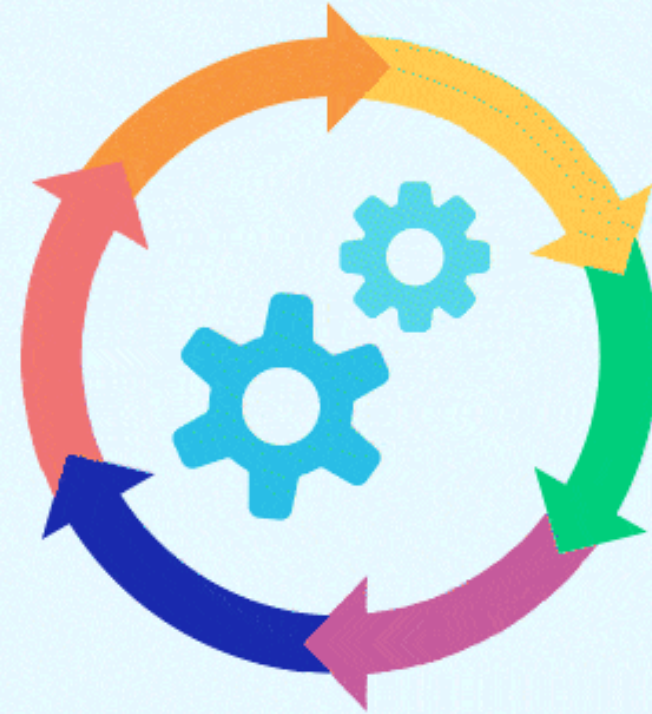


# EIN413B

## Programación



# UVA 3 Ciclos



# ¿Dónde Estamos?



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

SEMANA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
1	10-03-2025	11-03-2025	12-03-2025	13-03-2025	14-03-2025	15-03-2025	16-03-2025
2	17-03-2025	18-03-2025	19-03-2025	20-03-2025	21-03-2025	22-03-2025	23-03-2025
3	24-03-2025	25-03-2025	26-03-2025	27-03-2025	28-03-2025	29-03-2025	30-03-2025
4	31-03-2025	01-04-2025	02-04-2025	03-04-2025	04-04-2025	05-04-2025	06-04-2025
5	07-04-2025	08-04-2025	09-04-2025	10-04-2025	11-04-2025	12-04-2025	13-04-2025
6	14-04-2025	15-04-2025	16-04-2025	17-04-2025	18-04-2025	19-04-2025	20-04-2025
7	21-04-2025	22-04-2025	23-04-2025	24-04-2025	25-04-2025	26-04-2025	27-04-2025
8	28-04-2025	29-04-2025	30-04-2025	01-05-2025	02-05-2025	03-05-2025	04-05-2025
9	05-05-2025	06-05-2025	07-05-2025	08-05-2025	09-05-2025	10-05-2025	11-05-2025
10	12-05-2025	13-05-2025	14-05-2025	15-05-2025	16-05-2025	17-05-2025	18-05-2025
11	19-05-2025	20-05-2025	21-05-2025	22-05-2025	23-05-2025	24-05-2025	25-05-2025
12	26-05-2025	27-05-2025	28-05-2025	29-05-2025	30-05-2025	31-05-2025	01-06-2025
13	02-06-2025	03-06-2025	04-06-2025	05-06-2025	06-06-2025	07-06-2025	08-06-2025
14	09-06-2025	10-06-2025	11-06-2025	12-06-2025	13-06-2025	14-06-2025	15-06-2025
15	16-06-2025	17-06-2025	18-06-2025	19-06-2025	20-06-2025	21-06-2025	22-06-2025
16	23-06-2025	24-06-2025	25-06-2025	26-06-2025	27-06-2025	28-06-2025	29-06-2025
17	30-06-2025	01-07-2025	02-07-2025	03-07-2025	04-07-2025	05-07-2025	06-07-2025
18	07-07-2025	08-07-2025	09-07-2025	10-07-2025	11-07-2025	12-07-2025	13-07-2025

Semana	Tema	Control	SMOJ	Tarea
1	UVA 1: Python - Programas Secuenciales	-	-	-
2	UVA 1: Python - Programas Secuenciales	C1	UVA1	
3	UVA 2: Condicionales	C2	UVA2	TA1
4	UVA 3: Ciclos			
5	UVA 3: Ciclos	C3	UVA3	
6	UVA 4: Strings	C4	UVA4	TA2
7	UVA 5: Funciones	C5	UVA5	
8	Preparación Certamen 1			
9	Certamen 1 : Jueves 08/05 17:30 hrs.			
10	UVA 6: Listas	C6	UVA6	
11	Vacaciones Estudiantes			
12	UVA 6: Listas			TA3
13	UVA 7: Diccionarios	C7	UVA7	
14	UVA 8: Procesamiento de Texto y Archivos	C8	UVA8	TA4
15	Preparación Certamen 2			
16	Certamen 2 : Jueves 26/06 17:30 hrs.			
17	No hay más clases ni ayudantías			
18	Certamen Recuperativo : martes 08/07 17:30 hrs.			

# Objetivos de Aprendizaje



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

## Contenido:

- Comprender el uso de ciclos en Python.
- Implementar ciclos controlados por contadores y condiciones generales.
- Reconocer y aplicar patrones comunes de ciclos.
- Desarrollar soluciones con ciclos anidados.



# ¿Qué es un Ciclo?



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

Un ciclo es una estructura de control que permite repetir un conjunto de instrucciones múltiples veces de manera automática.

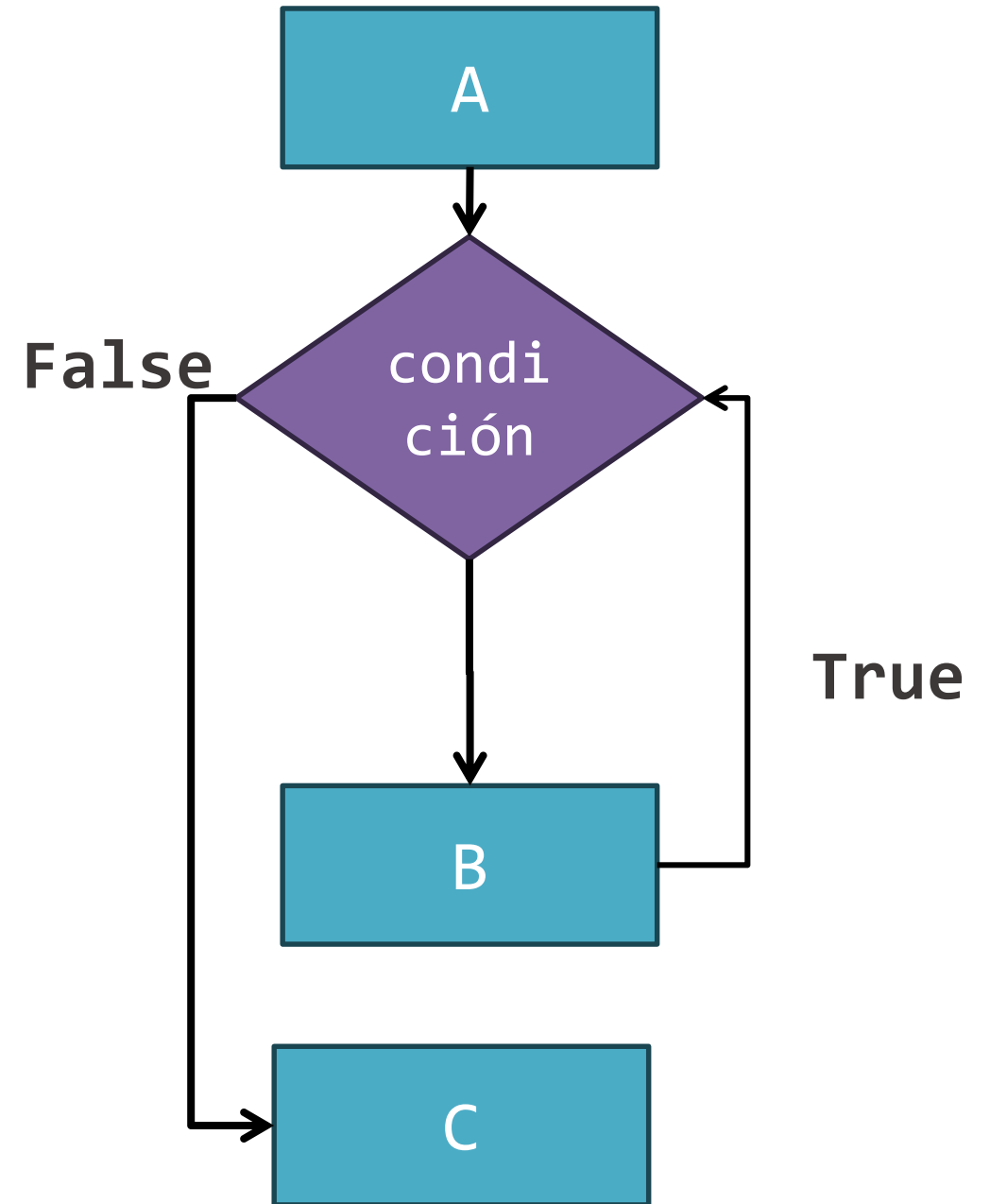
Los ciclos son fundamentales en la programación porque permiten ejecutar el mismo bloque de código varias veces, ya sea un número específico de veces o hasta que se cumpla una determinada condición.

```
1 class Repeat
2   def print_message
3     puts "I Will Not Repeat My Code"
4     puts "I Will Not Repeat My Code"
5     puts "I Will Not Repeat My Code"
6     puts "I Will Not Repeat My Code"
7     puts "I Will Not Repeat My Code"
8     puts "I Will Not Repeat My Code"
9     puts "I Will Not Repeat My Code"
10  end
11 end
```


## Ciclo while

El ciclo **while** es una estructura de control que permite repetir un bloque de código mientras se cumpla una condición lógica determinada.

En otras palabras, el ciclo **while** continúa ejecutando las instrucciones dentro de su cuerpo siempre que la condición evaluada al inicio sea **True**. Una vez que la condición se evalúa como **False**, el ciclo se detiene y el programa continúa con la siguiente instrucción después del ciclo.



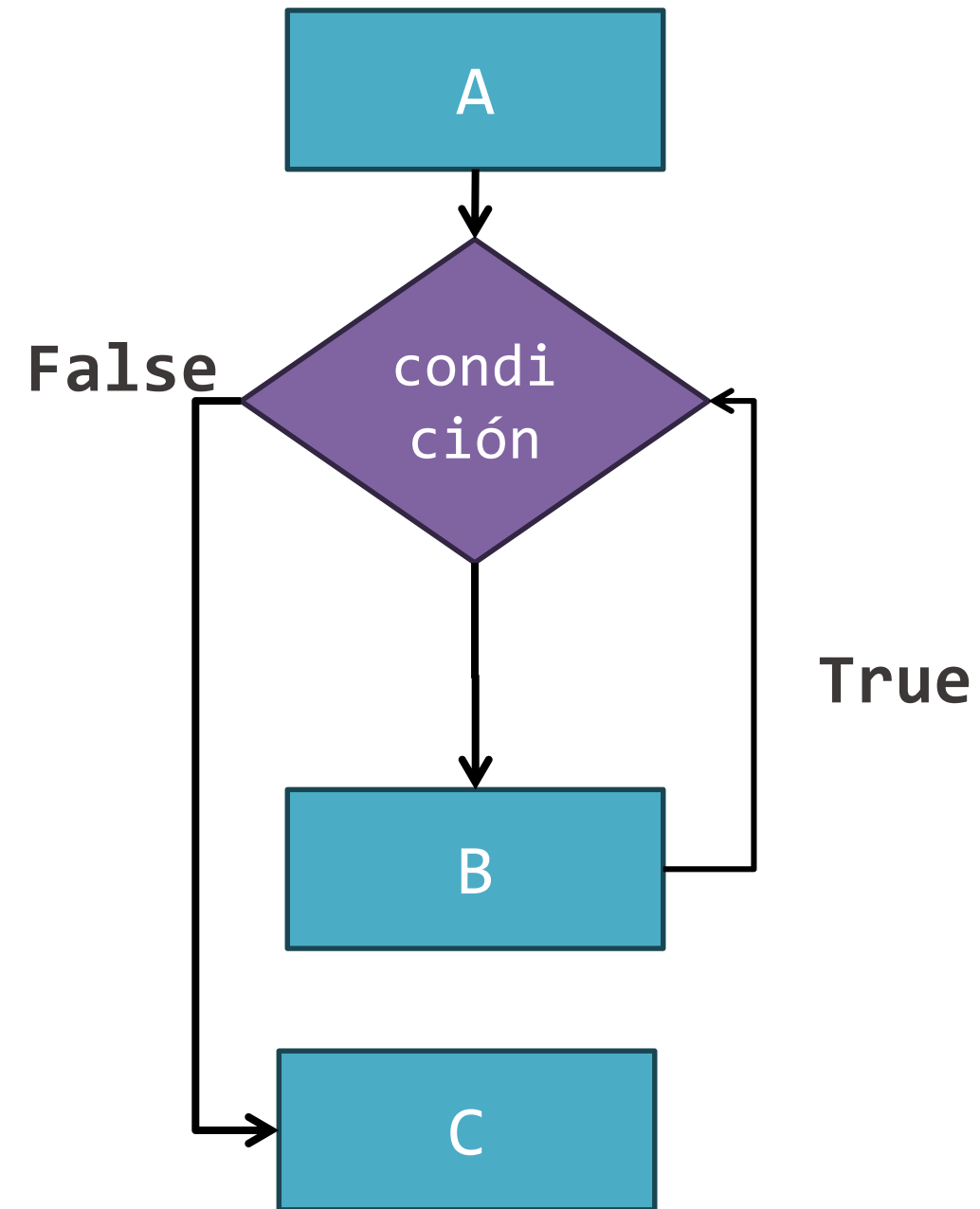
## Ciclo while - Sintaxis



```
1 while condición:  
2     # Bloque de código a repetir
```

## Ciclo while - Sintaxis

```
1 while condición:  
2     # Bloque de código a repetir
```





# Ciclos Controlados por Contador



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

Un ciclo controlado por contador es un tipo de ciclo que se ejecuta un número específico de veces, determinado por un contador.

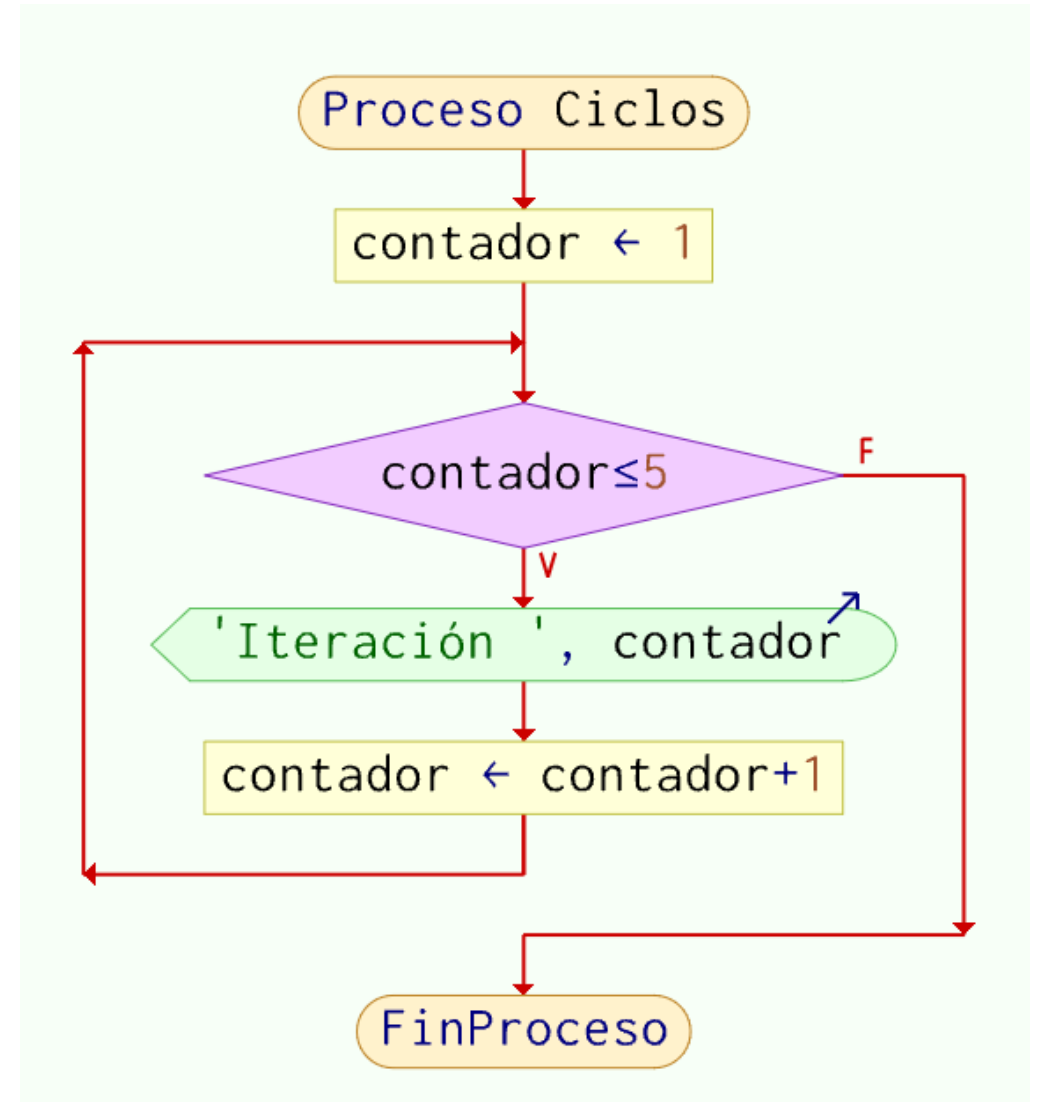
Este tipo de ciclo es ideal para situaciones en las que se conoce de antemano cuántas iteraciones son necesarias



# Ciclos Controlados por Contador



```
1 contador = 1
2 while contador <= 5:
3     print("Iteración ", contador)
4     contador = contador + 1
```



# Ciclos Basados en Condiciones Generales



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

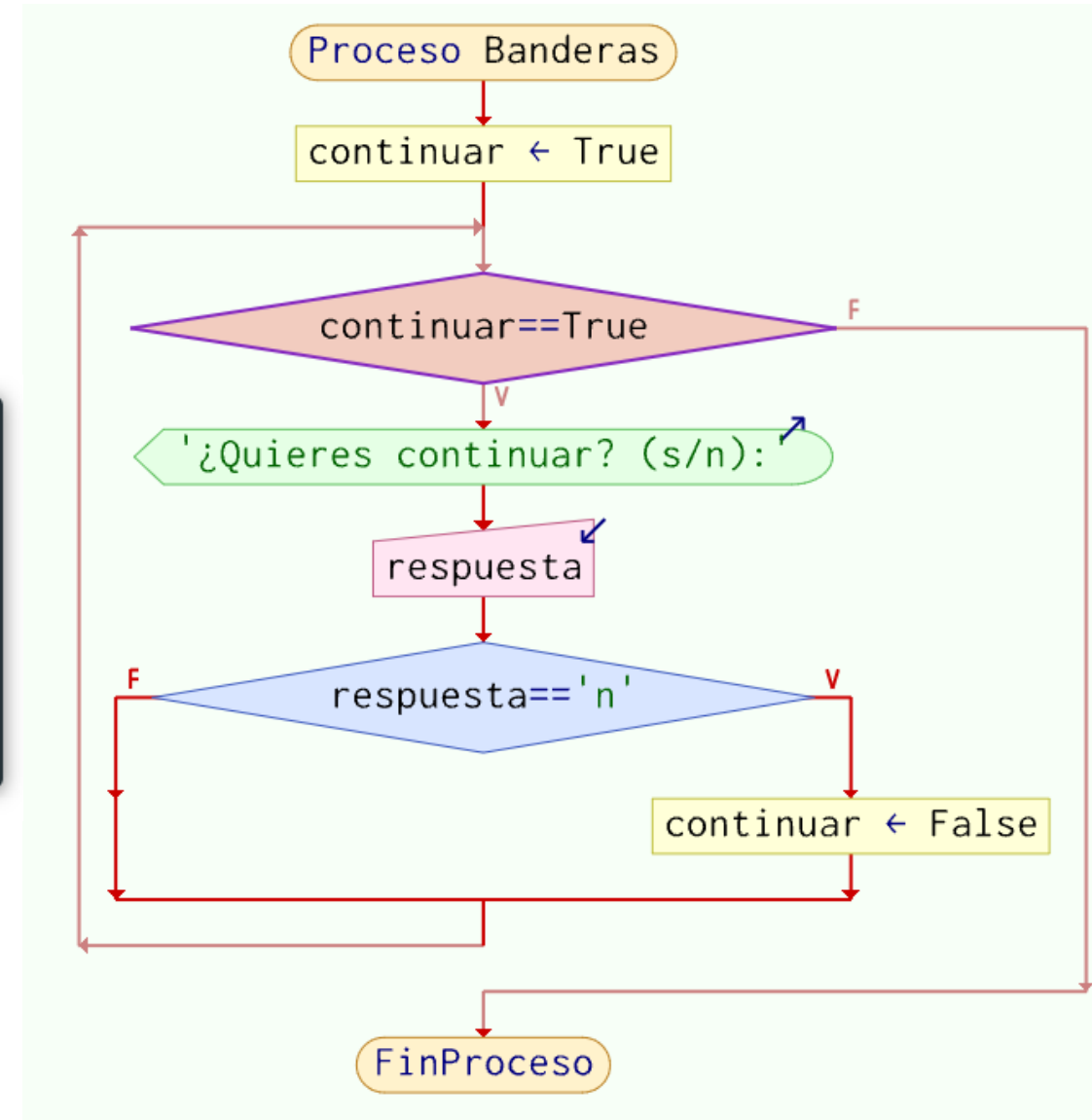
Un ciclo basado en condición general es aquel que continúa ejecutándose mientras se cumpla una condición específica. A diferencia de los ciclos controlados por contadores, que dependen de un número fijo de iteraciones, estos ciclos dependen de condiciones que pueden variar a lo largo de la ejecución del programa.

Las variables bandera son una técnica común utilizada para controlar la ejecución de un ciclo **while**. Estas variables actúan como indicadores que determinan cuándo debe continuar o detenerse el ciclo.



# Ciclos Basados en Condiciones Generales

```
1 continuar = True
2 while continuar == True:
3     respuesta = input("¿Quieres continuar? (s/n): ")
4     if respuesta == 'n':
5         continuar = False
```



# Ciclo For

Se utiliza para repetir un bloque de código un número específico de veces o para iterar sobre una secuencia (como una lista, una tupla o una cadena).

```
for i in range(5):  
    print(i)
```

## Ejemplo

```
for i in range(1, 6):  
    print(f"Iteración {i}")
```

# En resumen...

## 1. Ciclos while en Python:

Estructura básica y su uso para repetir tareas mientras se cumple una condición.

## 2. Ciclos Controlados por Contador:

Utilización de un contador para ejecutar un número fijo de iteraciones.

## 3. Ciclos Basados en Condiciones Generales:

Uso de variables bandera y condiciones dinámicas para controlar la repetición.

## Usos comunes

- **Búsqueda del Mayor/Menor:** Cómo encontrar el valor más alto o más bajo en una serie de entradas.
- **Acumulación y Conteo:** Métodos para sumar valores o contar elementos en un ciclo.
- **Ciclos Anidados:** Aplicaciones como la búsqueda de números primos en un rango o el cálculo de series matemáticas.

### Conclusión:

- Los ciclos son fundamentales en programación para automatizar tareas repetitivas y realizar cálculos complejos.
- La comprensión y aplicación de ciclos en Python te permite resolver una amplia variedad de problemas de manera eficiente.

### Próximos Pasos:

- Practica lo aprendido implementando ciclos en nuevos contextos.
- Explora cómo los ciclos y otras estructuras de control pueden complementar lo que has aprendido hoy.

