

Control de Flujo y Lógica Integrada

Instrucciones: Deberás escribir el código en tu editor, probar diferentes escenarios y asegurarte de que el programa no colapse. Lee detenidamente los requerimientos de cada sistema.

Reto 1: El Invernadero Inteligente

Contexto: Has sido contratado para programar el cerebro de un invernadero automatizado. El sistema debe evaluar las condiciones climáticas durante una semana (7 días) y activar los mecanismos físicos adecuados.

Requerimientos del sistema:

1. Crea un ciclo que se repita exactamente 7 veces (simulando los 7 días).
2. En cada día, el sistema debe pedirle al usuario que ingrese la **Temperatura** y la **Humedad**.
3. Evalúa las condiciones usando if / elif / else:
 - Si la temperatura es **mayor a 30** Y la humedad es **menor a 30**: Imprime "Alerta: Activando riego intensivo".
 - Si la temperatura es **mayor a 30** Y la humedad es **mayor o igual a 30**: Imprime "Alerta: Activando ventiladores".
 - Si la temperatura es **menor a 15**: Imprime "Alerta: Activando calentadores".
 - En cualquier otro caso: Imprime "Condiciones óptimas. Sistemas en reposo."
4. El sistema debe llevar una cuenta de cuántas alertas se activaron en la semana. Al finalizar los 7 días, debe imprimir el total de alertas.

Código Base:

```
total_alertas = 0
dias_semana = 7
```

```
print("--- Iniciando Sistema del Invernadero ---")
```

```
# Construye la lógica del programa aquí.
```

```
1  total_alertas = 0
2  dias_semana = 7
3
4  print("--- Iniciando Sistema del Invernadero ---")
5  # Construye la lógica del programa aquí.
6  |
```

Reto 2: Analizador de Tráfico de Red

Contexto: Tienes un registro del "ping" (tiempo de respuesta en milisegundos) de varios paquetes de datos que pasaron por tu servidor. Debes analizar la lista para detectar anomalías en la red.

Requerimientos del sistema:

1. Tienes la siguiente lista de pings: [25, 45, 800, 30, 1200, -5, 50, 95]
2. Recorre la lista con un ciclo for.
3. Aplica las siguientes reglas usando alteradores de flujo:
 - Si el ping es **menor a 0**: Esto es un error crítico del hardware. Imprime "ERROR FATAL: Ping negativo detectado. Deteniendo análisis." y **rompe** el ciclo por completo.
 - Si el ping es **mayor a 500**: Es un paquete perdido. Imprime "Advertencia: Paquete lento detectado (X ms)" y suma 1 a un contador de paquetes perdidos.
 - Si el ping es **menor a 100**: Es un paquete perfecto. Usa la instrucción que **salta** a la siguiente iteración sin hacer nada más.
4. Al final (si el ciclo no se rompió por un error fatal), imprime cuántos paquetes lentos se detectaron en total.

Código Base (Plantilla):

```
pings_red = [25, 45, 800, 30, 1200, -5, 50, 95]
```

```
paquetes_lentos = 0
```

```
print("--- Analizando Tráfico de Red ---")
```

```
# Escribe tu ciclo y condiciones aquí
```

```
print("Análisis finalizado. Total de paquetes lentos:", paquetes_lentos)
```

```
1  pings_red = [25, 45, 800, 30, 1200, -5, 50, 95]
2  paquetes_lentos = 0
3
4  print("--- Analizando Tráfico de Red ---")
5
6  # Escribe tu ciclo y condiciones aquí
7
8
9  print("Análisis finalizado. Total de paquetes lentos:", paquetes_lentos)
```

Reto 3: El Cajero Automático Seguro

Contexto: Vas a programar el software de un cajero automático desde cero. Este ejercicio requerirá ciclos anidados (un ciclo dentro de otro) o múltiples ciclos separados.

Fase 1: Seguridad (Login)

- El PIN correcto es 1234.
- El usuario tiene un máximo de **3 intentos** para ingresarlo usando un ciclo while.
- Si lo ingresa mal 3 veces, el programa debe imprimir "Tarjeta bloqueada" y terminar.
- Si lo ingresa bien, debe imprimir "Bienvenido" y pasar a la Fase 2 (puedes usar un break para salir del ciclo de intentos).

Fase 2: El Menú de Operaciones

- El saldo inicial es de \$5000.
- Crea un ciclo while True que muestre un menú: "1. Consultar, 2. Retirar, 3. Salir".
- Pide al usuario que elija una opción.
- Si elige 1: Imprime el saldo actual.
- Si elige 2: Pide una cantidad a retirar. Si la cantidad es mayor al saldo, imprime "Fondos insuficientes". Si no, resta la cantidad del saldo e imprime el nuevo saldo.
- Si elige 3: Imprime "Gracias por usar el banco" y rompe el ciclo para terminar el programa.

Código Base:

```
pin_correcto = "1234"
```

```
intentos = 0
```

```
saldo = 5000
```

```
print("--- Bienvenido al Banco Nacional ---")
```

```
# Construye la lógica del programa aquí.
```

```
1  pin_correcto = "1234"
2  intentos = 0
3  saldo = 5000
4
5  print("--- Bienvenido al Banco Nacional ---")
6
7  # Construye la lógica del programa aquí.
```