Async Await

En esta sección, reescribiremos el mismo código de la sección anterior utilizando la sintaxis de **async** y **await** en JavaScript.

Uso de async y await

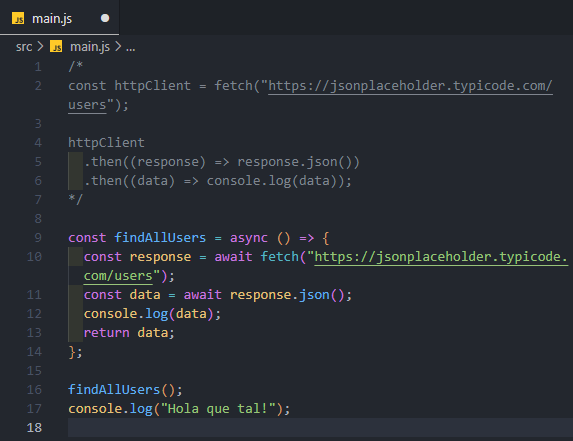
Los términos **async** y **await**. son características de JavaScript que permiten trabajar con promesas de una manera más limpia y legible.

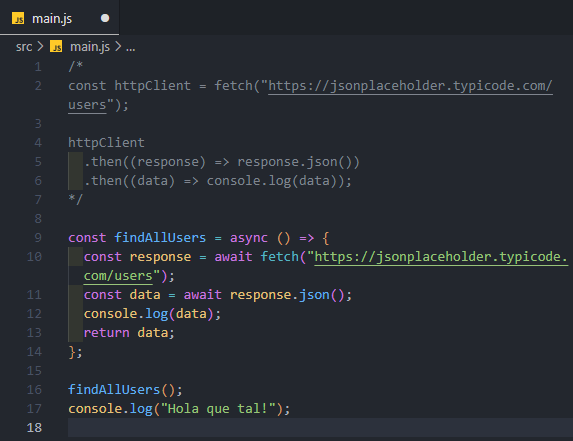
* **async**: Se utiliza para declarar que una función va a retornar una promesa.
* **await**: Se utiliza dentro de funciones marcadas como **async** para esperar a que una promesa se resuelva.

Aunque las operaciones se ejecutan en paralelo y tienen que manejar la comunicación con el servidor y el backend, estos métodos hacen que el código parezca más lineal.

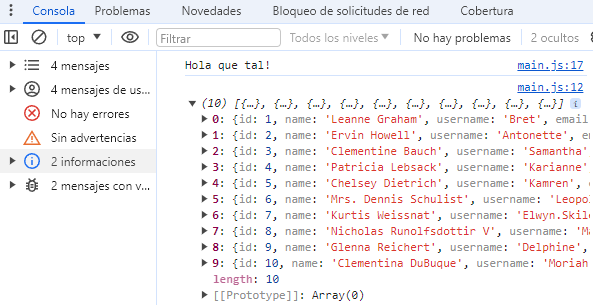
En el siguiente código, la función **findAllUsers** es una función asincrónica (**async**) que realiza una solicitud HTTP GET a **https://jsonplaceholder.typicode.com/users** con **fetch**. Luego, espera la respuesta y lo convierte a JSON con **await**. Finalmente, imprime los datos en la consola y los retorna.

Al llamar a **findAllUsers**, se ejecuta la función y se imprime **"Hola que tal!"** en la consola.





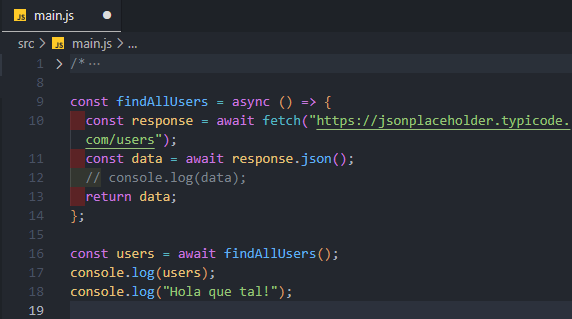
En resumen, al llamar a **findAllUsers**, la solicitud HTTP y la conversión a JSON se realizan de manera asincrónica, permitiendo que el mensaje **"Hola que tal!"** se imprima en la consola inmediatamente, antes de que se completen las operaciones asincrónicas dentro de **findAllUsers**.

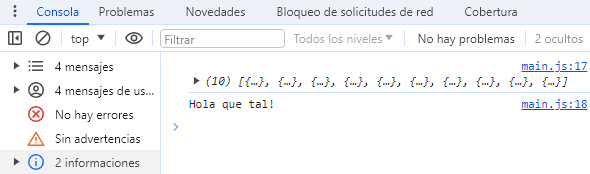


A pesar de que el código es asíncrono y se imprime primero **"Hola que tal"**, se ve más lineal con **await**. Resuelve y obtiene los datos sin necesidad del método **then**. El **await** se maneja en segundo plano en una tarea de procesamiento en paralelo que está pendiente y se resuelve cuando se pasa a la constante en **fetch**.

Función que devuelve una promesa

Si queremos que la función **findAllUsers** devuelva una promesa para manejarla y procesarla, podemos hacer lo siguiente:

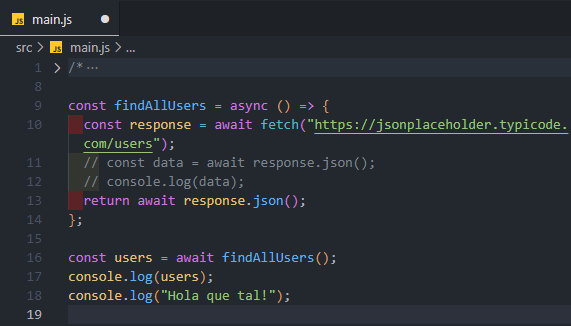


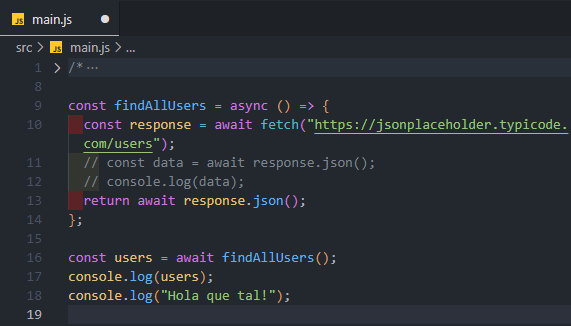


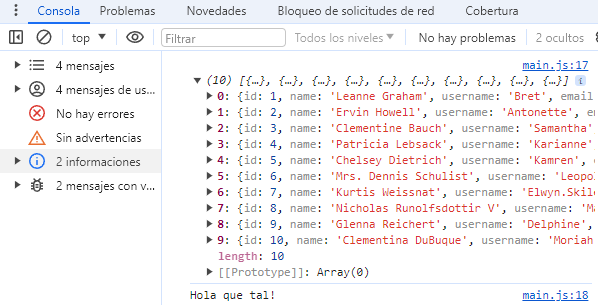
En este caso, se llama a la función **findAllUsers** utilizando **await** para esperar a que la promesa se resuelva y se asigna el resultado a la variable **users**.

Simplificar el código

El código se puede simplificar aún más, eliminando las variables intermedias:







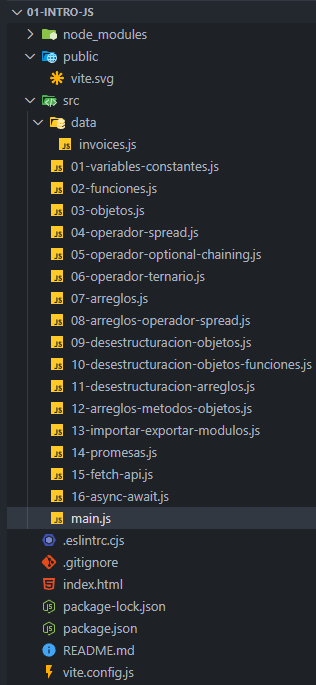
Antes de pasar al siguiente tema

Realiza estas acciones en Visual Studio Code:

1. Abre el panel izquierdo y busca el archivo **main.js**.
2. Haz clic derecho en el archivo y selecciona **Cambiar nombre**.
3. Cambia el nombre del archivo a **16-async-await.js** y presiona **Enter**.
4. Ahora, crea un nuevo archivo, dentro la carpeta raíz **src**, haciendo clic en el botón **Nuevo archivo** en la parte superior del panel izquierdo.



1. Nombre el nuevo archivo como **main.js** y presiona **Enter**.



A partir de este momento, se realizarán cambios en el archivo **main.js** que has creado.