

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE
DELEGACIONES EN LA UNIDAD NACIONAL DE CIBERDELITOS**

FRONTEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

MARIA IVETH COCHA GER

DIRECTOR: VANESSA KATHERINE GUEVARA BALAREZO

DMQ, agosto 2024

CERTIFICACIONES

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

MARIA IVETH COCHA GER
maria.cocha@epn.edu.ec
iveth.cocha.2001@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **MARIA IVETH COCHA GER**, bajo mi supervisión.

VANESSA KATHERINE GUEVARA BALAREZO
DIRECTOR
vanessa.guevarav@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

MARIA IVETH COCHA GER

AGRADECIMIENTO

Gracias a mi madre por apoyarme y ser mi compañía en todo momento, a mi tía por sus palabras de motivación, a mi familia por todos los consejos y recomendaciones para ser una mejor persona. Quiero agradecer también a mis profesores que contribuyeron a mi formación académica.

Gracias también a mi chiquilla Mirka por ser mi compañía en las noches de desvelo.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
AGRADECIMIENTO	III
ÍNDICE DE CONTENIDO	IV
RESUMEN	VI
ABSTRACT	VII
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo General	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
2. METODOLOGÍA	6
2.1. Metodología de Desarrollo	6
Roles	7
Artefactos	8
2.2. Diseño de interfaces	10
Herramienta utilizada para el diseño- Figma	11
2.3. Diseño de la arquitectura	12
Patrón Arquitectónico	12
2.4. Herramientas de desarrollo	13
3. RESULTADOS	15
3.1 Sprint 0. Configuración del Ambiente de Desarrollo	15
3.2 Sprint 1. Resultados del diseño y desarrollo del flujo de autenticación y navegación	18
3.3 Sprint 2. Resultados del diseño y desarrollo del flujo Delegaciones	22
3.4 Sprint 3. Resultados del diseño y desarrollo del flujo Delitos	31
3.5 Sprint 4. Resultados del diseño y desarrollo del Módulo Buscar	32
3.6 Sprint 5. Resultados de las pruebas del frontend	33
4. CONCLUSIONES	37
5. RECOMENDACIONES	38
6. REFERENCIAS BIBLIOGRÁFICAS	39
7. ANEXOS	42

ANEXO I	43
ANEXO II	44
ANEXO III	65
ANEXO IV.....	66

RESUMEN

Con los avances tecnológicos, se ha implementado la automatización de procesos, logrando una significativa reducción en los tiempos de ejecución y la tasa de errores. Estos beneficios se han extendido no solo a industrias, sino también a actividades de la vida diaria, como la programación de citas online y otras aplicaciones similares.

Este trabajo se centra en desarrollar el componente de frontend de un Sistema de Delegaciones diseñado para la Unidad Nacional de Ciberdelitos, con el fin de facilitar la gestión de delitos. En comparación con el método manual actual, que ha enfrentado numerosos desafíos, el nuevo sistema automatizado está diseñado para corregir errores comunes, como la entrada incorrecta de datos, la colocación de números en campos inapropiados y la falta de llenado adecuado de los campos esenciales.

Para el desarrollo del sistema se usa React JS, Material IU, PrimeReact en la codificación y estilización de las interfaces de Usuario, y la metodología SCRUM para la etapa de planeación y entrega a tiempo de todos los requerimientos solicitados. Finalmente, para la parte de pruebas se recurre a Vitest y React Testing junto con Lighthouse.

El documento se organiza de la siguiente manera: el capítulo 1 presenta la descripción del componente frontend, el capítulo 2 discute la metodología empleada, el capítulo 3 expone los resultados obtenidos, y los capítulos 4 y 5 ofrecen las conclusiones y recomendaciones. Finalmente, se incluyen las referencias bibliográficas utilizadas y el apartado de anexos.

PALABRAS CLAVE: frontend, sistema, React JS, SCRUM , Vitest, Lighthouse.

ABSTRACT

With technological advancements, process automation has been implemented, achieving a significant reduction in execution times and error rates. These benefits have extended not only to industries but also to daily activities, such as online appointment scheduling and similar applications.

This project focuses on developing the frontend component of a Delegation System designed for the National Cybercrime Unit to facilitate crime management. Compared to the current manual method, which has faced numerous challenges, the new automated system is designed to correct common errors such as incorrect data entry, placing numbers in inappropriate fields, and failing to adequately fill essential fields.

React JS, Material UI, and PrimeReact are used for coding and styling the user interfaces, while the SCRUM methodology is applied for planning and timely delivery of all requested requirements. Finally, for testing, Vitest and React Testing along with Lighthouse are utilized.

The document is organized as follows: Chapter 1 presents the description of the frontend component, Chapter 2 discusses the methodology employed, Chapter 3 outlines the results obtained, and Chapters 4 and 5 provide the conclusions and recommendations. Finally, the references and the annex section are included.

KEYWORDS: frontend, system, React JS, SCRUM , Vitest, Lighthouse.

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La Unidad Nacional de Ciberdelitos, con el fin de automatizar la gestión de delitos, ha decidido incorporar un sistema que cumpla con este objetivo. Basado en esta necesidad, se plantea el desarrollo del frontend para el Sistema de Delegaciones.

El uso de un sistema implica la automatización y simplificación de varios procesos, optimizando así los recursos. Las interfaces gráficas de usuario (GUI) resultantes deben ser claras y concisas para garantizar que el usuario pueda realizar sus actividades dentro del sistema. Si el usuario no puede completar una acción, no entiende los pasos necesarios del proceso o encuentra el diseño poco atractivo, podría abandonar el uso del sistema. Para evitar esto, la información presentada en pantalla debe ser legible, fácil de comprender y mantener su estructura al ajustarse a diferentes resoluciones de pantalla [1].

Además de la parte visual, el componente frontend incluye la funcionalidad necesaria para consumir los endpoints del backend, manejando correctamente las alertas y validando los datos ingresados, como números, letras y caracteres especiales.

Para cumplir este propósito, se ha utilizado React JS en el desarrollo de este sistema, debido a su capacidad para crear interfaces de usuario dinámicas y eficientes, su facilidad para manejar el estado de la aplicación y su compatibilidad con la creación de componentes reutilizables. También se ha incorporado el uso de Material UI y PrimeReact, que ofrecen componentes listos para usar y personalizables, agilizando así el diseño de las interfaces [2].

De esta forma, se permite al usuario gestionar eficientemente la asignación de responsabilidades autorizadas en varios módulos:

- Módulo Delegaciones: Permite registrar, editar y observar la información existente.
- Módulo Delitos: Permite ingresar y eliminar información.
- Módulo Usuarios: Permite agregar, actualizar roles y eliminar usuarios.

Este sistema proporciona las herramientas necesarias para el desarrollo de actividades mediante campos de autocompletado y el uso de filtros en algunas tablas, simplificando así el proceso de búsqueda. En contraste con el método manual actual, donde la información puede dispersarse en múltiples documentos y repetirse, el sistema asegura una mayor consistencia de datos sin necesidad de una revisión estrictamente detallada.

Además, permite un acceso rápido y fácil a los datos relevantes en tiempo real, agilizando las operaciones diarias al eliminar la necesidad de buscar y recopilar información manualmente.

1.1 Objetivo General

Desarrollar un frontend de un sistema para la gestión de delegaciones para la Unidad de Nacional de Ciberdelitos

1.2 Objetivos específicos

1. Definir los requerimientos para el frontend del sistema.
2. Prototipar las interfaces de usuario en base de los requerimientos determinados.
3. Codificar las interfaces según los prototipos realizados.
4. Realizar pruebas para verificar el rendimiento y funcionamiento del frontend del sistema

1.3 Alcance

El frontend del Sistema de Delegaciones se centra en ofrecer interfaces amigables e intuitivas para la gestión de delegaciones. Este componente se ha desarrollado utilizando React JS, junto con bibliotecas para el estilo de los componentes, validaciones, formularios y seguridad. Además, se ha incorporado el principio de mínimo privilegio, donde cada rol establecido en el sistema tiene únicamente los privilegios esenciales para desempeñar su trabajo. Este modelo proporciona alta protección de datos, una correcta funcionalidad frente a fallas y una mayor resistencia ante actos maliciosos [3].

Además, para lograr el objetivo general de este proyecto, se ha optado por la metodología ágil SCRUM. Esta metodología permite recopilar los requisitos, definir roles y, especialmente, producir entregas incrementales en intervalos de tiempo cortos conocidos como Sprints. El sistema se estructura de la siguiente manera:

Roles

- Administrador
- Registrador
- Visualizador

Consumo de endpoints para todos los Roles

- Confirmar cuenta
- Crear contraseña de acceso
- Iniciar y cerrar Sesión
- Recuperar contraseña

El Rol Administrador consume endpoints para:

- Agregar una delegación
- Actualizar una delegación
- Ver el detalle de una delegación en específico
- Ver todas las delegaciones registradas
- Agregar un nuevo Registrador o Visualizador
- Actualizar el rol de un usuario
- Eliminar un usuario

El Rol Registrador consume endpoints para:

- Solicitar registro para acceder al sistema
- Agregar una delegación
- Actualizar una delegación
- Ver el detalle de una delegación en específico
- Ver todas las delegaciones registradas
- Agregar un delito
- Eliminar un delito
- Ver todos los delitos registrados

El Rol Visualizador consume endpoints para:

- Solicitar registro para acceder al sistema
- Buscar y visualizar una delegación por Número de Investigación o Número de Instrucción Fiscal.

1.4 Marco Teórico

Frontend

El Frontend, o interfaz de usuario, es la parte visible de un sistema informático, ya sea de hardware o software, que permite al usuario interactuar con el sistema [5]. Sin embargo,

su creación requiere un conocimiento profundo de la ingeniería de software y la aplicación de métodos de experiencia de usuario [6].

En el desarrollo web, el Frontend se construye generalmente utilizando tres lenguajes: HTML, CSS y JavaScript. Estos se organizan en tres archivos principales: un archivo HTML para el contenido, un archivo CSS para los estilos y un archivo JavaScript para las validaciones [4].

Ingeniería de software

La ingeniería de software es una rama de la ingeniería que abarca todos los aspectos de la creación de software, empleando modelos de sistemas, notaciones, reglas, recomendaciones de diseño y guías de procesos [7].

Según el tipo de software resultante, se pueden distinguir dos categorías principales:

- **Productos genéricos:** sistemas que se venden en el mercado y son de código abierto.
- **Productos personalizados:** sistemas que son diseñados según las peticiones realizadas por el cliente [7].

Independientemente de si el sistema que se encuentra en el mercado o se desarrolla para un cliente determinado, debe ser mantenible y escalable. Para garantizar estas cualidades, se establecen evaluaciones y pruebas del sistema creado.

Pruebas de interfaz de usuario

Para validar el correcto funcionamiento del sistema, es importante considerar tres aspectos principales: velocidad de funcionamiento (cómo responde el sistema a las acciones del usuario), facilidad de aprendizaje (cuánto tiempo le toma al usuario volverse productivo en el sistema) y robustez (la capacidad del sistema para tolerar errores del usuario) [7]. Las pruebas de la interfaz de usuario se enfocan en asegurar que la interfaz funcione adecuadamente desde un punto de vista técnico y cumpla con los estándares y requisitos establecidos por el usuario [8]. Las pruebas clave a realizar incluyen:

- **Pruebas unitarias:** Se centran en probar cada uno de los módulos o clases del programa de forma independiente, verificando que cada componente funcione correctamente por sí solo [8].
- **Pruebas de integración:** Consisten en integrar los módulos o clases que ya han sido probados de forma independiente en las pruebas unitarias, centrándose en evaluar la correcta interacción entre ellos y las interfaces que los conectan [8].

- **Pruebas de rendimiento:** Miden la velocidad y la capacidad de respuesta del sistema ante diversas operaciones del usuario [8]. Estas pruebas pueden incluir:
 - Pruebas de Volumen: Evaluación de la capacidad del sistema para manejar grandes volúmenes de datos.
 - Pruebas de Resistencia: Pruebas del sistema en condiciones de sobrecarga o bajo estrés para identificar su capacidad de mantenerse operativo.
 - Pruebas de Carga: Evaluación del sistema en condiciones normales para verificar su comportamiento bajo la carga esperada.

El objetivo de estas pruebas es detectar errores y corregirlos antes de que el sistema entre en producción, asegurando que el producto final sea robusto, eficiente y cumpla con las expectativas de los usuarios.

Herramientas y Bibliotecas

En el desarrollo de páginas web con JavaScript, existen numerosas herramientas que potencian el desarrollo y la funcionalidad de las aplicaciones. Las bibliotecas y los frameworks en JavaScript facilitan la creación de sitios web y aplicaciones con características y funcionalidades muy variadas, gracias a las capacidades dinámicas, flexibles y atractivas de JavaScript [9].

Un framework es un conjunto de archivos y directorios que proporcionan una estructura predefinida para el desarrollo de aplicaciones. Estos frameworks incorporan funcionalidades ya desarrolladas y probadas, y ofrecen una base sobre la cual los desarrolladores pueden construir sus aplicaciones utilizando un determinado lenguaje de programación [10].

Por otro lado, las bibliotecas son colecciones de rutinas, procedimientos, clases, y otros recursos que pueden ser reutilizados en diferentes programas. Las bibliotecas permiten a los desarrolladores aprovechar soluciones existentes para tareas comunes sin tener que reinventar la rueda [11].

2. METODOLOGÍA

El estudio de casos es una metodología particularmente adecuada para ciertos tipos de problemas, especialmente cuando la investigación y la teoría están en sus fases preliminares, o para abordar problemas prácticos delicados donde las experiencias de los participantes y el contexto de la situación son fundamentales [12]. La investigación a través de estudios de casos ofrece una metodología rica en información, permitiendo al investigador obtener una comprensión profunda de la problemática y explorar en detalle las necesidades y soluciones [13].

En el contexto del desarrollo de software, el investigador asume el rol de diseñador y codificador del componente de frontend, aplicando conocimientos teóricos y prácticos para crear una solución efectiva. Para este proyecto, se ha optado por un enfoque de estudio de casos con el objetivo de abordar las necesidades específicas de los usuarios en relación con la gestión de delegaciones.

Este sistema de gestión de delegaciones se diseña y desarrolla teniendo en cuenta las características y requisitos específicos de los usuarios, con el objetivo de brindar una solución efectiva y satisfactoria para sus necesidades de gestión.

Este sistema de gestión de delegaciones se diseña y desarrolla teniendo en cuenta las características y requisitos específicos de los usuarios, con el objetivo de brindar una solución efectiva y satisfactoria para sus necesidades de gestión. A través de este enfoque, se busca una comprensión detallada de las problemáticas que enfrentan los usuarios y se desarrolla una solución que responda de manera precisa a sus requerimientos.

2.1. Metodología de Desarrollo

Las metodologías para el desarrollo de software son enfoques sistemáticos diseñados para mantener el orden y control de los componentes resultantes en el producto final. Estas metodologías definen artefactos, roles y actividades, y establecen prácticas y técnicas recomendadas para guiar el proceso de desarrollo del software. Existen dos categorías principales de metodologías de desarrollo de software: tradicionales y ágiles [14]. Estas metodologías se implementan en función de las necesidades específicas de cada proyecto. Sin embargo, en los últimos tiempos, las metodologías ágiles han ganado una mayor popularidad. Esto se debe a que están orientadas a proyectos que requieren soluciones a medida, ofreciendo una alta flexibilidad y adaptabilidad sin comprometer la calidad del producto final [15].

Entre las metodologías ágiles, SCRUM se destaca como una de las más efectivas para gestionar proyectos complejos y dinámicos. En SCRUM, la organización del desarrollo del proyecto se realiza mediante la división del trabajo en ciclos de trabajo cortos denominados Sprints. Cada Sprint tiene una duración fija y relativamente breve, durante la cual se planifican, desarrollan y entregan incrementos del producto [14].

Para el desarrollo de este Trabajo de Integración Curricular, se ha optado por utilizar la metodología SCRUM. A continuación, se detallan los roles y artefactos fundamentales que se han implementado en el proyecto, proporcionando una visión clara de cómo se organiza y gestiona el trabajo.

Roles

SCRUM tiene tres roles principales que colaboran para alcanzar los objetivos del proyecto, los cuales son:

Cliente (Product Owner)

El Product Owner es el representante de todas las partes interesadas en los resultados del proyecto. Su principal responsabilidad es crear y mantener el Product Backlog para alcanzar los objetivos del producto o proyecto. El Product Owner establece un calendario de entregas y trabaja en estrecha colaboración con el equipo para planificar, revisar y detallar los objetivos de cada Sprint [14].

Facilitador (Scrum Master)

El Scrum Master actúa como un facilitador y líder del equipo. Su rol es asegurarse de que el equipo SCRUM siga las prácticas y principios de SCRUM, eliminando obstáculos que puedan afectar el progreso del equipo. El Scrum Master también es responsable de garantizar que el Product Backlog esté priorizado y listo para la próxima iteración, asegurando así que el equipo pueda entregar resultados de manera efectiva [14].

Equipo (Development Team)

El Development Team es el grupo de personas que trabaja conjuntamente para desarrollar el producto del proyecto. En un proyecto típico, este equipo está compuesto por varios miembros con habilidades diversas, aunque en algunos casos puede ser una sola persona, como en este proyecto. El equipo es responsable de seleccionar los requisitos del Product Backlog que se comprometen a completar en cada Sprint, estimar la complejidad de los requisitos, y llevar a cabo el diseño, desarrollo y pruebas para garantizar la calidad del software [15].

En la **Tabla 2. 1** se muestra la asignación de roles siguiendo la metodología SCRUM

Tabla 2. 1 Asignación de Roles en la Metodología SCRUM

Rol	Nombre
Product Owner	Tcnl. Héctor García
Scrum Master	Ing. Vanessa Guevara
Team	María Iveth Cocha Ger

Artefactos

En SCRUM, los artefactos son elementos clave que el equipo y las partes interesadas utilizan para definir el producto en desarrollo, planificar las acciones necesarias para su producción, y gestionar las tareas del proyecto. Los principales artefactos en SCRUM son el Product Backlog, el Sprint Backlog y los incrementos. Cada uno cumple una función específica en el ciclo de vida del proyecto [16].

Recopilación de Requerimientos

La recopilación de requerimientos es una fase crucial en el desarrollo de software que se enfoca en definir el propósito del sistema. Durante este proceso, el cliente, los desarrolladores y los usuarios colaboran para identificar los problemas a resolver y especificar un sistema que satisfaga estas necesidades [15].

La **Tabla 2. 2** muestra el formato utilizado para la Recopilación de Requerimientos y en el **ANEXO II** se muestra la tabla completa con todos los requerimientos.

Tabla 2. 2 Formato Recopilación de Requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
CATEGORÍA DE SISTEMA	ID	DESCRIPCIÓN DEL REQUISITO
Frontend	RR-001	<p>Todos los usuarios necesitan consumir varios endpoints para:</p> <ul style="list-style-type: none"> ● Confirmar cuenta ● Iniciar sesión ● Crear contraseña ● Cerrar sesión ● Recuperar contraseña

Historias de Usuario

Las historias de usuario son una herramienta fundamental en el desarrollo ágil de software, utilizadas para capturar los requisitos del sistema desde la perspectiva de los usuarios finales. Estas historias ayudan a entender las funcionalidades que el sistema debe ofrecer, de manera sencilla y directa [17]. La **Tabla 2. 3** muestra el formato utilizado para las Historias de Usuario y en el **ANEXO II** se muestran todas las tablas realizadas para las Historias de Usuario.

Tabla 2. 3 Formato Historia De Usuario

HISTORIA DE USUARIO	
Identificador: HU001	Usuario: Todos
Nombre de las Historia: Confirmar cuenta, iniciar sesión, cerrar sesión, crear contraseña y recuperar contraseña	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración: 1	
Responsable: Iveth Cocha	
Descripción: Para el acceso al sistema los usuarios Administrador, Registrador y Visualizador deben ingresar sus credenciales en el formulario de ingreso (correo y contraseña). Si el usuario olvida su contraseña, se le permite crear una nueva contraseña. Para acceder al sistema el usuario debe validar su cuenta primero	
Observación: Para la confirmación de cuenta llega un mensaje al correo registrado. El usuario debe llenar todos los campos para validar sus credenciales, si es la primera vez que el usuario ingresa al sistema se le presenta una ventana para crear su contraseña.	

Product Backlog

El Product Backlog es una lista priorizada de objetivos y requisitos que define la visión y las expectativas del cliente en relación con el producto o proyecto. Esta herramienta fundamental en SCRUM ayuda a guiar el desarrollo del producto, asegurando que se cumplan los objetivos y se entreguen los resultados esperados. El Product Owner es el responsable de crear y gestionar la lista con la ayuda del Scrum Master y el Development Team es quien proporciona el coste estimado de completar cada requisito [14].

La **Tabla 2. 4** muestra el formato utilizado para el Product Backlog y en el **ANEXO II** se muestra la tabla completa.

Tabla 2. 4 Formato Product Backlog

Product Backlog				
ID-HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Confirmar cuenta, iniciar sesión, cerrar sesión, crear contraseña y recuperar contraseña	1	Finalizado	Alta

Sprint Backlog

El Sprint Backlog es un artefacto que detalla las tareas específicas que el equipo se compromete a completar durante un Sprint. Este documento es creado y gestionado durante la reunión de planificación del Sprint (Sprint Planning) y juega un papel fundamental en la organización del trabajo del equipo para cumplir con los objetivos de la iteración [14].

La **Tabla 2. 5** muestra el formato utilizado para el Sprint Backlog y en el **ANEXO II** se muestra la tabla completa.

Tabla 2. 5 Formato Sprint Backlog

Sprint Backlog						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del Ambiente de Desarrollo	NA	NA	NA	<ul style="list-style-type: none"> • Establecer los requerimientos para el frontend. • Definir la estructura del proyecto frontend. • Establecer los roles para cada usuario. 	10H

2.2. Diseño de interfaces

La Interfaz de Usuario (IU) es el componente del software que permite a los usuarios interactuar con el sistema. Es la parte visible del software donde los usuarios pueden comprender, navegar y operar el producto. Con el creciente número de usuarios, tanto

técnicos como no técnicos, que necesitan interactuar con interfaces digitales, el diseño de IU se ha vuelto una disciplina fundamental en el desarrollo de software [18].

Existen tres puntos de vista distintos sobre una interfaz de usuario, cada uno con un modelo mental propio, desarrollado a partir de diferentes experiencias y necesidades: del usuario, del programador y del diseñador.

Al desarrollar una aplicación, es crucial no solo cumplir con los requisitos funcionales del cliente, sino también garantizar que la interfaz sea clara y evidente en cuanto a cómo la aplicación satisface esos requisitos. La interfaz debe facilitar una experiencia de usuario efectiva y satisfactoria [1].

El diseño de interfaces de usuario debe seguir una serie de principios para asegurar que el software sea efectivo, fácil de usar y accesible para todos los usuarios. A continuación, se presentan los principios clave para el diseño de IU:

- **Anticipación:** Las aplicaciones deben anticipar las necesidades del usuario, evitando que el usuario tenga que buscar información o funcionalidades [18].
- **Autonomía:** La interfaz y el entorno de trabajo deben estar a disposición del usuario, proporcionando un entorno flexible que facilite el aprendizaje y uso de la aplicación [18].
- **Objetos de Interfaz Humana:** Los objetos de interfaz deben ser comprensibles, estables y consistentes en su comportamiento y apariencia, garantizando que los usuarios puedan interactuar con ellos de manera efectiva [18].
- **Legibilidad:** La información en la interfaz debe ser fácil de localizar y leer para facilitar la usabilidad del sistema de software. Se debe utilizar una tipografía clara, un contraste adecuado entre texto y fondo, y una disposición lógica de la información [18].

A continuación, se detalla la herramienta seleccionada para el diseño de las interfaces en este proyecto.

Herramienta utilizada para el diseño- Figma

El prototipado es una etapa esencial en el diseño de Interfaces de Usuario (IU), permitiendo la creación de simulaciones detalladas del sistema que se va a construir. Este proceso evolutivo es crucial para validar los requisitos de diseño y asegurar que las soluciones propuestas cumplen con las expectativas de los usuarios [19].

Para el prototipado de las pantallas accesibles según el tipo de usuario (Administrador, Registrador y Visualizador) de este sistema, se ha utilizado Figma. Figma es una herramienta de diseño colaborativo que permite a los diseñadores crear, compartir y probar diseños para sitios web, aplicaciones móviles y otras experiencias digitales [20].

En la **Figura 2. 1** se muestra el prototipado de inicio de sesión del frontend para el sistema de delegaciones y en el **ANEXO II** se presentan los demás prototipos.



Figura 2. 1 Prototipo de inicio de sesión

2.3. Diseño de la arquitectura

La arquitectura de software de un sistema es el conjunto de estructuras fundamentales que permiten razonar sobre el sistema en su totalidad. Incluye los elementos de software, las relaciones entre estos elementos y las propiedades que definen su comportamiento [21]. En este contexto, la arquitectura de software abarca todos los componentes visibles e invisibles del sistema, tales como botones, tablas, íconos y otros elementos que facilitan la interacción y comunicación entre las diferentes tareas ejecutadas en la interfaz de usuario.

Patrón Arquitectónico

Uno de los patrones arquitectónicos más reconocidos y utilizados en el desarrollo de software es el Modelo-Vista-Controlador (MVC). Este patrón clasifica y organiza los componentes del sistema en tres partes fundamentales: Modelo, Vista y Controlador [22]. La principal ventaja del patrón MVC es que separa las responsabilidades entre la interfaz de usuario, la lógica de negocio y el acceso a datos, lo que facilita la gestión, el mantenimiento y la escalabilidad del sistema [22].

El patrón MVC está compuesto por tres componentes principales, cada uno con responsabilidades específicas:

- **Modelo:** este componente se encarga de manipular, gestionar y actualizar los datos de los datos de la aplicación. Su función incluye la consulta, búsqueda, filtrado y actualización de datos en la base de datos, si la aplicación utiliza una [23].
- **Vista:** la Vista se encarga de mostrar al usuario final las pantallas, ventanas, páginas y formularios que conforman la interfaz de usuario [23]. Desde la perspectiva del programador, este componente corresponde al frontend de la aplicación.
- **Controlador:** actúa como el intermediario entre el Modelo y la Vista, gestionando las instrucciones del usuario, procesándolas y actualizando el Modelo y la Vista en consecuencia [23].

En la **Figura 2. 2** se presenta el patrón de arquitectura del frontend.

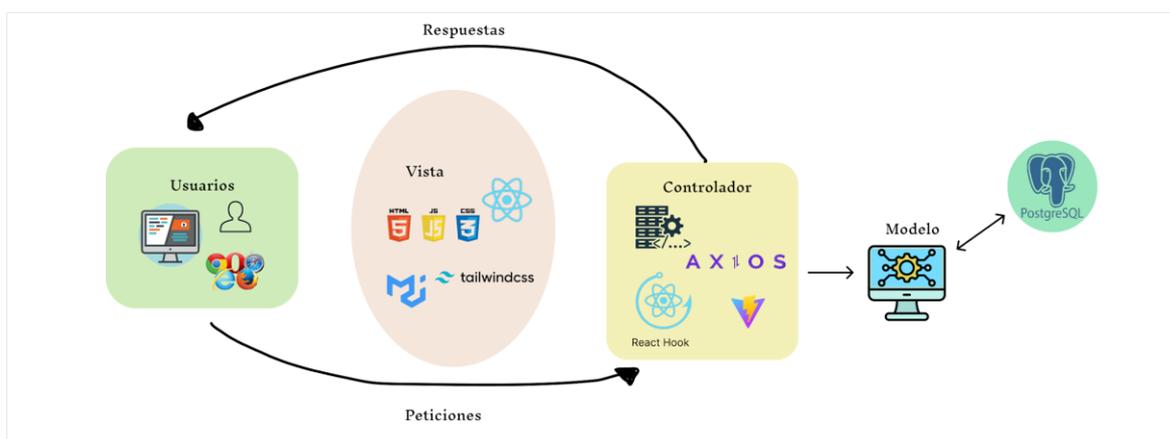


Figura 2. 2 Patrón de arquitectura del frontend

2.4. Herramientas de desarrollo

Las herramientas de desarrollo juegan un papel crucial en el ciclo de vida de un proyecto de software, ya que automatizan tareas repetitivas, facilitan la integración de componentes y optimizan diversos aspectos del proceso de desarrollo [24]. Su uso adecuado permite a los desarrolladores concentrarse en los aspectos críticos del desarrollo del sistema, asegurando una mayor eficiencia y calidad en el producto final. La descripción de las herramientas y la razón de su uso se detallan en la **Tabla 2. 6** que incluye las herramientas para el desarrollo del componente frontend.

Tabla 2. 6 Herramientas para el desarrollo del frontend

Herramienta	Justificación
Vite	Mejora el tiempo de inicio del servidor de desarrollo dividiendo primero los módulos de una aplicación en dos categorías: dependencias y código fuente. También viene con configuraciones predeterminadas listas para usar [24].
Tailwind CSS	Tailwind CSS opera analizando todos tus archivos HTML, componentes JavaScript y cualquier otra plantilla para encontrar nombres de clases, generando los estilos correspondientes y escribiéndolos en un archivo CSS estático. Es rápido, flexible y confiable, sin necesidad de tiempo de ejecución.
React Js	Toda aplicación web basada en React está compuesta por elementos reutilizables que forman componentes de la interfaz de usuario. React acelera la renderización sin recargar la página [25].
Material UI y PrimeReact	Ofrecen potentes herramientas y componentes para el desarrollo de interfaces de usuario en aplicaciones React, que ya tienen un estilo propio además de tener integrados funcionalidad en sus componentes [26] [27].
Vitest y React Testing Library	Vitest es un poco más rápido en la ejecución inicial de los tests unitarios [28]. React testing library. Es una biblioteca que permite testear componentes React proporcionando funciones para renderizarlos y buscar elementos por texto dentro del componente [28].
Axios	Es una biblioteca HTTP para el cliente que permite realizar solicitudes a un endpoint específico, ya sea una API externa o su propio servidor backend [29].

3. RESULTADOS

En el siguiente apartado se muestran los resultados obtenidos de la planificación del Product Backlog.

3.1 Sprint 0. Configuración del Ambiente de Desarrollo

Las actividades desarrolladas en el Sprint 0 son:

- Establecer los requerimientos del frontend
- Definir la estructura del proyecto
- Establecer los roles de usuario

Establecer los requerimientos del frontend

Consumo de endpoints para confirmar cuenta, iniciar sesión, cerrar sesión, modificar contraseña y recuperar contraseña para todos los roles

En el componente frontend, los usuarios deben confirmar su cuenta después de haber sido registrados en el sistema. Además, deben poder iniciar sesión para acceder al sistema. Una vez dentro, el usuario puede cerrar sesión. Si es la primera vez que el usuario accede al sistema, debe crear una contraseña. En caso de que el usuario olvide su contraseña, puede crear una nueva. Para llevar a cabo este proceso, se utilizan los endpoints del componente backend, proporcionando así la funcionalidad necesaria. El componente backend se encarga de crear un usuario Administrador por defecto la primera vez que se ejecuta. Este usuario tiene la posibilidad de asignar más usuarios.

Consumo de endpoints para solicitar registro y poder acceder al sistema

Los usuarios con roles de Registradores y Visualizadores deben llenar un formulario en el componente frontend. Este formulario incluye la cédula, el nombre del agente y el rol. La información recopilada se envía al correo del administrador del sistema mediante un endpoint del backend.

Consumo de endpoints para el Módulo Usuarios

El usuario Administrador puede agregar un nuevo usuario con un rol específico, actualizar el rol de un usuario existente y eliminar usuarios. Para agregar un nuevo usuario, el Administrador solo necesita ingresar la cédula o el nombre del agente. Si el agente existe en la lista de agentes registrados en el sistema, los campos de nombre, grado, cédula y email se completan automáticamente; de lo contrario, no. Esto se logra mediante el consumo de los endpoints del backend que proporcionan la funcionalidad de este módulo.

Consumo de endpoints para el Módulo Delegaciones

Los usuarios con roles de Administrador y Registrador pueden agregar, actualizar, buscar y ver el detalle de una delegación. Estas acciones se realizan consumiendo los endpoints proporcionados por el backend, lo que permite la gestión efectiva de las delegaciones.

Consumo de endpoints para el Módulo Delitos

Los usuarios Registradores pueden agregar y eliminar un delito. Dado que los delitos son de texto corto, son completamente visibles en la tabla que muestra todos los delitos registrados en el sistema. Estas acciones se realizan consumiendo los endpoints del backend, lo que proporciona la funcionalidad necesaria para este módulo.

Consumo de endpoints para buscar una delegación y ver el detalle de una delegación

El usuario Visualizador tiene acceso a la lista de todas las delegaciones registradas en el sistema. Dado que puede haber muchas delegaciones, debe poder buscar una delegación específica, para lo cual se consume un endpoint del backend. El usuario Registrador puede ver el detalle o la información de las delegaciones, lo cual también se realiza consumiendo los endpoints generados por el backend.

Definir la estructura del proyecto

El componente de frontend se ha desarrollado usando Visual Studio Code como entorno de desarrollo. Los archivos se han organizado según las necesidades, como páginas, formularios y otros componentes necesarios para el desarrollo del sistema. La **Figura 3. 1** muestra el esquema de estructura del proyecto.

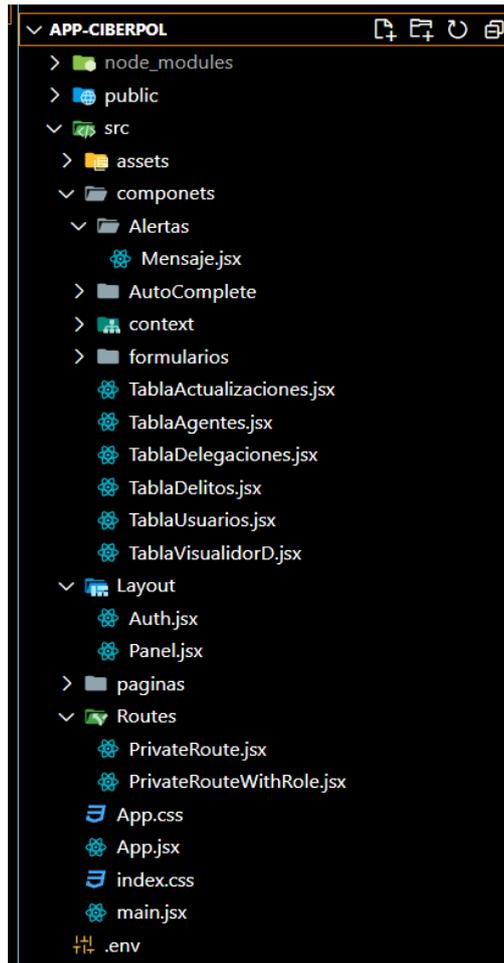


Figura 3. 1 Estructura general del Proyecto

Establecer los roles de usuario

En la **Figura 3. 2** se muestra los roles de usuario y las acciones que puede desempeñar cada uno de ellos



Figura 3. 2 Roles de Usuario

3.2 Sprint 1. Resultados del diseño y desarrollo del flujo de autenticación y navegación

En el Sprint 1 se realiza el diseño e implementación de las interfaces y consumo de endpoints para las siguientes actividades:

- Confirmar cuenta, iniciar sesión, cerrar sesión, crear contraseña y recuperar contraseña.
- Añadir, actualizar, y eliminar un Visualizador y/o Registrador.
- Solicitar registro para acceder al sistema.

Diseño e implementación de las interfaces y consumo de endpoints para confirmar cuenta, iniciar sesión, cerrar sesión, crear contraseña y recuperar contraseña

En el componente frontend, los usuarios deben haber confirmado su cuenta para poder crear su contraseña de acceso y así realizar actividades como el inicio de sesión, cierre de sesión y recuperación de contraseña. En la **Figura 3. 3** se muestra la pantalla de confirmación de cuenta. Después de confirmar la cuenta, el usuario debe crear una contraseña de acceso como se indica en la **Figura 3. 4**. Esta acción se ejecuta mediante un formulario en el que se debe ingresar la nueva contraseña dos veces para verificar su similitud.



Figura 3. 3 Página de Confirmación de Cuenta

Crear contraseña

Nueva Contraseña

Confirmar Nueva Contraseña

Crear Contraseña

Figura 3. 4 Formulario de creación de contraseña

Los usuarios deben ingresar su correo electrónico y contraseña para el inicio de sesión. En la **Figura 3. 5** se muestra el formulario para el inicio de sesión y la **Figura 3. 6** presenta la prueba unitaria correspondiente a esta funcionalidad.

Inicio de Sesión

Correo

Contraseña *****

Ingresar

Figura 3. 5 Formulario de Inicio de Sesión

```
RERUN src/test/Login.test.tsx x65
TAP version 13
✓ src/test/Login.test.tsx (2)
✓ Login (2)
  ✓ se renderiza el formulario de inicio de sesión
  ✓ El formulario actua como se espera
Test Files 1 passed (1)
Tests 2 passed (2)
Start at 00:36:46
Duration 366ms
PASS waiting for file changes...
press h to show help, press q to quit
```

Figura 3. 6 Prueba unitaria del formulario de inicio de sesión

Si el usuario olvida su contraseña, puede recuperarla en dos pasos. Primero, debe ingresar la dirección de correo registrado en el sistema, como se indica en la **Figura 3. 7**. Luego, recibe una notificación en su correo electrónico registrado con un enlace para cambiar la contraseña, utilizando un formulario similar al de crear la contraseña, como se indica en la **Figura 3. 8**.

Para cerrar sesión, el usuario debe hacerlo de forma manual, como se muestra en la **Figura 3. 9**. El detalle de las pruebas unitarias realizadas se encuentra en el **ANEXO II**.



Figura 3. 7 Recuperar contraseña (paso 1)



Figura 3. 8 Recuperar Contraseña (paso 2)



Figura 3. 9 Botón de cierre de sesión

Diseño e implementación de las interfaces y consumo de endpoints para añadir, actualizar, y eliminar un Visualizador y/o Registrador

El usuario Administrador tiene acceso a la opción de Usuarios, donde puede agregar un nuevo usuario, como se muestra en la **Figura 3. 10**. Además, el administrador puede observar el listado de todos los usuarios registrados mediante el consumo del endpoint del backend. En la **Figura 3. 11** se muestra el listado de usuarios registrados en el sistema. El

listado brinda acceso a las opciones de actualizar el rol del usuario, como se indica en la **Figura 3. 12**, y a eliminar al usuario como se muestra en la **Figura 3. 13**.

Cédula - Nombre

Cédula:

Nombre:

Correo:

Rol:

Figura 3. 10 Formulario para registrar Usuario

N°	Grado	Nombre y Apellido	Cédula	Rol	Acciones
1	POLL.	MARTHA LORENA	050111769	Registrador	 
2	POLL.	LUIS PAUL LOPES PERES	0400757530	Registrador	 
3	POLL.	MARIA IVETH COCHA GER	1728141571	Administrador	 

Figura 3. 11 Listado de usuarios

Cédula:

Nombre:

Correo:

Rol:

Figura 3. 12 Formulario actualizar rol usuario

Eliminar el usuario

Figura 3. 13 Eliminar usuario

Diseño e implementación de las interfaces y consumo de endpoints para solicitar registro para acceder al sistema

Mediante el frontend, se envía información al correo del administrador acerca de la solicitud de registro, como se indica en la **Figura 3. 14**, por medio del endpoint generado por el backend.



SOLICITUD DE REGISTRO PARA ACCEDER AL SISTEMA

Cédula

Nombres y Apellidos

Email

Mensaje (recuerde mencionar el rol)

Solicitar Registro

Figura 3. 14 Formulario de solicitud de registro

3.3 Sprint 2. Resultados del diseño y desarrollo del flujo Delegaciones

En el Sprint 2 se ha realizado la siguiente tarea:

- Diseño e implementación de interfaces y consumo de endpoints para mostrar las delegaciones existentes, ver el detalle de una delegación, actualizar y crear una delegación

Diseño e implementación de interfaces y consumo de endpoints para mostrar las delegaciones existentes, ver el detalle de una delegación, actualizar y crear una delegación.

Los usuarios Administrador y Registrador tienen acceso a la opción de delegaciones, donde puede agregar una nueva delegación, como se muestra desde la **Figura 3. 15** hasta la **Figura 3. 19** y la **Figura 3. 20** presenta la prueba unitaria correspondiente a esta funcionalidad.

ASIGANCIÓN DE LA INVESTIGACION

N° de Investigación Previa

N° de Instrucción Fiscal

Mes de ingreso de Disposiciones Fiscales -- Seleccione un mes -- v

Apellidos y Nombres del Agente

Grado Agente:

LOCALIZACIÓN

Realice la Búsqueda para la Localización

Cod Distrito:

Distrito:

Zona:

Cantón:

Provincia:

Figura 3. 15 Formulario agregar delegación (parte 1)

DELITO

Delito Tipificado en Delegación

Tipo Delito:

Delito Desagregación Policia Judicial:

Fecha de la Infracción/Delito

Apellidos y Nombres de la Victima

Sexo --Seleccione el sexo-- v

Edad

SOSPECHOSO

Llenar esta sección si existe en la boleta o en el proceso de Descargo

Apellidos y Nombres del Detenido o Sospechoso

Condición del Infractor Involucrado

Parentesco del Detenido o Sospechoso con la Victima

Alias del Sospechoso

Placas del Vehículo Involucrado en el Delito

Figura 3. 16 Formulario agregar delegación (parte 2)

FISCAL

Apellidos y Nombres del Fiscal

Buscar Fiscalía Fiscalía Nombre N° Fiscalía:

Unidad Especializada de Fiscalía:

Fecha de la Delegación

Fecha de Recepción en PJ

Fecha de Recepción por parte del Agente Investigador

N° de Oficio con la que recibe la Diligencia el Agente

Plazo Otorgado (Días)

N° art.444 COIP

Figura 3. 17 Formulario agregar delegación (parte 3)

CUMPLIMIENTO

¿Qué art. cumplió dentro del plazo?

Cumplimiento Parcial

Cumplimiento Total

Fecha de Cumplimiento o Descargo de Delegación

En Investigación

N° de Oficio de Descargo

Versiones (Número)

Reconocimiento de Lugar de los Hechos (Número)

¿Determinó posibles Responsables?

Comparecencia del Sospechoso

Peticiones a Fiscalía

Tipo de Requerimientos

Figura 3. 18 Formulario agregar delegación (parte 4)

RESULTADOS DE OPERATIVOS

Nombre del Requerido en la Boleta

Apellidos y Nombres de los Detenidos, producto del Cumplimiento de la Disposición Fiscal

N° Boletas Solicitadas

N° de Detenidos producto de la Investigación

N° de Allanamientos

Recuperación de Bienes o evidencias

N° de Recuperación de Automotores

N° de Recuperación Otros

N° de Notificaciones

N° de Citaciones

N° de Peritajes

N° de Traslados

INFORME

Informe o Descargo

Causas de Incumplimiento de la Investigación

Nombre de los Detenidos Producto de la Investigación

Observaciones

Cantidad Sustraida

Entidad Financiera

Agregar Delegacion

Figura 3. 19 Formulario agregar delegación (parte 5)

```

RERUN src/test/NuevaDelegacion.test.tsx x13
✓ src/test/NuevaDelegacion.test.tsx (4)
✓ Agregar Delegacion (4)
  ✓ Se muestra el formulario de agregar
  ✓ Funcionalidad de campos de autocompletado
  ✓ El boton realiza su accion
  ✓ Dialogo con el mensaje de confirmacion/error

Test Files 1 passed (1)
  Tests    4 passed (4)
  Start at 21:30:40
  Duration 630ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 3. 20 Prueba Unitaria del formulario agregar delegación

Además, se puede observar el listado de todas las delegaciones registradas mediante el consumo del endpoint del backend. En la **Figura 3. 21** se muestra el listado de las delegaciones y en la **Figura 3. 22** se indica la prueba unitaria de esta funcionalidad.

N° Orden Ingreso	Año	N° Investigación Previa	N° Instrucción Fiscal	Grado	Agente	Acciones
1	2024	123456776543211		CBOS.	VALDIVIEZO BORJA DIEGO PATRICIO	
2	2024	123443211234567		SGOP.	ARELLANO CATAGNA NELSON ORLANDO	

Figura 3. 21 Listado de Delegaciones

```

RERUN src/test/TablaDelegaciones.test.tsx x26

✓ src/test/TablaDelegaciones.test.tsx (4)
✓ Listar Delegaciones (4)
  ✓ Se muestra la tabla con el contenido de las delegaciones
  ✓ El icono actualizar lleva a Actualizar
  ✓ El icono detalle lleva a Visualizar
  ✓ Funciona el filtro de búsqueda

Test Files 1 passed (1)
Tests      4 passed (4)
Start at   16:21:41
Duration   570ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 3. 22 Prueba Unitaria listado de delegaciones

En el listado también se tiene la opción de actualizar la delegación mediante un formulario, como se aprecia desde la **Figura 3. 23** hasta la **Figura 3. 27**. La prueba unitaria correspondiente a la funcionalidad de actualizar delegación se muestra en la **Figura 3. 28**.

ASIGNACIÓN DE LA INVESTIGACION

N° de Investigación Previa: 123123123123456

N° de Instrucción Fiscal: [input type="text"]

Mes de ingreso de Disposiciones Fiscales: OCTUBRE

Apellidos y Nombres del Agente: MUÑOZ NAVARRETE RICARDO MARTÍN

Grado Agente: TNTE

LOCALIZACIÓN

Cod Distrito: 17D12

Distrito: NOROCCIDENTE

Zona: ZONA 2

Canton: SAN MIGUEL DE LOS BANCOS

Provincia: PICHINCHA

Figura 3. 23 Formulario Actualizar delegación (parte 1)

DELITO

Delito Tipificado en Delegación

Tipo Delito:

Tipo Desgregado:

Fecha de Infracción o Delito

Apellidos y Nombres de la Víctima

Sexo

Edad

Figura 3. 24 Formulario Actualizar delegación (parte 2)

SOSPECHOSO

Apellidos y Nombres del Detenido o Sospechoso

Condición del Infractor Involucrado

Parentesco del Detenido o Sospechoso con la Víctima

Alias del Sospechoso

Placas del Vehículo Involucrado en el Delito

FISCAL

Apellidos y Nombres del Fiscal

Unidad Especializada de Fiscalía:

Fecha de la Delegación

Fecha de Recepción en CIBERPOL

Fecha de Recepción por parte del Agente Investigador

N° de Oficio con la que recibe la Diligencia el Agente

Plazo Otorgado (Días)

N° art-444 COIP

Figura 3. 25 Formulario Actualizar delegación (parte 3)

CUMPLIMIENTO

¿Qué art. cumplió dentro del plazo?

Cumplimiento Parcial

Cumplimiento Total

Fecha de Cumplimiento o Descargo

En Investigación

N° de Oficio de Descargo

Versiones

Reconocimiento de Lugar de los Hechos

¿Determinó posibles Responsables?

Comparecencia del Sospechoso

Peticiones a Fiscalía

Tipo de Requerimientos

Figura 3. 26 Formulario Actualizar delegación (parte 4)

Figura 3. 27 Formulario Actualizar Delegación (parte 5)

```

RERUN src/test/ActualizarDelegacion.test.tsx x16
✓ src/test/ActualizarDelegacion.test.tsx (3)
✓ Visualizar Deleegacion (3)
  ✓ Se renderiza al formulario Actualizar Delegación
  ✓ El boton realiza su función
  ✓ Apertura del cuadro del dialogo con el mensaje esperado

Test Files  1 passed (1)
  Tests     3 passed (3)
Start at    11:43:14
Duration    622ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 3. 28 Prueba Unitaria formulario Actualizar Delegación

En el módulo Delegación también se tiene la opción para visualizar la delegación, como se muestra desde la **Figura 3. 29** hasta la **Figura 3. 33**. En la **Figura 3. 34** se muestra la prueba unitaria correspondiente. El detalle de las pruebas realizadas se encuentra el **ANEXO II**.

ASIGNACIÓN DE LA INVESTIGACION

N° de Investigación Previa 123456776543211

N° de Instrucción Fiscal No Registra

Mes de Ingreso de Disposiciones Fiscales: MARZO

Apellidos y Nombres del Agente VALDIVIEZO BORJA DIEGO

Grado Agente CBOS.

LOCALIZACIÓN

Cod Distrito: No Registra

Distrito: No Registra

Zona: No Registra

Canton: No Registra

Provincia: No Registra

Figura 3. 29 Formulario visualizar Delegación (parte 1)

DELITO

Delito Tipificado en Delegación No Registra

Tipo Delito: No Registra

Tipo Desgregado: No Registra

Fecha de Infracción o Delito No Registra

Apellidos y Nombres de la Victima No Registra

Sexo No Registra

Edad 22

SOSPECHOSO

Apellidos y Nombres del Detenido o Sospechoso No Registra

Condición del Infractor Involucrado No Registra

Parentesco del Detenido o Sospechoso con la Victima No Registra

Alias del Sospechoso No Registra

Placas del Vehiculo Involucrado en el Delito No Registra

Figura 3. 30 Formulario visualizar delegación (parte 2)

FISCAL

Apellidos y Nombres del Fiscal No Registra

Unidad Especializada de Fiscalía: No Registra

Fecha de la Delegación No Registra

Fecha de Recepción en CIBERPOL No Registra

Fecha de Recepción por parte del Agente Investigador No Registra

N° de Oficio con la que recibe la Diligencia el Agente No Registra

Plazo Otorgado (Días) No Registra

N° art.-444 COIP No Registra

Figura 3. 31 Formulario visualizar delegación (parte 3)

BOLETA

¿Qué art. cumplió dentro del plazo? No Registra

Cumplimiento Parcial No Registra

Cumplimiento Total No Registra

Fecha de Cumplimiento o Descargo No Registra

En Investigación No Registra

N° de Oficio de Descargo No Registra

Versiones No Registra

Reconocimiento de Lugar de los Hechos No Registra

¿Determinó posibles Responsables? No Registra

Comparecencia del Sospechoso No Registra

Peticiones a Fiscalía No Registra

Tipo de Requerimientos No Registra

Figura 3. 32 Formulario visualizar delegación (parte 4)

OBSERVACIONES

Nombre del Requerido en la Boleta No Registra

Apellidos y Nombres de los Detenidos, producto del Cumplimiento de la Disposición Fiscal No Registra

N° Boletas Solicitadas No Registra

N° de Detenidos producto de la Investigación No Registra

N° de Allanamientos No Registra

N° de Recuperaion de Bienes No Registra

N° de Recuperación de Automotores No Registra

N° de Recuperación Otros No Registra

N° de Notificaciones No Registra

N° de Citaciones No Registra

N° de Peritajes No Registra

N° de Traslados No Registra

INFORME

Informe o Descargo No Registra

Causas de Incumplimeto de la Investigación No Registra

Nombre de los Detenidos Producto de la Investigación No Registra

Observaciones No Registra

Cantidad Sustraída No Registra

Entidad Financiera No Registra

Figura 3. 33 Formulario visualizar delegación (parte 5)

```
RERUN src/test/DetalleDelegacion.test.tsx x8
✓ src/test/DetalleDelegacion.test.tsx (1)
✓ Visualizar Delelegacion (1)
  ✓ Se renderiza a Visualizar Delegación
Test Files 1 passed (1)
Tests      1 passed (1)
Start at   23:50:25
Duration   626ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 3. 34 Prueba Unitaria formulario visualizar delegación

3.4 Sprint 3. Resultados del diseño y desarrollo del flujo Delitos

En el Sprint 3 se ha realizado la siguiente tarea:

- Diseño e implementación de interfaces y consumo de endpoints para agregar, eliminar y ver los delitos.

Diseño e implementación de interfaces y consumo de endpoints para agregar, eliminar y ver los delitos.

El usuario Registrador tiene acceso a la opción de Delitos, donde puede agregar un nuevo delito, como se muestra en la **Figura 3. 35**. Además, puede observar el listado de todos los delitos registrados mediante el consumo del endpoint del backend, como se indica en la **Figura 3. 36**. También dispone de la opción de eliminar el delito, como se detalla en la **Figura 3. 37**.



Formulario para agregar un delito. Incluye dos campos de texto: "Delito" y "Sección", y un botón "Agregar".

Figura 3. 35 Formulario añadir delito

Delito		Seccion		Borrar
Buscar Delito		Buscar		
Abandono de persona		Delitos contra la integridad personal		
Abigeato		Delitos contra el derecho a la propiedad		
Abolición y suspensión de derechos de persona protegida		Delitos contra personas y bienes protegidos por el Derecho Internacional Humanitario		
Aborto con muerte		Delitos contra la inviolabilidad de la vida		
Aborto consentido		Delitos contra la inviolabilidad de la vida		
Aborto no consentido		Delitos contra la inviolabilidad de la vida		
Aborto no punible		Delitos contra la inviolabilidad de la vida		
Abstención de la ejecución de operaciones en comoción interna		Delitos contra la seguridad pública		
Abuso de arma de fuego		Delitos contra la seguridad pública		
Abuso de confianza		Delitos contra el derecho a la propiedad		

Figura 3. 36 Listar delitos

Eliminar el delito

Figura 3. 37 Eliminar delito

3.5 Sprint 4. Resultados del diseño y desarrollo del Módulo Buscar

En el Sprint 4 se ha realizado la siguiente tarea:

- Diseño e implementación de interfaces y consumo de endpoints para buscar y visualizar una delegación

Diseño e implementación de interfaces y consumo de endpoints para buscar y visualizar una delegación.

En el frontend, se consumen los endpoints del backend para listar las delegaciones, como se muestra en la **Figura 3. 38**. A través de este endpoint, el usuario Visualizador puede realizar la búsqueda de delegaciones utilizando los campos de Número de Instrucción fiscal o Número de Investigación Previa. Como se observa en la **Figura 3. 38**, el Visualizador solo tiene la capacidad de ver el detalle de una delegación, similar a lo que se hace en los roles de Administrador y Registrador.

Nº Orden	Año Ingreso	Nº Investigación Previa Buscar	Nº Instrucción Fiscal Buscar	Grado	Agente	Acciones
1	2024	123456776543211		CBOS.	VALDIVIEZO BORJA DIEGO PATRICIO	
2	2024	123443211234567		SGOP.	ARELLANO CATAGNA NELSON ORLANDO	

Figura 3. 38 Listado de Delegaciones para rol Visualizador

3.6 Sprint 5. Resultados de las pruebas del frontend

Para el Sprint 5 se ha realizado las siguientes tareas:

- Pruebas Unitarias
- Pruebas de Rendimiento
- Pruebas de Aceptación

Pruebas Unitarias

Las pruebas unitarias tienen como objetivo dividir el código del sistema en bloques más pequeños y verificar que cada uno de estos bloques funcione correctamente de manera aislada [30].

Las pruebas unitarias se han realizado solamente en el módulo de delegaciones, ya que es el componente más relevante del sistema, gestionando un mayor flujo de información. Además, los otros módulos comparten funcionalidades comunes (crear, actualizar, listar y visualizar). También se incluyó la página de inicio de sesión en las pruebas, debido a que es un componente diferente y controla el acceso de los usuarios al sistema.

En la **Figura 3. 39** se muestra la estructura del código para una prueba unitaria de un componente, mientras que en la **Figura 3. 40** se presenta el resultado de la prueba ejecutada. El detalle de todas las pruebas unitarias desarrolladas se encuentra en el **ANEXO II**.

```
describe('Login', () => {
  //imitar funcionalidad
  let mockSetAuth;
  let mockNavigate;
  beforeEach(() => {
    | mockSetAuth = vi.fn(); //limpiar cada proceso de prueba
  });

  it('se renderiza el formulario de inicio de sesión', () => {
    render(
      <Router>
      | <AuthContext.Provider value={{ setAuth: mockSetAuth }}>
      | | <Login />
      | </AuthContext.Provider>
      </Router>
    );
    const button = screen.getAllByRole('Ingresar');
    expect(button).toBeDefined();
  });

  it('El formulario actua como se espera', async () => {
    const emailInput = screen.getByLabelText('Correo');
    const passwordInput = screen.getByLabelText('Contraseña');
    const button = screen.getByText('Ingresar');
    await userEvent.type(emailInput, 'example@example.com');
    await userEvent.type(passwordInput, 'password123');
    act(() => {
      | fireEvent.click(button);
    })
    expect(mockSetAuth).toHaveBeenCalledTimes(1);
    expect(mockNavigate).toHaveBeenCalledTimes(1);
    expect(mockNavigate).toHaveBeenCalledWith('/delegaciones');
  });
});
```

Figura 3. 39 Código para la Prueba Unitaria

```
RERUN src/test/Login.test.tsx x65

TAP version 13
✓ src/test/Login.test.tsx (2)
  ✓ Login (2)
    ✓ se renderiza el formulario de inicio de sesión
    ✓ El formulario actua como se espera

Test Files 1 passed (1)
Tests      2 passed (2)
Start at   00:36:46
Duration   366ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 3. 40 Prueba Unitaria del componente

Las pruebas unitarias han revelado que todos los componentes del sistema funcionan según lo esperado. Esta fase de pruebas ha permitido detectar problemas de implementación, corregir errores y asegurar que cada unidad de código cumple con las especificaciones técnicas y los estándares de calidad requeridos.

Pruebas de Rendimiento

Las pruebas de rendimiento evalúan el producto en aspectos multidimensionales como velocidad, carga, tráfico y estrés, entre otros [31]. Para llevar a cabo este tipo de pruebas, se utiliza la herramienta Lighthouse de Google, la cual es automatizada y permite medir la calidad de una página web en términos de facilidad de uso y rendimiento [32].

En la **Figura 3. 41** se muestra los resultados obtenidos de la prueba de rendimiento realizada con Lighthouse. Cada prueba realizada proporciona parámetros consistentes, y el valor de estos parámetros puede variar en función de la página analizada.

El detalle de todas las pruebas de rendimiento desarrolladas se encuentra en el **ANEXO II**.



Figura 3. 41 Prueba de Rendimiento del componente

Para interpretar los resultados es necesario saber que:

- **Rendimiento:** evalúa la rapidez con la que se carga la página web y la velocidad con la que los usuarios pueden acceder a sus contenidos [33].
- **Accesibilidad:** examina la facilidad de uso del sitio web para personas que necesitan asistencia, revisando elementos como botones y enlaces para asegurarse de que están bien descritos [33].
- **Prácticas Recomendadas:** verifica si la página sigue los estándares modernos de desarrollo web, como cargar recursos desde servidores seguros con HTTPS, garantizar que todas las bibliotecas JavaScript sean seguras y libres de vulnerabilidades, asegurarse de que no haya errores de navegación, y evitar el uso de marcos y APIs obsoletos [33].
- **SEO:** verifica si la página web es compatible con dispositivos móviles, tiene datos estructurados válidos, incluye etiquetas de título y meta descripción, es indexable, y si su contenido no depende de plugins [33].

Los resultados de la prueba de rendimiento con Lighthouse muestran que, aunque el sistema está bien optimizado en términos de prácticas y SEO, hay áreas clave que requieren atención para mejorar su rendimiento y accesibilidad. Para mejorar el rendimiento se puede reducir el tamaño de los archivos CSS y JavaScript para mejorar los tiempos de carga. Así, como comprimir y ajustar las imágenes para que se carguen más rápido sin perder calidad. En el caso de la accesibilidad hay que asegurarse de que el contraste entre el texto y el fondo cumpla con las pautas de accesibilidad para mejorar la legibilidad del contenido.

Pruebas de Aceptación

Las pruebas de aceptación se enfocan en la validación del software por parte del cliente (Product Owner) para confirmar que el producto cumple con los requisitos definidos y está listo para su uso en producción [8]. Este proceso garantiza que el sistema cumple con las expectativas del cliente y está preparado para su implementación operativa. Durante estas pruebas, el cliente verifica que todas las funcionalidades y características del sistema estén completas y funcionen según los criterios acordados.

La **Tabla 3. 1** muestra el formato utilizado para las Pruebas de Aceptación. En el **ANEXO II** se encuentra todas las Pruebas de Aceptación.

Tabla 3. 1 Formato de Prueba de Aceptación

PRUEBA DE ACEPTACION	
Identificador: PA002	Identificador de HU: HU002
Nombre: Solicitar registro para acceder al sistema	
Descripción: Se debe llenar un formulario, cuyos datos son enviados al correo del Administrador.	
Pasos de Ejecución: <ul style="list-style-type: none"> ● Seleccionar la opción de solicitar registro que se muestra debajo del formulario de iniciar sesión ● Llenar el formulario con los datos solicitados ● Dar clic en solicitar registro 	
Resultado Deseado: El componente frontend permite solicitar registro para acceder al sistema	
Evaluación de la Prueba: El resultado esperado ha sido confirmado y ha obtenido una aprobación del 100%.	

Al finalizar las pruebas de aceptación, se confirma que todos los requisitos establecidos al inicio del proyecto han sido cumplidos de manera satisfactoria.

4. CONCLUSIONES

En esta sección se encuentran las conclusiones después de haber finalizado el Trabajo de Integración Curricular.

- Una definición precisa de los requerimientos para cada rol de usuario es fundamental para simplificar el proceso de codificación. Establecer una base clara sobre las expectativas del sistema facilita la estimación del tiempo y los recursos necesarios, optimizando la planificación del proyecto y reduciendo el riesgo de malentendidos y retrabajos.
- El prototipado de interfaces con Figma facilita la estructuración y alineación efectiva de los componentes en el diseño, al proporcionar una representación visual y funcional de las páginas antes de su codificación. Esto establece una base clara y coherente que optimiza el proceso de desarrollo, asegurando que la implementación final incluya requerimientos solicitados por el Product Owner.
- La planificación y organización al inicio del desarrollo son esenciales para el éxito del proyecto. Una planificación detallada asegura que cada Sprint se complete en el tiempo previsto, lo que contribuye a una gestión efectiva del tiempo y recursos, y facilita el cumplimiento de los plazos establecidos para el proyecto.
- El uso de React JS ha permitido la creación de interfaces de usuario dinámicas y eficientes. Este framework facilitó el desarrollo de componentes reutilizables y mejoró la experiencia del usuario mediante interfaces intuitivas, permitiendo a los usuarios realizar sus tareas de manera más fácil y eficiente.
- La integración de funcionalidades, como el consumo de endpoints del backend y la validación de datos, es importante para el correcto funcionamiento del sistema. Asegurar que estas funcionalidades estén implementadas adecuadamente garantiza la robustez del sistema y su capacidad para cumplir con los requisitos del proyecto.
- Las pruebas unitarias son fundamentales para verificar el comportamiento de los componentes y corregir errores tempranamente. Por otro lado, la herramienta Lighthouse proporciona una visión integral de la experiencia del usuario, permitiendo mejoras basadas en métricas objetivas como tiempos de carga y uso de recursos. Finalmente, las pruebas de aceptación garantizan que el software cumpla con las expectativas del cliente según los requisitos definidos.

5. RECOMENDACIONES

En esta sección se encuentran las recomendaciones para el presente trabajo.

- Los resultados de las pruebas de rendimiento han revelado que el marcador de accesibilidad se vio afectado por los colores empleados en el panel para las opciones de delegaciones, delitos y usuarios. Para mejorar la accesibilidad, se recomienda usar una paleta de colores con mayor contraste entre el texto y el fondo. Aunque se ha utilizado una paleta de colores definida por el Product Owner, es crucial considerar aspectos de accesibilidad en futuras versiones del sistema para cumplir con las pautas de WCAG (Web Content Accessibility Guidelines) y garantizar que todos los usuarios, incluidas personas con discapacidades visuales, puedan interactuar de manera efectiva con la interfaz.
- Se recomienda mantener las herramientas de desarrollo y las dependencias actualizadas para evitar errores al compilar el sistema. Las actualizaciones periódicas aseguran que se utilicen las versiones más recientes con correcciones de errores y mejoras de seguridad.
- Se recomienda incorporar pruebas de seguridad en el ciclo de desarrollo del sistema. Estas pruebas pueden incluir análisis de vulnerabilidades, pruebas de penetración y revisiones de seguridad del código para identificar y corregir posibles riesgos antes de la implementación en producción.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1] M. C. Albornoz, «Diseño de interfaz gráfica de usuario,» Octubre 2014. [En línea]. Available: <http://sedici.unlp.edu.ar/handle/10915/41578>. [Último acceso: mayo 2024].
- [2] React, «Una biblioteca de JavaScript para construir interfaces de usuario,» React.org, 2021. [En línea]. Available: <https://es.legacy.reactjs.org/>. [Último acceso: 26 Mayo 2024].
- [3] D. Morales, «TIPOS DE MECANISMOS PARA LA PROTECCIÓN DE LOS SERVICIOS INFORMÁTICOS Y SUS MODELOS DE SEGURIDAD,» [En línea]. Available: <https://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/4928/51257%20-%20Moreno%20Granados%20Diego.pdf?sequence=1&isAllowed=y>. [Último acceso: 26 Mayo 2024].
- [4] C. J. G. a. R. V. Rodríguez, «Editor web visual para HTML, CSS y JavaScript de apoyo a la docencia,» SABER. Revista Multidisciplinaria del Consejo de Investigación de la Universidad de Oriente, 2017. [En línea]. Available: <https://dialnet.unirioja.es/descarga/articulo/6047135.pdf>. [Último acceso: 26 Mayo 2024].
- [5] A. A. a. N. Zambrano, «IMPORTANCIA, PROBLEMAS Y SOLUCIONES EN EL DISEÑO DE LA INTERFAZ DE USUARIO,» 2024. [En línea]. Available: <https://www.redalyc.org/articulo.oa?id=427739430010>. [Último acceso: 26 Mayo 2024].
- [6] S. M. Labrada, «Principios del proceso de diseño de interfaz de usuario Revista Cubana de Transformación Digital,» 2020. [En línea]. Available: <https://rctd.uic.cu/rctd/article/view/96>. [Último acceso: 26 Mayo 2024].
- [7] O. Garcia, «Ingeniería del Software 7ma. Ed. - Ian Sommerville,» 2018. [En línea]. Available: https://www.academia.edu/34812442/Ingenieria_del_Software_7ma._Ed._-_Ian_Sommerville. [Último acceso: 26 Mayo 2024].
- [8] M. A. a. A. A. Daniel Bolaños, Pruebas de software y junit. un análisis en profundidad y ejemplos prácticos, PRENTICE HALL, 2007.
- [9] D. P. Acharya, «Las 40 mejores bibliotecas y frameworks de JavaScript para 2024,» Kinst, Abril 2021. [En línea]. Available: <https://kinsta.com/es/blog/bibliotecas-javascript/>. [Último acceso: 28 Mayo 2024].
- [10] acens, «Framework para el desarrollo ágil de aplicaciones,» Mar 2014. [En línea]. Available: <https://www.acens.com/comunicacion/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>. [Último acceso: 3 Mayo 2024].
- [11] M. L. y. A. H. González, «“Programación de Sistemas Bibliotecas,» [En línea]. Available:

https://www.uv.mx/personal/lizhernandez/files/2010/10/Bibliotecas_clase.pdf.
[Último acceso: 26 Mayo 2024].

- [12] T. V. Bonoma, «Case Research in Marketing: Opportunities, Problems, and a Process,» [En línea]. Available: <https://doi.org/10.2307/3151365>. [Último acceso: 02 Junio 2024].
- [13] V. Elizabeth, «El estudio de caso y su implementación en la investigación,» 2024. [En línea]. Available: http://scielo.iics.una.py/scielo.php?pid=S2226-40002012000100009&script=sci_arttext. [Último acceso: 2 Junio 2024].
- [14] E. M. y. J. Pacienza, «Metodologías de desarrollo de software,» 2015. [En línea]. Available: <https://repositorio.uca.edu.ar/handle/123456789/522>. [Último acceso: 02 Junio 2024].
- [15] A. O. y. M. Rojas, «Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo,» 2024. [En línea]. Available: <https://www.redalyc.org/articulo.oa?id=133115027022>. [Último acceso: 02 Junio 2024].
- [16] Atlassian, «Obtén más información sobre los artefactos del scrum ágil,» Atlassian, 2024. [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/artifacts#:~:text=del%20scrum%20%C3%A1gil%3F>. [Último acceso: 02 Junio 2024].
- [17] M. Cohn, «User Stories Applied for Agile Software Development,» 2012. [En línea]. Available: <https://www.mountaingoatsoftware.com/uploads/articles/User-Stories-Applied-Mike-Cohn.pdf>. [Último acceso: 02 Junio 2024].
- [18] L. S. y. M. Gómez, «Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación,» [En línea]. Available: <https://www.angelfire.com/journal2/lead/DIU.pdf>. [Último acceso: 02 Junio 2024].
- [19] Figma, «What is Figma?-Figma Learn - Help Center,» Figma Learn, 2019. [En línea]. Available: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>. [Último acceso: 02 Junio 2024].
- [20] Figma, «Figma: The Collaborative Interface Design Tool,» Figma, 2024. [En línea]. Available: <https://www.figma.com/>. [Último acceso: 02 Junio 2024].
- [21] P. V.-E. L. C. C. a. M. K. Humberto Cervantes Maceda, «Arquitectura de software,» 2016. [En línea]. Available: <https://dspace.itsjapon.edu.ec/jsui/bitstream/123456789/3811/1/Arquitectura%20de%20Software%20-%20Humberto%20Cervantes%20Maceda%20%28FreeLibros%29.pdf>. [Último acceso: 2 Enero 2024].
- [22] N. L. a. V. R. O. Danilo, «Análisis comparativo de Patrones de Diseño de Software,» 2022. [En línea]. Available: <https://doi.org/10.23857/pc.v7i7.4338>. [Último acceso: 26 Mayo 2024].
- [23] C. Giridhar, Learning Python Design Patterns, Packt Publishing Ltd, 2016.

- [24] ViteJS, «¿Por qué Vite?,» ViteJS, 2024. [En línea]. Available: <https://es.vitejs.dev/guide/why> . [Último acceso: 02 Junio 2024].
- [25] Tailwind CSS, «Installation - Tailwind CSS,» Tailwindcss.com, 2024. [En línea]. [Último acceso: 2 Junio 2024].
- [26] Material UI, «Overview - Material UI,» Mui.com, 2024. [En línea]. Available: <https://mui.com/material-ui/getting-started/>. [Último acceso: 2 Junio 2024].
- [27] PrimeReact, «PrimeReact | React UI Component Library,» Primereact.org, 2024. [En línea]. Available: <https://primereact.org/installation/> . [Último acceso: 2 Junio 2024].
- [28] J. F. Moreno, «Testing en React con Vitest,» Paradigmadigital.com, 11 Enero 2024. [En línea]. Available: <https://www.paradigmadigital.com/dev/testing-react-vitest/>. [Último acceso: 2 Junio 2024].
- [29] J. C. Guña, «Cómo usar Axios con React: La guía definitiva,» freeCodeCamp.org, 14 Julio 2022. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-usar-axios-con-react/> . [Último acceso: 9 Junio 2024].
- [30] R. B. Susana González, «Personalización del proceso de pruebas unitarias empleando la herramienta NUnit,» Abril 2016. [En línea]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=8590526>. [Último acceso: 21 Junio 2024].
- [31] S. & S. Y. K. Pradeep, «A Pragmatic Evaluation of Stress and Performance Testing Technologies for Web Based Applications,» 2019. [En línea]. Available: <https://ieeexplore.ieee.org/document/8701327>. [Último acceso: 21 Junio 2024].
- [32] A. FLORES, «Google Lighthouse, todo lo que debes saber,» CMA Comunicación, 2020. [En línea]. Available: <https://cmacomunicacion.com/google-lighthouse-todo-lo-que-debes-saber/>. [Último acceso: 21 Junio 2024].
- [33] T. Pol, «Google Lighthouse: Qué es y Cómo Utilizarlo,» Semrush Blog, 2023. [En línea]. Available: <https://es.semrush.com/blog/como-utilizar-google-lighthouse/>. [Último acceso: 26 Junio 2024].

7. ANEXOS

ANEXO I. Certificado de originalidad.

ANEXO II. Manual Técnico

ANEXO III. Manual de Usuario

ANEXO IV. Manual de Instalación

ANEXO I

F_AA_236

CERTIFICADO DE ORIGINALIDAD TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 22 de julio de 2024

De mi consideración:

Yo, VANESSA KATHERINE GUEVARA BALAREZO, en calidad de Director del Trabajo de Integración Curricular titulado DESARROLLO DE UN FRONTEND asociado DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE DELEGACIONES EN LA UNIDAD DE CIBERDELITOS elaborado por el estudiante MARIA IVETH COCHA GER de la carrera en TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta antiplagio "TURNITIN" para la revisión de originalidad del documento escrito (sin anexos) producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 7%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



VANESSA KATHERINE
GUEVARA BALAREZO

Vanessa Guevara

Docente Ocasional a Tiempo Completo

ESFOT

ANEXO II

Recopilación de Requerimientos

Los requerimientos definidos para el desarrollo del frontend del presente proyecto se encuentran en la **Tabla 1**.

Tabla 1 Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
CATEGORÍA DE SISTEMA	ID	DESCRIPCIÓN DEL REQUISITO
Frontend	RR-002	Los usuarios Registrador y Visualizador necesitan consumir el endpoint para: <ul style="list-style-type: none">• Solicitar registro para acceder al sistema.
	RR-003	Los usuarios Administrador y Registrador necesitan consumir varios endpoints para: <ul style="list-style-type: none">• Agregar una delegación• Actualizar una delegación• Ver el detalle de una delegación en específico.• Ver todas las delegaciones registradas
	RR-004	Los usuarios Administrador necesitan consumir varios endpoints para: <ul style="list-style-type: none">• Agregar un nuevo registrador o visualizador• Actualizar el rol de un usuario• Eliminar un usuario
	RR-005	Los usuarios Registrador necesitan consumir varios endpoints para: <ul style="list-style-type: none">• Agregar un delito• Eliminar un delito• Ver todos los delitos registrados
	RR-006	Los usuarios Visualizador necesitan consumir varios endpoints para: <ul style="list-style-type: none">• Buscar y visualizar una delegación por Número de Investigación o Número de Instrucción Fiscal.

Historias de Usuario

Una vez finalizado la etapa de recopilación de requerimientos, se elabora las Historias de Usuario para el componente, dando un total de 7 historias desde la **Tabla 2** hasta la **Tabla 6**.

Tabla 2 Solicitar registro para acceder al sistema- Registrador y Visualizador

HISTORIA DE USUARIO	
Identificador: HU002	Usuario: Registrador y Visualizador
Nombre de las Historia: Solicitar registro para acceder al sistema	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración: 1	
Responsable: Iveth Cocha	
Descripción: Para solicitar el registro en el sistema como Registrador y Visualizador, los usuarios deben llenar un formulario con su información personal y especificar el rol de acceso. Una vez que completen el formulario, este es remitido al administrador.	
Observación: El Administrador es quien recibe las notificaciones en el correo electrónico registrado y este se encarga de registrar o no al usuario que solicita el registro.	

Tabla 3 Gestionar Delegaciones - Administrador y Registrador

HISTORIA DE USUARIO	
Identificador: HU003	Usuario: Administrador y Registrador
Nombre de las Historia: Gestionar Delegaciones	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Iteración: 2	
Responsable: Iveth Cocha	
Descripción: Se consume los endpoints para <ul style="list-style-type: none"> ● Crear una delegación ● Ver el detalle de una delegación en específico ● Actualizar una delegación ● Ver todas las delegaciones registradas 	
Observación: En el módulo delegaciones no implica la eliminación. Para las actualizaciones no todos los campos ingresados por el Registrador o Administrador pueden ser actualizados.	

Tabla 4 Gestionar Usuarios- Administrador

HISTORIA DE USUARIO	
Identificador: HU004	Usuario: Administrador
Nombre de las Historia: Gestionar Usuarios	

Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Iteración: 1	
Responsable: Iveth Cocha	
<p>Descripción: Para que el usuario pueda cumplir con esta función debe consumir los siguientes endpoints:</p> <ul style="list-style-type: none"> ● Agregar un nuevo registrador o visualizador ● Actualizar el rol de un usuario ● Eliminar un usuario 	
<p>Observación: En este proceso, solo es necesario ingresar el número de cédula o el nombre del agente; el resto de la información se completa automáticamente. El usuario administrador tiene la capacidad de designar y seleccionar el perfil que se asigna al usuario recién agregado. Dependiendo de esta elección, el usuario agregado tiene distintos niveles de permisos. En cuanto a la actualización del registro o la visualización, solo se puede modificar el rol asignado.</p>	

Tabla 5 Gestionar Delitos - Registrador

HISTORIA DE USUARIO	
Identificador: HU005	Usuario: Registrador
Nombre de las Historia: Gestionar Delitos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración: 3	
Responsable: Iveth Cocha	
<p>Descripción: Para que el usuario pueda cumplir con esta función debe consumir los siguientes endpoints:</p> <ul style="list-style-type: none"> ● Agregar un delito ● Eliminar un delito ● Ver todos los delitos registrados 	
<p>Observación: En caso de querer saber si un delito ya se encuentra registrado o no, se incluye un filtro en la tabla para buscar el delito</p>	

Tabla 6 Buscar una delegación - Visualizador

HISTORIA DE USUARIO	
Identificador: HU006	Usuario: Visualizador
Nombre de las Historia: Buscar y visualizar una delegación	
Prioridad en Negocio:	Riesgo en Desarrollo: Medio

Iteración: 4
Responsable: Iveth Cocha
<p>Descripción: Para que el usuario pueda cumplir con esta función debe consumir los siguientes endpoints:</p> <ul style="list-style-type: none"> • Listar delegaciones • Buscar delegación • Ver el detalle de la delegación que haya buscado, mediante los parámetros de búsqueda mencionados anteriormente.
<p>Observación Para simplificar la búsqueda de una delegación, la tabla que lista las delegaciones registradas incluye un filtro en las columnas “Número de investigación Previa” y “Numero de Instrucción Fiscal”</p>

Product Backlog

En base a las Historias de Usuario y a la prioridad que tiene cada módulo para su implementación en la **Tabla 7** se indica la prioridad asignada a cada Historia de Usuario.

Tabla 7 Product Backlog

Product Backlog				
ID-HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU002	Solicitar registro para acceder al sistema - Registrador y Visualizador	1	Finalizado	Alta
HU003	Gestionar Delegaciones - Administrador y Registrador	2	Finalizado	Alta
HU004	Gestionar Usuarios - Administrador	1	Finalizado	Alta
HU005	Gestionar Delitos - Registrador	3	Finalizado	Alta
HU006	Buscar y visualizar una Delegación - Visualizador	4	Finalizado	Alta

Sprint Backlog

Se presentan un total de 5 Sprints para detallar el tiempo de culminación de cada uno. En la **Tabla 8** se muestra el Sprint Backlog.

Tabla 8 Sprint Backlog

Sprint Backlog						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB001	Diseño y desarrollo del flujo de autenticación y navegación	Inicio de Sesión	HU001	Confirmar cuenta, iniciar sesión, cerrar sesión, crear contraseña y recuperar contraseña - Administrador Registrador y Visualizador	<ul style="list-style-type: none"> Implementación de las interfaces de inicio, cierre, recuperación y creación de contraseña. Validaciones de campos y tipos de datos. Mensajes de confirmación, error o alerta. 	40H
		Usuarios	HU004	Gestionar Usuarios - Administrador	<ul style="list-style-type: none"> Implementación de las interfaces para añadir, actualizar, y eliminar un Visualizador y/o Registrador. Validaciones de campos y tipos de datos. Mensajes de confirmación, error o alerta. Protección de rutas por los Roles asignados 	
		Solicitud de Registro	HU002	Solicitar registro para acceder al sistema- Registrador y Visualizador	<ul style="list-style-type: none"> Implementación de la interfaz para solicitar el registro y acceso al Sistema de Delegaciones Validaciones de campos y tipos de datos. 	

					<ul style="list-style-type: none"> ● Mensajes de confirmación, error o alerta. 	
SB002	Diseño y desarrollo del flujo Delegaciones	Delegaciones	HU003	Gestionar Delegaciones - Administrador y Registrador	<ul style="list-style-type: none"> ● Implementación de las interfaces para mostrar las delegaciones existentes, ver el detalle de una delegación, actualizar y crear una delegación ● Validaciones de campos y tipos de datos. ● Mensajes de confirmación, error o alerta. 	60H
SB003	Diseño y desarrollo del flujo Delitos	Delitos	HU005	Gestionar Delitos - Registrador	<ul style="list-style-type: none"> ● Implementación de las interfaces para agregar, eliminar y ver los delitos. ● Validaciones de campos y tipos de datos. ● Mensajes de confirmación, error o alerta. 	30H
SB004	Diseño y desarrollo del Módulo Buscar	Visualizador de Delegaciones	HU006	Buscar y visualizar una delegación - Visualizador	<ul style="list-style-type: none"> ● Implementación en la interfaz de delegaciones para realizar la búsqueda de una delegación y ver el detalle de la delegación. ● Mensajes de confirmación, error o alerta. ● Validaciones del campo para búsqueda. ● Mostrar la delegación buscada. ● Mensajes de error o alerta. 	40H
SB005	Pruebas del frontend	<ul style="list-style-type: none"> ● Pruebas Unitarias. ● Pruebas de Rendimiento. 				30H

		<ul style="list-style-type: none"> • Pruebas de Aceptación. 	
	Documentación	Trabajo de Integración Curricular.	30H
TOTAL			240H

Diseño de Interfaces

A continuación, se presenta el Diseño de Interfaces para los roles de Administrador, Registrador y Visualizador desde la **Figura 1** hasta la **Figura 21**.



Figura 1 Página de Inicio de Sesión



Figura 2 Página de Solicitud de Registro

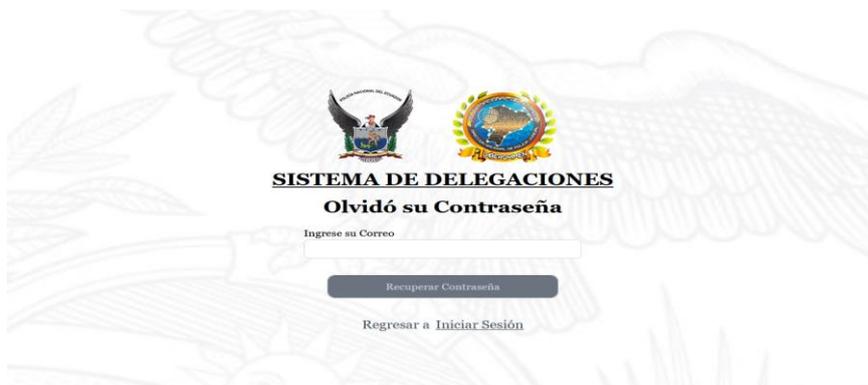


Figura 3 Página Olvido su contraseña



Figura 4 Página Reestablecer contraseña



Figura 5 Página de confirmación de cuenta



Figura 6 Página crear contraseña



Figura 7 Página Delegaciones (rol Administrador)



Figura 8 Página Agregar Delegación (rol Administrador)



Figura 9 Página Detalle Delegación (rol Administrador)

Bienvenido Salir

POLL MARIA IVETH COCHA GER - Administrador

Delegaciones

Usuarios

Agentes

ACTUALIZAR DELEGACIÓN

ASIGNACIÓN DE LA INVESTIGACION

Nº de Investigación Previa: 123456776543211

Nº de Instrucción Fiscal: [input type="text"]

Mes de ingreso de Disposiciones Fiscales: MARZO

Apellidos y Nombres del Agente: VALDIVIEZO BORJA DIEGO PATRICIO

Grado Agente: CBOS.

LOCALIZACIÓN

Cod Distrito: [input type="text"]

Dirección: [input type="text"]

Figura 10 Página Actualizar Delegación (rol Administrador)

Bienvenido Salir

POLL MARIA IVETH COCHA GER - Administrador

Delegaciones

Usuarios

Agentes

Usuarios

Nuevo Usuario

Nº	Grado	Nombre y Apellido	Cédula	Rol	Acciones
1	POLL.	MARTHA LORENA	050111769	Registrador	 
2	POLL.	MARIA IVETH COCHA GER	1728141571	Administrador	 

Figura 11 Página Usuarios (rol Administrador)

Bienvenido Salir

POLL MARIA IVETH COCHA GER - Administrador

Delegaciones

Usuarios

Agentes

REGISTRAR NUEVO USUARIO

Cédula - Nombre: [input type="text"]

Cédula: [input type="text"]

Nombre: [input type="text"]

Correo: [input type="text"]

Rol: Seleccione el rol ▼

Registrar Usuario

Figura 12 Página Registrar Nuevo Usuario (rol Administrador)

Bienvenido Salir

 POLL. MARIA IVETH COCHA GER - Administrador

Delegaciones

Usuarios

Agentes

ACTUALIZAR USUARIO

Cédula:

Nombre:

Correo:

Rol:

Figura 13 Página Actualizar Usuario (rol Administrador)

Bienvenido Salir

 POLL. MARTHA LORENA - Registrador

Delegaciones

Delitos

Agentes

DELEGACIONES

N° Orden	Año Ingreso	N° Investigación Previa	N° Instrucción Fiscal	Grado	Agente	Acciones
1	2024	123456776543211	<input type="text" value="Buscar"/>	CBOS.	VALDIVIEZO BORJA DIEGO PATRICIO	 
2	2024	123443211234567	<input type="text" value="Buscar"/>	SGOP.	ARELLANO CATAGNA NELSON ORLANDO	 

« ‹ › »

Figura 14 Página Delegaciones (rol Registrador)

Bienvenido Salir

 POLL. MARTHA LORENA - Registrador

Delegaciones

Delitos

Agentes

AGREGAR DELEGACIÓN

ASIGNACIÓN DE LA INVESTIGACIÓN

N° de Investigación Previa:

N° de Instrucción Fiscal:

Mes de ingreso de Disposiciones Fiscales: -- Seleccione un mes --

Apellidos y Nombres del Agente:

Grado Agente:

LOCALIZACIÓN

Realice la Búsqueda para la Localización

Figura 15 Página Agregar Delegación (rol Registrador)



Figura 16 Página Actualizar Delegación (rol Registrador)



Figura 17 Página Detalle Delegación (rol Registrador)



Figura 18 Pagina delitos (rol Registrador)



Figura 19 Página Agregar delito (rol Registrador)



Figura 20 Página Delegaciones (rol Visualizador)



Figura 21 Página Detalle Delegación (rol Visualizador)

Pruebas

Pruebas Unitarias

Desde la **Figura 22** hasta la **Figura 25** se presentan los resultados de las pruebas unitarias realizadas.

```
RERUN src/test/TablaDelegaciones.test.tsx x26

✓ src/test/TablaDelegaciones.test.tsx (4)
✓ Listar Delelegaciones (4)
  ✓ Se muestra la tabla con el contenido de las delegaciones
  ✓ El icono actualizar lleva a Actualizar
  ✓ El icono detalle lleva a Visualizar
  ✓ Funciona el filtro de busqueda

Test Files 1 passed (1)
Tests      4 passed (4)
Start at   16:21:41
Duration   570ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 22 Resultados componente listar Delegaciones

```
RERUN src/test/NuevaDelegacion.test.tsx x13

✓ src/test/NuevaDelegacion.test.tsx (4)
✓ Agregar Delelegacion (4)
  ✓ Se muestra el formulario de agregar
  ✓ Funcionalidad de campos de autocompletado
  ✓ El botton realiza su accion
  ✓ Dialogo con el mensaje de confirmacion/error

Test Files 1 passed (1)
Tests      4 passed (4)
Start at   21:30:40
Duration   630ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 23 Prueba Unitaria formulario Añadir Delegación

```
RERUN src/test/DetalleDelegacion.test.tsx x8

✓ src/test/DetalleDelegacion.test.tsx (1)
✓ Visualizar Delelegacion (1)
  ✓ Se renderiza a Visualizar Delegación

Test Files 1 passed (1)
Tests      1 passed (1)
Start at   23:50:25
Duration   626ms

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 24 Prueba Unitaria Visualizar Delegación

```

RERUN src/test/ActualizarDelegacion.test.tsx x16
✓ src/test/ActualizarDelegacion.test.tsx (3)
✓ Visualizar Delelegacion (3)
  ✓ Se renderiza al formulario Actualizar Delegación
  ✓ El boton realiza su función
  ✓ Apertura del cuadro del dialogo con el mensaje esperado

Test Files 1 passed (1)
Tests 3 passed (3)
Start at 11:43:14
Duration 622ms

PASS Waiting for file changes...
press h to show help, press q to quit

```

Figura 25 Prueba Unitaria Actualizar Delegación

Pruebas de Rendimiento

En la **Figura 26** se muestra cómo se usa la herramienta de Lighthouse para realizar la prueba de rendimiento. Primeramente, se debe inspeccionar la página deseada y seleccionar la opción de Lighthouse



Figura 26 Ventana de Lighthouse

Desde la **Figura 27** hasta la **Figura 31** se presentan los resultados de las pruebas de rendimiento realizadas.

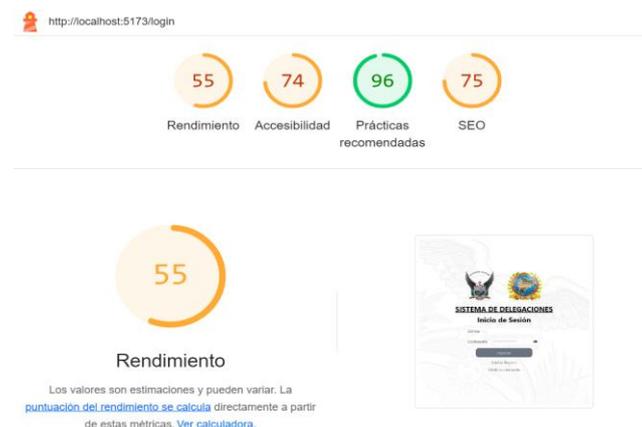


Figura 27 Prueba Rendimiento de Inicio de Sesión



Rendimiento

Los valores son estimaciones y pueden variar. La [puntuación del rendimiento se calcula](#) directamente a partir de estas métricas. [Ver calculadora.](#)



Figura 28 Prueba Rendimiento Listar Delegaciones

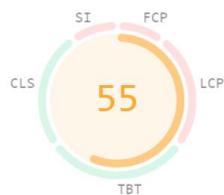


Rendimiento

Los valores son estimaciones y pueden variar. La [puntuación del rendimiento se calcula](#) directamente a partir de estas métricas. [Ver calculadora.](#)



Figura 29 Prueba Rendimiento Agregar Delegación



Rendimiento

Los valores son estimaciones y pueden variar. La [puntuación del rendimiento se calcula](#) directamente a



Figura 30 Prueba Rendimiento Detalle Delegación

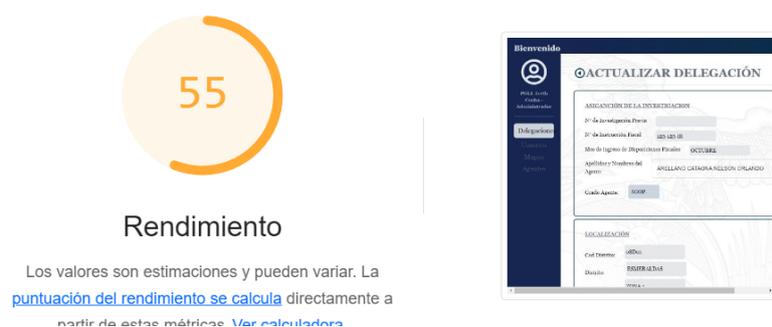


Figura 31 Prueba Rendimiento Actualizar Delegación

Pruebas de Aceptación

Las pruebas de aceptación se realizan en función de los requerimientos y las historias de usuario plantadas al comienzo del desarrollo del sistema. Las se encuentran detalladas desde la **Tabla 9** hasta la **Tabla 13**.

Tabla 9 Prueba de aceptación 1 Confirmar su cuenta, iniciar sesión, cerrar sesión, modificar contraseña y recuperar contraseña

PRUEBA DE ACEPTACION	
Identificador: PA001	Identificador de HU: HU001
Nombre: Confirmar su cuenta, iniciar sesión, cerrar sesión, modificar contraseña y recuperar contraseña.	
Descripción: Al correo del usuario le llega un mensaje con un enlace para confirmar la cuenta, esto lo lleva a una nueva ventana donde debe crear una contraseña para acceder al sistema. Para el inicio de sesión los usuarios deben ingresar sus credenciales, en caso de haber olvidado la contraseña puede generar una nueva.	
Pasos de Ejecución: Confirmación: <ul style="list-style-type: none"> ● Abrir el correo electrónico ● Hacer clic en el enlace del mensaje ● Crear una contraseña Iniciar Sesión: <ul style="list-style-type: none"> ● Ingresar sus credenciales correctamente ● Presionar el botón Ingresar Cerrar Sesión: <ul style="list-style-type: none"> ● Asegurarse de haber iniciado sesión 	

<ul style="list-style-type: none"> ● Hacer clic en el botón cerrar sesión <p>Recuperar contraseña:</p> <ul style="list-style-type: none"> ● Presionar sobre Olvidó su contraseña ● Ingresar el correo electrónico ● Hacer clic en el botón Recuperar Contraseña ● Revisar el correo y dar clic en el enlace para generar una nueva contraseña ● Crear la nueva contraseña
<p>Resultado Deseado: El componente frontend permite confirmar la cuenta, iniciar sesión, cerrar sesión, modificar contraseña y recuperar contraseña</p>
<p>Evaluación de la Prueba: El resultado esperado ha sido confirmado y ha obtenido una aprobación del 100%.</p>

Tabla 10 Prueba de aceptación 3 Módulo Delegaciones

PRUEBA DE ACEPTACION	
Identificador: PA003	Identificador de HU: HU003
Nombre: Módulo Delegaciones	
Descripción: Este módulo permite la gestión de delegaciones siempre y cuando el usuario haya accedido al sistema.	
Pasos de Ejecución: Para acceder a los pasos el usuario debe asegurarse de haber iniciado sesión	
Agregar delegación <ul style="list-style-type: none"> ● Dar clic en botón de Agregar delegación ● Llenar el formulario con los datos solicitados ● Guardar la información con el botón Añadir delegación 	
Listar delegaciones <ul style="list-style-type: none"> ● Se le mostraran en una tabla todas las delegaciones registradas 	
Actualizar delegación <ul style="list-style-type: none"> ● En el listado de delegaciones tiene el icono de actualizar (color verde) ● Hacer clic en el icono ● Actualizar los campos que sean necesarios ● Dar clic en botón Actualizar delegación 	
Ver el detalle de la delegación <ul style="list-style-type: none"> ● En el listado de delegaciones tiene el icono de visualizar (color azul) ● Hacer clic en el icono ● Visualizar la información de la delegación 	
Resultado Deseado: El componente frontend permite gestionar delegaciones	
Evaluación de la Prueba: El resultado esperado ha sido confirmado y ha obtenido una aprobación del 100%.	

Tabla 11 Prueba de aceptación 4 Módulo Usuarios

PRUEBA DE ACEPTACION	
Identificador: PA004	Identificador de HU: HU004
Nombre: Módulo Usuarios	
Descripción: El usuario Administrador puede agregar un usuario, actualizar el rol y eliminar al usuario	
Pasos de Ejecución: Para acceder a los pasos el usuario debe asegurarse de haber iniciado sesión. Debe seleccionar la opción de Usuarios en el panel Agregar usuario <ul style="list-style-type: none"> ● Hacer clic en el botón de Añadir usuario ● Llenar el cuadro de búsqueda con el número de cédula o el nombre del usuario a registrar. ● Seleccionar el rol para darle los permisos necesarios ● Dar clic en el botón Añadir Usuario para guardar la información Listar usuarios <ul style="list-style-type: none"> ● En la opción de Usuarios se le muestra una tabla con todos los usuarios registrados Actualizar Rol <ul style="list-style-type: none"> ● En el listado de Usuarios tiene el icono de actualizar (color verde) ● Hacer clic en el icono ● Actualizar el rol del usuario ● Guardar los cambios con el botón actualizar Usuario Eliminar <ul style="list-style-type: none"> ● En el listado de delegaciones tiene el icono de eliminar (color Rojo) ● Hacer clic en el icono ● Confirmar la acción de eliminar 	
Resultado Deseado: El componente frontend permite gestionar Usuarios	
Evaluación de la Prueba: El resultado esperado ha sido confirmado y ha obtenido una aprobación del 100%.	

Tabla 12 Prueba de aceptación 5 Módulo Delitos

PRUEBA DE ACEPTACION	
Identificador: PA005	Identificador de HU: HU005
Nombre: Módulo Delitos	
Descripción: El usuario Registrador puede agregar, visualizar y eliminar un delito.	
Pasos de Ejecución: Para acceder a los pasos el usuario debe asegurarse de haber iniciado sesión Debe seleccionar la opción de Delitos en el panel Agregar <ul style="list-style-type: none"> ● Hacer clic en el botón de Añadir delito ● Llenar los campos solicitados. 	

<ul style="list-style-type: none"> ● Dar clic en el botón Añadir delito <p>Listar</p> <ul style="list-style-type: none"> ● En la opción de delitos se le muestra una tabla con todos los delitos registrados <p>Visualizar</p> <ul style="list-style-type: none"> ● En el listado se muestra un filtro de búsqueda, llenar el cuadro de búsqueda <p>Eliminar</p> <ul style="list-style-type: none"> ● En el listado se muestra el icono de eliminar (color Rojo) ● Hacer clic en el icono ● Confirmar la acción de eliminar
<p>Resultado Deseado: El componente frontend permite gestionar delitos</p>
<p>Evaluación de la Prueba: El resultado esperado ha sido confirmado y ha obtenido una aprobación del 100%.</p>

Tabla 13 Prueba de aceptación 6 Buscar Delegación - Ver el Detalle de una Delegación - Visualizador

PRUEBA DE ACEPTACION	
Identificador: PA006	Identificador de HU: HU006
Nombre: Buscar una delegación	
Descripción: El usuario Visualizador puede buscar delegaciones por el número de investigación previa o número de instrucción fiscal.	
Pasos de Ejecución: Para acceder a los pasos el usuario debe asegurarse de haber iniciado sesión <ul style="list-style-type: none"> ● Se le muestra una tabla con todas las delegaciones registradas ● Debajo del título de las columnas N° de Investigación Previa y N° de Instrucción Fiscal tiene filtros de búsqueda, debe llenar cualquiera de los dos filtros ● Aparece la delegación buscada 	
Resultado Deseado: E El componente frontend permite buscar delegaciones	
Evaluación de la Prueba: El resultado esperado ha sido confirmado y ha obtenido una aprobación del 100%.	

ANEXO III

Para visualizar el Manual de Usuario se debe ingresar al siguiente enlace:

<https://www.youtube.com/watch?v=4j6foOG5MQw>

ANEXO IV

En esta sección se proporciona el enlace del repositorio del frontend.

Repositorio del frontend

El código del proyecto se aloja en GitHub, el cual puede ser accedido a través del siguiente enlace:

<https://github.com/iveth-cocha/Front>