

Uso de estructuras repetitivas

for, while, do while

for, while, do while

Las estructuras repetitivas en C permiten ejecutar un bloque de código múltiples veces mientras se cumpla una condición. Existen tres tipos principales de ciclos en C:

- for: Ideal para iteraciones con un número conocido de repeticiones.
- while: Se ejecuta mientras se cumpla una condición, sin conocer el número exacto de iteraciones.
- do-while: Similar a while, pero garantiza ejecutar el código al menos una vez.

Estructura de un ciclo for

```
for (inicialización; condición; incremento/decremento)
```

```
{
```

```
    // Bloque de código a ejecutar
```

```
}
```

- **Inicialización**: Se ejecuta una sola vez antes de iniciar el ciclo.
- **Condición**: Se evalúa antes de cada iteración. Si es true, se ejecuta el bloque de código.
- **Incremento/Decremento**: Se ejecuta al final de cada iteración.

Errores comunes

- Olvidar la inicialización o incremento puede provocar bucles infinitos.
- Usar `<=` en lugar de `<` cuando el índice empieza en `0`, lo que podría causar una iteración extra.

Ventajas/Desventajas

- Fácil de leer y estructurar cuando el número de repeticiones es conocido.
- Menos flexible que `while` para condiciones no predeterminadas.

Ejemplo de uso recomendado

- Recorrer arreglos o listas con un número conocido de elementos.
- Generar secuencias numéricas.

Estructura de uso de while

```
while (condición) {
```

```
    // Bloque de código a ejecutar mientras la condición sea verdadera
```

```
}
```

Errores comunes

- No actualizar la variable de control dentro del ciclo, causando un bucle infinito.
- Condiciones mal formuladas que nunca se cumplen.

Ventajas/Desventajas

- Adecuado cuando no se conoce el número exacto de iteraciones.
- Puede ser más propenso a errores lógicos si no se maneja bien la condición.

Ejemplo de uso recomendado

- Leer datos hasta que se cumpla una condición (ejemplo: leer números hasta que el usuario ingrese 0).
- Esperar la entrada del usuario.

Estructura de uso de do while

```
do {
```

```
    // Bloque de código a ejecutar
```

```
} while (condición);
```

Errores comunes

Puede ejecutar el bloque innecesariamente si la condición nunca se cumple.

Ventajas/Desventajas













Garantiza que el bloque de código se ejecuta al menos una vez.

No siempre es la mejor opción cuando se quiere validar primero una condición.

Ejemplo de uso recomendado

- Menús interactivos donde se debe mostrar el menú al menos una vez.
- Validación de entrada del usuario.

Comparativa

Característica	for	while	do while
Se usa cuando conocemos el número de iteraciones			
Se usa cuando las iteraciones dependen de una condición externa			
Garantiza ejecutar al menos una vez			
Mejor para recorrer arreglos o estructuras con un número fijo de elementos			

Referencias

IEEE. (2020). *IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®) Base Specifications, Issue 7*. IEEE.

ISO/IEC. (2018). *ISO/IEC 9899:2018 – Programming languages – C*. International Organization for Standardization.