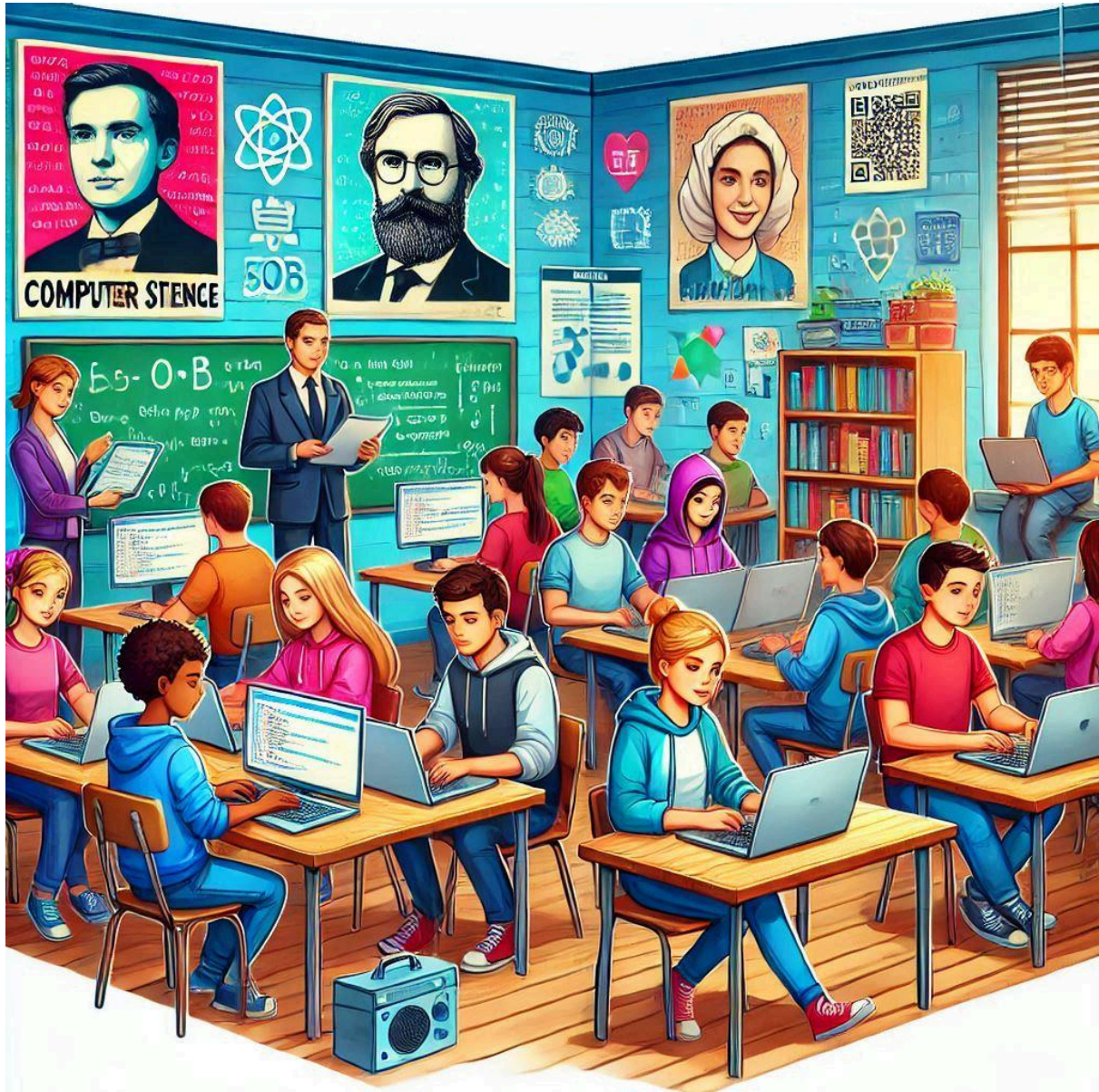


## Memória



## **Índice**

Introducción.....	3
HTML Y CSS.....	4
JavaScript.....	5
Conclusiones y Complicaciones.....	8

## **Introducción**

### **Objetivo del Proyecto**

Desarrollar una SPA interactiva y dinámica que permita a los usuarios gestionar información detallada sobre los Pokémon. A través de esta aplicación, los usuarios podrán:

Crear nuevos registros de Pokémon.

- Leer y visualizar detalles de los Pokémon.
- Actualizar información existente de los Pokémon.
- Eliminar registros de Pokémon

## HTML Y CSS

El HTML representa la caja general (container) donde guardaré dinámicamente tanto los formularios como la tabla interactiva, meto dentro un div de la tabla para interactuar y crear sobre ella y de allí podemos interactuar con JavaScript

El CSS mostrado es principalmente para darle ambiente de pokémon donde el usuario tenga una buena interactividad con la tabla y poder adentrarse mejor a lo que corresponde esta web

Así se ve como lo ve el usuario:

POKÉMON										
id	nombre	tipo	entrenador	pokemon	movimientos	nivel	region	Eliminar	Actualizar	Agregar
1	Squirtle	Agua			Aqua cola	15	Kanto	<button>Eliminar</button>	<button>Actualizar</button>	<button>Agregar</button>
2	Cyndaquil	Fuego			Lanzallamas	15	Johto	<button>Eliminar</button>	<button>Actualizar</button>	<button>Agregar</button>
3	Treecko	Planta			Hierba lazo	15	Hoenn	<button>Eliminar</button>	<button>Actualizar</button>	<button>Agregar</button>
4	Turtwig	Planta			Hoja afilada	15	Sinnoh	<button>Eliminar</button>	<button>Actualizar</button>	<button>Agregar</button>
5	Oshawott	Agua			Concha filo	15	Teselia	<button>Eliminar</button>	<button>Actualizar</button>	<button>Agregar</button>
6	Froakie	Agua			Surf	15	Kalos	<button>Eliminar</button>	<button>Actualizar</button>	<button>Agregar</button>

# JavaScript

## 1. TablaPokemon()

### Descripción:

Crea y muestra dinámicamente una tabla con la información de los Pokémon.

### Funcionamiento:

- Limpia el contenido del contenedor principal.
- Crea una tabla HTML con los encabezados basados en las propiedades del JSON de Pokémon.
- Itera sobre el array `pokemon` para generar filas con la información de cada Pokémon, incluyendo imágenes de Pokémon y entrenador.
- Añade botones en cada fila para **Eliminar**, **Actualizar** y **Agregar** Pokémon, permitiendo la gestión completa de los datos.

## 2. formAgregarPokemon(indice)

### Descripción:

Muestra un formulario interactivo para agregar un nuevo Pokémon.

### Funcionamiento:

- Limpia el contenido del contenedor principal.
- Crea un formulario con los siguientes campos:
  - **Nombre** (input de texto)
  - **Tipo** (input de texto)
  - **Entrenador** (input tipo file para imagen)
  - **Imagen del Pokémon** (input tipo file para imagen)
  - **Movimientos** (checkbox para seleccionar varios movimientos)
  - **Nivel** (input de número)
  - **Región** (select con opciones predefinidas)
- Agrega un botón para enviar el formulario y otro para volver a la tabla principal.

## 3. formEditarPokemon(indice)

### Descripción:

Despliega un formulario para editar la información de un Pokémon existente.

### Funcionamiento:

- Limpia el contenido del contenedor principal.
- Rellena los campos del formulario con los datos actuales del Pokémon seleccionado.
- Los campos incluyen:
  - **Nombre y Tipo** (input de texto)
  - **Entrenador e Imagen del Pokémon** (inputs tipo file)

- **Movimientos** (checkbox con opciones preseleccionadas)
- **Nivel** (input de número)
- **Región** (select con la región actual seleccionada)
- Incluye botones para **Guardar Cambios** y **Volver** a la tabla.

#### 4. validarFormulario(...inputs)

**Descripción:**

Valida que los campos del formulario no estén vacíos.

**Funcionamiento:**

- Itera sobre los inputs recibidos y verifica si tienen valor.
- Si un campo está vacío, su borde se colorea de rojo.
- Devuelve `true` si todos los campos están completos, y `false` si falta algún dato.

#### 5. editarPokemon(indice, nombre, tipo, entrenadorFile, pokemonFile, movimientos, nivel, region)

**Descripción:**

Actualiza los datos de un Pokémon existente.

**Funcionamiento:**

- Modifica el objeto Pokémon con los nuevos valores proporcionados.
- Si se cargan nuevas imágenes, actualiza sus URLs.
- Refresca la tabla de Pokémon para reflejar los cambios.

#### 6. agregarPokemon(nombre, tipo, entrenador, pokemonImg, movimientos, nivel, region)

**Descripción:**

Añade un nuevo Pokémon al array.

**Funcionamiento:**

- Crea un nuevo objeto Pokémon con los datos ingresados en el formulario.
- Si no se cargan imágenes, usa una imagen por defecto.
- Inserta el nuevo Pokémon en el array y actualiza la tabla.

#### 7. pokemonVacio()

**Descripción:**

Verifica si hay Pokémon registrados y muestra un mensaje si no hay ninguno.

**Funcionamiento:**

- Limpia el contenido del contenedor.

- Si el array `pokémon` está vacío, muestra un mensaje indicando que no hay Pokémon registrados.

## 8. eliminarPokemonConConfirmacion(indice)

### Descripción:

Muestra un mensaje de confirmación antes de eliminar un Pokémon.

### Funcionamiento:

- Muestra un mensaje con el nombre del Pokémon a eliminar.
- Incluye dos botones:
  - **Sí:** Llama a `eliminarPokemon(indice)` para eliminar el Pokémon.
  - **No:** Cancela la eliminación y vuelve a la tabla.

## 9. eliminarPokemon(indice)

### Descripción:

Elimina un Pokémon del array.

### Funcionamiento:

- Remueve el Pokémon del array `pokemon` según el índice.
- Refresca la tabla para reflejar la eliminación.

## **Conclusiones y Complicaciones**

Es un proyecto donde he aprendido a saber hacer un CRUD en base a un JSON interactivo, a hacer un SPA con JS, he entendido los conceptos y cómo hacer validaciones de formularios como saber validar y agregar objetos nuevos, editarlos y eliminarlos del JSON.

Complicaciones -> Las complicaciones que he padecido en este proyecto han sido principalmente, saber validar y subir checkbox o inputs, aprender a editar el pokémon me ha costado bastante.