

ANTEPROYECTO

1- INTRODUCCIÓN

Para adquirir videojuegos hay una gran variedad de plataformas, al igual que para series o películas existen multitud de servicios (Netflix, Prime Video, HBO, Disney +). Esto puede llevar a no saber en que plataforma está disponible un juego o incluso comprar uno que ya habías adquirido.

Esto ahorra tiempo al usuario con una búsqueda única para localizar un juego, evitando tener que visitar todas sus plataformas.

2- OBJETIVO

Desarrollaremos una aplicación web de consulta de videojuegos, pudiendo ver en que plataforma están o las fichas técnicas de los mismos (nombre, género, valoración, PEGI, etc).

Será una aplicación web SPA (Single-Page Application) y constará inicialmente de 3 vistas:

- Inicio / Home : página principal con información sobre las novedades y próximos lanzamientos.
- Búsqueda / Filtros : resultados de una búsqueda generado dinámicamente con una pequeña descripción y con opciones de filtrado.
- Ficha técnica : ficha dinámica con las plataformas donde se encuentra un videojuego, especificaciones, descripción y pudiéndose mostrar imágenes o capturas del mismo.

3- TECNOLOGÍAS

- Arquitectura del proyecto → se hará uso de **Next.js** que nos permite crear un arquetipo fullstack gracias a la posibilidad que ofrece para elegir que partes se ejecutan en servidor y cuales en el cliente. Next.js es un framework de React, por lo que también contará con partes de este último. El entorno de ejecución que utiliza es Node.js junto al gestor de dependencias npm.

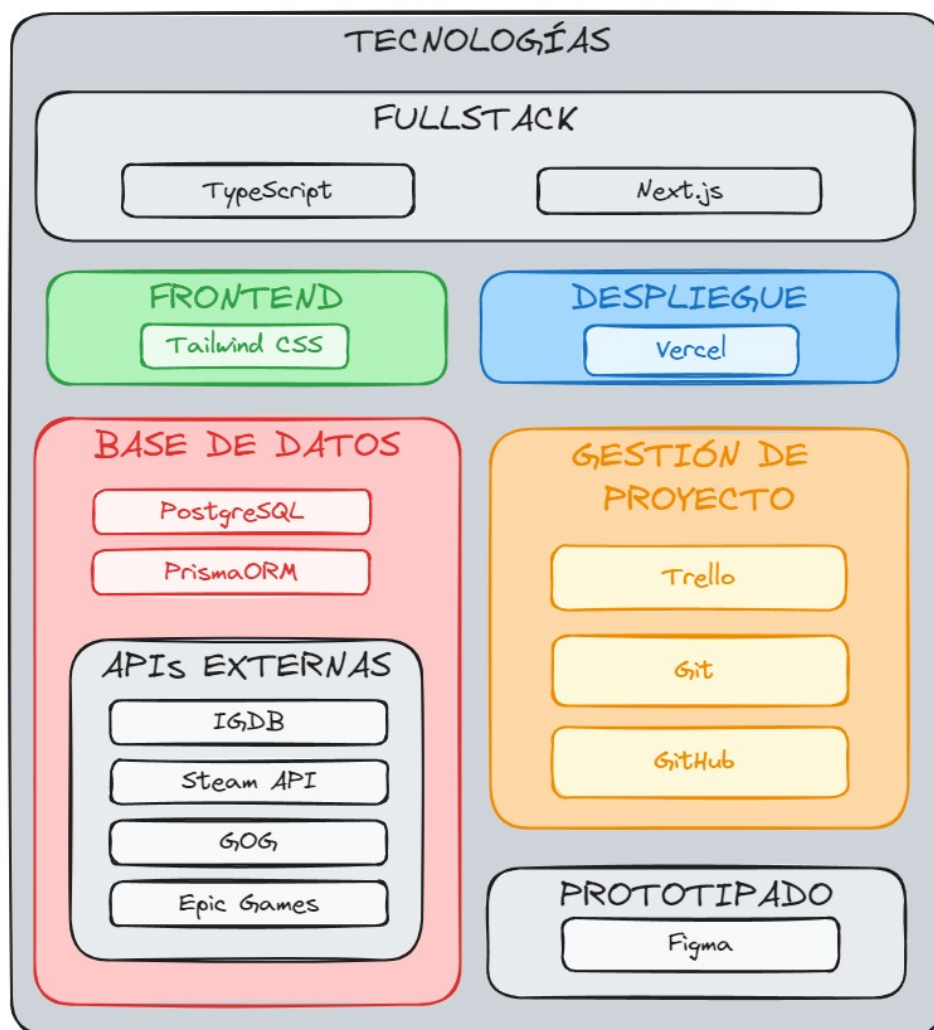
Como lenguaje para este conjunto se utilizará **Typescript** que agrega el tipado a JavaScript como característica a destacar, esto lo convierte en un superset.

Para los estilos contamos con el framework **TailwindCSS** que nos facilitará el proceso para estilizar la aplicación web.

- Base de datos → con **PostgreSQL** se creará y gestionará la base de datos necesaria para el proyecto. Para el mapeo de tablas se utilizará **PrismaORM**.

Este proyecto tiene una fuerte dependencia de APIs externas como **IGDB API**, **Steam API**, **Epic Games API** y **GOG API**. Las consultas hacia IGDB se volcarán en la base de datos propia para reducir el número de llamadas.

- Despliegue → Next.js está desarrollado por **Vercel** que proporciona servicio en la nube. Por este motivo tienen una completa compatibilidad e integración entre ellos aparte de ser gratuito.
- Gestión del proyecto → Para coordinar todas las tareas y los miembros del equipo se utilizará **Trello**. Con **Git** y **GitHub** haremos uso del control de versiones.
- Prototipado → El diseño se realizará con **Figma**, este también nos permite simular ciertas acciones como redireccionamiento o pop-ups para ver un prototipo lo más real posible.



4- FUNCIONALIDADES

A continuación se nombrarán los objetivos o funcionalidades que se pretenden conseguir en este proyecto:

1. Desarrollo de una aplicación web SPA - para obtener un mejor rendimiento y ofrecer más calidad a la experiencia de usuario.
2. Desarrollar un sistema de búsqueda - el usuario podrá escribir el nombre de un juego y en la vista de filtros saldrán todas las posibles coincidencias del mismo.
3. Aplicar filtros – una vez obtenidos los datos de una búsqueda se podrá filtrar por género, plataforma, modo de juego, año de lanzamiento, etc.
4. Establecer una base de datos a partir de la API IGDB – consumiremos la API y parsearemos los archivos JSON recibidos para ajustarlos a nuestra propia estructura.
5. Crear una API propia – estableceremos endpoints de la base de datos que creamos. Esto nos permitirá una mayor flexibilidad a la hora de trabajar y nos ahorraremos hacer llamadas continuas a IGDB API.
6. Consumir APIs de plataformas – consumiremos las APIs de Steam, GOG, Epic Games de manera esporádica cuando IGDB no cubra las necesidades que se requieran.

5- CONCLUSIONES

Esperamos conseguir todos los objetivos propuestos, aunque somos conscientes de que son ambiciosos, ya que no contamos con conocimientos en la mayoría de las tecnologías elegidas y tienen un curva de aprendizaje pronunciada.

Por ello concluimos que a lo largo del desarrollo algunas funcionalidades puedan sufrir cambios, modificaciones o incluso llegar a remplazarse por una nueva.

*« Los que pueden, lo hacen; los que no, solo saben quejarse. »
LinusTorvals*