Creación y clasificación de documentos**

La Figura 3.2 muestra el formulario de registro de nuevos documentos, a este formulario pueden acceder los usuarios registrados en el sistema de gestión documental, que no sean

administradores. Para guardar un documento es necesario contar con la clasificación del documento, el archivo PDF que se utilizará, el asunto y las respectivas observaciones. El sistema generara automáticamente una codificación en base a la facultad, departamento y año, acompañado de un dígito secuencial; esta codificación se realiza en base a lo descrito en la sección de Documentación.

**NUEVO DOCUMENTO**

Carrera:

Subárea:

Tipo de Documento:

Subtipo de Documento:

Período/Año:

Codificación:

Archivo PDF:  The\_Guide\_...Mockups.pdf

Número de Páginas:

Asunto:

Observaciones:

**Figura 3.2.** Formulario de registro de nuevos documentos

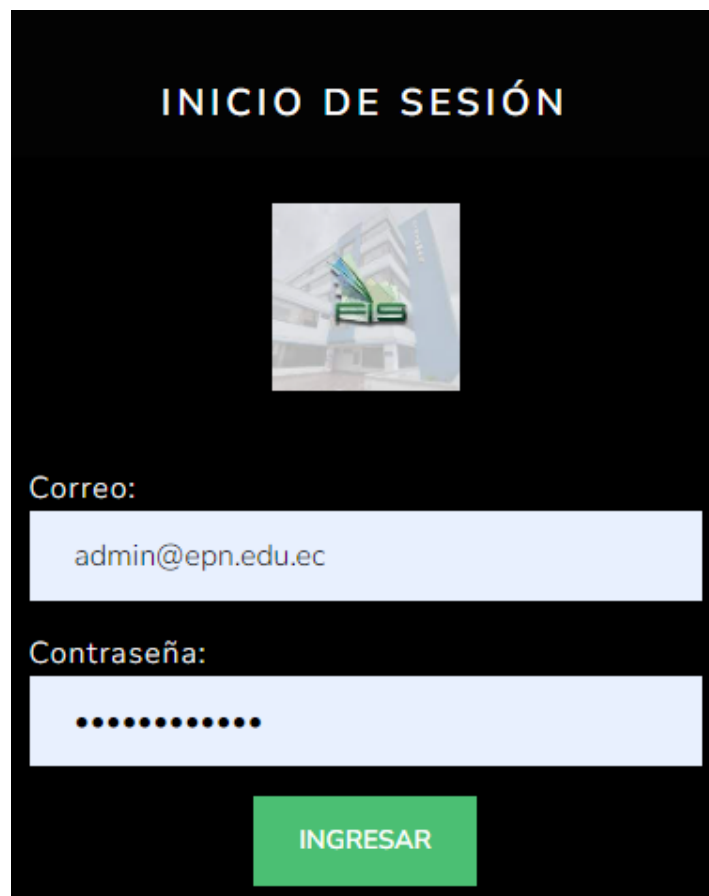
### 3.3 Sprint 2

Acorde a la planificación, durante el sprint 2 se realizó el desarrollo de:

- Inicio de sesión
- Gestión de documentos

### Inicio de sesión:

En la Figura 3.3 se visualiza el formulario de inicio de sesión, en este los usuarios registrados podrán iniciar sesión con sus correos y contraseñas. En el formulario de inicio de sesión se valida que exista el correo y que la contraseña sea la misma que la guardada en la base de datos. Una vez validado el correo y la contraseña, se verifica el rol que tiene el usuario. Durante la validación se verifica si es administrador o visualizador/editor, al usuario se le desplegará el módulo de usuarios en caso de ser administrador, por otra parte, el usuario visualizará el módulo de documentos en caso de ser visualizador/editor.



The image shows a login form with a black background. At the top, the text "INICIO DE SESIÓN" is displayed in white. Below this is a small square image of a modern building with a green and blue logo. The form contains two input fields: "Correo:" with the text "admin@epn.edu.ec" and "Contraseña:" with a masked password of ten dots. A green button labeled "INGRESAR" is positioned at the bottom center.

**Figura 3.3.** Formulario de inicio de sesión

### Gestión de documentos

En la Figura 3.4 y Figura 3.5 se visualiza el panel de administración de documentos, el cual puede ser accedido por los usuarios que sean visualizadores/editores y administradores. Estos usuarios podrán visualizar un listado de los documentos cargados en el sistema y correspondientes a su facultad y área, además, encontrarán las opciones para editar, visualizar o eliminar los documentos registrados. Para los administradores, podrán acceder

a todos los documentos sin importar a que facultad o área correspondan, pero no podrán editarse ni eliminarse.

HISTORIAL DE DOCUMENTOS												
Mostrar 10 registros											Buscar:	
N°	FACULTAD	CARRERA	ÁREA	SUBÁREA	SERIE	SUBSERIE	PERIODO/AÑO	CODIFICACIÓN	N° FOLIO	ASUNTO	OBSERVACIONES	ACCIONES
0	FIS	Computación	Decanato	Secretaria de Decanato	Consejo de Facultad	Actas/Resoluciones	2018	FIS-DC-2018-0001	2	asd	asdas	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a> <a href="#">VISUALIZAR</a>
1	FIS	Computación	Decanato	Secretaria de Decanato	Consejo de Facultad	Actas/Resoluciones	2018	FIS-DC-2018-0002	3	asdasd	asdasd	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a> <a href="#">VISUALIZAR</a>

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

**Figura 3.4.** Panel de gestión de documentos - Usuario visualizador/editor

HISTORIAL DE ARCHIVOS												
Mostrar 10 registros											Buscar:	
N°	FACULTAD	CARRERA	ÁREA	SUBÁREA	SERIE	SUBSERIE	PERIODO/AÑO	CODIFICACIÓN	N° FOLIO	ASUNTO	OBSERVACIONES	ACCIONES
0	FIS	Computación	Decanato	Secretaria de Decanato	Consejo de Facultad	Actas/Resoluciones	2018	FIS-DC-2018-0002	3	asdasd	asdasd	<a href="#">VISUALIZAR</a>
1	FIS	Computación	Decanato	Secretaria de Decanato	Consejo de Facultad	Actas/Resoluciones	2018	FIS-DC-2018-0001	2	asd	asdas	<a href="#">VISUALIZAR</a>

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

**Figura 3.5.** Panel de gestión de documentos - Usuario administrador

### 3.4 Sprint 3

Acorde a la planificación, durante el sprint 3 se realizó el desarrollo de:

- Gestión de usuarios
- Actualización de documentos

#### Gestión de usuarios

En la Figura 3.6 se visualiza el panel de administración de usuarios en un formato de tabla, este panel puede ser accedido únicamente por los administradores. En este panel los administradores podrán administrar a los usuarios teniendo la opción de actualizar los datos del usuario. Adicionalmente, como resultado de requerimientos para aspectos como auditorias, los usuarios no se pueden eliminar, únicamente se cambian de estado en activo o inactivo.



HISTORIAL DE USUARIOS CREADOS						
Mostrar <input type="text" value="10"/> registros				Buscar: <input type="text"/>		
N°	NOMBRE	FACULTAD	ROL	EMAIL	ACCIONES	
0	admin	FIS	Admin	admin@epn.edu.ec	<a href="#">EDITAR</a>	
1	Decanato	FIS	Decanato	decanato@epn.edu.ec	<a href="#">EDITAR</a>	
2	Subdecanato	FIS	Subdecanato	subdecanato@epn.edu.ec	<a href="#">EDITAR</a>	
3	Jefatura	FIS	Jefatura	jefatura@epn.edu.ec	<a href="#">EDITAR</a>	
4	Asistente-Decanato	FIS	Decanato	asistente.decanato@epn.edu.ec	<a href="#">EDITAR</a>	

Mostrando registros del 1 al 5 de un total de 5 registros

Anterior  Siguiente

**Figura 3.6.** Panel de gestión de usuarios - Usuario Administrador

### Actualización de documentos

En la Figura 3.7 se muestra el formulario de actualización de documentos, el cual puede ser accedido por los usuarios del sistema de gestión documental registrados como editores, sin embargo, no podrán modificar el archivo que haya sido cargado previamente en el sistema. Si se requiere una modificación del archivo, se debe solicitar al administrador de la base de datos.

**ACTUALIZAR DOCUMENTO**

Carrera:

Subárea:

Tipo de Documento:

Subtipo de Documento:

Perido/Año:

Codificación:

Asunto:

Observaciones:

X CANCELARGUARDAR

**Figura 3.7.** Formulario de actualización de documento

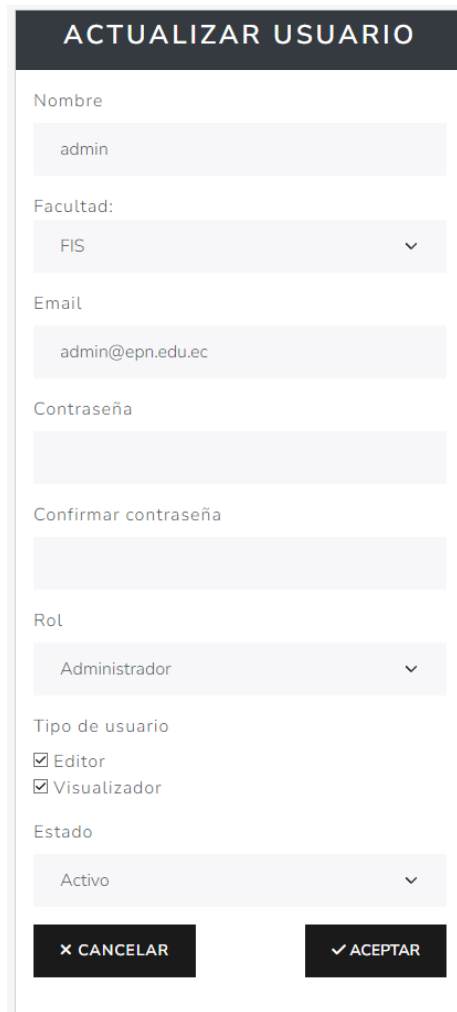
### 3.5 Sprint 4

Acorde a la planificación, durante el sprint 4 se realizó el desarrollo de:

- Actualización de usuarios
- Búsqueda de documentos

#### **Actualización de usuarios**

En la Figura 3.8 se visualiza el formulario de actualización de usuario, a este formulario pueden acceder únicamente los administradores del sistema de gestión documental. Si un usuario ya no debe tener permisos para acceder al sistema, los administradores no pueden eliminar a los usuarios, pero pueden deshabilitarlos para que no puedan volver a iniciar sesión. Al no eliminar a los usuarios registrados en el sistema se garantiza la consistencia de los datos registrados.



ACTUALIZAR USUARIO

Nombre  
admin

Facultad:  
FIS

Email  
admin@epn.edu.ec

Contraseña

Confirmar contraseña

Rol  
Administrador

Tipo de usuario  
 Editor  
 Visualizador

Estado  
Activo

× CANCELAR      ✓ ACEPTAR

**Figura 3.8.** Formulario de actualización de usuario

### **Búsqueda de documentos**

En la Figura 3.9 se visualiza la funcionalidad de búsqueda de documentos implementada en el panel de gestión documental. Los usuarios del sistema de gestión documental pueden realizar la búsqueda de los documentos cargados en el sistema haciendo uso de palabras claves que les sirva para encontrar los documentos cargados. La funcionalidad de búsqueda mostrará como resultado las filas en los que la palabra o palabras claves coincidan con una o varias columnas de la tabla que se visualiza.

HISTORIAL DE DOCUMENTOS												
Mostrar 10 registros												Buscar: 2023
N°	FACULTAD	CARRERA	ÁREA	SUBÁREA	SERIE	SUBSERIE	PERIODO/AÑO	CODIFICACIÓN	N° FOLIO	ASUNTO	OBSERVACIONES	ACCIONES
4	FIS	Software	Decanato	Secretaria de Decanato	Consejo de Facultad	Actas/Resoluciones	2023	FIS-DC-2023-0001	2	asdas	dasdas	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a> <a href="#">VISUALIZAR</a>
9	FIS	Computación	Decanato	Secretaria de Decanato	Consejo de Facultad	Actas/Resoluciones	2023	FIS-DC-2023-0002	2	asdasd	asdas	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a> <a href="#">VISUALIZAR</a>

Mostrando registros del 1 al 2 de un total de 2 registros (filtrado de un total de 10 registros)

Anterior 1 Siguiente

**Figura 3.9.** Cuadro de búsqueda de documentos

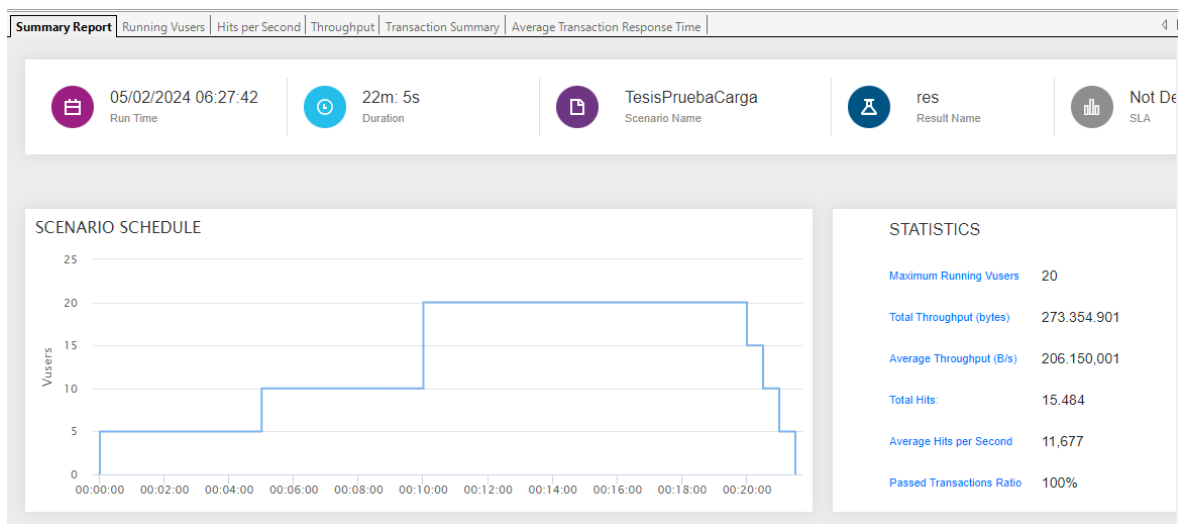
### 3.6 Sprint 5 – Pruebas

Durante el sprint 5 se realizó las pruebas de funcionamiento y usabilidad, en el caso de las pruebas de usabilidad se las realizo con los usuarios que utilizarán el sistema desarrollado. Con el feedback dado con los usuarios se realizó las correcciones necesarias antes del deploy en un servidor de la universidad.

Es fundamental abordar los resultados de las pruebas de carga, ya que estas han permitido a la organización determinar el comportamiento del aplicativo en situaciones de carga y evaluar su estabilidad en escenarios específicos donde los usuarios acceden de manera concurrente al aplicativo web.

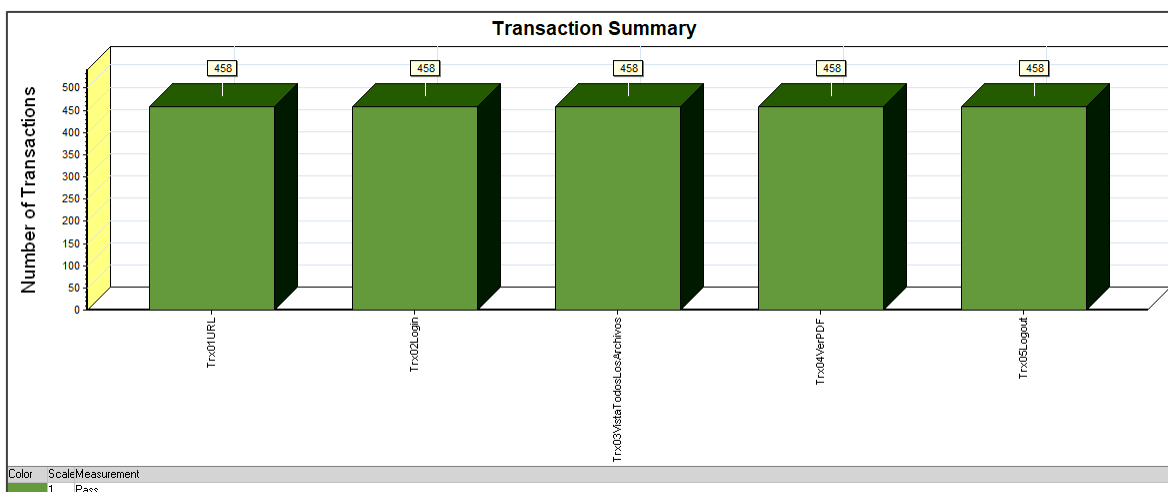
La ejecución del escenario se realizó el día 5 de febrero de 2024, con esta ejecución se buscó determinar tiempos de respuesta, número de transacciones, métricas de rendimiento y disponibilidad de las distintas transacciones descritas anteriormente.

Se muestra la Figura 3.10 con los detalles generales de la ejecución del escenario, dicha ejecución tuvo un tiempo de duración de 22 minutos y 5 segundos, llegando a un máximo de 20 usuarios virtuales.



**Figura 3.10.** Resultados de la ejecución de la prueba de carga

El número total de transacciones ejecutadas correctamente durante el tiempo total de la prueba se detalla en la Figura 3.11. En la gráfica presentada, se evidencia que todas las transacciones medidas han sido completadas exitosamente, totalizando 2290 pasos exitosos. Además de esta información, se exhibe la trazabilidad en las transacciones, apreciada mediante la igualdad en el número de transacciones individuales. Esta observación lleva a la conclusión de que ninguna de estas transacciones experimentó interrupciones en el flujo de trabajo.



**Figura 3.11.** Número total de transacciones realizadas

Respecto a los resultados en los tiempos de respuesta promedio en las transacciones obtenidos durante la prueba, se observa que todas las transacciones se encuentran en puntos óptimos, esto debido a que las transacciones no superan el 1 segundo de respuesta promedio como se observa en la Tabla 3.1.

**Tabla 3.1.** Tiempo de respuesta promedio de las transacciones

	<b>Tiempo de Respuesta (segundos)</b>
<b>Transacciones</b>	<b>Ejecución de 0-20 minutos</b>
Trx01URL	0.827
Trx02Login	0.373
Trx03VistaTodosLosArchivos	0.170
Trx04VerPDF	0.161
Trx05Logout	0.207

Como conclusión de esta prueba realizada, se confirma que el aplicativo se encuentra en óptimas condiciones y cuenta con la capacidad de soportar hasta un número de 20 usuarios concurrentes, estas condiciones también se encuentran sujetas a las especificaciones solicitadas para el alojamiento del sistema en un servidor de aplicación.

Como parte de las pruebas de seguridad, se presentan los resultados del escaneo de manera general en la Figura 3.12. En dicho informe se detallan las vulnerabilidades identificadas durante el análisis del código fuente de la aplicación Sistema de Gestión del Sistema de Archivos. Estas evaluaciones fueron llevadas a cabo mediante el uso del componente de la herramienta Fortify, utilizando el Audit Workbench, un módulo específicamente diseñado para llevar a cabo una detección exhaustiva de vulnerabilidades de seguridad. El enfoque adoptado se centra en cumplir con los requisitos más rigurosos, generando informes de cumplimiento que se ajustan a estándares ampliamente reconocidos en el mercado, tales como OWASP, CWE, MISRA, NIST, PCI, CERT, entre otros. La Tabla 3.2 proporciona un resumen de la ejecución del análisis SAST.

**Tabla 3.2.** Resumen de la ejecución del análisis SAST

<b>Descripción</b>	<b>Valor</b>
Tipo de análisis	Estático, SAST
Líneas de Código (LOC)	30.806
Número de archivos	435
Fecha del Análisis	Feb 4, 2024
Problemas encontrados	7

Fortify Security Report		opentext
Issue Count by Category		
Issues by Category		
Privacy Violation		6
Insecure Transport		1

**Figura 3.12.** Vulnerabilidades encontradas en el análisis SAST con la herramienta Fortify

Los problemas encontrados se refieren a una violación a la privacidad lo que nos indica que el archivo users.controller.js maneja mal la información confidencial en la línea 104, lo que puede comprometer la privacidad del usuario y, a menudo, es ilegal. Esta violación ocurre porque la información privada del usuario ingresa al programa y estos datos se encuentran en una ubicación externa como la consola. El detalle obtenido por la herramienta se presenta en la Figura 3.13.

users.controller.js, line 104 (Privacy Violation)			
<b>Fortify Priority:</b>	Critical	<b>Folder</b>	Critical
<b>Kingdom:</b>	Security Features		
<b>Abstract:</b>	The file users.controller.js mishandles confidential information on line 104, which can compromise user privacy and is often illegal.		
<b>Source:</b>	users.controller.js:111 Read password() 109           email, 110           rol, 111           password, 112           confirm_password, 113           _id		
<b>Sink:</b>	users.controller.js:104 ~JS_Generic.render() 102 103           if (errors.length > 0) { 104            res.render("users/editusers", { 105            //Devuelve los errores establecidos 106            errors,		

**Figura 3.13.** Detalle de la violación de privacidad encontrada en el código

La siguiente vulnerabilidad encontrada se refiere a la llamada a listening () en index.js en la línea 6 utiliza un protocolo inseguro en lugar de un protocolo seguro para comunicarse con el servidor. Esto se refiere a que toda la comunicación a través de HTTP, FTP o Gopher no está autenticada ni cifrada. Por lo tanto, está sujeto a compromisos, especialmente en el entorno móvil donde los dispositivos se conectan con frecuencia a redes inalámbricas públicas no seguras mediante conexiones WiFi. El detalle obtenido por la herramienta se presenta en la Figura 3.14.

index.js, line 6 (Insecure Transport)			
Fortify Priority:	Critical	Folder	Critical
Kingdom:	Security Features		
Abstract:	The call to listen() in index.js on line 6 uses an insecure protocol instead of a secure protocol to communicate with the server.		
Sink:	index.js:6 FunctionPointerCall: listen()		
4	require("./database");		
5			
6	app.listen(app.get("port"), () => {		
7	console.log("Server on port:", app.get("port"));		
8	});		

**Figura 3.14.** Detalle de la violación de privacidad encontrada en el código

La segunda prueba seguridad, se describe los resultados del escaneo realizado en el Sistema de Gestión del Sistema de Archivos ya desplegado en el servidor de aplicación. Esto se realizó con el componente de la herramienta Fortify WebInspect que es un módulo diseñado para realizar una detección rigurosa de vulnerabilidades de seguridad. Este enfoque se centra en cumplir con los requisitos más estrictos, generando informes de cumplimiento que se alinean con estándares reconocidos en el mercado, como OWASP, CWE, MISRA, NIST, PCI, CERT, entre otros. En la Figura 3.15, se muestra el resumen del análisis DAST.

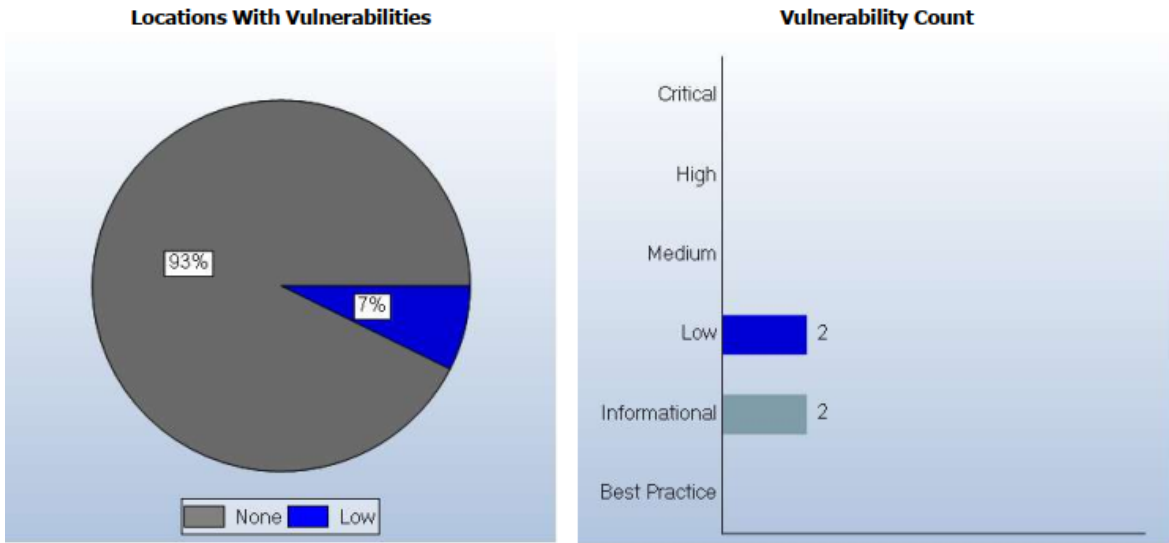
Scan Info						
Scan Name	# of Servers	Policy	Date	Duration	Vulns	
Site: http://172.28.36.46:4000/	1	Standard	02/04/24	11 mins	2	
Server Content						
Server	Cookies	Forms	Scripts	External Links	Broken Links	Server Types
http://172.28.36.46:4000	5	3	5	0	8	Java, Tomcat
<b>Total</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>0</b>	<b>8</b>	

**Figura 3.15.** Resumen del análisis DAST con Fortify WebInspect

En la Figura 3.16 se muestra la criticidad de las vulnerabilidades encontradas corresponden a las encontradas con respecto al lado del cliente



## Vulnerability Summary



## Vulnerability Count by Server

Server	Critical	High	Medium	Low	Info	BP
http://172.28.36.46:4000	0	0	0	2	2	0
<b>Total</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>0</b>

Figura 3.16. Dashboard de vulnerabilidades encontradas en el análisis DAST

### 3.7 Sprint 6 – Despliegue

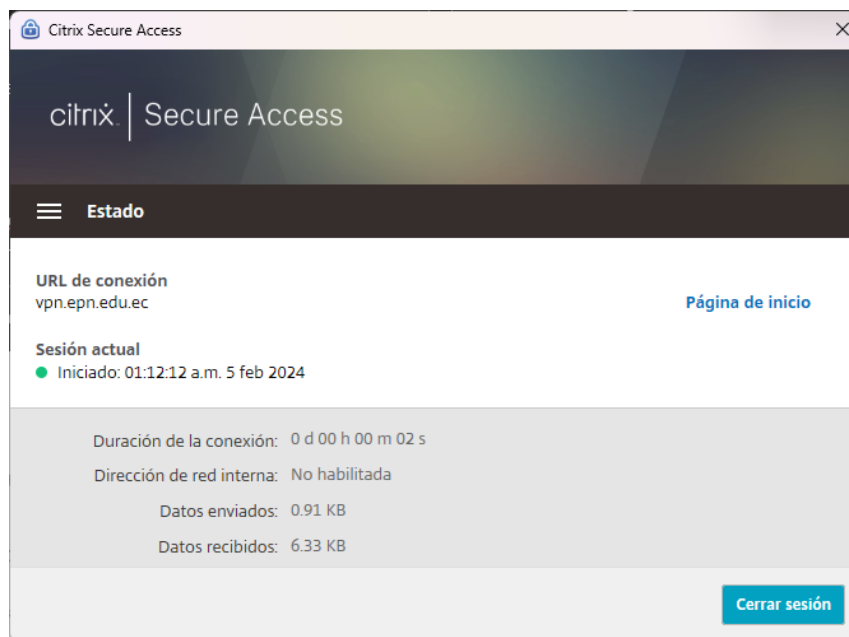
El despliegue del aplicativo se realiza en un servidor provisto por la Facultad de Ingeniería en Sistemas de la Escuela Politécnica Nacional, en la Figura 3.17 se puede visualizar las características del servidor utilizado para el despliegue del sistema. En este servidor se ha instalado la base de datos necesaria, junto con las herramientas necesarias para el funcionamiento.

## Device specifications

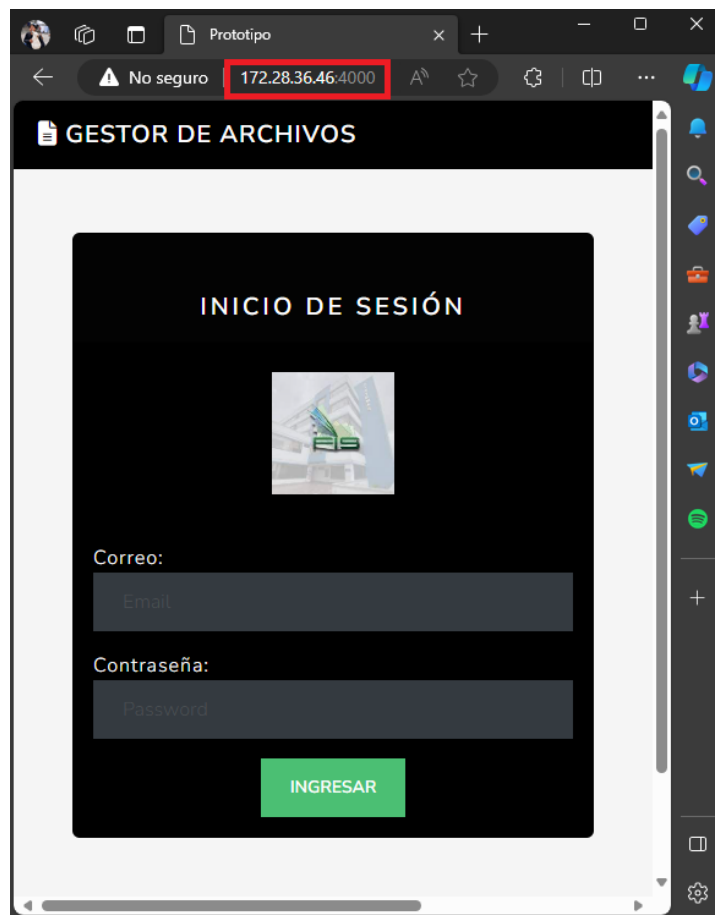
Device name	WIN-5NI96I27KTQ
Processor	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz 2.59 GHz (30 processors)
Installed RAM	12.0 GB
Device ID	BE491648-91B7-40F7-B6FD-257C02ABF83C
Product ID	00456-50926-18839-AA712
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

**Figura 3.17.** Especificaciones de servidor para despliegue de aplicativo

El aplicativo puede accederse desde el campus de la Escuela Politécnica Nacional haciendo uso de la IP del servidor, o desde el exterior al campus educativo haciendo uso del VPN “Citrix Secure Access” provisto por la CSIRT de la EPN, en la Figura 3.18 se puede visualizar la VPN utilizada para acceder al sistema. El acceso haciendo uso de la IP y puerto de acceso se puede visualizar en la Figura 3.19.



**Figura 3.18.** VPN de la Escuela Politécnica Nacional



**Figura 3.19.** Acceso al aplicativo con la IP del servidor y puerto de acceso.

## 4 CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

El aplicativo de gestión documental desarrollado cumple con todos los requerimientos establecidos durante la recopilación de requerimientos, el mismo permite gestionar usuarios registrados y documentos digitalizados al clasificarlos en base a los parámetros establecidos.

El sistema gestión documental está pensado para manejar un volumen grande de datos, el uso de una base de datos no relacional facilita la implementación del sistema desarrollado, al ser este tipo de base de datos fácilmente escalable y recomendado para manejar grandes volúmenes de datos debido a que los mismos no se almacenan en tablas si no en documentos.

Los resultados obtenidos en el análisis SAST nos pueden ayudar a guiar una planificación de acciones para abordar las vulnerabilidades detectadas, permitiendo una asignación más efectiva de recursos y esfuerzos para mejorar la seguridad general del aplicativo mediante

la corrección de las áreas más vulnerables. Es importante que este análisis detallado de cada una de las vulnerabilidades de seguridad encontradas se analice cuidadosamente, esto con el fin de obtener información correlacionada entre la vulnerabilidad identificada y el código fuente.

En el análisis realizado se obtiene información de tipos vulnerabilidades que son falsos positivos, esto porque se muestra nombres relacionados con las rutas declaradas para cada uno de los módulos, el ofuscamiento de estos nombres no garantiza una garantía mayor a la que ya se tiene con las seguridades proporcionadas a través de la protección de rutas propias del framework.

Respecto a la prueba de rendimiento, se evidencia que el tiempo de respuesta en todas las transacciones son óptimas, por lo que es un indicador positivo en el tiempo de respuesta por parte de estas transacciones. Es importante considerar que estos tiempos promedios se los obtuvieron al llegar hasta máximo de 20 usuarios concurrentes. Con estos resultados obtenidos, se concluye que el aplicativo está preparado para el uso concurrente con los tiempos y usuarios definidos en el escenario.

Finalmente, analizando los resultados de la prueba DAST, se observa que la ausencia de las vulnerabilidades críticas o altas es un indicador positivo para garantizar la solidez en las medidas de seguridad implementadas en el aplicativo. Adicional a esto, se tiene que las vulnerabilidades detectadas como informacionales son esenciales para mejorar la privacidad en la gestión de la información de aplicativo, esto es importante considerar debido a que el enfoque de este es sobre el trabajo para la gestión de archivos documentales y que tienen información propia de cada uno de los departamentos.

La combinación de Node.js y handlebars emerge como una poderosa sinergia que potencia la creación de sitios web interactivos. La versatilidad y eficiencia de Node.js en el lado del servidor, complementada por la simplicidad y elegancia de handlebars en la gestión de las vistas, proporcionan una base sólida para el desarrollo web moderno. Esta conjunción facilita la creación de experiencias interactivas dinámicas, permitiendo a los desarrolladores ofrecer a los usuarios una navegación fluida y enriquecedora.

La implementación del módulo de gestión documental en la Facultad de Sistemas ha demostrado ser un paso crucial hacia la modernización y eficiencia en la administración de información. Este sistema ha facilitado la organización, almacenamiento y recuperación de documentos de manera más ágil y estructurada. Al proporcionar una plataforma centralizada para la gestión documental, se ha mejorado la colaboración entre los diversos departamentos y se ha optimizado el flujo de trabajo. La trazabilidad y seguridad en el

manejo de documentos se han fortalecido, cumpliendo con estándares de calidad y proporcionando una respuesta efectiva a las necesidades administrativas de la facultad. En última instancia, este módulo contribuye significativamente a la eficacia operativa y a la toma de decisiones informada dentro de la Facultad de Sistemas.

## **4.2 Recomendaciones**

En el sistema, de ser necesario, se puede agregar campos adicionales para la codificación de los documentos. Actualmente se utiliza la facultad, área, período y un número secuencial para la codificación, sin embargo, se podría utilizar la subárea y subtipo de documento para facilitar la identificación de los documentos digitalizados.

Por temas de consistencia de los datos registrados, no se debe permitir eliminar usuarios o documentos registrados en la base de datos. Si un usuario no tiene acceso al sistema debe deshabilitarse y conservarse en la base de datos, lo mismo sucede con los documentos ingresados al sistema, estos solo deben marcarse como deshabilitados y no visibles para los usuarios, pero conservar los registros en la base de datos y los documentos en el servidor.

En caso de continuarse con el desarrollo del prototipo, se debe implementar la creación de logs en los que se registren los eventos importantes. Algunos de estos eventos importantes pueden ser: Creación de usuario, creación de documentos, actualización de documentos, actualización de datos de usuario; en estos logs se debe registrar el usuario que realizó los cambios correspondientes, para que en caso de modificaciones no autorizadas en los datos se pueda conocer al responsable de estas.

Se recomienda asegurar la capacitación del equipo de desarrollo en los diferentes estándares de referencia utilizados para el análisis estático del código (SAST). Esta acción permite reducir el número de defectos encontrados, lo que influye directamente en la categorización general del código. Adicional a esto, se debe garantizar la capacidad de mantenimiento del código mediante la refactorización, consolidando todas las secciones clonadas en una ubicación centralizada, como una nueva función, párrafo o método. Posteriormente, se debería invocar o llamar este código refactorizado en lugar de mantener múltiples instancias duplicadas. Esta práctica ayudará a reducir la duplicación, simplificará la gestión y facilitará futuras actualizaciones o modificaciones en el código.

Se recomienda ampliar la consideración de los escenarios durante las pruebas de rendimiento, centrándose en la garantía de un rendimiento óptimo del aplicativo bajo las condiciones específicas en las cuales se llevaron a cabo cada uno de los análisis y pruebas.

La importancia de esta recomendación radica en la necesidad de abordar de manera exhaustiva los distintos contextos operativos, cargas de trabajo y variaciones de tráfico que pueden influir significativamente en el comportamiento del sistema.

Al tener en cuenta una diversidad de escenarios representativos, se fortalece la confianza en la capacidad del aplicativo para mantener un desempeño eficiente y fiable, brindando así una visión más completa y realista de su capacidad para enfrentar situaciones diversas en entornos de producción.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] DINARDAP, «Norma de la digitalización de documentos de la DINARDAP,» DINARDAP, Quito, 2016.
- [2] Dirección de Gestión Documental y Archivo, *Norma de gestión documental para entidades de administración pública*, Quito, Pichincha: Presidencia de la Republica del Ecuador, 2016.
- [3] Azure DevOps, «Microsoft,» 11 Noviembre 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/devops/user-guide/services?view=azure-devops>. [Último acceso: 13 Febrero 2024].
- [4] Figma, «What is Figma?,» 27 Septiembre 2016. [En línea]. Available: <https://www.figma.com/prototyping/>. [Último acceso: 13 Febrero 2024].
- [5] MongoDB, «MongoDB,» MongoDB, 2007. [En línea]. Available: <https://www.mongodb.com/es>. [Último acceso: 13 Febrero 2024].
- [6] G. Bustos, «Hostinger,» Hostinger, 10 Enero 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-github>. [Último acceso: 13 Febrero 2024].
- [7] Citrix, «Citrix,» 25 Enero 2025. [En línea]. Available: <https://docs.netcaler.com/es-es/citrix-secure-internet-access.html>. [Último acceso: 22 Febrero 2024].
- [8] M. Palacio, «Scrum Manager,» Febrero 2022. [En línea]. Available: [https://www.scrummanager.com/files/scrum\\_master.pdf](https://www.scrummanager.com/files/scrum_master.pdf). [Último acceso: 1 Febrero 2024].
- [9] P. Deemer, G. Benefield, C. Larman y B. Vodde, «Información básica de SCRUM,» 2009. [En línea]. Available: [https://goodagile.com/scrumprimer/scrumprimer\\_es.pdf](https://goodagile.com/scrumprimer/scrumprimer_es.pdf).
- [10] T. Dimes, *Conceptos Básicos de Scrum: Desarrollo de Software Agile y manejo de proyectos Agile*, Babelcube Books, 2015.
- [11] A. Menzinsky, G. López, J. Palacio y M. Á. Sobrino, «Scrum Manager,» Agosto 2022. [En línea]. Available: [https://www.scrummanager.com/files/scrum\\_manager\\_historias\\_usuario.pdf](https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf).
- [12] B. Wake, «Exploring Extreme Programing,» 17 Agosto 2003. [En línea]. Available: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.

- [13] T. Stober y U. Hansmann, Agile Software Development, Best Practices for Large Software Development Projects, Alemania: Springer-Verlag, 2009.
- [14] R. Jeffries, «Essential XP: Card, Conversation, Confirmation,» Agosto 2001. [En línea]. Available: <https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>.
- [15] J. Cao, N. Khach y M. Ellis, The guide to mockups, UXPin, 2019.
- [16] G. Blandino, «Figma: qué es y cómo funciona,» 2023. [En línea]. Available: <https://www.pixartprinting.es/blog/figma-que-es/>.
- [17] S. Pinzón, R. Rodriguez y A. Vanegas, Java y el patrón Modelo-Vista-Controlador (MVC), Bogotá: Universidad Distrital Francisco José de Caldas, 2019.
- [18] E. López, A. Olivé, E. Mayol y C. Gómez, Diseño de sistemas software en UML, Barcelona: Universitat Politecnica de Catalunya. Iniciativa Digital Politecnica, 2004.
- [19] S. A. Pinzón Núñez , R. Rodríguez Guerrero y C. A. Vanegas, Java y el patrón Modelo-Vista-Controlador (MVC), Bogotá: Universidad Distrital Francisco Jose de Caldas, 2019.
- [20] Escuela Politécnica Nacional, «PROCEDIMIENTO DE GESTIÓN DE ARCHIVOS, CONFORMACIÓN DE EXPEDIENTES, INVENTARIO Y VALORACIÓN DOCUMENTAL,» Escuela Politécnica Nacional, Quito, 2019.
- [21] P. Kotler y K. Lane Keller, Dirección de Marketin, México: Pearson Education, 2012.

## 5 ANEXOS

### Anexo I. Historias de usuario

Enlace a las historias de usuario desarrolladas para la creación del presente Trabajo de Integración Curricular:

<https://www.dropbox.com/scl/fo/iibxblanc7vcejfsy02v1/h?rlkey=o2wxr8m0p2kce8zpgslmp7d4s&dl=0>

### Anexo II. Diseño de interfaces Figma

Enlaces al diseño de interfaces realizadas para el gestor de archivos institucionales para la Facultad de Ingeniería de Sistemas (FIS):

<https://www.figma.com/file/QiVPPkjhbm9AKpMXE2JuK3/PrototipoModuloAdminstrador?type=design&node-id=0%3A1&mode=design&t=GytmFnKzw2CCn2RL-1>

<https://www.figma.com/file/bfP0VCcGaM7K5ogKGJiUff/PrototipoModuloDecanato?type=design&mode=design&t=K8JLBgamqVRSzrxZ-1>

### **Anexo III. Macro de Excel**

Enlace de la macro de Excel que se utilizó como referencia para el desarrollo del prototipo:

<https://www.dropbox.com/scl/fo/6qayiqktj7dlhfj6zwrnm/h?rlkey=ukg9ginszk79rkxl0fwi1pdfp&dl=0>

### **Anexo IV. Cartas de confidencialidad de documentos y constancia de evidencia del prototipo**

Enlace a las cartas correspondientes:

<https://www.dropbox.com/scl/fo/cbshgjyikqvelk4wlngmk/h?rlkey=secyi8kvh7hr2kpfpi6igk99s&dl=0>

### **Anexo V. Código fuente del Prototipo**

Enlace del código fuente del prototipo del gestor de archivos institucionales para la Facultad de Ingeniería de Sistemas (FIS):

<https://github.com/MarlonP-28/TesisPrototipo>

### **Anexo VI. Video demostrativo**

En el siguiente enlace se encuentra un video en el cual se explican los pasos para instalar y ejecutar el sistema desarrollado en un equipo personal.

<https://www.dropbox.com/scl/fo/cwmghmbwjuergknvig2b2/h?rlkey=ynmnmgm47380fuzlkxvqu8vwrz&dl=0>