



Apache
NetBeans

#22 E. REPETITIVAS

CICLO WHILE



Java™

```
default.rb
# Scientist: Result
# 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
end

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: mientras (WHILE)

Es la estructura algorítmica que se utiliza en un ciclo cuando no se sabe el número de veces que éste se ha de repetir. Dicho número depende de las proposiciones dentro del ciclo.

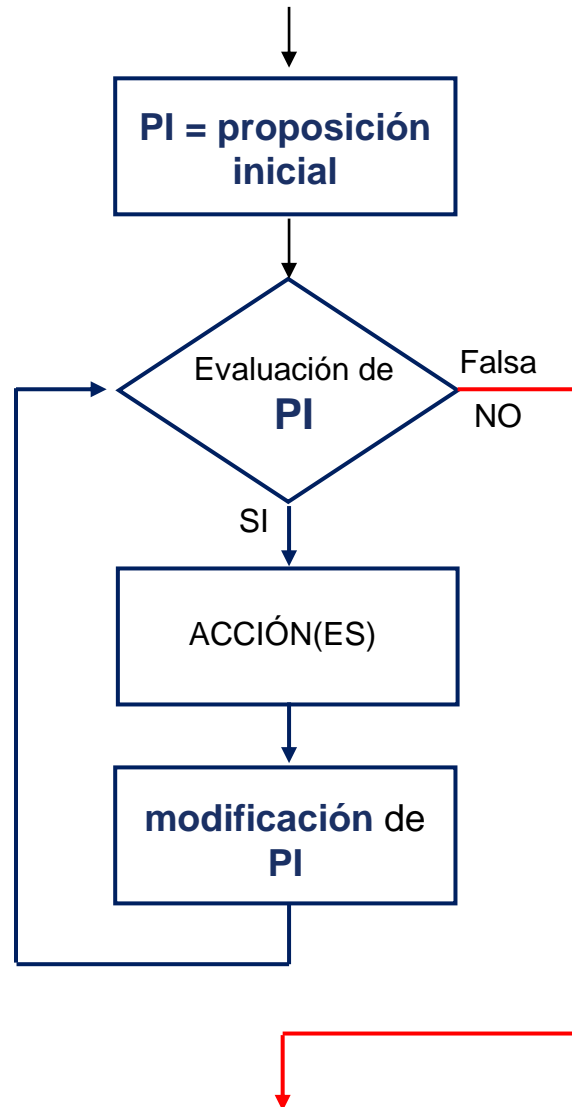
En la estructura mientras se distinguen dos partes:

Ciclo: Conjunto de instrucciones que se ejecutarán repetidamente.

Condición de terminación: La evaluación de esta condición permite decidir cuándo finalizará la ejecución del ciclo. La condición se evalúa al inicio del mismo.

Debe existir también un enunciado dentro del ciclo que afecte la condición, para evitar que el ciclo se ejecute indefinidamente.

Ciclo while



PI

```
while (condición con PI) {  
    acción(es)  
    modificación de PI  
}
```

Ejemplo 3.6

Construye el programa en Java que lea las descargas que un usuario realiza de un sitio web. La captura termina cuando el usuario introduce un cero.

```
import java.util.Scanner;

public class Ejemplo36 {

    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        int descarga;
        System.out.println("Tamaño de la descarga: ");
        descarga = Leer.nextInt(); //Proposición inicial
        while(descarga!=0) {
            System.out.print("Siguiente descarga: ");
            descarga = Leer.nextInt(); //Modificación Proposición inicial
        }
    }
}
```

Ejemplo 3.7

Construye el programa en Java que lea un indeterminado número de calificaciones. Se debe mostrar cuántas calificaciones fueron leídas. Para terminar de leer se debe introducir una calificación con -1.

```
import java.util.Scanner;

public class Ejemplo37 {

    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        int calif, cont=0;
        System.out.print("Introduce la calificación: ");
        calif = Leer.nextInt(); //Proposición inicial
        while (calif != -1) {
            cont++;
            System.out.print("Siguiente calificación: ");
            calif = Leer.nextInt(); //Modificación Proposición inicial
        }
        System.out.println("Se leyeron "+cont+" calificaciones");
    }
}
```

Ejemplo 3.8

Construye el programa en Java que obtenga la suma de un indeterminado número de gastos.

Para terminar de leer se debe introducir un sueldo en 0 (cero).

```
import java.util.Scanner;

public class Ejemplo38 {

    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        float gasto, suma=0;
        System.out.print("Introduce el gasto: ");
        gasto = Leer.nextFloat(); //Proposición inicial
        while (gasto != 0) {
            suma = suma + gasto;    // suma += sueldo;
            System.out.print("Siguiente sueldo: ");
            gasto = Leer.nextFloat(); //Modificación Prop. inicial
        }
        System.out.println("El total de gastos es $" + suma);
    }
}
```


Ejemplo 3.9

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

```
import java.util.Scanner;

public class Ejemplo39 {

    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        int N, i=1; //Proposición inicial
        System.out.print("¿Cuántas veces se mostrará la materia? ");
        N = Leer.nextInt();
        while(i<=N) {
            System.out.println(i+".-Fundamentos de Programación");
            i++;      //Modificación Proposición inicial
        }
    }
}
```