Ejercicios async/await

Ejercicio 1: Obtener Información de Usuarios con async/await

Enunciado:
Crea una función obtenerUsuarios utilizando async/await para obtener la lista de usuarios desde https://jsonplaceholder.typicode.com/users. Muestra en la consola el nombre y correo de cada usuario.

Requisitos:

1. Usa try/catch para manejar errores.
2. Muestra un mensaje en consola si ocurre un error durante la solicitud.

Ejercicio 2: Publicar Datos con async/await

Enunciado:
Crea una función publicarPost que envíe un nuevo post a https://jsonplaceholder.typicode.com/posts usando el método POST. Los datos a enviar deben incluir un título y un cuerpo.

Requisitos:

1. Utiliza async/await para realizar la solicitud.
2. Muestra la respuesta del servidor en la consola.
3. Usa try/catch para capturar y mostrar errores.

Ejercicio 3: Encadenar Múltiples Solicitudes con async/await

Enunciado:
Crea una función que primero obtenga una lista de usuarios desde https://jsonplaceholder.typicode.com/users, y luego, para el primer usuario, obtenga sus publicaciones desde https://jsonplaceholder.typicode.com/posts?userId=1. Muestra el nombre del usuario y los títulos de sus publicaciones.

Requisitos:

1. Usa async/await para ambas solicitudes.
2. Maneja errores con try/catch.

Ejercicio 4: Esperar Varias Promesas con async/await y Promise.all

Enunciado:
Crea una función que obtenga datos de las siguientes URLs en paralelo:

1. https://jsonplaceholder.typicode.com/posts
2. https://jsonplaceholder.typicode.com/users
3. https://jsonplaceholder.typicode.com/comments

Muestra en la consola la cantidad total de publicaciones, usuarios y comentarios.

Requisitos:

1. Usa Promise.all junto con async/await.
2. Maneja errores con try/catch.

Ejercicio 5: Actualización de Datos con PUT y async/await

Enunciado:
Crea una función actualizarUsuario que actualice el nombre y correo del usuario con ID 1 en https://jsonplaceholder.typicode.com/users/1. Cambia el nombre a "Nuevo Nombre" y el correo a "nuevo@correo.com".

Requisitos:

1. Utiliza el método PUT y async/await.
2. Muestra la respuesta del servidor en la consola.
3. Maneja errores con try/catch.

Ejercicio 6: Eliminar Datos con DELETE y async/await

Enunciado:
Crea una función eliminarPost que elimine el post con ID 1 en https://jsonplaceholder.typicode.com/posts/1.

Requisitos:

1. Usa el método DELETE con async/await.
2. Muestra un mensaje en consola confirmando la eliminación.
3. Maneja errores con try/catch.

Ejercicio 7: Manejo de Errores HTTP con async/await

Enunciado:
Realiza una solicitud a una URL inexistente (https://jsonplaceholder.typicode.com/invalid-url). Captura el error y muestra un mensaje personalizado que indique que la página no fue encontrada.

Requisitos:

1. Usa try/catch para detectar el error.
2. Muestra un mensaje de "Página no encontrada" si ocurre un error.

Ejercicio 8: Uso de Parámetros en la URL con async/await

Enunciado:
Crea una función que obtenga los comentarios de la publicación con ID 1 usando https://jsonplaceholder.typicode.com/comments?postId=1. Muestra en la consola el nombre del autor y el contenido de cada comentario.

Requisitos:

1. Muestra los comentarios en la consola.
2. Maneja errores con `try/catch`.

Ejercicio 9: Renderizado Dinámico de Datos con async/await

Enunciado:
Crea una aplicación que obtenga una lista de álbumes desde https://jsonplaceholder.typicode.com/albums y muestre los títulos de los primeros 10 álbumes en una lista HTML (<ul>).

Requisitos:

1. Inserta los títulos en el DOM dentro de una lista desordenada (<ul>).
2. Maneja errores mostrando un mensaje en la página.

Ejercicio 10: Simulación de Carga Progresiva con async/await

Enunciado:
Crea una función que obtenga fotos desde https://jsonplaceholder.typicode.com/photos y muestre las fotos una a una cada segundo. Muestra el título de cada foto mientras se carga.

Requisitos:

1. Usa async/await para obtener los datos.
2. Utiliza setTimeout para simular la carga progresiva.
3. Maneja errores mostrando un mensaje adecuado.