



Práctica Integradora

Objetivo

Vamos a ejercitar todo lo aprendido sobre **Ciclos o Bucles iterativos**, integrandolo con lo visto anteriormente.

Los ciclos o bucles constituyen un excelente recurso en la programación, ya que nos ayudan a evitar la repetición de líneas de código al momento de querer ejecutar algo varias veces. **Por ejemplo:** tratar de listar los elementos almacenados en un array.

Recuerda que siempre puedes revisar el material visto. Si no encuentras algo, puedes consultar con tus profesores y compañeros, así como también googlear la información que creas necesaria. Recomendamos consultar documentaciones oficiales como la de [MDN Web Docs](#), [W3School](#) o foros reconocidos como [Stack Overflow](#).

Cuando tengas alguna duda que te impida avanzar, puedes preguntarle a tus profesores.

¡ Éxitos !



Micro desafío 1:

Instrucciones

1. Crear una carpeta de trabajo y dentro de ella crea un archivo JavaScript (**.js**).

Saldo Bancario

El Tech Leader nos informa que debemos efectuar un código que permita **calcular el saldo de la cuenta de un cliente de la Institución Financiera**. Tu responsabilidad será ir ejecutando cada una de las siguientes consignas para lograr el objetivo:

1. Crear un array que contenga las operaciones bancarias realizadas por el cliente, tomando en cuenta que los **depósitos** serán representados por **valores positivos** y los **retiros** por **valores negativos**.
2. Crear una función ([Callback](#)) que reciba como parámetro el array creado. La función tendrá la responsabilidad de calcular y devolver el saldo total de depósitos, el saldo actual de retiros y el saldo actual del cliente.
3. Crear otra función que reciba como parámetros el *nombre* y el *apellido* del cliente así como el *array que contiene las operaciones bancarias*. Esta función tendrá la responsabilidad de **retornar** el *nombre y apellido del cliente* y el *saldo total de depósitos, el saldo actual de retiros y el saldo actual*.

Una vez creado el programa, ejecutalo y para verificar los resultados solicitados, invoca a la segunda función, asigna diferentes valores o argumentos. No olvides utilizar el [console.log\(\)](#) para mostrar los resultados al usuario, de acuerdo al siguiente mensaje:

```
Estimada Gloria Medina:  
  
El monto total de Los depósitos es de: $5500.  
  
El monto total de Los retiros es de: $15000.  
  
Por lo tanto, su saldo actual en la cuenta es de: $40000.
```



Micro desafío 2:

Instrucciones

En la carpeta ya creada, crea un nuevo archivo JavaScript (**.js**).

Tabla de multiplicar

Uno de los equipos de desarrollo te pide ayuda para desarrollar un código, donde se pueda visualizar la tabla de multiplicar del 1 al 10. El equipo indica que puedes usar algunos de los ciclos o bucles aprendidos: [for](#) - [while](#) o [do while](#),

Para ayudar al equipo de desarrollo, tu responsabilidad será:

1. Crear una función que no reciba ningún parámetro. Tendrá la responsabilidad de generar de manera automática la tabla de multiplicar del 1 al 10.
2. Lo ideal es que función muestre al usuario algo así:

```
Tabla de multiplicar
-----
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10
----- // Y así hasta la tabla del 10
```



Desafío extra (opcional):

Instrucciones

Si llegaste hasta acá, significa que vienes trabajando muy bien. **¡Felicitaciones!**

Para que no te quedes con las ganas y puedas seguir practicando, te proponemos que desarrolles las siguientes funciones:

En la carpeta ya creada, crea un nuevo archivo JavaScript (**.js**) por cada nueva función que desarrolles.

1. Crear una **función** que pida un valor por parámetro y muestre los 10 números siguientes.
2. **Imprimir** los números entre el 1 y el 57, saltando de tres en tres.
3. Escribir un **programa** que muestre la sumatoria de todos los números entre el 0 y el 100.
4. Crear una **función** que reciba por parámetro una cadena de texto: 'practicando el uso de los ciclos o bucles'. Tendrá la responsabilidad de mostrar al usuario cada una de las letras de la cadena de texto pero en MAYÚSCULA.

Pista: Investigar el método [.toUpperCase](#).

5. Crear una función que reciba como parámetro un array de números positivos. Tendrá la responsabilidad de retornar un nuevo array pero sólo con los valores pares.

Pista: Investiga sobre el uso del [operador módulo](#). Recordemos también que un número par es aquel que se puede dividir entre 2.