



# Promesas, async/await y Manejo de Errores en JavaScript

## Entrega de ejercicios

Recuerda que la entrega de los ejercicios se hará a través de un repositorio en GitHub. Cuando lo tengas listo, comparte el enlace del repositorio por Discord directamente con las formadoras.

### EJERCICIO 1: Promesa básica con .then()


 **Objetivo: Entender cómo funciona una promesa simple.**

 Paso a paso:

1. Crea un archivo llamado script.js en tu proyecto.
2. Escribe la función 'crearSaludo' que devuelva una promesa que se resuelve tras 1 segundo.
3. Llama a esa función pasándole tu nombre como argumento.
4. Usa .then() para recibir la respuesta y mostrarla en la consola.
5. Abre la consola del navegador y comprueba que aparece el mensaje 'Hola, [tu nombre]'.


### EJERCICIO 2: Rechazo con .catch()


 **Objetivo: Manejar errores cuando una promesa se rechaza.**

 Paso a paso:

1. Escribe una función 'calcularEdad' que reciba un nombre y devuelva una promesa.
2. La promesa debe rechazarse si el nombre es 'Desconocido'.
3. Llama a la función con el nombre 'Desconocido'.
4. Usa .then() y .catch() para mostrar el resultado o el error en consola.
5. Comprueba que el error aparece correctamente.


### EJERCICIO 3: Usar async y await


 **Objetivo: Simplificar una promesa usando async/await.**

 Paso a paso:

1. Escribe una función 'buscarUsuario' que devuelva una promesa con un mensaje tras 1.5 segundos.
2. Crea una función async llamada 'mostrarUsuario'.
3. Dentro de ella, usa await para esperar el resultado de obtenerUsuario.
4. Muestra el mensaje en consola.
5. Llama a 'mostrarUsuario' y revisa el resultado en consola.

### EJERCICIO 4: Encadenar varias promesas con .then()

 **Objetivo: Practicar cómo conectar varias promesas entre sí.**


 Paso a paso:

1. Crea tres funciones llamadas pasoUno, pasoDos y pasoTres, que devuelvan promesas resueltas con distintos mensajes.

2. Llama primero a `pasoUno` y luego encadena `pasoDos` y `pasoTres` con `.then()`.
3. Después de cada `.then()`, imprime el resultado en consola.
4. Comprueba que los mensajes aparecen en el orden correcto.

### EJERCICIO 5: Manejo de errores con `try / catch`

 **Objetivo: Atrapar errores dentro de una función `async`.**

 Paso a paso:

1. Escribe una función `dividir` que devuelva una promesa: si `b` es 0, rechaza con un error.
2. Crea una función `async` `hacerDivision`.
3. Dentro, usa `try/catch` para capturar el error si existe.
4. Usa `await` para llamar a `dividir(10, 0)` y muestra el resultado o el error.
5. Comprueba que el error aparece correctamente en consola.