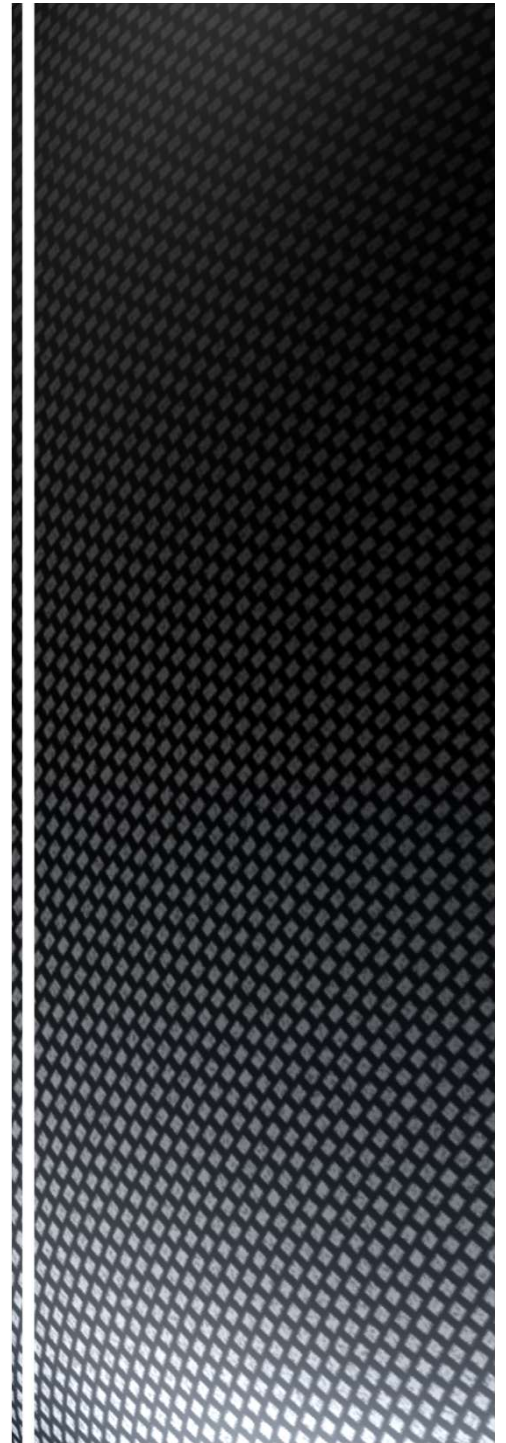


Estructuras de control

Unidad 4



Contenidos

- **Instrucciones de control**
- **Instrucciones condicionales**
 - Instrucciones if
 - Instrucciones if-else
 - Instrucciones if-else anidadas
 - Estructura condicional múltiple switch
- **Instrucciones repetitivas**
 - Instrucciones while
 - ✓ Ciclos controlador por contador
 - ✓ Ciclos controlador por condición
 - Ciclos sin fin
 - Instrucciones do-while
 - ✓ Creación de filtros
 - Instrucciones for
 - Ciclos anidados

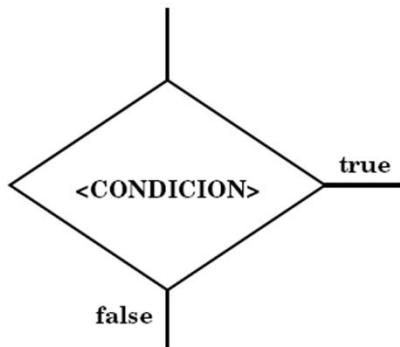
Instrucciones de Control

Las ***instrucciones de control*** determinan el orden en que se ejecutan las instrucciones de un programa:

- **Instrucciones Secuenciales:** Las instrucciones se van ejecutando sucesivamente, siguiendo el orden de aparición.
- **Instrucciones Condicionales:** Permiten decidir si un conjunto de instrucciones se ejecutan o no.
- **Instrucciones Repetitivas:** Permiten repetir un conjunto de instrucciones un determinado número de veces.

Instrucciones de Control

Ejemplo



Bifurcador de flujo

Instrucción
if-else

```
1 //Programa que muestra el mayor de dos números
2 import java.util.*;
3
4 public class Mayor {
5     public static void main(String args[]) {
6         int dato1, dato2, mayor;
7
8         Scanner entrada = new Scanner(System.in);
9
10        //Pedimos el primer dato
11        System.out.println("Introduce el primer dato:");
12        dato1 = entrada.nextInt();
13
14        //Pedimos el segundo dato
15        System.out.println("Introduce el segundo dato:");
16        dato2 = entrada.nextInt();
17
18        //Comprobamos
19        if(dato1 > dato2){
20            mayor = dato1;
21        }else{
22            mayor = dato2;
23        }
24
25        //Mostramos el resultado
26        System.out.println("El mayor es: " + mayor);
27    }
28 }
```

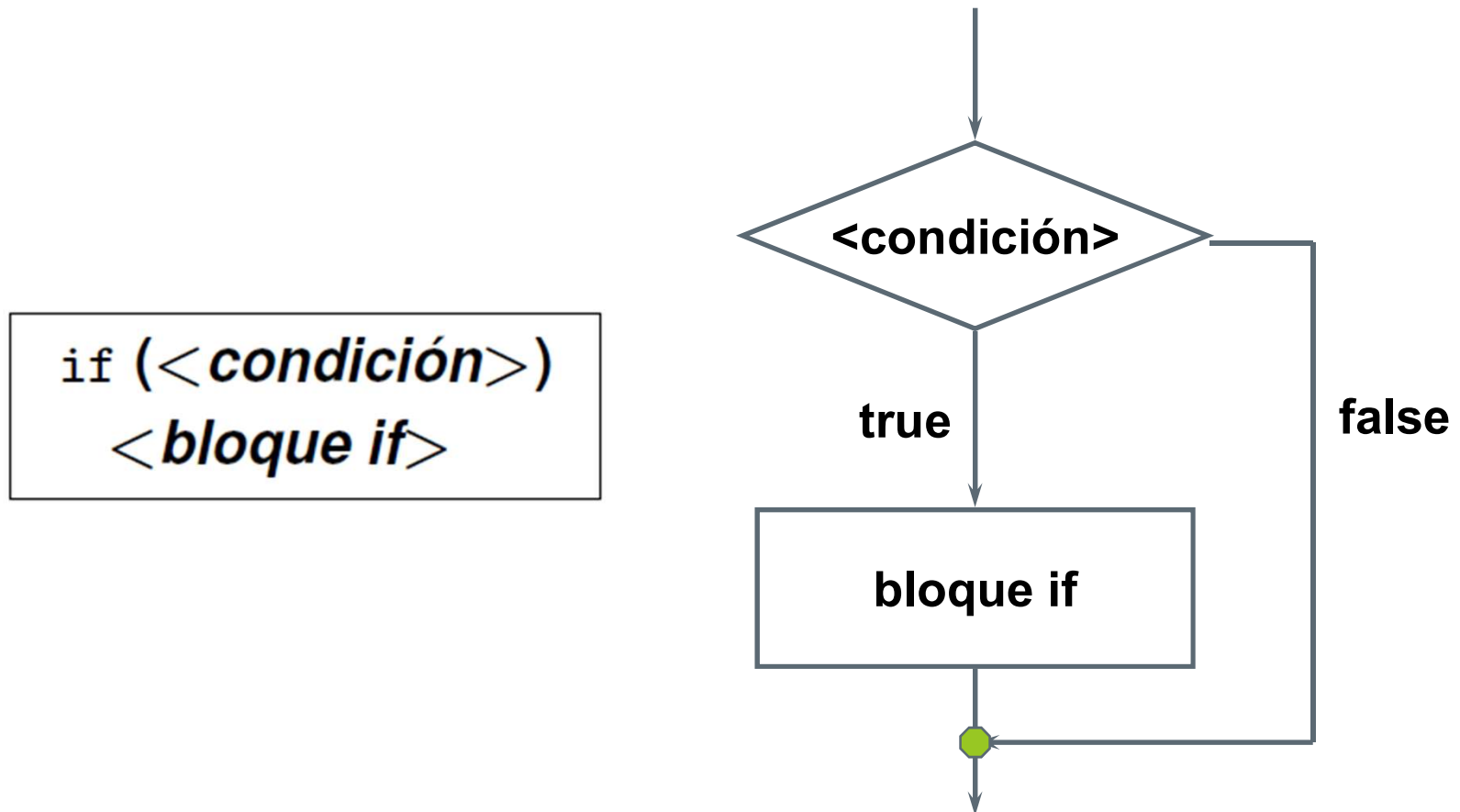
Expresión Booleana
(Exp. Relacional)

Instrucciones Condicionales

- Instrucción if
- Instrucción if-else
- Instrucciones if-else anidadas
- Estructura condicional múltiple switch

Instrucciones Condicionales

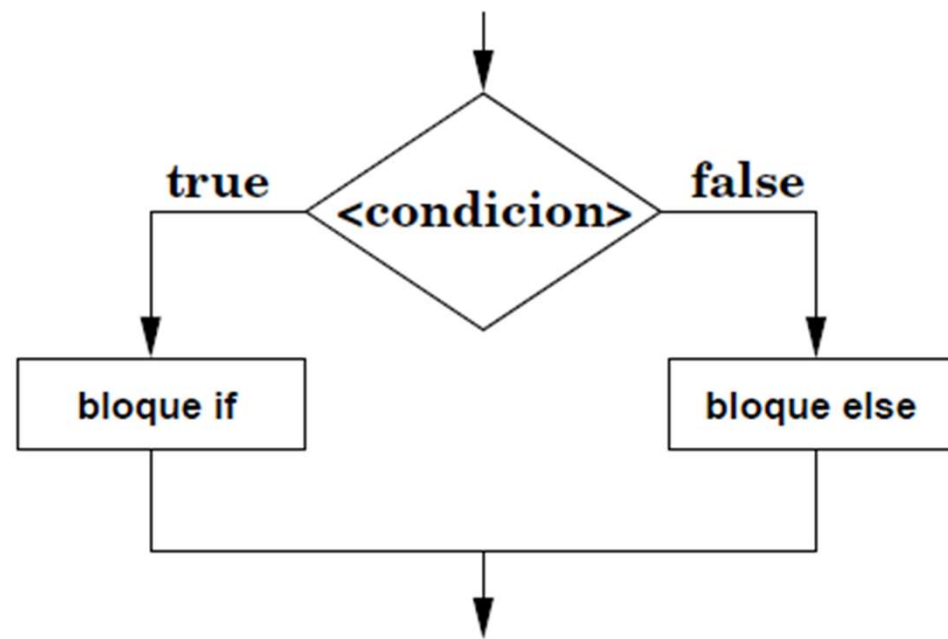
Instrucción if



Instrucciones Condicionales

Instrucción if-else

```
if <condición>  
    <bloque if>;  
else  
    <bloque else>;
```



Instrucciones Condicionales

Instrucciones anidadas if-else

- Dentro de un **if(else)** puede aparecer otro **if(else)**.

```
1  import java.util.*;
2
3  public class Numeros {
4      public static void main(String args[]) {
5          int x;
6
7          Scanner entrada = new Scanner(System.in);
8
9          System.out.println("Introduce un número:");
10         x = entrada.nextInt();
11
12         if(x < 0){
13             System.out.println(x + " Es negativo");
14         }else{
15             if(x > 0){
16                 System.out.println(x + " Es positivo");
17             }else{
18                 System.out.println(x + " Es cero");
19             }
20         }
21     }
22 }
```


Instrucciones Condicionales

Estructura condicional múltiple switch

- Permite elegir entre múltiples alternativas “por igualdad”.

```
switch (<expresión>) {  
    case <valor1>:  
        <sentencias>  
        break;  
    case <valor2>:  
        <sentencias>  
        break;  
    case <valor3>:  
        <sentencias>  
        break;  
    default:  
        <sentencias>  
}  
}
```

```
switch(opcion){  
    case 1:  
        System.out.println("Has aceptado.");  
        break;  
    case 2:  
        System.out.println("Has cancelado.");  
        break;  
    case 3:  
        System.out.println("Has solicitado ayuda.");  
        break;  
    default:  
        System.out.println("Respuesta no permitida.");  
        break;  
}
```

Instrucciones Condicionales

Ejemplo swicth

```
1  import java.util.*;
2
3  public class EjemploSwitch {
4      public static void main(String args[]) {
5          int opcion;
6
7          Scanner entrada = new Scanner(System.in);
8
9          System.out.println("Pulsa 1 para aceptar.\nPulsa 2 para cancelar.\nPulsa 3 para ayuda.");
10         System.out.println("Introduce un número:");
11         opcion = entrada.nextInt();
12
13         switch(opcion){
14             case 1:
15                 System.out.println("Has aceptado.");
16                 break;
17
18             case 2:
19                 System.out.println("Has cancelado.");
20                 break;
21
22             case 3:
23                 System.out.println("Has solicitado ayuda.");
24                 break;
25
26             default:
27                 System.out.println("Respuesta no permitida.");
28                 break;
29         }
30     }
31 }
```

Instrucciones Repetitivas

Las ***instrucciones repetitivas*** son también conocidas como *bucles, ciclos o lazos*.

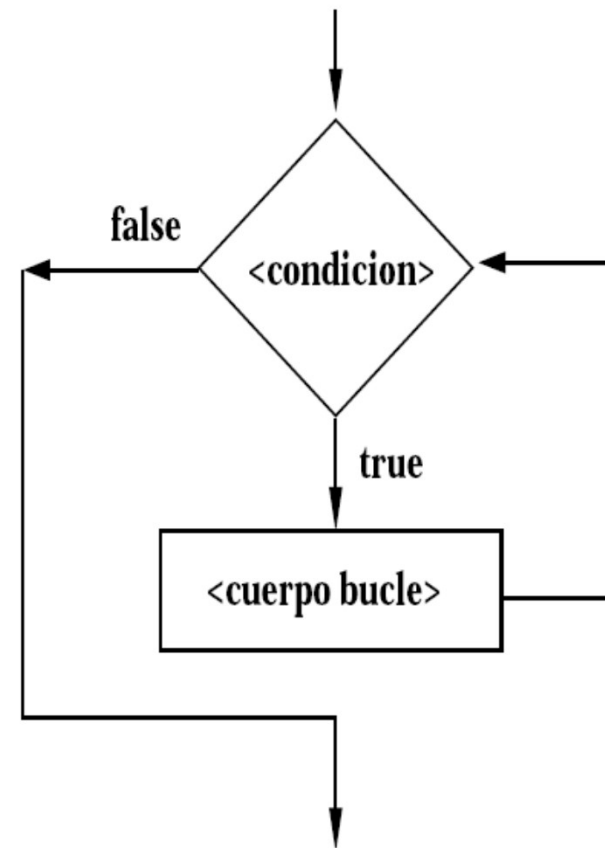
Una instrucción repetitiva permite la ejecución de un conjunto de instrucciones:

- *Ciclos Controlados por Contador*: Se ejecuta un número de veces conocido a priori.
- *Ciclos Controlados por Condición*: Se ejecuta mientras se satisface una condición.

Instrucciones Repetitivas

Instrucción while

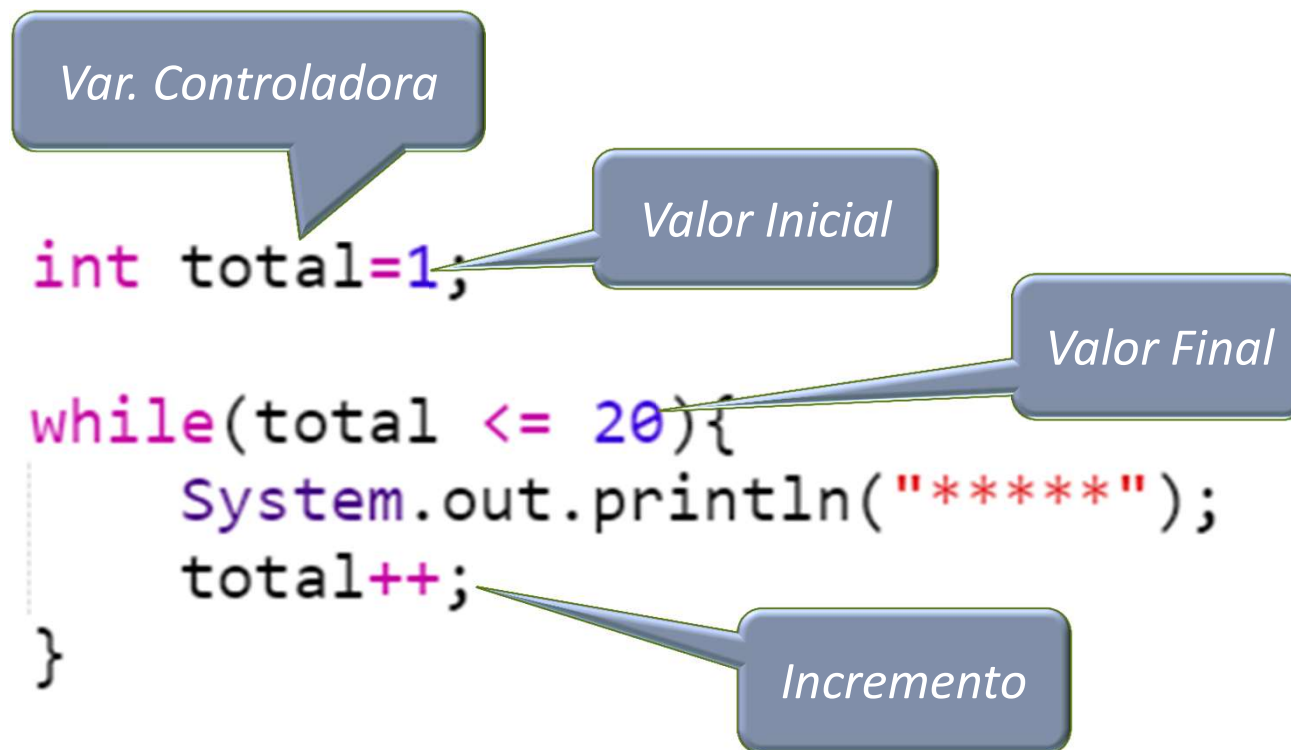
```
while (<condición>) {  
    <cuerpo bucle>  
}
```



Instrucciones Repetitivas

Ciclos controlados por CONTADOR con while

Ejemplo: Escribir 20 líneas con 5 asteriscos cada una.



Instrucciones Repetitivas

Ciclos controlados por CONTADOR con while

Ejemplo: Leer un número positivo tope desde teclado e imprimir tope líneas con 5 asteriscos cada una.

```
System.out.println("¿Cuántas líneas de asteriscos quiere imprimir?");  
tope = entrada.nextInt();  
total = 0;
```

```
while(total < tope){  
    System.out.println("*****");  
    total++;  
}
```

El valor final puede venir especificado por una variable

Instrucciones Repetitivas

Ciclos controlados por CONDICIÓN con while

Programa que suma los valores leídos desde teclado hasta que se introduzca el valor -1.

```
1  import java.util.*;
2
3  public class Suma {
4      public static void main(String args[]) {
5          Scanner entrada = new Scanner(System.in);
6
7          int suma, numero;
8
9          suma = 0;
10
11         System.out.println("Introduce números:");
12         numero = entrada.nextInt();
13
14         while(numero != -1){
15             suma = suma + numero;
16             numero = entrada.nextInt();
17         }
18
19         System.out.println("La suma es:" + suma);
20     }
21 }
```

Lectura Anticipada

Primer candidato

¿Es Bueno?

Leer siguiente candidato

Instrucciones Repetitivas

Ciclos sin fin

A la hora de diseñar un ciclo hay que **comprobar**:

- ¿Qué ocurre la primera vez que se entra en el ciclo?
- ¿Se han inicializado previamente las variables (contadores, acumuladores, etc.) que se modifican dentro del ciclo?
- ¿Qué ocurre al finalizar el ciclo?
- ¿Realmente finaliza el ciclo en algún momento?

¿Cuántas iteraciones se producen?

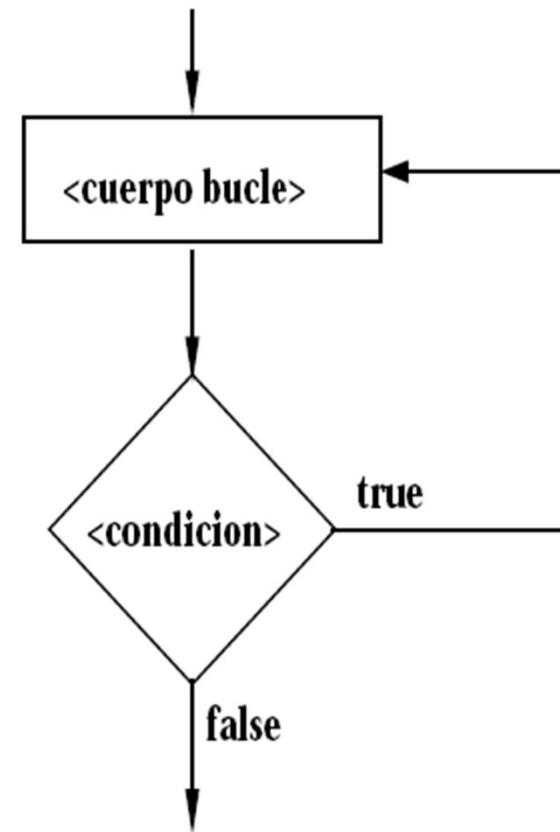
```
int contador = 2;  
  
while(contador < 3){  
    contador--;  
    System.out.println(contador);  
}
```

*Esta condición
nunca será false.
El ciclo nunca parará*

Instrucciones Repetitivas

Instrucciones do-while

```
do {  
    < cuerpo bucle >  
} while (< condición >);
```



Instrucciones Repetitivas

Creación de FILTROS con do-while

Objetivo: Leer un valor y no permitir al usuario que lo introduzca fuera de un rango determinado.

```
1  import java.util.*;
2
3  public class Valores {
4      public static void main(String args[]) {
5          Scanner entrada = new Scanner(System.in);
6
7          int valor;
8
9          do{
10             System.out.println("Introduce un valor positivo:");
11             valor = entrada.nextInt();
12         }while(valor < 0);
13
14         System.out.println(Math.sqrt(valor));
15     }
16 }
```

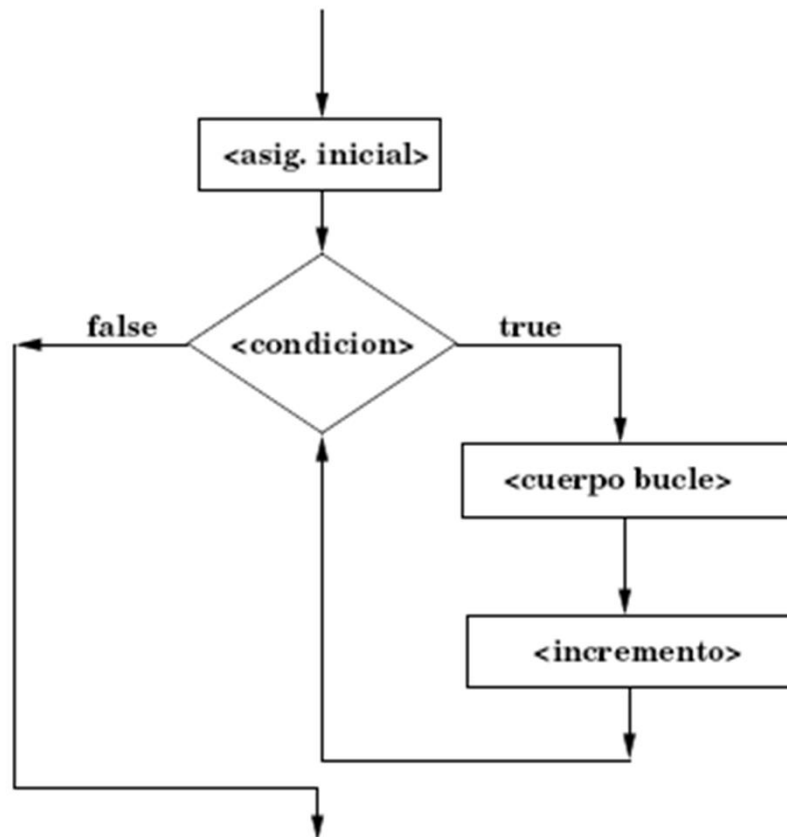
*Hay que realizar la lectura
mientras NO se cumpla la
condición requerida:*

Mientras no sea positivo

Instrucciones Repetitivas

Instrucciones for

```
for ([<asig.inicial>]; [<condicion>]; [<incremento>])  
  <cuerpo bucle>
```



```
<asig.inicial>;  
while (<condicion>)  
{  
  <cuerpo bucle>;  
  <incremento>;  
}
```

Instrucciones Repetitivas

Ciclos controlados por CONTADOR con for

```
1  import java.util.*;
2
3  public class Valores {
4      public static void main(String args[]) {
5          Scanner entrada = new Scanner(System.in);
6
7          int i, valor, suma;
8          double media;
9
10         suma = 0;
11
12         for(i=1; i <= 5; i++){
13             System.out.println("Introduce el número " + i + ":");
14             valor = entrada.nextInt();
15             suma = suma + valor;
16         }
17
18         media = suma/5.0;
19         System.out.println("La media es: " + media);
20     }
21 }
```

MUY LEGIBLE:

*Todos los elementos del ciclo controlado por contador se encuentran en la cabecera del **for***

Instrucciones Repetitivas

Ciclos controlador por CONTADOR con for

IMPORTANTE:

No se debe modificar el valor de la variable controladora, ni el valor final dentro del cuerpo del bucle.

Ejemplo. Sumar pares hasta un tope

```
suma = 0;

for (par=2; par<=tope ; par++){
    if (par%2 == 0)
        suma = suma+valor;
    else
        par++;    // :-(
}
```

Al modificar la variable controladora en este punto, se dificulta mucho la compresión del código.

Instrucciones Repetitivas

Ciclos anidados

Dentro de un ciclo puede aparecer otro ciclo.

```
1 //Imprime la tabla de multiplicar de los N primeros números
2
3 import java.util.*;
4
5 public class TablaMultiplicar {
6     public static void main(String args[]) {
7         Scanner entrada = new Scanner(System.in);
8
9         int i, j;
10        final int N = 3;
11
12        for(i=1; i <= N; i++){
13            for(j=0; j <= 9; j++){
14                System.out.println(i + "*" + j + "=" + i*j);
15            }
16            System.out.print("\n");
17        }
18    }
19 }
```

```
1*0=0
1*1=1
1*2=2
1*3=3
1*4=4
1*5=5
1*6=6
1*7=7
1*8=8
1*9=9
```

```
2*0=0
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
```

```
3*0=0
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
```