

Actividad 1 - Identificar el proyecto tecnológico a trabajar  
“Plataforma de gestión de laboratorios universitarios”

Julián David Triana Celis

Tatiana Cabrera



Ingeniería De Software, Facultad De Ingeniería

Universidad Iberoamericana

Bogotá D.C

2025

## Tabla de contenido

|      |   |    |
|------|---|----|
| 1.   | Introducción.....   | 3  |
| 2.   | Justificación.....  | 4  |
| 3.   | Desarrollo del contenido del documento.....                         | 5  |
| 3.1. | Contextualización de la necesidad.....                              | 5  |
| 3.2. | Planteamiento del problema .....                                    | 5  |
| 3.3. | Alcance del proyecto .....  | 5  |
| 3.4. | Objetivos.....  | 6  |
| 4.   | Metodología ágil seleccionada .....                                 | 6  |
| 5.   | Mapa de Stakeholders y clasificación .....                          | 8  |
| 6.   | Matriz de riesgos .....   | 8  |
| 7.   | Repositorio y ramas.....  | 9  |
| 8.   | Levantamiento de información.....                                   | 9  |
| 9.   | Diagrama de flujo .....   | 10 |
| 10.  | Historias de usuario .....  | 11 |
| 11.  | RQF-RQNF .....  | 14 |
| 12.  | Evidencias Desing Thinking.....                                     | 15 |
| 13.  | Competencias clave / plus competitivo .....                         | 15 |
| 14.  | Estándares / Relación con currículo / Criterios de evaluación ..... | 15 |
| 15.  | Método de evaluación del producto.....                              | 15 |
| 16.  | Desafío y producto final .....                                      | 16 |
| 17.  | Tareas (WBS) .....  | 16 |
| 18.  | Difusión .....  | 16 |
| 19.  | Recursos .....  | 16 |
| 20.  | Herramientas tecnológicas.....                                      | 16 |
| 21.  | Cronograma .....  | 17 |
| 22.  | Referencias .....   | 17 |

## **1. Introducción**

El presente documento desarrolla el proyecto tecnológico denominado Plataforma de Gestión de Laboratorios Universitarios, cuyo propósito es optimizar la reserva de espacios, el control de equipos y la gestión de mantenimiento de los laboratorios de la Facultad de Ingeniería.

La propuesta surge a partir de la necesidad de mejorar la eficiencia y trazabilidad de estos procesos, actualmente realizados de forma manual.

El proyecto se implementará mediante metodologías ágiles (Scrum/Scrumban) y principios de Design Thinking, garantizando un enfoque centrado en los usuarios y entregas iterativas de valor.

## **2. Justificación**

La gestión de laboratorios en las instituciones educativas suele presentar limitaciones derivadas del uso de herramientas no integradas.

Actualmente, la reserva de espacios y equipos depende de hojas de cálculo o correos, lo que genera errores, duplicidad y conflictos de horario.

Una plataforma tecnológica unificada permitirá digitalizar el proceso, aumentando la eficiencia y reduciendo el riesgo de fallas.

- Corto plazo: reducir tiempos de programación y evitar choques de reservas.
- Medio plazo: optimizar la utilización de espacios y aumentar la satisfacción de usuarios.
- Largo plazo: integrar la solución con plataformas académicas institucionales e incluir módulos de analítica de uso.
- Stakeholders principales: coordinación académica, técnicos de laboratorio, docentes y estudiantes.
- Innovación: validaciones automáticas, trazabilidad de mantenimientos y generación de indicadores de uso.
- Relevancia: aporta transparencia, eficiencia y control de recursos tecnológicos y físicos.

### **3. Desarrollo del contenido del documento**

#### **3.1. Contextualización de la necesidad**

En la Facultad de Ingeniería, los procesos de reserva y mantenimiento de laboratorios son esenciales para la docencia y la investigación.

Sin embargo, la falta de un sistema centralizado ha generado ineficiencia, conflictos y pérdida de información.

Una plataforma digital permitirá planificar y supervisar el uso de espacios, controlando equipos y su mantenimiento.

#### **3.2. Planteamiento del problema**

El proceso manual de gestión de laboratorios provoca retrasos, duplicidad de información y errores en la programación.

- Problema: falta de integración tecnológica que gestione reservas, equipos y mantenimientos.
- Causas: uso de herramientas dispersas, comunicación ineficiente y ausencia de trazabilidad.
- Consecuencias: pérdida de tiempo, fallas en equipos y baja disponibilidad de laboratorios.

#### **3.3. Alcance del proyecto**

Incluye:

- Gestión de reservas de laboratorios.
- Control de inventario de equipos.
- Registro y seguimiento de mantenimiento preventivo y correctivo.
- Validaciones automáticas de disponibilidad.
- Generación de reportes y métricas.

Excluye: integración financiera y nómina de personal.

Restricciones: tres sprints de desarrollo, máximo cinco integrantes, uso de infraestructura cloud gratuita.

Criterios de aceptación:

- No permite reservas duplicadas.
- Genera historial de mantenimiento.
- Muestra disponibilidad de espacios en tiempo real.
- Exporta reportes a PDF o CSV.

### **3.4. Objetivos**

Objetivo general: desarrollar una plataforma que automatice la gestión de laboratorios, optimizando el uso de espacios, equipos y mantenimiento.

Objetivos específicos:

- Implementar un calendario centralizado de reservas.
- Registrar y monitorear el estado de los equipos.
- Automatizar alertas de mantenimiento.
- Generar reportes e indicadores de utilización.
- Validar conflictos y restricciones de uso.

### **4. Metodología ágil seleccionada**

Para la ejecución del proyecto “Plataforma de Gestión de Laboratorios Universitarios” se aplicará el marco de trabajo Scrum con soporte visual Kanban (Scrumban), ya que combina la planificación iterativa de Scrum con la flexibilidad de Kanban, permitiendo visualizar el flujo de trabajo y priorizar tareas de forma dinámica.

Roles definidos:

- Product Owner: Julián David Triana Celis.
- Scrum Máster: Tatiana Cabrera (docente guía).
- Equipo de desarrollo: integrantes del grupo de proyecto.

Duración y estructura:

Cada Sprint tendrá una duración de dos semanas, al final de las cuales se realizará una entrega funcional del producto.

Los principales eventos establecidos son: planificación del sprint, reuniones diarias (dailys), revisión del sprint y retrospectiva, garantizando comunicación constante y mejora continua.

Artefactos:

Se emplean el Backlog del producto, el Backlog del sprint y el Incremento funcional, los cuales permiten gestionar los requerimientos y medir el progreso del desarrollo.

El proyecto se organiza mediante un tablero ágil en Trello, estructurado con las columnas: Backlog del producto, Sprint actual, En desarrollo, En revisión/pruebas y Hecho. Cada tarjeta representa una historia de usuario (HU) asociada a una funcionalidad del sistema.

En la primera iteración (Sprint 1) se priorizaron las historias:

- HU01 – Reservar laboratorio
- HU02 – Validar disponibilidad

Las demás historias (HU03, HU04 y HU05) se mantienen en el Backlog del producto para futuros sprints.

El tablero Trello permite una visualización clara del flujo de trabajo, el seguimiento del avance y la asignación de tareas al equipo. Además, las etiquetas de colores clasifican las tarjetas según su prioridad, tipo de funcionalidad y sprint.

Enlace del tablero:

<https://trello.com/invite/b/68e30cd66050e0a9ea7b6521/ATTI9001aa27790d6059d0c5667667f4b8606C720153/plataforma-de-gestion-de-laboratorios-universitarios-tablero-agil>

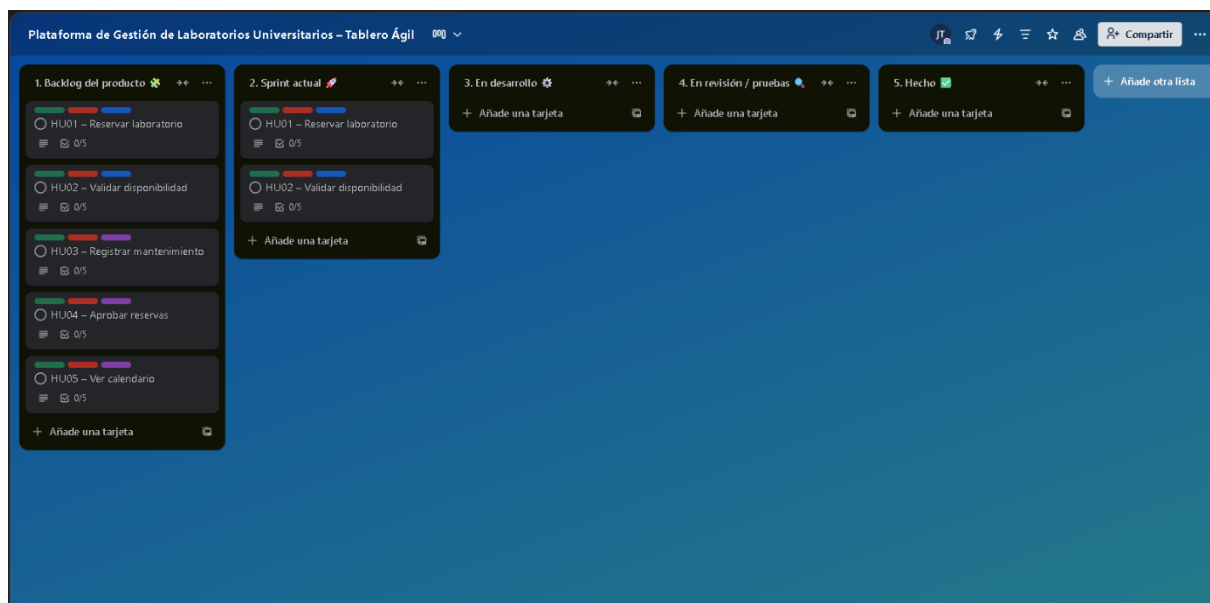


Ilustración 1 Tablero ágil del proyecto en Trello bajo la metodología Scrum-Kanban

## 5. Mapa de Stakeholders y clasificación

Primarios: coordinación académica, técnicos de laboratorio, docentes.

Secundarios: estudiantes, área TI, mantenimiento y bienestar.

Clasificación según poder–interés:

- Alta/Alta: coordinación (gestionar de cerca).
- Alta/Baja: TI (mantener satisfechos).
- Baja/Alta: docentes y técnicos (mantener informados).
- Baja/Baja: personal administrativo (monitorear).

## 6. Matriz de riesgos

| ID | Riesgo                          | Prob. | Impacto | Nivel | Estrategia                             |
|----|---------------------------------|-------|---------|-------|--|
| R1 | Retrasos por alcance indefinido | Media | Alta    | Alto  | Controlar alcance y priorizar backlog. |
| R2 | Fallas en despliegue            | Media | Media   | Medio | Entorno de pruebas y CI/CD.            |
| R3 | Baja adopción del sistema       | Media | Alta    | Alto  | Capacitación y pruebas piloto.         |
| R4 | Datos inconsistentes            | Media | Media   | Medio | Validaciones automáticas.              |
| R5 | Sobrecarga de equipo            | Media | Media   | Medio | Planificación realista.                |
| R6 | Fallas de seguridad             | Baja  | Alta    | Medio | Autenticación y HTTPS.                 |



## **7. Repositorio y ramas**

Repositorio GitHub: <https://github.com/JulianTriana11/Plataforma-Gestion-Laboratorios>

Convención:

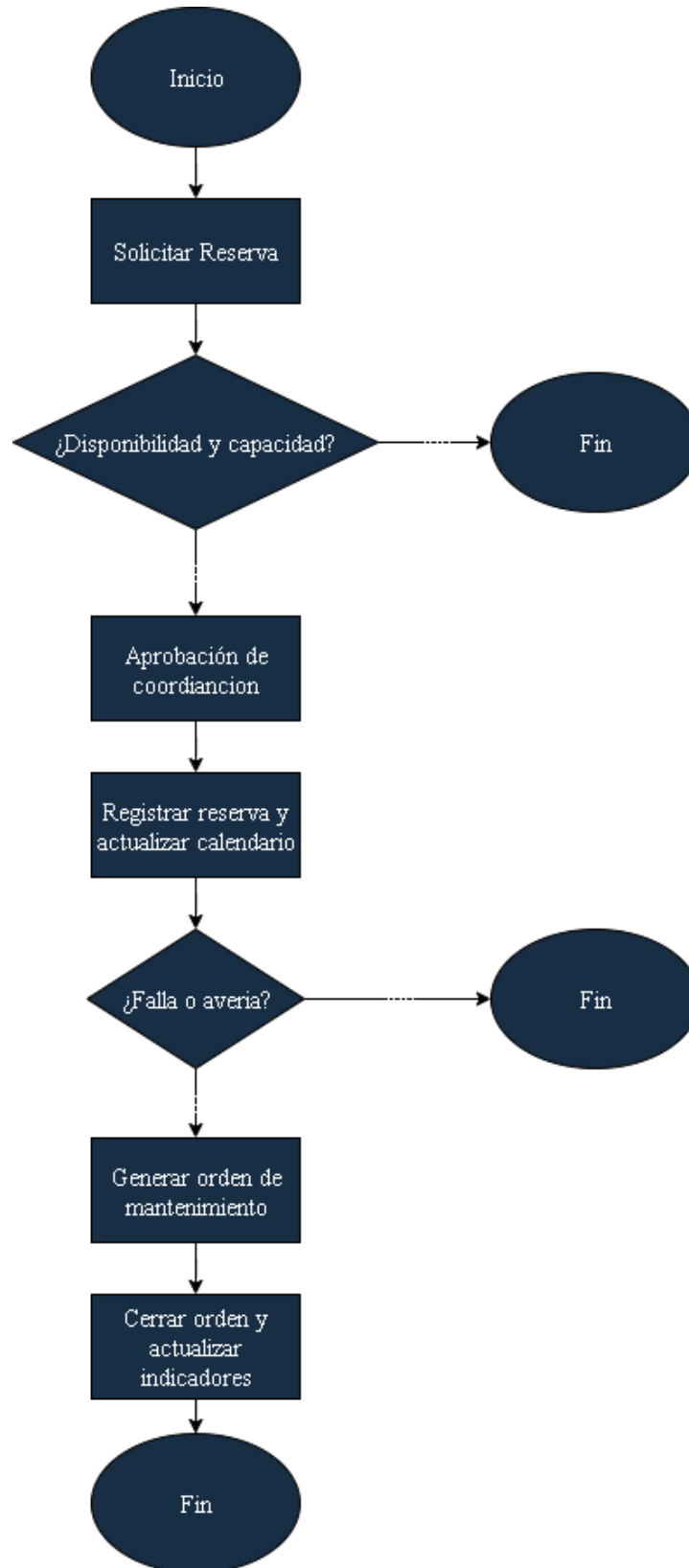
- main → versión estable
- dev → integración
- feat/huXX-descripcion → por historia de usuario

## **8. Levantamiento de información**

- Herramientas: entrevistas a docentes y técnicos, encuestas a estudiantes, observación directa y análisis documental.
- Conclusiones: los usuarios requieren visibilidad de horarios, validaciones automáticas de disponibilidad y trazabilidad en el mantenimiento de los equipos.

## 9. Diagrama de flujo

Diagrama de flujo del proceso de gestión de laboratorios universitarios.



*Ilustración 2 Diagrama de flujo del proceso de gestión de laboratorios universitarios.*

## 10. Historias de usuario

Las historias de usuario representan las funcionalidades principales que debe cumplir el sistema “Plataforma de Gestión de Laboratorios Universitarios”. Cada historia se documentó en el tablero ágil de Trello, siguiendo la estructura de la metodología Scrum y utilizando el formato estándar:

- Como [tipo de usuario], quiero [acción o necesidad] para [beneficio o resultado esperado].

Cada historia de usuario incluye criterios de aceptación específicos y un checklist de tareas técnicas que permiten medir el avance de desarrollo, garantizando que cada funcionalidad cumpla con los requerimientos definidos.

Las historias de usuario registradas en el tablero son las siguientes:

- HU01 – Reservar laboratorio

Descripción:

Como docente, quiero reservar un laboratorio para programar mis clases sin choques de horario.

Criterios de aceptación:

- El sistema valida disponibilidad.
- Se guarda la reserva en el calendario.
- Envía confirmación al usuario.

Checklist:

- Diseñar formulario de reserva.
- Implementar validación de disponibilidad y capacidad.
- Conectar formulario con base de datos.
- Generar confirmación automática al usuario.
- Probar reserva y verificar conflictos.

- **HU02 – Validar disponibilidad**

Descripción:

Como sistema, quiero validar la disponibilidad y capacidad del laboratorio seleccionado para evitar conflictos de reservas.

Criterios de aceptación:

- El sistema detecta choques de horario o exceso de capacidad.
- Muestra mensaje de advertencia si la reserva no es posible.
- Permite continuar solo si hay disponibilidad.

Checklist:

- Crear algoritmo de validación de horarios.
- Configurar parámetros de capacidad por laboratorio.
- Implementar mensajes de error dinámicos.
- Probar validaciones con diferentes escenarios.
- Registrar resultados de prueba en el tablero QA.

- **HU03 – Registrar mantenimiento**

Descripción:

Como técnico de laboratorio, quiero registrar las fallas o mantenimientos de equipos para llevar un control actualizado y evitar interrupciones en clases.

Criterios de aceptación:

- Permite crear una orden de mantenimiento con detalles del equipo y tipo de falla.
- Actualiza el estado del equipo automáticamente.
- Genera historial de mantenimientos realizados.

Checklist:

- Diseñar formulario de mantenimiento.
- Configurar base de datos para registro de fallas.
- Crear módulo de historial de mantenimientos.
- Implementar actualización de estado del equipo.
- Probar creación y cierre de órdenes.

- **HU04 – Aprobar reservas**

Descripción:

Como coordinación académica, quiero aprobar o rechazar las reservas de laboratorios para garantizar la disponibilidad y el uso adecuado de los espacios.

Criterios de aceptación:

- El sistema muestra las reservas pendientes de aprobación.
- Permite aprobar o rechazar con comentario.
- Guarda el registro del usuario y fecha de decisión.

Checklist:

- Crear vista de reservas pendientes.
- Implementar botones de aprobación y rechazo.
- Registrar auditoría de usuario y fecha.
- Enviar notificación automática al docente.
- Validar que la reserva aprobada se actualice en el calendario.

- **HU05 – Ver calendario de laboratorios**

Descripción:

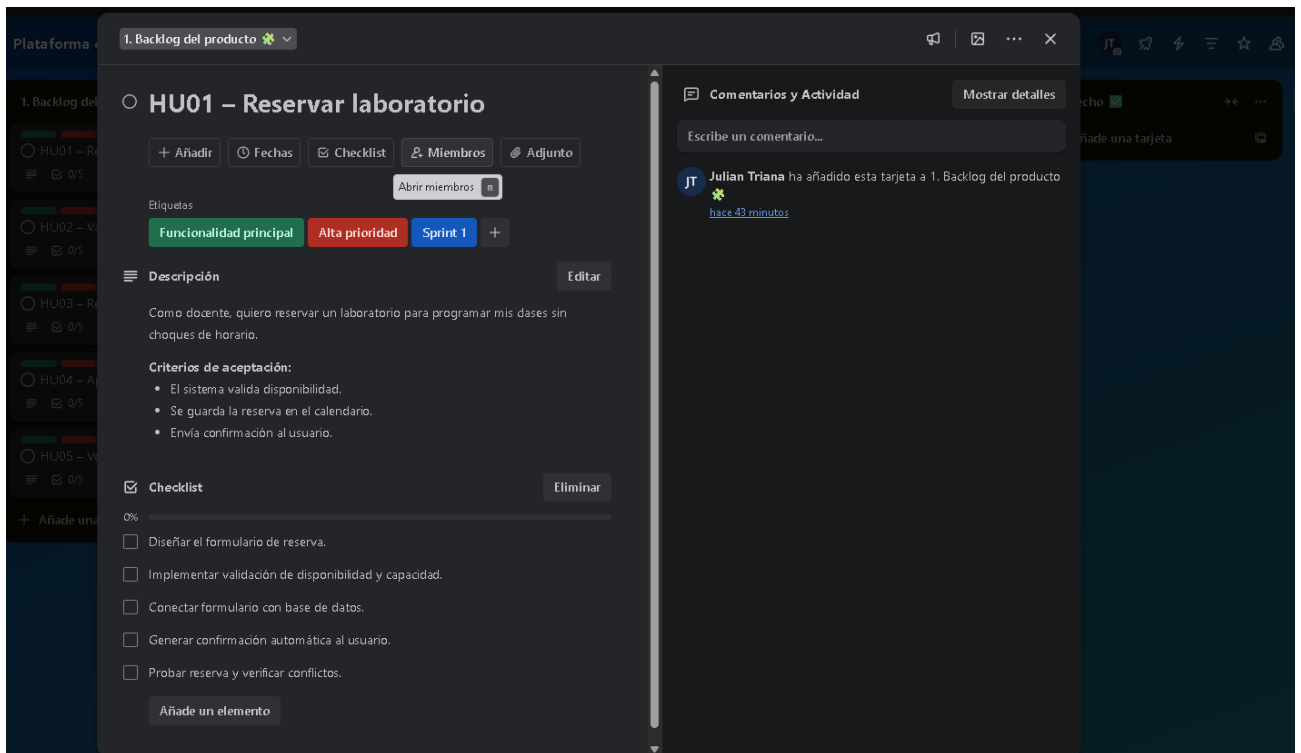
Como usuario, quiero visualizar el calendario general de los laboratorios para conocer horarios disponibles, reservas activas y mantenimientos programados.

Criterios de aceptación:

- Muestra calendario filtrable por laboratorio y fecha.
- Permite distinguir reservas, mantenimientos y disponibilidad.
- Exporta a PDF o CSV.

Checklist:

- Integrar biblioteca de calendario (FullCalendar o similar).
- Conectar API de reservas y mantenimientos.
- Implementar filtros por fecha y laboratorio.
- Diseñar leyenda de colores para estados.
- Agregar función de exportación a PDF/CSV.



## 11. RQF-RQNF

### Funcionales (RQF):

- Registro y edición de reservas.
- Validación automática de conflictos.
- Gestión de mantenimiento y fallas.
- Generación de reportes.

### No funcionales (RQNF):

- Respuesta < 2 segundos.
- Disponibilidad del 99 %.
- Usabilidad accesible en móvil.
- Seguridad mediante roles y HTTPS.
- Mantenibilidad modular.

## **12. Evidencias Design Thinking**

Para el desarrollo del proyecto “Plataforma de Gestión de Laboratorios Universitarios” se aplicó la metodología Design Thinking, la cual permitió comprender las necesidades de los usuarios finales, definir el problema central y generar ideas innovadoras orientadas a una solución tecnológica funcional.

- **Fase de empatizar:**  
Se realizaron entrevistas y observaciones directas con docentes, estudiantes y técnicos de laboratorio para identificar las principales dificultades en la gestión de reservas, disponibilidad de equipos y comunicación de mantenimientos. Esta fase permitió conocer la experiencia real del usuario y recopilar información valiosa sobre sus expectativas y frustraciones.
- **Fase de definir:**  
Con base en la información recopilada, se identificaron los puntos críticos del proceso actual: ausencia de trazabilidad, duplicidad de reservas y falta de confirmaciones automáticas. A partir de ello, se formuló el problema principal: “¿Cómo mejorar la eficiencia y el control en la gestión de laboratorios universitarios mediante una plataforma digital centralizada?”
- **Fase de idear:**  
El equipo propuso distintas alternativas de solución, evaluando su viabilidad técnica y su impacto en los usuarios. Se seleccionó como propuesta principal el desarrollo de una plataforma web que gestione reservas, valide disponibilidad y registre mantenimientos en tiempo real.

La aplicación del enfoque Design Thinking facilitó el diseño centrado en el usuario, promoviendo la colaboración y la creatividad en la búsqueda de una solución tecnológica ajustada a las necesidades reales de la comunidad académica.

## **13. Competencias clave / plus competitivo**

El sistema destaca por su enfoque especializado en laboratorios de ingeniería, validaciones automáticas y flujo de mantenimiento integrado.

## **14. Estándares / Relación con currículo / Criterios de evaluación**

Alineado con PMBOK 7 (stakeholders, equipo, planificación, entrega, medición) y con los resultados de aprendizaje del programa.

## **15. Método de evaluación del producto**

- **Funcionalidad:** cumplimiento de criterios de aceptación.

- Calidad: defectos por sprint.
- Satisfacción: encuestas SUS.
- Impacto: reducción de conflictos y tiempos.

## **16. Desafío y producto final**

Desafío: integrar reglas, trazabilidad y roles múltiples.

Producto final: MVP web funcional con datos de prueba.

## **17. Tareas (WBS)**

- a) Levantamiento de requerimientos.
- b) Diseño de arquitectura y UX.
- c) Implementación del MVP.
- d) Pruebas y validación.
- e) Despliegue y capacitación.

## **18. Difusión**

Presentación interna a coordinación, voz a voz y publicación institucional.

## **19. Recursos**

Humanos: PO, SM, desarrolladores, QA.

Tecnológicos: hosting, repositorio Git, Trello.

Financieros: herramientas gratuitas y recursos institucionales.

## **20. Herramientas tecnológicas**

- Gestión: Trello, Notion.
- Código: GitHub, Node.js, React.
- Base de datos: PostgreSQL.
- Pruebas: Postran, Jest.
- DevOps: Docker, GitHub Activos.



## 21. Cronograma

| Sprint | Actividades principales                    | Duración  | Entregable              |
|--------|--|-----------|-------------------------|
| 0      | Diseño inicial y configuración del entorno | 1 semana  | Mockups y setup técnico |
| 1      | Implementación módulo de reservas          | 2 semanas | HU01, HU02 completadas  |
| 2      | Desarrollo módulo de mantenimiento         | 2 semanas | HU03 lista              |
| 3      | Reportes y pruebas de validación           | 2 semanas | HU04, HU05 y testing    |
| 4      | Documentación y despliegue final           | 1 semana  | Versión MVP funcional   |

## 22. Referencias

- **PMI.** (2021). *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)* (7.ª ed.). Project Management Institute.
- **Schwaber, K. y Sutherland, J.** (2020). *La Guía de Scrum: La Guía Definitiva para Scrum, el marco de trabajo ágil más utilizado en el mundo*. Scrum.org & Scrum Alliance.
- **ISO/IEC.** (2011). *ISO/IEC 25010:2011 Ingeniería de sistemas y software — Requisitos y evaluación de la calidad del software (SQuaRE)*. Organización Internacional de Normalización.
- **IDEO.org.** (2015). *Guía práctica para el diseño centrado en el ser humano*. IDEO.org.