

La Pokédex en FLUTTER 2

Objetivo:

El objetivo de estas mejoras es permitir que los alumnos profundicen en el desarrollo de aplicaciones en Flutter, mejorando su capacidad para trabajar con APIs, interfaces gráficas y gestión de datos.

Entregable:

Fecha: 20/03/25

Deberás **subir el código a un repositorio en GitHub** y compartir el correspondiente enlace.

Extensiones y mejoras de la Pokédex:

1. Mejoras en la interfaz:

- Diseño más atractivo con *Material Design*:
 - Agregar tarjetas (*Card*) para cada Pokémon con bordes redondeados y sombras. **HECHO.**
 - Usar colores y tipografías llamativas. **HECHO**
 - Animaciones al cargar los datos. **HECHO.**
- Vista detallada de cada Pokémon: Al hacer clic en un Pokémon, abrir una pantalla con más detalles como:
 - Tipo(s) - Peso y altura - Estadísticas base (HP, Ataque, Defensa, etc.) **HECHO PERO SENCILLO, SOLO PONE EL TIPO.**
- Modo oscuro: Implementar un *ThemeData* para que el usuario pueda elegir entre modo claro y oscuro. **HECHO.**
- Cambio de vista: Permite alternar entre una vista en lista y una vista en cuadrícula con botones de cambio. **HECHO.**

- Cambio de color según tipo: Hacer que el fondo o los bordes de cada tarjeta cambien según el tipo del Pokémon (agua = azul, fuego = rojo, etc.). **HECHO**

2. Mejoras en la funcionalidad:

- Búsqueda de Pokémon: Agregar un *TextField* para filtrar la lista de Pokémon por nombre. **HECHO.**
- Cargar más Pokémon: En lugar de solo los primeros 20, permitir que el usuario cargue más al hacer scroll. **HECHO.**
- Favoritos: Agregar la posibilidad de marcar Pokémon como favoritos y guardarlos localmente con *SharedPreferences*. **HECHO.**
- Filtros por tipo: Permitir filtrar los Pokémon según su tipo (agua, fuego, eléctrico, etc.). **HECHO.**
- Ordenar Pokémon: Agregar una opción para ordenar los Pokémon alfabéticamente o por número en la Pokédex. **HECHO.**
- Modo aleatorio: Botón para mostrar un Pokémon al azar con sus detalles. **HECHO.**
- Notificación tras seleccionar un Pokémon favorito: **FALTA**
 - Una vez que el usuario marca un Pokémon como favorito, después de 3 segundos aparecerá una notificación con el mensaje: "¡[Nombre del Pokémon] ahora es tu favorito!"
 - Para esto, se debe usar el paquete *Flutter Local Notifications*.

3. Mejoras en el manejo de datos:

- Manejo de errores: Mostrar un mensaje personalizado si la API falla o no hay conexión a internet. **HECHO.**

4. Opcionales para subir nota:

- Optimización del consumo de la API:
 - Carga eficiente y rápida de los Pokémon (Ejemplo: Se puede usar *Pagination* y *Lazy Loading*). **FALTA**
 - Guardar los Pokémon descargados en una base de datos local (*sqflite*) para evitar llamadas repetitivas a la API. **FALTA.**
- Animaciones avanzadas:
 - Usar *Hero* para transiciones suaves entre pantallas. **HECHO.**
 - Aplicar efectos como fade-in o zoom al mostrar los Pokémon. **HECHO.**
- Animación de carga: En lugar del *CircularProgressIndicator*, usar una animación con un ícono de Poké Ball girando. **HECHO.**

- Sonidos al interactuar: Reproducir un sonido cuando se toca un Pokémon, como su grito característico. **HECHO**
- Temas personalizados: Permitir a los usuarios cambiar el tema de la app con opciones predefinidas (Pokédex clásica, retro, etc.). **FALTA**
- Comparador de Pokémon: Permitir seleccionar dos Pokémon y compararlos en estadísticas. **FALTA**