**Index.js**

**<!DOCTYPE html>**

**<html lang="es">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Document</title>**

**</head>**

**<body>**

**<!-- Ejercicio 2.1 – Nombre de un Pokémon -->**

**<h2 id="nombre-pokemon">Cargando...</h2>**

**<!-- Ejercicio 2.2 – Chiste Aleatorio -->**

**<p id="chiste">Cargando chiste...</p>**

**<!-- Ejercicio 2.3 – Capital de Argentina -->**

**<div id="info-pais">Cargando datos...</div>**

**<!-- 3. Renderizando Objetos Complejos -->**

**<!-- Ejercicio 3.1 – Tarjeta Pokémon Mejorada -->**

**<div id="pokemon-card"></div>**

**<!-- Ejercicio 3.2 – Perfil Rick and Morty -->**

**<div id="rick-morty-card"></div>**

**<!-- 5. Renderizado de Listas -->**

**<!-- Ejercicio 5.1 – Lista de Tareas -->**

**<ul id="lista-tareas"></ul>**

**<!-- Ejercicio 5.2 – Galería de Gatos -->**

**<div id="galeria-gatos" style="display:grid;grid-template-columns:repeat(3, 1fr);gap:10px;"></div>**

**<!-- 6. Interactividad con Usuario -->**

**<!-- Ejercicio 6.1 – Buscador Rick and Morty -->**

**<input type="text" id="nombre-personaje">**

**<button id="buscar-btn">Buscar</button>**

**<div id="resultado-busqueda"></div>**

**<!-- Ejercicio 6.2 – Generador de Consejos -->**

**<p id="consejo-texto">Haz clic para un nuevo consejo</p>**

**<button id="nuevo-consejo-btn">Obtener Consejo</button>**

**<script src="app.js"></script>**

**</body>**

**</html>**

**App.js**

**1. Fundamentos: APIs y fetch()**

**Ejemplo básico con comentarios**

fetch('https://jsonplaceholder.typicode.com/todos/1')

.then(response => {

    if (!response.ok) throw new Error('Error en la respuesta de red');

    return response.json();

})

.then(data => {

    console.log("Datos recibidos:", data);

})

.catch(error => {

    console.error("Hubo un problema con fetch:", error);

});

2. Peticiones Básicas y Renderizado Simple

Objetivo: Acostumbrarse al flujo de fetch y a mostrar un único dato en la página.

**Ejercicio 2.1: El Nombre de un Pokémon**

API: PokéAPI

Tarea:

1.   HTML: Crea un index.html con <h2 id="nombre-pokemon">Cargando...</h2>.

2.   JS: Crea un app.js. Usa fetch para pedir datos del Pokémon "charmander" a la URL https://pokeapi.co/api/v2/pokemon/charmander.

3.   Cuando obtengas los datos, selecciona el h2 y cambia su textContent por el nombre del Pokémon (que se encuentra en la propiedad name del objeto de datos).

fetch('https://pokeapi.co/api/v2/pokemon/ninetales')

.then(res => res.json())

.then(data => {

    document.getElementById("nombre-pokemon").textContent = data.name;

});

**Ejercicio 2.2 – Chiste Aleatorio**

fetch('https://api.chucknorris.io/jokes/random')

.then(res => res.json())

.then(data => {

    document.getElementById("chiste").textContent = data.value;

});

**Ejercicio 2.3 – Capital de Argentina**

fetch('https://restcountries.com/v3.1/name/argentina')

.then(res => res.json())

.then(data => {

    let capital = data[0].capital[0];

    document.getElementById("info-pais").textContent = `Capital: ${capital}`;

});

**3. Renderizando Objetos Complejos**

**Ejercicio 3.1 – Tarjeta Pokémon Mejorada**

fetch('https://pokeapi.co/api/v2/pokemon/pikachu')

.then(res => res.json())

.then(datos => {

    let contenedor = document.getElementById("pokemon-card");

    contenedor.innerHTML = ''; // Limpiar

let nombre = document.createElement("h2");

    nombre.textContent = datos.name;

    let imagen = document.createElement("img");

    imagen.src = datos.sprites.front\_default;

    let tipo = document.createElement("p");

    tipo.textContent = `Tipo: ${datos.types[0].type.name}`;

    let habilidad = document.createElement("p");

    habilidad.textContent = `Habilidad: ${datos.abilities[0].ability.name}`;

    contenedor.append(nombre, imagen, tipo, habilidad);

});

**Ejercicio 3.2 – Perfil Rick and Morty**

fetch('https://rickandmortyapi.com/api/character/2')

.then(res => res.json())

.then(data => {

    let contenedor = document.getElementById("rick-morty-card");

    contenedor.innerHTML = '';

    let nombre = document.createElement("h2");

    nombre.textContent = data.name;

    let imagen = document.createElement("img");

    imagen.src = data.image;

    let especie = document.createElement("p");

    especie.textContent = `Especie: ${data.species}`;

    let origen = document.createElement("p");

    origen.textContent = `Origen: ${data.origin.name}`;

    contenedor.append(nombre, imagen, especie, origen);

});

**4. async/await: Sintaxis Moderna**

**Ejercicio 4.1 – Refactor de Tarjeta Pokémon**

async function mostrarPokemon(nombrePokemon) {

try {

    let res = await fetch(`https://pokeapi.co/api/v2/pokemon/${nombrePokemon}`);

    let datos = await res.json();

    let card = document.getElementById("pokemon-card");

    card.innerHTML = '';

    card.innerHTML = `

    <h2>${datos.name}</h2>

    <img src="${datos.sprites.front\_default}">

    <p>Tipo: ${datos.types[0].type.name}</p>

    <p>Habilidad: ${datos.abilities[0].ability.name}</p>

    `;

} catch (error) {

    console.error("Error al mostrar Pokémon:", error);

}

}

mostrarPokemon("ninetales");

**Ejercicio 4.2 – Simular API de Clima**

async function consultarClima(ciudad) {

try {

    console.log(`Consultando clima para ${ciudad}...`);

    let clima = await new Promise(resolve => {

    setTimeout(() => resolve({ temperatura: 25, condicion: "Soleado" }), 2000);

    });

    console.log(`La temperatura en ${ciudad} es de ${clima.temperatura}°C y está ${clima.condicion}.`);

} catch (error) {

    console.error("Error al consultar clima:", error);

}

}

consultarClima("Buenos Aires");

**5. Renderizado de Listas**

**Ejercicio 5.1 – Lista de Tareas**

async function mostrarTareas() {

let res = await fetch('https://jsonplaceholder.typicode.com/todos?\_limit=10');

let tareas = await res.json();

let ul = document.getElementById("lista-tareas");

ul.innerHTML = '';

tareas.forEach(tarea => {

    let li = document.createElement("li");

    li.textContent = tarea.title;

    if (tarea.completed) li.classList.add("completada");

    ul.appendChild(li);

});

}

mostrarTareas();

**Ejercicio 5.2 – Galería de Gatos**

async function cargarGatos() {

let res = await fetch('https://api.thecatapi.com/v1/images/search?limit=9');

let gatos = await res.json();

let galeria = document.getElementById("galeria-gatos");

galeria.innerHTML = '';

gatos.forEach(gato => {

    let img = document.createElement("img");

    img.src = gato.url;

    img.style.width = "100%";

    galeria.appendChild(img);

});

}

cargarGatos();

**6. Interactividad con Usuario**

**Ejercicio 6.1 – Buscador Rick and Morty**

document.getElementById("buscar-btn").addEventListener("click", async () => {

let nombre = document.getElementById("nombre-personaje").value.trim();

let contenedor = document.getElementById("resultado-busqueda");

contenedor.innerHTML = "Buscando...";

try {

    let res = await fetch(`https://rickandmortyapi.com/api/character/?name=${nombre}`);

    let datos = await res.json();

    contenedor.innerHTML = '';

    datos.results.forEach(personaje => {

    let card = document.createElement("div");

    card.innerHTML = `<h4>${personaje.name}</h4><img src="${personaje.image}" width="100">`;

    contenedor.appendChild(card);

    });

} catch (err) {

    contenedor.innerHTML = "No se encontraron personajes con ese nombre.";

}

});

**Ejercicio 6.2 – Generador de Consejos**

async function obtenerConsejo() {

let res = await fetch("https://api.adviceslip.com/advice");

let data = await res.json();

document.getElementById("consejo-texto").textContent = data.slip.advice;

}

document.getElementById("nuevo-consejo-btn").addEventListener("click", obtenerConsejo);

window.addEventListener("DOMContentLoaded", obtenerConsejo);