Universidad Politécnica de Querétaro

Ingeniería en Tecnologías de la Información e Innovación Digital

Grupo: TIID-214

Desarrollo de Aplicaciones Móviles

Iván Isay Guerra López

Reporte 2 Import & Asincronía

Integrantes

Galarza Piña Ruth Verónica

25 - 09 – 2025

1. Se hizo la función pedida para la resta de la función

```
JS utils.js > ...
1   function restar(a, b) {
2       return a - b;
3   }
4
```

En un nuevo archivo creamos las opciones para comprobar si realmente funciona

```
JS main.js
1   console.log(restar(10, 5));
2   console.log(restar(7, 3));
3   console.log(restar(20, 4));
4
```

Una vez hecho lo anterior se instaló una extensión en VS de live server para que se ejecutara correctamente, una vez hecho el main daremos click derecho y ahí vamos a open with live server, y nos aparece algo como lo siguiente

~ / main.js [fullscreen]

```
const restar = require("./utils");// Aqui se utiliza el "./utils" para

// Pruebas, las puse asi como en ejercicios vistos anteriormente

console.log(restar(10, 5));
console.log(restar(7, 3));
console.log(restar(20, 4));
```

Después debería ejecutar lo realizado, no me salió, investigue en varias paginas, pero no logre realizarlo, abajo puse los links consultados.

```
▶ Uncaught ReferenceError: require is not defined   main.js:1
▶ Uncaught SyntaxError: Identifier 'restar' has     utils.js:1
already been declared
```

2. Se hizo una promesa para resolver esta problemática, se utiliza new promise para que funcione de forma correcta, después de esto tenemos que poner dos opciones importantes “resolve” que se utiliza cuando la condición se cumple y “reject” para cuando no se cumple, aquí utilizamos un if ya que si el usuario es idéntico a “admin” nos mostrara “Acceso concedido” en caso de que no se cumpla esta condición pasarla al paramento de reject, el cual nos dirá acceso denegado, todo lo demás ya nada mas es la verificación, es decir es lo que nos mostrara el código, por así decirlo.

```
function verificarUsuario(usuario) {
 return new Promise ((resolve, reject) =>{
     if(usuario === "admin"){
        resolve("Acceso concedido");
     }else{
        reject("Acceso denegado");
     }
 });
}

// Usa .then() y .catch() para manejar el resultado
verificarUsuario("admin")
  .then(res => console.log(res)) // Acceso concedido
  .catch(err => console.error(err));

verificarUsuario("Ivan")
  .then(res => console.log(res)) // Acceso denegado
  .catch(err => console.error(err));
```

Una vez guardado esto nos aparecerá en la consola de nuestro navegador lo que se dice en el código.

3.Para finalizar vamos a hacer una supuesta llamada a API con la función de async/await, la llamamos obtener datos, como podemos ver esto es una forma mas simplificadad pr asi decirlo de una promesa, por eso podemos ver que se utiliza una, async siempre nos va a devolver la promesa y await como su nombre mas o menos lo dice, o asi entendi yo pausa la ejecucucion de la promesa hasta que termine, entonces llamamos la funcion ontener datos, en el await de

simularPeticionAPI nos devuelve a promesa con el temporizador,

```
function simularPeticionAPI() {
  return new Promise(resolve => {
    setTimeout(() => {
      resolve("Datos recibidos correctamente");
    }, 5000);
  });
}

async function obtenerDatos() {
 let resultado= await simularPeticionAPI();
 console.log(resultado);
}
obtenerDatos();
// Usa la funcion async
```

Aquí podemos ver la ejecucucion, este es casi lo mismo que las promesas tradicionales, pero con una sintaxis mas limpia por asi decirlo, nos ahorra codigo cuando son promesas grandes

```
Datos recibidos correctamente                Ejercicio3.js:11
>
```

Conclusión:

Puse a prueba mis conocimientos vistos en las exposiciones, la verdad no entendí casi nada el 1er ejercicio, se pusieron en practica el uso de promesas y el como usar su forma simplificada que en este caso es el async/away

Links Consultados:

https://share.google/xJuOMksGSabdzveiE

https://www.freecodecamp.org/news/module-exports-how-to-export-in-node-js-and-javascript/

https://share.google/4S5CcGQiJLGCmdKsi

Link Git