

UNIVERSIDAD PRIVADA DOMINGO SAVIO

FACULTAD DE INGENIERIA



Actividad

Título: Ejercicios de Replit

Docente: Ing. Jimmy Nataniel Requena Llorentty

Materia: Programación 2

Estudiante: Rodrigo Andres Miranda Borda

Junio del 2025
Santa Cruz – Bolivia

rod-cmd/peliculacompleta: pri x | (1) WhatsApp x tablasdemul.py - peliculacompleta x +

replit.com/@mirandarodrigo2/peliculacompleta#tablasdemul.py

0/5 checkpoints used

```
1 num_tabla = int(input("Ingresa un número para ver su tabla de multiplicar: "))
2 print(f"--- Tabla del {num_tabla} ---")
3 for i in range(1, 11):
4     resultado = num_tabla * i # i tomará valores de 1 a 10     resultado = num_tabla *
5     print(f"{num_tabla} x {i} = {resultado}")
6
```

```
~/workspace$ python tablasdemul.py
Ingresa un número para ver su tabla de multiplicar: 4
--- Tabla del 4 ---
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri x | (1) WhatsApp x obtener_clasificacion_pelicula... x +

replit.com/@mirandarodrigo2/peliculacompleta#obtener_clasificacion_pelicula_completa.py

0/5 checkpoints used

```
62 def ejecutar_pruebas():
63     """
64     Ejecuta todas las pruebas unitarias para verificar que la función
65     obtener_clasificacion_pelicula() funciona correctamente.
66
67     Utiliza asserts para validar diferentes casos de prueba:
68     - Casos límite (edades en los bordes de cada clasificación)
69     - Casos normales (edades típicas en cada rango)
70     - Casos de error (edades inválidas)
71
72     Raises:
73     AssertionError: Si alguna prueba falla, se detiene la ejecución
74                     y muestra qué prueba específica falló.
75     """
76
77     print("Ejecutando pruebas de la función
78     obtener_clasificacion_pelicula()...")
79     print("=" * 60)
80
81     # PRUEBAS PARA ADULTOS (10 años o más)
82     # Caso normal: adulto joven
83     assert obtener_clasificacion_pelicula(20) == "¡Puedes ver películas
```

```
~/workspace$ python obtener_clasificaci...
[✓] Prueba 11 PASADA: Edad excesivamente alta (15 años)
[✓] Prueba 12 PASADA: Bebé de 0 años

¡TODAS LAS PRUEBAS PASARON EXITOSAMENTE!
Nuestra función es robusta y maneja todos los casos correctamente.

Ahora vamos a verificar TU edad para película S...

Por favor, ingresa tu edad: 15

=====
RESULTADO DE CLASIFICACIÓN DE PELÍCULAS
=====
Edad ingresada: 15 años
Clasificación: Puedes ver películas clasificadas PG-13.

¡Gracias por usar nuestro verificador de edad!
¡Disfruta de tus películas!
~/workspace$
```

Temperatura

rod-cmd/peliculacompleta: pri x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

ordenamientos.py

```
1 def ordenamiento_de_burbuja(lista):
2     n = len(lista)
3     for i in range(n):
4         hubo_intercambio = False
5         for j in range(n - 1 - i):
6             if lista[j] > lista[j + 1]:
7                 lista[j], lista[j + 1] = lista[j + 1], lista[j]
8                 hubo_intercambio = True
9         if not hubo_intercambio:
10             break
11     return lista
12
13 if __name__ == "__main__":
14     numeros = [6, 3, 8, 2, 5]
15     print("Antes:", numeros)
16     ordenamiento_de_burbuja(numeros)
17     print("Después:", numeros)
```

Shell

```
~/workspace$ python ordenamientos.py
Antes: [6, 3, 8, 2, 5]
Después: [2, 3, 5, 6, 8]
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

ejemplo_refactorizacion.py

```
1 # Ejemplo de Refactorización
2 def calcular_area_rectangulo(base, altura):
3     """Calcula el área de un rectángulo"""
4     return base * altura
5
6 def mostrar_area_rectangulo(numero, base, altura):
7     """Muestra el área de un rectángulo con formato"""
8     area = calcular_area_rectangulo(base, altura)
9     print(f"El área del rectángulo {numero} ({base}x{altura}) es: {area}")
10
11 def main():
12     """Función principal del programa"""
13     # Ejemplo de uso
14     mostrar_area_rectangulo(1, 10, 5)
15
16     # Puedes probar con más rectángulos
17     mostrar_area_rectangulo(2, 8, 3)
18     mostrar_area_rectangulo(3, 15, 7)
19
20 if __name__ == "__main__":
21     main()
```

Shell

```
~/workspace$ python ejemplo_refactorizacion.py
El área del rectángulo 1 (10x5) es: 50
El área del rectángulo 2 (8x3) es: 24
El área del rectángulo 3 (15x7) es: 105
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used Run

Files Search

- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numero_secreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

- .upm

File Folder

Configure Repl

bucles.py

```
1 mensaje = input("escribe el mensaje que deseas repetir: ")
2 veces = int(input("¿cuántas veces deseas repetirlo? "))
3 for i in range(veces):
4     print(mensaje)
5
6 mensaje = input("escribe el mensaje que deseas repetir: ")
7 veces = int(input("¿cuántas veces deseas repetirlo? "))
8 contador = 0
9 while contador < veces:
10     print(mensaje)
11     contador += 1
12
13
```

Shell

```
~/workspace$ python bucles.py
escribe el mensaje que deseas repetir: hola mundo
¿cuántas veces deseas repetirlo? 10
hola mundo
hola mundo
hola mundo
hola mundo
hola mundo
hola mundo
hola mundo
hola mundo
hola mundo
hola mundo
escribe el mensaje que deseas repetir: hello worl
d
¿cuántas veces deseas repetirlo? 2
hello world
hello world
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

Ln 1, Col 1 • Spaces: 4 History

rod-cmd/peliculacompleta: pri x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used Run

Files Search

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numero_secreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

- .upm

Configure Repl

lista_promedio.py

```
1 cantidad_notas = int(input("¿cuántas notas vas a ingresar? "))
2 suma = 0
3 for i in range(cantidad_notas):
4     nota = float(input(f"ingrese la nota # {i + 1}: "))
5     suma += nota
6 promedio = suma / cantidad_notas
7 print(f"\nSuma total de notas: {suma}")
8 print(f"Promedio: {promedio}")
9
```

Shell

```
~/workspace$ python listacomidas.py
Mi hobbie favorito es pique macho
Mi segundo hobbie favorito es picante
Mi tercer hobbie favorito es sopa de mani
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

Ln 1, Col 1 • Spaces: 4 History

rod-cmd/peliculacompleta: pri... (1) WhatsApp peliculacompleta - Replit

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

listagrupos.py

```
1 nombres_integrantes = ['Rodrigo Miranda', 'Fabricio Lora', 'Flavia Gutierrez', 'Joel Escalante', 'Sara Vidaurre', 'Lizbeth Apaza']
2 for nombre in nombres_integrantes:
3     print(f'¡Bienvenido a 404 NOT FOUND, {nombre}!')
```

Shell

```
~/workspace$ python listagrupos.py
¡Bienvenido a 404 NOT FOUND, Rodrigo Miranda!
¡Bienvenido a 404 NOT FOUND, Fabricio Lora!
¡Bienvenido a 404 NOT FOUND, Flavia Gutierrez!
¡Bienvenido a 404 NOT FOUND, Joel Escalante!
¡Bienvenido a 404 NOT FOUND, Sara Vidaurre!
¡Bienvenido a 404 NOT FOUND, Lizbeth Apaza!
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri... (1) WhatsApp peliculacompleta - Replit

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

listamishobbies.py

```
1 hobbies = ["escuchar musica", "mirar peliculas", "jugar videojuegos"]
2 print(f'Mi hobbie favorito es {hobbies[0]}')
3 print(f'Mi segundo hobbie favorito es {hobbies[1]}')
4 print(f'Mi tercer hobbie favorito es {hobbies[2]}')
```

Shell

```
~/workspace$ python listamishobbies.py
Mi hobbie favorito es escuchar musica
Mi segundo hobbie favorito es mirar peliculas
Mi tercer hobbie favorito es jugar videojuegos
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri... x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

File Folder

Configure Repl

numerossecreto.py

```
1 numerossecreto = 7
2
3 intento = int(input("Adivina el número secreto (entre 1 y 10): "))
4 while intento != numerossecreto:
5     if intento > numerossecreto:
6         print("Demasiado alto.")
7     else:
8         print("Demasiado bajo.")
9     intento = int(input("Intenta otra vez: "))
10 print(f"¡Correcto! El número era {numerossecreto}.")
11
12
```

Shell

```
~/workspace$ python numerossecreto.py
Adivina el número secreto (entre 1 y 10): 4
Demasiado bajo.
Intenta otra vez: 6
Demasiado bajo.
Intenta otra vez: 9
Demasiado alto.
Intenta otra vez: 7.
¡Correcto! El número era 7.
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri... x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

File Folder

Configure Repl

lista1.py

```
1 def contar_ocurrencias(lista, elemento_buscado):
2     contador = 0 # Inicializa el contador en 0
3     for elemento in lista: # Itera sobre cada elemento de la lista
4         if elemento == elemento_buscado: # Si el elemento es igual al
5             elemento_buscado
6             contador += 1 # Incrementa el contador en 1
7     return contador # Retorna el contador # Fin de la función
8
9 mis_numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5]
10 elemento_buscado = 10
11 resultado = contar_ocurrencias(mis_numeros, elemento_buscado)
12 print(f"El elemento {elemento_buscado} aparece {resultado} veces en la
13 lista.")
```

Shell

```
~/workspace$ python lista1.py
El elemento 10 aparece 1 veces en la lista.
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: prin x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files Search

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

File Folder

Configure Repl lista2.py

```
1 lista_invertida = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5]
2 print("Lista original:", lista_invertida)
3 lista_invertida.reverse()
4 print("Lista invertida:", lista_invertida)
5
6
```

Shell

```
~/workspace$ python lista2.py
Lista original: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5]
Lista invertida: [5, 4, 3, 2, 1, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: prin x (1) WhatsApp x peliculacompleta - Replit x +

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files Search

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

File Folder

Configure Repl lista3.py

```
1 def sumar_elementos(lista_numeros):
2     acumulador_suma = 0
3     for numero in lista_numeros:
4         acumulador_suma += numero
5     return acumulador_suma
6
7
8 mi_lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5]
9 resultado = sumar_elementos(mi_lista)
10
11 print(f"La suma de los elementos de la lista es: {resultado}")
12
13 print("-Con assert-")
14
15 def sumar_elementos(lista_numeros):
16     acumulador_suma = 0
17     for numero in lista_numeros:
18         acumulador_suma += numero
19     return acumulador_suma
20
21 assert sumar_elementos([1, 2, 3, 4, 5]) == 15
22 print("Prueba 1 pasada = 15")
23
24 assert sumar_elementos([10, 20, 30, 40, 50]) == 150
```

Shell

```
~/workspace$ python lista3.py
La suma de los elementos de la lista es: 70
(-Con assert-)
Prueba 1 pasada = 15
Prueba 2 pasada = 150
Prueba 3 pasada = 70
Prueba 4 pasada = 13
Prueba 5 pasada = 0
Prueba 6 pasada = 100
Todas las pruebas pasaron.
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

rod-cmd/peliculacompleta: pri... (1) WhatsApp peliculacompleta - Replit

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

- File
- Folder

Configure Repl

lista4.py

```
1 def encontrar_maximo(ingresa_los_numeros):
2     if len(ingresa_los_numeros) == 0:
3         return None # Mejor explicito
4
5     maximo = ingresa_los_numeros[0]
6     for numero in ingresa_los_numeros:
7         if numero > maximo:
8             maximo = numero
9     return maximo # Fuera del bucle
10
11
12 print("\nProbando encontrar_maximo")
13 assert encontrar_maximo([1, 9, 2, 8, 3, 7]) == 9
14 print("Prueba 1 pasada = 9")
15 assert encontrar_maximo([-1, -9, -2, -8]) == -1
16 print("Prueba 2 pasada = -1")
17 assert encontrar_maximo([42, 42, 42]) == 42
18 print("Prueba 3 pasada = 42")
19
```

Shell

```
~/workspace$ python lista4.py
Probando encontrar_maximo
Prueba 1 pasada = 9
Prueba 2 pasada = -1
Prueba 3 pasada = 42
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

24°C Ventoso 10:08 17/6/2025

rod-cmd/peliculacompleta: pri... (1) WhatsApp peliculacompleta - Replit

replit.com/@mirandarodrigo2/peliculacompleta

peliculacompleta 0/5 checkpoints used

Files

- .git
- bucles.py
- ejemplo_refactoriza...
- lista_promedio.py
- lista1.py
- lista2.py
- lista3.py
- lista4.py
- listacomidas.py
- listagrupos.py
- listamishobbies.py
- numerossecreto.py
- obtener_clasificacio...
- obtener_clasificacio...
- ordenamientos.py
- README.md
- tablasdemul.py

Package files

- File
- Folder

Configure Repl

obtener_clasificacion_pelicu...

```
1 # Ejercicio: Verificador de Edad para Películas usando Funciones
2
3 def obtener_clasificacion_pelicula(edad_persona):
4     """
5     Función que determina qué tipo de películas puede ver una persona según
6     su edad.
7
8     Parámetro:
9         edad_persona (int): La edad de la persona
10
11     Retorna:
12         str: Mensaje con la clasificación de película recomendada
13     """
14     # Validar que la edad sea válida
15     if edad_persona < 0 or edad_persona > 120:
16         return "Edad no válida."
17
18     # Lógica de clasificación por edad
19     if edad_persona < 13:
20         return "Te recomendamos películas G (General Audiences)."
```

Shell

```
~/workspace$ python obtener_clasificaci...
la.py
=== VERIFICADOR DE EDAD PARA PELÍCULAS ===
Por favor, ingresa tu edad: 18
Resultado: ¡Puedes ver películas de cualquier cl
asificación incluyendo R!
¡Gracias por usar nuestro verificador!
¿Quieres ver las pruebas con diferentes edades?
(s/n): s
=== PRUEBAS CON DIFERENTES EDADES ===
Edad 5: Te recomendamos películas G (General Aud
tencias).
Edad 12: Te recomendamos películas G (General Au
dientes).
Edad 13: Puedes ver películas G y PG-13.
Edad 16: Puedes ver películas G y PG-13.
Edad 17: ¡Puedes ver películas de cualquier clas
ificación incluyendo R!
Edad 25: ¡Puedes ver películas de cualquier clas
ificación incluyendo R!
Edad 65: ¡Puedes ver películas de cualquier clas
ificación incluyendo R!
Edad -5: Edad no válida.
Edad 150: Edad no válida.
~/workspace$
```

Activar Windows
Ve a Configuración para activar Windows.

24°C Ventoso 10:09 17/6/2025

