

Memoria del Proyecto: Anime Den

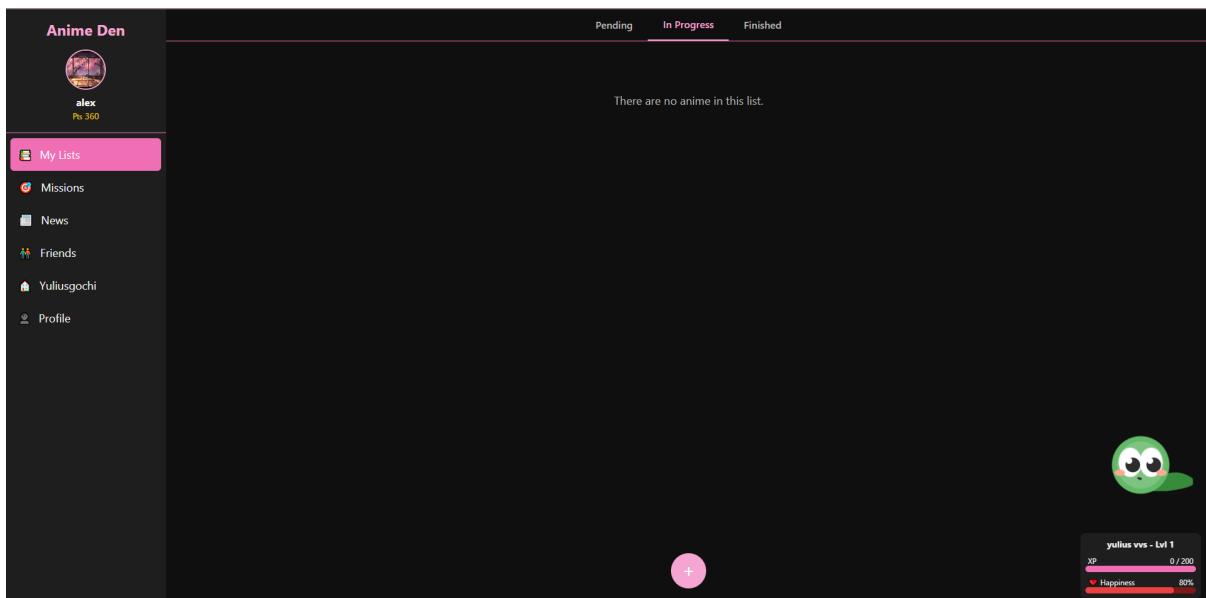
Alejandro Alarcón López

Mario Cárcel Navarro

1. Introducción y Descripción General

Anime Den es una aplicación web *Full Stack* diseñada para transformar la experiencia de seguimiento de animes. A diferencia de las listas tradicionales, este proyecto gamifica el hábito de ver series mediante la integración de una mascota virtual ("Yulusguchi") que evoluciona y crece conforme el usuario completa animes y misiones.

El objetivo principal ha sido crear un ecosistema donde la productividad del usuario (ver y registrar animes) se traduzca directamente en recompensas visuales y progreso dentro de la aplicación.



1.1. Fase de Diseño y Prototipado

Antes de escribir una sola línea de código, realizamos una fase de planificación visual. Se crearon **Mockups** iniciales para definir la distribución de los elementos, la paleta de colores (Naranja/Rosa) y la experiencia de usuario (UX). Esto nos permitió tener una guía clara de qué componentes programar.



2. Tecnologías Usadas (Tech Stack)

Para el desarrollo de esta aplicación se ha optado por una arquitectura moderna basada en componentes y microservicios contenedrizados.

Frontend (Cliente)

- **React + Vite**: Utilizado para construir una interfaz de usuario rápida y reactiva. Vite se eligió por su velocidad de compilación superior a Create-React-App.
- **TypeScript**: Implementado para añadir tipado estático, reduciendo errores en tiempo de desarrollo y mejorando la calidad del código.
- **Tailwind CSS**: Framework de estilos "utility-first" que ha permitido diseñar una interfaz atractiva sin escribir hojas de estilo CSS tradicionales.
- **Lucide React**: Para la iconografía de la interfaz.

Backend (Servidor)

- **Node.js + Express**: Servidor ligero encargado de gestionar la API RESTful. Maneja las peticiones del frontend y la lógica de conexión con la base de datos.
- **MySQL (driver mysql2)**: Base de datos relacional utilizada para la persistencia de usuarios y sus listas de animes.
- **CORS**: Middleware para gestionar la seguridad y permitir la comunicación entre el puerto del frontend (5173) y el backend (3000).

Infraestructura y Herramientas

- **Docker & Docker Compose**: Utilizado para encapsular la base de datos MySQL y el gestor phpMyAdmin, garantizando que el entorno de desarrollo sea idéntico en cualquier máquina.
- **phpMyAdmin**: Herramienta de administración visual para gestionar la base de datos y verificar registros manualmente.

□	Name	Container ID	Image	Port(s)	Actions
□	● proyectointermodular	-	-	-	■ : └
□	● animelist-frontend	443b6337d25e	proyectointermodular-fronte	8080:80 ↗	■ : └
□	● animelist-backend	8a6f6c96bf32	proyectointermodular-backe	3000:3000 ↗	■ : └
□	● animelist-pma	1c2097305bef	phpmyadmin/phpmyadmin	8081:80 ↗	■ : └
□	● animelist-db	0f27cbd230f4	mysql:8.0	3306:3306 ↗	■ : └

3. Inteligencia Artificial en el Desarrollo

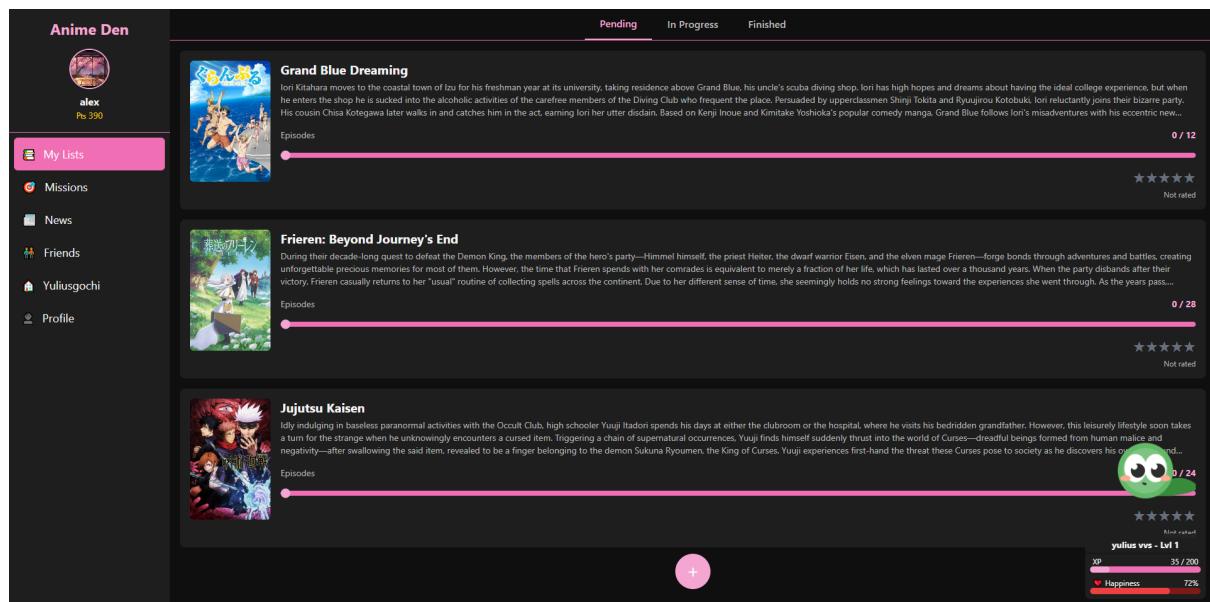
El desarrollo de Anime Den ha sido potenciado mediante el uso de herramientas avanzadas de IA, específicamente **IA Studio** y modelos generativos como **Gemini**.

- **IA Studio como Entorno de Ayuda:** Utilizamos IA Studio para refinar prompts, estructurar la lógica compleja y generar explicaciones detalladas del código.
- **Generación de Código Boilerplate:** La IA ayudó a generar la estructura inicial de los componentes de React y la configuración del servidor Express.
- **Depuración (Debugging):** Fundamental para resolver errores complejos de conexión (CORS, fallos de SQL) y problemas de lógica en el renderizado de listas.
- **Optimización de Consultas SQL:** Creación de scripts para la inserción de usuarios y tablas relacionales.

4. Funcionalidades Principales

4.1. Gestión de Listas de Anime

El núcleo de la aplicación. Permite buscar animes, añadirlos a listas personalizadas (Pendiente, En Curso, Finalizado) y actualizarlos en tiempo real.



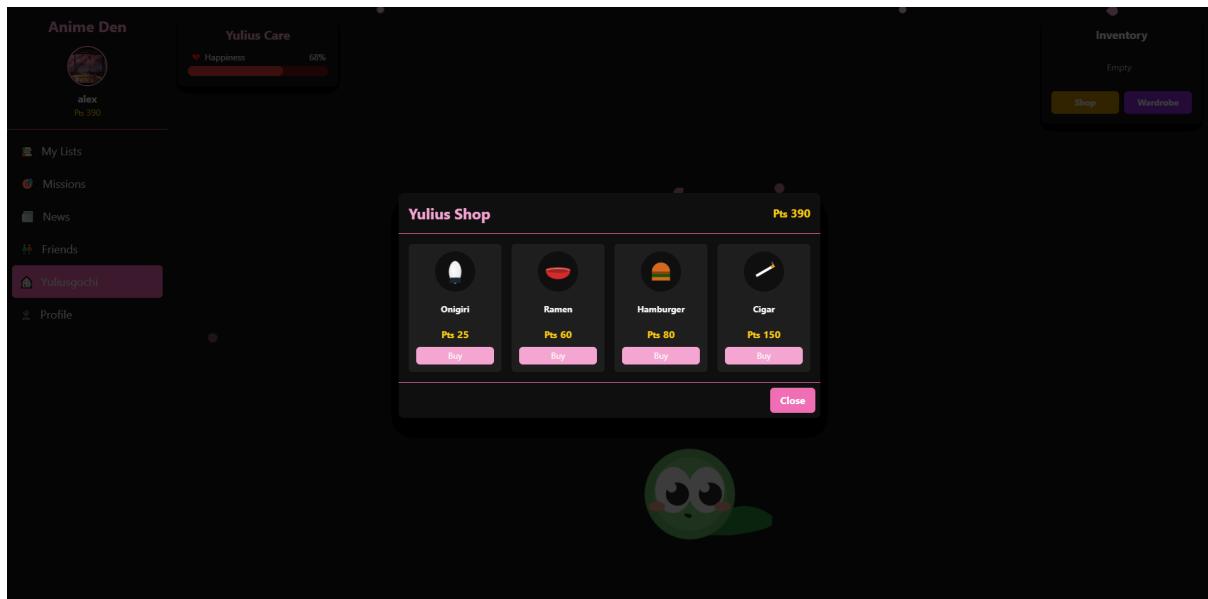
4.2. Sistema Yulusgochi (Mascota)

Una mascota interactiva que reacciona al estado del usuario.

- **Evolución:** Cambia de apariencia (Renacuajo -> Ninja) según el nivel.
- **Felicidad:** Sistema que decrece con el tiempo y requiere interacción.
- **Animaciones CSS:** Estados de ánimo visuales (rebote feliz, temblor triste).

4.3. Economía y Tienda

Los usuarios ganan "Pesetas" completando misiones y viendo capítulos. Estas monedas se canjean en una tienda virtual por comida (para subir felicidad) o ropa (cosméticos).



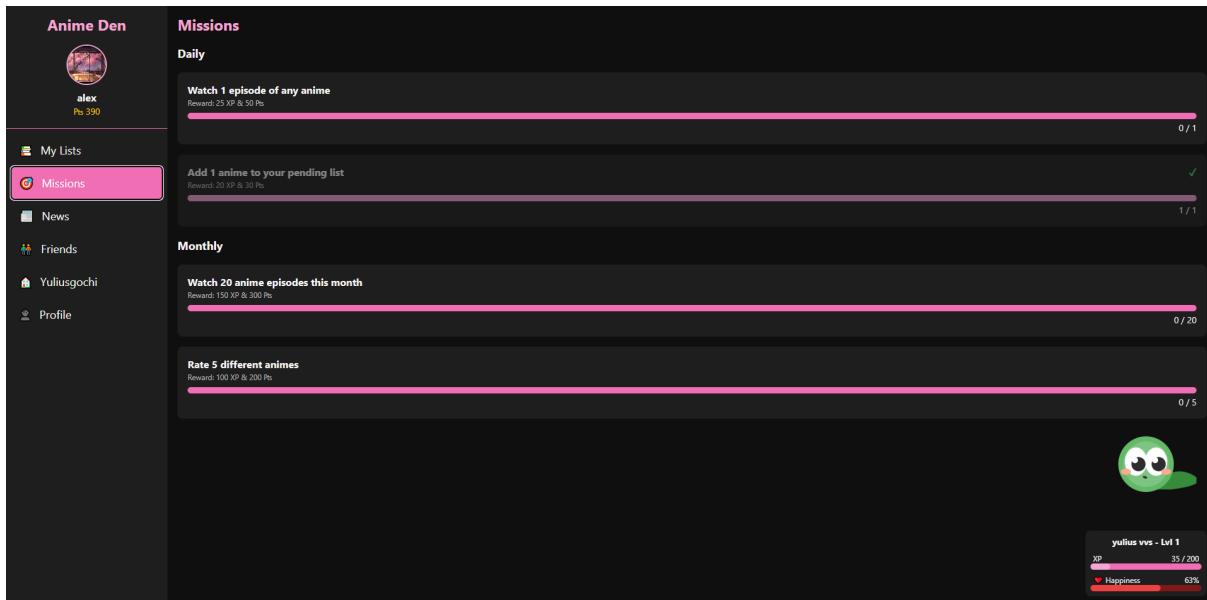
4.5. Sistema de Misiones y Recompensas

Para fomentar la retención y el uso diario de la aplicación, se ha implementado un sistema de **Misiones** gamificadas. Estas actúan como el motor económico del juego, incentivando al usuario a realizar acciones de productividad (ver y registrar animes) a cambio de beneficios virtuales.

El sistema se divide en dos categorías temporales:

- **Misiones Diarias:** Objetivos a corto plazo diseñados para completarse rápidamente (ej: "Ver 1 capítulo de cualquier anime" o "Añadir 1 anime a pendientes").
- **Misiones Mensuales:** Metas de mayor envergadura que requieren constancia (ej: "Ver 20 capítulos este mes" o "Calificar 5 animes diferentes").

Cada misión cuenta con una **barra de progreso visual** que se actualiza en tiempo real. Al completarse, el usuario recibe una doble recompensa: **XP** (Experiencia para subir de nivel a la mascota) y **Pesetas/Puntos** (Moneda para comprar ítems).

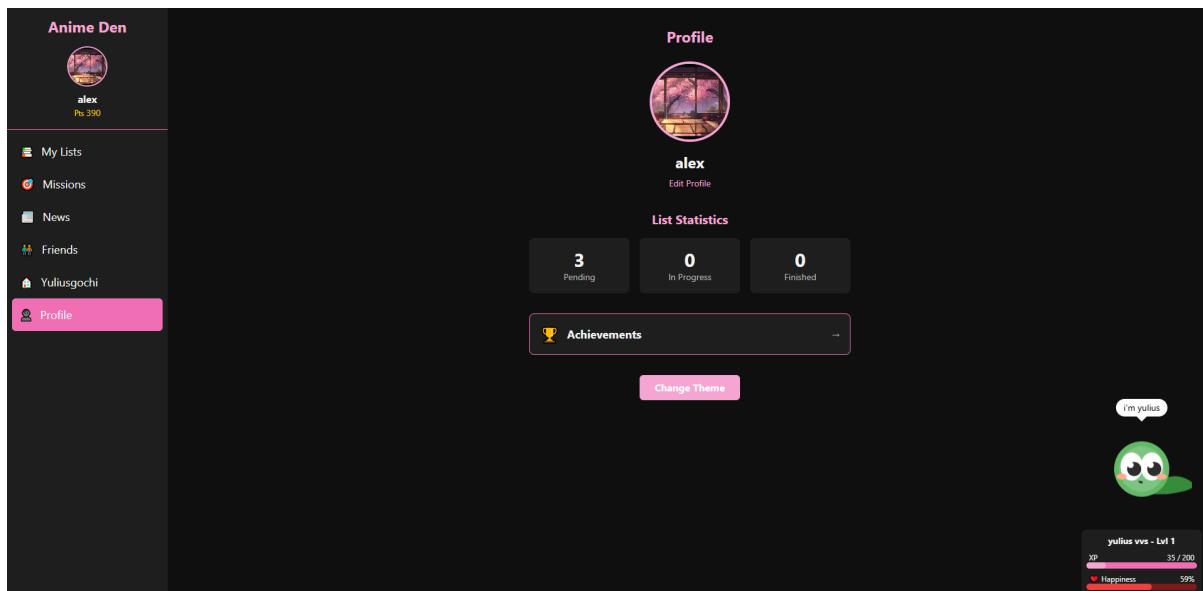


4.6. Perfil de Usuario y Personalización

El perfil es el centro de mando donde el usuario gestiona su identidad y visualiza su progreso global en la plataforma. Esta sección no solo muestra información estática, sino que integra funcionalidades de configuración y estadísticas.

Las características clave incluyen:

- **Estadísticas de Visionado:** Un resumen numérico que desglosa cuántos animes tiene el usuario en cada estado (Pendientes, En Curso, Finalizados), permitiendo un análisis rápido de sus hábitos.
- **Sistema de Logros:** Una vitrina de trofeos o medallas que se desbloquean al alcanzar hitos importantes (ej: "Otaku Nivel 5"), reforzando la sensación de progresión.
- **Configuración de la App:** Desde aquí se gestionan las preferencias globales, como la **Internacionalización (i18n)** para cambiar entre Español e Inglés, y la personalización del tema visual (cambio de paleta de colores entre Naranja y Rosa).



5. Desafíos y Soluciones (Lo que nos costó)

El desarrollo presentó varios retos técnicos que fueron superados durante el proceso:

5.1. Conexión Frontend-Backend-Base de Datos

- **El Problema:** Al principio, el botón de "Login" parecía no funcionar. No había errores visibles, pero la petición no llegaba o la base de datos rechazaba la conexión.
- **La Solución:** Implementamos "logs chivatos" (`console.log`) en cada paso de la función `handleLogin` para trazar la petición. Descubrimos que faltaba crear el usuario en MySQL y usamos un script SQL manual en phpMyAdmin para solucionarlo.

5.2. Arquitectura Híbrida (Persistencia)

- **El Problema:** Migrar todo el estado de la aplicación (inventario, misiones, mascota) a MySQL de golpe era demasiado complejo para el tiempo disponible.
- **La Solución:** Diseñamos una arquitectura híbrida. Los datos críticos (Usuarios y Animes) se guardan en **MySQL** para seguridad, mientras que el estado del juego (Mascota, Inventario) se mantiene en **LocalStorage**. Esto permitió avanzar rápido sin perder funcionalidad.

6. Conclusión

Anime Den es un ejemplo funcional de una aplicación web moderna. Ha permitido poner en práctica conocimientos de bases de datos relacionales, creación de APIs y gestión de estado complejo en el Frontend. La integración de mecánicas de juego sobre una herramienta de productividad demuestra cómo el software puede ser funcional y divertido al mismo tiempo.