

UD3E10 Proyecto grupal – API de Pokémon (PokeAPI)

Introducción

El objetivo de esta práctica es planificar, desarrollar, defender y valorar un proyecto desarrollado en grupo. Vamos a construir una aplicación sobre la API de Pokémon en la que expondremos la información más relevante en formato wiki. Una vez familiarizados con el modelo de datos desarrollaremos un pequeño juego basado en la información disponible.

Documentación

La documentación oficial se encuentra en la ruta: <https://pokeapi.co/docs/v2>

La ruta base es: <https://pokeapi.co/api/v2/> + {endpoint}

Los “endpoints” se agrupan en torno a 11 bloques (en negrita los que considero especialmente relevantes).

Grupo “Berries”

- berry
- berry-firmness
- berry-flavor

Grupo “Contest”

- contest-type
- contest-effect
- super-contest-effect
- encounter-method
- encounter-condition
- encounter-condition-value

Grupo “Evolution”

- evolution-chain
- evolution-trigger

Grupo “Games”

- **generation**
- pokedex
- version
- version-group

Grupo “Items”

- item
- item-attribute
- item-category
- item-fling-effect
- item-pocket

Grupo “Locations”

- location
- location-area
- pal-park-area
- **region**

Grupo “Machines”

- machine

Grupo “Moves”

- **move**
- move-ailment
- move-battle-style
- move-category
- move-damage-class
- move-learn-method
- move-target

Grupo “Pokemon”

- **ability**
- characteristic
- egg-group
- gender
- growth-rate
- nature
- pokeathlon-stat
- **pokemon**
- pokemon-color
- pokemon-form
- pokemon-habitat
- pokemon-shape
- pokemon-species
- stat
- **type**

Grupo “Utility”

- language

Debes identificar el modelo de datos y ponerlo en común, el punto de entrada es la entidad “Pokémon”.

Por ejemplo, la ficha de “Bulbasaur”.

Endpoint: <https://pokeapi.co/api/v2/pokemon/1>

Si estudiamos sus propiedades además de las propiedades básicas encontramos otras algo escondidas de gran utilidad.

- Stats: Valores numéricos de hp, attack, defense, special-attack, special-defense, speed.
- Sprites: Contiene imágenes del Pokémon.
 - Ejemplo: sprites.other.showdown incluye gifs.

Objetivos “guía” de la práctica

NOTA: LO QUE SE INDICA A CONTINUACIÓN SON MÍNIMOS, QUE PUEDES AMPLIAR.

1. Buscador y ficha de Pokémon

- Implementar un buscador con paginador para listar los Pokémon disponibles.
 - Permitir filtro por generación.
- La ficha del Pokémon debe mostrar:
 - Sus estadísticas (HP, Attack, Defense, etc.).
 - Sus tipos y una tabla de daños:
 - A qué tipos hace más daño.
 - Contra qué tipos es débil.
- Sus imágenes y sprites.

2. Comparador de Pokémon

- Implementar dos buscadores que permitan seleccionar dos Pokémon.
- Comparar los Pokémon seleccionados según sus estadísticas y tipos.
- Opcional: Desde la ficha de un Pokémon, incluir un botón "Comparar" para seleccionar otro Pokémon.

3. Generador y gestor de equipos

- Crear equipos de 6 Pokémon de manera aleatoria o manual.
- Filtro por generación y/o por tipo.
- Funcionalidades de gestión:
 - Asignar un nombre al equipo.
 - Editar manualmente los Pokémon del equipo.
 - Borrar equipos.
- Los Pokémon del equipo deben pertenecer a la misma generación.
- Los equipos podrán ser humanos o máquina.
- Opcional: Desde la ficha de equipo, incluir un botón para “Jugar contra” dicho equipo.

4. Batalla entre Pokémon o equipos

- Implementar una batalla 1 contra 1:

- El Pokémon con mayor velocidad comienza.
 - Simular un combate basado en ataque y defensa, considerando un rango aleatorio para variar los valores.
 - Tener en cuenta las fortalezas y debilidades de tipo.
- Implementar una batalla entre equipos:
 - Se participa con un equipo humano.
 - El equipo contrincante se puede elegir manualmente o de manera aleatoria.
 - Los Pokémon del equipo tienen un orden aleatorio en cada batalla.
 - En cada iteración, el Pokémon ganador conserva el daño recibido.
 - El jugador humano elige cual es el siguiente Pokémon del equipo a participar cada vez que pierde un Pokémon.
 - Guardar resultados y generar una clasificación de equipos.

5. Opciones auxiliares

- Optimizar cacheando los resultados de los endpoint para mejorar el rendimiento (localStorage o sessionStorage).
- Poder limpiar el Local Storage.
- Exportar/importar los datos de Local Storage en formato JSON (los endpoint no).
 - Datos a almacenar: equipos, batallas entre equipos.

Entregables

- El equipo deberá realizar una presentación de su proyecto, en el que se incluirán al menos los apartados:
 - Funcionalidades
 - Planificación
 - Arquitectura
 - Ejemplo de uso
 - Dificultades/Resumen
- Código y documentación del proyecto.
 - Memoria del proyecto (básicamente los mismos apartados del punto anterior pero desarrollados).
- Seguimiento del proyecto, que incluye:
 - Planificación de tareas y cronología.
 - Reparto de trabajo por miembro y día.
 - Commits diarios.

Evaluación

Cada miembro del equipo deberá rellenar una ficha de evaluación personal y coevaluación.

Habrà una defensa y se calificarán los distintos entregables.

La nota del proyecto se repartirá entre los miembros del equipo según la coevaluación.