



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y
DISEÑO DIGITAL
SOFTWARE (REDISEÑO)**

APLICACIONES WEB [5TO NIVEL] - A - SOFT-R

Sistema web de gestión de laboratorios informáticos

PROYECTO GRUPAL DE GRADO

Administración de base de datos
Verificación y validación del software



QUEVEDO LOS RIOS

Junio 2

Título:

- Documentación del sistema web para controlar los laboratorios informáticos.

Autores:

- Freddy Vladimir Farinango Guandinango
- Urbina Romero Isaías Abraham
- Villamarín Cuenca Iván Andrés
- Vinueza Sánchez Harold Nicolas

Afiliaciones:

- Universidad Técnica Estatal de Quevedo
- Ing. Gleiston Cicerón Guerrero Ulloa

Tabla de contenido

| | |
|---|----|
| 1. Resumen | 4 |
| 2. Abstract | 5 |
| 3. Introducción | 6 |
| 4. Revisión del estado del arte | 8 |
| 5. Sistema Propuesto | 10 |
| Objetivo Principal | 10 |
| Objetivos Específicos | 10 |
| 5.1. Alcance del proyecto: | 11 |
| 5.3. Estructuración del sistema | 12 |
| 5.4. Requisitos del sistema | 12 |
| 5.4.1. Requisitos Funcionales | 13 |
| 5.4.2. Requisitos No funcionales | 13 |
| 6. Metodología de desarrollo | 13 |
| 5.1. Planeación | 13 |
| 5.2. Diseño de alto nivel | 14 |
| 5.3. Revisión del diseño de alto nivel | 15 |
| 5.4. Desarrollo | 15 |
| 5.5. Post mórtem | 17 |
| 5. Resultados | 18 |
| 5.1. Planificación | 18 |
| Tabla 1: Planificación del desarrollo | 19 |
| 5.2. Requisitos | 19 |
| Tabla: Cantidad de requisitos recolectados | 19 |
| 5.3. Roles y responsabilidades en V&V | 22 |
| Tabla 3: Roles y responsabilidades de lo integrantes | 23 |
| 5.4. Prototipos y diseño inicial de interfaces | 23 |
| Tabla 4: Cantidad de prototipos e interfaces | 23 |
| 6. Discusión | 24 |
| 7. Conclusiones | 24 |
| 9. Referencias | 27 |
| 10. Anexos Diagrama entidad relación Modelo conceptual | 29 |

1. Resumen

El proyecto realizado es un sistema web para organizar el uso de los laboratorios informáticos de la universidad. Nace de problemas comunes: horarios confusos, reservas dispersas y poco registro del estado de los equipos. El sistema reúne en un solo lugar funciones para ver disponibilidad, reservar, registrar asistencia, llevar inventario y reportar fallas, con accesos según el rol de cada usuario.

El desarrollo se realizó con una base de datos en PostgreSQL, un backend en Java con Spring Boot y una interfaz web en HTML, CSS y JavaScript. Para ordenar el trabajo se tomó como guía el Proceso Personal del Software (PPS), pasando por etapas de planeación, diseño, construcción y revisión. En esta versión ya se encuentran listas las pantallas principales y varias funciones clave, y de tal manera el sistema quedó implementado con sus pantallas principales, validaciones básicas y flujos de uso completos.

El objetivo es que docentes, estudiantes y personal técnico tengan información clara y a mano. A futuro podrían evaluarse mejoras como mejor soporte para móvil o enlaces con sistemas internos, siempre que existan permisos y recursos.

Palabras clave:

gestión de laboratorios informáticos; sistema web; reservas y horarios; inventario de equipos; verificación y validación (V&V); Spring Boot y PostgreSQL

2. Abstract

This work presents a web system to organize the use of the university's computer labs. It starts from common issues: unclear schedules, scattered reservations, and limited tracking of equipment status. The system brings together in one place the main functions to check availability, make reservations, record attendance, keep an inventory, and report failures, with access according to each user role.

The development uses PostgreSQL for the database, Java with Spring Boot for the backend, and an HTML/CSS/JavaScript web interface. The Personal Software Process (PSP) was used as a guide to structure the work into planning, design, construction, and review. In this version, the main screens and several key features are ready; This, the system was implemented with its main screens, basic validations, and complete usage flows.

The goal is to give teachers, students, and technical staff clear, accessible information. Possible improvements such as better mobile support or links to internal systems can be evaluated later, if permissions and resources are available

Keywords:

computer lab management; web system; scheduling and reservations; equipment inventory; verification and validation (V&V); Spring Boot and Postgre

3. Introducción

En la universidad, los laboratorios informáticos son espacios muy importantes para el aprendizaje práctico de los estudiantes. En estos lugares no solo se hacen clases, también se desarrollan trabajos, proyectos e incluso actividades extracurriculares. Sin embargo, a pesar de su importancia, en muchos casos la gestión de estos laboratorios todavía se realiza de forma manual, lo que trae bastantes complicaciones. Por ejemplo, se pueden presentar confusiones con los horarios, sobreuso de equipos, pérdida de información sobre reservas, o simplemente no hay un seguimiento claro de quién usa qué y cuándo.

El presente proyecto nace precisamente al observar estas situaciones. La idea es crear un sistema web que permita tener un mejor control y organización de los laboratorios. Se tiene la visión de que el sistema ayude tanto a docentes como a estudiantes y al personal encargado, para que puedan consultar horarios disponibles, reservar laboratorios, registrar asistencia, llevar el control del uso de los equipos, reportar daños y gestionar todo de forma más sencilla.

Desde el inicio, nos centramos en identificar las necesidades reales de quienes hacen uso de estos espacios. Se conversó con usuarios, se analizaron situaciones comunes y se fueron anotando las funciones que deberían formar parte del sistema. Luego se trabajaron bocetos simples en papel y computadora para tener una idea visual clara, y también se empezó a diseñar la estructura de la base de datos y los módulos del sistema.

Una de las decisiones importantes fue la metodología de trabajo. En lugar de aplicar una metodología muy estructurada como cascada o Scrum, se optó por seguir un proceso más personalizado, adaptado a nuestras posibilidades como estudiantes. Basándonos en el modelo de Proceso Personal del Software (PPS), se realizó la división del desarrollo en etapas claras, como la planificación, el diseño, la programación y la revisión, permitiendo avanzar sin la acumulación del estrés, corrigiendo sobre la marcha y entendiendo mejor cada parte del proyecto.

En el tiempo presente, ya se ha desarrollado una buena parte del sistema. Las interfaces ya están diseñadas, algunas funciones clave ya están programadas y

seguimos trabajando para completar el resto. Ha sido un proceso de bastante aprendizaje y también de enfrentarse a varios retos, pero poco a poco el sistema va tomando forma.

4. Revisión del estado del arte

La revisión de antecedentes académicos sobre sistemas de manejo de datos e información en laboratorios informáticos muestra una variedad de propuestas dirigidas a resolver problemas comunes en estos espacios. Se analizaron diez artículos científicos actuales que plantean soluciones tecnológicas en el contexto educativo, principalmente en universidades y centros de formación técnica [1] a [8].

En la mayoría de estos trabajos, se destaca el uso de plataformas con interfaz web que permiten registrar información, consultar disponibilidad o gestionar datos sobre actividades realizadas en los laboratorios [1], [2], [3], [4], [5]. Algunos de ellos también abordan el tema de supervisión o automatización de procesos básicos, como la asignación de horarios o el uso compartido de recursos, con el fin de mejorar la organización interna [1], [6], [7].

Algunos trabajos, como el de Jiménez et al. [7], proponen entornos virtuales que permiten el acceso remoto a laboratorios a través de plataformas educativas, favoreciendo el aprendizaje a distancia. Sin embargo, estos enfoques priorizan la enseñanza práctica y no incluyen elementos administrativos como reservas o asistencia. Otros estudios, como el de Huang [8] y Yu et al. [4], se centran más en el apoyo a la docencia y la planificación académica, pero no incorporan funcionalidades relacionadas con el registro de fallos técnicos o el seguimiento de inventario.

Algunas propuestas como las de Lu et al. [1], Li et al. [6] y Zhang y Liu [5] presentan funcionalidades innovadoras como el uso de tecnología RFID, sistemas híbridos (web mas móvil) o integración con herramientas especializadas para experimentación y control, lo cual fortalece la operatividad técnica de los laboratorios.

También se identifican propuestas como la de Nur Ariffin et al. [9], que presentan sistemas de reservas en línea para laboratorios, mejorando la organización interna. Aunque útiles, estos sistemas tienden a cubrir solo aspectos específicos. Lo mismo ocurre en el trabajo de Zambrano et al. [10], que desarrolla un sistema funcional de gestión, pero sin abordar una integración completa con la estructura académica ni con reportes históricos o perfiles diferenciados por usuario. La mayoría de las investigaciones

revisadas adoptan una solución parcial o dirigida a necesidades puntuales, sin abarcar una visión global del entorno universitario.

Pese a sus aportes, todos los sistemas revisados presentan limitaciones importantes en cuanto a una visión integral del entorno universitario. En particular, ninguno contempla simultáneamente aspectos clave como:

- Gestión de inventario de equipos y mobiliario
- Control de reservas y horarios académicos
- Registro de asistencia de docentes y estudiantes
- Generación de reportes técnicos y de uso
- Trazabilidad histórica y mantenimiento de laboratorios
- Roles diferenciados (docente, técnico, administrador, estudiante)

Además, algunos sistemas están diseñados únicamente para entornos científicos o de investigación [2], [6], [5], por lo que no se adaptan fácilmente al manejo académico y administrativo de facultades o carreras. Ninguno de los trabajos contemplados abarca simultáneamente áreas como planificación de horarios, gestión de equipos, control de asistencia o generación de reportes consolidados. Tampoco se observa un diseño con múltiples niveles de acceso según el tipo de usuario, lo cual limita su utilidad en entornos donde participan docentes, estudiantes, técnicos y personal administrativo.

Frente a estas limitaciones, el sistema propuesto en el presente proyecto se plantea como una solución más integral. Su enfoque no solo busca mejorar el uso académico de los laboratorios, sino también organizar de manera efectiva la información sobre reservas, equipos, asistencia y reportes. A través de una interfaz web simple y funcional, los usuarios podrán acceder a las herramientas que necesiten, según su perfil. Además, se permitirá una trazabilidad completa de todas las actividades realizadas en los laboratorios, facilitando la toma de decisiones y la planificación institucional.

Este presente sistema no pretende reemplazar las ideas anteriores, sino complementarlas y adaptarlas a las necesidades reales de un entorno universitario, aprovechando los aprendizajes detectados en los estudios revisados, la propuesta busca resolver vacíos funcionales y estructurales para brindar una herramienta completa, accesible y útil para la gestión diaria de laboratorios informáticos.

5. Sistema Propuesto

Para resolver las necesidades que encontramos en el uso diario de los laboratorios, se ha planteado el desarrollo de una sistema web de gestión laboratorio informático que ayude a organizar y manejar mejor toda la información que se genera en estos espacios. No solo se trata de reservar aulas o registrar equipos, sino de facilitar la manera en que docentes, estudiantes y responsables interactúan con el entorno del laboratorio.

La idea es que todo esté en un solo lugar: desde los horarios hasta el inventario de computadoras o sillas. Así, si un docente necesita reservar un espacio para su clase, puede hacerlo desde la misma plataforma, sin necesidad de procesos largos o depender de terceros. También se podrá registrar cuándo se usó un laboratorio, por quién, con qué materia y si hubo algún problema con los equipos.

Otro punto clave es que el sistema ayudará a que el personal encargado pueda tener un mejor control del estado de los equipos y el mobiliario. Si algo se daña, se podrá reportar fácilmente y hacerle seguimiento. Incluso se podrá tener un historial para saber cuántas veces un equipo falló, cuándo fue reparado o si ya se cambió.

La plataforma también considera los diferentes tipos de usuarios. No es lo mismo lo que necesita un estudiante que un administrador o un docente. Por eso, cada uno tendrá acceso a lo que realmente le corresponde, sin complicar la interfaz o sobrecargarla con funciones innecesarias.

Este sistema fue pensado con base en las debilidades que observamos en otros trabajos y herramientas existentes. En lugar de enfocarnos solo en una parte, como reservas o inventario, se propuso una solución más completa, adaptada a la realidad académica y organizativa de la universidad. De esta forma, buscamos que realmente sea útil y fácil de aplicar en el día a día.

Objetivo Principal

Optimizar la gestión de los laboratorios informáticos mediante el desarrollo de un sistema web que mejore la organización interna y facilite el acceso a la información para una administración más eficiente.

Objetivos Específicos

1. Automatizar el registro y control de los equipos y mobiliario utilizados en los laboratorios.

2. Coordinar la asignación de horarios y reservas de uso para mejorar la organización de los espacios.
3. Registrar la asistencia de las actividades realizadas en los laboratorios para contar con datos históricos.
4. Facilitar la notificación de fallas técnicas para una atención oportuna por parte del personal encargado.
5. Generar reportes que ayuden a visualizar el estado y uso de los recursos tecnológicos y físicos.

5.1. Alcance del proyecto:

El presente proyecto se enfoca en el desarrollo de un sistema web que ayude a organizar y mejorar el uso de los laboratorios de computación dentro de la universidad. Esta herramienta permitirá llevar un mejor control de las actividades que se realizan en esos espacios, como saber cuándo están ocupados, qué equipos están disponibles y reportar si algo falla. Está pensado para ser utilizado por docentes, personal técnico y quienes tienen a cargo el manejo de los laboratorios. Todo el sistema podrá usarse desde cualquier navegador, lo que lo hace accesible desde distintos lugares dentro del campus. Como se trata de un trabajo realizado por estudiantes, el desarrollo estará centrado en crear una versión básica pero funcional, cumpliendo con los objetivos planteados y ajustándose al tiempo y recursos que tenemos disponibles. La idea es dejar listo un sistema que pueda ser probado en condiciones reales, aunque aún no esté listo para aplicarse a toda la universidad.

5.2. Limitaciones

El sistema se usa en navegador de escritorio. En teléfonos y tabletas puede abrirse, pero no está pensado para pantallas pequeñas ni para gestos táctiles. Tampoco se han cubierto todas las combinaciones de navegador, resolución y sistema operativo; por eso podrían aparecer diferencias de visualización o comportamiento según el equipo.

Por ahora no existe conexión con el SGA ni con otros servicios institucionales. Esta integración depende de accesos, políticas y datos que maneja la universidad. Si en algún momento se autorizan estos cruces de información, se podría evaluar cómo enlazar los módulos sin afectar la seguridad ni la privacidad.

El control que ofrece la aplicación es de tipo administrativo. No hay comunicación con hardware del laboratorio, como sensores de ambiente, lectores de huella o reconocimiento facial. Si más adelante se decide incorporar dispositivos físicos de control de acceso o monitoreo, habría que añadir interfaces y pruebas específicas para ese uso.

Tampoco se incluyen funciones basadas en inteligencia artificial, como un chatbot de ayuda, ni chat en tiempo real entre usuarios. Además, algunos módulos complementarios (por ejemplo, analítica más profunda, reportes avanzados o carga masiva de datos) no forman parte del alcance actual. Podrían considerarse en el futuro si la universidad lo ve útil y existen recursos para su implementación y mantenimiento.

5.3.Estructuración del sistema

- **Módulo de Inventario:** Registra y gestiona los equipos informáticos y mobiliario de cada laboratorio, incluyendo su estado y ubicación.
- **Módulo de Horarios y Reservas:** Permite asignar horarios, consultar disponibilidad y realizar reservas por parte de docentes o administradores.
- **Módulo de Asistencia:** Controla el ingreso de estudiantes y docentes al laboratorio, generando un historial por fecha y materia.
- **Módulo de Reportes Técnicos:** Facilita el registro de fallas y mantenimiento de equipos o mobiliario, asignando seguimiento por parte del personal técnico.
- **Módulo de Administración:** Gestiona usuarios, roles y permisos (administrador, técnico, docente, etc.).
- **Módulo de Historial y Reportes:** Permite consultar y exportar informes sobre el uso del laboratorio, asistencia, fallas y mantenimientos.

5.4.Requisitos del sistema

Antes de empezar con el desarrollo del sistema, fue necesario identificar con claridad qué es lo que se espera que haga y cómo debe funcionar. Esto nos permitió organizar mejor las ideas y tener una guía clara durante todo el proceso. A continuación, se explican los dos tipos de requisitos más importantes: funcionales y no funcionales.

5.4.1. Requisitos Funcionales

Los requisitos funcionales tienen que ver con las acciones que el sistema debe permitir realizar. En el presente proyecto, se enfocan en todo lo relacionado con el manejo de los laboratorios, como registrar los equipos y muebles, organizar las reservas y los horarios de uso, y permitir reportar cuando algo se daña. También se incluye la forma en que las carreras acceden a los laboratorios, cómo se llevan los registros de asistencia y cómo se asocian las materias que se imparten y cada una de estas funciones tiene como objetivo facilitar la organización interna sin depender de procesos manuales o confusos.

5.4.2. Requisitos No funcionales

En cambio, los requisitos no funcionales están relacionados con la forma en que el sistema debe comportarse. Por ejemplo, que sea fácil de entender para los usuarios, que funcione en diferentes navegadores sin errores, que responda rápido cuando se usa, y que mantenga la información segura. También se tomó en cuenta que pueda ser mejorado en el futuro sin complicaciones y que se puedan hacer copias de seguridad por si ocurre algún problema permitiendo que el sistema no solo funcione, sino que sea cómodo y seguro para quienes lo usen.

6. Metodología de desarrollo

Para el desarrollo del presente sistema web de gestión de laboratorios informáticos se aplicó el **Proceso Personal del Software (PPS)**, propuesto por Watts Humphrey y descrito por Roger Pressman en su obra Ingeniería de Software [11].

La presente metodología PPS consta de cinco actividades estructurales, que se detallan a continuación y fueron adaptadas progresivamente en el presente proyecto:

5.1. Planeación

En esta primera etapa se definieron los requisitos del sistema a través de entrevistas con usuarios docentes y personal encargados de los laboratorios y un análisis del entorno real de los laboratorios universitarios. Con base en estos

requerimientos se realizó la estimación de las tareas necesarias, así como una planificación general del trabajo en fases.

Además, se establecieron las siguientes acciones clave:

- Estimación inicial del esfuerzo (días/persona) y tiempo por tarea.
- Identificación de posibles dificultades (riesgos técnicos y de alcance).
- Registro de todas las actividades planificadas mediante un cronograma diario semanal.
- División de responsabilidades entre los integrantes del grupo.

Todas estas decisiones quedaron documentadas y organizadas en hojas de planificación y control.

5.2. Diseño de alto nivel

Se definió la arquitectura general del sistema, compuesta por un backend desarrollado en Java con Spring Boot, un frontend con tecnologías web (HTML, CSS y JavaScript) y una base de datos relacional en PostgreSQL.

En esta etapa también se elaboraron:

- **Diagramas de modelo Conceptual, Lógico y Físico** son modelos de diagramas específicamente para la creación de la base de datos [12].
- **Diagrama de clases:** Representa las clases del sistema, sus atributos, métodos y las relaciones entre ellas. Se usa para modelar la estructura del software[13].
- **Diagrama de contexto:** Muestra el sistema como una única entidad y su interacción con actores externos. Define los límites del sistema y sus entradas/salidas principales [13].
- Prototipos visuales de las interfaces mediante bocetos a mano alzada y de manera digital utilizando Paint.
- Un enfoque modular para dividir el sistema en componentes reutilizables: inventario, reservas, asistencia, reportes, etc.

Se registraron las decisiones técnicas clave, y cuando existieron dudas sobre el diseño de ciertos módulos, se elaboraron pequeños prototipos funcionales para evaluar su viabilidad.

5.3. Revisión del diseño de alto nivel

Antes de comenzar la programación, se revisaron todos los diseños para verificar su coherencia, integridad y viabilidad. Este proceso incluyó:

- Revisión por pares entre los integrantes del grupo.
- Validación cruzada de los diagramas de base de datos y estructuras de navegación.
- Análisis anticipado de posibles errores lógicos o conflictos en las relaciones entre entidades.

Se utilizó una lista de verificación (checklist) para asegurarse de que se cumplan criterios de consistencia, simplicidad y cumplimiento de requisitos.

5.4. Desarrollo

Durante el desarrollo del sistema web se emplearon diversas herramientas agrupadas por su función en el proyecto. A continuación, se describen con sus respectivos conceptos y justificaciones de uso:

a) Backend (Servidor y lógica del negocio)

- **Java**

Java es un lenguaje de programación orientado a objetos, ampliamente utilizado en aplicaciones empresariales por su robustez, portabilidad y seguridad. Se utilizó para construir la lógica del sistema, permitiendo desarrollar un backend sólido y mantenible [14].

- **Spring Boot**

Spring Boot es un framework de desarrollo basado en Java que permite crear aplicaciones web y servicios REST de manera rápida, con configuración mínima y estructuras predefinidas. Facilita la integración con bases de datos y proporciona herramientas para pruebas, seguridad y gestión de dependencias [15].

b) Base de datos

- **PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, reconocido por su estabilidad, integridad de datos y extensibilidad. Soporta consultas complejas, claves foráneas, funciones personalizadas y procedimientos almacenados, lo que lo hace ideal para aplicaciones con estructuras de datos robustas [16].

- **pgAdmin**

pgAdmin es una herramienta gráfica oficial para la administración de bases de datos PostgreSQL. Permite visualizar tablas, ejecutar consultas, generar backups y realizar tareas de mantenimiento fácilmente [16].

c) Frontend (Interfaz de usuario)

- **HTML5**

HTML5 (HyperText Markup Language) es el lenguaje de marcado estándar para la creación de páginas web. Define la estructura y contenido de la interfaz del sistema, incluyendo formularios, botones, tablas, entre otros elementos visuales [17].

- **CSS3**

CSS3 (Cascading Style Sheets) se encarga del diseño y estilo visual de las páginas web, permitiendo aplicar colores, tipografías, márgenes y adaptaciones responsivas. Facilita una experiencia más amigable e intuitiva para el usuario final [18].

- **JavaScript**

JavaScript es un lenguaje de programación del lado del cliente, utilizado para dotar de interactividad a las páginas web. Se usó para validar formularios, gestionar eventos y mejorar la usabilidad del sistema[14].

d) Entornos de desarrollo

- **IntelliJ IDEA**

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) especializado en el desarrollo de software con Java. Proporciona herramientas avanzadas para el análisis de código, autocompletado inteligente, integración con frameworks y ejecución de pruebas [19]

- **Visual Studio Code**

Visual Studio Code es un editor de código multiplataforma ligero pero potente, compatible con HTML, CSS, JavaScript y muchos otros lenguajes. Ofrece extensiones, depuración integrada y control de versiones, lo que mejora la productividad del desarrollador [20].

- e) **Prototipado y diseño de interfaces**

Para el diseño preliminar de las interfaces del sistema se utilizó un enfoque tradicional mediante bocetos digitales elaborados en PowerPoint y Paint. Estos prototipos permitieron visualizar la distribución de elementos, formularios y navegación general del sistema antes de su implementación. Este método fue útil para validar ideas rápidamente, hacer correcciones inmediatas y compartirlas entre los miembros del equipo de forma sencilla y económica.

- **GitHub**

<https://github.com/ffarinangog2/APLICACIONES-WEB-5TO-NIVEL---A--SOFT-R>

GitHub es una plataforma basada en la nube que permite alojar repositorios Git. Ofrece herramientas para seguimiento de errores, revisión de código, gestión de ramas y trabajo colaborativo entre múltiples desarrolladores [21].

Durante la programación, cada módulo fue probado individualmente mediante pruebas manuales. Se registraron los errores detectados, el tiempo invertido por tarea y las correcciones aplicadas. Esto permitió tener una trazabilidad sobre la calidad del trabajo y una mejora continua.

5.5. Post mórtem

El proyecto avanzó bien cuando se implementó las necesidades en casos simples: reservar un laboratorio, registrar una asistencia, reportar un daño. Esas historias cortas ayudaron a decidir qué construir primero y qué dejar para después. También fue útil dibujar pantallas rápidas en PowerPoint o Paint; evitó rehacer vistas completas más adelante.

Lo que más costó fue mantener ordenados los datos de ejemplo y los nombres en la base. Al principio hubo cambios de columnas y llaves que rompían consultas. Se resolvió dejando un esquema estable y acordando nombres claros para entidades y campos. Con eso, las pruebas manuales fueron más parejas y apareció menos ruido.

En el código, separar responsabilidades dio resultado: controladores livianos, servicios con la lógica y repositorios para el acceso a datos. Cuando algo fallaba, era más fácil ubicar la causa. Las validaciones en formularios y mensajes simples al usuario redujeron errores tontos en las pruebas.

Si tocara empezar otra vez, fijaríamos antes los datos mínimos obligatorios de cada módulo inventario, reservas, asistencia y una plantilla base de interfaz para no decidir estilos en cada pantalla. Quedan pendientes temas como compatibilidad en más navegadores, uso en móvil y posibles enlaces con sistemas internos; se pueden evaluar más adelante según lo que pida la universidad

5. Resultados

Se ha completado gran parte del sistema, incluyendo las pantallas principales y funciones básicas como reservas, registro de equipos y control de asistencia. El sistema ya funciona bien en pruebas internas y se adapta a lo que se necesita en los laboratorios.

5.1. Planificación

En la Tabla 1 se muestra la planificación general del desarrollo del sistema. Está organizada por fases, y cada una contiene las actividades principales realizadas en ese momento del proyecto.

| Planificación del desarrollo | |
|------------------------------|---|
| Fase | Actividad |
| Fase 1 | Planificación, investigación, recolección de requisitos |
| Fase 2 | Implementación del backend (Spring Boot y PostgreSQL) |

| | |
|---------------|---|
| Fase 3 | Desarrollo del frontend (HTML, CSS, JavaScript) |
| Fase 4 | Integración de módulos y pruebas funcionales |
| Fase 5 | Validación final del sistema y entrega del proyecto |

Tabla 1: Planificación del desarrollo

5.2 Requisitos

La Tabla 2 resume la cantidad total de requisitos que se recopilamos durante la fase de análisis. Se dividen en dos tipos: funcionales, que definen lo que el sistema debe hacer, y no funcionales, que establecen condiciones de calidad como seguridad o usabilidad.

| Resumen de requisitos recolectados | |
|------------------------------------|----------|
| Recolección de Requisitos | Cantidad |
| Funcionales | 17 |
| No funcionales | 9 |

Tabla: Cantidad de requisitos recolectados

RF1. Registro y Control de Equipos Informáticos

El sistema permitirá registrar todos los equipos como computadoras, impresoras u otros dispositivos de cada laboratorio, con sus características principales, estado actual y ubicación para llevar un mejor control.

RF2. Registro de Mobiliario

Se podrá guardar información del mobiliario (como escritorios, sillas o pizarras) asignado a cada laboratorio, incluyendo su cantidad y estado físico.

RF3. Gestión de Laboratorios y Ubicación Física

El sistema organizará los laboratorios según su número, nombre o ubicación dentro del edificio, para facilitar su identificación y asignación.

RF4. Asignación de Laboratorios a Carreras

Se podrá indicar qué carreras tienen acceso a qué laboratorios, permitiendo así una mejor planificación del uso de espacios.

RF5. Registro de Materias por Laboratorio

Se permitirá asociar materias específicas a los laboratorios donde se imparten, ayudando a controlar el uso de los espacios según la planificación académica.

RF6. Gestión de Horarios y Disponibilidad

El sistema permitirá asignar horarios de uso para cada laboratorio, mostrando su disponibilidad y evitando cruces o sobreuso.

RF7. Registro de Docentes Frecuentes

Se llevará un registro de los docentes que más usan los laboratorios, para análisis posteriores o para darles acceso preferente.

RF8. Registro de Ingreso de Estudiantes

El sistema almacenará los datos de los estudiantes que asisten a clases en los laboratorios, incluyendo la fecha, hora y materia, como una especie de historial.

RF9. Solicitud de Laboratorios por Carreras Externas

Carreras que no tengan un laboratorio asignado de forma directa podrán hacer solicitudes de uso, y estas serán gestionadas por el administrador.

RF10. Reporte de Daños y Fallas Técnicas

Se podrán registrar daños o problemas técnicos que ocurran en los laboratorios, para darles seguimiento y solucionarlos pronto.

RF11. Gestión de Mantenimiento y Sustituciones

El sistema llevará el control de las acciones de mantenimiento que se realicen, así como de los equipos o mobiliario que sean reemplazados.

RF12. Cambios o Reubicaciones de Laboratorios

Cuando un laboratorio cambie de ubicación o se reorganice, estos cambios podrán registrarse para mantener actualizado el sistema.

RF13. Gestión de Roles de Usuario

Cada usuario tendrá un rol específico (como docente, estudiante, administrador), y el sistema permitirá asignar y controlar estos permisos.

RF14. Historial de Laboratorios

Se guardará un registro histórico de cada laboratorio, con información sobre su uso, mantenimiento, asignaciones, entre otros.

RF15. Reservas de Laboratorios

El sistema permitirá que los docentes o responsables hagan reservas de laboratorios para clases o actividades específicas.

RF16. Validación y Gestión de Reservas

Cada reserva deberá pasar por un proceso de validación para evitar conflictos de horarios o duplicaciones.

RF17. Exportación de Reportes

Se podrá generar y exportar reportes con información útil del sistema, como uso de laboratorios, mantenimiento, asistencia, etc.

RNF1. Seguridad de Acceso y Roles

Solo los usuarios autorizados podrán acceder al sistema, y cada uno verá solo lo que le corresponde según su rol.

RNF2. Usabilidad e Interfaz Amigable

El sistema tendrá una interfaz sencilla, clara y fácil de usar para que cualquier usuario pueda manejarlo sin complicaciones.

RNF3. Compatibilidad con Navegadores

Podrá usarse en distintos navegadores web como Chrome, Firefox o Edge, sin que existan errores de visualización o funcionamiento.

RNF4. Disponibilidad del Sistema

El sistema estará disponible la mayor parte del tiempo para que los usuarios puedan acceder cuando lo necesiten.

RNF5. Rendimiento

Debe funcionar de forma rápida, sin tardar demasiado al cargar datos o realizar operaciones.

RNF6. Escalabilidad

Permitirá agregar nuevas funciones o manejar más usuarios/equipos en el futuro sin que se vuelva lento o inestable.

RNF7. Mantenibilidad

El sistema será fácil de actualizar o corregir en caso de errores, para que siempre se mantenga funcional.

RNF8. Seguridad de Datos Personales

Se protegerán los datos personales de los usuarios para evitar accesos no autorizados o pérdida de información.

RNF9. Respaldo y Recuperación

Tendrá copias de seguridad que permitirán recuperar la información en caso de fallas o errores inesperados.

5.3. Roles y responsabilidades en V&V

En la Tabla 3 se detallan los roles y responsabilidades asignados a cada integrante del equipo durante el proceso de verificación y validación (V&V). Cada persona cumplió una función específica para asegurar que el sistema funcione correctamente.

| Roles y responsabilidades | | |
|----------------------------------|--|--|
| Integrante | Rol en V&V | Responsabilidades específicas |
| Iván Villamarín | Responsable de Revisión de Requisitos y Validación | - Liderar la validación de requisitos con usuarios reales (docentes, técnicos) |
| | | - Aplicar listas de verificación |
| Freddy Farinango | Encargado de Pruebas de Unidad y Verificación de Código Temprana | - Coordinar la matriz de trazabilidad |
| | | - Diseñar y ejecutar pruebas unitarias |
| | | - Asegurar que cada módulo cumpla estándares funcionales |

| | | |
|----------------|--|---|
| Isaías Urbina | Responsable de Pruebas de Integración y Validación Funcional | - Validar la lógica interna del código |
| | | - Verificar la interacción entre módulos (reservas, horarios) |
| | | - Coordinar escenarios funcionales completos |
| | | - Generar informes de fallos |
| | | - Ejecutar pruebas de aceptación y usabilidad |
| Harold Vinueza | Encargado de Validación del Sistema, Usabilidad y Reporte Final de Calidad | - Coordinar sesiones con usuarios para validar el sistema |
| | | - Redactar el informe final de V&V |

Tabla 3: Roles y responsabilidades de los integrantes

5.4. Prototipos y diseño inicial de interfaces

La Tabla 4 muestra un resumen del trabajo realizado en cuanto a diseño visual. Aquí se indican los prototipos e interfaces creados para representar la estructura y el comportamiento esperado del sistema antes de su desarrollo completo.

| Resumen de prototipos e interfaces | |
|------------------------------------|----------|
| Recolección de Diseños | Cantidad |
| Interfaces | 8 |
| Prototipados | 4 |

Tabla 4: Cantidad de prototipos e interfaces

6. Discusión

Durante el desarrollo del proyecto, hasta esta etapa, se ha logrado avanzar en aspectos clave como la recolección de requisitos, la planificación por fases, el diseño del sistema y la definición de herramientas. Todo esto nos ha permitido tener una visión mucho más clara de lo que se va a construir y cómo lo vamos a hacer. Aunque aún no iniciamos la parte técnica del desarrollo, creemos que se ha sentado una buena base para que el proceso sea más ordenado y llevadero.

Uno de los puntos más positivos hasta ahora ha sido el trabajo en equipo. Cada integrante asumió un rol claro, lo que ha facilitado la distribución de tareas y la comunicación. Además, el hecho de haber utilizado prototipos nos ayudó bastante para imaginar cómo lucirá el sistema y detectar posibles mejoras desde el diseño.

Sin embargo, también reconocemos que todavía queda bastante por hacer. La implementación y las pruebas serán etapas exigentes, y sabemos que pueden surgir dificultades técnicas o retrasos. Por eso, es importante seguir con el mismo nivel de compromiso y organización que hemos mantenido hasta ahora.

En general, consideramos que el proyecto va bien encaminado. Lo más importante ha sido tener claro lo que se quiere lograr y haberlo documentado correctamente. Ahora el reto será convertir esas ideas en una solución funcional y que realmente aporte a la gestión de los laboratorios informáticos.

7. Conclusiones

Durante el desarrollo de este proyecto se pudo notar que en muchas universidades aún falta una forma clara y organizada para llevar el control de los laboratorios. A veces no se sabe quién los usa, en qué horarios o qué equipos están dañados. Por eso, fue importante pensar en una herramienta que ayude a mejorar la organización de estos espacios.

El trabajo en equipo fue clave para avanzar con cada parte del sistema. Aunque hubo algunas dificultades en el camino, como decidir qué funciones incluir o cómo dividir las tareas, se logró avanzar con base en reuniones, pruebas y correcciones. También se valoró el uso de ideas claras y simples para que cualquier persona pueda entender y manejar la plataforma sin problemas.

Al comparar con otros trabajos parecidos, se vio que algunos se enfocan solo en un aspecto, como el acceso remoto o la parte técnica. En cambio, el sistema que aquí se propone busca tener varias funciones en un solo lugar, adaptándose mejor a cómo se usan los laboratorios en la universidad y pensando en lo que realmente necesitan los docentes, técnicos y estudiantes.

El sistema quedó implementado y operativo con sus flujos principales: reservas de laboratorios, registro de asistencia e inventario de equipos. Al centralizar estas tareas en una sola plataforma, disminuyeron las confusiones de horarios y el doble registro, en línea con lo que reportan soluciones web para contextos educativos cuando unifican la gestión en un punto de acceso común [1], [3], [8].

A diferencia de trabajos que cubren solo una parte del problema reservas o inventario por separado, aquí se integraron módulos bajo roles diferenciados (administración, docentes, soporte). Esa integración hizo más directa la operación diaria y evitó mantener planillas o sistemas paralelos, superando el alcance parcial observado en propuestas similares [9], [10].

El enfoque fue administrativo y de aula presencial. No se incorporó automatización con hardware (RFID, biometría o esquemas híbridos web-móvil) como plantean algunos autores [5], [6], ni se trabajó con laboratorios remotos o virtuales orientados a prácticas a distancia [7]. Esta decisión ayudó a concentrar el esfuerzo en el uso real del campus y en asegurar que los datos básicos estén completos y coherentes.

Para organizar el trabajo se aplicó el Proceso Personal del Software (PPS), con etapas claras de planeación, diseño, construcción y revisión. Otros estudios reportan buenos resultados con marcos ágiles como Scrum [4], pero en este proyecto PPS permitió un avance ordenado con entregas verificables y trazabilidad simple de cambios.

La base de datos en PostgreSQL se comportó estable para las operaciones del sistema. Esto coincide con comparativas que muestran buen rendimiento de motores relacionales en aplicaciones web cuando el esquema está bien definido y las consultas están controladas [16]. Durante las pruebas, acordar nombres consistentes y validar

formularios en la interfaz redujo errores de carga y facilitó reproducir los casos de uso completos

8. Recomendaciones

Para aquellos piensen continuar o mejorar este sistema, sería bueno escuchar las sugerencias de los mismos usuarios. Ellos son quienes lo usarán a diario, así que sus opiniones ayudan a saber qué falta o qué se puede hacer mejor. A veces pequeños detalles hacen la diferencia.

También es importante que se mantenga el sistema actualizado, tanto en funciones como en la información registrada. Esto ayuda a evitar errores y que los datos siempre estén correctos. Si se olvida revisar o actualizar, poco a poco el sistema puede dejar de ser útil.

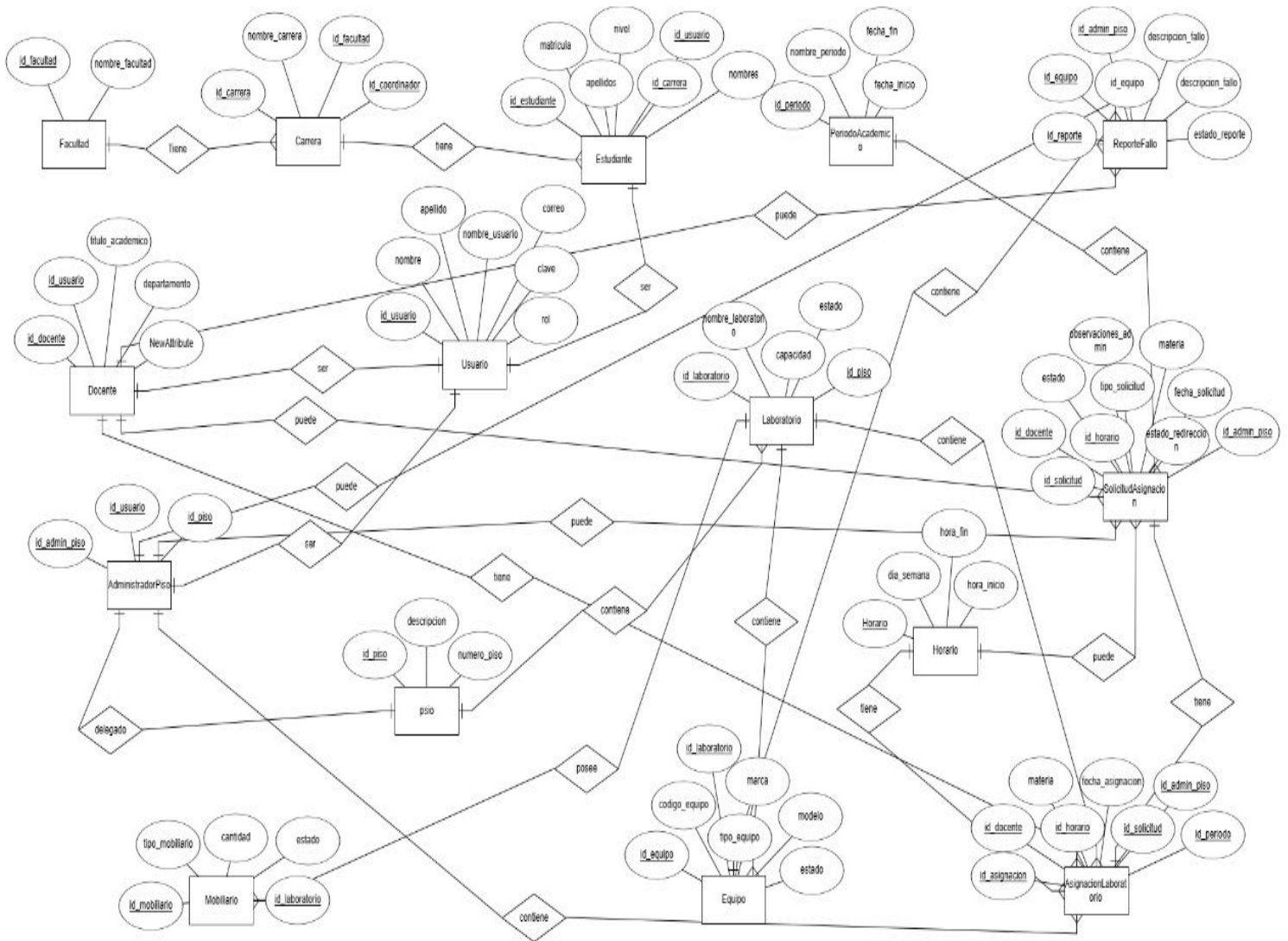
9. Referencias

- [1] Y. Tang and J. Hu, "Design of management system for computer laboratory based on ITIL," *Proceedings of 2009 4th International Conference on Computer Science and Education, ICCSE 2009*, pp. 568–571, 2009, doi: 10.1109/ICCSE.2009.5228366.
- [2] H. Zou, Z. Lin, Y. Luo, and T. Cai, "A Light Real-time Laboratory Information Management System," pp. 231–234, Apr. 2024, doi: 10.1109/IMSE61332.2023.00052.
- [3] B. Li, "Design and implementation of web-based laboratory management system in colleges and universities," *ICCRD2011 - 2011 3rd International Conference on Computer Research and Development*, vol. 2, pp. 468–471, 2011, doi: 10.1109/ICCRD.2011.5764176.
- [4] J. Yu, J. Zhang, Y. Chen, N. Wu, Y. Mei, and W. Sun, "Development of a Lightweight Software Engineering Laboratory Management System Based on Scrum," *Proceedings - 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2022*, pp. 818–821, 2022, doi: 10.1109/AEMCSE55572.2022.00164.
- [5] H. Jin and F. Li, "Design and implementation of laboratory information management system based on hybrid mode," *Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education, ITME 2008*, pp. 631–635, 2008, doi: 10.1109/ITME.2008.4743942.
- [6] A. Lu, Y. Fan, and Z. Q. Guan, "An open laboratory automatic management system based on RFID," *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010*, vol. 3, pp. 57–60, 2010, doi: 10.1109/ICCAE.2010.5451858.
- [7] P. Bistak, "Matlab and Java based virtual and remote laboratories for control engineering," pp. 1439–1444, Jul. 2009, doi: 10.1109/MED.2009.5164749.
- [8] L. Huang, "Research and Implementation of Information Management System of University Computer Laboratory," *Proceedings - 2022 11th International Conference on Information Communication and Applications, ICICA 2022*, pp. 13–17, 2022, doi: 10.1109/ICICA56942.2022.00009.
- [9] N. A. Harron *et al.*, "ComTrack: Implementation of innovative computer lab management tool for academic institutions," *ISCAIE 2017 - 2017 IEEE Symposium on Computer Applications and Industrial Electronics*, pp. 132–135, Oct. 2017, doi: 10.1109/ISCAIE.2017.8074964.
- [10] A. de Investigación, H. I. Javier Rentería-Macias, R. I. Alejandro Macías-Lara, and C. Simón Plata-Cabrera III, "Aplicación Web para Control del Inventario de Laboratorios de Computación," *Dominio de las Ciencias, ISSN-e 2477-8818, Vol. 6, N° 3, 2020 (Ejemplar dedicado a: Julio-Septiembre 2020), págs. 1422-1443*, vol. 6, no. 3, pp. 1422–1443, 2020, doi: 10.23857/dc.v6i3.2443.
- [11] Roger S. Pressman, *Ingeniería De Software Un enfoque práctico*, 7ma Edición. 1997. Accessed: Jul. 13, 2025. [Online]. Available: http://books.google.com/books?hl=en&lr=&id=L5tVdqFU3jcC&oi=fnd&pg=PA45&dq=Ingenieria+de+Software&ots=Dgx7drEqNa&sig=Kkm4_vRo_scAHzqrSZo9Dd_pyOw
- [12] P. G. T. H. Kashmira and S. Sumathipala, "Generating Entity Relationship Diagram from Requirement Specification based on NLP," *2018 3rd International Conference on Information Technology Research, ICITR 2018*, Dec. 2018, doi: 10.1109/ICITR.2018.8736146.

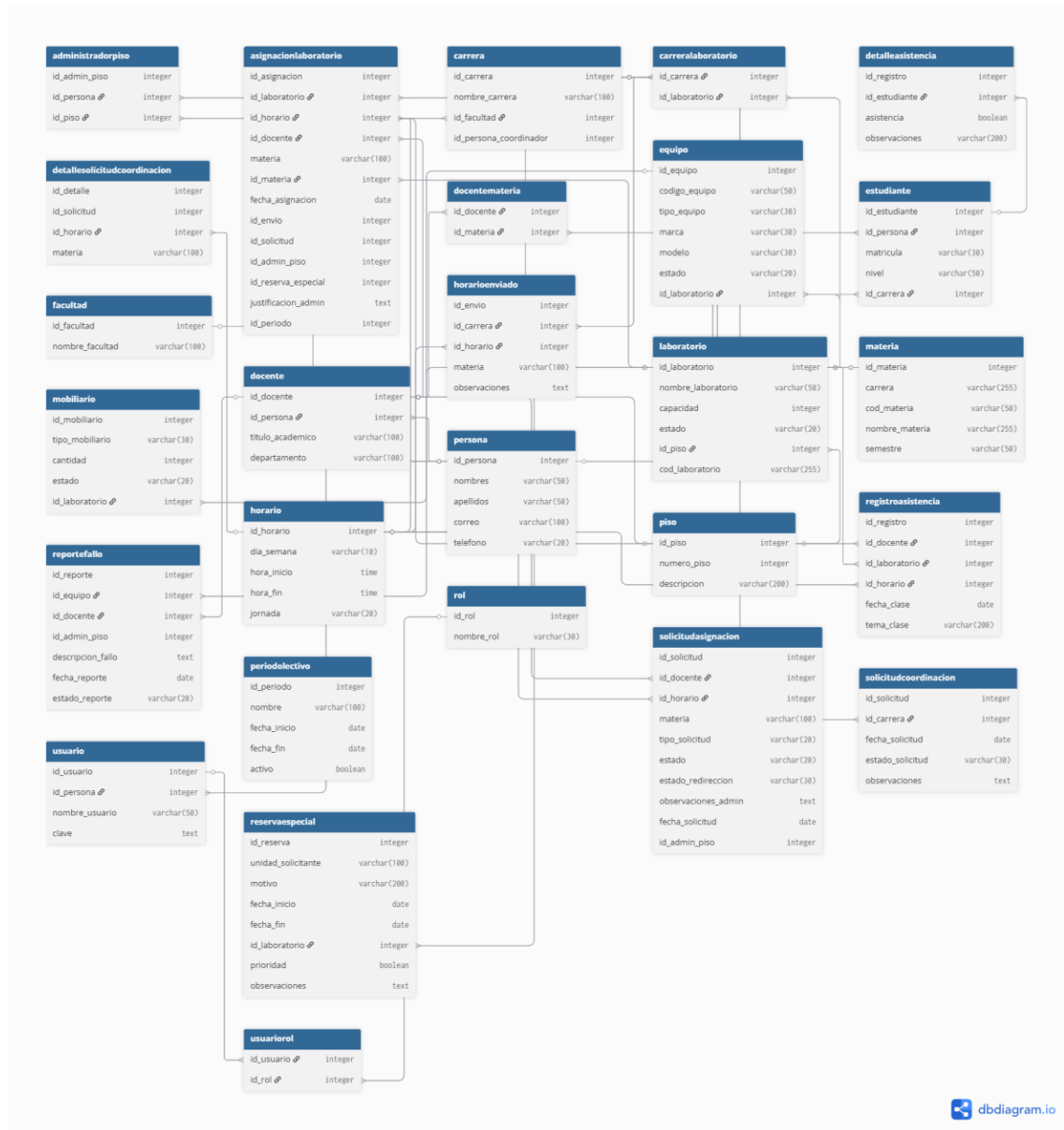
- [13] J. M. Almendros-Jiménez and L. Iribarne, “UML modeling of user and database interaction,” *Computer Journal*, vol. 52, no. 3, pp. 348–367, 2009, doi: 10.1093/COMJNL/BXN028.
- [14] A. Wirfs-Brock and B. Eich, “JavaScript: The first 20 years,” *Proceedings of the ACM on Programming Languages*, vol. 4, no. HOPL, p. 189, Jun. 2020, doi: 10.1145/3386327;WGROUP:STRING:ACM.
- [15] Walls Craig, *Spring Boot en acción | Libros de Manning | IEEE Xplore*. Manning, 2015. Accessed: Jul. 13, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10280114>
- [16] R. Wodyk and M. Skublewska-Paszkowska, “Performance comparison of relational databases SQL Server, MySQL and PostgreSQL using a web application and the Laravel framework,” *Journal of Computer Sciences Institute*, vol. 17, pp. 358–364, Dec. 2020, doi: 10.35784/JCSI.2279.
- [17] A. C. Schwickert, “HTML – Hypertext Markup Language,” *Informatik-Spektrum 1997 20:3*, vol. 20, no. 3, pp. 168–169, Jun. 1997, doi: 10.1007/S002870050065.
- [18] B. D. Blansit, “An introduction to Cascading Style Sheets (CSS),” *Journal of Electronic Resources in Medical Libraries*, vol. 5, no. 4, pp. 395–409, 2008, doi: 10.1080/15424060802453811;WGROUP:STRING:PUBLICATION.
- [19] “Compare IntelliJ IDEA Ultimate vs. IntelliJ IDEA Community - JetBrains IDE.” Accessed: Jul. 13, 2025. [Online]. Available: <https://www.jetbrains.com/products/compare/?product=idea&product=idea-ce#>
- [20] “Visual Studio Code: Edición de código redefinida.” Accessed: Jul. 13, 2025. [Online]. Available: <https://code.visualstudio.com/>
- [21] “Acelere DevOps con GitHub: Mejore el rendimiento de la entrega de software con problemas, proyectos, acciones y seguridad avanzada de GitHub | Libros de Packt Publishing | IEEE Xplore.” Accessed: Jul. 13, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10162298>

10.Anexos

Diagrama entidad relación Modelo conceptual



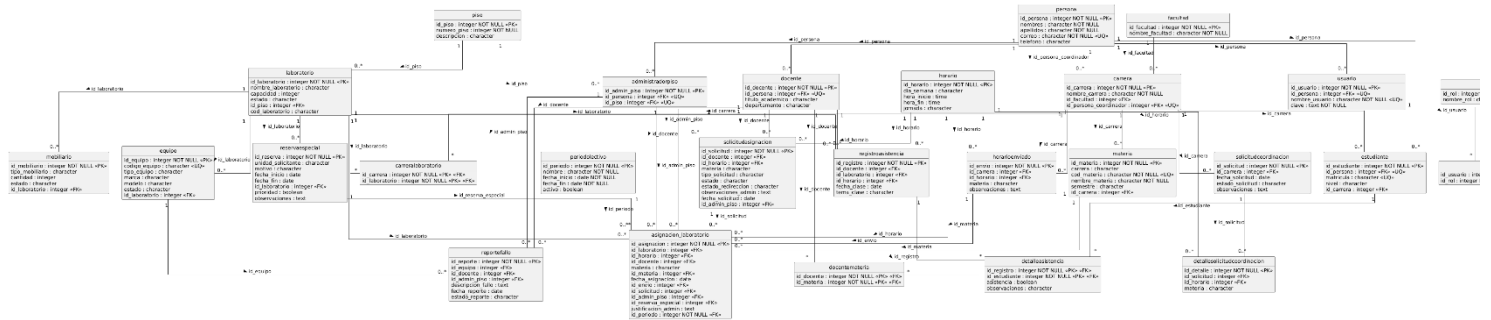
Logico



Modelo Físico



Diagrama de clases



Prototipos

Noticias:

Actualizaciones

Noticias:

Actualizaciones

Inicio de sesión

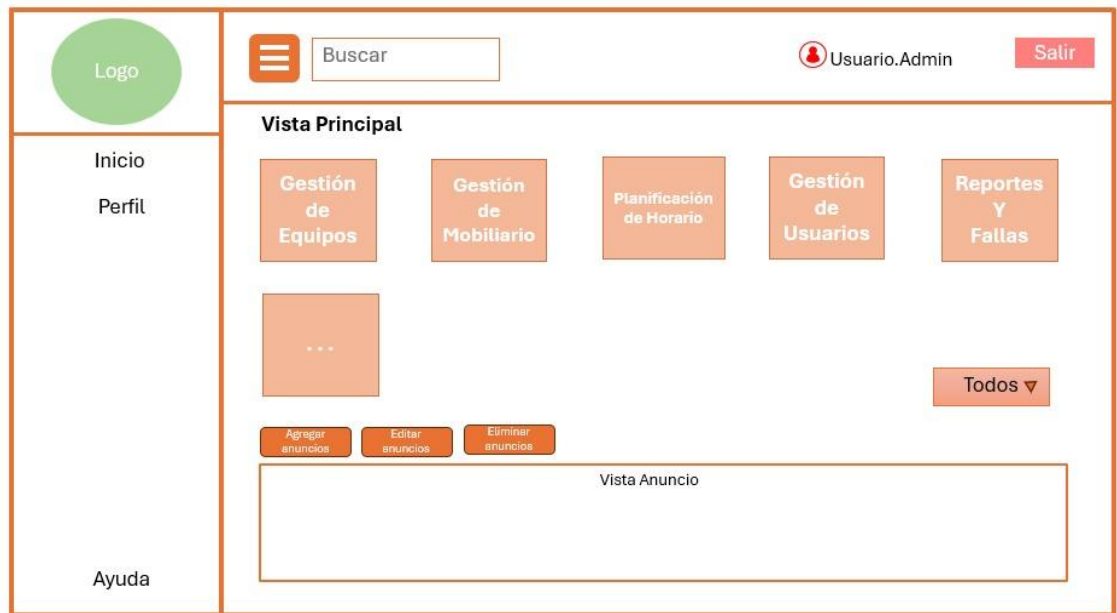
Nombre de Usuario

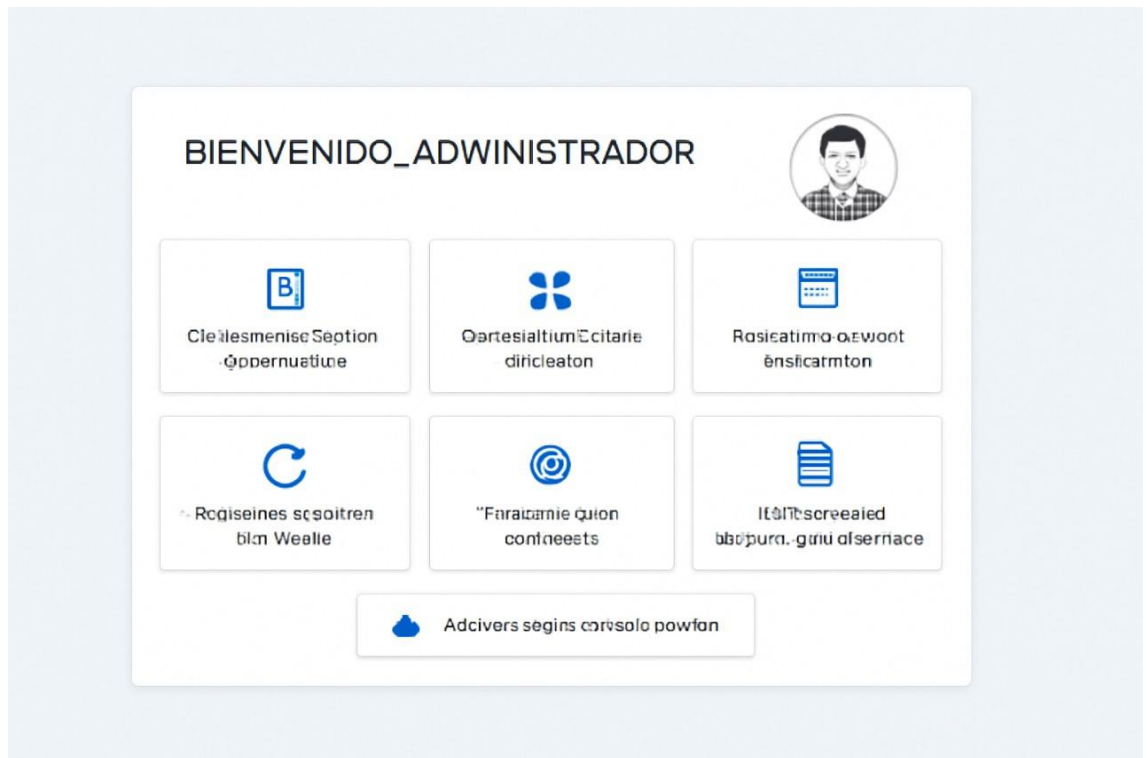
Contraseña

¿Olvidaste tu contraseña? [Clic aqui](#)

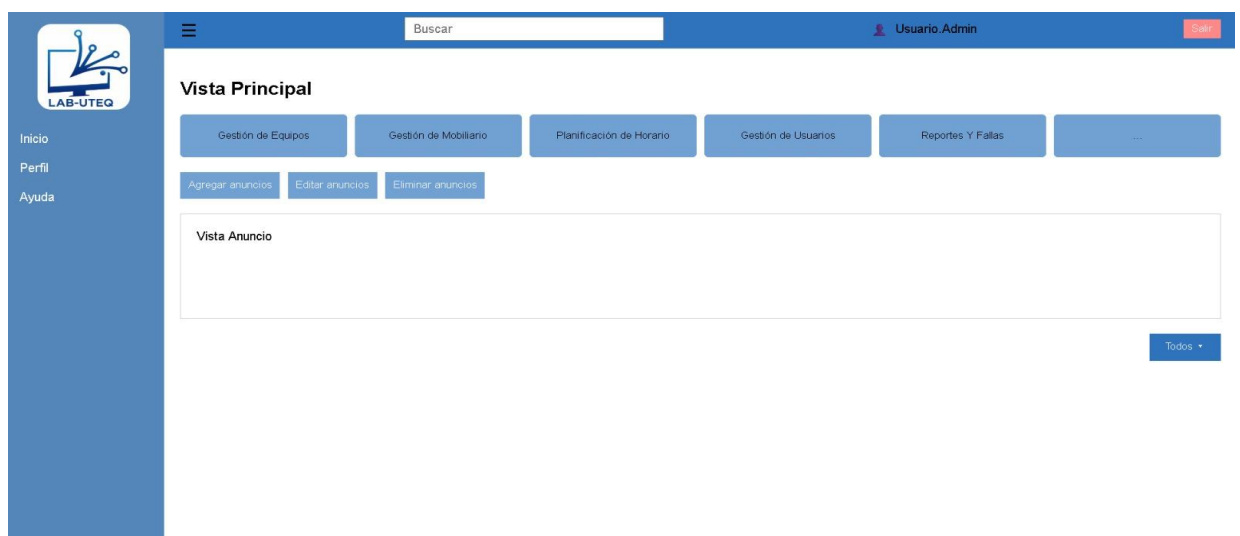
Iniciar

[¿No tienes cuenta? Regístrate](#)





Interfaces



Noticias

Actualización del sistema

Se ha mejorado el rendimiento en la carga de datos del usuario.

Nueva funcionalidad

Ahora puedes cambiar tu foto de perfil desde el panel de usuario.

Inicio de sesión

Nombre de Usuario

Contraseña

[¿Has olvidado los datos de la cuenta? Clic aquí](#)

Iniciar

[¿No tienes cuenta? Regístrate](#)

Entrevista para la recopilación de requisitos

