



Proyecto Final

De Haro Rivas Emiliano

Espinosa Camacho Emanuel Alexander

Gomez Estrada Brian Ismael



Visitar el proyecto terminado https://briange05.github.io/Api_Pokemon/

Repositorio: https://github.com/BrianGE05/Api_Pokemon

Documento o presentación con los puntos del primer avance y la documentación del código y los pasos de creación de su sitio desde el Back y el Front.

Back-end

Describiremos un poco el código:

HTML:

Se define un documento HTML básico con un contenedor principal `<main>` con la clase `.flex`.

Se utiliza un elemento `<template>` con el id `card` que define la estructura de una tarjeta de Pokémon. Contiene una imagen del Pokémon, su nombre, y estadísticas de ataque, ataque especial y defensa.

CSS:

Se aplican estilos simples para diseñar las tarjetas de Pokémon. Se utiliza flexbox para organizar las tarjetas en una fila y se establecen estilos de borde, sombra y tamaño de imagen.

JavaScript:

La función `getRandomInt(min, max)` devuelve un número aleatorio entre `min` (incluido) y `max` (excluido).

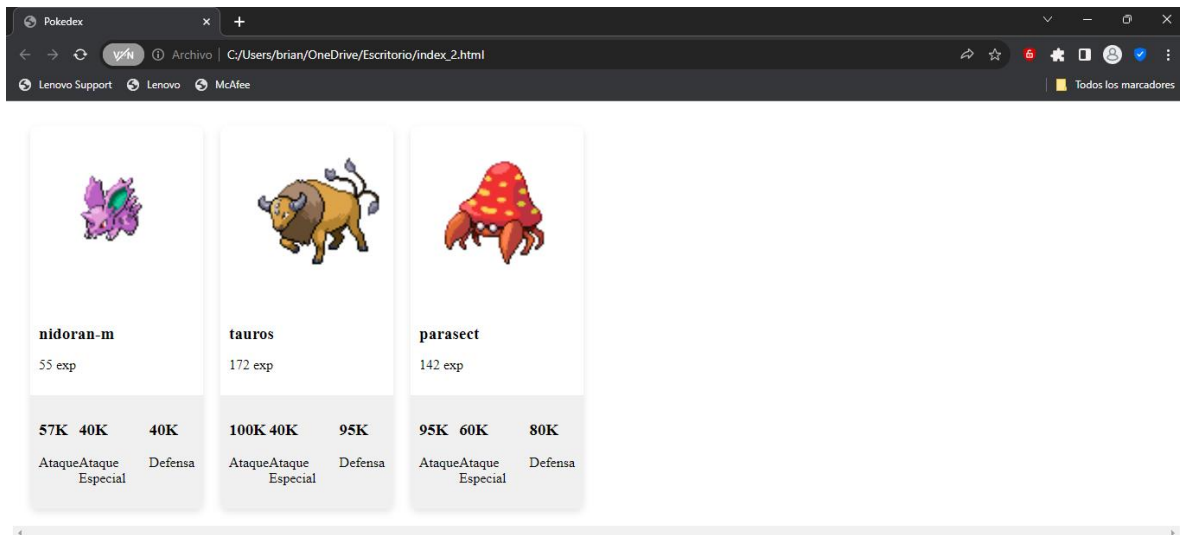
El evento `DOMContentLoaded` se activa cuando el DOM está completamente cargado. En este evento, se obtienen tres números aleatorios que representan los IDs de los Pokémon y se llama a la función `fetchData` para obtener datos de cada uno.

La función `fetchData(id, flex)` realiza una solicitud a la API de PokeAPI para obtener los datos de un Pokémon específico según su ID. Luego, los datos se pasan a la función `pintarCard` para mostrar la tarjeta del Pokémon en la página.

La función `pintarCard(pokemon, flex)` toma los datos del Pokémon obtenidos de la API y los inserta en la plantilla de tarjeta de Pokémon. Luego, agrega la tarjeta completa al contenedor principal `.flex`.

Front-end

Se muestra al usuario son tres diferentes pokemones que al actualizar la página se muestran otros tres diferentes pokemones, las tarjetas de los pokemones muestran su información como ataque, ataque especial y defensa.



Modificaciones al back-end y front-end

Back-end

HTML:

Se define un documento HTML básico con un contenedor principal `<main>` con la clase `.flex`.

Se utiliza un elemento `<template>` con el id `card` que define la estructura de una tarjeta de Pokémon. Contiene una imagen del Pokémon, su nombre, y estadísticas de ataque, ataque especial y defensa.

Hay un título `<h1>` con la clase `.title` que se encuentra centrado en la página y muestra el texto "Pokemones".

CSS:

Se definen estilos para dar formato al diseño de la página web. Se utiliza flexbox para organizar las tarjetas de Pokémon en una cuadrícula y se establecen estilos de borde, sombra y tamaño de imagen para las tarjetas.

JavaScript:

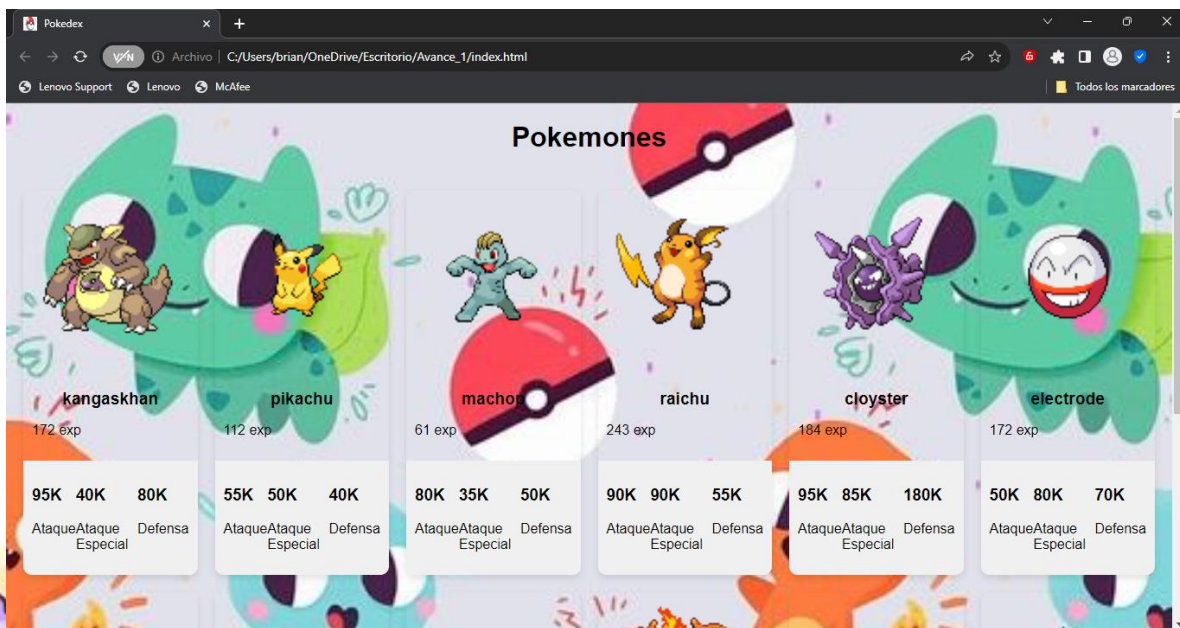
La función `getRandomInt(min, max)` devuelve un número aleatorio entre `min` (incluido) y `max` (excluido).

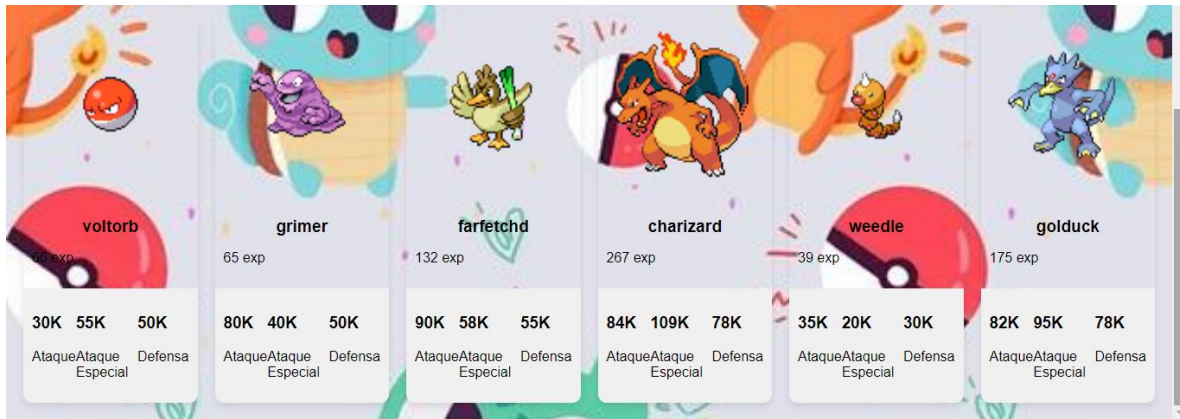
El evento `DOMContentLoaded` se activa cuando el DOM está completamente cargado. En este evento, se obtienen 12 números aleatorios que representan los IDs de los Pokémon y se llama a la función `fetchData` para obtener datos de cada uno.

La función `fetchData(id, flex)` realiza una solicitud a la API de PokeAPI para obtener los datos de un Pokémon específico según su ID. Luego, los datos se pasan a la función `pintarCard` para mostrar la tarjeta del Pokémon en la página.

La función `pintarCard(pokemon, flex)` toma los datos del Pokémon obtenidos de la API y los inserta en la plantilla de tarjeta de Pokémon. Luego, agrega la tarjeta completa al contenedor principal `.flex`.

Front-end





Ahora el usuario puede visualizar 12 pokemones, y ver la imagen de fondo, es un ambiente más amigable.

Carpeta del proyecto generada con la estructura del proyecto del lado del back y del front, es decir, todo lo generado de manera local para su sitio.

```
pokedex_project/
|
├── backend/
|   ├── node_modules/
|   ├── index.js
|   ├── package-lock.json
|   └── package.json
└── frontend/
    ├── index.html
    ├── styles.css
    ├── script.js
    └── images/
        └── pokemon1.png
```

Documentación del proyecto en su Github.

Se adjunta link del repositorio en Github

4. Conclusión. Quisiera redactaran su experiencia de trabajo y aprendizaje a lo largo de la materia, pero más específicamente durante el desarrollo de este su proyecto final.

Nuestra experiencia a lo largo de la materia ha sido extremadamente buena ya que aprendimos bastante. Exploramos el fascinante mundo de las APIs, lo cual nos brindó una comprensión más profunda de cómo interactúan las aplicaciones con los datos y los servicios externos. Al trabajar en nuestro proyecto, descubrimos el poder y la versatilidad de las APIs al utilizarlas para obtener información sobre Pokémones.

Además, la capacidad de clonar un repositorio y trabajar en colaboración nos brindó una excelente oportunidad para aprender sobre control de versiones y trabajo en equipo. Esta práctica nos permitió gestionar eficazmente nuestro código y coordinar nuestros esfuerzos para lograr nuestros objetivos.

