



**Certified Tech
Developer**

The Ultimate Degree

Examen parcial de Programación Imperativa

¡Llegó el momento de poner a prueba todo lo que estuvimos viendo a lo largo de estas semanas!

Metodología de evaluación

Se evaluarán los siguientes conceptos sobre el código entregado:

- **FORMA**
 - Que el código esté prolijo e implemente buenas prácticas
 - Que las variables, métodos y funciones tengan nombres descriptivos
 - Que utilices nombres en español o en inglés pero no ambos
 - Que utilices camelCase donde corresponda
- **LÓGICA**
 - Que la lógica corresponda con lo que solicitan las consignas
 - Que utilices los métodos más adecuados para cada caso
- **FUNCIONAMIENTO**
 - Que el código funcione correctamente, sin arrojar errores
 - Que el código produzca el resultado esperado a partir de los datos suministrados

Duración, formato y entrega

El examen tendrá una duración de 100 minutos (aprox). La hora de finalización será informada por el docente. ⚠️ **Las entregas realizadas después del tiempo estipulado no serán tenidas en cuenta.**

Al terminar el parcial, deben entregar un solo archivo .js que contenga todos los ejercicios, haciendo uso del formulario que les enviará el docente.

ACLARACIÓN: Si usaron prompt van a tener que tener la línea 1 de config con lo del prompt entonces con eso sabemos que lo instalaron bien.

⚠️ **Recuerden verificar la recepción del formulario con el/la docente antes de retirarse del zoom** ⚠️

Algunos consejos 🧐💡

- Es normal tener nervios en esta etapa. Recordá que practicamos mucho estas semanas y tratá de mantener la mayor calma que puedas.
- Lee todas las consignas antes de empezar el parcial para tener una idea general de lo que se pide.
- Si alguna consigna no queda clara, recordá que podés pedir aclaración por privado al docente.
- Si una consigna te bloquea demasiado, pasá a la siguiente, lo más probable es que luego se te ocurra una solución.
- El uso de prompt es opcional.

Ejercicio 1:

Realizar una **función** que reciba por parámetro dos valores y verifique qué tipo de dato son:

- Si ambos son tipo de dato **Number** deberá **retornar** su multiplicación.
- Si ambos son tipo de dato **String** deberá concatenarlos y **retornar** el resultado de esa concatenación.
- Si ambos son tipo de dato **Booleano** deberán **retornar** false.

Ejercicio 2:

- 1) Crear un **objeto literal** que contenga las siguientes propiedades y asignarles un valor:
 - a) nombre
 - b) apellido
 - c) edad
 - d) esArgentino (bool)
- 2) Declarar un **función** que reciba al objeto por parámetro:
 - a) Si la edad es mayor o igual a 18 años y es argentino, **retornará** una frase que diga: *[nombre] [apellido] vive en Argentina y es mayor de edad.*
 - b) Si es mayor de edad pero no es argentino, **retornará** una frase que diga: *[nombre] [apellido] NO vive en Argentina y es mayor de edad.*
 - c) Si NO es mayor de edad pero es argentino, **retornará** una frase que diga: *[nombre] [apellido] vive en Argentina y NO es mayor de edad.*
 - d) Si no es mayor ni es argentino, **retornará** una frase que diga: *[nombre] [apellido] NO vive en Argentina y NO es mayor de edad.*

Realizar una función llamada *verificacionDeTiposDeDatos* que reciba dos valores por parámetro, estos pueden ser String o Booleano. Nuestra función deberá determinar si los valores recibidos son del mismo tipo **sin utilizar typeof**. Por ejemplo:

- Si recibe por parámetro (true, "true") deberá retornar un mensaje que indique "No son del mismo tipo"
- Si recibe por parámetro (false, true) deberá retornar un mensaje que indique "Son del mismo tipo"
- Si recibe por parámetro ("true", "false") deberá retornar un mensaje que indique "Son del mismo tipo"
- Si recibe por parámetro ("false", false) deberá retornar un mensaje que indique "No son del mismo tipo"

Ejercicio 3:

1. Crear un array vacío llamado **destinos**.
2. Dados estos tres objetos:

```
let brasil = {  
  cantidadDeVisitas : 3,  
  clima: "soleado",  
  habitantes: "212 millones"  
}  
  
let rusia = {  
  cantidadDeVisitas : 4,  
  clima: "frío",
```

```
    habitantes: "144 millones"
  }
  let estadosUnidos = {
    cantidadDeVisitas : 1,
    clima: "nublado",
    habitantes: "329 millones"
  }
```

- a. Agregarlos al array. (**utilizando un método JS, piensa cuál podrías utilizar**)
 - b. **Mostrar por consola** el array para ver que contenga esos objetos.
3. Crea una función que reciba el array **destinos** como parámetro, e incremente en 1 la cantidad de visitas de cada uno de los destinos, llámala **siguienteViaje**.

* *Debes utilizar alguna estructura de repetición vista en clase.*
4. Mostrar por consola el array de destinos antes y después de la ejecución de la función.