

## **7. Plan de Trabajo - SWAPPLY**

### **7.1. Fases del Proyecto**

El proyecto Swapply se estructura en 6 fases principales que se distribuyen desde principios de noviembre de 2025 hasta el 1 de junio de 2026. Cada fase tiene objetivos claros, actividades definidas y entregables específicos.

### **ANÁLISIS (Principios de noviembre – mediados de diciembre)**

Durante esta fase se recopila y define toda la información necesaria para construir Swapply.

#### **Actividades principales:**

→ Noviembre (hasta el 30 de noviembre):

- Descripción del proyecto y contexto general.
- Especificación de roles y responsabilidades del equipo.
- Recopilación de requisitos funcionales: registro de usuarios, solicitud de intercambios, chat interno, notificaciones.
- Definición de requisitos no funcionales: seguridad (protección de datos personal), rendimiento, escalabilidad, fiabilidad y usabilidad.
- Gestión de riesgos: análisis de riesgos técnicos como fraudes, sobrecarga del sistema, fallos de pago si existieran servicios premium.

→ Diciembre (hasta el 14 de diciembre):

- Especificación detallada de casos de uso que describen cómo los usuarios interactuarán con la plataforma.
- Diciembre (hasta el 29 de diciembre):
  - Elaboración de diagramas de casos de uso usando notación UML.
  - Cierre del documento de requisitos y especificación funcional completa.

**Entregable:** Documento de requisitos, especificación de casos de uso y análisis de riesgos completados.

**Hito H1** (29 de diciembre): Documento de requisitos y casos de uso aprobado.

### **DISEÑO (Mediados de diciembre – finales de enero)**

Se define la arquitectura técnica y la estructura de Swapply, basándose en los requisitos validados.

### **Actividades principales:**

- Diciembre – Principios de enero (hasta el 6 de enero):
  - Elaboración de diagramas de actividad para los casos de uso más importantes.
  - Diseño de la arquitectura general: estructura cliente-servidor, uso de [APIs](#), [gestión de sesiones](#), autenticación y control de acceso.
- Diciembre – Enero (hasta el 26 de enero):
  - Diseño de la base de datos: definición de tablas para usuarios, productos, categorías, solicitudes de intercambio, valoraciones y mensajes.
  - Elaboración de diagramas UML: diagramas de clases, secuencia y componentes que modelan el sistema.
- Enero (hasta el 8 de febrero):
  - Diseño de la interfaz de usuario (UI): creación de maquetas para las pantallas principales (inicio, perfil, bandeja de intercambios, chat, notificaciones).
  - Definición del flujo de usuario ([UX](#)) para asegurar una navegación intuitiva y coherente.

**Entregables:** Arquitectura definida, modelo de datos completo, diagramas UML y prototipos de interfaz.

**Hito H2** (26 de enero): Arquitectura, modelo de datos y prototipos de interfaz validados.

## **DESARROLLO (Febrero – mediados de abril)**

Se implementa Swappy como producto real, dividiendo el trabajo entre módulos del servidor e interfaz de usuario.

### **Actividades principales:**

- Febrero (hasta el 2 de marzo):
  - Implementación de la lógica del lado del servidor (módulos del servidor):
    - Desarrollo de [APIs](#) para gestionar usuarios, productos e intercambios.
    - Implementación de autenticación y mecanismos de seguridad.
    - Lógica del sistema de intercambio (validación de solicitudes, estados de intercambio, confirmación).

**Nota:** Los "módulos del servidor" son componentes que administran recursos, almacenan archivos y datos, y procesan las solicitudes de los clientes a través de una red, permitiendo que la plataforma funcione como servicio backend.

- Febrero – Marzo (hasta el 9 de marzo):
  - Implementación de la interfaz de usuario ([frontend](#)):
    - Desarrollo de las vistas basadas en los prototipos de diseño.
    - Integración con las APIs del servidor para mostrar información en tiempo real.
    - Implementación de aspectos visuales: estilos, navegación, componentes interactivos.
- Marzo (hasta el 30 de marzo):
  - Integración de funcionalidades clave:
    - Sistema de chat entre usuarios (mensajería en tiempo real).
    - Sistema de valoraciones y reputación.
    - Notificaciones de intercambios y eventos.
    - Integración continua mediante GitHub: automatización de despliegues en entornos de prueba.
- Mediados de abril (hasta el 17 de abril):
  - Completar funcionalidades pendientes y ajustes finales.
  - Primera integración completa de [backend](#) y [frontend](#).
  - Verificación interna de que todos los requisitos críticos se han implementado.

**Entregables:** Primera versión funcional de Swapply con [backend](#) y [frontend](#) integrados.

**Hito H3** (17 de abril): Primera versión funcional de Swapply completada.

## PRUEBAS (Mediados de abril – principios de mayo)

Se ejecutan pruebas exhaustivas para garantizar la calidad, estabilidad y usabilidad de la plataforma.

### Actividades principales:

- Enero – Enero (hasta el 31 de enero):
  - Pruebas de integración continua tempranas: validación de que los primeros prototipos de subsistemas se comunican correctamente, comprobando interfaces y flujos básicos entre módulos de usuarios, productos y base de datos.
- Marzo (hasta el 16 de marzo):

- Segundo ciclo de pruebas de integración: validación de que la lógica del servidor e interfaz de usuario funcionan correctamente en conjunto.

→ Mediados de abril (hasta el 30 de abril):

- Pruebas unitarias 1 y 2: validación de funciones, clases y componentes de forma aislada en [backend](#) (APIs, lógica de negocio) y [frontend](#).

**Nota:** Las pruebas unitarias dependen de que exista código implementado, es decir, de la implementación de la lógica del servidor y de la interfaz, porque se prueban funciones, clases o componentes concretos ya desarrollados.

→ Abril (hasta el 12 de abril):

- Pruebas de usabilidad 1: evaluación de la facilidad de uso de Swapply con usuarios de prueba (compañeros del equipo o usuarios externos), validando que los flujos principales (registro, solicitar intercambio, chat) son intuitivos.

→ Abril (hasta el 25 de abril):

- Subsanación de errores tras pruebas: corrección de los errores detectados en las pruebas unitarias y de usabilidad más importantes, priorizando los que afectan a los flujos principales (registro, login, intercambio, chat, etc.).

→ Mayo (hasta el 2 de mayo):

- Pruebas de usabilidad 2: validación adicional de flujos secundarios y refinamientos tras correcciones.

→ Mayo (hasta el 11 de mayo):

- Pruebas de rendimiento para verificar la estabilidad del servidor.
- Corrección final de errores detectados.

**Entregables:** Validación del sistema, garantía de funcionamiento estable, documentación de pruebas.

**Hito H4** (11 de mayo): Sistema validado y estable, listo para implementación.

## IMPLEMENTACIÓN (Mediados – finales de mayo)

Se despliega Swapply para su uso real en producción.

**Actividades principales:**

→ Mediados de mayo (hasta el 23 de mayo):

- Configuración final del servidor.
- Instalación y configuración de la base de datos en producción.
- Configuración de dominio y certificado [SSL](#) para comunicación segura.

→ Finales de mayo (hasta el 1 de junio):

- Despliegue de la aplicación web.
- Publicación de la versión v.1.0 para acceso público.
- Monitorización inicial del funcionamiento en producción.

**Entregables:** Swapply disponible públicamente para los usuarios.

**Hito H5 (1 de junio):** Publicación de Swapply v1.0.

## **MANTENIMIENTO (Desde publicación hasta 1 de junio y más allá)**

Se realiza monitorización y mejora continua de la plataforma después del lanzamiento.

**Actividades principales:**

- Corrección de errores que aparecen con el uso real.
- Actualización de funcionalidades existentes basada en feedback de usuarios.
- Implementación de mejoras y nuevas características.
- Mejora del rendimiento del servidor según métricas observadas.
- Monitorización continua de seguridad y protección de datos personales.

**Entregables:** Plataforma funcionando de manera estable y mejoras implementadas con el tiempo.

## **7.2. Cronograma Gantt**

El cronograma Gantt refleja la distribución temporal de todas las fases y tareas del proyecto desde el inicio de noviembre de 2025 hasta el 1 de junio de 2026, un período de aproximadamente 6 meses.

**Principales características del Gantt:**

- Las barras de color representan el rango temporal de cada actividad.
- Las dependencias se indican mediante flechas o líneas de conexión entre tareas.
- Los hitos (milestones) se marcan como puntos clave de decisión o entrega.
- Las solapamientos entre fases indican trabajo paralelo (ej.: diseño y primeras pruebas de integración conceptual, desarrollo e integración continua).

Desglose temporal aproximado:

Fase	Período	Duración
Análisis	Nov 30 – Dic 29	~4 semanas
Diseño	Dic 6 – Ene 26	~8 semanas
Desarrollo	Feb 1 – Abr 17	~11 semanas
Pruebas	Ene 31 – May 11	~14 semanas (solapadas)
Implementación	May 11 – Jun 1	~3 semanas
Mantenimiento	Jun 1 en adelante	Continuo

## 7.3. Dependencias e Hitos Clave

### DEPENDENCIAS DEL PROYECTO

Las siguientes dependencias aseguran una secuencia lógica y ordenada del trabajo:

**1. Diseño depende de Análisis:**

El diseño de arquitectura, base de datos e interfaz requiere que los requisitos y casos de uso estén completamente definidos y aprobados (Hito H1).

**2. Desarrollo depende de Diseño:**

La implementación de los módulos del servidor y la interfaz depende de que la arquitectura, y los prototipos de interfaz estén validados (Hito H2). Los desarrolladores usan estos documentos como guía.

**3. Pruebas de integración tempranas dependen de arquitectura:**

Las primeras pruebas de integración (enero) validan que los subsistemas diseñados pueden comunicarse, lo que depende de tener la arquitectura definida.

**4. Pruebas unitarias dependen de desarrollo:**

Las pruebas unitarias requieren código implementado en los módulos del servidor e interfaz. No pueden ejecutarse sin que exista lógica desarrollada y componentes creados.

**5. Pruebas de integración de segundo ciclo dependen de pruebas unitarias:**

Una vez validados los módulos de forma individual, se prueban en conjunto para asegurar que interactúan correctamente.

**6. Pruebas de usabilidad dependen de interfaz implementada:**

Las pruebas de usabilidad requieren que la interfaz esté completa y funcional (Hito H3) para ser evaluada por los usuarios.

**7. Implementación depende de pruebas aprobadas:**

Solo se despliega a producción cuando se han superado las pruebas y el sistema se ha validado como estable (Hito H4).

**8. Mantenimiento depende de publicación:**

El mantenimiento comienza después de que la v1.0 está disponible públicamente (Hito H5).

## HITOS PRINCIPALES ("MILESTONES")

Se definen 5 hitos principales que representan puntos de decisión críticos y entregas importantes:

	Fecha	Descripción
H1	29 Dic 2025	Documento de requisitos y casos de uso aprobado
H2	26 Ene 2026	Arquitectura, modelo de datos y prototipos de interfaz validados
H3	17 Abr 2026	Primera versión funcional de Swapply completada
H4	11 May 2026	Sistema validado y estable, pruebas de aceptación superadas
H5	1 Jun 2026	Publicación de Swapply v1.0

**API:** conecta información y funcionalidades con los requerimientos de la aplicación

**Gestión de sesiones:** proceso de administrar la interacción continua de un usuario con una aplicación para mantener su estado y datos seguros y eficientes.

**UX:** Camino que sigue un usuario a través de una plataforma digital para completar una tarea específica, como registrarse o comprar un producto.

**Backend:** lógica de negocio de Swapply (gestión de usuarios, seguridad, anuncios, intercambios, reputación, notificaciones) y el acceso a la base de datos.

**Frontend:** Interfaz web que ven los usuarios (pantallas de inicio, crear anuncio, perfil, bandeja de intercambios).

**SSL:** protocolo de seguridad que cifra la conexión entre un servidor web y un navegador para proteger la privacidad y la integridad de los datos transmitidos, como información personal o financiera

