

## Tema: La sentencia while en Python

Objetivo: Comprender el funcionamiento de la sentencia while como estructura de control repetitiva en Python, para aplicarla en la solución de problemas que requieran ejecutar un conjunto de instrucciones de manera continua mientras se cumpla una condición lógica.

### Conceptos Teóricos

La **sentencia while** en programación es una **estructura de control** que permite **repetir un bloque de instrucciones** mientras se cumpla una **condición lógica** (es decir, mientras la condición sea *verdadera*).

👉 En otras palabras:

El ciclo while repite una acción **hasta que** la condición deje de cumplirse.

---

```
while condición:
    # Bloque de código a repetir
    # (debe modificarse algo para que la condición pueda volverse falsa)
```

### ♦ Sintaxis general:

#### ♦ Explicación paso a paso:

1. **Evalúa la condición** (una expresión que puede ser verdadera o falsa).
2. Si la **condición es verdadera (True)**, ejecuta el bloque de código dentro del ciclo.
3. Al terminar ese bloque, **vuelve a evaluar la condición**.
4. Si sigue siendo verdadera, **repite** el proceso.
5. Cuando la condición sea **falsa (False)**, el ciclo **se detiene** y el programa continúa con la siguiente instrucción.

#### ♦ Ejemplo básico en Python:

```
python

contador = 1

while contador <= 5:
    print("Número:", contador)
    contador = contador + 1
```

## Salida:

```
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
```

## ■ Explicación:

- La condición inicial es contador  $\leq 5$ .
- Mientras esa condición sea verdadera, se muestra el número y se aumenta el valor de contador.
- Cuando contador llega a 6, la condición ya no se cumple, y el ciclo termina.

---

## ⚠ Importante:

- Si olvidas modificar la variable que controla la condición (por ejemplo, no aumentas el contador), el ciclo **nunca termina**, y se produce un **bucle infinito**.

Ejemplo de error:

```
while True:
    print("Esto nunca termina...")
```

---

## ♦ Usos comunes:

- Repetir acciones hasta que el usuario decida salir.
- Leer datos hasta que se cumpla una condición.
- Simular procesos que dependen de un evento externo (por ejemplo, esperar una señal o un valor específico).

## Instrucciones para el Estudiante:

Desarrolle la **Parte 1** de la actividad en su **cuaderno de trabajo**, y la **Parte 2** en **Google Colab**, aplicando los conceptos aprendidos sobre la sentencia while. Cada parte tendrá un **valor del 50%** de la **nota total (nota 7)**, por lo que se evaluará tanto la comprensión teórica como la correcta aplicación práctica del tema.

### PARTE 1: Cuaderno

#### Instrucción:

Copie en su **cuaderno de trabajo** el **tema** y el **objetivo** que se presentan a continuación. Luego, escriba **cada pregunta junto con su respuesta correcta**, cuidando la presentación y la ortografía.

## Preguntas sobre la sentencia while:

1. ¿Qué función cumple la sentencia while en un programa en Python?
  - a) Repetir un bloque de instrucciones mientras se cumpla una condición
  - b) Ejecutar una instrucción solo una vez
  - c) Comparar dos valores numéricos
  - d) Detener la ejecución del programa
2. ¿Qué sucede cuando la condición del while nunca se vuelve falsa?
  - a) El programa se detiene automáticamente
  - b) Se produce un **bucle infinito**
  - c) No se ejecuta el ciclo
  - d) El ciclo se ejecuta solo una vez
3. En la estructura de un ciclo while, ¿qué parte se debe modificar dentro del bloque para evitar repeticiones infinitas?
  - a) El nombre del ciclo
  - b) La condición inicial
  - c) La variable de control
  - d) El tipo de dato
4. ¿Cuál de las siguientes afirmaciones sobre el ciclo while es verdadera?
  - a) Solo puede usarse con números enteros
  - b) Siempre necesita una estructura else para funcionar
  - c) Evalúa la condición antes de ejecutar el bloque de instrucciones
  - d) Se usa únicamente para recorrer listas
5. ¿Qué palabra clave se utiliza para iniciar un ciclo repetitivo en Python?
  - a) loop
  - b) while
  - c) repeat
  - d) for

## PARTE 2: Colab

### Instrucción:

Transcriba cada ejemplo de código en Python en su cuaderno digital de trabajo utilizando Google Colab, ejecútelo para comprobar su funcionamiento y observe los resultados. Luego, comparta el vínculo del archivo en Classroom en el espacio habilitado para tal fin, asegurándose de que el enlace esté configurado con permiso de visualización para el docente.

#### Ejemplo 1: Contar del 1 al 5

```
python

contador = 1

while contador <= 5:
    print("Número:", contador)
    contador = contador + 1
```

#### Explicación:

El ciclo imprime los números del 1 al 5.

Cada vez que se repite, la variable contador aumenta en 1.

#### Ejemplo 2: Solicitar una contraseña

```
python

contraseña = ""

while contraseña != "python123":
    contraseña = input("Ingrese la contraseña: ")

print("¡Acceso concedido!")
```

### ■ Explicación:

El ciclo se repite hasta que el usuario escribe la contraseña correcta.  
Cuando la condición deja de cumplirse, el programa muestra el mensaje final.

#### ✿ Ejemplo 3: Sumar números hasta que el usuario escriba 0

```
python

suma = 0
numero = int(input("Ingrese un número (0 para terminar): "))

while numero != 0:
    suma = suma + numero
    numero = int(input("Ingrese otro número (0 para terminar): "))

print("La suma total es:", suma)
```

### ■ Explicación:

El programa va sumando los números que el usuario ingresa hasta que digita 0, momento en el que termina el ciclo.

#### ✿ Ejemplo 4: Contador regresivo

```
python

contador = 5

while contador > 0:
    print("Cuenta regresiva:", contador)
    contador = contador - 1

print("¡Despegue!")
```

### ■ Explicación:

El ciclo comienza en 5 y va restando 1 en cada repetición hasta llegar a 0, simulando una cuenta regresiva.