

1. Estructuras de Control

¿Qué son las Estructuras de Control?

Las estructuras de control son bloques fundamentales en programación que permiten controlar el flujo de ejecución de un programa, determinando qué instrucciones se ejecutan y cuándo. Son esenciales para implementar lógica condicional, repeticiones y decisiones.

1.1. Estructuras de Control Secuenciales

Son las más básicas y ejecutan instrucciones una tras otra, en el orden en que aparecen en el código.

Ejemplo en Python:

```
print('Inicio')  
x = 5 + 3  
print('Resultado:', x)
```

1.2. Estructuras de Control Condicionales (Decisión)

Permiten ejecutar diferentes bloques de código dependiendo del cumplimiento de ciertas condiciones.

if - else:

```
if x > 0:  
    print('Positivo')  
else:  
    print('No positivo')
```

if - elif - else:

```
if x == 0:  
    print('Cero')  
elif x > 0:  
    print('Positivo')  
else:  
    print('Negativo')
```

1.3. Estructuras de Control Repetitivas (Bucles)

Permiten repetir un bloque de instrucciones múltiples veces.

Bucle while:

```
i = 0
while i < 5:
    print(i)
    i += 1
```

Bucle for:

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

1.4. Estructuras de Control de Selección Múltiple

Permiten tomar decisiones entre múltiples opciones. En Python, se utiliza 'if - elif - else'.

En otros lenguajes como C, Java o JavaScript se usa 'switch'.

Ejemplo en Java:

```
switch(dia) {
    case 1: System.out.println('Lunes'); break;
    case 2: System.out.println('Martes'); break;
    default: System.out.println('Otro día');
}
```

1.5. Palabras Clave de Control Adicionales

- break → Finaliza un bucle prematuramente.
- continue → Salta la iteración actual y continúa con la siguiente.
- pass → No hace nada, se usa como marcador de lugar en Python.

1.6. Importancia de las Estructuras de Control

Permiten diseñar algoritmos con decisiones, repeticiones y flujos complejos.

Hacen que los programas puedan adaptarse a diferentes entradas y condiciones.

1.7. Buenas Prácticas

- Usar nombres descriptivos para las variables.
- Evitar bucles infinitos.
- Mantener indentación clara.
- No abusar de estructuras anidadas (if dentro de if, etc.).