



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN CENTRALIZADA DE
INVENTARIO IT Y REPORTES DE CAMPO, CON AUTOMATIZACIÓN DE
FORMULARIOS Y CONTROL EN TIEMPO REAL DE LA EMPRESA KLUANE
DRILLING ECUADOR**

**Proyecto de titulación previo a la obtención del título de: Tecnología Superior en
Desarrollo de Software**

Autor: Victor Paul Alvarez Alvarez

Tutor: Ing. Christian Roberto Tapia Gaibor

Quito, Ecuador

2025

Dedicatoria

Este trabajo lo dedico con todo mi corazón a **mi madre, que está en el cielo**, porque aunque no esté físicamente conmigo, su amor sigue presente en mi vida y en cada meta que alcanzo.

A mi novia, por ser mi compañera, mi apoyo y mi motivación durante todo este proceso. Gracias por caminar conmigo, por sostenerme en los días pesados y por impulsarme a seguir.

A la madre de mi novia, por su apoyo y por brindarme su confianza y ánimo en cada etapa de este recorrido.

A la señora **Mercy** y a su hija, por aparecer en mi vida y tenderme la mano en los momentos más difíciles. Su ayuda, su cercanía y su apoyo fueron fundamentales para no rendirme.

Y a mis amigos de la universidad, los que se quedaron hasta el final: gracias por su lealtad, su compañía y por hacer más llevadero este camino. Este logro también lleva un pedazo de ustedes.

Tabla de contenidos

Introducción	9
Problema Técnico/Tecnológico	10
Objetivos	11
Objetivo general:	11
Objetivos específicos:	11
Marco Conceptual.....	12
Capítulo I.....	13
Levantamiento de Requisitos y Diseño del Sistema	13
1.1 Antecedentes	13
1.2 Metodología Scrum.....	13
1.3 Etapas del proceso	14
1.4 Actores principales.....	15
1.5 Historias de usuario	15
1.5.1 Flujo de historias	16
1.5.2 Caso de usos.....	17
1.5.3 Caso de usos extendidos.....	18
1.6 Planificación de los Sprint	40
1.7 Herramientas y software	42
1.7.1 Flujo de trabajo con Git.....	44
1.7.2 Procedimiento de validación antes de integrar a producción	45
1.8 Hardware	46
1.9 Viabilidad del proyecto	47
1.9.1 Viabilidad técnica	47
1.9.2 Viabilidad económica	48
1.9.3 Viabilidad operativa	49
Capítulo II	50
Construcción del Sistema.....	50
2.1 Alcance.....	50
2.2 Análisis de diseño.....	51
2.3 Mapa de navegación	52
2.4 Lógica de negocio.....	54
2.4.1 Ilustración de Arquitectura del Proyecto (Arquitectura de Aplicaciones)	54
2.4.2 Ilustración de Arquitectura Tecnológica (Frontend / Backend / Base de Datos)	55
2.4.3 Reglas de negocio	56

2.4.4 Diagrama de Secuencia	57
2.5 Diseño de la arquitectura de la aplicación.....	57
2.5.1 Arquitectura tecnológica y despliegue (entorno de operación)	59
2.6 Modelo de Base de Datos (BDD).....	60
2.6.1 Entidades principales	61
2.6.2 Relaciones del modelo	62
2.7 Requisitos Funcionales.....	64
2.8 Requisitos no Funcionales.....	64
Capítulo III	66
Pruebas y Estabilización	66
3.1 Reuniones con Product Owner	66
3.1.1 Objetivo de las reuniones	66
3.1.2 Participantes y roles.....	66
3.2 Pruebas y estabilización	71
3.2.1 Proceso de pruebas en Scrum.....	71
3.3 Tipos de pruebas realizadas.....	73
3.3.1 Herramientas utilizadas	74
3.3.2 Planilla de pruebas Sprint	74
3.3.3 Estabilización.....	76
Conclusiones	78
Recomendaciones	79
Referencias bibliográficas	81

Índice de tablas

Tabla 1.	Historias de usuario.....	15
Tabla 2.	Caso de usos	17
Tabla 3.	Credenciales incorrectas	18
Tabla 4.	Inicio de sesión con campos faltantes.....	19
Tabla 5.	Inicio de sesión exitoso	20
Tabla 6.	Registrar activo con marca faltante	21
Tabla 7.	Registrar activo con modelo faltante.....	22
Tabla 8.	Registrar activo con categoría no seleccionada	22
Tabla 9.	Registrar activo exitoso.....	23
Tabla 10.	Registrar activo con código interno duplicado	24
Tabla 11.	Consultar matriz de activos (dashboard + tabla)	25
Tabla 12.	Listar usuarios del sistema	27
Tabla 13.	Registrar usuario con datos faltantes	27
Tabla 14.	Registrar usuario con contraseña inválida.....	28
Tabla 15.	Registrar usuario exitoso.....	29
Tabla 16.	Eliminar usuario (administración)	30
Tabla 17.	Consultar inventario en sitio (campamento)	31
Tabla 18.	Editar datos de campamento (exitoso).....	32
Tabla 19.	Transferir/Devolver con destino faltante.....	33
Tabla 20.	Transferencia/Devolución exitosa.....	34
Tabla 21.	Exportar PDF oficial (campamento).	35
Tabla 22.	Consultar historial de movimientos	36
Tabla 23.	Registrar reporte de campo con datos	37
Tabla 24.	Registrar reporte de campo exitoso.....	38
Tabla 25.	Editar reporte de campo	39
Tabla 26.	Planificación general de Sprints	41
Tabla 27.	Requisitos mínimos de hardware.....	46
Tabla 28.	Historias de usuario.....	51
Tabla 29.	Mapa de navegación del sistema	52
Tabla 30.	Requisitos funcionales.....	64
Tabla 31.	Requisitos no funcionales	64
Tabla 32.	Planilla de pruebas por sprint (Resumen).....	74

Índice de figuras

Gráfico 1.	Flujo de historias	16
Gráfico 2.	Casos de uso iniciar sesión	18
Gráfico 3.	Caso de uso registrar activo	21
Gráfico 4.	Caso de uso matriz de activos	25
Gráfico 5.	Caso de uso administración de usuarios	26
Gráfico 6.	Caso de uso gestión de campamentos.....	31
Gráfico 7.	Caso de uso Exportación PDF	35
Gráfico 8.	Caso de uso historial / trazabilidad	36
Gráfico 9.	Caso de uso reportes de campo	37
Gráfico 10.	Flujo de trabajo frontend y backend	40
Gráfico 11.	Ramificación de github	44
Gráfico 12.	Arquitectura de Aplicaciones.....	55
Gráfico 13.	Arquitectura Tecnológica.....	56
Gráfico 14.	Diagrama de Secuencia.....	57
Gráfico 15.	Arquitectura de la aplicación	59
Gráfico 16.	Arquitectura de despliegue	60
Gráfico 17.	Modelo de base de datos	63
Gráfico 18.	Reunión Presentación	67
Gráfico 19.	Pantalla de inicio (dashboard – matriz de activos)	68
Gráfico 20.	Ingreso de un nuevo activo (formulario de registro).....	68
Gráfico 21.	Formulario de registro de activos con validaciones	69
Gráfico 22.	Asignación de rol y sede en el registro de usuarios.....	69
Gráfico 23.	Confirmación de registro exitoso y visualización en la matriz de activos	70
Gráfico 24.	Generación de PDF oficial (Matriz 09) del inventario en campamento	70
Gráfico 25.	Gestión de usuarios del sistema	70
Gráfico 26.	Prueba de inicio de sesión	72
Gráfico 27.	Verificación de trazabilidad en base de datos (historial de movimientos).....	72
Gráfico 28.	Ejecución de prueba de generación de PDF oficial.....	73

DECLARACIÓN y AUTORIZACIÓN

Yo, Victor Paul Alvarez Alvarez con C.I. 1752780732 autor(a) del trabajo de titulación intitulado: “DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN CENTRALIZADA DE INVENTARIO IT Y REPORTES DE CAMPO, CON AUTOMATIZACIÓN DE FORMULARIOS Y CONTROL EN TIEMPO REAL DE LA EMPRESA KLUANE DRILLING ECUADOR”, previa a la obtención del título de Tecnología Superior en Desarrollo de Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1. Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 23 de agosto de 2025



Victor Paul Alvarez Alvarez

C.I. 1752780732

Agradecimientos

Agradezco a Dios por la fortaleza y constancia para culminar este proyecto.

A mi familia, por su apoyo, comprensión y motivación durante todo el proceso académico.

A mi tutor, Ing. Christian Roberto Tapia Gaibor, por su guía, observaciones y acompañamiento en el desarrollo del proyecto.

A la empresa Kluane Drilling Ecuador, por permitir el levantamiento de información y brindar el contexto real para aplicar los conocimientos adquiridos.

Finalmente, a la Pontificia Universidad Católica del Ecuador – PUCE TEC, por la formación recibida y por impulsar proyectos que aportan al sector productivo.

Introducción

Hoy en día, las organizaciones que manejan activos tecnológicos que enfrentan el reto de llevar un control riguroso y actualizado de sus inventarios, especialmente cuando los dispositivos son utilizados por personal de campo. En el caso de Kluane Drilling Ecuador, la administración de laptops, celulares, impresoras, radios y otros equipos se realiza con matrices en Excel separados, lo que provoca errores, duplicaciones y demoras en la actualización de la información.

Asimismo, los técnicos en campo remiten sus reportes en formatos Excel adicionales, los cuales deben ser unificados manualmente por el área de Tecnología de la Información utilizando funciones como BUSCARV. Este procedimiento es poco eficiente y vulnerable a equivocaciones.

Este proyecto tiene como objetivo desarrollar una solución web que permita centralizar el inventario tecnológico y facilite a los técnicos enviar sus reportes directamente desde campo. Con esto se eliminará el uso de hojas de cálculo, se mejorará el seguimiento de los activos y se fortalecerá el control del estado y la localización de los equipos tecnológicos, optimizando los procesos internos y favoreciendo la toma de decisiones.

Problema Técnico/Tecnológico

Actualmente, la empresa administra su inventario de tecnología mediante hojas de cálculo en Excel que requieren actualización manual. De igual manera, los técnicos en campo envían sus reportes en este mismo formato, obligando al equipo de IT a realizar procesos repetitivos de validación y actualización. Esta situación acarrea múltiples inconvenientes técnicos:

- Demoras en la consolidación de información: cada archivo debe revisarse y compararse de forma individual.
- Pérdida o sobrescritura de datos: es común que se omitan registros o se modifiquen sin respaldo.
- Carga administrativa innecesaria: se dedica mucho tiempo a tareas que podrían automatizarse.
- Falta de trazabilidad: no es posible identificar fácilmente quién realizó un cambio o cuando.
- Descentralización de la información: cada sede opera con archivos distintos y no conectados entre sí.

Objetivos

Objetivo general:

Implementar un sistema web que permita automatizar la gestión del inventario IT y el envío de reportes técnicos desde campo, centralizando la información en una base de datos accesible desde cualquier sede, para mejorar los tiempos de respuesta, precisión de los datos y facilitar el acceso a la información.

Objetivos específicos:

- Diseñar una base de datos relacional que integre tanto el inventario como los reportes del personal técnico, a través del modelo entidad-relación, para asegurar la integridad y acceso eficiente a los datos desde cualquier módulo.
- Desarrollar formularios web que permitan registrar reportes de campo en tiempo real, utilizando tecnologías web (HTML, CSS, JavaScript etc), para agilizar el envío de la información técnica.
- Construir una interfaz administrativa que permita al personal de IT consultar, modificar y exportar información, para reemplazar el uso de hojas de cálculo por un sistema digital centralizado.

Marco Conceptual

El proyecto se fundamenta en principios de ingeniería de software orientados a la automatización y control de procesos administrativos asociados a la gestión de activos tecnológicos. La solución propuesta adopta una arquitectura web cliente–servidor, con una base de datos relacional y módulos que garantizan integridad, trazabilidad y acceso centralizado.

Tecnologías y herramientas seleccionadas:

- PHP: lenguaje de programación del lado del servidor utilizado para implementar la lógica de negocio y la generación dinámica de páginas web.
- MySQL (con phpMyAdmin): sistema gestor de bases de datos relacional empleado para almacenar inventario, usuarios, sedes, reportes e historial de cambios.
- HTML, CSS y JavaScript: tecnologías del frontend para construir interfaces accesibles desde cualquier navegador y dispositivo.
- Bootstrap: framework para diseñar interfaces limpias, responsivas y consistentes en distintos tamaños de pantalla.
- cPanel: entorno de hosting utilizado para el despliegue del sistema y la administración de recursos del servidor.
- GitHub: control de versiones para organizar el código fuente, respaldar avances y facilitar mejoras futuras.
- GitHub Projects: herramienta de gestión visual para el seguimiento de tareas y organización del backlog y sprints.
- Scrum: marco de trabajo ágil que divide el desarrollo en sprints de dos semanas, permitiendo revisiones, retroalimentación continua y entregas incrementales.

Capítulo I

Levantamiento de Requisitos y Diseño del Sistema

1.1 Antecedentes

Kluane Drilling Ecuador cuenta con operaciones distribuidas en varias sedes y proyectos, donde el uso y traslado de equipos tecnológicos es constante. La administración del inventario IT se ha realizado históricamente mediante hojas de cálculo independientes, lo que limita la actualización oportuna y dificulta el control centralizado de activos.

Adicionalmente, el personal técnico de campo reporta novedades de equipos y actividades mediante formatos Excel separados, los cuales deben consolidarse manualmente en el área de TI. Este escenario evidencia la necesidad de un sistema web propio que centralice el inventario y permita el registro de reportes desde campo de forma estandarizada y en tiempo real.

1.2 Metodología Scrum

Se seleccionó la metodología ágil Scrum por su adaptabilidad, enfoque iterativo e incremental, y por facilitar la entrega de incrementos funcionales de software en periodos cortos. Esto permite incorporar retroalimentación continua del área de TI de Kluane Drilling Ecuador y reducir riesgos durante el desarrollo.

Roles definidos:

- **Scrum Master:** Ing. Christian Roberto Tapia Gaibor.
- **Product Owner:** Ing. Iván Pérez.
- **Desarrollador:** Victor Paul Alvarez Alvarez.

Ceremonias y seguimiento:

- Planificación de sprint: definición de objetivo del sprint y tareas priorizadas.
- Daily (adaptada): seguimiento diario documentado en GitHub Project.

- Revisión de sprint: demostración de avances y recopilación de observaciones.
- Retrospectiva: identificación de mejoras para el siguiente sprint.

Herramientas de gestión y control: GitHub Project para seguimiento de tareas y GitHub para control de versiones.

1.3 Etapas del proceso

Levantamiento de Requisitos

- Reuniones con el responsable del área de TI para identificar necesidades, activos gestionados y flujo de reportes.
- Revisión de matrices actuales (Excel) de inventario y formatos de reportes de campo.
- Identificación de puntos críticos: duplicidad de registros, falta de trazabilidad, demoras en consolidación y ausencia de control en tiempo real.

Diseño

- Diseño del modelo entidad–relación y normalización de tablas para inventario, usuarios y reportes.
- Diseño de interfaces (mockups) para módulos: inventario, reportes de campo, administración y exportación.
- Definición de validaciones, roles/permisos y bitácora de cambios.

Construcción

- Desarrollo por sprints de módulos (inventario, reportes, administración), integrando base de datos y frontend.
- Control de versiones en GitHub y seguimiento de tareas en GitHub Projects.

Pruebas

- Pruebas funcionales por módulo, validación de formularios y verificación de permisos por rol.

- Pruebas de integración (formularios -> BD, exportación, consultas) y pruebas de usabilidad con usuarios del área de TI.

Implementación

- Despliegue del sistema en hosting con cPanel y configuración de base de datos MySQL.
- Capacitación básica al personal y entrega de documentación (manual de usuario y guía técnica).

1.4 Actores principales

- **Administrador TI:** gestiona inventario, catálogos, usuarios, permisos y auditoría.
- **Técnico de campo:** registra reportes desde terreno y consulta asignaciones.
- **Colaborador:** visualiza asignaciones.

1.5 Historias de usuario

Tabla 1. Historias de usuario

ID	Historia de usuario	Rol	Prioridad
HU01	Como usuario, quiero iniciar sesión para acceder al sistema según mi rol.	Usuario	Alta
HU02	Como Admin TI, quiero registrar activos IT para centralizar el inventario.	Admin TI	Alta
HU03	Como Admin TI, quiero editar activos para mantener información correcta.	Admin TI	Alta
HU04	Como Admin TI, quiero asignar/transferir activos por sede/responsable para controlar custodia y ubicación.	Admin TI	Alta

HU05	Como Técnico, quiero registrar y editar reportes de campo para reportar novedades sin usar Excel.	Técnico	Alta
HU06	Como Admin TI, quiero administrar usuarios/roles para controlar accesos.	Admin TI	Media
HU07	Como Admin TI, quiero exportar inventario/reportes a PDF para auditorías y reportes internos.	Admin TI	Media
HU08	Como Admin TI, quiero consultar historial de movimientos para trazabilidad.	Admin TI	Media

1.5.1 Flujo de historias

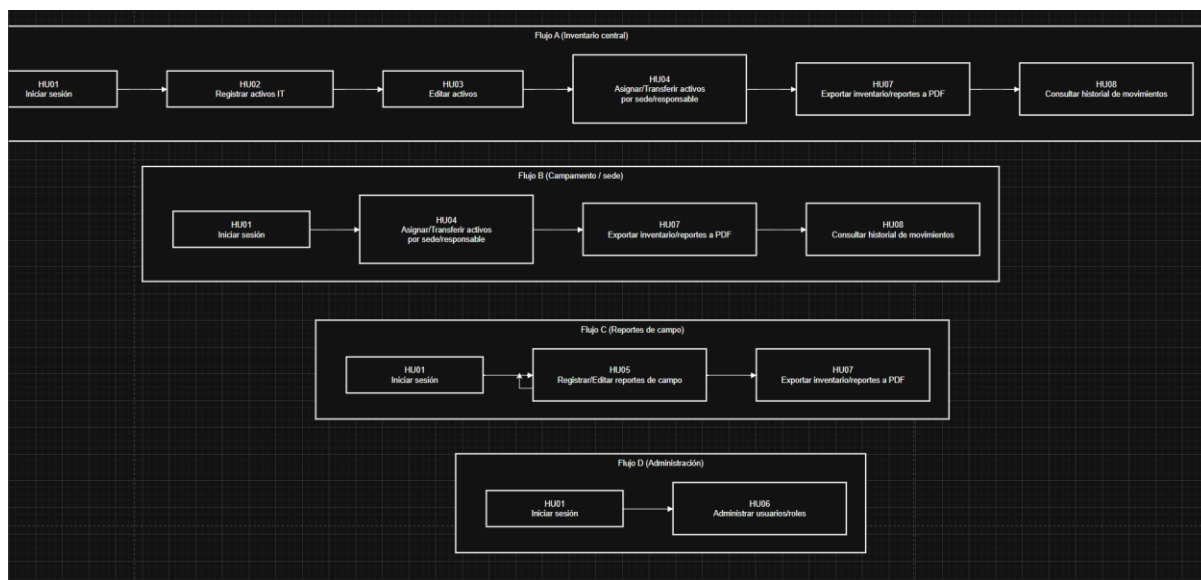


Gráfico 1. Flujo de historias

1.5.2 Caso de usos.

Tabla 2. Caso de usos

Código	Caso de uso	Actor principal
CU01	Iniciar sesión	Usuario
CU02	Registrar activo (Nuevo Activo)	Admin TI
CU03	Editar activo	Admin TI
CU04	Gestión de campamento: editar datos + transferir/devolver	Técnico/Admin
CU05	Reportes de campo: registrar/editar/consultar	Técnico
CU06	Administrar usuarios (crear/editar/eliminar)	Admin TI
CU07	Exportar PDF (matriz / PDF oficial campamento)	Admin TI
CU08	Consultar historial de movimientos	Admin TI

1.5.3 Caso de usos extendidos.

CU01 — INICIAR SESIÓN

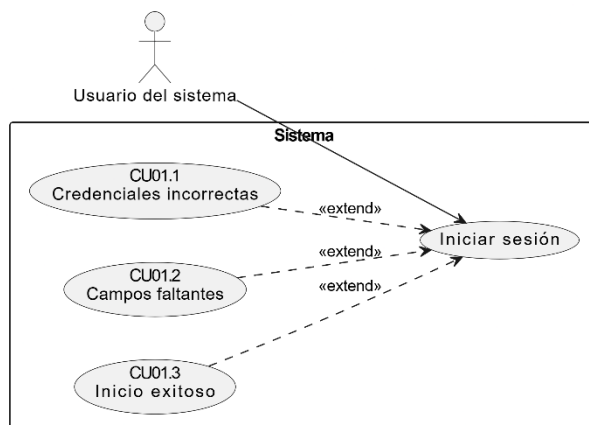


Gráfico 2. Casos de uso iniciar sesión

Tabla 3. Credenciales incorrectas

Campo	Detalle
Nombre del caso de uso	CU01.1 – Credenciales incorrectas
Actor principal	Usuario
Descripción	El sistema valida credenciales y, si son inválidas, niega el acceso.
Precondiciones	Usuario en la pantalla de login.
Postcondiciones	No inicia sesión; permanece en login.
RNF asociados	RNF-S01, RNF-U01
Flujo normal	1) Ingresar correo y contraseña. 2) Presionar INGRESAR. 3) Sistema valida.

	4) Deniega acceso y muestra error.
Flujos alternos	A1) Si hay múltiples intentos, el sistema puede limitar intentos (si aplica).

Tabla 4. Inicio de sesión con campos faltantes

Campo	Detalle
Nombre del caso de uso	CU01.2 – Campos faltantes
Actor principal	Usuario
Descripción	Valida que correo y contraseña estén completos (mensaje “Completa este campo”).
Precondiciones	Usuario intenta ingresar con campos vacíos.
Postcondiciones	No se procesa autenticación.
RNF asociados	RNF-U01
Flujo normal	1) Dejar vacío correo o contraseña. 2) Presionar INGRESAR. 3) Sistema/navegador marca campo requerido (“Completa este campo”).
Flujos alternos	A1) Formato email inválido: mostrar validación (si aplica).

Tabla 5. Inicio de sesión exitoso

Campo	Detalle
Nombre del caso de uso	CU01.3 – Inicio exitoso
Actor principal	Usuario
Descripción	Autentica y redirige al módulo según rol.
Precondiciones	Usuario registrado y activo.
Postcondiciones	Sesión iniciada y acceso por rol.
RNF asociados	RNF-S01, RNF-R01
Flujo normal	1) Ingresar credenciales válidas. 2) Presionar INGRESAR. 3) Sistema valida y crea sesión. 4) Carga permisos y redirige a dashboard.
Flujos alternos	A1) Usuario sin rol: mostrar error (inconsistencia).

CU02 — REGISTRAR ACTIVO (NUEVO ACTIVO)

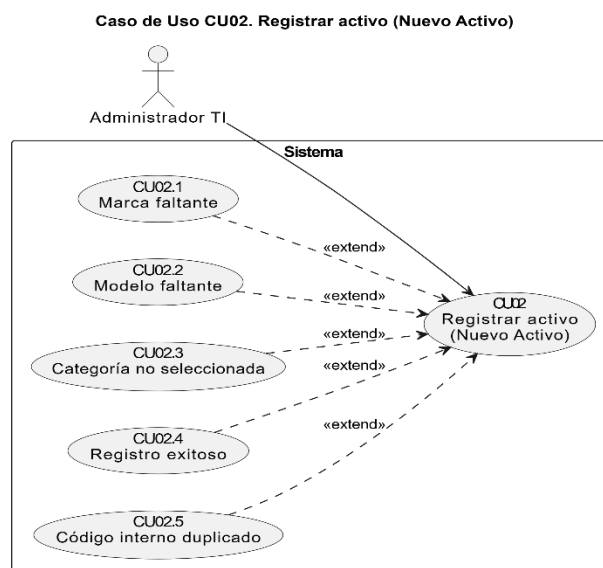


Gráfico 3. Caso de uso registrar activo

Tabla 6. Registrar activo con marca faltante

Campo	Detalle
Nombre del caso de uso	CU02.1 – Datos faltantes (Marca)
Actor principal	Admin TI
Descripción	Valida que “Marca” sea obligatoria.
Precondiciones	Admin TI autenticado; formulario Nuevo Activo.
Postcondiciones	No se guarda el activo.
RNF asociados	RNF-U01, RNF-I01
Flujo normal	1) Completar formulario dejando Marca vacía. 2) Guardar. 3) Validación “Completa este campo”.

Flujos alternos	A1) Mostrar guía para completar correctamente.
-----------------	--

Tabla 7. Registrar activo con modelo faltante

Campo	Detalle
Nombre del caso de uso	CU02.2 – Datos faltantes (Modelo)
Actor principal	Admin TI
Descripción	Valida que “Modelo” sea obligatorio.
Precondiciones	Admin TI autenticado.
Postcondiciones	No se guarda el activo.
RNF asociados	RNF-U01, RNF-I01
Flujo normal	1) Dejar Modelo vacío. 2) Guardar. 3) Validación “Completa este campo”.
Flujos alternos	A1) Reintento del usuario.

Tabla 8. Registrar activo con categoría no seleccionada

Campo	Detalle
Nombre del caso de uso	CU02.3 – Datos faltantes (Categoría)

Actor principal	Admin TI
Descripción	Valida selección de Categoría antes de guardar.
Precondiciones	Admin TI autenticado.
Postcondiciones	No se guarda el activo.
RNF asociados	RNF-U01, RNF-I01
Flujo normal	1) No seleccionar categoría. 2) Guardar. 3) Validación “Completa este campo”.
Flujos alternos	A1) Si faltan categorías, gestionar catálogo (administración).

Tabla 9. Registrar activo exitoso.

Campo	Detalle
Nombre del caso de uso	CU02.4 – Registro exitoso
Actor principal	Admin TI
Descripción	Crea un activo con código interno, serie, marca, modelo, categoría, sede y estado inicial.
Precondiciones	Admin TI autenticado; catálogos disponibles.
Postcondiciones	Activo creado y visible en la matriz de activos.

RNF asociados	RNF-I01, RNF-R01
Flujo normal	1) Completar todos los campos. 2) Guardar. 3) Sistema registra en BD. 4) Muestra activo en listado.
Flujos alternos	A1) Error BD: informar y no guardar.

Tabla 10. Registrar activo con código interno duplicado

Campo	Detalle
Nombre del caso de uso	CU02.5 – Código duplicado
Actor principal	Admin TI
Descripción	Impide guardar un activo con codigo_interno existente (UNIQUE).
Precondiciones	Existe el código en inventario.
Postcondiciones	No se guarda; se muestra error de duplicidad.
RNF asociados	RNF-I01
Flujo normal	1) Ingresar código repetido. 2) Guardar. 3) Sistema valida unicidad. 4) Error y no guarda.
Flujos alternos	A1) Sugerir editar el activo existente.

CU03 — EDITAR ACTIVO / MATRIZ

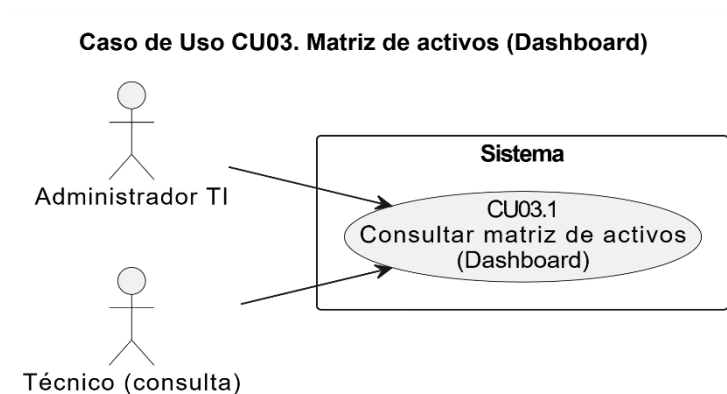


Gráfico 4. Caso de uso matriz de activos

Tabla 11. Consultar matriz de activos (dashboard + tabla)

Campo	Detalle
Nombre del caso de uso	CU03.1 – Consultar matriz de activos
Actor principal	Admin TI / Técnico (consulta)
Descripción	Visualiza totales y listado (código, equipo, serie, categoría, sede, custodio, estado, acciones).
Precondiciones	Usuario autenticado.
Postcondiciones	Datos visibles; acciones habilitadas según rol.
RNF asociados	RNF-R01, RNF-U01

Flujo normal	1) Entrar al dashboard. 2) Ver tarjetas (Total activos, Operativos). 3) Ver tabla con acciones por fila.
Flujos alternos	A1) Sin datos: mostrar “No hay activos registrados”.

CU06 — USUARIOS (LISTAR/CREAR/ELIMINAR)

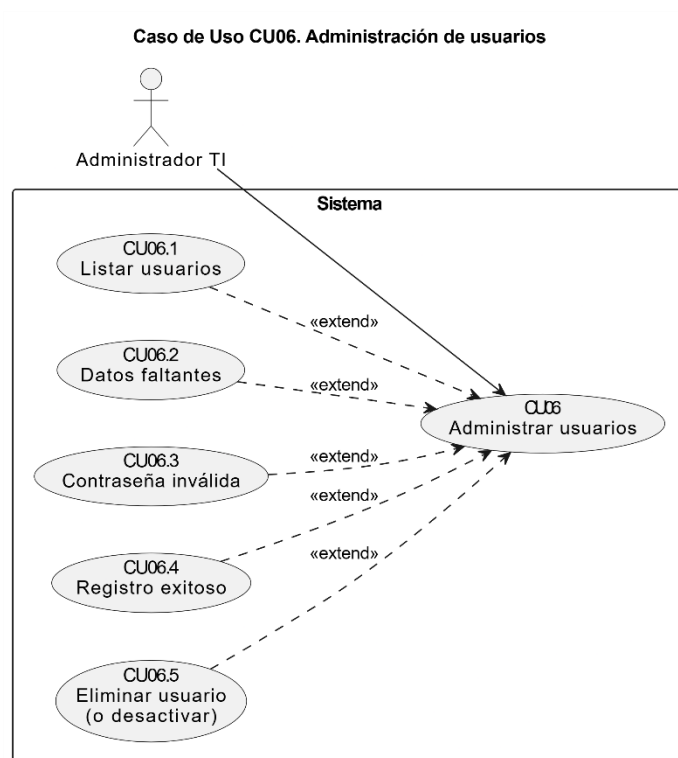


Gráfico 5. Caso de uso administración de usuarios

Tabla 12. Listar usuarios del sistema

Campo	Detalle
Nombre del caso de uso	CU06.1 – Listar usuarios
Actor principal	Admin TI
Descripción	Visualiza usuarios con nombre, email, rol, sede y acciones.
Precondiciones	Admin TI autenticado.
Postcondiciones	Lista cargada y lista para acciones.
RNF asociados	RNF-R01, RNF-U01
Flujo normal	1) Admin abre “Usuarios”. 2) Sistema lista usuarios. 3) Muestra tabla.
Flujos alternos	A1) Sin usuarios: mensaje informativo.

Tabla 13. Registrar usuario con datos faltantes

Campo	Detalle
Nombre del caso de uso	CU06.2 – Datos faltantes (Usuario)
Actor principal	Admin TI

Descripción	Valida campos obligatorios (Nombre completo y Email).
Precondiciones	Formulario “Registrar Usuario” abierto.
Postcondiciones	Usuario no creado; validación “Completa este campo”.
RNF asociados	RNF-U01, RNF-S01
Flujo normal	1) Dejar vacío nombre o email. 2) Guardar. 3) Validación “Completa este campo”.
Flujos alternos	A1) Formato de email inválido: validar formato.

Tabla 14. Registrar usuario con contraseña inválida.

Campo	Detalle
Nombre del caso de uso	CU06.3 – Contraseña inválida
Actor principal	Admin TI
Descripción	Valida longitud mínima de contraseña.
Precondiciones	Formulario abierto.
Postcondiciones	Usuario no creado.
RNF asociados	RNF-S01, RNF-U01

Flujo normal	1) Contraseña < 6. 2) Guardar. 3) Sistema rechaza por regla mínima.
Flujos alternos	A1) Contraseña válida: continuar a CU06.4.

Tabla 15.Registrar usuario exitoso

Campo	Detalle
Nombre del caso de uso	CU06.4 – Registro exitoso
Actor principal	Admin TI
Descripción	Crea usuario con rol, sede y área.
Precondiciones	Roles y sedes disponibles.
Postcondiciones	Usuario creado y visible en listado.
RNF asociados	RNF-S01, RNF-I01
Flujo normal	1) Completar campos. 2) Seleccionar rol y sede. 3) Guardar. 4) Sistema crea y confirma.
Flujos alternos	A1) Email duplicado: error y no guarda.

Tabla 16. Eliminar usuario (administración)

Campo	Detalle
Nombre del caso de uso	CU06.5 – Eliminar usuario
Actor principal	Admin TI
Descripción	Elimina usuario desde acciones (según política del sistema).
Precondiciones	Admin TI autenticado; usuario existe.
Postcondiciones	Usuario eliminado (o desactivado si se aplica baja lógica).
RNF asociados	RNF-S01, RNF-T01
Flujo normal	1) Admin selecciona eliminar. 2) Confirma. 3) Sistema elimina/desactiva. 4) Actualiza lista.
Flujos alternos	A1) Si se requiere conservar trazabilidad, usar desactivación en lugar de borrado.

CU04 — GESTIÓN DE CAMPAMENTO (EDITAR + TRANSFERIR/DEVOLVER)

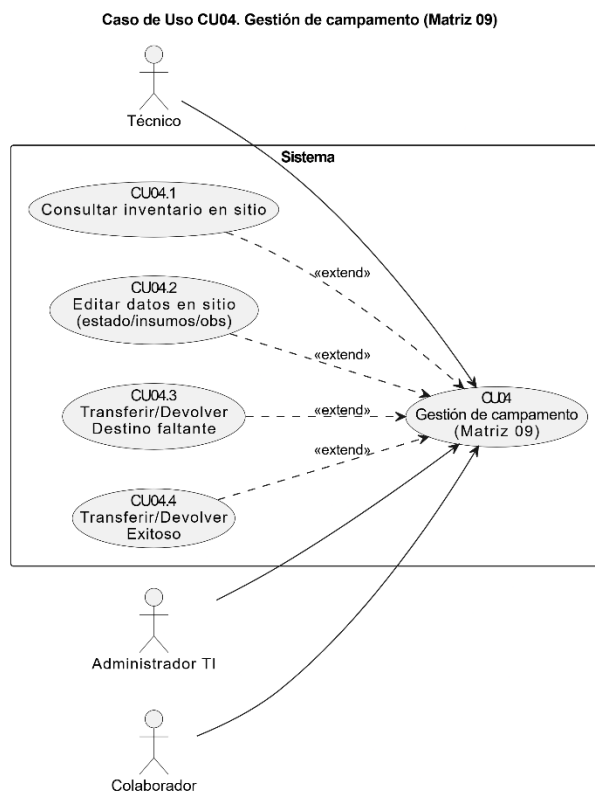


Gráfico 6. Caso de uso gestión de campamentos

Tabla 17. Consultar inventario en sitio (campamento)

Campo	Detalle
Nombre del caso de uso	CU04.1 – Consultar inventario en sitio
Actor principal	Técnico/Colaborador/Admin (según permisos)
Descripción	Muestra activos del campamento con responsable/área, estado e insumos.

Precondiciones	Usuario autenticado; campamento asociado.
Postcondiciones	Tabla visible; acciones disponibles según rol.
RNF asociados	RNF-R01, RNF-U01
Flujo normal	1) Entrar a “Gestión de campamento”. 2) Sistema lista inventario en sitio.
Flujos alternos	A1) Sin activos: mostrar mensaje “No hay activos”.

Tabla 18. Editar datos de campamento (exitoso)

Campo	Detalle
Nombre del caso de uso	CU04.2 – Editar datos en sitio
Actor principal	Usuario con permiso
Descripción	Edita responsable, estado, insumos y observación del equipo en sitio.
Precondiciones	Usuario autenticado; activo visible en campamento.
Postcondiciones	Datos actualizados en tabla.
RNF asociados	RNF-I01, RNF-U01, RNF-T01
Flujo normal	1) Abrir modal “Editar datos de campamento”. 2) Cambiar campos. 3) Guardar cambios. 4) Sistema actualiza y confirma.

Flujos alternos	A1) Error BD: informar y no guardar.
-----------------	--------------------------------------

Tabla 19. Transferir/Devolver con destino faltante

Campo	Detalle
Nombre del caso de uso	CU04.3 – Destino no seleccionado
Actor principal	Usuario con permiso
Descripción	Valida que “Destino (Sede)” sea obligatorio antes de confirmar envío.
Precondiciones	Modal “Transferir/Devolver” abierto.
Postcondiciones	No se transfiere/devuelve; mensaje “Selecciona un elemento de la lista”.
RNF asociados	RNF-U01, RNF-I01
Flujo normal	1) Abrir modal. 2) No seleccionar destino. 3) Confirmar envío. 4) Validación impide continuar.
Flujos alternos	A1) Si no hay sedes, notificar administrador.

Tabla 20. Transferencia/Devolución exitosa

Campo	Detalle
Nombre del caso de uso	CU04.4 – Transferir/Devolver exitoso
Actor principal	Usuario con permiso
Descripción	Registra movimiento a sede destino con motivo y actualiza inventario (el equipo desaparece del sitio).
Precondiciones	Activo existe; destino seleccionado.
Postcondiciones	Movimiento registrado; activo cambia de sede; inventario del sitio se actualiza.
RNF asociados	RNF-T01, RNF-I01
Flujo normal	1) Seleccionar destino. 2) Ingresar motivo. 3) Confirmar. 4) Sistema registra movimiento e actualiza sede/estado.
Flujos alternos	A1) Motivo vacío: permitir o exigir según regla interna.

CU07 — EXPORTACIÓN PDF

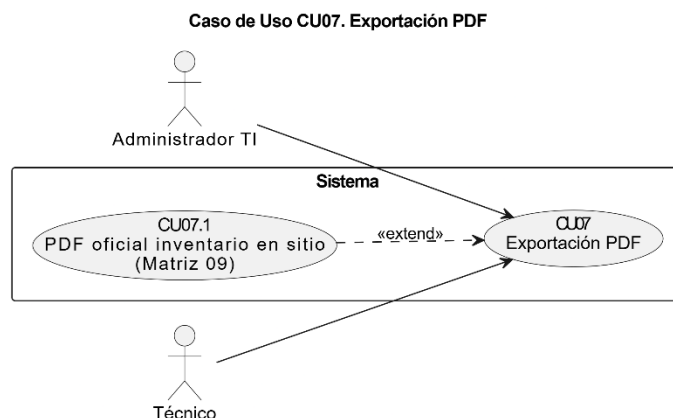


Gráfico 7. Caso de uso Exportación PDF

Tabla 21. Exportar PDF oficial (campamento).

Campo	Detalle
Nombre del caso de uso	CU07.1 – PDF oficial inventario en sitio
Actor principal	Usuario con permiso
Descripción	Genera PDF “oficial” del inventario del campamento.
Precondiciones	Usuario autenticado; existen activos para exportar.
Postcondiciones	PDF generado y entregado al navegador.
RNF asociados	RNF-R01, RNF-S01
Flujo normal	1) Presionar “PDF OFICIAL”. 2) Sistema genera PDF. 3) Descarga/vista.
Flujos alternos	A1) Sin datos: “No hay registros para exportar”.

CU08 — HISTORIAL / TRAZABILIDAD (movimientos)

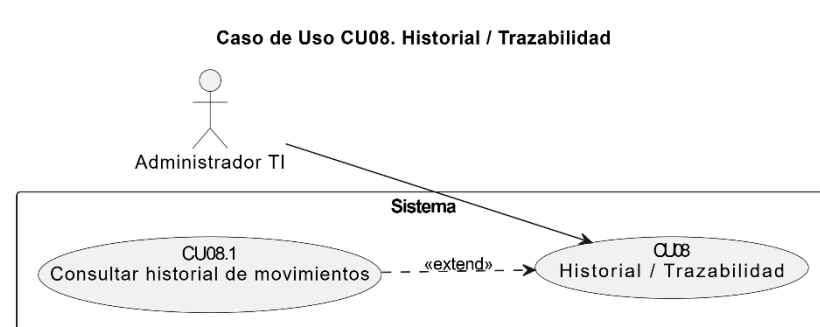


Gráfico 8. Caso de uso historial / trazabilidad

Tabla 22. Consultar historial de movimientos

Campo	Detalle
Nombre del caso de uso	CU08.1 – Consultar historial de movimientos
Actor principal	Admin TI
Descripción	Consulta asignaciones/transferencias/devoluciones/mantenimiento para auditoría.
Precondiciones	Admin autenticado; existen movimientos registrados.
Postcondiciones	Historial visible para trazabilidad.
RNF asociados	RNF-T01, RNF-R01

Flujo normal	1) Abrir módulo historial. 2) Aplicar filtros (activo/usuario/fecha). 3) Sistema consulta BD. 4) Muestra resultados ordenados.
Flujos alternos	A1) Sin resultados: “No hay movimientos” y sugerir revisar filtros.

CU05 — REPORTES DE CAMPO

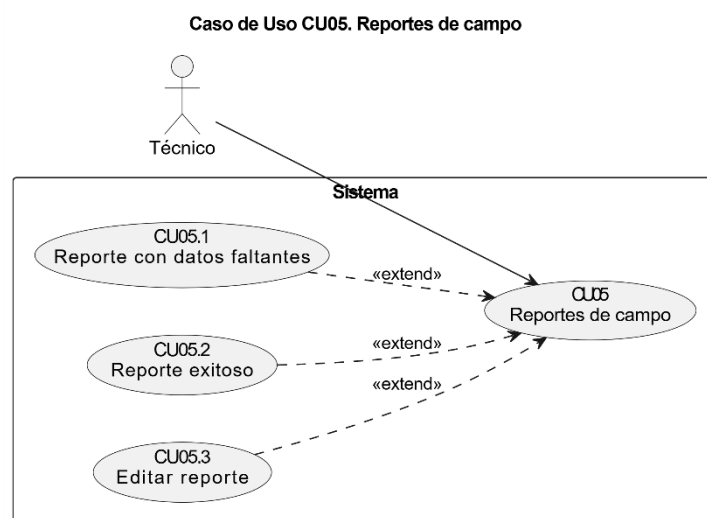


Gráfico 9. Caso de uso reportes de campo

Tabla 23. Registrar reporte de campo con datos

Campo	Detalle
Nombre del caso de uso	CU05.1 – Reporte con datos faltantes
Actor principal	Técnico
Descripción	Valida campos obligatorios del reporte antes de guardar.
Precondiciones	Técnico autenticado; acceso a módulo reportes.

Postcondiciones	Reporte no guardado; se muestran validaciones.
RNF asociados	RNF-U01, RNF-I01
Flujo normal	1) Técnico abre “Nuevo reporte”. 2) Deja campos obligatorios vacíos (ej.: sede/activo/actividad). 3) Guardar. 4) Sistema valida y muestra error/“Completa este campo”.
Flujos alternos	A1) Activo no encontrado: permitir “reporte sin activo” o solicitar alta (según política).

Tabla 24. Registrar reporte de campo exitoso.

Campo	Detalle
Nombre del caso de uso	CU05.2 – Reporte exitoso
Actor principal	Técnico
Descripción	Registra un reporte desde campo en tiempo real (sin Excel).
Precondiciones	Técnico autenticado; sedes/activos disponibles.
Postcondiciones	Reporte guardado y disponible para consulta.

RNF asociados	RNF-U01, RNF-T01
Flujo normal	1) Técnico abre “Nuevo reporte”. 2) Selecciona sede/proyecto. 3) Selecciona activo (por serie/código). 4) Registra actividad/observaciones/estado. 5) Guarda. 6) Sistema confirma y almacena fecha/hora y usuario.
Flujos alternos	A1) Sin conexión: guardar borrador o avisar (si aplica).

Tabla 25. Editar reporte de campo

Campo	Detalle
Nombre del caso de uso	CU05.3 – Editar reporte
Actor principal	Técnico
Descripción	Permite corregir datos del reporte dentro de permisos/tiempo permitido.
Precondiciones	Reporte existe; técnico autorizado.
Postcondiciones	Reporte actualizado; cambios registrados (si aplica).
RNF asociados	RNF-I01, RNF-T01
Flujo normal	1) Técnico abre historial de reportes. 2) Selecciona reporte. 3) Edita campos permitidos. 4) Guarda. 5) Sistema actualiza.

Flujos alternos	A1) Si el reporte está cerrado, bloquear edición y mostrar aviso.
-----------------	---

1.6 Planificación de los Sprint

Para estructurar el desarrollo del sistema web, se aplicó la metodología ágil Scrum dividiendo el trabajo en **sprints de 14 días**. Cada sprint contiene un conjunto de funcionalidades priorizadas del Product Backlog y se valida mediante entregables funcionales al finalizar el ciclo.

El seguimiento y evidencia del avance se realizó mediante **GitHub Projects** (backlog, estados y tareas) y mediante **GitHub** (ramas, commits y merges), donde cada sprint se representó con una rama específica con el formato `sprint-n-<nombre>` y al final del sprint se aplicó el DOD.

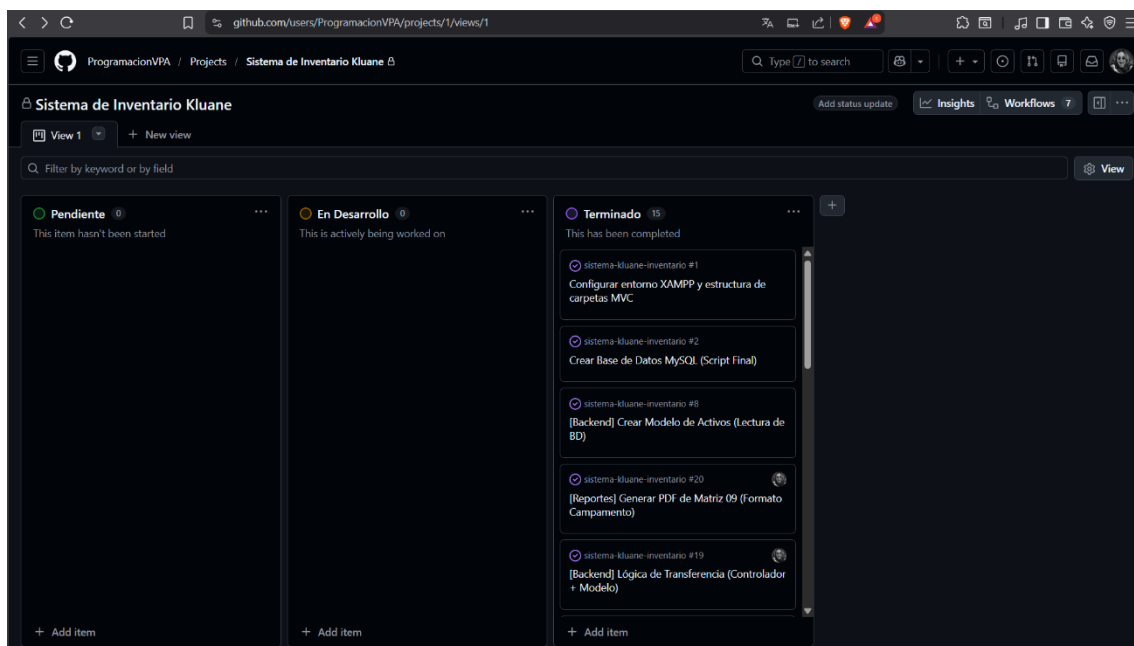


Gráfico 10. Flujo de trabajo frontend y backend

Tabla 26. Planificación general de Sprints

Sprint	Objetivo del sprint	Actividades (Issues – GitHub)	Entregables principales	Estado
Sprint 1 – Preparación y acceso	Preparar entorno, estructura base del proyecto y autenticación	Issue #1 Configurar entorno XAMPP y estructura MVC; Issue #2 Crear BD MySQL (Script final); Issue #3 Conexión PHP a BD; Issue #4 Diseño Login (HTML/Bootstrap); Issue #5 Validación de usuarios	Proyecto base + conexión BD + login funcional + validación de usuarios	Terminado
Sprint 2 – Inventario (Dashboard + Registro)	Implementar módulo central de inventario (lectura y registro)	Issue #8 Modelo de activos (lectura BD); Issue #9 Tabla de inventario en Dashboard; Issue #10 Formulario registro de activos; Issue #11 Lógica de guardado (INSERT)	Matriz de activos en dashboard + formulario “Nuevo Activo” + guardado en BD	Terminado

Sprint 3 – Trazabilidad (Historial + UI)	Implementar historial y mejoras de visualización	Issue #14 Vista Historial de Movimientos; Issue #13 Mostrar columna Responsable en Dashboard	Historial de movimientos + responsable/custodio visible en matriz	Terminado
Sprint 4 – Operación en campo (Matriz 09)	Completar el flujo de campamento: formulario, transferencia y PDF	Issue #17 Crear tabla guia_remision para Matriz 09; Issue #18 Formulario Matriz 09 (Envío a Campo); Issue #19 Lógica de Transferencia (Controlador + Modelo); Issue #20 PDF Matriz 09 (Formato Campamento)	Gestión de campamento (Matriz 09) + transferir/devolver + PDF oficial	Terminado

1.7 Herramientas y software

Para el desarrollo, pruebas y despliegue del sistema web de inventario IT y gestión operativa en campo, se utilizaron herramientas orientadas a garantizar compatibilidad con el entorno de la empresa, trazabilidad del desarrollo y facilidad de mantenimiento. El conjunto de herramientas empleadas se detalla a continuación:

a) Entorno de desarrollo y servidor local

- **XAMPP:** utilizado como entorno local para ejecutar PHP y MySQL durante el desarrollo y pruebas, permitiendo simular el entorno de ejecución del sistema antes del despliegue.
- **Arquitectura MVC (estructura de carpetas):** se organizó el proyecto bajo una estructura tipo MVC para separar lógica de negocio, vistas y controladores, facilitando mantenibilidad y escalabilidad.

b) Backend y persistencia de datos

- **PHP:** lenguaje principal del lado del servidor para implementar la lógica del sistema, validaciones y conexión con la base de datos.
- **MySQL:** motor de base de datos relacional para almacenar usuarios, roles, activos, sedes y movimientos, asegurando centralización e integridad de la información.
- **Script SQL final (phpMyAdmin):** se generó el script final para creación de la base de datos y tablas, garantizando reproducibilidad y facilidad de instalación.

c) Frontend e interfaz de usuario

- **HTML5, CSS3 y JavaScript:** tecnologías utilizadas para construir interfaces web accesibles desde navegadores, con formularios y tablas para inventario, usuarios y operaciones en campo.
- **Bootstrap 5:** framework empleado para diseñar una interfaz responsiva y consistente, optimizada para escritorio y dispositivos móviles.

d) Gestión del proyecto y control de versiones

- **GitHub (repositorio):** utilizado para el control de versiones y respaldo del código fuente, permitiendo trazabilidad de cambios por commits y ramas.
- **GitHub Projects:** empleado para la planificación y seguimiento del trabajo bajo Scrum, gestionando backlog y tareas por estado (Terminado/En progreso), alineado con la evidencia del proyecto.

- **GitHub Issues:** utilizado para documentar tareas técnicas por sprint (configuración de entorno, conexión BD, inventario, historial, transferencias, PDF Matriz 09), conservando evidencia verificable del avance.

1.7.1 Flujo de trabajo con Git

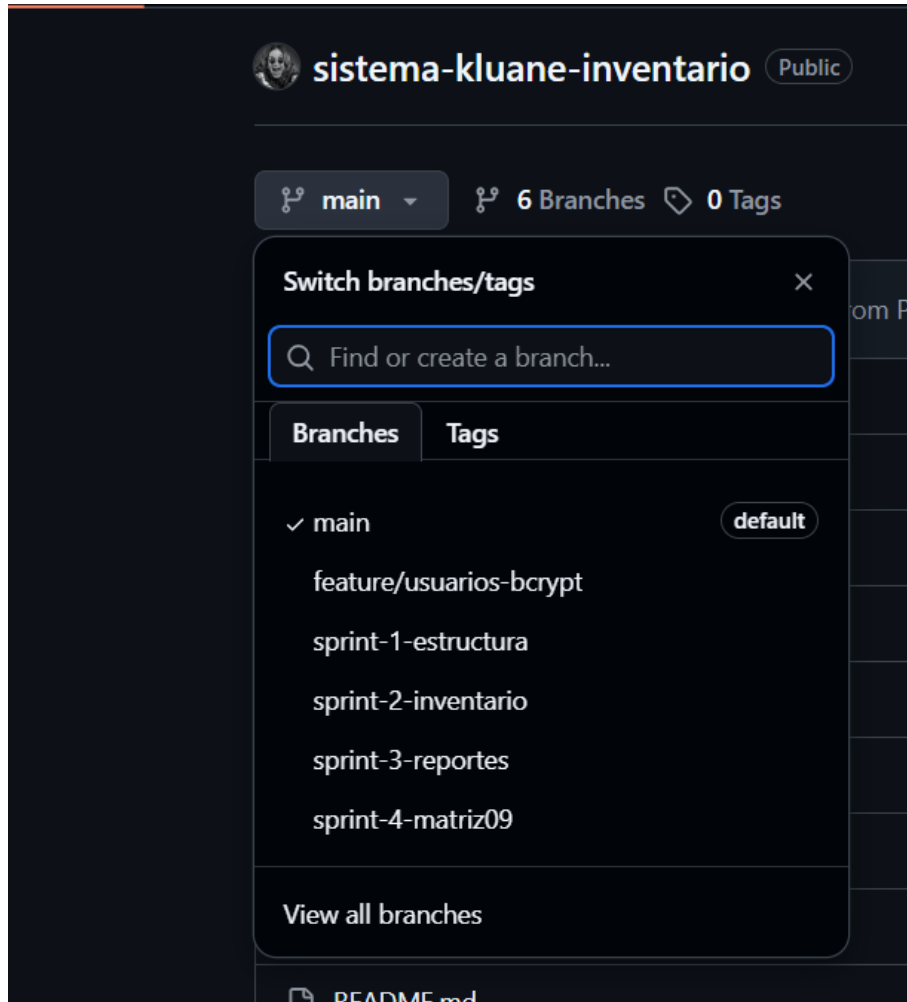


Gráfico 11. Ramificación de github

Rama principal: main

- Contiene la versión **estable** del sistema, lista para demostración, entrega y eventual despliegue.
- **No se trabaja directamente** sobre main; los cambios se integran desde ramas de sprint o ramas de funcionalidad una vez verificados.

- Representa el estado consolidado del proyecto después de completar y validar los entregables de cada sprint.

Ramas de sprint: sprint-*

- Son ramas creadas para desarrollar y consolidar funcionalidades **por iteración Scrum**, manteniendo trazabilidad del avance.
- Cada rama agrupa el trabajo planificado en GitHub Projects, issues para un sprint específico.
- Facilitan la verificación del entregable antes de fusionarlo en main.

Ramas por funcionalidad: feature/*

- Se utilizan cuando un módulo requiere desarrollo aislado o independiente.
- Permiten implementar una funcionalidad completa sin afectar el estado estable del proyecto en main.

1.7.2 Procedimiento de validación antes de integrar a producción

1. Cada nueva funcionalidad o mejora se desarrolla en una rama de **sprint** (sprint-*) o una rama **feature** (feature/*).
2. Al finalizar la implementación, se realizan **commits** con mensajes claros y descriptivos (por ejemplo: “Implementa formulario de registro de activos”).
3. Se ejecutan pruebas funcionales locales para validar:
4. formularios (campos obligatorios y mensajes de validación),
5. persistencia en base de datos (lectura/insert/actualización),
6. control por rol (si aplica).
7. La tarea se marca como **Terminado** en GitHub Projects y queda respaldada con su Issue correspondiente (evidencia del sprint).

8. Si las validaciones son correctas, se realiza el **merge** de la rama del sprint/feature hacia main, dejando la rama principal en estado estable.

1.8 Hardware

Tabla 27. Requisitos mínimos de hardware

Componente	Cliente Web (usuario final)	Entorno de desarrollo (PC del desarrollador)	Servidor/Hosting (producción)
Procesador	Intel i3 / Ryzen 3 o superior	Intel i5 / Ryzen 5 o superior	Administrado por el proveedor (cPanel)
Memoria RAM	4 GB (mínimo)	8 GB (recomendado)	Administrado por el proveedor (cPanel)
Almacenamiento	2 GB libres (uso general)	20–50 GB libres (proyecto, BD, backups)	Administrado por el proveedor (cPanel)
Sistema operativo	Windows / macOS / Linux	Windows / macOS / Linux	Linux (hosting)
Navegador	Chrome / Edge / Firefox actualizado	Chrome / Edge / Firefox actualizado	N/A

Conectividad	Internet estable (acceso web desde sedes/campo)	Internet estable + acceso a GitHub	Acceso privado según configuración
--------------	---	------------------------------------	------------------------------------

1.9 Viabilidad del proyecto

La viabilidad del proyecto se evaluó considerando tres dimensiones: **viabilidad técnica**, **viabilidad económica** y **viabilidad operativa**. Este análisis permite determinar si la solución propuesta puede desarrollarse, mantenerse y adoptarse en el contexto de Kluane Drilling Ecuador, donde actualmente el inventario IT y la consolidación de información se realizaban con procesos manuales y archivos dispersos, generando duplicidad, demoras y falta de trazabilidad.

1.9.1 Viabilidad técnica

El proyecto es **técnicamente viable** debido a que la solución se implementa como un sistema web basado en tecnologías ampliamente utilizadas, compatibles con entornos empresariales de hosting y con documentación disponible para su mantenimiento.

- **Compatibilidad del stack tecnológico:** El sistema se desarrolló con **PHP** y **MySQL**, lo cual es compatible con entornos típicos de despliegue como hosting con **cPanel/phpMyAdmin**, facilitando su implementación en infraestructura web existente.
- **Entorno de desarrollo controlado:** El uso de **XAMPP** y una estructura organizada (MVC) permitió realizar pruebas locales y validar funcionalidades antes de integrar entregables por sprint.

- **Trazabilidad y evidencia técnica:** La gestión mediante **GitHub Projects/Issues** proporciona evidencia clara de avance por funcionalidades (inventario, historial, transferencias, PDF), permitiendo control de versiones y respaldo del código fuente.
- **Funcionalidades críticas implementadas:** El módulo de inventario (dashboard, formulario de activos y lógica de guardado), la trazabilidad (historial de movimientos), la lógica de transferencia y la generación de PDF para Matriz 09 demuestran que la solución puede operar correctamente y responder a necesidades reales del proceso.

En conjunto, estas condiciones indican que la arquitectura web propuesta es adecuada, escalable a mejoras futuras y alineada al objetivo de centralizar información y reducir trabajo manual en la gestión del inventario IT y operaciones asociadas.

1.9.2 Viabilidad económica

El proyecto es **económicamente viable** porque utiliza herramientas y tecnologías de bajo costo o de uso gratuito, reduciendo la inversión necesaria para su desarrollo y adopción.

- **Uso de software libre y herramientas gratuitas:** El desarrollo con PHP/MySQL y la gestión del proyecto con GitHub Projects/Issues evita costos de licenciamiento, ya que se apoya en herramientas accesibles.
- **Costos operativos controlados:** La infraestructura necesaria para ejecutar el sistema puede mantenerse en un hosting estándar (cPanel/MySQL) y no requiere equipos especializados para su operación, ya que se accede desde navegador.
- **Optimización del tiempo del área TI:** Al reemplazar consolidación manual (archivos Excel y procesos repetitivos), se reduce carga administrativa, lo cual representa un ahorro indirecto de tiempo y esfuerzo operativo.

En consecuencia, el proyecto puede implementarse con una inversión moderada y sostenida, maximizando el beneficio operativo al centralizar inventario y procesos relacionados.

1.9.3 Viabilidad operativa

El proyecto es **operativamente viable** debido a que la solución está diseñada para ser utilizada por roles reales del proceso (Administrador TI, Técnico y Colaborador), con validaciones de formularios, control por roles y módulos enfocados en las necesidades del trabajo diario.

- **Facilidad de uso:** La interfaz web basada en formularios y tablas permite registrar, consultar y actualizar información sin requerir conocimientos avanzados, mejorando la adopción por el personal.
- **Soporta operación en diferentes sedes/campamentos:** La gestión por sedes y la funcionalidad de transferencia/devolución permiten controlar activos distribuidos en proyectos y campamentos, alineándose al contexto operativo del negocio.
- **Mejora la trazabilidad:** La existencia de historial de movimientos (asignación/transferencia/devolución/mantenimiento) permite seguimiento y auditoría, resolviendo una de las limitaciones principales del uso de archivos dispersos.
- **Capacitación y continuidad:** Debido a que la solución es web y se apoya en tecnologías estándar, la capacitación requerida es mínima y el mantenimiento puede realizarse por el área TI con documentación y control de versiones.

Por lo anterior, se concluye que la solución es viable para ser utilizada en el entorno real de la empresa, contribuyendo a mejorar el control del inventario IT, reducir errores y fortalecer la toma de decisiones con información centralizada y confiable.

Capítulo II

Construcción del Sistema

2.1 Alcance

El sistema de inventario IT y control operativo en campo abarca el desarrollo completo de una aplicación web destinada a la **gestión centralizada de activos tecnológicos**, la administración de usuarios por roles y la operación por sede/campamento, incluyendo trazabilidad de movimientos y generación de documentos PDF.

Incluye:

- **Frontend:** Implementado con **HTML, CSS, JavaScript y Bootstrap**, encargado de proporcionar interfaces amigables y responsivas para el inicio de sesión, la matriz de activos (dashboard), el registro de activos, la gestión de usuarios y el módulo de campamento (Matriz 09), permitiendo consulta y operación desde distintos dispositivos.
- **Backend:** Desarrollado en **PHP** bajo una estructura tipo MVC, responsable de manejar la lógica de negocio, validar formularios, gestionar sesiones por rol y realizar las operaciones CRUD (inventario/usuarios), además de la lógica de transferencia/devolución y la generación de documentos PDF.
- **Base de datos:** Diseñada e implementada en **MySQL**, almacenando la información de **activos, usuarios, roles, sedes, categorías y historial de movimientos**, asegurando integridad y soporte para auditoría operativa mediante trazabilidad de cambios/movimientos.
- Este alcance se limita a los procesos relacionados con la **gestión de inventario IT**, la operación por campamento y los movimientos de equipos entre sedes, sin contemplar la integración con sistemas externos (ERP, compras, contabilidad o MDM), aunque el

sistema queda preparado para ampliaciones futuras mediante nuevos módulos y reportes adicionales

2.2 Análisis de diseño

Se derivan de los requisitos desde las historias de usuario.

A partir de las historias de usuario definidas en el levantamiento de requisitos, se concretaron los siguientes elementos funcionales del sistema:

Tabla 28. Historias de usuario

CÓDIGO	DETALLE
HU01:	Como usuario , quiero iniciar sesión , para acceder al sistema según mi rol (Administrador/Técnico/Colaborador).
HU02:	Como Administrador TI , quiero registrar activos IT , para centralizar el inventario y reemplazar matrices Excel.
HU03:	Como Administrador TI , quiero visualizar la matriz de activos en el dashboard , para consultar estado, sede y responsable del inventario.
HU04:	Como Administrador TI , quiero gestionar usuarios y roles , para controlar permisos y accesos al sistema.
HU05:	Como usuario en campamento , quiero editar datos operativos del equipo (estado/insumos/observación) , para mantener actualizado el inventario en sitio.
HU06:	Como Administrador/Técnico , quiero transferir o devolver equipos entre sedes , para mantener trazabilidad y control de ubicación.

HU07:	Como Administrador/Técnico , quiero generar un PDF oficial (Matriz 09) , para documentar el inventario en campamento y soportar control interno.
HU08:	Como Administrador TI , quiero consultar el historial de movimientos , para auditoría y trazabilidad del inventario.

2.3 Mapa de navegación

El sistema se organiza mediante una navegación estructurada por **módulos y roles**, permitiendo que cada usuario acceda únicamente a las pantallas habilitadas según su perfil (Administrador, Técnico o Colaborador). La navegación prioriza el acceso rápido al inventario central (dashboard), a la operación en sitio/campamento (Matriz 09), a los movimientos de equipos (transferir/devolver) y a la generación de documentos PDF oficiales para control interno.

Tabla 29. Mapa de navegación del sistema

Sección / Pantalla	Descripción	Rol con acceso
Login	Pantalla de autenticación para ingreso al sistema con validación de usuarios.	Admin / Técnico / Colaborador
Dashboard (Matriz de Activos)	Vista principal del inventario central: listado de activos y estado general; muestra campos clave como equipo/código/serie/categoría/sede/responsable.	Admin / Técnico (consulta) / Colaborador

		(consulta limitada)
Registrar Nuevo Activo	Formulario para crear activos en el inventario (alta), con validaciones y guardado en base de datos.	Admin TI
Historial de Movimientos	Pantalla para consultar trazabilidad de movimientos (asignaciones/transferencias/devoluciones), orientada a auditoría operativa.	Admin TI (principal) / Técnico (si se habilita consulta)
Usuarios del Sistema	Listado de usuarios con rol, sede y área; permite administración de accesos.	Admin TI
Registrar Usuario / Gestión de Roles	Formulario para crear/actualizar usuarios y asignar roles, controlando permisos del sistema.	Admin TI
Gestión de Campamento (Matriz 09 / EC-IT-F-09)	Módulo operativo en sitio: permite controlar equipos en campamento, editar estado/insumos/observaciones y preparar documentación de control.	Técnico / Admin TI (según rol)
Transferir / Devolver Equipo	Flujo para mover equipos entre sedes/campamentos, registrando el movimiento y su motivo (trazabilidad).	Técnico / Admin TI

		(según permisos)
Generación de PDF (Matriz / PDF oficial)	Generación de documento PDF para control interno (Matriz 09 / formato campamento), utilizado como evidencia operativa.	Admin TI / Técnico
Cerrar sesión	Opción para finalizar sesión y proteger el acceso al sistema.	Todos los roles

2.4 Lógica de negocio.

2.4.1 Ilustración de Arquitectura del Proyecto (Arquitectura de Aplicaciones)

El sistema de inventario IT se organiza por **roles y procesos**, donde cada rol accede a funcionalidades específicas: Administrador (gestión completa), Técnico (operación en campo/campamento y movimientos) y Colaborador (consulta limitada).

Los procesos principales del negocio abarcan el **registro y administración de activos**, la **operación en campamento (Matriz 09)**, el **registro de movimientos** (asignación/devolución/mantenimiento) y la **generación de documentos PDF** para evidencia operativa.

La trazabilidad se garantiza mediante el almacenamiento de eventos y observaciones en la tabla historial_movimientos, asociando cada registro a un activo y un usuario responsable.

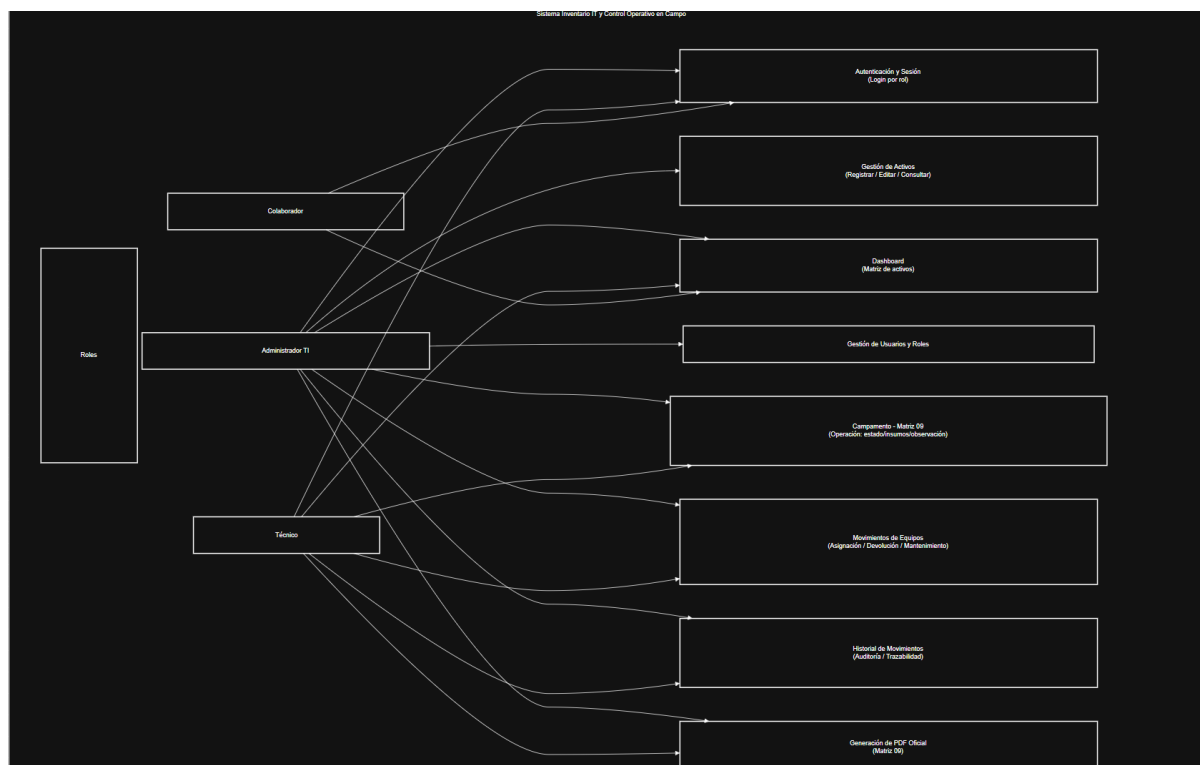


Gráfico 12. *Arquitectura de Aplicaciones*

2.4.2 Ilustración de Arquitectura Tecnológica (Frontend / Backend / Base de Datos)

La arquitectura tecnológica se implementa en tres capas:

Frontend web (interfaz de usuario), **Backend** (lógica de negocio en PHP MVC) y **Base de datos MySQL** (persistencia y trazabilidad).

Las consultas (búsqueda, filtros, listados) se realizan mediante solicitudes HTTP desde el frontend al backend, el cual responde con datos estructurados para renderizar el dashboard y módulos operativos (por ejemplo, activos por sede/campamento).

La integridad se apoya en restricciones: `activos.codigo_interno` es único y existen claves foráneas entre `activos`–`categorías`–`sedes`–`usuarios` e `historial`–`activos`–`usuarios`.

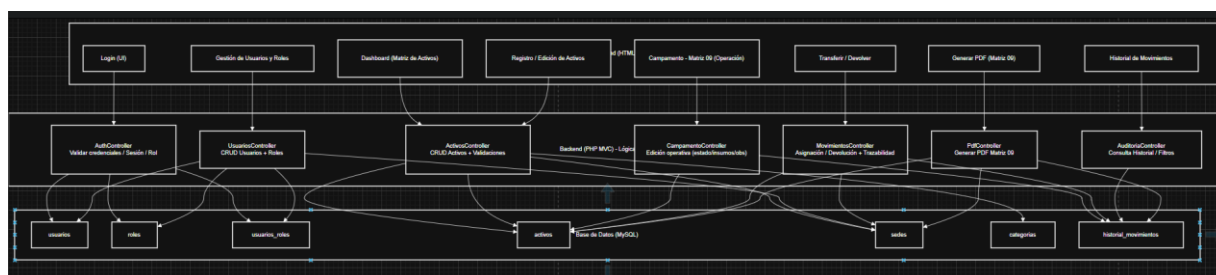


Gráfico 13. *Arquitectura Tecnológica*

2.4.3 Reglas de negocio

- Al **registrar un activo**, el sistema almacena la información base del equipo (código interno, serie, marca, modelo, categoría, sede, estado y observaciones) y aplica validaciones para evitar inconsistencias. El inventario se centraliza en la base de datos, reemplazando el uso de hojas de cálculo dispersas.
- Al **editar un activo o actualizar información operativa en campamento (Matriz 09)**, se actualizan campos operativos como estado, observación y requerimientos de insumos, permitiendo que la información refleje la situación real en sitio y fortaleciendo la consistencia del inventario en tiempo real.
- Al **transferir o devolver un equipo**, el sistema actualiza la ubicación del activo y registra el evento en el historial de movimientos, incluyendo motivo y fecha del cambio. Este registro funciona como evidencia para auditoría operativa, ya que permite identificar cuándo ocurrió el movimiento y quién ejecutó la acción.
- Las **búsquedas y filtros** ejecutados en el dashboard o en módulos operativos solicitan información al backend, el cual procesa la petición, consulta la base de datos y devuelve resultados estructurados para actualizar la interfaz. De esta forma, el usuario visualiza información consolidada por sede, categoría, estado o responsable.
- Finalmente, la **generación del PDF oficial (Matriz 09)** consolida el inventario del campamento/sede y produce un documento descargable que sirve como soporte de

control interno. Este documento se utiliza como evidencia documental del inventario en campo y contribuye al cumplimiento de procedimientos internos de verificación.

2.4.4 Diagrama de Secuencia

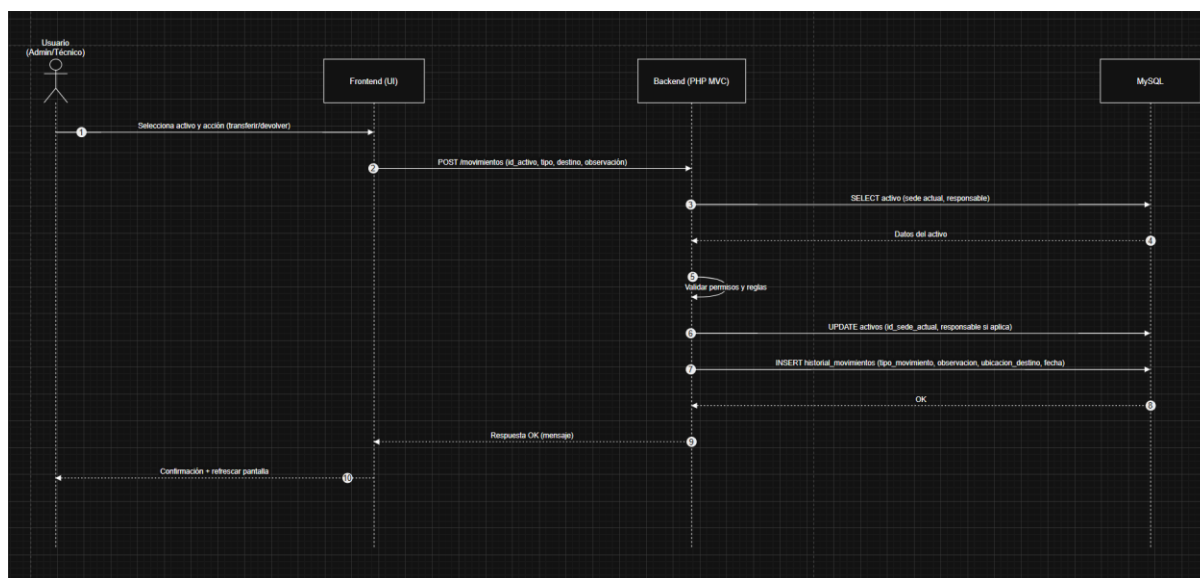


Gráfico 14. Diagrama de Secuencia

2.5 Diseño de la arquitectura de la aplicación.

La solución propuesta adopta una **arquitectura web en tres capas** (presentación, negocio y datos), orientada a centralizar la gestión del inventario IT y soportar la operación en campo mediante módulos específicos por rol. La separación por capas permite mantener responsabilidades claras: la interfaz se enfoca en la experiencia de usuario, la lógica de negocio se concentra en el servidor y la persistencia se gestiona de forma estructurada en una base de datos relacional.

La **capa de presentación** está compuesta por una aplicación web desarrollada con **HTML, CSS, JavaScript y Bootstrap**, accesible desde navegadores de escritorio y dispositivos móviles. Esta capa ofrece las pantallas de autenticación (login), visualización de inventario (dashboard/matriz de activos), registro y edición de activos, administración de

usuarios, operación de campamento (Matriz 09), gestión de movimientos (transferir/devolver) y generación de documentos PDF.

La **capa de negocio** se implementa en **PHP** bajo un patrón **MVC**, concentrando las reglas operativas del sistema. En esta capa se ejecuta la autenticación, la administración de sesión y permisos por rol (Administrador, Técnico y Colaborador), la validación de formularios y la lógica asociada a operaciones críticas: creación/actualización de inventario, administración de usuarios, registro de movimientos y generación del PDF oficial. La comunicación entre la presentación y el backend se realiza mediante solicitudes HTTP/HTTPS, permitiendo que el usuario reciba respuestas actualizadas para listados, filtros y confirmaciones de operaciones.

La **capa de datos** se gestiona mediante **MySQL**, donde se almacenan de manera estructurada las entidades principales del dominio: activos, usuarios, roles, sedes y categorías. Para garantizar trazabilidad operativa, el sistema registra eventos relevantes en una tabla de historial de movimientos, lo que facilita auditoría y seguimiento de cambios relacionados con asignaciones, devoluciones y acciones de mantenimiento.

El procesamiento inicia cuando el usuario interactúa con el frontend y ejecuta una acción (por ejemplo, registrar un activo o actualizar información en campamento). La interfaz envía la solicitud al backend PHP, el cual aplica validaciones y reglas de negocio, consulta o actualiza la base de datos y devuelve una respuesta estructurada para refrescar la vista correspondiente. El control de accesos se realiza por roles, asegurando que operaciones sensibles —como administración de usuarios, transferencias/devoluciones o emisión de PDF— se encuentren disponibles únicamente para perfiles autorizados.

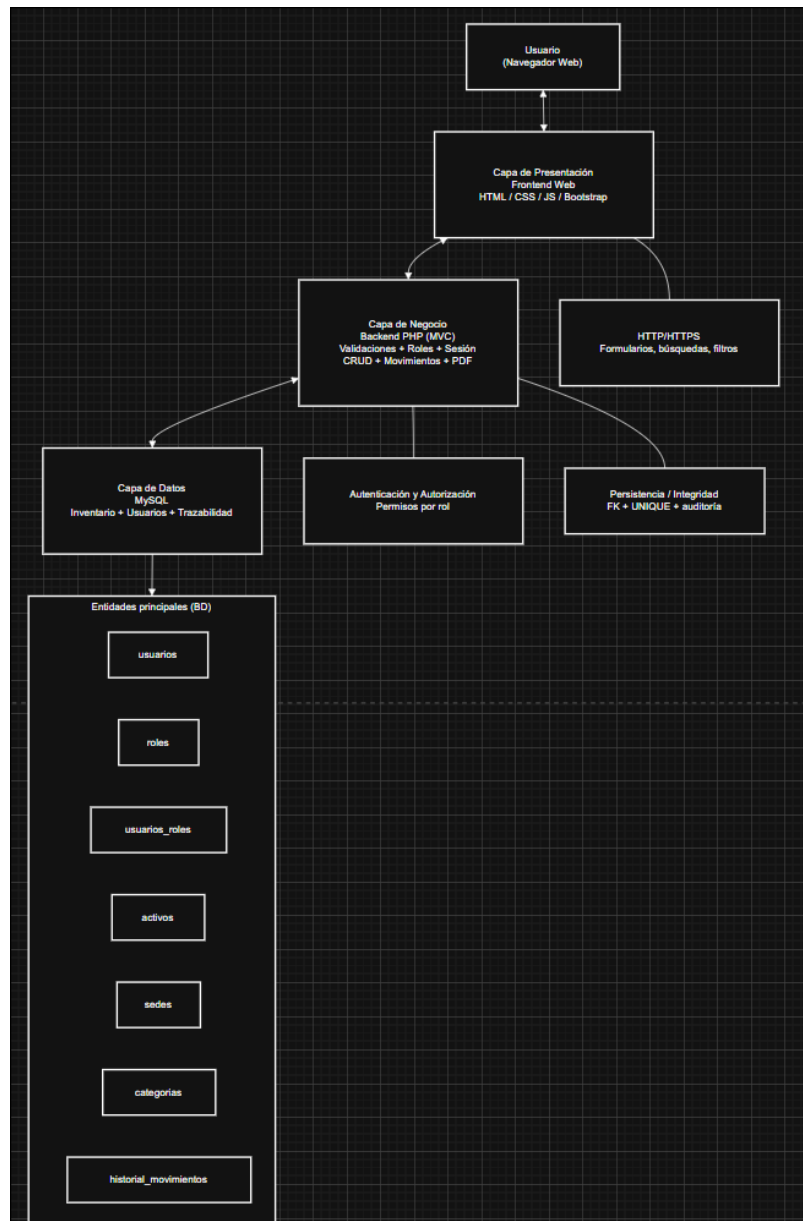


Gráfico 15. Arquitectura de la aplicación

2.5.1 Arquitectura tecnológica y despliegue (entorno de operación)

El sistema se desarrolla y prueba en un entorno local controlado mediante herramientas de servidor web y base de datos, y posteriormente se despliega en un entorno de hosting administrado. En producción, el sistema opera sobre un servidor web que hospeda la aplicación PHP (estructura MVC) y se conecta a una base de datos MySQL para la persistencia de la información del inventario y la trazabilidad.

El acceso de los usuarios se realiza desde navegador, lo que permite operación distribuida desde distintas sedes o proyectos. Además, el control de versiones del código fuente se gestiona mediante GitHub y el seguimiento del desarrollo se organiza en GitHub Projects, asegurando trazabilidad del avance por iteraciones y facilitando mejoras futuras. Esta organización respalda la mantenibilidad del sistema, ya que los cambios pueden implementarse y verificarse de forma controlada antes de integrarse a la versión estable.

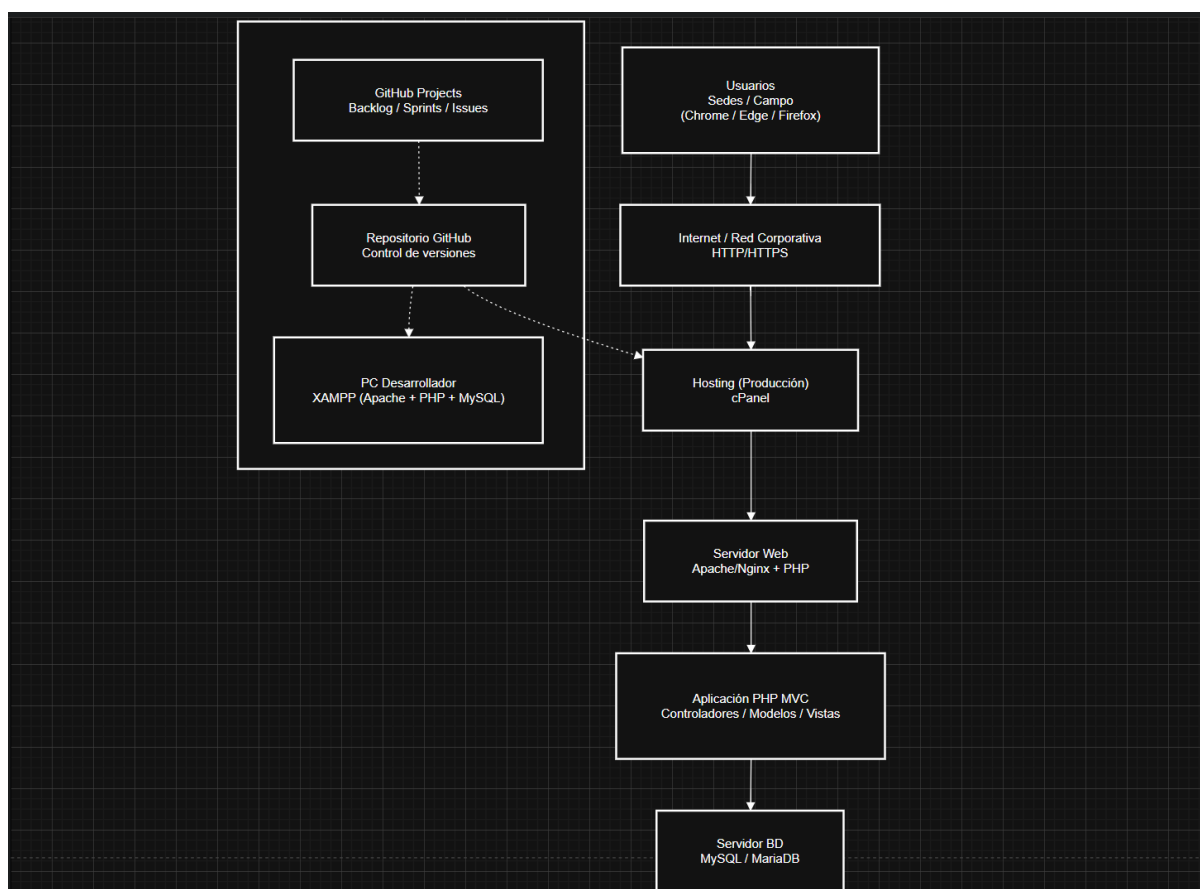


Gráfico 16. Arquitectura de despliegue

2.6 Modelo de Base de Datos (BDD).

El modelo de base de datos del sistema de inventario IT fue diseñado en **MySQL** con el objetivo de centralizar la información del inventario, administrar usuarios y roles, controlar sedes/campamentos y asegurar **trazabilidad** mediante un historial de movimientos. El esquema se compone de tablas normalizadas que separan entidades principales (activos,

usuarios, sedes, categorías y roles) y una tabla de auditoría operativa (historial_movimientos) que registra eventos relevantes para control interno y revisiones posteriores

La estructura mantiene **integridad referencial** mediante claves foráneas entre activos–categorías–sedes–usuarios y entre historial–activos–usuarios, lo cual permite relacionar cada movimiento con el activo afectado y el usuario responsable que ejecutó la acción. Además, se incorpora una restricción de unicidad sobre el código interno del activo para prevenir registros duplicados dentro del inventario.

2.6.1 Entidades principales

A continuación, se describen las tablas principales del sistema y su propósito dentro del modelo:

- **usuarios:** almacena los datos de acceso y perfil del usuario (nombre, email, contraseña cifrada), así como su rol y sede asociada. El campo email es único para evitar duplicidad de cuentas
- **roles:** define los perfiles del sistema (Administrador, Técnico y Colaborador) que determinan permisos de acceso a módulos y operaciones.
- **sedes:** registra las ubicaciones operativas (Matriz Quito, Proyecto Macas, Proyecto Warintza), usadas para controlar inventario por ubicación y operaciones de campo.
- **categorías:** clasifica los activos (Laptops, Periféricos, Herramientas IT), lo cual facilita filtros, reportes y control por tipo de equipo.
- **activos:** entidad central del inventario. Contiene identificación del activo (codigo_interno único), datos técnicos (serie, marca, modelo), estado (Operativo, Dañado, Mantenimiento, Baja), sede actual, responsable, observaciones y si requiere insumos.

- **historial_movimientos:** tabla de trazabilidad/auditoría. Registra movimientos o acciones relevantes (Asignación, Devolución, Mantenimiento) vinculando activo, usuario responsable, fechas, observación y ubicación de destino.
- **usuarios_rols:** tabla puente para asignación de roles a usuarios con fecha de asignación. Aunque usuarios ya contiene id_rol, esta tabla sirve como soporte para registrar cambios de rol en el tiempo o mantener historial de asignaciones.

2.6.2 Relaciones del modelo

El modelo implementa las siguientes relaciones principales:

1. **categorias (1) → (N) activos:** una categoría puede clasificar múltiples activos; cada activo pertenece a una sola categoría (activos.id_categoria).
2. **sedes (1) → (N) activos:** una sede puede contener múltiples activos; cada activo tiene una sede actual (activos.id_sede_actual).
3. **usuarios (1) → (N) activos:** un usuario puede ser responsable de varios activos; un activo puede tener responsable opcional (activos.id_usuario_responsable).
4. **roles (1) → (N) usuarios:** un rol puede asignarse a varios usuarios; cada usuario tiene un rol principal (usuarios.id_rol).
5. **sedes (1) → (N) usuarios:** una sede puede tener varios usuarios asociados; el usuario puede tener sede asignada (usuarios.id_sede).
6. **activos (1) → (N) historial_movimientos:** un activo puede registrar múltiples movimientos; cada movimiento corresponde a un activo (historial_movimientos.id_activo).
7. **usuarios (1) → (N) historial_movimientos:** un usuario responsable puede generar múltiples registros en el historial (historial_movimientos.id_usuario_responsable).

8. **usuarios (1) → (N) usuarios_rols y roles (1) → (N) usuarios_rols:** permite asignar roles a usuarios con fecha.

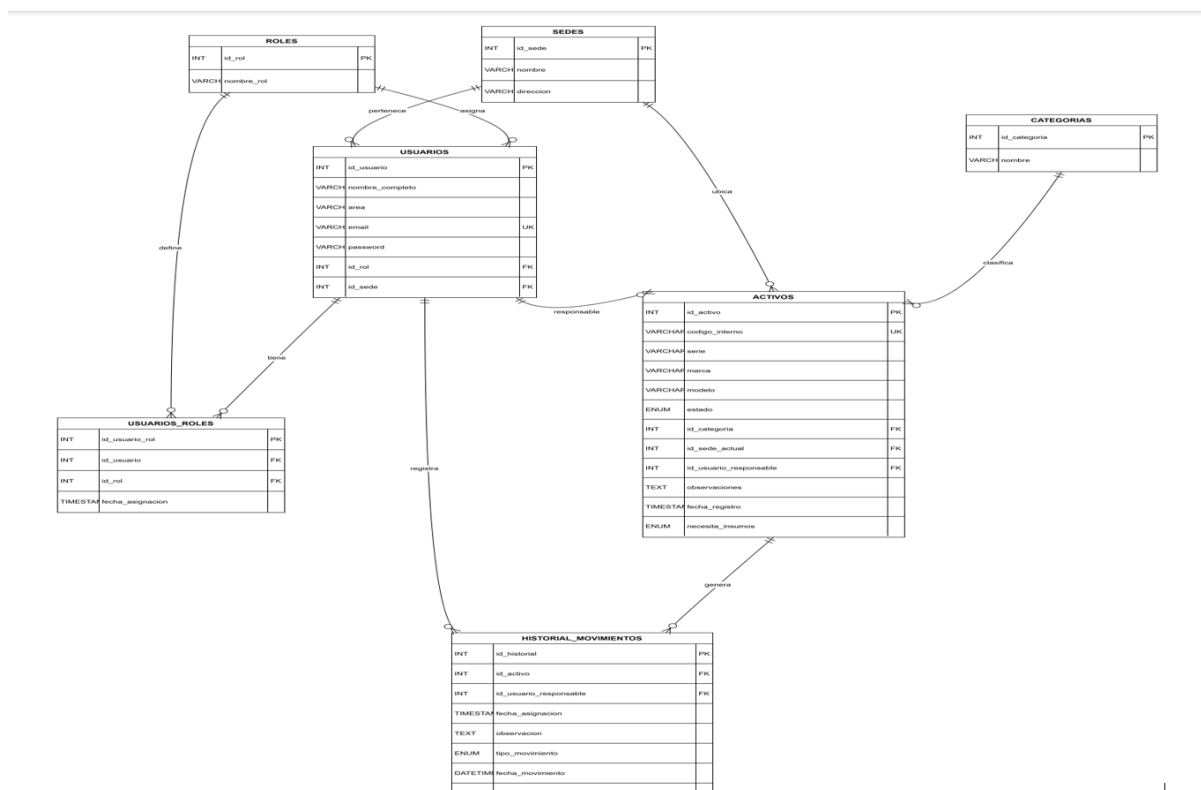


Gráfico 17. Modelo de base de datos

2.7 Requisitos Funcionales

Tabla 30. Requisitos funcionales

Código	Detalle
RF01	Autenticación de usuarios por correo y contraseña, acceso por roles.
RF02	CRUD de activos (crear, editar, eliminar).
RF03	Gestión de campamento: editar estado/insumos/responsable del activo en sitio.
RF04	Transferir/Devolver activos entre sedes con motivo.
RF05	Administración de usuarios (crear, editar, eliminar).
RF06	Exportación a PDF (matriz y PDF oficial por campamento).
RF07	Registro y consulta de historial de movimientos para trazabilidad.
RF08	Registro/edición/consulta de reportes de campo (formularios web).

2.8 Requisitos no Funcionales

Tabla 31. Requisitos no funcionales

Código	Detalle
RNF-S01	Seguridad: roles, sesiones seguras, contraseñas cifradas.
RNF-U01	Usabilidad: validaciones claras, interfaz responsiva.

RNF-I01	Integridad: unicidad de códigos, consistencia de sedes/usuarios.
RNF-T01	Trazabilidad: historial de movimientos y motivos.
RNF-R01	Rendimiento: carga rápida de tablas y consultas.

Capítulo III

Pruebas y Estabilización

3.1 Reuniones con Product Owner

3.1.1 Objetivo de las reuniones

Las reuniones con el **Product Owner (PO)** se realizaron para validar que el sistema desarrollado responda a las necesidades reales del área de TI de Kluane Drilling Ecuador y para asegurar que cada incremento entregado al final de cada sprint cumpla con la **Definición de Hecho (DoD)**. Estas sesiones permitieron revisar módulos implementados, levantar observaciones, priorizar ajustes y aprobar entregables antes de integrarlos a la rama estable del proyecto. Este enfoque es coherente con el uso del PO como enlace entre necesidades del negocio y el equipo técnico, tal como se presenta en el modelo de tesis guía.

3.1.2 Participantes y roles

En el proyecto se consideraron los siguientes participantes para las reuniones de seguimiento y validación:

- **Scrum Master:** Ing. Christian Roberto Tapia Gaibor.
- **Product Owner:** Ing. Iván Pérez.
- **Desarrollador:** Victor Paul Alvarez Alvarez.

Se realizaron reuniones virtuales

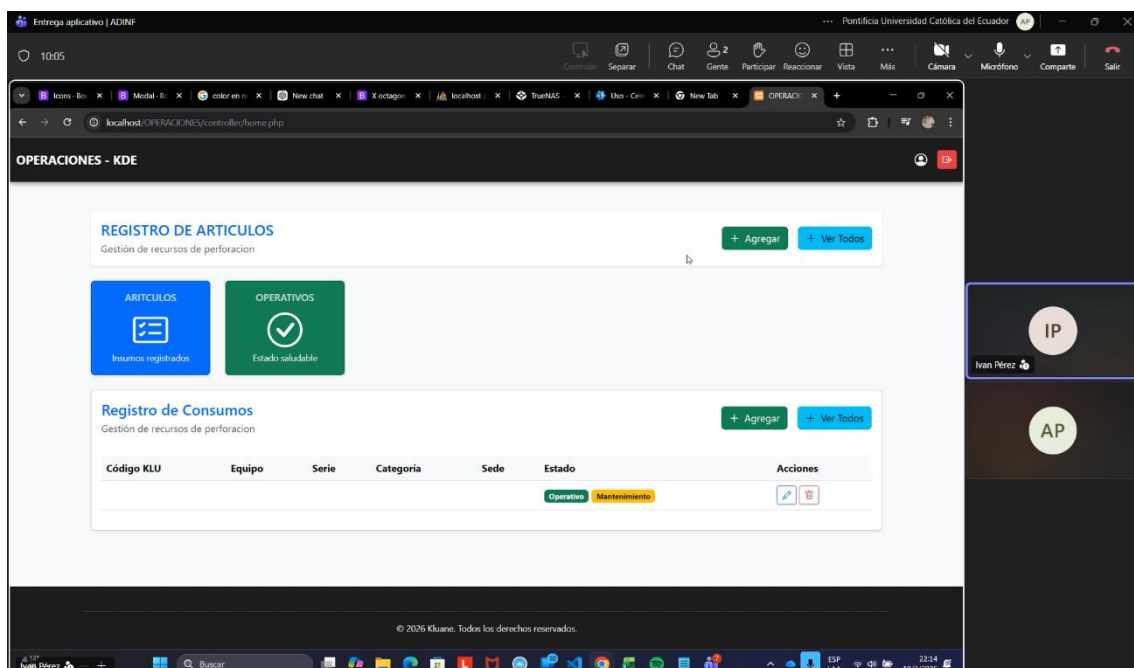


Gráfico 18. Reunión Presentación

En el transcurso del desarrollo del sistema web se presentaron observaciones relacionadas con el diseño de interfaces, ajustes de navegación por rol, validaciones de formularios y consistencia de reportes (PDF oficial). Dichas observaciones fueron revisadas en reuniones con el Product Owner y solventadas mediante mejoras incrementales, quedando evidenciadas en las siguientes capturas del sistema.

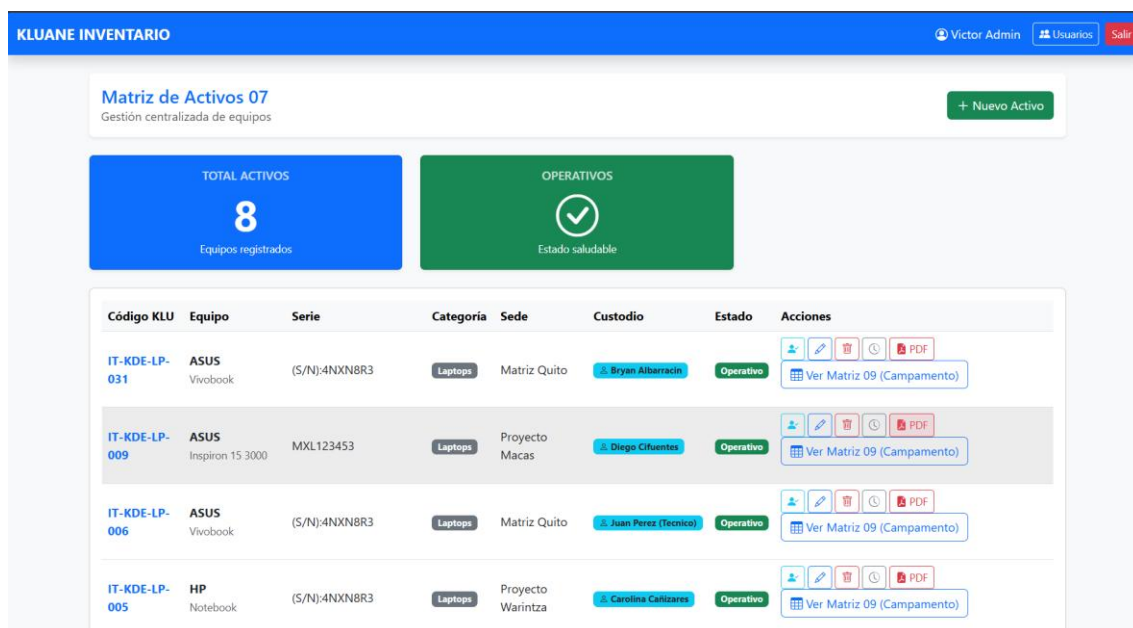


Gráfico 19. Pantalla de inicio (dashboard – matriz de activos)

Registrar Nuevo Activo

Código Interno (Etiqueta): Ej: KLU-LAP-055

Número de Serie: SN...

Marca: Ej: Dell

Modelo: Ej: Latitude S420

Categoría: Computación

Sede / Ubicación: Matriz Quito

Estado Inicial: Operativo

Cancelar Guardar Activo

Gráfico 20. Ingreso de un nuevo activo (formulario de registro)

Registrar Nuevo Activo

Código Interno (Etiqueta): IT-KDE-LP-024

Número de Serie: N... Completa este campo

Marca: Ej: Dell

Modelo: Ej: Latitude 5420

Categoría: Computación

Sede / Ubicación: Matriz Quito

Estado Inicial: Operativo

Cancelar Guardar Activo

Gráfico 21. Formulario de registro de activos con validaciones

Registrar Usuario

Nombre Completo:

Email:

Contraseña: Mínimo 6 caracteres. Se guardará segura.

Rol: Administrador (dropdown menu open showing: -- Seleccionar --, Administrador, Tecnico, Colaborador)

Sede: Matriz Quito

Guardar Cancelar

Gráfico 22. Asignación de rol y sede en el registro de usuarios

Código KLU	Equipo	Serie	Categoría	Sede	Custodio	Estado	Acciones
IT-KDE-LP-032	ASUS Vivobook	(S/N):4NXN8R3	Laptops	Matriz Quito	Victor Admin	Operativo	Ver Matriz 09 (Campamento)

Gráfico 23. Confirmación de registro exitoso y visualización en la matriz de activos

KLUANE DRILLING ECUADOR S.A.							
MATRIZ DE GESTIÓN DE EQUIPOS TECNOLÓGICOS EN CAMPAMENTO							
PROYECTO: Proyecto Macas FECHA IMP: 2026-02-11 06:47							
Total Operativo	Necesita Reparación	Laptops	Radio				
1	0	1	0				
N	Equipo	Serie	Responsable	Área	Fecha	Ubicación	Estado
1	Laptops	MXL123453	Diego Cifuentes	Logística	2026-02-09 09:34:23	Proyecto Macas	Operativo

Gráfico 24. Generación de PDF oficial (Matriz 09) del inventario en campamento

Volver

Gestión de Person

Usuarios del Sistema

Nuevo Usuario

Nombre	Email	Rol	Sede	Acciones
Bryan Albarracin	bryan@kluane.com	Admin	Matriz Quito	
Carolina Cañizares	carito@kluane.com	Colaborador	Proyecto Warintza	
Diego Cifuentes	logistica.macas@kluane.com	Técnico	Proyecto Macas	
Ivan Perez	ivan@kluane.com	Admin	Matriz Quito	
Juan Perez (Tecnico)	juan.perez@kluane.com	Técnico	Proyecto Macas	
Maria Lopez (RRHH)	maria.lopez@kluane.com	Técnico	Matriz Quito	
Prueba logistico	prueba@kluane.com	Técnico	Proyecto Macas	
Victor Admin	admin@kluane.com	Admin	Matriz Quito	

Gráfico 25. Gestión de usuarios del sistema

3.2 Pruebas y estabilización

Objetivo

Siguiendo la metodología Scrum, las pruebas y la estabilización del sistema se realizaron de forma **iterativa en cada sprint**, garantizando que cada incremento del producto fuera funcional y estable antes de su integración a la rama principal. Este enfoque permitió **detectar y corregir errores de manera temprana**, optimizar el rendimiento de consultas y mejorar la experiencia del usuario (validaciones, mensajes y navegación por rol), asegurando un flujo de trabajo ágil y controlado.

En el contexto del sistema de inventario IT, las pruebas se enfocaron especialmente en: autenticación y control de roles, operaciones CRUD de activos y usuarios, operación en campamento (Matriz 09), transferir/devolver con trazabilidad, generación de PDF oficial y consulta del historial de movimientos, respaldado por las tablas activos, usuarios, roles, sedes, categorías e historial_movimientos.

3.2.1 Proceso de pruebas en Scrum

En cada sprint, una historia de usuario solo se consideró finalizada al cumplir con la **Definición de Hecho (Definition of Done)**, que incluyó:

- **Ejecución de pruebas funcionales manuales** por módulo (login, dashboard, formularios y operaciones críticas).
- **Ejemplo:** Se verificó el inicio de sesión ingresando credenciales válidas e inválidas y comprobando el acceso por rol (Administrador/Técnico/Colaborador). Esto valida la autenticación contra la tabla usuarios y la relación con roles.

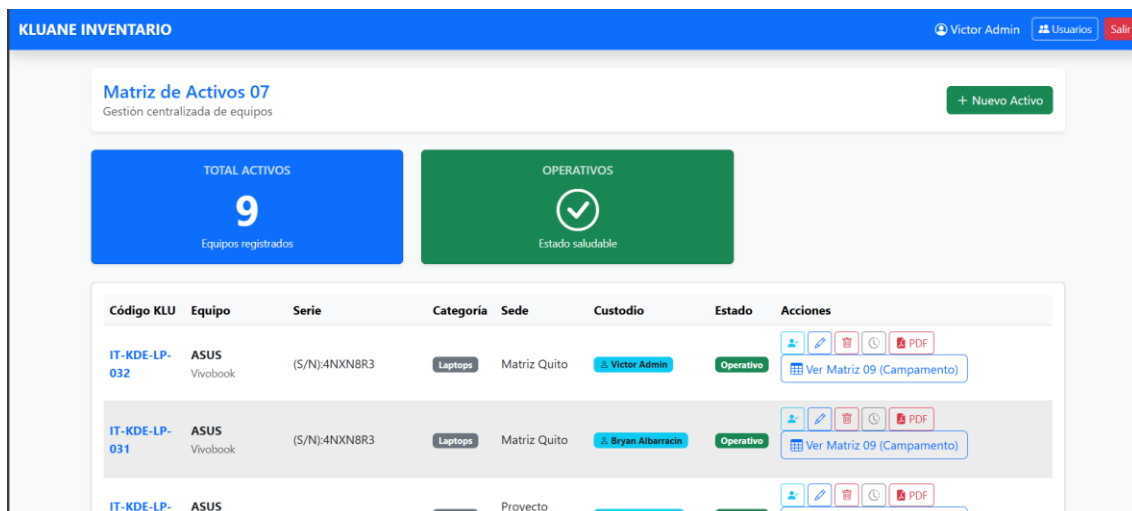


Gráfico 26. Prueba de inicio de sesión

- **Validación de integridad de datos** en la base MySQL (verificación de inserciones/actualizaciones y consistencia de relaciones).
- **Ejemplo:** Se validó en MySQL que cada movimiento (asignación/devolución/mantenimiento) cree un registro en `historial_movimientos` asociados al activo y al usuario responsable, confirmando auditoría operativa.

id_historial	id_activo	id_usuario_responsable	fecha_asignacion	observacion	tipo_movimiento	fecha_movimiento	ubicacion_destino
1	6	3	2026-02-06 10:11:14	Entrega completo de insumos	Asignacion	2026-02-08 20:58:46	Desconocido
2	2	3	2026-02-06 10:12:08	Prueba	Asignacion	2026-02-08 20:58:46	Desconocido
3	2	3	2026-02-06 10:15:11	completo	Asignacion	2026-02-08 20:58:46	Desconocido
4	6	2	2026-02-06 12:09:48	Coller etc ect	Asignacion	2026-02-08 20:58:46	Desconocido
5	8	3	2026-02-06 13:32:20	Idassdsd	Asignacion	2026-02-08 20:58:46	Desconocido
6	9	2	2026-02-08 16:26:16	prueba	Asignacion	2026-02-08 20:58:46	Desconocido
7	6	4	2026-02-08 21:01:08	Cambio realizado en sitio. cambio		2026-02-08 21:01:08	2
8	2	4	2026-02-08 21:02:55		Asignacion	2026-02-08 21:02:55	Desconocido
9	6	4	2026-02-08 21:16:42	Transferencia enviada por Diego Cifuentes. Motivo...		2026-02-08 21:16:42	1
10	6	3	2026-02-08 21:18:34		Asignacion	2026-02-08 21:18:34	Desconocido
11	6	3	2026-02-08 21:26:33		Asignacion	2026-02-08 21:26:33	2
12	6	4	2026-02-08 21:34:26	Cambio realizado en sitio. mouse		2026-02-08 21:34:26	2
13	8	4	2026-02-08 21:35:35		Asignacion	2026-02-08 21:35:35	2

Gráfico 27. Verificación de trazabilidad en base de datos (historial de movimientos).

- Validación visual y funcional de la interfaz (PDF oficial)
- **Ejemplo:** En el frontend se comprobó que al presionar el botón “**PDF OFICIAL**” (Matriz 09) se genere el documento con los datos correctos del campamento (activos filtrados por sede, estado, insumos/observación). El listado proviene de activos y la trazabilidad se soporta con `historial_movimientos`.

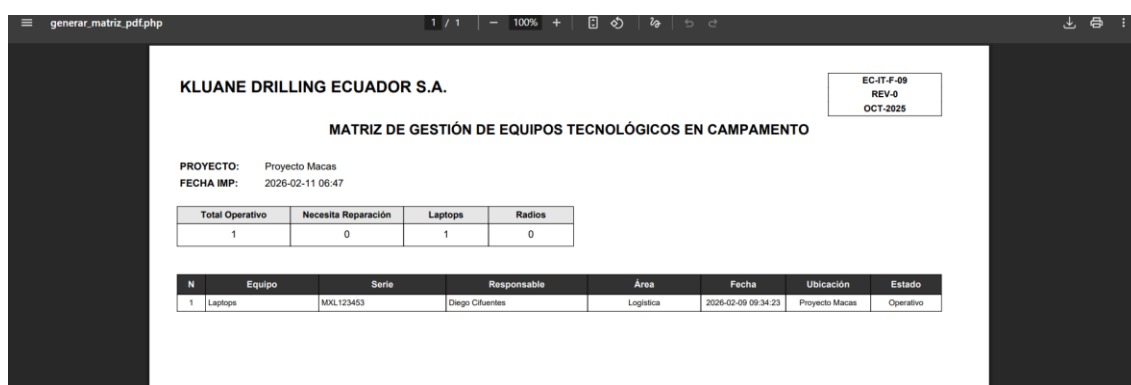


Gráfico 28. Ejecución de prueba de generación de PDF oficial

3.3 Tipos de pruebas realizadas

Frontend (HTML/Bootstrap/JS)

- **Pruebas funcionales:** validación de botones, formularios (campos requeridos), navegación por módulos y respuesta de tablas (listados).
- **Pruebas de usabilidad:** revisión de mensajes claros (errores/confirmaciones), consistencia visual, uso en diferentes tamaños de pantalla (responsive con Bootstrap).

Backend (PHP MVC)

- **Pruebas de base de datos:** validación de inserción, actualización y consulta de datos en MySQL (activos/usuarios/historial).

- **Pruebas de endpoints o flujos del backend:** verificación de respuestas del controlador (login, CRUD activos, CRUD usuarios, transferir/devolver, historial, PDF).
- **Pruebas de integración:** comprobación de interacción correcta entre controladores, modelos y vistas; validando que una acción en UI impacte correctamente en MySQL y quede trazada.

3.3.1 Herramientas utilizadas

- **Navegador (Chrome/Edge/Firefox):** pruebas funcionales y validación visual.
- **phpMyAdmin:** verificación de datos, consultas y consistencia del historial de movimientos.
- **Logs del servidor (Apache/PHP en XAMPP o cPanel):** seguimiento de errores, depuración y monitoreo de ejecución.
- **GitHub / GitHub Projects:** registro de incidencias, seguimiento por sprint y evidencia de cierre de tareas.

3.3.2 Planilla de pruebas Sprint

Tabla 32. Planilla de pruebas por sprint (Resumen)

Sprint	Tarea / Módulo	Descripción	Tipo de prueba	Responsable	Resultado esperado	Estado
S1	Autenticación	Validar login, credenciales inválidas/válidas y sesión por rol	Funcional / Seguridad	Victor P. Alvarez	Acceso correcto según rol o	OK

					error controlado	
S1	BD inicial	Creación de tablas y relaciones (activos, usuarios, roles, sedes, historial)	DB / Integridad	Victor P. Alvarez	Script ejecuta sin errores; relaciones correctas	OK
S2	Dashboard	Carga de matriz de activos, filtros básicos y visualización	Funcional / UI	Victor P. Alvarez	Listado consistente con BD	OK
S2	Registro Activo	Validaciones + guardado de activo	Funcional / DB	Victor P. Alvarez	Inserción correcta; código interno único	OK
S3	Historial	Consulta de movimientos y evidencia de trazabilidad	Funcional / DB	Victor P. Alvarez	Historial muestra movimientos por activo/usuario	OK
S3	Usuarios/Roles	Crear/editar usuarios, asignar rol y sede	Funcional / Seguridad	Victor P. Alvarez	Control de permisos consistente	OK

S4	Campamento (Matriz 09)	Editar estado/insumos/observación y reflejo en inventario	Funcional / Integración	Victor P. Alvarez	Cambios guardados y visibles	OK
S4	Transferir/Devolver	Actualiza sede del activo y registra historial con motivo	Funcional / DB	Victor P. Alvarez	Cambio de sede + registro en historial	OK
S4	PDF Oficial	Generación PDF con datos correctos por sede/campamento	Funcional / Reporte	Victor P. Alvarez	PDF descargable con información correcta	OK

3.3.3 Estabilización

En las últimas iteraciones antes de la entrega:

- Se corrigieron errores detectados en pruebas manuales (validaciones de formularios, mensajes de error, navegación).
- Se depuró el backend PHP (controladores y modelos) y se optimizaron consultas a MySQL para mejorar tiempos de carga del dashboard.
- Se revisó y validó la correcta generación del **PDF oficial (Matriz 09)**, verificando estructura y consistencia de datos del campamento.
- Se garantizó la trazabilidad registrando movimientos en `historial_movimientos` asociados a activos y usuarios responsables.

- Se ajustó la experiencia de usuario: mejoras visuales en Bootstrap, responsividad y consistencia de menús según rol.

Conclusiones

El desarrollo e implementación del sistema web de inventario IT para Kluane Drilling Ecuador permitió centralizar la gestión de activos tecnológicos, reemplazando el uso de múltiples archivos Excel y reduciendo problemas de duplicidad e inconsistencias en la información.

La arquitectura en tres capas (presentación, negocio y datos) facilitó la separación de responsabilidades entre la interfaz web, la lógica en PHP (MVC) y la base de datos MySQL, lo que contribuyó a un sistema más mantenible y escalable para futuras mejoras.

El control de acceso por roles y la trazabilidad mediante el historial de movimientos fortalecieron el control operativo, permitiendo auditoría y seguimiento de cambios sobre los equipos, incluyendo asignaciones, devoluciones y acciones operativas en campamento.

Finalmente, el módulo de campamento (Matriz 09) y la generación de PDF oficial aportaron evidencia formal del inventario en sitio, mejorando el control interno y agilizando el trabajo del área de TI al disponer de información confiable y actualizada.

Recomendaciones

Se recomienda mantener un **ciclo de actualización continua** de las dependencias y librerías utilizadas en el frontend (Bootstrap/JavaScript) y en el backend (PHP), con el fin de asegurar compatibilidad con nuevas versiones y aplicar parches de seguridad cuando corresponda.

Es conveniente **documentar y versionar** todo el código fuente en GitHub, aplicando una estrategia de ramas definida (por ejemplo, rama principal estable y ramas de desarrollo por sprint o funcionalidad), de modo que se garantice trazabilidad, control de cambios y facilidad de mantenimiento evolutivo.

Asimismo, se sugiere implementar **pruebas sistemáticas** (unitarias e integración) para validar nuevas funcionalidades antes de publicarlas en producción, reduciendo la probabilidad de regresiones en módulos críticos como autenticación, registro/edición de activos, transferencias/devoluciones, generación de PDF e historial de movimientos. Este enfoque permite estabilizar incrementos de manera controlada al cierre de cada sprint.

También es recomendable configurar un **entorno de pruebas o preproducción (staging)** en el hosting (cPanel) por ejemplo, mediante un subdominio o carpeta independiente con una base de datos de prueba para validar cambios sin afectar la operación diaria del sistema en producción. Esta práctica facilita revisiones con el Product Owner y permite desplegar mejoras con menor riesgo.

En cuanto al rendimiento, se aconseja realizar **revisiones periódicas** de consultas y cargas del sistema (dashboard, filtros y listados), optimizando el modelo de datos e índices si se detectan cuellos de botella. De igual manera, se puede considerar la aplicación de

estrategias de optimización (paginación, filtros eficientes y reducción de consultas repetitivas) para mejorar tiempos de respuesta en escenarios de uso concurrente.

Finalmente, es fundamental adoptar un **plan de mantenimiento preventivo** del entorno donde se ejecuta el sistema (servidor/hosting), incluyendo respaldos programados de la base de datos, monitoreo básico y capacitación periódica a usuarios clave (Administrador TI y Técnicos), con el fin de asegurar continuidad operativa, correcto uso de los módulos y aprovechamiento sostenido de la solución desarrollada.

Referencias bibliográficas

En su N., & Dirige, S. (s. f.). Sommerville. Scalahed.com. Recuperado el 25 de julio de 2025, de

https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf

Barzanallana, R. (s. f.). Desarrollo de aplicaciones web: Lenguaje PHP y MySQL.

Universidad de Murcia. Recuperado el 25 de julio de 2025, de

https://www.um.es/docencia/barzana/DAWEB/2017_18/daweb-tema-16-php-mysql.html

Apache Friends. (s. f.). *XAMPP installers and downloads for Apache Friends*. Recuperado el 11 de febrero de 2026, de <https://www.apachefriends.org/>

Bootstrap. (s. f.). *Get started with Bootstrap (v5.3)*. Recuperado el 11 de febrero de 2026, de <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

GitHub. (s. f.). *About Projects*. Recuperado el 11 de febrero de 2026, de

<https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>