Actividad 4: Control de flujo PYTHON con funciones y excepciones

Nombre: Alejandro Ramirez Cruz

Matricula: 379551

Fecha: Lunes 08 septiembre de 2025

Ejercicio 1: Números aleatorios par e impar

Descripción:

Generar 40 números aleatorios entre 0 y 200. Desplegar los números y clasificarlos como "par" o "impar".

[Explicación del problema]:

Se generan 40 números aleatorios entre 0-200, luego se imprimen dichos números y se clasifican en pares e impares segun el caso del número, al final se contabiliza el total de c/u.

Solución:

Se importa la biblioteca random, la cual permite generar números aleatorios. Mediante un ciclo for se generan 40 números aleatorios en el rango de 0 a 200. Cada número se valida para determinar si es par o impar, y con ayuda de contadores (para contar la cantidad) y acumuladores (para sumar los valores) se van registrando los resultados. Además, los números se guardan en listas separadas: una para pares y otra para impares. Finalmente, el programa imprime:

La lista completa de números generados.

Los números clasificados en pares e impares.

El total y la suma de pares.

El total y la suma de impares.

```
import random   # Importar la librería para generar números aleatorios
```

```
def numeros_aleatorios():     #Funcion creada
    # Contadores
    par = 0
    impar = 0

    # Acumuladores de sumas
    suma_par = 0
    suma_impar = 0

    # Listas para almacenar los números
    numeros = []          # Lista con todos los números generados
    lista_pares = []      # Lista solo para los pares
    lista_impares = []    # Lista solo para los impares

    # Generar 40 números aleatorios
    for i in range(40):
        num_random = random.randint(0, 200)   # Genera un número aleatorio entre 0 y 200
        numeros.append(num_random)            # Agrega el número generado a la lista "numeros"

        # Clasificar entre par o impar
        if num_random % 2 == 0:
            par += 1                          # Contador de pares
            suma_par += num_random            # Acumula la suma de pares
            lista_pares.append(num_random)    # Guarda el número en la lista de pares
        else:
            impar += 1                        # Contador de impares
            suma_impar += num_random          # Acumula la suma de impares
            lista_impares.append(num_random)  # Guarda el número en la lista de impares

    # Mostrar resultados
    print("=== NÚMEROS GENERADOS ===")
    print(numeros)
    print("\n=== CLASIFICACIÓN ===")
    print("Pares: ", lista_pares)
    print("Impares:", lista_impares)

    print("\n=== RESULTADOS ===")
    print(f"Total de pares: {par}")
    print(f"Total de impares: {impar}")
    print(f"Suma de pares: {suma_par}")
```

```
        print(f"Suma de impares: {suma_impar}")
```

```
numeros_aleatorios()        # Mandar a llamar la función
```

```
=== NÚMEROS GENERADOS ===
[5, 11, 65, 47, 153, 8, 121, 16, 82, 91, 63, 185, 69, 15, 172, 54, 84, 43, 60, 183, 27, 5, 186, 153, 84, 5, 151, 187, 175, 30,

=== CLASIFICACIÓN ===
Pares:  [8, 16, 82, 172, 54, 84, 60, 186, 84, 30, 164, 172, 62, 44, 180]
Impares: [5, 11, 65, 47, 153, 121, 91, 63, 185, 69, 15, 43, 183, 27, 5, 153, 5, 151, 187, 175, 121, 189, 119, 181, 95]

=== RESULTADOS ===
Total de pares: 15
Total de impares: 25
Suma de pares: 1398
Suma de impares: 2459
```

Ejercicio 2: Tabla de Multiplicar

Descripción:

Desplegar la tabla de mutiplicar de un numero dado entre 1 y 20.

[Explicación del problema]:

Se solicita al usuario un número comprendido entre 1 y 20. Luego, se valida que el número ingresado esté dentro del rango permitido. Si es válido, mediante un ciclo for se generan los múltiplos del número, mostrando su tabla de multiplicar desde 1 hasta 10. Finalmente, se despliega en pantalla la tabla correspondiente. En caso de que el número no esté en el rango, el programa indica que el valor ingresado es inválido.

Solución:

El programa solicita al usuario ingresar un número entre 1 y 20, que corresponde a la tabla de multiplicar que desea visualizar. Primero se valida que el número ingresado esté dentro del rango permitido. Si es correcto, mediante un ciclo for se generan y muestran en pantalla las multiplicaciones del número elegido desde 1 hasta 10. En caso de que el valor no se encuentre dentro del rango, el programa muestra un mensaje de error indicando que el número está fuera de los límites establecidos.

```
def tabla_multiplicar():     # Funcion creada

    num = int(input("Ingrese su tabla deseada (1-20): "))   #Pedir al usario la tabla deseada

    if 1 <= num <= 20:     # Validar que el numero este entre el rango dado
        for i in range(1, 11):     #Imprimir la tabla del numero elegido
            print(f"{num} × {i} = {num * i}")
    else:
        print("Número fuera del rango (1-20).")    # Si el numero esta fuera del rango se imprime el siguiente mensaje
```

```
tabla_multiplicar()     # Mandar a llamar a la funcion
```

```
Ingrese su tabla deseada (1-20): 7
7 × 1 = 7
7 × 2 = 14
7 × 3 = 21
7 × 4 = 28
7 × 5 = 35
7 × 6 = 42
7 × 7 = 49
7 × 8 = 56
7 × 9 = 63
7 × 10 = 70
```

Ejercicio 3: Validacion de calificacion

Descripción:

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).

[Explicación del problema]:

Se solicita al usuario una calificación. Por medio de un bloque try-except se manejan errores de captura (por ejemplo, si el usuario ingresa texto en lugar de un número). Validamos que la calificación esté dentro del rango de 0 a 100. Se determina si la calificación corresponde

a "Aprobado" o "Reprobado", y se despliega el resultado.

Solución:

El programa solicita al usuario ingresar su calificación mediante la función input(). Primero se utiliza un bloque try-except para asegurarse de que el valor ingresado sea un número entero; si no lo es, se muestra un mensaje de error y la función termina.

Luego, se realiza la validación de la entrada para comprobar que la calificación esté dentro del rango permitido (0 a 100). Si la calificación es válida, se utiliza un operador condicional (if-else) para determinar si el estudiante aprobó o reprobó: imprime "Aprobado" si la calificación es mayor o igual a 60, y "Reprobado" si es menor.

Si la calificación ingresada está fuera del rango, se muestra un mensaje de error indicando que el número debe estar entre 0 y 100.

```
def validar_cal():     # Funcion creada
    try:
        # Solicitar al usuario que ingrese su calificación
        cal = int(input("Ingrese su calificación: "))
    except ValueError:
        # Si el usuario ingresa algo que no sea un número
        print("Error: Debe ingresar solo números.")
        return  # Termina la función si hubo error

    # Validar que la calificación esté dentro del rango permitido (0-100)
    if 0 <= cal <= 100:
        # Verificar si el estudiante aprobó o reprobó
        if cal >= 60:
            print("Aprobado")    # Imprime aprobado si su calificacion es mayor igual a 60
        else:
            print("Reprobado")   # Imprime reprobado si su calificacion es menor a 60
    else:
        print("Error: la calificación debe estar entre 0 y 100.")  # Mensaje si la calificación está fuera del rango
```

```
validar_cal()    # Mandar a llamar a la función
```

```
Ingrese su calificación: 12
Reprobado
```

Ejercicio 4: Suma y media de numeros

Descripción:

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

[Explicación del problema]:

Se utliza un ciclo while para leer numeros hasta que el usuario ingrese 0. Se valida que los numeros sean positivos, se acomulan y se cuentan, despues se calcula la suma total y la media de los numeros. Finalmente se imprimen los resultados.

Solución:

El programa solicita al usuario ingresar números enteros positivos hasta que ingrese 0, utilizando un ciclo while. Se valida con try-except que la entrada sea un número y que sea positiva; si no, se muestra un mensaje de error y se pide otro número.

Cada número válido se acumula en un acumulador y se cuenta con un contador. Al finalizar, se calcula la media y se muestran la suma total, la media y la cantidad de números ingresados. Si no se ingresó ningún número válido, se indica que se intente de nuevo.

```
def suma_y_media():    # Funcion creada
    # Inicializar variables
    suma = 0          # Acumulador para la suma de números
    contador = 0      # Contador para la cantidad de números ingresados

    while True:   # Ciclo infinito hasta que el usuario escriba 0
        try:
            num = int(input("Ingrese un número positivo (0 para terminar): "))  # Solicita al usuario un número entero positi
        except ValueError:
            print("Error: debe ingresar solo números enteros.")  # Si el usuario ingresa algo que no es un número entero
            continue   # Regresa al inicio del ciclo

        if num < 0:     # Se valida que el número no sea negativo
            print("Solo se permiten números positivos.")
            continue   # Regresa al inicio del ciclo

        # Condición de salida: si el usuario ingresa 0
        if num == 0:
            break
```

```
            # Acumular la suma y aumentar el contador
            suma += num
            contador += 1

        if contador > 0:     # Se verifica que se ingresaron números válidos
            media = suma / contador   # Calcular la media
            # Mostrar resultados
            print(f"\nSuma total = {suma}")   # Imprime la suma total de los numeros
            print(f"Media = {media:.2f}")   # Imprime la media de ls numeros
            print(f"Cantidad de números ingresados = {contador}")  # Imprime el total de numeros ingresados
        else:
            print("\nIntente de nuevo.")    # Se imprime si no se ingresaron números válidos
```

```
suma_y_media()      # Mandar a llamar a la funcion
```

```
Ingrese un número positivo (0 para terminar): 13
Ingrese un número positivo (0 para terminar): 4
Ingrese un número positivo (0 para terminar): 5
Ingrese un número positivo (0 para terminar): 0

Suma total = 22
Media = 7.33
Cantidad de números ingresados = 3
```

Ejercicio 5: Promedio de materia

Descripción:

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".

[Explicación del problema]:

Se solicita al usuario el promedio de la materia. Se valida que el promedio esté dentro del rango permitido (0 a 100). Mediante el uso de un ciclo for para permitir hasta 3 intentos, si el usuario aprueba (promedio ≥ 60), felicitarlo y finalizar el programa; y si el usuario reprueba 3 veces, mostrar "Baja Académica".

Solución:

El programa permite al usuario ingresar su calificación hasta 3 intentos, utilizando un ciclo for. Se valida con try-except que la entrada sea un número entero; si no, se muestra un mensaje de error y se repite el intento.

Cada calificación se comprueba para asegurarse de que esté dentro del rango 0-100. Si la calificación es mayor o igual a 60, se imprime un mensaje de aprobación y el programa termina. Si es menor, se indica que se reprobó y se permite otro intento.

Si después de 3 intentos el usuario no aprueba, el programa muestra un mensaje de Baja Académica.

```
def prom_mat():    # Funcion creada
    for i in range(1, 4):  # Permite un máximo de 3 intentos
        try:
            cal = int(input(f"Intento {i} - Ingrese su calificación: ")) # Solicita al usuario su calificación para este inte
        except ValueError:
            # Captura errores si el usuario no ingresa un número entero
            print("Error: Debe ingresar solo números.")
            continue  # No cuenta como intento válido, pide de nuevo

        if 0 <= cal <= 100:     # Verifica que la calificación esté dentro del rango permitido
            if cal >= 60:   # Verifica si la calificación  es aprobatoria
                print("Felicidades, has aprobado.")
                return  # Termina la función si aprobó
            else:        # Entra si la calificación no es aprobatoria
                print("Reprobaste esta vez.")
        else:
            # Mensaje si la calificación está fuera del rango 0-100
            print("Error: la calificación debe estar entre 0 y 100.")
            continue  # No cuenta como intento válido

    # Mensaje si no se aprueba después de 3 intentos
    print("Baja Académica")
```

```
prom_mat()    # Mandar a llamar a la funcion
```

```
Intento 1 - Ingrese su calificación: 1
Reprobaste esta vez.
Intento 2 - Ingrese su calificación: 23
Reprobaste esta vez.
Intento 3 - Ingrese su calificación: 34
```

```
Reprobaste esta vez.
Baja Académica
```

Ejercicio 6: Funcion para suma, media, mayor y menor

Descripción:

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.

[Explicación del problema]:

Se crea una función que utilice un ciclo while para leer números hasta que el usuario decida detenerse. Se calcula la suma, media, valor mayor y menor de los números ingresados. Y se imprimen los resultados.

Solución:

El programa permite ingresar números enteros hasta que se ingrese 0, usando un ciclo while y validando con try-except que sean números válidos. Cada número se acumula y se cuenta, mientras se determina el mayor y el menor. Al finalizar, si se ingresaron números, se calcula la media y se muestran el mayor, el menor y la media; si no, se indica que no se ingresaron números.

```
def suma_media_mayor_y_menor():     # Funcion creada
    numeros = []           # Lista para almacenar los números ingresados
    suma = 0               # Acumulador para la suma de los números
    contador = 0           # Contador de números ingresados

    while True:            # Ciclo infinito hasta que el usuario ingrese 0
        try:
            num = int(input("Ingrese un número (0 para salir): "))  # Solicita un número entero
        except ValueError:
            print("Error: debe ingresar un número entero.")         # Captura errores de tipo
            continue                                                  # Vuelve al inicio del ciclo

        if num == 0:           # Condición de salida del ciclo
            break

        numeros.append(num)    # Agrega el número a la lista
        suma += num            # Acumula la suma
        contador += 1          # Incrementa el contador

        if contador == 1:      # Inicializa mayor y menor con el primer número
            mayor = num
            menor = num
        else:
            if num > mayor:    # Actualiza el mayor si corresponde
                mayor = num
            if num < menor:    # Actualiza el menor si corresponde
                menor = num

    if contador > 0:           # Si se ingresaron números
        media = suma / contador # Calcula la media
        print(f"El número mayor ingresado es: {mayor}")   # Muestra el mayor
        print(f"El número menor ingresado es: {menor}")   # Muestra el menor
        print(f"La media de los números es: {media}")     # Muestra la media
    else:
        print("No se ingresaron números.")               # Mensaje si no se ingresó nada
```

```
suma_media_mayor_y_menor()    # Mandar a llamar a la funcion
```

```
Ingrese un número (0 para salir): 53
Ingrese un número (0 para salir): 5
Ingrese un número (0 para salir): 4
Ingrese un número (0 para salir): 0
El número mayor ingresado es: 53
El número menor ingresado es: 4
La media de los números es: 20.666666666666668
```

Ejercicio 7: Generacion de numeros impares

Descripción:

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.

[Explicación del problema]:

Se crea una función que utilice un ciclo while para leer números hasta que el usuario decida detenerse. Se calcula la suma, media, valor mayor y menor de los números ingresados. Y se imprimen los resultados.

Solución:

El programa genera hasta 25 números aleatorios entre 10 y 60 usando un ciclo for y la función random.randint(). Cada número se clasifica como par o impar y se almacena en su respectiva lista.

Se usan acumuladores y contadores para ir sumando y contando los números pares e impares. Cuando se alcanzan 15 números impares, el ciclo se detiene.

Al final, se calculan las medias de los números pares e impares y se muestran en pantalla junto con las listas generadas.

```
import random  # Importar la libreria random para generar números aleatorios
```

```
def generar_impares():    # Funcion creada
    impares = []          # Lista para almacenar números impares
    pares = []            # Lista para almacenar números pares
    suma_impares = 0      # Acumulador de números impares
    suma_pares = 0        # Acumulador de números pares
    contador_impares = 0  # Contador de impares
    contador_pares = 0    # Contador de pares

    for i in range(25):  # Máximo 25 intentos para generar números
        num = random.randint(10, 60)  # Genera un número aleatorio entre 10 y 60

        if num % 2 != 0:              # Verifica si el número es impar
            impares.append(num)       # Agrega a la lista de impares
            suma_impares += num       # Acumula la suma de impares
            contador_impares += 1     # Incrementa el contador de impares

        else:                          # Si es par
            pares.append(num)          # Agrega a la lista de pares
            suma_pares += num          # Acumula la suma de pares
            contador_pares += 1        # Incrementa el contador de pares

        if contador_impares == 15:    # Condición para detener si hay 15 impares
            break

    # Calcular medias
    media_impares = suma_impares / contador_impares if contador_impares > 0 else 0  # Media de impares
    media_pares = suma_pares / contador_pares if contador_pares > 0 else 0        # Media de pares

    # Mostrar resultados
    print(f"Números impares generados: {impares}")     # Lista de impares
    print(f"Números pares generados: {pares}")         # Lista de pares
    print(f"Media de impares: {media_impares:.2f}")    # Media de impares
    print(f"Media de pares: {media_pares:.2f}")        # Media de pares
```

```
generar_impares()     # Mandar a llamar a la funcion
```

```
Números impares generados: [23, 19, 25, 27, 31, 59, 49, 31, 21, 39, 53]
Números pares generados: [40, 38, 10, 60, 12, 34, 32, 60, 10, 58, 48, 32, 34, 12]
Media de impares: 34.27
Media de pares: 34.29
```

Ejercicio 8: Validacion de numero en rango

Descripción:

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.

[Explicación del problema]:

Se crea una función que solicite al usuario un número y un rango (mínimo y máximo), y se valida que el número esté dentro del rango especificado. Se permite al usuario repetir el proceso hasta que decida detenerse. Se calcula la cantidad de números ingresados y su promedio. Y se imprime los resultados.

Solución:

El programa permite al usuario ingresar números dentro de un rango definido por el usuario. Se utiliza un ciclo while para pedir números hasta que se ingrese 0, y se valida con try-except que la entrada sea un número entero.

Cada número dentro del rango se acumula en un total y se cuenta con un contador. Al finalizar, se muestran la suma total, el número de entradas válidas y, si corresponde, el promedio de los números válidos.

```
def validar_numrango():         # Funcion creada
    numeros = 0                 # Contador de números válidos ingresados
    total_num_ingresado = 0     # Acumulador de la suma de los números válidos

    try:
        print("Ingrese un rango mínimo y máximo")          # Se manda mensaje
        min_val = int(input("Ingrese un valor mínimo: "))  # Solicita el valor mínimo
        max_val = int(input("Ingrese un valor máximo: "))  # Solicita el valor máximo
    except ValueError:
        print("Error: Debe ingresar valores numéricos para el rango.")  # Captura errores
        return  # Termina la función si hay error en el rango

    while True:  # Ciclo hasta que el usuario ingrese 0
        try:
            num = int(input(f"Ingrese un número entre {min_val} y {max_val} (0 para salir): "))  # Solicita número
        except ValueError:
            print("Error: Caracter no reconocido.")  # Error si no es número entero
            continue  # Vuelve a solicitar el número

        if num == 0:  # Condición de salida del ciclo
            break

        if min_val <= num <= max_val:  # Verifica que el número esté dentro del rango
            total_num_ingresado += num  # Acumula la suma
            numeros += 1                # Incrementa el contador de válidos
        else:
            print(f"El número {num} está fuera del rango definido ({min_val}-{max_val}).")  # Mensaje de error

    # Mostrar resultados
    print(f"\nLa suma de los números válidos ingresados es: {total_num_ingresado}")
    print(f"El total de números válidos ingresados es: {numeros}")

    if numeros > 0:  # Calcula y muestra el promedio si hay números válidos
        promedio = total_num_ingresado / numeros
        print(f"El promedio de los números válidos es: {promedio:.2f}")
    else:
        print("No se ingresaron números válidos para calcular el promedio.")  # Mensaje si no hubo números válidos
```

```
validar_numrango()      # Mandar a llamar a la funcion
```

```
Ingrese un rango mínimo y máximo
Ingrese un valor mínimo: 1
Ingrese un valor máximo: 100
Ingrese un número entre 1 y 100 (0 para salir): 99
Ingrese un número entre 1 y 100 (0 para salir): 999
El número 999 está fuera del rango definido (1-100).
Ingrese un número entre 1 y 100 (0 para salir): 35
Ingrese un número entre 1 y 100 (0 para salir): 34
Ingrese un número entre 1 y 100 (0 para salir): 34
Ingrese un número entre 1 y 100 (0 para salir): 56
Ingrese un número entre 1 y 100 (0 para salir): 0

La suma de los números válidos ingresados es: 258
El total de números válidos ingresados es: 5
El promedio de los números válidos es: 51.60
```

Ejercicio 9: Area de un triangulo

Descripción:

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

[Explicación del problema]:

Se crea una función que reciba como parámetros la base y la altura de un triángulo. Luego se calcula el área utilizando la fórmula: (base * altura) / 2 Regresa el resultado, y se llama a la función desde el programa principal y desplegar el resultado.

Solución:

El programa calcula el área de un triángulo utilizando una función personalizada que recibe la base y la altura como parámetros. La función aplica la fórmula matemática (base * altura) / 2 y retorna el resultado.

En el programa principal se solicita al usuario ingresar la base y la altura, se validan como números, se llama a la función y se muestra el área calculada en pantalla.

```
def area_triangulo(base, altura):  # Funcion creada con los parametros
    area = (base * altura) / 2   # Calcula el área usando la fórmula (base*altura)/2
    return area                  # Regresa el área calculada
```

```
# Programa principal
try:
    base = float(input("Ingrese la base del triángulo: "))     # Solicita la base
    altura = float(input("Ingrese la altura del triángulo: ")) # Solicita la altura
except ValueError:
    print("Error: Debe ingresar un número válido.")            # Captura errores si no es número
else:
    resultado = area_triangulo(base, altura)                  # Mandar a llamar a la función con los parámetros
    print(f"El área del triángulo es: {resultado:.2f}")       # Imprime el resultado del area
```

```
Ingrese la base del triángulo: 10
Ingrese la altura del triángulo: 12
El área del triángulo es: 60.00
```

Ejercicio 10: Validacion de numero en rango

Descripción:

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

[Explicación del problema]:

Se crea una función que solicite al usuario un número y un rango (mínimo y máximo). Se valida que el número esté dentro del rango especificado. Y se despliega un mensaje indicando si el número es válido o no.

Solución:

El programa permite al usuario ingresar números dentro de un rango definido por el usuario. Se utiliza un ciclo while para pedir números hasta que se ingrese 0, y se valida con try-except que la entrada sea un número entero.

```
def validar_num_rango():      # Funcion creada
    try:
        print("Ingrese un rango mínimo y máximo")               # Mensaje para indicar que se va a ingresar el rango
        min_val = int(input("Ingrese un valor mínimo: "))       # Solicita el valor mínimo del rango
        max_val = int(input("Ingrese un valor máximo: "))       # Solicita el valor máximo del rango
    except ValueError:
        print("Error: Debe ingresar valores numéricos para el rango.")  # Captura si el usuario no ingresa números
        return                                                  # Termina la función si hay error

    while True:                                                 # Ciclo infinito para ingresar números hasta salir
        try:
            num = int(input(f"Ingrese un número entre {min_val} y {max_val} (0 para salir): "))  # Solicita un número
        except ValueError:
            print("Error: Caracter no reconocido, intente de nuevo.")  # Captura si el número no es válido
            continue                                              # Continúa el ciclo sin evaluar nada más

        if num == 0:                                            # Entra aqui si el usuario ingresa 0
            print("Saliendo del programa...")                   # Mensaje al salir del ciclo
            break                                               # Sale del ciclo
        elif min_val <= num <= max_val:
            print(f"El número {num} es válido")                 # Indica que el número ingresado está dentro del rango
        else:
            print(f"El número {num} está fuera del rango definido ({min_val}-{max_val}).")  # Indica que el número no es válic
```

```
validar_num_rango()      # Mandar a llamar a la funcion
```

```
Ingrese un rango mínimo y máximo
Ingrese un valor mínimo: 1
Ingrese un valor máximo: 100
Ingrese un número entre 1 y 100 (0 para salir): 124
El número 124 está fuera del rango definido (1-100).
Ingrese un número entre 1 y 100 (0 para salir): 24
El número 24 es válido
Ingrese un número entre 1 y 100 (0 para salir): 45
El número 45 es válido
Ingrese un número entre 1 y 100 (0 para salir): 55
El número 55 es válido
Ingrese un número entre 1 y 100 (0 para salir): 0
Saliendo del programa...
```