

# UNIVERSIDAD PRIVADA DOMINGO SAVIO

## FACULTAD DE INGENIERIA



### Actividad

**Título:** Ejercicios de Replit

**Docente:** Ing. Jimmy Nataniel Requena Llorentty

**Materia:** Programación 2

**Estudiante:** LORA COLODRO FABRIZIO

**Junio del 2025**  
Santa Cruz – Bolivia

prog2-ejer-prac-class - Replit

https://replit.com/@snwxsulles/prog2-ejer-prac-class

0/5 checkpoints used

Files: edad-cine.py, verificador\_edad.py, README.md

```
1 # Verificador de Edad en una Película
2 # Ingresar tu edad
3 edad = int(input("Ingresa tu edad: "))
4
5 if edad >= 18:
6     print("¡Puedes ver películas clasificadas R!")
7 elif edad >= 13:
8     print("Puedes ver películas clasificadas PG-13.")
9 elif edad >= 0:
10    print("Te recomendamos películas clasificadas G o PG.")
11 else:
12    print("Edad no válida.")
```

Shell

```
~/workspace: python edad-cine.py
3 files changed, 49 insertions(+), 16 deletions(-)
create mode 100644 README.md
~/workspace$ git push origin main --force
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.34 KiB | 1.34 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:FaXx0/prog2-ejer-prac-class.git
 a35afe..13eeec3 main -> main
~/workspace$ python edad-cine.py
Ingresa tu edad: 7
Te recomendamos películas clasificadas G o PG.
~/workspace$
```

09:40 a.m. 10/06/2025

prog2-ejer-prac-class - Replit

https://replit.com/@snwxsulles/prog2-ejer-prac-class

0/5 checkpoints used

Files: edad-cine.py, verificador\_edad.py, README.md, tabla\_multiplicar.py, Adivina\_el\_numero.py, Refactorizar.py

```
5 # Función para obtener la clasificación de películas según la edad
6 def obtener_clasificacion(edad): # Función principal
7     if edad < 0 or edad > 120: # Validación de edad
8         return "Edad no válida." # Mensaje de error
9     elif edad < 13: # Clasificación para menores de 13 años
10        return "Te recomendamos películas clasificadas G o PG." # Mensaje de recomendación
11    elif edad < 18: # Clasificación para mayores de 13 años y menores de 18 años
12        return "Puedes ver películas clasificadas PG-13." # Mensaje de recomendación
13    else: # Clasificación para mayores de 18 años
14        return "¡Puedes ver películas clasificadas R!" # Mensaje de recomendación
15
16 def pruebas(): # Función para pruebas de la función obtener_clasificacion (pruebas unitarias)
17     assert obtener_clasificacion(10) == "Te recomendamos películas clasificadas G o PG."
18     "Prueba fallida: Niño de 10 años"
19     print("Pruebas 7 PASADA: Niño de 10 años") # Pruebas 1
```

Shell

```
~/workspace: python verificador_edad.py
~/workspace$ python verificador_edad.py
Verificador de Edad para Películas
[✓] Prueba 7 PASADA: Niño de 10 años
[✓] Prueba 5 PASADA: Límite adolescente (13 años)
[✓] Prueba 6 PASADA: Adolescente de 17 años
[✓] Prueba 2 PASADA: Límite adulto (18 años)
[✓] Prueba 3 PASADA: Adulto mayor de 65 años
[✓] Prueba 12 PASADA: Bebé de 0 años
[✓] Prueba 10 PASADA: Edad negativa (-1)
[✓] Prueba 11 PASADA: Edad excesivamente alta (150)
Todas las pruebas pasaron.
Ingresa tu edad: 15
```

03:58 a.m. 11/06/2025

The screenshot shows a Replit IDE window titled "prog2-ejer-prac-class". The browser address bar displays "https://replit.com/@snowsoulles/prog2-ejer-prac-class". The file explorer on the left lists several files: "edad\_cine.py", "verificador\_edad.py", "README.md", "tabla\_multiplicar.py", "Adivina\_el\_numero.py", and "factorizar.py". The "tabla\_multiplicar.py" file is open in the editor, showing a Python script that takes a number as input and prints a multiplication table for that number. The script includes a range check and a loop to calculate the products. Below the editor, a terminal window shows the execution of the script. The user has entered the number 6, and the output displays a multiplication table for 6, ranging from 6 x 1 to 6 x 10.

```
5
6 numero = int(input("Introduce un número del 1 al 10: "))
7
8 # Validamos que el número esté dentro del rango permitido
9 if 1 <= numero <= 10:
10     print(f"\nTabla de multiplicar del {numero}:\n")
11     for i in range(1, 11): # del 1 al 10
12         resultado = numero * i
13         print(f"{numero} x {i} = {resultado}")
14 else:
15     print("Número fuera de rango. Por favor, ingresa un número del 1 al 10.")
16
```

```
~/workspace$ python tabla_multiplicar.py
Introduce un número del 1 al 10: 6

Tabla de multiplicar del 6:

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
~/workspace$
```

The screenshot shows the same Replit IDE window, but now the "Adivina\_el\_numero.py" file is open in the editor. This script implements a number guessing game where the user has to guess a secret number (7) within a range of 1 to 10. The script uses a while loop to keep asking for guesses until the correct number is entered. The terminal window shows the execution of the script. The user has entered several incorrect guesses (5, 6, 9, 10, 7) and the program provides feedback like "Demasiado bajo." or "Demasiado alto." until the correct guess (7) is entered, at which point it says "¡Correcto! El número era 7."

```
3 numero_secreto = 7 # Numero secreto fijo
4 intento = int(input("Adivina el número secreto (entre 1 y 10): "))
5
6 while intento != numero_secreto:
7     if intento > numero_secreto:
8         print("Demasiado alto.")
9     else:
10        print("Demasiado bajo.")
11        intento = int(input("Intenta otra vez: "))
12
13 print(f"¡Correcto! El número era {numero_secreto}.")
```

```
~/workspace$ python Adivina_el_numero.py
Adivina el número secreto (entre 1 y 10): 5
Demasiado bajo.
Intenta otra vez: 6
Demasiado bajo.
Intenta otra vez: 9
Demasiado alto.
Intenta otra vez: 10
Demasiado alto.
Intenta otra vez: 7
¡Correcto! El número era 7.
~/workspace$
```

The screenshot shows a Replit IDE window titled "prog2-ejer-prac-class". The browser address bar displays "https://replit.com/@snowouilles/prog2-ejer-prac-class#README.md". The file explorer on the left lists several files: README.md, edad\_cine.py, verificador\_edad.py, tabla\_multiplicar.py, Adivina\_el\_numero.py, and Refactorizar.py. The main editor displays the content of README.md, which includes a table of contents with links to each file. Below the editor, a terminal window shows the output of running the program. The output indicates that the user is in the workspace, runs the Refactorizar.py file, and receives prompts for age verification and area calculations for three rectangles.

```
1 # ~Presentacion de trabajos y ejercicios Programacion2~
2 # -1:verificador de edad de cine
3 # -2:verif-d-edad cine completo
4 # -3:README.md
5 # -4:Tabla de multiplicar
6 # -5:Adivina_el_numero.py
7 # -6:Refactorizar
```

```
Intenta otra vez: 6
Demasiado bajo.
Intenta otra vez: 9
Demasiado alto.
Intenta otra vez: 10
Demasiado alto.
Intenta otra vez: 7
¡Correcto! El número era 7.
~/workspace$ python Refactorizar.py
El área del rectángulo 1 (10x5) es: 50
El área del rectángulo 2 (7x3) es: 21
El área del rectángulo 3 (15x8) es: 120
~/workspace$
```

The screenshot shows the same Replit IDE window, but now the "Refactorizar.py" file is open in the editor. The code defines a function "mostrar\_area\_rectangulo" that takes "numero", "base", and "altura" as arguments. It calculates the area by calling "calcular\_area\_rectangulo" and prints the result. Below the function definition, there are three calls to "mostrar\_area\_rectangulo" for different rectangles. The terminal window shows the same output as the first screenshot, indicating that the program was run successfully.

```
6 # fin de la función
7 # Función principal con indentación FIXED
8 def mostrar_area_rectangulo(numero, base, altura): # Función principal
9     area = calcular_area_rectangulo(base, altura) # area = base * altura
10    # Fin de la función
11    print(f"El área del rectángulo {numero} ({base}x{altura}) es: {area}") # Salida de datos
12    # Fin de la función
13
14 # Rectángulos
15
16 # Rectángulo 1 (usando la función) # Llamada a la función principal con los parámetros correspondientes
17 mostrar_area_rectangulo(1, 10, 5)
18 # Rectángulo 2 (usando la función) # Llamada a la función principal con los parámetros correspondientes
19 mostrar_area_rectangulo(2, 7, 3)
20 # Rectángulo 3 (usando la función) # Llamada a la función principal con los parámetros correspondientes
```

```
Intenta otra vez: 6
Demasiado bajo.
Intenta otra vez: 9
Demasiado alto.
Intenta otra vez: 10
Demasiado alto.
Intenta otra vez: 7
¡Correcto! El número era 7.
~/workspace$ python Refactorizar.py
El área del rectángulo 1 (10x5) es: 50
El área del rectángulo 2 (7x3) es: 21
El área del rectángulo 3 (15x8) es: 120
~/workspace$
```