



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**SISTEMA DE PRÉSTAMOS PARA EL DEPARTAMENTO DE LTIC EN LA
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**Proyecto de titulación previo a la obtención del título de: Tecnólogo Superior en
Desarrollo de Software**

Autor: Pablo Andrés Brito Chicaiza

Tutor: Carlos Miguel Cárdenas Riofrio

Quito, Ecuador

2026

Dedicatoria

Pablo Brito:

Dedicó esta tesis a mis amigos, a todos los que estuvieron acompañándome en este proceso académico, a los que estuvieron conmigo en la universidad, y mis amigos fuera de ella, a mi amiga Antonella también, que me apoyó como nadie en mi proceso académico.

Deseo también dedicar esta tesis a mis abuelos y padres, que confiaron siempre en mí y me apoyaron incansablemente, en especial mis padres, muchas gracias por ser depositar su confianza en mí y ser mi pilar en este proceso. Los quiero a todos.

Tabla de contenido

Dedicatoria	2
Introducción	7
1. Contexto general	7
Capítulo I	9
Levantamiento de Requisitos y Diseño del Sistema	9
1.1 Levantamiento de requisitos	9
1.1.1 Requisitos funcionales	9
1.1.2 Requisitos no funcionales	12
1.2 Diseño del sistema	14
1.2.1 Arquitectura general	15
1.2.2 Modelado del sistema	16
1.3 Herramientas y tecnologías utilizadas	19
1.4 Metodología	21
1.5 Alcance y limitaciones del sistema	22
Capítulo II	24
Construcción del Sistema	24
2.1 Estructura general del desarrollo	24
2.2 Organización del proyecto	25
2.3 Desarrollo del frontend	27
2.4 Desarrollo del backend	30
2.5 Integración y conexión entre capas	32
2.5.1 Ejemplos de integración entre capas	33
Capítulo III	35
Pruebas y Estabilización	35
3.1 Pruebas funcionales del sistema	35
3.2 Pruebas de seguridad y autenticación	37
3.3 Pruebas de rendimiento	38
3.4 Validación en diferentes navegadores	39
3.5 Pruebas de usabilidad con usuarios del LTIC	40
Conclusiones	42
Recomendaciones	43
Referencias bibliográficas	44

DECLARACIÓN y AUTORIZACIÓN

Yo, **Pablo Andrés Brito Chicaiza** con C.I. 1755693817 autor(a) del trabajo de titulación intitulado: “Sistema de préstamos para el departamento de LTIC en la pontificia universidad católica del ecuador”, previa a la obtención del título de **Tecnología en desarrollo de software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 2 de febrero del 2026



Pablo Andrés Brito Chicaiza

C.I. 1755693817

Índice de figuras:

<i>Ilustración 1</i>	<i>Fig 1.- Un esquema que muestre cómo interactúan el Modelo, la Vista y el Controlador.....</i>	<i>15</i>
<i>Ilustración 2</i>	<i>Fig 2.-Diagrama de casos de uso (Casos: iniciar sesión, editar/registrar equipo, añadir estudiantes, reservar equipo, ver historial)</i>	<i>17</i>
<i>Ilustración 3</i>	<i>Fig 3.- Modelo entidad-relación del sistema.....</i>	<i>18</i>
<i>Ilustración 4</i>	<i>Fig 4.- Orden de las carpetas</i>	<i>26</i>
<i>Ilustración 5</i>	<i>Fig 5.- Formulario del login</i>	<i>27</i>
<i>Ilustración 6</i>	<i>Fig 6.- Pantalla principal del sistema</i>	<i>27</i>
<i>Ilustración 7</i>	<i>Fig 7.- Formulario nuevo préstamo.....</i>	<i>28</i>
<i>Ilustración 8</i>	<i>Fig 8.- Historial de préstamos, con filtros específicos en la parte superior.....</i>	<i>28</i>
<i>Ilustración 9</i>	<i>Fig 9.- Pruebas de control de errores (si no se llena un campo obligatorio).....</i>	<i>37</i>

Introducción

1. Contexto general

En la actualidad, la integración de las tecnologías de la información y comunicación constituye un elemento fundamental para el funcionamiento y desarrollo de las instituciones educativas superiores. La transformación digital ha permeado todos los ámbitos de la gestión universitaria, desde los procesos académicos hasta los servicios de soporte técnico, generando la necesidad de contar con plataformas eficientes que faciliten el acceso y la administración de recursos tecnológicos. En este contexto, los departamentos encargados de brindar soporte tecnológico desempeñan un papel estratégico, no solo como proveedores de equipos y servicios, sino como facilitadores del proceso de enseñanza-aprendizaje en entornos cada vez más digitalizados.

Problemática y solución propuesta:

El Laboratorio de Tecnologías de la Información y Comunicación (LTIC) de la Pontificia Universidad Católica del Ecuador (PUCE) constituye una unidad de soporte fundamental para la comunidad académica, proporcionando servicios de préstamo de equipos portátiles, mantenimiento de dispositivos tecnológicos y asistencia técnica especializada, principalmente a estudiantes de carreras tecnológicas y de ingeniería. Sin embargo, el sistema actual presenta limitaciones significativas: ausencia de historial de préstamos que permita el rastreo de equipos y responsables, carencia de mecanismos de búsqueda avanzada y filtrado, e interfaz desactualizada que dificulta la interacción eficiente con la plataforma.

Frente a esta problemática, el presente proyecto propone el rediseño integral de la plataforma web del LTIC mediante la implementación de tecnologías contemporáneas como Laravel, PHP, JavaScript y CSS. El sistema incorporará un módulo de historial completo con registro detallado de transacciones, funcionalidades de búsqueda avanzada con filtrado por múltiples criterios, ordenamiento alfabético y cronológico, además de una interfaz rediseñada bajo principios modernos de usabilidad y experiencia de usuario.

Objetivos del proyecto

El objetivo general del proyecto consiste en diseñar y desarrollar una plataforma web moderna, eficiente y organizada que mejore sustancialmente la administración de recursos del LTIC, al tiempo que brinde una experiencia de usuario optimizada y acorde con las necesidades actuales de estudiantes y personal administrativo. Se espera que la implementación de este sistema genere beneficios tangibles en términos de reducción de tiempos de procesamiento de solicitudes, mejora en el control de inventario, incremento en la trazabilidad de las operaciones, y mayor satisfacción de los usuarios del servicio.

Estructura del documento

El presente documento se organiza en tres capítulos principales. El Capítulo I aborda el levantamiento de requisitos y el diseño arquitectónico del sistema. El Capítulo II detalla la construcción e implementación de la plataforma. El Capítulo III presenta las pruebas realizadas y la validación del sistema. Finalmente, se incluyen las conclusiones del proyecto y recomendaciones para futuras mejoras.

Capítulo I

Levantamiento de Requisitos y Diseño del Sistema

1.1 Levantamiento de requisitos

El proceso de levantamiento de requisitos constituyó la fase inicial y fundamental del proyecto, determinando las especificaciones funcionales y no funcionales que el sistema debía cumplir para satisfacer las necesidades del Laboratorio de Tecnologías de la Información y Comunicación (LTIC). Este proceso se llevó a cabo mediante reuniones periódicas con el personal del LTIC, específicamente con el coordinador del departamento, quien fungió como Product Owner del proyecto, proporcionando información detallada sobre los procesos actuales, las deficiencias identificadas y las expectativas respecto al nuevo sistema.

Durante estas sesiones de trabajo, se documentaron los flujos de trabajo existentes para la gestión de préstamos, identificando puntos críticos donde el sistema actual presentaba fallas o ineficiencias. Se analizaron casos de uso reales, escenarios problemáticos frecuentes y necesidades específicas del personal administrativo y de los usuarios finales. Este análisis permitió establecer dos categorías principales de requisitos: funcionales y no funcionales, cada una con especificaciones concretas y medibles que guiarían el desarrollo del sistema.

1.1.1 Requisitos funcionales

Los requisitos funcionales definen las capacidades específicas y comportamientos que el sistema debe exhibir para cumplir con sus objetivos operativos. Estos requisitos describen qué debe hacer el sistema en términos de procesamiento de información, interacción con usuarios y gestión de datos.

RF01 - Gestión de préstamos de equipos

El sistema debe permitir el registro completo de préstamos de equipos tecnológicos, capturando información esencial como la identificación del usuario solicitante, el equipo asignado mediante su código de inventario, la fecha y hora de préstamo, el personal del LTIC responsable de la entrega, y observaciones sobre el estado inicial del equipo. Este módulo debe facilitar tanto el registro de nuevos préstamos como la actualización del estado cuando el equipo sea devuelto.

RF02 - Sistema de historial con trazabilidad completa

El sistema debe implementar un módulo de historial que registre de manera permanente todas las transacciones de préstamo realizadas, manteniendo un registro histórico completo que permita consultar quién solicitó cada equipo, quién lo entregó, cuándo fue prestado y devuelto, y cualquier incidencia reportada durante el período de préstamo. Esta funcionalidad es crítica para la rendición de cuentas y el control administrativo.

RF03 - Búsqueda avanzada y sistema de filtrado

El sistema debe proporcionar mecanismos de búsqueda avanzada que permitan localizar préstamos y equipos mediante múltiples criterios. Los usuarios deben poder filtrar registros por nombre de estudiante, código de equipo, rango de fechas, estado del préstamo (activo, finalizado), tipo de equipo, y responsable de la entrega. Adicionalmente, el sistema debe permitir ordenamiento alfabético ascendente y descendente, así como ordenamiento cronológico de los resultados.

RF04 - Gestión de inventario de equipos

El sistema debe mantener un catálogo actualizado de todos los equipos disponibles para préstamo, registrando información como código de inventario, tipo de equipo, marca, modelo, especificaciones técnicas, estado operativo (disponible, prestado, en mantenimiento, dado de baja) y ubicación física. Este módulo debe actualizar automáticamente la disponibilidad cuando un equipo sea prestado o devuelto.

RF05 - Autenticación y autorización de usuarios

El sistema debe implementar un mecanismo de autenticación seguro que valide la identidad de los usuarios antes de permitir el acceso a las funcionalidades del sistema. Se deben definir diferentes niveles de autorización: personal del LTIC con permisos completos de administración, y estudiantes con permisos limitados de consulta de sus propios préstamos. La autenticación debe realizarse mediante credenciales institucionales.

RF06 - Generación de reportes administrativos

El sistema debe generar reportes estadísticos y administrativos que faciliten la toma de decisiones y el análisis de la operación del LTIC. Estos reportes deben incluir: equipos más solicitados, estudiantes con mayor número de préstamos, duración promedio de préstamos, equipos con mayor incidencia de problemas, y análisis temporal de la demanda de servicios. Los reportes deben poder exportarse en formatos PDF y Excel.

RF07 - Notificaciones automáticas

El sistema debe enviar notificaciones automáticas a los usuarios en eventos relevantes, tales como confirmación de préstamo, recordatorio de devolución próxima, y notificación de devolución completada. Estas notificaciones deben enviarse al correo electrónico institucional del estudiante.

1.1.2 Requisitos no funcionales

Los requisitos no funcionales establecen las características de calidad que el sistema debe poseer, definiendo restricciones y criterios relacionados con el rendimiento, la seguridad, la usabilidad y otros atributos que determinan la calidad general de la solución tecnológica.

RNF01 - Rendimiento del sistema

El sistema debe responder a las solicitudes de los usuarios en un tiempo máximo de 3 segundos para operaciones de consulta y 5 segundos para operaciones de registro o actualización de datos. La interfaz debe cargar completamente en menos de 2 segundos en conexiones de red estándar. El sistema debe ser capaz de manejar al menos 50 usuarios simultáneos sin degradación significativa del rendimiento.

RNF02 - Seguridad de la información

El sistema debe implementar medidas de seguridad robustas para proteger la información sensible. Todas las comunicaciones entre el cliente y el servidor deben realizarse mediante protocolo HTTPS. Las contraseñas deben almacenarse mediante algoritmos de hash seguros (bcrypt). El sistema debe prevenir ataques comunes como inyección SQL, cross-site scripting (XSS) y falsificación de peticiones entre sitios (CSRF). Debe implementarse un sistema de sesiones seguro con tokens de autenticación y tiempos de expiración configurables.

RNF03 - Usabilidad e interfaz de usuario

El sistema debe presentar una interfaz intuitiva que permita a usuarios sin experiencia técnica avanzada realizar las operaciones básicas sin necesidad de capacitación extensa. La navegación debe ser clara y consistente en todas las secciones del sistema. Los formularios deben incluir validaciones en tiempo real que informen al usuario sobre errores antes del envío de información. Los mensajes de error deben ser claros y orientar al usuario sobre cómo corregir el problema.

RNF04 - Mantenibilidad y calidad del código

El código fuente debe estructurarse siguiendo las mejores prácticas del framework Laravel y los principios SOLID de diseño de software. Debe mantenerse una separación clara entre la lógica de negocio, la presentación y el acceso a datos mediante el patrón MVC. El código debe incluir comentarios en secciones complejas y documentación técnica que facilite futuras modificaciones o expansiones del sistema. Se debe mantener un control de versiones mediante Git con commits descriptivos y organizados.

RNF05 - Escalabilidad

El sistema debe diseñarse con una arquitectura que permita su crecimiento futuro sin requerir reescrituras significativas. La base de datos debe estructurarse de manera normalizada para facilitar la adición de nuevas tablas o relaciones. El código debe ser modular, permitiendo agregar nuevas funcionalidades mediante la creación de nuevos módulos sin afectar los existentes.

RNF06 - Disponibilidad y confiabilidad

El sistema debe mantener una disponibilidad del 99% durante el horario de operación del LTIC (8:00 a 18:00, lunes a viernes). Debe implementarse un sistema de respaldos automáticos de la base de datos que se ejecuten diariamente, conservando copias de los últimos 30 días. El sistema debe incluir manejo de errores que evite la pérdida de datos en caso de fallos y permita una recuperación controlada.

RNF07 - Compatibilidad

El sistema debe funcionar correctamente en los navegadores web más utilizados: Google Chrome (versiones recientes), Mozilla Firefox (versiones recientes), Microsoft Edge y Safari. La interfaz debe ser responsive, adaptándose adecuadamente a diferentes tamaños de pantalla, desde computadoras de escritorio hasta dispositivos móviles, garantizando una experiencia de usuario consistente en todos los dispositivos.

1.2 Diseño del sistema

El diseño del sistema constituye la fase de planificación arquitectónica donde se definen la estructura general, los componentes principales, sus interacciones y las representaciones visuales mediante diagramas que facilitan la comprensión del funcionamiento del sistema. Esta etapa transforma los requisitos funcionales y no funcionales previamente identificados en especificaciones técnicas concretas que guiarán la implementación del software. El diseño propuesto para el sistema de gestión de préstamos del LTIC se fundamenta en principios de arquitectura de software bien establecidos, garantizando escalabilidad, mantenibilidad y separación adecuada de responsabilidades.

1.2.1 Arquitectura general

La arquitectura del sistema se fundamenta en el patrón Modelo-Vista-Controlador (MVC), implementado mediante el framework Laravel. Este patrón organiza la aplicación en tres componentes principales que trabajan de manera coordinada: el Modelo gestiona los datos y la información del sistema, la Vista presenta la interfaz visual al usuario, y el Controlador procesa las acciones y coordina las respuestas entre ambos componentes.

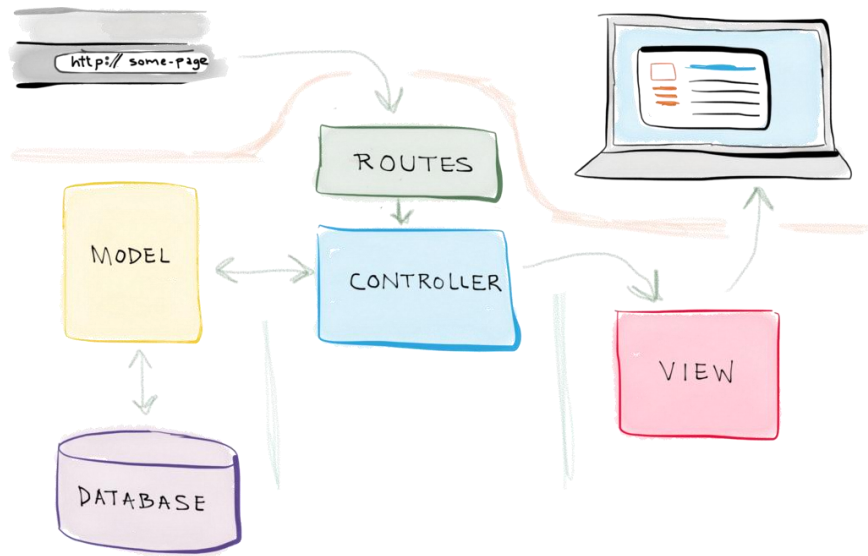


Fig 1.- Un esquema que muestre cómo interactúan el Modelo, la Vista y el Controlador

Organización del sistema

El sistema se estructura en capas diferenciadas. La capa de presentación implementa la interfaz gráfica con la que interactúan los usuarios, incluyendo formularios, pantallas de consulta y paneles de administración. La capa de lógica de aplicación contiene las reglas de negocio, procesa las solicitudes, valida información y gestiona los permisos de acceso. La capa de acceso a datos administra todas las interacciones con la base de datos, almacenando y recuperando información de manera segura y eficiente.

Flujo de operación

Cuando un usuario realiza una acción en el sistema, la interfaz captura la información y la envía para su procesamiento. El sistema valida los datos, verifica que la operación sea permitida, actualiza la información en la base de datos y genera una respuesta que confirma el éxito de la operación o indica los errores que deben corregirse. Este flujo garantiza el procesamiento ordenado y seguro de todas las operaciones del sistema.

1.2.2 Modelado del sistema

El modelado del sistema proporciona representaciones gráficas que facilitan la comprensión de su estructura y funcionamiento. Estos diagramas constituyen herramientas fundamentales de documentación y comunicación entre los participantes del proyecto.

Diagrama de casos de uso

El diagrama de casos de uso identifica las principales interacciones del usuario con la aplicación. Se representan los recursos disponibles para el usuario, las acciones que puede realizar, y el flujo general de operación. En el sistema de gestión de préstamos del LTIC, los estudiantes pueden consultar equipos disponibles, verificar su historial de préstamos, y recibir notificaciones. Los administradores pueden registrar préstamos y devoluciones, gestionar el inventario, consultar historiales completos, generar reportes, y aplicar búsquedas con filtros avanzados.

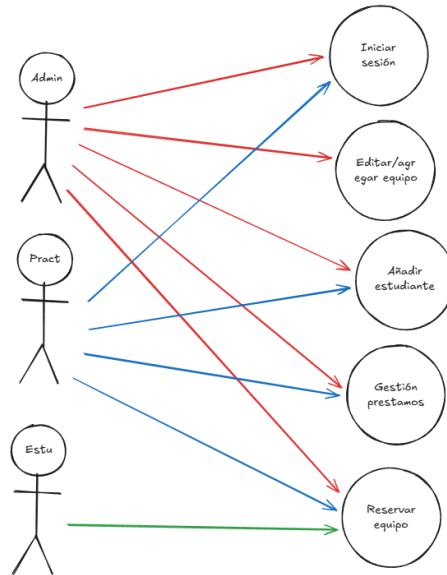


Fig 2.-Diagrama de casos de uso (Casos: iniciar sesión, editar/registrar equipo, añadir estudiantes, reservar equipo, ver historial)

Diagrama de clases

El diagrama de clases describe la estructura del sistema, detallando las entidades principales como Usuario, Equipo, Préstamo, Categoría y Reporte. Cada entidad agrupa la información relacionada y define cómo se conecta con las demás partes del sistema.

Modelo entidad-relación

El modelo entidad-relación define cómo se organiza y almacena la información en la base de datos. Se establecen las tablas principales (usuarios, equipos, préstamos, categorías) y las relaciones entre ellas, garantizando que los datos se mantengan organizados y sin duplicaciones innecesarias.

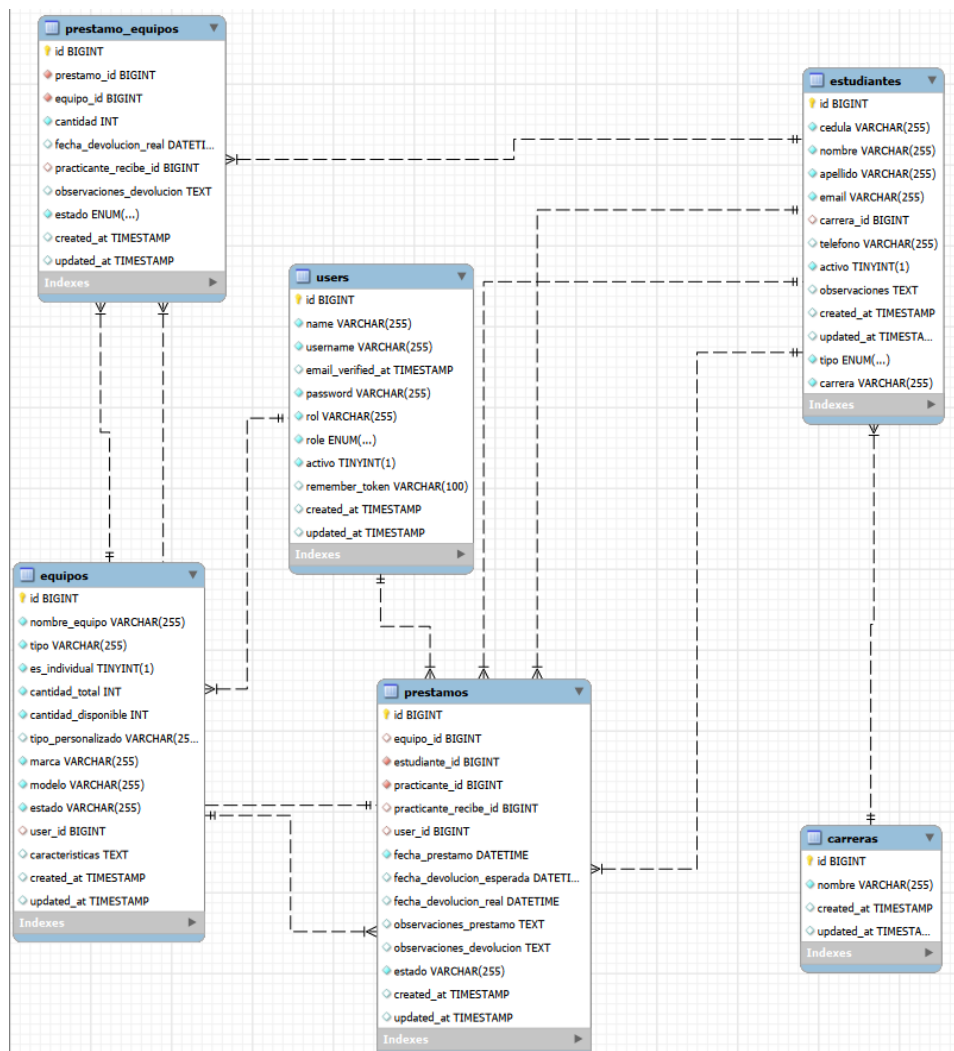


Fig 3.- Modelo entidad-relación del sistema

Prototipo de interfaces

Se diseñaron pantallas preliminares que muestran cómo se visualizarán los formularios de registro de préstamos, las consultas de historial, y los módulos de administración dentro de la aplicación. Estas interfaces permiten al usuario ingresar información de manera sencilla y guiada, facilitando todas las operaciones necesarias. El prototipo incluye pantallas para la confirmación de operaciones, búsqueda de equipos, y visualización de reportes, buscando asegurar una experiencia de usuario intuitiva y eficiente durante todo el proceso.

1.3 Herramientas y tecnologías utilizadas

El desarrollo del sistema de gestión de préstamos del LTIC requirió la selección de herramientas y tecnologías que garantizaran la construcción de una plataforma robusta, segura y escalable. La elección de cada componente se fundamentó en criterios de madurez, documentación disponible, facilidad de mantenimiento y compatibilidad con los requisitos establecidos.

Laravel

Laravel constituye el framework principal para el desarrollo del backend del sistema. Este framework de código abierto para PHP implementa el patrón MVC y proporciona herramientas que facilitan la construcción de aplicaciones web de manera estructurada y eficiente. Su selección se justifica por incluir un sistema de enrutamiento intuitivo, Eloquent ORM para simplificar operaciones con bases de datos, autenticación preconfigurada, y Blade como motor de plantillas para vistas dinámicas. Estas características aceleran el desarrollo y garantizan código mantenible.

PHP

PHP es el lenguaje de programación del lado del servidor sobre el cual se construye Laravel. Este lenguaje interpretado, diseñado específicamente para desarrollo web, permite procesar formularios, interactuar con bases de datos, gestionar sesiones de usuario y ejecutar la lógica de negocio del sistema. Su amplia adopción garantiza disponibilidad de documentación y soporte de la comunidad.

JavaScript

JavaScript proporciona interactividad y dinamismo a la interfaz del usuario, ejecutándose directamente en el navegador. En el sistema se utiliza para validaciones en tiempo real de formularios, actualización dinámica de contenido sin recargar la página,

confirmación de acciones críticas mediante diálogos, y mejora general de la experiencia de usuario mediante carga asíncrona de información.

CSS y Bootstrap

CSS define la presentación visual del sistema, mientras que Bootstrap proporciona un framework de componentes prediseñados y un sistema de rejilla responsive. Esta combinación permite desarrollar interfaces modernas que se adaptan automáticamente a diferentes dispositivos (móviles, tabletas, computadoras) sin necesidad de implementar versiones separadas, garantizando accesibilidad y profesionalismo visual.

MySQL

MySQL es el sistema gestor de bases de datos relacional seleccionado para el almacenamiento de información. Su elección se fundamenta en la compatibilidad nativa con Laravel, estabilidad probada en entornos de producción, rendimiento adecuado para el volumen de operaciones del LTIC, y disponibilidad de herramientas de administración. MySQL garantiza integridad de datos mediante restricciones referenciales y soporta transacciones para mantener consistencia en las operaciones.

Git y GitHub

Git proporciona control de versiones distribuido, permitiendo registrar todos los cambios en el código fuente y facilitando la colaboración. GitHub complementa Git ofreciendo alojamiento en la nube, respaldo automático del código, y funcionalidades de documentación del proyecto. Esta combinación permite mantener un historial completo de modificaciones, trabajar en ramas independientes, y proteger el trabajo contra pérdidas accidentales.

1.4 Metodología

Scrum es un enfoque ágil que se aplica para la administración y creación de proyectos de alta complejidad. En lugar de hacer un esfuerzo por entregar un producto final completo de una vez, Scrum facilita la entrega incremental e iterativa, permitiendo responder rápidamente a los cambios y mejorar continuamente.

Para el desarrollo de este sistema se utilizó Scrum como metodología debido a su capacidad para gestionar proyectos tecnológicos de forma eficiente y colaborativa. Scrum se adapta de forma perfecta a proyectos que requieren flexibilidad y entregas incrementales, permitiendo ajustar la solución tecnológica a las necesidades y cambios que puedan surgir durante el desarrollo.

Por lo tanto, Scrum ofrece las siguientes ventajas:

Organización del trabajo en equipo: Define roles claros (Scrum Master, Product Owner y equipo de desarrollo) que optimizan la colaboración y responsabilidad de cada miembro.

Ciclos iterativos (Sprints): Permiten planificar, desarrollar y entregar partes funcionales del proyecto en períodos cortos, asegurando avances constantes y medibles.

Adaptabilidad: Su enfoque flexible facilita responder a cambios en los requerimientos o prioridades del proyecto.

Mejora continua: Mediante reuniones como las retrospectivas de sprint se analizan los resultados de cada ciclo para identificar áreas de mejora.

1.5 Alcance y limitaciones del sistema

El alcance del proyecto define las funcionalidades implementadas y aquellas excluidas del desarrollo actual, permitiendo gestionar las expectativas del LTIC y enfocar los recursos disponibles en entregar un producto funcional que resuelva las necesidades prioritarias.

Funcionalidades incluidas

El sistema implementa las capacidades fundamentales para la gestión de préstamos de equipos tecnológicos. Se desarrolló un módulo de registro de préstamos que documenta cada entrega, capturando información del estudiante, equipo asignado, fechas y observaciones, actualizando automáticamente la disponibilidad de los dispositivos. El sistema de historial proporciona trazabilidad completa de transacciones, permitiendo consultas diferenciadas según el tipo de usuario: el personal administrativo accede a registros completos, mientras los estudiantes consultan únicamente su historial personal.

Las funcionalidades de búsqueda avanzada y filtrado permiten localizar información mediante múltiples criterios (nombre, código de equipo, fechas, estado) con ordenamiento alfabético y cronológico. El módulo de inventario mantiene actualizado el catálogo de equipos con características técnicas, estado operativo y ubicación. La autenticación mediante credenciales institucionales garantiza acceso controlado según permisos asignados.

Funcionalidades excluidas

Algunas funcionalidades se reservaron para versiones futuras. No se implementó un sistema automatizado de notificaciones por correo electrónico, ni generación de reportes estadísticos con visualizaciones gráficas avanzadas, priorizándose reportes

básicos que satisfacen necesidades inmediatas. El sistema opera únicamente con solicitudes presenciales, excluyendo funcionalidades de reserva anticipada. No se desarrolló aplicación móvil nativa, aunque la interfaz responsive permite acceso funcional desde dispositivos móviles.

Restricciones y dependencias

El sistema presenta restricciones inherentes a su diseño. El funcionamiento requiere conexión permanente a internet, sin capacidades offline, debido al modelo cliente-servidor adoptado. El sistema depende de la infraestructura de red de la PUCE y requiere navegadores modernos compatibles con HTML5, CSS3 y JavaScript, restricción que afecta únicamente a navegadores significativamente desactualizados.

Capítulo II

Construcción del Sistema

2.1 Estructura general del desarrollo

La construcción del sistema de gestión de préstamos del LTIC se organizó siguiendo un enfoque modular e incremental que facilitó el desarrollo ordenado de las funcionalidades. El sistema se dividió en componentes independientes pero interconectados, permitiendo que el equipo trabajara de manera estructurada y validara progresivamente cada parte desarrollada.

El desarrollo se fundamentó en la arquitectura de tres capas previamente diseñada: primero se implementó la base de datos, luego la lógica del sistema en el backend, y finalmente la interfaz visual en el frontend. Este orden garantizó que cada componente pudiera apoyarse en las partes ya funcionales, facilitando la detección temprana de errores.

Módulos principales del sistema

El sistema se organizó en cinco módulos que agrupan funcionalidades relacionadas. El módulo de autenticación gestiona el inicio de sesión de usuarios y la verificación de credenciales institucionales, constituyendo la base para los demás componentes. El módulo de gestión de préstamos implementa las operaciones centrales: registro de préstamos, devoluciones y consultas de equipos activos.

El módulo de inventario administra el catálogo de equipos del LTIC, permitiendo agregar dispositivos, modificar información existente y actualizar estados. El módulo de historial proporciona capacidades de búsqueda avanzada sobre transacciones pasadas, aplicando filtros según el tipo de usuario. El módulo de reportes genera análisis estadísticos sobre la utilización del servicio.

Secuencia de implementación

El desarrollo siguió una secuencia lógica que construyó funcionalidades complejas sobre componentes simples previamente validados. Primero se creó la estructura de la base de datos con sus tablas y relaciones. Luego se desarrollaron los modelos que representan las entidades del sistema y sus reglas de validación. Posteriormente se implementaron los controladores que procesan las solicitudes y ejecutan la lógica de negocio.

Paralelamente se construyó la interfaz de usuario con diseño responsive para diferentes dispositivos. Finalmente se agregó interactividad mediante JavaScript, incluyendo validaciones en tiempo real y actualizaciones dinámicas de contenido.

La integración de componentes se realizó de manera progresiva, validando la correcta comunicación entre capas antes de continuar. Se verificó primero la conexión entre backend y base de datos, luego la comunicación entre vistas y controladores, y finalmente se probaron flujos completos que simulan operaciones reales de usuarios.

2.2 Organización del proyecto

La organización del proyecto establece cómo se estructuran los archivos y directorios del sistema, facilitando el desarrollo ordenado y el mantenimiento futuro del sistema.

Estructura de directorios

Laravel proporciona una estructura predefinida que organiza el código según su función. El directorio app contiene los modelos y controladores que implementan la lógica principal del sistema. El directorio resources almacena las vistas, archivos CSS y JavaScript que conforman la interfaz visual. El directorio database incluye las migraciones que definen la estructura de las tablas. El directorio routes establece las

rutas de la aplicación, determinando qué URLs están disponibles y qué controladores procesan cada solicitud.

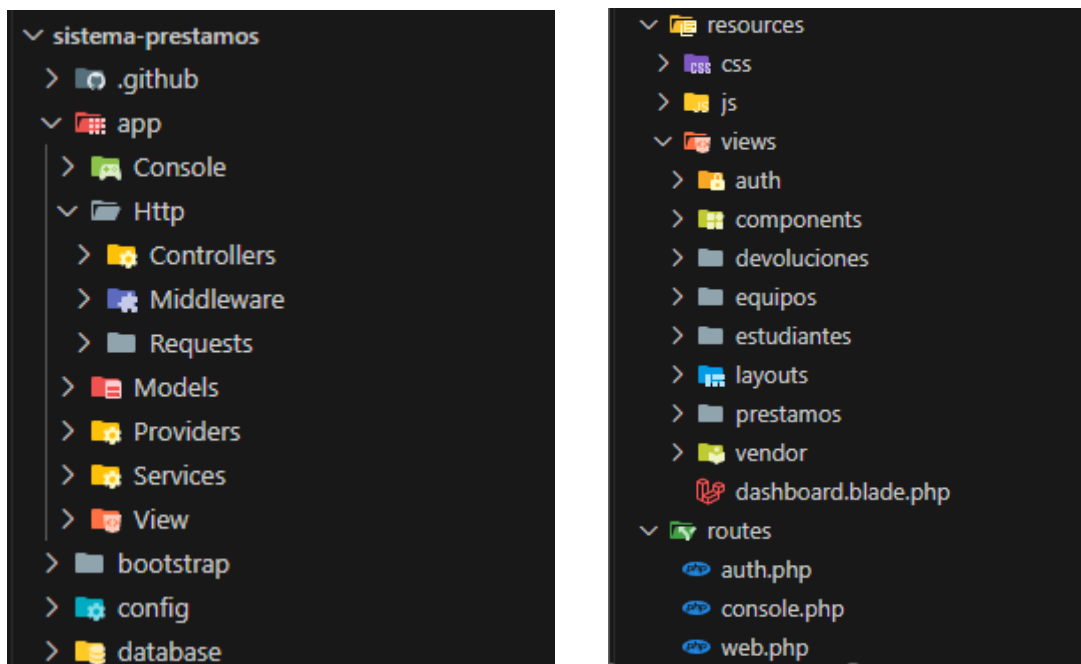


Fig 4.- Orden de las carpetas

Separación de componentes

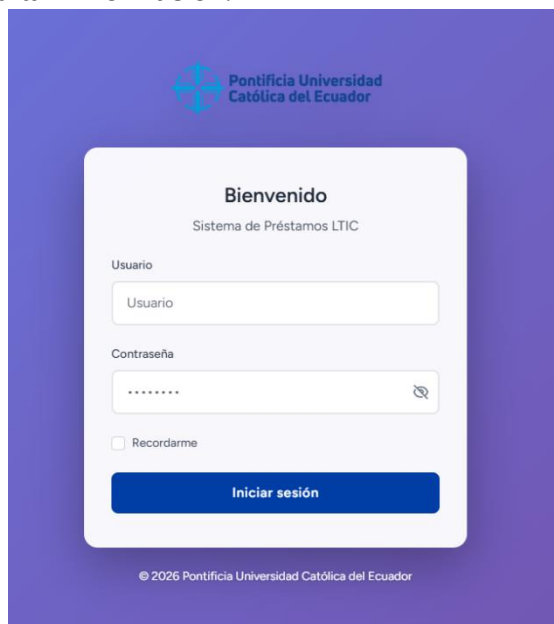
El proyecto mantiene separación clara entre sus componentes principales. El backend reside en el directorio app y contiene la lógica de procesamiento y validación de datos. El frontend se encuentra en resources y constituye la interfaz con la que interactúan los usuarios. La base de datos se gestiona mediante migraciones que permiten crear y modificar su estructura de manera controlada.

Configuración del entorno

El entorno de desarrollo se configuró utilizando Laragon, que proporciona PHP, Apache y MySQL integrados. El archivo .env almacena variables de configuración específicas como credenciales de base de datos. Composer gestiona las dependencias del proyecto, permitiendo instalar Laravel y las bibliotecas necesarias mediante el archivo composer.json.

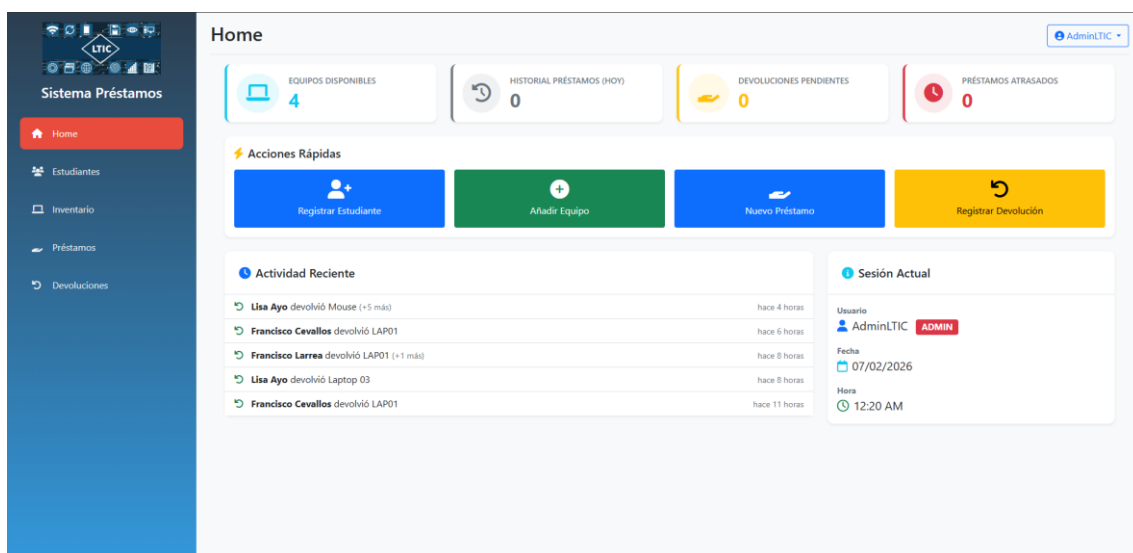
2.3 Desarrollo del frontend

El frontend constituye la parte visual del sistema, es decir, todo aquello que los usuarios ven y con lo que interactúan directamente en sus pantallas. Esta capa del sistema se desarrolló con el objetivo de proporcionar una interfaz clara, intuitiva y accesible que facilite las tareas cotidianas tanto del personal del LTIC como de los estudiantes que consultan información.



Formulario de login del Sistema de Préstamos LTIC. El formulario está centrado en una pantalla de fondo morada. Encabezado: Pontificia Universidad Católica del Ecuador. Título: Bienvenido. Subtítulo: Sistema de Préstamos LTIC. Campos: Usuario (campo de texto), Contraseña (campo de texto con icono de ojo para alternar visibilidad). Opciones: Recordarme (checkbox). Botón: Iniciar sesión. Pie de página: © 2026 Pontificia Universidad Católica del Ecuador.

Ilustración 5 Fig 5.- Formulario del login



Pantalla principal del Sistema de Préstamos LTIC. La interfaz incluye un menú lateral con opciones: Home, Estudiantes, Inventario, Préstamos y Devoluciones. El contenido principal muestra un resumen de estadísticas: EQUIPOS DISPONIBLES (4), HISTORIAL PRÉSTAMOS (HOY) (0), DEVOLUCIONES PENDIENTES (0) y PRÉSTAMOS ATRASADOS (0). Sección de Acciones Rápidas con botones: Registrar Estudiante, Añadir Equipo, Nuevo Préstamo y Registrar Devolución. Sección de Actividad Reciente con una lista de devoluciones: Lisa Ayo devolvió Mouse (+5 más) hace 4 horas, Francisco Cevallos devolvió LAP01 hace 6 horas, Francisco Larrea devolvió LAP01 (+1 más) hace 8 horas, Lisa Ayo devolvió Laptop 03 hace 8 horas y Francisco Cevallos devolvió LAP01 hace 11 horas. Sección de Sesión Actual que muestra el usuario Admin.LTIC ADMIN, la fecha 07/02/2026 y la hora 12:20 AM.

Ilustración 6 Fig 6.- Pantalla principal del sistema

Nuevo Préstamo

Registrar Salida de Equipo

Datos del Préstamo

Estudiante Solicitante *
Test test
Cédula: 1785858522 | Sin camera

Equipos a Entregar *
LAP01
HP - Modelo 800H

Detalles y Responsables

Atendido por (Practicante) *
Seleccione quién atiende...

Fecha Esperada de Devolución *
07/02/2026

Estado de entrega / Observaciones
Ej: Se entrega con cargador original...

Cancelar Confirmar Préstamo

Ilustración 7 Fig 7.- Formulario nuevo préstamo

Gestión de Préstamos

Historial de Préstamos

Buscar por estudiante, código o equipo... Buscar

Todos los estados

Desde dd/mm/aaaa Hasta dd/mm/aaaa

Equipo(s)	Estudiante Solicitante	Practicante Receptor	Fecha y Atención	Fecha Devolución	Estado	Acciones
Todos devueltos	Lisa Ayo Contabilidad	Maya Guachamin (última devolución)	06/02/2026 06:19 PM Entregó: Pablo Cevallos	06/02/2026 07:37 PM (última)	✓ Completado (6/6)	
Todos devueltos	Francisco Cevallos	Maya Guachamin	06/02/2026 05:43 PM Entregó: Pablo Cevallos	06/02/2026 06:17 PM	✓ Completado (1/1)	
Todos devueltos	Francisco Larrea Desarrollo de software	Francisco Larrea (última devolución)	06/02/2026 12:22 PM Entregó: Pablo Cevallos	06/02/2026 03:40 PM (última)	✓ Completado (0/0)	
Todos devueltos	Lisa Ayo Contabilidad	Maya Guachamin	06/02/2026 12:14 PM Entregó: Pablo Cevallos	06/02/2026 03:40 PM	✓ Completado (1/1)	
Todos devueltos	Francisco Cevallos	Francisco Larrea	06/02/2026 04:27 AM Entregó: Pablo Cevallos	06/02/2026 12:21 PM	✓ Completado (1/1)	
Todos devueltos	Francisco Cevallos	Francisco Larrea	06/02/2026 03:52 AM Entregó: Pablo Cevallos	06/02/2026 04:00 AM	✓ Completado (1/1)	
Todos devueltos	Francisco Cevallos	Pablo Cevallos	06/02/2026 03:46 AM Entregó: Pablo Cevallos	06/02/2026 03:47 AM	✓ Completado (1/1)	

Ilustración 8 Fig 8.- Historial de préstamos, con filtros específicos en la parte superior

Implementación de la interfaz de usuario

La interfaz se construyó utilizando HTML para estructurar el contenido de las páginas, CSS para definir la apariencia visual mediante colores, tipografías y distribución de elementos, y JavaScript para agregar interactividad que mejora la experiencia del usuario. La combinación de estas tecnologías permitió crear páginas web dinámicas que responden a las acciones del usuario de manera inmediata.

Se empleó Bootstrap como herramienta de diseño, que proporciona componentes visuales prediseñados como botones, formularios y tablas con apariencia profesional y

consistente. Esto aceleró el desarrollo y garantizó que todos los elementos del sistema mantuvieran un estilo visual coherente.

Pantallas principales del sistema

El sistema incluye varias pantallas diseñadas para funciones específicas. La pantalla de inicio de sesión solicita las credenciales institucionales del usuario y valida su identidad antes de permitir el acceso al sistema. Una vez autenticado, el usuario accede al panel principal que presenta las opciones disponibles según su tipo de cuenta.

Para los administradores del LTIC, la pantalla de registro de préstamos presenta un formulario donde se ingresa información del estudiante, se selecciona el equipo a prestar, y se registran observaciones sobre el estado del dispositivo. Esta pantalla incluye listas desplegables que facilitan la selección de opciones y campos de texto para capturar información adicional.

La pantalla de consulta de historial permite visualizar los préstamos registrados en formato de tabla organizada, mostrando información como fecha de préstamo, estudiante, equipo asignado y estado actual. Esta pantalla incluye campos de búsqueda que permiten filtrar la información mostrada según diferentes criterios.

La pantalla de gestión de inventario lista todos los equipos disponibles en el LTIC, mostrando su código, tipo, marca, modelo y estado actual. Desde esta pantalla los administradores pueden agregar nuevos equipos o modificar información de los existentes.

Diseño responsive y accesibilidad

El sistema se diseñó para funcionar adecuadamente en diferentes dispositivos, adaptándose automáticamente al tamaño de pantalla disponible. Esto significa que la

misma aplicación funciona correctamente tanto en computadoras de escritorio con pantallas grandes como en tabletas y teléfonos móviles con pantallas más pequeñas.

Esta adaptabilidad se logró mediante técnicas de diseño responsive que reorganizan automáticamente los elementos visuales según el espacio disponible. En pantallas grandes, la información se presenta en múltiples columnas aprovechando el espacio horizontal. En pantallas pequeñas, los elementos se apilan verticalmente para facilitar la lectura y navegación sin necesidad de desplazamiento horizontal.

Validaciones del lado del cliente

Para mejorar la experiencia del usuario y evitar errores, se implementaron validaciones que verifican la información antes de enviarla al servidor. Estas validaciones informan inmediatamente al usuario si ha olvidado completar algún campo obligatorio, si ha ingresado información en formato incorrecto, o si existen inconsistencias en los datos proporcionados.

Por ejemplo, cuando un administrador registra un préstamo, el sistema verifica que todos los campos requeridos estén completos antes de permitir el envío del formulario. Si falta información, se resaltan los campos incompletos y se muestra un mensaje indicando qué debe corregirse. Esta retroalimentación inmediata reduce errores y agiliza el proceso de registro.

2.4 Desarrollo del backend

El backend constituye el núcleo del sistema que opera en el servidor, ejecutando las operaciones fundamentales que los usuarios no ven directamente. Mientras el frontend presenta la información, el backend la procesa, toma decisiones según las reglas establecidas y gestiona el almacenamiento en la base de datos.

Implementación de controladores y modelos

Los controladores actúan como coordinadores que reciben las solicitudes de los usuarios y determinan qué acciones ejecutar. Los modelos representan las entidades principales del sistema (usuarios, equipos, préstamos), definiendo qué información se almacena y qué reglas debe cumplir. Por ejemplo, el modelo de Equipo establece que cada dispositivo debe tener un código único y que el estado solo puede ser disponible, prestado, en mantenimiento o dado de baja.

Lógica de negocio principal

La lógica de negocio implementa las reglas que gobiernan el funcionamiento del sistema. Al registrar un préstamo, verifica que el equipo esté disponible y que el usuario tenga permisos adecuados. Al registrar una devolución, actualiza el estado del equipo, registra la fecha y hora, y almacena observaciones sobre el estado del dispositivo devuelto.

Gestión de autenticación y sesiones

El sistema verifica la identidad de los usuarios comparando las credenciales ingresadas con los datos almacenados. Una vez autenticado, se crea una sesión que identifica al usuario durante su interacción con la aplicación, verificando en cada operación que tenga los permisos necesarios. Los administradores pueden registrar préstamos, mientras los estudiantes únicamente consultan su historial personal.

Operaciones con la base de datos

El backend gestiona todas las interacciones con la base de datos mediante Eloquent de Laravel. Ejecuta consultas para recuperar información aplicando filtros

especificados, e inserta o actualiza registros coordinadamente. Al registrar un préstamo, inserta el nuevo registro, actualiza el estado del equipo a prestado, y registra la fecha y hora de la transacción, garantizando consistencia en todas las operaciones.

2.5 Integración y conexión entre capas

La integración entre capas permite que el frontend y el backend se comuniquen para ejecutar las funcionalidades del sistema de manera coordinada.

Comunicación frontend-backend

Cuando un usuario realiza una acción en la interfaz, el frontend recopila la información y la envía al backend mediante una solicitud HTTP. El backend procesa esta información, ejecuta las operaciones necesarias en la base de datos, y genera una respuesta que envía de vuelta al frontend. Esta respuesta puede ser una confirmación de éxito, un mensaje de error, o información solicitada. Laravel facilita este proceso mediante un sistema de rutas que asocia direcciones específicas con los controladores encargados de procesarlas.

Manejo de peticiones y respuestas

El sistema utiliza métodos HTTP para distinguir tipos de operaciones: GET para consultas de información y POST para operaciones que modifican datos. Cuando el backend completa el procesamiento, genera una respuesta que el frontend recibe y utiliza para actualizar la interfaz visual, mostrando la información obtenida o presentando mensajes de error cuando corresponda.

2.5.1 Ejemplos de integración entre capas

La consulta del historial ilustra la integración básica: el frontend envía los criterios de búsqueda al backend, que construye y ejecuta la consulta en la base de datos, y retorna los resultados formateados que el frontend presenta en una tabla organizada.

El registro de préstamos ejemplifica una operación más compleja: el frontend valida los datos ingresados y los envía al backend, que verifica disponibilidad del equipo, registra el préstamo, actualiza el estado del equipo, y confirma la operación. El frontend recibe esta confirmación y muestra el mensaje de éxito al usuario.

2.6 Control de versiones y pruebas iniciales

Durante el desarrollo del sistema se implementaron prácticas de control de versiones y validación continua que garantizaron la calidad del código y facilitaron la detección temprana de errores.

Uso de Git para gestión de código

Git permitió registrar todos los cambios realizados en el código fuente del proyecto, manteniendo un historial completo que documenta qué modificaciones se efectuaron, cuándo ocurrieron y quién las realizó. Esta herramienta facilita recuperar versiones anteriores del código si alguna modificación genera problemas, y permite trabajar en diferentes funcionalidades simultáneamente sin que los cambios interfieran entre sí.

El código se almacenó en un repositorio de GitHub, proporcionando respaldo automático en la nube y facilitando la colaboración entre los miembros del equipo. Cada vez que se completaba una funcionalidad o se corregía un error, los cambios se registraban en el repositorio con un mensaje descriptivo que explica qué se modificó.

Pruebas durante el desarrollo

Las pruebas se realizaron de manera continua conforme se desarrollaban las funcionalidades, validando que cada componente operara correctamente antes de integrarlo con el resto del sistema. Estas pruebas incluyeron verificar que los formularios capturaran información correctamente, que las consultas a la base de datos retornaran los resultados esperados, y que las validaciones identificaran apropiadamente datos incorrectos o incompletos.

Al completar cada módulo, se ejecutaron pruebas de funcionalidad básica que simulaban operaciones reales de usuarios, garantizando que el flujo completo desde la interfaz hasta la base de datos funcionara adecuadamente. Esta validación temprana permitió identificar y corregir problemas cuando su resolución era más sencilla, evitando que errores menores se acumularan y complicaran el desarrollo posterior.

Capítulo III

Pruebas y Estabilización

3.1 Pruebas funcionales del sistema

Las pruebas funcionales constituyen el proceso de validación mediante el cual se verifica que cada componente del sistema ejecute correctamente las funcionalidades para las que fue diseñado. Este proceso garantiza que el sistema cumpla con los requisitos establecidos y que las operaciones se ejecuten de manera confiable y consistente.

Validación de módulos principales

Se validó el funcionamiento independiente de cada módulo del sistema antes de verificar su integración. El módulo de autenticación se probó verificando que aceptara credenciales válidas y rechazara intentos de acceso con información incorrecta. El módulo de gestión de préstamos se validó registrando transacciones de prueba y confirmando que la información se almacenara correctamente en la base de datos con todos los campos requeridos.

El módulo de inventario se probó agregando equipos de prueba, modificando su información, y verificando que los cambios de estado se reflejaran apropiadamente. El módulo de historial se validó ejecutando búsquedas con diferentes criterios y confirmando que los filtros retornaran únicamente los registros que cumplían las condiciones especificadas.

Verificación de flujos de trabajo completos

Las pruebas de flujos completos simularon operaciones reales desde el inicio hasta la finalización. Se validó el proceso completo de registro de préstamo: inicio de

sesión del administrador, navegación al formulario de préstamos, selección de estudiante y equipo, ingreso de observaciones, envío del formulario, y confirmación exitosa de la operación. Este flujo se verificó múltiples veces con diferentes combinaciones de datos para garantizar su funcionamiento consistente.

El flujo de consulta de historial se probó desde la perspectiva de diferentes tipos de usuario, validando que los administradores visualizaran información completa mientras que los estudiantes accedieran únicamente a sus propios registros. El flujo de devolución de equipos se validó verificando que actualizara correctamente los estados, registrara las fechas de devolución, y mantuviera la integridad de la información.

Pruebas de casos de uso críticos

Se identificaron escenarios críticos que representan situaciones importantes para la operación del LTIC y se ejecutaron pruebas específicas para cada uno. Se validó que el sistema impidiera registrar préstamos de equipos ya asignados a otros usuarios, que las validaciones de campos obligatorios funcionaran correctamente rechazando formularios incompletos, y que los mensajes de error proporcionaran información clara sobre cómo corregir los problemas identificados.

Estas pruebas funcionales confirmaron que el sistema opera según lo esperado, ejecutando correctamente las funcionalidades definidas en los requisitos y proporcionando respuestas apropiadas tanto en operaciones exitosas como en situaciones de error.

Nuevo Préstamo

Registrar Salida de Equipo

Registrar estudiante

Error al procesar el préstamo:

- El campo estudiante id es obligatorio.
- Debes especificar la cantidad para cada equipo.
- El equipo_id0 seleccionado no es válido.

Datos del Préstamo

Estudiante Solicitante *

Buscar por cédula o nombre...

Equipos a Entregar *

Buscar equipo 1...

+ Añadir equipo/accesorio

Solo se muestran equipos disponibles.

Detalles y Responsables

Atendido por (Practicante) *

Seleccione quién atiende...

Responsable de entregar el equipo.

Fecha Esperada de Devolución *

07/02/2026

Fecha límite para la devolución del equipo.

Estado de entrega / Observaciones

Ej: Se entrega con cargador original...

X Cancelar

Confirmar Préstamo

Fig 9.- Pruebas de control de errores (si no se llena un campo obligatorio)

3.2 Pruebas de seguridad y autenticación

Las pruebas de seguridad verifican que el sistema proteja adecuadamente la información y que solo usuarios autorizados accedan según sus permisos establecidos.

Validación del sistema de inicio de sesión

Se verificó que el sistema aceptara únicamente credenciales válidas y rechazara intentos con información incorrecta. Se probaron escenarios con correos no registrados, contraseñas incorrectas y cuentas deshabilitadas.

Verificación de permisos de acceso

Se confirmó que los estudiantes que son practicantes consultaran únicamente su historial personal, sin acceso a información de otros usuarios ni funcionalidades administrativas. Los administradores del LTIC tienen acceso completo a todas las funcionalidades. Se intentó acceder directamente a páginas administrativas sin permisos apropiados, verificando que el sistema bloqueara estos accesos.

Protección de datos sensibles

Se validó que las contraseñas se almacenaran cifradas en la base de datos, que las sesiones expiraran tras inactividad prolongada, y que la comunicación utilizara

protocolos seguros. Estas validaciones confirmaron que el sistema implementa medidas apropiadas para proteger la información.

3.3 Pruebas de rendimiento

Las pruebas de rendimiento evalúan la velocidad y eficiencia con la que el sistema ejecuta sus operaciones, garantizando que proporcione respuestas rápidas y mantenga su funcionalidad incluso cuando múltiples usuarios lo utilizan simultáneamente.

Tiempo de respuesta del sistema

Se midió el tiempo que el sistema requiere para ejecutar operaciones comunes, verificando que las respuestas se generen en períodos aceptables. Las consultas de historial retornan resultados en menos de dos segundos, incluso cuando se filtran grandes volúmenes de registros. El registro de préstamos se completa en aproximadamente tres segundos, incluyendo todas las validaciones y actualizaciones en la base de datos. La carga de pantallas principales ocurre en menos de un segundo, proporcionando una experiencia fluida al usuario.

Capacidad de manejo de usuarios simultáneos

Se simulon escenarios con múltiples usuarios accediendo al sistema al mismo tiempo para verificar que el rendimiento se mantuviera estable. Se confirmó que el sistema opera adecuadamente con hasta diez usuarios simultáneos sin degradación apreciable en los tiempos de respuesta. Esta capacidad resulta suficiente para el volumen de operaciones del LTIC, donde raramente coinciden más de tres o cuatro usuarios accediendo simultáneamente.

Optimización de consultas a la base de datos

Se revisaron las consultas que el sistema ejecuta en la base de datos, identificando aquellas que consumían más tiempo y aplicando optimizaciones. Se implementaron índices en los campos utilizados frecuentemente para búsquedas, reduciendo significativamente el tiempo de ejecución de consultas complejas. Estas optimizaciones garantizan que el sistema mantenga un rendimiento adecuado conforme crece la cantidad de información almacenada.

3.4 Validación en diferentes navegadores

Las pruebas de compatibilidad verifican que el sistema funcione correctamente en los navegadores web más utilizados y en diferentes dispositivos, garantizando accesibilidad universal independientemente de la plataforma empleada.

Compatibilidad con navegadores principales

Se validó el funcionamiento del sistema en Google Chrome, Mozilla Firefox, Microsoft Edge y Safari. Se verificó que la interfaz se visualizara correctamente, que los formularios capturaran información apropiadamente, y que las funcionalidades operaran sin errores en cada navegador. Se identificaron y corrigieron inconsistencias menores de visualización, garantizando una experiencia uniforme.

Pruebas de diseño responsive

Se verificó que la interfaz se adaptara correctamente a diferentes tamaños de pantalla mediante pruebas en monitores de escritorio, laptops, tabletas y teléfonos móviles. El diseño responsive reorganiza automáticamente los elementos según el espacio disponible. Se validó que los formularios fueran utilizables en pantallas táctiles, que los botones tuvieran tamaño adecuado para selección táctil, y que el texto se visualizara legiblemente sin necesidad de ampliar la pantalla.

3.5 Pruebas de usabilidad con usuarios del LTIC

Las pruebas de usabilidad constituyen el proceso de validación del sistema con usuarios reales, permitiendo identificar aspectos que pueden mejorarse en la interfaz o en los flujos de trabajo. Estas pruebas proporcionan retroalimentación valiosa desde la perspectiva de quienes utilizarán el sistema cotidianamente.

Validación con personal administrativo

Se realizaron sesiones de prueba con el personal del LTIC, solicitándoles ejecutar operaciones habituales como registrar préstamos, consultar historiales y gestionar el inventario. Durante estas sesiones se observó cómo interactuaban con el sistema, identificando aspectos confusos o que requerían pasos innecesarios. El personal administrativo proporcionó sugerencias sobre la organización de la información en pantalla, la ubicación de botones frecuentemente utilizados, y la claridad de los mensajes presentados por el sistema.

Retroalimentación de usuarios finales

Se solicitó a los administradores que utilizaran el sistema para consultar información sobre equipos disponibles y revisar el historial de préstamos. Esta validación confirmó que la interfaz resultaba intuitiva para los usuarios. La navegación del sistema fue sin dificultad, se comprendió la información presentada, y completaron las tareas asignadas exitosamente. Se identificaron oportunidades de mejora en la presentación de ciertas pantallas y en la redacción de algunas instrucciones.

Correcciones y ajustes finales

Basándose en la retroalimentación recibida, se implementaron ajustes que mejoraron la experiencia de usuario. Se reorganizaron algunos elementos de la interfaz para facilitar el acceso a funciones frecuentemente utilizadas. Se clarificaron mensajes de error para proporcionar orientación más específica sobre cómo resolver problemas. Se ajustaron tamaños de fuentes y espaciados para mejorar la legibilidad. Estas correcciones refinaron el sistema, garantizando que cumpliera efectivamente con las necesidades de sus usuarios.

Conclusiones

El sistema de gestión de préstamos desarrollado para el Laboratorio de Tecnologías de la Información y Comunicación de la PUCE cumplió satisfactoriamente con los objetivos establecidos, proporcionando una solución que moderniza los procesos administrativos del departamento.

El sistema resuelve las limitaciones identificadas mediante un módulo de historial con trazabilidad completa, funcionalidades de búsqueda avanzada con múltiples criterios de filtrado, y una interfaz moderna e intuitiva. La arquitectura basada en Laravel proporcionó una base escalable que facilita el mantenimiento futuro y la incorporación de nuevas funcionalidades.

La aplicación de la metodología Scrum permitió gestionar eficientemente el desarrollo, facilitando la comunicación continua con el LTIC y la adaptación a cambios en los requisitos. Las pruebas realizadas confirmaron que el sistema cumple con los requisitos funcionales y no funcionales, operando con tiempos de respuesta adecuados, protegiendo apropiadamente la información, y funcionando correctamente en diferentes navegadores y dispositivos.

Recomendaciones

Se recomienda implementar en versiones futuras un sistema automatizado de notificaciones por correo electrónico que informe a los estudiantes sobre el estado de sus préstamos y fechas de devolución. La incorporación de reportes estadísticos avanzados con visualizaciones gráficas proporcionaría herramientas analíticas para optimizar la asignación de recursos.

Se sugiere desarrollar funcionalidades de reserva anticipada de equipos y establecer integración con sistemas centralizados de la universidad para simplificar el registro de usuarios. Es importante realizar campañas de difusión del sistema entre la comunidad universitaria y establecer un proceso de revisión periódica que evalúe su funcionamiento e identifique oportunidades de mejora continua.

Referencias bibliográficas

- Atlassian. (2024). *Qué es Scrum y cómo empezar*.
<https://www.atlassian.com/es/agile/scrum>
- Laravel. (2024). *Laravel Documentation - The PHP Framework for Web Artisans*.
<https://laravel.com/docs>
- Mozilla Developer Network. (2024). *JavaScript - MDN Web Docs*.
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- Otwell, T. (2023). *Laravel: Up & Running* (3ra ed.). O'Reilly Media.
- Pontificia Universidad Católica del Ecuador. (2024). *Reglamento de Titulación - PUCE TEC*.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. <https://scrumguides.org>
- Stauffer, M. (2019). *Laravel: Up and Running - A Framework for Building Modern PHP Apps* (2da ed.). O'Reilly Media.
- W3C. (2024). *HTML & CSS Standards*.
<https://www.w3.org/standards/webdesign/htmlcss>
- Bootstrap Team. (2024). *Bootstrap Documentation - The most popular HTML, CSS, and JS library*. <https://getbootstrap.com/docs>
- GitHub, Inc. (2024). *GitHub Documentation - Get started with GitHub*.
<https://docs.github.com>
- Chacon, S., & Straub, B. (2022). *Pro Git* (2da ed.). Apress. <https://git-scm.com/book/es/v2>
- Oracle Corporation. (2024). *MySQL Documentation - MySQL 8.0 Reference Manual*.
<https://dev.mysql.com/doc>
- PHP Group. (2024). *PHP Manual - Hypertext Preprocessor*.
<https://www.php.net/manual/es>
- Pressman, R. S., & Maxim, B. R. (2021). *Ingeniería del Software: Un Enfoque Práctico* (9na ed.). McGraw-Hill Education.
- Sommerville, I. (2020). *Ingeniería de Software* (10ma ed.). Pearson Educación.

Anexos:

- Manual de usuario: [Manual de usuario SISTEMA DE PRÉSTAMOS LTIC.pdf](#)
- Repositorio: <https://github.com/Crisck-m/LTIC.git>