



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

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

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

### **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

<b>1. Resumen .....</b>	<b>4</b>
<b>2. Introducción.....</b>	<b>5</b>
<b>3. Revisión del arte .....</b>	<b>6</b>
<b>4. Sistema Propuesto .....</b>	<b>7</b>
4.1.    Objetivo Principal .....	7
4.2.    Objetivos Específicos .....	8
4.3.    Alcance del proyecto: .....	8
4.4.    Estructuración del sistema.....	8
4.5.    Requisitos del sistema.....	9
<b>4.5.1.    Requisitos Funcionales .....</b>	<b>9</b>
<b>4.5.2.    Requisitos No funcionales.....</b>	<b>10</b>
<b>5. Metodología de desarrollo .....</b>	<b>11</b>
<b>5.1. Planeación.....</b>	<b>12</b>
<b>5.2. Diseño de alto nivel.....</b>	<b>12</b>
<b>5.3. Revisión del diseño de alto nivel.....</b>	<b>13</b>
<b>5.4. Desarrollo.....</b>	<b>13</b>
<b>5.5. Post mórtem (evaluación final) .....</b>	<b>16</b>
<b>5.6.    Conclusión de la metodología .....</b>	<b>16</b>
<b>6. Resultados .....</b>	<b>16</b>
6.1.    Planificación.....	16
6.2.    Requisitos.....	17
6.3.    Roles y responsabilidades en V&V.....	17
<b>7. discusión.....</b>	<b>19</b>
<b>9. Referencias .....</b>	<b>20</b>
<b>10. Anexos .....</b>	<b>22</b>

## **1. Resumen**

Mas adelante.....

## **2. Introducción**

El presente proyecto consiste en crear un sistema web para el control y la gestión de laboratorios informáticos dentro de la universidad. El presente sistema facilitará a los docentes, estudiantes y personal técnico la reserva de espacios, la visualización de la disponibilidad y el uso adecuado de los equipos disponibles. El sistema es el núcleo del proyecto. Todas las fases y tareas giran en torno a su construcción y funcionamiento. Para su desarrollo, se aplicará la metodología en cascada, lo que significa que el trabajo se organizará en etapas consecutivas. Esto permitirá un mejor control del progreso y una validación más clara en cada paso. Desde la fase de investigación hasta la entrega de los componentes funcionales, cada actividad se enfocará en lograr que el sistema cumpla con los requisitos definidos y sea útil para los usuarios finales.

### 3. Revisión del estado del arte

La revisión de antecedentes académicos sobre sistemas de gestión de laboratorios informáticos y plataformas similares revela un amplio espectro de propuestas orientadas a resolver distintos aspectos del control y uso de estos espacios. Se analizaron ocho artículos científicos relevantes que abordan soluciones tecnológicas enfocadas en la supervisión, automatización y administración de laboratorios en contextos educativos, científicos e ingenieriles [1] a [8].

Una característica común en la mayoría de las propuestas es la implementación de **sistemas centralizados con interfaz web o cliente-servidor**, los cuales permiten controlar el acceso a los laboratorios, monitorear el uso de equipos y gestionar datos asociados a actividades realizadas [1], [2], [3], [4], [5]. Además, varias investigaciones resaltan la importancia de mejorar la **eficiencia en la administración** del laboratorio mediante automatización de tareas rutinarias, como el registro de uso o el acceso a los recursos [1], [6], [7].

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. En esa misma línea, el sistema desarrollado por Jiménez et al. [7] destaca por permitir el acceso remoto a laboratorios de ingeniería de control mediante entornos virtuales construidos en MATLAB y Java, lo cual promueve el aprendizaje práctico en escenarios simulados. No obstante, este enfoque se orienta principalmente a la enseñanza remota y no aborda funcionalidades administrativas como reservas, asistencia o gestión de inventario. Otras investigaciones, como las de Huang [8] y Yu et al. [4], se centran más en **la mejora del aprendizaje práctico** mediante el soporte a la enseñanza y la planificación académica, pero no abordan en profundidad la gestión técnica ni administrativa del laboratorio.

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, en varios casos, los sistemas están orientados únicamente a laboratorios científicos o de investigación [2], [6], [5], sin adaptarse a la estructura académica y administrativa típica de una universidad.

En contraste, el sistema propuesto en este proyecto se diferencia por su enfoque integral, educativo y técnico, al integrar en una sola plataforma web todos los elementos necesarios para la administración eficiente de laboratorios informáticos universitarios. Este sistema incorpora módulos especializados para inventario, reservas, horarios, asistencia, fallas técnicas y reportes, lo cual permite una trazabilidad completa del uso de los recursos y mejora tanto la planificación académica como el mantenimiento operativo. Asimismo, se implementa una gestión de usuarios y roles acorde a la estructura real de las instituciones, permitiendo personalización del acceso y control de funciones según el perfil del usuario. Por tanto, este proyecto no solo recopila y mejora las mejores prácticas detectadas en estudios anteriores, sino que amplía significativamente su alcance, resolviendo vacíos funcionales y estructurales presentes en las soluciones analizadas.

#### **4. Sistema Propuesto**

El sistema propuesto es una aplicación web destinada al control, gestión y seguimiento de los laboratorios informáticos de una universidad. Su diseño responde a la necesidad de centralizar las actividades relacionadas con el uso académico, técnico y administrativo de estos espacios, superando las limitaciones identificadas en soluciones previas. A través de una interfaz intuitiva y segura, el sistema permite registrar los equipos y mobiliario, gestionar horarios, reservar laboratorios, llevar control de asistencia y reportar fallas técnicas, generando así una trazabilidad completa del uso de cada laboratorio.

##### **4.1. Objetivo Principal**

Crear un sistema web para controlar y organizar los laboratorios de computación de la universidad, permitiendo registrar equipos, horarios, reservas, asistencia y reportes de fallas, para así usar mejor los recursos, llevar un buen control y ayudar a que todo funcione de forma ordenada y eficiente.

#### **4.2.Objetivos Específicos**

1. Registrar y controlar los equipos y mobiliario que hay en cada laboratorio, como computadoras, sillas y proyectores.
2. Organizar los horarios y reservas de los laboratorios para evitar cruces y saber cuándo están ocupados o disponibles.
3. Llevar un registro de asistencia de los docentes y estudiantes, para tener un control de quién usa los laboratorios y cuándo.
4. Permitir reportar fallas o problemas técnicos en los equipos o el mobiliario para que puedan ser solucionados rápidamente.
5. Generar reportes claros y útiles sobre el uso de los laboratorios, los daños reportados y las actividades realizadas.

#### **4.3.Alcance del proyecto:**

El presente sistema web estará enfocado en la gestión de laboratorios informáticos dentro de la universidad, permitiendo registrar equipos y mobiliario, gestionar reservas y horarios, llevar control de asistencia y reportar fallas técnicas. Se desarrollará una plataforma accesible desde cualquier navegador, dirigida principalmente a docentes, técnicos y personal administrativo. El presente proyecto cubrirá el diseño, implementación y pruebas del sistema, con módulos funcionales básicos que permitan su uso real en un entorno académico controlado, considerando los recursos y tiempo disponibles como estudiantes.

#### **4.4.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.



## **4.5.Requisitos del sistema**

### **4.5.1. Requisitos Funcionales**

#### **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.

### **4.5.2. Requisitos No funcionales**

#### **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. 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 [9]. El PPS fue seleccionada por ser flexible, adaptable a proyectos individuales o de pequeño equipo, y porque se enfoca en mejorar el proceso de trabajo personal mediante mediciones, revisiones constantes y planeación disciplinada.

A diferencia de metodologías estándar como Cascada o Scrum, el PPS se basa en un enfoque estructurado que pone énfasis en la responsabilidad individual o del equipo pequeño respecto a la calidad, planificación y mejora continua del desarrollo del software.

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

### 5.1. Planeación

En esta primera etapa se definieron los **requisitos funcionales y no funcionales** 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 elaborarán:

- **El modelo entidad-relación (ER), diagramas de modelo Conceptual lógico y físico** para la base de datos, representando tablas como Laboratorio, Reserva, Usuario, Asistencia, Equipos, etc.

Un modelo de datos entidad-relación es un modelo conceptual de alto nivel que describe la información como entidades, atributos, relaciones y restricciones[10] . Los diagramas entidad-relación se utilizan para diseñar la base de datos del software.

- **Diagrama de clases:** Representa las clases del sistema, sus atributos, métodos y las relaciones entre ellas. Se usa para modelar la estructura del software[11].
- **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 [11].
- Prototipos visuales de las interfaces mediante bocetos a mano alzada.
- 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 [12].

- **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 [13].

## **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 [14].

- **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 [14].

## **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 [15].

- **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 [16].

- **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[12].

**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 [17]

- **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 [18].

**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. 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 [19].

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 (evaluación final)

Mas adelante.....

## 5.6. Conclusión de la metodología

El uso del **Proceso Personal del Software (PPS)** permitió aplicar una metodología realista, disciplinada y adecuada al contexto académico, permitiendo gestionar el tiempo, la calidad y los recursos de forma organizada. Aunque fue una metodología personalizada y adaptada, se cumplieron todas las fases esenciales del desarrollo de software, asegurando así la calidad y funcionalidad del sistema entregado.

# 5. Resultados

## 6.1. Planificación

<b>Tabla 3: Planificación del desarrollo</b>	
<b>Fase</b>	<b>Actividad</b>
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

## 6.2. Requisitos

<b>Resumen de requisitos recolectados</b>	
<b>Recolección de Requisitos</b>	<b>Cantidad</b>
Funcionales	17
No funcionales	9

## 6.3. Roles y responsabilidades en V&V

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 Vinuesa	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

#### 6.4. Prototipos y diseño inicial de interfaces

<b>Resumen de prototipos e interfaces</b>	
<b>Recolección de Requisitos</b>	<b>Cantidad</b>

Interfaces	4
Prototipados	2

## 7. 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.

## 8. Conclusiones y recomendaciones

Mas adelante.....

## 9. Referencias

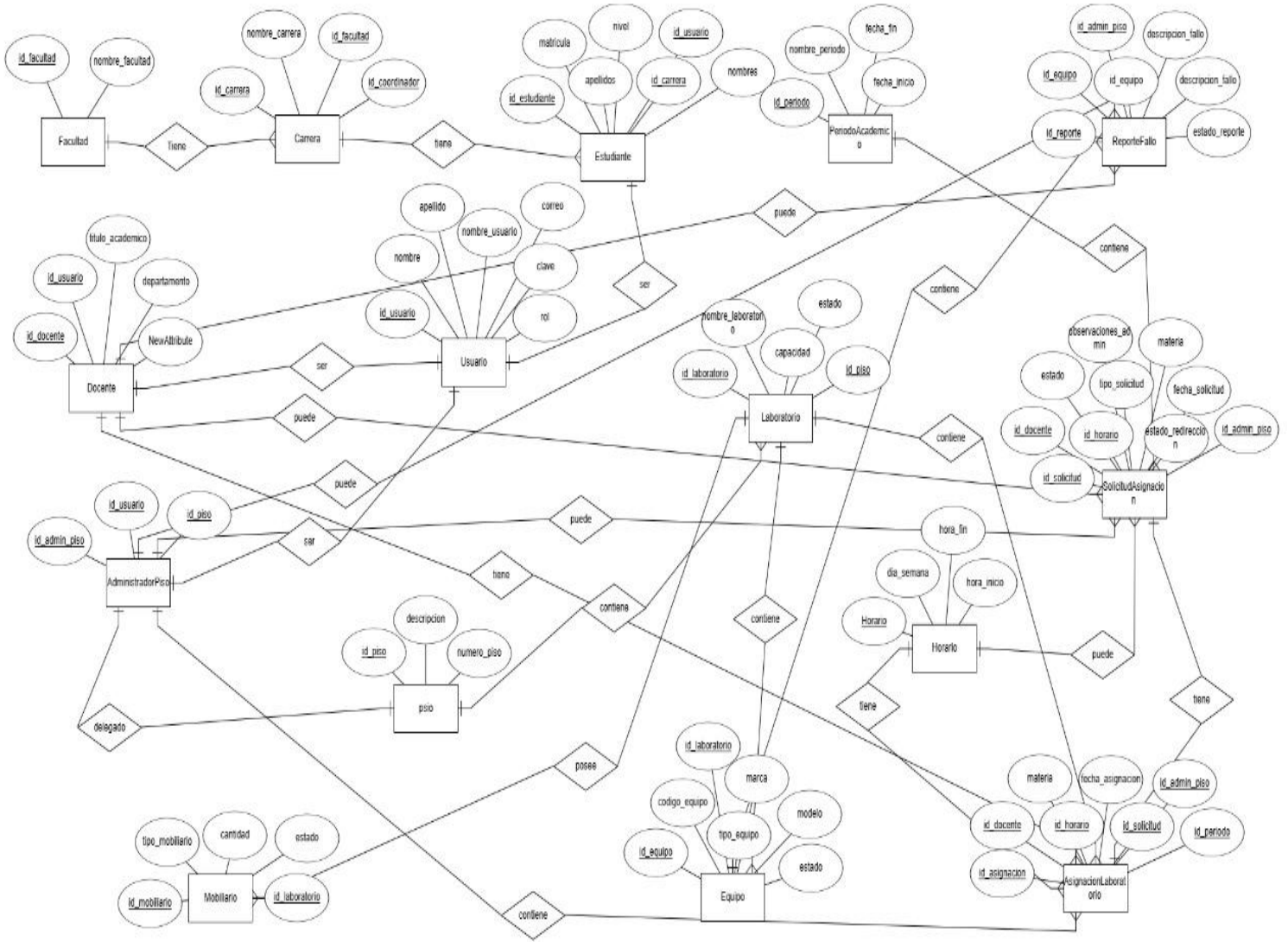
- [1] Y. Tang and J. Hu, "Design of management system for computer laboratory based on ITIL," *Proceedings of 2009 4th International Conference on Computer Science and Education, ICCSE 2009*, pp. 568–571, 2009, doi: 10.1109/ICCSE.2009.5228366.
- [2] H. Zou, Z. Lin, Y. Luo, and T. Cai, "A Light Real-time 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] Roger S. Pressman, *Ingenieria 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](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)
- [10] 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.

- [11] 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.
- [12] 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.
- [13] 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>
- [14] 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.
- [15] 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.
- [16] 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.
- [17] “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#>
- [18] “Visual Studio Code: Edición de código redefinida.” Accessed: Jul. 13, 2025. [Online]. Available: <https://code.visualstudio.com/>
- [19] “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



## Modelo Físico



## Prototipos

**Noticias:**  
Actualizaciones

**Noticias:**  
Actualizaciones

### Inicio de sesión

Nombre de Usuario

Contraseña

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

Iniciar

¿No tienes cuenta? [Regístrate](#)

Logo

Inicio

Perfil

Ayuda

Buscar

Usuario.Admin

Salir

#### Vista Principal

Gestión de Equipos

Gestión de Mobiliario

Planificación de Horario

Gestión de Usuarios

Reportes Y Fallas

...

Todos ▼

Agregar anuncios

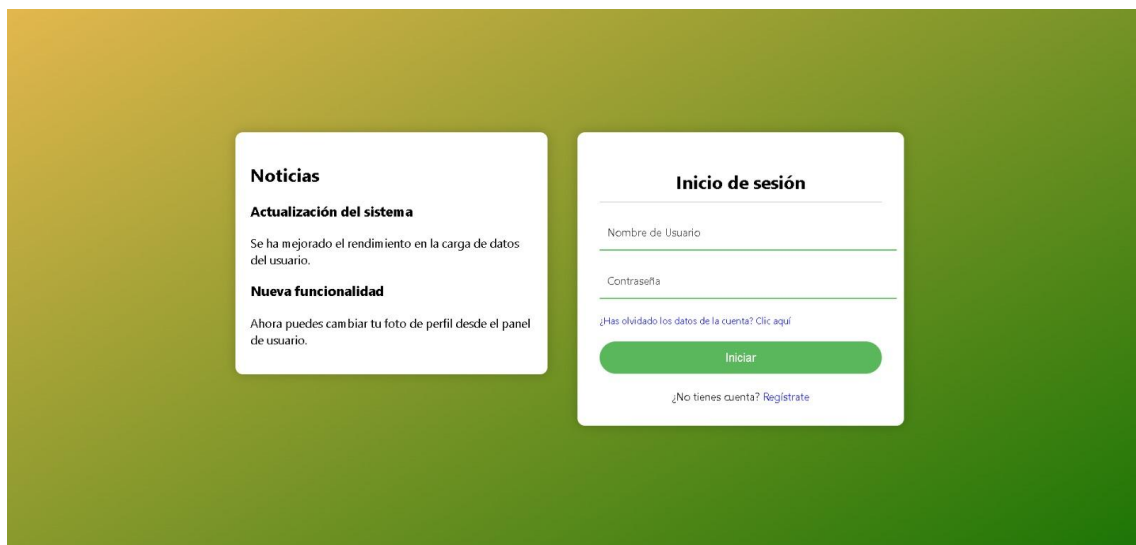
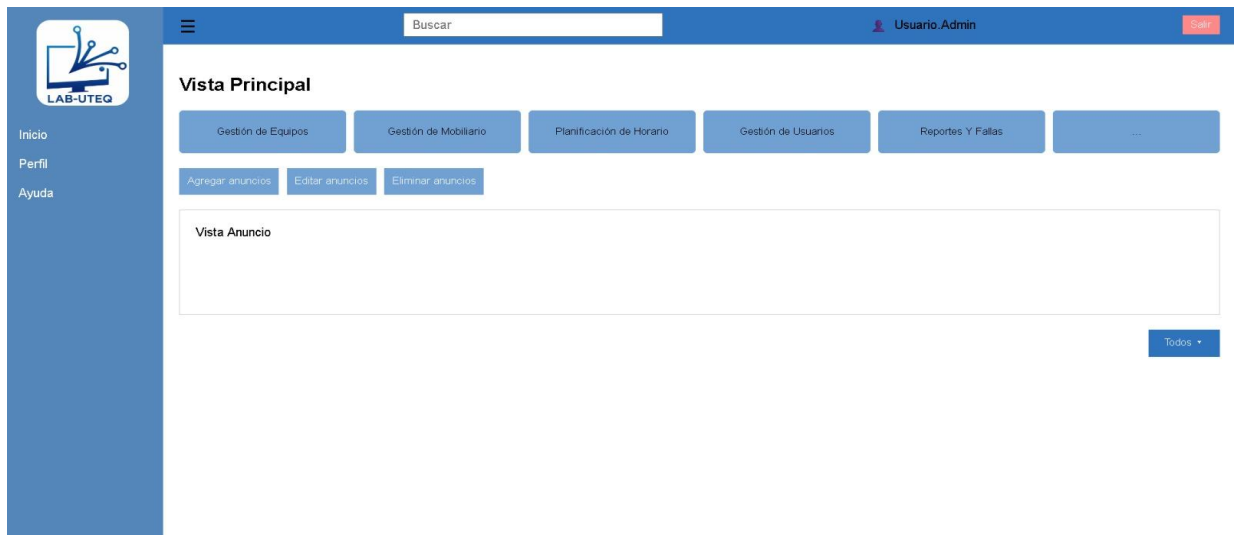
Editar anuncios

Eliminar anuncios

Vista Anuncio

## Interfaces





## Entrevista para la recopilación de requisitos

