

Documentación técnica: El Oro Pa' Las Ratas

1. Introducción al proyecto

“El Oro Pa' Las Ratas” nace como una idea personal, un pasatiempo de verano (2025) donde decidí dibujar a Hommer sujetando una copa, como el Rey de Copas de la baraja española. Alguien cercano me dijo ‘¿y porque no haces una baraja en vez de una sola carta?’ asique manos a la obra. En noviembre acabé 40 cartas desde 0, adaptadas a la baraja española, pero muy vistosas, orginales y jugables. Decidí hacer una pequeña inversión para hacer 20 barajas + caja, y fue un éxito entre mi entorno.

Un poco explicado esto, en octubre 2025, nuestro tutor de 2º, Jose Luís, nos mandó un proyecto en PHP, y yo lo hice sobre una tienda de pegatinas (de algunas de mis cartas, ya que la baraja no estaba acabada), y una vez aprobado este proyecto de PHP y con la baraja, por fin, física, ya tenía claro sobre qué quería que fuera mi TFG, inspirándome en ‘Solitarie & Casual Games’ de Windows, y haciendo juegos típicos de esta baraja española que tantas risas y momentos memorables me han dado (Mus, Mentiroso, Culo, Escoba ...)

2. Arquitectura y Back-End

Para garantizar la estabilidad y seguridad de mi proyecto, he optado por una arquitectura robusta basada en las siguientes tecnologías:

- **Laravel:** Framework principal de PHP utilizado para el enrutamiento, la lógica de negocio y la gestión de la BBDD.
- **Docker y Laravel Sail:** Todo el entorno de desarrollo está contenerizado. Utilizo contenedores de Docker para desplegar el servidor web y la BBDD, lo que asegura que el proyecto funcione de manera idéntica en cualquier máquina.
- **MySQL8:** Motor de Base de Datos (juego_escoba_tfg). Debido a los altos estándares de seguridad de esta versión, la conexión la configuré en un puerto alternativo (3307) y habilitando la recuperación de claves públicas. (allowPublicKeyRetrieval=true) para su correcta administración a través de clientes, como DBeaver.
- **Laravel Breeze:** Sistema de autenticación de usuarios integrado. He modificado y desacoplado sus vistas por defecto para adaptarlas a la identidad gráfica de mi proyecto, manteniendo intacta su potente lógica de seguridad y gestión de sesiones.

3. Front-End e Interfaz de usuario

En la parte frontal del proyecto decido no usar plantillas genéricas, sino que se construyen mediante componentes reutilizables y reactivos:

- **Tailwind CSS:** Framework de estilos utilitarios empleado para maquetar toda la web. He configurado variables de colores personalizados (p.ej: ratas-oscuro, para el background) en el archivo de configuración para mantener una coherencia visual estricta en todas las vistas.
- **Alpine.js:** Micro-Framework de JavaScript utilizado para manejar la reactividad en el lado del cliente sin sobrecarga de herramientas mas pesadas como React. Gracias a Alpine, he implementado:
 - Menús desplegados condicionales según el estado de autenticación (directivas @auth y @guest)
 - Ventanas modales fluidas (como este mismo modal 'Acerca del proyecto')
 - Persistencia de datos mediante localStorage, permitiendo guardar preferencias de usuario (modo oscuro) sin necesidad de consultar a la BBDD.
- **Componentes de Blade:** la interfaz está modularizada en piezas independientes (header, sidebar, modales) que se inyectan en los layouts principales, consiguiendo así un código limpio y fácil de mantener/reparar

4. Decisiones de diseño y Navegabilidad

La estructura de la aplicación sigue siendo un enfoque híbrido de rendimiento:

- **Ventanas principales:** Las secciones pesadas (los juegos de mesa, la tienda, las estadísticas) se cargan en rutas independientes para mantener el código ordenado y usar controladores específicos de Laravel.
- **Modales asíncronos:** Las interacciones de apoyo (ajustes, reglas de juego, información técnica) se abren en modales superpuestos, evitando tiempos de carga innecesarios y manteniendo la inmersión del jugador en la plataforma.
- **Accesibilidad condicionada:** La interfaz evoluciona según el estado del usuario. Un invitado ve una interfaz restringida ("candados" en el sidebar), incitándole al registro, mientras que un usuario autenticado disfruta de acceso total y controles personalizados.

Autor: Hugo Recio Ortega