



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

# **Diplomatura en programación web full stack con React JS**

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

p. 2

## **Módulo 2: JavaScript**

### **Unidad 3: Asincronismo**

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



## Presentación:

Algunas tareas pueden demorar más tiempo, por ejemplo la respuesta de un servidor, la escritura en un archivo, la impresión de un texto, etc. Como la programación en JavaScript se ejecuta línea tras línea de código, frente a sucesos que demoran, el sistema se quedaría “parado” esperando una respuesta. Sin embargo, JavaScript tiene un manejo especial para los procesos asíncronos (los que demoran más tiempo) y en esta unidad vamos a aprender acerca de ellos!



## Objetivos:

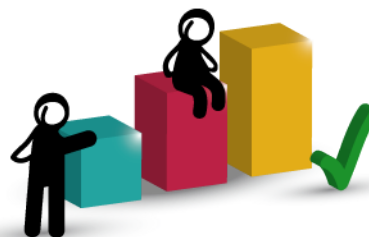
### Que los participantes:

- Comprendan el concepto de procesos asincrónicos.
- Sean capaces de identificar los mecanismos de manejo del asincronismo.



## Bloques temáticos:

1. Concepto de asincronismo
2. Callbacks
3. Promises
4. Async/Away
5. Ejemplo de asincronismo en JS
6. Trabajo Práctico



## Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

*\* El MEC es el modelo de E-learning colaborativo de nuestro Centro.*



## Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.



## 1. Concepto de asincronismo

### Operaciones asincrónicas

Las operaciones asincrónicas son aquellas que al llamarlas, la respuesta demora y/o no puede ser procesada en el momento. Las operaciones más comunes que son tratadas de esta forma son:

- Operaciones sobre archivos (solo para JavaScript del lado del servidor)
- Operaciones sobre bases de datos (solo para JavaScript del lado del servidor)
- Conexión y consulta a servidores externos

Las operaciones asincrónicas se pueden tratar de diferentes formas: callbacks, promises, async/await

### Procesos Sincrónicos

1. Son las 11PM y aún estoy programando para llegar con una entrega
2. Tengo hambre
3. Guardo los cambios del código (commit?)
4. Camino hasta la pizzería cerca de mi casa
5. Pido la pizza y espero que me la entreguen

### Procesos Asincrónicos

1. Son las 11PM y aún estoy programando para llegar con una entrega
2. Tengo hambre
3. Llamo a la pizzería y hago mi pedido





4. Continúo con mi desarrollo
5. Luego de un rato suena el timbre <- Evento
6. Guardo los cambios
7. Salgo, pago la pizza, me como la pizza

## **Cómo trabaja JavaScript**

En JavaScript todas las operaciones de E/S son asíncronas

- Conexión con un servidor remoto por medio de HTTP
- Timers
- Del lado del servidor (NodeJS)
  - Consultas a la base de datos
  - Operaciones sobre archivos

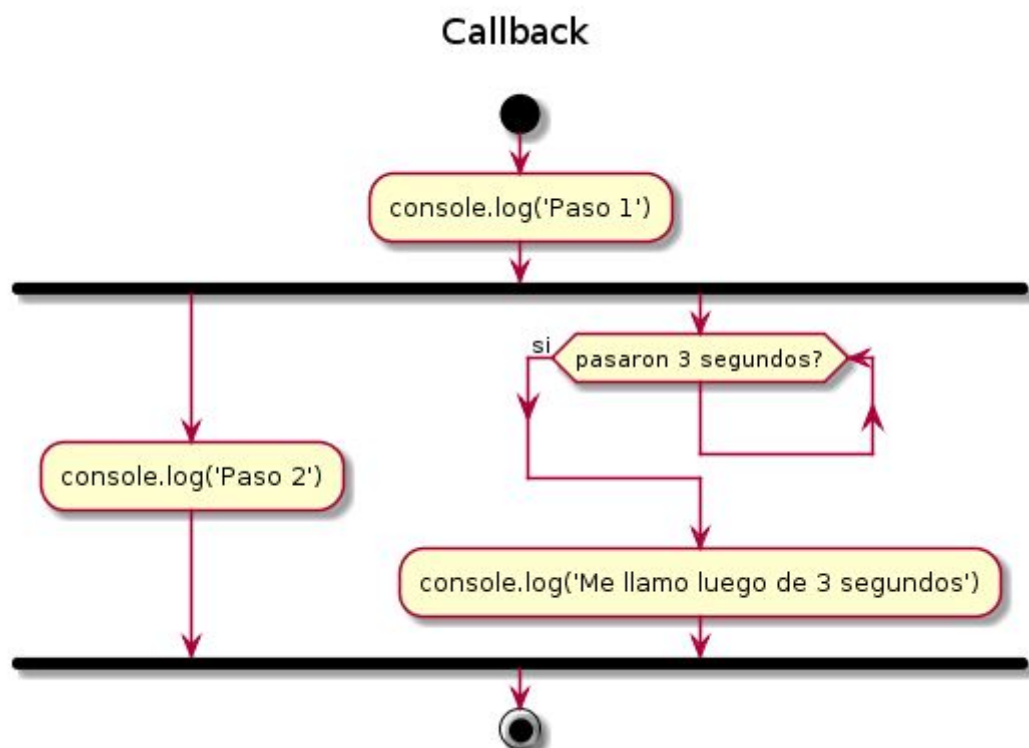
Puede ser por medio de:

- Callbacks
- Promises



## 2. Callbacks

```
function hacerALos3seg() {  
  console.log('Me llamo luego de 3 segundos');  
}  
  
console.log("Paso 1");  
setTimeout(hacerALos3seg, 3000);  
console.log("Paso 2");
```





```
function doAsync1(fn) {  
  console.log("doAsync1");  
  fn();  
}  
function doAsync2(fn) {  
  console.log("doAsync2");  
  fn();  
}  
function doAsync3(fn) {  
  console.log("doAsync3");  
  fn();  
}  
function doAsync4(fn) {  
  console.log("doAsync4");  
  fn();  
}  
  
console.log("Iniciando");  
doAsync1(function() {  
  doAsync2(function() {  
    doAsync3(function() {  
      doAsync4(function() {});  
    });  
  });  
});  
  
console.log("Finalizando");
```

## Callbacks: ventajas y desventajas

### **Ventajas**

- Muy conocidas (posiblemente las hayas usado)
- Se usa en muchas funciones
- Es sencillo para operaciones básicas

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**

## **Desventajas**

- El código puede quedar ilegible fácilmente



### 3. Promises

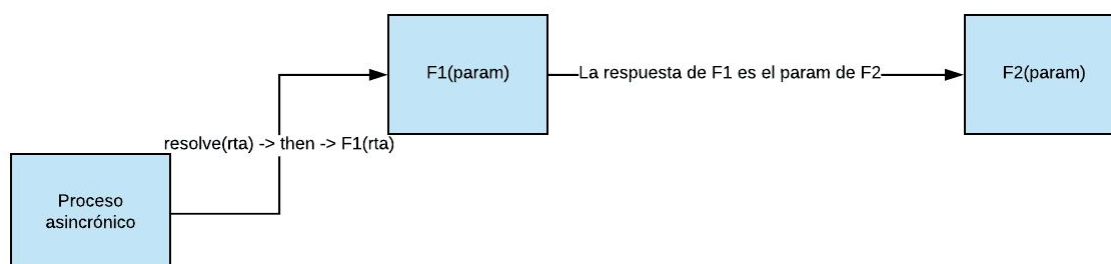
Es una abstracción de la programación asíncrona. Es un objeto que hace de “interlocutor” entre la función que hace una operación asíncrona, y el llamador. El cual (el “interlocutor”) maneja la respuesta de la función asíncrona (ya sea un valor o una excepción).

Si usamos una definición simplificada, podemos decir que es el resultado de una operación asíncrona, cuyo estado puede ser pendiente, completada, o rechazada.

```
console.log("Paso 1");

procesoAsincronicoConPromise()
  .then(funcionQueProcesaLaRespuestaOk)
  .catch(funcionQueProcesaElError);

console.log("Paso 2");
```





```
console.log("Paso 1");

procesoAsincronicoConPromise()
  .then(funcionQueProcesaLaRespuestaOk)
  .then(funcionQueProcesaAlFinalRespuestaOk)
  .catch(funcionQueProcesaElError);

console.log("Paso 2");
```

## Promises: Ventajas y Desventajas

### Ventajas

- Código más legible.
- Es la forma más habitual de trabajar con procesos asincrónicos.
- Se utiliza en varios frameworks JavaScript (Angular, React, Node)
- Podemos concatenar funciones que procesan la respuesta.
- Podemos enfocarnos y trabajar por “partes”, dividiendo un problema complejo en varios más pequeños.

### Desventajas

- El código puede quedar ilegible fácilmente



## 4. Async/Away

Supongamos que tenemos una función que realiza una consulta a un servidor externo, esta operación es del tipo asincrónica, ya que el servidor puede tardar en responder (JavaScript obliga que todas estas operaciones sean asincrónicas), y debemos mostrar la respuesta del servidor por consola.

### Ejemplo

```
async function realizarPeticiónAServidorExterno() {  
    var respuesta = await http.get('http://unServer.com');  
    return respuesta;  
}
```

La variable **respuesta**, no será asignada hasta que el servidor no retorne la respuesta. Para ello, la palabra **await**, hace que no continúe la ejecución del programa hasta no recibir la respuesta del servidor.

Al utilizar **await** dentro de una función, debemos “marcar” la misma como una función asincrónica, por ello es que incorporamos, en la definición de la función, antes del nombre la palabra **async**.



## 5. Ejemplo de asincronismo en JS

Vamos a realizar un carrousel de imágenes muy rudimentario.

El HTML es este:

```
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <link rel="stylesheet" href="css/style.css">
8  </head>
9  <body>
10
11      
12
13      <h2 onclick="cambiarImgIz()">cambiarIzq</h2>
14      <h2 onclick="cambiarImgDe()">cambiarDer</h2>
15      <h1>Hola Mundo!</h1>
16
17      <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Harum obcaecati perferend
18      <script src="js/script.js"></script>
19  </body>
20 </html>
```

El CSS sólo contiene una pequeña configuración de la vista de las imágenes





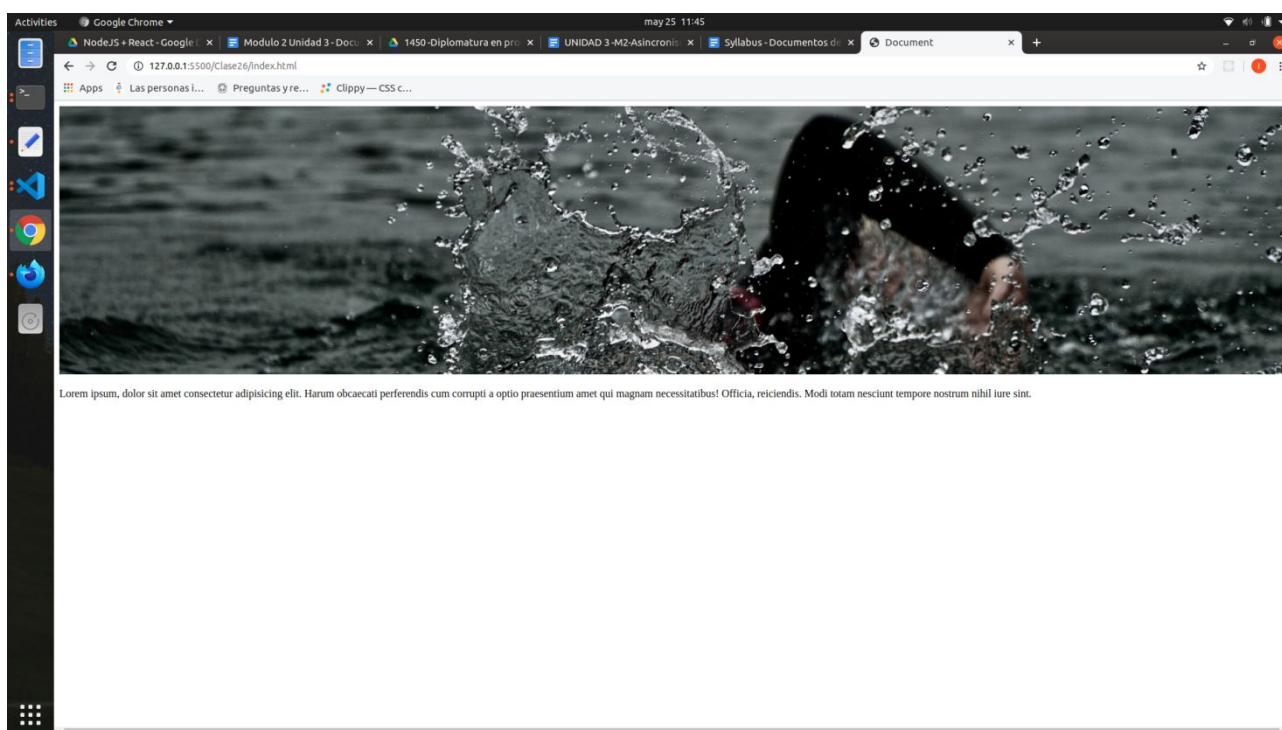
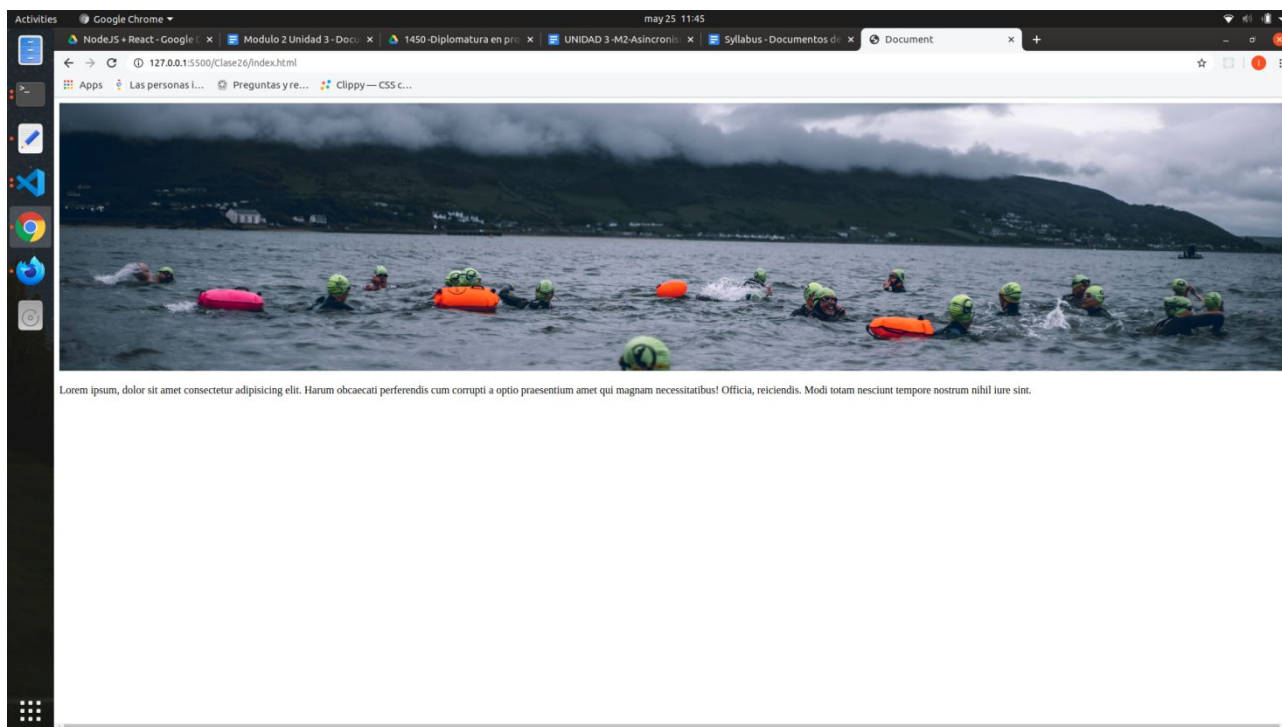
```
6  img {
7
8      width: 100vw;
9      height: 400px;
10
11     align-items: center;
12
13     z-index: -1;
14
15     object-fit: cover;
16
17 }
```

En el JS utilizamos un `setInterval` para ir cambiando el `src` de la imagen. Lo hacemos de modo indefinido y cada mil milisegundos, es decir, una vez por segundo.



```
1 //setInterval(); Hace "algo" (funcion) cada tanto tiempo
2 //setTimeout(); Hace "algo"(funcion) despues de transcurrido un tiempo
3 // los tiempos se calculan en milisegundos
4
5 var titulo = document.querySelector("h1");
6 var texto = document.querySelector("p");
7
8 var tbody = document.querySelector("tbody");
9
10
11 var imagen = document.querySelector("img");
12
13
14 var imagenes = ["img1.jpg", "img2.jpg", "img3.jpg", "img4.jpg", "img5.jpg"];
15 var i = 0;
16
17 setInterval(()=>{
18     imagen.src = "image/"+imagenes[i];
19     i = i + 1;
20
21     if(i == 5) {
22         i = 0;
23     }
24 }, 1000);
25
26
27
28
```

El resultado final es un cambio de imagen constante

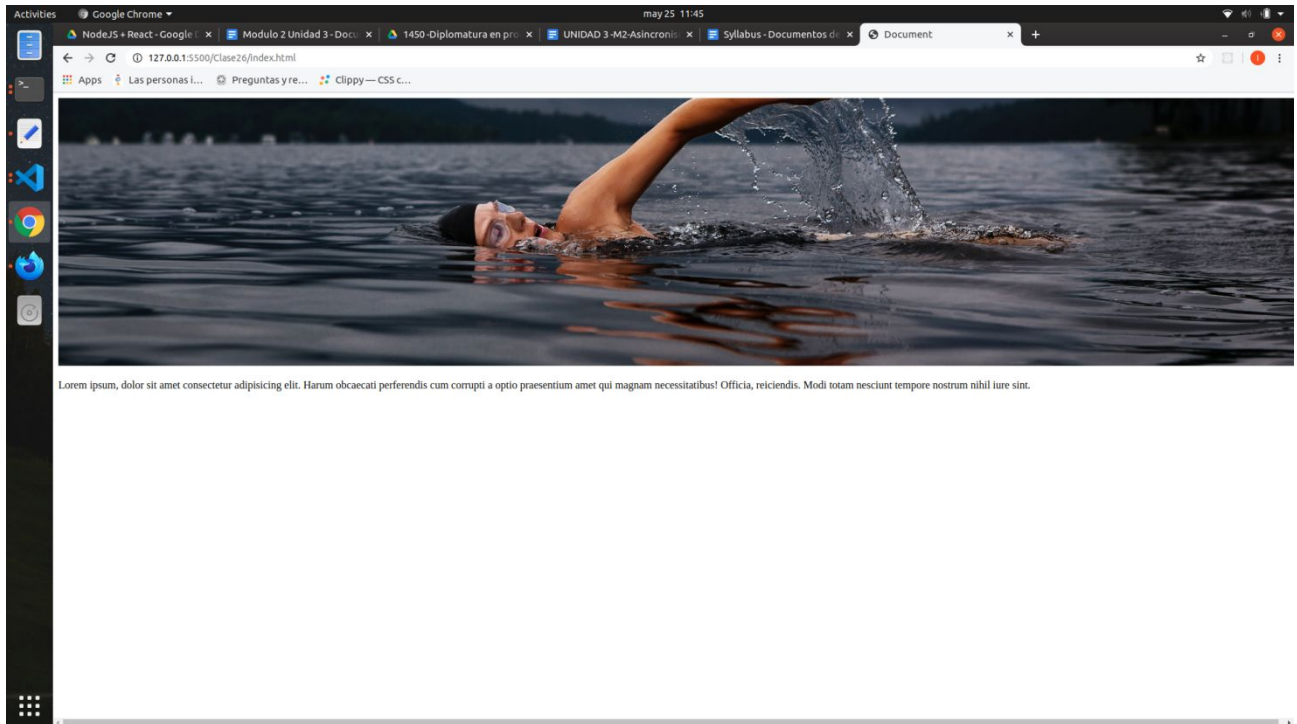




**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

p. 20



**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



## 6. Trabajo Práctico

Modificar la tabla de meses desarrollada en la Unidad para que al recargar (o cargar por primera vez) la página, la tabla se empiece a dibujar de arriba hacia abajo con un movimiento perceptible (dar un tiempo entre el dibujo de una fila y la siguiente para que se perciba el movimiento).



## Bibliografía utilizada y sugerida

MDN - JavaScript. (n.d.) Recuperado de:

<https://developer.mozilla.org/es/docs/Web/JavaScript>

Udemy. (n.d.) Recuperado de: <https://udemy.com>

Wikipedia - JavaScript. (n.d.) Recuperado de <https://es.wikipedia.org/wiki/JavaScript>

w3schools.com - JavaScript. (n.d.) Recuperado de <https://www.w3schools.com/js/>



## Lo que vimos:

- Asincronismo:



## Lo que viene:

- Funciones anónimas y funciones arrow

