

## Introducción

Cuando nos referimos a una función de callback entendemos que es una función que se pasa a otra función como un parámetro. Esta función enviada por parámetro se ejecuta dentro de la función que la contiene para completar algún tipo de acción. De esta forma, podemos realizar ciertas acciones siempre y cuando se ejecute previamente la función primaria.

¿Cuáles son los principales objetivos en este proyecto?

Entender que es una función callback y cuando usarla

Mejorar tus conocimientos en JavaScript

## Descripción del Proyecto

### 1. Crea un nuevo proyecto JavaScript

Para comprender que es un callback primero necesitas entender el porqué de la necesidad de usarlos. Para ello, crea un script con el siguiente contenido y ejecútalo:

*esto es lo que nos muestra la función realizada por callbacks*

```
function a() {  
    console.log("Se ha ejecutado la función a");  
}  
  
function b() {  
    console.log("Se ha ejecutado la función b");  
}  
b();  
a();
```

Como podrás comprobar, se mostrará por consola los siguientes mensajes:

Se ha ejecutado la función a

Se ha ejecutado la función b

A continuación modifica la función a para que tenga un tiempo de espera en su ejecución mediante la función `setTimeout` de JavaScript:

```
function a(){  
    // Simulamos un tiempo de espera en la ejecución de la función a  
    setTimeout( function(){  
        console.log("Se ha ejecutado la función a");  
    }, 1000 );
```

```

}

function b(){

  console.log("Se ha ejecutado la función b");

}

a();

b();

```

¿Qué pasa si ejecutas ahora el código? Si realmente deseas ejecutar primero la función a, y cuando esta finalice ejecutar la función b, tendrás que modificar el script y hacer uso de los callbacks.

## 2. Actualiza el código JavaScript para poder hacer uso de callbacks

Actualiza el script implementado el uso de los callbacks para que únicamente se ejecute la función b si previamente se ha ejecutado la función a y ha finalizado.

## 3. Busca 5 casos reales

A continuación busca 5 casos reales donde el uso de callbacks sea necesario y ejemplifique claramente su necesidad de uso.

## 4. Crea una presentación

Por último tendrás que crear una presentación explicando todo lo que has aprendido acerca de los callbacks y cuando recomiendas su uso. También tendrás que analizar que desventajas tienen y en qué casos no es recomendable usarlos.

### Requisitos

Debes de hacer uso de GIT

Debes de documentar correctamente todo el código

### Objetivo del proyecto

- Entender cómo funciona callbacks
- Mejorar tus conocimientos en JavaScript
- Mejorar tus conocimientos en lógica y programación

## La vamos a priorizar de la siguiente manera

TAREA	PRIORIDAD	MINUTOS	DIFICULTAD
Documentación	Alta	30 minutos	ALTA
Organización	Alta	20 minutos	NORMAL
Búsqueda Previa de información	Normal	30 minutos	NORMAL

Creación de repositorio	Normal	10 minutos	BAJA
Documentando el proyecto	Alta	20 minutos	Alta
Creación HTML	Alta	10 minutos	Baja
Creación css	Normal	5 minutos	Baja
Creación js callbacks	Normal	10 minutos	Normal
Creación js case	Alta	30 minutos	Normal

## Registro de incidencia

Ninguna

## Calendario

11/12/19

Realización 2 horas de proyectos

## Riesgos

Que se moje el ordenador

## Workflow de git que se va usar

[Documentación acerca del WORKFLOW de git que vas a usar](#)

- Creación Git Hub <https://github.com/robertfox11/JSCallback.git>
- Hacemos commits de la estructura de la página principal.
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación) Pedir ayuda a compañeros
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación) Pedir ayuda a compañeros
- No encontrar con facilidad la información relacionada con el proyecto
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación)
- Pedir ayuda a compañeros

A partir de la realización de la estructura se continuó trabajando solamente en la rama “master”, a través del Workflow “Gitflow”.

Mas información --> <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow>



### Herramientas utilizadas para el proyecto

Son las siguientes

- Sistema operativo Windows
- Navegadores compatibles Chrome, Firefox
- Visual Studio Code