

## Ciclos y sentencias repetitivas

<b>Ciclos y sentencias repetitivas</b>	<b>1</b>
¿Qué aprenderás?	2
Introducción	2
Control de flujo y condiciones	3
Bucles y representación en diagramas de flujos	5
Bucle FOR	5
Bucle While	6
Bucle DO-While	7



**¡Comencemos!**

## ¿Qué aprenderás?

- Comprender la lógica de las sentencias repetitivas

## Introducción

Las sentencias repetitivas nos permiten, en resumen, ejecutar una cantidad de líneas de códigos tantas veces hasta que se cumpla una condición. A esto le llamamos **bucles** en la jerga de la programación de software. A continuación, introduciremos la lógica de estas sentencias repetitivas mediante algoritmos de diagrama de flujo y ejemplificando mediante códigos sencillos en java.

**¡Vamos con todo!**



## Control de flujo y condiciones

En un programa computacional la mayoría de las veces, por no decir siempre, es necesario controlar el flujo del programa y evaluar condiciones, las cuales si se cumplen nos permitirán realizar ciertas acciones o procesos, tales como cálculos u operaciones varias. Por ejemplo, necesitamos crear un programa que evalúe identificar si un número es par. El programa debiese terminar si o solo si el usuario ingresa un número par, si no debiese volver a preguntar. La solución, diagrama de flujo y algoritmo en Java sería el siguiente:

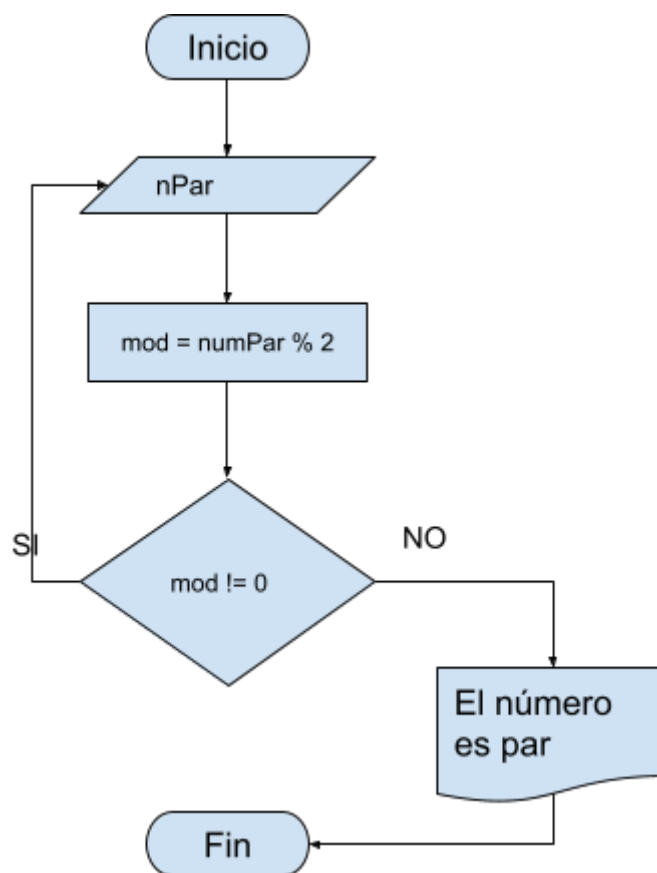


Imagen 1. Diagrama de flujos - Solución para ingresar un número par.  
Fuente: Desafío Latam.

El diagrama de la imagen representa la solución para ingresar un número par. Nótese que en el caso que la condición se cumpla, el flujo del programa se devuelve al inicio, y este solo terminará si la condición no se cumple. Este bucle o sentencia repetitiva se llama DO-WHILE..

Por último, la solución en lenguaje Java del flujo anterior, sería como se muestra a continuación:

```
package cl.desafiolatam;
import java.util.Scanner;
public class MiPrimerPrograma {
    public static void main(String[] args) {
        //Se declaran las variables de entrada de tipo double
        double nPar = 0;
        double mod = 0;
        /*Ciclo DO-WHILE, bucle que no terminará hasta que el usuario
ingrese un número par */
        do{
            System.out.printf("Ingrese un número: ");
            Scanner sc = new Scanner(System.in);
            nPar = sc.nextLong();
            mod = nPar % 2;
        }while(mod != 0);

        System.out.println("El número es par");
    }
}
```

## Bucles y representación en diagramas de flujos

Como se mencionaba antes, un bucle es una sentencia que permite repetir una serie de instrucciones hasta que se cumpla una condición. En este ámbito, tenemos tres tipos de sentencias que vamos a estudiar:

- Bucle For
- Bucle While
- Bucle Do While

### Bucle FOR

El bucle FOR es aquel que nos permitirá repetir las sentencias desde y hasta un número determinado. En este caso, este bucle siempre se apoya de un contador.

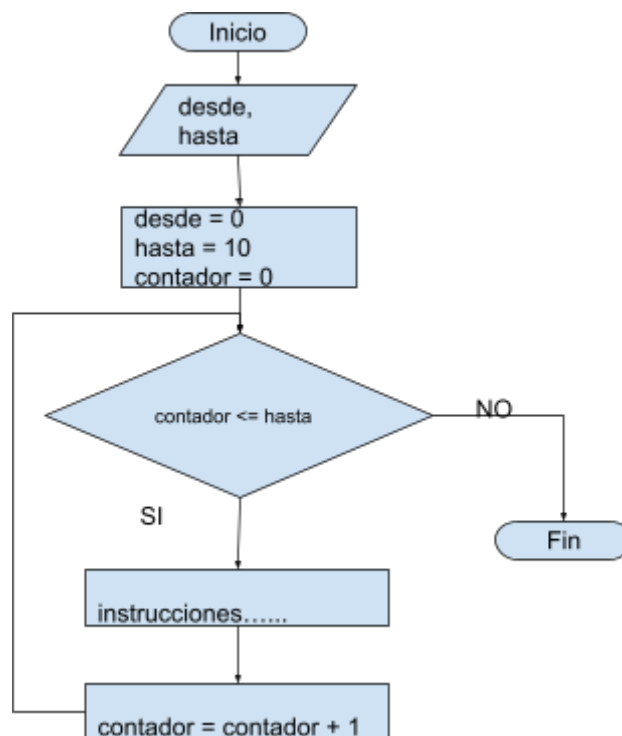


Imagen 2. Diagrama de flujo bucle FOR.

Fuente: Desafío Latam.

En el flujo anterior, se declara una variable que indica desde dónde vamos a contar, otra que indica hasta donde o cuantas veces se repetirán las instrucciones y por último, un contador que cuenta cada una de las repeticiones. Para efecto del flujo, este repetirá desde 0 hasta 10 veces las instrucciones.

## Bucle While

El bucle While es otra sentencia repetitiva que ejecuta una serie de instrucciones mientras se cumple una condición. En este caso, puede que este Bucle nunca ejecute las instrucciones ya que la condición puede que nunca se cumpla.

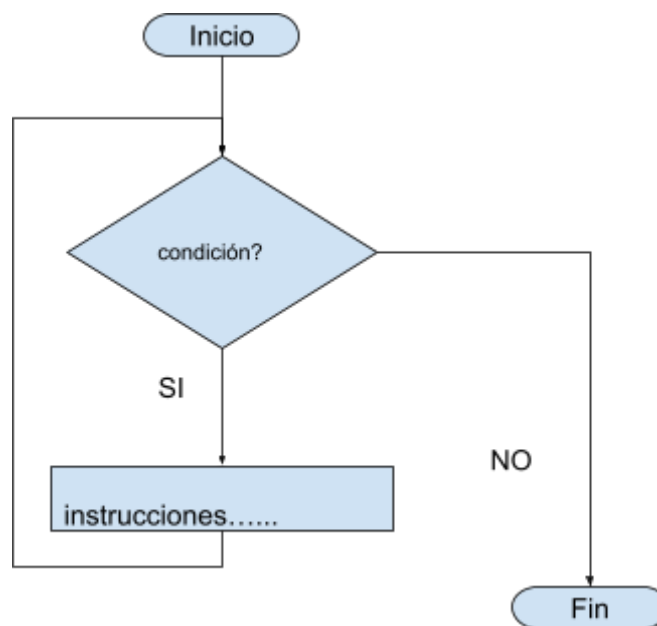


Imagen 3. Diagrama de flujo bucle While.  
Fuente: Desafío Latam.

## Bucle DO-While

El Bucle DO-While es muy similar al Bucle While, no obstante este siempre va a ejecutar las instrucciones al menos una vez antes de salir del bucle.

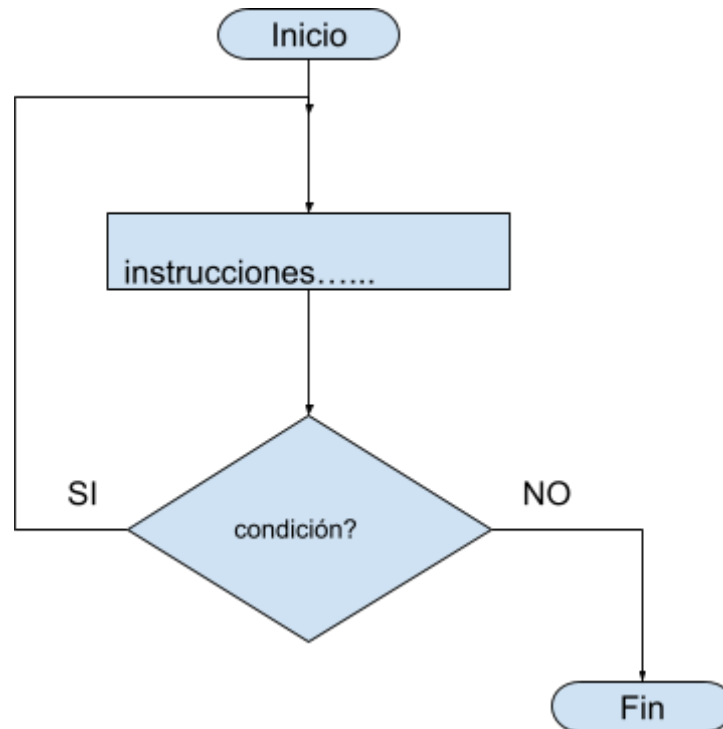


Imagen 4. Diagrama de flujos Bucle DO-WHlle.  
Fuente: Desafío Latam.