**Memoria del Proyecto: Catálogo de Animes**

**1. Introducción**

**1.1 Objetivo del Proyecto**

Este proyecto tiene como objetivo desarrollar una plataforma web para gestionar un catálogo de animes, permitiendo a los usuarios consultar información sobre los animes, agregar animes a su lista de favoritos y enviar solicitudes para agregar nuevos animes. Los administradores tendrán acceso a un panel de gestión para añadir, modificar y eliminar animes del catálogo, así como gestionar las solicitudes enviadas por los usuarios y otorgar privilegios a otros usuarios.

**1.2 Justificación**

En el contexto actual, existen varias plataformas para gestionar animes, como MyAnimeList o AniList. Sin embargo, estas plataformas no siempre ofrecen la posibilidad de solicitar animes para añadir a la base de datos ni un sistema de gestión de privilegios entre los usuarios. Este proyecto busca cubrir esa necesidad mediante una plataforma personalizada, ofreciendo una interacción más directa entre usuarios y administradores.

**1.3 Alcance**

* **Usuarios:**
  + Consultar el catálogo de animes.
  + Añadir animes a su lista de favoritos.
  + Enviar solicitudes para añadir animes nuevos al catálogo.
* **Administradores:**
  + Gestionar el catálogo (añadir, modificar, eliminar animes).
  + Gestionar las solicitudes de animes realizadas por los usuarios.
  + Asignar privilegios de administrador a otros usuarios.

**2. Análisis y Planificación**

**2.1 Estudio Previo del Problema o Necesidad**

Los usuarios necesitan una manera organizada y fácil de consultar animes, agregar los que les gustan a una lista de favoritos y poder sugerir animes nuevos. Además, los administradores necesitan herramientas para gestionar el catálogo y las solicitudes de los usuarios, así como controlar los privilegios de cada usuario.

**2.2 Requisitos Funcionales**

* **Usuarios:**
  + Registro y autenticación de usuarios.
  + Consultar animes en el catálogo.
  + Agregar animes a la lista de favoritos.
  + Enviar solicitudes para agregar nuevos animes.
* **Administradores:**
  + Añadir, modificar y eliminar animes del catálogo.
  + Gestionar las solicitudes de animes de los usuarios (aprobar o rechazar).
  + Asignar privilegios de administrador a otros usuarios.

**2.3 Requisitos No Funcionales**

* Interfaz responsive para adaptación a dispositivos móviles.
* Seguridad en la gestión de contraseñas y datos de usuarios (uso de sesiones y protección de datos).
* Optimización de rendimiento para tiempos de carga rápidos.
* Accesibilidad para personas con discapacidades visuales.

**2.4 Tecnologías Utilizadas**

* **Frontend:**
  + **HTML5, CSS3, JavaScript:** Utilizados para la creación de la interfaz de usuario, ofreciendo una experiencia de navegación intuitiva.
* **Backend:**
  + **PHP:** Lenguaje utilizado para gestionar la lógica de negocio, la autenticación de usuarios y la interacción con la base de datos. PHP es el encargado de procesar las solicitudes de los usuarios, gestionar las sesiones, y ejecutar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en el catálogo de animes.
  + **MySQL:** Sistema de gestión de bases de datos utilizado para almacenar toda la información de la plataforma, como los datos de los usuarios, los animes, las solicitudes, y los favoritos.
* **Autenticación y Seguridad:**
  + **PHP y sesiones:** PHP se encarga de gestionar las sesiones de usuario, asegurando que solo los usuarios autenticados puedan acceder a funciones específicas.
  + **Protección contra inyecciones SQL:** Uso de consultas preparadas para prevenir ataques de inyección SQL y proteger los datos.

**3. Diseño y Arquitectura**

**3.1 Diagrama de Casos de Uso**

**Actores:**

* **Usuario:** Puede consultar animes, añadirlos a favoritos, y enviar solicitudes de animes.
* **Administrador:** Puede gestionar el catálogo de animes, gestionar solicitudes de usuarios, y asignar privilegios de administrador.

(Incluir un diagrama UML de Casos de Uso aquí, mostrando cómo interactúan los actores con el sistema).

**3.2 Diagrama de Clases (Simplificado)**

* **Usuario:** id, nombre, correo, contraseña, rol (usuario/admin).
* **Anime:** id, título, género, año de estreno, sinopsis.
* **Favorito:** id\_usuario, id\_anime.
* **Solicitud:** id\_usuario, anime\_solicitado, estado (pendiente, aprobado, rechazado).
* **Administrador:** id\_usuario (con privilegios para gestionar los animes y usuarios).

(Incluir un diagrama UML de Clases aquí).

**3.3 Diagrama de Flujo del Sistema**

**Flujo de registro de usuario:**

1. El usuario ingresa los datos en el formulario de registro.
2. Se valida que el correo no esté registrado.
3. Si los datos son válidos, se guarda el usuario en la base de datos.
4. Se envía un correo de confirmación y el usuario puede acceder a su cuenta.

(Incluir un diagrama de flujo aquí).

**4. Desarrollo e Implementación**

**4.1 Desarrollo Backend con PHP y MySQL**

El backend se desarrolló utilizando **PHP** como lenguaje principal para gestionar las solicitudes de los usuarios y administrar el catálogo de animes. PHP se encargó de recibir y procesar las solicitudes HTTP, así como de manejar la lógica de negocio del sistema (registro, autenticación, operaciones CRUD sobre animes y solicitudes).

**Funcionalidades principales implementadas en PHP:**

* **Registro y autenticación de usuarios:** Los usuarios se registran y autentican a través de un sistema de sesiones.
* **Operaciones CRUD de animes:** Los administradores pueden agregar, modificar y eliminar animes desde el panel de administración.
* **Gestión de solicitudes:** Los usuarios pueden enviar solicitudes para añadir nuevos animes, y los administradores pueden aprobar o rechazar estas solicitudes.

**4.2 Base de Datos MySQL**

La base de datos **MySQL** se utilizó para almacenar la información del proyecto. Las tablas clave incluyen:

* **Usuarios:** Almacena información de los usuarios, como nombre, correo electrónico, contraseña (cifrada), y el rol (usuario o administrador).
* **Animes:** Contiene los datos de los animes (título, género, sinopsis, año de estreno).
* **Favoritos:** Una relación entre usuarios y animes para almacenar los animes que un usuario ha marcado como favoritos.
* **Solicitudes:** Los usuarios pueden solicitar nuevos animes para agregar al catálogo, y estas solicitudes se gestionan en esta tabla.

**4.3 Seguridad**

Se implementó un sistema de autenticación utilizando **sesiones PHP**, lo que garantiza que los usuarios solo puedan acceder a las funcionalidades protegidas si han iniciado sesión correctamente. Además, se utilizó **consultas preparadas** en PHP para prevenir inyecciones SQL y garantizar la seguridad de los datos.

**5. Pruebas**

**5.1 Pruebas Unitarias**

Se realizaron pruebas unitarias sobre las funciones clave del sistema, como el registro de usuarios, la autenticación, y las operaciones CRUD de animes. Estas pruebas aseguran que cada módulo funcione correctamente de manera independiente.

**5.2 Pruebas de Integración**

Se verificó la correcta integración entre el frontend y el backend, asegurando que los datos se recuperen correctamente de la base de datos y que las interacciones entre los usuarios y el servidor se manejen de manera eficiente.

**5.3 Pruebas de Usabilidad**

La interfaz de usuario fue probada con un grupo pequeño de usuarios para verificar la facilidad de uso y la claridad de la navegación. También se hicieron ajustes basados en la retroalimentación obtenida.

**6. Conclusiones y Futuras Mejoras**

**6.1 Logros**

El proyecto cumple con los objetivos establecidos, proporcionando una plataforma funcional donde los usuarios pueden consultar animes, agregar favoritos y sugerir nuevos animes, y los administradores tienen control sobre el catálogo y los permisos de los usuarios.

**6.2 Lecciones Aprendidas**

Durante el desarrollo, aprendí a manejar adecuadamente las sesiones en PHP, la gestión de bases de datos con MySQL, y la implementación de sistemas de autenticación y autorización.

**6.3 Futuras Mejoras**

* Añadir un sistema de valoraciones y comentarios para los animes.
* Implementar un sistema de recomendaciones personalizadas basado en los animes favoritos de los usuarios.
* Mejorar el diseño de la interfaz para hacerlo más accesible y atractivo.

**7. Documentación Adicional**

* **Código fuente:** El código completo del proyecto se encuentra disponible en [enlace al repositorio en GitHub].
* **Instrucciones de instalación:** Para desplegar el proyecto en un entorno local o en producción, sigue estos pasos:
  1. Clona el repositorio.
  2. Configura la base de datos MySQL.
  3. Realiza el ajuste de las credenciales de la base de datos en el archivo de configuración.
  4. Ejecuta el servidor PHP para probar la aplicación.