

Desarrollo de un Módulo Administrativo para la Disponibilidad de los Dispositivos Electrónicos de los Laboratorios del Instituto Tecnológico Superior de Lerdo

Flores-Luevanos, M.G.¹; Moreno-Núñez, E. M.², Rodríguez-Lozano, K. V.³

Datos de Adscripción:

¹ María Guadalupe Flores Luevanos. Tecnológico Nacional de México/Instituto Tecnológico Superior de Lerdo. maria.fl@itslerdo.edu.mx. <https://orcid.org/0000-0003-2613-1431>

² ✉ Eida Moreno Núñez. Tecnológico Nacional de México/Instituto Tecnológico Superior de Lerdo. elda.mn@itslerdo.edu.mx. <https://orcid.org/0000-0002-5912-082X>

³ Karla Verónica Rodríguez Lozano. Tecnológico Nacional de México/Instituto Tecnológico Superior de Lerdo. karla.rl@itslerdo.edu.mx. <https://orcid.org/0000-0003-2832-4165>

Resumen - Este proyecto desarrolla un módulo de administración para la disponibilidad de dispositivos electrónicos en aulas y laboratorios del Instituto Tecnológico Superior de Lerdo. Se emplean Django, Vue.js y PostgreSQL para optimizar la gestión de equipos y resolver problemas como la falta de control centralizado, errores en registros manuales y ausencia de reportes confiables. La metodología sigue un enfoque mixto, combinando métodos cuantitativos para evaluar el desempeño del sistema y cualitativos que se implementarán en fases posteriores. En esta etapa inicial, se validan las funcionalidades principales mediante pruebas con datos simulados. El prototipo integra módulos clave: gestión de usuarios, registro de fallas y mantenimientos, consulta en tiempo real del estado de los dispositivos y generación automatizada de reportes. Estas funcionalidades cumplen con normativas institucionales y la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados. La implementación del sistema busca centralizar la administración de dispositivos, mejorando la eficiencia operativa, la precisión en los registros y la toma de decisiones. Aunque las pruebas en un entorno real están previstas para futuras etapas, los resultados preliminares sugieren que el prototipo es escalable y replicable en otros espacios de la institución. En conclusión, este trabajo no solo atiende una necesidad específica del Instituto, sino que también establece las bases para nuevas estrategias en la gestión de recursos educativos.

Palabras Clave – Gestión de dispositivos, Infraestructura tecnológica, Optimización administrativa, Reportes automatizados.

Abstract - This project develops a management module for overseeing the availability of electronic devices in classrooms and laboratories at the Instituto Tecnológico Superior de Lerdo. The system leverages Django, Vue.js, and PostgreSQL to optimize equipment management and address key challenges, such as the lack of centralized control, errors in manual records-keeping, and the absence of reliable reporting. A mixed-methods approach is employed, combining quantitative techniques to assess system performance with qualitative methods to be implemented in

later stages. In this initial phase, core functionalities are validated through tests using simulated data. The prototype includes key modules: user management, failure and maintenance logging, real-time device status monitoring, and automated report generation. These features comply with institutional regulations and the General Law on Personal Data Protection Held by Obligated Subjects. The system is designed to centralize device management, enhancing operational efficiency, improving record accuracy, and supporting data-driven administrative decision-making. While real-world deployment is planned for future stages, preliminary results indicate that the prototype is scalable and replicable across similar institutional contexts. In conclusion, this project not only addresses a specific operational need within the Institute, but also lays the groundwork for broader strategies in educational resources management.

Keywords - Administrative optimization, Automated reports, Device management, Technological infrastructure

I. INTRODUCCIÓN

En el ámbito educativo, la integración de tecnologías en aulas y laboratorios ha cobrado gran relevancia, especialmente en instituciones de educación superior con programas de formación en áreas tecnológicas. Sin embargo, la gestión eficiente de los dispositivos electrónicos sigue representando un desafío para muchas instituciones, debido a la falta de herramientas adecuadas para el monitoreo, mantenimiento y administración de estos recursos (García & López, 2021).

El Instituto Tecnológico Superior de Lerdo enfrenta problemas similares en la administración de los dispositivos electrónicos de sus laboratorios. Actualmente, la gestión de estos equipos se realiza mediante hojas de cálculo en Excel, lo que incrementa el riesgo de errores humanos, dificulta el seguimiento del mantenimiento y limita la confiabilidad de los registros. Esta situación no solo afecta la eficiencia operativa, sino que también impacta negativamente en la experiencia de los estudiantes y docentes al utilizar los laboratorios, lo cual repercute en los procesos de enseñanza y aprendizaje, particularmente en asignaturas que requieren el uso intensivo de equipamiento especializado.

A nivel internacional, existen soluciones comerciales como LabStats o Google Workspace, diseñadas para apoyar la administración de recursos tecnológicos en entornos educativos. No obstante, estas herramientas presentan limitaciones importantes en términos de licenciamiento, altos costos de implementación y poca flexibilidad para adaptarse a contextos institucionales específicos. En consecuencia, surge la necesidad

de desarrollar soluciones propias, basadas en tecnologías de código abierto, que se ajusten a las particularidades y recursos disponibles de instituciones públicas como el Instituto Tecnológico Superior de Lerdo.

En respuesta a esta problemática, diversos estudios han explorado soluciones tecnológicas aplicadas a la gestión de dispositivos electrónicos en instituciones educativas. El informe titulado “Aplicación de Internet de las Cosas (IoT) para escalar el monitoreo y gestión remota de dispositivos en instituciones educativas”, desarrollado por Quintanilla Padilla y Cartagena Lobos (2020), representa un referente en esta línea. En dicho trabajo se diseñó un sistema basado en IoT para supervisar de manera remota y escalable los dispositivos electrónicos en el ITCA-FEPADE de El Salvador. A través de sensores conectados a una red de monitoreo centralizada, el sistema permite visualizar en tiempo real el estado de los recursos tecnológicos y generar reportes automáticos que optimizan la toma de decisiones administrativas. Además, introduce el concepto de escalabilidad vertical y horizontal, lo que permite adaptar el sistema a diferentes tipos de infraestructura tecnológica institucional.

Complementando esta perspectiva, Espinoza Barrios (2023) realiza un estudio correlacional en Lima, Perú, donde demuestra que una adecuada gestión de los recursos tecnológicos se relaciona positivamente con el rendimiento académico de los estudiantes. El estudio evidenció una alta correlación ($Rho = .852$) entre ambos factores, lo que resalta la importancia de implementar sistemas que no solo gestionen los dispositivos, sino que también faciliten su uso pedagógico.

Otros trabajos, como el de Gómez (2023), exploran la aplicación de tecnologías de identificación por radiofrecuencia (RFID) para el control de activos de cómputo, destacando la trazabilidad y seguridad que ofrecen estos sistemas al monitorear dispositivos en tiempo real. Asimismo, la tesis de Baez Bagatella (2014) detalla el desarrollo de un sistema integral para la administración de laboratorios de cómputo en la BUAP, con módulos para inventarios, control de accesos y generación de reportes estadísticos, lo que refuerza la necesidad de contar con soluciones centralizadas y adaptables. Por su parte, Lasso Guaman y Masache Mosquera (2024) presentan la implementación del software ElabFTW como una herramienta eficaz de gestión de equipos de laboratorio y proyectos de investigación, destacando su bajo costo y facilidad de adopción institucional.

Con base en estos antecedentes, se propone el desarrollo de un sistema web para la gestión de dispositivos electrónicos en los laboratorios del Instituto Tecnológico Superior de Lerdo. Este sistema está diseñado con tecnologías modernas como Django para el backend, Vue.js para la interfaz de usuario y PostgreSQL como sistema de gestión de bases de datos. El objetivo es mejorar la precisión en el registro de mantenimientos, automatizar la generación de reportes, facilitar la consulta en tiempo real del estado de los dispositivos y garantizar la trazabilidad de los procesos técnicos.

En el Instituto Tecnológico Superior de Lerdo se cuenta con una diversidad de dispositivos electrónicos que resultan fundamentales para el desempeño de las actividades académicas y administrativas. Cada departamento depende en distinta medida de estos recursos según sus funciones específicas. Sin embargo,

el control actual sobre el estado de los equipos, sus mantenimientos y disponibilidad se realiza de manera rudimentaria, principalmente en hojas de cálculo sin mecanismos de validación ni control de versiones. Esta situación hace que la información sea altamente vulnerable a modificaciones accidentales o intencionadas, lo que limita la precisión de los registros y dificulta la toma de decisiones fundamentadas. Además, no existen alertas o indicadores que permitan anticipar necesidades de mantenimiento, lo que incrementa la probabilidad de fallas inesperadas y reduce la vida útil de los dispositivos.

El desarrollo de un sistema automatizado para la gestión del estado operativo de los dispositivos electrónicos busca atender esta necesidad institucional crítica. La propuesta permitirá abordar problemáticas clave como la ausencia de un registro estandarizado, la incapacidad para generar reportes confiables y la demora en la atención de fallas técnicas. Al centralizar los datos y automatizar procesos como el registro de fallas, la actualización del estado de los equipos y la generación de reportes, el sistema proporcionará a los distintos usuarios —docentes, administrativos y jefes de laboratorio— una herramienta robusta, accesible y adaptada a las necesidades operativas del Instituto. Esta solución no solo contribuirá a la eficiencia institucional, sino que también estará alineada con la misión del Tecnológico de Lerdo de fomentar la calidad, la innovación y la competitividad en sus procesos académicos y administrativos.

El sistema propuesto contará con funcionalidades esenciales: login de usuario con autenticación por contraseña, interfaz para la actualización del estado de cada dispositivo, visualización de un conteo resumen que refleje el número total de dispositivos y aquellos con fallas, carga de evidencia documental para cada mantenimiento y generación de reportes automatizados exportables en formato Excel. Todo ello será implementado utilizando frameworks modernos como Django (para la lógica del backend), Vue.js (para una experiencia de usuario dinámica en el frontend) y PostgreSQL como motor de base de datos relacional. Esta arquitectura permitirá escalar la solución a futuro y la integración con otras plataformas tecnológicas institucionales, garantizando sostenibilidad, seguridad y eficiencia.

II. PARTE TÉCNICA DEL ARTÍCULO

A. Metodología

El presente estudio se enmarca en una investigación aplicada, cuyo objetivo es desarrollar un sistema de gestión para dispositivos electrónicos, con el fin de solucionar una problemática específica identificada en el Instituto Tecnológico Superior de Lerdo. Para garantizar un desarrollo progresivo y eficiente, se ha adoptado una metodología incremental, la cual permite la construcción del sistema en iteraciones sucesivas, facilitando su validación, ajuste y optimización en cada fase del proceso.

Específicamente, el sistema fue desarrollado utilizando:

1. Django para la lógica del backend.
2. Vue.js para la interfaz de usuario.
3. PostgreSQL como sistema de base de datos.

En el capítulo del fundamento teórico se explica a detalle las ventajas de estas herramientas de software.

La metodología incremental ha sido seleccionada debido a las características relevantes, que se mencionan a continuación:

- a) Iteraciones pequeñas: El desarrollo se divide en ciclos cortos y manejables.
- b) Entrega progresiva: Se ofrecen versiones funcionales del software desde etapas tempranas.
- c) Flexibilidad: Permite ajustes continuos según nuevas necesidades o cambios del cliente.
- d) Menor riesgo: Identificar errores pronto reduce fallos graves al final del proyecto.

Esas características ofrecen las siguientes ventajas en el desarrollo del software en cuestión:

- a) Satisfacción del cliente: El cliente ve resultados tempranos y da retroalimentación continua.
- b) Mayor calidad: Las pruebas frecuentes ayudan a encontrar y corregir errores pronto.
- c) Uso eficiente de recursos: El trabajo por etapas mejora la planificación y asignación de esfuerzos.
(Sentrio, 2023)

Las fases de la metodología incremental pueden ser descritas brevemente de la siguiente manera (PhoenixNAP, 2025)

1. Recolección de requisitos: Se definen los objetivos y funciones necesarias del software.
2. Planificación de incrementos: Se dividen tareas en bloques entregables.
3. Diseño de cada incremento: Se organiza cómo debe funcionar cada bloque.
4. Codificación: Se desarrolla el incremento según el diseño.
5. Pruebas: Se verifica que cada parte funcione correctamente.
6. Integración: Se combinan los incrementos para construir el producto total.
7. Entrega: Se entrega la versión actualizada al cliente.

El sistema ha sido implementado como prueba piloto en el Laboratorio de cómputo 5E, del Instituto Tecnológico Superior de Lerdo. Este laboratorio cuenta con una capacidad de 30 equipos de cómputo, es utilizado por al menos 200 estudiantes diariamente, con una ocupación promedio del 80% entre las 10:00 a.m. y las 4:00 p.m. Además, el laboratorio tiene el equipamiento de un proyector y un pintarrón inteligente para facilitar las actividades docentes y estudiantiles.

Enfoque Metodológico

Además de la metodología empleada en el desarrollo del software explicada en el inciso A, este proyecto tiene un enfoque de investigación mixto, pues combina técnicas cuantitativas y cualitativas para la evaluación tanto del desempeño técnico del sistema como de su impacto en la gestión de los dispositivos electrónicos del instituto. Este enfoque permite el empleo de:

- Métodos cuantitativos: Se seleccionan un conjunto de métricas objetivas para evaluar el rendimiento del sistema, considerando indicadores como tiempos de respuesta, exactitud de registros, reducción de errores en el registro de dispositivos y eficiencia en la consulta de datos, para este fin se aplican cuestionarios con escalas de Likert para la cuantificación de los datos.
- Métodos cualitativos: Se analiza la experiencia del

usuario a través de observaciones estructuradas y entrevistas semiestructuradas a los usuarios de la prueba piloto, con el propósito de identificar oportunidades de mejora en la usabilidad e interacción con el sistema.

B. Fases del Proyecto

El desarrollo del sistema se realiza en las siete fases descritas en el inciso A, las tareas específicas realizadas en cada fase son listadas a continuación:

1. **Recolección de requerimientos**
 - Identificación y documentación de las necesidades específicas del laboratorio.
 - Definición de los requerimientos funcionales y no funcionales del sistema.
2. **Planificación de los incrementos**
 - Definición de los entregables y de las tareas específicas involucradas en cada uno.
 - Entregable 1: Gestión de los dispositivos
 - Entregable 2: Gestión de las fallas y del mantenimiento correctivo
 - Entregable 3: Creación de los reportes del catálogo de dispositivos, de las fallas registradas y de las acciones de mantenimiento realizadas.
3. **Diseño por incrementos**
 - Elaboración de modelos básicos para la comprensión del software a desarrollar, con sus respectivos diagramas, basados en el lenguaje unificado de modelado (UML).
 - Modelo de casos de uso, basados en los requerimientos funcionales relacionados con las tareas de ingeniería del software de cada incremento.
 - Modelo estático: definición de la arquitectura de las clases involucradas en el incremento.
 - Modelado de la base de datos en PostgreSQL, se decide utilizar este gestor para garantizar la integridad y eficiencia en el almacenamiento de la información.
 - Modelo de flujos de trabajo: con las tareas correspondientes a cada incremento y sus entregables.
 - Diseño de las interfaces gráficas de usuario (IGU) con un enfoque centrado en la usabilidad, compaginando los modelos del usuario, del diseñador y de los programadores.
4. **Desarrollo de códigos por incremento**
 - Implementación del Backend en Django, encargado de la lógica del negocio y la gestión de APIs REST.
 - Desarrollo del Frontend en Vue.js, proporcionando una interfaz interactiva y dinámica.
 - Integración del sistema con PostgreSQL para asegurar la confiabilidad y disponibilidad de los datos.
5. **Pruebas y validación**
 - Diseño y aplicación de pruebas unitarias, de caja blanca y caja negra, para verificar la correcta funcionalidad de cada módulo del sistema.

6. Integración de los incrementos.

- Diseño y ejecución de casos de prueba integrales para evaluar la funcionalidad del sistema en su totalidad.
- Evaluación preliminar con un grupo piloto de usuarios en el Laboratorio 5E para identificar posibles ajustes antes de su implementación a mayor escala.

7. Entrega al cliente

- Implementación y despliegue del sistema en un entorno controlado, en un laboratorio de prácticas seleccionado.
- Recopilación de datos sobre su desempeño, eficiencia operativa y percepción de los usuarios, para la evaluación y mejora del producto de software.

adecuada a la vista. En este proyecto, el controlador se implementa en Django mediante su sistema de vistas (views).

La Programación Orientada a Objetos (POO) es un paradigma de desarrollo de software que organiza el código en torno a objetos, Según Booch, Maksimchuk, Engle, Young, Connallen y Houston (2007) la POO facilita la abstracción de conceptos del mundo real en entidades del software, lo que mejora la comprensión, diseño y desarrollo del sistema.

Django: es un framework web de alto nivel desarrollado en Python, diseñado para facilitar el desarrollo rápido y estructurado de aplicaciones web. Creado en 2005 por Adrian Holovaty y Simon Willison, Django sigue el principio DRY (Don't Repeat Yourself). Django se utiliza como la tecnología principal para la construcción del backend, proporcionando la lógica de negocio, la interacción con la base de datos y la gestión de las APIs REST.

Vue.js: es un framework progresivo de JavaScript diseñado para construir interfaces de usuario dinámicas y aplicaciones web interactivas. Creado por Evan You en 2014, Vue.js se distingue por su simplicidad, flexibilidad y enfoque en la reactividad, lo que lo ha convertido en una de las herramientas más populares en el desarrollo frontend. Las funcionalidades desarrolladas en el proyecto fueron: Reactividad de Componentes, Gestión de Componentes, Enrutamiento Dinámico, Interfaz Intuitiva y Preparación para la escalabilidad.

SQL (Structured Query Language): SQL (Structured Query Language) es un lenguaje de programación es un lenguaje de programación ampliamente aceptado que se emplea para administrar y modificar bases de datos relacionales.

PostgreSQL: es un sistema de gestión de bases de datos relacional de código abierto que ha ganado reconocimiento mundial por su robustez, escalabilidad y extensibilidad. La aplicación en el proyecto En el prototipo del módulo de administración de disponibilidad de dispositivos electrónicos, PostgreSQL desempeñó un papel crucial como base para el almacenamiento y la gestión de datos. Las principales características aplicadas al proyecto fueron: Definición de Tablas y Esquemas, Relaciones entre Entidades, Consultas para el Backend, Índices y Optimización de Consultas, e Integridad de datos.

Visual Studio Code (VS Code): Visual Studio Code, conocido como VS Code, es un editor de código creado por Microsoft que se ha consolidado como una herramienta esencial para los programadores gracias a su versatilidad, eficiencia y capacidad para ser ampliado mediante extensiones.

DBeaver: Es una herramienta de administración de bases de datos multiplataforma diseñada para facilitar la gestión, visualización y análisis de datos. Compatible con múltiples sistemas de bases de datos, como PostgreSQL, MySQL, Oracle, y más, DBeaver es ampliamente utilizado por desarrolladores, administradores de bases de datos y analistas debido a su interfaz intuitiva y funcionalidades avanzadas.

Postman: es una herramienta popular para diseñar, probar y documentar APIs (Application Programming Interfaces).

C. Técnicas e Instrumentos

Para garantizar la validez y fiabilidad del proyecto, se emplearon diversas herramientas y técnicas de análisis:

- **Herramientas de desarrollo:** Fueron utilizados Visual Studio Code para la implementación del código; Postman para la prueba de las interfaces de programación de aplicaciones (API) y pgAdmin/DBeaver para la administración de la base de datos.
- **Pruebas de rendimiento:** Evaluación del tiempo de respuesta del sistema y análisis de registros de errores.
- **Evaluación cualitativa:** Aplicación de entrevistas semiestructuradas y observación directa del uso del sistema para obtener información sobre la experiencia de usuario.

El uso de la metodología mixta y de un enfoque incremental permitió validar de manera progresiva la funcionalidad del sistema, garantizando que su diseño e implementación responden a las necesidades reales del Instituto Tecnológico Superior de Lerdo.

D. Fundamento teórico

Modelo-Vista-Controlador (MVC): El patrón arquitectónico Modelo-Vista-Controlador (MVC) es uno de los enfoques más utilizados en el desarrollo de software para organizar aplicaciones de manera modular, escalable y eficiente. Sommerville (2015).

Modelo: El Modelo representa la lógica de negocio y la estructura de los datos del sistema. Es responsable de gestionar las interacciones con la base de datos y de mantener la integridad de los datos. En este prototipo, el modelo se implementa utilizando el framework Django, que proporciona un ORM (Object-Relational Mapping) robusto y eficiente.

Vista: La Vista se encarga de la presentación de la información al usuario y de la interacción con este. En este proyecto, las vistas se implementan con Vue.js, un framework frontend progresivo que permite crear interfaces dinámicas y altamente interactivas.

Controlador: El Controlador actúa como intermediario entre el modelo y la vista. Procesa las solicitudes del usuario, consulta o modifica los datos en el modelo y devuelve una respuesta

III. RESULTADOS Y DISCUSIÓN

Optimización Administrativa: Anteriormente, la administración de dispositivos era descentralizada, con registros inconsistentes. Con este sistema se contará con validaciones automáticas, historial técnico de cada dispositivo y visibilidad inmediata del estado operativo. La prueba piloto ha demostrado la reducción de los errores y la agilización en la atención de fallas, en comparación al proceso anterior.

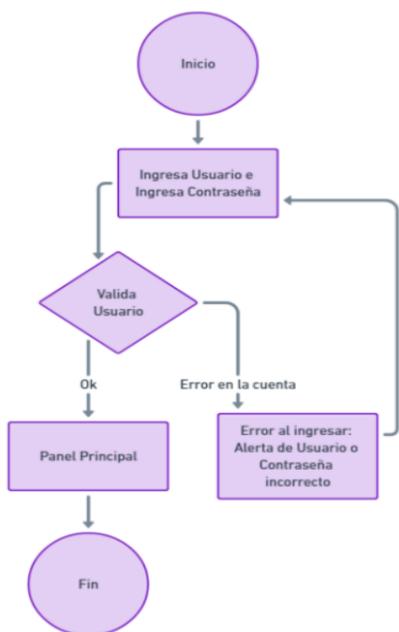
Reportes automatizados: El sistema genera reportes dinámicos exportables en Excel, categorizados por fecha, ubicación, tipo de dispositivo o tipo de falla. Estos reportes se actualizan automáticamente a partir de los registros internos, eliminando la necesidad de consolidación manual. Esta funcionalidad ha mejorado los tiempos de respuesta en decisiones administrativas.

Comparativa con soluciones comerciales: A diferencia de herramientas como LabStats, que requieren pagos anuales y configuración técnica externa, el modelo desarrollado permite: personalización completa, integración con plataformas institucionales, cero costos de licenciamiento y alineación con procesos internos. Esto representa una alternativa estratégica más viable para instituciones públicas.

Alcance: Con el fin de identificar y jerarquizar los requerimientos del Módulo de Administración, se procedió a la elaboración de los diagramas de flujo, casos de uso y de clases. Para ello, se empleó la herramienta orientada al Lenguaje Unificado de Modelado (UML) llamada ArgoUML®. En las figuras 1 a 4, se muestran los diagramas de casos de uso correspondientes a las funciones principales del sistema.

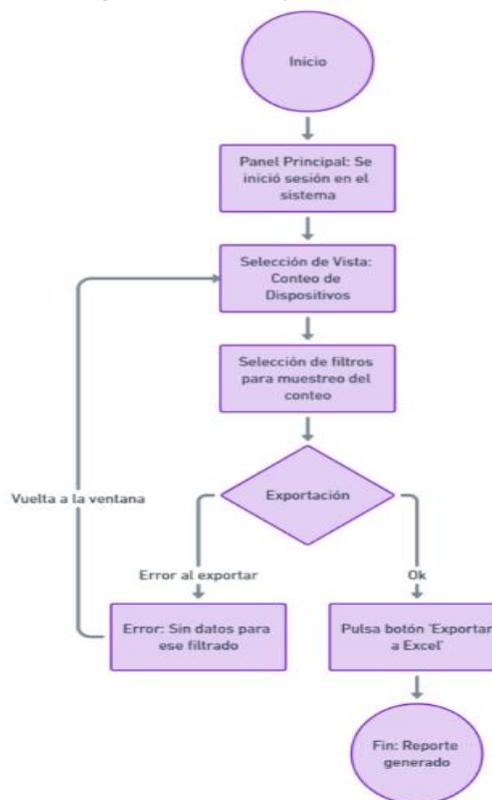
En la figura 1 se muestran los posibles flujos que se obtienen en el momento de inicio de sesión en el sistema.

Figura 1
Diagrama de flujo login.



En el diagrama de la figura 2 se muestra el flujo de la aplicación cuando un usuario desea revisar el conteo de los dispositivos.

Figura 2
Diagrama de flujo de conteo de dispositivos.



El diagrama de la figura 3 representa el flujo de gestión de dispositivos dentro de un sistema, iniciando desde un panel principal donde se visualizan los sistemas y se accede al módulo de Gestión de Dispositivos. Desde allí, el usuario puede elegir una de tres acciones: actualización, creación o eliminación de un dispositivo. Si se elige actualizar, se pulsa un botón, se rellenan los datos en un formulario y, si no hay errores, se actualiza exitosamente el dispositivo. En la creación, se pulsa el botón "Agregar Dispositivo", se completan los datos en el formulario, y si la validación es correcta, el dispositivo se crea exitosamente. Para la eliminación, se selecciona el dispositivo, se abre un modal, se confirma la acción, y tras validar, el dispositivo es eliminado exitosamente. En cualquiera de las acciones, si ocurre un error durante la validación, se muestra una alerta de error y se retorna al paso correspondiente. Finalmente, el proceso concluye cuando la acción se realiza correctamente.

En el diagrama presentado en la figura 4 se muestran los diferentes casos de uso para los 2 tipos de usuarios existentes en el sistema, siendo la principal diferencia que el usuario de tipo Administrador cuenta con la opción de gestionar a los usuarios del sistema.

Figura 3
Diagrama de gestión de dispositivos.

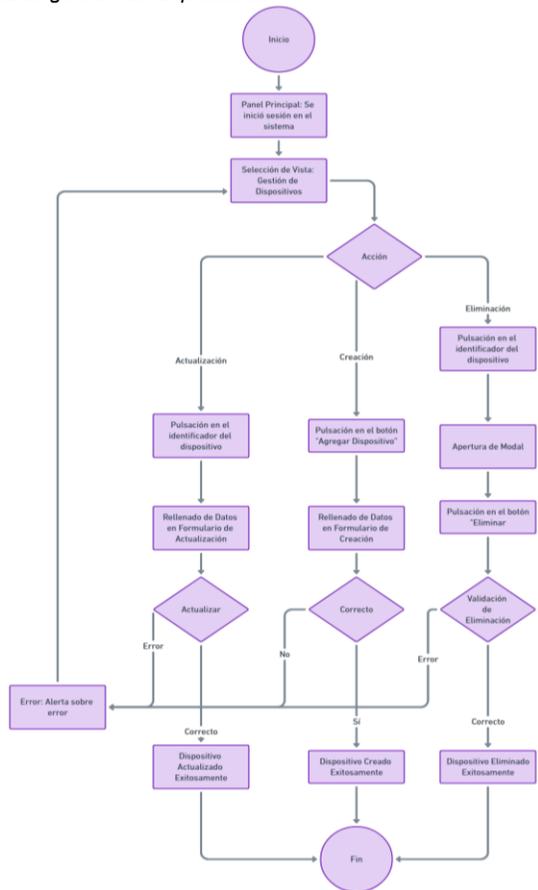


Figura 4
Diagrama de Casos de Uso.

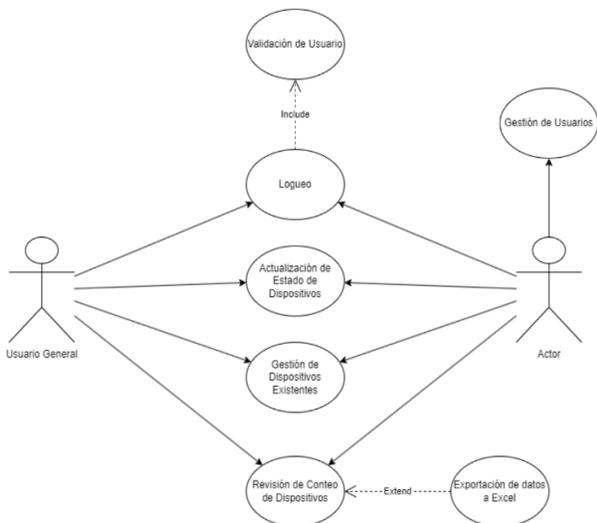
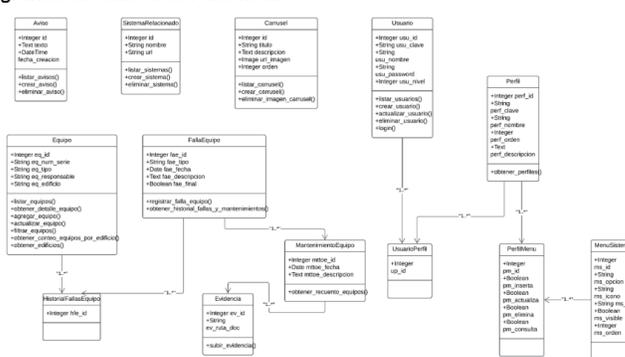


Figura 5
Diagrama de Clases del sistema.



El diagrama de la figura 5 muestra las características del sistema, en las cuales se encuentra el equipo, con la representación de sus atributos y métodos; Falta Equipo representa al equipo con detalles, fecha y estado; Mantenimiento Equipo detalla los mantenimientos realizados en los equipos con atributos como fecha y descripción.

La interfaz de usuario del módulo de Login (figura 6) presenta un diseño sencillo pero funcional, con un fondo dinámico que proporciona una experiencia visual atractiva. Incluye el nombre del sistema: M-DISP ITSL, junto con los campos para ingresar las credenciales de usuario (nombre de usuario y contraseña) y el botón de inicio de sesión. La implementación incluye validaciones robustas para garantizar que las credenciales ingresadas sean correctas. La herramienta implementada en las interfaces fue Vue.js, que permite la construcción de aplicaciones complejas, combinando simplicidad y flexibilidad.

Figura 6
Vista de Login.



El módulo de administración de usuarios y el menú dinámico sientan las bases para las funcionalidades avanzadas que se desarrollarán en fases posteriores, asegurando una arquitectura robusta y escalable (ver figura 7)

Figura 7
Vista de gestión de Usuarios.

Gestión de Usuarios			
ID	Nombre TI	Perfil TI	Acciones
admin	Administrador	Administrador	
General1	Usuario General	General	✔ ✘
General2	Usuario General 2	General	✔ ✘

La figura 8 muestra un formulario de una interfaz de usuario para crear un nuevo usuario en un sistema. Contiene los siguientes campos de entrada:

- Clave (probablemente una identificación única del usuario),
- Nombre (nombre del usuario),
- Contraseña (para iniciar sesión),
- Perfil (selección de rol o tipo de usuario mediante un menú desplegable).

La figura 9 muestra un formulario de interfaz gráfica para editar un usuario en un sistema. El título "Editar Usuario" indica que los datos existentes se pueden modificar. Los campos visibles son:

- Clave
- Nombre
- Nueva Contraseña (Opcional): permite ingresar una contraseña nueva, si se desea cambiarla,
- Perfil: muestra "General" y es un campo desplegable para cambiar el rol del usuario.

Figura 8
Vista de Creación de Usuarios.

Figura 9
Vista de edición de usuarios.

La interfaz gráfica mostrada en la figura 10, consolida las funcionalidades necesarias para un sistema integral de gestión de dispositivos, registrando fallas y mantenimientos, y generando reportes dinámicos y personalizados que permiten un monitoreo efectivo.

Figura 10
Vista de Gestión de Equipos y Estado.



La figura 11 muestra el conteo de equipos tecnológicos y las fallas registradas por cada edificio. En total, se contabilizan 27 equipos distribuidos entre switches (7), routers (7), monitores (4), laptops (5) y PCs de escritorio (4). En cuanto a las fallas, se registran cinco en total: una en switches, una en routers, dos en laptops y una en PC de escritorio. Por tipo de falla, se identifican tres fallas físicas, dos de software y dos eléctricas. Los edificios con más incidencias son el Edificio B y el Edificio D, cada uno con dos fallas, mientras que el Edificio C presenta solo una. Las fallas en el Edificio B fueron de tipo físico, en el D se dividieron entre física y de software, y en el C la única falla fue eléctrica.

Figura 11
Vista de Conteo de Dispositivos.

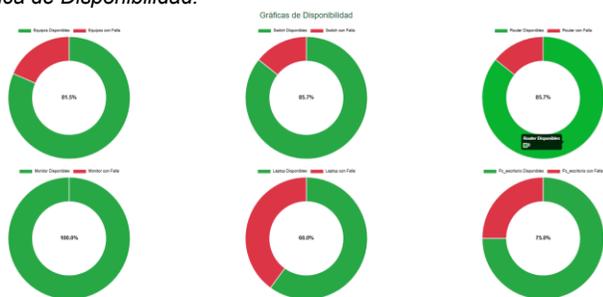
Filtrar por Edificio		Conteo de Equipos por Edificio													
Todos los edificios		+ Restricción a Edificio													
EDIFICIO	TOTAL EQUIPOS	SWITCH	ROUTER	MONITOR	LAPTOP	PC_ESCRITORIO	TOTAL FALLAS	FALLAS SWITCHE	FALLAS ROUTER	FALLAS MONITOR	FALLAS LAPTOP	FALLAS PC_ESCRITORIO	FALLA FISICA	FALLA DE SOFTWARE	FALLA ELECTRICA
Edificio B	8	2	3	2	1	0	2	1	0	0	1	0	2	0	0
Edificio D	6	2	1	0	1	2	1	0	0	0	0	1	1	0	0
Edificio E	4	2	0	0	1	1	0	0	0	0	0	0	0	0	0
Edificio A	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Edificio C	7	1	3	2	1	0	2	0	1	0	1	0	0	0	2
Total	27	7	7	4	5	4	5	1	1	0	2	1	3	0	2

La figura 12 presenta gráficas circulares que muestran el porcentaje de disponibilidad de distintos tipos de equipos tecnológicos. En general, el 81.5 % de los equipos están disponibles y el 18.5 % presentan fallas. Al analizar por tipo de equipo:

- Switches: presentan una disponibilidad del 85.7 %, con un 14.3 % con fallas.
- Routers: también muestran un 85.7 % de disponibilidad y 14.3 % de fallas.
- Monitores: se encuentran 100 % disponibles, sin fallas registradas.
- Laptops: son el tipo de equipo con mayor cantidad de fallas, ya que solo el 60.0 % están disponibles, mientras que el 40.0 % presentan fallas.
- PCs de escritorio: tienen una disponibilidad del 75.0 %, con un 25.0 % con fallas.

Estas gráficas evidencian que, aunque la mayoría de los equipos están en funcionamiento, las laptops y PC de escritorio requieren mayor atención por su mayor proporción de fallas.

Figura 12
Gráfica de Disponibilidad.



A continuación, se presentan los resultados de las encuestas aplicadas a 50 usuarios sobre el módulo de administración para la disponibilidad de dispositivos, utilizando métodos cuantitativos y escala de Likert (1 a 5), en la cual la respuesta 1 es “totalmente en desacuerdo” con la afirmación; el 2 expresa “desacuerdo”; 3 “ni de acuerdo ni en desacuerdo”; el 4 “de acuerdo” y el 5 “totalmente de acuerdo” con la afirmación. Las preguntas están formuladas para cubrir aspectos clave del uso, efectividad y percepción del módulo.

Tabla 1
Resultados con 50 usuarios.

No.	Pregunta	Media	% Respuestas 4-5	Desv. estándar
1	El módulo permite gestionar fácilmente la disponibilidad de dispositivos.	4.36	86%	0.65
2	La interfaz del módulo es intuitiva y fácil de usar.	4.28	80%	0.72
3	El tiempo para actualizar el estado de un dispositivo es adecuado.	4.20	78%	0.79
4	El módulo refleja en tiempo real la disponibilidad de los dispositivos.	4.10	76%	0.84
5	Se han reducido los errores al marcar dispositivos como disponibles o no disponibles.	4.22	82%	0.77
6	El módulo facilita la toma de decisiones sobre asignación de dispositivos.	4.18	74%	0.81
7	Se ha incrementado la eficiencia en el control del inventario de dispositivos.	4.25	80%	0.70
8	El sistema notifica correctamente los cambios en la disponibilidad de los dispositivos.	4.06	70%	0.89
9	La información del módulo es clara y confiable.	4.30	84%	0.68
10	En general, estoy satisfecho con el módulo de administración de disponibilidad de dispositivos.	4.40	88%	0.61

Figura 13
Gráfica de Resultados.



En la inferencia estadística el módulo es bien valorado por los usuarios, con medias entre 4.06 y 4.40, reflejando una percepción positiva en términos de usabilidad, exactitud y utilidad. La pregunta mejor evaluada fue la de satisfacción general (4.40), mientras que la más baja fue sobre la notificación de cambios (4.06), lo cual podría indicar un área de mejora. Las desviaciones estándar relativamente bajas sugieren consistencia en las respuestas.

IV. CONCLUSIONES

El desarrollo del módulo ha demostrado que es posible implementar una solución efectiva, sostenible y económica para la gestión de recursos tecnológicos en contextos institucionales. Las funcionalidades desarrolladas no sólo atienden una necesidad operativa, sino que también favorecen la autonomía tecnológica y administrativa del instituto.

El enfoque en reportes automatizados y optimización administrativa aporta valor agregado frente a soluciones preexistentes en el mercado, validando así el potencial de replicabilidad de este desarrollo en otras instituciones públicas con condiciones similares.

El presente trabajo logró desarrollar un prototipo funcional para la administración de la disponibilidad de dispositivos electrónicos, utilizando tecnologías modernas como Django, Vue.js y PostgreSQL. Este sistema aborda las principales problemáticas identificadas en el instituto, relacionadas con la falta de control centralizado, la ineficiencia en el registro de datos y la carencia de reportes confiables para la toma de decisiones.

Los resultados preliminares, obtenidos mediante pruebas controladas con datos simulados, validaron la correcta operación de los módulos desarrollados, incluyendo la gestión de usuarios, el registro de fallas y mantenimientos, la consulta en tiempo real del estado de los dispositivos y la generación automatizada de reportes. Estos logros confirman la viabilidad técnica del prototipo y su alineación con los objetivos del proyecto.

Sin embargo, el alcance de esta fase se limita a la validación funcional con datos simulados. La implementación y evaluación del sistema en un entorno real, como el Laboratorio de Cómputo 5E, sigue siendo una tarea pendiente que permitirá evaluar de manera más completa el impacto del sistema en usuarios finales y su interacción con datos operativos.

Entre las principales limitaciones identificadas se encuentra la ausencia de pruebas cualitativas con usuarios finales, así como la necesidad de incorporar funcionalidades adicionales, como la optimización de reportes según las necesidades específicas de los administradores.

V. AGRADECIMIENTOS

Las autoras expresan su más sincero agradecimiento, en primera instancia, al Dr. Jesús Alejandro Valdés Nieblas, titular de la Subdirección de Posgrado e Investigación del Instituto Tecnológico Superior de Lerdo (ITSL), por el valioso respaldo brindado en el desarrollo de este proyecto. Su apoyo institucional así como su disposición para facilitar los recursos y espacios necesarios, han sido fundamentales para la consolidación y avance de esta iniciativa.

De igual forma, se extiende un reconocimiento profundo a los estudiantes que, con compromiso, entusiasmo y dedicación, han participado activamente en las distintas etapas de implementación y mejora del sistema. La constancia y el esfuerzo demostrados por cada uno de ellos han contribuido significativamente a fortalecer la calidad del trabajo desarrollado, reflejando el espíritu de colaboración y superación que caracteriza a nuestra comunidad académica. Su participación ha sido clave para alcanzar los objetivos propuestos y continuar impulsando la innovación en el entorno tecnológico del ITSL.

VI. REFERENCIAS

Baez Bagatella, J. A. (2014). Sistema web para la administración de servicios informáticos de los laboratorios de cómputo de la FCC. Benemérita Universidad Autónoma de Puebla.

Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Connallen, J., & Houston, K. A. (2007). *Object-oriented analysis and design with applications*. (3rd ed.) Addison-Wesley.

DBeaver Community. (2025). *DBeaver: Universal Database Tool*. <https://dbeaver.com/docs/dbeaver/>

Django Software Foundation. (2024.). Django Documentation. <https://www.djangoproject.com>

Espinoza, F. (2023). Gestión de los recursos tecnológicos y el proceso de enseñanza y aprendizaje. Estudio correlacional en la ciudad de Lima, Perú. *UCV-SCIENTIA*, 15(1), 15–22. <https://doi.org/10.18050/revucv-scientia.v15n1a2>

Fielding, R. T. (2014). RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. <https://datatracker.ietf.org/doc/rfc7231/>

Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.

García, M., & López, R. (2021). Gestión de dispositivos electrónicos en entornos educativos: Desafíos y soluciones tecnológicas. *Revista de Innovación Educativa*, 25(2), 45-60.

Gómez, G. (2023). Diseño prototipo sistema para geolocalización y control de activos de cómputo por medio de tecnologías de RFID. [Tesis de Especialización, Universidad Distrital Francisco José de Caldas]. <https://repository.udistrital.edu.co/server/api/core/bitstreams/9eadbaa5-f067-4816-8af8-7f75090afadc/content>

Google. (2024.). *Google Workspace*. <https://workspace.google.com>

Hernández Sampieri, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. McGraw-Hill.

Holovaty, A., & Willison, S. (2005). *Django Documentation*. <https://www.djangoproject.com>

LabStats. (n.d.). LabStats Computer Lab Management. <https://www.labstats.com>

Lasso Guaman, J. J., & Masache Mosquera, J. C. (2024). Implementación del software ElabFTW para la gestión de proyectos de investigación y de equipos de laboratorio de CAREN-UTC y ECFN-YachayTech. Universidad Técnica de Cotopaxi.

Ley Federal del Derecho de Autor. D.O.F. 24 de diciembre de 1996. <https://www.diputados.gob.mx/LeyesBiblio/pdf/122.pdf>

Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados (LGPDPSSO). Nueva Ley D.O.F. 20-03-2025. <https://www.diputados.gob.mx/LeyesBiblio/pdf/LGPDPPSO.pdf>

PhoenixNAP. (2025). *Desarrollo incremental – Glosario de TI*. <https://phoenixnap.mx/glosario/desarrollo-incremental>

Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.) McGraw-Hill Education.

Quintanilla Padilla, R. E., & Cartagena Lobos, C. L. (2020). Aplicación de Internet de las Cosas IoT para escalar el monitoreo y gestión remota de dispositivos en instituciones educativas: Aplicación en ITCA-FEPADE Centro Regional Santa Ana. ITCA-FEPADE.

Sentrio (2023). *Diferencias entre el Desarrollo Iterativo e Incremental*. <https://sentrio.io/blog/diferencias-entre-desarrollo-iterativo-e-incremental/>

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.

Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.

Utz, M. (2013). *Learning Python* (5th ed.) O'Reilly Media.