



**Nombre: Jonathan Ramallo Rubio**

**Grupo: 2DAW**

**Curso: 2023-2024**

**Proyecto: MangaSculp**

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

## Contenidos

1. Introducción .....	4
1.1- Presentación y Objetivos.....	4
1.2- Contexto .....	4
1.3- Planteamiento del Problema .....	5
1.4- Analisis de costes. ....	6
Ejemplo de desglose de costes (mensual): .....	7
1.5- Plan de Financiación.....	8
Estrategia de financiación .....	8
1.6- Plan de Recursos Humanos .....	9
Proceso de Contratación y Capacitación.....	10
1.7- Plan de prevención de riesgos .....	11
Identificación y Evaluación de Riesgos.....	11
Medidas Preventivas y Correctivas .....	12
Plan de Implementación y Revisión .....	12
2. Especificación de requisitos .....	13
2.1- Introducción .....	13
2.2- Descripción general.....	13
2.3- Requisitos Específicos .....	14
2.3.1 Requerimientos Funcionales.....	14
2.3.2 Requerimientos de Interfaces Externas .....	15
2.3.3 Requerimientos Rendimiento .....	18
2.3.4 Obligaciones del Diseño .....	18
2.3.5 Atributos.....	19
3. Análisis.....	22
3.1- Introducción .....	22
Objetivos del Análisis .....	22
3.2- Diagrama de Clases. ....	22
Relación entre tablas.....	23
3.3- Diagrama de Casos de Uso .....	26
Diagrama de Casos de Uso .....	26
4. Diseño.....	32

ALUMNO: Jonathan Ramallo Rubio

CURSO: 2º DAW

PROYECTO: MangaSculp

4.1	Introducción .....	32
4.2	Capa de Presentación.....	32
4.3	Capa de Negocio o Lógica de la Aplicación .....	33
4.4	Capa de Persistencia o Datos .....	34
	Conclusión .....	36
5.	Implementación .....	37
6.	Evaluación .....	42
6.1-	Introducción .....	42
6.2-	Validaciones de páginas de Estilo .....	42
6.3-	Validación de Enlaces .....	42
6.4-	Validación de la Resolución.....	43
6.5-	Validación de Navegadores .....	43
6.6-	Otras validaciones .....	44
7.	Conclusión .....	45
7.1-	Valoración Personal del Trabajo Realizado .....	45
7.2-	Posibles Ampliaciones .....	46
	Bibliografía .....	46

## 1. Introducción

### 1.1 Presentación y Objetivos

#### Presentación:

El proyecto de desarrollo de esta web surge como respuesta a una necesidad creciente en el mercado español: la escasez de tiendas que ofrezcan figuras de anime de calidad y planos de impresiones 3D. A pesar del creciente interés en la cultura del manga y el anime, muchos aficionados se enfrentan a dificultades para adquirir estos productos debido a la falta de opciones locales y a las reticencias a importar directamente desde Asia. Esta web pretende abordar esta brecha en el mercado al ofrecer una plataforma accesible y confiable para la compra de figuras de anime y planos de impresiones 3D de alta calidad, producidos tanto localmente como a nivel global.

#### Objetivos Generales:

- Ampliar el espectro de conocimiento y afianzar los conocimientos adquiridos en el grado superior, aplicándolos en un proyecto práctico y relevante en el campo del comercio electrónico.
- Resolver el problema de los altos costes y la falta de accesibilidad de las figuras de anime de calidad en el mercado español, proporcionando una alternativa local y asequible para los aficionados.
- Fomentar la confianza y la seguridad en la compra de figuras de anime y planos de impresiones 3D mediante la garantía de calidad y la transparencia en los procesos de producción y distribución.

#### Objetivos Específicos:

- Facilitar el acceso a una amplia variedad de productos de anime y manga, incluyendo figuras de colección y planos de impresiones 3D de personajes populares y exclusivos.

- Educar y promover el uso de tecnologías emergentes como la impresión 3D, brindando a los usuarios la oportunidad de explorar y participar en la creación de sus propias figuras y diseños.
- Mejorar la experiencia del usuario mediante un diseño intuitivo y funcional de la web, que facilite la navegación y la búsqueda de productos, y fomente la interacción y la participación de la comunidad de aficionados al anime.

## 1.2 Contexto

1. **Tendencias del mercado:** El mercado de figuras de calidad está experimentando una creciente demanda, especialmente entre los aficionados al manga. Esta tendencia ha generado un aumento en los precios de las figuras importadas, lo que ha creado una oportunidad para explorar alternativas de producción local.
2. **Impacto de los costes de importación:** Los altos costes de importación de las figuras de calidad han sido un obstáculo significativo para su acceso en el mercado europeo. La necesidad de reducir estos costes se ha convertido en un objetivo clave para mejorar la accesibilidad y la competitividad del producto.
3. **Tecnología y fabricación:** Los avances en tecnología de fabricación, como la impresión 3D, han abierto nuevas posibilidades para la producción de figuras de calidad de forma más eficiente y económica. Estas tecnologías pueden aprovecharse para desarrollar un enfoque innovador en la producción de figuras de calidad.
4. **Cambio en las tendencias culturales:** El manga está experimentando un auge en popularidad, con una base de fans cada vez más diversa y exigente. Esto ha generado una mayor demanda de figuras de calidad que representen una amplia variedad de personajes y estilos.

## 1.3 Planteamiento del Problema (la idea).

1. La idea del proyecto es conseguir sacar ese producto de Asia y empezar a producirlo en Europa (o a nivel global) para reducir esos costes de importación tan elevados.

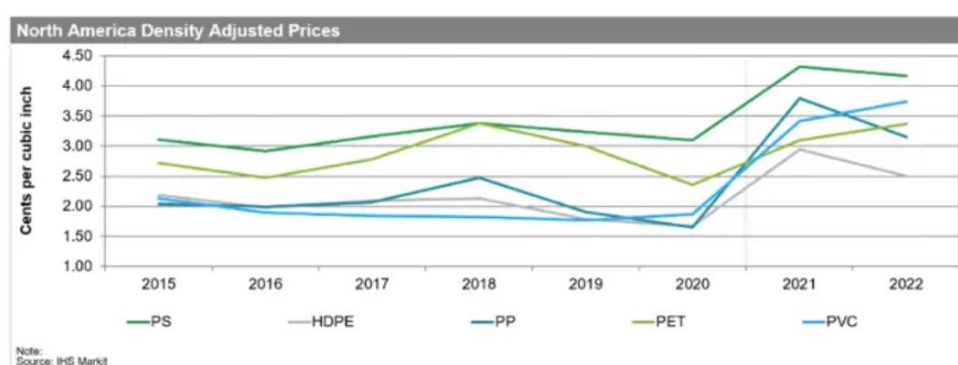
ALUMNO: Jonathan Ramallo Rubio

CURSO: 2º DAW

PROYECTO: MangaSculp

2. En España es difícil encontrar una tienda que las venda a buen precio, e incluso encontrar los planos por si quieres hacerlas tu, por lo que resolveríamos el problema de tener que dar el salto desde el mercado asiático a traerlo a Europa
3. Los precios se siguen disparando a nivel global, debido a que el mayor coste es de importación y de materiales de calidad.

Overall prices are elevated going forward



4. En el marco del Foro de la Comisión de la Industria del Plástico, Responsabilidad y Desarrollo Sustentable (Cipres) 2021, de la Asociación Nacional de la Industria Química (Aniq), Joel Morales, director ejecutivo de Poliolefinas para América de IHS Markit, pronosticó que los precios de las materias primas bajarán el próximo año, pero los costos de importación continuarán altos.
5. El manga está en pleno auge.
6. Analizando las tendencias, se observa una clara influencia de los valores actuales y entorno en las obras. El público demanda historias y personajes muy diferentes a los de hace unos años; y hoy en día estamos en un contexto donde las mujeres tienen voz, son independientes, y llevan las riendas de sus vidas. Cuanto más diversa es la sociedad, más diversa es la cultura.

## 1.4 Análisis de costes.

El análisis de costes de una aplicación web implica considerar varios factores:

1. **Desarrollo:**
  - **Recursos humanos:**

ALUMNO: Jonathan Ramallo Rubio

CURSO: 2º DAW

PROYECTO: MangaSculp

- Desarrolladores backend (Python, Django).
  - Desarrolladores frontend (TypeScript, React.js).
  - Diseñadores de UI/UX.
  - Administradores de base de datos (SQLite).
  - **Costos estimados:** Sueldos mensuales de los desarrolladores y diseñadores, calculados en base a tarifas locales o tarifas por hora.
2. **Infraestructura:**
    - **Servidores:** Costos de hosting, almacenamiento y ancho de banda.
    - **Servicios en la nube:** Servicios adicionales como almacenamiento en la nube (AWS, Azure, Google Cloud).
    - **Base de datos:** Mantenimiento y escalabilidad de SQLite (puede ser migrada a un sistema más robusto si se requiere).
  3. **Licencias y herramientas:**
    - Licencias de software para desarrollo y herramientas de gestión de proyectos.
    - Herramientas de CI/CD (Integración continua y entrega continua).
  4. **Marketing y promoción:**
    - Campañas de marketing digital.
    - Presencia en redes sociales.
    - Publicidad pagada.
  5. **Mantenimiento y soporte:**
    - Actualizaciones periódicas.
    - Soporte técnico y corrección de errores.
  6. **Otros costes:**
    - Dominio web.
    - Certificados SSL para seguridad.

### Ejemplo de desglose de costes (mensual):

- **Desarrollo:**
  - Desarrollador backend: 2700€
  - Desarrollador frontend: 2700€
  - Diseñador UI/UX: 2250€
  - Administrador de base de datos: 2250€
- **Infraestructura:**
  - Servidores: 450€
  - Servicios en la nube: 180€
- **Marketing:** 900€
- **Mantenimiento y soporte:** 720€
- **Otros:**
  - Dominio web y certificados SSL: 90€

**Total estimado mensual: 12.240€**

## 1.5 Plan de Financiación

Para financiar la aplicación MangaSculp, se pueden considerar varias fuentes y estrategias:

1. **Inversión inicial:**
  - Aportación de los fundadores.
  - Familia y amigos.
2. **Capital de riesgo:**
  - Buscar inversores ángeles o fondos de capital de riesgo interesados en proyectos tecnológicos.
3. **Crowdfunding:**
  - Plataformas como Kickstarter o Indiegogo para obtener fondos de futuros usuarios interesados en el proyecto.
4. **Subvenciones y programas de apoyo:**
  - Subvenciones gubernamentales o programas de aceleración para startups.
5. **Modelo de negocio:**
  - **Publicidad:** Incluir anuncios en la plataforma para generar ingresos adicionales.
  - **Comisiones:** Cobrar una comisión por ventas o transacciones realizadas a través de la plataforma.
6. **Préstamos:**
  - Préstamos bancarios o de instituciones financieras con planes de pago flexibles.
7. **Socios estratégicos:**
  - Asociarse con empresas del sector que puedan estar interesadas en invertir a cambio de participación en el proyecto o en sus beneficios futuros.

### Estrategia de financiación

1. **Fase inicial (6 meses):**
  - Capital propio y aportaciones de familia y amigos.
  - Buscar subvenciones y programas de aceleración.
2. **Fase de crecimiento (6-12 meses):**
  - Campaña de crowdfunding para validar el interés del mercado y obtener fondos.
  - Buscar inversores ángeles interesados en el proyecto.
3. **Fase de expansión (12-24 meses):**
  - Iniciar rondas de financiación con fondos de capital de riesgo.
  - Implementar modelos de monetización como suscripciones y publicidad.

Estas estrategias ayudarán a asegurar los fondos necesarios para el desarrollo, lanzamiento y escalado de MangaSculp.



## 1.6 Plan de Recursos Humanos

### 1. Desarrolladores Backend:

- **Número:** 2
- **Responsabilidades:**
  - Desarrollar y mantener la lógica del servidor y las APIs.
  - Integrar la base de datos con la aplicación.
  - Asegurar la seguridad y escalabilidad del backend.
- **Habilidades:**
  - **Tecnologías:** Python, Django.
  - **Experiencia:** Diseño de APIs RESTful, gestión de bases de datos (SQLite), buenas prácticas de seguridad.

### 2. Desarrolladores Frontend:

- **Número:** 2
- **Responsabilidades:**
  - Desarrollar la interfaz de usuario.
  - Implementar funcionalidades del lado del cliente.
  - Asegurar una buena experiencia de usuario.
- **Habilidades:**
  - **Tecnologías:** TypeScript, Node.js, frameworks como React o Angular.
  - **Experiencia:** Diseño responsivo, optimización de rendimiento, integración con APIs backend.

### 3. Diseñadores UI/UX:

- **Número:** 1
- **Responsabilidades:**
  - Diseñar interfaces atractivas y funcionales.
  - Crear wireframes, mockups y prototipos.
  - Realizar pruebas de usabilidad y recoger feedback de usuarios.
- **Habilidades:**
  - **Herramientas:** Figma, Sketch, Adobe XD.
  - **Experiencia:** Diseño centrado en el usuario, conocimiento de principios de diseño, capacidad de prototipado rápido.

### 4. Administrador de Bases de Datos:

- **Número:** 1
- **Responsabilidades:**
  - Administrar la base de datos.
  - Optimizar consultas y asegurar la integridad de los datos.
  - Realizar backups y planes de recuperación.
- **Habilidades:**
  - **Tecnologías:** SQLite, SQL.

- **Experiencia:** Optimización de bases de datos, planes de recuperación ante desastres, mantenimiento de integridad y seguridad de datos.

5. **Gerente de Proyecto:**

- **Número:** 1
- **Responsabilidades:**
  - Planificar y coordinar el desarrollo del proyecto.
  - Asegurar el cumplimiento de plazos y objetivos.
  - Gestionar la comunicación entre los equipos.
- **Habilidades:**
  - **Metodologías:** Gestión de proyectos ágil (Scrum, Kanban).
  - **Experiencia:** Coordinación de equipos multidisciplinares, planificación de recursos y tiempos, resolución de conflictos.

6. **Especialista en Marketing:**

- **Número:** 1
- **Responsabilidades:**
  - Desarrollar y ejecutar estrategias de marketing.
  - Gestionar la presencia en redes sociales y campañas publicitarias.
  - Analizar métricas y ajustar estrategias.
- **Habilidades:**
  - **Áreas:** Marketing digital, SEO/SEM.
  - **Experiencia:** Creación de contenido, análisis de datos, gestión de campañas publicitarias en línea.

7. **Soporte Técnico:**

- **Número:** 1
- **Responsabilidades:**
  - Atender consultas y problemas de los usuarios.
  - Realizar mantenimiento y actualizaciones.
  - Resolver incidencias técnicas.
- **Habilidades:**
  - **Tecnologías:** Conocimientos generales en desarrollo web.
  - **Experiencia:** Atención al cliente, resolución de problemas técnicos, documentación de incidencias.

## Proceso de Contratación y Capacitación

1. **Reclutamiento:**

- **Estrategias:**
  - Publicación de ofertas en portales de empleo y redes sociales.
  - Colaboración con agencias de reclutamiento.
  - Participación en ferias de empleo y eventos del sector.

2. **Selección:**

- **Proceso:**
  - Revisión de currículos y portafolios.
  - Entrevistas técnicas y de competencias.
  - Pruebas prácticas y de aptitudes.

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

### 3. Capacitación:

- **Programas:**
  - Programa de inducción para nuevos empleados.
  - Cursos y talleres sobre tecnologías específicas.
  - Mentoría y seguimiento continuo.

## 1.7 Plan de prevención de riesgos

### Identificación y Evaluación de Riesgos

#### 1. Riesgos Técnicos:

- **Fallo del servidor:**
  - **Impacto:** Interrupción del servicio, pérdida de datos.
  - **Mitigación:** Implementar soluciones de redundancia, backups regulares, monitorización continua.
- **Vulnerabilidades de seguridad:**
  - **Impacto:** Compromiso de datos, pérdida de confianza de los usuarios.
  - **Mitigación:** Auditorías de seguridad, actualizaciones periódicas, uso de protocolos seguros.
- **Problemas de escalabilidad:**
  - **Impacto:** Degradación del rendimiento, insatisfacción del usuario.
  - **Mitigación:** Planificar y probar la escalabilidad de la infraestructura, uso de soluciones de carga balanceada.

#### 2. Riesgos Humanos:

- **Falta de personal clave:**
  - **Impacto:** Retrasos en el desarrollo, disminución de la calidad.
  - **Mitigación:** Mantener una base de datos de candidatos, contratos de contingencia.
- **Baja productividad:**
  - **Impacto:** Retrasos, baja calidad del producto.
  - **Mitigación:** Evaluaciones de rendimiento, programas de motivación, ambiente de trabajo positivo.
- **Errores humanos:**
  - **Impacto:** Fallos en el sistema, bugs.
  - **Mitigación:** Capacitación continua, revisión de código, automatización de pruebas.

#### 3. Riesgos Operativos:

- **Interrupciones del servicio:**
  - **Impacto:** Pérdida de usuarios, daño a la reputación.
  - **Mitigación:** Planes de recuperación ante desastres, redundancia en servidores.
- **Fallas en la comunicación:**
  - **Impacto:** Descoordinación, retrasos.

- **Mitigación:** Uso de herramientas de gestión de proyectos, reuniones regulares de equipo.

## Medidas Preventivas y Correctivas

1. **Medidas Preventivas:**
  - **Plan de Contingencia:** Desarrollo de planes detallados para diferentes escenarios de riesgo.
  - **Capacitación Continua:** Programas regulares de formación y actualización para el equipo.
  - **Monitoreo y Alertas:** Implementación de sistemas de monitoreo continuo y alertas tempranas.
2. **Medidas Correctivas:**
  - **Respuesta Rápida:** Protocolos claros para la respuesta a incidencias.
  - **Evaluación Post-incidente:** Análisis detallado de incidentes para evitar recurrencias.
  - **Comunicación Transparente:** Informar a los usuarios sobre problemas y soluciones en tiempo real.

## Plan de Implementación y Revisión

1. **Implementación:**
  - **Cronograma:** Definir un calendario de implementación de todas las medidas preventivas y correctivas.
  - **Responsables:** Asignar responsables específicos para cada área de riesgo.
2. **Revisión y Actualización:**
  - **Auditorías Periódicas:** Realizar auditorías de riesgos y seguridad regularmente.
  - **Feedback:** Recoger feedback del equipo y usuarios para mejorar los planes.
  - **Actualización Continua:** Adaptar el plan de prevención de riesgos a nuevas amenazas y tecnologías.

## 2. Especificación de requisitos

### 2.1 Introducción

La aplicación "MangaSculp" es una plataforma de comercio electrónico diseñada para la compra y venta de productos relacionados con el mundo del manga y figuras coleccionables. Esta especificación de requisitos detalla las funcionalidades y características necesarias para el desarrollo y operación de la aplicación. Se describen los objetivos del sistema, las características generales y los requisitos específicos, incluyendo los funcionales y de interfaces externas. Esta especificación servirá como base para el desarrollo, la validación y la verificación del sistema.

### 2.2 Descripción general.

"MangaSculp" es una aplicación web que permite a los usuarios registrar cuentas, navegar por un catálogo de productos, agregar productos a su carrito de compras, realizar compras, dejar reseñas y gestionar su perfil. Los administradores pueden gestionar productos, usuarios, y pedidos. La aplicación está diseñada para ser intuitiva, segura y eficiente, utilizando tecnologías modernas para asegurar una experiencia de usuario satisfactoria.

#### Objetivos del Sistema:

- Facilitar la compra y venta de productos de manga y figuras coleccionables.
- Proporcionar una plataforma segura y eficiente para la gestión de productos y usuarios.
- Ofrecer una interfaz de usuario intuitiva y accesible para todos los usuarios.
- Permitir a los administradores gestionar y moderar el contenido de la plataforma.

#### Características Principales:

- Registro y autenticación de usuarios.
- Navegación y búsqueda de productos.
- Carrito de compras y procesamiento de pagos.
- Gestión de perfil de usuario.
- Sistema de reseñas y calificaciones.
- Gestión de productos, usuarios y pedidos para administradores.

## 2.3 Requisitos Específicos

### 2.3.1 Requerimientos Funcionales

#### 1. Registro de Usuario

- El sistema debe permitir a los usuarios crear una cuenta proporcionando nombre, correo electrónico y contraseña.

#### 2. Iniciar Sesión

- El sistema debe permitir a los usuarios iniciar sesión con su correo electrónico y contraseña.
- El sistema debe generar un token JWT para la autenticación del usuario.

#### 3. Administrar Perfil

- El sistema debe permitir a los usuarios ver y actualizar su información personal.
- El sistema debe guardar los cambios realizados en el perfil del usuario.

#### 4. Ver Productos

- El sistema debe permitir a los usuarios navegar y buscar productos utilizando filtros y categorías.
- El sistema debe mostrar una lista de productos que coincidan con los criterios de búsqueda.

#### 5. Ver Detalle de Producto

- El sistema debe mostrar información detallada de un producto seleccionado, incluyendo imágenes, descripción, precio y reseñas.

#### 6. Agregar Producto al Carrito

- El sistema debe permitir a los usuarios agregar productos a su carrito de compras.
- El sistema debe actualizar el carrito de compras con los productos seleccionados.

#### 7. Ver Carrito de Compras

- El sistema debe permitir a los usuarios ver los productos en su carrito de compras.
- El sistema debe permitir a los usuarios modificar las cantidades de los productos o eliminarlos del carrito.

## 8. Realizar Compra

- El sistema debe permitir a los usuarios proceder con la compra de los productos en su carrito.
- El sistema debe permitir a los usuarios ingresar detalles de envío y de pago.
- El sistema debe procesar los pagos a través de PayPal.

## 9. Ver Historial de Pedidos

- El sistema debe permitir a los usuarios ver sus pedidos anteriores.
- El sistema debe mostrar detalles de envío y estado de los pedidos.

## 10. Escribir Reseña

- El sistema debe permitir a los usuarios escribir reseñas y calificar los productos que han comprado.
- El sistema debe guardar las reseñas en la base de datos y asociarlas al producto y al usuario.

## 11. Ver Reseñas

- El sistema debe permitir a los usuarios ver las reseñas de otros usuarios sobre un producto.

## 12. Gestionar Productos (Administrador)

- El sistema debe permitir a los administradores agregar, actualizar y eliminar productos en el catálogo.

## 13. Gestionar Usuarios (Administrador)

- El sistema debe permitir a los administradores ver la lista de usuarios registrados.
- El sistema debe permitir a los administradores eliminar cuentas de usuarios.

## 2.3.2 Requerimientos de Interfaces Externas

### 2.3.2.1 Interfaces de los Usuarios

**Descripción:** Las interfaces de usuario son cruciales para garantizar una experiencia de usuario intuitiva, accesible y eficiente. Estas interfaces incluyen las páginas web, formularios, y elementos interactivos que los usuarios utilizarán para interactuar con la aplicación.

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

#### Requisitos:

- **Interfaz Web:** La aplicación debe proporcionar una interfaz web accesible desde navegadores modernos como Google Chrome, Mozilla Firefox, Microsoft Edge y Safari.
- **Diseño Responsivo:** La interfaz debe ser responsiva, adaptándose a diferentes tamaños de pantalla (escritorio, tableta, móvil) para asegurar una experiencia de usuario consistente y accesible en todos los dispositivos.
- **Accesibilidad:** La interfaz debe seguir las directrices de accesibilidad web (WCAG 2.1) para asegurar que todas las funcionalidades sean accesibles para usuarios con discapacidades.
- **Navegación Intuitiva:** La navegación debe ser intuitiva, con menús claros y una estructura lógica que permita a los usuarios encontrar rápidamente la información y las funcionalidades que buscan.
- **Feedback del Usuario:** La interfaz debe proporcionar feedback inmediato para acciones del usuario, como validaciones de formularios en tiempo real, mensajes de confirmación, y notificaciones de errores.

#### 2.3.2.2 Interfaces Hardware

**Descripción:** Las interfaces de hardware definen los requisitos mínimos y las especificaciones de los dispositivos físicos que se utilizarán para acceder y operar la aplicación.

#### Requisitos:

- **Dispositivos de Usuario:** La aplicación debe ser accesible desde dispositivos de usuario comunes, incluyendo:
  - **Ordenadores de Escritorio y Portátiles:** Con navegadores web modernos instalados.
  - **Tabletas y Teléfonos Móviles:** Con sistemas operativos iOS y Android.
- **Servidores:** Los servidores que alojaran la aplicación deben cumplir con los siguientes requisitos:
  - **CPU:** Procesadores multi-núcleo.
  - **Memoria:** Mínimo de 8GB de RAM para el entorno de producción.
  - **Almacenamiento:** Espacio suficiente para almacenar la base de datos y archivos estáticos (imágenes de productos, avatares de usuarios, etc.), con un mínimo de 100GB recomendado.
  - **Conectividad:** Conexión a internet de alta velocidad para manejar las solicitudes de los usuarios y la comunicación con servicios externos (como PayPal).



### 2.3.2.3 Interfaces Software

**Descripción:** Las interfaces de software describen cómo la aplicación interactúa con otros sistemas y servicios de software, incluyendo APIs, bibliotecas y frameworks.

#### Requisitos:

- **Sistema Operativo del Servidor:** La aplicación debe ser compatible con sistemas operativos de servidor comunes, como Ubuntu Linux y Windows Server.
- **Frameworks y Librerías:**
  - **Backend:** Django (versión 5.0.6), Django REST Framework, Django CORS Headers, Django SimpleJWT, Pillow, PyJWT, Python Dotenv, Sqlparse, Tzdata.
  - **Frontend:** React (versión 18.3.1), React DOM, React Router DOM, Axios, JWT Decode, Zustand, React Icons, Headless UI, Heroicons, React Hot Toast, Tanstack React Query, Tailwind CSS, Font Awesome, PayPal React SDK.
  - **Desarrollo:** TypeScript, Vite, ESLint, PostCSS, Autoprefixer.
- **APIs Externas:**
  - **PayPal:** Para el procesamiento de pagos.
- **Bases de Datos:**
  - **SQLite:** Para el entorno de desarrollo.
  - **PostgreSQL:** Para el entorno de producción.
- **Caché y Sesiones:**
  - **Redis:** Para la gestión de la caché y las sesiones.

### 2.3.2.4 Interfaces de Comunicaciones

**Descripción:** Las interfaces de comunicaciones describen cómo la aplicación maneja la comunicación y la transferencia de datos entre los diferentes componentes del sistema y con servicios externos.

#### Requisitos:

- **Protocolo de Comunicación:** La aplicación debe utilizar HTTP/HTTPS para todas las comunicaciones entre el cliente y el servidor.
- **Seguridad:** Todas las comunicaciones deben ser seguras, utilizando HTTPS para proteger los datos en tránsito.
- **Autenticación y Autorización:**
  - **JWT (JSON Web Tokens):** Para autenticar a los usuarios y autorizar el acceso a recursos protegidos.
  - **OAuth 2.0:** Para la integración con servicios externos como PayPal.

### 2.3.3 Requerimientos Rendimiento

**Descripción:** Los requisitos de rendimiento aseguran que la aplicación funcione de manera eficiente bajo diversas condiciones de carga, manteniendo tiempos de respuesta aceptables y utilizando los recursos de manera óptima.

**Requisitos:**

- **Tiempo de Respuesta:** Las páginas deben cargar en menos de 2 segundos bajo condiciones normales de uso.
- **Escalabilidad:** La aplicación debe poder manejar un aumento en el número de usuarios y transacciones sin una degradación significativa en el rendimiento.
- **Capacidad de Carga:** La aplicación debe soportar al menos 1000 usuarios concurrentes.
- **Procesamiento de Pedidos:** El sistema debe poder procesar al menos 50 pedidos por minuto sin errores ni retrasos significativos.
- **Optimización de Consultas:** Las consultas a la base de datos deben estar optimizadas para minimizar el tiempo de ejecución y el uso de recursos.
- **Uso de Recursos:** El sistema debe utilizar de manera eficiente la memoria y la CPU, evitando picos de uso excesivo.

### 2.3.4 Obligaciones del Diseño

#### 2.3.4.1 Estándares Cumplidos

**Descripción:** El diseño de la aplicación debe cumplir con ciertos estándares para garantizar la calidad, la seguridad y la interoperabilidad.

**Requisitos:**

- **WCAG 2.1:** Cumplimiento de las directrices de accesibilidad web para asegurar que la aplicación sea usable por personas con discapacidades.
- **OWASP:** Implementación de las mejores prácticas de seguridad recomendadas por el Open Web Application Security Project para prevenir vulnerabilidades comunes como XSS, CSRF y SQL injection.
- **W3C:** Cumplimiento de los estándares del World Wide Web Consortium (W3C) para HTML, CSS y JavaScript para asegurar la compatibilidad y la interoperabilidad.
- **GDPR:** Cumplimiento con el Reglamento General de Protección de Datos (GDPR) de la Unión Europea para proteger la privacidad y los datos personales de los usuarios.

#### 2.3.4.2 Limitaciones Hardware

**Descripción:** Las limitaciones de hardware definen las restricciones relacionadas con el equipo físico necesario para operar la aplicación.

**Requisitos:**

- **Requisitos Mínimos de Servidor:**
  - **CPU:** Procesadores multi-núcleo con al menos 4 núcleos.
  - **Memoria:** Mínimo de 8GB de RAM.
  - **Almacenamiento:** Al menos 100GB de espacio en disco para la base de datos y archivos estáticos.
  - **Conectividad:** Conexión a Internet de alta velocidad con un ancho de banda suficiente para manejar el tráfico esperado.

#### 2.3.5 Atributos

##### 2.3.5.1 Seguridad

**Descripción:** La seguridad es un atributo crítico para proteger los datos de los usuarios y la integridad de la aplicación.

**Requisitos:**

- **Autenticación:** Uso de tokens JWT para la autenticación de usuarios.
- **Autorización:** Implementación de controles de acceso basados en roles (RBAC) para restringir el acceso a funciones administrativas y sensibles.
- **Cifrado:** Uso de HTTPS para cifrar todas las comunicaciones entre el cliente y el servidor.
- **Almacenamiento Seguro:** Cifrado de datos sensibles en la base de datos, como contraseñas y tokens.
- **Auditoría:** Registro de actividades críticas y cambios en el sistema para fines de auditoría y monitoreo.

##### 2.3.5.2 Facilidades de Mantenimiento

**Descripción:** El mantenimiento eficiente asegura que la aplicación pueda actualizarse y mejorarse fácilmente con el tiempo.

**Requisitos:**

- **Documentación:** Documentación clara y completa del código, APIs y procedimientos de mantenimiento.

- **Modularidad:** Diseño modular del código para facilitar la actualización y el reemplazo de componentes individuales.

### 2.3.5.3 Portabilidad

**Descripción:** La portabilidad asegura que la aplicación pueda ser trasladada y ejecutada en diferentes entornos con mínima adaptación.

#### Requisitos:

- **Compatibilidad Multiplataforma:** La aplicación debe ser compatible con múltiples sistemas operativos de servidor, incluyendo Ubuntu Linux y Windows Server.
- **Contenedorización:** Uso de Docker para empaquetar la aplicación y todas sus dependencias, asegurando que se ejecute de manera consistente en diferentes entornos.
- **Configuración Independiente:** Separación de la configuración de la aplicación del código fuente, utilizando archivos de configuración y variables de entorno.

### 2.3.5.4 Otros Requerimientos

**Descripción:** Otros requerimientos incluyen aspectos adicionales que son importantes para el funcionamiento y la usabilidad de la aplicación.

#### Requisitos:

- **Usabilidad:** La aplicación debe ser fácil de usar, con una interfaz de usuario intuitiva y una experiencia de usuario coherente.
- **Disponibilidad:** La aplicación debe estar disponible al menos el 99.9% del tiempo, excluyendo mantenimientos programados.
- **Escalabilidad:** La arquitectura de la aplicación debe permitir la escalabilidad horizontal para manejar incrementos en la carga de trabajo.
- **Soporte Técnico:** Disponibilidad de soporte técnico para resolver problemas y asistir a los usuarios y administradores.

### Conclusión

La especificación de requisitos presentada para la aplicación proporciona una base sólida para el desarrollo, asegurando que todas las funcionalidades clave y las interfaces externas estén claramente definidas. Esta especificación permitirá a los desarrolladores construir una aplicación robusta, escalable y fácil de mantener, que satisfaga las necesidades de los usuarios y administradores. Las obligaciones de diseño, los requisitos de rendimiento y los atributos adicionales aseguran que la aplicación no solo funcione correctamente, sino que también ofrezca una experiencia de usuario

**ALUMNO:** Jonathan Ramallo Rubio  
**CURSO:** 2º DAW  
**PROYECTO:** MangaSculp

superior y cumpla con los estándares de la industria. Esta especificación servirá como guía detallada durante todo el proceso de desarrollo y operación de la aplicación.

## 3. Análisis

### 3.1 Introducción

La aplicación "MangaSculp" está construida sobre una arquitectura robusta que incluye diversas funcionalidades críticas, tales como la gestión de usuarios, productos, pedidos y reseñas, así como un sistema avanzado de permisos y autenticación.

Este análisis explorará en profundidad las diferentes capas de la aplicación, desde su base de datos hasta la lógica de negocio y la interfaz de usuario, proporcionando una visión integral de su diseño y funcionamiento.

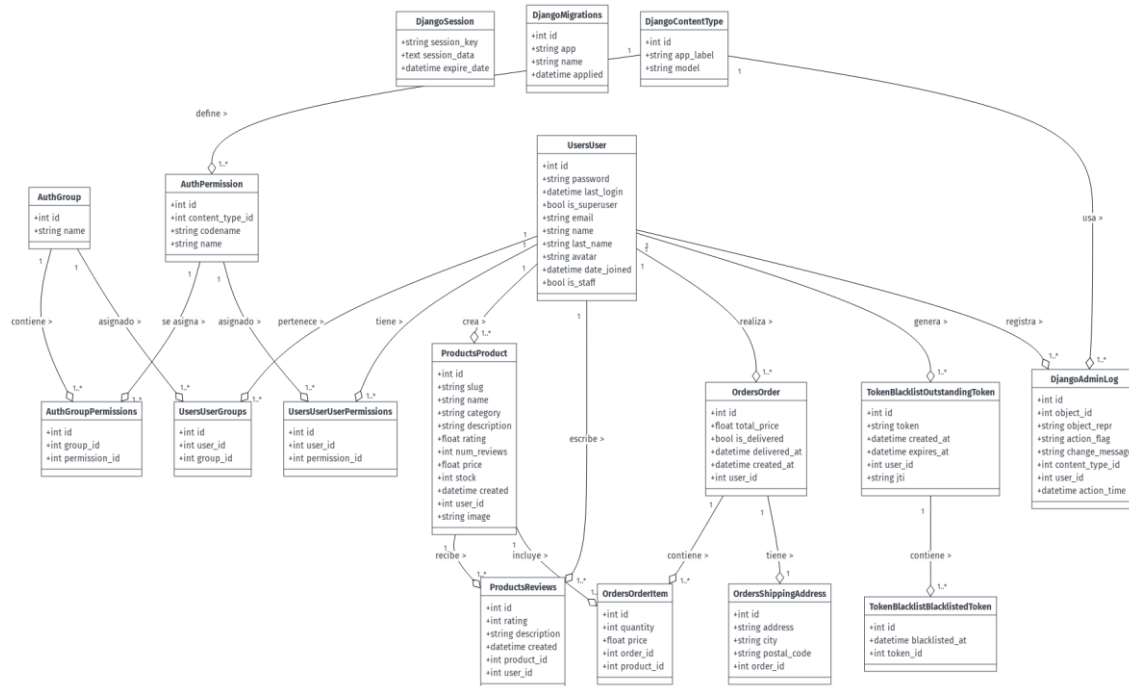
#### Objetivos del Análisis

El objetivo principal de este análisis es proporcionar una comprensión completa de la estructura y las funcionalidades de "MangaSculp".

Para lograr esto, se abordarán los siguientes puntos:

1. **Diseño de la Base de Datos:** Examinar las entidades principales y sus relaciones, lo que incluye tablas como usuarios, productos, pedidos y reseñas, así como los permisos y grupos de autenticación.
2. **Lógica de Negocio:** Analizar cómo la aplicación maneja las operaciones comerciales, desde la creación de productos y la gestión de inventarios hasta el procesamiento de pedidos y la gestión de envíos.
3. **Casos de Uso:** Presentar diagramas de casos de uso para ilustrar cómo interactúan los diferentes actores con la aplicación y qué funcionalidades específicas están disponibles para ellos.

### 3.2 Diagrama de Clases.



## Relación entre tablas

### 1. AuthGroup y AuthGroupPermissions

- **Relación:** 1 a muchos
- **Descripción:** Cada grupo de autenticación (AuthGroup) puede tener múltiples permisos asignados (AuthGroupPermissions). Esto permite la asociación de varios permisos a un solo grupo.

### 2. AuthPermission y AuthGroupPermissions

- **Relación:** 1 a muchos
- **Descripción:** Cada permiso (AuthPermission) puede ser asignado a múltiples grupos a través de la tabla AuthGroupPermissions. Esto permite que un solo permiso sea reutilizado por diferentes grupos.

### 3. UsersUser y UsersUserGroups

- **Relación:** 1 a muchos
- **Descripción:** Cada usuario (UsersUser) puede pertenecer a múltiples grupos (UsersUserGroups). Esto permite asignar un usuario a varios grupos diferentes.

### 4. AuthGroup y UsersUserGroups

- **Relación:** 1 a muchos
- **Descripción:** Cada grupo (AuthGroup) puede tener múltiples usuarios asociados a través de la tabla UsersUserGroups. Esto permite gestionar los usuarios que pertenecen a cada grupo.

### 5. UsersUser y UsersUserPermissions

- **Relación:** 1 a muchos

- **Descripción:** Cada usuario (UsersUser) puede tener múltiples permisos asignados directamente (UsersUserUserPermissions). Esto permite que un usuario tenga permisos específicos que no están relacionados con su grupo.
- 6. **AuthPermission y UsersUserUserPermissions**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada permiso (AuthPermission) puede ser asignado a múltiples usuarios a través de la tabla UsersUserUserPermissions. Esto permite la reutilización de permisos específicos para diferentes usuarios.
- 7. **UsersUser y ProductsProduct**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada usuario (UsersUser) puede crear múltiples productos (ProductsProduct). Esto permite que un usuario sea el autor de varios productos en la base de datos.
- 8. **ProductsProduct y ProductsReviews**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada producto (ProductsProduct) puede recibir múltiples reseñas (ProductsReviews). Esto permite la evaluación de un producto por varios usuarios.
- 9. **UsersUser y ProductsReviews**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada usuario (UsersUser) puede escribir múltiples reseñas (ProductsReviews). Esto permite que un usuario reseñe diferentes productos.
- 10. **UsersUser y OrdersOrder**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada usuario (UsersUser) puede realizar múltiples pedidos (OrdersOrder). Esto permite que un usuario tenga varios pedidos registrados.
- 11. **OrdersOrder y OrdersOrderItem**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada pedido (OrdersOrder) puede contener múltiples artículos de pedido (OrdersOrderItem). Esto permite que un solo pedido incluya varios productos.
- 12. **ProductsProduct y OrdersOrderItem**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada producto (ProductsProduct) puede ser incluido en múltiples artículos de pedido (OrdersOrderItem). Esto permite que un producto se venda en varios pedidos.
- 13. **OrdersOrder y OrdersShippingAddress**
  - **Relación:** 1 a 1
  - **Descripción:** Cada pedido (OrdersOrder) tiene una única dirección de envío (OrdersShippingAddress). Esto asegura que cada pedido se envíe a una dirección específica.
- 14. **TokenBlacklistOutstandingToken y TokenBlacklistBlacklistedToken**
  - **Relación:** 1 a muchos
  - **Descripción:** Cada token pendiente (TokenBlacklistOutstandingToken) puede tener múltiples tokens en la lista negra



(TokenBlacklistBlacklistedToken). Esto permite un seguimiento detallado de los tokens invalidados.

**15. UsersUser y TokenBlacklistOutstandingToken**

- **Relación:** 1 a muchos
- **Descripción:** Cada usuario (UsersUser) puede generar múltiples tokens pendientes (TokenBlacklistOutstandingToken). Esto permite gestionar los tokens de acceso activos de un usuario.

**16. DjangoContentType y AuthPermission**

- **Relación:** 1 a muchos
- **Descripción:** Cada tipo de contenido (DjangoContentType) puede definir múltiples permisos (AuthPermission). Esto permite asignar permisos específicos a diferentes tipos de contenido en la aplicación.

**17. DjangoContentType y DjangoAdminLog**

- **Relación:** 1 a muchos
- **Descripción:** Cada tipo de contenido (DjangoContentType) puede ser usado en múltiples registros de log de administración (DjangoAdminLog). Esto permite un seguimiento detallado de las acciones administrativas relacionadas con diferentes tipos de contenido.

**18. UsersUser y DjangoAdminLog**

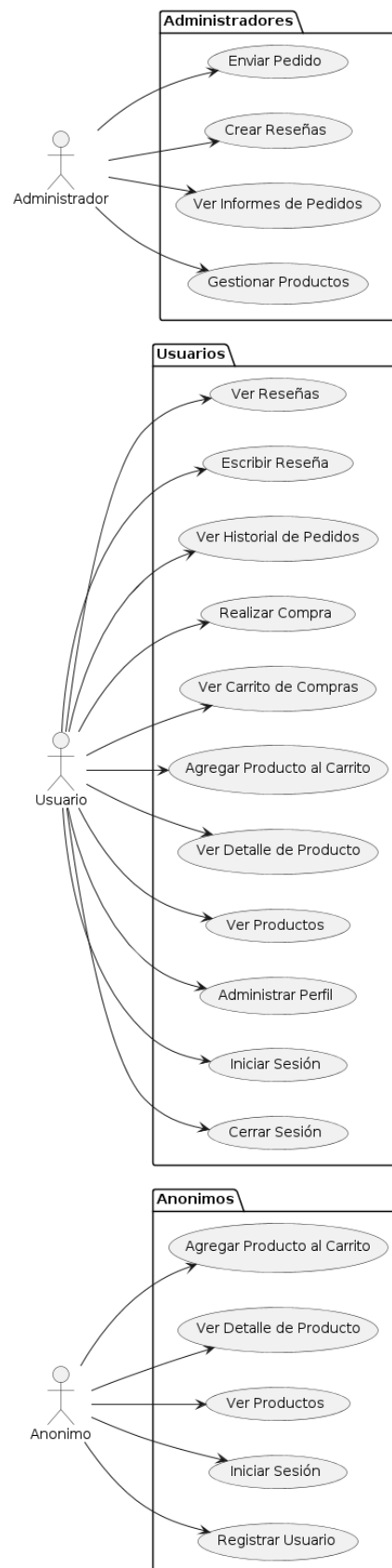
- **Relación:** 1 a muchos
- **Descripción:** Cada usuario (UsersUser) puede registrar múltiples acciones en el log de administración (DjangoAdminLog). Esto permite auditar las actividades administrativas de los usuarios.

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

### 3.3 Diagrama de Casos de Uso

#### Diagrama de Casos de Uso

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp



ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

### Actores

- **Usuario anónimo:** Navega por el sitio sin iniciar sesión.
- **Usuario registrado:** Inicia sesión y accede a funcionalidades adicionales.
- **Administrador:** Gestiona usuarios, productos y pedidos.

### Casos de Uso Detallados

#### Anónimos

1. **Registrar Usuario Descripción:** Permite a una persona crear una cuenta de usuario en la aplicación. **Actor:** Anónimo **Flujo Principal:**
  - El usuario accede a la página de registro.
  - El usuario ingresa sus datos personales (nombre, correo electrónico, contraseña).
  - El sistema valida los datos ingresados.
  - El sistema crea una nueva cuenta de usuario.
  - El sistema envía un correo de verificación al usuario.
  - El usuario verifica su cuenta a través del enlace en el correo.
2. **Iniciar Sesión Descripción:** Permite a un usuario autenticarse en la aplicación. **Actor:** Anónimo **Flujo Principal:**
  - El usuario accede a la página de inicio de sesión.
  - El usuario ingresa su correo electrónico y contraseña.
  - El sistema valida las credenciales ingresadas.
  - El sistema genera un token JWT para el usuario.
  - El sistema redirige al usuario a la página principal.
3. **Ver Productos Descripción:** Permite a los usuarios navegar y buscar productos en el catálogo. **Actor:** Anónimo **Flujo Principal:**
  - El usuario accede a la página de productos.
  - El usuario puede buscar productos utilizando filtros y categorías.
  - El sistema muestra la lista de productos que coinciden con los criterios de búsqueda.
4. **Ver Detalle de Producto Descripción:** Permite a los usuarios ver información detallada sobre un producto específico. **Actor:** Anónimo **Flujo Principal:**
  - El usuario selecciona un producto de la lista.
  - El sistema muestra la página de detalles del producto, incluyendo imágenes, descripciones, precio, y reseñas.

5. **Agregar Producto al Carrito Descripción:** Permite a los usuarios añadir productos a su carrito de compras. **Actor:** Anónimo **Flujo Principal:**
- El usuario accede a la página de detalles del producto.
  - El usuario selecciona la cantidad deseada.
  - El usuario hace clic en el botón "Agregar al carrito".
  - El sistema actualiza el carrito de compras del usuario.

### Usuarios

6. **Administrar Perfil Descripción:** Permite a un usuario ver y editar su perfil. **Actor:** Usuario **Flujo Principal:**
- El usuario accede a la página de perfil.
  - El usuario actualiza su información personal (nombre, apellido, correo electrónico, avatar).
  - El sistema valida los datos ingresados.
  - El sistema guarda los cambios en la base de datos.
7. **Ver Carrito de Compras Descripción:** Permite a los usuarios ver y modificar los productos en su carrito de compras. **Actor:** Usuario **Flujo Principal:**
- El usuario accede a la página del carrito de compras.
  - El sistema muestra los productos en el carrito, incluyendo cantidades y precios.
  - El usuario puede modificar las cantidades o eliminar productos del carrito.
8. **Realizar Compra Descripción:** Permite a los usuarios proceder con la compra de los productos en su carrito. **Actor:** Usuario **Flujo Principal:**
- El usuario accede a la página del carrito de compras.
  - El usuario hace clic en el botón "Proceder al pago".
  - El usuario ingresa los detalles de envío y de pago.
  - El sistema valida los datos ingresados.
  - El sistema procesa el pago a través de PayPal.
  - El sistema confirma la compra y envía una confirmación por correo electrónico.
9. **Ver Historial de Pedidos Descripción:** Permite a los usuarios ver sus pedidos anteriores. **Actor:** Usuario **Flujo Principal:**
- El usuario accede a la página de historial de pedidos.

- El sistema muestra una lista de los pedidos anteriores del usuario, incluyendo detalles de envío y estado del pedido.

10. **Escribir Reseña Descripción:** Permite a los usuarios escribir reseñas y calificar los productos que han comprado. **Actor:** Usuario **Flujo Principal:**

- El usuario accede a la página de detalles del producto.
- El usuario hace clic en el botón "Escribir reseña".
- El usuario ingresa su reseña y calificación.
- El sistema guarda la reseña en la base de datos y la asocia al producto y al usuario.

11. **Ver Reseñas Descripción:** Permite a los usuarios ver las reseñas de otros usuarios sobre un producto. **Actor:** Usuario **Flujo Principal:**

- El usuario accede a la página de detalles del producto.
- El sistema muestra las reseñas y calificaciones de otros usuarios para el producto.

12. **Cerrar Sesión Descripción:** Permite a un usuario autenticado cerrar su cuenta de usuario. **Actor:** Usuario **Flujo Principal:**

- El usuario accede a la página principal.
- El usuario clicca en su perfil.
- El usuario clicca en cerrar sesión.
- El sistema destruye la sesión del usuario.
- El sistema destruye el token JWT de sesión del usuario.

13. **Iniciar Sesión Descripción:** Permite a un usuario autenticado volver a iniciar sesión (útil para cambios de cuenta). **Actor:** Usuario **Flujo Principal:**

- El usuario accede a la página de inicio de sesión.
- El usuario ingresa su correo electrónico y contraseña.
- El sistema valida las credenciales ingresadas.
- El sistema genera un token JWT para el usuario.
- El sistema redirige al usuario a la página principal.

### *Administradores*

14. **Gestionar Productos Descripción:** Permite a los administradores crear, actualizar y eliminar productos en el catálogo. **Actor:** Administrador **Flujo Principal:**

ALUMNO: Jonathan Ramallo Rubio

CURSO: 2º DAW

PROYECTO: MangaSculp

- El administrador accede a la página de gestión de productos.
- El administrador puede agregar un nuevo producto ingresando los detalles del producto.
- El administrador puede actualizar los detalles de un producto existente.
- El administrador puede eliminar un producto del catálogo.

15. **Ver Pedidos Descripción:** Permite a los administradores ver el estado y estadísticas sobre los pedidos y el uso de la aplicación. **Actor:** Administrador **Flujo Principal:**

- El administrador accede a la página de pedidos.
- El sistema muestra todos los pedidos de los usuarios y los datos de cada pedido.

16. **Crear Reseñas Descripción:** Permite a los administradores escribir reseñas de productos. **Actor:** Administrador **Flujo Principal:**

- El administrador accede a la página de detalles del producto.
- El administrador hace clic en el botón "Escribir reseña".
- El administrador ingresa su reseña y calificación.
- El sistema guarda la reseña en la base de datos y la asocia al producto y al administrador.

17. **Enviar Pedido Descripción:** Permite a los administradores actualizar el estado de los pedidos y marcar los pedidos como enviados. **Actor:** Administrador **Flujo Principal:**

- El administrador accede a la página de gestión de pedidos.
- El administrador selecciona un pedido.
- El administrador actualiza el estado del pedido a "Enviado".

18. **Gestionar Usuarios Descripción:** Permite a los administradores gestionar las cuentas de los usuarios. **Actor:** Administrador **Flujo Principal:**

- El administrador accede a la página de gestión de usuarios.
- El administrador puede ver la lista de usuarios registrados.
- El administrador puede eliminar una cuenta de usuario.

## 4. Diseño

### 4.1 Introducción

El diseño de la aplicación web "MangaSculp" sigue una arquitectura en capas que garantiza la separación de responsabilidades, facilita el mantenimiento y la escalabilidad, y asegura una experiencia de usuario fluida y consistente. Cada capa está diseñada con tecnologías y patrones específicos que optimizan su función. A continuación, se presenta una descripción detallada del diseño de cada capa de la aplicación.

### 4.2 Capa de Presentación

#### Objetivos de Diseño:

- Crear una interfaz de usuario intuitiva y atractiva.
- Garantizar la accesibilidad y responsividad en diferentes dispositivos.
- Facilitar la interacción fluida entre el usuario y las funcionalidades de la aplicación.

#### Diseño Seguido:

- **Componentización:** Se utilizó React para dividir la interfaz de usuario en componentes reutilizables y modulares. Cada componente se encarga de una parte específica de la UI, como formularios, listas de productos, carritos de compras y páginas de detalle de productos.
- **State Management:** Se implementó Zustand para gestionar el estado global de la aplicación, permitiendo compartir y sincronizar datos entre componentes de manera eficiente.
- **Routing:** React Router DOM se utilizó para manejar la navegación y las rutas dentro de la aplicación, asegurando una transición fluida entre las diferentes vistas y páginas.
- **Estilizado:** Se empleó Tailwind CSS para un diseño rápido y responsivo, aplicando utilidades de CSS directamente en los componentes. Esto permitió una rápida iteración y adaptación del diseño visual.
- **Accesibilidad:** Se siguieron las directrices de accesibilidad web (WCAG) para asegurar que todos los usuarios, independientemente de sus capacidades, puedan utilizar la aplicación de manera efectiva.



- **Integración de APIs:** Axios se utilizó para realizar solicitudes HTTP a la API del backend, gestionando la comunicación y el intercambio de datos con el servidor.
- **Notificaciones:** Se integró React Hot Toast para mostrar notificaciones y mensajes de estado a los usuarios.
- **Iconografía:** Se utilizó React Icons y Font Awesome para los iconos, proporcionando una interfaz visualmente atractiva y fácil de entender.
- **Pagos:** Se integró el SDK de PayPal React para facilitar los pagos en la aplicación.
- **Consultas de Datos:** Tanstack React Query se utilizó para gestionar el estado de las solicitudes y la caché de datos, mejorando el rendimiento y la experiencia del usuario.

### 4.3 Capa de Negocio o Lógica de la Aplicación

#### Objetivos de Diseño:

- Implementar la lógica de negocio de manera modular y reutilizable.
- Asegurar la integridad y consistencia de las operaciones del negocio.
- Facilitar la integración con la capa de presentación y la capa de datos.

#### Diseño Seguido:

- **Framework:** Se utilizó Django como el framework principal debido a su robustez y flexibilidad. Django proporciona una estructura clara y modular que facilita la organización de la lógica de negocio en aplicaciones web.
- **APIs RESTful:** Se creó una API RESTful utilizando Django REST Framework (DRF), definiendo endpoints claros y bien documentados para manejar las operaciones de CRUD (crear, leer, actualizar, eliminar) de las entidades principales como usuarios, productos, pedidos y reseñas.
- **Autenticación y Autorización:** Se implementó Django SimpleJWT para la gestión de autenticación basada en tokens JWT, asegurando que solo los usuarios autenticados puedan acceder a ciertas funcionalidades. Además, se definieron roles y permisos específicos para controlar el acceso a recursos y acciones dentro de la aplicación.
- **Gestión de Imágenes:** Se utilizó Pillow para el procesamiento y manejo de imágenes dentro de la aplicación.
- **Configuración de Entorno:** Se empleó Python Dotenv para cargar variables de entorno desde un archivo .env, facilitando la gestión de configuraciones sensibles y específicas del entorno.
- **Validaciones y Reglas de Negocio:** Se definieron y aplicaron validaciones de datos y reglas de negocio directamente en los modelos y serializers de Django, asegurando que los datos ingresados cumplan con los requisitos y restricciones necesarios.
- **CORS:** Se configuró django-cors-headers para manejar las solicitudes CORS y permitir el acceso a la API desde diferentes dominios.

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

## 4.4 Capa de Persistencia o Datos

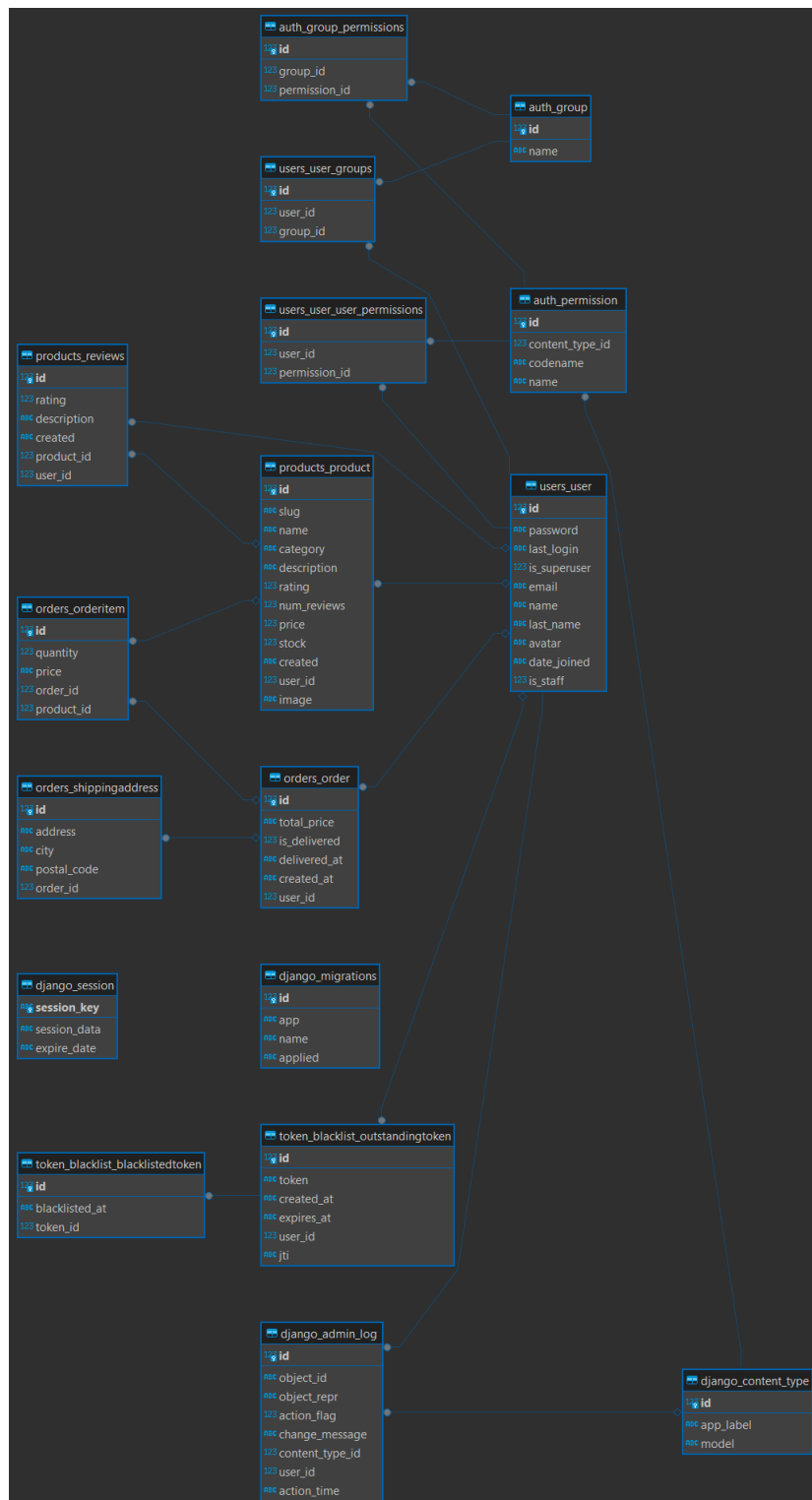
### Objetivos de Diseño:

- Almacenar y gestionar los datos de manera eficiente y segura.
- Asegurar la integridad y consistencia de los datos.
- Facilitar el acceso y la manipulación de los datos por parte de la capa de negocio.

### Diseño Seguido:

- **Base de Datos Relacional:** Se utilizó SQLite como la base de datos relacional principal, aprovechando su simplicidad y facilidad de uso. SQLite es adecuado para el desarrollo y las pruebas debido a su naturaleza ligera y configuración mínima, permitiendo un desarrollo ágil y eficiente.
- **Modelo de Datos:** Se diseñó un modelo de datos relacional bien estructurado, utilizando el ORM de Django para mapear las entidades de la aplicación a tablas en la base de datos. Esto incluyó la definición de relaciones claras entre las tablas mediante claves primarias y foráneas.

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp



- **Migraciones:** Se utilizaron las migraciones de Django para manejar cambios en el esquema de la base de datos de manera controlada y versionada. Esto permitió evolucionar la estructura de la base de datos junto con la aplicación.

ALUMNO: Jonathan Ramallo Rubio

CURSO: 2º DAW

PROYECTO: MangaSculp

- **Optimización de Consultas:** Se optimizaron las consultas a la base de datos mediante el uso de índices en campos frecuentemente consultados, mejorando el rendimiento de la aplicación.
- **Caché y Sesiones:** Se integró Redis para la gestión de la caché y las sesiones, mejorando el rendimiento y la velocidad de respuesta de la aplicación.
- **Seguridad de Datos:** Se implementaron medidas de seguridad para proteger los datos almacenados, incluyendo el cifrado de datos sensibles y la implementación de políticas de acceso restringido.
- **Procesamiento de Consultas SQL:** Se utilizó sqlparse para el análisis y formateo de consultas SQL, asegurando que las consultas sean correctas y optimizadas.

## Conclusión

El diseño de la aplicación "MangaSculp" se ha basado en un enfoque de arquitectura en capas, que permite una clara separación de responsabilidades y facilita el desarrollo, mantenimiento y escalabilidad de la aplicación. Cada capa ha sido diseñada utilizando tecnologías y patrones específicos que optimizan su función, asegurando una integración eficiente y una experiencia de usuario coherente y fluida. Este enfoque modular no solo mejora la calidad del software, sino que también permite una mayor flexibilidad y adaptabilidad a futuras necesidades y cambios.

## 5. Implementación

### 5.1 Tecnologías utilizadas en el desarrollo del proyecto

El desarrollo de la aplicación web "MangaSculp" ha requerido la integración de diversas tecnologías modernas y herramientas de desarrollo, permitiendo construir una solución robusta, escalable y eficiente. A continuación, se detallan las tecnologías utilizadas en cada capa de la aplicación:

#### Backend:

- **Django:** Framework web de alto nivel en Python que permite un desarrollo rápido y limpio con un enfoque en la reutilización de componentes. Utilizado para la lógica de negocio y la capa de persistencia.
- **Django REST Framework (DRF):** Extensión de Django que facilita la creación de APIs RESTful.
- **Django CORS Headers:** Permite manejar los headers CORS en las aplicaciones Django, permitiendo el acceso a la API desde diferentes dominios.
- **Django SimpleJWT:** Biblioteca para la autenticación basada en tokens JWT, facilitando el manejo seguro de sesiones.
- **Pillow:** Biblioteca de Python para el procesamiento de imágenes.
- **Python Dotenv:** Permite cargar variables de entorno desde un archivo .env.
- **asgiref:** Biblioteca de Python que proporciona una implementación de referencia para el servidor de aplicaciones asincrónicas ASGI.
- **sqlparse:** Herramienta de análisis y formateo de SQL.
- **tzdata:** Base de datos de zonas horarias.
- **PyJWT:** Biblioteca de Python para codificar y decodificar JSON Web Tokens.

#### Frontend:

- **React:** Biblioteca de JavaScript para construir interfaces de usuario.
- **React DOM:** Paquete para manejar el DOM en aplicaciones React.
- **React Router DOM:** Biblioteca para manejar rutas y navegación en aplicaciones React.
- **Axios:** Cliente HTTP basado en promesas para hacer solicitudes a la API.
- **JWT Decode:** Biblioteca para decodificar JSON Web Tokens.
- **Zustand:** Biblioteca para la gestión de estado en React.
- **React Icons:** Conjunto de iconos SVG para aplicaciones React.
- **Headless UI y Heroicons:** Componentes de interfaz de usuario accesibles y estilizados para React.
- **React Hot Toast:** Biblioteca para mostrar notificaciones y mensajes de estado.
- **Tanstack React Query:** Herramienta para gestionar el estado de las solicitudes y la caché de datos.
- **Tailwind CSS:** Framework de CSS utilitario para un diseño rápido y receptivo.
- **Font Awesome:** Biblioteca de iconos vectoriales.

- **PayPal React SDK:** Biblioteca para integrar pagos con PayPal en aplicaciones React.

#### DevDependencies:

- **TypeScript:** Superset de JavaScript que agrega tipos estáticos.
- **Vite:** Herramienta de construcción rápida para aplicaciones modernas de JavaScript.
- **ESLint:** Herramienta de linting para identificar y reportar patrones en el código.
- **PostCSS y Autoprefixer:** Herramientas para transformar CSS con JavaScript.
- **Standard:** Estilo de código JavaScript.
- **React Refresh:** Herramienta para la recarga en caliente de módulos en React durante el desarrollo.

## 5.2 Descripción del Proyecto

La implementación de "MangaSculp" se organiza en tres capas principales: la capa de presentación, la capa de negocio o lógica de la aplicación, y la capa de persistencia o de datos. Cada una de estas capas está diseñada para interactuar entre sí de manera eficiente y cohesiva, garantizando una experiencia de usuario fluida y una gestión eficaz de la lógica de negocio y los datos.

### 5.2.1 Capa de Presentación

La capa de presentación es responsable de la interfaz de usuario y la interacción del usuario con la aplicación. Esta capa se ha desarrollado utilizando tecnologías modernas para asegurar una experiencia de usuario atractiva y responsiva.

#### Componentes Implementados:

1. **Interfaz de Usuario:**
  - **Registro e Inicio de Sesión:** Formularios con validaciones en tiempo real y mensajes de error claros, implementados con React y gestionados con componentes reutilizables.
  - **Perfil de Usuario:** Páginas para que los usuarios puedan ver y editar su información personal, diseñadas para ser intuitivas y fáciles de usar.
  - **Catálogo de Productos:** Listados de productos con opciones de filtrado y búsqueda avanzada. Implementación de componentes dinámicos para manejar la paginación y la carga de datos de manera eficiente.
  - **Página de Detalles del Producto:** Vista detallada de cada producto con descripciones completas, imágenes, y reseñas de usuarios.
  - **Carrito de Compras:** Componente interactivo que permite a los usuarios agregar y eliminar productos, y proceder al pago.
  - **Historial de Pedidos:** Sección donde los usuarios pueden ver el estado de sus pedidos anteriores y detalles de envío.
  - **Sistema de Reseñas:** Formularios para que los usuarios dejen reseñas y calificaciones, con moderación por parte de los administradores.
2. **Comunicación con el Servidor:**

- **Axios:** Uso de Axios para realizar solicitudes HTTP asíncronas a la API del backend sin recargar la página.
- **WebSockets:** Implementación de comunicación en tiempo real para ciertas funcionalidades como notificaciones y actualizaciones en vivo.

#### Tecnologías y Herramientas Utilizadas:

- **React:** Para construir la interfaz de usuario dinámica.
- **Tailwind CSS:** Framework de CSS utilitario para un diseño rápido y receptivo.
- **React Router DOM:** Para manejar la navegación y las rutas.
- **Axios:** Para realizar solicitudes HTTP asíncronas.
- **JWT Decode:** Para decodificar tokens JWT y manejar la autenticación.
- **Zustand:** Para la gestión de estado.
- **React Icons y Font Awesome:** Para los iconos SVG.
- **React Hot Toast:** Para mostrar notificaciones y mensajes de estado.
- **Headless UI y Heroicons:** Para componentes de UI accesibles y estilizados.
- **PayPal React SDK:** Para integrar pagos con PayPal.
- **Vite:** Para la construcción y desarrollo rápido de la aplicación.
- **TypeScript:** Para añadir tipos estáticos a JavaScript.

### 5.2.2 Capa de Negocio o Lógica de la Aplicación

La capa de negocio se encarga de la lógica que gestiona las operaciones entre la interfaz de usuario y la capa de datos. Esta capa se ha implementado utilizando Django, aprovechando su estructura modular y sus potentes funcionalidades de ORM y autenticación.

#### Componentes Implementados:

1. **Gestión de Usuarios:**
  - **Registro y Autenticación:** Módulos para la creación de cuentas, inicio de sesión, y autenticación de usuarios utilizando tokens JWT.
  - **Roles y Permisos:** Implementación de un sistema de roles y permisos que controla el acceso a diferentes funcionalidades de la aplicación.
2. **Gestión de Productos:**
  - **CRUD de Productos:** Funcionalidades para crear, leer, actualizar y eliminar productos en la base de datos.
  - **Inventario:** Gestión del inventario de productos con alertas para niveles bajos de stock.
  - **Categorías y Filtros:** Organización de productos en categorías y subcategorías, con soporte para filtros avanzados en las búsquedas.
3. **Gestión de Pedidos:**
  - **Proceso de Compra:** Flujo de trabajo que abarca desde agregar productos al carrito hasta procesar el pago y confirmar el pedido.
  - **Gestión de Pedidos:** Funcionalidades para rastrear el estado de los pedidos y notificar a los usuarios sobre cambios en el estado.
  - **Logística de Envío:** Integración con servicios de envío para calcular costos y tiempos de entrega, y generar etiquetas de envío.

#### 4. Reseñas y Calificaciones:

- **Sistema de Reseñas:** Módulos que permiten a los usuarios dejar reseñas y calificaciones de los productos.
- **Moderación:** Herramientas para que los administradores puedan moderar y gestionar las reseñas de los usuarios.

#### Tecnologías y Herramientas Utilizadas:

- **Django:** Framework web para el backend, que proporciona una estructura modular y funcionalidades avanzadas.
- **Django REST Framework:** Para la creación de APIs RESTful que gestionan la comunicación entre la capa de presentación y la capa de datos.
- **Celery:** Biblioteca para la ejecución de tareas asíncronas, utilizada para manejar tareas en segundo plano como el envío de correos electrónicos y la actualización de inventarios.

### 5.2.3 Capa de Persistencia o de Datos

La capa de persistencia es responsable de almacenar y gestionar los datos de la aplicación. Esta capa utiliza una base de datos relacional para asegurar la integridad y consistencia de los datos.

#### Componentes Implementados:

##### 1. Base de Datos:

- **Modelo Relacional:** Diseño de un modelo de datos relacional utilizando PostgreSQL, con tablas bien definidas para usuarios, productos, pedidos, reseñas, y más.
- **ORM (Object-Relational Mapping):** Uso de Django ORM para mapear las entidades de la aplicación a tablas en la base de datos, facilitando las operaciones CRUD.

##### 2. Entidades Principales:

- **Usuarios (UsersUser):** Tabla que almacena información sobre los usuarios, incluyendo credenciales y datos personales.
- **Grupos y Permisos (AuthGroup, AuthPermission):** Tablas para gestionar los roles y permisos de los usuarios.
- **Productos (ProductsProduct):** Tabla que contiene detalles de los productos disponibles en el catálogo.
- **Reseñas (ProductsReviews):** Tabla que guarda las reseñas y calificaciones de los productos por parte de los usuarios.
- **Pedidos (OrdersOrder):** Tabla que registra los pedidos realizados por los usuarios, incluyendo detalles de los productos y estado del pedido.
- **Artículos de Pedido (OrdersOrderItem):** Tabla que detalla los productos incluidos en cada pedido.
- **Direcciones de Envío (OrdersShippingAddress):** Tabla que almacena las direcciones de envío asociadas a los pedidos.



ALUMNO: Jonathan Ramallo Rubio

CURSO: 2º DAW

PROYECTO: MangaSculp

- **Sesiones (DjangoSession):** Tabla que gestiona las sesiones activas de los usuarios.
- **Tokens de Lista Negra (TokenBlacklistBlacklistedToken):** Tabla que almacena tokens que han sido invalidados.
- **Log de Administración (DjangoAdminLog):** Tabla que registra las acciones administrativas realizadas en la aplicación.

### 3. Tecnologías y Herramientas Utilizadas:

- **PostgreSQL:** Base de datos relacional utilizada para el almacenamiento persistente de datos.
- **Redis:** Utilizado como almacenamiento en memoria para la caché y la gestión de sesiones, mejorando el rendimiento de la aplicación.
- **Django ORM:** Para mapear las entidades de la aplicación a la base de datos, facilitando las operaciones CRUD y asegurando la integridad de los datos.
- **Alembic/Migrations de Django:** Herramientas para manejar migraciones de esquema de la base de datos, asegurando que la estructura de la base de datos evoluciona junto con la aplicación.

### Consideraciones de Diseño:

- **Integridad de Datos:** Uso de claves primarias y foráneas para asegurar la integridad referencial entre las tablas.
- **Rendimiento:** Indexación de campos frecuentemente consultados y optimización de consultas para mejorar el rendimiento de la aplicación.

## 6. Evaluación

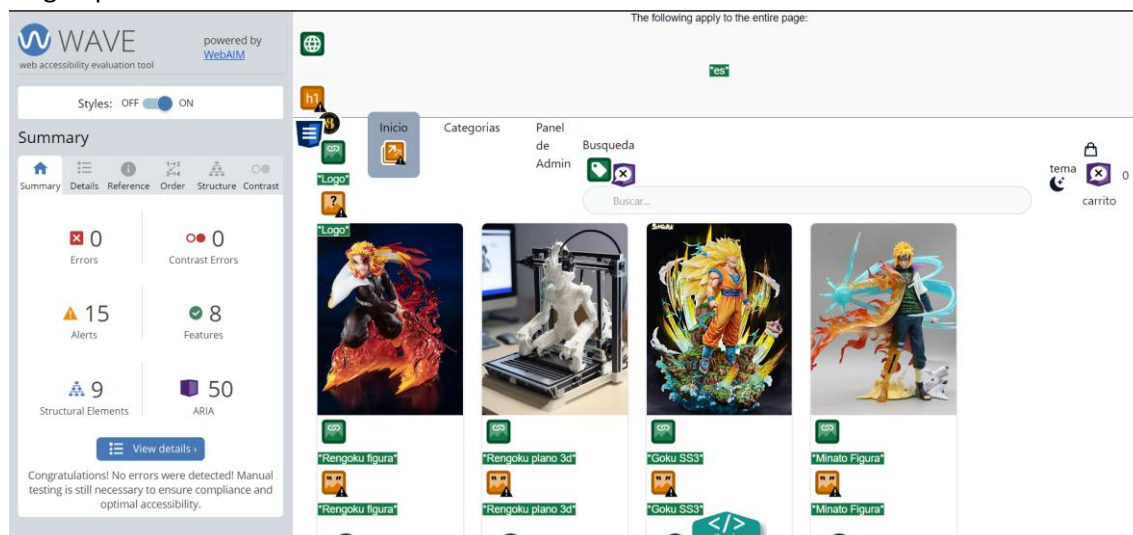
### 6.1- Introducción

En esta sección se detallará el proceso de evaluación realizado para garantizar la calidad y la funcionalidad de la aplicación web MangaSculp. Se abordarán diferentes aspectos como la validación de estilos, enlaces, resolución y navegadores, así como otras validaciones realizadas dentro del código para mejorar la seguridad y la experiencia del usuario.

### 6.2- Validaciones de páginas de Estilo

Dado que toda la presentación visual de la aplicación se ha desarrollado utilizando Tailwind CSS, no se aplicaron validaciones específicas a páginas de estilos CSS individuales.

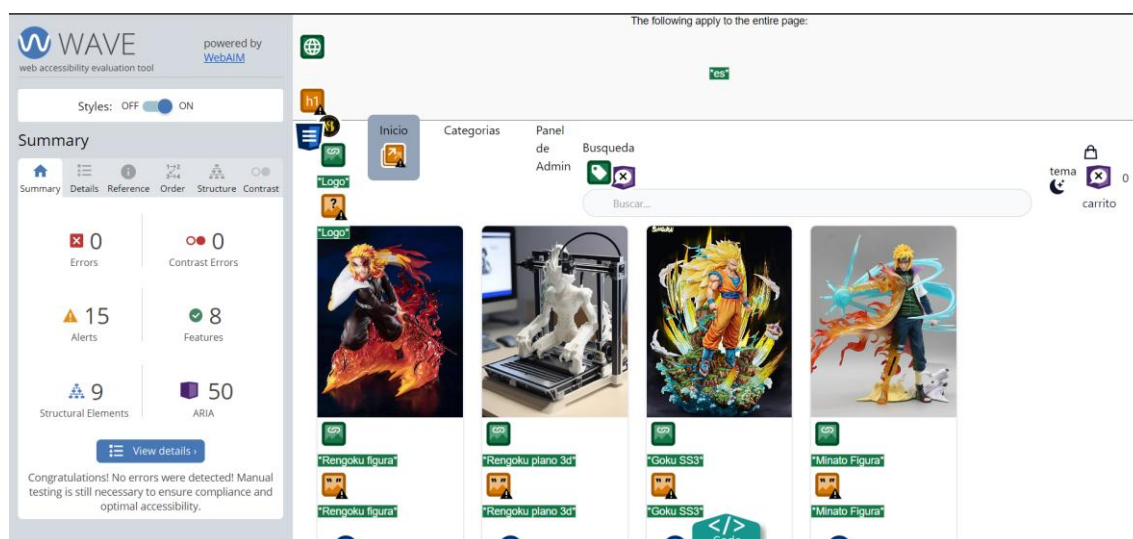
Adicionalmente se usó la herramienta automatizada WAVE para ver que no había ningún problema con el css de contraste



### 6.3- Validación de Enlaces

Se llevó a cabo una validación exhaustiva de los enlaces presentes en la aplicación. Cada enlace fue probado manualmente para asegurarse de que redirigiera correctamente al destino previsto. Actualmente, todos los enlaces han sido verificados y funcionan correctamente.

Adicionalmente se llevó a cabo una validación exhaustiva de los enlaces presentes en la aplicación utilizando la herramienta WAVE. Cada enlace fue probado manualmente y también se verificó su accesibilidad con WAVE para asegurarse de que cumpliera con las pautas de accesibilidad web. Se confirmó que todos los enlaces cumplen con las normativas de accesibilidad y redirigen correctamente al destino previsto.



## 6.4- Validación de la Resolución

La aplicación ha sido diseñada con Tailwind CSS, lo que garantiza que sea completamente responsiva. Esto significa que la página se renderizará correctamente en una amplia gama de resoluciones, incluyendo dispositivos móviles, tablets y ordenadores.

## 6.5- Validación de Navegadores

La aplicación MangaSculp ha sido desarrollada utilizando React, Tailwind y Django. Se recomienda y se espera que la aplicación sea compatible con los navegadores más

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

recientes y populares, incluyendo Chrome, Firefox, Safari y Edge, en sus versiones más actualizadas.

## 6.6- Otras validaciones

Dentro del código de la aplicación, se han implementado validaciones adicionales para mejorar la seguridad y la integridad de los datos.

Se han utilizado expresiones regulares en los formularios para garantizar que la información proporcionada por los usuarios cumpla con ciertos criterios predefinidos. Además, se ha proporcionado información adicional y orientación a los administradores para ayudarles a evitar entradas incorrectas o maliciosas.

Estas medidas adicionales contribuyen a mejorar la calidad y la fiabilidad de la aplicación MangaSculp.

## 7. Conclusión

### 7.1- Valoración Personal del Trabajo Realizado

#### Análisis DAFO

- **Debilidades (Weaknesses):**
  - Falta de experiencia previa en proyectos similares.
    - La curva de aprendizaje ha sido pronunciada, lo que ha afectado el tiempo de desarrollo y la eficiencia en ciertas etapas del proyecto.
  - Dependencia de tecnologías específicas que podrían no ser adecuadas a largo plazo.
    - La base de datos se tendría que migrar a una PostgreSQL
- **Amenazas (Threats):**
  - Competencia intensa en el mercado de aplicaciones web.
  - Rápida evolución tecnológica que podría hacer obsoleta la aplicación.
  - Riesgos de seguridad y privacidad de los datos de los usuarios.
    - La gestión de datos de usuarios implica riesgos significativos de seguridad y privacidad, que deben ser manejados con rigor para evitar brechas y pérdida de confianza.
- **Fortalezas (Strengths):**
  - Alta motivación y compromiso del equipo.
  - Innovación en la combinación de tecnologías utilizadas.
  - Comunidad de usuarios potencialmente amplia y apasionada.
- **Oportunidades (Opportunities):**
  - Crecimiento del mercado de manga digital.
  - Posibilidad de expansión internacional.
  - Colaboraciones con editoriales de manga y eventos de anime.

#### Análisis CAME

- **Corregir Debilidades:**
  - Invertir en formación y desarrollo del equipo.
  - Buscar financiación adicional para aumentar los recursos disponibles.
  - Evaluar y adaptar tecnologías conforme a las necesidades a largo plazo.
- **Afrontar Amenazas:**
  - Realizar análisis de mercado y mantenerse actualizado sobre tendencias.
  - Desarrollar un plan de contingencia para la evolución tecnológica.
  - Implementar medidas de seguridad más robustas y actualizarlas periódicamente.
- **Mantener Fortalezas:**
  - Continuar innovando y mejorando la aplicación.
  - Fortalecer la comunidad de usuarios mediante interacción constante y feedback.

- **Explotar Oportunidades:**
  - Desarrollar estrategias de marketing para penetrar en el mercado de manga digital.
  - Explorar oportunidades de expansión a otros mercados.
  - Buscar asociaciones estratégicas con editoriales y eventos relevantes.

## 7.2- Posibles Ampliaciones

1. **Funciones Adicionales:**
  - Integrar funciones de realidad aumentada para una experiencia más interactiva.
  - Implementar un sistema de recomendaciones personalizadas basado en el comportamiento del usuario.
2. **Expansión de la Plataforma:**
  - Ampliar la aplicación a otras plataformas como iOS y Android.
  - Desarrollar una versión web optimizada para dispositivos móviles.
3. **Colaboraciones y Alianzas:**
  - Establecer colaboraciones con editoriales de manga para obtener contenido exclusivo.
  - Alianzas con tiendas de manga y productos relacionados para promociones cruzadas.
4. **Monetización:**
  - Integrar publicidad de manera no intrusiva para generar ingresos adicionales.
5. **Internacionalización:**
  - Traducir la aplicación a varios idiomas para captar una audiencia global.
  - Adaptar el contenido a las preferencias culturales de diferentes regiones.

## Bibliografía

1.3- Plástico.com. (2022, 5 de enero). Precios de resinas continuarán altos en 2022. Plástico.com. <https://www.plastico.com/es/noticias/precios-de-resinas-continuaran-altos-en-2022>

1.3- Fnac. (s.f.). Las tendencias del manga en Fnac: Más global y más en femenino. Fnac. Recuperado de <https://www.fnac.es/Las-tendencias-del-manga-en-Fnac-Mas-global-y-mas-en-femenino/cp9868/w-4>

1.4- ProjectManager.com. (2023). Cost-Benefit Analysis: A Quick Guide with Examples and Templates. ProjectManager.com. Recuperado de

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

<https://www.projectmanager.com/blog/cost-benefit-analysis-for-projects-a-step-by-step-guide>

1.5- blazebusinessandlegal.com. (2023). The Ultimate Guide to Startup Funding: BlazeBusinessAndLegal.com. Recuperado de <https://blazebusinessandlegal.com.au/startup-funding/>

1.6- Indeed Editorial Team. (2022). What Is HR Management Training. Indeed. Recuperado de <https://www.indeed.com/career-advice/career-development/hr-management-training>

1.6- Indeed Editorial Team. (2022). 11 HR Trends To Know for 2024. Indeed. Recuperado de <https://www.indeed.com/career-advice/career-development/hr-trends>

1.7- ProjectManager.com. (2024). Project Strategy. ProjectManager.com. Recuperado de <https://www.projectmanager.com/blog/project-strategy>

2.3.2.1- W3C. (2018). Web Content Accessibility Guidelines (WCAG) 2.1. W3C. Recuperado de <https://www.w3.org/TR/WCAG21/>

2.3.2.1- Mozilla Developer Network (MDN). (2023). Web Performance. MDN Web Docs. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/Performance>

2.3.2.2- Django Software Foundation. (2023). Django Documentation. Django Project. Recuperado de <https://docs.djangoproject.com/en/stable/>

2.3.2.3- React. (2023). React Documentation. Recuperado de <https://react.dev/>

2.3.2.3- PayPal. (2023). PayPal Developer Documentation. PayPal. Recuperado de <https://developer.paypal.com/docs/api/overview/>

2.3.2.3- Docker. (2023). Docker Documentation. Docker. Recuperado de <https://docs.docker.com/>

2.3.2.4- Mozilla Developer Network (MDN). (2023). HTTP. MDN Web Docs. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/HTTP>

2.3.4.1- OWASP Foundation. (2021). OWASP Top Ten Project. OWASP. Recuperado de <https://owasp.org/www-project-top-ten/>

2.3.4.1- European Union. (2018). General Data Protection Regulation (GDPR). Recuperado de <https://gdpr.eu/>

3.2- Lucid.app (2024) Lucid/Lucidchart. lucid.app Recuperado de <https://lucid.app>

3.3- PlantUML.com (2024) Plant Uml plantuml.com Recuperado de <https://www.plantuml.com>

ALUMNO: Jonathan Ramallo Rubio  
CURSO: 2º DAW  
PROYECTO: MangaSculp

6.5- Legacy ReactJS Documentation. (2024). Requisitos del entorno JavaScript. Recuperado de <https://es.legacy.reactjs.org/docs/javascript-environment-requirements.html>

7.1- blog.hubspot.es (2024) Como hacer un análisis DAFO: blog.hubspot.es Recuperado de <https://blog.hubspot.es/sales/como-hacer-analisis-dafo>

7.2- Simla.com (2021) ¿Qué es un análisis CAME y cómo hacer uno para tu empresa? simla.com Recuperado de [https://www.simla.com/blog/analisis-came#:~:text=El%20Análisis%20CAME%20\(Corregir%2C%20Adaptar,las%20amenazas%20exter,nas%20del%20mercado.](https://www.simla.com/blog/analisis-came#:~:text=El%20Análisis%20CAME%20(Corregir%2C%20Adaptar,las%20amenazas%20exter,nas%20del%20mercado.)