

LAB: The Web Storage APIs

Objetivo:

Crear una app web donde:

1. **Buscarás Pokémon** por nombre (usando la PokéAPI).
2. **Guardarás tus favoritos** en el navegador (con localStorage).

Parte 1: Entendiendo las herramientas a usar

1. La PokéAPI

- Es una API gratuita que devuelve datos de Pokémon en formato JSON.
- Cómo usarla: Haremos peticiones a esta URL:

`https://pokeapi.co/api/v2/pokemon/[nombre-del-pokemon]`

- Ejemplo para Pikachu: <https://pokeapi.co/api/v2/pokemon/pikachu>.

Datos importantes en la respuesta:

- Nombre: `response.name` ("pikachu").
- Imagen: `response.sprites.front_default` (URL de la imagen frontal).

2. fetch, Cómo obtener datos:

- Es una función de JavaScript para hacer peticiones HTTP.
- **Estructura básica:** Ej de como usarlo, pero la investigación que hagan será fundamental, no necesariamente debe quedar igual, es un ejemplo.

```
fetch(url)
  .then(function(response) {
    return response.json();
  })
  .then(function(data) {
    // Datos del Pokémon aquí
    console.log(data);
  })
  .catch(function(error) {
    alert("¡Error! Pokémon no encontrado");
  });
```

3. localStorage, Guardar favoritos:

Almacena datos en el navegador, como un diccionario clave-valor.

Métodos clave:

- `localStorage.setItem("clave", valor)`: Guarda datos (String).

- `localStorage.getItem("clave")`: Recupera datos.
- Para guardar arrays/objetos, usen `JSON.stringify()` al guardar y `JSON.parse()` al leer.

Parte 2: Tareas a realizar

Tarea 1: Estructura HTML

Debe tener:

1. Un `<input>` para escribir el nombre del Pokémon.
2. Un `<button>` para buscar.
3. Un `<div>` para mostrar el resultado, aquí pondrán la imagen y nombre del Pokémon.
4. Un `<button>` para guardar como favorito.
5. Una sección (`<div id="favoritos">`) para listar los Pokémon favoritos.

Tarea 2: Funciones JavaScript

Implementarán estas funciones:

1. **searchPokemon()** (Buscar Pokémon):

- Obtiene el nombre del `<input>`.
- Usa `fetch` para pedir datos a la PokéAPI.
Si el Pokémon existe:
- Muestra su **imagen** (`sprites.front_default`) y **nombre** en el `<div>` de resultados.
- Guarda los datos en una variable global (para usarla luego al guardar).
- Si no existe: Muestra un `alert` "Pokémon no encontrado."

2. **saveFavorite()** (Guardar favorito):

- Verifica que la variable global tenga datos.
- Obtiene la lista actual de favoritos de `localStorage`, si no hay, crea un array vacío.
- Añade el Pokémon al array **solo si no está repetido**.
- Guarda el array en `localStorage` con `JSON.stringify()`.
- Llama a `updateFavoritesList()` para actualizar la lista en pantalla.

3. **updateFavoritesList()** (Actualizar lista de favoritos):

- Lee los favoritos de `localStorage` (con `JSON.parse()`).
- Borra el contenido anterior del `<div id="favoritos">`.

- Por cada Pokémon en la lista:
 - Crea un elemento (ej: `<div>` con su imagen y nombre).
 - Lo añade al `<div id="favoritos">`.
- **Al cargar la página:**
 - Llama a `updateFavoritesList()` para mostrar los favoritos guardados.