

DESARROLLO WEB FULL STACK

Promesas

Trabajando con operaciones asíncronicas


DWFS COR



¿Qué es una Promesa?

Una promesa representa un valor que puede estar disponible ahora, en el futuro, o nunca, representa la terminación o el fracaso eventual de una **operación asíncrona**.



A photograph of two men in an office environment, overlaid with a semi-transparent blue and purple gradient. The man in the foreground, on the right, is a young man with dark hair, wearing a striped polo shirt, pointing his right index finger towards a computer screen. The man in the background, on the left, has a beard and glasses, wearing a light-colored shirt, and is looking at the same screen. The background is slightly blurred, showing office furniture and plants.

¿Cómo se
implementa una
promesa?



Promesa

Resuelve o rechaza inmediatamente según el valor del parámetro de la función pedido.

```
const pedido = function(resuelvo) {  
  var promesa = new Promise(function(resolve, reject) {  
    if (resuelvo) {  
      resolve('Promesa resuelta correctamente.');    } else {  
      reject('Promesa no resuelta. :(');  
    }  
  });  
  
  return promesa;  
}
```



Promesa

Veo el valor devuelto por
mi operación asincrónica.

```
pedido(true).then(function(respuesta) {  
    console.log(respuesta);  
}).catch(function(error) {  
    console.log(error);  
});
```



Promesas

Agreguemos tiempo para simular que la operación tarda, lo hacemos con un **setTimeout**.

En la realidad no se usa setTimeout, ahí realmente tarda la operación.

```
const pedido = function(resuelvo) {  
  var promesa = new Promise(function(resolve, reject) {  
    setTimeout(function() {  
      if (resuelvo) {  
        resolve('Promesa resuelta correctamente.');      } else {  
        reject('Promesa no resuelta. :(');  
      }  
    }, 5000);  
  });  
  
  return promesa;  
};
```



Funcionalidades extras

- Encadenamiento
- Async - Await
- Bloque finally
- Métodos (all, race, allSettled)



Fetch API



Fetch

La API Fetch proporciona una interfaz JavaScript para hacer uso del protocolo HTTP, directamente desde el navegador.

Provee un método global `fetch()` que proporciona una forma fácil de obtener recursos de un servidor forma asíncrona.

Está basado en la API de Promesas.



Fetch

Usando una promesa y trayendo la información de una API.

En el ejemplo no se hace, pero los casos de **errores** son importantes que los manejemos usando el **catch()**.

```
fetch('https://api.chucknorris.io/jokes/random')  
  .then(function(respuesta) {  
    return respuesta.json();  
  }).then(function(json) {  
    console.log(json);  
  });
```



Fetch

Enviando los datos a una función que los procese.

```
function dibujar() {  
    // Preparar el DOM para mostrar visualmente  
}  
  
fetch('https://api.chucknorris.io/jokes/random')  
    .then(function(respuesta) {  
        return respuesta.json();  
    }).then(function(json) {  
        dibujar(json);  
    }).catch(function(error) {  
        console.log('Algo salió mal.');    });
```



A photograph of two men in an office environment. The man in the foreground, wearing a striped polo shirt, is pointing at a computer monitor. The man in the background, wearing glasses and a beard, is looking at the same screen. The image has a blue and purple color overlay.

Axios



Axios

Cliente HTTP basado en promesas para utilizar este protocolo de manera fácil. Se puede usar desde el navegador y también desde Node.js.

Es de desarrollo privado pero de código abierto, no está incluido en el estándar EcmaScript pero es el más usado por los desarrolladores web.

<https://github.com/axios/axios>



Axios

Haciendo un get a una api.

```
axios.get('https://api.chucknorris.io/jokes/random')  
  .then(function(response) {  
    console.log(response);  
  }).catch(function(error) {  
    console.log(error);  
  });
```



Refactorizar nuestra pokedex para en vez de usar XMLHttpRequest usemos **Fetch** o **Axios**.



squirtle

squirtle



Default character: true

Base experience: 63

Height: 5

Weight: 90

Abilities: rain-dish, torrent