**Universidad de Las Américas**

**Facultad de Ingeniería y Ciencias Aplicadas**

**Ingeniería de Software**

***Sistema web responsive para seguimiento y control de activos tecnológicos de una Universidad***

**Isaac Gabriel Cerda Sánchez**

**Esteban Andrés Enríquez**

**Ing. Víctor Gómez Regalado**

***23 de enero del 2026***

**Quito, Ecuador**

Contenido

[Resumen 2](#_Toc219715685)

[Abstract 4](#_Toc219715686)

[1. Introducción 6](#_Toc219715687)

[**1.1.** **Identificación y descripción del problema o necesidad** 6](#_Toc219715688)

[**1.2.** **Descripción de la organización** 14](#_Toc219715689)

[2. Análisis de posibles soluciones 15](#_Toc219715690)

[**2.1.** **Identificación y selección de la mejor solución** 16](#_Toc219715691)

[**2.2.** **Impacto del proyecto en la sociedad** 32](#_Toc219715692)

[3. Objetivos 34](#_Toc219715693)

[**3.1.** **Objetivo General** 34](#_Toc219715694)

[**3.2.** **Objetivos Específicos** 34](#_Toc219715695)

[4. Alcance 35](#_Toc219715696)

[**4.1.** **Alcance de la solución seleccionada** 35](#_Toc219715697)

[**4.2.** **Limitaciones y restricciones del proyecto** 40](#_Toc219715698)

[5. Planificación y costos del proyecto 45](#_Toc219715699)

[6. Descripción de estudios realizados 53](#_Toc219715700)

[7. Desarrollo del proyecto 56](#_Toc219715701)

[**7.1.** **Diseño de la solución** 56](#_Toc219715702)

[**7.2.** **Desarrollo de la solución** 61](#_Toc219715703)

[**7.3.** **Pruebas y evaluación de la solución** 75](#_Toc219715704)

[**7.4.** **Resultados y Discusión.** 88](#_Toc219715705)

[**7.5.** **Implicaciones éticas** 89](#_Toc219715706)

[8. Conclusiones y Recomendaciones 91](#_Toc219715707)

[9. Trabajo futuro 93](#_Toc219715708)

[10. Referencias bibliográficas 95](#_Toc219715709)

[11. Anexos 96](#_Toc219715710)

**Índice de Gráficos**

[Figure 1. Organigrama del área de Tecnología de la Información. 15](#_Toc219933615)

[Figure 2. Diagrama de contenedores de la Alternativa 1: arquitectura del sistema de control de activos tecnológicos. Arquitectura: 20](#_Toc219933616)

[Figure 3. Diagrama de contenedores de la arquitectura del sistema móvil de control y seguimiento de activos tecnológicos. 25](#_Toc219933617)

[Figure 4. Diagrama de casos de uso del Sistema de Activos de la Universidad de Las Américas. 36](#_Toc219933618)

[Figure 5. Plan de sprints para la implementación del sistema de activos tecnológicos. 49](#_Toc219933619)

[Figure 6.Arquitectura nivel 1 (Contexto) 56](#_Toc219933620)

[Figure 7. Arquitectura nivel 2 (Contenedores) 57](#_Toc219933621)

[Figure 8. Arquitectura nivel 3 Backend 58](#_Toc219933622)

[Figure 9. Arquitectura nivel 3 Frontend. 60](#_Toc219933623)

[Figure 10. Diagrama de Clases del Dominio A continuación, se detalla la responsabilidad de las entidades en cada microservicio: 61](#_Toc219933624)

[Figure 11. Página del Frontend con el módulo de AssetService 62](#_Toc219933625)

[Figure 12.Página del Frontend con el módulo de Movements 63](#_Toc219933626)

[Figure 13. Modulo de activos en funcionamiento 69](#_Toc219933627)

[Figure 14. Modulo de movimientos en funcionamiento 70](#_Toc219933628)

[Figure 15. Modulo de mantenimientos en funcionamiento 71](#_Toc219933629)

[Figure 16. Módulo de dashboard en funcionamiento 72](#_Toc219933630)

[Figure 17. Archivo descargado para subir a SAP 73](#_Toc219933631)

[Figure 18. Módulo de usuarios en funcionamiento 74](#_Toc219933632)

[Figure 19. Backlog en Azure Boards 75](#_Toc219933633)

**Índice de Tablas**

[Table 1. Análisis FODA de la gestión de activos tecnológicos en la Universidad de Las Américas 14](#_Toc219933705)

[Table 2. Comparación de alternativas Web responsive y aplicación móvil (Flutter) según ISO/IEC 9126. 31](#_Toc219933706)

[Table 3. Modelo de procesos de casos de uso para el sistema de activos tecnológicos 37](#_Toc219933707)

[Table 4. Modelo de requerimientos funcionales y no funcionales del sistema de activos tecnológicos 42](#_Toc219933708)

[Table 5. Resumen de herramientas y costos aproximados para desarrollo y gestión de proyectos de activos tecnológicos. 51](#_Toc219933709)

[Table 6. Resumen de costos del proyecto de activos tecnológicos con y sin uso de infraestructura de una Universidad. 53](#_Toc219933710)

[Table 7. Épica 1 69](#_Toc219933711)

[Table 8. Épica 2 70](#_Toc219933712)

[Table 9. Épica 3 71](#_Toc219933713)

[Table 10. Épica 4 72](#_Toc219933714)

[Table 11. Épica 5 73](#_Toc219933715)

[Table 12. Épica 6 73](#_Toc219933716)

[Table 13. Tabla final: Product Backlog Consolidado 74](#_Toc219933717)

Resumen

Esta investigación aborda la falta de trazabilidad operativa en la gestión de activos tecnológicos de una institución con múltiples sedes, donde el control diario se apoya en hojas de cálculo, correos y documentos físicos, generando información fragmentada, errores y demoras. En este contexto, los procesos de devolución presentan tiempos elevados por la necesidad de validar manualmente qué equipos tiene cada usuario y su ubicación, registrándose un SLA promedio de 133 horas.

El objetivo del trabajo fue desarrollar un sistema web responsive que centralice la información y automatice los registros de movimientos, mantenimientos y documentación institucional, reduciendo la dependencia de controles manuales y habilitando analítica para la gestión. La metodología combinó levantamiento y validación de requerimientos con actores operativos, diseño arquitectónico con el modelo C4 y un enfoque de microservicios, y desarrollo iterativo con Scrum en sprints de dos semanas. La solución se implementó sobre una base de datos SQL Server y un backend distribuido en servicios especializados (Activos/DataWarehouse, Movimientos, Mantenimientos y Plantillas/Actas), con arquitectura limpia por capas y consumo de APIs desde un cliente web.

Como resultados, se obtuvo una plataforma funcional validada mediante revisiones iterativas y pruebas (unitarias, integración, usabilidad y aceptación), que fortalece la trazabilidad al vincular historial del activo, responsables, ubicaciones, actas en PDF y evidencias de mantenimiento. Se concluye que el sistema mejora el control operativo y la calidad de la información; no obstante, la verificación completa del impacto en SLA requiere acordar formalmente la definición del indicador, su fuente de datos y un plan de pruebas de aceptación en el entorno operativo.

Palabras clave: gestión de activos; trazabilidad; microservicios; Scrum; SQL Server; actas PDF; dashboards; integración SAP; SysAid.

Abstract

This research addresses the lack of operational traceability in the management of technological assets across multiple campuses, where daily control relies on spreadsheets, emails, and paper-based records, leading to fragmented information, errors, and delays. In this context, return and verification processes become slow because staff must manually validate which devices are assigned to each user and where they are located, resulting in an average SLA of 133 hours.

The aim of this work was to develop a responsive web system that centralizes asset information and automates the registration of movements, maintenance activities, and institutional documentation, reducing reliance on manual controls and enabling analytics for decision-making. The methodology combined requirements elicitation and validation with operational stakeholders, architectural design using the C4 model and a microservices approach, and iterative development with Scrum in two-week sprints. The solution was implemented on a SQL Server database and a backend distributed into specialized services (Assets/DataWarehouse, Movements, Maintenance, and Templates/Records), following a clean layered architecture and exposing APIs consumed by a web client.

As a result, a functional platform was delivered and validated through iterative reviews and testing (unit, integration, usability, and acceptance). The system strengthens traceability by linking each asset to its history, responsible users, locations, PDF records, and maintenance evidence. It is concluded that the solution improves operational control and information quality; however, a complete assessment of SLA improvement requires a formal definition of the indicator, an agreed data source, and an acceptance testing plan in the operational environment.

**Keywords**: asset management; traceability; microservices; Scrum; SQL Server; PDF records; dashboards; SAP integration; SysAid.

# Introducción

## **Identificación y descripción del problema o necesidad**

La administración eficiente de los activos tecnológicos es fundamental para el funcionamiento diario de cualquier institución académica de una universidad, el área de soporte técnico maneja un aproximado de 10.572 equipos (Anexo 1) entre ellos computadoras de escritorio, laptops, monitores, teclados, ratón e impresoras distribuidos en varias sedes. Aunque estos activos se registran en SAP, este registro es enfocado parte la parte contable (número de serie, modelo, proveedor y fecha de adquisición) pero no a la parte operativa diaria (Préstamo de equipos, mantenimientos programados, devoluciones, ingresos), su ingreso al sistema solo ocurre tras una revisión manual por parte del equipo de soporte. De este modo, cada operación que ocurre internamente, préstamo-mantenimiento-devoluciones-ingresos se valida con un proceso disperso y poco automatizado mediante hojas de cálculo y documentos físicos, este proceso mantiene una verificación manual de los datos, introduciendo riesgos de errores, complica la trazabilidad y retrasa la operación diaria.

**Necesidades surgidas a raíz de la pandemia y que se mantienen hasta la actualidad**

Durante la emergencia sanitaria por COVID-19, se implementaron medidas extraordinarias dentro de la Universidad como el envío de equipos tecnológicos a los hogares del personal, traslado de equipos a sedes con mayor espacio para evitar contagios.

Tras el retorno a la presencialidad, se evidenciaron múltiples inconsistencias entre lo registrado en SAP y la ubicación o condición real de los equipos, estas inconsistencias persisten incluso en 2025, afectando la capacidad del área de soporte para operar con precisión o atender solicitudes de manera oportuna.

**Manejo manual y fragmentación de la información**

La administración actual de activos depende de todo el grupo de soporte, quienes llevan a cabo de forma manual y descentralizada (en al menos cuatro hojas de cálculo independientes) este proceso. Esta falta de centralización dificulta que el personal de TI acceda de manera eficiente a la información necesaria, lo que genera demoras en la asignación, retiro, préstamo de los activos y errores en el manejo diario operativo.

Actualmente, la información se distribuye en los siguientes registros ([Figura 1](bookmark://Bookmark1)):

* Registro general de activos asignados por empleado.
* Control de activos devueltos a soporte (IMAC).
* Inventario de activos disponibles para asignación.
* Préstamos temporales utilizados en eventos u otras solicitudes específicas.

Además de estos registros, no existe claridad sobre si hay otras fuentes adicionales con información relacionada, lo que aumenta la incertidumbre y el riesgo de pérdida de control sobre los activos. A esto se suman otros documentos de uso recurrente, como actas de entrega, devolución, préstamo todas estas firmadas en PDF y validadas una por una con los registros llevados en las hojas de cálculo. Todo este proceso se desarrolla fuera de SAP, con registros aislados que no se sincronizan automáticamente.   
Esta estructura conlleva una gran carga operativa, duplicidad de esfuerzos, retrasos en la actualización de información y un alto riesgo de errores humanos. Además, muchas asignaciones temporales ni siquiera llegan a registrarse formalmente en SAP, lo que incrementa la brecha entre lo que aparece en el sistema y la realidad del inventario.

**Falta de trazabilidad y visibilidad operativa**

Actualmente, como se puede observar en el [**Anexo 1**](bookmark://Marcador1) existen 3.293 CPU y 492 iMac en uso, lo que implica al menos 3.785 teclados y ratón asociados.Sin embargo, estos periféricos no están registrados individualmente como activos, por lo que cuando se dañan o extravían, son reemplazados sin dejar constancia, ya que no forman parte del control formal de inventario. Esta falta de trazabilidad de consumibles representa pérdidas silenciosas y reiteradas, y evidencia un riesgo mayor: la pérdida de activos sin registro o seguimiento, especialmente en el caso de equipos dañados que se sustituyen sin actualizar los sistemas de control.

Esta carencia de información estructurada genera un desfase en la toma de decisiones, ya que la falta de visibilidad clara y actualizada sobre los equipos disponibles dificulta responder con agilidad a las necesidades del personal académico y administrativo. Además, representa una seria dificultad para enfrentar auditorías o procesos de control interno, dado que la información se encuentra dispersa, fragmentada y, en muchos casos, desactualizada, comprometiendo la transparencia institucional.

Aunque el equipo de soporte dispone de un stock de equipos y consumibles para atender incidentes críticos en aulas o brindar soporte al personal administrativo, el manejo de esta información se realiza mediante archivos Excel desactualizados por sede, lo que ocasiona una distribución ineficiente de los recursos y genera demoras en la asignación de equipos de reemplazo.

**Datos y magnitud del problema**

Entre [**Anexo 2**](bookmark://Marcador2) 2011 y 2023, se ha llevado el control de un inventario superior a los 10.572 activos tecnológicos. De estos:

* Al menos 9.600 activos presentan números de serie únicos, pero más de 40% de ellos no tienen una codificación estandarizada o contienen datos genéricos como “S/N”. Esto impide asociar un activo a su responsable de forma confiable.
* Se identifican al menos 54 marcas y más de 500 modelos distintos de equipos, lo que incrementa la complejidad técnica.
* 1.194 personas distintas han estado asociadas como responsables de uno o más activos.
* El período 2011–2015 concentra más de 5.178 activos, lo que sugiere que muchos equipos ya han superado su vida útil sin que exista un proceso automatizado de baja o renovación.

Del[**Anexo 3**](bookmark://Marcador3)1 de enero de 2024 al 9 de mayo de 2025, se han generado 3.602 tickets vinculados a activos, en categorías como:

* 885: Asignación y préstamo
* 602: Salida de funcionarios
* 234: Préstamos con soporte
* 663: Ingreso de funcionarios
* 587: Formateo y cambios
* 420: Reemplazo de periféricos (teclado, ratón, audífonos, etc.)
* 101: Devolución

Los tickets de devolución presentan un SLA promedio de 133 horas ([**Anexo 4**](bookmark://Marcador4)), evidencian cuellos de botella innecesarios en el proceso. Esto se debe, en gran parte, a la necesidad de verificar manualmente los datos de los activos asignados al usuario, así como a la búsqueda de su ubicación física o la confirmación de qué equipos tiene en su poder. Esta falta de trazabilidad centralizada retrasa la administración, aumenta la carga operativa del personal de soporte y afecta los tiempos de atención al usuario.

La situación actual presenta diversos riesgos que afectan directamente la eficiencia operativa y el control institucional, uno de los más críticos es la falta de trazabilidad real, ya que el sistema SAP no permite conocer en tiempo real el estado, ubicación y uso de los activos tecnológicos. Además, existe una fuerte dependencia de procesos manuales expuestos a errores, como la generación no automatizada de actas en PDF o el uso de múltiples archivos sin integración entre sí, lo que incrementa la probabilidad de inconsistencias, duplicidades y pérdida de información. A esto se suma la ausencia de un plan de mantenimiento preventivo, lo que conlleva una administración reactiva a fallos, aumentando el riesgo de averías en equipos críticos y reduciendo la vida útil de los activos.

La universidad se encontró con una dificultad importante: la falta de mantenimiento preventivo y continuo de los equipos tecnológicos. Como consecuencia, los proyectores comenzaron a presentar fallas debido a la acumulación excesiva de polvo, y las computadoras también comenzaron a deteriorarse. A partir de marzo de este año, los mantenimientos se han realizado únicamente cuando los usuarios reportan problemas, lo que limita el alcance y efectividad de estas intervenciones. Además, los únicos registros existentes de estos mantenimientos se encuentran en los correos electrónicos, lo que impide llevar un control centralizado y confiable. Esta situación hace que no sea posible saber con certeza qué equipos han recibido mantenimiento recientemente, ni en qué condiciones se encuentran proyectores u ordenadores específicos

**¿Por qué esto se vuelve una necesidad?**

Actualmente, como se ha evidenciado en los apartados anteriores, la universidad carece de una herramienta que permita el control centralizado de sus activos tecnológicos, con la capacidad de registrar de forma estructurada su ubicación, estado, responsable y la fecha del último mantenimiento realizado.

Aunque los activos están registrados en SAP, el sistema no ofrece trazabilidad operativa, esta limitación ha obligado a implementar procesos manuales y paralelos, con registros distribuidos en hojas de cálculo, correos electrónicos y documentos físicos, lo que genera errores, demoras y una información altamente fragmentada.

A esta situación se suma la ausencia de un plan de mantenimiento preventivo y programado para los equipos tecnológicos, como proyectores y computadoras. Los únicos registros disponibles se encuentran en correos electrónicos, lo que impide llevar un historial técnico confiable y dificulta conocer qué equipos han recibido mantenimiento, en qué fecha y en qué condiciones. La falta de control técnico incrementa el riesgo de fallos, reduce la vida útil de los dispositivos y puede comprometer el desarrollo de actividades académicas y administrativas.

Por tanto, se vuelve una necesidad institucional prioritaria implementar una solución que centralice, automatice y garantice tanto el control de activos como el seguimiento sistemático de sus mantenimientos.

**FODA NECESIDAD**

|  |  |
| --- | --- |
| ***FORTALEZAS*** | ***OPORTUNIDADES*** |
| La universidad dispone de un inventario formal de activos tecnológicos y utiliza un ERP (SAP) de uso internacional, enfocado en el control financiero. | Desarrollo de una plataforma o sistema integral para el control de activos que reemplace el uso de hojas de cálculo y mejore la trazabilidad. |
| El área de soporte técnico cuenta con experiencia en la atención de tickets y la administración de activos, incluso en contextos complejos como el generado durante la pandemia. | Integración con herramientas de soporte técnico para mejorar los tiempos de atención y la disponibilidad de información para el personal de soporte. |
| La universidad cuenta con infraestructura tecnológica, incluyendo un stock suficiente de equipos como monitores, que permite realizar reemplazos oportunos en áreas críticas sin depender de compras urgentes. | Fortalecimiento de políticas de inventario y control de consumibles para reducir pérdidas silenciosas y aumentar la eficiencia en el uso de recursos. |
| El personal técnico ha demostrado capacidad para adaptarse y sostener operatividad durante escenarios de alta exigencia como la pandemia, implementando soluciones improvisadas pero funcionales. | Implementar un plan de mantenimiento preventivo y programado que reduzca fallos, extienda la vida útil de los equipos y mejore la disponibilidad tecnológica. |
| Existe experiencia previa en generación y validación de documentación institucional (actas de entrega, devolución, préstamo), lo cual puede ser aprovechado para la automatización futura. | Desarrollar o integrar una herramienta digital para registrar y monitorear mantenimientos, con historial por equipo. |
| **DEBILIDADES** | **AMENAZAS** |
| SAP no permite realizar una trazabilidad operativa de los activos (ubicación exacta, movimientos, responsables, prestamos, ticket asociado). | Existe un riesgo latente de pérdida o robo de equipos debido a la falta de un control actualizado, lo que impide su detección oportuna y deja al área sin capacidad de respuesta efectiva una vez ocurrido el incidente. |
| La administración depende de varias personas sin llevar un registro formal y de procesos manuales no escalables, lo que representa un alto riesgo operativo. | Incumplimiento de SLA en asignaciones o préstamos de activos debido a la falta de información confiable. |
| - Uso de múltiples hojas de cálculo sin integración entre sí ni con SAP, lo que genera duplicidades, errores y pérdida de información. | Posibles observaciones o sanciones en auditorías internas o externas por inconsistencias o falta de trazabilidad. |
| Falta de trazabilidad de consumibles como teclados, mouse y otros periféricos, que se reponen sin dejar registro. | Sobrecarga operativa y desgaste del personal por depender de tareas manuales repetitivas y críticas. |
| Procesos lentos y propensos a errores como la transcripción manual de actas en PDF adjuntas a tickets. | Desconfianza institucional hacia el área de soporte o la administración de inventario por parte de usuarios y autoridades. |
| Dificultad para responder con agilidad a solicitudes, auditorías o incidentes por falta de visibilidad en tiempo real. | Aumento del riesgo de fallos masivos en equipos críticos por acumulación de polvo o falta de limpieza técnica. |
| El mantenimiento técnico de los equipos se realiza de forma mayormente reactiva, lo que evidencia una falta de planificación preventiva | La falta de visibilidad operativa en tiempo real impide la toma de decisiones ágiles y oportunas, lo que puede afectar procesos académicos o administrativos clave. |

Table 1. Análisis FODA de la gestión de activos tecnológicos en la Universidad de Las Américas

## **Descripción de la organización**

Por medio del análisis FODA presentado en la Tabla 1, se identificó la necesidad de desarrollar una solución de software que integre y centralice el control de activos, vinculándola con el ERP SAP y el sistema de tickets de soporte. Esta integración busca evitar duplicidades y garantizar la trazabilidad operativa de cada equipo y consumible.

Al estandarizar los roles y procesos mediante flujos de trabajo definidos para asignaciones, préstamos y devoluciones, se reducirán los riesgos operativos derivados de registros manuales dispersos y se favorecerá el cumplimiento eficiente de los SLA establecidos. Asimismo, la plataforma permitirá programar y registrar mantenimientos preventivos y limpiezas técnicas, lo que contribuirá a prolongar la vida útil de los activos y mejorar la disponibilidad tecnológica.  
La Universidad es una institución de educación superior reconocida por su compromiso con la calidad académica, la innovación y el desarrollo tecnológico. Ofrece una amplia variedad de programas de grado y posgrado en distintas áreas del conocimiento, con un enfoque en la formación integral de sus estudiantes.

Cuenta con varios campus distribuidos geográficamente, lo que exige una gestión tecnológica eficiente que garantice la continuidad de las actividades académicas y administrativas en todas sus sedes.

Para responder a estas necesidades, la universidad cuenta con la Dirección de Tecnologías de la Información (TI), responsable de administrar los sistemas, servicios y recursos tecnológicos institucionales. Dentro de esta dirección opera la Coordinación de Soporte a Usuarios, liderada por Ana María Ariza, que se encarga de administrar el soporte técnico de primer nivel (N1). Sus funciones abarcan desde la entrega y mantenimiento de equipos hasta la atención de incidentes y requerimientos tecnológicos de estudiantes, docentes y personal administrativo.

El equipo de soporte técnico cumple un rol fundamental en el día a día de la universidad, asegurando que los recursos tecnológicos estén disponibles y funcionando adecuadamente para todas las áreas académicas y administrativas. Esta coordinación trabaja bajo lineamientos operativos claros, orientados a ofrecer un servicio eficiente y con enfoque en el usuario final.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 1. Organigrama del área de Tecnología de la Información.

# Análisis de posibles soluciones

Frente a la problemática identificada en la Universidad, relacionada con la falta de trazabilidad, control y mantenimiento adecuado de los activos tecnológicos, es necesario evaluar alternativas que permitan optimizar el proceso operativo del área de soporte técnico.

Por esta razón, se presentan a continuación tres posibles soluciones enfocadas en resolver los puntos críticos detectados: trazabilidad de equipos, control de préstamos y devoluciones, administrar de stock, registro de mantenimientos, y generación de reportes que reflejen con precisión la situación real del inventario tecnológico.

## **Identificación y selección de la mejor solución**

**Alternativa 1(Seleccionada):** Desarrollo de un sistema web responsive para la trazabilidad integral y control operativo de activos tecnológicos de una Universidad

Se propone el desarrollo de una aplicación web responsiva que centralice la administración de activos tecnológicos, mejorando la trazabilidad operativa, automatización de procesos y control documental. Esta solución permite registrar el ciclo completo de cada activo (ingreso, movimientos, mantenimientos, tickets relacionados y baja), vinculando responsables, ubicaciones y documentación automática como actas de asignación cuando se vaya a entregar un activo, acta de devolución y acta de préstamo de activos todas estas generadas automáticamente en PDF y actualizadas en el sistema.

El sistema integra dashboards analíticos y estadísticos, alertas preventivas y conexión con el sistema de tickets SysAid mediante API, evitando duplicidades y mejorando la eficiencia del soporte técnico. Utiliza autenticación Microsoft (SSO) y asignación de roles para control de acceso y responsabilidades por perfil. Toda la información es respaldada por una base de datos optimizada que administra los activos, usuarios, ubicaciones, mantenimientos y documentos, y se alinea con categorías SAP y estructuras contables para que cuando se exporte el documento en formato Excel, se pueda subir los cambios sin problemas al ERP.

Esta solución reduce los procesos manuales, fortalece el control operativo y mejora la visibilidad de los activos en tiempo real, sentando las bases para una administración tecnológica más eficiente.

**Módulos**

**1. Siguimiento de activos tecnológicos**

* Permite conocer todo el historial del activo desde que ingresa a soporte hasta su baja o reasignación:
* Movimientos internos entre usuarios o sedes.
* Préstamos temporales o asignaciones definitivas.
* Historial detallado por responsable, ubicación, uso, mantenimientos, devoluciones y bajas.
* Descargar el archivo Excel con los activos que requieren ser actualizados en SAP.

**2. Dashboard y reportes analíticos**

Permite visualizar información clave para la toma de decisiones:

* Total de activos por sede, tipo y estado (asignado, disponible, en mantenimiento, en baja).
* Alertas de activos sin mantenimiento reciente o con uso prolongado.
* Equipos en préstamo, pendientes de devolución o vencidos.
* Movimientos y asignaciones por área o período.

**3. Aplicación web responsive con autenticación por rol**

Aplicación segura, accesible por navegador o dispositivos móviles, con autenticación vía Microsoft (Single Sign-On), permitiendo el acceso solo al personal autorizado. Roles definidos:

* **Asistente de Soporte a Usuarios**: Verifica que el etiquetado y datos del activo estén correctos. Marca los activos listos para SAP.
* **Coordinadora de Soporte a Usuarios**: Administra usuarios, valida movimientos, visualiza dashboards, supervisa el estado de los activos.
* **Asistente de Activos Fijos**: Descarga reportes en CSV, valida activos para cruce con SAP.
* **Soporte Técnico**: Crea y editar activos, documenta mantenimientos
* **Usuarios Finales (profesores, administrativos, estudiantes)**: Consultan sus activos asignados y el historial de préstamos/devoluciones.

**4. Módulo de generación automática de PDFs**

Generación automática de actas para:

* Asignación de activos
* Devolución de activos
* Préstamos temporale
* Mantenimientos

Los documentos se construyen a partir de una plantilla HTML con variables dinámicas y se generan listos para firma digital o manual. Se almacenan junto al activo correspondiente.

**5. Integración con sistema de tickets (SysAid)**

Conexión vía API para visualizar desde la plataforma el estado del ticket asociado al activo:

* Estado del ticket (abierto, cerrado, pendiente, en proceso).

**6. Módulo de mantenimiento**

Permite registrar y documentar de forma estructurada todos los trabajos de mantenimiento realizados sobre un activo, tanto correctivos como preventivos:

* Registro de fecha y hora de inicio y fin del mantenimiento.
* Datos del técnico responsable y del solicitante (si aplica).
* Descripción del problema reportado y solución aplicada.
* Fotografías del estado del equipo antes y después del mantenimiento.
* Clasificación del mantenimiento (preventivo o correctivo).
* Asociación del mantenimiento con el activo y, si corresponde, con un ticket en SysAid.

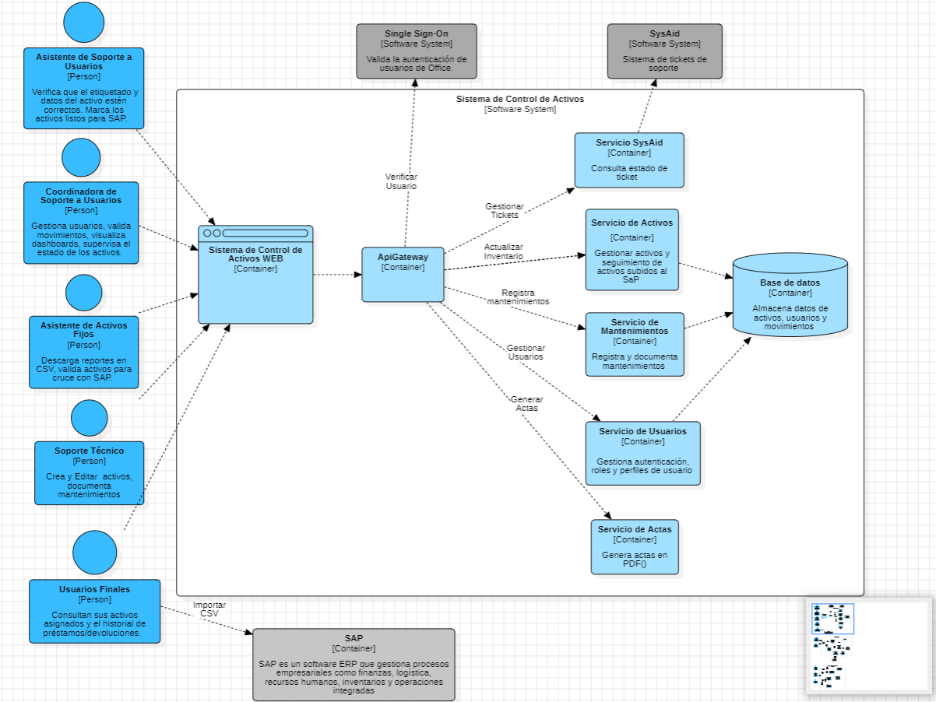


Figure 2. Diagrama de contenedores de la Alternativa 1: arquitectura del sistema de control de activos tecnológicos.  
Arquitectura:

La arquitectura del sistema de control de activos tecnológicos de una Universidad está basada en un modelo de microservicios como muestra en la Figura 2, lo que permite distribuir las responsabilidades en contenedores independientes, facilitando la escalabilidad, el mantenimiento y la integración con otros sistemas institucionales.

Los usuarios acceden al sistema a través de una aplicación web llamada Sistema de Control de Activos WEB, que actúa como interfaz principal. Esta aplicación es utilizada por diferentes perfiles, como el asistente de soporte a usuarios, la coordinadora de soporte, el asistente de activos fijos, el soporte técnico y los usuarios finales. Cada uno de ellos tiene un conjunto de funciones específicas: desde marcar activos listos para SAP, registrar mantenimientos o consultar los equipos asignados.

Para asegurar un acceso seguro y centralizado, la autenticación se realiza mediante Single Sign-On, utilizando cuentas institucionales de Microsoft Office 365. Este proceso de validación es administrado por el Servicio de Usuarios, el cual se encarga de los perfiles, roles y autenticación de cada usuario.

Una vez autenticado, el usuario interactúa con el sistema a través de un API Gateway, que actúa como orquestador entre el frontend y los microservicios internos. Este componente dirige las solicitudes hacia servicios especializados, cada uno responsable de una parte crítica del proceso.

El Servicio de Activos controla todo lo relacionado con los equipos: su registro, movimientos entre usuarios o sedes, préstamos temporales y actualizaciones de inventario. Cuando un activo es revisado y considerado listo, este servicio se encarga de enviar la información al sistema SAP, validado previamente por el personal correspondiente.

Por su parte, el Servicio de Mantenimientos permite registrar mantenimientos preventivos o correctivos. Este servicio documenta la intervención con datos como fechas, descripciones técnicas, responsables y fotografías antes y después del procedimiento.

Para los procesos que requieren respaldo formal, como la entrega o devolución de equipos, el sistema cuenta con un Servicio de Actas, que genera documentos en PDF de forma automática, usando plantillas HTML personalizables. Estas actas se almacenan y se vinculan al activo y al usuario involucrado.

El sistema también se integra con SysAid, una plataforma externa para el manejo de tickets de soporte. Esta integración es realizada por el Servicio SysAid, que permite consultar el estado de un ticket asociado a un activo, aportando contexto adicional en el momento de procesar un mantenimiento o una devolución.

Toda la información del sistema se almacena de forma estructurada en una Base de Datos central, que guarda datos de activos, usuarios, movimientos, mantenimientos y actas. Este diseño asegura la trazabilidad completa de cada equipo desde su ingreso hasta su baja.

**Alternativa 2**(Flutter/Dart): Aplicación móvil para gestión semiautomática de inventario

Se propone desarrollar una aplicación móvil nativa para Android e iOS, construida con Dart y Flutter, que permita gestionar de forma integral todo el ciclo de vida de los activos tecnológicos se encuentra directamente desde el dispositivo del usuario. La app ofrecerá formularios optimizados para captura en terreno incluyendo fotografías y geolocalización y funcionará en modo offline, sincronizando automáticamente los datos cuando recupere conexión.

La autenticación se realiza mediante Microsoft SSO, garantizando control de acceso por rol. A través de la integración con SysAid, los usuarios podrán consultar y actualizar el estado de los tickets asociados a cada activo sin abandonar la aplicación. Cada noche (o en el horario definido), un proceso programado genera y envía por correo al Coordinador un Excel plano con el listado de activos a validar en SAP, que servirá como respaldo y base de datos offline.

Las actas de asignación, devolución y préstamo se envían por correo como plantillas PDF; una vez completadas y firmadas (ya sea en formato papel o digital), el responsable las sube a la app como imagen o PDF escaneado. Notificaciones push y correos electrónicos alertan a coordinadores y usuarios finales sobre mantenimientos próximos, devoluciones pendientes o documentos por completar.

Esta alternativa maximiza la movilidad del equipo de soporte, reduce los tiempos de registro y asegura que toda la información quede centralizada y sincronizada en la nube tan pronto como sea posible.    
**Módulos**

**1. Autenticación y roles**

* Inicio de sesión con Microsoft SSO
* Gestión de permisos por rol (Coordinador, Soporte Técnico, Usuario Final)

**2. Gestión de activos desde móvil**

* Formularios adaptados para creación y edición de activos
* Captura de fotografías y geolocalización en cada movimiento o mantenimiento
* Sincronización automática en segundo plano

**3. Integración con SysAid**

* Consulta de estado y comentarios de tickets asociados
* Añadir notas o cerrar tickets directamente desde la app

**4. Envío programado de listados SAP**

* Generación nocturna de Excel plano con activos a validar
* Envío automático por correo al Coordinador
* Marcado en la app de registros que requieren actualización en SAP

**5. Notificaciones y recordatorios**

* Push notifications para mantenimientos próximos y devoluciones vencidas
* Correos electrónicos automáticos con enlaces directos a formularios pendientes

**6. Gestión de actas por correo**

* Envío de plantillas PDF para asignación, devolución y préstamo
* Subida de actas firmadas (foto o PDF escaneado) desde la aplicación

**7. Panel de seguimiento para el Coordinador**

* Vista consolidada de pendientes de validación, estados de activos y cargas recientes
* Filtros por sede, tipo de activo, estado operativo y responsable

**8. Reportes y exportaciones básicas**

* Descarga de listados en CSV directamente desde la app
* Historial de actividades y documentos cargados por usuario

**9. Soporte offline y sincronización**

* Almacenamiento local de datos cuando no hay conexión
* Resolución automática de conflictos al restaurarse el acceso a internet

Con esta combinación de Flutter en el cliente y Go en el servidor, se consigue una solución ágil, segura y capaz de soportar altas tasas de concurrencia, ideal para la gestión en tiempo real de los activos tecnológicos de la Universidad.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 3. Diagrama de contenedores de la arquitectura del sistema móvil de control y seguimiento de activos tecnológicos.

**Arquitectura:**

La solución se estructura en torno a un contenedor denominado **"Sistema Móvil"**, como se muestra en la **Figura 3**, el cual autentica a los usuarios (Asistente de Soporte, Coordinadora, Soporte Técnico, Usuarios Finales y Asistente de Activos Fijos) mediante Single Sign-On de Office 365 y expone sus peticiones a un ApiGateway. Este Gateway enruta las llamadas a varios microservicios empaquetados también como contenedores: el **Servicio de Activos**para CRUD y seguimiento de equipos (con integración a SAP), el **Servicio de Mantenimientos**para registrar intervenciones con fotos y ubicación, el **Servicio SysAid**para consultar y actualizar tickets, y el **Servicio de Usuarios**para gestionar perfiles y permisos. Toda la información de activos, usuarios y movimientos se persiste en **Firebase**(Realtime Database/Firestore y Storage), mientras que el **Servicio de Correo**envía automáticamente los listados Excel y las plantillas PDF de actas (asignación, devolución, préstamo) al responsable. Las notificaciones push se entregan a través de **Firebase Cloud Messaging**y **Apple Push Notification Service**, alertando a coordinadores y usuarios sobre mantenimientos próximos, devoluciones pendientes o documentos por completar. Finalmente, el Asistente de Activos Fijos descarga desde la app los CSV de activos para su validación y cruce en SAP.    
**¿Por qué usar ISO/IEC 9126 para elegir una alternativa?**

ISO/IEC 9126 es un estándar que nos da un conjunto claro de criterios de calidad para evaluar software. En lugar de opinar de forma vaga, medimos cada opción según seis grandes características:

1. Funcionalidad: ¿Cumple con lo que necesitamos hacer?

2. Fiabilidad: ¿Se ejecuta sin fallos y recupera bien de errores?

3. Usabilidad: ¿Es fácil de entender y usar?

4. Eficiencia: ¿Consume pocos recursos (CPU, memoria, red)?

5. Mantenibilidad: ¿Se puede corregir, ampliar o adaptar sin gran esfuerzo?

6. Portabilidad: ¿Funciona en distintos entornos (Windows, Android, iOS, etc.)?

Al calificar cada alternativa con estos seis ejes, obtenemos una comparación objetiva y replicable, que evita debates subjetivos y alinea la elección con lo que realmente importa para el proyecto.

**Justificación de la elección de la Alternativa 1 en base a ISO/IEC 9126**   
A continuación comparamos sólo las dos opciones propuestas usando los seis atributos de calidad de ISO/IEC 9126, con sus respectivos pesos y puntuaciones:

1. **Funcionalidad (25 %)**

La Alternativa 1 ofrece un registro exhaustivo de todo el ciclo de vida de cada activo: ingreso, asignación, traslado, mantenimiento y baja. Incluye generación automática de actas en PDF (asignación, devolución, préstamo), paneles analíticos con métricas clave y alertas preventivas, así como conexión bidireccional con SysAid y exportación directa de líneas pendientes a SAP. Esta cobertura integral asegura que cada operación quede trazada y disponible para consulta o auditoría sin depender de procesos manuales.

Por su parte, la Alternativa 2 permite gestionar gran parte de estos procesos desde el móvil, incluso en modo offline, y sincronizar con SysAid; sin embargo, carece de dashboards en tiempo real y no genera PDFs de forma nativa. Los listados SAP se envían por correo, pero no existen paneles de análisis ni workflows automáticos de aprobación masiva, lo que limita la visibilidad y control operativo.

1. **Fiabilidad(15%)**   
   La Alternativa 1 descansa sobre una base de datos relacional optimizada que garantiza transacciones ACID, replicación para alta disponibilidad y validaciones en backend que evitan datos incompletos o duplicados. Además, cuenta con logging centralizado y alertas ante cualquier excepción, de modo que los administradores pueden reaccionar rápidamente ante fallos o picos de carga.

En cambio, la Alternativa 2 introduce un componente de sincronización offline/online que, aunque útil en zonas sin cobertura, añade complejidad a la reconciliación de datos. Los conflictos de versiones, registros duplicados o pérdidas de información pueden surgir si un usuario permanece mucho tiempo desconectado o si la conexión es intermitente.

1. **Usabilidad (15 %)**

La interfaz web responsive de la Alternativa 1 está diseñada con vistas específicas por rol y accesible desde cualquier dispositivo. Su navegación intuitiva, apoyada por onboarding contextual y cumplimiento de estándares de accesibilidad (WCAG 2.1), facilita que técnicos, coordinadores y usuarios finales encuentren y completen sus tareas con un mínimo de entrenamiento.

La Alternativa 2, basada en Flutter, proporciona formularios adaptados al móvil y captura de fotos y geolocalización, lo que resulta muy práctico en terreno. No obstante, el espacio limitado de la pantalla obliga a segmentar la información en varias pantallas, aumentando la curva de aprendizaje y el número de toques necesarios para acceder a datos complejos.**Eficiencia (15 %)**   
Toda la lógica pesada (generación de PDFs, agregados de datos) se ejecuta en servidor en la Alternativa 1, ahorrando recursos en el cliente. En la Alternativa 2, la sincronización y el procesamiento local pueden consumir batería y datos móviles.

1. **Mantenibilidad (15 %)**

La Alternativa 1 adopta una arquitectura de microservicios desacoplados, cada uno con su propio ciclo de vida y despliegue independiente. Esto facilita la integración continua, las pruebas automatizadas y la incorporación de nuevos módulos sin afectar al resto del sistema. La documentación centralizada (Swagger/OpenAPI) y el versionado semántico garantizan trazabilidad y control de cambios.

En la Alternativa 2, aunque el código también está modularizado, existe la necesidad de mantener dos ecosistemas tecnológicos (Flutter/Dart y el backend), así como coordinar actualizaciones en tiendas de aplicaciones. Este doble stack aumenta la complejidad de pruebas y despliegues y exige más coordinación entre equipos.

1. **Portabilidad (15 %)**

La Alternativa 1, al ser una aplicación web, funciona en cualquier navegador y dispositivo sin instalaciones previas. Puede desplegarse tanto on-premise como en la nube, aprovechando contenedores Docker y Kubernetes para ajuste automático de capacidad. No depende de tiendas de apps ni de procesos de certificación adicionales.

La Alternativa 2 requiere instalar paquetes nativos en Android e iOS y pasar por los ciclos de publicación y revisión de cada tienda. Aunque ofrece acceso offline y notificaciones push nativas, la gestión de versiones y actualizaciones en múltiples plataformas introduce pasos adicionales en el ciclo de vida del producto.**Conclusión**   
Con un 100 % frente al 74 %, la Alternativa 1 se alinea mejor con los criterios de ISO/IEC 9126. Su funcionalidad más completa, mayor fiabilidad y eficiencia, junto a la facilidad de mantenimiento y portabilidad, la convierten en la opción óptima para garantizar un control operativo robusto y escalable de los activos tecnológicos de una Universidad.

1. **Eficiencia (15 %)**

En la Alternativa 1, toda la lógica pesada (generación de PDFs, agregados estadísticos y exportaciones) se procesa en servidor, de modo que el cliente sólo recibe los resultados ya filtrados. El uso de caching, paginación y microservicios escalables permite responder rápidamente incluso ante consultas masivas, optimizando el consumo de recursos y ancho de banda.

Por el contrario, la Alternativa 2 implicará más trabajo en el dispositivo móvil: sincronización de datos, procesamiento local de imágenes y cálculos preliminares. Esta carga adicional puede traducirse en un mayor consumo de batería, uso de datos móviles y tiempos de espera perceptibles para el usuario.

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo de Calidad** | **Peso (%)** | **Alternativa 1** | **Alternativa 2** |
| Funcionalidad | 25% | 5 | 4 |
| Fiabilidad | 15% | 5 | 3 |
| Usabilidad | 15% | 5 | 4 |
| Eficiencia | 15% | 5 | 3 |
| Mantenibilidad | 15% | 5 | 4 |
| Portabilidad | 15% | 5 | 4 |
| **Total** | **100%** | **100%** | **74%** |

Table 2. Comparación de alternativas Web responsive y aplicación móvil (Flutter) según ISO/IEC 9126.

La **Alternativa 1** se presenta como la opción más sólida según los seis atributos de calidad definidos en la norma **ISO/IEC 9126**, como se muestra en la **Tabla 2**. Ofrece una cobertura funcional completa, mayor confiabilidad gracias al control transaccional y monitoreo, una mejor experiencia de usuario multirrol, eficiencia al concentrar la lógica pesada en el servidor, facilidad de mantenimiento mediante una arquitectura modular y alta portabilidad al no requerir instalación.

En contraste, la **Alternativa 2** presenta limitaciones como la ausencia de dashboards en tiempo real, riesgos inherentes a la sincronización offline y la complejidad de operar en un doble ecosistema (cliente Flutter y backend), lo que la hace menos adecuada para un entorno de control operativo escalable.

Por tanto, se recomienda implementar la **Alternativa 1** como solución principal. En caso de que en el futuro se identifiquen necesidades específicas de operación en campo con conectividad limitada, se sugiere desarrollar un complemento móvil liviano que se integre sin comprometer la coherencia del sistema central.

## **Impacto del proyecto en la sociedad**

**2.2 Político**

El proyecto fortalece la transparencia institucional al proporcionar trazabilidad completa sobre los activos tecnológicos, lo cual es clave ante auditorías internas y externas. Su adopción permite alinearse con normativas de control público y mejora la rendición de cuentas ante entes gubernamentales y autoridades académicas, favoreciendo el cumplimiento de políticas de gobernanza universitaria y control de bienes públicos.

**Económico**

Desde la perspectiva económica, el sistema permite optimizar el uso de recursos tecnológicos mediante control riguroso del inventario, mantenimiento preventivo y reducción de pérdidas o duplicidades. Esto genera ahorros financieros sostenidos al evitar compras innecesarias y alargar la vida útil de los equipos. Asimismo, mejora la eficiencia operativa del personal técnico, reduciendo tiempos improductivos y reprocesos asociados a errores humanos.

**Social**

A nivel social, se fortalece la percepción de eficiencia institucional y se mejora la experiencia de los usuarios finales (estudiantes, docentes, administrativos) al contar con activos tecnológicos bien gestionados y disponibles oportunamente. El sistema también contribuye al bienestar del personal técnico al reducir la carga operativa repetitiva y permitir un entorno de trabajo más estructurado, predecible y profesionalizado.

**2.2.3 Tecnológico**

El proyecto impulsa la transformación digital en el entorno universitario mediante el uso de tecnologías modernas como MongoDB (NoSQL), arquitectura de microservicios y autenticación segura (SSO). Además, sienta las bases para escalabilidad futura, integración con sistemas externos como SysAid y SAP, y para la incorporación de inteligencia analítica mediante dashboards con indicadores clave de rendimiento (KPIs).

**Ambiental**

El sistema reemplaza la necesidad de imprimir documentos físicos como actas, formularios o reportes, lo cual contribuye a reducir el consumo de papel y materiales de oficina. También se minimiza la obsolescencia tecnológica al permitir mantenimientos programados, reduciendo el desecho innecesario de equipos reparables, lo que tiene un impacto positivo en la gestión responsable de residuos tecnológicos.

**Ético**

Desde el enfoque ético, el sistema mejora la responsabilidad institucional en la gestión de activos, promoviendo el uso adecuado de bienes comunes y evitando su mal uso, pérdida o apropiación indebida. A nivel de usuario, se fomenta la responsabilidad individual al permitir consultar el historial de activos asignados, favoreciendo una cultura de transparencia, trazabilidad y compromiso con los recursos institucionales.

**Solución replicable en otras instituciones educativas**

El sistema puede adaptarse fácilmente a otras universidades o institutos que enfrentan problemas similares en la administración de activos tecnológicos. Al documentar su arquitectura y procesos bajo buenas prácticas, la Universidad puede posicionarse como un referente nacional en trazabilidad tecnológica y digitalización de procesos administrativos, compartiendo su experiencia en foros académicos o redes interuniversitarias.

# Objetivos

## **Objetivo General**

Desarrollo de un sistema web responsive para la trazabilidad integral y control operativo de activos tecnológicos de una Universidad, mediante la automatización de registros de movimientos, mantenimientos y documentación institucional, con el objetivo de eliminar el uso de hojas de cálculo como principal herramienta de control y reducir en al menos un 30 % el tiempo promedio de atención de solicitudes (SLA), pasando de 133 horas a menos de 93 horas

## **Objetivos Específicos**

Ejecutar el levantamiento, análisis y validación de requerimientos del sistema de gestión de activos tecnológicos, garantizando una comprensión profunda de los procesos actuales y las necesidades operativas de los usuarios clave.

Diseñar la arquitectura funcional y técnica del sistema web, incluyendo estructuras de base de datos, flujos de navegación y control de acceso por roles, asegurando modularidad, escalabilidad y compatibilidad con SAP y SysAid.

Implementar la generación automática de actas en formato PDF (asignación, devolución y préstamo), integrando plantillas dinámicas y almacenamiento vinculado al historial de cada activo.

Desarrollar e implementar el sistema web responsive de forma iterativa, abarcando el ciclo completo de funcionalidades para la gestión de activos, automatización documental y visualización analítica.

Testear el sistema de forma integral, aplicando pruebas funcionales, de usabilidad, rendimiento y regresión durante cada iteración, con participación activa de usuarios finales para garantizar calidad y mejora continua.

Desplegar el sistema de forma controlada y brindar capacitación a los usuarios, asegurando su correcta operación en el entorno institucional y facilitando su integración en los procesos operativos establecidos.

# Alcance

## **Alcance de la solución seleccionada**

El alcance del proyecto inicia con el levantamiento estructurado de información técnica y operativa sobre los activos tecnológicos existentes en la Universidad, a través de entrevistas, revisión documental y catálogos institucionales. Esta información será normalizada y almacenada en una base de datos Sql Server, permitiendo gestionar datos estructurados e incrustar documentos como actas PDF y evidencias fotográficas importantes para llevar un registro correcto tanto de mantenimientos como el estado en el que llega o es entregado un equipo, a partir de estos datos se construirá, bajo metodología ágil Scrum, un sistema web responsive que abordará de forma modular los procesos definidos en los casos de uso: administración del ciclo de vida de los activos (alta, edición, baja), gestión de movimientos internos(asignación, prestamos, devoluciones), administrar mantenimientos, y visualización analítica a través de dashboards. Cada funcionalidad será desarrollada en sprints iterativos que incluirán análisis, diseño, desarrollo, pruebas y validación por actores clave. El proyecto finaliza con la entrega operativa y funcional del sistema, asegurando trazabilidad completa de los activos, integración con SysAid (para lectura de tickets) y generación de reportes estructurados para su carga en SAP, cumpliendo con los objetivos técnicos y estratégicos de la institución.

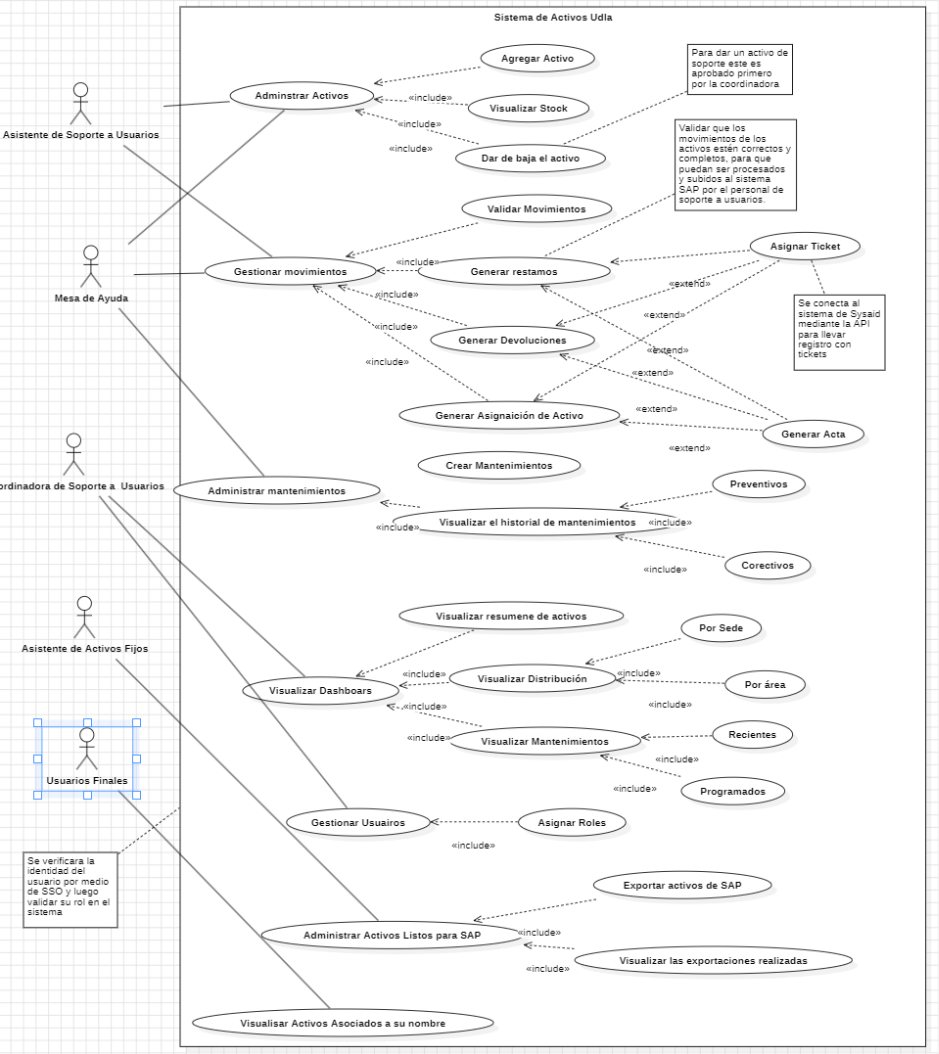


Figure 4. Diagrama de casos de uso del Sistema de Activos de la Universidad de Las Américas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Nombre del Caso de Uso | Frecuencia | Importancia | Urgencia | Referencia (Anexo) |
| UC01 | Administrar Activos | Alta | Vital | Inmediata | [Anexo 15](bookmark://Anexo15) |
| UC02 | Gestionar Movimientos | Alta | Vital | Inmediata | [Anexo 16](bookmark://Anexo16) |
| UC03 | Administrar Mantenimientos | Alta | Vital | Alta | [Anexo 17](bookmark://Anexo17) |
| UC04 | Visualizar Dashboards | Media | Importante | Puede Esperar | [Anexo 18](bookmark://Anexo18) |
| UC05 | Administrar Activos para SAP | Alta | Vital | Inmediata | [Anexo 19](bookmark://Anexo19) |
| UC06 | Visualizar Activos Asignados | Media | Importante | Puede Esperar | [Anexo 20](bookmark://Anexo20) |

Table 3. Modelo de procesos de casos de uso para el sistema de activos tecnológicos

A partir de los procesos descritos en la **Figura 4**y la **Tabla 3**, el sistema se organiza funcionalmente en seis módulos principales que reflejan el flujo operativo real del área de soporte de la universidad, cada módulo responderá a casos de uso previamente modelados [Anexo 7](bookmark://Marcador5), abarcando desde la gestión de activos hasta el control de mantenimientos y la trazabilidad. Para representar gráficamente estos flujos, se utilizó el estándar BPMN (Business Process Model and Notation) [Anexo 21](bookmark://Anexo21), mediante la herramienta StarUML, lo que permitió una visualización clara y estructurada del proceso actual.

A continuación, se detalla el funcionamiento de cada módulo desarrollado en el sistema.

**a) Módulo de seguimiento de activos tecnológicos (**[**Anexo 8**](bookmark://Marcador6)**)**

Permite registrar y consultar la trazabilidad completa de cada activo, incluyendo:

* Historial de movimientos (asignación, devolución, préstamo).
* Estado actual (disponible, asignado, en mantenimiento, dado de baja).
* Ubicación física y responsable del activo.
* Generación automática de actas en PDF para cada operación (entrega, devolución, préstamo).

**b) Módulo de mantenimiento (**[**Anexo 9**](bookmark://Marcador7)**y**[**Anexo 10**](bookmark://Marcador8)**)**

Sistema estructurado para registrar mantenimientos preventivos y correctivos:

* Fecha, técnico responsable, tipo de mantenimiento.
* Descripción del problema y solución aplicada.
* Fotografías del estado del equipo antes y después del procedimiento.
* Asociación del mantenimiento con el activo y, si aplica, con ticket de SysAid.

**c) Dashboard analítico y reportes (**[**Anexo 11**](bookmark://Marcador9)**)**

Visualización de información clave para la toma de decisiones:

* Total, de activos por sede, tipo y estado.
* Alertas sobre equipos sin mantenimiento reciente.
* Indicadores de cumplimiento de SLA.

**d) Aplicación web responsive (**[**Anexo 12**](bookmark://Marcador10)**)**

Desarrollada bajo arquitectura de microservicios, con frontend accesible desde navegador o dispositivos móviles. Incluye autenticación institucional y control de acceso por roles:

* **Soporte técnico**:
* ***Soporte en Sitio y Mesa de Ayuda*:** Visualización de Activos
* ***Encargado Acitvos:***creación y edición de activos, registro de mantenimientos
* **Coordinación de soporte**: validación de movimientos, supervisión de dashboard.
* **Área de activos fijos**: exportación para cruce con SAP.
* **Usuarios finales**: consulta de historial de sus propios activos.

**e) Integración con SysAid y SAP (**[**Anexo 13**](bookmark://Marcador13)**)**

* Consulta del estado de tickets asociados a activos mediante API con SysAid.
* Aprobación por parte de persona de activos de finanzas para confirmar cambios y poder exportar a SAP.
* Exportación directa de activos listos para SAP con codificación/formato compatible.
* Reducción de duplicidades y errores por sincronización con sistemas existentes.

**f) Módulo de generación automática de actas en PDF (**[**Anexo 14**](bookmark://Marcador14)**)**

* Actas de entrega, devolución y préstamo generadas desde plantillas dinámicas.
* Almacenamiento vinculado directamente al historial del activo.
* Listas para firma digital o manual.

## **Limitaciones y restricciones del proyecto**

El desarrollo e implementación del sistema web para el seguimiento y control de activos tecnológicos de la Universidad estará sujeto a las siguientes restricciones y limitaciones técnicas y operativas:

* El sistema será desarrollado utilizando la tecnología .NET, lo que implica que tanto el backend como los servicios estarán construidos dentro de este entorno de desarrollo, aprovechando su robustez y compatibilidad con entornos corporativos.
* Para el almacenamiento estructurado de información se utilizará **MongoDB** como base de datos principal, lo que permite mayor flexibilidad en el manejo de documentos, escalabilidad horizontal y un diseño orientado a objetos, adecuado para los registros de activos, mantenimientos y movimientos. Además, MongoDB facilitará el almacenamiento eficiente de archivos y fotografías asociadas a los activos, evitando la rigidez de modelos relacionales y permitiendo una gestión más dinámica de la información no estructurada.
* El acceso al sistema estará restringido únicamente a usuarios pertenecientes a la Universidad, autenticados mediante Single Sign-On (SSO) con sus cuentas institucionales de Microsoft 365, lo que garantiza seguridad y control de acceso por rol.
* La aplicación web estará disponible únicamente desde la red interna (intranet) de la universidad, por lo tanto, no será accesible desde redes externas sin configuración previa de VPN u otro mecanismo de acceso seguro validado por el área de Tecnologías de la Información.
* En esta fase no se contempla el desarrollo de una aplicación móvil nativa, sin embargo, el sistema será totalmente responsive, permitiendo su uso desde navegadores en dispositivos móviles siempre que estén conectados a la red institucional.
* La integración con SAP estará limitada a la generación y exportación de archivos en formato Excel estructurado, compatibles con el proceso de importación estándar del ERP, sin conexión directa o sincronización en tiempo real.
* El sistema no administrará licencias de software ni realizará escaneos automáticos del hardware o software instalado. Su enfoque estará centrado exclusivamente en la trazabilidad física y administrativa de los activos tecnológicos, incluyendo asignaciones, movimientos, mantenimientos y bajas.
* La interacción con el sistema de tickets SysAid se realizará mediante consumo de API REST en modo de solo lectura. Esto permitirá consultar el estado del ticket asociado a cada activo (abierto, cerrado, en proceso), sin permitir la creación, edición ni cierre de tickets desde la plataforma desarrollada.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimiento** | **Prioridad** |
| RF1 | Administrar activos tecnológicos (agregar, editar, baja, stock) | Must Have |
| RF2 | Gestionar movimientos (asignación, préstamo, devolución con ticket SysAid y acta) | Must Have |
| RF3 | Administrar mantenimientos (con fotos y ticket opcional) | Must Have |
| RF4 | Alertas de mantenimiento mensual | Should Have |
| RF5 | Dashboards analíticos y exportación | Must Have |
| RF6 | Exportación a SAP en Excel | Must Have |
| RF7 | Visualización de activos por usuario final | Should Have |
| RF8 | Acceso web responsive | Must Have |
| RF9 | Autenticación SSO | Must Have |
| RF10 | Trazabilidad completa histórica | Must Have |
| RNF1 | Escalabilidad horizontal | Should Have |
| RNF2 | Modularidad arquitectónica | Must Have |
| RNF3 | Respuesta ≤ 12 s por operación | Must Have |
| RNF4 | Cifrado y seguridad de datos | Must Have |
| RNF5 | Funcionamiento en intranet/VPN | Must Have |
| RNF6 | Tolerancia a fallos | Should Have |
| RNF7 | Documentación técnica clara | Should Have |
| RNF8 | Usabilidad e interfaz intuitiva | Must Have |

Table 4. Modelo de requerimientos funcionales y no funcionales del sistema de activos tecnológicos

A continuación, se detallan los requerimientos identificados durante la fase de levantamiento de información, alineados con los casos de uso modelados y el alcance funcional establecido. Estos requerimientos se han clasificado en funcionales y no funcionales, y se han priorizado utilizando la técnica **MoSCoW**, con el objetivo de orientar el desarrollo iterativo del sistema. La priorización correspondiente se presenta en la **Tabla 4**.

**Requerimientos funcionales**

1. El sistema debe permitir al asistente de soporte agregar, editar, dar de baja y visualizar el stock de activos tecnológicos.
2. El sistema debe registrar asignaciones, préstamos y devoluciones, generando automáticamente tickets en SysAid y actas PDF.
3. El sistema debe registrar mantenimientos preventivos y correctivos, con detalle técnico, evidencia fotográfica y (opcional) asociación de ticket SysAid.
4. El sistema debe enviar alertas mensuales sobre activos sin mantenimiento reciente.
5. El sistema debe mostrar dashboards analíticos con indicadores clave (SLA, distribución, KPIs) y permitir exportación a Excel.
6. El sistema debe validar activos y generar archivos Excel compatibles para importación manual en SAP.
7. El sistema debe permitir a los usuarios finales visualizar los activos asignados y su historial, en modo solo lectura.
8. El sistema debe ser accesible vía web y responsive para uso en dispositivos móviles.
9. El sistema debe autenticarse mediante Single Sign-On (SSO) con cuentas institucionales Microsoft 365.
10. El sistema debe mantener trazabilidad histórica completa de todos los activos (movimientos, mantenimientos, actas).

**Requerimientos no funcionales**

1. El sistema debe ser escalable horizontalmente para soportar incremento de activos y usuarios.
2. La arquitectura debe ser modular para permitir evolución funcional con bajo acoplamiento.
3. El sistema debe garantizar tiempos de respuesta ≤ 5–12 segundos por operación según el módulo.
4. Los datos en tránsito deben ser cifrados mediante HTTPS y los documentos sensibles en reposo deben ser protegidos.
5. El sistema debe operar en la red interna de la Universidad, accesible solo desde la intranet o vía VPN.
6. El sistema debe ser tolerante a fallos menores y permitir recuperación tras pérdida momentánea de conexión.
7. La documentación técnica del sistema debe ser clara para facilitar mantenimiento evolutivo.
8. La interfaz debe ser usable e intuitiva para los diferentes perfiles de usuario.

# Planificación y costos del proyecto

El proyecto será ejecutado por un equipo de dos personas, que asumirán las funciones de desarrollo backend, frontend, análisis, pruebas y documentación, coordinados bajo la metodología ágil Scrum para asegurar la entrega iterativa de funcionalidades completas por sprint. Para la gestión y seguimiento de las tareas asignadas, se utilizará la herramienta Jira, lo que permitirá llevar control sobre las historias de usuario y las tareas específicas derivadas de los casos de uso. Cada integrante del equipo tendrá responsabilidades distribuidas en las distintas fases del ciclo de vida del software, con entregas incrementales al final de cada sprint para revisión y retroalimentación.    
**Fase 1: Análisis**

* **Levantamiento de información:**
* Reuniones con las áreas de soporte técnico, activos fijos y usuarios finales.
* Revisión documental de catálogos institucionales de activos tecnológicos.
* Identificación de procesos actuales de seguimiento y control.

1. **Obtención de requisitos:**

* Definición de requisitos funcionales y no funcionales.
* Redacción y aprobación del documento de requisitos con actores clave.

1. **Investigación tecnológica:**

* Análisis y validación del stack tecnológico elegido:
* .NET Core para backend.
* Sql Server como base de datos principal.
* Arquitectura modular con microservicios.
* SSO con Microsoft 365.

1. **Definición del alcance y casos de uso:**

* Consolidación de los casos de uso generales y específicos.
* Documentación del alcance funcional y las restricciones técnicas.

**Fase 2: Diseño**

1. **Diseño arquitectónico:**

* Diagramas de arquitectura física y lógica.
* Diseño del esquema de base de datos en MongoDB.
* Definición de endpoints RESTful y flujos de datos.

1. **Diseño de interfaces:**

* Prototipos de las principales pantallas web (responsive).
* Diagramas de navegación y usabilidad.

1. **Diseño de seguridad:**

* Definición de estrategias de autenticación y roles.
* Implementación de cifrado de datos y control de sesiones.

**Fase 3: Implementación**

1. **Módulo de Administración de Activos:**

* Alta, edición, baja, visualización de stock y actas.

1. **Módulo de Movimientos:**

* Asignaciones, préstamos y devoluciones, integración con SysAid y generación de actas.

1. **Módulo de Mantenimientos:**

* Registro de mantenimientos preventivos/correctivos con evidencias.

1. **Módulo de Dashboards:**

* Indicadores, distribución y cumplimiento de SLA.

1. **Integración con SAP:**

* Exportación estructurada para carga en el ERP.

1. **Seguridad y autenticación:**

* Implementación completa de SSO y permisos por roles.

**Fase 4: Pruebas**

1. **Pruebas unitarias:**

* Validación de componentes individuales (.NET y MongoDB).

1. **Pruebas de integración:**

* Comprobación de flujos entre módulos y servicios externos (SysAid y SAP).

1. **Pruebas de sistema:**

* Verificación del comportamiento integral.
* Despliegue de la aplicación
* Configuración del despliegue

1. **Pruebas de usuario:**

* Sesiones de validación y retroalimentación con los usuarios finales.

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto., Imagen

Figure 5. Plan de sprints para la implementación del sistema de activos tecnológicos.

El desarrollo del proyecto se planifica desde el 1 de septiembre de 2025 hasta el 10 de enero de 2026, siguiendo un enfoque ágil e iterativo basado en la metodología Scrum. Se contemplan 10 sprints, cada uno con una duración de dos semanas, tal como se ilustra en la Figura 5. Cada sprint entregará incrementos funcionales completamente desarrollados, probados y validados por los actores clave, asegurando que cada módulo y subproceso sea finalizado y aprobado antes de avanzar a la siguiente iteración.

**Costos**

En esta sección se detallan los costos asociados con el desarrollo del proyecto *Sistema Web Responsive para el Seguimiento y Control de Activos Tecnológicos de una Universidad*. La estimación contempla mano de obra, infraestructura, licencias de software, gastos administrativos y despliegue, considerando el período comprendido entre agosto 1 de Septiembre 2025 y enero 5 de 2026, dividido en 10 sprints bajo metodología ágil Scrum.    
**Justificación de la tarifa horaria**

En Ecuador, para el año 2025, el salario promedio mensual de un desarrollador semi-senior especializado en tecnologías web y backend se estima en aproximadamente **$1,248 USD**, lo que equivale a una tarifa referencial de **$7.21 USD/hora**, considerando jornadas de 30 horas semanales. Para este proyecto, se ha establecido una tarifa ajustada de **$10 USD/hora**, la cual contempla no solo las actividades de desarrollo, sino también el análisis de requerimientos, diseño arquitectónico, documentación, pruebas, despliegue y un margen asociado a sobretiempos derivados del enfoque ágil adoptado.

**Costos de desarrollo**

**Mano de obra**

El equipo de trabajo está conformado por dos desarrolladores, Isaac y Esteban, quienes asumirán todas las fases del ciclo de vida del software. Cada uno trabajará **30 horas semanales durante 17 semanas**, desde el **1 de septiembre de 2025 hasta el 5 de enero de 2026**.

**Cálculo:**   
2 personas × 30 h/semana × 17 semanas × $10/hora = **$10,200 USD**

**Infraestructura y equipos.**   
Ambos integrantes utilizarán sus propios equipos, considerados como aportes personales:

* Equipo de Esteban: $800
* Equipo de Isaac: $1,650

**Total equipos:** $2,450

**Licencias de software**

Para las actividades de gestión, desarrollo, diseño y pruebas se utilizarán herramientas especializadas, con un costo total proyectado de **$850 USD**. Los detalles se presentan en la **Tabla 5**.

|  |  |
| --- | --- |
| Herramienta | Costo aproximado |
| Jira (gestión ágil) | $120 (6 meses) |
| IDE Visual Studio Professional | $250 (6 meses) |
| Postman Pro (1 cuenta) | $120 (6 meses) |
| Otros plugins y utilidades | $120 (6 meses) |

Table 5. Resumen de herramientas y costos aproximados para desarrollo y gestión de proyectos de activos tecnológicos.

**Gastos administrativos y misceláneos**

Incluyen consumo de internet, energía eléctrica, insumos y papelería durante todo el proyecto: **$400 USD**.

**Despliegue: escenarios posibles.**

* **Opción A: Infraestructura de la Universidad.**   
  En caso de contar con soporte institucional, se utilizará la infraestructura tecnológica de la universidad, sin costos adicionales para el equipo de desarrollo.
* **Opción B: Infraestructura externa.**   
  Si no se dispone del soporte institucional, se recurrirá a una solución básica en la nube para pruebas y despliegue:
* Servidor web (Azure/Linux para .NET): $20/mes × 6 meses = $120
* SqlServer Atlas (M0 gratuito): $0
* Certificados SSL / Dominio: $0
* Costo adicional total: $120

|  |  |  |
| --- | --- | --- |
| **Ítem** | **Costo total** | **Por adquirir** |
| Mano de obra | $10,200 | $10,200 |
| Equipos (aportes personales) | $2,450 | - |
| Licencias de software | $610 | $610 |
| Gastos administrativos y misceláneos | $400 | - |
| Despliegue (infraestructura de la Universidad) | $0 | - |
| Despliegue (infraestructura externa) | $840 | $840 |
| **Total (con la Universidad)** | **$14,500** | **$11650** |
| **Total (sin la Universidad)** | **$13,668** | **$10810** |

Table 6. Resumen de costos del proyecto de activos tecnológicos con y sin uso de infraestructura de una Universidad.

# Descripción de estudios realizados

Esta sección presenta una revisión documental de investigaciones y desarrollos académicos relevantes que sustentan conceptualmente la propuesta de este proyecto: un sistema web responsive para el seguimiento y control de activos tecnológicos en una Universidad. Se consideran temas como la gestión de inventarios mediante aplicaciones web, la integración con sistemas ERP y ITSM, y la incorporación de dashboards analíticos para la toma de decisiones.

**1. IoT-Enabled Real-Time Data Integration in ERP Systems**

Kumar (2022) analiza cómo la integración de sensores IoT y aplicaciones web con sistemas ERP tradicionales mejora la trazabilidad de activos y la gestión de inventario. El estudio muestra que incorporar capas de integración web y dashboards permite a los usuarios acceder a información en tiempo real, mejorando significativamente los procesos de control operativo y reduciendo errores administrativos al sincronizar datos de múltiples fuentes. Esto respalda la decisión de integrar el sistema con SAP y SysAid para mantener una visión centralizada del estado de los activos y los tickets relacionados.    
*Referencia: Kumar, N. (2022). IoT-Enabled Real-Time Data Integration in ERP Systems.*[*PDF*](https://www.academia.edu/download/120862327/12253.pdf)

**2. Data-Driven Decision Making: Advanced Database Systems for Business Intelligence**

Hosen et al. (2024) exploran cómo las bases de datos NoSQL y las herramientas de visualización como dashboards mejoran las capacidades de toma de decisiones en entornos empresariales. En comparación con modelos relacionales tradicionales, los NoSQL son más flexibles para manejar documentos y archivos multimedia, como los que almacenará el sistema propuesto (actas PDF, fotos de mantenimientos). El artículo también destaca la relevancia de Power BI y similares para presentar métricas clave y SLA, una práctica que se replica en el módulo de dashboards de este proyecto.    
*Referencia: Hosen, M.S., et al. (2024). Data-Driven Decision Making: Advanced Database Systems for Business Intelligence.*[*PDF*](https://www.researchgate.net/profile/Esther-Folorunso-3/publication/382695215_Data-Driven_Decision_Making_Advanced_Database_Systems_for_Business_Intelligence/links/66a9e30775fcd863e5eae75d/Data-Driven-Decision-Making-Advanced-Database-Systems-for-Business-Intelligence.pdf)

**3. Database and Dashboard Design of a CRM/BI Application**

Geresics-Földi y Msc (2023) documentan la implementación de un sistema CRM con dashboards integrados que facilita el seguimiento de activos y relaciones con clientes mediante una arquitectura flexible basada en NoSQL. Este estudio demuestra que centralizar la información y ofrecer paneles de control interactivos simplifica los flujos de trabajo y mejora la visibilidad operativa, principios que se aplican directamente en los módulos de administración y visualización del sistema de la Universidad.    
*Referencia: Geresics-Földi, E. (2023). Database and Dashboard Design of a CRM/BI Application for a State Export Agency. PDF*

**4. Adaptive Cloud-Based Big Data Analytics Model for Sustainable Supply Chain Management**

Stefanovic et al. (2025) proponen un modelo basado en analítica en la nube para gestión sostenible de cadenas de suministro, enfatizando el uso de dashboards y almacenamiento NoSQL para optimizar decisiones de inventario. Su investigación resalta que la trazabilidad digital de bienes a través de interfaces web adaptativas incrementa la eficiencia operativa, un enfoque que motiva el diseño responsive del sistema de la Universidad.    
*Referencia: Stefanovic, N., et al. (2025). Adaptive Cloud-Based Big Data Analytics Model for Sustainable Supply Chain Management. Sustainability.*[*Enlace*](https://www.mdpi.com/2071-1050/17/1/354)

**5. Big Data Engineering for Real-Time Inventory Optimization**

Pamisetty (2019) presenta un marco para optimización en tiempo real de inventarios mediante bases NoSQL y dashboards interactivos en la nube, integrados con sistemas ERP. Este trabajo subraya la importancia de mantener sincronización entre plataformas heterogéneas para evitar duplicidad de registros, elemento fundamental para justificar la integración del sistema de la Universidad con SAP y SysAid.

Referencia: Pamisetty, A. (2019). Big Data Engineering for Real-Time Inventory Optimization in Wholesale Distribution Networks. PDF

# Conclusión

# Estos estudios demuestran la viabilidad y las ventajas competitivas de implementar un sistema web basado en NoSQL y dashboards para la trazabilidad y control de activos. Además, evidencian que integrar plataformas externas (ERP e ITSM) mediante APIs incrementa la eficiencia organizacional, reduce la dependencia de procesos manuales y proporciona información clave para la toma de decisiones. La propuesta para la Universidad se alinea con estas prácticas de vanguardia, garantizando una solución moderna, escalable y adaptada a las necesidades institucionales actuales.

# Desarrollo del proyecto

## **Diseño de la solución**

El diagrama de contexto presentado a continuación sitúa al Sistema de Gestión de Activos dentro del ecosistema de la organización. Muestra a los usuarios principales y su interacción con el sistema, así como la dependencia crítica con el servicio de identidad corporativo.   
Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 6.Arquitectura nivel 1 (Contexto)

Este diagrama de nivel 1 define los límites del sistema. El Personal de TI es el actor principal que interactúa con la plataforma para realizar sus labores diarias.

Un componente fundamental en esta arquitectura es la integración con Microsoft Entra ID. El sistema no almacena contraseñas localmente; en su lugar, confía en Entra ID para la autenticación de usuarios mediante protocolos estándar como OpenID Connect (OIDC). Esto permite que el personal utilice sus credenciales institucionales existentes (Single Sign-On), garantizando que solo usuarios autorizados y activos en el directorio de la organización puedan acceder a la información sensible del inventario y los procesos de soporte.

**Arquitectura Nivel 2**

* El diagrama mostrado a continuación corresponde al nivel de Contenedores (Nivel 2) dentro del modelo C4. Este diagrama explica cómo el cliente web interactúa con los diferentes microservicios del backend y cómo estos colaboran entre sí para cumplir con los requisitos de negocio.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 7. Arquitectura nivel 2 (Contenedores)

Este diagrama ilustra la arquitectura distribuida de la solución. La aplicación web (**WebClient**) actúa como el punto de entrada único para los usuarios, comunicándose con cuatro microservicios especializados:

* **DataWarehouse (Assets Service):** Gestiona el inventario maestro.
* **Movements Service:** Controla las asignaciones y devoluciones.
* **Maintenance Service:** Administra los mantenimientos y evidencias.
* **Template Service:** Genera la documentación oficial.

El flujo de datos es orquestado principalmente por el frontend, aunque existe comunicación entre servicios (Service-to-Service) para mantener la integridad referencial, como cuando el *Movements Service* notifica al *Assets Service* que un activo ha cambiado de estado 

**Arquitectura nivel 3 Backend (Microservicios):**

* El diagrama presente a continuación detalla la arquitectura interna típica de los microservicios del backend (ejemplificado con el servicio de Activos y Movimientos). En este nivel 3, se observa la separación de responsabilidades mediante capas.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 8. Arquitectura nivel 3 Backend

Para entender este diagrama se puede explicar que cada microservicio sigue una arquitectura limpia. Los Controladores reciben las peticiones HTTP y validan la entrada. La Capa de Servicios contiene la lógica de negocio pura; por ejemplo, en MovementsService, esta capa valida las reglas de negocio (ej. "no se puede asignar un activo dado de baja") antes de proceder.

* Un componente clave son los Service Clients (como AssetsServiceClient), que encapsulan la comunicación HTTP con otros microservicios, permitiendo que el servicio de Mantenimiento, por ejemplo, verifique si un activo existe en el DataWarehouse sin acceder directamente a su base de datos.

**Arquitectura nivel 3 SPA (Frontend):**

* El diagrama a continuación representa la arquitectura detallada de nivel 3 del cliente web ([WebClient](https://vscode-file/vscode-app/c:/Users/esteb/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \t "_blank)). Se descompone la aplicación React en sus módulos funcionales y gestión de estado.
* el backend recibe las solicitudes HTTP enviadas desde el cliente (SPA) en sus controladores principales (AssetsController, BrandsController). Estos controladores actúan como puntos de entrada, recibiendo los DTOs (Data Transfer Objects).
* Los controladores delegan la lógica compleja a la capa de Servicios. Por ejemplo, el ImportService se encarga de procesar los archivos Excel/CSV cargados, validando que el formato sea correcto antes de intentar guardar los datos. La capa de Acceso a Datos (Repository pattern o Entity Framework directo) abstrae la comunicación con la base de datos, ejecutando las operaciones de inserción, lectura y actualización sobre las tablas de Activos.

**Arquitectura nivel 3 SPA (Frontend):**

El diagrama a continuación representa la arquitectura detallada de nivel 3 de la Single Page Application (SPA) construida con React y Vite. Se descompone el contenedor en sus componentes internos y sus interacciones

Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 9. Arquitectura nivel 3 Frontend.

Como se muestra en el diagrama, el AppRouter.tsx gestiona la navegación. La aplicación está estructurada en torno a Features (Activos, Movimientos, Usuarios).

El manejo de datos se realiza a través de Custom Hooks (como useAssets.ts, useActaWizard.ts) que abstraen la complejidad de las llamadas a la API. Estos hooks utilizan la capa de servicios (src/services/) donde se definen las instancias de Axios para cada microservicio.

## **Desarrollo de la solución**

**Modelo de clases**

El diseño de clases del sistema sigue el patrón de arquitectura de microservicios, donde el dominio se divide en contextos delimitados para garantizar la independencia y escalabilidad de cada módulo. A continuación, se presenta el diagrama de clases global, segmentado por los cuatro servicios principales: DataWarehouse (Activos), Movements (Movimientos), Maintenance (Mantenimiento) y Template (Documentación).

Aunque estas entidades residen en bases de datos separadas, mantienen una consistencia lógica a través de identificadores únicos (Business IDs) que permiten la trazabilidad de la información a través del sistema distribuido.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figure 10. Diagrama de Clases del Dominio  
A continuación, se detalla la responsabilidad de las entidades en cada microservicio:

**1. Contexto de Inventario (DataWarehouse Service)**

Este servicio actúa como la "Fuente de la Verdad". La clase central es **Asset**, que modela el bien tecnológico físico.

* **Asset:** Contiene los atributos inmutables e identificativos (Serie, Código de Barras, BusinessId).
* **Brand, Model, Campus:** Son entidades de catálogo que normalizan la información para evitar redundancia y errores de escritura.
* **Employee y Department:** Modelan la estructura organizacional y los custodios potenciales de los activos.

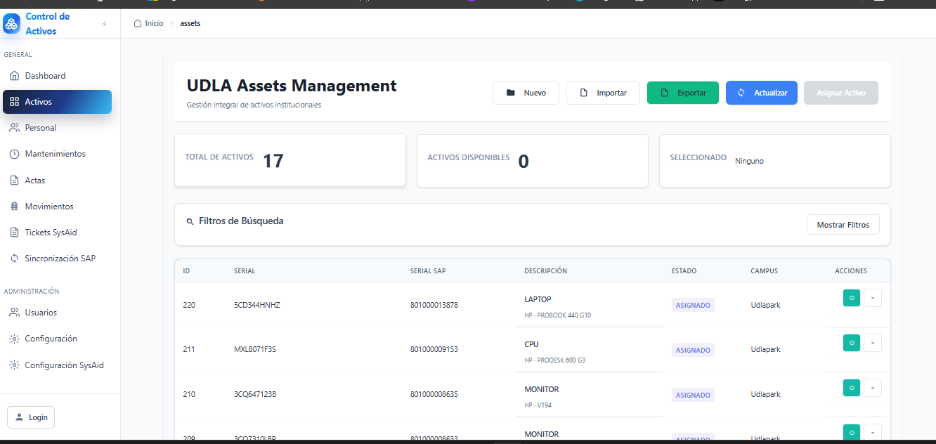


Figure 11. Página del Frontend con el módulo de AssetService

**2. Contexto de Ciclo de Vida (Movements Service)**

Este módulo gestiona la transaccionalidad. No duplica la información del activo, sino que referencia su ID.

* **Movement:** Es la entidad principal que registra eventos temporales (Asignaciones, Devoluciones). Incluye atributos de control como SemaphoreStatus para indicar la urgencia o estado del proceso.

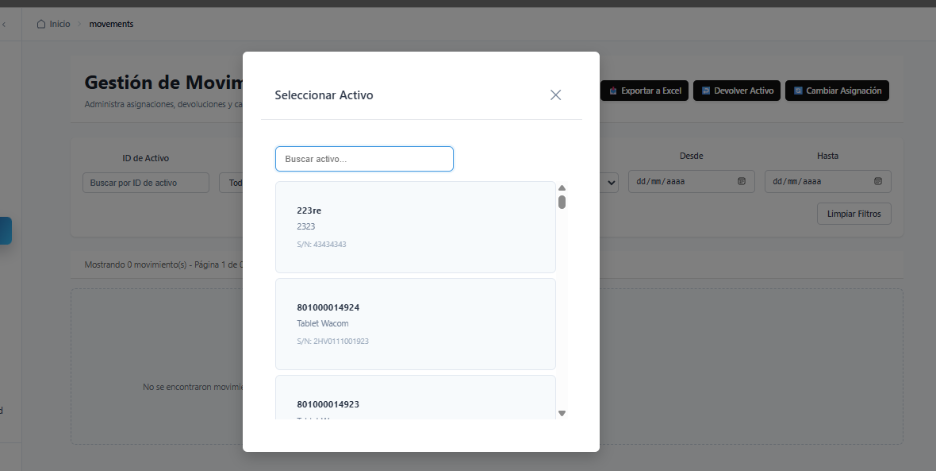


Figure 12.Página del Frontend con el módulo de Movements

**3. Contexto de Soporte (Maintenance Service)**

Enfocado en la operatividad técnica.

* **Maintenance:** Representa un ticket de soporte. Almacena la descripción del problema y la resolución técnica. Se vincula lógicamente al activo mediante el AssetId.
* **MaintenancePhoto:** Gestiona la evidencia visual. Dado que el sistema integra Azure Blob Storage, esta entidad almacena las URLs de las fotografías del estado del equipo antes y después del mantenimiento.

**4. Contexto de Documentación (Template Service)**

Encargado de la formalización legal de los procesos.

* **Acta:** Representa el documento generado (PDF/HTML) que firma el usuario. Vincula un movimiento específico con un formato de documento.
* **Template:** Almacena las plantillas HTML base que se utilizan para renderizar las actas dinámicamente, permitiendo cambiar el formato de los

documentos sin modificar el código fuente.

**7.2.2 Metodología**

Para el desarrollo del Sistema Web de Gestión y Control de Activos Tecnológicos, se adoptó una metodología ágil Scrum, la cual permitió una planificación iterativa, adaptación continua a los requerimientos institucionales y entregas incrementales orientadas a valor.

La elección de Scrum responde a la naturaleza del proyecto, que integra múltiples microservicios, dependencias externas como Microsoft Entra ID, SysAid y Azure Blob Storage, y requiere validación frecuente de funcionalidades por parte de los stakeholders del área de Tecnología de la Información. Scrum proporcionó un marco estructurado para gestionar esta complejidad de forma controlada y medible.

**Metodología Seleccionada: Scrum**

El marco de trabajo Scrum fue aplicado como metodología principal, organizando el desarrollo en sprints de dos semanas, cada uno enfocado en entregar un incremento funcional, estable y demostrable del sistema.

Cada sprint estuvo alineado a los módulos principales de la solución:

* **Módulo de Activos (DataWarehouse Service)**
* **Módulo de Movimientos y Actas (Movements Service + Template Service)**
* **Módulo de Mantenimientos (Maintenance Service)**
* **Módulo de Dashboards y Analítica**
* **Integración con Microsoft Entra ID (SSO) y SysAid**

Cada sprint tuvo como resultado una versión parcial pero funcional del sistema, permitiendo revisiones constantes por parte de stakeholders como:

* Coordinación de Soporte TI
* Mesa de Ayuda
* Asistentes de TI responsables de activos
* Usuarios institucionales finales (pruebas de “Mis activos”)

**Ceremonias aplicadas**

A pesar de tratarse de un proyecto individual, se mantuvieron las ceremonias formales de Scrum:

**• Sprint Planning**

Antes de cada sprint se definieron los Product Backlog Items (PBIs) a desarrollar, priorizados según valor institucional y complejidad técnica.   
Se consideraron dependencias entre microservicios y actividades como:

* Creación de catálogos
* Implementación de endpoints REST
* Conexión con claves Entra ID
* Gestión de actas PDF
* Pruebas en entorno local

**• Daily Meetings (adaptadas)**

Se realizó una autoevaluación diaria del avance del sprint, revisando el estado de los PBIs y las tareas técnicas asociadas.

Este seguimiento permitió identificar bloqueos tempranos y asegurar el cumplimiento de los objetivos del sprint.

**• Sprint Review**

Al finalizar cada sprint se revisó el incremento logrado, validando funcionalidades como:

* Registro de activos
* Movimientos con generación de actas PDF
* Mantenimiento con fotos almacenadas en Azure Blob
* Dashboards en tiempo real
* Gestión para SAP

**• Sprint Retrospective**

Después de cada sprint se analizaron oportunidades de mejora, tales como:

* División de PBIs demasiado grandes
* Ajustes en la planificación de tareas técnicas
* Mejora en la modularidad del frontend y backend
* Optimización de la secuencia de desarrollo entre microservicios

**Artefactos de desarrollo**

Durante el proceso se elaboraron:

**• Product Backlog estructurado en Epics → Features → PBIs → Tasks**

Cada ítem se relacionó con un caso de uso del sistema.

**• Dashboard de Seguimiento en Azure Boards**

Incluyendo burndown charts, progreso por sprint, PBIs completados y estados de avance.

**• Historias de Usuario**

Todas redactadas bajo la estructura:   
**Como [rol], quiero [funcionalidad], para [valor agregado].**

**• Incrementos funcionales por sprint**

Cada entrega correspondió a una parte operativa del sistema.

**Relación con los sprints**

El trabajo se dividió en **cinco sprints**, cuyos productos principales fueron:

* **Sprint 1:** CRUD de Activos
* **Sprint 2:** Movimientos + Actas PDF
* **Sprint 3:** Exportación para SAP
* **Sprint 4:** Registro de Mantenimientos
* **Sprint 5:** Dashboards + SSO + integración final

Todos estos avances se documentan en las siguientes secciones, incluyendo evidencia del incremento entregado, PBIs generados y criterios de aceptación.

**7.2.3 Product Backlog**

A continuación se presenta el Product Backlog oficial del Sistema de Gestión, Trazabilidad y Control de Activos Tecnológicos de la UDLA.   
El backlog se construyó siguiendo la estructura Épica → Feature → Historia de Usuario, permitiendo una trazabilidad clara entre los requerimientos de alto nivel y la implementación técnica en los microservicios.

**7.2.3.1 ÉPICA 1 – Gestión del Módulo de Activos (AssetService)**

**Descripción:** Construcción del microservicio principal encargado del inventario maestro de activos tecnológicos, su CRUD, su validación y su trazabilidad base.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción** | **Historias de Usuario** |
| **F1.1 – CRUD de Activos** | Implementación de creación, edición y validación de activos. | **PBI 1.1.1 – Crear un activo**   **PBI 1.1.2 – Editar un activo existente**   **PBI 1.1.3 – Validar campos del formulario** |
| **F1.2 – Baja de Activos** | Funcionalidad para gestionar el ciclo final del activo. | **PBI 1.2.1 – Dar de baja un activo** |
| **F1.3 – Visualización y Filtros** | Consultas del inventario según sede, estado y categoría. | **PBI 1.3.1 – Visualizar lista de activos**   **PBI 1.3.2 – Aplicar filtros de stock** |

Table 7. Épica 1

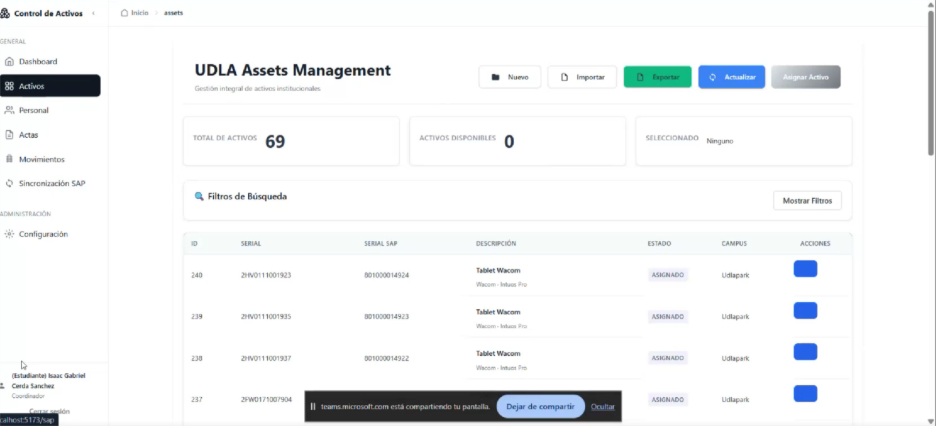


Figure 13. Modulo de activos en funcionamiento

**ÉPICA 2 – Gestión de Movimientos (MovementsService + TemplateService)**

**Descripción:** Administración del ciclo de vida del activo: asignaciones, préstamos, devoluciones y la generación automática de actas PDF. 

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción** | **Historias de Usuario** |
| **F2.1 – Registrar asignaciones y préstamos** | Registra la entrega de activos a colaboradores. | **PBI 2.1.1 – Registrar asignación** |
| **F2.1.2 – Generar acta PDF de asignación** | Generación automática del documento oficial. | **PBI 2.1.2 – Generar acta PDF de asignación** |
| **F2.2 – Registrar préstamo** | Registro temporal del uso del activo. | **PBI 2.2.1 – Registrar préstamo** |
| **F2.3 – Registrar devolución** | Cierra el ciclo del préstamo o asignación. | **PBI 2.3.1 – Registrar devolución** |

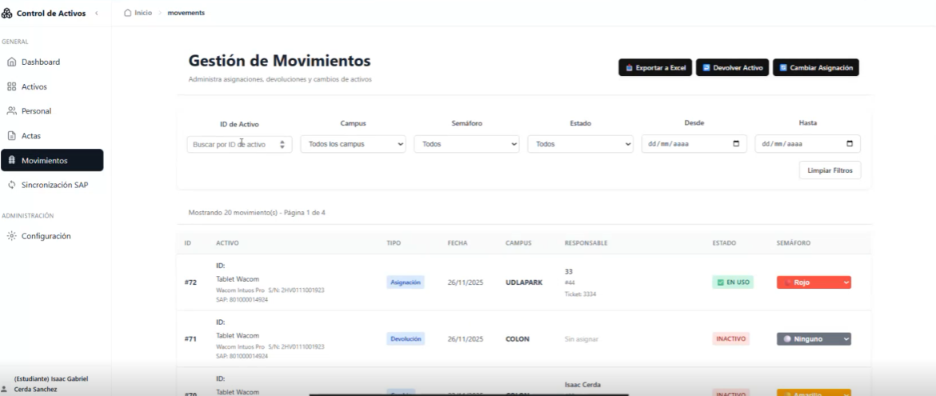
Table 8. Épica 2  


Figure 14. Modulo de movimientos en funcionamiento

**ÉPICA 3 – Gestión de Mantenimientos (MaintenanceService)**

**Descripción:** Registro de mantenimientos correctivos y preventivos con evidencia multimedia y validación entre servicios.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción** | **Historias de Usuario** |
| **F3.1 – Registro de mantenimiento** | Permite documentar la intervención técnica. | **PBI 3.1.1 – Registrar mantenimiento** |
| **F3.2 – Asociación con ticket externo** | Permite enlazar el mantenimiento a un ticket de SysAid. | **PBI 3.2.1 – Asociar ticket al mantenimiento** |

Table 9. Épica 3

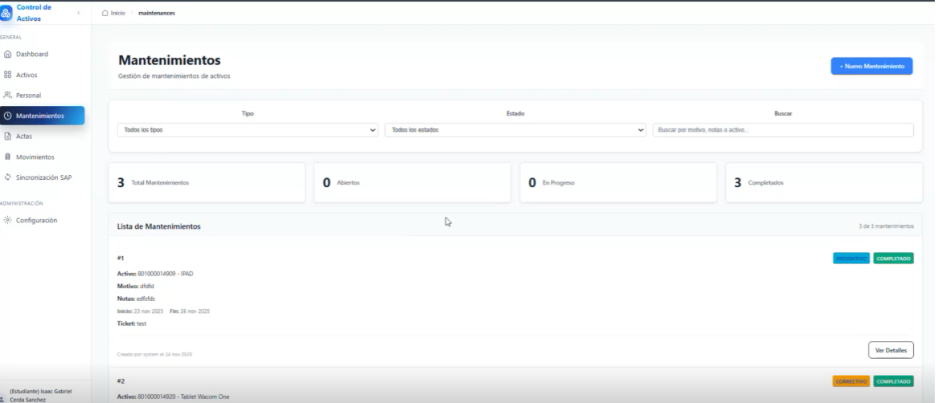


Figure 15. Modulo de mantenimientos en funcionamiento

**ÉPICA 4 – Dashboards y Analítica (Frontend + Aggregations)**

**Descripción:** Visualización en tiempo real de inventario, movimientos, mantenimiento y cumplimiento de políticas.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción** | **Historias de Usuario** |
| **F4.1 – Dashboard de activos** | Vista ejecutiva del inventario. | **PBI 4.1.1 – Ver dashboard de activos** |
| **F4.2 – Dashboard de mantenimientos** | Vista de mantenimientos realizados o pendientes. | **PBI 4.2.1 – Ver mantenimientos realizados/pendientes** |
| **F4.3 – Dashboard de SLAs** | Vista de indicadores de desempeño. | **PBI 4.3.1 – Ver KPIs de SLA** |

Table 10. Épica 4

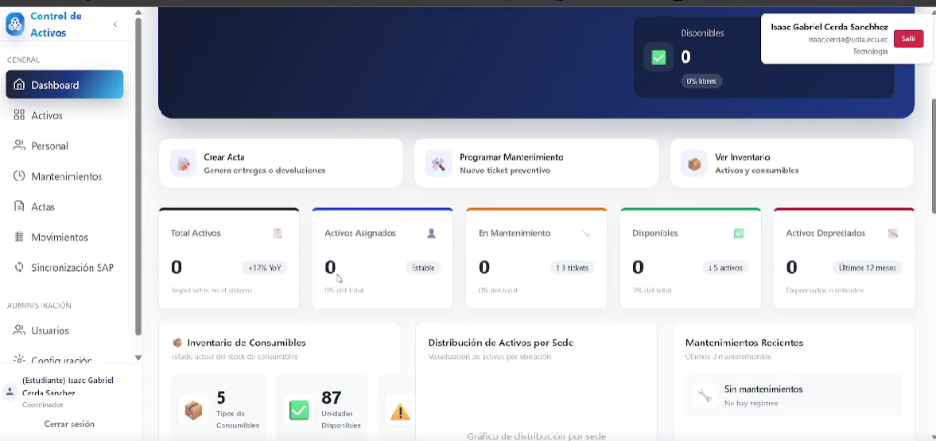


Figure 16. Módulo de dashboard en funcionamiento

**ÉPICA 5 – Exportación y Control para SAP (SAP Export Module)**

**Descripción:** Herramientas para exportar activos completados al sistema SAP institucional.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción** | **Historias de Usuario** |
| **F5.1 – Exportación de activos a SAP** | Genera archivo Excel con activos listos para SAP. | **PBI 5.1.1 – Exportar archivo Excel SAP** |
| **F5.4 – Registro de activos exportados** | Evita duplicidad y mantiene control histórico. | **PBI 5.4.1 – Registrar activos exportados** |

Table 11. Épica 5

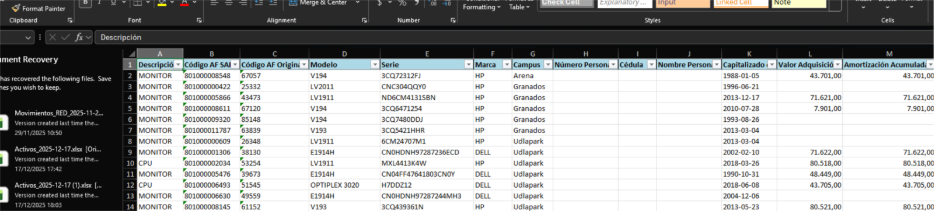


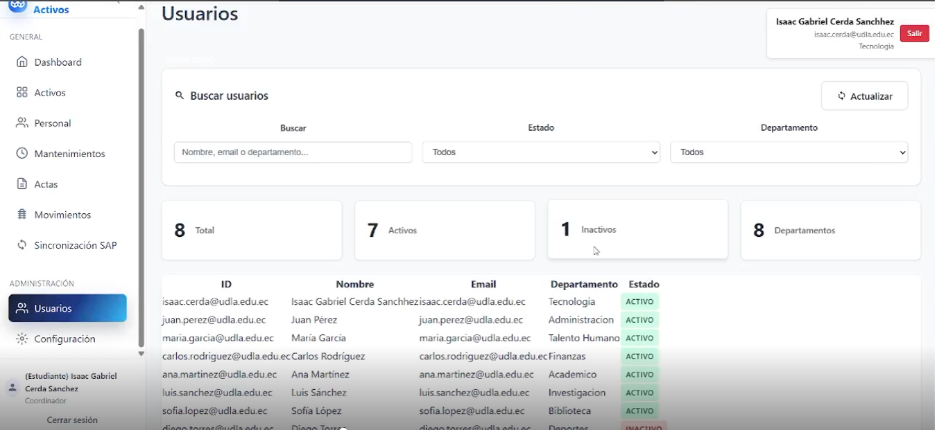
Figure 17. Archivo descargado para subir a SAP

**ÉPICA 6 – Módulo “Mis Activos” (Frontend + API Gateway)**

**Descripción:** Vista personalizada para usuarios finales (docentes/colaboradores) con activos asignados.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción** | **Historias de Usuario** |
| **F6.1 – Activos asignados a usuario** | Consulta del inventario personal. | **PBI 6.1.1 – Ver activos asignados** |
| **F6.2 – Historial del activo** | Trazabilidad del uso del activo. | **PBI 6.2.1 – Ver historial del activo** |

Table 12. Épica 6

Figure 18. Módulo de usuarios en funcionamiento  
**Tabla final: Product Backlog Consolidado**

|  |  |  |
| --- | --- | --- |
| **Épica** | **Features** | **Historias de Usuario** |
| **Gestión del Módulo de Activos** | F1.1, F1.2, F1.3 | PBI 1.1.1 – 1.3.2 |
| **Gestión de Movimientos** | F2.1 – F2.3 | PBI 2.1.1 – 2.3.1 |
| **Gestión de Mantenimientos** | F3.1 – F3.2 | PBI 3.1.1 – 3.2.1 |
| **Dashboards** | F4.1 – F4.3 | PBI 4.1.1 – 4.3.1 |
| **Exportación SAP** | F5.1, F5.4 | PBI 5.1.1 – 5.4.1 |

Table 13. Tabla final: Product Backlog Consolidado

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Figure 19. Backlog en Azure Boards

## **Pruebas y evaluación de la solución**

De manera paralela al desarrollo del proyecto, se ejecutó un proceso de evaluación continua por fases y por sprints, con el propósito de asegurar que cada incremento del sistema sea verificable, funcional y alineado con los requerimientos definidos. Bajo este enfoque, cada producto entregable (módulos, servicios e integraciones) fue revisado y validado antes de considerarse finalizado, reduciendo riesgos y evitando inconsistencias entre componentes.

La etapa de pruebas y evaluación fue clave para garantizar que la solución cumpla las expectativas de los usuarios finales, no solo en términos de funcionamiento, sino también de estabilidad y experiencia de uso. Probar permitió identificar y corregir errores de forma temprana, prevenir regresiones y asegurar que los flujos críticos operen correctamente antes de la implementación final.  
Para lograr una validación integral se aplicaron pruebas unitarias, de integración, de usabilidad y de aceptación. Las unitarias verificaron reglas de negocio y validaciones; las de integración confirmaron el comportamiento correcto entre módulos y contratos de comunicación; las de usabilidad evaluaron claridad, consistencia y eficiencia de la interfaz; y las de aceptación comprobaron, mediante escenarios reales, que cada entrega de sprint cumpliera los criterios establecidos, asegurando una solución robusta y confiable.

**Tipos de pruebas que se aplicarán:**

1. **Pruebas de Integración:**Validan el comportamiento de una unidad específica (función, clase o componente) de forma aislada. Aseguran que la lógica interna cumpla lo esperado y facilitan detectar errores temprano.

**Responsable:**Isaac Cerda

1. **Pruebas Unitarias:**Validan el comportamiento de una unidad específica (función, clase o componente) de forma aislada. Aseguran que la lógica interna cumpla lo esperado y facilitan detectar errores temprano.

**Responsable:**Isaac Cerda

1. **Pruebas de Usabilidad:**Evalúan qué tan fácil, claro y eficiente es usar el sistema para completar tareas reales. Identifican problemas de navegación, comprensión, accesibilidad, consistencia y experiencia de usuario.   
   **Responsable:**Esteban Enríquez
2. **Pruebas de aceptación:**Comprueban que el sistema cumpla los requisitos y criterios de aceptación definidos por el negocio/usuarios.   
   **Responsable:**Esteban Enríquez

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Prueba** | **Paso** | **Descripción** | **Responsable** |
| **Pruebas de Integración** | Paso 1 | Definir el alcance de integración (módulos involucrados, endpoints/servicios y flujo end-to-end) y preparar el ambiente con datos de prueba. | Isaac Cerda |
| Paso 2 | Diseñar casos de prueba con: precondiciones, entradas, pasos, resultado esperado y criterio de aprobación (incluye casos normales y de error). | Isaac Cerda |
| Paso 3 | Ejecutar las pruebas y recolectar evidencia (logs, respuestas, capturas, trazas) registrando incidencias con severidad y prioridad. | Isaac Cerda |
| Paso 4 | Analizar fallos de integración (contratos, validaciones, tiempos, dependencias), corregir y documentar el cambio. | Isaac Cerda |
| Paso 5 | Re-ejecutar pruebas (incluida regresión del flujo) y cerrar la incidencia cuando el resultado sea conforme. | Isaac Cerda |
| **Pruebas unitarias** | Paso 1 | Identificar unidades críticas (servicios, validaciones, reglas de negocio) y definir criterios mínimos (casos positivos, negativos y borde). | Isaac Cerda |
| Paso 2 | Preparar datos de prueba y aislar dependencias con mocks/fakes (sin acceso real a BD/servicios externos). | Isaac Cerda |
| Paso 3 | Implementar pruebas con estructura Arrange–Act–Assert y aserciones claras sobre salidas/errores esperados. | Isaac Cerda |
| Paso 4 | Ejecutar la suite y revisar cobertura/fallos; corregir la lógica o refactorizar para que el comportamiento sea determinístico. | Isaac Cerda |
| Paso 5 | Integrar la ejecución en CI/CD y actualizar pruebas cuando cambien requisitos o contratos. | Isaac Cerda |
| **Pruebas de aceptación** | Paso 1 | Definir criterios de aceptación por requisito/historia de usuario y preparar escenarios con datos reales o representativos. | Esteban Enríquez |
| Paso 2 | Ejecutar los escenarios end-to-end validando resultados contra lo esperado y registrando evidencias e incidencias. | Esteban Enríquez |
| Paso 3 | Aprobar o rechazar la entrega según cumplimiento; corregir hallazgos y repetir hasta cumplir los criterios. | Esteban Enríquez |
| **Pruebas de usabilidad** | Paso 1 | Seleccionar tareas clave y definir el perfil de usuario, objetivos, tiempos esperados y métricas (errores, claridad, satisfacción). | Esteban Enríquez |
| Paso 2 | Realizar la prueba con usuarios observando la interacción, recopilando comentarios y registrando fricciones/puntos de confusión. | Esteban Enríquez |
| Paso 3 | Analizar resultados y aplicar mejoras de UI/UX (priorizar cambios), y revalidar las tareas para confirmar la mejora. | Esteban Enríquez |

**Aprobación inicial de requerimientos:**   
Para la definición y validación inicial de los requerimientos del sistema se realizaron varias reuniones de levantamiento y revisión con la Responsable de TI, en las cuales se estableció el alcance funcional y técnico de la aplicación, así como las necesidades principales que debía cubrir. En estas sesiones se presentaron los avances de manera progresiva, permitiendo ajustar funcionalidades, priorizar entregables y asegurar que lo desarrollado respondiera a lo solicitado desde las primeras etapas del proyecto.

Posteriormente, se desarrollaron reuniones específicas con los equipos de Arquitectura y Seguridad con el fin de identificar restricciones, lineamientos institucionales y criterios de cumplimiento (por ejemplo, políticas de acceso, estándares de seguridad y componentes permitidos). A partir de estas validaciones se definieron decisiones clave sobre lo que era viable implementar, así como cambios necesarios en el diseño inicial; uno de los ajustes más relevantes fue la modificación del esquema de usuarios para integrarlo con Active Directory, garantizando un control de acceso alineado con los estándares de la organización.

Finalmente, se mantuvieron reuniones técnicas con el arquitecto y el responsable de base de datos para consolidar la estructura de la solución, validar el modelo de datos y asegurar la correcta implementación de los procesos críticos. Si bien el proceso se vio afectado por la renuncia de la Responsable de TI, la continuidad del levantamiento y verificación de información se aseguró mediante coordinación con la persona designada como reemplazo, con quien se completaron los requerimientos pendientes y se confirmaron los puntos necesarios para cerrar la fase de definición.

**Pruebas de integración**

Las pruebas de integración se ejecutaron una vez que los módulos principales del sistema web responsive estuvieron completados y estables, con el objetivo de verificar que los componentes funcionen como un todo y que los flujos críticos end-to-end no presenten fallos antes del despliegue. En esta solución era indispensable validar la interacción entre la autenticación SSO, el API Gateway y los microservicios, ya que cualquier inconsistencia podía impactar directamente la operación (por ejemplo: registro/movimiento de activos, generación de actas y consulta de tickets).

Un caso representativo fue comprobar el flujo completo desde el inicio de sesión hasta la gestión operativa: autenticación mediante SSO con Microsoft 365, enrutamiento de solicitudes por el API Gateway, registro de movimientos del activo desde el Servicio de Activos, generación automática de actas en PDF mediante el Servicio de Actas, consulta del ticket asociado a través del Servicio SysAid y preparación de información para su validación/exportación hacia SAP cuando correspondía. La finalidad fue asegurar que los contratos entre servicios, validaciones, permisos por rol y respuestas del sistema se mantuvieran consistentes bajo condiciones normales y escenarios de error.

El proceso se realizó de forma controlada, definiendo primero el alcance de integración (módulos involucrados, endpoints y flujo end-to-end), diseñando casos de prueba con precondiciones, entradas, resultados esperados y criterios de aprobación, y luego ejecutando las pruebas recolectando evidencia (logs, respuestas y capturas). Cada ejecución se documentó en un registro con identificador, componente probado, procedimiento, resultado esperado vs. resultado obtenido, estado (aprobada/rechazada) y observaciones; cuando se detectaron fallos, se corrigieron y se repitieron las pruebas incluyendo regresión del flujo hasta obtener resultados conformes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Nombre** | **Prueba** | **Resultado** | **Estado** | **Observaciones** |
| Identificador de la Prueba | Nombre del módulo | Cual fuel el proceso para realizar las pruebas y verificar que todo funcione | El resultado esperado | Aprobado o rechazado | Notas en caso de fallos |

**Pruebas Unitarias**   
Durante el desarrollo de la solución de gestión de activos, se ejecutó un proceso riguroso de aseguramiento de calidad enfocado en las funcionalidades críticas de los microservicios, mediante la implementación y ejecución sistemática de pruebas unitarias. Estas pruebas se diseñaron para validar el comportamiento de los componentes de forma aislada, asegurando el cumplimiento de los requisitos funcionales, las reglas de negocio y las especificaciones técnicas definidas para el sistema.

El alcance de las pruebas se determinó a partir del conteo de casos identificados por los atributos [Fact] y [Theory] en los proyectos de pruebas de cada microservicio. Adicionalmente, en TemplateService se consideró el conteo de pruebas de integración únicamente como referencia, manteniéndolas separadas del total unitario para preservar la trazabilidad del indicador.

Las pruebas implementadas cubren escenarios representativos y de alta criticidad, incluyendo: flujos exitosos (happy path) con datos válidos, validaciones de restricciones de entrada y reglas de negocio, simulación de dependencias externas mediante mocks (por ejemplo, servicios de almacenamiento y servicios relacionados), casos límite (edge cases), condiciones excepcionales y manejo de errores. Este enfoque permitió verificar que los servicios respondan de manera consistente ante datos inválidos, incompletos o situaciones no previstas, reduciendo el riesgo de fallos en ambientes de producción.

Los resultados fueron documentados en registros estructurados, definiendo para cada caso de prueba las precondiciones, el procedimiento de ejecución, el resultado esperado, el resultado obtenido, la evidencia asociada y el estado final (aprobado/rechazado), junto con observaciones relevantes. Como resultado del proceso, se registran 752 casos de pruebas unitarias distribuidos entre los microservicios del sistema, y 761 casos si se incluyen las pruebas de integración de TemplateService como referencia. La distribución por microservicio y módulo se presenta en la Tabla X, evidenciando una cobertura significativa en servicios clave como AssetsService, MovementsService, MaintenanceService, SysAidService y TemplateService, con estado general Aprobado.

|  |  |  |  |
| --- | --- | --- | --- |
| Microservicio | Módulo | Casos de prueba | Estado |
| AssetsService | Services | 51 | ✓ Aprobado |
| AssetsService | Generales | 1 | ✓ Aprobado |
| MaintenanceService | Controllers | 62 | ✓ Aprobado |
| MaintenanceService | Services | 25 | ✓ Aprobado |
| MaintenanceService | Validators | 8 | ✓ Aprobado |
| MaintenanceService | Middleware | 12 | ✓ Aprobado |
| MovementsService | Services | 131 | ✓ Aprobado |
| MovementsService | Entities | 17 | ✓ Aprobado |
| SysAidService | Services | 84 | ✓ Aprobado |
| SysAidService | Controllers | 65 | ✓ Aprobado |
| TemplateService | Unit | 255 | ✓ Aprobado |
| TemplateService | Services | 12 | ✓ Aprobado |
| TemplateService | Validation | 29 | ✓ Aprobado |
| TemplateService | Integration *(referencial)* | 9 | ✓ Aprobado |
| TOTAL unitario |  | **752** | **✓ Aprobado** |
| TOTAL general (incluye integración) |  | **761** | **✓ Aprobado** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Microservicio** | **Módulo** | **Casos de Prueba** | **Estado** |
| MaintenanceService | Creación de Mantenimientos | 7 | ✓ Aprobado |
| MaintenanceService | Consulta de Mantenimientos | 11 | ✓ Aprobado |
| MaintenanceService | Actualización de Mantenimientos | 8 | ✓ Aprobado |
| MaintenanceService | Gestión de Fotografías | 16 | ✓ Aprobado |
| TemplateService | Creación/Actualización de Plantillas | 13 | ✓ Aprobado |
| TemplateService | Consulta de Plantillas | 19 | ✓ Aprobado |
| TemplateService | Previsualización de PDF | 14 | ✓ Aprobado |
| TemplateService | Gestión de Versiones | 15 | ✓ Aprobado |
| TemplateService | Creación de Actas | 8 | ✓ Aprobado |
| TemplateService | Consulta de Actas | 10 | ✓ Aprobado |
| TemplateService | Generación PDF y Firma Digital | 8 | ✓ Aprobado |
| TemplateService | Actualización/Eliminación de Actas | 10 | ✓ Aprobado |
| **TOTAL** |  | **139** | **✓ Aprobado** |

**7.3.3. Pruebas de usabilidad**

Para la ejecución de las pruebas de usabilidad, se trabajó directamente con usuarios y personal de soporte mediante reuniones de revisión y uso guiado del sistema, con el objetivo de evaluar la experiencia de usuario e identificar oportunidades de mejora en navegación, claridad de la interfaz, consistencia visual y facilidad para completar tareas clave. Durante estas sesiones, los participantes interactuaron con la aplicación en escenarios reales o simulados, mientras se observaban sus acciones para detectar puntos de fricción, pasos confusos y elementos que generaran dudas.

La recolección de información se realizó únicamente a través de las reuniones, sin utilizar encuestas ni formularios. En cada encuentro se documentaron observaciones, comentarios y problemas reportados, priorizando aquellos que afectaban el flujo normal de trabajo. Este enfoque permitió obtener retroalimentación inmediata y contextualizada, además de contrastar el comportamiento esperado con el uso real del sistema.

Con base en los hallazgos de cada reunión, se aplicaron mejoras iterativas orientadas a simplificar procesos, reducir pasos innecesarios, mejorar etiquetas y mensajes, y hacer más intuitiva la ubicación de acciones frecuentes. La participación constante de las personas de soporte facilitó validar el funcionamiento del sistema desde una perspectiva operativa, asegurando que los ajustes realizados respondan a necesidades reales y contribuyan a una solución más accesible y fácil de utilizar. Los cambios y conclusiones obtenidos en estas sesiones se detallan en el anexo correspondiente. **7.3.4. Pruebas de aceptación**

Para la ejecución de las pruebas de aceptación, se realizaron reuniones periódicas con los usuarios responsables y personal de soporte, con el objetivo de confirmar que la solución cumpla con los requerimientos definidos y sea apta para su utilización en el entorno operativo. En estas sesiones se validaron los criterios de aceptación asociados a los procesos principales del sistema, verificando que cada funcionalidad entregue los resultados esperados y que los flujos completos puedan ejecutarse sin inconsistencias.

La validación se llevó a cabo mediante la ejecución guiada de escenarios de uso durante las reuniones, revisando paso a paso casos representativos como: autenticación y acceso por roles, creación y actualización de registros, consulta y filtrado de información, generación de documentos, y registro de movimientos o acciones relevantes. Para cada escenario se documentaron los resultados observados, los hallazgos detectados y las recomendaciones de ajuste, registrando el estado de cada punto evaluado (aprobado, aprobado con observaciones o rechazado).

Con base en la retroalimentación obtenida en cada reunión, se aplicaron correcciones y mejoras hasta alcanzar la conformidad con los requerimientos. Posteriormente, se repitieron los escenarios afectados para asegurar que los cambios solucionen el problema sin introducir nuevos errores. Este enfoque permitió consolidar una versión final estable y alineada con las expectativas de los usuarios.

## **Resultados y Discusión.**

La implementación de la solución web responsive para la gestión, seguimiento y control de activos tecnológicos cumplió con los objetivos y el alcance definidos inicialmente, validándose mediante pruebas funcionales, revisiones iterativas y retroalimentación directa de los actores involucrados. Durante la etapa de verificación se evaluaron los procesos principales del sistema, incluyendo el registro y administración de activos, asignación y devolución, gestión de mantenimientos con evidencias, así como la generación de documentos (plantillas/actas) y sus flujos asociados, confirmando un comportamiento estable y conforme a los requerimientos planteados.

A nivel tecnológico, el uso de React en el frontend y .NET 8 en el backend permitió construir una plataforma moderna, modular y escalable. La arquitectura basada en servicios facilitó la separación de responsabilidades y la evolución independiente de componentes, mejorando la mantenibilidad y reduciendo el impacto de cambios. Adicionalmente, el despliegue mediante Docker aseguró consistencia entre entornos, simplificó la configuración y permitió publicar la solución de forma controlada y reproducible.

En cuanto a la validación con usuarios y soporte, las reuniones periódicas permitieron identificar oportunidades de mejora en navegación, mensajes, pasos del flujo y consistencia de la interfaz, lo que condujo a ajustes incrementales orientados a simplificar tareas frecuentes y mejorar la experiencia de uso. No obstante, se presentó una limitación: no fue posible comprobar completamente la medición y cumplimiento del SLA en los tickets debido a un cambio de responsable, lo que impidió ejecutar las pruebas finales con los criterios y aprobaciones necesarios. A pesar de ello, el proyecto avanzó hasta el despliegue y se validaron satisfactoriamente las demás funcionalidades; como trabajo futuro, queda completar la verificación del SLA una vez establecidos los responsables y criterios definitivos de evaluación.

## **Implicaciones éticas**

El desarrollo e implementación de la solución web para la trazabilidad y control operativo de activos tecnológicos implica consideraciones éticas relacionadas con la responsabilidad institucional, la transparencia y el uso correcto de información operativa. En particular, el sistema se orienta a fortalecer la rendición de cuentas mediante trazabilidad completa, aspecto relevante para auditorías internas y externas y para reducir riesgos de mal uso, pérdida o apropiación indebida de bienes.

**Privacidad y protección de datos operativos**

Aunque el sistema se centra en activos, procesa datos vinculados a personas (por ejemplo, activos asignados a usuarios e historial asociado). Por ello, es éticamente necesario aplicar el principio de mínimo acceso y limitar la exposición de información según rol, considerando que el sistema contempla autenticación SSO y control de acceso por roles. Adicionalmente, se establecen requerimientos de cifrado y seguridad de datos, reduciendo el riesgo de accesos no autorizados o filtraciones.

**Seguridad, entorno de acceso y reducción de superficie de exposición**

Desde una perspectiva ética, restringir el acceso a la plataforma a entornos controlados ayuda a proteger la información y disminuir riesgos. En el alcance definido, se especifica el funcionamiento en intranet/VPN y el acceso desde red interna, lo que refuerza la seguridad y evita exposición innecesaria a redes externas.

**Transparencia, trazabilidad y responsabilidad**

El sistema promueve una cultura de transparencia y trazabilidad al permitir consultar el historial de activos asignados, lo que incentiva el uso adecuado de recursos comunes y mejora la responsabilidad individual y organizacional.

Asimismo, la trazabilidad es un componente ético clave porque reduce arbitrariedad, facilita verificaciones y mejora la rendición de cuentas ante revisiones y auditorías.

**Integración con sistemas externos y principio de minimización**

En integraciones con plataformas de terceros (p. ej., tickets), existe el riesgo de ampliar permisos o manipular información fuera del propósito original. Para mitigar esto, se definió que la interacción con el sistema de tickets sea solo lectura, permitiendo consultar estados sin crear/editar/cerrar tickets desde la solución. Esto reduce el impacto sobre procesos oficiales y limita usos indebidos.

**Uso ético de indicadores y SLA**

El sistema contempla analítica con indicadores (incluyendo SLA) y objetivos de reducción de tiempos de atención. Ética y operativamente, esto exige: (1) que los indicadores se midan con definiciones claras, (2) que no se utilicen para sancionar sin contexto (p. ej., dependencias externas o falta de información), y (3) que existan mecanismos de revisión cuando el dato no pueda validarse completamente.

**Propiedad intelectual y uso de software**

La solución debe respetar licencias y componentes utilizados (frameworks, librerías, plantillas, íconos, etc.) y garantizar que los artefactos generados (como actas/plantillas) se usen únicamente con fines autorizados y bajo políticas institucionales, manteniendo control de versiones y trazabilidad documental según el alcance funcional del sistema.

# Conclusiones y Recomendaciones

# Se logró desarrollar una solución web responsive orientada a la trazabilidad integral y control operativo de activos tecnológicos, automatizando movimientos, mantenimientos y documentación, con el fin de reemplazar el uso de hojas de cálculo como herramienta principal de control.

# El proyecto permitió estructurar e implementar un proceso formal de levantamiento, análisis y validación de requerimientos, lo cual mejoró la comprensión de los procesos operativos reales y redujo ambigüedades durante el desarrollo.

# La solución fue diseñada con una arquitectura funcional y técnica que contempla control de acceso por roles, modularidad, escalabilidad y compatibilidad con sistemas externos como SAP y SysAid, facilitando la integración con el ecosistema institucional existente.

# Se implementó la generación automática de actas en PDF (asignación, devolución y préstamo) mediante plantillas dinámicas, vinculando la documentación al historial del activo y fortaleciendo la trazabilidad documental.

# El sistema se desarrolló de manera iterativa, incorporando validación continua y pruebas durante las iteraciones para asegurar calidad y mejora progresiva.

# Respecto al objetivo de reducir el tiempo promedio de atención (SLA), se avanzó en el diseño e instrumentación de indicadores y analítica; sin embargo, la verificación completa del SLA puede depender de disponibilidad de responsables y condiciones organizacionales para ejecutar pruebas de aceptación específicas.

# Recomendaciones

# Definir desde el inicio el stack tecnológico y las integraciones críticas (SSO, SysAid, SAP, almacenamiento, generación de PDFs y despliegue), ya que una definición temprana reduce retrabajo, incompatibilidades y cambios tardíos de arquitectura.

# Mantener un proceso sólido de requerimientos y priorización (alcance claro, criterios de aceptación por requerimiento y priorización MoSCoW), porque facilita el control del proyecto y asegura que lo entregado realmente simplifique el trabajo diario de quienes operan el sistema.

# Para asegurar la medición real del SLA, se recomienda acordar formalmente: definición del indicador, fuente de datos, responsables, y un plan de pruebas de aceptación. Esto evita bloqueos por cambios de personal y mejora la continuidad operativa.

# Fortalecer la estrategia de pruebas (unitarias, integración, regresión y rendimiento) y monitorear el cumplimiento de requisitos no funcionales como tiempos de respuesta por operación, seguridad y usabilidad, para sostener la calidad en producción.

# Implementar un plan de adopción que incluya capacitación, guías rápidas y documentación técnica/operativa, para facilitar el uso y reducir dependencia del equipo de desarrollo en tareas repetitivas.

# Considerar desde la planificación que, además de lo técnico, existen procesos internos y aprobaciones (burocracia) que influyen en tiempos y validaciones; por ello, es recomendable reservar tiempo para aprendizaje del contexto, alineación con soporte y coordinación con áreas dueñas de procesos (por ejemplo, tickets e importaciones).

# Trabajo futuro

A partir de los resultados obtenidos con la implementación de los módulos principales del sistema (gestión de activos, movimientos, mantenimientos, generación de actas y despliegue en contenedores), se identifican varias líneas de evolución que pueden desarrollarse como mejoras incrementales o como proyectos complementarios. Estas iniciativas buscan fortalecer la trazabilidad, aumentar la automatización operativa, mejorar la experiencia de usuario y consolidar la integración con plataformas externas.

Validación completa y automatización del SLA de tickets: completar la verificación del cálculo de SLA y sus reglas de negocio (inicio/pausa/cierre, horarios laborales, estados que contabilizan) una vez definidos los responsables del proceso. Como mejora adicional, incorporar tableros con alertas por vencimiento y reportes de cumplimiento por categoría, prioridad o área.

* **Aplicación móvil o modo “mobile-first” avanzado:** extender la solución a una experiencia móvil (PWA o app) enfocada en operaciones rápidas como inventario, reasignación, devoluciones, verificación de estado y consulta de historial; útil para trabajo en campo o bodegas.
* **Inventario con códigos QR / códigos de barras y/o RFID:** integrar lectura por cámara/lector para identificar activos y consumibles, reducir errores de digitación y acelerar procesos de control físico. Esto puede incluir impresión/gestión de etiquetas y auditorías guiadas por escaneo.
* **Módulo de auditoría e inventarios cíclicos:** implementar campañas de inventario por zonas/ubicaciones con conteos, conciliación automática, hallazgos (faltantes/sobrantes), evidencias y cierre con acta. Esto reforzaría control y trazabilidad.
* **Flujos de aprobación y firma:** fortalecer el flujo de aprobación para asignaciones/devoluciones/préstamos y mantenimientos (por rol o jerarquía), incluyendo notificaciones, trazabilidad de aprobaciones y firma digital cuando aplique.
* **Analítica y reportes ejecutivos:** ampliar el módulo de indicadores con métricas de rotación, antigüedad, costos de mantenimiento, frecuencia de fallas, tiempos de atención, disponibilidad por ubicación y cumplimiento de políticas. Idealmente con exportación programada (PDF/Excel) y dashboards por perfil.
* **Integración más profunda con sistemas externos:** estandarizar sincronizaciones con sistemas corporativos (por ejemplo, exportación/conciliación de inventario, centros de costo, responsables, ubicaciones) y definir un modelo de datos “fuente de verdad” para evitar duplicidad y errores.
* **Seguridad avanzada**: añadir controles como autenticación multifactor (MFA), políticas de sesión, auditoría de accesos, y permisos más granulares por operación (lectura/escritura/aprobación). También reforzar cifrado, gestión de secretos y hardening de infraestructura.
* **Observabilidad y operación:** incorporar monitoreo centralizado (logs, métricas y trazas), alertas por fallos de microservicios, dashboards de salud, y seguimiento de errores para soporte. Esto mejora estabilidad en producción y reduce tiempos de diagnóstico.
* **Optimización de rendimiento y escalabilidad:** realizar pruebas de carga y ajustes (caching, paginación, índices, colas para tareas pesadas como generación de PDFs), para asegurar tiempos de respuesta consistentes en escenarios de alta concurrencia.
* **Automatización de procesos y reducción de burocracia**: documentar y formalizar procedimientos internos (políticas, responsables, plantillas) y automatizar tareas repetitivas (recordatorios, vencimientos, renovaciones, devoluciones pendientes). Esto reduce fricción operativa y facilita adopción.

# Referencias bibliográficas

Geresics-Földi, E. (2023). Database and dashboard design of a CRM/BI application for a state export agency [PDF].

Hosen, M. S., et al. (2024). Data-driven decision making: Advanced database systems for business intelligence [PDF].

Kumar, N. (2022). IoT-enabled real-time data integration in ERP systems [PDF].

Schwaber, K., & Sutherland, J. (2020, November). The Scrum Guide: The definitive guide to Scrum: The rules of the game. Scrumguides.org.

Fowler, M., & Lewis, J. (2014, March 25). Microservices: A definition of this new architectural term. MartinFowler.com.

Newman, S. (2021). Building microservices: Designing fine-grained systems (2nd ed.). O’Reilly Mdia.

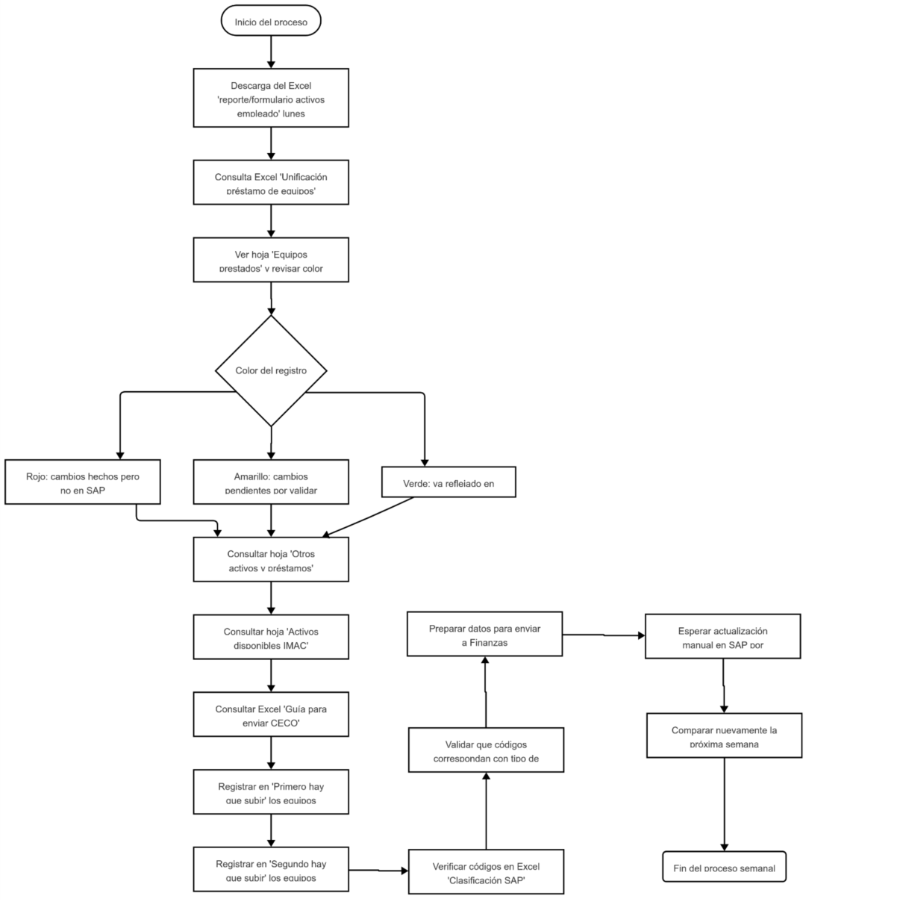
Richardson, C. (2018). Microservices patterns: With examples in Java. Manning Publications.

Dragoni, N., et al. (2016). Microservices: Yesterday, today, and tomorrow. arXiv. <https://arxiv.org/abs/1606.04036>

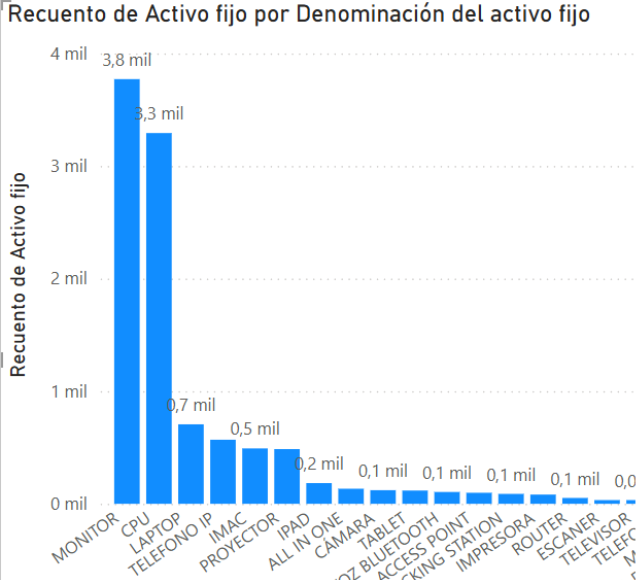
Las 4 primeras (Kumar/Hosen/Geresics-Földi y tu tesis) están tomadas del PDF.

# Anexos

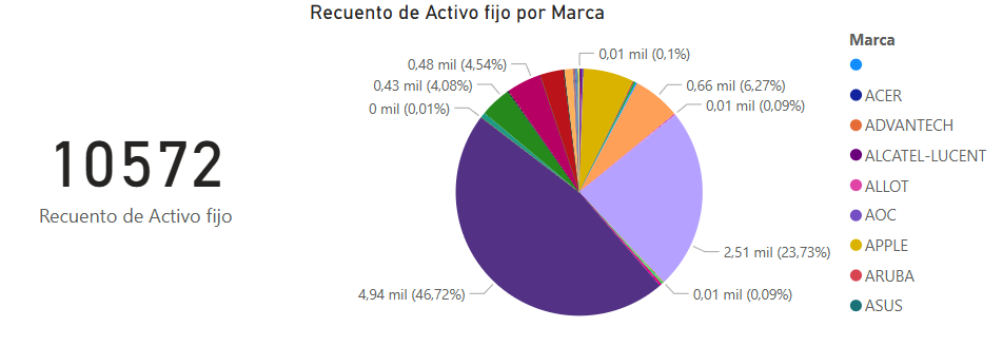
* 1. **Anexo 1:  Diagrama de flujo actual proceso con el que se llevan los activos**

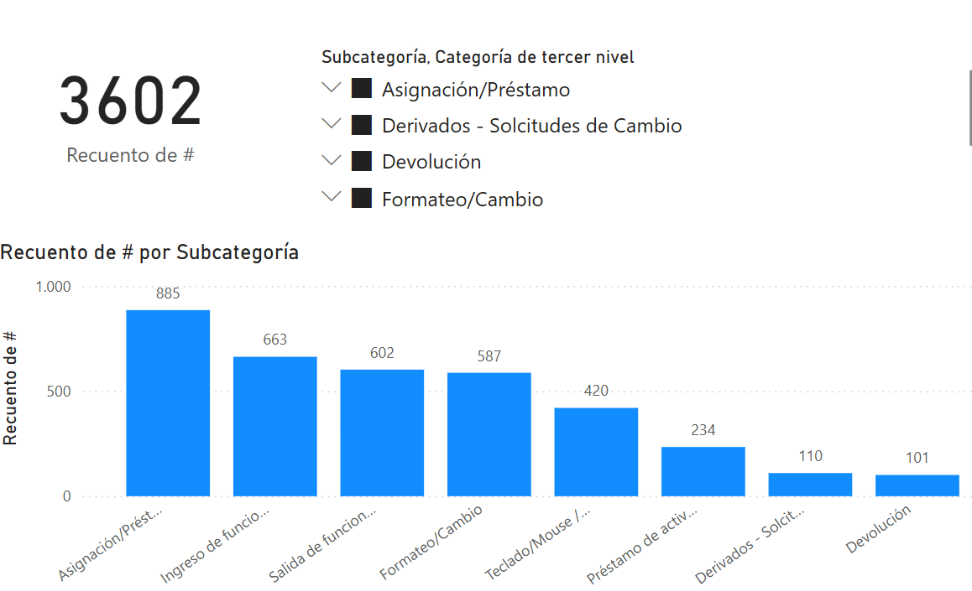


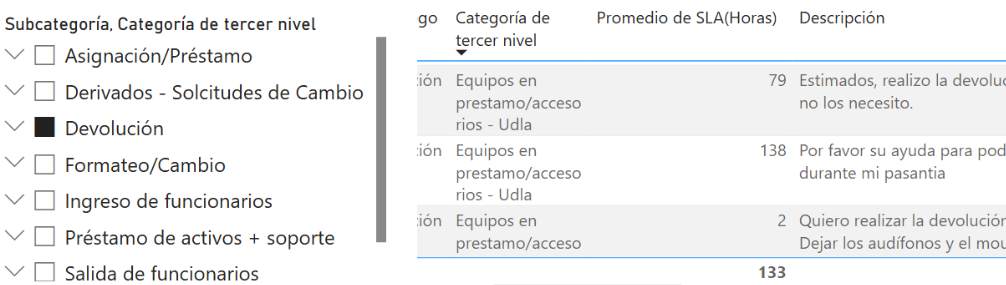
* 1. **Anexo 2: Recuento de Activos Fijos por Denominación del activo fijo**



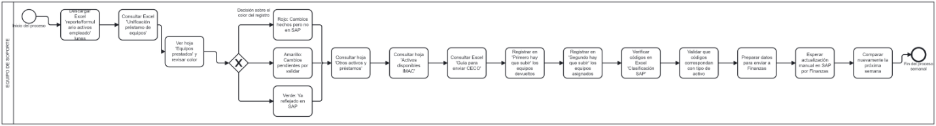
* 1. **Anexo 3: Recuento de Activos Fijos y Recuento de Activos Fijos por Marca**



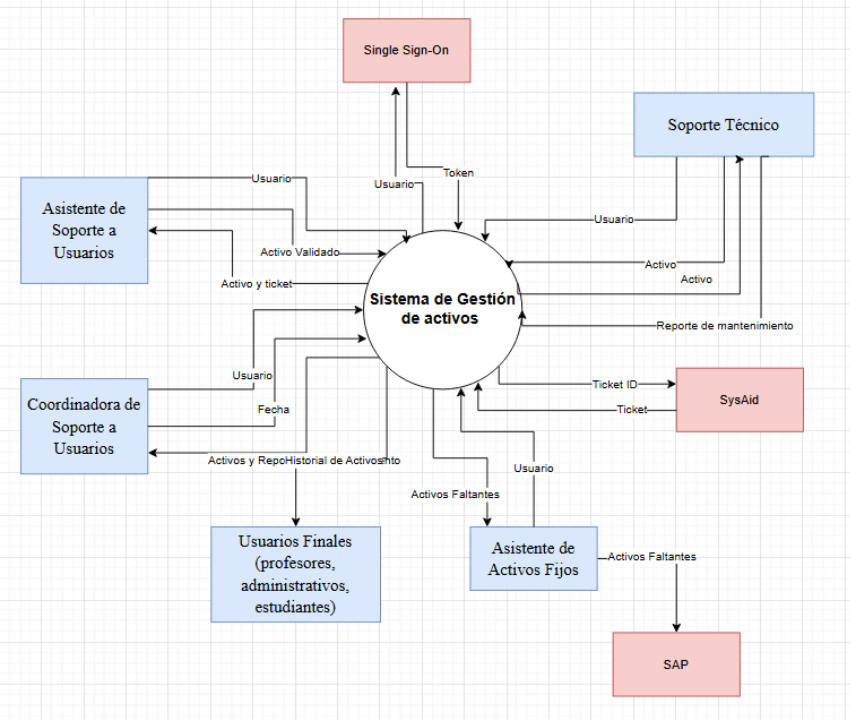
* 1. **Anexo 4: Recuento de tickets relacionados con activos y desglose del número de tickets por subcategoría.**
  2. **Anexo 5: Tiempo de Respuesta Promedio (SLA) de la Subcategoría "Devolución"**



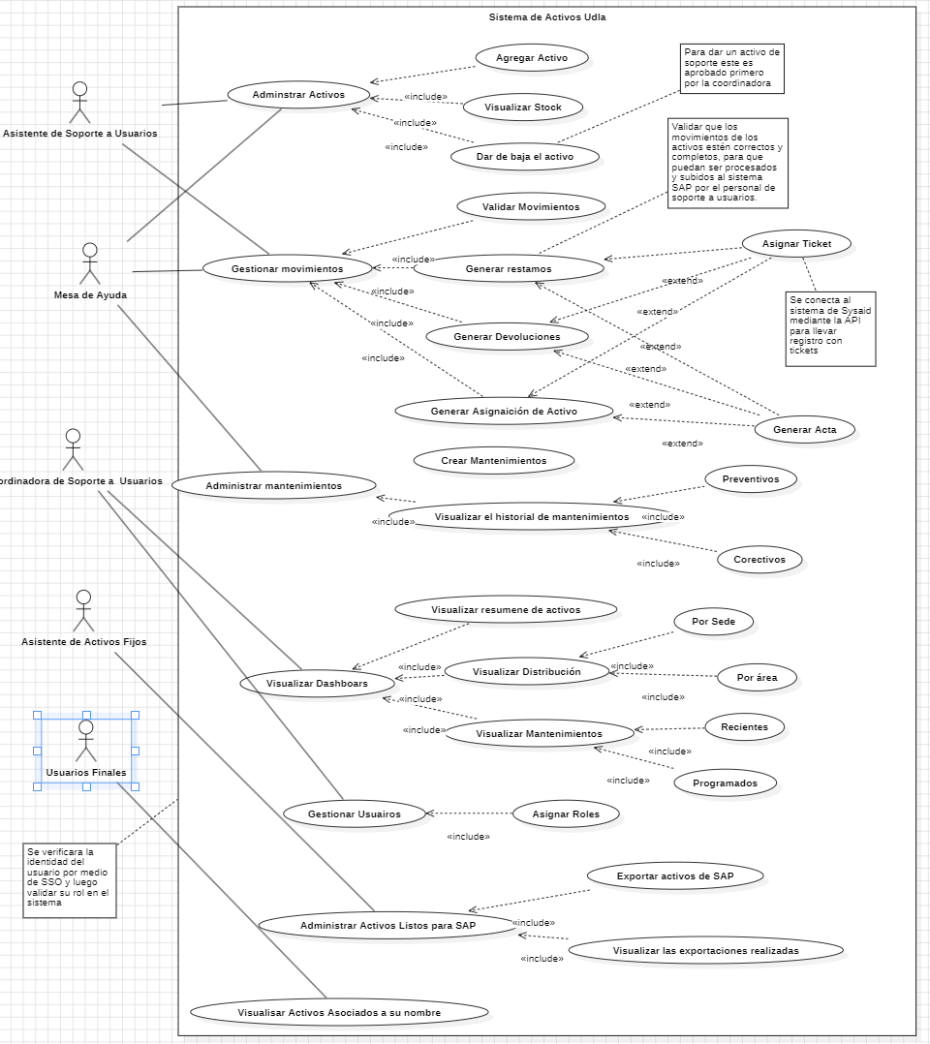
* 1. **Anexo 6: Diagrama BPMN flujo de control activos**



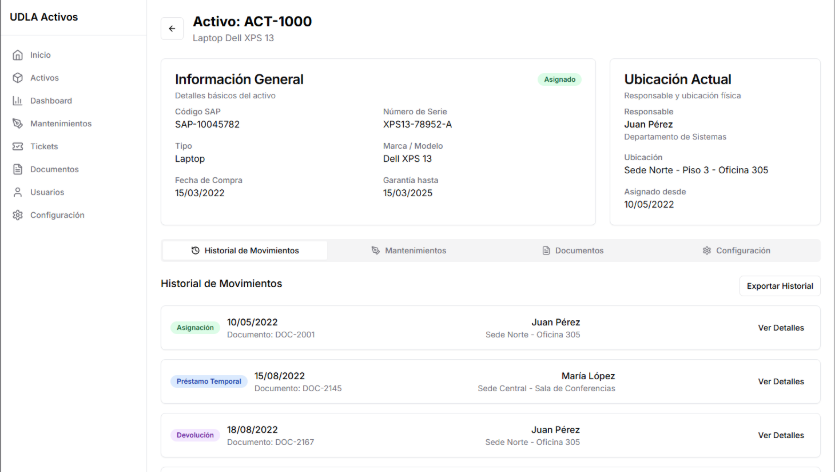
* 1. **Anexo 7: Alternativa 1 diagrama de Contexto**



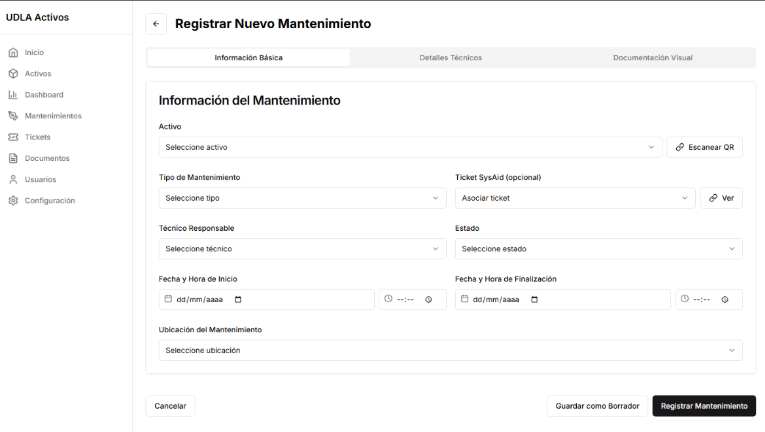
* 1. **Anexo 8: Diagrama de Casos de Uso**

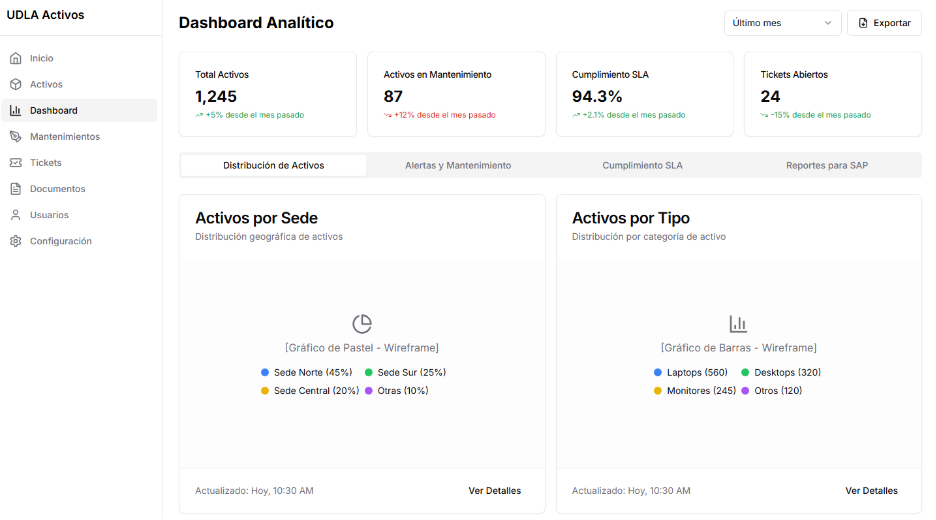
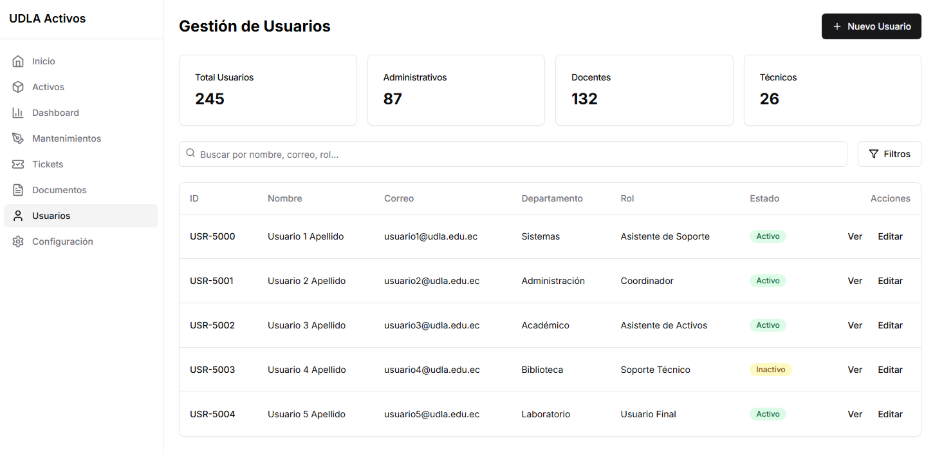


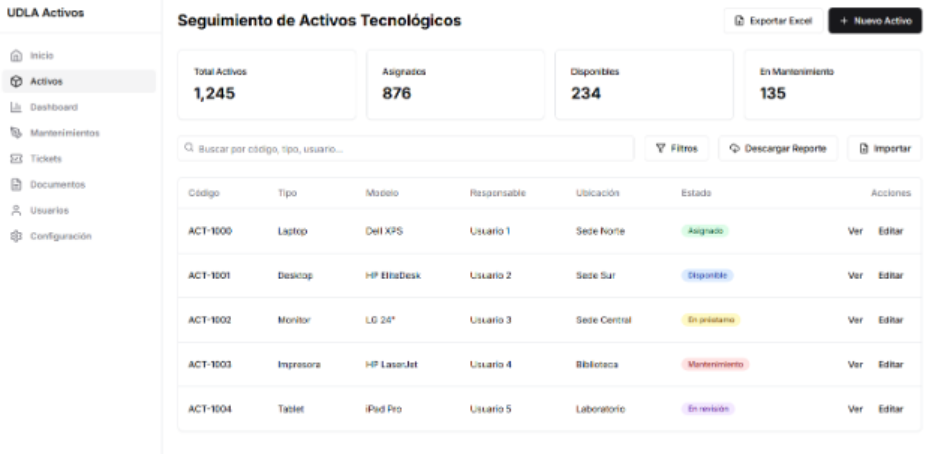
* 1. **Anexo 9. Modulo de activos**



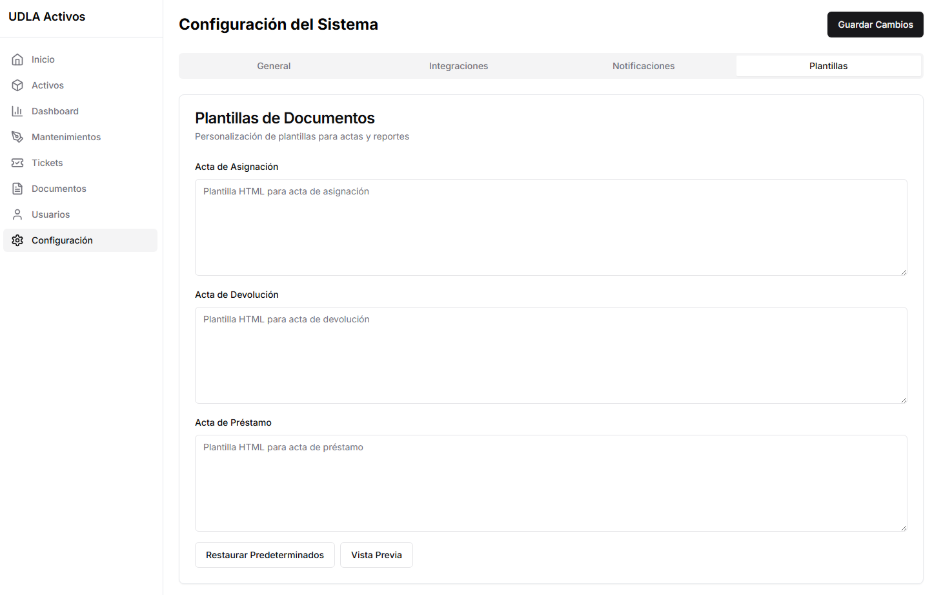
* 1. **Anexo 10: Módulo de mantenimientos parte 1**



* 1. **Anexo 11: Módulo de mantenimientos parte 2**
  2. **Anexo 12: Módulo de Dashboards**
  3. **Anexo 13. Módulo de gestión de usuario**
  4. **Anexo 14. Módulo de exportación de SAP del sistema de seguimiento de activos tecnológicos.**



* 1. **Anexo 15. Panel de configuración de plantillas de actas del sistema de activos tecnológicos.**



* 1. **Anexo 16. UC01**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[UC01]** | | | **Administrar Activos** | |
| **Descripción** | El asistente de soporte a usuarios que maneja los activos accede al sistema para administrar activos tecnológicos con el fin de mantener actualizado su estado, ubicación y trazabilidad. Esto incluye agregar, editar, dar de baja activos y visualizar el stock | | | | |
| **Actores** | * Asistente de Soporte a Usuarios * Coordinadora de Soporte | | | | |
| **Pre**  **condiciones** | * El usuario debe estar autenticado en el sistema mediante SSO institucional. * Debe contar con rol autorizado para modificar activos. | | | | |
| **Post condiciones** | * El activo queda registrado o actualizado en la base de datos. * Se registra el historial del movimiento. * Se genera acta si corresponde. | | | | |
| **Secuencia Normal** | # | Acción (actor) | | Reacción (sistema) | |
| 1 | El asistente accede al módulo de administración de activos. | | El sistema muestra las opciones disponibles. | |
| 2 | El actor selecciona “Agregar Activo”. | | El sistema muestra el formulario correspondiente. | |
| 3 | El actor ingresa los datos del activo. | | El sistema valida y guarda la información. | |
| 4 | El actor puede editar un activo ya registrado. | | El sistema actualiza la información y registra el cambio. | |
| 5 | El actor puede dar de baja un activo. | | El sistema solicita confirmación, genera acta y actualiza estado.  ● 5.1 Si el activo tiene pendientes, el sistema deberá mostrar advertencia y bloquear la acción  ● 5.2 Si el activo está listo, el sistema deberá generar acta de baja y actualizar base de datos | |
| **Excepciones** | # | Acción (actor) | | Reacción (sistema) | |
|  | p | En caso de pérdida de conexión, el sistema deberá mostrar mensaje de error y guardar temporalmente los datos en caché. | | En el caso de que se pierda la conexión, el sistema deberá almacenar la información en caché local y reintentar la operación una vez restablecida la conexión. | |
|  | Q | En caso de ingreso incompleto, el sistema deberá solicitar corrección antes de continuar. | | En el caso de que falte un dato obligatorio, el sistema deberá resaltar los campos faltantes y bloquear el envío hasta que se completen. | |
|  | r | En caso de intento de baja de un activo con un ticket abierto asociado. | | El sistema deberá bloquear la operación y notificar al usuario que debe cerrar el ticket en SysAid antes de proceder. | |
| **Rendimiento** | El sistema deberá realizar las acciones descritas en los pasos 1 al 5 en un máximo de 5 segundos por operación individual. | | | | |
| **Frecuencia** | Este caso de uso se espera que se lleve a cabo una media de 10 veces al día. | | | | |
| **Importancia** | Vital | | | | |
| **Urgencia** | Inmediatamente | | | | |
| **Comentarios** | Este caso de uso representa la columna vertebral del sistema, pues centraliza las acciones sobre el ciclo de vida del activo. Las validaciones y actualizaciones están sujetas a aprobación de la Coordinadora de Soporte si se trata de dar de baja el equipo. | | | | |

* 1. **Anexo 17. UC02**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[UC02]** | | | **Gestionar Movimientos** | |
| **Descripción** | La Mesa de Ayuda y el Asistente de Soporte a Usuarios acceden al sistema para registrar asignaciones, préstamos o devoluciones de activos tecnológicos. Cada movimiento genera un ticket en SysAid y una acta en PDF como respaldo documental del proceso. | | | | |
| **Actores** | * Mesa de Ayuda * Asistente de Soporte a Usuarios * Coordinadora de Soporte (valida movimientos) * SysAid (actor externo) | | | | |
| **Pre**  **condiciones** | * El usuario debe estar autenticado mediante SSO. * El activo debe estar registrado y disponible para movimiento. * El usuario destino del activo debe existir y tener ROL válido. | | | | |
| **Post condiciones** | * El movimiento queda registrado con sello de tiempo. * Se genera ticket en SysAid asociado al activo. * Se genera acta dinámica en PDF lista para firma. | | | | |
| **Secuencia Normal** | # | Acción (actor) | | Reacción (sistema) | |
| 1 | El actor accede al módulo “Gestionar Movimientos”. | | El sistema muestra las opciones: asignación, préstamo o devolución. | |
| 2 | El actor selecciona el tipo de movimiento. | | El sistema muestra el formulario correspondiente. | |
| 3 | El actor selecciona el activo y completa los datos del usuario destino. | | El sistema valida disponibilidad del activo y existencia del usuario. | |
| 4 | El actor confirma el movimiento. | | El sistema registra el movimiento y genera ticket en SysAid. | |
| 5 | El sistema extiende a “Generar Acta”. | | Se crea y almacena automáticamente el PDF del acta con los datos ingresados. | |
| **Excepciones** | # | Acción (actor) | | Reacción (sistema) | |
|  | p | El actor intenta registrar un préstamo sobre un activo ya asignado a otro usuario. | | El sistema bloquea la operación y muestra mensaje de conflicto. | |
|  | Q | El actor intenta registrar un movimiento sin seleccionar usuario destino. | | El sistema solicita corrección de los campos obligatorios. | |
|  | r | En caso de fallo de conexión con SysAid al generar ticket. | | El sistema guarda el movimiento en espera y alerta al usuario para reintento manual. | |
| **Rendimiento** | Las acciones descritas entre los pasos 1 y 5 deberán ejecutarse en un máximo de 8 segundos por operación. | | | | |
| **Frecuencia** | Este caso de uso se espera que se ejecute entre 20 y 40 veces por semana. | | | | |
| **Importancia** | Vital | | | | |
| **Urgencia** | Inmediatamente | | | | |
| **Comentarios** | La generación del acta es automática e inmediata tras registrar el movimiento, y el número de ticket queda vinculado para trazabilidad desde el historial del activo. | | | | |

* 1. **Anexo 18. UC03**

|  |  |  |  |
| --- | --- | --- | --- |
| **[UC04]** | **Visualizar Dashboards** | | |
| **Descripción** | La Coordinadora de Soporte, el Asistente de Activos Fijos y usuarios autorizados acceden al módulo de dashboards para visualizar de forma analítica los activos por sede, tipo y estado; el resumen de mantenimientos realizados o pendientes, y las distribuciones por área o alertas asociadas al cumplimiento de SLA. Esta información permite tomar decisiones informadas sobre el estado del parque tecnológico. | | |
| **Actores** | * Coordinadora de Soporte * Asistente de Activos Fijos * Sistema (componente analítico interno) | | |
| **Pre**  **condiciones** | * El actor debe haber iniciado sesión con credenciales institucionales. * El actor debe contar con permisos de lectura sobre los dashboards. * Debe existir data registrada en el sistema (activos, mantenimientos, movimientos). | | |
| **Post condiciones** | * Se generan visualizaciones con los datos actualizados en tiempo real. * El actor podrá exportar parte de los datos (solo si tiene permisos). * Se actualizan los indicadores de cumplimiento de SLA en base a los tickets y mantenimientos registrados. | | |
| **Secuencia Normal** | # | Acción (actor) | Reacción (sistema) |
| 1 | El actor accede al módulo “Visualizar Dashboards”. | El sistema carga los componentes visuales y las opciones disponibles. |
| 2 | El actor selecciona una vista: resumen de activos, distribución por sede, por área o mantenimientos. | El sistema consulta y filtra los datos en tiempo real desde Sql Server. |
| 3 | El actor aplica filtros (por sede, área, fecha o tipo de activo). | El sistema actualiza dinámicamente las gráficas e indicadores.  ● 3.1 Si no existen datos para el filtro aplicado, el sistema deberá {mostrar mensaje: “No existen registros para este criterio”}. |
| 4 | El actor consulta el cumplimiento de SLA u otros KPI de mantenimiento. | El sistema calcula los valores y muestra alertas si hay desviaciones.  ● 4.1 Si el indicador de SLA está por debajo del umbral definido, el sistema deberá {resaltar el KPI en color de advertencia}. |
| 5 | (Opcional) El actor exporta datos a Excel para análisis externo. | El sistema genera un archivo estructurado con los datos filtrados. |
| **Excepciones** | # | Acción (actor) | Reacción (sistema) |
|  | p | Si el actor no tiene permisos sobre ciertos filtros o exportaciones. | El sistema oculta esas opciones o las bloquea con un mensaje de acceso restringido. |
|  | Q | Si se produce un error al obtener los datos (Mongo caído o sin conexión). | El sistema muestra mensaje de error y sugiere reintentar más tarde. |
|  | r | Si se supera el tiempo de espera al cargar métricas pesadas. | El sistema cancelará la carga e informará que se intentará automáticamente más tarde. |
| **Rendimiento** | El sistema deberá cargar los dashboards en menos de 8 segundos con filtros aplicados. | | |
| **Frecuencia** | Consultado diariamente por usuarios administrativos y semanalmente por soporte técnico. | | |
| **Importancia** | Importante (clave para la toma de decisiones operativas). | | |
| **Urgencia** | Puede esperar (no bloquea procesos, pero apoya control estratégico). | | |
| **Comentarios** | Este módulo funciona como una capa visual de consulta que no afecta la integridad de los datos; permite identificar cuellos de botella operativos o incumplimientos de mantenimiento por sede o área. | | |

* 1. **Anexo 19. UC04**

|  |  |  |  |
| --- | --- | --- | --- |
| **[UC04]** | **Visualizar Dashboards** | | |
| **Descripción** | La Coordinadora de Soporte, el Asistente de Activos Fijos y usuarios autorizados acceden al módulo de dashboards para visualizar de forma analítica los activos por sede, tipo y estado; el resumen de mantenimientos realizados o pendientes, y las distribuciones por área o alertas asociadas al cumplimiento de SLA. Esta información permite tomar decisiones informadas sobre el estado del parque tecnológico. | | |
| **Actores** | * Coordinadora de Soporte * Asistente de Activos Fijos * Sistema (componente analítico interno) | | |
| **Pre**  **condiciones** | * El actor debe haber iniciado sesión con credenciales institucionales. * El actor debe contar con permisos de lectura sobre los dashboards. * Debe existir data registrada en el sistema (activos, mantenimientos, movimientos). | | |
| **Post condiciones** | * Se generan visualizaciones con los datos actualizados en tiempo real. * El actor podrá exportar parte de los datos (solo si tiene permisos). * Se actualizan los indicadores de cumplimiento de SLA en base a los tickets y mantenimientos registrados. | | |
| **Secuencia Normal** | # | Acción (actor) | Reacción (sistema) |
| 1 | El actor accede al módulo “Visualizar Dashboards”. | El sistema carga los componentes visuales y las opciones disponibles. |
| 2 | El actor selecciona una vista: resumen de activos, distribución por sede, por área o mantenimientos. | El sistema consulta y filtra los datos en tiempo real desde MongoDB. |
| 3 | El actor aplica filtros (por sede, área, fecha o tipo de activo). | El sistema actualiza dinámicamente las gráficas e indicadores.  ● 3.1 Si no existen datos para el filtro aplicado, el sistema deberá {mostrar mensaje: “No existen registros para este criterio”}. |
| 4 | El actor consulta el cumplimiento de SLA u otros KPI de mantenimiento. | El sistema calcula los valores y muestra alertas si hay desviaciones.  ● 4.1 Si el indicador de SLA está por debajo del umbral definido, el sistema deberá {resaltar el KPI en color de advertencia}. |
| 5 | (Opcional) El actor exporta datos a Excel para análisis externo. | El sistema genera un archivo estructurado con los datos filtrados. |
| **Excepciones** | # | Acción (actor) | Reacción (sistema) |
|  | p | Si el actor no tiene permisos sobre ciertos filtros o exportaciones. | El sistema oculta esas opciones o las bloquea con un mensaje de acceso restringido. |
|  | Q | Si se produce un error al obtener los datos (Mongo caído o sin conexión). | El sistema muestra mensaje de error y sugiere reintentar más tarde. |
|  | r | Si se supera el tiempo de espera al cargar métricas pesadas. | El sistema cancelará la carga e informará que se intentará automáticamente más tarde. |
| **Rendimiento** | El sistema deberá cargar los dashboards en menos de 8 segundos con filtros aplicados. | | |
| **Frecuencia** | Consultado diariamente por usuarios administrativos y semanalmente por soporte técnico. | | |
| **Importancia** | Importante (clave para la toma de decisiones operativas). | | |
| **Urgencia** | Puede esperar (no bloquea procesos, pero apoya control estratégico). | | |
| **Comentarios** | Este módulo funciona como una capa visual de consulta que no afecta la integridad de los datos; permite identificar cuellos de botella operativos o incumplimientos de mantenimiento por sede o área. | | |

* 1. **Anexo 20. UC05**

|  |  |  |  |
| --- | --- | --- | --- |
| **[UC05]** | **Administrar Activos para SAP** | | |
| **Descripción** | El Asistente de Activos Fijos accede al módulo de administración de activos listos para SAP, desde donde valida y marca los activos tecnológicos que han sido correctamente gestionados en el sistema (movimientos aprobados, sin tickets abiertos y con información completa), generando archivos estructurados en formato Excel para su posterior importación manual en SAP. Esto permite sincronizar ambos sistemas sin errores ni duplicidad. | | |
| **Actores** | * Asistente de Activos Fijos * Coordinadora de Soporte (validación de movimientos) * Sistema SAP (actor externo) | | |
| **Pre**  **condiciones** | * El activo debe contar con historial actualizado, sin tickets abiertos asociados en SysAid. * Los movimientos del activo deben haber sido validados por la Coordinadora de Soporte. * El usuario debe tener rol con privilegios de exportación a SAP. | | |
| **Post condiciones** | * Se genera y almacena un archivo Excel con los activos listos para SAP. * El sistema registra qué activos fueron exportados, para no repetir la operación. * Se notifica al usuario sobre el éxito de la exportación y su ubicación. | | |
| **Secuencia Normal** | # | Acción (actor) | Reacción (sistema) |
| 1 | El Asistente de Activos Fijos accede al módulo “Administrar Activos para SAP”. | El sistema carga los activos con movimientos completados y sin conflictos. |
| 2 | El actor aplica filtros (fecha, sede, tipo de activo). | El sistema presenta la lista filtrada de activos exportables. |
| 3 | El actor selecciona los activos y pulsa “Exportar”. | El sistema valida que cada activo cumpla las condiciones. |
| 4 | Si todo es correcto, se genera el archivo Excel compatible con SAP. | El archivo se almacena en la carpeta definida y se actualiza el historial del activo. |
| 5 | El actor exporta datos a Excel para análisis externo. | El sistema genera un archivo estructurado con los datos filtrados. |
| **Excepciones** | # | Acción (actor) | Reacción (sistema) |
|  | p | Si se intenta exportar sin seleccionar activos. | El sistema oculta esas opciones o las bloquea con un mensaje de acceso restringido. |
|  | Q | Si el archivo no se puede generar por fallo de permisos. | El sistema muestra mensaje de error y sugiere reintentar más tarde. |
|  | r | Si hay conflicto entre estados del activo y el historial. | El sistema no permitirá exportar hasta que se corrija la inconsistencia. |
| **Rendimiento** | La exportación de hasta 100 activos deberá completarse en menos de 6 segundos. | | |
| **Frecuencia** | Se espera que este proceso se rdealice semanalmente o quincenalmente. | | |
| **Importancia** | Vital (sin esta operación no se refleja el inventario en SAP). | | |
| **Urgencia** | Inmediatamente (forma parte del control financiero institucional). | | |
| **Comentarios** | Este módulo requiere sincronización entre áreas operativas y administrativas. Si bien no realiza conexión directa con SAP, su diseño permite una transición limpia y controlada entre sistemas. | | |

* 1. **Anexo 21. UC06**

|  |  |  |  |
| --- | --- | --- | --- |
| **[UC06]** | **Visualizar Activos Asociados a su Nombre** | | |
| **Descripción** | Los usuarios finales de la Universidad (docentes, personal administrativo o estudiantes) acceden a la plataforma para consultar los activos asignados a su nombre, así como su historial de préstamos, devoluciones y mantenimientos realizados. Esta vista ofrece transparencia y control al usuario sobre su responsabilidad institucional. | | |
| **Actores** | * Usuario Final (funcionario de la Universidad) | | |
| **Pre**  **condiciones** | * El usuario debe iniciar sesión mediante SSO institucional (Microsoft 365). * Debe haber al menos un activo asignado a su nombre registrado en el sistema. * El sistema debe tener trazabilidad registrada de cada movimiento del activo. | | |
| **Post condiciones** | * El usuario podrá visualizar todos sus activos, historial de movimientos y mantenimientos. * No podrá modificar la información, solo consultar. * Podrá descargar su historial en formato PDF (opcional si se implementa). | | |
| **Secuencia Normal** | # | Acción (actor) | Reacción (sistema) |
| 1 | El Usuario Final inicia sesión en la plataforma. | El sistema valida credenciales con SSO y permite el acceso al módulo de activos personales. |
| 2 | El usuario selecciona la opción “Mis activos”. | El sistema consulta y muestra la lista de  activos asignados al usuario. |
| 3 | El usuario selecciona un activo para  consultar detalles. | El sistema despliega su historial completo  de movimientos, mantenimientos, tickets  relacionados y estado actual. |
| **Excepciones** | # | Acción (actor) | Reacción (sistema) |
|  | p | Si el usuario intenta acceder sin estar autenticado. | El sistema redirige al portal de inicio de  sesión institucional. |
|  | Q | Si ocurre error de conexión con la base   de datos. | El sistema muestra mensaje: “Error al cargar activos, intente más tarde”. |
|  | r | Si el usuario intenta modificar algún  dato del activo. | El sistema bloquea la acción y muestra mensaje de acceso solo lectura. |
| **Rendimiento** | El sistema debe cargar el historial completo de un usuario en menos de 4 segundos, incluso si tiene más de 10 activos. | | |
| **Frecuencia** | Variable. Se espera que el usuario lo consulte al menos una vez por mes, o ante una auditoría interna o devolución. | | |
| **Importancia** | Importante (refuerza transparencia y responsabilidad sobre activos tecnológicos). | | |
| **Urgencia** | Puede esperar (acceso informativo, no operacional). | | |
| **Comentarios** | Este caso de uso promueve la autogestión responsable del usuario institucional sobre los bienes asignados, y se alinea con las políticas de trazabilidad y control operativo establecidas por la Universidad. | | |

* 1. **Anexo 21. BPMN del proceso con la aplicación**

Diagrama

El contenido generado por IA puede ser incorrecto.