

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

SISTEMAS DE INFORMACIÓN



TRABAJO DE TITULACIÓN

DESARROLLO DE LOS MÓDULOS DE GESTIÓN DE CASOS, ACTIVIDADES Y
ASIGNACIONES PARA LOS CGJ DE LA PUCE

AUTOR

SERGIO ALEXANDER DÁVILA AGUILAR

DIRECTOR:

GUIDO EDUARDO OCHOA MORENO

QUITO DM, 2025

DEDICATORIA

Dedico el presente trabajo a mis padres y abuelitos, personas que desde el inicio me dieron la fuerza necesaria y me brindaron de sus ánimos durante toda mi formación académica. También dedico a una buena amiga Karen Tituaña, quien ha sido mi gran soporte y me ha brindado su apoyo incondicional desde que inicié la carrera, sin importar la situación, sin ella, nada de esto hubiera sido posible.

AGRADECIMIENTOS

Agradezco profundamente a mis padres, quienes me dieron la oportunidad de seguir mi carrera universitaria en la PUCE, por todo lo que han sacrificado por mi y por siempre creer en que voy a lograr cosas grandes. A mis amigos más cercanos por siempre apoyar mis decisiones y alegrarse por mis logros, además de darme su amistad sincera. A todas las personas que conocí a lo largo de mi formación universitaria, cada uno de ellos aportó en pequeña o gran medida a la persona que ahora soy, espero que logren cumplir sus metas y sueños al igual que yo.

RESUMEN

El presente trabajo de titulación detalla el proceso de análisis, diseño y desarrollo del Sistema “Balanza Web” para los Consultorios Jurídicos de la PUCE, el cual tiene como propósito automatizar procesos de la Gestión de Casos. En este desarrollo, se aplicó la metodología ágil de desarrollo Prototipado. El patrón de arquitectura utilizado para las interfaces es MVC (Modelo Vista-Controlador), haciendo uso del framework Vue.js, además de los lenguajes de programación Typescript, TailwindCSS, Javascript, HTML.

Palabras clave: Modelo, Vista, Controlador, Vue.js, Prototipado, TailwindCSS, Typescript, Javascript, Sistema.

ABSTRACT

This degree project presents the analysis, design, and development of the “Balanza Web” system for the Legal Clinics of PUCE, aimed at automating case management processes. The agile Prototyping methodology was applied throughout the development process. The architectural pattern used for the interfaces is MVC (Model-View-Controller), with the implementation of the Vue.js framework, along with programming languages such as Typescript, TailwindCSS, Javascript, and HTML.

Keywords: Model, View, Controller, Vue.js, Prototyping, TailwindCSS, Typescript, Javascript, System.

ÍNDICE

DEDICATORIA	I
RESUMEN	III
ABSTRACT	IV
CAPÍTULO I: INTRODUCCIÓN.....	1
1.1 Planteamiento del problema	1
1.2 Justificación	1
1.3 Objetivos	2
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos	2
1.4 Alcance	2
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA	3
2.1 Aplicación Web	3
2.1.1 Aplicaciones Web basadas en servicios	3
2.2 Metodología de desarrollo de software	3
2.2.1 Prototipado Evolutivo.....	4
2.3 Base de Datos	5
2.3.1 MySQL	5
2.4 Lenguaje de Programación	5
2.4.1 JavaScript	5
2.4.2 TypeScript.....	5
2.5 Entorno de Desarrollo Integrado (IDE)	6
2.5.1 Visual Studio Code	6
2.6 Framework de Desarrollo Web	6
2.6.1 Node.js.....	6
2.6.2 Vue.js	6
2.6.3 Middleware y Backend.....	6
2.7 Control de Versiones	7
2.7.1 Importancia del control de versiones.....	7
2.7.2 Git y GitHub	7
2.8 Buenas Prácticas de Desarrollo Web	7
2.8.1 Principios de diseño y usabilidad	7

CAPÍTULO III: ANÁLISIS Y DISEÑO DE LA APLICACIÓN	8
3.1 Análisis de Requerimientos	8
3.1.1 Requisitos Funcionales	8
3.1.2 Requisitos No Funcionales	8
3.2 Diagramas Generales de Casos de Uso	9
3.2.1 Módulo de Casos	9
3.2.2 Módulo de Actividades	11
3.2.3 Módulo de Asignaciones	13
3.3 Diagramas de Casos de Uso Específicos	15
3.3.1 Módulo de Casos	15
3.3.1.1 Caso de uso Crear Caso	15
3.3.1.2 Caso de Uso Consultar Caso	17
3.3.1.3 Caso de uso Actualizar Caso	19
3.3.2 Módulo de Actividades	21
3.3.2.1 Caso de Uso Crear Actividad.....	21
3.3.2.2 Caso de Uso Consultar Actividad.....	23
3.3.2.3 Caso de uso Actualizar Actividad.....	25
3.3.2.4 Caso de Uso Generar Reporte de Actividades.....	27
3.3.3 Módulo de Asignaciones	29
3.3.3.1 Caso de Uso Crear Asignación	29
3.3.3.2 Caso de Uso Consultar Asignación	31
3.3.3.3 Caso de uso Actualizar Asignación	32
3.4 Modelado del Sistema	34
3.4.1 Diagrama Entidad-Relación	34
3.5 Diseño de la Arquitectura del Sistema	35
3.5.1 Diseño de la Base de Datos	35
3.5.2 Diseño de la Interfaz de Usuario	38
CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN	42
4.1 Preparación del Entorno de Desarrollo	42
4.2 Estándares de Codificación	42
4.3 Implementación de la Funcionalidad	43
4.3.1 Backend	43
4.3.2 Frontend.....	44

4.4 Plan de Pruebas	44
4.4.1 Tipos de Pruebas	44
4.4.2 Criterios de Aceptación	45
4.4.3 Casos de Prueba	46
4.4.3.1 Casos de Prueba Módulo de Casos.....	46
4.4.3.2 Casos de Prueba Módulo de Casos.....	47
4.4.3.3 Casos de Prueba Módulo de Reportes enfocado a Actividades.....	48
4.4.3.4 Casos de Prueba Módulo de Asignaciones	48
4.5 Post Mortem	49
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	52
5.1 Conclusiones	52
5.2 Recomendaciones	53
Bibliografía:.....	54

ÍNDICE DE ILUSTRACIONES

Ilustración 1 – Diagrama Prototipado Evolutivo.....	4
Ilustración 2 - Diagrama Caso de Uso Módulo Casos.....	9
Ilustración 3 Diagrama Caso de Uso Módulo Actividades.....	11
Ilustración 4 - Diagrama Caso de Uso Módulo Asignaciones.....	13
Ilustración 5 - Diagrama Caso de Uso Crear Caso.....	15
Ilustración 6 - Diagrama Caso de Uso Consultar Caso.....	17
Ilustración 7 - Diagrama Caso de Uso Actualizar Caso.....	19
Ilustración 8 - Diagrama Caso de Uso Crear Actividad.....	21
Ilustración 9 – Diagrama Caso de Uso Consultar Actividad.....	23
Ilustración 10 - Diagrama Caso de Uso Actualizar Actividad.....	25
Ilustración 11 – Diagrama Caso de Uso Generar reporte de Actividad.....	27
Ilustración 12 – Diagrama Caso de uso Crear asignación.....	29
Ilustración 13 - Diagrama Caso de uso Consultar asignación.....	31
Ilustración 14 – Diagrama Caso de uso Actualizar Asignación.....	32
Ilustración 15 - Diagrama Entidad-Relación.....	34
Ilustración 16- Entidad Users en la base de datos.....	36
Ilustración 17- Entidad Assignments en la base de datos.....	36
Ilustración 18- Entidad Activities en la base de datos.....	37
Ilustración 19- Entidad Initial_Consultations en la base de datos.....	37
Ilustración 20 - Pantalla de casos.....	38
Ilustración 21 - Vista Actividades impuestas de un caso.....	38
Ilustración 22 - Formulario para completar actividad.....	39
Ilustración 23 - Detalle de la actividad.....	39
Ilustración 24- Panel de asignaciones pendientes.....	40
Ilustración 25 - Panel casos ya asignados.....	40
Ilustración 26 - Pantalla de generación de reportes en Excel.....	41

INDICE DE TABLAS

Tabla 1 - Descripción de Caso de Uso general de Módulo de Casos	10
Tabla 2- Descripción Caso de Uso General de Módulo de Actividades.....	12
Tabla 3 - Descripción Caso de Uso General de Módulo de Asignaciones	14
Tabla 4 - Caso de Uso Crear Caso	15
Tabla 5 - Caso de Uso Consultar Caso	17
Tabla 6 - Caso de Uso Actualizar Caso	19
Tabla 7 - Caso de Uso Crear Actividad	21
Tabla 8 - Caso de Uso Consultar Actividad.....	23
Tabla 9 - Caso de Uso Actualizar Actividad.....	25
Tabla 10 - Caso de Uso Generar reporte de Actividad	27
Tabla 11 - Caso de uso Crear asignación.....	29
Tabla 12 - Caso de uso Consultar asignación	31
Tabla 13 - Caso de uso Actualizar Asignación	32
Tabla 14- Casos de Prueba Módulo de Casos.....	46
Tabla 15 - Casos de Prueba Módulo de Casos.....	47
Tabla 16 - Casos de Prueba Módulo de Reportes enfocado a Actividades.....	48
Tabla 17 - Casos de prueba Módulo de Asignaciones	48

CAPÍTULO I: INTRODUCCIÓN

1.1 Planteamiento del problema

Los Consultorios Jurídicos Gratuitos de la Pontificia Universidad Católica del Ecuador cumplen una función importante en cuanto a la atención legal a personas. Sin embargo, la gestión semiautomatizada de casos, actividades y asignaciones ha generado ineficiencias y errores operativos en sus procesos internos. La falta de una plataforma tecnológica centralizada dificulta el seguimiento de los casos, además de la correcta distribución de tareas entre estudiantes y tutores, y la trazabilidad de los casos y las actividades realizadas. Esta situación ha comprometido tanto la calidad del servicio brindado como la experiencia de aprendizaje de los estudiantes practicantes de derecho. Por tanto, se requiere el desarrollo de una aplicación web especializada que permita gestionar eficientemente los casos jurídicos, asignar actividades de forma ordenada y mantener un control transparente de las asignaciones.

1.2 Justificación

El desarrollo del módulo de gestión de casos, actividades y asignaciones representa una mejora significativa en la operatividad de los CJG (Consultorios Jurídicos Gratuitos) de la PUCE. Este sistema ayudará y permitirá automatizar procesos clave, optimizando tiempos de respuesta y mejorando la organización institucional. Por otro lado, favorecerá la supervisión académica, el aprendizaje práctico de los estudiantes y la calidad del servicio legal brindado. La implementación de herramientas tecnológicas modernas, la cual se encuentra alineada a metodologías ágiles y buenas prácticas de desarrollo, garantiza que la solución propuesta sea escalable, fácil de mantener y robusta, además de que también se alinea con los objetivos institucionales de innovación y compromiso social establecidos por la universidad.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un módulo web para la gestión de casos, actividades y asignaciones de los consultorios jurídicos gratuitos de la PUCE, que permita mejorar la organización, trazabilidad y eficiencia en los procesos de atención legal.

1.3.2 Objetivos Específicos

- Analizar los procesos actuales de gestión en los CJG.
- Implementar una arquitectura de software adecuada para el desarrollo del módulo web.
- Implementar las funcionalidades necesarias para el registro, seguimiento y asignación de casos y actividades.
- Validar el funcionamiento del sistema a través de pruebas funcionales con los usuarios clave de los CJG.

1.4 Alcance

Esta parte del proyecto se enfocará en el desarrollo del módulo de gestión de casos, actividades y asignaciones, que será parte de un sistema web completo y destinado al uso interno de CJG (Consultorios Jurídicos Gratuitos) de la PUCE. Este sistema permitirá registrar nuevos casos, asignar actividades a estudiantes o tutores, y mantener un historial de los procedimientos realizados. La solución será desarrollada utilizando tecnologías web modernas (Vue.js, Typescript, Javascript MySQL), y se diseñará para su fácil implementación y escalabilidad futura.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2.1 Aplicación Web

Una aplicación web es un sistema interactivo que puede ser accesible desde un navegador, ya sea móvil o de computador, que permite a los usuarios realizar tareas específicas mediante una interfaz gráfica (GUI). Estas aplicaciones actualmente son muy utilizadas por su portabilidad y facilidad de acceso. En el contexto del presente proyecto, una aplicación web facilita el acceso multiusuario mediante roles a los datos de casos jurídicos desde cualquier lugar con conexión a internet, lo cual es esencial para estudiantes, abogados y administradores de los consultorios.

2.1.1 Aplicaciones Web basadas en servicios

Las aplicaciones web basadas en servicios dividen la lógica del sistema en distintos componentes, los cuales se encuentran comunicados entre sí a través de interfaces de APIs. Este modelo permite separar el frontend del servidor backend, lo que ayuda a tener una arquitectura desacoplada y reutilizable. Los servicios suelen implementarse como microservicios o APIs RESTful, lo que facilita la escalabilidad, mantenimiento y despliegue de funcionalidades específicas sin afectar a otras.

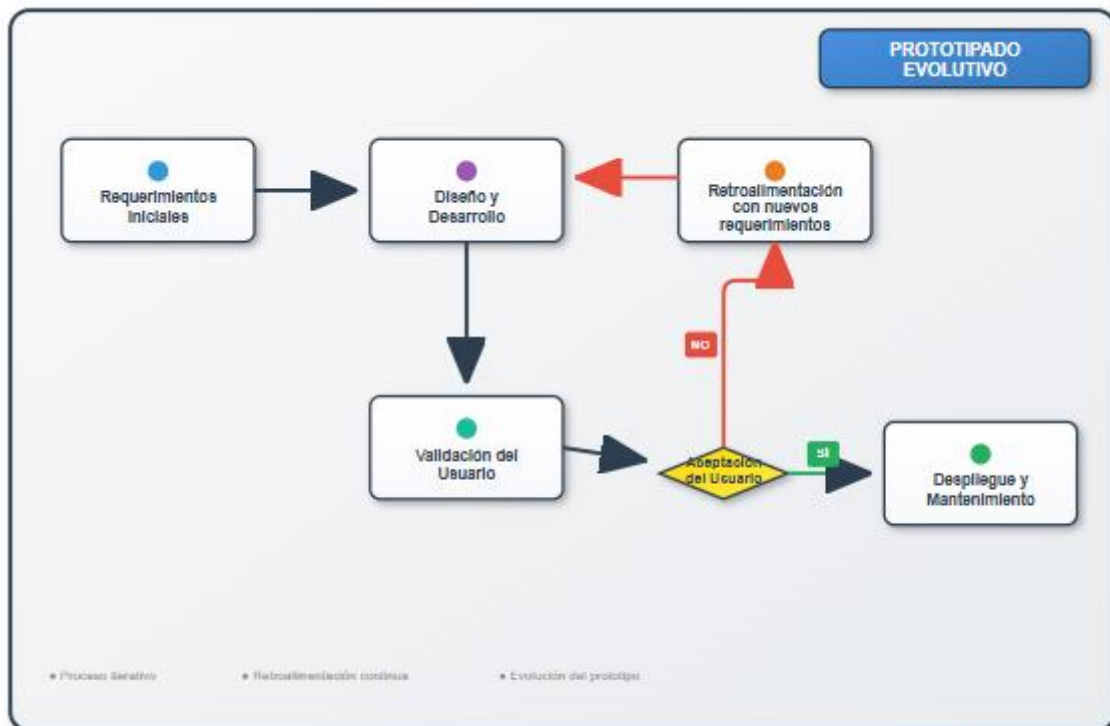
2.2 Metodología de desarrollo de software

Para el desarrollo de este módulo, se empleó la metodología de prototipado, la cual está enfocada en la creación rápida de versiones preliminares del sistema, lo cual nos permite obtener retroalimentación temprana por parte de los usuarios finales. Esta metodología es especialmente útil cuando los requerimientos del sistema no están completamente definidos desde el inicio, esto debido a que permite al equipo de desarrollo analizar de mejor manera las necesidades del usuario a través de la interacción con prototipos funcionales.

2.2.1 Prototipado Evolutivo

El prototipado evolutivo es una metodología ágil de desarrollo de software que se basa en la construcción inicial de un prototipo funcional que evoluciona progresivamente a través de iteraciones sucesivas y revisiones con los usuarios finales. En cada iteración, los usuarios o stakeholders prueban una versión del sistema, brindan cierta retroalimentación detallada sobre funcionalidades o diseño y, a partir de esta, el prototipo se modifica hasta convertirse en el producto final. Esta estrategia resulta particularmente útil en proyectos donde los requisitos no están completamente definidos desde el inicio o tienden a cambiar con el tiempo. En lugar de desarrollar el sistema completo en una sola fase, se construyen versiones incrementales con funcionalidades parciales pero operativas, lo que permite validar de forma continua tanto la usabilidad como los requerimientos funcionales. Durante este proceso, se promueve una participación activa del usuario, lo que permite detectar errores o mejoras desde etapas tempranas del desarrollo. Esto reduce el riesgo de desarrollar funcionalidades innecesarias o mal planteadas, y asegura que el sistema evolucione de acuerdo con las expectativas reales del usuario final.

Ilustración 1 – Diagrama Prototipado Evolutivo



Nota: Diagrama de Ciclo del Prototipado Evolutivo. Elaboración Propia

2.3 Base de Datos

La base de datos es uno de los principales componentes para el funcionamiento del sistema, ya que almacena y organiza la información que utiliza la aplicación. En este proyecto, la base de datos almacenará la información relacionada con casos jurídicos, actividades asignadas, usuarios del sistema y su historial de acciones.

2.3.1 MySQL

MySQL es un sistema gestor de bases de datos relacionales de código abierto, el cual es ampliamente utilizado por su fiabilidad y rendimiento. Es ideal para aplicaciones web por su compatibilidad con múltiples lenguajes de programación, además de que permite consultas eficientes a través del lenguaje SQL. Se utilizará MySQL para garantizar una adecuada estructuración y acceso a los datos de todos los módulos del sistema.

2.4 Lenguaje de Programación

Un lenguaje de programación es un conjunto de reglas y sintaxis que permite a los desarrolladores dar instrucciones a una computadora de una forma para realizar tareas específicas. El desarrollo de un sistema web requiere el uso de lenguajes que permitan implementar una interacción fluida entre el usuario y la aplicación, además de una experiencia de usuario óptima.

2.4.1 JavaScript

JavaScript es un lenguaje de programación interpretado, utilizado principalmente en el desarrollo frontend de aplicaciones web, que nos permite crear interfaces dinámicas, validar formularios en el cliente y manipular el DOM, lo cual mejora la interactividad de la aplicación. JavaScript se ha ido expandiendo al entorno del servidor con tecnologías como Node.js, y su ecosistema incluye bibliotecas como Axios para consumo de APIs, Express para construcción de servidores web.

2.4.2 TypeScript

TypeScript es un superset de JavaScript que añade tipado estático y características de programación orientada a objetos. Su uso permite detectar errores en tiempo de compilación y mantener un código más organizado y escalable, lo cual es especialmente útil en proyectos de mediana a gran escala como el presente módulo. En proyectos con Vue y Node.js como el actual, TypeScript nos permite integrar tipado estático en todo el flujo

de desarrollo. Hay algunas bibliotecas como Vue Router, Pinia o VeeValidate que cuentan con soporte completo para TypeScript.

2.5 Entorno de Desarrollo Integrado (IDE)

El IDE es una herramienta que facilita el desarrollo de software al integrar funciones como edición de código, depuración y control de versiones.

2.5.1 Visual Studio Code

Visual Studio Code es un editor de código fuente potente y de código abierto, que incluye herramientas de depuración, con soporte para múltiples lenguajes de programación así como una gran variedad de extensiones útiles para la detección y corrección de errores, formato de los archivos, visualización, etc. Se utilizará este entorno por lo ideal de su integración con tecnologías modernas como JavaScript, TypeScript, Node.js y Git.

2.6 Framework de Desarrollo Web

Un framework web es una “plantilla” que nos proporciona una estructura base para el desarrollo de aplicaciones, lo que facilita la implementación de funcionalidades comunes y promueve buenas prácticas de programación.

2.6.1 Node.js

Node.js es un entorno de ejecución para JavaScript en el servidor. Permite a los desarrolladores crear aplicaciones escalables y de alto rendimiento mediante un modelo de I/O no bloqueante. Se utilizará en el backend del sistema para gestionar la lógica de negocio, los modelos, la capa de datos y las conexiones con la base de datos.

2.6.2 Vue.js

Vue.js es un framework progresivo para construir interfaces de usuario. Su enfoque en la reactividad y especialmente sus componentes reutilizables lo convierten en una excelente opción para el desarrollo de un frontend ligero y eficiente. Vue.js nos permitirá construir una interfaz clara y rápida para los usuarios.

2.6.3 Middleware y Backend

El backend representa la lógica de negocio del sistema, la cual está encargada de procesar los datos, gestionar las reglas del negocio y conectar con la base de datos. En Node.js, se emplea comúnmente Express o Sequelize como middleware para definir rutas, autenticar usuarios y manejar peticiones. El middleware actúa como un intermediario entre la petición del cliente y la respuesta del servidor, lo que permite añadir capas de seguridad,

alguna validación y la transformación de datos. En este proyecto, Express y Sequelize permitieron construir un backend robusto y escalable.

2.7 Control de Versiones

El control de versiones es fundamental al momento de realizar el desarrollo colaborativo de un sistema y el seguimiento de cambios en el código fuente.

2.7.1 Importancia del control de versiones

La importancia del control de versiones en el desarrollo de cualquier aplicación radica en mantener un historial de modificaciones y colaborar en equipo sin sobrescribir código, además de que nos permite revertir fácilmente cambios en caso de errores. Esta es una práctica esencial en cualquier desarrollo profesional.

2.7.2 Git y GitHub

Git es el sistema de control de versiones más utilizado actualmente en todo el mundo. Por otro lado, GitHub es una plataforma web que permite alojar repositorios y gestionar colaboraciones para el desarrollo de aplicaciones. En este proyecto, es esencial emplear ambos para asegurar la trazabilidad del código y facilitar el trabajo en equipo.

2.8 Buenas Prácticas de Desarrollo Web

Aplicar buenas prácticas durante el desarrollo de un sistema asegurará que el sistema sea funcional, mantenible y accesible para cualquier desarrollador, lo que además lo hace escalable.

2.8.1 Principios de diseño y usabilidad

La aplicación debe ser intuitiva, accesible y visualmente clara. Se aplicarán principios como consistencia, retroalimentación al usuario, jerarquía visual y adaptación a diferentes dispositivos, para garantizar una experiencia de usuario satisfactoria. Además, se aplicaron otros principios como el diseño centrado en el usuario, jerarquía visual para destacar elementos clave, y el uso de colores neutros para no saturar la vista. También se priorizó la accesibilidad mediante etiquetas semánticas, contraste adecuado y tamaños de fuente legibles. Se utilizó TailwindCSS para mantener consistencia visual, con un diseño responsive que adapta las interfaces a distintos tamaños de pantalla, mejorando la experiencia de uso.

CAPÍTULO III: ANÁLISIS Y DISEÑO DE LA APLICACIÓN

3.1 Análisis de Requerimientos

La presente sección detalla los requerimientos necesarios para el desarrollo e implementación del módulo de Gestión de Casos, Actividades y Asignaciones dentro del sistema de “Balanza Web”, cuyo propósito es optimizar y digitalizar la gestión de los Consultorios Jurídicos Gratuitos de la PUCE.

3.1.1 Requisitos Funcionales

Los requisitos funcionales extraídos mediante entrevistas y observaciones con usuarios de los CJG de la PUCE fueron:

- Permitir a los usuarios visualizar los casos y su información relacionada según su rol y área.
- Permitir a los usuarios crear y completar las actividades asociadas a cada caso.
- Permitir a los usuarios asignar y reasignar casos y actividades a los estudiantes o abogados de los casos correspondientes.
- Visualizar el detalle de los casos, evidencias, información de los usuarios. etc.
- Generar reportes de actividades según fechas establecidas.

3.1.2 Requisitos No Funcionales

Los requisitos no funcionales definen las características de calidad y restricciones del sistema:

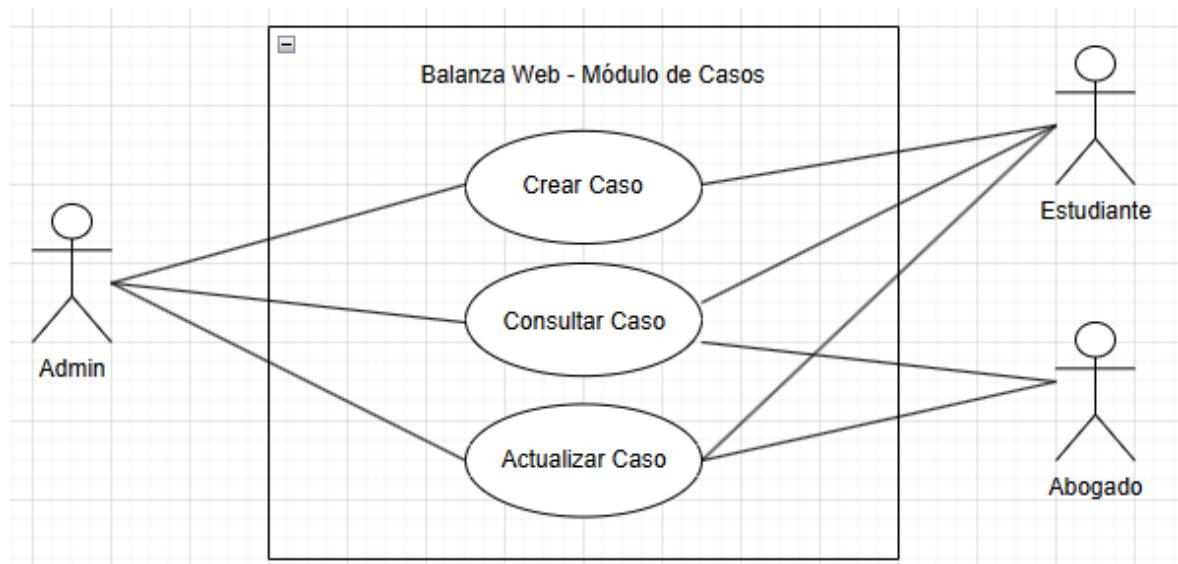
- La aplicación debe ser accesible a través de navegadores web actuales.
- La interfaz debe ser intuitiva y amigable para el usuario.
- |

3.2 Diagramas Generales de Casos de Uso

A continuación, se presentan los diagramas generales de casos de uso para los módulos:

3.2.1 Módulo de Casos

Ilustración 2 - Diagrama Caso de Uso Módulo Casos



Nota: Elaboración propia. Diagrama de Casos de Uso del Módulo de Gestión de Casos con sus Actores.

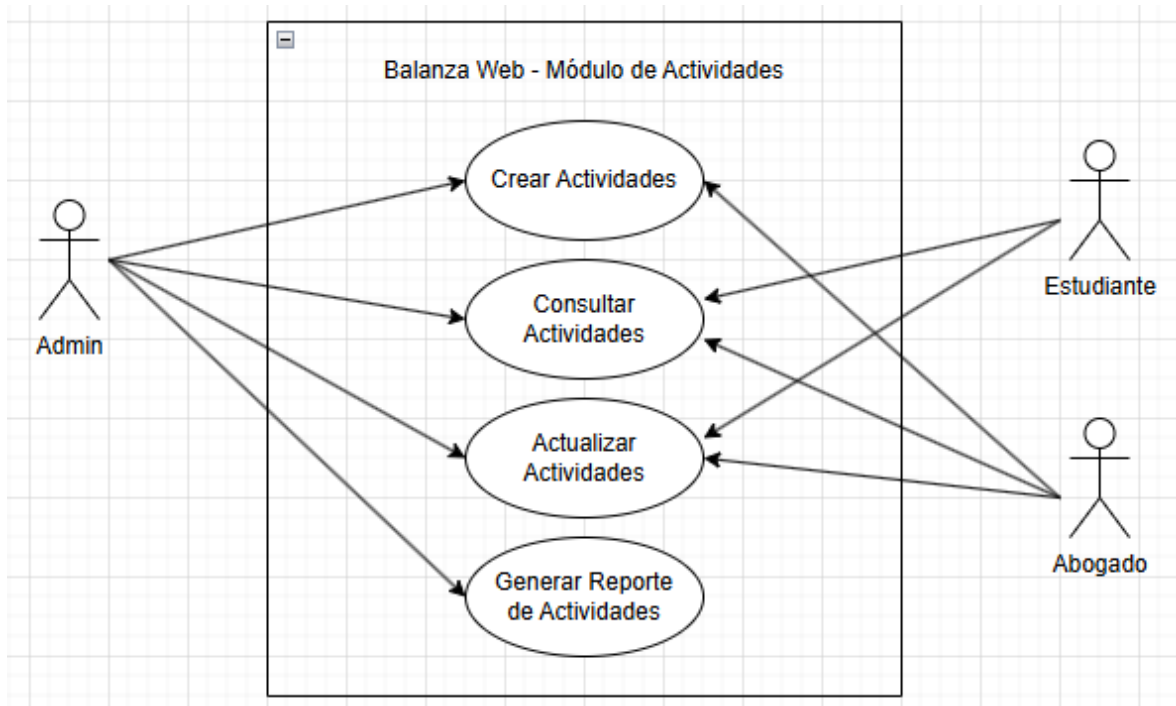
Tabla 1 - Descripción de Caso de Uso general de Módulo de Casos

Nombre del caso de uso: Caso de Uso Módulo de Casos		ID Único: CUG-01
Área: -		
Actor(es): Administrador, Estudiante		
Interesados: Estudiante		
Nivel: -		
Descripción: Permitir que un usuario con credenciales válidas y una cuenta activa pueda crear Casos nuevos, consultar la información de los casos y actualizar los mismos según las necesidades.		
Evento desencadenador: El actor navega a la página de Balanza Web (Login), introduce su correo electrónico y contraseña, y hace clic en el botón "Iniciar Sesión"		
Tipo de desencadenador: <input checked="" type="checkbox"/> Externo <input type="checkbox"/> Temporal		
Pasos realizados (ruta principal):		Información para los pasos
1. El actor ingresa al sistema Balanza Web con su correo y contraseña en el formulario de la página de login.		Correo electrónico, Contraseña
2. El sistema valida localmente que los campos no estén vacíos.		
3. El actor ejerce sus funciones normales normales de crear, actualizar y consultar el caso.		Información de sus tareas asignadas.
4. El sistema crea, cambia o trae la información según la acción realizada.		
Precondiciones: El actor debe tener una cuenta creada en el sistema con un rol específico de Administrador o Estudiante.		
Postcondiciones: El actor está autenticado en el sistema, se crea una sesión válida y se cargan sus permisos de rol.		
Suposiciones: El actor tiene acceso a un navegador web compatible y conexión a la red de la PUCE.		
Garantía de éxito: El actor visualiza la información de los casos, las opciones de edición o creación correspondientes a su rol.		
Garantía mínima: El sistema envía un mensaje de error o alerta al tratar de crear, consultar o actualizar la actividad o el actor pudo realizar cualquiera de las acciones descritas.		
Requerimientos cumplidos: Permitir a los usuarios visualizar los casos y su información relacionada según su rol y área y Visualizar el detalle de los casos, evidencias, información de los usuarios. etc.		
Cuestiones pendientes: ¿Se implementará una validación de datos correctos por cada campo?		
Prioridad: Alta		
Riesgo: Medio		

Nota: Esta tabla describe el funcionamiento general del módulo de casos, con el formato único para este caso de uso GENERAL. Fuente: Elaboración Propia.

3.2.2 Módulo de Actividades

Ilustración 3 Diagrama Caso de Uso Módulo Actividades



Nota: Elaboración propia. Diagrama de Casos de Uso del Módulo de Actividades con sus Actores.

Tabla 2- Descripción Caso de Uso General de Módulo de Actividades

Nombre del caso de uso: Caso de Uso Módulo de Actividades		ID Único: CUG-02
Área: -		
Actor(es): Administrador, Abogado, Estudiante		
Interesados: Usuario del Sistema		
Nivel: -		
Descripción: Permite al usuario consultar, crear y actualizar actividades asociadas a los casos jurídicos asignados, además de generar reportes por fecha.		
Evento desencadenador: El actor accede a la interfaz de casos, selecciona un caso y elige ver o registrar actividades.		
Tipo de desencadenador: <input checked="" type="checkbox"/> Externo <input type="checkbox"/> Temporal		
Pasos realizados (ruta principal):		Información para los pasos
1. El actor ingresa al sistema Balanza Web con su correo y contraseña en el formulario de la página de login.		Correo electrónico, Contraseña
2. El sistema valida localmente que los campos no estén vacíos.		
3. El actor navega hacia la pestaña “Mis Casos”		
4. El actor ejerce sus funciones normales normales de crear, actualizar y consultar las actividades dentro de un caso específico.		Información de sus tareas asignadas.
5. El sistema guarda, consulta o actualiza datos de las actividades.		
Precondiciones: El actor debe tener una cuenta activa y rol de Administrador, Abogado o Estudiante.		
Postcondiciones: Las actividades son almacenadas o actualizadas en la base de datos, asociadas a los casos.		
Suposiciones: El actor tiene acceso a un navegador web compatible y conexión a la red de la PUCE.		
Garantía de éxito: El actor puede crear, consultar o completar actividades según su rol, además de generar repo		
Garantía mínima: El sistema notifica errores si hay campos faltantes o datos incorrectos.		
Requerimientos cumplidos: Permitir a los usuarios crear y completar las actividades asociadas a cada caso y Generar reportes de actividades según fechas establecidas.		
Cuestiones pendientes: ¿Se integrará filtrado por estado de actividad o responsables?		
Prioridad: Alta		
Riesgo: Medio		

Nota: Esta tabla describe el funcionamiento general del módulo de actividades, con el formato único para este caso de uso GENERAL. Fuente: Elaboración Propia.

3.2.3 Módulo de Asignaciones

Ilustración 4 - Diagrama Caso de Uso Módulo Asignaciones

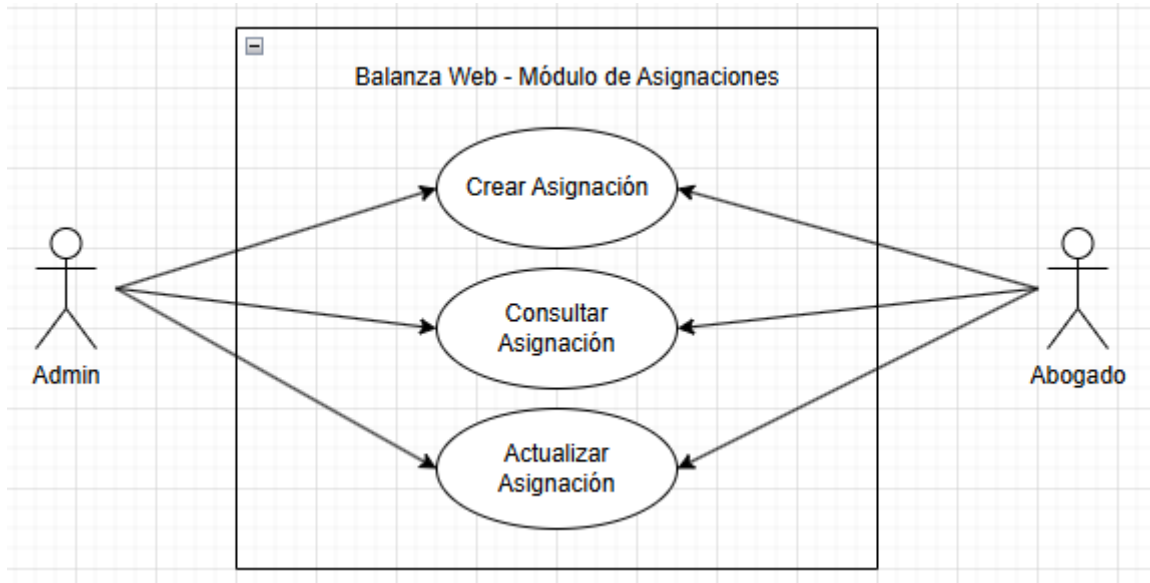


Tabla 3 - Descripción Caso de Uso General de Módulo de Asignaciones

Nombre del caso de uso: Caso de Uso Módulo de Asignaciones		ID Único: CUG-03
Área: -		
Actor(es): Administrador, Abogado		
Interesados: Usuario del Sistema		
Nivel: -		
Descripción: Permite asignar o reasignar casos a estudiantes, y consultar el estado de las asignaciones realizadas.		
Evento desencadenador: El actor accede a la opción “Asignar Casos” dentro del sistema.		
Tipo de desencadenador: <input checked="" type="checkbox"/> Externo <input type="checkbox"/> Temporal		
Pasos realizados (ruta principal):		Información para los pasos
1. El actor ingresa al sistema Balanza Web con su correo y contraseña en el formulario de la página de login.		Correo electrónico, Contraseña
2. El sistema valida localmente que los campos no estén vacíos.		
3. El actor navega hacia la pestaña “Asignar Casos”		
4. El actor ejerce sus funciones normales normales de asignar casos a un estudiante o reasignar casos a otro estudiante.		Información de sus tareas asignadas.
5. El sistema guarda, consulta o actualiza datos de las asignaciones.		
Precondiciones: El actor debe tener cuenta activa con rol de Administrador o Abogado.		
Postcondiciones: El sistema almacena o actualiza la asignación en la base de datos.		
Suposiciones: El actor tiene acceso a un navegador web compatible y conexión a la red de la PUCE.		
Garantía de éxito: El actor visualiza y gestiona correctamente las asignaciones.		
Garantía mínima: El sistema muestra errores si no se selecciona un estudiante o si se intenta reasignar al mismo usuario.		
Requerimientos cumplidos: Permitir a los usuarios asignar y reasignar casos y actividades a los estudiantes o abogados de los casos correspondientes.		
Cuestiones pendientes: ¿Se permitirá asignación múltiple o filtrado por disponibilidad de estudiantes?		
Prioridad: Alta		
Riesgo: Bajo		

Nota: Esta tabla describe el funcionamiento general del módulo de asignaciones, con el formato único para este caso de uso GENERAL. Fuente: Elaboración Propia.

3.3 Diagramas de Casos de Uso Específicos

3.3.1 Módulo de Casos

3.3.1.1 Caso de uso Crear Caso

Ilustración 5 - Diagrama Caso de Uso Crear Caso

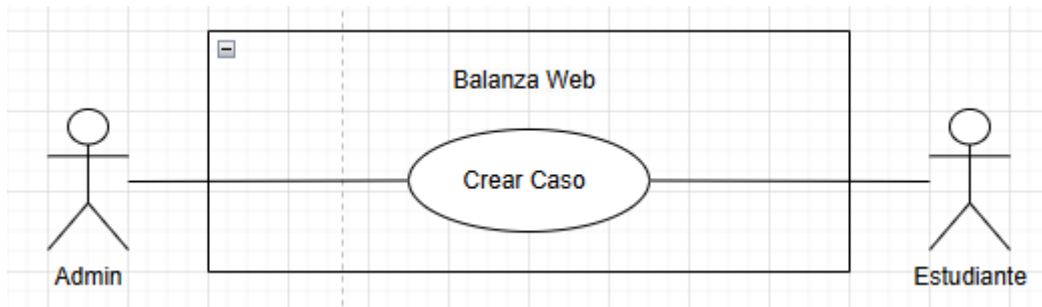


Tabla 4 - Caso de Uso Crear Caso

Nombre	Crear Caso
Descripción	Crea un caso con todos sus atributos
Actores	Administrador – Estudiante
Precondiciones	El usuario logueado debe ser de tipo Administrador o Estudiante
Flujo principal	<ol style="list-style-type: none">1. Inicia cuando un usuario estudiante o administrador ingresa al sistema.2. El usuario ingresa a la interfaz de “Nuevo Caso”.3. El usuario escribe en la barra de búsqueda una cédula o pasaporte para verificar si el usuario ya existe.4. Si no existe, el usuario ingresa todos los datos necesarios para el caso.5. El usuario guarda los datos del caso y del cliente con la opción “Crear Caso”.6. El sistema guarda los datos del caso y del cliente.

Postcondiciones	El Caso junto con el cliente son creados en la base de datos.
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario ingresa una cédula o pasaporte existentes. <ol style="list-style-type: none"> 1. El Sistema carga todos los datos del cliente y los casos ligados a él. • El usuario no ingresa datos obligatorios <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de datos obligatorios en los campos correspondientes. 2. El usuario ingresa los datos faltantes. 3. Volver al paso 5 del flujo principal.

3.3.1.2 Caso de Uso Consultar Caso

Ilustración 6 - Diagrama Caso de Uso Consultar Caso

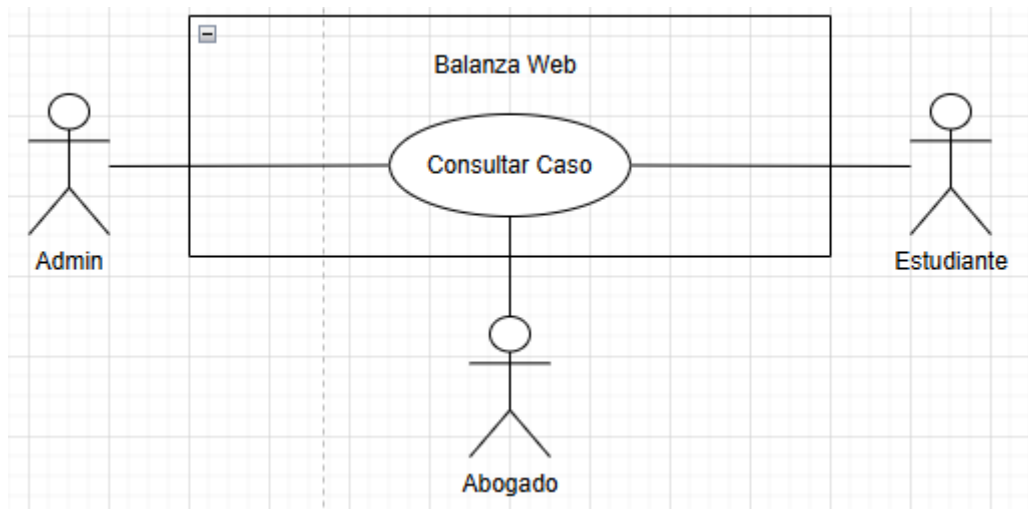


Tabla 5 - Caso de Uso Consultar Caso

Nombre	Consultar Caso
Descripción	Consulta uno o varios casos con todos sus atributos.
Actores	Administrador – Estudiante - Abogado
Precondiciones	<ul style="list-style-type: none"> • El usuario logueado debe ser de tipo Administrador, Estudiante o Abogado. • Debe existir el caso.
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario estudiante, abogado o administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Mis Casos”. 3. El sistema muestra todos los casos existentes para el usuario administrador, los casos asignados para los estudiantes y todos los casos de su área para el abogado.

	<ol style="list-style-type: none"> 4. El usuario ingresa en la barra de búsqueda la cédula del cliente, la fecha o el ID del caso. 5. El sistema trae los datos del caso.
Postcondiciones	Los datos del caso junto con los datos de las actividades y del cliente son cargados desde la base de datos.
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario ingresa una cédula, fecha o ID inexistentes. <ol style="list-style-type: none"> 2. El Sistema muestra una alerta de que no existen casos para los datos ingresados. 3. Vuelve al paso 4 del flujo principal.

3.3.1.3 Caso de uso Actualizar Caso

Ilustración 7 - Diagrama Caso de Uso Actualizar Caso

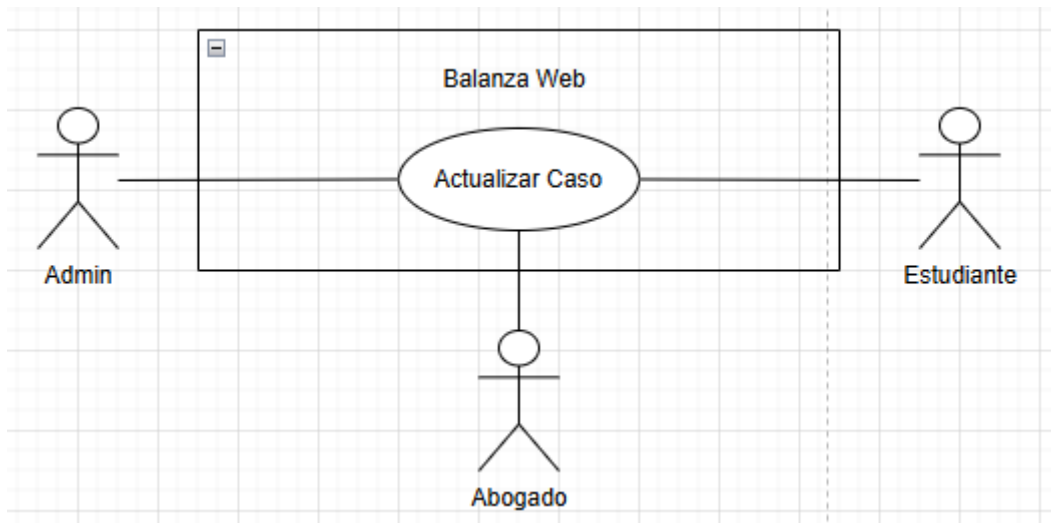


Tabla 6 - Caso de Uso Actualizar Caso

Nombre	Actualizar Caso
Descripción	Modifica los atributos de un caso
Actores	Administrador – Estudiante - Abogado
Precondiciones	El usuario logueado debe ser de tipo Administrador, Estudiante o Abogado
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario estudiante, abogado o administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Mis Casos”. 3. El sistema muestra todos los casos correspondientes según el rol. 4. El usuario ingresa al apartado de Finalizar Caso de un Caso deseado. 5. El sistema despliega un modal con atributos a llenar. 6. El usuario llena los atributos. 7. El usuario da click en la opción Finalizar.

	8. El sistema actualiza los campos del caso en la base de datos.
Postcondiciones	El caso se modifica en la base de datos.
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario no completa los campos requeridos para finalizar el caso. <ol style="list-style-type: none"> 1. El Sistema muestra una alerta de que todos los campos son obligatorios. 2. El usuario llena todos los campos. 3. Vuelve al paso 7 del flujo principal.

3.3.2 Módulo de Actividades

3.3.2.1 Caso de Uso Crear Actividad

Ilustración 8 - Diagrama Caso de Uso Crear Actividad

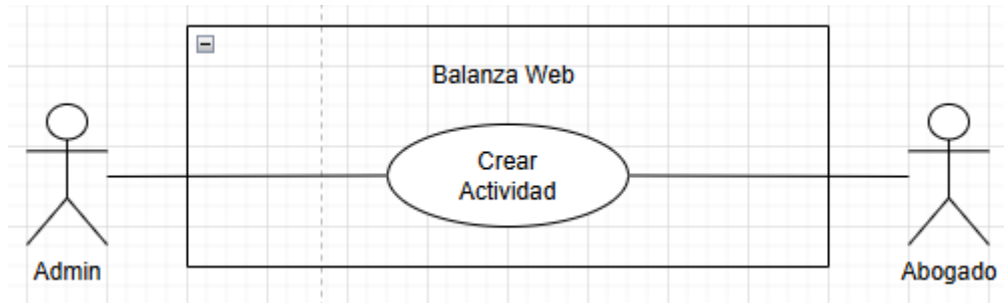


Tabla 7 - Caso de Uso Crear Actividad

Nombre	Crear Actividad
Descripción	Crea una Actividad con sus atributos.
Actores	Administrador – Abogado
Precondiciones	El usuario logueado debe ser de tipo Administrador o Abogado
Flujo principal	<ol style="list-style-type: none">1. Inicia cuando un usuario abogado o administrador ingresa al sistema.2. El usuario ingresa a la interfaz de “Crear Actividades”.3. El sistema muestra todos los casos correspondientes según el rol.4. El usuario selecciona la opción de “Ver Actividades” de un caso en específico.5. El sistema despliega un modal con las actividades y opciones.6. El usuario selecciona la opción “Crear nueva actividad”.7. El sistema presenta un modal con opciones para completar.8. El usuario ingresa todos los datos.

	<p>9. El usuario selecciona la opción “Crear Actividad”</p> <p>10. El sistema guarda los datos de la nueva actividad en la base de datos.</p>
Postcondiciones	La actividad se crea en la base de datos
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario no completa los campos requeridos para crear la actividad. <ol style="list-style-type: none"> 1. El Sistema muestra una alerta de que todos los campos son obligatorios. 2. El usuario llena todos los campos. 3. Vuelve al paso 20 del flujo principal.

3.3.2.2 Caso de Uso Consultar Actividad

Ilustración 9 – Diagrama Caso de Uso Consultar Actividad

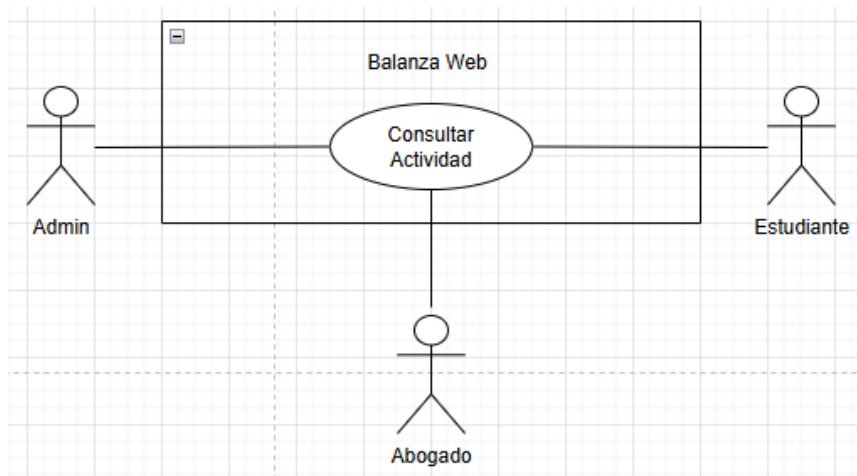


Tabla 8 - Caso de Uso Consultar Actividad

Nombre	Consultar Actividad
Descripción	Consulta una Actividad con sus atributos.
Actores	Administrador – Abogado - Estudiante
Precondiciones	El usuario logueado debe ser de tipo Administrador, Abogado o Estudiante
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario abogado, estudiante o administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Mis Casos”. 3. El sistema muestra todos los casos correspondientes según el rol. 4. El usuario selecciona la opción de “Ver Actividades” de un caso en específico. 5. El sistema despliega en un modal todas las actividades del caso seleccionado.
Postcondiciones	Las actividades son cargadas desde la base de datos.

Flujo Alternativo	<ul style="list-style-type: none">• El Caso seleccionado no tiene actividades. <ol style="list-style-type: none">1. El Sistema muestra una alerta de que no existen actividades para el Caso seleccionado.
-------------------	--

3.3.2.3 Caso de uso Actualizar Actividad

Ilustración 10 - Diagrama Caso de Uso Actualizar Actividad

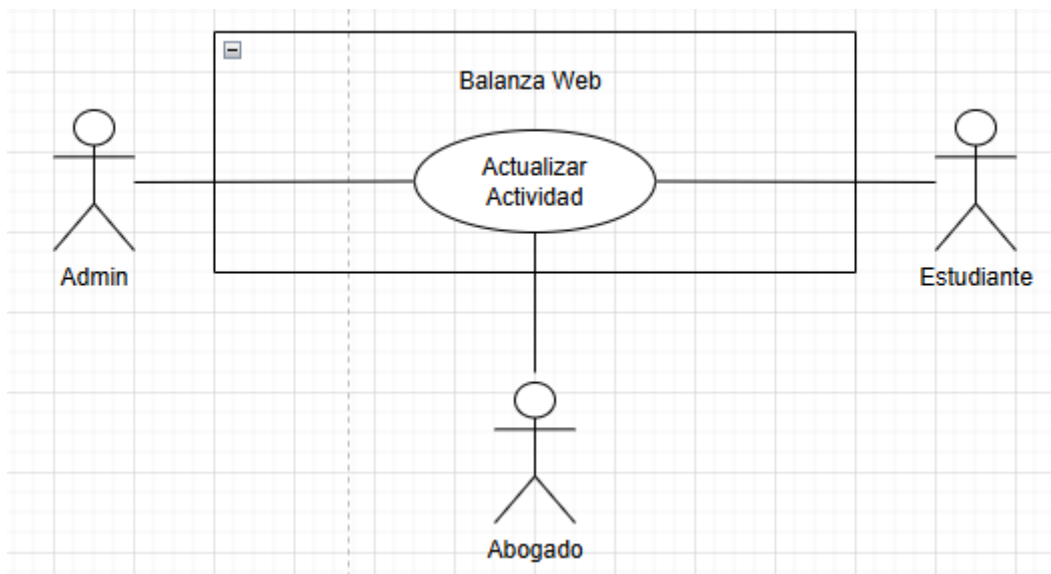


Tabla 9 - Caso de Uso Actualizar Actividad

Nombre	Actualizar Actividad
Descripción	Actualiza una Actividad con sus atributos.
Actores	Administrador – Abogado - Estudiante
Precondiciones	El usuario logueado debe ser de tipo Administrador, Abogado o Estudiante
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario abogado, estudiante o administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Mis Casos”. 3. El sistema muestra todos los casos correspondientes según el rol. 4. El usuario selecciona la opción de “Ver Actividades” de un caso en específico. 5. El sistema despliega en un modal todas las actividades del caso seleccionado.

	<ol style="list-style-type: none"> 6. El usuario selecciona la opción “Completar” en una actividad. 7. El sistema muestra un modal con campos a completar. 8. El usuario completa todos los campos correspondientes 9. El usuario selecciona la opción “Completar Actividad” 10. El sistema actualiza la actividad en la base de datos.
Postcondiciones	La actividad es actualizada con los nuevos datos en la base de datos.
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario no completó los campos obligatorios. <ol style="list-style-type: none"> 1. El sistema muestra una alerta de campos obligatorios para ser completados. 2. El usuario completa todos los campos obligatorios. 3. Vuelve al paso 9 del flujo principal.

3.3.2.4 Caso de Uso Generar Reporte de Actividades

Ilustración 11 – Diagrama Caso de Uso Generar reporte de Actividad

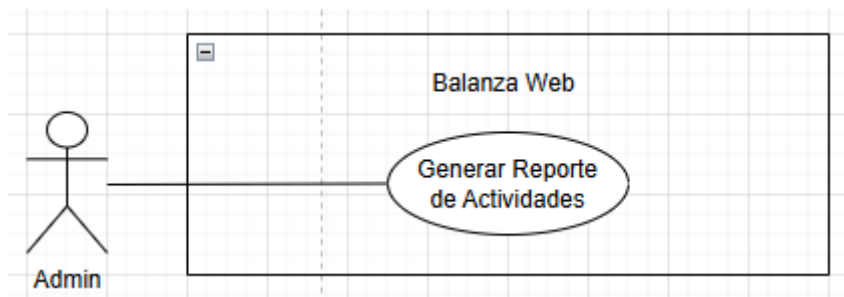


Tabla 10 - Caso de Uso Generar reporte de Actividad

Nombre	Generar Reporte de Actividades
Descripción	Genera un reporte de las actividades en un intervalo de fechas, con ciertos atributos de las actividades.
Actores	Administrador
Precondiciones	El usuario logueado debe ser de tipo Administrador.
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Generar Reportes”. 3. El sistema muestra la Interfaz de Generación de Reportes. 4. El usuario selecciona la opción de “Reporte de Actividades”. 5. El usuario selecciona un rango de fechas en las que quiere generar el reporte de las actividades. 6. El usuario selecciona la opción “Generar Reporte”. 7. El sistema carga la información desde la base de datos.

	<p>8. El sistema descarga un archivo Excel con la información necesaria de las actividades para los reportes.</p>
Postcondiciones	<p>La base de datos trae los datos y el Excel es descargado con la información.</p>
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario no seleccionó un intervalo de fechas. <ol style="list-style-type: none"> 1. El sistema muestra una alerta de que debe seleccionar 2 fechas para saber que actividades traer y descargar. 2. El usuario selecciona el intervalo de fechas. 3. Vuelve al paso 6 del flujo principal.

3.3.3 Módulo de Asignaciones

3.3.3.1 Caso de Uso Crear Asignación

Ilustración 12 – Diagrama Caso de uso Crear asignación

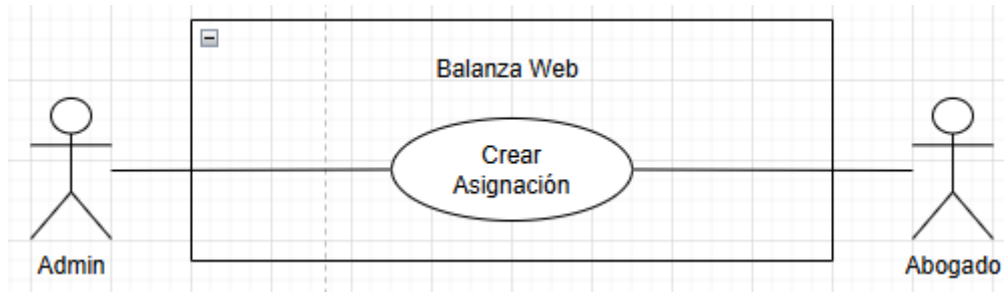


Tabla 11 - Caso de uso Crear asignación

Nombre	Crear Asignación
Descripción	Crea una asignación con todos sus atributos.
Actores	Administrador - Abogado
Precondiciones	El usuario logueado debe ser de tipo Administrador o Abogado.
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Asignar Casos”. 3. El sistema muestra la Interfaz de asignación de casos, con todos los casos sin asignar. 4. El usuario selecciona la columna de “Asignar a” de la tabla. 5. El usuario selecciona un estudiante de los desplegados. 6. El usuario selecciona la opción “Asignar” 7. El sistema crea la Asignación a ese estudiante en la base de datos.
Postcondiciones	La asignación se crea en la base de datos.

Flujo Alternativo	<ul style="list-style-type: none">• El usuario no seleccionó un estudiante para la asignación.<ol style="list-style-type: none">1. El sistema no habilita el botón de "Asignar".2. El usuario selecciona un estudiante para asignar.3. Vuelve al paso 6 del flujo principal.
-------------------	--

3.3.3.2 Caso de Uso Consultar Asignación

Ilustración 13 - Diagrama Caso de uso Consultar asignación

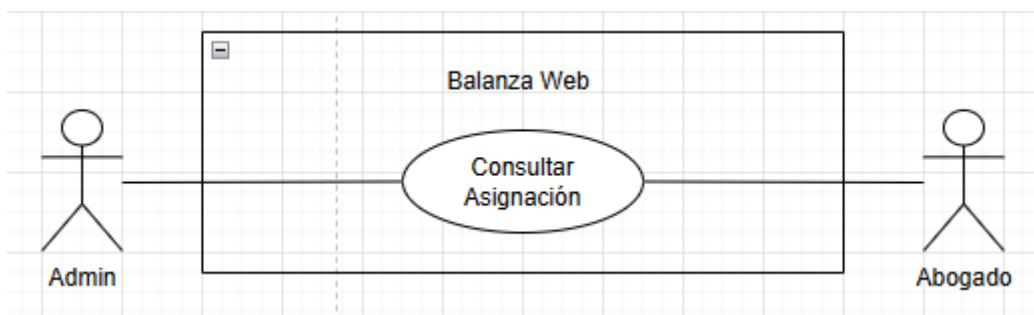


Tabla 12 - Caso de uso Consultar asignación

Nombre	Consultar Asignación
Descripción	Consulta las asignaciones con todos sus atributos.
Actores	Administrador - Abogado
Precondiciones	El usuario logueado debe ser de tipo Administrador o Abogado.
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Asignar Casos”. 3. El sistema muestra la Interfaz de asignación de casos. 4. El usuario selecciona la pestaña “Casos Asignados” de la tabla. 5. El sistema trae la información de las asignaciones de la base de datos. 6. El sistema muestra la información de las asignaciones.
Postcondiciones	La información de las asignaciones se trae de la base de datos.
Flujo Alternativo	<ul style="list-style-type: none"> • No existen asignaciones de ningún caso

	1. El sistema muestra que no existen casos asignados.
--	---

3.3.3.3 Caso de uso Actualizar Asignación

Ilustración 14 – Diagrama Caso de uso Actualizar Asignación

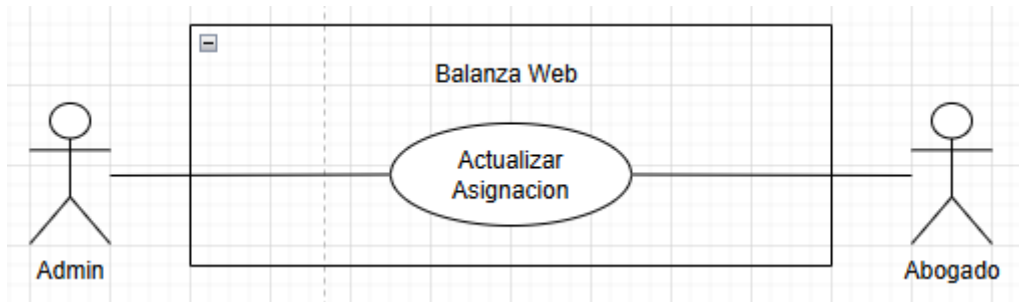


Tabla 13 - Caso de uso Actualizar Asignación

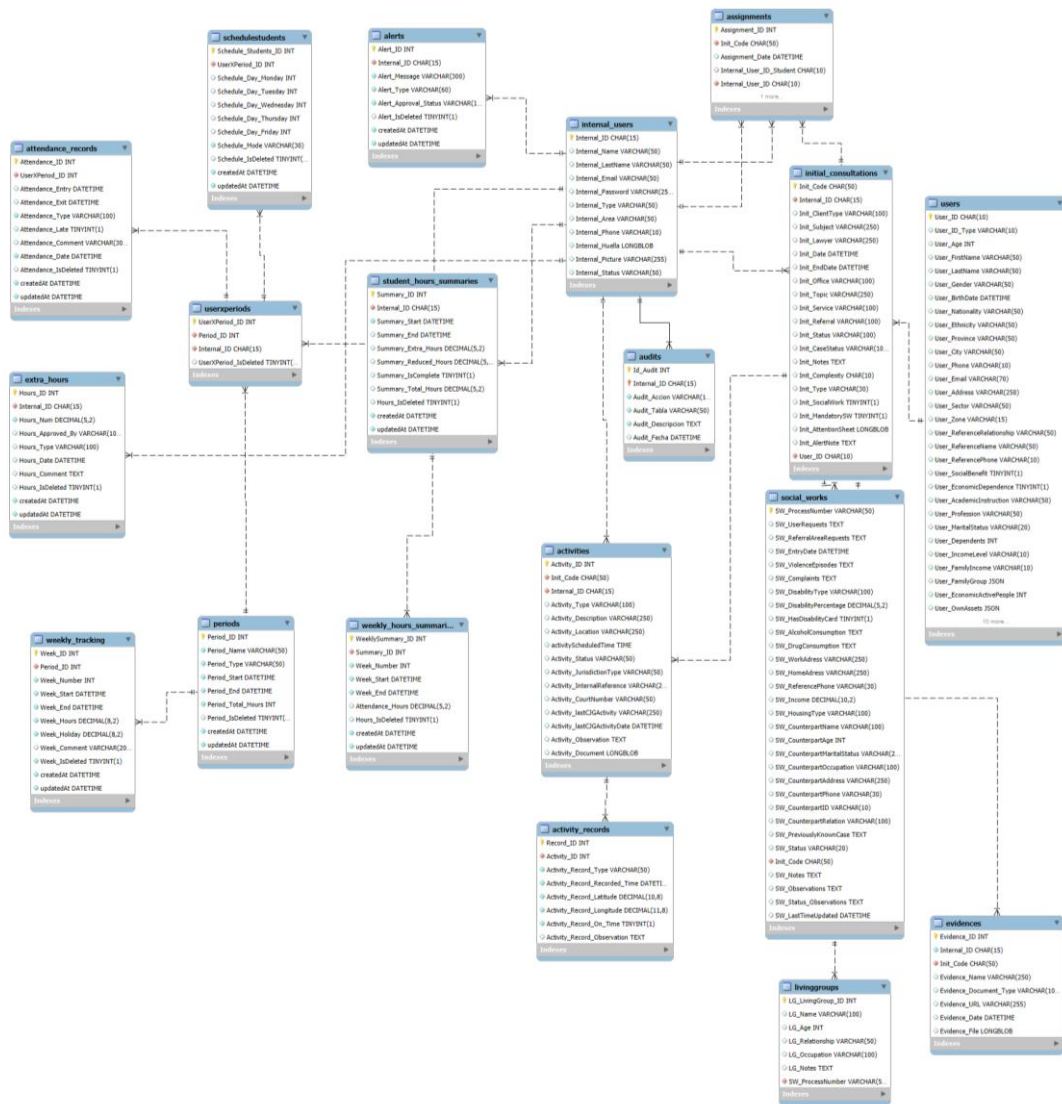
Nombre	Actualizar Asignación
Descripción	Actualiza una asignación con sus atributos.
Actores	Administrador - Abogado
Precondiciones	El usuario logueado debe ser de tipo Administrador o Abogado.
Flujo principal	<ol style="list-style-type: none"> 1. Inicia cuando un usuario administrador ingresa al sistema. 2. El usuario ingresa a la interfaz de “Asignar Casos”. 3. El sistema muestra la Interfaz de asignación de casos. 4. El usuario selecciona la pestaña “Casos Asignados” de la tabla. 5. El sistema trae la información de las asignaciones de la base de datos. 6. El sistema muestra la información de las asignaciones. 7. El usuario selecciona la opción Reasignar Caso

	<ol style="list-style-type: none"> 8. El sistema muestra un modal con campos a rellenar. 9. El usuario completa todos los datos. 10. El usuario selecciona la opción “Reasignar”. 11. El sistema actualiza la información de la asignación en la base de datos.
Postcondiciones	La información de la asignación se actualiza en la base de datos.
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario no completó campos obligatorios. <ol style="list-style-type: none"> 1. El sistema muestra una alerta de los campos que son obligatorios. 2. El usuario rellena todos los campos obligatorios. 3. Vuelve al paso 10 del flujo principal. • El usuario selecciona de los campos el mismo estudiante al que se le asignó el caso. <ol style="list-style-type: none"> 1. El sistema no habilita la función de “Reasignar” 2. El usuario escoge otro estudiante que no sea el mismo para reasignar. 3. Volver al paso 10 del flujo principal.

3.4 Modelado del Sistema

3.4.1 Diagrama Entidad-Relación

Ilustración 15 - Diagrama Entidad-Relación



Se diseñó un modelo entidad-relación que representa la estructura de datos del módulo de Gestión de Casos, Actividades y Asignaciones:

- Entidad:** Caso
Atributos: ID, Cliente_ID, Abogado_ID, Estado, Área, Descripción, Fecha de Creación, Fecha de Actualización.

- **Entidad:** Actividad
Atributos: ID, Caso_ID, Responsable_ID, Descripción, Estado, Fecha de Registro, Fecha de Finalización, Observaciones.
- **Entidad:** Usuario
Atributos: ID, Nombre, Apellido, Rol, Correo Electrónico, Contraseña, Estado, Fecha de Creación.
- **Entidad:** Asignación
Atributos: ID, Caso_ID, Estudiante_ID, Fecha de Asignación, Estado.

3.5 Diseño de la Arquitectura del Sistema

El sistema “Balanza Web” se desarrolló bajo una arquitectura de Modelo Vista-Controlador, lo que facilita tanto el desarrollo, como la gestión de los módulos, además de que hace la aplicación web sea altamente escalable. Para esto, se emplearon las siguientes tecnologías:

- **Frontend:** Vue.js, Typescript.
- **Backend:** Javascript, Node.js, Sequelize.
- **Base de Datos:** MySQL.
- **Control de Versiones:** Git y GitHub.

3.5.1 Diseño de la Base de Datos

Se definieron las tablas relevantes para soportar la gestión eficiente de los casos, las actividades y las asignaciones. La base de datos, con las entidades relevantes para los módulos de Gestión de Casos, Actividades y Asignaciones, garantiza la integridad referencial mediante claves foráneas entre las entidades **Caso, Actividad, Usuario y Asignación.**

Ilustración 16- Entidad Users en la base de datos



Ilustración 17- Entidad Assignments en la base de datos

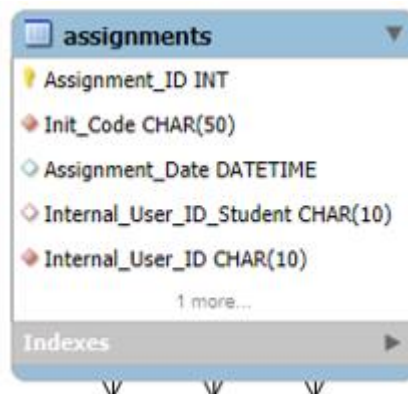
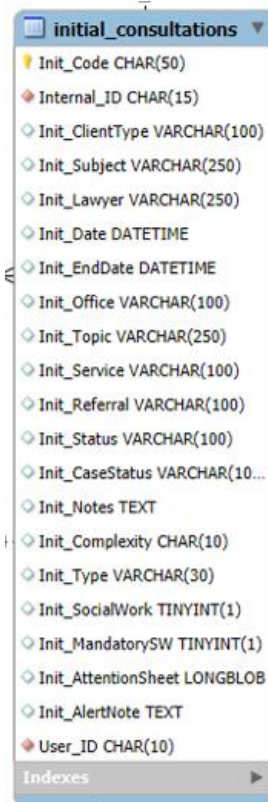


Ilustración 18- Entidad Activities en la base de datos



Ilustración 19- Entidad Initial_Consultations en la base de datos

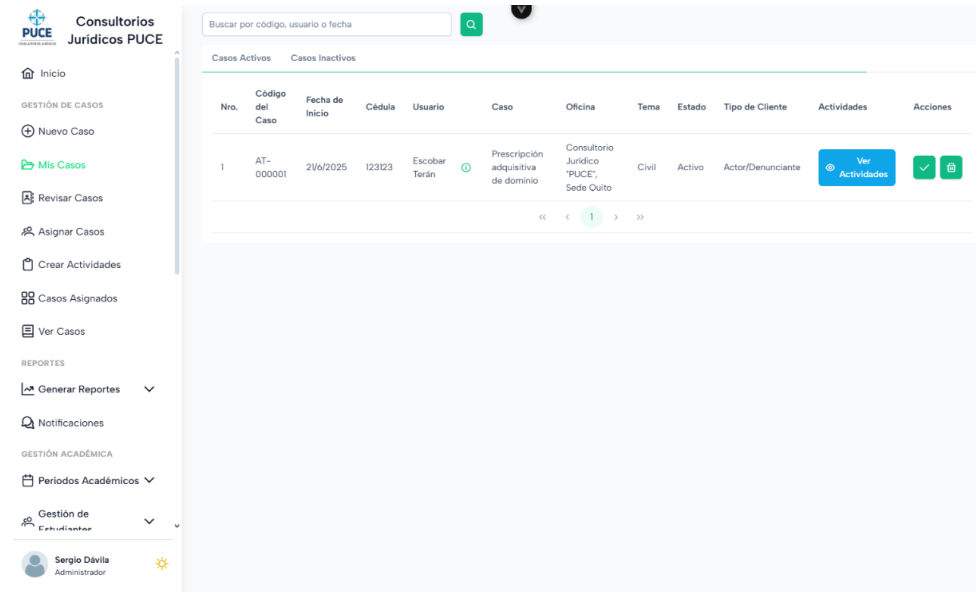


3.5.2 Diseño de la Interfaz de Usuario

Se diseñaron interfaces sencillas y accesibles gracias al Framework de desarrollo frontend Vue.js, con el fin de facilitar el uso del módulo:

- **Pantalla de Casos.**

Ilustración 20 - Pantalla de casos



- **Vista de Detalle y Actividades de un Caso.**

Ilustración 21 - Vista Actividades impuestas de un caso

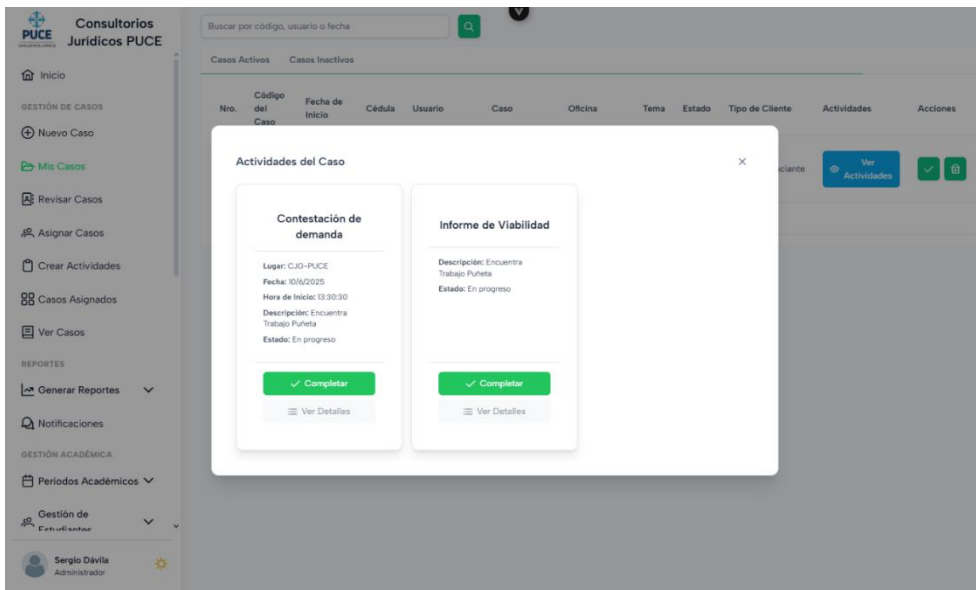


Ilustración 22 - Formulario para completar actividad

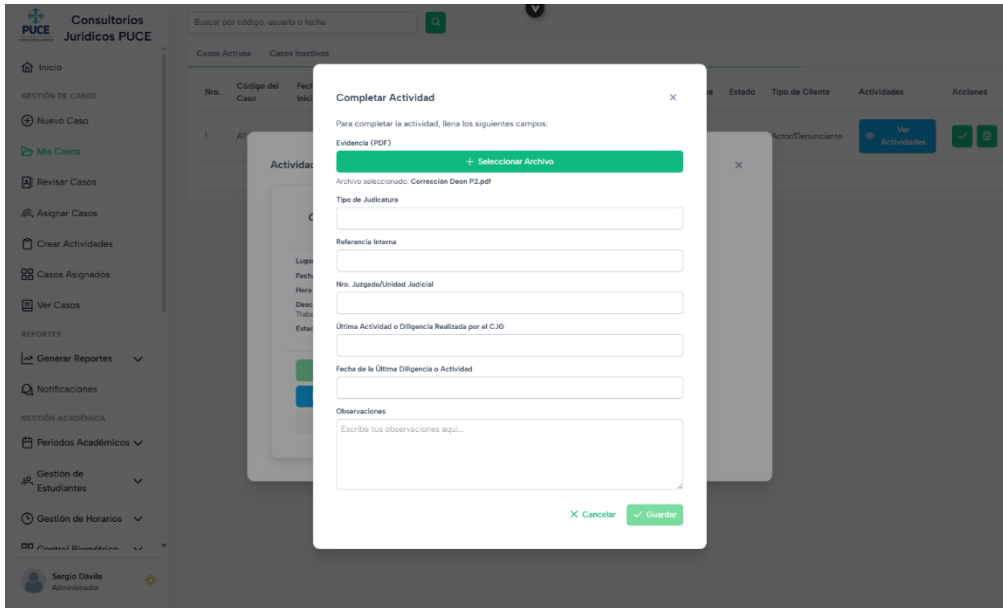
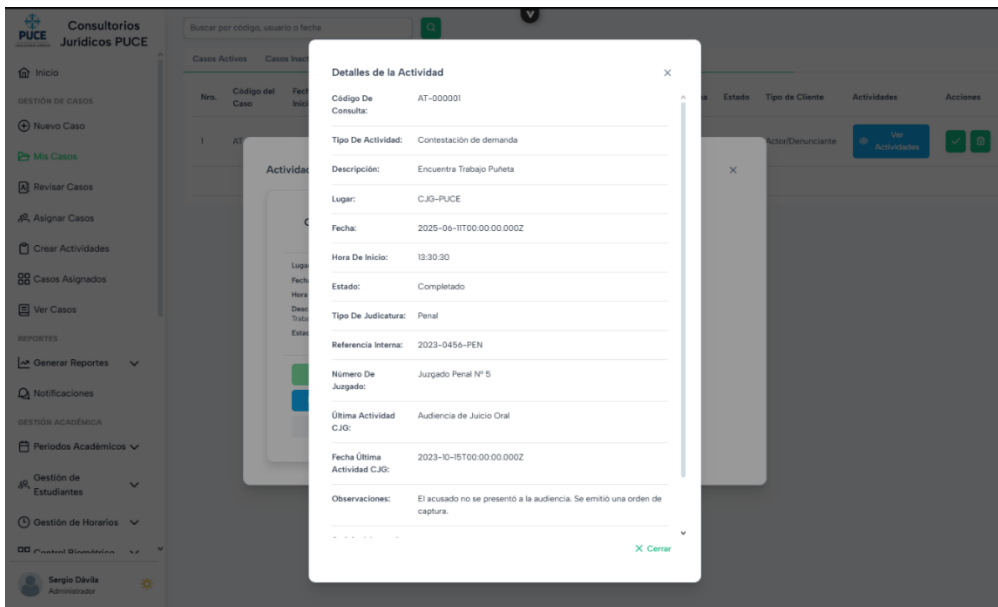


Ilustración 23 - Detalle de la actividad



- **Panel de Asignaciones.**

Ilustración 24- Panel de asignaciones pendientes

Consulterios Jurídicos PUCE

Inicio

GESTIÓN DE CASOS

- Nuevo Caso
- Mis Casos
- Revisar Casos
- Asignar Casos
- Crear Actividades
- Casos Asignados
- Ver Casos

REPORTES

- Generar Reportes
- Notificaciones

GESTIÓN ACADÉMICA

- Periodos Académicos
- Gestión de Estudiantes
- Gestión de Horarios
- Control Biométrico

Sergio Dávila
Administrador

Buscar caso...

Asignar automáticamente todos los casos

Casos por Asignar Casos Asignados

#	Código de Caso	Área	Estado	Complejidad	Asignar a	Acciones
No se encontraron casos para asignar que coincidan con los filtros.						

Ilustración 25 - Panel casos ya asignados

Consulterios Jurídicos PUCE

Inicio

GESTIÓN DE CASOS

- Nuevo Caso
- Mis Casos
- Revisar Casos
- Asignar Casos
- Crear Actividades
- Casos Asignados
- Ver Casos

REPORTES

- Generar Reportes
- Notificaciones

GESTIÓN ACADÉMICA

- Periodos Académicos
- Gestión de Estudiantes
- Gestión de Horarios
- Control Biométrico

Sergio Dávila
Administrador

Buscar caso...

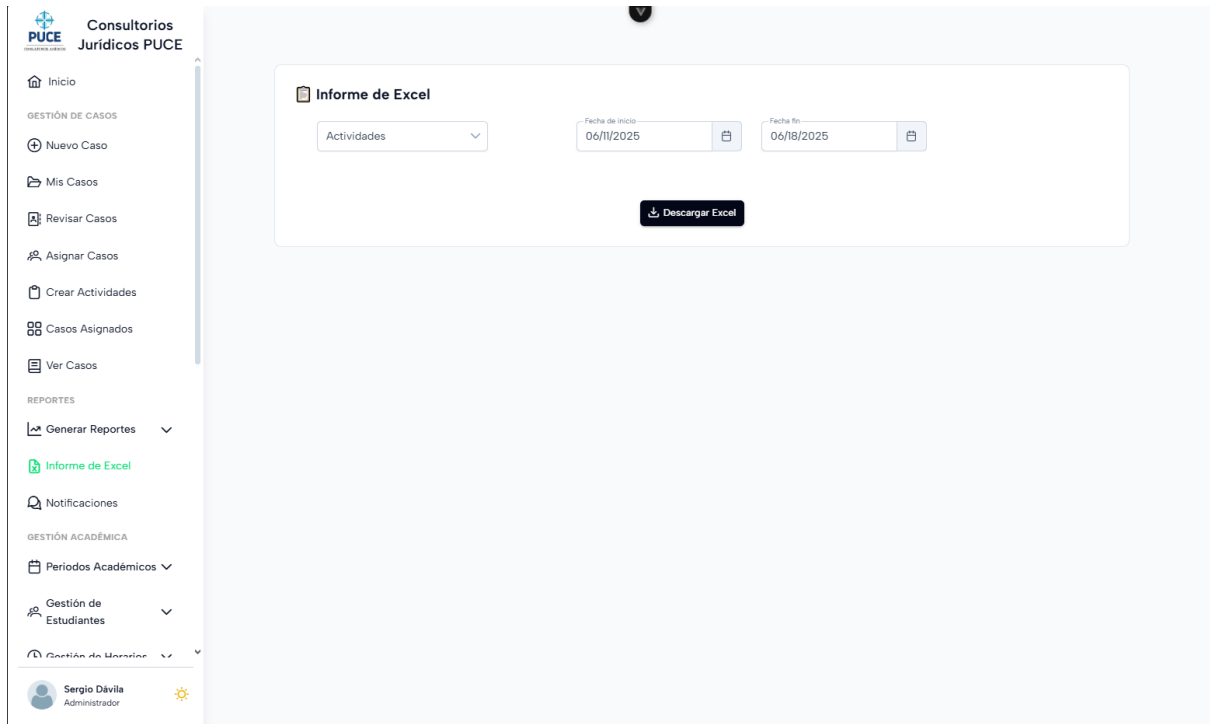
Asignar automáticamente todos los casos

Casos por Asignar Casos Asignados

#	Código de Caso	Área	Estado	Complejidad	Asignado a	Acciones
1	AT-000001	Civil	Asignado	Bajo	xaad xaad	

- **Pantalla de Reportes.**

Ilustración 26 - Pantalla de generación de reportes en Excel



Estas interfaces se implementaron siguiendo principios de usabilidad y accesibilidad, lo que permite que el sistema web sea usable desde diferentes dispositivos.

CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN

4.1 Preparación del Entorno de Desarrollo

Para garantizar un desarrollo ordenado y eficiente del sistema de los CJG de la PUCE, “Balanza Web”, se configuró un entorno de desarrollo acorde a las necesidades, que incluyó la instalación y configuración de las siguientes herramientas:

- **VS Code:** Entorno de desarrollo integrado, la cual brindó una alta compatibilidad con tecnologías modernas.
- **Node.js:** Entorno de ejecución para el backend.
- **MySQL:** Sistema gestor de bases de datos.
- **Git y GitHub:** Herramienta para el control de versionamiento del sistema y almacenamiento del código fuente.
- **Framework Vue.js:** Enfocado en la creación del frontend, herramienta la cual permitió un desarrollo ágil y modular.

La configuración del entorno con estas herramientas proporcionó un entorno que resultó siendo escalable, accesible y adecuado para las necesidades del sistema.

4.2 Estándares de Codificación

Para garantizar la calidad y consistencia del código en el desarrollo del sistema "Balanza Web", se siguieron los estándares establecidos por el Departamento de Informática de la PUCE, junto con buenas prácticas universales de codificación.

Lenguaje del código:

Todo el código fuente se escribió en inglés, incluyendo nombres de variables, funciones, clases y archivos, mientras que los comentarios se redactaron en español para facilitar la comprensión del equipo y de cualquier desarrollador que ingrese en futuras versiones.

Convenciones de nombrado:

- Las variables y funciones fueron escritas en camelCase.
- Las clases, componentes e interfaces fueron escritas en PascalCase.
- Los archivos y carpetas fueron escritas en snake_case.

Estructura del proyecto:

- Se realizó una modularización clara siguiendo la arquitectura MVC.
- Se separaron las capas (rutas, vistas, componentes)
- Se uso el tipado estático con TypeScript para una mayor robustez.

Directrices específicas del Departamento de Informática:

- Programación en inglés
- Uso de nombres descriptivos y autoexplicativos.
- Documentación interna con comentarios claros en español.
- Validación de tipos en tiempo de compilación.

Los anteriores estándares establecidos aseguraron un código mantenible, escalable, sobre todo al trabajar en equipo, además de que estuvo alineado con los requisitos institucionales.

4.3 Implementación de la Funcionalidad

El desarrollo se dividió en los 2 clásicos componentes, Backend y Frontend.

4.3.1 Backend

El backend fue desarrollado haciendo uso de Node.js, usando Sequelize como ORM para interactuar con la base de datos MySQL desde el sistema. Para el módulo de gestión de casos, actividades y asignaciones, las principales funcionalidades implementadas fueron:

- Creación, consulta y actualización de casos.
- Registro, seguimiento y finalización de actividades.

- Asignación y reasignación de casos a estudiantes.
- Generación de reportes en formato Excel para el módulo de actividades.

Tanto las rutas como los controladores fueron diseñados bajo la arquitectura REST, lo que consiguió una comunicación eficiente entre el servidor y el cliente.

4.3.2 Frontend

El frontend se desarrolló con el Framework de desarrollo de interfaces Vue.js y TailwindCSS, lo que ayudó en la creación de interfaces modernas y fáciles de usar. Las funcionalidades principales implementadas dentro de la interfaz de los módulos correspondientes son:

- Visualización de casos y actividades.
- Creación y actualización de casos y actividades.
- Asignación y consulta de casos a estudiantes.
- Generación de reportes por fechas.

Este diseño se enfocó en la simplicidad y usabilidad, para así asegurar una experiencia positiva para los usuarios de los CJG.

4.4 Plan de Pruebas

Para garantizar la calidad y el correcto funcionamiento del sistema “Balanza Web”, se elaboró un plan de pruebas exhaustivo, que, al ejecutarlo, permitió identificar errores, validar los requerimientos extraídos de los usuarios de los CJG y asegurar que el sistema cumpla con las necesidades de estos.

4.4.1 Tipos de Pruebas

Se aplicaron los siguientes tipos de pruebas para el sistema:

- **Pruebas Unitarias:**

Esta sección estuvo centrada en verificar de manera aislada el comportamiento de

las funciones, métodos y APIs de forma individual tanto en el backend como en el frontend.

- **Pruebas de Integración:**

Estas pruebas se realizaron para asegurar que los diferentes módulos del sistema, como backend, frontend y base de datos, se comuniquen de forma correcta y sin errores. Se simularon escenarios completos de flujo de datos entre los distintos componentes, con el equipo de desarrollo y con los usuarios finales de los CJG.

- **Pruebas Funcionales:**

Se realizaron para validar que el sistema cumpla con los requerimientos funcionales definidos. Aquí se incluyó la validación de las interfaces de usuario, flujo de navegación y respuesta del sistema ante diferentes acciones.

- **Pruebas de Usabilidad:**

Estas pruebas fueron realizadas con los usuarios reales de los CJG para determinar la facilidad de uso, la claridad de las interfaces y la comprensión de los procesos por parte de los estudiantes, abogados y administradores.

- **Pruebas de Seguridad:**

Se hicieron algunas validaciones básicas para identificar posibles vulnerabilidades como accesos no autorizados o exposición de datos sensibles.

4.4.2 Criterios de Aceptación

Para que un módulo sea considerado como aceptado y listo para su despliegue, tuvo que cumplir con los siguientes criterios de forma individual:

- Superar correctamente el 100% de los casos de prueba funcionales.
- No presentar errores críticos o bloqueantes para el flujo principal de trabajo.
- Cumplir con los requisitos funcionales y no funcionales establecidos.
- Responder adecuadamente ante entradas válidas e inválidas.
- Contar con la validación y aprobación de los usuarios clave durante las pruebas de aceptación.

- El sistema debía ser accesible y funcional en los principales navegadores web: Google Chrome, Mozilla Firefox, Microsoft Edge y Opera.

4.4.3 Casos de Prueba

Se realizaron pruebas específicas para los módulos de Casos, Actividades y Asignaciones.

4.4.3.1 Casos de Prueba Módulo de Casos

Tabla 14- Casos de Prueba Módulo de Casos

Caso de Prueba	Entrada	Resultado Esperado	Estado
Crear Caso	Datos completos: cédula, nombres, contacto, descripción del caso, área.	El caso y el cliente se registran exitosamente en la base de datos. El sistema muestra confirmación y el nuevo caso es visible en la interfaz.	Superado
Crear Caso con datos incompletos	Solo cédula, sin ingresar otros campos obligatorios.	El sistema bloquea la acción y muestra mensajes de advertencia en los campos vacíos.	Superado
Consultar Caso	Cédula, fecha o ID válidos de un caso existente.	El sistema muestra toda la información del caso, incluyendo datos del cliente, descripción, estado y actividades.	Superado
Consultar Caso inexistente	Número de cédula o ID que no existe en la base de datos.	El sistema informa que no existen resultados, sin generar errores ni fallos en la interfaz.	Superado
Actualizar Caso	Modificación en la descripción, estado u otros campos.	Los datos se actualizan correctamente en la base de datos y el sistema muestra un mensaje de confirmación.	Superado

4.4.3.2 Casos de Prueba Módulo de Casos

Tabla 15 - Casos de Prueba Módulo de Casos

Caso de Prueba	Entrada	Resultado Esperado	Estado
Crear Actividad	Selección de caso existente, descripción, fecha y estado de la actividad.	La actividad se registra exitosamente y aparece en la lista de actividades del caso.	Superado
Crear Actividad con campos vacíos	Se omite la descripción o fecha de la actividad.	El sistema impide la creación y muestra mensajes de error en los campos obligatorios.	Superado
Consultar Actividades	Caso con actividades previamente registradas.	El sistema lista todas las actividades asociadas al caso, mostrando detalles como descripción, fecha y estado.	Superado
Consultar Actividades sin registros	Caso sin actividades asociadas.	El sistema informa que no existen actividades para el caso seleccionado.	Superado
Completar Actividad	Selección de actividad pendiente, ingreso de los datos requeridos.	La actividad se actualiza correctamente, el estado cambia a "Finalizada" y los datos quedan guardados en la base de datos.	Superado

4.4.3.3 Casos de Prueba Módulo de Reportes enfocado a Actividades

Tabla 16 - Casos de Prueba Módulo de Reportes enfocado a Actividades

Caso de Prueba	Entrada	Resultado Esperado	Estado
Generar Reporte de Actividades	Selección de un rango de fechas válido.	El sistema genera y descarga un archivo Excel con las actividades realizadas en el período seleccionado, mostrando información estructurada.	Superado
Generar Reporte sin seleccionar fechas	Se intenta generar el reporte sin definir el rango de fechas.	El sistema bloquea la acción y muestra un mensaje indicando que se deben seleccionar las fechas obligatorias.	Superado

4.4.3.4 Casos de Prueba Módulo de Asignaciones

Tabla 17 - Casos de prueba Módulo de Asignaciones

Caso de Prueba	Entrada	Resultado Esperado	Estado
Crear Asignación	Selección de un caso sin asignar y un estudiante disponible.	El caso se asigna correctamente al estudiante y queda registrado en la base de datos.	Superado
Crear Asignación sin seleccionar estudiante	Se intenta asignar un caso sin elegir un estudiante.	El sistema bloquea la acción y deshabilita el botón de "Asignar" hasta que se seleccione un estudiante válido.	Superado
Reasignar Caso	Selección de un caso previamente asignado y un nuevo estudiante.	El caso se reasigna correctamente al nuevo estudiante y el sistema actualiza la base de datos.	Superado
Reasignar Caso al mismo estudiante	Se intenta reasignar el caso	El sistema bloquea la acción, deshabilita la opción de "Reasignar"	Superado

	al mismo estudiante.	hasta que se seleccione un estudiante diferente.	
--	----------------------	--	--

4.5 Post Mortem

El desarrollo del sistema para los CJG de la PUCE, “Balanza Web”, representó un gran desafío, a la vez que una experiencia enriquecedora tanto a nivel técnico como profesional. El siguiente análisis Post Mortem permitirá identificar tanto los aspectos positivos como los desafíos enfrentados y las lecciones aprendidas durante el ciclo de vida del proyecto.

Aspectos Positivos

- **Definición clara de objetivos:** Los objetivos fueron establecidos de manera precisa desde el inicio del proyecto, lo que permitió enfocar los esfuerzos en resolver las necesidades reales de los Consultorios Jurídicos de la PUCE.
- **Metodología adecuada:** El uso de la metodología de Prototipado facilitó de gran manera la obtención de retroalimentación temprana por parte de los usuarios pertinentes, lo que permitió realizar ajustes e implementar nuevas funcionalidades requeridas a lo largo del ciclo de vida del proyecto.
- **Uso de tecnologías modernas:** El uso en conjunto de tecnologías como Vue.js, Node.js, TypeScript y MySQL resultó adecuada para garantizar el rendimiento, escalabilidad y mantenibilidad del sistema, además de alinearse a los estándares de la Dirección de Informática de la PUCE.
- **Trabajo colaborativo:** Git y GitHub fueron indispensables en el desarrollo del proyecto y permitieron un control de versiones eficiente. Esto ayudó a evitar conflictos en el código y mantuvo un historial de los cambios.
- **Validación con usuarios reales:** La ejecución de las pruebas con usuarios finales nos permitió validar la funcionalidad del sistema en un entorno real, lo que garantizó su usabilidad y eficacia, además de validar los requisitos funcionales y no funcionales establecidos.

Dificultades Encontradas

- **Incertidumbre en requisitos iniciales:** Al inicio del proyecto existió cierta ambigüedad y falta de explicación en algunos requisitos funcionales, lo que generó algunos retrabajos en las primeras etapas de desarrollo, además los ciertos requisitos funcionales que fueron surgiendo a medida que avanzaba el proyecto.
- **Falta de familiaridad con algunas herramientas:** Al inicio del proyecto hubo una curva de aprendizaje mediana en tecnologías como Vue.js y Typescript, por lo que todo el equipo de desarrollo requirió una inversión adicional de tiempo para su aprendizaje.
- **Limitaciones de tiempo:** Los plazos establecidos inicialmente representaron un desafío para cumplir con todas las fases de desarrollo, pruebas y documentación, sobre todo por la aparición de nuevos requisitos funcionales y limitaciones de infraestructura por parte de la Dirección de Informática de la PUCE.
- **Comunicación con usuarios finales:** En algunos momentos, coordinar reuniones o validaciones con los usuarios finales resultó complicado debido a sus propias cargas, tanto académicas como laborales.

Lecciones Aprendidas

- La planificación inicial es una parte fundamental, y esta debe ser flexible para adaptarse a cambios que surgen durante el desarrollo, sobre todo en proyectos donde los usuarios descubren necesidades adicionales en el camino.
- El uso de prototipos funcionales y las pruebas de validación continuas con usuarios reales nos aportan gran valor, ya que nos permitió detectar errores y ajustar algunas funcionalidades antes de que el sistema esté en su etapa final.
- El proceso de documentación constante y el control de versiones son elementos importantes que aseguran la calidad del código y además le brinda continuidad al proyecto, sobre todo si otros desarrolladores desean involucrarse en el futuro.
- La capacitación previa en algunas tecnologías no vistas puede ayudar a optimizar los tiempos de desarrollo y evitar contratiempos durante la implementación.

En general, el desarrollo del sistema para los CJG, “Balanza Web”, cumplió con los objetivos planteados y aportó una solución efectiva para la gestión de los Consultorios Jurídicos Gratuitos de la PUCE, aunque existen módulos a los cuales es posible realizar mejoras para futuras versiones. A pesar de que se enfrentaron desafíos técnicos y organizativos, los mismos permitieron fortalecer habilidades y mejorar la capacidad de respuesta ante situaciones imprevistas. Este proyecto o desarrollo no solo representa un aporte tecnológico, sino también un proceso de aprendizaje que da algunas bases sólidas para futuros desarrollos.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La creación del sistema "Balanza Web" para los Consultorios Jurídicos Gratuitos de la PUCE facilitó la automatización y mejora de los procesos esenciales vinculados a la administración de casos, tareas y responsabilidades. Durante el desarrollo del proyecto se lograron alcanzar las metas propuestas, demostrando la importancia y eficacia de una herramienta tecnológica que contribuya tanto al sector académico como al trabajo social de la Universidad.

A través de la implementación del método de Prototipado y la utilización de tecnologías actuales como Vue.js, Node.js, TypeScript y MySQL, se logró establecer un sistema operativo, accesible y escalable, acorde a las demandas particulares de los CJG. La estructura Modelo Vista Controlador simplificó la estructuración y la conservación del código.

Las evaluaciones efectuadas evidenciaron que el sistema es estable, sencillo e intuitivo, satisfaciendo los requerimientos tanto funcionales como no funcionales establecidos. Además, la participación de usuarios reales confirmó que la solución incrementa notablemente la rastreabilidad de los casos y la distribución de tareas, mejorando el tiempo de trabajo y disminuyendo los fallos operativos.

Además, el proceso de desarrollo demostró la relevancia de tener una documentación técnica exhaustiva, además de adherirse a buenas prácticas de codificación, lo que asegura la sostenibilidad y el progreso futuro del sistema.

5.2 Recomendaciones

Para garantizar el adecuado desempeño y progreso del sistema "Balanza Web" a largo plazo, se ofrecen las siguientes sugerencias:

- **Formación constante:** Es fundamental llevar a cabo sesiones de capacitación regulares para los usuarios finales con el objetivo de que utilicen todas las características del sistema.
- **Conservación progresiva:** Tomar en cuenta futuras mejoras y actualizaciones del sistema, dependiendo de las demandas que puedan presentarse en los Consultorios Jurídicos y de los progresos tecnológicos.
- **Supervisión y retroalimentación:** La implementación de sistemas de seguimiento y recolección de propuestas de los usuarios facilitará la identificación de áreas de mejora.
- **Protección de los datos:** Potenciar los sistemas de protección, tanto en el ámbito de la infraestructura como en el de la aplicación, para salvaguardar la información delicada de los casos y los usuarios.
- **Escalabilidad:** Investigar a futuro la incorporación del módulo a otros sistemas institucionales de la PUCE, con el fin de alcanzar una interoperabilidad y eficacia superiores.

Bibliografía:

GitHub. (s.f.). *About GitHub and Git*. GitHub Docs. Recuperado el 1 de junio de 2025, de <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>

Inesdi. (2022, 7 de febrero). *TypeScript vs JavaScript: ¿Cuál deberías aprender primero?* Inesdi Digital Business School. <https://www.inesdi.com/blog/typescript-vs-javascript/>

W3Schools. (s.f.). *Node.js Introduction*. W3Schools. Recuperado el 1 de junio de 2025, de https://www.w3schools.com/nodejs/nodejs_intro.asp

Vue.js. (s.f.). *Guía oficial Vue 3 en español*. Vue 3 Documentation (Traducción no oficial). Recuperado el 1 de junio de 2025, de <https://vue3-spanish.netlify.app/guide/introduction.html>

Recluit. (s.f.). *¿Qué es Visual Studio Code?* Recluit. Recuperado el 1 de junio de 2025, de <https://recluit.com/que-es-visual-studio-code/>

Oracle. (s.f.). *¿Qué es MySQL?* Oracle. Recuperado el 1 de junio de 2025, de <https://www.oracle.com/ar/mysql/what-is-mysql/>

García, O. (2013, septiembre 2). *Modelo de prototipos*. Proyectum. <https://proyectum.com/sistema/blog/modelo-de-prototipos/>

Los 10 principios de usabilidad para diseño de interfaces de usuario. (2021, septiembre 9). KSchool, tu centro de estudios especializado en internet. <https://kschool.com/blog/usabilidad-ux/los-10-principios-de-usabilidad-para-diseno-de-interfaces-de-usuario/>

Plan de pruebas de software: Claves para asegurar la calidad. (s/f). Qalified.com. Recuperado el 8 de julio de 2025, de <https://qalified.com/es/blog/plan-pruebas-software/>

¿Qué es el middleware? (s/f). Amazon.com. Recuperado el 8 de julio de 2025, de <https://aws.amazon.com/es/what-is/middleware/>