



**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
 - Las invocaciones correspondientes

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 con extensión .txt** 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 llamada `restarPares` que reciba por parámetro **dos** valores numéricos.

- Si ambos números recibidos son **pares** entonces se deberá **retornar** el **resultado** de restarle al primer parámetro el segundo.
- Si alguno o los dos **no** son pares , se deberá retornar un string que diga **"Lo siento, uno o mas parámetros no son pares"**

Ejercicio 2:

Debemos poder controlar el acceso a un recital de música para una determinada persona. Para eso, debemos realizar una **función** que reciba **dos** parámetros

- 1) La edad de la persona, Number
- 2) Si tiene o no, el ticket de entrada , booleano

La función deberá evaluar las siguientes **condiciones** y **retornar** en cada caso lo que se pida:

- A) Si tiene entrada y es mayor o igual de 18 años: **retorna un valor booleano** `true`.
- B) Si tiene entrada pero es menor de 18 años: **retorna un string** que diga "Sólo puedes acceder con un adulto acompañante".
- C) Si no tiene entrada: **retorna un booleano** `false`.



Ejercicio 3:

Dado el siguiente arreglo de objetos:

```
let productos = [  
  {  
    producto : "remera",  
    tipo: "indumentaria",  
    precio: 2100  
  },  
  {  
    producto : "notebook",  
    tipo: "tecnologia",  
    precio: 200000  
  },  
  {  
    producto : "celular",  
    tipo: "tecnologia",  
    precio: 27000  
  },  
  {  
    producto : "protector solar",  
    tipo: "cosmetica",  
    precio: 2500  
  },  
  {  
    producto : "pantalon",  
    tipo: "indumentaria",  
    precio: 7500  
  },  
  {  
    producto : "tablet",  
    tipo: "tecnologia",  
    precio: 60000  
  },  
]
```



Realizar una **función** que reciba un **array** como parámetro, y lo recorra para **filtrar** los productos que sean de tipo “tecnología” y que su precio sea **mayor o igual** a 50000, la función deberá **retornar** un **nuevo arreglo** con los productos que cumplan con las condiciones antes mencionadas.