



# Ex JS - Asincronía & Promesas

## Investigación

### 1. ¿Qué es un archivo JSON?

(Fuente: <https://rockcontent.com/es/blog/archivo-json/>)

En la práctica, .json es un archivo que contiene una serie de datos estructurados en formato de texto y se usa para transferir información entre sistemas. Es importante decir que, a pesar de su origen estar en el lenguaje JavaScript, JSON no es un lenguaje de programación.

JSON es una notación para la transferencia de datos que sigue un estándar específico. Por eso, puede emplearse en diferentes lenguajes de programación y de sistemas.

Los datos contenidos en un archivo en formato JSON deben estructurarse por medio de una colección de pares con nombre y valor o deben ser una lista ordenada de valores. Sus elementos tienen que contener:

- **Clave:** corresponde al identificador del contenido. Por eso, debe ser una string delimitada por comillas.
- **Valor:** representa el contenido correspondiente y puede contener los siguientes tipos de datos: string, array, object, number, boolean o null.

No tiene como objetivo la creación de programas

### 2. ¿Qué es la asincronía en Javascript?

Es una técnica que permite a tu programa iniciar una tarea de larga duración y seguir respondiendo a otros eventos mientras esa tarea se ejecuta, en lugar de tener que esperar hasta que esa tarea haya terminado.

### 3. ¿Qué son las Promesas?

Las promesas son como un contrato entre dos partes en la programación. Una parte hace una promesa de hacer algo, y la otra parte espera el resultado. Es como hacer un pedido en un restaurante: tú haces el pedido (promesa), y luego esperas a que te sirvan la comida (resultado). Por ejemplo, imagina que haces una solicitud a un servidor para obtener datos. En lugar de esperar bloqueado hasta recibir la respuesta, haces una promesa de manejar los datos cuando estén disponibles. Mientras tanto, puedes continuar realizando otras tareas. Cuando los datos llegan, la promesa se cumple y puedes manejarlos de acuerdo a tus necesidades. En resumen, las promesas son una forma de trabajar con operaciones asíncronas de manera más elegante y estructurada. Te permiten continuar con tu código mientras esperas resultados y manejarlos cuando estén listos.

(Fuente: <https://lenguajejs.com/javascript/asincronia/promesas/>)

Las promesas son un concepto para resolver el problema de asincronía de una forma mucho más elegante y práctica que, por ejemplo, utilizando funciones callbacks directamente.

### 4. ¿Qué es y cómo se usa 'Fetch' en JS?

Es un mecanismo que le permite realizar llamadas simples AJAX (JavaScript asíncrono y XML) con JavaScript. Proporciona una interfaz JavaScript para acceder y manipular partes del canal HTTP, tales como peticiones y respuestas

### 5. ¿Cómo se usa Async/Await?

Async/Await es una característica de JavaScript que permite escribir código asíncrono de forma más sencilla y con una apariencia similar a código síncrono.

Dentro de una función asíncrona, puedes utilizar la palabra clave await para esperar a que se complete una operación asíncrona. Esto hace que la ejecución de la función se detenga hasta que se resuelva la promesa o se complete la operación asíncrona.

### 6. ¿Cuándo deberíamos usar código asíncrono?

(Fuente: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Asynchronous/Introducing>)

La programación asíncrona es una técnica que permite a tu programa iniciar una tarea de larga duración y seguir respondiendo a otros eventos mientras esa tarea se ejecuta, en lugar de tener que esperar hasta que esa tarea haya terminado. Una vez que dicha tarea ha finalizado, tu programa presenta el resultado.

## **Presentación**

Libre para explicar y desarrollar los conceptos.

## Práctica

1.- Usa este JSON para imprimir en consola del navegador la lista de títulos de los 'Todos'.

```
{
  "results": [
    {
      "title": "Estudiar Js",
      "priority": "alta",
      "isDone": false
    },
    {
      "title": "Estudiar CSS",
      "priority": "alta",
      "isDone": true
    },
    {
      "title": "Estudiar OOP",
      "priority": "media",
      "isDone": false
    },
    {
      "title": "Estudiar IA",
      "priority": "baja",
      "isDone": false
    }
  ]
}
```

2.- Usa el mismo JSON para imprimir en el documento del navegador una tabla con todos los elementos 'Todos' y sus propiedades. (hacer con java script)

## **Nota**

Se puede utilizar Bootstrap,

## **Extras**

Mostrar los 'Todos' terminados y no terminados de forma gráfica.

## **Recursos**

<https://www.chartjs.org/>

[https://youtu.be/uSEdI\\_FhMWE](https://youtu.be/uSEdI_FhMWE)