



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

ADSO

Taller: Explorando el Mundo Asíncrono con Node.js

Propósito del taller

Aprender a manejar la programación asíncrona en Node.js usando callbacks, promesas, async/await, ejecuciones paralelas y control de tiempo.

Sección 1: Introducción a la Asincronía

Ejemplo 1: Lectura de archivo con Callback

```
const fs = require('fs');
fs.readFile('archivo1.txt', 'utf8', (error, datos) => {
  if (error) {
    console.error('Error:', error);
    return;
  }
  console.log('Contenido:', datos);
});
console.log('Mensaje asíncrono');
```

Ejercicio 1:

- Crea 3 archivos de texto.
- Escribe una función que los lea uno por uno usando callbacks.

Ejercicio 2:

- Modifica el ejercicio anterior para concatenar el contenido de los tres archivos y mostrarlo junto.

Sección 2: Uso de Promesas

Ejemplo 2: Lectura de archivo con Promesas

```
const fs = require('fs').promises;
```



```
fs.readFile('archivo1.txt', 'utf8')
  .then(data => console.log(data))
  .catch(err => console.error(err));
```

Ejercicio 3:

- Convierte el ejercicio 1 para que funcione usando promesas.

Ejercicio 4:

- Lee tres archivos con promesas y concatena su contenido.

✨ Sección 3: Async / Await

Ejemplo 3: Lectura con async/await

```
const fs = require('fs').promises;
async function leer() {
  try {
    const data = await fs.readFile('archivo1.txt', 'utf8');
    console.log(data);
  } catch (e) {
    console.error(e);
  }
}
leer();
```

Ejercicio 5:

- Convierte el ejercicio 4 para usar async/await.

Ejercicio 6:

- Agrega manejo de errores con try/catch y muestra un mensaje personalizado si falla la lectura.

Sección 4: Lectura en Paralelo

Ejemplo 4: Promise.all

```
const fs = require('fs').promises;
async function leerArchivos() {
```



```
try {
  const [a, b, c] = await Promise.all([
    fs.readFile('archivo1.txt', 'utf8'),
    fs.readFile('archivo2.txt', 'utf8'),
    fs.readFile('archivo3.txt', 'utf8')
  ]);
  console.log(a + b + c);
} catch (e) {
  console.error(e);
}
}
leerArchivos();
```

Ejercicio 7:

- Lee 5 archivos en paralelo y muestra cuántas palabras tiene cada uno.

Ejercicio 8:

- Guarda los resultados del ejercicio anterior en un archivo `informe.txt`.

Sección 5: Temporizadores y control de tiempo

Ejemplo 5: Temporizador con Promise

```
function esperar(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
```

Ejercicio 9:

- Escribe una función que reciba un arreglo de tiempos y los imprima uno por uno con ese retraso.

Ejercicio 10:

- Escribe una función que use `Promise.race()` para cancelar la lectura de un archivo si tarda más de cierto tiempo.

Proyecto Final: Procesador de Archivos Asíncrono

Requisitos:



- Leer archivos de un directorio.
- Filtrar solo los `.txt`.
- Contar palabras de cada archivo.
- Guardar un resumen en `informe.txt`.

Formato del informe:

Archivo: `archivo1.txt` - Palabras: 35
Archivo: `archivo2.txt` - Palabras: 40

