

# Estudio de Requerimientos –

## Proyecto Práctico - TC2 -

### CTT 2025

Instituto Técnico de Informática - CETP

*(Basada en ISO/IEC/IEEE 29148 e ISO/IEC 9126)*

---

#### Portada

SGLAB

Sistema de Gestión de Laboratorios

TC2

Juan Damián Pajares

Estado (Borrador / Aprobado / **Revisado**)

---

#### Historial de cambios

Versión	Fecha	Autor	Descripción del cambio	Revisor
0.1	03/09/2025	Nombre	Creación inicial	Tutor
1.0	8/10/2025	Nombre	Versión final aprobada	Tutor

---

# Índice

[Estudio de Requerimientos –](#)

[Proyecto Práctico - TC2 -](#)

[CTT 2025](#)

[Portada](#)

[Historial de cambios](#)

[Índice](#)

[1. Introducción](#)

[1.1 Propósito](#)

[1.2 Alcance](#)

[1.3 Referencias](#)

[1.4 Definiciones y acrónimos](#)

[2. Requerimientos](#)

[2.1 Requerimientos funcionales](#)

[2.2 Requerimientos no funcionales](#)

[2.3 Restricciones](#)

[2.4 Trazabilidad](#)

[3. Arquitectura y Diseño](#)

[3.1 Diagramas Utilizados](#)

[3.2 Descripción de Componentes y Relaciones](#)

[Relaciones Relevantes](#)

[3.3 Herramientas Utilizadas](#)

[4. Plan de Implementación](#)

[5. Plan de Verificación y Validación](#)

[6. Evidencias](#)

[7. Evaluación de Calidad \(ISO/IEC 9126\)](#)

[8. Conclusiones](#)

[9. Anexos](#)

---

# 1. Introducción

## 1.1 Propósito

Este documento tiene como objetivo definir los **requisitos funcionales (RF)** y **no funcionales (RNF)** del sistema **SGLAB (Sistema de Gestión de Laboratorio)** para el laboratorio LAB6, correspondiente al hito de **Estudio de Requerimientos** del Proyecto Práctico de la unidad curricular **TC2 - CTT 2025**. El sistema SGLAB permite registrar el estado de los PC 's del laboratorio mediante un servidor Apache alojado en un entorno Linux (WSL o VirtualBox), utilizando conexiones SSH y scripts de diagnóstico en PowerShell. Incluye una interfaz web con autenticación de usuarios (estudiantes y administradores) y un panel visual de administración para gestionar equipos y usuarios con filtros. El documento sirve como base para el diseño, implementación y validación del sistema, asegurando que cumpla con los estándares de calidad establecidos en las normativas ISO/IEC/IEEE 29148 e ISO/IEC 9126, de seguridad (ISO 27001) y las metodologías ágiles (Scrum), así como con los objetivos de robustez y velocidad de registro destacados por el docente como stakeholder principal.

## 1.2 Alcance

Este documento cubre la especificación de requisitos para el desarrollo del sistema SGLAB orientado al laboratorio LAB6, incluyendo:

- La configuración e instalación del servidor Apache +PHP 8.3/8.2 +PDO en Linux (Debian/Ubuntu) mediante PROXMOX.
- El establecimiento de conexiones SSH entre el servidor y las PC 's con Windows mediante claves SSH (pública y privada).
- La ejecución de scripts en PowerShell para diagnóstico de hardware y detallado de software, generación de archivos TOML con la información de equipo.
- La subida segura de archivos TOML al servidor mediante SCP/SSH.
- La implementación de una interfaz web con login para estudiantes (usando cédula como credenciales) y administradores. (**deseable** 2FA mediante email/token).
- El panel de administración para listar, filtrar y gestionar registros de Pc's y usuarios.

- La gestión de roles de usuario (estudiante, docente, asistente técnico, directora, subdirectora) y permisos.
- Operaciones CRUD para gestión de equipos, usuarios y registros (alta, baja, modificación, consulta).
- La conexión a base de datos MySQL/MariaDB mediante PDO y ODBC. (puerto 3306)
- Control de versiones y organización de tareas por medio de GitHub.

Quedan fuera del alcance de este hito:

- La integración con otros sistemas externos.
- El mantenimiento posterior a la entrega del proyecto.

### 1.3 Referencias

- ISO/IEC/IEEE 29148:2018 – *Systems and software engineering — Life cycle processes — Requirements engineering* (Ingeniería de sistemas y software — Procesos del ciclo de vida — Ingeniería de requisitos)  
<https://www.iso.org/obp/ui/en/#iso:std:iso-iec-ieee:29148:ed-2:v1:en>
- ISO/IEC 9126:2001 – *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)*  
<https://www.verity.cl/que-es-norma-iso-iec-9126-2001/>
- ISO/IEC 27001:2022 – Gestión de seguridad de la información  
<https://www.iso.org/obp/ui/es/#iso:std:iso-iec:27001:ed-3:v1:en>
- Manual de Gestión de Proyectos – CTT 2025
- Programa planeamiento educativo curso terciario redes y software (UTU)  
[https://planeamientoeducativo.utu.edu.uy/sites/planeamiento.utu.edu.uy/files/2017-11/INTRO\\_PROY\\_II.pdf](https://planeamientoeducativo.utu.edu.uy/sites/planeamiento.utu.edu.uy/files/2017-11/INTRO_PROY_II.pdf)
- Manual de Gestión de proyectos TI:  
<https://es.slideshare.net/slideshow/manual-de-gestin-de-proyectos-ti/59458108>
- Guía Scrum Guide 2020:  
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

- Diagramas oficiales del docente: Casos de uso, Diagrama de clases:  
[https://drive.google.com/file/d/1JPjG7z0fNYL-jdHCVfg2DYNj1a\\_Msc6O/view?usp=sharing](https://drive.google.com/file/d/1JPjG7z0fNYL-jdHCVfg2DYNj1a_Msc6O/view?usp=sharing)  
<https://drive.google.com/file/d/10X6Qur8rbAlxe-Pw91eKMk7zT9uk8dC4/view?usp=sharing>
- Material didáctico de la unidad curricular TC2 – Instituto Técnico de Informática CETP
- Documentación oficial de Apache HTTP Server: <https://httpd.apache.org/docs/>
- Documentación oficial de SSH: <https://www.ssh.com/ssh/>
- Documentación oficial de PowerShell:  
<https://docs.microsoft.com/enus/powershell/>
- Documentación oficial de TOML: <https://toml.io/en/>
- Documentación oficial de MySQL: <https://dev.mysql.com/doc/>
- Documentación oficial de MariaDB: <https://mariadb.com/kb/en/documentation/>
- Documentación oficial de PDO: <https://www.php.net/manual/es/book.pdo.php>

Repositorio GitHub del proyecto: <https://github.com/UTU-ITI/SGLAB.git>

## 1.4 Definiciones y acrónimos

- **SGLAB**: Sistema de Gestión de Laboratorio
- **LAB6**: Laboratorio 6, sala de Pc's objeto del sistema
- **RF**: Requisito Funcional
- **RNF**: Requisito No Funcional
- **WSL**: Windows Subsystem for Linux (característica de Windows que permite ejecutar un entorno Linux en la máquina Windows)
- **SSH**: Secure Shell (protocolo de red que permite el acceso remoto seguro a computadoras a través de una conexión cifrada)
- **SCP**: Secure Copy Protocol (Protocolo de copia segura)
- **2FA**: Autenticación de dos factores (para roles administrativos)
- **TOML**: Tom's Obvious, Minimal Language (formato de archivo de configuración)
- **PDO**: PHP Data Objects (interfaz de acceso a bases de datos en PHP)

- **ODBC:** Open Database Connectivity (estándar de acceso a bases de datos)
- **MySQL:** Sistema de gestión de bases de datos relacional
- **MariaDB:** Fork de MySQL, sistema de gestión de bases de datos relacional
- **Apache:** Servidor HTTP de código abierto
- **CRUD:** Create, Read, Update, Delete (operaciones base de persistencia)
- **PHP:** Lenguaje de programación utilizado para el desarrollo web del sistema
- **GitHub:** por medio de la plataforma el estudiante puede hacer manejo de versiones, actualizar el repositorio y dar orden a las tareas solicitadas por el docente que corresponden al proyecto en cada sprint.
- **ISO:** International Organization for Standardization (Organización Internacional de Normalización)
- **IEEE:** Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)
- **CETP:** Consejo de Educación Técnico Profesional
- **CTT:** Curso Técnico Terciario
- **Roles de usuario:**
  - **Estudiante:** Usuario que puede iniciar sesión y actualizar estado de PC.
  - **Docente:** Usuario con permisos extendidos (visualización de registro, importación de listado de estudiantes y pcs).
  - **Asistente técnico:** Usuario con permisos para gestionar equipos y registros.
  - **(administrativo)Directora/subdirectora:** Usuario con permisos administrativos completos.

- **Entidades principales** (según diagrama de clases del docente):
  - o Equipos: serial, hostname, CPU, RAM, diskFree, diskType.
  - o Registro: Fecha, Estado, Descripción.
  - o Personas: ci, nombre, apellido, dirección, fechaNacimiento.
  - o Usuarios: id, ci, nombre, apellido, email, password, rol.

---

## 2. Requerimientos

*(Adaptado de la estructura de ISO/IEC/IEEE 29148)*

## **2.1 Requerimientos funcionales**

Lista clara y numerada de funcionalidades que el sistema debe cumplir en este hito.  
Formato sugerido:

### **1) Base de Datos — DB (6 requerimientos)**

- **RF-DB-01** — Definir entidades principales del dominio.
- **RF-DB-02** — Modelar relaciones y cardinalidades entre entidades.
- **RF-DB-03** — Establecer el tipo de dato de la fecha de los registros como INT (epoch time).
- **RF-DB-04** — Elaborar diagrama MER completo y validado.
- **RF-DB-05** — Crear usuario dedicado para la aplicación (CREATE USER) con privilegios mínimos necesarios.
- **RF-DB-06** — Escribir script SQL inicial (DDL + datos semilla si aplica).

### **2) Ingeniería (8 requerimientos)**

- **RF-ING-01** — Redactar el Estudio de Requerimiento (ESRE) completo.
- **RF-ING-02** — Elaborar Diagrama de Casos de Uso.
- **RF-ING-03** — Elaborar Diagrama de Clases.
- **RF-ING-04** — Organizar Formación de grupos y responsabilidades.
- **RF-ING-05** — Elaborar Diagrama de Implementación (Servidor Proxmox y componentes).
- **RF-ING-06** — Elaborar Diagrama de Dominio.
- **RF-ING-07** — Construir Cronograma GANTT del proyecto.
- **RF-ING-08** — Crear Planilla de Registro (definición de campos y flujos).

### 3) Linux / Infra (14 requerimientos)

- **RF-LNX-01** — Crear script de instalación del servidor.
- **RF-LNX-02** — Instalar el servidor base.
- **RF-LNX-03** — Instalar MySQL.
- **RF-LNX-04** — Instalar Apache2.
- **RF-LNX-05** — Instalar PHP 8.3.
- **RF-LNX-06** — Configurar PDO (extensión php-mysql).
- **RF-LNX-07** — Generar claves SSH genéricas para automatización de estado.
- **RF-LNX-08** — Instalar y configurar OpenSSH-Server.
- **RF-LNX-09** — Crear usuario genérico para la tecnicatura (mínimos privilegios).
- **RF-LNX-10** — SPAWN: Script que ingresa información de archivos .TOM a la base de datos.
- **RF-LNX-11** — Respaldo de la base de datos con crontab.
- **RF-LNX-12** — Instalar y configurar Git en el servidor.
- **RF-LNX-13** — Cargar a la base cada hora los archivos generados por PowerShell (tarea programada).
- **RF-LNX-14** — CRONTAB: Mantenimiento y limpieza programada de archivos .TOM.

### 4) Backend PHP (18 requerimientos)

- **RF-PHP-01** — Clase Registro.
- **RF-PHP-02** — Clase Persona.
- **RF-PHP-03** — Clase Periférico.
- **RF-PHP-04** — Clase Equipos.
- **RF-PHP-05** — Clase Usuario.
- **RF-PHP-06** — Clase Perfiles (roles/permiso).
- **RF-PHP-07** — Sistema de login/logout y manejo de sesiones.
- **RF-PHP-08** — Protección de rutas (middleware/admin).
- **RF-PHP-09** — Perfiles de Usuario (Admin / Estudiante / Docente).



- **RF-PHP-10** — Importación de estudiantes (archivo fuente y validaciones).
- **RF-PHP-11** — POST /registros (crear registro).
- **RF-PHP-12** — PUT /registros/{id} (actualizar registro).
- **RF-PHP-13** — GET /registros (listar con filtros/paginación si aplica).
- **RF-PHP-14** — Conversión de fechas a/desde epoch time.
- **RF-PHP-15** — Visualización de gráfico (chart) con estados.
- **RF-PHP-16** — Formulario de inicio de sesión (perfil estudiante).
- **RF-PHP-17** — Archivo .TOM con usuario administrador (bootstrap/seed).
- **RF-PHP-18** — Manejo de sesiones (expiración, regeneración de ID, seguridad).

## **5) RedesMS / PowerShell (11 requerimientos)**

- **RF-NET-01** — Verificar directorio de destino en el servidor antes de copias.
- **RF-NET-02** — Diagrama de red del laboratorio.
- **RF-NET-03** — Cargar script al inicio del sistema.
- **RF-NET-04** — Obtener datos básicos del PC.
- **RF-NET-05** — Ver recursos disponibles (almacenamiento, RAM, CPU, etc.).
- **RF-NET-06** — Borrar archivos generados por PowerShell con antigüedad > 7 días.
- **RF-NET-07** — Obtener diagnóstico de red (IP, gateway, DNS, latencia).
- **RF-NET-08** — Respaldo de la base de datos en dispositivo extraíble.
- **RF-NET-09** — Verificar estado del servidor SSH antes de copiar.
- **RF-NET-10** — Obtener número de serie de la computadora.
- **RF-NET-11** — Enviar e-mail si no está ingresado el S/N.

## **6) Web / UI (8 requerimientos)**

- **RF-WEB-01** — Dashboard de Profesor.
- **RF-WEB-02** — Dashboard de Administrador.

- **RF-WEB-03** — Mockup / Diseño de pantallas en Figma o Balsamiq.
- **RF-WEB-04** — Formulario de Registro de Estado (Alumno).
- **RF-WEB-05** — Formulario de Registro de Estado (Admin).
- **RF-WEB-06** — Formulario de Administración (importación de PCs y Estudiantes).
- **RF-WEB-07** — Formulario de Inicio de sesión (Admin).
- **RF-WEB-08** — Listado Web de PCs (tabla con filtros/búsqueda).

## 2.2 Requerimientos no funcionales

Características de calidad que deben cumplirse (**ISO/IEC 9126**): funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad.

Formato sugerido:

- RNF-01: [Descripción clara y medible]RNF-01 Seguridad **MUST**: hash de contraseñas (bcrypt/argon2), RBAC, HTTPS, mitigación CSRF/XSS.
- RNF-02 Rendimiento **SHOULD**: listados/búsquedas 3 s (10k equipos); exportaciones 10 s (50k registros).
- RNF-03 Disponibilidad **SHOULD**: 99% mensual.
- RNF-04 Usabilidad **SHOULD**: responsive y accesible (WCAG AA).
- RNF-05 Mantenibilidad **SHOULD**: logs estructurados, migraciones versionadas, API/FE desacoplados.
- RNF-06 Portabilidad **SHOULD**: navegadores modernos; backend desplegable en Linux.

## 2.3 Restricciones

Limitaciones técnicas, de tiempo, de recursos o normativas.

## 2.4 Trazabilidad

Tabla que vincula cada requerimiento con:

- Artefacto que lo implementa.
- Caso de prueba que lo valida.
- Evidencia documental.

# 3. Arquitectura y Diseño

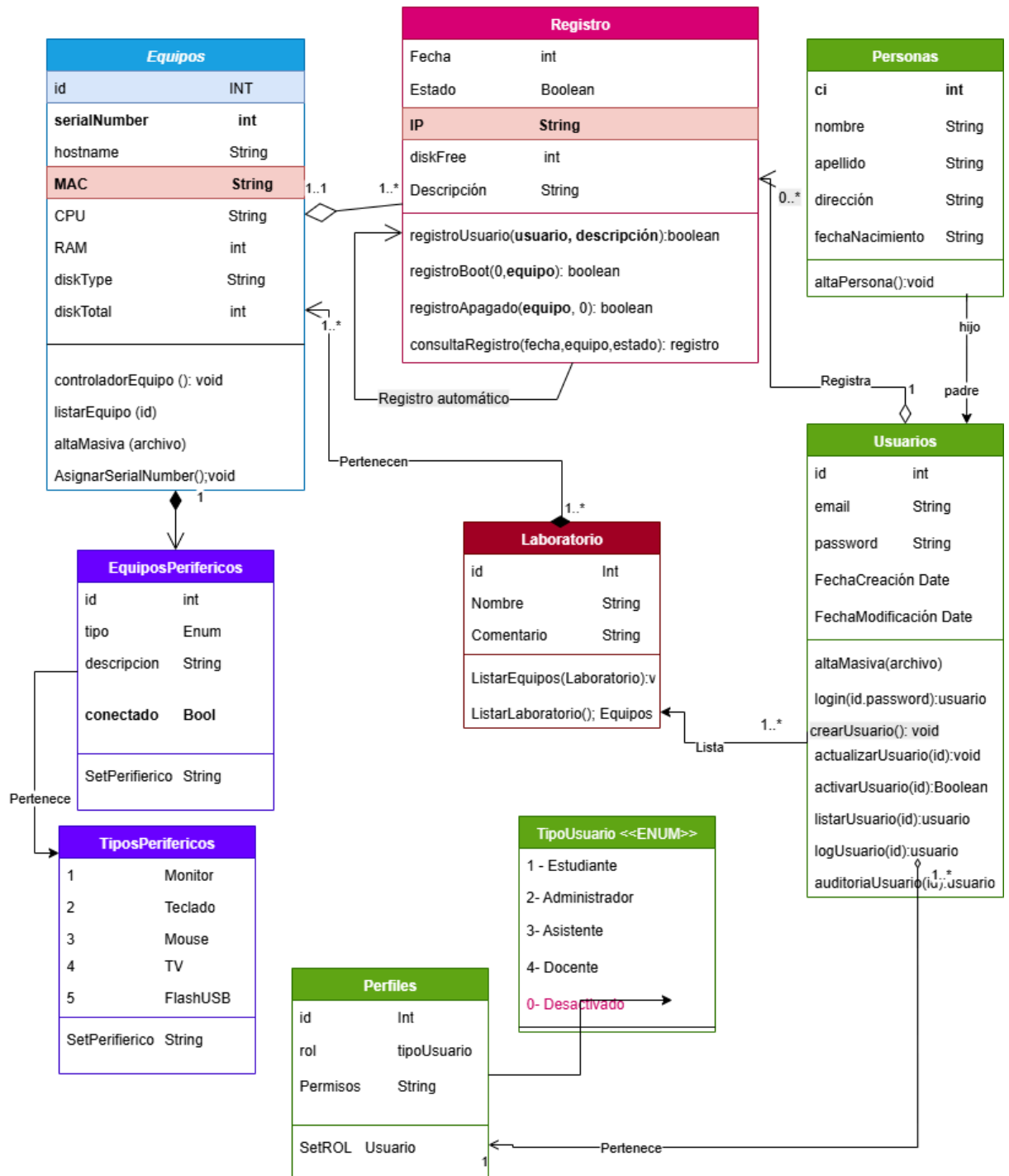
### 3.1 Diagramas Utilizados

Durante el desarrollo del sistema **SGLAB**, se emplearon diferentes diagramas UML y modelos de representación con el fin de documentar, analizar y diseñar la solución propuesta.

- **Diagrama de Clases UML:**  
Representa las entidades principales del sistema, sus atributos, métodos y las

relaciones entre ellas.

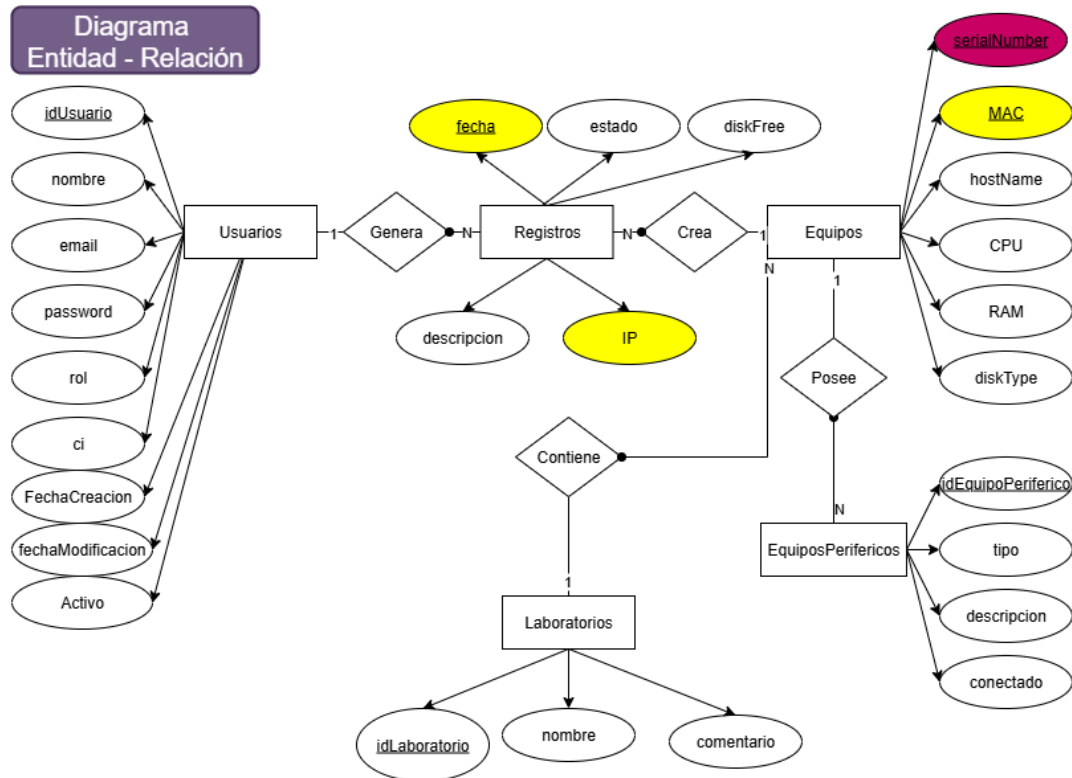
## Diagrama de Clases SGLAB



[Enlace al diagrama de clases](#)

- **Modelo Entidad-Relación (M-E-R) y Diagrama Entidad Relación (D-E-R):**  
Se construyó el modelo E-R para definir la estructura de la base de datos, identificando las entidades, atributos y relaciones. Este modelo asegura la

consistencia de los datos y facilita la implementación en un gestor de base de datos relacional.



[Enlace al diagrama de entidad relación](#) **Diagramas de Red:**

Se consideró la topología lógica necesaria para la conexión de equipos y laboratorios, representando los nodos principales y la comunicación entre ellos.

[Enlace al diagrama de Red](#)

### 3.2 Descripción de Componentes y Relaciones

El sistema está compuesto por los siguientes **componentes principales**:

- **Equipos:** representan las computadoras dentro de los laboratorios, con atributos técnicos como serialNumber, hostName, CPU, RAM y tipo de disco.
- **Periféricos:** dispositivos asociados a los equipos (monitor, teclado, mouse, TV, USB), gestionados a través de relaciones con la entidad **EquiposPerifericos**.
- **Laboratorios:** representan los salones en los cuales se implementa el software.
- **Usuarios:** representan a los distintos actores que interactúan con el sistema (estudiantes, administradores, docentes, asistentes). Están asociados a la entidad

## **Personas .**

- **Registros:** almacenan la trazabilidad de uso de los equipos, indicando usuario, estado del equipo, espacio en disco y fecha de uso.

## Relaciones Relevantes

- Un **Laboratorio** contiene múltiples **Equipos**.
  - Un **Equipo** puede estar asociado a varios **Periféricos**.
  - Un **Usuario** corresponde a una **Persona** y puede realizar múltiples **Registros**.
  - Los **Registros** vinculan a los **Usuarios** con los **Equipos** que utilizan.
  - Cada **Usuario** posee un **Tipo de Rol** (estudiante, administrador, docente, asistente).
- 

## 3.3 Herramientas Utilizadas

Para el diseño y documentación se emplearon las siguientes herramientas:

- **Draw.io (Diagrams.net)**: para la creación de diagramas UML:
    - Diagrama de Clases.
    - Diagrama de E-R.
    - Diagrama de Red.
  - **Entorno de desarrollo (Visual Studio Code)**
- 

## 4. Plan de Implementación

- Actividades a realizar.
  - Cronograma (por hito).
  - Responsables.
- 

## 5. Plan de Verificación y Validación

- Casos de prueba asociados a cada requerimiento.
- Herramientas de testing utilizadas.
- Criterios de aceptación.

---

## 6. Evidencias

- Capturas de pantalla.
- Scripts o configuraciones relevantes.
- Logs de ejecución.

---

## 7. Evaluación de Calidad (ISO/IEC 9126)

### Funcionalidad:

- Descripción de cómo se evaluó.
- Resultado obtenido.
- Observaciones y acciones de mejora.

### Usabilidad:

- Descripción de cómo se evaluó.
- Resultado obtenido.
- Observaciones y acciones de mejora.

### Mantenibilidad:

- Descripción de cómo se evaluó.
- Resultado obtenido.
- Observaciones y acciones de mejora.

---

## 8. Conclusiones

Resumen de lo logrado en el hito y próximos pasos.

---

## 9. Anexos

- Glosarios.
- Listados de comandos.



- Configuraciones extendidas.

