

Desarrollo de una aplicación web de lectura digital de Mangas

Eyder Huayta Tantavilca

Servicio Nacional de Adiestramiento en el Trabajo Industrial

Nota del autor

No se ha recibido ninguna beca o ayuda financiera para el desarrollo de este proyecto.

### Resumen

La aplicación web en desarrollo está diseñada para facilitar la consulta y gestión de información sobre cines, abordando la falta de plataformas accesibles y fáciles de usar en el mercado. La aplicación permite a los usuarios consultar horarios, funciones y detalles de películas de una forma sencilla. Actualmente, la aplicación está destinada a cualquier persona con acceso a internet y un navegador web.

## Inicio

Resumen.....	2
1. Requisitos del Sistema .....	5
1.1 Requisitos Funcionales: .....	5
1.2 Requisitos no Funcionales: .....	5
1.3 Requisitos Técnicos: .....	5
1.4 Requisitos de Plataforma .....	6
1.5 Diseño de la Aplicación .....	6
1.5.1 Home .....	6
1.5.2 Sección de Mangas Recientes .....	6
1.5.3 Formulario de Inicio de Sesión .....	<b>¡Error! Marcador no definido.</b>
1.5.4 Formulario de Registro .....	7
1.5.5 Lectura del manga .....	<b>¡Error! Marcador no definido.</b>
2. Plan de Desarrollo.....	9
2.1 Fases del Proyecto.....	9
2.2 Herramientas de Gestión.....	9
3. Implementación.....	9
3.1 Tecnologías Utilizadas .....	9
3.2 Configuración del Entorno de Desarrollo .....	9
4. Pruebas.....	10
4.1 Estrategia de Pruebas .....	10
4.2 Casos de Prueba .....	10

5.	Manual de Usuario.....	10
5.1	Guía de Uso.....	10
5.2	Preguntas Frecuentes (FAQ) .....	11
6.	Desarrollo de Funcionalidades.....	11
6.1	Clases Principales .....	11
6.2	Flujo de la Aplicación .....	12
7.	Conclusiones .....	13
7.1	Resultados Esperados.....	13
7.2	Futuras Mejoras .....	13
8.	Referencias.....	14
8.1	Código Fuente.....	14

## 1. Requisitos del Sistema

### 1.1 Requisitos Funcionales:

- Los usuarios podrán registrarse e iniciar sesión para acceder a funcionalidades personalizadas.
- Los usuarios administradores podrán gestionar las películas y funciones, mientras que los usuarios regulares solo podrán consultar la información.  
Los usuarios podrán consultar la lista de películas en cartelera, con detalles como el título, sinopsis, género, duración, calificación y horarios de función.
- Los usuarios podrán filtrar películas por género, fecha de estreno, o por horario.
- Los usuarios podrán consultar los horarios de funciones para cada película en cines específicos.

### 1.2 Requisitos no Funcionales:

- La interfaz debe ser intuitiva y fácil de navegar para los usuarios de todas las edades, especialmente para aquellos con poca experiencia tecnológica.
- La aplicación debe cargar rápidamente, con tiempos de respuesta menores a 2 segundos para todas las consultas de información de películas y funciones.
- El sistema debe ser capaz de manejar un alto volumen de usuarios concurrentes sin disminuir su rendimiento.
- Las imágenes y el contenido deben cargarse en menos de 2 segundos bajo buenas condiciones de red.
- La aplicación debe ser accesible a través de cualquier navegador web moderno (Chrome, Firefox, Edge, Safari).

### 1.3 Requisitos Técnicos:

- **Lenguajes de Programación:**
  - Java con Spring Boot para el desarrollo backend.
  - React y CSS para el diseño de la interfaz de usuario.
- **Herramientas de Desarrollo:**
  - **Visual Studio Code:** Entorno de desarrollo integrado (IDE) para crear aplicaciones Android.
  - **XAMPP:** Herramienta que proporciona un entorno de servidor local.
  - **MySQL:** Sistema de gestión de bases de datos relacional.
  - **Spring Boot:** Marco de desarrollo en Java.

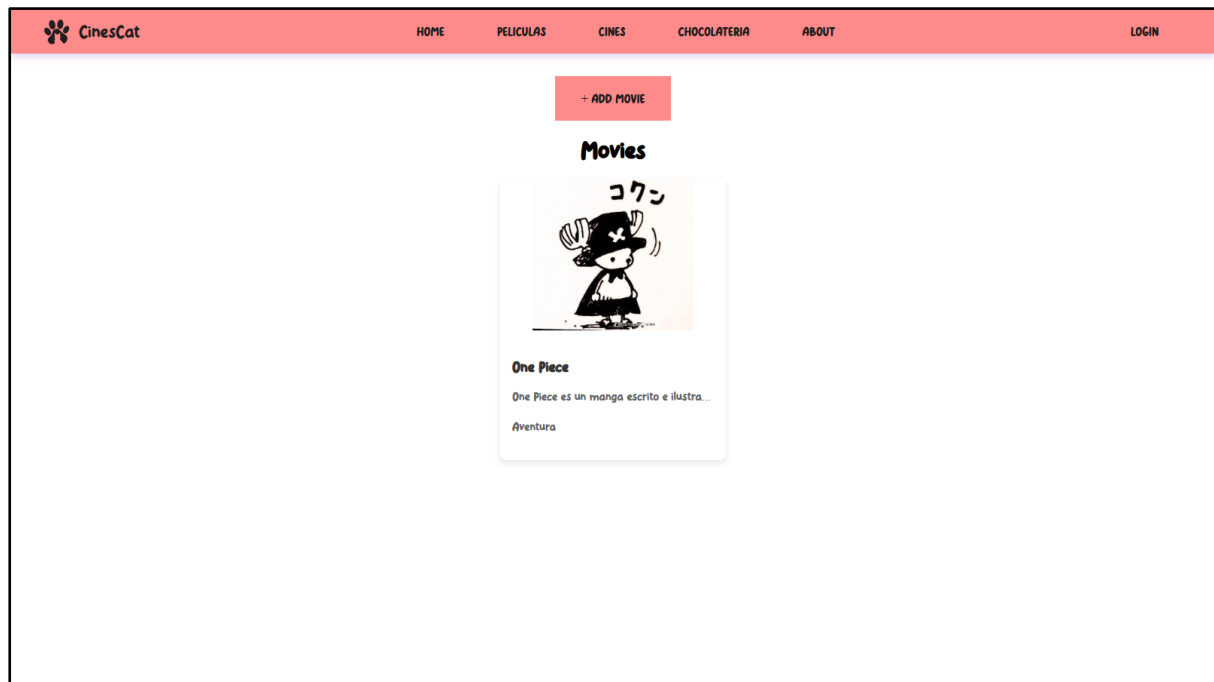
## 1.4 Requisitos de Plataforma

### - Navegadores Soportados:

- Google Chrome (última versión y versiones anteriores de los últimos dos años).
- Mozilla Firefox (última versión y versiones anteriores de los últimos dos años).
- Microsoft Edge (última versión).
- Safari (última versión en macOS).

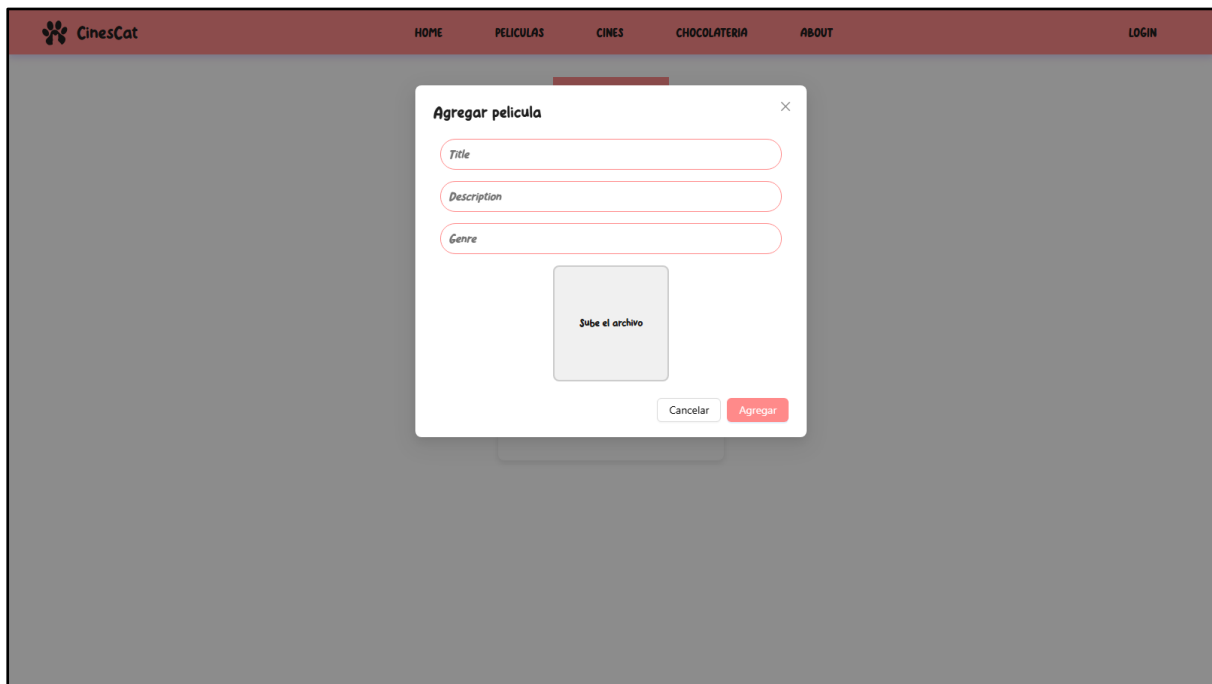
## 1.5 Diseño de la Aplicación

### 1.5.1 Home



Fuente: [GitHub](#)

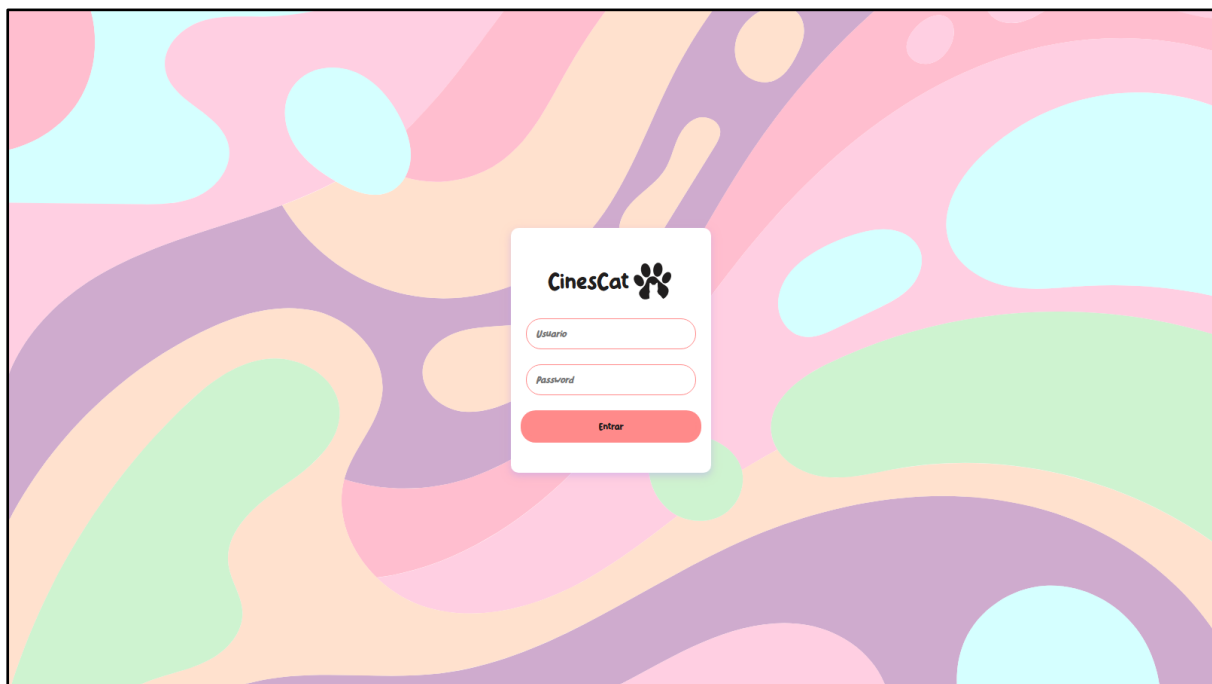
### 1.5.2 Modal - Añadir película



The screenshot shows a web application interface with a dark red header. The header contains the CinesCat logo on the left and navigation links (HOME, PELICULAS, CINES, CHOCOLATERIA, ABOUT, LOGIN) on the right. A modal window titled 'Agregar película' is centered on the screen. The modal has a close button (X) in the top right corner. It contains three input fields labeled 'Title', 'Description', and 'Genre'. Below these fields is a large gray button labeled 'Sube el archivo'. At the bottom of the modal are two buttons: 'Cancelar' and 'Agregar'.

*Fuente: [GitHub](#)*

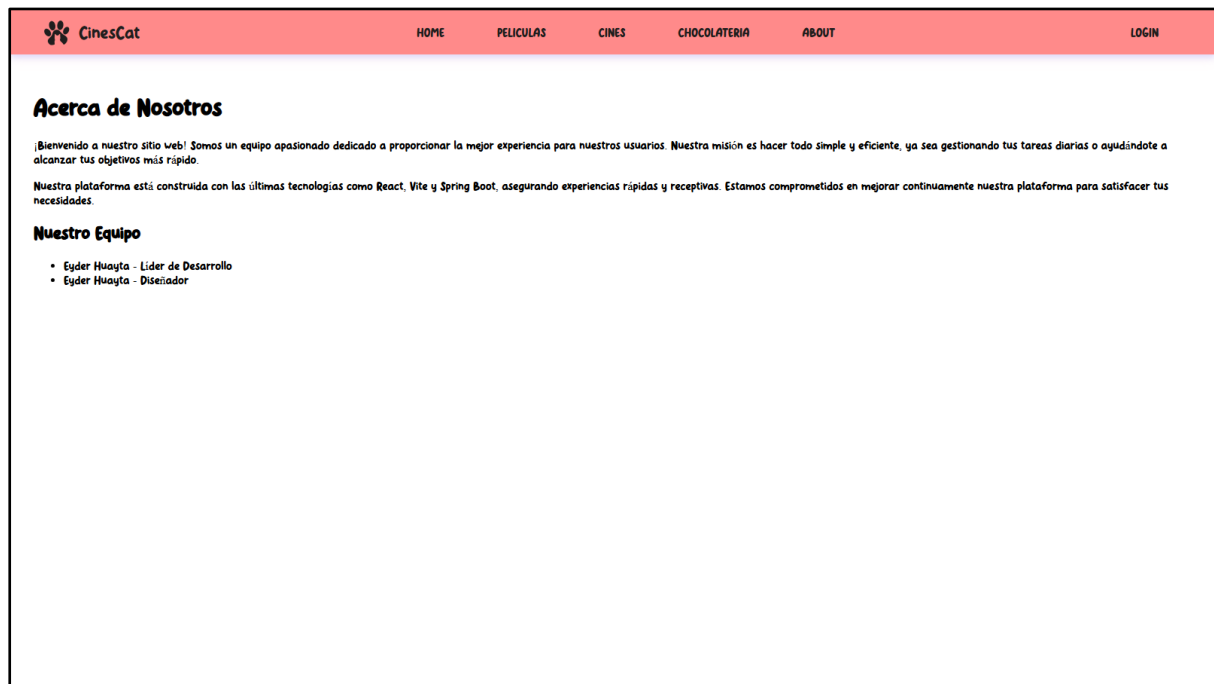
### 1.5.1 Login



The screenshot shows a login form centered on a background with a colorful, abstract pattern of pink, purple, orange, and green shapes. The login form is a white rectangle with the CinesCat logo at the top. It contains two input fields labeled 'Usuario' and 'Password'. Below these fields is a red button labeled 'Entrar'.

*Fuente: [GitHub](#)*

### 1.5.2 About



*Fuente: [GitHub](#)*



## 2. Plan de Desarrollo

### 2.1 Fases del Proyecto

- **Planificación:** Definición de objetivos y requisitos.
- **Diseño:** Estructuración de las páginas utilizando HTML, CSS y JavaScript (React).
- **Desarrollo:** Programación e implementación de las funcionalidades.
- **Pruebas:** Verificación de la aplicación web y corrección de errores.
- **Despliegue:** Implementación y lanzamiento de la aplicación en un servidor web, asegurando su accesibilidad a través de navegadores.

### 2.2 Herramientas de Gestión

- GitHub (<https://github.com/EL Eyder/CinesCat>) para control de versiones y colaboración.

## 3. Implementación

### 3.1 Tecnologías Utilizadas

- **Visual Studio:** Entorno de desarrollo para la programación en C#.

### 3.2 Configuración del Entorno de Desarrollo

- Instalar Java JDK 21: Instalar la versión más reciente de Java JDK.
- Instalar Visual Studio Code: Asegúrate de tener instalada la última versión de Visual Studio Code.
- Instalar Node.js y npm: Asegúrate de tener instalados Node.js y npm (gestor de paquetes de Node) para poder trabajar con React.
- Clonar el repositorio GitHub: Utiliza Git para clonar el repositorio del proyecto desde GitHub a tu máquina local.
- Configurar la base de datos: En el proyecto Spring Boot, configura la conexión a la base de datos en el archivo application.properties. (Asegúrate de tener MySQL (o el sistema de base de datos que estés usando) en funcionamiento).
- Instalar dependencias: Abre una terminal en la carpeta del frontend del proyecto (React) y ejecuta npm install
- Iniciar el servidor local:

- **Backend (Spring Boot):** Desde el IDE (como IntelliJ IDEA o Eclipse), ejecuta el proyecto **Spring Boot** en modo desarrollo. Esto iniciará un servidor local en el puerto predeterminado (por defecto `http://localhost:8080`).
- **Frontend (React):** Abre una terminal en la carpeta del frontend y ejecuta `npm run dev`
- Acceder a la aplicación: Accede al backend de la aplicación en el navegador en `http://localhost:8080` y al frontend en `http://localhost:3000`. Ambos deben estar corriendo simultáneamente.
- Probar funcionalidades: Realiza pruebas en las distintas funcionalidades de la aplicación para asegurarte de que todo funcione como se espera.

## 4. Pruebas

### 4.1 Estrategia de Pruebas

- Pruebas Unitarias: Verificación de cada módulo de la aplicación.
- Pruebas de Integración: Asegurar que los componentes funcionen bien juntos.
- Pruebas de Aceptación: Validar que la aplicación cumple con los requisitos del usuario.

### 4.2 Casos de Prueba

- Caso de Prueba 1: Validar el funcionamiento del sistema de inicio de sesión.
- Caso de Prueba 2: Verificar el funcionamiento de la conexión a la base de datos.
- Caso de Prueba 3: Controlar las respuestas que devuelve el servidor.

## 5. Manual de Usuario

### 5.1 Guía de Uso

- **Ingresa a la url de la aplicación web:**
  - Ingresa al enlace de la web oficial.
- **Primer Inicio:**
  - Al abrir la aplicación por primera vez, se te presentará unos banners de bienvenida. Visualiza la información proporcionada para familiarizarte con las funciones principales de la aplicación.

Si ya tienes una cuenta, ingresa tus credenciales para iniciar sesión, si no, regístrate en el formulario correspondiente

- **Seleccionar la película a visualizar:**
  - En la pantalla principal, verás una lista de las películas más recientes.
  - Toca la película deseada para empezar a leer la información de la misma.
- **Navegar por el Contenido:**
  - Utiliza el Header para visualizar las distintas páginas.

## 5.2 Preguntas Frecuentes (FAQ)

- ¿Es necesario el inicio sesión?: No es necesario tener una cuenta para usar la aplicación, esto debido a que se busca la lectura rápida y menos restrictiva.
- ¿Cómo reporto un problema?: Se pueden reportar los problemas que se encuentran en el repositorio de GitHub.
- ¿Por qué no cargan las películas?: Esto se puede deber a 2 razones principales, la primera es que la API no esté activa al momento de usar la aplicación lo cual es lo mas probable al ser una API que se aloja de manera local, y la segunda causa es por la conexión lenta de internet.

## 6. Desarrollo de Funcionalidades

### 6.1 Clases Principales

- **CinescatApplication**

Clase principal de la aplicación que inicializa el entorno de ejecución de Spring Boot. Esta clase contiene el método main, que ejecuta el método run de SpringApplication, iniciando el servidor de la aplicación.
- **User**

Clase que representa la entidad de usuario en la base de datos. Contiene atributos como id, username, y password, que almacenan la información de cada usuario. Esta clase se mapea a la tabla users en la base de datos, utilizando anotaciones de JPA.

- **Movie**

Clase que representa la entidad de película en la base de datos. Incluye atributos como id, title, description, genre e image. Estos campos almacenan la información de cada película y permiten la gestión de los datos en la base de datos. La clase también proporciona el método getImageUrl para obtener la URL completa de una imagen.

- **MovieController**

Clase que maneja las solicitudes HTTP relacionadas con la gestión de películas. Permite recuperar la lista de películas (getMovies) y agregar nuevas películas (addMovie) con detalles como título, descripción, género e imagen. Incluye el método saveImage, que guarda las imágenes de las películas en un directorio local y gestiona posibles errores de subida.

- **UserController**

Clase que gestiona las solicitudes relacionadas con los usuarios. Ofrece métodos para obtener la lista de usuarios (getUsers) y para crear un nuevo usuario (userCreate). Esta clase maneja las interacciones con el repositorio IUserRepo, que conecta la aplicación con la base de datos.

## 6.2 Flujo de la Aplicación

- **Inicio de la Aplicación:** Al iniciar la aplicación, el usuario se presenta en la vista inicial (Home.jsx), donde se muestra un botón de "Login" en la barra de navegación.
- **Navegación a la Página de Inicio de Sesión:** El usuario hace clic en el botón "Login" en la barra de navegación, lo que redirige a la vista del formulario de inicio de sesión (Login.jsx).
- **Ingreso de Credenciales:** En la página de inicio de sesión, el usuario ingresa su usuario y contraseña.
- **Autenticación:** Al enviar el formulario, valida las credenciales ingresadas.

- **Verificación de Usuario:** El sistema consulta la base de datos utilizando para verificar si el usuario existe y si las credenciales son correctas.
- **Redirección al Menú Principal:** Si las credenciales son válidas, el usuario es redirigido a la página principal (Home.jsx) donde se muestra el contenido de la aplicación.
- **Visualización de Contenido:** En la página principal, el usuario ve las películas agregadas recientemente.
- **Interacción con el Contenido:** El usuario selecciona una película para visualizar, lo que lo lleva a la vista de los detalles de la película.
- **Manejo de Errores:** Si ocurre un error durante la autenticación o en la aplicación, el usuario es redirigido a la vista de error (Error ()) para mostrar el mensaje correspondiente.

## 7. Conclusiones

### 7.1 Resultados Esperados

Se espera que la aplicación facilite el acceso a la información de películas nuevas, así como su fecha de salida y también su visualización si está disponible. La aplicación debe ser sencilla, gratuita y ofrecer una experiencia de usuario agradable.

### 7.2 Futuras Mejoras

- **Expansión de Contenidos:** Incluir más categorías y tipos de contenido.
- **Buscador Integrado:** Implementar un buscador para encontrar el contenido deseado.
- **Sección de comentarios:** Implementar una sección de comentarios para mostrar una opinión de la película.
- **Funcionalidades Adicionales:** Implementar características como recomendaciones personalizadas y sincronización entre dispositivos.

## **8. Referencias**

### **8.1 Código Fuente**

- [CinesCat \(GitHub\)](#)