

CICLOS 2 JAVA

El Ciclo while

El ciclo while (mientras) es un bucle de "pre-prueba". Esto significa que **primero evalúa una condición** y, solo si es verdadera, ejecuta el código en su interior. Repetirá este proceso hasta que la condición sea falsa.

Sintaxis:

```
while (condicion) {  
    // Código a ejecutar mientras la condición sea verdadera  
}
```

Característica principal: Si la condición es falsa desde el principio, el código dentro del ciclo **nunca se ejecutará**.

Ejemplo 1: while con Acumulador

Un **acumulador** es una variable que "acumula" un valor en cada iteración, como en una suma o un producto.

Objetivo: Sumar los números del 1 al 5.

```
public class EjemploWhile {  
    public static void main(String[] args) {  
        int suma = 0;        // Acumulador, inicia en 0  
        int contador = 1;    // Variable de control  
  
        while (contador <= 5) {  
            suma = suma + contador; // Se acumula el valor del  
contador  
            System.out.println("Suma parcial: " + suma);  
            contador++; // ¡Importante! Incrementar el contador  
para evitar un bucle infinito  
        }  
  
        System.out.println("La suma total es: " + suma);  
    }  
}
```

Resultado del código:

```
Suma parcial: 1
Suma parcial: 3
Suma parcial: 6
Suma parcial: 10
Suma parcial: 15
La suma total es: 15
```

El Ciclo do-while

El ciclo do-while (hacer... mientras) es un bucle de "post-prueba". Esto significa que **primero ejecuta el bloque de código una vez** y luego evalúa la condición. Si la condición es verdadera, el ciclo se repite.

Sintaxis:

```
do {
    // Código a ejecutar
} while (condicion);
```

Característica principal: El código dentro del ciclo se ejecutará **al menos una vez**, sin importar la condición. Es ideal para menús o validación de datos.

Ejemplo 2: do-while con Bandera

Una **bandera** (o *flag*) es una variable, usualmente booleana, que controla el flujo de un programa. Actúa como un interruptor: el ciclo continúa mientras la bandera esté en un estado (ej. false) y se detiene cuando cambia al otro (true).

Objetivo: Pedir una contraseña hasta que el usuario ingrese la correcta ("1234").

```
import java.util.Scanner;

public class EjemploDoWhile {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String contrasenaCorrecta = "1234";
        String contrasenaIngresada;
```

```

    boolean accesoPermitido = false; // Bandera, inicia en
falso

    do {
        System.out.print("Ingresa la contraseña: ");
        contrasenaIngresada = teclado.nextLine();

        if (contrasenaIngresada.equals(contrasenaCorrecta)) {
            accesoPermitido = true; // Cambiamos el estado de
la bandera
            System.out.println("¡Acceso concedido!");
        } else {
            System.out.println("Contraseña incorrecta.
Inténtalo de nuevo.");
        }

    } while (!accesoPermitido); // El ciclo continúa mientras
la bandera sea falsa

    teclado.close();
}
}

```

Resultado del código: El programa pedirá la contraseña repetidamente hasta que escribas "1234".

| Característica | while | do-while |
|------------------|-------------------------------------|--------------------------------------|
| Verificación | Pre-prueba (evalúa y luego ejecuta) | Post-prueba (ejecuta y luego evalúa) |
| Ejecución Mínima | 0 veces (si la condición es falsa) | 1 vez (siempre se ejecuta al menos) |

| | | |
|---------------------|---------------------------------------|---|
| Punto y Coma | No lleva punto y coma al final | Sí lleva punto y coma después de <code>while(condicion);</code> |
| Uso Común | Cuando no sabes cuántas veces iterar. | Menús, validación de entrada de datos. |

Ejercicios para Practicar

Nivel Básico

1. **Contador while:** Imprime los números del 1 al 10.
2. **Tabla de multiplicar con while:** Pide un número al usuario y muestra su tabla de multiplicar del 1 al 10.
3. **Cuenta regresiva con while:** Pide un número y muestra una cuenta regresiva desde ese número hasta 1.
4. **Entrada con do-while:** Pide al usuario un número. Si introduce un número negativo o cero, vuelve a pedirselo.

Nivel Intermedio (Acumuladores y Banderas)

6. **Suma de pares con while:** Calcula la suma de todos los números pares entre 1 y 100.
7. **Factorial con while:** Pide un número y calcula su factorial (ej. $5! = 5 * 4 * 3 * 2 * 1$).
8. **Contador de intentos con while:** Simula un login. El usuario tiene 3 intentos para ingresar un usuario y contraseña correctos. Usa una bandera para saber si el acceso fue exitoso y un contador para los intentos.
9. **Cajero automático con do-while:** Simula un cajero que pide un monto a retirar. Si el monto es mayor a un saldo inicial (ej. 1000), muestra un mensaje de error y vuelve a pedirlo. Si es válido, resta el monto al saldo y pregunta si desea realizar otra operación.

Nivel Avanzado

11. **Promedio de notas con do-while:** Pide al usuario que ingrese notas. El programa se detiene cuando el usuario ingresa un -1. Al final, muestra el promedio de las notas ingresadas (sin contar el -1).

12. **Validación de correo con do-while:** Pide al usuario un correo electrónico hasta que ingrese uno que contenga el símbolo @ y un .. Usa una bandera para validar.
13. **Contador de vocales con while:** Pide una frase y cuenta cuántas vocales (a, e, i, o, u) tiene, sin importar si son mayúsculas o minúsculas.