

# Promise y Fetch

Módulo 04

# Operaciones asincrónicas

A yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing JavaScript.

# API Promise

La API Promise es una interfaz que me permite representar operaciones asincrónicas (ejecutadas en un proceso de carga distinto del principal de la página) que pueden ser potencialmente resueltas o rechazadas.

## API Fetch

La API Fetch implementa la API Promise en el manejo de las solicitudes HTTP. Permite realizar solicitudes AJAX al servidor. Mediante el evento 'fetch' podemos interceptar peticiones entrantes a nuestra página.

Fuente: [https://developer.mozilla.org/es/docs/Referencia\\_DOM\\_de\\_Gecko/Introducci%C3%B3n](https://developer.mozilla.org/es/docs/Referencia_DOM_de_Gecko/Introducci%C3%B3n)

# API Promise

## CREAR UNA PROMESA

Las promesas se crean mediante la Función Constructora *Promise*. Dicha función recibe como parámetro otra función cuyos argumentos son los siguientes:

- El primer argumento toma un callback que será ejecutado en caso de que la promesa esté resuelta, es decir, que la operación esté cumplida correctamente
- El segundo argumento toma un callback que será ejecutado en caso de que la promesa esté rechazada, es decir, que la operación no se haya podido concretar.

```
const delay = new Promise(  
  function(resolver, rechazar){  
    setTimeout(resolver, 3000);  
  });
```

Este código ejecuta el callback 'resolver' al cabo de 3 segundos

# API Promise

## USAR UNA PROMESA YA CREADA

Usar una promesa significa asignar los callbacks resolver y rechazar. En otras palabras: usar una promesa es definir que sucede en caso de que esté correcta o de que esté rechazada.

- El `.then()` asigna el callback que se ejecutará cuando la promesa se resuelva.
- El `.catch()` asigna el callback que se ejecutará en el caso en que la promesa se rechace

```
delay.  
  then(function(){  
    console.log("Tiempo cumplido");  
  })  
  .catch(function(){  
    console.log("Ocurrió un error");  
  })
```

Este código define, siguiendo el caso anterior, qué ocurrirá al cabo de 3 segundos (`.then()`) y muestra un mensaje en caso de error (`.catch()`)

# API Fetch

¿Qué es?

Es una interfaz que implementa la API Promise

¿Para que sirve?

Sirve para ejecutar operaciones HTTP asincrónicas (llamadas AJAX por ejemplo) de forma mucho más simple.

¿Cómo se usa?

- Request y Response. Estos dos objetos son la base de la API Fetch. Request representa una petición http (tiene cuerpo, cabeceras, ec.). Response representa una respuesta HTTP (tiene un cuerpo y, por ejemplo, podemos convertirla a json)
- fetch(). El método fetch() recibe como parámetro un Request y retorna una promesa que resuelve en un objeto Response.
- Evento “fetch”. Se dispara al interceptar una petición HTTP entrante. Lo usaremos para darle funcionalidad offline a nuestra aplicación

# ¡Muchas gracias!

¡Sigamos trabajando!