

UNIVERSIDAD ESTATAL A DISTANCIA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
INGENIERÍA INFORMÁTICA

TAREA No.2

03101 – Programación avanzada en web

Alumno: Christian Céspedes Cubillo

Octubre 2024

Introducción

Con el presente trabajo se busca recopilar información valiosa para profundizar sobre la temática de los API's, conocer su funcionalidad y dar los primeros pasos en la aplicación, por medio de la puesta en práctica de un ejemplo sencillo.

Al finalizar la investigación se espera tener un conocimiento más amplio sobre algunas de las API's más conocidas y los beneficios de saber utilizarlas y con el desarrollo de la aplicación de prueba, nos permita interiorizar aún más los conceptos y su funcionamiento para el momento de realizar un sistema más robusto, podamos estar más familiarizados con esta herramienta.

1. Justificación de la elección del API:

¿Por qué API REST?:

Luego de la investigación que pude realizar, la API que he seleccionado para estudiar más a profundidad y utilizar para realizar el ejercicio de esta tarea es REST, ya que por lo leído, puedo interpretar que los sistemas que la utilizan pueden escalar de manera más eficiente gracias a las interacciones entre cliente y servidor, además es el API más utilizado, esto puede simplificar el desarrollo del sistema, al facilitar el acceso a información.

Beneficios de utilizar REST:

Puede resumir que la elección del API Rest se basa en los siguientes beneficios que se puede obtener al utilizarla:

- 1- Es ligera, al ser basada en el estándar HTTP quiere decir que no tiene formato y puede ser utilizado XML, JSON, HTML, entre otros, esto hace las aplicaciones livianas y veloces.
- 2- Es independiente el cliente y el servidor, es decir el almacenamiento de datos y la interfaz de usuario son separadas del servidor, esto quiere decir que desarrolladores pueden desarrollar en diferentes áreas del proyecto independientemente y probar ambos entornos.
- 3- Su escalabilidad y flexibilidad. Al estar separado del cliente y el servidor, se puede escalar velozmente y el programador pueden integrar las API Rest sin mucho trabajo adicional.
- 4- Son eficientes, de alto rendimiento, el consumo de ancho de banda es bajo y son muy rentables, ya que los desarrolladores las puede utilizar sin depender de terceros.

Para la elaboración de la práctica seleccione la API llamada PokeApi, ya que dentro de las que pude ver, fue una de las que pude tener mayor acceso, ya que al principio había seleccionado algunas que cuando hacía el llamado, no podía utilizarla por algunos permisos o porque no estaba habilitado. En mi caso, me pareció acertado esta selección, ya que pude encontrar mayor guía a la hora de realizar la programación de la sintaxis para acceder a la información de la API. Quizás, no sea una API muy funcional en cuanto a su uso para sistemas más robustos o serios, pero para comenzar aprender del tema, me pareció muy acertado.

2. Investigación preliminar

Las API RESTful son adecuadas para escenarios que buscan mayor escalabilidad, simplicidad y más interoperabilidad. Al utilizarlas, genera una gran flexibilidad en la recuperación de datos.

Las API RESTful, son ideales para las operaciones de crear, leer, actualizar y eliminar.

Leer (GET): recupera la representación del recurso en un URI especificado.

Actualizar (PUT): actualiza un recurso en un URI especificado.

Crear (POST): crear un nuevo recurso, el servidor genera el nuevo URI del nuevo objeto y devuelve como parte de la respuesta el URI

Eliminar (DELETE): elimina el recurso especificado

El formato de intercambio de datos que se utiliza en REST, es un formato ligero y fácil de leer, este formato es conocido como JSON (JavaScript Object Notation) y se utiliza para estructurar los datos en formato texto y permite el intercambio de la información entre aplicaciones de manera fácil, liviana y veloz.

3. Programación utilizando API:

Creamos el componente API.jsx:

Para iniciar creamos el componente API, donde vamos a importar useState y useEffect de react. Se crea la función llamada Pokemon_API, donde vamos a generar las constantes donde guardaremos la información que vamos a obtener de la API que utilizamos.

Por medio del método Fetch, accedamos el PokeApi y obtenemos para cada uno los endpoints que me interesa mostrar que son: el nombre, id, experiencia, estatura y peso, así como su imagen respectiva.

Por medio de la creación de dos botones (Anterior y Siguiente), nos movilizamos dentro de los diferentes Pokemons para ver su información.

```

App.js*  API.jsx*  Pokemon_API
Varios  Pokemon_API  Siguiente

1  import React, { useState,useEffect } from 'react'
2  3 referencias
3  function Pokemon_API() {
4      const [data, setData] = useState("");
5      const [id, setId] = useState(1);
6
7      const Siguiente = () => {
8          setId(id + 1);
9      }
10
11     const Anterior = () => {
12         if (id > 1)
13         {
14             setId(id - 1)
15         }
16     }
17
18     useEffect(() => {
19         fetch("https://pokeapi.co/api/v2/pokemon/" + id)
20             .then((Response) => Response.json())
21             .then((data) => setData(data))
22     }, [id]);
23
24
25     if (data) {
26         return (
27             <div>{
28                 data &&
29                 <div>
30                     <center>
31                         <hr></hr>
32                         <h1>Ejemplo de llamado a API (PokeAPI)</h1>
33                         <h3>Puedes avanzar con el boton SIGUIENTE o ANTERIOR para conocer los diferentes personajes Pokemon</h3>
34                         <h2>{data.name}</h2>
35
36                         <img src={data.sprites.front_default} alt={data.name} />
37                         <h2>Id: {data.id}</h2>
38                         <h2>Experiencia:{data.base_experience}</h2>
39                         <h2>Peso:{data.weight}</h2>
40                         <h2>Altura:{data.height}</h2>
41                         <br />
42                         {
43                             id > 1 ? <button onClick={Anterior}> ANTERIOR </button> : <button disabled> ANTERIOR </button>
44                         }
45                         <button onClick={Siguiente}> SIGUIENTE </button>
46                     </center>
47                 </div>
48             </div>
49         )
50     }
51 }
52
53 export default Pokemon_API;

```

Desde el APP.js se hace el import del nuevo componente

```

1
2  import Pokemon_API from './Componentes/API';
3  import Counter from './Componentes/Counter';
4  import { Encabezado } from './Componentes/Encabezado/Encabezado';
5  import { Usuario } from './Componentes/Usuario';
6
7
8  function App() {
9
10     return (
11         <div>
12             <Encabezado />
13             <Usuario nombre='Christian Cespedes Cubillo' cedula={110970383} materia='03101 Programacion Avanzada en Web' />
14             <Counter />
15             <Pokemon_API />
16         </div>
17     );
18 }
19 export default App;

```

Y el resultado es el siguiente:

UNIVERSIDAD ESTATAL A DISTANCIA

Ejercicio Tarea Programada N1 y N2

Practica de componentes, CSS, Estados con React y API

Alumno: Christian Cespedes Cubillo

Cedula: 110970383

Materia: 03101 Programacion Avanzada en Web

Ejemplo de contador utilizando useState (estado)

0

SUMAR | RESTAR

Ejemplo de llamado a API (PokeAPI)

Puedes avanzar con el boton SIGUIENTE o ANTERIOR para conocer los diferentes personajes Pokemon

bulbasaur



Id: 1

Experiencia:64

Peso:69

Altura:7

ANTERIOR | SIGUIENTE

Ejemplo de llamado a API (PokeAPI)

Puedes avanzar con el boton SIGUIENTE o ANTERIOR para conocer los diferentes personajes Pokemon

ivysaur



Id: 2

Experiencia:142

Peso:130

Altura:10

ANTERIOR | SIGUIENTE

Referencias:

Astera, Yunay Baig. (2024, Junio 27). *Una guía completa de los diferentes tipos de API*. <https://www.astera.com/es/type/blog/types-of-apis/>

Mulesoft. (2024). *Los 3 principales beneficios de las API REST*. https://www-mulesoft-com.translate.goog/resources/api/top-3-benefits-of-rest-apis?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq#:~:text=1.-,Lightweight,of%20things%20devices%2C%20and%20more.