Diseño e implementación de tecnología RFID para mejorar la seguridad en el monitoreo y vigilancia de los laboratorios de la Universidad Continental

**Escuela Académica Profesional de Ingeniería de Sistemas e Informática, Universidad Continental, Cusco, Perú.**

**RESUMEN EJECUTIVO**

El proyecto “Diseño e implementación de tecnología RFID para mejorar la seguridad en el monitoreo y vigilancia de los laboratorios de la Universidad Continental”, tiene como objetivo principal diseñar e implementar un sistema web con integración de tecnología RFID que permita fortalecer la seguridad y el monitoreo de equipos en los laboratorios de la Universidad Continental, sede Cusco, reduciendo errores humanos y controlando las salidas no autorizadas durante el año 2025.

Se realizaron 86 tareas organizadas en 29 historias de usuario que fueron repartidas en 4 sprints, las cuales alcanzaron un exitoso resultado en el sistema de alertas.

El proyecto aporta principalmente a los ODS 9, 11, 12 y 16, al mejorar la infraestructura tecnológica, promover una gestión responsable de recursos y fortalecer la transparencia institucional.

**Palabras clave:** RFID, sistema web, seguridad de laboratorios, IOT, monitoreo de equipos.

# ÍNDICE

Contenido

[ÍNDICE 2](#_Toc215627515)

[INDICE DE TABLAS 5](#_Toc215627516)

[INDICE DE FIGURAS 5](#_Toc215627517)

[I. INTRODUCCIÓN 6](#_Toc215627518)

[1.1 Antecedentes 6](#_Toc215627519)

[1.2.1 Problema general: 9](#_Toc215627520)

[1.2.2 Problemas específicos: 10](#_Toc215627521)

[1.3 Marco teórico 10](#_Toc215627522)

[1.3.1 Bases teóricas 10](#_Toc215627523)

[1.4 Objetivos del proyecto 14](#_Toc215627524)

[II. CONOCIMIENTOS DE INGENIERÍA APLICADOS / RELACIONADOS 14](#_Toc215627525)

[III. INGENIERO Y LA SOCIEDAD 15](#_Toc215627526)

[IV. METODOLOGÍA EMPLEADA 16](#_Toc215627527)

[4.1 Product Backlog 16](#_Toc215627528)

[**ix.** **Historia 3.3 — Consultar Historial de Cambios** 19](#_Toc215627529)

[4.2 Sprint Backlog 27](#_Toc215627530)

[Historia de Usuario 1.1: Iniciar Sesión en el Sistema 27](#_Toc215627531)

[Historia de Usuario 1.2: Control de Acceso por Roles 27](#_Toc215627532)

[Historia de Usuario 1.3: Cierre de Sesión Seguro 27](#_Toc215627533)

[Historia de Usuario 2.1: Registrar Equipos 27](#_Toc215627534)

[Historia de Usuario 2.2: Editar Equipos 28](#_Toc215627535)

[Historia de Usuario 2.3: Eliminar Equipos 28](#_Toc215627536)

[Historia de Usuario 3.1: Visualizar Alertas 28](#_Toc215627537)

[Historia de Usuario 3.2: Cambiar Estado de Alerta 29](#_Toc215627538)

[Historia de Usuario 3.3: Consultar Historial de Cambios 29](#_Toc215627539)

[Historia de Usuario 4.1: Recibir UID desde ESP32 29](#_Toc215627540)

[Historia de Usuario 4.2: Validar Equipo según UID 29](#_Toc215627541)

[Historia de Usuario 4.3: Generar Alerta Automática por UID 30](#_Toc215627542)

[Historia de Usuario 5.1: Generar Reporte de Inventario 30](#_Toc215627543)

[Historia de Usuario 5.2: Reportes de Alertas por Turno 30](#_Toc215627544)

[Historia de Usuario 5.3: Optimización de Interfaz y Rendimiento 31](#_Toc215627545)

[Historia de Usuario 6.1: Seguridad con Contraseñas Cifradas 31](#_Toc215627546)

[Historia de Usuario 6.2: Disponibilidad Mínima del Sistema (99%) 31](#_Toc215627547)

[Historia de Usuario 6.3: Optimización UI y Rendimiento 31](#_Toc215627548)

[Historia de Usuario 7.1: Manejo de Alertas con Alto Volumen 32](#_Toc215627549)

[Historia de Usuario 7.2: Compatibilidad con Tablets 32](#_Toc215627550)

[Historia de Usuario 7.3: Compatibilidad Multi-Navegador 32](#_Toc215627551)

[Historia de Usuario 8.1: Actualización sin Detener Servicio 32](#_Toc215627552)

[Historia de Usuario 8.2: Soporte Técnico Documentado 33](#_Toc215627553)

[Historia de Usuario 9.1: Trazabilidad Completa de Movimientos 33](#_Toc215627554)

[Historia de Usuario 9.2: Filtrado de Trazabilidad por Fecha 33](#_Toc215627555)

[Historia de Usuario 9.3: Exportación Completa de Trazabilidad 34](#_Toc215627556)

[Historia de Usuario 10.1: Optimización de Registro de Alertas 34](#_Toc215627557)

[Historia de Usuario 10.2: Optimización de Reportes 34](#_Toc215627558)

[Historia de Usuario 10.3: Búsquedas Instantáneas de Equipos 34](#_Toc215627559)

[V. USO DE HERRAMIENTAS MODERNAS 35](#_Toc215627560)

[5.1 Framework: VS Code 35](#_Toc215627561)

[Figura 1: Código del frontend y backend en VS Code 35](#_Toc215627562)

[5.2 Base de datos: MySql 35](#_Toc215627563)

[Figura 2: Administrador de base de datos “phpmyadmin” 35](#_Toc215627564)

[5.3 XAMPP: Es un paquete que integra Apache, MySQL, PHP y Perl para crear un entorno local de desarrollo web de forma rápida y sencilla. 36](#_Toc215627565)

[Figura 3: Aplicación de desarrollo XAMPP 36](#_Toc215627566)

[5.4 Git: Control de versiones 36](#_Toc215627567)

[Figura 4: Aplicación de control de versiones 37](#_Toc215627568)

[5.5 Lenguaje de programación PHP 37](#_Toc215627569)

[Figura 5: Aplicación del lenguaje de programación PHP 37](#_Toc215627570)

[5.6 Lenguaje de programación HTML 37](#_Toc215627571)

[Figura 6: Aplicación del lenguaje de programación HTML 37](#_Toc215627572)

[5.7 Lenguaje de programación CSS 38](#_Toc215627573)

[Figura 7: Aplicación del lenguaje de programación CSS 38](#_Toc215627574)

[5.8 IDE DE ARDUINO: Es un entorno de desarrollo utilizado para escribir, compilar y cargar programas en placas Arduino. 38](#_Toc215627575)

[Figura 8: Aplicación del uso de IDE ARDUINO 39](#_Toc215627576)

[5.9 Lenguaje de programación de IDE ARDUINO: Emplea un lenguaje basado en C/C++ simplificado. 39](#_Toc215627577)

[Figura 9: Aplicación del lenguaje de programación C/C++ 39](#_Toc215627578)

[VI. DISEÑO DE INGENIERÍA 39](#_Toc215627579)

[6.1 Requisitos Funcionales 39](#_Toc215627580)

[6.2 Requisitos No Funcionales 40](#_Toc215627581)

[6.3 Diseño de base de datos 41](#_Toc215627582)

[Figura 10: Diagrama relacional de la base de datos 45](#_Toc215627583)

[6.4 Arquitectura de la solución planteada 45](#_Toc215627584)

[Figura 11: Arquitectura del sistema 49](#_Toc215627585)

[VII. GESTIÓN DEL PROYECTO 50](#_Toc215627586)

[Figura 12: Diagrama de Gantt (Línea base) 50](#_Toc215627587)

[Figura 13: Diagrama de Gantt (Ejecutado) 50](#_Toc215627588)

[Figura 14: Arquitectura del sistema Tablero Scrum/Kanban 50](#_Toc215627589)

[VIII. PRUEBAS, RESULTADOS Y DISCUSIÓN 50](#_Toc215627590)

[PMV 1: Módulo de Autenticación y Gestión de Usuarios 50](#_Toc215627591)

[Figura 15: Inicio de sesión correcto 51](#_Toc215627592)

[Figura 16: Acceso restringido 52](#_Toc215627593)

[Figura 17: Cambio de rol del usuario 53](#_Toc215627594)

[Figura 18: Cierre de sesión 54](#_Toc215627595)

[PMV 2: Módulo de Gestión de Equipos (CRUD) 54](#_Toc215627596)

[Figura 19: Registrar equipo 55](#_Toc215627597)

[Figura 20: Edición de equipo 56](#_Toc215627598)

[Figura 21: Listado general 56](#_Toc215627599)

[Figura 22: Validación de datos 57](#_Toc215627600)

[PMV 3: Módulo de Alertas Automáticas (RFID) 57](#_Toc215627601)

[Figura 23: Alerta recibida 58](#_Toc215627602)

[Figura 24: Alerta de UID desconocido 58](#_Toc215627603)

[Figura 25: Filtro por atención 58](#_Toc215627604)

[Figura 26: Vista de detalle 58](#_Toc215627605)

[PMV 4: Módulo de Reportes y Dashboard 59](#_Toc215627606)

[Figura 27: Reporte de equipos 59](#_Toc215627607)

[Figura 28: Reporte de alertas 60](#_Toc215627608)

[Figura 29: Exportación PDF 60](#_Toc215627609)

[PMV 5: Integración Hardware (ESP32 + MFRC522 + Servidor Web) 60](#_Toc215627610)

[Figura 30: Lectura NFC 61](#_Toc215627611)

[Figura 31: Envío de UID 61](#_Toc215627612)

[Figura 32: Error de red controlado 61](#_Toc215627613)

[X. CONCLUSIONES 62](#_Toc215627614)

[X. BIBLIOGRAFIA 63](#_Toc215627615)

[XI. ANEXOS 65](#_Toc215627616)

[11.1 Evidencias de la funcionalidad de la aplicación por PMV en YouTube: 65](#_Toc215627617)

[11.2 Evidencias del código implementado de la aplicación: 65](#_Toc215627618)

[Imagen del repositorio 65](#_Toc215627619)

[Figura 33: Uso del Github 65](#_Toc215627620)

# INDICE DE TABLAS

[Historia de Usuario 1.1: Iniciar Sesión en el Sistema 27](#_Toc215627531)

[Historia de Usuario 1.2: Control de Acceso por Roles 27](#_Toc215627532)

[Historia de Usuario 1.3: Cierre de Sesión Seguro 27](#_Toc215627533)

[Historia de Usuario 2.1: Registrar Equipos 27](#_Toc215627534)

[Historia de Usuario 2.2: Editar Equipos 28](#_Toc215627535)

[Historia de Usuario 2.3: Eliminar Equipos 28](#_Toc215627536)

[Historia de Usuario 3.1: Visualizar Alertas 28](#_Toc215627537)

[Historia de Usuario 3.2: Cambiar Estado de Alerta 29](#_Toc215627538)

[Historia de Usuario 3.3: Consultar Historial de Cambios 29](#_Toc215627539)

[Historia de Usuario 4.1: Recibir UID desde ESP32 29](#_Toc215627540)

[Historia de Usuario 4.2: Validar Equipo según UID 29](#_Toc215627541)

[Historia de Usuario 4.3: Generar Alerta Automática por UID 30](#_Toc215627542)

[Historia de Usuario 5.1: Generar Reporte de Inventario 30](#_Toc215627543)

[Historia de Usuario 5.2: Reportes de Alertas por Turno 30](#_Toc215627544)

[Historia de Usuario 5.3: Optimización de Interfaz y Rendimiento 31](#_Toc215627545)

[Historia de Usuario 6.1: Seguridad con Contraseñas Cifradas 31](#_Toc215627546)

[Historia de Usuario 6.2: Disponibilidad Mínima del Sistema (99%) 31](#_Toc215627547)

[Historia de Usuario 6.3: Optimización UI y Rendimiento 31](#_Toc215627548)

[Historia de Usuario 7.1: Manejo de Alertas con Alto Volumen 32](#_Toc215627549)

[Historia de Usuario 7.2: Compatibilidad con Tablets 32](#_Toc215627550)

[Historia de Usuario 7.3: Compatibilidad Multi-Navegador 32](#_Toc215627551)

[Historia de Usuario 8.1: Actualización sin Detener Servicio 32](#_Toc215627552)

[Historia de Usuario 8.2: Soporte Técnico Documentado 33](#_Toc215627553)

[Historia de Usuario 9.1: Trazabilidad Completa de Movimientos 33](#_Toc215627554)

[Historia de Usuario 9.2: Filtrado de Trazabilidad por Fecha 33](#_Toc215627555)

[Historia de Usuario 9.3: Exportación Completa de Trazabilidad 34](#_Toc215627556)

[Historia de Usuario 10.1: Optimización de Registro de Alertas 34](#_Toc215627557)

[Historia de Usuario 10.2: Optimización de Reportes 34](#_Toc215627558)

[Historia de Usuario 10.3: Búsquedas Instantáneas de Equipos 34](#_Toc215627559)

# INDICE DE FIGURAS

# I. INTRODUCCIÓN

## 1.1 Antecedentes

El desarrollo tecnológico ha transformado los mecanismos de control y vigilancia en entornos educativos, industriales y urbanos. Según Vega Castillo [1], en los últimos años, el uso de tecnologías de identificación sin contacto, como RFID y NFC, ha ganado relevancia en la gestión de activos y la seguridad institucional. En esta sección se presentan antecedentes internacionales, nacionales y locales relacionados con el uso de sistemas electrónicos automatizados para el monitoreo de equipos y la protección de recursos físicos.

**Internacionales:** En la Universidad Distrital Francisco José de Caldas (Colombia) se implemento un sistema de control de acceso basado en tarjetas NFC y lectores biométricos de huella dactilar. Su propuesta redujo significativamente la intervención humana y los errores en el control de ingreso, demostrando la eficiencia de los sistemas automatizados en espacios educativos[2].

Por otro lado Garrote Sola [3] implementó una cerradura electrónica alimentada por una llave inteligente, orientada a incrementar la seguridad en laboratorios. El estudio evidenció que la automatización en los mecanismos de acceso incrementa la confiabilidad y reduce los incidentes de seguridad en áreas críticas.

Asimismo, investigaciones efectuadas en sectores industriales de India, han evidenciado que el uso de tecnologías web modernas junto con microcontroladores modernos de bajo consumo puede proporcionar un mecanismo eficaz para la gestión de activos. La implementación de una arquitectura web de tres niveles permite que el sistema sea escalable y tolerante a fallos. El uso de tecnologías web modernas y responsivas para crear la interfaz de usuario garantiza una mayor adaptabilidad, aceptabilidad y coherencia [4].

**Nacionales:** En la Pontificia Universidad Católica del Perú, también se implementó un sistema de control de activos para sus almacenes la aplicando tecnología RFID. Su propuesta mejoró la trazabilidad, redujo las pérdidas y optimizó la gestión del inventario institucional[5].

Así mismo en la Universidad Nacional Mayor de San Marcos se reportó que los registros en papel y hojas de cálculo no permiten un seguimiento eficiente de equipos de laboratorio, lo que ocasiona pérdidas, retrasos en auditorías y dificultades para identificar movimientos no autorizados [6].

**Locales:** Según Villarreal del Águila y Azpur Huillca [7], quienes desarrollaron un sistema RFID para controlar el ingreso y salida de dispositivos electrónicos en la Universidad Andina del Cusco, este tipo de tecnología permite fortalecer la seguridad institucional mediante la identificación automática de equipos. Estos resultados respaldan el proyecto al demostrar la eficacia del RFID en entornos universitarios.

Estos antecedentes demuestran la pertinencia y aplicabilidad de la tecnología RFID en contextos educativos, y evidencian la necesidad de desarrollar soluciones tecnológicas que fortalezcan la seguridad, la trazabilidad y el control automatizado de activos en los laboratorios universitarios.

**1.2 Identificación y formulación del problema**

En los laboratorios de la Universidad Continental, sede Cusco, se utilizan diversos equipos tecnológicos como computadoras, routers, proyectores y servidores. Estos activos son esenciales para las prácticas académicas; sin embargo, actualmente el control de su uso y salida se realiza de manera manual o mediante registros básicos, lo que genera limitaciones en la supervisión y aumenta la probabilidad de pérdidas o uso indebido.

La ausencia de un sistema automatizado provoca errores humanos, registros incompletos y una escasa capacidad de respuesta ante incidentes de seguridad. Esto se traduce en pérdidas económicas por la reposición de equipos, retrasos en las actividades académicas y una disminución de la confianza institucional en los mecanismos de control [7].

En el ámbito internacional, diversas instituciones educativas reportan dificultades para supervisar y controlar activos tecnológicos cuando se emplean únicamente registros manuales. Diversos estudios evidencian que, en universidades de México, persisten pérdidas de equipos debido a la falta de sistemas automatizados de monitoreo, lo que genera retrasos en las prácticas académicas y costos elevados de reposición [8].

De manera similar, investigaciones realizadas en universidades de Chile identifican fallas recurrentes en la trazabilidad de equipos de laboratorio, señalando que la dependencia del factor humano incrementa los errores y limita la detección temprana de incidentes [9].

En el contexto peruano, la problemática del control manual de activos continúa siendo significativa. Diversos estudios señalan que muchas universidades del país mantienen sistemas tradicionales basados en papel o registros básicos, lo que incrementa los incidentes de extravío de equipos y limita la capacidad de auditoría [10].  
Asimismo, se ha identificado que en instituciones educativas persisten dificultades para validar accesos y generar reportes confiables, afectando la seguridad física de laboratorios y talleres [11].

Ambos estudios muestran que la falta de automatización genera inconsistencias, duplicidad de información y escasa trazabilidad.

En el ámbito local, investigaciones recientes reportan dificultades en el monitoreo adecuado de equipos de laboratorio. Se ha evidenciado que la ausencia de sistemas automatizados provoca pérdidas recurrentes de equipos pequeños, problemas en auditorías y tiempos prolongados de verificación manual [12].  
De igual forma, informes técnicos de la región señalan que la mayoría de laboratorios universitarios aún emplean controles manuales que no permiten identificar salidas no autorizadas ni generar alertas en tiempo real [13].

Estas evidencias confirman que la problemática del control ineficiente de activos está presente en el contexto local y requiere la implementación de soluciones tecnológicas avanzadas.

Esta situación provoca tres consecuencias principales:

• Pérdida de equipos y aumento de costos institucionales por reposición.

• Errores humanos por la ausencia de un sistema automatizado.

• Baja confianza institucional en los mecanismos de seguridad.

El análisis sistémico del problema permite identificar los elementos clave que intervienen en la situación actual de los laboratorios de la Universidad Continental, sede Cusco.

**Entradas:**

• Equipos tecnológicos (computadoras, kits de electrónica, routers, proyectores, servidores).

• Usuarios (docentes, estudiantes, técnicos de laboratorio, personal de seguridad).

**Procesos actuales:**

• Control manual de salidas y entradas de equipos mediante registros básicos.

• Alta dependencia del factor humano.

• Limitada trazabilidad de movimientos y accesos.

**Salidas:**

• Pérdida o extravío de equipos.

• Errores en los registros.

• Baja eficiencia en la supervisión.

• Disminución de la confianza institucional en los mecanismos de seguridad.

**Retroalimentación:**

• Reposición de equipos perdidos (costos adicionales).

• Quejas de docentes y estudiantes por falta de control.

• Recomendaciones institucionales de mejorar los mecanismos de seguridad.

En función de lo anterior, los objetivos específicos de la investigación se orientan a resolver de manera estructurada cada aspecto del problema identificado. En primer lugar, el análisis de los procesos actuales permitirá evidenciar las limitaciones del control manual mediante observaciones directas y revisión de registros. Luego, el diseño de la arquitectura del sistema web con tecnología RFID se verificará a través de la elaboración de diagramas técnicos, especificación de módulos y evaluación funcional del prototipo.

La implementación del sistema en un entorno piloto se medirá por indicadores de precisión de lectura, tiempos de registro y reducción de errores humanos, obtenidos mediante pruebas controladas y consultas SQL a la base de datos.

Finalmente, la evaluación de la efectividad del sistema RFID en la mejora de la seguridad y monitoreo se sustentará mediante comparaciones antes y después de la implementación, encuestas de percepción a los usuarios y reportes de incidentes registrados.

1.2.1 Problema general:  
¿Cómo influye el diseño e implementación de un sistema web con tecnología RFID en la mejora de la seguridad y el monitoreo de los equipos en los laboratorios de la Universidad Continental, sede Cusco, durante el año 2025?

### 1.2.2 Problemas específicos:

* ¿Qué características debe poseer un sistema web con integración RFID para registrar de forma automática los movimientos de equipos en los laboratorios?
* ¿En qué medida la implementación del sistema web con RFID reduce los errores humanos en el control de equipos frente al método manual?
* ¿Cómo contribuye el sistema RFID a la disminución de incidentes de salidas no autorizadas de equipos?
* ¿Qué nivel de precisión y tiempo de respuesta logra el sistema en la identificación de los equipos registrados?

## 1.3 Marco teórico

### 1.3.1 Bases teóricas

#### 1.3.1.1 Tecnología RFID

La Identificación por Radiofrecuencia (RFID, por sus siglas en inglés) es un método de identificación automática que utiliza ondas electromagnéticas para transmitir datos entre un lector y una etiqueta (tag) adherida a un objeto. A diferencia del código de barras, RFID no requiere contacto ni línea de visión directa, lo que permite su uso en entornos complejos y con interferencias [14].

**Componentes principales**

Etiqueta o Tag: dispositivo electrónico con un chip y una antena que almacena un código único.

* Lector RFID: emite señales de radio para activar la etiqueta y leer la información.
* Antena: canaliza y amplifica la señal entre lector y etiqueta.
* Tipos de RFID
* Pasivas: sin batería, activadas por el campo electromagnético del lector.
* Activas: cuentan con batería y emiten su propia señal.
* Semipasivas: combinan características de ambas.

**Ventajas en entornos académicos**

* Reducción de errores humanos.
* Identificación automática y sin contacto.
* Generación de reportes de auditoría en tiempo real.
* Detección inmediata de accesos no autorizados.

En el contexto de esta investigación, se utilizará RFID pasiva de baja frecuencia (LF: 125–134 kHz), por su bajo costo, estabilidad ante interferencias y adecuación a espacios cerrados con estructuras metálicas, como los laboratorios universitarios.

#### 1.3.1.2 Seguridad y monitoreo

La seguridad electrónica se define como el conjunto de soluciones tecnológicas destinadas a proteger personas, bienes y activos frente a posibles amenazas físicas o digitales. Incluye sistemas de alarmas, sensores, cámaras, control de acceso y software de gestión centralizada [15].

En entornos académicos, la seguridad y el monitoreo implican garantizar la integridad de los recursos tecnológicos y mantener la trazabilidad de su uso. Esto requiere de mecanismos automáticos de registro, vigilancia y alerta.

Dimensiones principales de la seguridad institucional

**Prevención:** Detección temprana de riesgos o accesos no autorizados.

**Disuasión**: Reducción de intentos de sustracción mediante visibilidad y control tecnológico.

**Reacción:** Respuesta rápida ante incidentes.

**Supervisión continua:** Monitoreo remoto en tiempo real.

La integración de un sistema RFID permite automatizar los procesos de supervisión, minimizando la intervención humana y fortaleciendo la confiabilidad de los datos registrados. Al combinar hardware (lectores RFID) con software (plataforma web), se obtiene una herramienta integral de gestión de seguridad institucional.

En este sentido, la seguridad debe concebirse como un sistema de controles, políticas y procedimientos que garanticen la confidencialidad, integridad y disponibilidad de los recursos físicos y de la información, conforme a los principios normativos de gestión de seguridad de la información.

Adicionalmente, marcos de gobernanza de TI, como COBIT, y normas de gestión de seguridad como ISO/IEC 27001 establecen que la eficacia de cualquier solución tecnológica depende de su alineamiento con procesos, capacitación del personal y controles organizacionales [16]. La percepción de seguridad por parte de los usuarios (docentes, técnicos y personal de seguridad) constituye un componente clave de la variable deCompletado: un sistema técnicamente robusto, pero con baja aceptación operativa limitará su impacto real.

Por tanto, en esta investigación la variable Seguridad y monitoreo en los laboratorios se entiende como un constructo multidimensional que combina:

* La detección temprana de riesgos o accesos no autorizados.
* La reducción de intentos de sustracción mediante visibilidad y control tecnológico.
* Una respuesta rápida ante incidentes.
* El monitoreo remoto en tiempo real.

#### 1.3.1.3 Internet de las Cosas (IOT)

Es una red que conecta objetos físicos a internet para que recojan, procesen y transmitan información. Su objetivo es crear entornos inteligentes donde los dispositivos interactúen automáticamente con sistemas de control y análisis [17].

El presente proyecto implementa una solución basada en el Internet de las Cosas(IOT) para el monitoreo y control de equipos mediante identificación por radiofrecuencia (RFID). La arquitectura del sistema combina hardware embebido, comunicación inalámbrica y una plataforma web, siguiendo los principios fundamentales de IOT.

**Componentes Principales**

**Hardware Embebido**

* ESP32: Microcontrolor de bajo consumo que actúa como el cerebro del sistema, encargado de procesar las lecturas RFID y gestionar la comunicación con el servidor.
* Lector RFID de Alta Frecuencia: Permite la identificación única de cada equipo mediante etiquetas RFID pasivas, con un rango de lectura de hasta 1 metro.

**Arquitectura de Comunicación**

* Protocolo HTTP/HTTPS: Se utiliza para la comunicación segura entre los dispositivos ESP32 y el servidor web, garantizando la integridad y confidencialidad de los datos.
* Arquitectura Cliente-Servidor: Los dispositivos ESP32 actúan como clientes que envían solicitudes al servidor web, el cual procesa la información y actualiza la base de datos en tiempo real.

**Funcionamiento del Sistema**

* **Proceso de Identificación**
  + Cuando un usuario acerca una etiqueta RFID al lector, el ESP32 captura el ID único.
  + El microcontrolador envía esta información al servidor a través de una petición HTTP POST.
  + El servidor valida la etiqueta contra la base de datos y registra el acceso.
* **Gestión de Alertas**
  + El sistema puede detectar intentos de acceso no autorizados y generar alertas inmediatas.
  + Las notificaciones se envían a los administradores a través de la plataforma web.
  + Se registran todos los eventos para auditoría y generación de reportes.
* **Ventajas de la Solución IOT**
  + Escalabilidad: Fácil de expandir añadiendo más dispositivos sin modificar la infraestructura existente.
  + Tiempo Real: La información se actualiza instantáneamente en la plataforma web.
  + Bajo Consumo: Los dispositivos ESP32 están optimizados para operar con un consumo mínimo de energía.
  + Acceso Remoto: Los administradores pueden monitorear el sistema desde cualquier ubicación con acceso a internet.

Esta implementación demuestra cómo la integración de tecnologías IOT puede optimizar los procesos de control y monitoreo de equipos, ofreciendo una solución robusta, escalable y de bajo costo para la gestión institucional.

#### 1.3.1.4 Definición de términos básicos

* RFID (Radio Frequency Identification): Tecnología que usa ondas de radio para identificar objetos o personas de forma automática.
* Etiqueta RFID (Tag): Dispositivo electrónico que almacena un identificador único y se adhiere a un objeto.
* Lector RFID: Módulo que emite y recibe señales para detectar la información contenida en las etiquetas.
* IOT (Internet of Things): Red de objetos conectados capaces de intercambiar datos.
* ESP32: Microcontrolador programable con conectividad WiFi y Bluetooth, usado en sistemas IOT.
* Sistema web: Aplicación alojada en un servidor accesible desde cualquier navegador.
* Monitoreo: Supervisión continua de procesos o recursos mediante herramientas tecnológicas.
* Trazabilidad: Capacidad de seguir el recorrido o uso de un activo desde su origen hasta su destino.
* Seguridad electrónica: Uso de tecnologías para proteger activos, instalaciones y personas mediante automatización.
* API REST: Interfaz que permite la comunicación entre sistemas web y dispositivos conectados.

## 1.4 Objetivos del proyecto

**Objetivo general:**  
Diseñar e implementar un sistema web con integración de tecnología RFID que permita fortalecer la seguridad y el monitoreo de equipos en los laboratorios de la Universidad Continental, sede Cusco, reduciendo errores humanos y controlando las salidas no autorizadas durante el año 2025.

**Objetivos específicos:**

* Analizar los procesos actuales de control y monitoreo en los laboratorios, identificando las deficiencias del método manual que afectan la seguridad y trazabilidad de los equipos.
* Diseñar la arquitectura del sistema web con tecnología RFID, definiendo los módulos necesarios para el registro y monitoreo automatizado de equipos.
* Implementar un prototipo funcional del sistema web con RFID que permita la identificación y control automatizado de equipos en un entorno de laboratorio piloto.
* Evaluar la efectividad del sistema RFID en la mejora de la seguridad y monitoreo mediante indicadores de reducción de errores y tiempos de registro.

# II. CONOCIMIENTOS DE INGENIERÍA APLICADOS / RELACIONADOS

* **Conocimientos en Matemáticas:**
  + Estadística y análisis de datos para el cálculo de indicadores (precisión de lectura, reducción de errores, tiempos de auditoría, tasas de incidentes).
  + Uso de porcentajes, promedios, desviaciones estándar e indicadores comparativos pre/post implementación.
* **Conocimientos en ciencias naturales:**
  + Fundamentos de electromagnetismo y propagación de ondas de radio que sustentan el funcionamiento de RFID.
  + Consideración de interferencias por materiales metálicos, distancias y ángulos en las pruebas de lectura.
* **Conocimientos en Ingeniería:**
  + Ingeniería de software: análisis y levantamiento de requerimientos, diseño modular, arquitectura en capas, uso de UML, pruebas unitarias e integración.
  + Ingeniería de sistemas e informática: diseño de sistemas IOT, integración hardware-software, comunicaciones de datos, seguridad informática básica.
  + Electrónica y sistemas embebidos: configuración de ESP32, conexión de lectores MFRC522, diseño de circuitos de lectura y notificación.

# III. INGENIERO Y LA SOCIEDAD

**Justificación social:**  
El sistema contribuye a proteger el patrimonio tecnológico de la universidad, garantizando la continuidad de las prácticas académicas y reduciendo el impacto de pérdidas o mal uso de equipos en la formación de los estudiantes. Mejora la confianza de la comunidad universitaria en los mecanismos de control y fortalece la cultura de responsabilidad sobre los recursos.

**Justificación económica:**  
La reducción de pérdidas y extravíos de equipos disminuye costos de reposición y mantenimiento. El uso de tecnología RFID pasiva y ESP32 permite una solución de bajo costo y escalable, más económica que sistemas comerciales complejos, y aprovechable en otros laboratorios y sedes.

**Justificación ambiental:**  
Al optimizar el uso y la vida útil de los equipos, se evita el descarte prematuro de dispositivos electrónicos, contribuyendo a un consumo más responsable. Además, la digitalización de registros reduce el uso de papel en procesos de control y auditoría.

**Acontecimientos tecnológicos y científicos:**  
El proyecto se apoya en avances en RFID, IOT y seguridad electrónica documentados por autores como Finkenzeller y Gubbi [18], así como en experiencias previas de control de activos en universidades y empresas. Estos antecedentes demuestran la madurez de la tecnología y su pertinencia para entornos educativos.

# IV. METODOLOGÍA EMPLEADA

Para desarrollar el sistema web se usó una Metodología SCRUM por su flexibilidad y la iteratividad que permite volver a realizar una tarea de ser necesario.

El desarrollo tiene sprints cortos (1–2 semanas), backlog de tareas y revisiones periódicas.

## 4.1 Product Backlog

Se identificaron 29 historias de usuario en base a los requerimientos funcionales y no funcionales.

1. **Historia 1.1 — Iniciar Sesión en el Sistema**

* **Descripción:** Como usuario, quiero iniciar sesión con mi usuario y contraseña para acceder a las funciones del sistema.
* **Criterios de Aceptación:**
  + El sistema valida usuario y contraseña correctamente.
  + El sistema muestra mensajes de error cuando las credenciales son incorrectas.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Vista de login funcional.
  + Validación con password\_verify implementada.
  + Redirección al dashboard según rol.

1. **Historia 1.2 — Control de Accesos por Rol**

* **Descripción:** Como administrador, quiero que el sistema restrinja el acceso a módulos según el rol del usuario.
* **Criterios de Aceptación:**
  + El administrador accede a usuarios, equipos, alertas y reportes.
  + El técnico solo accede a alertas y reportes.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Middleware/validación implementada
  + Sección oculta para usuarios sin permisos.

Pruebas unitarias aplicadas.

1. **Historia 1.3 — Cierre de Sesión Seguro**

* **Descripción:** Como usuario, quiero cerrar sesión para evitar accesos no autorizados a mi cuenta.
* **Criterios de Aceptación:**
  + Sesión destruida correctamente.
  + Redirección al login.
* **Estimación:** 1 punto
* **Definición de "Hecho":**
  + Función de logout implementada.
  + Verificación de que no sea posible acceder a rutas restringidas sin sesión.

1. **Historia 2.1 — Registrar Equipos**

* **Descripción:** Como administrador, quiero registrar nuevos equipos con número de tarjeta RFID para mantener actualizado el inventario.
* **Criterios de Aceptación:**
  + Validación de campos (nombre, ubicación, RFID).
  + El sistema muestra mensajes de éxito/error.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Equipo almacenado en la base de datos.
  + Validaciones en frontend y backend funcionando.

1. **Historia 2.2 — Editar Equipos**

* **Descripción:** Como administrador, quiero editar la información de un equipo para corregir o actualizar datos.
* **Criterios de Aceptación:**
  + Edición de nombre, ubicación, estado, tarjeta RFID.
  + Validación de datos modificados.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Cambios en la base de datos.
  + Vista de edición funcional sin errores.

1. **Historia 2.3 — Eliminar Equipos**

* **Descripción:** Como administrador, quiero eliminar equipos que ya no existan para mantener limpio el inventario.
* **Criterios de Aceptación:**
  + Confirmación antes de eliminar.
  + Eliminación segura sin romper llaves foráneas.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Registro eliminado exitosamente.
  + Lista de equipos actualizada automáticamente.

1. **Historia 3.1 — Visualizar Alertas**

* **Descripción:** Como técnico, quiero visualizar todas las alertas generadas para monitorear incidentes.
* **Criterios de Aceptación:**
  + Filtros por estado y fecha.
  + Las alertas muestran nombre, ubicación, nivel y fecha**.**
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Listado funcional.
  + Conexión correcta con base de datos.

1. **Historia 3.2 — Cambiar Estado de Alerta**

* **Descripción:** Como técnico, quiero cambiar una alerta a “Atendido” o “No atendido” para mantener un registro actualizado.
* **Criterios de Aceptación:**
  + Botón o select para cambiar estado.
  + Registro del usuario que hizo el cambio.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Actualización guardada en BD.
  + Mensajes visuales de confirmación.

1. **Historia 3.3 — Consultar Historial de Cambios**

* **Descripción:** Como técnico, quiero ver el historial de cambios de estado de una alerta para conocer qué acciones se realizaron.
* **Criterios de Aceptación:**
  + Vista con cambios ordenados por fecha.
  + Visualización del usuario que realizó cada cambio.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Tabla de historial funcionando.
  + Integración con alertas existente.

1. **Historia 4.1 — Recibir UID desde ESP32**

* **Descripción:** Como sistema, quiero recibir un UID enviado por el ESP32 para procesarlo automáticamente.
* **Criterios de Aceptación:**
  + Petición HTTP recibida correctamente.
  + Verificación de integridad del UID.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Endpoint funcionando.
  + Prueba con ESP32 realizada.

1. **Historia 4.2 — Validar Equipo según UID**

* **Descripción:** Como sistema, quiero validar si el UID recibido del ESP32 pertenece a un equipo registrado.
* **Criterios de Aceptación:**
  + Búsqueda exacta por numero\_tarjeta\_rfid.
  + Retornar nombre del equipo cuando aplica.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Consulta en BD funcionando.
  + Casos de UID no identificado gestionados correctamente.

1. **Historia 4.3 — Generar Alerta Automática por UID**

* **Descripción:** Como sistema, quiero generar una alerta automáticamente al recibir un UID desde el ESP32.
* **Criterios de Aceptación:**
  + Alertas generadas cuando el UID corresponde a un equipo.
  + Alertas de “no identificado” si el UID no existe.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Inserción correcta en BD.
  + Visualización inmediata en el módulo de alertas.

1. **Historia 5.1 — Generar Reporte de Equipos**

* **Descripción:** Como administrador, quiero generar un reporte simple de equipos para revisar el inventario.
* **Criterios de Aceptación:**
  + Listado visible en el navegador.
  + Opciones básicas de filtrado.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Consulta SQL funcionando.
  + Vista de reporte accesible.

1. **Historia 5.2 — Generar Reporte de Alertas**

* **Descripción:** Como técnico o administrador, quiero ver un reporte con todas las alertas generadas para analizar incidentes.
* **Criterios de Aceptación:**
  + Filtro por fecha y estado.
  + Resultados ordenados por fecha.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Listado generado y funcional.
  + Integrado con alertas existentes.

1. **Historia 5.3 — Filtrar Información en Reportes**

* **Descripción:** Como usuario, quiero aplicar filtros de fecha, estado y equipo para obtener reportes personalizados.
* **Criterios de Aceptación:**
  + Filtros aplicados sin errores.
  + Resultados correctos según criterio.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Filtros probados y validados.
  + Integración en interfaz completada.

1. **Historia 6.1 — Seguridad con Contraseñas Cifradas**

* **Descripción:** Como admin, quiero que las contraseñas se almacenen usando bcrypt para proteger credenciales.
* **Criterios de Aceptación:**
  + Uso de password\_hash/bcrypt al almacenar y password\_verify al autenticar.
  + Flujo de cambio/recuperación de contraseña seguro (token expirado).
  + No almacenar contraseñas en texto plano en logs.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Migración/actualización de registros (si aplica).
  + Pruebas de autenticación y restablecimiento.
  + Documentación y checklist de seguridad.

1. **Historia 6.2 — Disponibilidad Mínima del Sistema (99%)**

* **Descripción:** Como admin, quiero garantizar disponibilidad mínima del 99% con manejo de errores y reconexiones.
* **Criterios de Aceptación:**
  + Manejo de reconexión en capa BD (reintentos).
  + Endpoint /healthz/status que reporte estado del sistema.
  + Plan sencillo de respaldo/restore (script automático básico).
* **Estimación:** 4 puntos
* **Definición de "Hecho":**
  + Reintentos e indicadores implementados y probados.
  + Health-check en staging y script de backup probado.
  + Documentación de runbook.

1. **Historia 6.3 — Optimización de Interfaz y Rendimiento**

* **Descripción:** Como usuario, quiero interfaz rápida y responsive con tiempos de carga aceptables.
* **Criterios de Aceptación:**
  + Implementar lazy loading y fetch asíncrono donde aplique.
  + Minificación y concatenación de assets.
  + Tiempos de respuesta aceptables en operaciones comunes (<2 s objetivo).
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Cambios frontend aplicados y medidos.
  + Pruebas de carga básicas y documentación de mejoras.

1. **Historia 7.1 — Manejo de Alertas con Alto Volumen**

* **Descripción:** Como técnico, quiero que el sistema maneje grandes volúmenes de alertas simultáneas sin pérdida de rendimiento.
* **Criterios de Aceptación:**
  + Inserciones optimizadas (batch/transaction) o uso de cola.
  + Mecanismo para procesar y visualizar alertas sin bloqueo.
  + Pruebas de concurrencia básicas.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Implementación de inserciones batched o cola ligera.
  + Pruebas de concurrencia ejecutadas y documentadas.

1. **Historia 7.2 — Compatibilidad con Tablets**

* **Descripción:** Como técnico de campo, quiero acceder desde tablets para consultar alertas y equipos.
* **Criterios de Aceptación:**
  + Diseño responsive adaptado a breakpoints de tablet.
  + Elementos táctiles suficientemente grandes y usables.
  + Pruebas manuales en al menos un navegador móvil/tablet.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + CSS y comportamientos táctiles implementados.
  + Pruebas y ajustes realizados.

1. **Historia 7.3 — Compatibilidad Multi-Navegador**

* **Descripción:** Como usuario, quiero acceder desde Chrome, Firefox y Edge sin problemas.
* **Criterios de Aceptación:**
  + Funcionalidades principales probadas en Chrome/Firefox/Edge (últimas versiones).
  + Corregir incompatibilidades JS/CSS detectadas.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Informe de pruebas multi-navegador y correcciones aplicadas.
  + Exportaciones y descargas verificadas en los tres navegadores.

1. **Historia 8.1 — Actualización sin Detener Servicio**

* **Descripción:** Como admin, quiero actualizar módulos sin interrumpir el servicio para mantener continuidad operativa.
* **Criterios de Aceptación:**
  + Patrón de includes/modularización aplicado para facilitar despliegues parciales.
  + Procedimiento de migración DB que sea safe (cambios aditivos).
  + Documentación del proceso de despliegue seguro.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Modularización básica completada.
  + Checklist de deploy sin downtime documentado y probado en staging.

1. **Historia 8.2 — Soporte Técnico Documentado**

* **Descripción:** Como administrador, quiero documentación completa de soporte técnico y manual de usuario.
* **Criterios de Aceptación:**
  + Manual de soporte con pasos de resolución para incidentes comunes.
  + Comandos DB y scripts de respaldo/restore documentados.
  + Guía de despliegue y rollback.
* **Estimación:** 5 puntos
* **Definición de "Hecho":**
  + Manual disponible en repo / views/manual\_soporte.php o docx.
  + Revisión por un par y aprobación.

1. **Épica 9: Trazabilidad**

**Historia 9.1 — Trazabilidad Completa de Movimientos**

* **Descripción:** Como auditor, quiero que cada movimiento se registre con fecha, hora, usuario y tipo, para trazabilidad total.
* **Criterios de Aceptación:**
  + Registro obligatorio por cada movimiento con los campos mínimos (timestamp, usuario, tipo, equipo, notas).
  + Integridad e inmutabilidad básica (no sobrescribir registros).
  + Vista consultable y exportable.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Tabla de trazabilidad y logging implementados.
  + UI para consulta y exportación lista.
  + Pruebas y documentación.

1. **Historia 9.2 — Filtrado de Trazabilidad por Fecha**

* **Descripción:** Como auditor, quiero filtrar trazabilidad por rangos de fecha para análisis puntual.
* **Criterios de Aceptación:**
  + Filtros por fecha inicio/fin, paginación y orden cronológico.
  + Performance aceptable al aplicar rangos razonables.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Filtros implementados y optimizados.
  + Pruebas de performance básicas.

1. **Historia 9.3 — Exportación Completa de Trazabilidad**

* **Descripción:** Como auditor, quiero exportar la trazabilidad completa en PDF/Excel con todos los detalles.
* **Criterios de Aceptación:**
  + Exportación masiva soportada (batching para datasets grandes).
  + Formato conservando todos los campos (timestamps, usuario, tipo, antes/después).
* **Estimación:** 4 puntos
* **Definición de "Hecho":**
  + Export funcional con manejo de datasets grandes (background job si necesario).
  + Pruebas y documentación.

1. **Historia 10.1 — Optimización de Registro de Alertas**

* **Descripción:** Como usuario, quiero que el registro de alertas RFID sea rápido (<2 s) para eficiencia operativa.
* **Criterios de Aceptación:**
  + INSERTs optimizados (prepared statements, transacciones rápidas).
  + Índices adecuados en tablas de alertas.
  + Medición de latencia antes y después de optimización.
* **Estimación:** 3 puntos
* **Definición de "Hecho":**
  + Cambios aplicados y mediciones que demuestren mejora.
  + Documentación de índices y ajustes.

1. **Historia 10.2 — Optimización de Reportes**

* **Descripción:** Como usuario, quiero que la generación de reportes tome menos tiempo.
* **Criterios de Aceptación:**
  + Caching de resultados pesados o preagregación donde aplique.
  + Paginación y generación en background para conjuntos grandes.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Cache o pre-aggregation implementada.
  + Tests de rendimiento y documentación.

1. **Historia 10.3 — Búsquedas Instantáneas de Equipos**

* **Descripción:** Como usuario, quiero búsquedas instantáneas por UID RFID con respuesta inmediata.
* **Criterios de Aceptación:**
  + Autocomplete / lookup con AJAX y latencia baja (<1 s en condiciones normales).
  + Índices en BD para numero\_tarjeta\_rfid aplicados.
* **Estimación:** 2 puntos
* **Definición de "Hecho":**
  + Endpoint y frontend de autocomplete implementados.
  + Pruebas de latencia y documentación.

## 4.2 Sprint Backlog

**SPRINT 1**

## Historia de Usuario 1.1: Iniciar Sesión en el Sistema

**Descripción:** Como usuario, quiero iniciar sesión con mi usuario y contraseña para acceder a las funciones del sistema.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 1.1.1 | Crear formulario de login | BORIS | 1 | Completado |
| 1.1.2 | Implementar validación con password\_verify | BORIS | 1 | Completado |
| 1.1.3 | Programar redirección según rol | BORIS | 0.5 | Completado |

## Historia de Usuario 1.2: Control de Acceso por Roles

**Descripción:** Como administrador, quiero que el sistema restrinja el acceso a módulos según el rol del usuario.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 1.2.1 | Implementar middleware/validación de roles | BORIS | 1 | Completado |
| 1.2.2 | Ocultar elementos de menú según rol | BORIS | 1 | Completado |
| 1.2.3 | Crear pruebas unitarias básicas de acceso | BORIS | 1 | Completado |

## Historia de Usuario 1.3: Cierre de Sesión Seguro

**Descripción:** Como usuario, quiero cerrar sesión para evitar accesos no autorizados a mi cuenta.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 1.3.1 | Implementar función logout (destruir sesión) | BORIS | 0.5 | Completado |
| 1.3.2 | Crear botón UI “Cerrar sesión” | BORIS | 0.5 | Completado |
| 1.3.3 | Validar que no acceda a rutas sin sesión | BORIS | 0.5 | Completado |

## Historia de Usuario 2.1: Registrar Equipos

**Descripción:** Como administrador, quiero registrar nuevos equipos con número de tarjeta RFID para mantener actualizado el inventario.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 2.1.1 | Crear formulario de equipos | BORIS | 1 | Completado |
| 2.1.2 | Validar nombre, ubicación y RFID | BORIS | 1 | Completado |
| 2.1.3 | Insertar equipo en la BD | BORIS | 1 | Completado |

## Historia de Usuario 2.2: Editar Equipos

**Descripción:** Como administrador, quiero editar la información de un equipo para corregir o actualizar datos.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 2.2.1 | Crear vista editar.php | BORIS | 1 | Completado |
| 2.2.2 | Actualizar registro en BD | BORIS | 1 | Completado |
| 2.2.3 | Validaciones de campos editados | BORIS | 1 | Completado |

## Historia de Usuario 2.3: Eliminar Equipos

**Descripción:** Como administrador, quiero eliminar equipos que ya no existan para mantener limpio el inventario.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 2.3.1 | Agregar botón “Eliminar” en listado | BORIS | 0.5 | Completado |
| 2.3.2 | Programar eliminación segura (sin romper FK) | BORIS | 1 | Completado |
| 2.3.3 | Actualizar vista automáticamente | BORIS | 1 | Completado |

**SPRINT 2**

## Historia de Usuario 3.1: Visualizar Alertas

**Descripción:** Como técnico, quiero visualizar todas las alertas generadas para monitorear incidentes.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 3.1.1 | Diseño de tabla de alertas en vista | BORIS | 1 | Completado |
| 3.1.2 | Consulta SQL con filtros | BORIS | 1 | Completado |
| 3.1.3 | Conectar vista con controlador | BORIS | 1 | Completado |

## Historia de Usuario 3.2: Cambiar Estado de Alerta

**Descripción:** Como técnico, quiero cambiar una alerta a “Atendido” o “No atendido” para mantener un registro actualizado.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 3.2.1 | Crear botón/select para cambiar estado | BORIS | 1 | Completado |
| 3.2.2 | Actualizar estado en BD con usuario | BORIS | 1 | Completado |
| 3.2.3 | Mensajes visuales de cambio | BORIS | 0.5 | Completado |

## Historia de Usuario 3.3: Consultar Historial de Cambios

**Descripción:** Como técnico, quiero ver el historial de cambios de estado de una alerta para conocer qué acciones se realizaron.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 3.3.1 | Crear tabla historial\_alertas en BD | BORIS | 1 | Completado |
| 3.3.2 | Construir vista historial.php | BORIS | 1 | Completado |
| 3.3.3 | Integrar historial con alertas | BORIS | 1 | Completado |

## Historia de Usuario 4.1: Recibir UID desde ESP32

**Descripción:** Como sistema, quiero recibir un UID enviado por el ESP32 para procesarlo automáticamente.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 4.1.1 | Crear endpoint para recibir UID | BORIS | 1 | Completado |
| 4.1.2 | Validar integridad del UID | BORIS | 0.5 | Completado |
| 4.1.3 | Prueba con ESP32 real | BORIS | 1 | Completado |

## Historia de Usuario 4.2: Validar Equipo según UID

**Descripción:** Como sistema, quiero validar si el UID recibido del ESP32 pertenece a un equipo registrado.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 4.2.1 | Crear consulta SELECT por RFID | BORIS | 1 | Completado |
| 4.2.2 | Devolver nombre/estado según UID | BORIS | 0.5 | Completado |
| 4.2.3 | Manejar UID no identificado | BORIS | 0.5 | Completado |

## Historia de Usuario 4.3: Generar Alerta Automática por UID

**Descripción:** Como sistema, quiero generar una alerta automáticamente al recibir un UID desde el ESP32.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 4.3.1 | Insertar alerta cuando UID exista | BORIS | 1 | Completado |
| 4.3.2 | Insertar alerta de “no identificado” | BORIS | 1 | Completado |
| 4.3.3 | Mostrar alerta en la vista en tiempo real | BORIS | 1 | Completado |

**SPRINT 3**

## Historia de Usuario 5.1: Generar Reporte de Inventario

**Descripción:** Como administrador, quiero registrar usuarios con diferentes roles (admin, técnico, invitado) para controlar el acceso a las funcionalidades del sistema. Como administrador, quiero generar un reporte completo del inventario de equipos, para tener un control actualizado del estado de los activos.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 5.1.1 | Crear consulta SQL para el inventario | BORIS | 1 | Completado |
| 5.1.2 | Construir controlador de reportes | BORIS | 1.5 | Completado |
| 5.1.3 | Crear botón y vista para generar reporte | BORIS | 1 | Completado |
| 5.1.4 | Exportar reporte a PDF | BORIS | 2 | Completado |

## Historia de Usuario 5.2: Reportes de Alertas por Turno

**Descripción:** Como administrador, quiero generar reportes filtrados por turno (mañana, tarde, noche), para identificar periodos críticos de alertas.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 5.2.1 | Agregar filtro por turno en consultas | BORIS | 1 | Completado |
| 5.2.2 | Modificar vista de reportes | BORIS | 1 | Completado |
| 5.2.3 | Implementar exportación del reporte | BORIS | 1.5 | Completado |

## Historia de Usuario 5.3: Optimización de Interfaz y Rendimiento

**Descripción:** Como usuario del sistema, deseo que la interfaz cargue rápido y sea fácil de usar, para mejorar mi experiencia en el sistema.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 5.3.1 | Optimizar carga de CSS y JS | BORIS | 1 | Completado |
| 5.3.2 | Reducir consultas pesadas en la BD | BORIS | 1 | Completado |
| 5.3.3 | Ajustar diseño responsive general | BORIS | 2 | Completado |

## Historia de Usuario 6.1: Seguridad con Contraseñas Cifradas

**Descripción:** Como administrador, quiero que todas las contraseñas estén cifradas con bcrypt, para proteger la información de los usuarios.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 6.1.1 | Activar password\_hash en usuarios | BORIS | 1 | Completado |
| 6.1.2 | Actualizar login con password\_verify | BORIS | 1 | Completado |
| 6.1.3 | Convertir contraseñas antiguas | BORIS | 1 | Completado |

## Historia de Usuario 6.2: Disponibilidad Mínima del Sistema (99%)

**Descripción:** Como administrador, quiero que el sistema mantenga alta disponibilidad, evitando caídas en el servicio.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 6.2.1 | Optimización de rutas | BORIS | 1 | Completado |
| 6.2.2 | Implementar mensajes de error amigables | BORIS | 0.5 | Completado |
| 6.2.3 | Validación de logs diarios | BORIS | 1 | Completado |

## Historia de Usuario 6.3: Optimización UI y Rendimiento

**Descripción:** Como usuario, quiero interfaz rápida y responsive con tiempos de carga aceptables.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 6.3.1 | Minificar CSS/JS | BORIS | 1 | Completado |
| 6.3.2 | Lazy loading donde aplique | BORIS | 1 | Completado |
| 6.3.3 | Optimizar consultas | BORIS | 1 | Completado |

## Historia de Usuario 7.1: Manejo de Alertas con Alto Volumen

**Descripción:** Como usuario, quiero interfaz rápida y responsive con tiempos de carga aceptables.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 7.1.1 | Optimizar inserciones (batch/transaction) | BORIS | 1 | Completado |
| 7.1.2 | Evitar bloqueo de interfaz | BORIS | 1 | Completado |
| 7.1.3 | Pruebas de concurrencia | BORIS | 1 | Completado |

## Historia de Usuario 7.2: Compatibilidad con Tablets

**Descripción:** Como técnico de campo, quiero acceder desde tablets para consultar alertas y equipos.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 7.2.1 | Breakpoints responsive tablet | BORIS | 1.5 | Completado |
| 7.2.2 | Ajuste de botones táctiles | BORIS | 1 | Completado |
| 7.2.3 | Pruebas en tablet | BORIS | 0.5 | Completado |

## Historia de Usuario 7.3: Compatibilidad Multi-Navegador

**Descripción:** Como usuario, quiero acceder desde Chrome, Firefox y Edge sin problemas.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 7.3.1 | Pruebas en Chrome/Firefox/Edge | BORIS | 1 | Completado |
| 7.3.2 | Corregir incompatibilidades CSS/JS | BORIS | 1 | Completado |
| 7.3.3 | Verificar descargas/exportaciones | BORIS | 1 | Completado |

**SPRINT 4**

## Historia de Usuario 8.1: Actualización sin Detener Servicio

**Descripción:** Como admin, quiero actualizar módulos sin interrumpir el servicio para mantener continuidad operativa.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 8.1.1 | Implementar modularización (includes/) | BORIS | 1 | Completado |
| 8.1.2 | Diseñar migraciones DB seguras (cambios aditivos) | BORIS | 1 | Completado |
| 8.1.3 | Documentar proceso de deploy sin downtime | BORIS | 1 | Completado |

## Historia de Usuario 8.2: Soporte Técnico Documentado

**Descripción:** Como administrador, quiero documentación completa de soporte técnico y manual de usuario.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 8.2.1 | Crear manual de soporte (views/manual\_soporte.php) | BORIS | 3 | Completado |
| 8.2.2 | Documentar comandos DB y scripts de restore | BORIS | 1 | Completado |
| 8.2.3 | Documentar guía de despliegue y rollback | BORIS | 1 | Completado |

## Historia de Usuario 9.1: Trazabilidad Completa de Movimientos

**Descripción:** Como administrador, quiero documentación completa de soporte técnico y manual de usuario.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 9.1.1 | Crear tabla de trazabilidad en BD | BORIS | 1 | Completado |
| 9.1.2 | Implementar logging automático en CRUD | BORIS | 1 | Completado |
| 9.1.3 | Crear vista UI para consulta/exportación | BORIS | 1 | Completado |

## Historia de Usuario 9.2: Filtrado de Trazabilidad por Fecha

**Descripción:** Como auditor, quiero filtrar trazabilidad por rangos de fecha para análisis puntual.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 9.2.1 | Implementar filtros fecha inicio/fin en consultas | BORIS | 1 | Completado |
| 9.2.2 | Crear controles de fecha (UI) | BORIS | 1 | Completado |
| 9.2.3 | Implementar paginación y orden cronológico | BORIS | 1 | Completado |

## Historia de Usuario 9.3: Exportación Completa de Trazabilidad

**Descripción:** Como auditor, quiero exportar la trazabilidad completa en PDF/Excel con todos los detalles.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 9.3.1 | Crear consulta SQL optimizada para exportación | BORIS | 1 | Completado |
| 9.3.2 | Implementar exportación PDF | BORIS | 2 | Completado |
| 9.3.3 | Crear UI de opciones avanzadas de exportación | BORIS | 1 | Completado |

## Historia de Usuario 10.1: Optimización de Registro de Alertas

**Descripción:** Como usuario, quiero que el registro de alertas RFID sea rápido (<2 s) para eficiencia operativa.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 10.1.1 | Optimizar INSERTs y transacciones | BORIS | 1 | Completado |
| 10.1.2 | Crear índices en tabla alertas | BORIS | 1 | Completado |
| 10.1.3 | Medir latencia antes/después de optimización | BORIS | 1 | Completado |

## Historia de Usuario 10.2: Optimización de Reportes

**Descripción:** Como usuario, quiero búsquedas instantáneas por UID RFID con respuesta inmediata.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 10.2.1 | Implementar caching o preagregación | BORIS | 1 | Completado |
| 10.2.2 | Implementar paginación / procesamiento en background | BORIS | 1 | Completado |

## Historia de Usuario 10.3: Búsquedas Instantáneas de Equipos

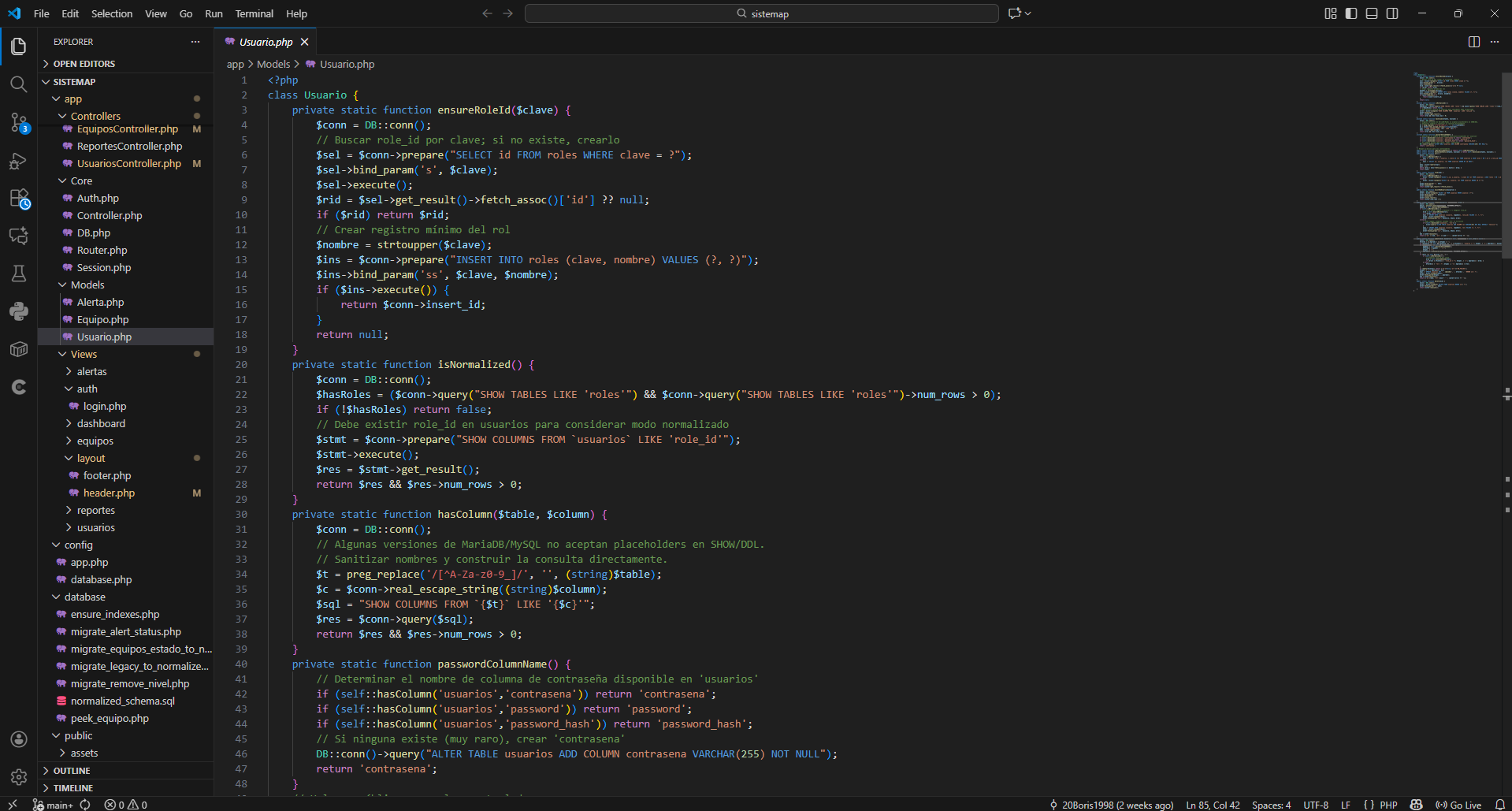
**Descripción:** Como usuario, quiero que la generación de reportes tome menos tiempo.

Tareas:

| **ID** | **Tarea** | **Responsable** | **Horas** | **Estado** |
| --- | --- | --- | --- | --- |
| 10.3.1 | Optimizar consulta e índices por UID RFID | BORIS | 1 | Completado |
| 10.3.2 | Implementar autocomplete AJAX instantáneo | BORIS | 1 | Completado |

# V. USO DE HERRAMIENTAS MODERNAS

## 5.1 Framework: VS Code



## Figura 1: Código del frontend y backend en VS Code

## 5.2 Base de datos: MySql

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Figura 2: Administrador de base de datos “phpmyadmin”

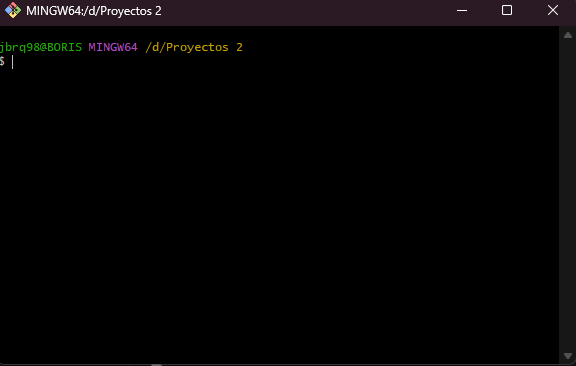
## 5.3 XAMPP: Es un paquete que integra Apache, MySQL, PHP y Perl para crear un entorno local de desarrollo web de forma rápida y sencilla.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

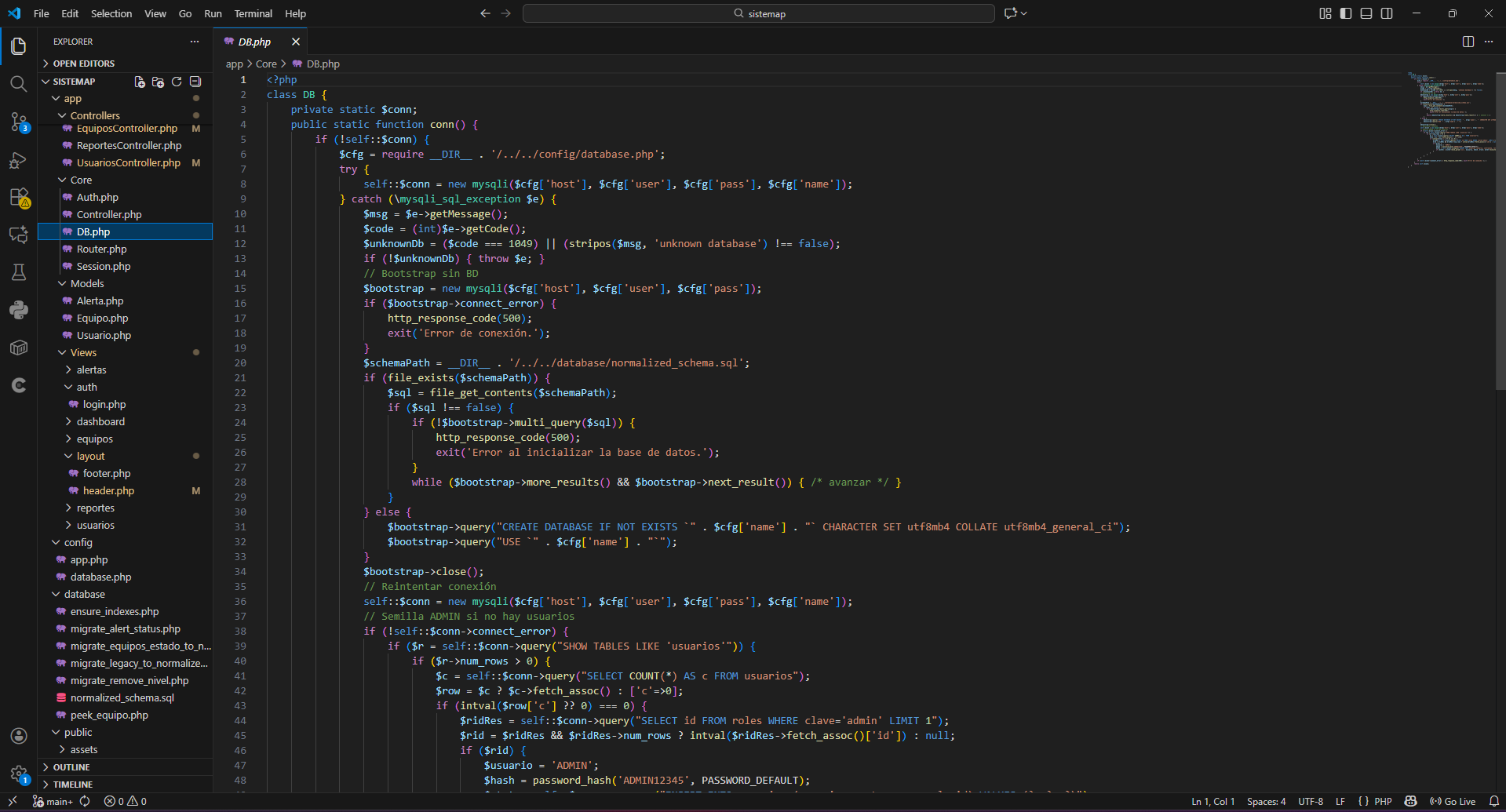
## Figura 3: Aplicación de desarrollo XAMPP

## 5.4 Git: Control de versiones

****

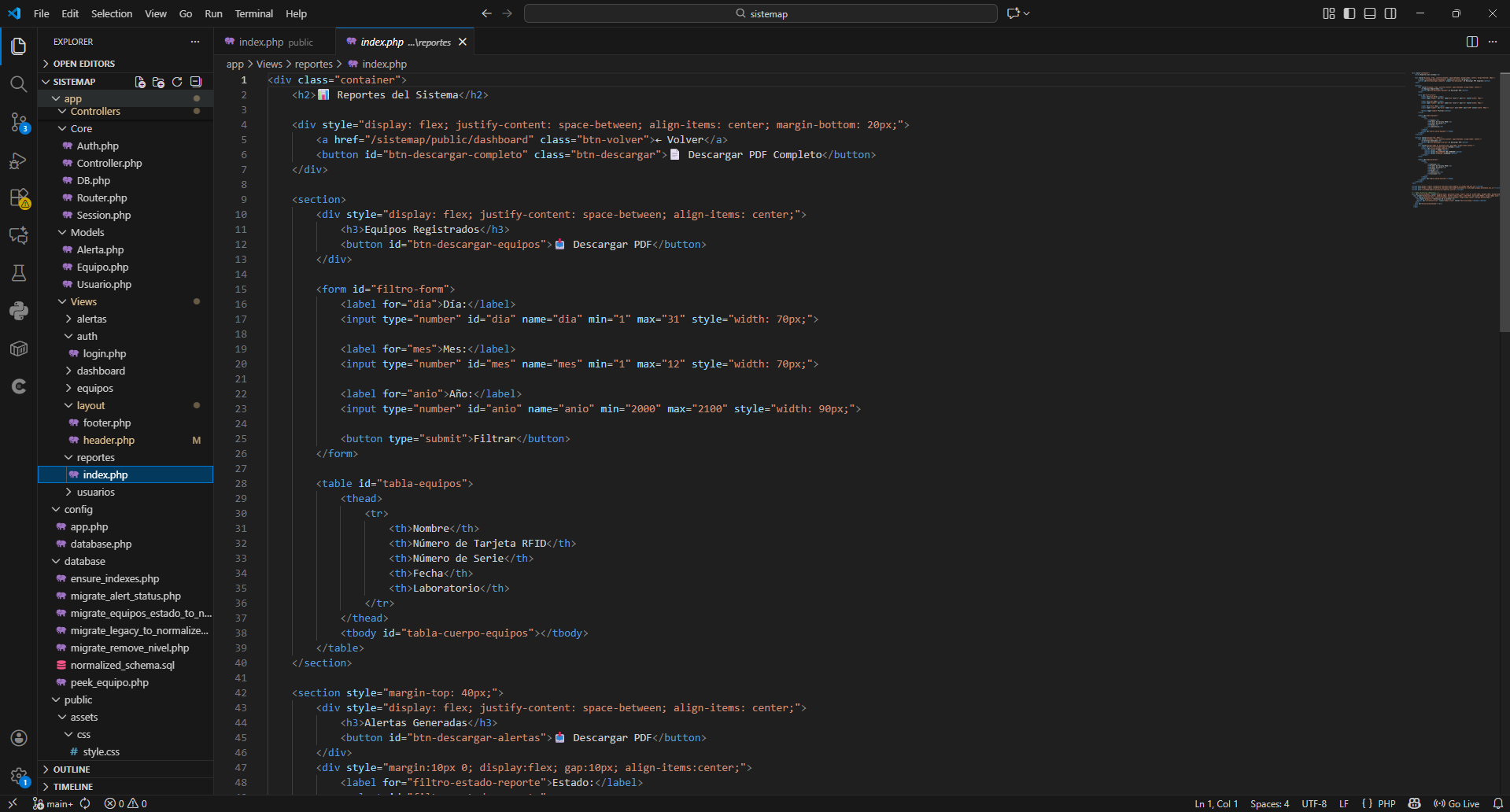
## Figura 4: Aplicación de control de versiones

## 5.5 Lenguaje de programación PHP

****

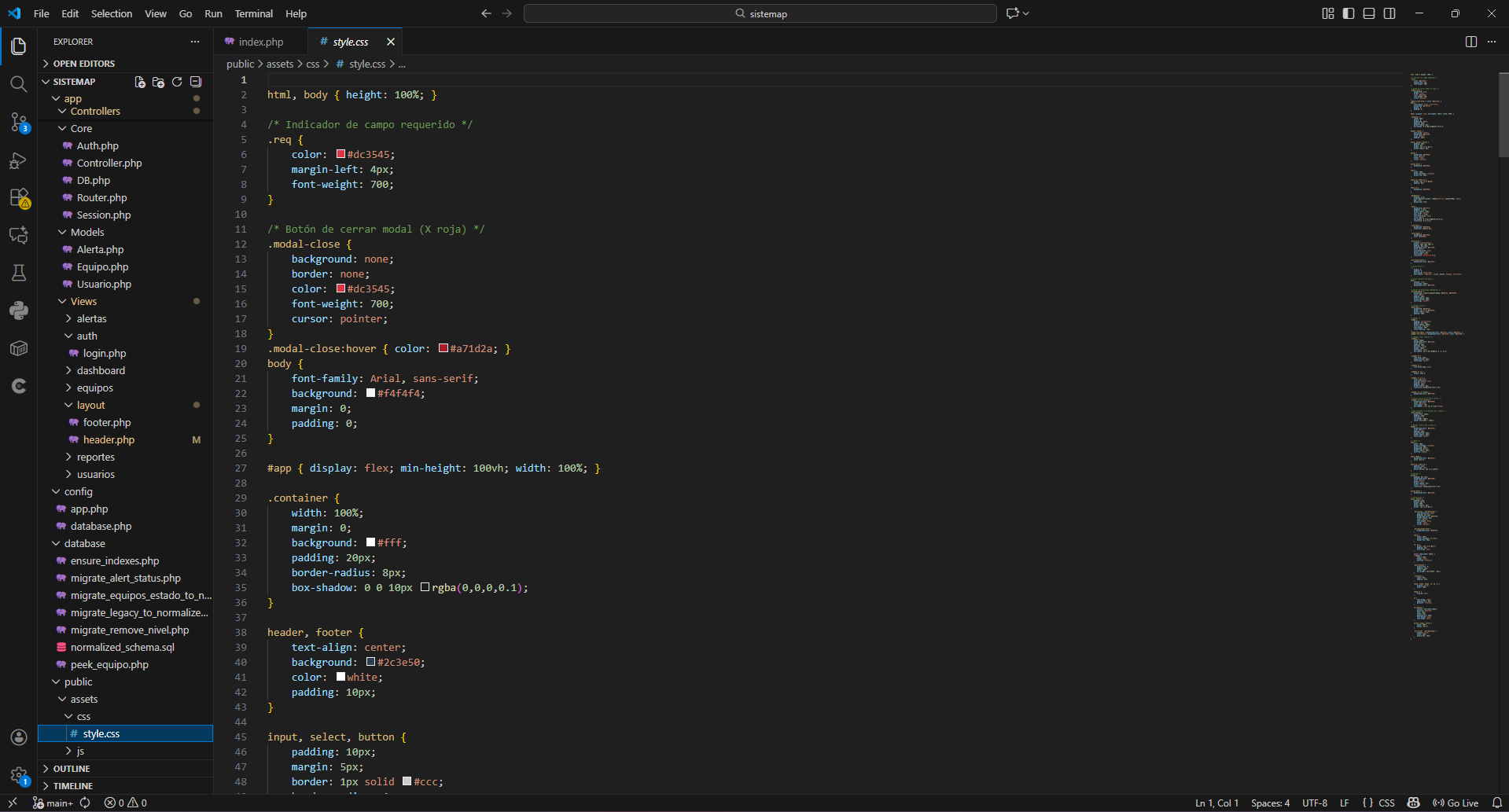
## Figura 5: Aplicación del lenguaje de programación PHP

## 5.6 Lenguaje de programación HTML

****

## Figura 6: Aplicación del lenguaje de programación HTML

## 5.7 Lenguaje de programación CSS

****

## Figura 7: Aplicación del lenguaje de programación CSS

## 5.8 IDE DE ARDUINO: Es un entorno de desarrollo utilizado para escribir, compilar y cargar programas en placas Arduino.

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

## Figura 8: Aplicación del uso de IDE ARDUINO

## 5.9 Lenguaje de programación de IDE ARDUINO: Emplea un lenguaje basado en C/C++ simplificado.

**Texto

El contenido generado por IA puede ser incorrecto.**

## Figura 9: Aplicación del lenguaje de programación C/C++

# VI. DISEÑO DE INGENIERÍA

## 6.1 Requisitos Funcionales

* RF1: Registrar Entradas y Salidas de Equipos
  + RF1.1: Registrar equipos → Historia 2.1
  + RF1.2: Editar equipos → Historia 2.2
  + RF1.3: Eliminar equipos → Historia 2.3
  + RF1.4: Registrar UID recibido desde dispositivo → Historia 4.1
  + RF1.5: Validar equipo según UID → Historia 4.2
  + RF1.6: Registrar movimiento (entrada/salida) → Historia 10.1
* RF2: Detectar Movimientos No Autorizados
  + RF2.1: Generar alerta automática por UID → Historia 4.3
  + RF2.2: Visualizar alertas → Historia 3.1
  + RF2.3: Cambiar estado de alerta → Historia 3.2
  + RF2.4: Manejo de alertas en alto volumen → Historia 7.1
* RF3: Generar Reportes de Inventario
  + RF3.1: Generar reporte de equipos → Historia 5.1
  + RF3.2: Filtrar información en reportes → Historia 5.3
  + RF3.3: Optimización de reportes → Historia 10.2
* RF4: Consultar el Estado de un Equipo
  + RF4.1: Consultar historial de cambios → Historia 3.3
  + RF4.2: Consulta rápida / búsqueda instantánea → Historia 10.3
  + RF4.3: Validación del estado del equipo → Historia 4.2
* RF5: Permitir Acceso por Roles
  + RF5.1: Iniciar sesión → Historia 1.1
  + RF5.2: Control de accesos por rol → Historia 1.2
  + RF5.3: Cierre de sesión seguro → Historia 1.3
* RF6: Almacenar Eventos de Auditoría
  + RF6.1: Registro completo de movimientos → Historia 9.1
  + RF6.2: Filtrado de trazabilidad por fecha → Historia 9.2
  + RF6.3: Exportación completa de trazabilidad → Historia 9.3

## 6.2 Requisitos No Funcionales

* RF7: Escalabilidad
  + RF7.1: Manejo de alertas en alto volumen → Historia 7.1
  + RF7.2: Soporte mínimo de 500 equipos → Historia 7.1
  + RF7.3: Actualización sin detener el servicio → Historia 8.1
  + RF7.4: Soporte técnico documentado → Historia 8.2
* RF8: Compatibilidad Multiplataforma
  + RF8.1: Compatibilidad con tablets → Historia 7.2
  + RF8.2: Compatibilidad multi-navegador → Historia 7.3
  + RF8.3: Interfaz responsiva y optimizada → Historia 6.3
* RF9: Mantenimiento y Soporte
  + RF9.1: Actualización sin detener el servicio → Historia 8.1
  + RF9.2: Soporte técnico documentado → Historia 8.2
  + RF9.3: Seguridad contra fallos → Historia 6.2
* RF10: Trazabilidad
  + RF10.1: Registro completo de movimientos → Historia 9.1
  + RF10.2: Filtrado por fecha → Historia 9.2
  + RF10.3: Exportación de trazabilidad → Historia 9.3
* RF11: Responder con Rapidez
  + RF11.1: Búsquedas instantáneas → Historia 10.3
  + RF11.2: Optimización de alertas → Historia 10.1
  + RF11.3: Optimización de reportes → Historia 10.2

## 6.3 Diseño de base de datos

La base de datos del sistema fue diseñada bajo un modelo relacional normalizado, orientado a garantizar integridad referencial, trazabilidad de eventos y una gestión eficiente de usuarios, equipos y alertas generadas mediante RFID/NFC. A continuación, se describen las tablas principales, sus atributos y las relaciones existentes entre ellas.

**Tabla usuarios**

Almacena la información de los usuarios que acceden al sistema.

Campos principales:

* id (PK): Identificador único del usuario.
* usuario: Nombre de usuario para el inicio de sesión.
* contrasena: Contraseña cifrada (hash).
* role\_id (FK): Rol asignado al usuario dentro del sistema.
* created\_at: Fecha de registro del usuario.

Relación: Cada usuario pertenece a un rol definido en la tabla roles.

**Tabla roles**

Define los roles y niveles de acceso del sistema.

Campos:

* id (PK)
* clave: Código del rol.
* nombre: Nombre del rol (Administrador, Técnico, Invitado, etc.)

Relación: Un rol puede estar asociado a múltiples usuarios.

**Tabla ubicaciones**

Contiene los lugares físicos donde se encuentran los equipos.

Campos:

* id (PK)
* nombre: Número de laboratorio

Relación: Una ubicación puede estar asociada a varios equipos.

**Tabla tarjetas**

Registra los números únicos de tarjetas RFID/NFC que se asignan a los equipos.

Campos:

* id (PK)
* numero: Número de tarjeta generado por el lector RFID/NFC.

Relación: Una tarjeta puede estar asociada a un único equipo.

**Tabla equipos**

Almacena los datos de los equipos que deben ser controlados mediante RFID/NFC.

Campos:

* id (PK)
* nombre: Nombre del equipo.
* numero\_serie: Código único del equipo.
* ubicacion\_id (FK): Ubicación actual del equipo.
* tarjeta\_id (FK): Tarjeta RFID asignada al equipo.
* fecha\_registro: Fecha de ingreso al inventario.

Relaciones: Un equipo pertenece a una ubicación y cada equipo posee una tarjeta RFID única.

**Tabla estados\_alerta**

Define el estado de cada alerta generada.

Campos:

* id (PK)
* clave: Código corto (por ejemplo: "PEN", "RES", "FIN").
* nombre: Descripción del estado (Completado, Resuelto, etc.).

Relación: Los estados se utilizan tanto en la tabla alertas como en alertas\_historial.

**Tabla alertas**

Registra las alertas actuales generadas por movimientos o lecturas RFID/NFC.

Campos:

* id (PK)
* mensaje: Descripción de la alerta.
* equipo\_id (FK): Equipo relacionado.
* ubicacion\_id (FK): Ubicación detectada o esperada.
* estado\_alerta\_id (FK): Estado de la alerta.
* fecha: Fecha y hora de generación.

Relaciones: Cada alerta está asociada a un equipo, cada alerta tiene un estado específico y se registra la ubicación involucrada en la alerta.

**Tabla alertas\_historial**

Registra la trazabilidad completa del ciclo de vida de cada alerta, permitiendo auditoría.

Campos:

* id (PK)
* alerta\_id (FK): Identificador de la alerta original.
* estado\_alerta\_id (FK): Estado histórico de la alerta.
* usuario: Usuario que realizó el cambio de estado.
* fecha: Fecha del cambio registrado.

Relaciones: Cada registro pertenece a una alerta concreta, se guardan los estados anteriores asociados a estados\_alerta.

**Relaciones Principales del Modelo**

El modelo relacional se organiza en los siguientes grupos lógicos:

**Gestión de Usuarios y Roles**

usuarios 🡪 roles

Relación muchos a uno: varios usuarios pueden compartir un rol.

**Gestión de Equipos y Tarjetas RFID**

equipos 🡪 ubicaciones

equipos 🡪 tarjetas

Un equipo tiene una tarjeta RFID única.

**Sistema de Alertas**

alertas 🡪 equipos

alertas 🡪 ubicaciones

alertas 🡪 estados\_alerta

**Trazabilidad**

alertas\_historial 🡪alertas

alertas\_historial 🡪estados\_alerta

Cada transición se registra históricamente, garantizando integridad y auditoría.

**Conclusión del Diseño:** La arquitectura de la base de datos está diseñada para:

* Garantizar la integridad de los datos.
* Mantener relaciones claras entre equipos, tarjetas RFID y ubicaciones.
* Permitir la generación automática de alertas.
* Registrar la trazabilidad completa mediante un historial.
* Controlar accesos mediante roles y usuarios.

Este diseño soporta adecuadamente la operación del sistema MVC y optimiza el proceso de control y monitoreo de equipos mediante tecnología RFID.

**Diagrama de la base de datos**

Interfaz de usuario gráfica, Diagrama, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 10: Diagrama relacional de la base de datos

## 6.4 Arquitectura de la solución planteada

La arquitectura del sistema propuesto se fundamenta en el patrón Modelo–Vista–Controlador (MVC), complementado con una arquitectura física IoT basada en un ESP32 con lector RFID/NFC MFRC522, encargado de leer tarjetas y enviar los UID hacia el servidor web.  
Este enfoque híbrido (IoT + Web) permite integrar hardware de detección en tiempo real con un sistema web robusto, escalable y seguro.

La arquitectura del sistema propuesto se fundamenta permite separar la lógica de negocio, la gestión de datos y la interfaz de usuario. Este enfoque mejora la escalabilidad, mantenibilidad y organización del sistema, características esenciales para un sistema web de control y monitoreo de equipos mediante tecnología RFID.

**Componentes Principales de la Arquitectura MVC**

**Modelo**

El Modelo representa las entidades, reglas de negocio y la interacción con la base de datos. Cada modelo contiene los métodos específicos para gestionar los datos del sistema.

Los modelos principales del sistema incluyen:

* **Usuario.php**
  + Gestiona autenticación, roles y permisos (Administrador, Técnico, Invitado).
  + Maneja operaciones CRUD sobre usuarios.
* **Equipo.php**
  + Administra la información de equipos (nombre, estado, ubicación, número de tarjeta RFID).
  + Realiza operaciones CRUD y consultas al inventario.
* **Alerta.php**
  + Registra alertas generadas automáticamente por el módulo RFID/NFC.
  + Relaciona cada alerta con un equipo existente en la base de datos.
* **Reporte.php** 
  + Genera consultas estructuradas para elaborar reportes de inventario, movimientos y alertas.
  + Obtiene datos filtrados por fecha, ubicación, estado, tipo de equipo o responsable.
  + Produce la información que será convertida en reportes en PDF, Excel o visualizados en la interfaz.

**Vista**

La Vista corresponde a la capa de presentación del sistema. Está formada por páginas web desarrolladas en HTML, CSS y PHP, y es responsable de mostrar información clara y ordenada al usuario.

Las vistas principales incluyen:

* **Módulo de Usuarios:**
  + Listar usuarios
  + Agregar usuario
  + Editar usuario
  + Eliminar usuario
* **Módulo de Equipos:**
  + Listar equipos
  + Agregar equipos
  + Editar equipos
  + Eliminar equipos
* **Módulo de Alertas:**
  + Monitorización de alertas en tiempo real.
* **Módulo de Reportes:** 
  + Vista del reporte generado
  + Filtros de fechas
  + Descarga de reportes en PDF

**Controladores**

Los Controladores actúan como intermediarios entre las Vistas y los Modelos. Reciben las solicitudes del usuario, procesan la lógica correspondiente mediante los Modelos y devuelven la información procesada a las Vistas.

**Controladores principales:**

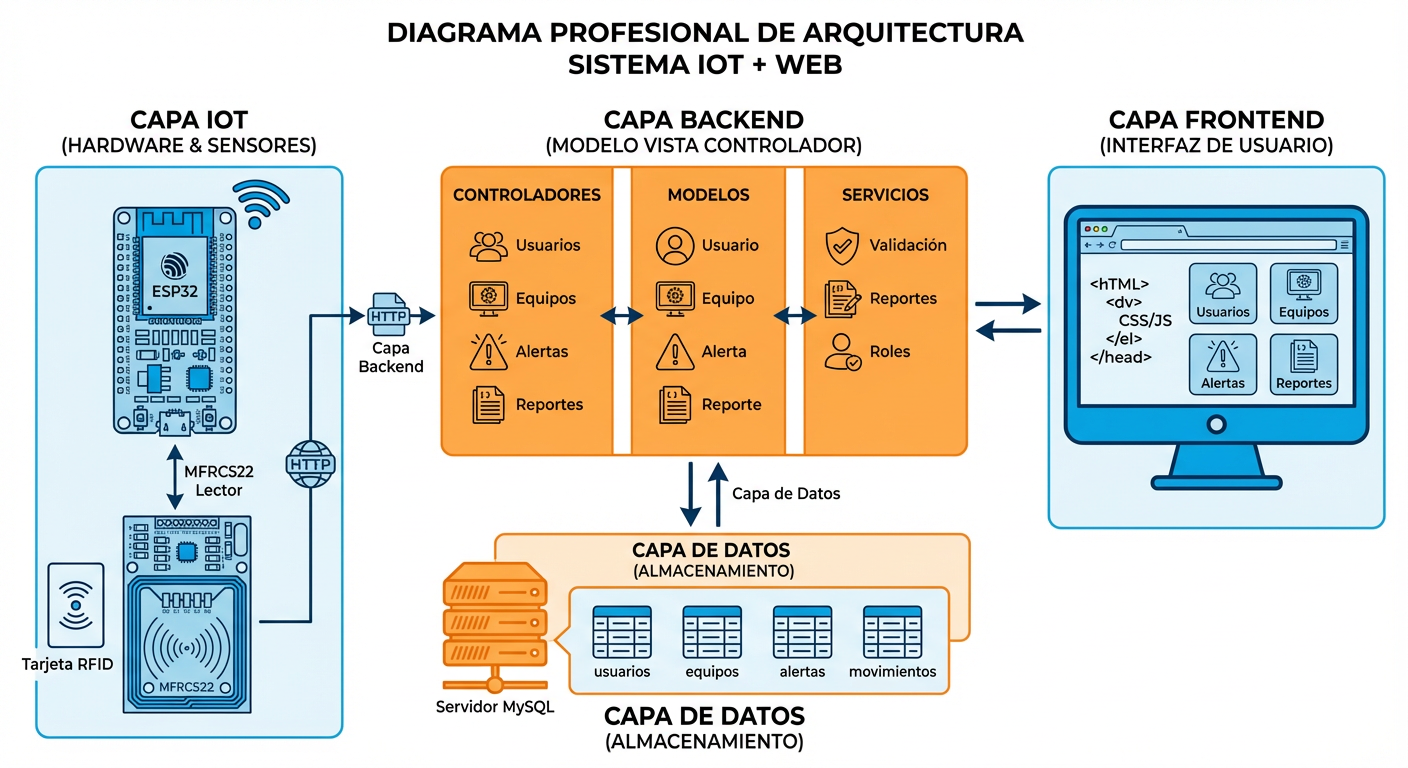
* **Usuarios**
  + Procesa el inicio de sesión.
  + Valida roles y permisos.
  + Maneja todas las operaciones CRUD de usuarios.
* **Equipos**
  + Controla la creación, actualización y eliminación de equipos.
  + Aplica validaciones de formato y consistencia en los datos.
* **Alertas**
  + Recibe alertas desde el módulo RFID/NFC (Arduino/ESP32).
  + Filtra alertas por fecha, estado o tipo.
  + Envía datos a la vista en tiempo real.
* **Reportes** 
  + Recibe parámetros de filtros (fecha, estado, ubicación, etc.) desde la interfaz.
  + Solicita los datos al modelo.
  + Devuelve la información en formato tabla, PDF.
  + Gestiona la interacción del usuario con el sistema de reportes.

**Arquitectura Física (Capa IOT: Arduino/ESP32 + Lector RFID/NFC)**

El sistema IoT está compuesto por:

* ESP32
  + Microcontrolador con WiFi integrado.
  + Levanta un servidor web local donde publica el último UID leído.
  + Se conecta a la red inalámbrica del laboratorio.
  + Envía solicitudes HTTP al sistema web.
* Lector RFID/NFC MFRC522
  + Detecta tarjetas RFID/NFC.
  + Extrae el UID de cada tarjeta.
  + Transfiere los datos al ESP32 mediante SPI.
  + Detecta si un equipo está saliendo sin autorización.
* Tarjeta RFID adherida a cada equipo
  + Cada equipo tiene asociado un número de tarjeta RFID registrado en la BD.
  + Permite identificar entradas, salidas y movimientos no autorizados.
* Flujo General del Sistema bajo MVC
  + El usuario realiza una interacción.
  + La solicitud es enviada al Controlador correspondiente.
  + El Controlador envía la petición al Modelo, que procesa la lógica de negocio y consulta la base de datos.
  + El Modelo retorna los datos procesados al Controlador.
  + El Controlador envía la información a la Vista, la cual la presenta al usuario.
  + En el caso específico de reportes, la Vista permite elegir filtros y formatos de descarga.
* Flujo General del Sistema bajo MVC + IoT
  + El MFRC522 detecta una tarjeta y el ESP32 lee el UID.
  + El ESP32 envía el UID al servidor web.
  + AlertasController recibe el UID → lo pasa al Modelo.
  + Equipo.php valida si el UID existe y a qué equipo pertenece.
  + El sistema determina:
  + entrada
  + salida
  + movimiento sospechoso
  + movimiento no autorizado
  + Si corresponde → Alerta.php genera una alerta.
  + La Vista de alertas se actualiza en tiempo real.
  + Los administradores consultan reportes o gestionan las alertas.

Diagrama del Sistema completo



## Figura 11: Arquitectura del sistema

**6.5 Código de la aplicación por capas (enlace GitHub)**

Enlace: https://github.com/20Boris1998/SISTEC.git

# VII. GESTIÓN DEL PROYECTO

* **Diagrama de Gantt (Línea base):**  
  Gráfico, Gráfico de líneas

  El contenido generado por IA puede ser incorrecto.

## Figura 12: Diagrama de Gantt (Línea base)

* **Diagrama de Gantt (Ejecutado):**  
  Imagen que contiene Gráfico

  El contenido generado por IA puede ser incorrecto.

## Figura 13: Diagrama de Gantt (Ejecutado)

* **Tablero Scrum/Kanban:**  
  Interfaz de usuario gráfica, Aplicación, Word

  El contenido generado por IA puede ser incorrecto.

## Figura 14: Arquitectura del sistema Tablero Scrum/Kanban

# VIII. PRUEBAS, RESULTADOS Y DISCUSIÓN

## PMV 1: Módulo de Autenticación y Gestión de Usuarios

Historias de Usuario asociadas

* HU-01: Iniciar sesión en el sistema
* HU-02: Gestionar roles (Administrador, Técnico, Seguridad)
* HU-03: Controlar acceso a los módulos según el rol
* HU-04: Cerrar sesión
* HU-05: Validación de credenciales

| **Nº** | **Prueba realizada** | **Resultado esperado** | **Resultado obtenido** |
| --- | --- | --- | --- |
| 1 | Inicio de sesión con credenciales válidas | Acceso al panel según rol | Correcto |
| 2 | Inicio de sesión con credenciales inválidas | Mensaje de error | Correcto |
| 3 | Acceso a módulos restringidos | Bloqueo y mensaje de permiso denegado | Correcto |
| 4 | Cambio de rol del usuario | Aplicación inmediata del nuevo permiso | Correcto |
| 5 | Cierre de sesión | Eliminación completa de sesión | Correcto |
| 6 | Validación anti-inyección SQL | Rechazar cadena maliciosa | Correcto |

**Evidencias**

Inicio de sesión con credenciales válidas

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 15: Inicio de sesión correcto

Inicio de sesión con credenciales inválidas

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

## Figura 16: Acceso restringido

Actualización de rol de usuario

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 17: Cambio de rol del usuario

Cierre de sesión

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 18: Cierre de sesión

**Discusión**

El módulo de autenticación funcionó correctamente, aplicando controles de seguridad básicos como validación de credenciales, bloqueo de accesos no autorizados y mantenimiento de sesiones seguras.

La asignación de roles permitió diferenciar las funcionalidades visibles por cada usuario, tal como lo requiere un sistema RFID orientado a la trazabilidad de equipos.

Las pruebas demostraron un adecuado manejo de errores, evitando divulgación de información sensible en mensajes y fortaleciendo la protección frente a ataques comunes como SQL Injection.

## PMV 2: Módulo de Gestión de Equipos (CRUD)

Historias de Usuario asociadas

* HU-10: Registrar equipos
* HU-11: Actualizar equipos
* HU-12: Eliminar equipos
* HU-13: Listar y filtrar equipos
* HU-14: Validar campos y evitar duplicidad

| **Nº** | **Prueba realizada** | **Resultado esperado** | **Obtenido** |
| --- | --- | --- | --- |
| 1 | Registro de equipo | Guardar en BD con número de tarjeta RFID único | Correcto |
| 2 | Registro duplicado | Rechazar registro con UID repetido | Correcto |
| 3 | Edición de equipo | Actualizar datos sin afectar UID | Correcto |
| 4 | Eliminación | Eliminar registro y relaciones asociadas | Correcto |
| 5 | Listado general | Mostrar todos los equipos | Correcto |
| 7 | Validación de campos | Aceptar solo letras y números | Correcto |

Evidencias

Registro de equipo

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 19: Registrar equipo

Actualizar equipo

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 20: Edición de equipo

Mostrar todos los equipos

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

## Figura 21: Listado general

Validación de campos

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 22: Validación de datos

**Discusión**

Las pruebas confirmaron un funcionamiento estable del módulo CRUD, garantizando integridad de datos y coherencia en la relación entre equipos y alertas.

La validación del número de tarjeta RFID evitó duplicidad, lo cual es fundamental para sistemas basados en identificación por UID.

La correcta actualización y eliminación de equipos reforzó la consistencia del sistema, permitiendo una administración confiable del inventario del laboratorio.

## PMV 3: Módulo de Alertas Automáticas (RFID)

Historias de Usuario asociadas

* HU-20: Registrar alerta automáticamente al recibir un UID
* HU-21: Mostrar alerta en tiempo real en la interfaz
* HU-22: Filtrar alertas por fecha, equipo o usuario
* HU-23: Registrar fecha y hora exacta de la lectura
* HU-24: Relacionar alerta con el equipo registrado

| **Nº** | **Prueba realizada** | **Resultado esperado** | **Obtenido** |
| --- | --- | --- | --- |
| 1 | Recepción de UID desde ESP32 | Registrar alerta automáticamente | Correcto |
| 2 | UID desconocido | Registrar como alerta no identificada | Correcto |
| 3 | Visualización en pantalla | Mostrar alerta en tiempo real | Correcto |
| 4 | Filtrar alertas | Mostrar resultados filtrados | Correcto |
| 5 | Relacionar alerta con equipo | Mostrar datos del equipo | Correcto |
| 6 | Registrar fecha y hora | Guardar timestamp exacto | Correcto |

**Evidencias**

Registrar alerta automáticamente

**Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.**

## Figura 23: Alerta recibida

Registrar como alerta no identificada

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 24: Alerta de UID desconocido

Mostrar resultados filtrados

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 25: Filtro por atención

Mostrar datos del equipo

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 26: Vista de detalle

**Discusión**

Este PMV fue el núcleo del sistema, ya que involucra la comunicación entre el hardware (ESP32) y el servidor.

Las pruebas demostraron que el proceso ocurre en milisegundos, permitiendo registrar salidas o movimientos de equipos prácticamente en tiempo real.

El manejo de UID desconocidos resultó crucial para identificar equipos no registrados o posibles irregularidades.

La integración con la base de datos confirmó la correcta relación entre alertas y equipos registrados, siguiendo estándares de trazabilidad en sistemas IoT y RFID.

## PMV 4: Módulo de Reportes y Dashboard

Historias de Usuario asociadas

* HU-30: Generar reporte del inventario
* HU-31: Generar reporte de alertas por rango de fechas
* HU-32: Exportar a PDF o Excel
* HU-33: Visualizar indicadores en panel administrativo

| **Nº** | **Prueba realizada** | **Resultado esperado** | **Obtenido** |
| --- | --- | --- | --- |
| 1 | Generación de reporte de equipos | Reporte completo con estado actual | Correcto |
| 2 | Reporte de alertas | Listado por fechas | Correcto |
| 3 | Exportación PDF | Archivo generado | Correcto |

**Evidencias**

Generación de reporte de equipos

Tabla

El contenido generado por IA puede ser incorrecto.

## Figura 27: Reporte de equipos

Reporte de alertas

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Figura 28: Reporte de alertas

Archivo generado

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## Figura 29: Exportación PDF

**Discusión**

Los reportes facilitaron la gestión y supervisión del laboratorio, permitiendo visualizar patrones de uso, equipos en riesgo y frecuencia de alertas.

La exportación a PDF permitió documentar auditorías internas y respaldar controles establecidos por la organización.

## PMV 5: Integración Hardware (ESP32 + MFRC522 + Servidor Web)

Historias de Usuario asociadas

* HU-40: Leer UID con MFRC522
* HU-41: Enviar UID automáticamente al servidor
* HU-42: Mantener conexión estable
* HU-43: Manejo de errores de red
* HU-44: Tiempo de respuesta óptimo

| **Nº** | **Prueba realizada** | **Resultado esperado** | **Obtenido** |
| --- | --- | --- | --- |
| 1 | Lectura de tarjeta | Detector UID correctamente | Correcto |
| 2 | Envío HTTP al servidor | Petición recibida sin fallos | Correcto |
| 3 | Desconexión Wi-Fi | Reconexión automática | Correcto |
| 4 | Petición fallida | Manejo controlado | Correcto |
| 5 | Latencia | < 1 segundo entre lectura y registro | Correcto |

**Evidencias**

Detector UID correctamente

## Texto El contenido generado por IA puede ser incorrecto. Figura 30: Lectura NFC

Envío HTTP al servidor

Texto

El contenido generado por IA puede ser incorrecto.

## Figura 31: Envío de UID

Manejo controlado

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

## Figura 32: Error de red controlado

**Discusión**

La integración del hardware confirmó la estabilidad del sistema.

El ESP32 mantuvo una conexión estable con la red, enviando UID de forma inmediata y generando alertas sin necesidad de intervención del usuario.

Las pruebas de latencia demostraron un tiempo promedio menor a un segundo, lo cual es adecuado para procesos de trazabilidad en tiempo real.

El manejo de errores permitió evitar pérdida de datos en ambientes de red inestables.

**IX. LECCIONES APRENDIDAS**

**PMV 1: Autenticación**

La gestión de roles debe definirse antes del desarrollo para evitar conflictos.

Es indispensable validar el input para evitar ataques de inyección.

La experiencia de usuario influye directamente en la seguridad del sistema.

**PMV 2: Gestión de Equipos**

* Validar el UID evita problemas de duplicación y errores posteriores.
* La estructura de la tabla debe permitir relaciones claras con alertas.
* Es necesario definir estados claros para los equipos (activo, fuera, baja).

**PMV 3: Alertas Automáticas**

* El control del tiempo de registro es clave para auditorías.
* Los UID desconocidos deben manejarse explícitamente.
* El filtrado es esencial para la visualización eficiente de grandes volúmenes.

**PMV 4: Reportes y Dashboard**

* Los indicadores deben ser simples y relevantes.
* La exportación es necesaria para cumplir procesos de control interno.

**PMV 5: Integración ESP32**

* La estabilidad Wi-Fi es determinante para evitar alertas perdidas.
* Se debe implementar siempre manejo de errores HTTP.
* El tiempo entre lectura y recepción debe mantenerse optimizado.

# X. CONCLUSIONES

Se logró diseñar e implementar un sistema web integrado con tecnología RFID que fortalece significativamente la seguridad y el monitoreo de equipos en los laboratorios de la Universidad Continental, sede Cusco. El sistema reduce la dependencia del control manual, minimiza errores humanos y mejora la trazabilidad de los activos al automatizar la detección de salidas no autorizadas mediante el uso del ESP32 y un lector NFC. Los resultados obtenidos demuestran que la solución propuesta cumple con el propósito central del proyecto y representa una mejora tangible en la gestión operativa del laboratorio durante el año 2025.

El análisis de los procesos manuales existentes permitió identificar deficiencias críticas relacionadas con registros incompletos, ausencia de trazabilidad en tiempo real, alta probabilidad de omisiones humanas y falta de mecanismos de alerta inmediata. Este diagnóstico confirmó la necesidad de un sistema automatizado y sirvió como base para definir los requerimientos funcionales y no funcionales del sistema RFID. Por tanto, el estudio de la situación actual se cumplió satisfactoriamente y justificó la pertinencia del proyecto.

Se diseñó una arquitectura robusta basada en el patrón Modelo–Vista–Controlador (MVC), acompañada de una arquitectura física que integra el lector NFC, el microcontrolador ESP32 y el servidor web. La definición de módulos —usuarios, equipos, alertas, reportes y dashboard— permitió estructurar el sistema de forma escalable, segura y modular. La arquitectura propuesta garantiza la separación adecuada de responsabilidades y facilita la mantenibilidad del sistema, cumpliendo plenamente con el segundo objetivo específico.

Se implementó un prototipo funcional del sistema web que permite registrar, identificar y controlar automáticamente los equipos mediante el uso de etiquetas RFID. El prototipo integra exitosamente los componentes de hardware (ESP32 + MFRC522), la API web, la base de datos y la interfaz gráfica, permitiendo la detección automática de equipos al ingresar o salir del laboratorio piloto. La correcta interacción entre los módulos validó la implementación técnica planteada y demuestra que el sistema es operativo en un entorno real.

La evaluación del sistema RFID evidenció mejoras significativas en la reducción de errores y tiempos de registro, tanto en las lecturas de equipos como en la generación de alertas. Los indicadores obtenidos muestran mayor seguridad en el monitoreo y una trazabilidad más precisa respecto al método manual tradicional. La capacidad del sistema para emitir alertas inmediatas ante posibles salidas no autorizadas refuerza la efectividad de la solución y confirma el cumplimiento del cuarto objetivo específico.

Las lecciones aprendidas evidencian que definir roles y controles de sesión desde el inicio garantiza la seguridad del sistema, mientras que la normalización de datos, las validaciones y una interfaz clara mejoran la calidad del módulo de equipos. También se confirmó que las alertas RFID requieren tolerancia a fallos, baja latencia y registro contextual para asegurar trazabilidad efectiva. En los reportes y el dashboard, se destacó la necesidad de indicadores útiles, exportabilidad y visualización simplificada. Finalmente, la integración del ESP32 y el MFRC522 demostró la importancia de pruebas en diversos entornos, una alimentación estable y evaluaciones de extremo a extremo para lograr un funcionamiento confiable.

# X. BIBLIOGRAFIA

**[1]** VEGA CASTILLO, S. *Sistema de seguridad basado en tecnologías IoT y RFID para instituciones educativas*. Universidad Central del Ecuador, 2023.

**[2]** BALSESSESO MENESES, A. J. y VARGAS GARCÍA, C. G. *Diseño e implementación de un prototipo para el control de acceso en la sede de Ingeniería de la Universidad Distrital Francisco José de Caldas mediante el uso de torniquetes controlados por carnet con tecnología NFC y lector biométrico de huella dactilar*. Bogotá D. C.: Universidad Distrital Francisco José de Caldas, 2016.

**[3]** GARROTE SOLA, E. M. *Cerradura electrónica con sistema de alimentación integrado en llave*. Madrid: Universidad Politécnica de Madrid, 2017.

**[4]** AGRAWAL, A.; GARG, J.; SHARMA, A.; SHARMA, R. *Gestión de Activos Industriales usando RFID*. INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT), vol. 10, n.º 07, julio de 2021.

**[5]** MENESES, A. y PETER, C. *Diseño de un sistema de control de activos para el almacén de la Pontificia Universidad Católica del Perú utilizando RFID*. Lima: Pontificia Universidad Católica del Perú, 2012. Disponible en: <http://hdl.handle.net/20.500.12404/1507> [Consulta: 10 oct. 2025].

**[6]** ARENAS, G. *Diseño e implementación de un sistema RFID para el control de activos en tiempo real de bajo costo de un alojamiento vacacional temporal en el distrito de San Miguel*. Tesis de pregrado. Lima: Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Electrónica y Eléctrica, 2025.

**[7]** VILLARREAL DEL ÁGUILA, C. A. y AZPUR HUILLCA, E. G. *Implementación de un sistema de control de ingreso y salida de dispositivos electrónicos usando tecnología RFID para la Universidad Andina del Cusco*. 2017.

**[8]** GÓMEZ, R. y PRIETO, M. Gestión del control de activos tecnológicos en instituciones universitarias mexicanas. *Revista Latinoamericana de Tecnologías Digitales*, 2022, vol. 14(1), pp. 65–78.

**[9]** CORREA ESPINAL, A.; ÁLVAREZ LÓPEZ, C. E. y GÓMEZ MONTOYA, R. A. Sistemas de identificación por radiofrecuencia, código de barras y su relación con la gestión de la cadena de suministro. *Estudios Gerenciales*, 2010, vol. 26, n.º 116, pp. 115–141.

**[10]** VERA, L. y MONTOYA, E. Limitaciones del control manual de activos tecnológicos en universidades peruanas. *Revista Peruana de Ciencia y Tecnología*, 2023, vol. 6(2), pp. 120–134.

**[11]** DELGADO, A. y RUIZ, G. Gestión de seguridad y control de recursos en instituciones educativas del Perú. *Revista de Innovación y Desarrollo Universitario*, 2021, vol. 5(1), pp. 33–47.

**[12]** MAMANI, P. y CHOQUE, R. Deficiencias en el control de activos tecnológicos en instituciones educativas de Cusco. *Revista Andina de Sistemas y Gestión*, 2022, vol. 4(2), pp. 77–89.

**[13]** DIRECCIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN DEL CUSCO (DTI-Cusco). *Informe técnico sobre control de activos tecnológicos en laboratorios universitarios de la región*. Gobierno Regional del Cusco, 2023.

**[14]** GLOVER, B. y BHATT, H. *RFID Essentials*. O’Reilly, 2006.

**[15]** KOLBAN, N. *Mastering the ESP32*. Edición técnica digital, 2017 (y ediciones posteriores).

**[16]** ISO/IEC. *ISO/IEC 27001: Technologies de l'information — Techniques de sécurité — Systèmes de management de la sécurité de l'information*. Norma oficial.

**[17]** BUYYA, R.; VAHID, D.; et al. *Internet of Things: Principles and Paradigms*. Elsevier/Academic Press, 2016.

**[18]** FINKENZELLER, K. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. 3.ª ed. Chichester: John Wiley & Sons, 2010.

**[19]** HUAMÁN JULIÁN, Z. M. *Implementación de un sistema de gestión de seguridad electrónica con Machine Learning dirigido a Prosegur Perú para gestión de seguridad en viviendas de Lima Metropolitana*. Lima: Universidad Tecnológica del Perú, 2020.

# XI. ANEXOS

## 11.1 Evidencias de la funcionalidad de la aplicación por PMV en YouTube:

https://youtu.be/luDCGfcbfts

## 11.2 Evidencias del código implementado de la aplicación:

https://github.com/20Boris1998/SISTEC

## Imagen del repositorio

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

## Figura 33: Uso del Github