Metodología SCRUM

Integrantes: Matias Roman

Frank Vogt

Patricio Quintanilla  
Fechas: 13-10-2025  
Profesor: Marco Antonio

Contenido

[Introducción 3](#_Toc211266999)

[Metodología y justificación de la elección 3](#_Toc211267000)

[Fases y ejecución 3](#_Toc211267001)

[Roles, herramientas y artefactos 4](#_Toc211267002)

[Aseguramiento de calidad y mitigación de dependencias 4](#_Toc211267003)

[Gestión de riesgos, métricas y lecciones aprendidas 4](#_Toc211267004)

[Conclusión 4](#_Toc211267005)

# Introducción

Se utilizó la metodología Scrum para organizar el trabajo del equipo y cumplir con los plazos establecidos. El objetivo principal fue entregar avances funcionales del proyecto de manera rápida y eficiente, asegurando calidad en cada etapa y reduciendo problemas relacionados con dependencias externas. Para lograr esto, se priorizó la creación de un producto mínimo viable (MVP) que permitiera validar los requisitos y minimizar riesgos antes de avanzar con tareas más complejas.

# Metodología y justificación de la elección

Scrum fue elegido porque permite trabajar de forma flexible y adaptarse a cambios en los requisitos o problemas externos. Este enfoque es ideal para proyectos con incertidumbres, ya que facilita la replanificación sin detener el desarrollo. A diferencia de métodos más rígidos como Waterfall, Scrum ofrece reuniones periódicas para planificar, revisar y ajustar el trabajo, lo que asegura que cada entrega cumpla con los objetivos. Además, se definieron criterios claros para considerar cada tarea como completada, lo que ayudó a mantener el orden y la calidad del proyecto.

# Fases y ejecución

El proyecto se dividió en varias etapas para facilitar su desarrollo:

1. Identificación de requisitos: Se analizaron las necesidades del proyecto y se priorizaron las tareas más importantes.
2. Diseño de la base de datos: Se creó el modelo de datos y se prepararon datos iniciales para pruebas.
3. Desarrollo del backend y frontend: Se trabajó en funciones básicas como el manejo de usuarios, autenticación y operaciones CRUD.
4. Validación de integraciones: Se utilizaron herramientas simuladas (stubs y webhooks) para probar funciones sin depender de sistemas externos.
5. Pruebas y ajustes finales: Se realizó un sprint dedicado a pruebas y correcciones para asegurar que el proyecto cumpliera con los estándares de calidad.

Cada etapa incluyó objetivos claros y requisitos mínimos, como pruebas funcionales, código limpio y documentación para facilitar el trabajo en equipo.

# Roles, herramientas y artefactos

El equipo se organizó en roles específicos para garantizar un desarrollo eficiente:

Product Owner: Responsable de definir los objetivos y prioridades del proyecto.

Scrum Master: Encargado de coordinar el trabajo y resolver problemas.

Desarrolladores: Trabajaron en el backend, frontend y pruebas.

QA: Aseguró la calidad del proyecto mediante revisiones y pruebas.

Se utilizaron herramientas como GitHub para el control de versiones, Docker para crear entornos de trabajo reproducibles, ESLint para revisar el código y Jest para realizar pruebas. Además, se crearon documentos importantes como la base de datos inicial, guías para integrar sistemas externos y scripts para facilitar el despliegue.

# Aseguramiento de calidad y mitigación de dependencias

Para garantizar la calidad del proyecto, se definieron pasos básicos que cada entrega debía cumplir, como pruebas funcionales, revisión del código y documentación. También se programó un sprint dedicado a implementar un sistema de pruebas automatizadas y generar reportes de cobertura. Para evitar retrasos por dependencias externas, se crearon herramientas simuladas (stubs y webhooks) que permitieron probar funciones de manera local antes de integrarlas con sistemas reales.

# Gestión de riesgos, métricas y lecciones aprendidas

Se tomaron medidas para reducir riesgos, como dividir tareas complejas en partes más pequeñas, planificar pruebas y proteger los datos con medidas de seguridad. Se utilizaron métricas simples para medir el progreso, como el porcentaje de tareas completadas y la cantidad de errores detectados. Entre las lecciones aprendidas, se destacó la importancia de automatizar pruebas desde el inicio, planificar bien las integraciones con sistemas externos y documentar todo el proceso para facilitar el trabajo en equipo.

# Conclusión

El uso de Scrum permitió organizar el trabajo del equipo y entregar un MVP funcional con las funciones básicas del proyecto. Este enfoque ayudó a reducir riesgos, adaptarse a cambios y establecer una base sólida para completar el proyecto con éxito. Aunque aún quedan tareas por realizar, como pruebas más avanzadas y ajustes finales, el trabajo realizado hasta ahora demuestra que la metodología y las decisiones tomadas fueron efectivas.