**Implementación de async/await**

**Tabla de contenido**

* Instrucciones
* Parte 1: Fetch de Usuarios y Renderizado en Cards
* Ejemplo de Estructura de la Card:
* Parte 2: Manejo de Errores
* Parte 3: Estilizado con CSS y Bootstrap
* Recursos Utilizables

**Entregables**

El objetivo de esta práctica es comprender y aplicar los conceptos de funciones asíncronas, async y await, en JavaScript. Utilizarás la API pública de JSONPlaceholder para obtener datos de usuarios y mostrarlos dinámicamente en una card.

**Instrucciones**

Lee el siguiente artículo sobre async y await en FreeCodeCamp:

[Explicación de async y await en JavaScript](https://www.freecodecamp.org/espanol/news/explicando-async-y-await-en-javascript-con/)

Utiliza la API pública de JSONPlaceholder para obtener los usuarios:

URL de la API: *https://jsonplaceholder.typicode.com/users*

**Parte 1: Fetch de Usuarios y Renderizado en Cards**

Crea una función asíncrona llamada fetchUsers() que haga un fetch a la URL *https://jsonplaceholder.typicode.com/users* para obtener la lista de usuarios.

Utiliza async y await para esperar a que los datos se obtengan y luego procesarlos.

Por cada usuario obtenido, crea dinámicamente una card utilizando JavaScript. La card debe incluir al menos los siguientes datos del usuario:

Nombre (name)

Nombre de usuario (username)

Correo electrónico (email)

Empresa (company.name)

Muestra todas las cards en el DOM (por ejemplo, en una sección con la clase .users-container).

**Ejemplo de Estructura de la Card:**

<div class="card">

<h3>Nombre: John Doe</h3>

<p>Usuario: johndoe</p>

<p>Email: johndoe@example.com</p>

<p>Empresa: Doe Inc.</p>

</div>

**Parte 2: Manejo de Errores**

Asegúrate de manejar los posibles errores de la API utilizando un bloque try/catch. Si ocurre un error al obtener los datos, muestra un mensaje de error en el DOM, indicando que no se pudieron cargar los usuarios.

El mensaje de error debe ser claro, por ejemplo: “Error al cargar los usuarios. Inténtalo de nuevo más tarde.”

**Parte 3: Estilizado con CSS y Bootstrap**

Utiliza Bootstrap para diseñar las cards de manera responsiva. Cada card debe tener un diseño limpio y estructurado, utilizando las clases de Bootstrap.

Asegúrate de que el diseño sea responsivo, adaptándose correctamente a diferentes tamaños de pantalla.

**Recursos Utilizables**

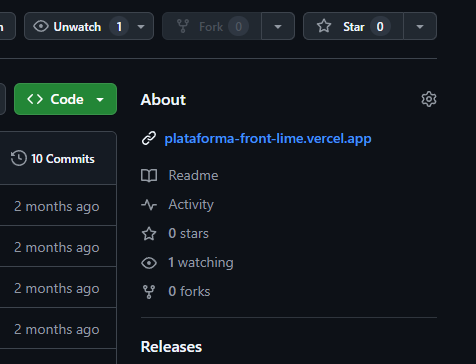
URL de la API de usuarios: https://jsonplaceholder.typicode.com/users

Artículo de referencia: Explicación de async y await en FreeCodeCamp

Documentación de JSONPlaceholder

**Entregables**

Para la entrega de esta práctica calificada, deberás subir el link a tu repositorio de GitHub, el cual debe contener el código del proyecto. Además, debe incluir el link a la página desplegada en el apartado “About” del repositorio. Como se muestra a continuación:

alt text