



**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:

1. Deberás escribir un objeto literal llamado **estudiante** que contenga las siguientes propiedades que deberás completar con tus datos:
  - a. nombre
  - b. apellido
  - c. camada
  - d. esParcial (VERDADERO)
2. Crear una función que, pasado un número entero (la futura nota), redondee según la siguiente regla:
  - $NOTA < 4$  : desaprobado
  - $4 < NOTA < 7$  : debe rendir examen final
  - $NOTA \geq 7$  : promocionado.

La función deberá retornar la condición final del alumno (solo retorna la palabra).

Al momento de llamar a la función, deberás mostrar la siguiente frase:

“El alumno obtuvo una nota igual a [NOTA] por lo tanto su condición es [CONDICIÓN]”

## Ejercicio 2:

1. Crea 1 array llamado **clase**.
2. Crea 3 objetos literales con las propiedades nombres, apellidos y bimestre de 3 compañeros.  
Ej: Esteban Piazza 1 - Leandro Borrelli 1 - Martín Cejas 1
  - a. Agrega estos 3 objetos al array llamado **clase**.

- b. Imprimir el array para ver que contenga esos objetos.
3. Crea una función que reciba el array **clase** como parámetro e incremente en 1 el número de bimestre de cada uno de los alumnos, llámala ***siguienteCursada***. Debes utilizar una estructura de repetición.
4. Deberás hacer el correcto llamado de la función mostrando el array **clase** antes y después de llamar a ***siguienteCursada***.

### Ejercicio 3:

1. Crear una función que apruebe o desaprobe alumnos en base a su asistencia y a su progreso completo en PlayGround (true, false). Para ello deberán:
  - Crear una constante **asistenciaMinima** que almacene el número mínimo de asistencias permitidas para la aprobación de la materia, en este caso serán 23 asistencias mínimas.
  - La función deberá recibir como parámetros la cantidad de asistencias y la condición final ante PlayGround del alumno.
  - Un alumno, será aprobado si tiene la cantidad de asistencias mínima y su progreso final está completo. Si alguna de estas opciones no se cumple, el alumno será automáticamente desaprobado.
  - Deberá retornar un string aprobado o desaprobado según corresponda