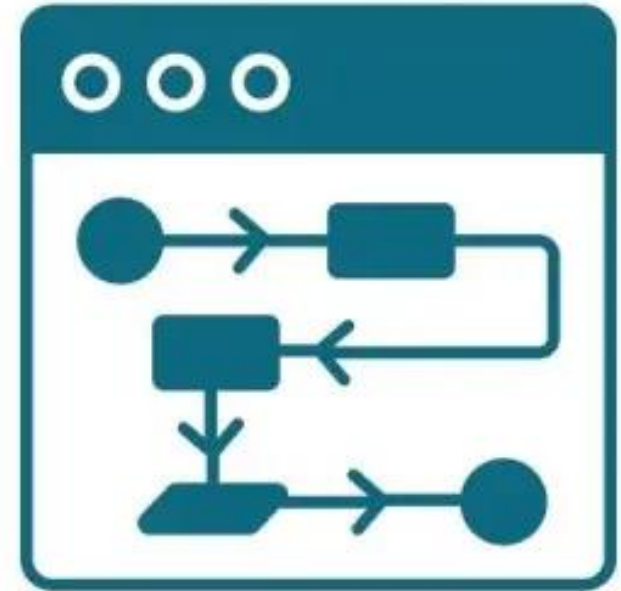




Algoritmos y Estructura de Datos

INSTRUCCIONES DE CONTROL DE PROCESAMIENTO



DOCENTE: MARX DANLY LEÓN TRUJILLO
FACULTAD DE ING. INDUSTRIAL Y SISTEMAS
E.P. INGENIERÍA DE SISTEMAS

PROGRAMACIÓN EN JAVA

- Ciclos Repetitivos
- While
- Do while
- For
- Codificación de Algoritmos



PROGRAMACIÓN EN JAVA

CICLOS REPETITIVOS

<code>iterar</code>	<code>(loop)</code>
<code>mientras</code>	<code>(while)</code>
<code>hacer-mientras</code>	<code>(do-while)</code>
<code>repetir</code>	<code>(repeat)</code>
<code>desde</code>	<code>(for)</code>

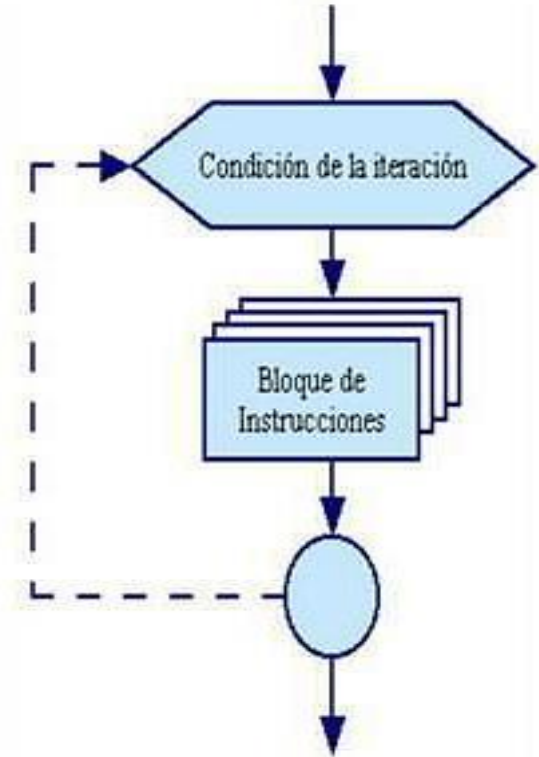
PROGRAMACIÓN EN JAVA

CICLOS REPETITIVOS

Las estructuras repetitivas se utilizan cuando se quiere que un conjunto de instrucciones se ejecuten un cierto número finito de veces.

BUCLES: Son estructuras que repiten una secuencia de instrucciones un número determinado de veces.

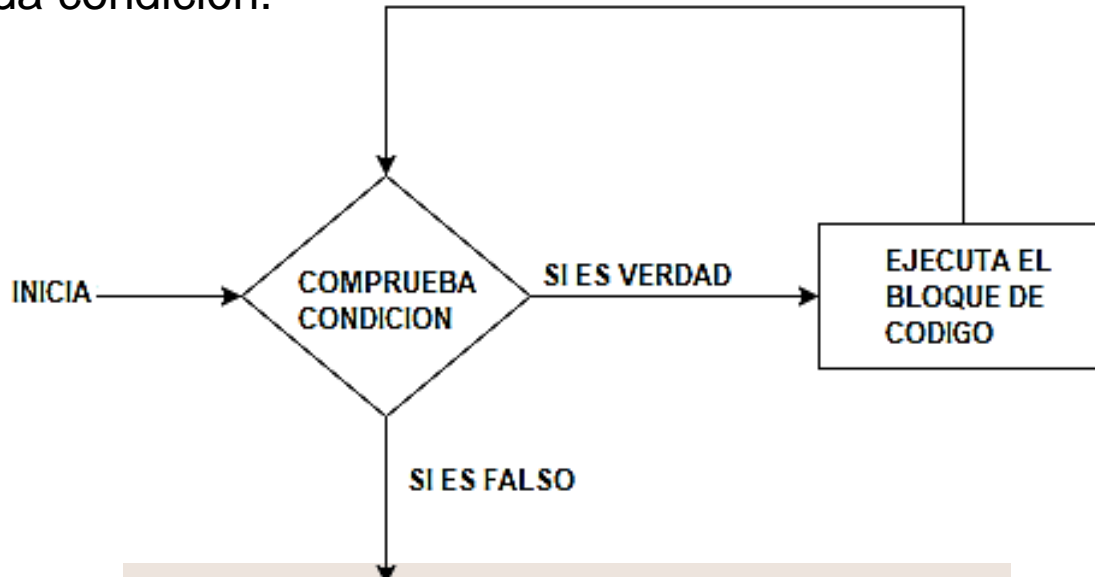
ITERACIÓN: Es el hecho de repetir la ejecución de una secuencia de acciones.



PROGRAMACIÓN EN JAVA

WHILE

La estructura repetitiva mientras (WHILE o DO WHILE: hacer mientras) es aquella en que el cuerpo del bucle se repite mientras se cumple una determinada condición.



PROGRAMACIÓN EN JAVA

WHILE

Pseudocódigo en español

```
mientras condicion hacer  
    accion S1  
    accion S2  
    .  
    .  
    acción Sn  
fin_mientras
```

Pseudocódigo en inglés

```
while condicion do  
    <acciones>
```

```
    .  
    .  
endwhile
```

o bien

```
dowhile condicion  
    <acciones>
```

```
    .  
    .  
enddo
```

PROGRAMACIÓN EN JAVA

WHILE

Ejemplo:

```
algoritmo suma_numero
var
    entero : N, TOTAL
    real : NUMERO, SUMA
inicio
    leer(N)
    TOTAL ← N
    SUMA ← 0
    mientras TOTAL > 0 hacer
        leer(NUMERO)
        SUMA ← SUMA + NUMERO
        TOTAL ← TOTAL - 1
    fin_mientras
    escribir('La suma de los', N, 'números es', SUMA)
fin
```

PROGRAMACIÓN EN JAVA

Estructura Mientras (“WHILE”)

Para la terminación de bucles con datos de entrada existen 4 métodos:

1. preguntar antes de la iteración,
2. encabezar la lista de datos con su tamaño,
3. finalizar la lista con su valor de entrada,
4. agotar los datos de entrada.

Estructura Hacer-Mientras (“DO-WHILE”)

El bucle **hacer-mientras** es análogo al bucle **mientras** y el cuerpo del bucle se ejecuta una y otra vez mientras la condición (expresión *booleana*) sea verdadera. El cuerpo del bucle siempre se ejecuta, al menos una vez, incluso aunque la expresión *booleana* sea falsa.

PROGRAMACIÓN EN JAVA

```
while (Expresion_lógica)
{
    sentencia_1;
    sentencia_2;
    ...
    sentencia_n;
}
```

```
do
{
    sentencia_1;
    sentencia_2;
    ...
    sentencia_n;
} while (expresion_lógica)
```

sentencia compuesta

```
while (Expresión_lógica)
    sentencia
```

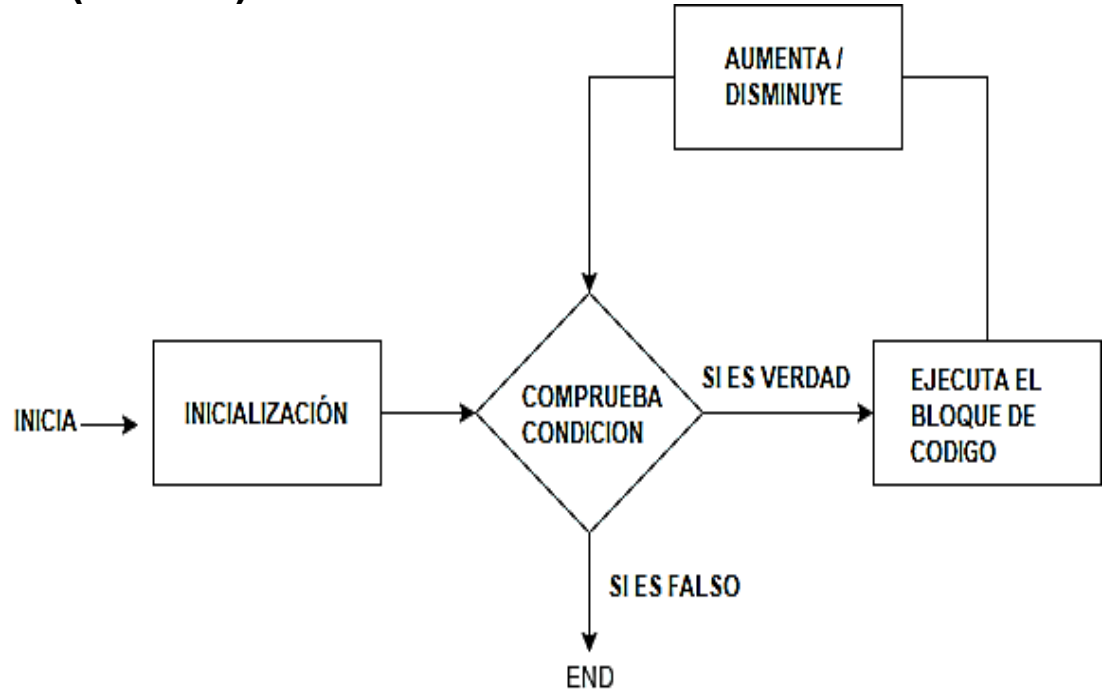
```
do sentencia
while (expresión_lógica)
```

sentencia simple

PROGRAMACIÓN EN JAVA

ESTRUCTURA DESDE/PARA ("FOR")

Ejecuta las acciones del cuerpo del bucle un número especificado de veces y de modo automático controla el número de iteraciones o pasos a través del cuerpo del bucle.



PROGRAMACIÓN EN JAVA

ESTRUCTURA DESDE/PARA (“FOR”)

Ejemplo:

```
algoritmo suma
var
    entero : I, N, S
inicio
    S ← 0 ,
    leer (N)
    desde I ← 1 hasta N hacer
        S ← S + I
    fin_desde
    escribir('Suma =', S)
fin
```

PROGRAMACIÓN EN JAVA

ESTRUCTURA DESDE/PARA ("FOR")

`desde v ← vi hasta vf`

`inc paso hacer`

`{inc, incremento}`

`dec`

`{dec, decremento}`

`<acciones>`

`.`

`.`

`.`

`fin_desde`

`desde i ← 20 hasta 10 decremento 1 hacer`

`<acciones>`

`fin_desde`

PROGRAMACIÓN EN JAVA

Ejercicios: FOR - WHILE

1. Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de Programación I. (while)
2. Ingresar 10 números e imprimir solamente los números pares. (while)
3. Realice un algoritmo que simule un reloj. (for)
4. Realizar un algoritmo que ingresado un número muestre la Secuencia Fibonacci (Do while).
5. Suponga que se tiene las calificaciones de un grupo de 15 alumnos. Realizar un algoritmo para calcular la calificación media y la calificación mas baja de todo el grupo. (for)

PROGRAMACIÓN EN JAVA

EJERCICIOS

1. Diseñar un algoritmo que dado un número total de horas, devuelva el número de semanas, días y horas equivalentes. Por ejemplo, dado un total de 1000 horas debe mostrar 5 semanas, 6 días y 16 horas.
2. Se necesita calcular el salario de un empleado, teniendo en cuenta que su salario mensual es de S/. 1500 y el empleado ha trabajado 18 horas extras. Se tiene en cuenta que la hora extra se paga 25% mas y tiene los descuentos de seguro, y aporte a la ONP.

PROGRAMACIÓN EN JAVA

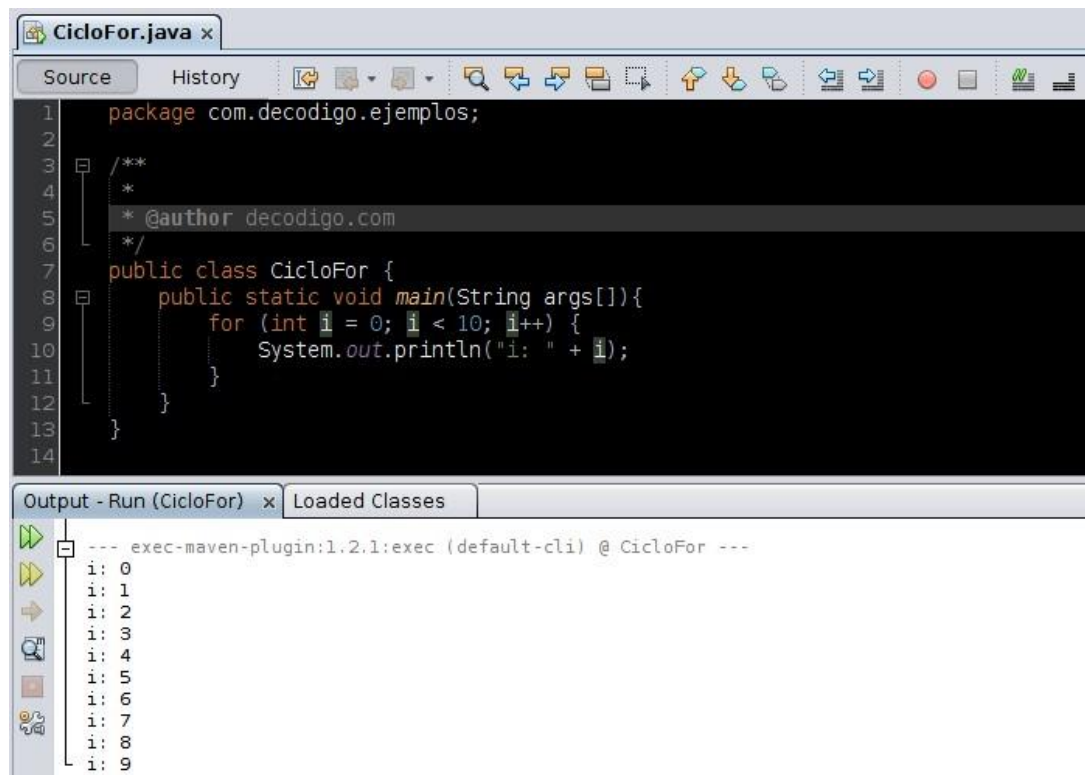
EJERCICIOS

3. Hacer un programa que simule un cajero automático con un saldo inicial de 1000 dólares, con el siguiente menú de opciones:
 - Ingresar dinero a la cuenta
 - Retirar dinero
 - Salir

PROGRAMACIÓN EN JAVA

```
public class JavaApplication42 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de programacion I  
        int notas[]={12,14,10,8,12,14,17,12};  
        int tamaño = notas.length;  
        int x = 0;  
        int suma=0;  
        while (x < tamaño){  
            suma = suma + notas[x];  
            x++;  
        }  
        double prom_nota = suma / 7;  
        System.out.println("El promedio es: "+ prom_nota);  
        System.out.println(suma);  
    }  
}
```


PROGRAMACIÓN EN JAVA



The screenshot shows an IDE window titled "CicloFor.java". The editor displays the following Java code:

```
1 package com.decodigo.ejemplos;
2
3 /**
4  *
5  * @author decodigo.com
6  */
7 public class CicloFor {
8     public static void main(String args[]){
9         for (int i = 0; i < 10; i++) {
10             System.out.println("i: " + i);
11         }
12     }
13 }
14
```

Below the editor, the "Output - Run (CicloFor)" tab is active, showing the execution output:

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ CicloFor ---
i: 0
i: 1
i: 2
i: 3
i: 4
i: 5
i: 6
i: 7
i: 8
i: 9
```