



Certified Tech Developer

The Ultimate Degree

JS101

Práctica CALLBACKS

Objetivos

Comprender cuándo y cómo se usan estas estructuras de control de flujo.

ref:

<https://docs.google.com/document/d/1tqjBxgTtZv-LmLLLe4TodO1bIrwH6MrmNo0h-ZRapbk/edit#>

**Micro desafíos - Paso I**

Tomando el siguiente ejercicio de bucles:

```
const a = [23, 45, 32];
const b = [12, 67, 75];

const encontrarGanador = function (a, b) {
  let primerParticipante = 0;
  let segundoParticipante = 0;
  for (let i = 0; i < 3; i++) {
    if (a[i] > b[i]) {
      primerParticipante++;
    } else if (a[i] < b[i]) {
      segundoParticipante++;
    }
  }
  if (primerParticipante > segundoParticipante) {
    return "primer";
  } else {
    return "segundo";
  }
}

// primera etapa
console.log("El ganador es el: " + encontrarGanador(a, b) + " participante");
```

Veamos si podemos aprovechar el poder de los CALLBACKS para agregar una función que nos permita definir de qué manera mostrar el output de esa función buscarGanador(a, b). Nos pidieron tener mas de una forma de mostrar al ganador. Que podamos decirle a la función encontrarGanador(a, b) que mediante un tercer parámetro podamos proponer diferentes modos de mostrar la salida por pantalla. Mostar en dos formatos: a) enFormatoTexto b) enFormatoTabla

O sea usando un CALLBACK como 3º parámetro.:



Micro desafíos - Paso II

En esta otra oportunidad tenemos un servicio web que recibe arrays con nombres dentro que deben convertirse en URLs para saber si esas páginas existen con .com. Por ejemplo ["google", "vivalapepa", "youtube"]



Micro desafíos - Paso III

Realizar un contador de cuenta regresiva usando setInterval().

Esta función permite ejecutar un callback cada x cantidad de tiempo.