

# DESARROLLO DE SOFTWARE FRONTEND I

## ESPINOSA MARTINEZ CARLOS LEONEL

Aarón Hernández García

### 6.1 Ejercicio PokeAPI

## Realizar un análisis del proyecto adjunto que consume la api de productos por medio de fetch de Javascript.

### Objetivo de la Web (Pokedex):

Aprender a consumir y manipular datos de una API pública utilizando fetch en JavaScript, manejando respuestas y errores, y mostrando información en una aplicación web interactiva.

### Código HTML:

```
index.html
EjemploAPI_Pokedex > EjemploClase > index.html > html > head > link
1 <html lang="es">
2 <head>
3   <meta charset="UTF-8">
4   <meta http-equiv="X-UA-Compatible" content="IE=edge">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="preconnect" href="https://fonts.gstatic.com">
7   <link rel="stylesheet" href="styles.css">
8   <link href="https://fonts.googleapis.com/css2?family=
9     =Ubuntu:ital,wght@0,300;0,400;0,500;0,700;1,300;1,400;1,500;1,700&d
10  rel="stylesheet">
11 </head>
12 <body>
13   <div class="head">
14     <h1>Pokedex</h1>
15   </div>
16   <div id="container">
17     <script src="main.js"></script>
18   </div>
19 </body>
20 </html>
```

## Código CSS:

```
<> index.html JS main.js # styles.css X
EjemploAPI_Pokedex > EjemploClase > # styles.css > .head
1  /* Establece estilos básicos para todos los elementos */
2  * {
3      margin: 0; /* Elimina los márgenes por defecto */
4      padding: 0; /* Elimina el relleno por defecto */
5      box-sizing: border-box; /* Incluye el padding y borde en el tamaño total */
6  }
7
8  /* Estilos del cuerpo del documento */
9  body {
10     background-color: #ecec; /* Color de fondo del cuerpo */
11     font-family: 'Ubuntu', sans-serif; /* Fuente principal */
12 }
13
14 /* Estilos para el encabezado */
15 .head {
16     background-color: rgb(49, 109, 187); /* Color de fondo del encabezado */
17     padding: 20px; /* Relleno alrededor del contenido */
18     position: fixed; /* Fija la posición en la parte superior */
19     top: 0; /* Coloca el encabezado en la parte superior de la ventana */
20     width: 100%; /* Ocupa todo el ancho */
21     color: white; /* Color del texto */
22 }
```

```
/* Contenedor principal */
#container {
    width: min(100%, 1000px); /* Ancho máximo de 1000px o 100% */
    margin: 100px auto; /* Margen superior e inferior de 100px, centrado horizontalmente */
    display: flex; /* Utiliza flexbox para la disposición de elementos */
    flex-wrap: wrap; /* Permite que los elementos se ajusten a la siguiente fila */
    justify-content: space-between; /* Distribuye el espacio entre los elementos */
}

/* Estilos para las tarjetas */
.card {
    width: 230px; /* Ancho de cada tarjeta */
    margin-top: 20px; /* Margen superior de 20px (ajustado para mejor alineación) */
    background-color: white; /* Fondo blanco */
    padding: 20px; /* Relleno interno */
    border-radius: 20px; /* Bordes redondeados */
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1); /* Añade una sombra sutil */
    transition: transform 0.2s; /* Transición suave para efectos */
}

.card:hover {
    transform: translateY(-5px); /* Eleva la tarjeta al pasar el ratón */
}













.card img {
    width: 100%; /* Las imágenes ocupan todo el ancho de la tarjeta */
    border-radius: 10px; /* Bordes redondeados en las imágenes */
}









.card span {
    color: #6e6e6e; /* Color gris para el texto */
    font-weight: 500; /* Peso medio para el texto */
    display: block; /* Asegura que el texto ocupe toda la línea */
    margin-top: 10px; /* Espacio entre el texto y el elemento anterior */
}
```

## Código JS:

```
index.html × JS main.js ×
EjemploAPI_Pokedex > EjemploClase > JS main.js > ...
1 // Obtenemos los datos de todos
2 //los pokemon con esta funcion
3 function fechPokemon(id){
4   fetch(`https://pokeapi.co/api/v2/pokemon/${id}/`)
5     .then(response => response.json())
6     .then(json => {
7       crearpokemon(json);
8     });
9   }
10 // Funcion que crea el ciclo para
11 //listar los pokemnes en en cantidad
12 function printPokemons(number) {
13   for(let i=1 ;i<=number;i++){
14     fechPokemon(i);
15   }
16 }
17 // Pinta el pokemo enviado insertando un HTML dentro del #container
18 function crearpokemon(pokemon){
19   container.innerHTML = `
20     ${container.innerHTML}
21     <div class="card">
22       
23       <span>Nº. ${pokemon.id}</span>
24       <h2>${pokemon.name}</h2>
25     </card>
26   `;
27 }
28 //Ejecutamos la funcion enviando la cantidad de pokemon a pintar.
29 printPokemons(30);
30
```

## Pagina Web (Pokedex)

Pokedex			
			
Nº.1 bulbasaur	Nº.2 ivysaur	Nº.3 venusaur	Nº.5 charmeleon
			
Nº.8 wartortle	Nº.4 charmander	Nº.6 charizard	Nº.7 squirtle
			

Pokedex			
pidgeotto	pidgeot	rattata	raticate
			
Nº.21 spearow	Nº.22 fearow	Nº.23 ekans	Nº.24 arbok
			
Nº.25 pikachu	Nº.26 raichu	Nº.27 sandshrew	Nº.28 sandslash

**Experiencia:**

Trabajar con fetch y la PokéAPI ha sido una experiencia enriquecedora que me ha permitido comprender mejor cómo interactuar con APIs en JavaScript. A través de este proyecto, aprendí a realizar solicitudes para obtener datos de Pokémon, lo que me permitió explorar cómo manejar respuestas JSON y presentar la información de manera dinámica en una interfaz web.