









Proyecto: POKEDEX con Node.js y React

Descripción del Proyecto

Este proyecto consiste en desarrollar una **Pokedex** completa con un backend en **Node.js** y un frontend en **React + TailwindCSS 4.0**. El backend obtendrá datos de la **PokeAPI** y los almacenará en una base de datos como **MongoDB** (o cualquier otra elegida por el alumno). El frontend permitirá visualizar los Pokémon, buscarlos, ver detalles y habilidades destacadas.

Funcionalidades Principales

-  **Obtención y almacenamiento de datos desde PokeAPI** en una base de datos.
-  **Búsqueda de Pokémon** por nombre.
-  **Visualización de detalles de los Pokémon**, incluyendo imágenes, tipos y habilidades.
-  **Backend con API REST** para servir la información al frontend.
-  **Uso obligatorio de `Promise.all([...promesas])`** para optimizar el consumo de datos en el frontend.
-  **Frontend en React + TailwindCSS 4.0 + React Router v7**.
-  **Dockerización completa** del backend y frontend.
-  **README.md** detallado con instrucciones de instalación y despliegue.

Tecnologías a Utilizar

- **Node.js** (v20+)
- **Express.js** para la gestión del servidor
- **MongoDB** o alternativa para almacenamiento
- **Axios** para consumir la PokeAPI
- **React + TailwindCSS 4.0** para el frontend
- **React Router v7** para la navegación
- **Docker** para la contenedorización







Requisitos del Proyecto

1. **Backend** en Node.js que obtenga datos de **PokeAPI**, los almacene en la base de datos y exponga endpoints para el frontend.
2. **Endpoints requeridos:**
 - **GET /pokemons** → Devuelve todos los Pokémon almacenados.
 - **GET /pokemons/:id** → Devuelve la información detallada de un Pokémon.

- `GET /pokemons/search?name=pikachu` o `/pokemons/pikachu` o →
Permite buscar Pokémon por nombre.
- 3. **Frontend en React** que permita:
 - **Buscar Pokémon** y listarlos con imágenes.
 - **Ver detalles completos de un Pokémon.**
 - **Navegar entre diferentes vistas usando React Router v7.**
- 4. **Optimización con `Promise.all([...promesas])`** para agilizar la carga de datos.
- 5. **Dockerización completa** del backend y frontend.
- 6. **README.md** detallado con instrucciones de instalación y uso.



Rúbrica de Evaluación

Criterio	Descripción	Puntuación
 Obtención de datos y almacenamiento	Consume datos de PokeAPI y los almacena correctamente	2 pts
 Búsqueda de Pokémon	Permite buscar Pokémon de manera efectiva	2 pts
 Detalles de Pokémon	Muestra información detallada en el frontend	2 pts
 Uso de <code>Promise.all([...promesas])</code>	Implementa esta técnica para optimizar el consumo de API	1 pts
 Dockerización	Backend y frontend correctamente dockerizados	1 pts
 Documentación	Instrucciones detalladas en <code>README.md</code> para instalación y uso	1 pts
TOTAL	10 pts	



Ejemplo de Interfaz

(Se recomienda incluir capturas de pantalla del frontend para referencia)