

## 7. Plan de Trabajo - SWAPPLY

### 7.1. FASES DEL PROYECTO

#### ANÁLISIS:

Recopilación y definición de requisitos necesarios para construir Swapply.

- Identificación de actores principales: usuarios registrados, visitantes y administradores.
- Recopilación de **requisitos funcionales**: registro de usuarios, creación de anuncios, solicitud de intercambios, chat interno, sistema de reputación, notificaciones, etc.
- Definición de **requisitos no funcionales**: seguridad (protección de datos), rendimiento, escalabilidad, disponibilidad, usabilidad.
- Elaboración de **diagramas de casos de uso** que describen cómo los usuarios interactuarán con la plataforma.
- Análisis de **riesgos técnicos** (fraudes, sobrecarga, fallos de pago si existieran servicios premium).

★ Documento de requisitos, casos de uso y una especificación funcional completa para comenzar el diseño.

#### DISEÑO:

Definición de arquitectura y estructura técnica de Swapply.

- Diseño de la **arquitectura general**: estructura cliente-servidor, uso de [APIs](#), [gestión de sesiones](#), autenticación.
- Diseño de la **base de datos**: tablas para usuarios, productos, categorías, solicitudes de intercambio, valoraciones.
- Elaboración de **diagramas UML**: diagramas de clases, secuencia, y componentes.
- Diseño de **interfaz de usuario (UI)**: creación de maquetas para las pantallas principales (inicio, crear anuncio, perfil, bandeja de intercambios).
- Definición del **flujo de usuario (UX)** para asegurar una navegación intuitiva.

★ Arquitectura definida, base de datos modelada y prototipos de la interfaz.

## DESARROLLO:

Implementación y construcción de Swapply como producto real

- Implementación del **backend**:
  - API para usuarios, productos e intercambios.
  - Gestión de autenticación y seguridad.
  - Lógica para el sistema de intercambio.
- Implementación del **frontend**:
  - Desarrollo de las vistas basadas en los prototipos de diseño.
  - Integración con la API para mostrar la información en tiempo real.
- Desarrollo de **funcionalidades clave**:
  - Publicación de productos y servicios.
  - Sistema de chat entre usuarios.
  - Sistema de valoraciones.
  - Notificaciones de intercambios.
- **Integración continua** mediante herramientas como GitHub.

★ Primera versión funcional del sistema

## PRUEBAS:

Realización de pruebas exhaustivas para asegurar la calidad antes del lanzamiento.

- Pruebas **unitarias** de cada módulo del backend y frontend
- Pruebas de **integración** entre los distintos componentes del sistema
- Pruebas de **aceptación del usuario**: si la plataforma cumple con los requisitos establecidos
- Pruebas de **usabilidad**: si Swapply es fácil de usar
- Prueba de carga y **rendimiento** para verificar la estabilidad del servidor
- Corrección de **errores** detectados

★ Validación del sistema y garantía de funcionamiento estable

## IMPLEMENTACIÓN:

Despliegue de Swapply para su uso real.

- Configuración del servidor o entorno en la nube (AWS, Google Cloud, etc).
- Instalación de la base de datos en producción
- Configuración de dominio y certificado SSL
- Despliegue de la aplicación web o móvil
- Publicación de la primera versión (v.1.0)

★ Swapply está disponible públicamente para los usuarios

## MANTENIMIENTO:

Monitorización y mejora continua de la plataforma después del lanzamiento.

- Corrección de errores que aparecen con el uso real
- Actualización de funcionalidades existentes
- Implementación de nuevas características (ej.: sistema de verificación de identidad)
- Mejora del rendimiento del servidor
- Monitorización de seguridad y protección de datos

★ Plataforma funcionando de manera estable y mejoras con el tiempo.

## 7.2. CRONOGRAMA: GANTT

<https://www.canva.com/design/DAG5u261lrE/nZ90VO27NJ06pB21tCn3Pg/edit>

1. **ANÁLISIS** (principios de noviembre – mediados de diciembre)
  - Noviembre: requerimientos, análisis de riesgos.
  - Principios de diciembre: casos de uso y documento de requisitos cerrado ([Hito H1](#)).
2. **DISEÑO** (mediados de diciembre – finales de enero)
  - Mediados de diciembre – principios de enero: arquitectura, APIs, [gestión de sesiones](#) / autenticación.
  - Diciembre – enero: diseño de base de datos y diagramas UML.
  - Enero: prototipos de interfaz.
  - Finales de enero: arquitectura + BD + prototipos aprobados ([Hito H2](#)).

### 3. DESARROLLO (febrero – mediados de abril)

- Febrero: backend (APIs, lógica de intercambio, seguridad).

“Módulos del **servidor**”: programa que administra recursos, almacena archivos y datos, y procesa las solicitudes de clientes a través de una red. Páginas web, aplicaciones, para que los usuarios puedan acceder a la información y usar los servicios de la plataforma.

- Febrero – marzo: frontend (aspectos visuales, integración con API).
- Marzo – mediados de abril: funcionalidades clave (chat, valoraciones, notificaciones) e integración continua.
- Mitad de abril: primera versión funcional lista ([Hito H3](#)).

### 4. PRUEBAS (mediados de abril – principios de mayo)

- Mediados de abril: pruebas unitarias y de integración.
- Finales de abril: pruebas de usabilidad y aceptación.
- Principios de mayo: pruebas de carga/rendimiento y corrección de errores.
- Primera semana de mayo: sistema validado y estable ([Hito H4](#)).

### 5. IMPLEMENTACIÓN (mediados – finales de mayo)

- Mediados de mayo: configuración servidor/nube, base de datos en producción, dominio y [SSL](#).
- Finales de mayo: despliegue de la versión v1.0 para uso real.
- Última semana de mayo / 1 de junio: Swapply publicado ([Hito H5](#)).

### 6. MANTENIMIENTO (desde publicación hasta junio)

- Desde publicación hasta 1 de junio: correcciones menores, mejoras de rendimiento, pequeños añadidos (p.ej. verificación de identidad), monitorización de seguridad.

## 7.3. DEPENDENCIAS E HITOS CLAVE

- **DEPENDENCIAS:**

- El **diseño** depende de tener el Análisis completado (requisitos cerrados).
- El **desarrollo** depende de la arquitectura y la base de datos definidas.
- Las **pruebas** dependen de tener una primera versión funcional desarrollada.
- La **implementación** depende de superar las pruebas críticas.

- **HITOS (“MILESTONES” IMPORTANTES):**

- H1: Documento de requisitos y casos de uso aprobado (fin de Análisis).
- H2: Arquitectura y modelo de datos validados + prototipo de interfaz (fin de Diseño).
- H3: Primera versión funcional de Swapply (backend + frontend integrados).
- H4: Pruebas de aceptación superadas y sistema estable.
- H5: Publicación de Swapply v1.0 antes del 1 de junio.

**API:** mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos

**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