

INTRODUCCIÓN A LA PROGRAMACIÓN

**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

Luz A. Sánchez Gálvez
sanchez.galvez@correo.buap.mx
luzsg@hotmail.com

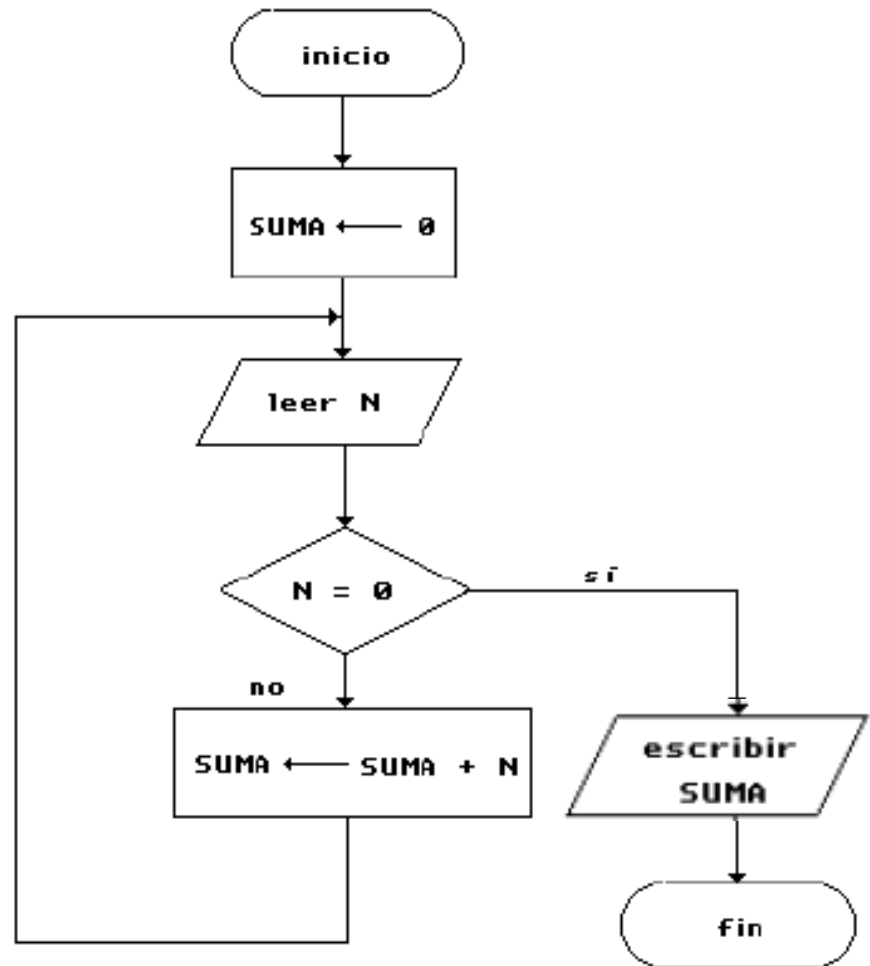
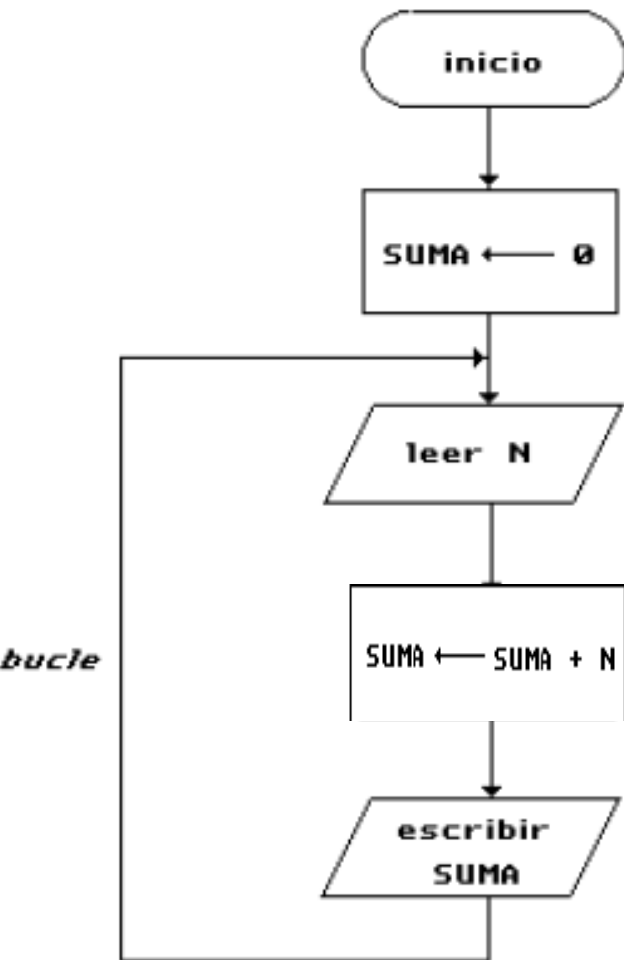
Estructuras Repetitivas

- ▶ La computadora está especialmente diseñada para aplicaciones en las que una o más operaciones se deben repetir muchas veces.
- ▶ Un **bucle (ciclo o loop)**, es un segmento de un programa cuyas instrucciones se repiten un número determinado de veces o mientras se cumpla una determinada condición.
- ▶ Es necesario que se establezcan mecanismos para controlar esta tarea repetitiva, de no ser así el bucle puede convertirse en un proceso infinito.



Estructuras Repetitivas

- ▶ Ejemplo de un bucle infinito y bucle finito



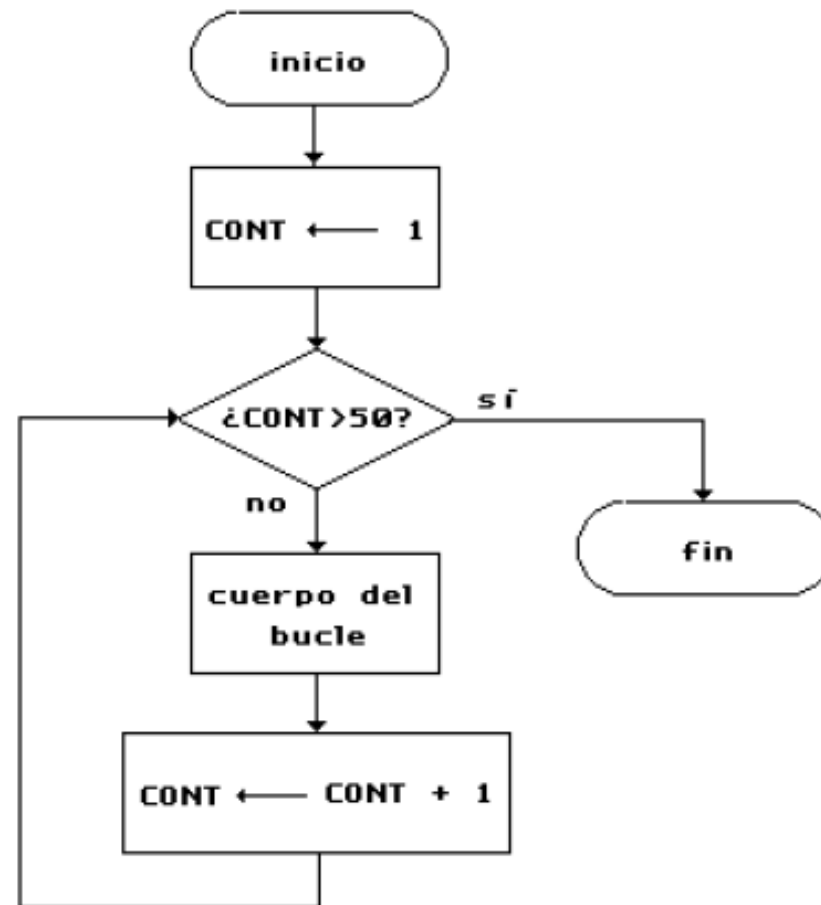
Estructuras Repetitivas

- ▶ Los procesos que se repiten varias veces en un programa necesitan en muchas ocasiones contar el número de repeticiones.
- ▶ Una forma de hacerlo es utilizar una variable llamada **contador**, cuyo valor se incrementa o decrementa en una cantidad constante en cada repetición que se produzca.
- ▶ El contador puede ser positivo (incrementos de uno en uno) o negativo (decrementos de uno en uno).



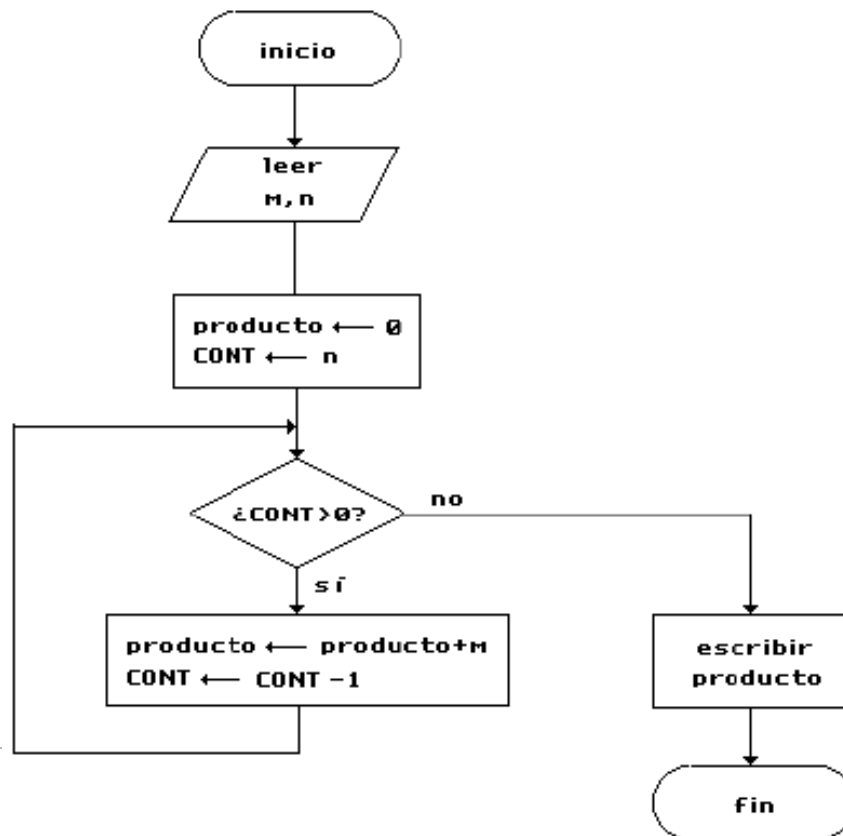
Estructuras Repetitivas

- El diagrama muestra un grupo de instrucciones (cuerpo del bucle) que se repiten, CONT es la variable contador. La instrucción que la actualiza es: $CONT = CONT + 1$.



Estructuras Repetitivas

- ▶ La multiplicación $n \times m$, se realiza sumando m , un número n de veces.
- ▶ El contador se decrementa: comienza en n y se va decrementando hasta llegar a cero; se termina el bucle y se realiza la acción escribir.



Estructuras Repetitivas

- ▶ Otro tipo de variable, asociada al funcionamiento de un bucle es un **acumulador**, cuya misión es almacenar una cantidad variable, resultante de operaciones sucesivas y repetidas.
- ▶ Un acumulador realiza una función parecida a la de un contador, con la diferencia de que el incremento o decremento, de cada operación es variable en lugar de constante.
- ▶ La instrucción, **suma = suma + n**, añade en cada iteración el valor de la variable n, (que es leída) por lo que **suma** es un ejemplo de acumulador.

Estructuras Repetitivas

- ▶ Una **estructura repetitiva** marca la iteración de una serie de acciones basándose en un bucle.
- ▶ Esta estructura debe constar de tres partes básicas:
 - ▶ **condición** (para finalizar la repetición)
 - ▶ **cuerpo del bucle** (conjunto de instrucciones que se repiten)
 - ▶ **salida del bucle** (instrucción a la que se accede una vez que se decide finalizar)



Estructuras Repetitivas

- ▶ Para obtener la suma de una serie de números, tras leer cada número, se añade una variable SUMA que contenga las sumas sucesivas parciales (SUMA se inicia en cero).
- ▶ El código correspondiente utilizará sucesivamente instrucciones tales como:

```
suma=0;
```

```
int i=Integer.parseInt(JOptionPane.showInputDialog("Introduzca  
un numero"));
```

```
if (i== 0)
```

```
    System.out.println(suma);
```

```
else
```

```
    suma =suma+i;
```

- ▶ que se pueden repetir muchas veces; éstas constituyen el **cuerpo del bucle**.



Estructuras Repetitivas

- ▶ Una vez que se tiene el cuerpo del bucle, se debe plantear cuántas veces se debe repetir.
- ▶ Se puede pedir al usuario la cantidad **n** de números que desea sumar, esto es, el número de iteraciones del bucle.
- ▶ Se utiliza un contador de iteraciones, **total**, que se inicializa en **n** y se decrementa en uno cada vez que el bucle se repite; se introduce al cuerpo del bucle:
total = total - 1;
- ▶ También se podría inicializar la variable total en 0 ó en 1, e ir incrementándolo en uno, en cada iteración, hasta llegar al número deseado **n**.

Estructuras Repetitivas

- ▶ Aunque la condición de finalización puede evaluarse en distintos lugares del código, no es recomendable que ésta se pueda efectuar a mitad del cuerpo del bucle, por lo que es bueno que se produzca al principio o al final del mismo.
 - ▶ Según donde se sitúe la condición de salida, dará lugar a distintos tipos de estructuras repetitivas:
 - ▶ ***do{...}while(condición)*** **[hacer-mientras...]**
 - ▶ ***while (condición){ .. }*** **[mientras...]**
 - ▶ ***for(){..}*** **[para...]**
-



Estructuras Repetitivas

► Estructura **for** (*para*)

- Estructura más simple, consiste en que la condición de salida se basa en un contador que cuenta el número de iteraciones, se utiliza cuando se conoce de antemano el número de iteraciones.

► Ejemplo:

```
suma = 0;
```

```
for ( i = 1; i <= n; i++)
```

```
{  numero;
```

```
    suma = suma + numero;
```

```
}
```

Inicializa
contador i

Condición de paro
del contador i

Incremento en 1
del contador i

- i es un contador que cuenta desde un valor inicial (1) hasta el valor final (n) con los incrementos que se consideren (de uno en uno en este caso).

Estructuras Repetitivas

- ▶ La variable índice o de control normalmente será de tipo entero y se utilizan como identificador las letras: i,j,k
- ▶ El incremento de la variable índice es 1 en cada iteración si no se indica lo contrario.
- ▶ Si se deben expresar incrementos distintos de +1 el formato de la estructura es:

**si ($i > n$) usar
 incremento $i++$
else decremento $i--$**

Estructuras Repetitivas

- ▶ Obviamente, si el valor inicial de la variable contador *i* es menor que el valor de *n*, los incrementos deben ser positivos, ya que en caso contrario la secuencia de acciones no se ejecutaría.
 - ▶ De igual modo si el valor inicial de *i* es mayor que el valor de *n*, entonces se deben hacer decrementos.
 - ▶ Así:

```
for (i =20; i<10; i--) {  
    <acciones>  
}
```
 - ▶ no ejecutaría nunca el cuerpo del bucle.
-



Estructuras Repetitivas

- ▶ **Estructura while (mientras)**

- ▶ Cuando la condición de salida del bucle se realiza al principio del mismo, éste se ejecuta mientras se verifica una cierta condición.

- ▶ **Sintaxis**

while (condición)

{

 <acciones>

}

- ▶ El cuerpo del bucle se repite mientras se cumple una determinada condición.
-



Estructuras Repetitivas

- ▶ Cuando se ejecuta la instrucción mientras, lo primero que sucede es la evaluación de la condición:
 - ▶ Si es **falsa**, no se ejecuta ninguna acción y el programa prosigue en la siguiente instrucción a la finalización del bucle;
 - ▶ Si es **verdadera**, entonces se ejecuta el cuerpo del bucle.



Ejemplo

- ▶ Código que determina si un número es par. La idea es restar consecutivamente dos al número leído, si se obtiene 2, es par, si se obtiene otro valor (el 1), entonces es impar:

```
int n=Integer.parseInt(JOptionPane.showInputDialog("Introduzca un  
numero entero positivo"));
```

```
while (n > 2)
```

```
    n = n -2;
```

```
if (n == 2)
```

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

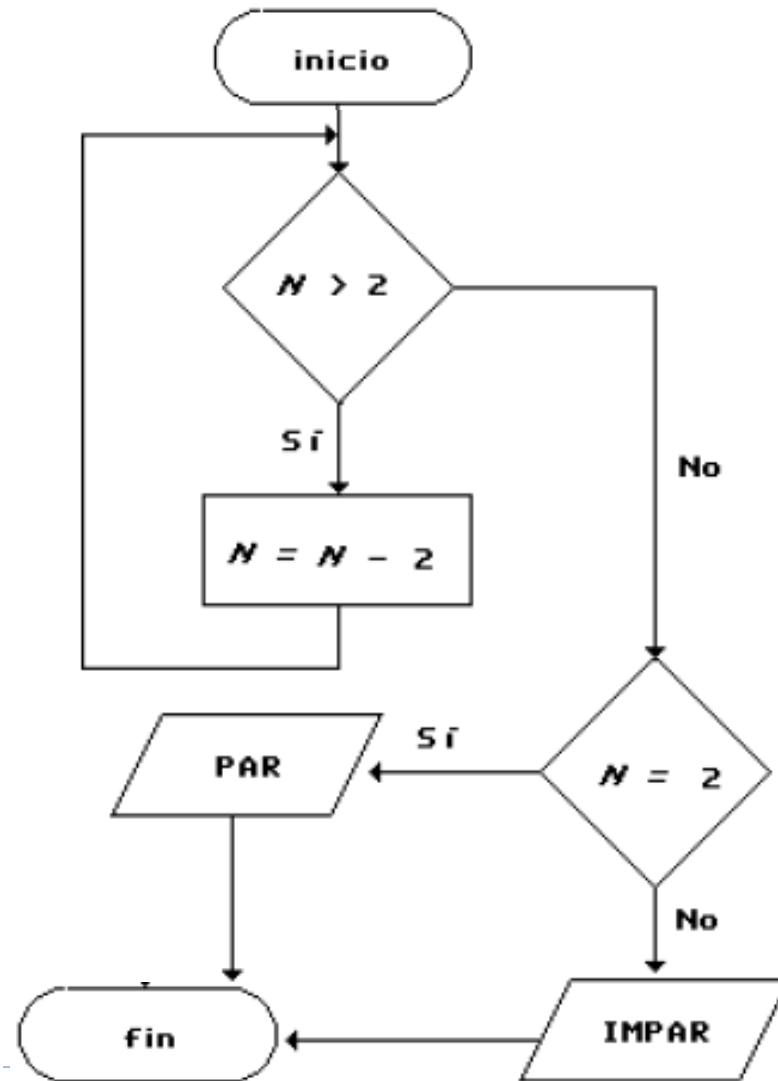
```
else
```

```
    System.out.println("es un número impar");
```



Ejemplo

► Diagrama de Flujo



Estructuras Repetitivas

- ▶ En una estructura **mientras** si la primera evaluación de la condición es falsa, el cuerpo del bucle nunca se ejecuta.
- ▶ Puede parecer inútil ejecutar el cuerpo del bucle cero veces, ya que no tendrá efecto en ningún valor o salida; aunque, puede ser una acción deseada.



Ejemplo

- Un bucle para leer las calificaciones de un estudiante y obtener su promedio, dejará de ejecutarse cuando la calificación leída sea negativa.

```
int c,suma, calif;  
c = 0;  
suma = 0;  
int calif=Float.parseInt(JOptionPane.showInputDialog("Introduzca  
calificacion"));  
while (calificacion >= 0)  
{  
    suma = suma + calif;  
    c = c+1;  
int calif=Float.parseInt(JOptionPane.showInputDialog("Introduzca  
calificacion"));  
}
```

► promedio=suma/c;

Estructuras Repetitivas

- ▶ **Estructura do { ... }while(condición)**
- ▶ En esta estructura la condición de salida se sitúa al final del bucle, el cual se ejecuta hasta que se verifique una cierta condición.
- ▶ Existen muchas situaciones en las que se desea que un bucle se ejecute al menos una vez antes de comprobar la condición de repetición.
- ▶ La estructura **do-while**, se ejecuta hasta que se cumpla una determinada condición, que se comprueba al final del bucle.

```
do {  
    <acciones>  
} while(condición);
```

Estructuras Repetitivas

```
contador = 0;
```

```
do
```

```
{
```

```
int n =Float.parseInt(JOptionPane.showInputDialog("Introduzca  
un número"));
```

```
contador =contador+1;
```

```
    } while (contador < 10)
```

```
System.out.println("números leídos:" + contador);
```



Estructuras Repetitivas

- ▶ Este bucle se repite hasta que el valor de la variable contador llegue a 10, lo que sucederá después de 10 ejecuciones del mismo.
- ▶ **Nota:** si en vez de 10 se pusiera 0, el cuerpo del bucle se ejecutará siempre al menos una vez.
- ▶ El bucle **do...while** se repite mientras la condición sea falsa, justo lo opuesto a la estructura **mientras**.



Estructuras Repetitivas

- ▶ Las tres estructuras repetitivas son susceptibles de intercambio entre ellas, así por ejemplo es posible, sustituir una estructura **for**, por una **mientras**; con incrementos positivos o negativos de la variable índice.



Estructuras Repetitivas

- ▶ La estructura **for** con incremento positivo es equivalente a la estructura **while** A), y la estructura **for** con decremento negativo es equivalente a la estructura **while** B).

A) `i = valorinicial;`
 `while (i <= valorfinal)`
 `i++;`

B) `i = valorinicial;`
 `while (i <= valorfinal)`
 `i--;`

