<

**Documento de Especificación de Arquitectura**

Realizado por:

<<Diego Castillo>>

**<<**Carolina Marcela Blanco Mendoza>>

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
|  |  | Carolina Blanco  Diego Castillo |  |  |

**Contenido**

Tabla de contenido

[Documento de Arquitectura de Software 5](#_Toc194960083)

[1.1 Introducción 5](#_Toc194960084)

[1.2 Propósito 5](#_Toc194960085)

[1.3 Referencias 5](#_Toc194960086)

[1.4 Definiciones acrónimos y abreviaciones 5](#_Toc194960087)

[Generalidades del Proyecto 6](#_Toc194960088)

[2. 1Problema a Resolver 6](#_Toc194960089)

[2.2 Descripción General del Sistema a Desarrollar (General y por módulo) 7](#_Toc194960090)

[2.3 Identificación de los Stakeholders y sus responsabilidades 7](#_Toc194960091)

[Vistas de la arquitectura 9](#_Toc194960092)

[3.1 Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el software) 9](#_Toc194960093)

[VISTA DE PROCESOS 10](#_Toc194960094)

[4.1 Diagrama de Actividades 11](#_Toc194960095)

[VISTA LÓGICA 12](#_Toc194960096)

[5.1 Diagramas - Clases (Actualizado) 13](#_Toc194960097)

[5.2 Diagrama - Objetos 14](#_Toc194960098)

[5.3 Diagramas - Secuencia 15](#_Toc194960099)

[VISTA DE IMPLEMENTACIÓN 18](#_Toc194960100)

[6.1 Diagrama de Componentes 19](#_Toc194960101)

[6.2 Diagrama de Paquetes 19](#_Toc194960102)

[6.3 VISTA DE DESPLIEGUE 21](#_Toc194960103)

[Arquitectura en capas 21](#_Toc194960104)

[VISTA DE DATOS 22](#_Toc194960105)

[8.1 Modelo Relacional normalizado-tercera forma normal de la base de datos (SGBD). 22](#_Toc194960106)

[Definición de Interfaces de Usuario 23](#_Toc194960107)

[Características Generales de Calidad 26](#_Toc194960108)

[10. 1 Tamaño y performance 26](#_Toc194960109)

[10.2 Calidad 27](#_Toc194960110)

[10.3 Usabilidad 27](#_Toc194960111)

[10.4 Eficiencia 27](#_Toc194960112)

[10.5 Seguridad 28](#_Toc194960113)

[10.6 Confiabilidad 28](#_Toc194960114)

[10.7 Mantenimiento 28](#_Toc194960115)

[10.8 Estándares 29](#_Toc194960116)

# Documento de Arquitectura de Software

## 1.1 Introducción

Este documento describe el diseño y funcionamiento del software, destacando los componentes clave y sus interrelaciones. Expone la estructura lógica y la organización de las capas de presentación, lógica del negocio y persistencia, así como los objetivos tecnológicos de la aplicación. Además, incluye diagramas de clases, casos de usos, secuencia, actividades, paquetes, y despliegue que representan visualmente las funcionalidades, tecnologías utilizadas y cómo estas interactúan, garantizando una solución integral para el acceso a anime, manga y novelas ligeras en un solo lugar.

## 1.2 Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema, teniendo un mayor objetivo fomentar el deporte, la visión del proyecto, teniendo en cuenta las necesidades de los clientes y sus respectivos usos y alcances de este software

## 1.3 Referencias

Este documento se crea a partir de las anteriores fases de desarrollo las cuales se consignaron en los siguientes documentos:

* Documento de historias de usuario.
* Documento de Visión del Proyecto.
* Planilla de análisis del software.

## 1.4 Definiciones acrónimos y abreviaciones

**ARQUITECTURA DE SOFTWARE:** conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCIÓN DE ARQUITECTURA**: colección de productos de documentación.

**VISTAS:** es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

# Generalidades del Proyecto

## 2. 1Problema a Resolver

En la actualidad, los fanáticos de las series japonesas, como animes, mangas y novelas ligeras, se enfrentan a la dificultad de acceder a todo el contenido relacionado con una misma historia de manera centralizada. Cada uno de estos formatos de contenido (anime, manga y novela ligera) se encuentra disperso en diversas plataformas o sitios web especializados, lo que obliga al usuario a navegar por diferentes servicios y enfrentar una experiencia fragmentada.

Las páginas de anime, por ejemplo, no suelen ofrecer acceso directo al manga o a las novelas ligeras correspondientes a la misma serie. De manera similar, los sitios dedicados a mangas o novelas ligeras no vinculan de manera eficiente el contenido visualizado con el contenido en otros medios, lo que puede resultar confuso para el usuario que desea conocer todos los aspectos de una serie.

Esta falta de centralización afecta la experiencia del usuario, ya que requiere una gran inversión de tiempo y esfuerzo para encontrar y organizar toda la información y el contenido relacionado con una serie. Además, para los nuevos fanáticos que se introducen en el universo de las series japonesas, la fragmentación del contenido puede ser abrumadora y dificultar su inmersión en el mundo de las historias que tanto les interesan.

La ausencia de una plataforma integral que agrupe el anime, el manga y la novela ligera en un solo lugar limita la posibilidad de disfrutar de manera óptima y sencilla de toda la narrativa y la evolución de las series. Esto genera una necesidad de una solución que permita a los usuarios acceder rápidamente a todo el contenido relacionado con una serie sin tener que navegar entre múltiples páginas y plataformas.

.

## 2.2 Descripción General del Sistema a Desarrollar (General y por módulo)

Este software es una plataforma digital que integra la visualización y gestión de contenido relacionado con animes, historietas (mangas, manhwa) y novelas ligeras. Su objetivo es proporcionar a los usuarios una experiencia completa para consumir, publicar y discutir este tipo de contenido. Los usuarios pueden registrarse como lectores o editores, y un administrador supervisa y gestiona la actividad en la plataforma. También se fomenta la creación de comunidades mediante grupos y herramientas sociales.

**El Módulo de Autenticación y Usuarios** gestiona el acceso al sistema y la administración de usuarios. Registro y login (correo/contraseña, redes sociales). Recuperación de contraseña. Gestión de roles (Lector, Editor, Administrador). Configuración de perfil.

**El Módulo de Contenido** gestiona animes, mangas y novelas ligeras.

Subida y edición de contenido por editores.

Organización por categorías y etiquetas.

Gestión de episodios/capítulos.

Control de versiones y actualizaciones.

**El Módulo de Grupos y Comunidad** facilita la interacción entre los usuarios.

Creación de grupos de lectura y discusión.

 Administración de miembros.

 Publicación de comentarios y debates.

 Eventos en comunidad.

## 2.3 Identificación de los Stakeholders y sus responsabilidades

|  |  |
| --- | --- |
| **STAKEHOLDER** | **DESCRIPCIÓN** |
| Editor | Persona responsable de gestionar el contenido que los usuarios (Lectores) ven. Puede subir nuevas historietas (mangas, manhwas, etc.), novelas ligeras y animes, organizarlos por capítulos o episodios, actualizar información como sinopsis, autores y géneros, y asegurarse de que todo el material esté correctamente presentado. Además, tiene la capacidad de modificar el contenido existente y agregar nuevos lanzamientos para mantener actualizado el catálogo*.* |
| Administrador | Persona que supervisa y gestiona toda la plataforma. Se encarga de revisar las publicaciones de los editores, resolver problemas técnicos, y responder a reportes de los lectores. Además, puede bloquear usuarios que infrinjan normas, enviar notificaciones masivas, y acceder a estadísticas detalladas como las series más vistas o las tendencias por región. Su rol es clave para mantener el orden, asegurar el buen funcionamiento del software y tomar decisiones informadas sobre el contenido y las políticas de la plataforma. |
| Lector | Persona que utiliza la plataforma para explorar, leer y ver el contenido disponible. Puede personalizar su perfil, recibir recomendaciones basadas en su historial, calificar series, y llevar un seguimiento del progreso de sus lecturas o visualizaciones. También tiene opciones para personalizar su experiencia, como ajustar la velocidad de reproducción y ocultar series ya vistas, garantizando así́ una experiencia fluida y adaptada a sus gustos. |

# Vistas de la arquitectura

## 3.1 Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el software)

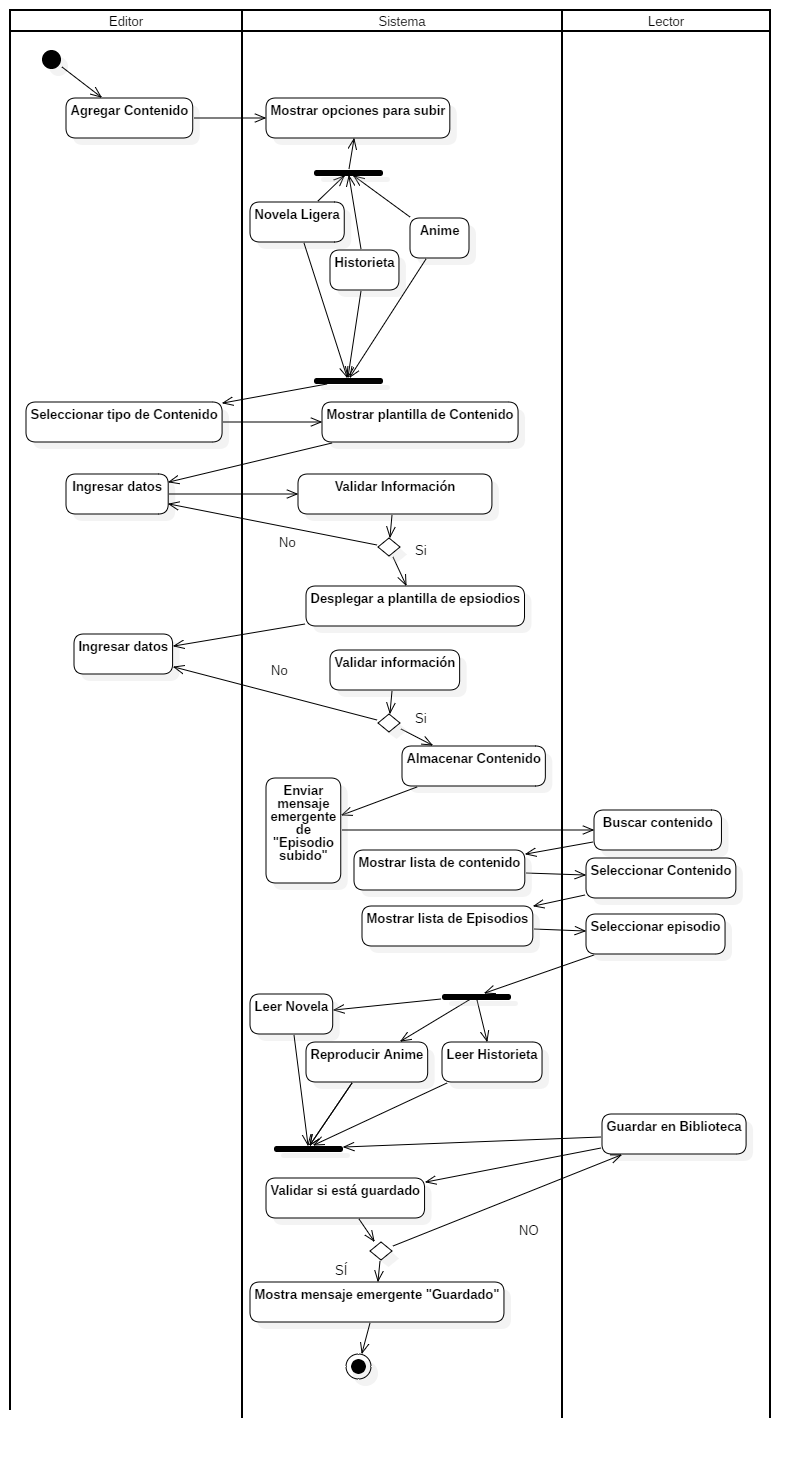
descripción breve

# VISTA DE PROCESOS

Todos los diagramas que hayan elaborado en fase II, por módulo (aplica para los puntos 3.2.1 al 3.3.2), debidamente argumentados cada uno de los diagramas.

## 4.1 Diagrama de Actividades

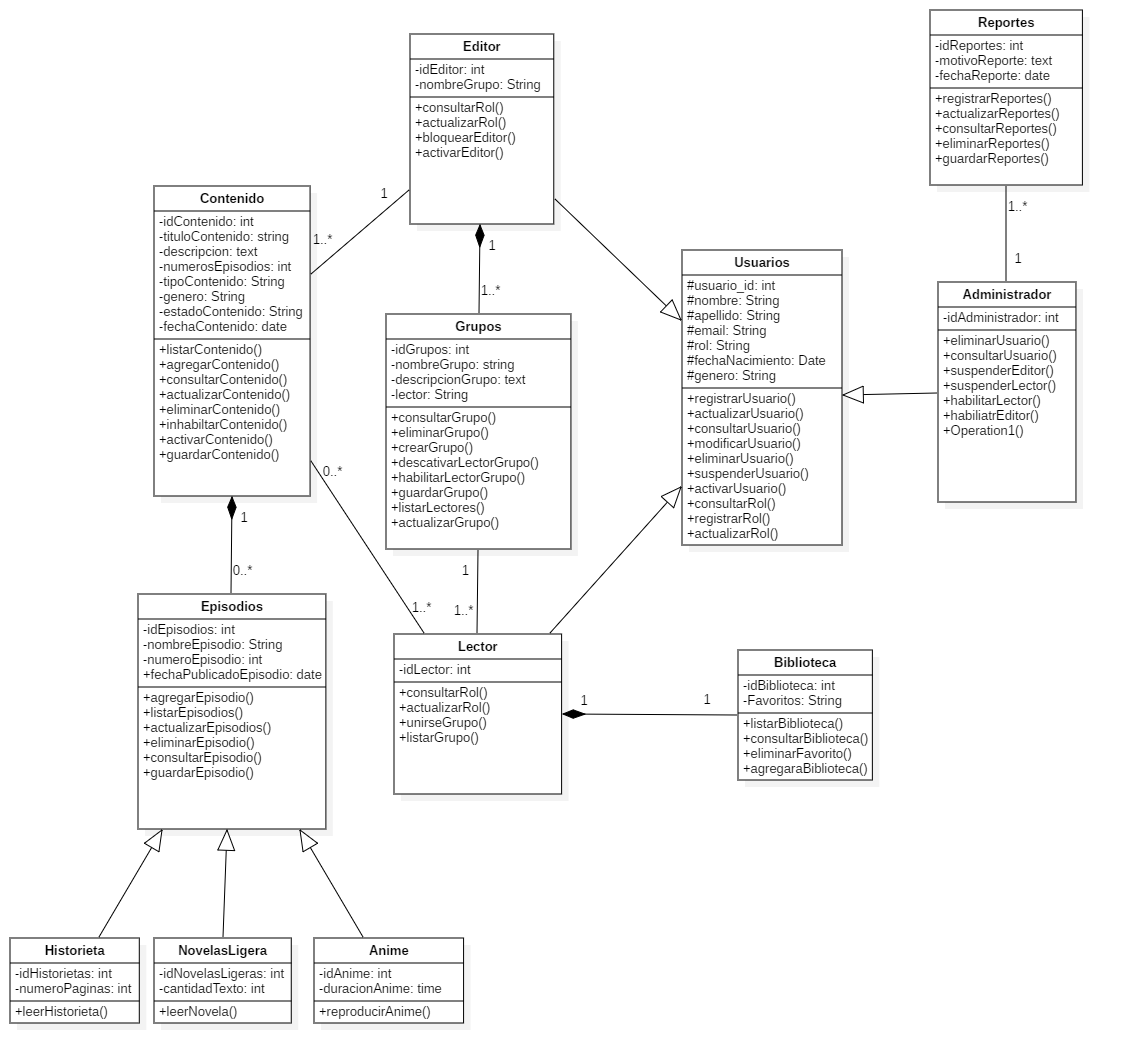
**Diagrama módulo 1: Contenido**

****

Este diagrama de actividades describe el flujo del **módulo de contenido** de la plataforma, abarcando tanto las acciones del **editor** como del **lector**, y las validaciones internas del **sistema**. El proceso inicia cuando el editor decide **agregar contenido**, tras lo cual el sistema muestra las opciones de tipo de contenido: **novela ligera**, **anime** o **historieta**. Luego, el editor selecciona un tipo y procede a **ingresar los datos**, que el sistema valida para asegurar su completitud y corrección. Si se requieren episodios, el sistema despliega una plantilla específica donde el editor continúa ingresando información por episodio. Una vez validados los datos, el contenido se **almacena**, y el sistema **confirma la subida** mediante un mensaje emergente. Posteriormente, los lectores pueden **buscar contenido**, **seleccionar** el deseado y visualizar la **lista de episodios**. Dependiendo del tipo, el lector puede **leer una novela o historieta** o **reproducir un anime**. Además, el lector puede optar por **guardar el contenido en su biblioteca**, acción que es validada por el sistema para evitar duplicados. Si no está guardado, el sistema lo añade y **muestra una notificación de confirmación**. Todo el flujo está distribuido entre tres swimlanes: **Editor**, **Sistema** y **Lector**, reflejando claramente la responsabilidad de cada actor.

# VISTA LÓGICA

## 5.1 Diagramas - Clases (Actualizado)



El diagrama de clases representa la estructura del software de gestión de contenido multimedia, organizado en tres módulos principales: **Usuarios, Contenido y Comunidad**. En el **Módulo de Usuarios**, la clase **Usuarios** es la base para los roles **Administrador, Editor y Lector**, cada uno con funciones específicas como gestionar usuarios, administrar contenido y acceder a la plataforma. Los administradores pueden suspender o habilitar usuarios, los editores pueden gestionar contenido y grupos, y los lectores pueden unirse a grupos y gestionar su biblioteca. En el **Módulo de Contenido**, la clase **Contenido** representa animes, historietas y novelas ligeras, con atributos como título, descripción, número de episodios y género. Las clases derivadas (**Historietas, Novelas Ligeras y Animes**) extienden la funcionalidad según el tipo de contenido, y los **Episodios** están vinculados al contenido mediante una relación de composición. En el **Módulo de Comunidad**, los **Grupos** permiten la interacción entre editores y lectores, mientras que la clase **Reportes** gestiona las denuncias dentro de la plataforma. La **Biblioteca**, exclusiva de los lectores, permite almacenar contenido favorito. En cuanto a las relaciones, se usa **herencia** para roles de usuario y tipos de contenido, **asociación** entre contenido y episodios, **agregación** entre grupos y lectores (un grupo puede existir sin un lector específico) y **composición** entre contenido y episodios (si se elimina el contenido, los episodios desaparecen). Este modelo estructurado facilita la escalabilidad del software, asegurando que los usuarios puedan interactuar con el contenido de forma organizada y segura.

## 5.2 Diagrama - Objetos

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

El diagrama de objetos representa una instancia del sistema de gestión de contenido, mostrando ejemplos concretos de los datos almacenados en las clases del diagrama de clases. En él, se observa a **Camila**, una editora del grupo "Sakura", quien ha registrado la serie **"Naruto"** como contenido de tipo anime, con 500 episodios y en estado "en proceso". Dentro de esta serie, se ha registrado el episodio **"¡Entra! ¡Naruto Uzumaki!"**, con fecha de publicación 24/08/2024. **Carlos**, un lector identificado con ID 1, puede visualizar este contenido. Además, el objeto **"Naruto: Anime"** representa una instancia de la clase Anime, con una duración de 22 minutos por episodio. Curiosamente, el diagrama también incluye un objeto ajeno al dominio del sistema, una dieta llamada **"Keto"**, con información sobre su duración, contenido nutricional y fechas de inicio y fin, lo que parece un error o inclusión accidental dentro del modelo del sistema.

## 5.3 Diagramas - Secuencia

Diagrama

El contenido generado por IA puede ser incorrecto.

1. El editor, a través de la Interfaz de Contenido, invoca crearContenido().
2. La clase Contenido procesa el contenido y responde con contenidoCreado().
3. Luego se llama a agregarEpisodio() y crearEpisodio() en la clase Episodio.
4. Posteriormente, se llama a asociarTipo(anime) para indicar que el contenido es de tipo anime.
5. La clase Anime registra el tipo de contenido y responde animeRegistrado().

**Objetivo:** Permitir al editor subir nuevo contenido, agregar episodios y definir si es anime, novela ligera o historieta.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. El lector inicia con consultarContenido() sobre la clase Contenido.
2. Luego realiza verEpisodios(idContenido) en Episodios.
3. Según el tipo de contenido, puede consultar:

* leerHistorieta(idHistorieta)
* verAnime(idAnime)
* leerNovelaLigera(idNovela)

**Objetivo:** Permitir que el lector explore un contenido específico y acceda a sus episodios según el tipo (anime, historieta o novela ligera).

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

# VISTA DE IMPLEMENTACIÓN

## 6.1 Diagrama de Componentes

**Diagrama

El contenido generado por IA puede ser incorrecto.**

## 6.2 Diagrama de Paquetes

**Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.**

## 6.3 VISTA DE DESPLIEGUE

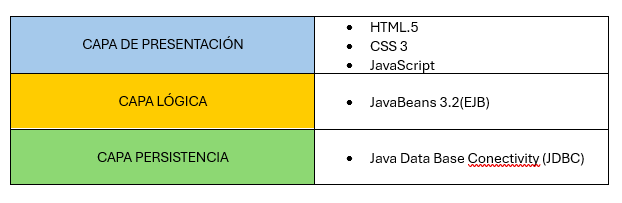
* + 1. **Diagrama de despliegue**

1. Una captura de pantalla de un celular con texto e imágenes

   El contenido generado por IA puede ser incorrecto.

# Arquitectura en capas

* Cuadro de Tecnologías por capas- versiones



# VISTA DE DATOS

## 8.1 Modelo Relacional normalizado-tercera forma normal de la base de datos (SGBD).

Diagrama

El contenido generado por IA puede ser incorrecto.

# Definición de Interfaces de Usuario

* Página principal, formularios, páginas error.
* Landing Page (Página Principal)

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* Pantallazos interfaces modulares
* Interfaz de usuario gráfica, Aplicación, Correo electrónico

  El contenido generado por IA puede ser incorrecto.
* Interfaz de usuario gráfica, Aplicación

  El contenido generado por IA puede ser incorrecto.
* Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

  El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

# Características Generales de Calidad

## 10. 1 Tamaño y performance

* Tiempo de respuesta en el acceso a la Base de Datos: 3 s
* Tiempo de respuesta de transacciones: 6s
* Espacio en disco para el cliente: 8 GB
* Espacio en disco para el servidor de Base de datos: 1 TB

## 10.2 Calidad

El software está diseñado para proporcionar una experiencia fluida y completa a los usuarios interesados en anime, manga y novelas ligeras, garantizando una gestión eficiente del contenido y la comunidad:

* **Gestión de Contenido:** Permite a los editores subir y administrar series, episodios y volúmenes con una interfaz estructurada y validaciones para evitar inconsistencias.
* **Interacción con la Comunidad:** Facilita la creación y administración de grupos por parte de editores, asegurando una dinámica segura y organizada.
* **Experiencia Personalizada:** Ofrece recomendaciones basadas en el historial de visualización/lectura y permite marcar contenido como visto o favorito.

## 10.3 Usabilidad

El sistema presenta una interfaz intuitiva y adaptable para mejorar la experiencia del usuario:

* **Registro y Perfiles:** Los usuarios pueden registrarse fácilmente, elegir su rol (lector o editor) y personalizar su perfil con foto, descripción y preferencias.
* **Menús Claros y Funcionales:** Navegación optimizada con opciones específicas para cada rol, facilitando el acceso a listas personalizadas, biblioteca y comunidad.
* **Lectura y Visualización Optimizada:** Funciones de zoom y desplazamiento fluido para capítulos, junto con opciones de personalización del tema (modo claro/oscuro).
* **Diseño Responsivo:** Accesible desde computadoras, tablets y dispositivos móviles sin pérdida de funcionalidad.

## 10.4 Eficiencia

El software está diseñado para ofrecer rapidez y estabilidad en el procesamiento de información:

* **Carga Rápida de Contenido:** Optimización para mostrar listas de series, capítulos y recomendaciones en menos de 10 segundos.
* **Búsqueda Eficiente:** Permite localizar series por nombre, género o tipo (anime, manga, manhua, manhwa o novela ligera) con resultados instantáneos.
* **Sincronización de Datos:** Registra el progreso de lectura/visualización en la nube para mantenerlo actualizado en distintos dispositivos.
* **Rendimiento Multiplataforma:** Soporta múltiples usuarios simultáneos sin afectar la velocidad o estabilidad del sistema.

## 10.5 Seguridad

El software implementa medidas de protección para garantizar la seguridad de la información y la interacción entre usuarios:

* **Autenticación Segura:** Opción de autenticación en dos pasos para mayor protección de cuentas.
* **Privacidad de Datos:** Protección de la información personal de los usuarios, asegurando que solo ellos puedan acceder a su cuenta.
* **Gestión de Roles y Permisos:** Solo los editores pueden administrar contenido y crear grupos, mientras que los lectores pueden interactuar dentro de los permisos establecidos.
* **Moderación de Contenido:** Sistema de reportes para eliminar contenido inapropiado y garantizar un entorno seguro para la comunidad.

## 10.6 Confiabilidad

El sistema mantiene la integridad y disponibilidad de la información en todo momento:

* **Alta Disponibilidad:** La plataforma se mantiene operativa el 90% del tiempo, con respaldo en caso de fallas.
* **Copias de Seguridad Diarias:** Respaldo automático de la base de datos para evitar pérdida de información.
* **Recuperación Rápida:** En caso de fallas, el sistema permite restaurar los datos en menos de una hora.

## 10.7 Mantenimiento

El software es escalable y adaptable a cambios futuros sin afectar su funcionamiento:

* **Actualización Continua:** Permite agregar nuevas funciones (como filtros de búsqueda avanzados) sin afectar el rendimiento.
* **Soporte Técnico y Mejora Constante:** Se pueden implementar optimizaciones en la interfaz y estructura sin interrupciones en el servicio.
* **Modularidad:** Posibilidad de expandir la plataforma con nuevos módulos para contenido, interacción social o funcionalidades premium.

## 10.8 Estándares

El desarrollo del software sigue buenas prácticas y normativas para garantizar calidad y compatibilidad:

* **Código Limpio y Documentado:** Estructuración clara para facilitar el mantenimiento y futuras mejoras.
* **Cumplimiento de Normas Web:** Uso de tecnologías como HTML5, CSS3 y frameworks modernos para asegurar una experiencia fluida en distintos dispositivos y navegadores.
* **Compatibilidad Multiplataforma:** Soporte para computadoras, dispositivos móviles y tablets con diferentes resoluciones de pantalla.
* **Seguridad y Privacidad:** Cumple con regulaciones de protección de datos y estándares de accesibilidad.

Normas de calidad, mencionar dos normas de calidad de Desarrollo de Software y argumentar cómo esas normas se aplican para el software.