

✓ Actividad 4 Control de flujo en Python con funciones y excepciones

Estudiante: Ramos Santamaria Fatima

Matrícula: 379156

Fecha: 08 de septiembre de 2025

Introducir conceptos básicos de manejo de excepciones (try), ciclos iterativos (for), generación de rangos (range()), y números aleatorios (random()).

✓ Ejercicio 1 Números Aleatorios (Par/Impar)

Enunciado: Generar 40 números aleatorios entre 0 y 200. Desplegar los números y clasificarlos como "par" o "impar". Mostrar la cantidad de números pares e impares, así como la suma de ambos grupos.

Desarrollo: Generar 40 números aleatorios utilizando la función random.randint(0, 200).\

- Clasificar cada número como "par" o "impar" utilizando el operador módulo (%).
- Contabilizar la cantidad de números pares e impares.
- Calcular la suma de los números pares y la suma de los números impares.
- Desplegar los resultados en un formato claro y organizado.

Conceptos a Aplicar: Ciclo for, función random(), validación de par/impar, acumuladores y contadores.

```
import random

par = 0
suma_par = 0
suma_impar = 0
for i in range(40):
    num = random.randint(0,200)
    if num % 2 == 0:
        print(f"{num} -- par")
        par += 1
        suma_par += num
    else:
        print(f"{num} -- impar")
        suma_impar += num

print(f"\nSe ingresaron {par} números pares")
print(f"La suma de los números pares es {suma_par}\n")

print(f"Se ingresaron {40 - par} números impares")
print(f"La suma de los números impares es {suma_impar}")
```

```
7 -- impar
76 -- par
41 -- impar
80 -- par
87 -- impar
161 -- impar
13 -- impar
189 -- impar
95 -- impar
169 -- impar
89 -- impar
157 -- impar
112 -- par
90 -- par
186 -- par
152 -- par
4 -- par
63 -- impar
198 -- par
176 -- par
4 -- par
160 -- par
104 -- par
55 -- impar
35 -- impar
34 -- par
```

```

126 -- par
21 -- impar
108 -- par
12 -- par
183 -- impar
9 -- impar
44 -- par
190 -- par
13 -- impar
143 -- impar
194 -- par
109 -- impar
30 -- par
5 -- impar

```

Se ingresaron 20 números pares
La suma de los números pares es 2080

Se ingresaron 20 números impares
La suma de los números impares es 1644

✓ Ejercicio 2 Tabla de Multiplicar

Enunciado: Desplegar la tabla de multiplicar de un número dado (entre 1 y 20). Ejemplo: Si el número es 5, mostrar:

```

5 * 1 = 5
5 * 2 = 10
...
5 * 10 = 50

```

Desarrollo:

- Solicitar al usuario un número entre 1 y 20.
- Validar que el número esté dentro del rango permitido.
- Utilizar un ciclo for con la función range() para generar los múltiplos del número ingresado.
- Desplegar la tabla de multiplicar en un formato legible.

Conceptos a Aplicar: Ciclo for, función range(), validación de entradas.

```

try:
    for i in range(1,11):
        if i == 1:
            num = int(input("Ingrese un número entre el 1 y el 20 "))

            if num <= 1:
                continue
            if num >= 20:
                continue

            print(f"{num} * {i} = {num * i}")

except ValueError:
    print(f"ERROR Ingrese un número")
except:
    print("ERROR")

```

```

Ingrese un número entre el 1 y el 20 14
14 * 1 = 14
14 * 2 = 28
14 * 3 = 42
14 * 4 = 56
14 * 5 = 70
14 * 6 = 84
14 * 7 = 98
14 * 8 = 112
14 * 9 = 126
14 * 10 = 140

```

✓ Ejercicio 3 Validación de Calificación

Enunciado: Leer una calificación dentro del rango de 0 a 100. Si hay un error de captura, mostrar un mensaje de error. Con la calificación correcta, indicar si está "Aprobado" (≥ 60) o "Reprobado" (< 60).

Desarrollo: \

- Solicitar al usuario una calificación.
- Utilizar un bloque try-except para manejar errores de captura (por ejemplo, si el usuario ingresa texto en lugar de un número).
- Validar que la calificación esté dentro del rango de 0 a 100.
- Determinar si la calificación corresponde a "Aprobado" o "Reprobado".
- Desplegar el resultado.

Conceptos a Aplicar: Manejo de excepciones (try-except), validación de entradas, operadores de comparación.

```
try:
    calif = int(input("Ingrese una calificación "))

    if num <= 0:
        error = 0
    else:
        if num >= 100:
            error = 0

    if calif >= 60:
        print("Aprobado")
    else:
        print("Reprobado")

except ValueError:
    print(f"ERROR Ingrese un número")
except error == 0:
    print("Ingrese un número entre el 0 y el 100")
except:
    print("ERROR")
```

Ingrese una calificación 120
Aprobado

✓ Ejercicio 4 Suma y Media de Números

Enunciado: Leer n números enteros positivos. El programa terminará cuando el usuario introduzca el número cero. Desplegar la suma de los números y su media.

Desarrollo:

- Utilizar un ciclo while para leer números hasta que el usuario ingrese 0.
- Validar que los números ingresados sean positivos.
- Acumular los números ingresados y contar cuántos se han introducido.
- Calcular la suma total y la media de los números.
- Desplegar los resultados.

Conceptos a Aplicar: Ciclo while, acumuladores, contadores, validación de entradas.

```
try:
    num = 1
    suma = 0
    i = 0

    while num != 0:
        num = int(input("Ingrese un numero: "))

        if num <= 0:
            continue

        suma += num
        i += 1

    prom = suma / i

    print(f"Se ingresaron {i} numeros")
    ...print(f"La suma es {suma}")
    ...print(f"El promedio es de {prom}")
```

```
except ValueError:
    print(f"ERROR Ingrese un número")
except:
    print("ERROR")
```

```
Ingrese un numero: 3
Ingrese un numero: 3
Ingrese un numero: 3
Ingrese un numero: 3
Ingrese un numero: 3
Ingrese un numero: 3
Ingrese un numero: 0
Se ingresaron 6 numeros
La suma es 18
El promedio es de 3.0
```

✓ Ejercicio 5 Promedio de Materia

Enunciado: Leer el promedio de una materia. El usuario tiene un máximo de 3 oportunidades para cursar la materia. Si aprueba, felicitarlo y continuar al siguiente semestre. Si reprueba 3 veces, mostrar "Baja Académica".

Desarrollo:

- Solicitar al usuario el promedio de la materia.
- Validar que el promedio esté dentro del rango permitido (0 a 100).
- Usar un ciclo for para permitir hasta 3 intentos.
- Si el usuario aprueba (promedio ≥ 60), felicitarlo y finalizar el programa.
- Si el usuario reprueba 3 veces, mostrar "Baja Académica".

Conceptos a Aplicar: Ciclo for, validación de entradas, manejo de excepciones (try-except).

```
try:
    for i in range(3):
        prom = int(input("Ingrese promedio: "))
        if prom < 0:
            continue
        if prom > 100:
            continue

        if prom >= 60:
            print("Felicidades, aprobaste")
            break
        elif i == 2:
            print("Baja academica")

except ValueError:
    print(f"ERROR Ingrese un número")
except:
    print("ERROR")
```

```
Ingrese promedio: 59
Ingrese promedio: 70
Felicidades, aprobaste
```

✓ Ejercicio 6 Función para Suma, Media, Mayor y Menor

Enunciado: Crear una función que lea n números hasta que el usuario lo desee. Desplegar la suma, media, valor mayor y menor de los números ingresados.

Desarrollo:

- Crear una función que utilice un ciclo while para leer números hasta que el usuario decida detenerse.
- Calcular la suma, media, valor mayor y menor de los números ingresados.
- Desplegar los resultados.

Conceptos a Aplicar: Funciones personalizadas, ciclo while, acumuladores, contadores.

```
def numeros():
    ...try:
    .....suma = 0
```

```

.....band==1
.....mayor==suma
.....menor==None
.....n==0

.....while band==1:
.....    print("Menu")
.....    print("1--Ingresar un numero")
.....    print("0--Detener")

.....    band=int(input(""))
.....    if band==0:
.....        break
.....    elif band==1:
.....        pass
.....    else:
.....        print("ERROR")
.....        break

.....    num=int(input("Ingrese un numero: "))

.....    if num>mayor:
.....        mayor=num
.....    if menor is None or num<menor:
.....        menor=num

.....    suma+=num
.....    n+=1
.....
.....    prom=suma/n

.....    print(f"La suma es {suma}")
.....    print(f"El promedio es de {prom}")
.....    print(f"El mayor es {mayor}")
.....    print(f"El menor es {menor}")

...except ValueError:
.....    print(f"ERROR Ingrese un número")
...except:
.....    print("ERROR")

```

numeros()

```

Menu
1 -- Ingresar un numero
0 -- Detener
1
Ingrese un numero: 4
Menu
1 -- Ingresar un numero
0 -- Detener
1
Ingrese un numero: 2
Menu
1 -- Ingresar un numero
0 -- Detener
1
Ingrese un numero: 5
Menu
1 -- Ingresar un numero
0 -- Detener
1
Ingrese un numero: 7
Menu
1 -- Ingresar un numero
0 -- Detener
1
Ingrese un numero: 1
Menu
1 -- Ingresar un numero
0 -- Detener
0
La suma es 19
El promedio es de 3.8
El mayor es 7
El menor es 1

```

✓ Ejercicio 7 Generación de Números Impares

Enunciado: Crear una función que genere 15 números impares entre 10 y 60 o un máximo de 25 números. Desplegar la media de los números pares e impares.

Desarrollo:

- Crear una función que utilice un ciclo for y la función random.randint() para generar números impares dentro del rango especificado.
- Contabilizar los números generados y calcular la media de los números pares e impares.
- Desplegar los resultados.

Conceptos a Aplicar: Funciones personalizadas, ciclo for, función random(), acumuladores.

```
import random
def numeros_random():
    try:
        impar = 0
        suma_par = 0
        suma_impar = 0

        for i in range(1, 26):
            num = random.randint(10,60)
            print(f"{i}. --> {num}")

            if num % 2 != 0:
                impar += 1
                suma_impar += num
            elif num % 2 == 0:
                suma_par += num

            if impar >= 10:
                break

        prom_impar = suma_impar / impar
        prom_par = suma_par / (i - impar)

        print(f"Numeros pares: {i - impar}")
        print(f"Numeros impares: {impar}")
        print(f"Promedio numeros pares: {prom_par:.2f}")
        print(f"Promedio numeros impares: {prom_impar:.2f}")

    except ValueError:
        print(f"ERROR Ingrese un número")
    except:
        print("ERROR")
```

numeros_random()

```
1. --> 49
2. --> 29
3. --> 21
4. --> 21
5. --> 36
6. --> 52
7. --> 35
8. --> 38
9. --> 29
10. --> 16
11. --> 13
12. --> 24
13. --> 16
14. --> 39
15. --> 21
16. --> 11
Numeros pares: 6
Numeros impares: 10
Promedio numeros pares: 30.33
Promedio numeros impares: 26.80
```

✓ Ejercicio 8 Validación de Número en Rango

Enunciado: Crear una función que valide un número dentro de un rango dado por el usuario. Repetir esta acción hasta que el usuario lo desee. Desplegar la cantidad de números ingresados y su promedio.

Desarrollo:

- Crear una función que solicite al usuario un número y un rango (mínimo y máximo).
- Validar que el número esté dentro del rango especificado.
- Permitir al usuario repetir el proceso hasta que decida detenerse.
- Calcular la cantidad de números ingresados y su promedio.
- Desplegar los resultados.

Conceptos a Aplicar: Funciones personalizadas, validación de entradas, ciclo while.

```
def rango(minimo, maximo):
    try:
        suma = 0
        n = 0
        band = 1

        while band == 1:
            print("Menú")
            print("1 -- Ingresar un número")
            print("0 -- Detener")

            band = int(input("Seleccione una opción: "))

            if band == 0:
                break
            elif band == 1:
                num = int(input("\n    Ingrese un número: "))

                if num < minimo or num > maximo:
                    print("El número está fuera del rango")
                    continue
                else:
                    suma += num
                    n += 1
            else:
                print("ERROR")
                break

        prom = suma / n

        print(f"\nNúmeros ingresados: {n}")
        print(f"Promedio: {prom:.2f}\n")

    except ValueError:
        print("ERROR Ingrese un número válido")
    except:
        print("ERROR")
```

```
try:
    minimo = int(input("Ingrese el valor mínimo del rango: "))
    ...maximo = int(input("Ingrese el valor máximo del rango: "))

    rango(minimo, maximo)

except ValueError:
    print(f"ERROR Ingrese un número")
except:
    print("ERROR")
```

```
Ingrese el valor mínimo del rango: 4
Ingrese el valor máximo del rango: 8
Menú
1 -- Ingresar un número
0 -- Detener
Seleccione una opción: 1

    Ingrese un número: 4
Menú
1 -- Ingresar un número
0 -- Detener
Seleccione una opción: 1
```

```

    Ingrese un número: 3
    El número está fuera del rango
    Menú
    1 -- Ingresar un número
    0 -- Detener
    Seleccione una opción: 1

    Ingrese un número: 6
    Menú
    1 -- Ingresar un número
    0 -- Detener
    Seleccione una opción: 1

    Ingrese un número: 8
    Menú
    1 -- Ingresar un número
    0 -- Detener
    Seleccione una opción: 1

    Ingrese un número: 2
    El número está fuera del rango
    Menú
    1 -- Ingresar un número
    0 -- Detener
    Seleccione una opción: 0

    Números ingresados: 3
    Promedio: 6.00

```

✓ Ejercicio 9 Área de un Triángulo

Enunciado: Crear una función que reciba como parámetros los valores para calcular el área de un triángulo y retorne su resultado.

Desarrollo:

- Crear una función que reciba como parámetros la base y la altura de un triángulo.
- Calcular el área utilizando la fórmula: $(base * altura) / 2$.
- Retornar el resultado.
- Llamar a la función desde el programa principal y desplegar el resultado.

Conceptos a Aplicar: Funciones personalizadas, fórmulas matemáticas, retorno de valores.

```

def area_triangulo(base, altura):
    try:
        area = (base * altura) / 2
        return area

    except:
        print("ERROR")

```

```

try:
    base = int(input("Ingrese la base del triangulo: "))
    altura = int(input("Ingrese la altura del triangulo: "))

    print(f"El area es: {area_triangulo(base, altura)} u^2")

except ValueError:
    print(f"ERROR Ingrese un número")
except:
    print("ERROR")

```

```

Ingrese la base del triangulo: 3
Ingrese la altura del triangulo: 6
El area es: 9.0 u^2

```

✓ Ejercicio 10 Validación de Número en Rango

Enunciado: Crear una función que valide un número dentro de un rango dado.

Desarrollo:

- Crear una función que solicite al usuario un número y un rango (mínimo y máximo).
- Validar que el número esté dentro del rango especificado.
- Desplegar un mensaje indicando si el número es válido o no.

Conceptos a Aplicar: Funciones personalizadas, validación de entradas.

```
def validar_rango(num, minimo, maximo):  
    try:  
        if num < minimo or num > maximo:  
            print("Numero fuera de rango")  
        else:  
            print("Numero valido")  
  
    except:  
        print("ERROR")
```

```
try:  
    minimo = int(input("Ingrese el valor mínimo del rango: "))  
    maximo = int(input("Ingrese el valor máximo del rango: "))  
  
    num = int(input("Ingrese un numero: "))  
  
    validar_rango(num, minimo, maximo)  
  
except ValueError:  
    print(f"ERROR Ingrese un número")  
except:  
    print("ERROR")
```

```
Ingrese el valor mínimo del rango: 4  
Ingrese el valor máximo del rango: 8  
Ingrese un numero: 3  
Numero fuera de rango
```