

UNIVERSIDAD PRIVADA DOMINGO SAVIO

FACULTAD DE INGENIERIA



Actividad

Título: Ejercicios en Replit

Docente: Ing.Jimmy Nataniel Requena Llorentty

Materia: Programación 2

Estudiante: Yoel Escalante Escobar

Fecha y hora actual: 2025-06-17 09:12:49

Santa Cruz – Bolivia

Junio del 2025

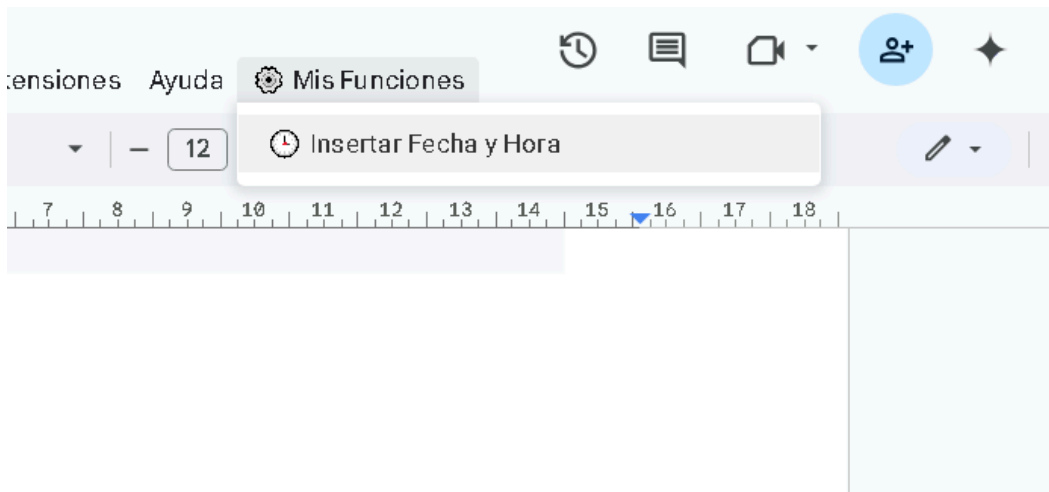
Scrip del documento	3
Actividad 1 Ejercicios	3
Ejercicio 1	3
Ejercicio 2	4
Ejercicio 3	4
Ejercicio 4	5
Ejercicio 5	6
Ejercicio 6	6
Ejercicio 7	7
Actividad 2 Ejercicio de listas	8
Ejercicio 1	8
Ejercicio 2	8
Ejercicio 3	9
Ejercicio 4	9
Ejercicio 5	10
Ejercicio 6	11
Ejercicio 7	12
Ejercicio 8	12
Ejercicio 9	13
Ejercicio 10	14

Scrip del documento

Implementamos el scrip para el google docs

```
function onOpen() {  
  DocumentApp.getUi()  
    .createMenu('⚙ Mis Funciones')  
    .addItem('🕒 Insertar Fecha y Hora', 'insertarFechaYHora')  
    .addToUi();  
}  
  
function insertarFechaYHora() {  
  var body = DocumentApp.getActiveDocument().getBody();  
  var fecha = Utilities.formatDate(new Date(), Session.getScriptTimeZone(),  
    "yyyy-MM-dd HH:mm:ss");  
  var parrafo = body.appendParagraph("Fecha y hora actual: " + fecha);  
  
  // Aplicar formato Arial tamaño 16  
  parrafo.setFontFamily("Arial");  
  parrafo.setFontSize(16);  
}
```

Con la implementación ahora aparece el botón para añadir fecha y hora



Actividad 1 Ejercicios

Ejercicio 1

En este ejercicio es para conocer los comandos del replit para subir al github

```
~/workspace$ git add .
~/workspace$ git commit -m "subido y ordenado todos los ejercicios hasta el 6"
[main e0fe282] subido y ordenado todos los ejercicios hasta el 6
1 file changed, 28 insertions(+), 15 deletions(-)
~/workspace$ git push origin main
Enumerating objects: 5, done.
```

Ejercicio 2

En este Ejercicio se busca aprender la condición for y lo que hace el range

```
5 #practica
6 #ejercicio2
7 print("Recorriendo un String")
8 for numero in range(3):
9     print(numero)
10 print ("Recorriendo un String:")
11 nombre = (input())
12 for letra in nombre:
13     print(letra)
14 #fin del programa
```

```
> Console Shell x +
~/workspace: python clase02_mi_codigo.py

15
Puedes ver peliculas clasificacion PG-13!
Recorriendo un String
0
1
2
Recorriendo un String:
hol
h
o
l
```

Ejercicio 3

Usamos la condición while para contar hasta el número que le dimos y los imprimamos 1 x 1

```
14 #ejercicio3
15 print ("Bucle while:")
16 num = int(input())
17 contador = 0
18 while contador < num:
19     print (f"contador es : {contador}")
20     contador+=1
21 print("Bucle while terminado")
22 #fin del programa
```

```
> Console Shell x +
~/workspace: python clase02_mi_codigo.py

Bucle while:
9
contador es : 0
contador es : 1
contador es : 2
contador es : 3
contador es : 4
contador es : 5
contador es : 6
contador es : 7
contador es : 8
Bucle while terminado
```

Ejercicio 4

Usamos la condición if para preguntar la edad y imprimir dependiendo que edad tenga

```
class02_mi_codigo.py > ...  
22 #ejercicio4  
23 """Estudiante Yoel Escalante Escobar"""  
24 print("Escriba su edad")  
25 edad = int(input())  
26 v if edad >= 18:  
27     print("Puedes ver peliculas clasificacion R!")  
28 v if edad >= 13 and edad < 18:  
29     print("Puedes ver peliculas clasificacion PG-13!")  
30 v if edad < 13 and edad >=0:  
31     print("Te recomendamos ver peliculas clasificacion G o PG!")  
32 v if edad < 0:  
33     print("Edad no valida!")
```

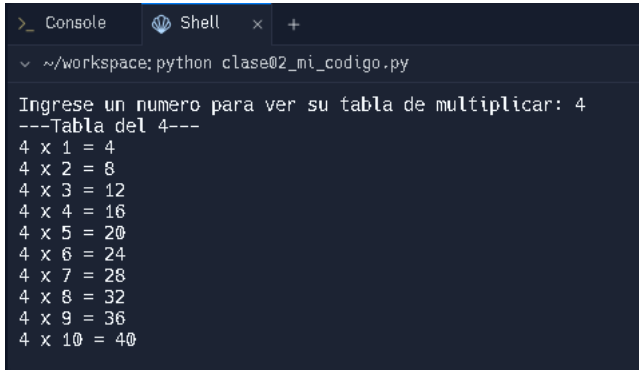
Como puede puse la edad de 5 y me devolvió que podía ver películas con clasificación G o PG

```
class02_mi_codigo.py > ...  
5 """Estudiante Yoel Escalante Escobar"""  
6 print("Escriba su edad")  
7 edad = int(input())  
8 v if edad >= 18:  
9     print("Puedes ver peliculas clasificacion R!")  
10 v if edad >= 13 and edad < 18:  
11     print("Puedes ver peliculas clasificacion PG-13!")  
12 v if edad < 13 and edad >=0:  
13     print("Te recomendamos ver peliculas clasificacion G o PG!")  
AI (→) Python  Diff  
_ Console  Shell x +  
v ~/workspace: python class02_mi_codigo.py  
^  
SyntaxError: invalid character '¡' (U+00A1)  
~/workspace$ python class02_mi_codigo.py  
File "/home/runner/workspace/class02_mi_codigo.py", line 1  
    print(¡Estoy aprendiendo Git y Github en Prog II!);  
          ^  
SyntaxError: invalid character '¡' (U+00A1)  
~/workspace$ python class02_mi_codigo.py  
Escriba su edad  
5  
Te recomendamos ver peliculas clasificacion G o PG!  
~/workspace$
```

Ejercicio 5

Imprimir tablas de multiplicacion del 1 al 10 del número que le digamos

```
34 #ejercicio5
35 num_tabla = int(input("Ingrese un numero para ver su tabla de multiplicar: "))
36 print(f"---Tabla del {num_tabla}---")
37 for i in range(1,11):
38     resultado = num_tabla * i
39     print(f"{num_tabla} x {i} = {resultado}")
40
```



>_ Console Shell x +
~/workspace: python clase02_mi_codigo.py
Ingrese un numero 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

Ejercicio 6

Adivina el numero secreto cada que le pongamos un número dice si es mayor o menor

```
#ejercicio6
sec = 6
adiv = int(input(f"Adivina el numero: "))
while adiv != sec:
    if adiv > sec:
        print("El numero es demasiado alto")
    if adiv < sec:
        print("El numero es demasiado bajo")
    adiv = int(input(f"Vuelva a intentarlo: "))
if adiv == sec:
    print("Adivinaste!")
```

```
40 #ejercicio6
41 sec = 6
42 adiv = int(input(f"Adivina el numero: "))
43 while adiv != sec:
44     if adiv > sec:
```

AI Python Diff

> Console Shell x +

~/workspace: python clase02_mi_codigo.py

```
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
Adivina el numero: 4
El numero es demasiado bajo
Vuelva a intentarlo : 7
El numero es demasiado alto
Vuelva a intentarlo : 6
Adivinaste!
```


Ejercicio 7

Un ejercicio de refactorización y calcula el área de un rectángulo

```
51 #ejercicio6refactorizaci
52 def calcular_area_rectangulo(base, altura):
53     area = base * altura
54     return area
55 def mostrar_area_rectangulo(numero, base, altura):
56     area = calcular_area_rectangulo(base, altura)
57     print(f"El area del rectangulo {numero}({base}x{altura} )es : {area}")
58     mostrar_area_rectangulo(1, 10, 5)
59
```

```
Adivinaste!
El area del rectangulo 1(10x5 )es : 50
Por favor, especifica una función.
~/workspace$
```

Actividad 2 Ejercicio de listas

Ejercicio 1

Imprime la palabras que añadimos en la lista

```
clase04_listas.py / ...
#Listas
#Listas de hobbies
hobbys = ["leer", "viajar", "jugar videojuegos", "escuchar musica", "ver peliculas"]
for nombre in hobbys:
    print(f"Mis hobbies son {nombre}")
print()
```

```
> Console Shell x +
~/workspace: git push origin main

Bienvenido al equipo, Diego!
~/workspace$ python clase04_listas.py
Mis hobbies son leer
Mis hobbies son viajar
Mis hobbies son jugar videojuegos
Mis hobbies son escuchar musica
Mis hobbies son ver peliculas
Bienvenido al equipo, Diego!
```

Ejercicio 2

Añade comidas favoritas y luego las imprime

```
#comidas favoritas
# Lista vacía para almacenar las comidas favoritas
comidas_favoritas = []
cantidad = int(input("¿Cuántas comidas favoritas quieres ingresar? "))
# Usamos un bucle para agregar cada comida
✓ for i in range(cantidad):
    comida = input(f"Ingrese la comida favorita #{i + 1}: ")
    comidas_favoritas.append(comida)
    print("\nMis comidas favoritas son:")
✓ for comida in comidas_favoritas:
    print(f"- {comida}")
print()
```

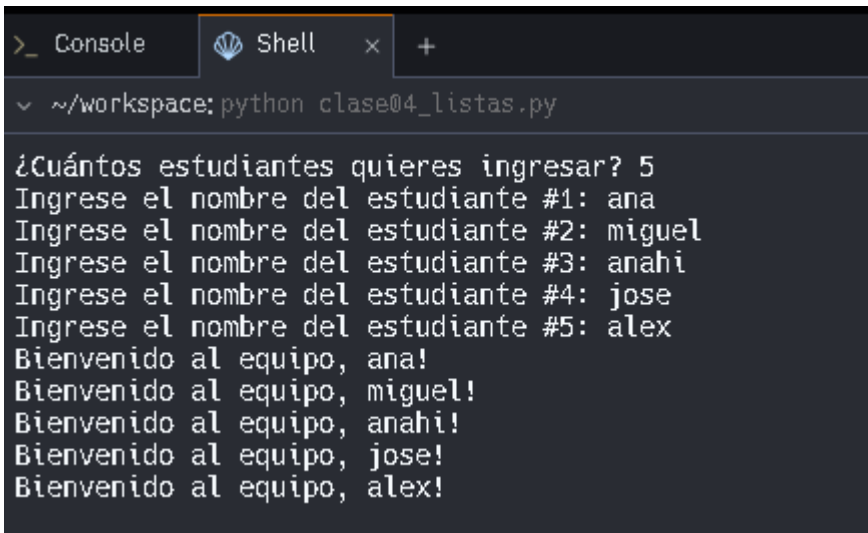
```
¿Cuántas comidas favoritas quieres ingresar? 3
Ingrese la comida favorita #1: picante
Ingrese la comida favorita #2: tallarin
Ingrese la comida favorita #3: hamburguesa

Mis comidas favoritas son:
- picante
- tallarin
- hamburguesa
```

Ejercicio 3

En la lista añadir los nombres de mis compañeros de grupo y luego imprimir

```
#Listas de nombres de estudiantes
nombres_estudiantes = []
cantidad = int(input("¿Cuántos estudiantes quieres ingresar? "))
for i in range(cantidad):
    nombre = input(f"Ingrese el nombre del estudiante #{i + 1}: ")
    nombres_estudiantes.append(nombre)
for nombre in nombres_estudiantes:
    print(f"Bienvenido al equipo, {nombre}!")
print()
```



```
>_ Console  Shell x +
v ~/workspace:python clase04_listas.py

¿Cuántos estudiantes quieres ingresar? 5
Ingrese el nombre del estudiante #1: ana
Ingrese el nombre del estudiante #2: miguel
Ingrese el nombre del estudiante #3: anahi
Ingrese el nombre del estudiante #4: jose
Ingrese el nombre del estudiante #5: alex
Bienvenido al equipo, ana!
Bienvenido al equipo, miguel!
Bienvenido al equipo, anahi!
Bienvenido al equipo, jose!
Bienvenido al equipo, alex!
```

Ejercicio 4

Calcula el promedio de las notas de una lista

```
18  ##Crear una lista de notas numericas
19  mis_notas = [85.5,92,78,88.5,95,82]
20  #inicializar Variable para la suma
21  suma_total = 0
22  #Usar un bucle for para calcular la suma total sin usar sum()
23  for nota in mis_notas:
24      suma_total += nota
25  #calcular el promedio
26  promedio = suma_total / len(mis_notas)
27  #Imprimir resultados de forma clara
28  print(f"Suma total de mis notas {mis_notas} : {suma_total}")
29  print(f"Promedio de las notas de Yoel son: {promedio:.2f}")
30  print()
31  #suma de elementos de un vectores
```

```
Suma total de las notas: 521.0
Promedio de las notas de Yoel son: 86.83
```

Vamos a provocar un error cambiando o tratando de cambiar una valor en una posicion que no existe

```
print()
##Crear una lista de notas numericas
mis_notas = [85.5,92,78,88.5,95,82]
mis_notas[14] = 90
#inicializar Variable para la suma
suma_total = 0
#Usar un bucle for para calcular la suma to
for nota in mis_notas:
    suma_total += nota
```

Cambiamos mis notas en la posición 14 y no existe esa posición

```
Traceback (most recent call last):
  File "/home/runner/workspace/clase04_listas.py", line 19, in <module>
    mis_notas[14] = 90
    ~~~~~^~~~~~
IndexError: list assignment index out of range
~/workspace$
```

Ejercicio 5

Suma todos los elementos dentro un lista/vector

```
#suma de elementos de un vectores
def suma_elementos(vector):
    suma = 0
    for elemento in vector:
        suma += elemento
    return suma

# Casos de Prueba con assert
print("Probando sumar_elementos...")
assert suma_elementos([1, 2, 3, 4, 5]) == 15
assert suma_elementos([10, -2, 5]) == 13
assert suma_elementos([]) == 0 # ¡Importante probar con una lista vacía!
assert suma_elementos([100]) == 100
print("¡Pruebas para sumar_elementos pasaron! ")
vector = [1, 5, 4]
resultado = suma_elementos(vector)
print(f"La suma del vector {vector} es: {resultado}")
```

```
Probando sumar_elementos...
¡Pruebas para sumar_elementos pasaron!
La suma del vector [1, 5, 4] es: 10
```

Ejercicio 6

Encuentra el número mayor dentro de una lista

```
def encontrar_mayor(lista):
    if len(lista) == 0:
        return None
    mayor = lista[0]
    for numero in lista:
        if numero > mayor:
            mayor = numero
    return mayor

print("\nProbando encontrar_mayor...")
assert encontrar_mayor([1, 9, 2, 8, 3, 7]) == 9
assert encontrar_mayor([-1, -9, -2, -8]) == -1
assert encontrar_mayor([42, 42, 42]) == 42
assert encontrar_mayor([]) == None # Prueba del caso especial
assert encontrar_mayor([5]) == 5
print("¡Pruebas para encontrar_mayor pasaron! ")
vector = [1, 5, 4]
resultado = encontrar_mayor(vector)
print(f"El numero mayor del vector {vector} es: {resultado}")
print()
```

```
Probando encontrar_mayor...
¡Pruebas para encontrar_mayor pasaron!
El numero mayor del vector [1, 5, 4] es: 5
```

Ejercicio 7

Contar cuantas veces se repite un elemento dentro de una lista

```
#contar cuantas veces hay un elemento en una lista
def contar_elemento(lista, elemento):
    contador = 0
    for i in lista:
        if i == elemento:
            contador += 1
    if contador == 0:
        return f"El elemento {elemento} no se encuentra en la lista"
    return f"El elemento {elemento} de la lista {lista} se repite {contador} veces"
resultado= contar_elemento([1, 2, 3, 4, 5, 6, 7, 5, 9, 10], 5)
print(resultado)
print()
```

El número mayor del vector [1, 3, 4] es: 4

El elemento 5 de la lista [1, 2, 3, 4, 5, 6, 7, 5, 9, 10] se repite 2 veces

Ejercicio 8

Invertir una lista que los elementos vayan de el último al principio

```
#invertir el orden de una lista
def lista_invertida(lista):
    lista_invertida = []
    for i in range(len(lista)-1, -1, -1):
        lista_invertida.append(lista[i])
    return lista_invertida
print("\nProbando invertir_lista...")
lista_prueba = [1, 2, 3, 4, 5]
lista_resultante = lista_invertida(lista_prueba)
assert lista_resultante == [5, 4, 3, 2, 1]
assert lista_prueba == [1, 2, 3, 4, 5] # ¡Verifica que la original no cambió!
assert lista_invertida(["a", "b", "c"]) == ["c", "b", "a"]
assert lista_invertida([]) == []
print("¡Pruebas para invertir_lista pasaron! ")
vector = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
resultado = lista_invertida(vector)
print(f"Invertir la lista {vector} : la lista invertida es {resultado}")
print()
```

Probando invertir_lista...
¡Pruebas para invertir_lista pasaron!
Invertir la lista [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] : la lista invertida es [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Ejercicio 9

Busca un número en una lista ordenada o desordenada de forma lineal

```
#busqueda secuencial o lineal
def busqueda_lineal(lista, elemento):
    for i in range(len(lista)):
        if lista[i] == elemento:
            return i
    return -1

# ... (definición de la función aquí arriba) ...
mi_lista_desordenada = [10, 5, 42, 8, 17, 30, 25]
print("Probando busqueda_lineal...")
assert busqueda_lineal(mi_lista_desordenada, 42) == 2
assert busqueda_lineal(mi_lista_desordenada, 10) == 0 # Al inicio
assert busqueda_lineal(mi_lista_desordenada, 25) == 6 # Al final
assert busqueda_lineal(mi_lista_desordenada, 99) == -1 # No existe
assert busqueda_lineal([], 5) == -1
print("¡Pruebas para busqueda_lineal pasaron! ")
vector = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
elemento = 5
resultado = busqueda_lineal(vector, elemento)
if resultado == -1:
    print(f"Buscar si el {elemento} esta en la lista {vector} : No se a encontrado el {elemento}")
print(f"Buscar si el {elemento} esta en la lista {vector} : Si se a encontrado el {resultado}")
```

El if del final es para cuando ejecute el programa se vea mas bonito añadiendo oraciones

```
Probando busqueda_lineal...
¡Pruebas para busqueda_lineal pasaron!
Buscar si el 5 esta en la lista [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] : Si se a encontrado el 4
```


Ejercicio 10

Busca un elemento de forma binaria en una lista ordenada

```
def busqueda_binaria(lista, elemento):
    izquierda, derecha = 0, len(lista) - 1
    while izquierda <= derecha:
        medio = (izquierda + derecha) // 2
        if lista[medio] == elemento:
            return medio
        elif lista[medio] < elemento:
            izquierda = medio + 1
        else:
            derecha = medio - 1
    return -1

lista_ordenada = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]
print("\nProbando busqueda_binaria...")
assert busqueda_binaria(lista_ordenada, 23) == 5
assert busqueda_binaria(lista_ordenada, 91) == 9 # Último
assert busqueda_binaria(lista_ordenada, 2) == 0 # Primero
assert busqueda_binaria(lista_ordenada, 3) == -1 # No existe
assert busqueda_binaria(lista_ordenada, 100) == -1 # Fuera de rango (mayor)
print("¡Pruebas para busqueda_binaria pasaron! ")
Lista = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]
elemento = 23
resultado = busqueda_binaria(Lista, elemento)
if resultado == -1:
    print(f"Buscar si el {elemento} esta en la lista {Lista} : No se a encontrado el {elemento}")
print(f"Buscar si el {elemento} esta en la lista {Lista} : Si se a encontrado el {resultado}")
```

```
Probando busqueda_binaria...
¡Pruebas para busqueda_binaria pasaron!
Buscar si el 23 esta en la lista [2, 5, 8, 12, 16, 23, 38, 56, 72, 91] : Si se a encontrado el 5
```