

Evaluación Técnica Laravel

Proyecto: Pokédex con API

DESCRIPCIÓN DEL PROYECTO

Vas a crear una aplicación web tipo Pokédex que consuma la **PokéAPI** (<https://pokeapi.co/>) para mostrar información de Pokémon.

REQUISITOS FUNCIONALES OBLIGATORIOS

1. Consumo de API

Debés consumir la PokéAPI para obtener información de los Pokémon:

- API Base: <https://pokeapi.co/api/v2/>
- Endpoint ejemplo: <https://pokeapi.co/api/v2/pokemon/pikachu>

2. Listado de Pokémon

Creá una vista que muestre un listado de Pokémon.

Requisitos mínimos:

- Mostrar al menos 20 Pokémon
- Cada Pokémon debe mostrar: imagen, nombre y número
- Al hacer clic en un Pokémon debe llevar al detalle

Tip: La API tiene un endpoint `/pokemon?limit=20` para obtener listados

3. Detalle de Pokémon

Creá una vista que muestre el detalle completo de un Pokémon específico.

Debe incluir:

- Imagen del Pokémon
- Nombre y número de Pokédex
- Tipos (Fuego, Agua, Planta, etc.)
- Estadísticas (HP, Ataque, Defensa, etc.)

- Altura y peso

4. Buscador

Implementá un buscador que permita buscar un Pokémon por nombre o número.

ASPECTOS TÉCNICOS A EVALUAR

Código y Buenas Prácticas

- Uso correcto de Eloquent y migraciones (si usás base de datos)
- Organización del código (controladores, modelos, servicios)
- Validación de datos donde corresponda
- Código limpio y legible

Frontend/Vistas

- Uso de Blade correctamente
- Diseño básico presentable (podés usar Bootstrap, Tailwind, o lo que prefieras)
- Interfaz intuitiva y funcional

Base de Datos (si aplica)

- Migraciones bien estructuradas
 - Relaciones correctas
 - Uso apropiado de la base de datos
-

ENTREGABLES

1. **Repositorio Git** con el código del proyecto
 2. **README.md** con:
 - Instrucciones para instalar y ejecutar el proyecto
 - Explicación breve de cómo organizaste el código
 - Qué funcionalidad adicional implementaste y por qué la elegiste
 - Qué decisiones técnicas tomaste y por qué
 3. **Capturas de pantalla** o video corto mostrando la aplicación funcionando
-

QUÉ VAMOS A EVALUAR

- Funcionalidades obligatorias completas y funcionando
- Calidad y organización del código
- Consumo correcto de la API

-  Capacidad de resolver problemas de forma autónoma
 -  Creatividad en la funcionalidad adicional
 -  Documentación y claridad para explicar tu trabajo
-

NOTAS IMPORTANTES

- Podés usar cualquier paquete de Composer que necesites
 - Podés consultar la documentación de Laravel y PokéAPI
 - Lo importante es que funcione y que puedas explicar lo que hiciste
 - No hay una única forma correcta de resolverlo, queremos ver **tu** forma de resolver problemas
 - Valoramos más un código limpio y funcional que algo sobrecargado de features
-

RECURSOS ÚTILES

- **PokéAPI Docs:** <https://pokeapi.co/docs/v2>
- **Laravel Docs:** <https://laravel.com/docs>