Estructuras Condicionales

Semestre 02, 2025

Introducción

Los condicionales permiten que un programa tome decisiones según ciertas condiciones.

El condicional `if` evalúa si una expresión es verdadera. Si lo es, se ejecuta el bloque de código asociado.

El uso de condicionales permite que nuestros programas sean dinámicos y respondan de forma distinta según los datos. `if` y `else` son herramientas esenciales para la lógica en programación.

Expresiones booleanas

Una **expresión booleana** es una expresión que se evalúa como `True` o `False`.

Estas expresiones son fundamentales para tomar decisiones usando `if`.

Ejemplos:

```
1
2   x = 5
3
4
5
6   print(x > 3) # True
7
8   print(x == 10) # False
9
10  print((x % 2) == 1) # True (x es impar)
11
```

También podemos combinarlas con operadores lógicos:

```

x = 7

y = 10



print((x > 5) and (y < 20)) # True

print(not(x == y)) # True

```

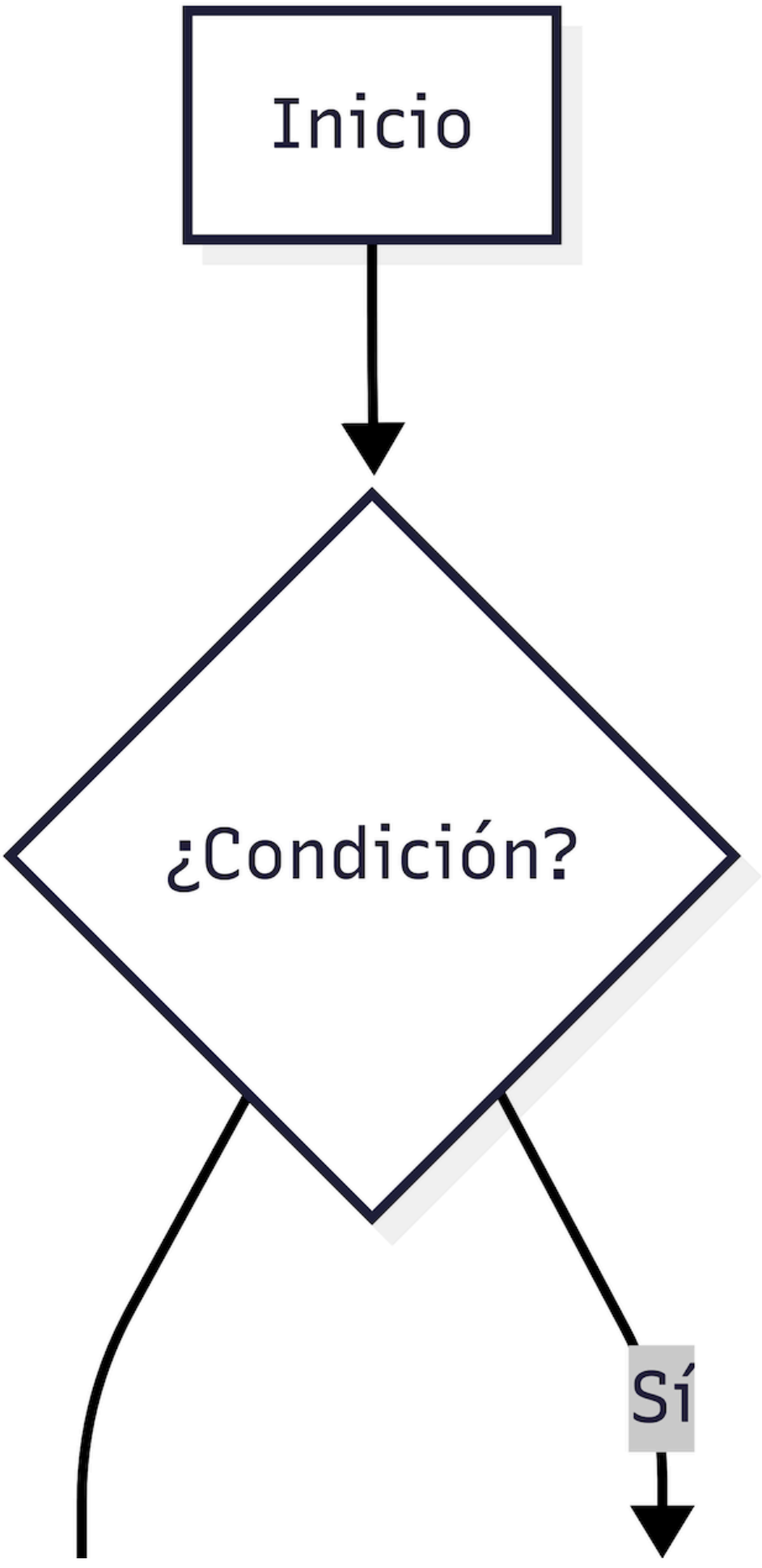
Sintaxis

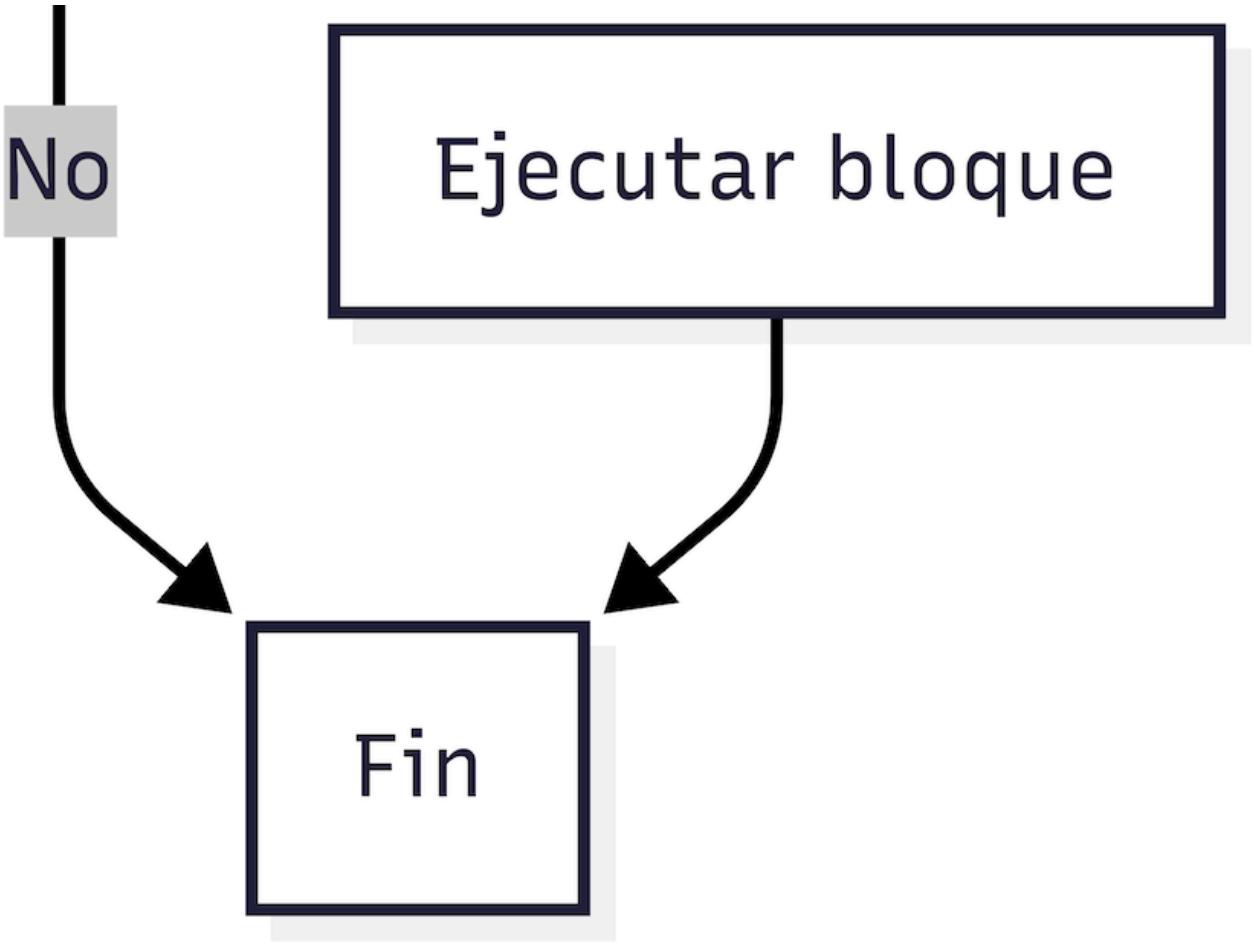
```

if condicion:

# bloque de código

```

Ejemplo

```

edad = 20



if edad >= 18:

print("Eres mayor de edad")

```

Diagráma de Flujo


Inicio
¿Condición?
Sí



Sangría (indentación)

La indentación es obligatoria en Python. El bloque de código dentro del `if` debe estar tabificado.

```

  if True:

  print("Esto se imprime")

```

Alternativas

No siempre solo tengo una opción, muchas veces tengo al menos 2.

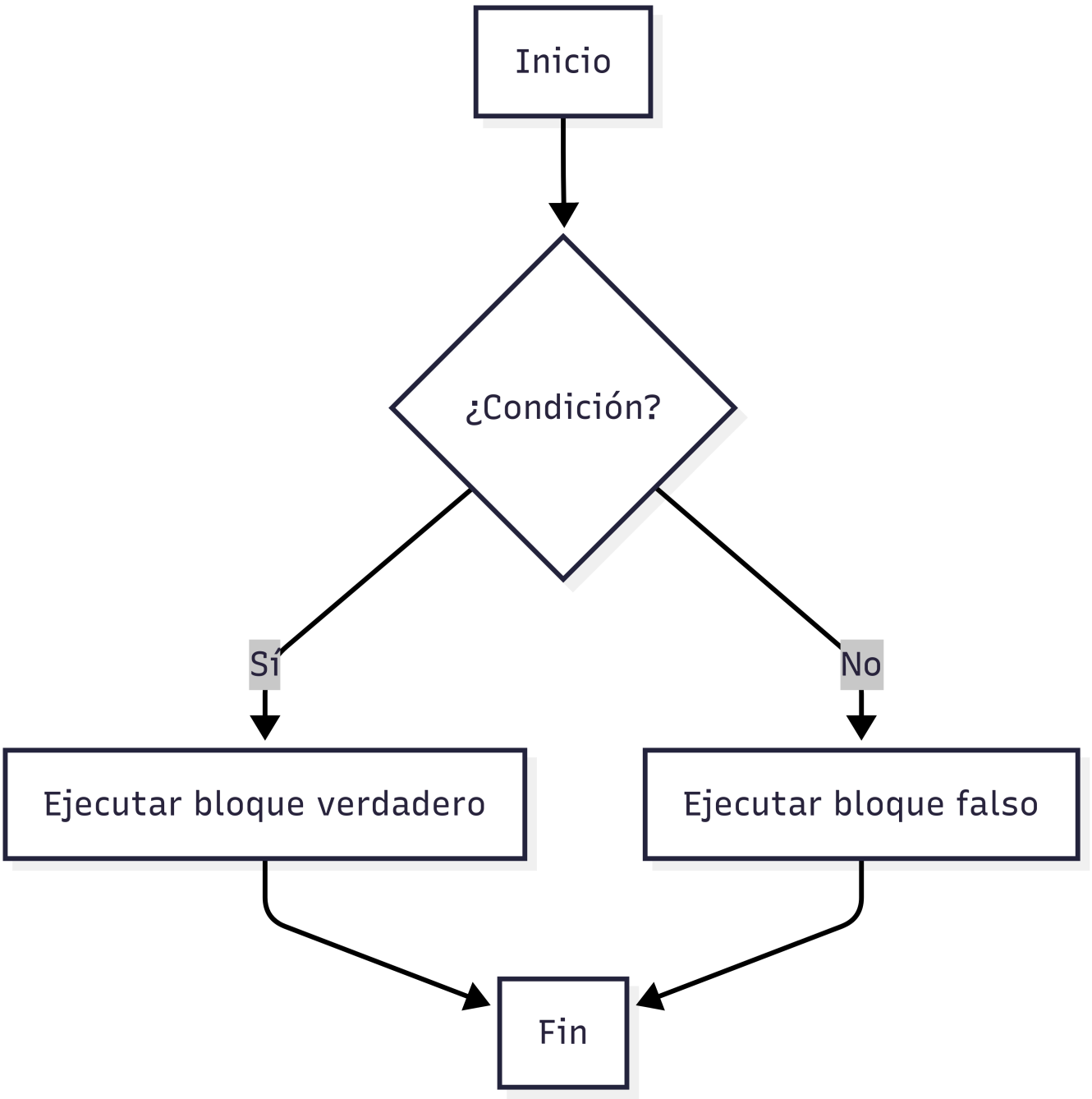
```

  if condicion:

```

```
# bloque si se cumple

else:

# bloque si no se cumple

```

Ejemplo

```

numero = 7


if numero % 2 == 0:

print("Es par")

else:

print("Es impar")

```

Diagráma de Flujo



Anidación de condicionales

Es posible colocar un `if` dentro de otro.

```
1
2   numero = 0
3
4
5
6   if numero >= 0:
```

```
if numero == 0:
print("Es cero")
else:
print("Es positivo")
else:
print("Es negativo")
```

Buenas prácticas

- Comentar la lógica si es compleja.
- Evitar condicionales anidados innecesarios.
- Usar paréntesis para agrupar condiciones complejas.

Errores comunes

1. **Falta de indentación**:

```
if True:
print("Esto falla") # Error de indentación
```

2. Usar = en lugar de == para comparación:

```
if x = 5: # Error de sintaxis
```

3. **Olvidar los dos puntos**:

```
```

```
if x > 3 # SyntaxError: expected ':'

print("OK")

```

4. **No convertir tipos antes de comparar**:

```

edad = input("Edad: ")

if edad >= 18: # Error: compara str con int

```

Ejemplos

Ejemplo 1 – Verificar número positivo

```

numero = int(input("Ingrese un número: "))



if numero > 0:

print("El número es positivo")

else:

print("El número no es positivo")

```

Ejemplo 2 – Verificar si un número es par

```

numero = int(input("Ingrese un número: "))


```

```
5
6   if numero % 2 == 0:
7
8   print("Es una vocal")
9
10  else:
11
12  print("No es una vocal")
13
```

Ejemplo 3 – Verificar rango de edad

```
1
2   edad = int(input("Ingrese su edad: "))
3
4
5
6   if edad >= 18 and edad <= 25:
7
8   print("Estás en el rango joven-adulto")
9
10  else:
11
12  print("Fuera del rango")
13
```