



Apache  
**NetBeans**

# #19 E. REPETITIVAS

## CICLO FOR



Java™

```
default.rb
# Available result of running an experiment.
Scientist::Result

# An array of candidate Observations.
attr_reader :candidates

# The control Observation to which the rest are compared.
attr_reader :control

# An experiment.
attr_reader :experiment

# An array of observations which didn't match the control.
attr_reader :ignored

# An array of observations which didn't match the control.
attr_reader :mismatched

# An array of Observations in execution order.
attr_reader :observations

# Internal: Create a new result.
#
# experiment - the Experiment this result is for
# observations - an Array of Observations, in execution order
# control - the control Observation
#
def initialize(experiment, observations = [], control = nil)
  @experiment = experiment
  @observations = observations
  @control = control
  @candidates = observations - [control]
  evaluate_candidates

  freeze
end

# Public: the experiment's context
def context
  experiment.context
end

# Public: the name of the experiment
def experiment_name
```

**PROGRAMA FÁCIL CON JAVA**



## 3.3 Estructuras iterativas

### Estructura repetitiva: repetir (FOR)

Es la estructura algorítmica adecuada para utilizar en un ciclo que se ejecutará un número definido de veces.

El número de veces se obtiene del planteamiento del problema, o de una lectura que indica que el número de iteraciones se debe realizar para N ocurrencias.

# Estructura repetitiva (FOR)

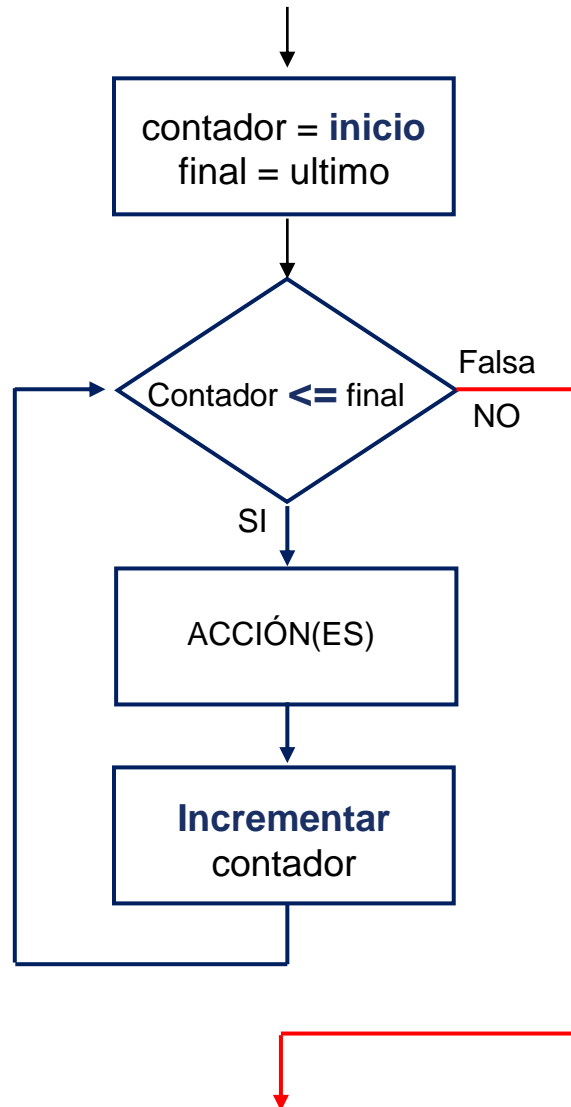
## Ejemplo:

Imagina que debes crear un programa que obtenga la nómina de los sueldos de 5 trabajadores de una empresa, pero que no puedes utilizar estructuras repetitivas.

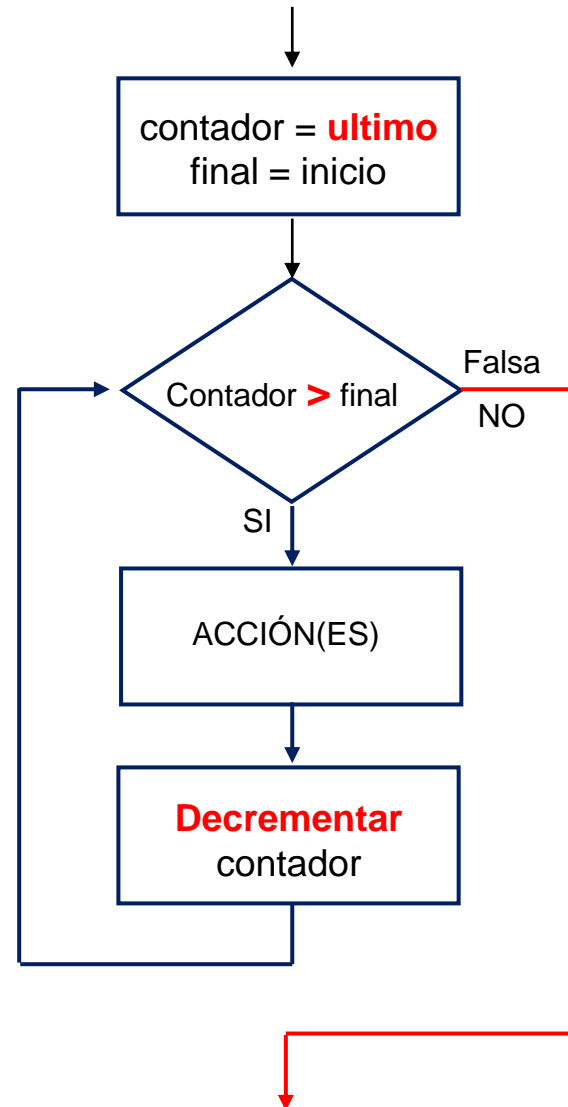
¿Qué sucedería si en lugar de tener 5 empleados, la empresa tuviera 100 o 1000?

```
System.out.print("Sueldo trabajador 1: ");  
sue1 = Leer.nextFloat();  
System.out.print("Sueldo trabajador 2: ");  
sue2 = Leer.nextFloat();  
System.out.print("Sueldo trabajador 3: ");  
sue3 = Leer.nextFloat();  
System.out.print("Sueldo trabajador 4: ");  
sue4 = Leer.nextFloat();  
System.out.print("Sueldo trabajador 5: ");  
sue5 = Leer.nextFloat();  
nomina = sue1 + sue2 + sue3 + sue4 + sue5;  
System.out.println("La nómina es "+nomina);
```

## Ciclo ascendente



## Ciclo descendente



```
for (inicio; condición; modificador) {  
    acción(es)  
}
```

## Ejemplo 3.1

Construye el programa en Java que muestre 5 veces el texto “Fundamentos de Programación”.

```
public class Ciclos {  
    public static void main(String[] args) {  
        // TODO code application logic here  
        for (int i=1; i<=5; i++) {  
            System.out.println(i+".- Fundamentos de Programación");  
        }  
    }  
}
```

Modifica el programa para que en lugar de que se muestre 5 veces ahora se muestre N veces (N será un valor introducido por teclado)

## Ejemplo 3.2

Construye el programa en Java que lea las calificaciones de N alumnos del grupo de Fundamentos de Programación.

```
public class Ejemplo32 {  
  
    public static void main(String[] args) {  
        Scanner Leer = new Scanner(System.in);  
        int N, calif;  
        System.out.print("¿Cuántas calificaciones son? ");  
        N = Leer.nextInt();  
        for (int i=1; i<=N; i++) {  
            System.out.print("Calificación "+i+": ");  
            calif = Leer.nextInt();  
        }  
    }  
}
```