**Ejercicio Sencillo: Consulta a una API en React**

Voy a crear un componente React simple que consume la API de Pokémon (una API libre y popular) para mostrar información e imágenes de un Pokémon.

**Código completo**

import React, { useState, useEffect } from 'react';

function PokemonApp() {

const [pokemon, setPokemon] = useState(null);

const [loading, setLoading] = useState(false);

const [pokemonId, setPokemonId] = useState(1);

useEffect(() => {

const fetchPokemon = async () => {

setLoading(true);

try {

const response = await fetch(`https://pokeapi.co/api/v2/pokemon/${pokemonId}`);

const data = await response.json();

setPokemon(data);

} catch (error) {

console.error("Error fetching Pokemon:", error);

} finally {

setLoading(false);

}

};

fetchPokemon();

}, [pokemonId]);

const handleNext = () => {

setPokemonId(prev => prev + 1);

};

const handlePrev = () => {

if (pokemonId > 1) {

setPokemonId(prev => prev - 1);

}

};

return (

<div style={{ padding: '20px', maxWidth: '500px', margin: '0 auto' }}>

<h1>Consulta API Pokémon</h1>

<div style={{ margin: '20px 0' }}>

<button onClick={handlePrev} disabled={pokemonId <= 1}>Anterior</button>

<span style={{ margin: '0 10px' }}>ID: {pokemonId}</span>

<button onClick={handleNext}>Siguiente</button>

</div>

{loading ? (

<p>Cargando...</p>

) : pokemon ? (

<div style={{ border: '1px solid #ccc', padding: '15px', borderRadius: '8px' }}>

<h2>{pokemon.name.toUpperCase()}</h2>

<img

src={pokemon.sprites.front\_default}

alt={pokemon.name}

style={{ width: '200px', height: '200px' }}

/>

<div>

<h3>Habilidades:</h3>

<ul>

{pokemon.abilities.map((ability, index) => (

<li key={index}>{ability.ability.name}</li>

))}

</ul>

</div>

<div>

<h3>Estadísticas:</h3>

<ul>

{pokemon.stats.map((stat, index) => (

<li key={index}>

{stat.stat.name}: {stat.base\_stat}

</li>

))}

</ul>

</div>

</div>

) : (

<p>No se encontró el Pokémon</p>

)}

</div>

);

}

export default PokemonApp;

**Explicación paso a paso:**

1. **Estado del componente**:
   * pokemon: Almacena los datos del Pokémon obtenidos de la API
   * loading: Indica si está cargando los datos
   * pokemonId: El ID del Pokémon a consultar
2. **useEffect**:
   * Se ejecuta cuando cambia pokemonId
   * Hace una petición fetch a la API de Pokémon
   * Actualiza el estado con la respuesta
3. **Manejo de eventos**:
   * handleNext y handlePrev cambian el ID del Pokémon
4. **Renderizado**:
   * Muestra controles para navegar entre Pokémon
   * Muestra el nombre, imagen, habilidades y estadísticas del Pokémon
   * Maneja estados de carga y errores

**Cómo usar este componente:**

1. Crea un nuevo proyecto React con create-react-app o tu herramienta preferida
2. Copia este código en un archivo (por ejemplo PokemonApp.js)
3. Importa y usa el componente en tu aplicación principal

Este ejemplo es perfecto para principiantes porque:

* Usa una API pública y bien documentada
* Muestra datos e imágenes
* Implementa loading states
* Maneja navegación básica
* Es fácil de modificar y extender