

Universidad Autónoma de Ciudad Juárez  
Instituto de Arquitectura Diseño y Arte  
Departamento de Diseño



# Investigación de Async, Await y Promise en JavaScript

Hugo Inés Gallegos Gaucín

Matrícula: 154677

Diseño Digital de Medios Interactivos

Proyecto Integral Web

Maestro: Jesús Roberto Saucedo Franco

27 de febrero del 2019

## Async

La palabra `async` se utiliza para señalar que se está implementando una función asíncrona, la función asíncrona se diferencia porque esta no necesita esperar que se obtengan los datos para ejecutarse, en cambio se ejecuta y cuando se obtengan los datos esto se envía al usuario. (Stanciu, 2013) Esta función se ejecuta en un orden diferente al resto de las funciones, lo cual regresa `promise`, por lo cual están funciones se cumplen en el momento actual o después.

### Ejemplo

```
async function concurrentStart() {
  console.log('==CONCURRENT START with await==');
  const slow = resolveAfter2Seconds() // starts timer in
  const fast = resolveAfter1Second() // starts timer imm

  // 1. Execution gets here almost instantly
  console.log(await slow) // 2. this runs 2 seconds after
  console.log(await fast) // 3. this runs 1 second after
}
```

## Await

La palabra `await` se utiliza solo en las funciones asíncronas, su objetivo es pausar la ejecución de la función asíncrona a la espera de un `promise`, para después obtener un resultado. En caso de que el `promise` sea rechazado el `await` nos proporciona el valor rechazado, además de que el `await` nos permite dividir el flujo de ejecución del código. (“`await`,” 2020)

### Ejemplo

```
async function f1() {
  var x = await resolveAfter2Seconds(10);
  console.log(x); // 10
}

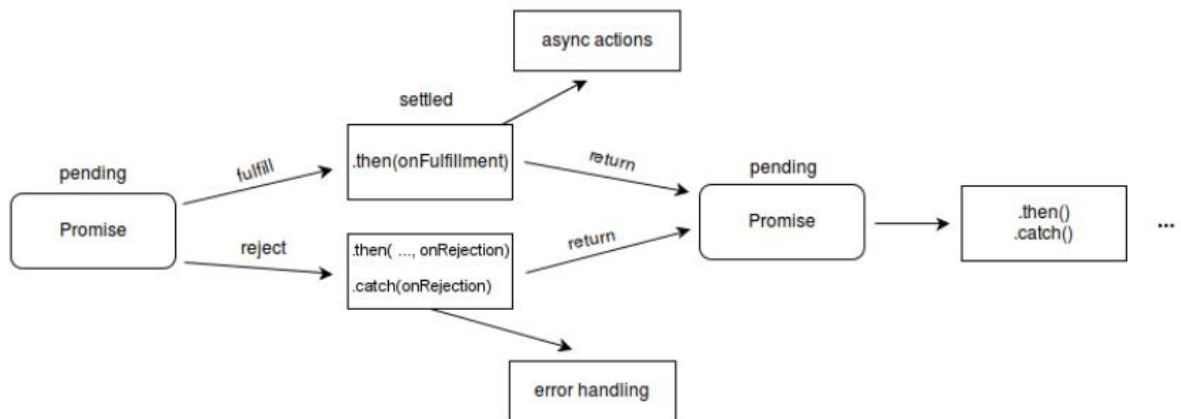
f1();
```

## Promise

Consiste en un objeto que nos permite representar el fin de ejecución de una función asíncrona, proporcionando un valor resultante de esta función. Los promise otorgan la capacidad de que los valores no sean regresados al momento, si no que estos puedan ser obtenidos en un tiempo futuro, ya sea que el valor resultante sea un éxito o una falla.

Así mismo, estos representan un proceso que está ocurriendo y que puede desencadenar una serie de devoluciones de llamada. Los promise cuenta con tres tipos de estado pendientes, cumplidos y rechazados. ("promise," 2020)

- Pendiente: Inicial, ni cumplido ni rechazado.
- Cumplido: Se completo la operación exitosamente.
- Rechazado: La operación resulto en una falla.



## Bibliografía

await. (2020). Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await>

promise. (2020). Retrieved from [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)

Stanciu, C. C. (2013). *CATALYDE: UN ENTORNO DE INICIACIÓN A LA PROGRAMACIÓN CONFINES EDUCATIVOS*.