

Programación Frontend y Backend

BLOQUE JAVA

Estructuras Básicas
(Condicionales e Iterativas)

01

Estructuras Condicionales



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

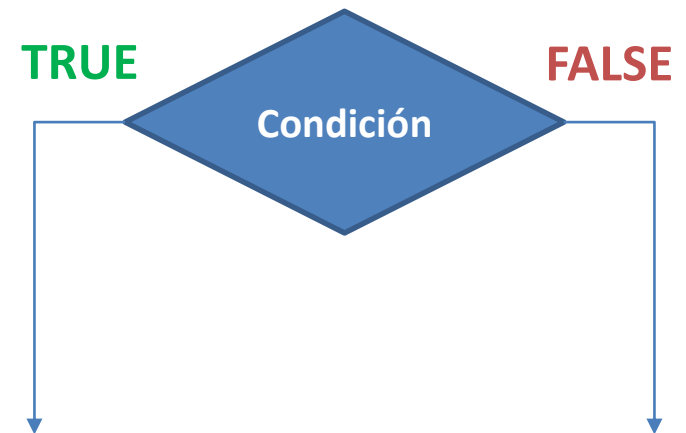


Estructuras Condicionales

- IF
- OPERADOR TERNARIO
- SWITCH

Estructuras Condicionales (IF)

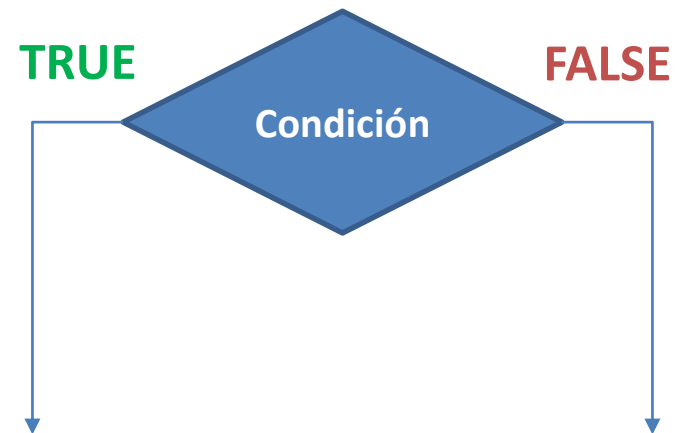
- Es la estructura condicional más simple y más utilizada en **Java**, ya que nos permite evaluar condiciones.
- Nos permite ejecutar o no ejecutar un bloque de código en concreto.



Estructuras Condicionales (IF)

- Ej: VOTO ELECTRÓNICO

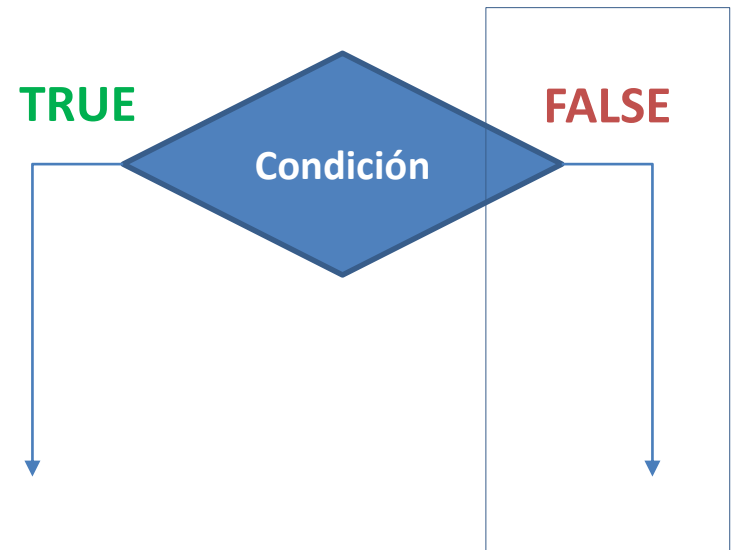
Si la persona es mayor de edad se le permite votar, en caso contrario no permitimos votar.



Estructuras Condicionales (IF)

- Ej: VOTO ELECTRÓNICO

```
int edad = 17;  
  
//Es mayor de edad.  
if (edad >= 18) {  
    System.out.println("¡Adelante!, puede votar.");  
} else {  
    System.out.println("Lo siento, no puede votar.");  
}
```



Lo siento, no puede votar.



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



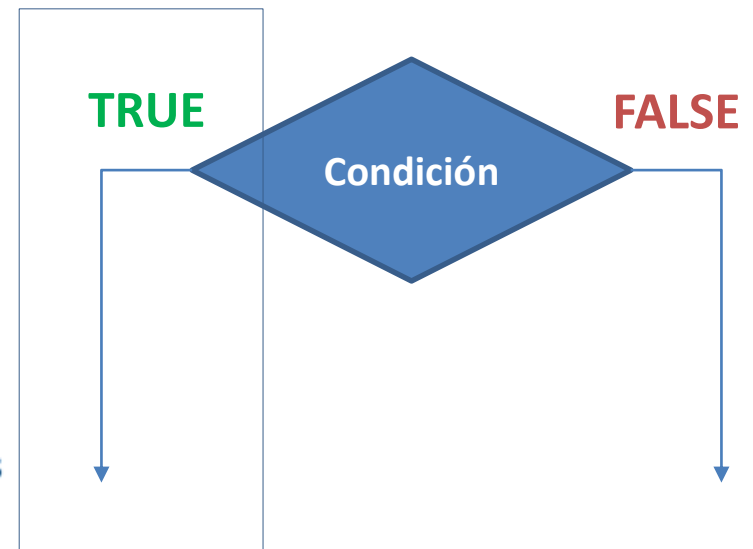
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Estructuras Condicionales (IF)

- Ej: VOTO ELECTRÓNICO

```
int edad = 18;  
  
//Es mayor de edad.  
if (edad >= 18) {  
    System.out.println("¡Adelante!, puede votar.");  
} else {  
    System.out.println("Lo siento, no puede votar.");  
}
```

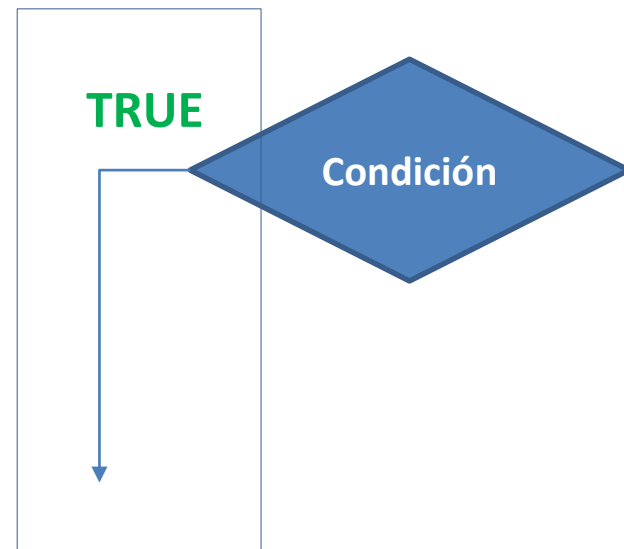


¡Adelante!, puede votar.

Estructuras Condicionales (IF)

- Ej: VOTO ELECTRÓNICO

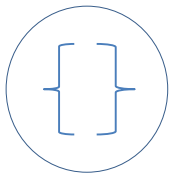
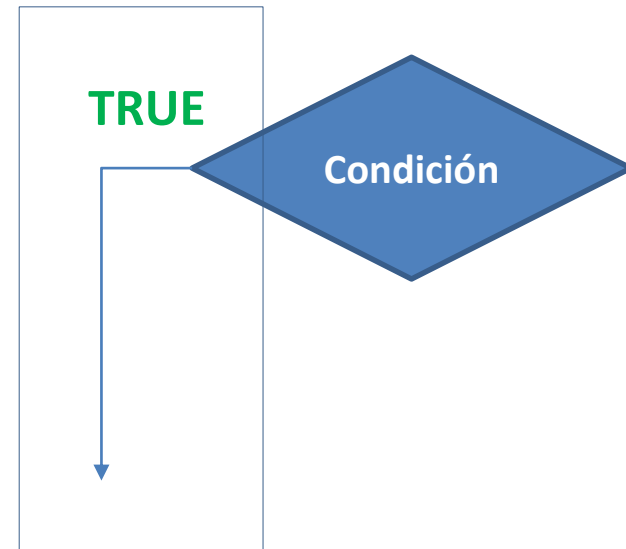
```
int edad = 18;  
  
//Es mayor de edad.  
if (edad >= 18) {  
    System.out.println("¡Adelante!, puede votar.");  
}
```



Estructuras Condicionales (IF)

- Ej: VOTO ELECTRÓNICO

```
int edad = 18;  
  
//Es mayor de edad.  
if (edad >= 18)  
    System.out.println("¡Adelante!, puede votar.");
```



Los {} son opcionales aunque muy recomendables.



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



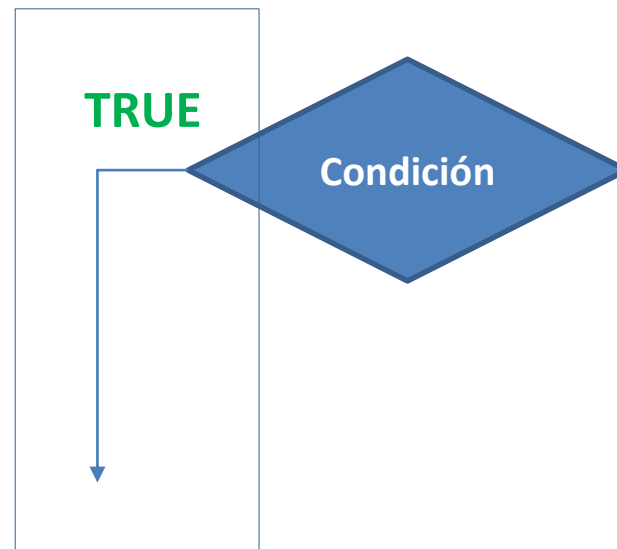
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Estructuras Condicionales (IF)

- Ej: VOTO ELECTRÓNICO

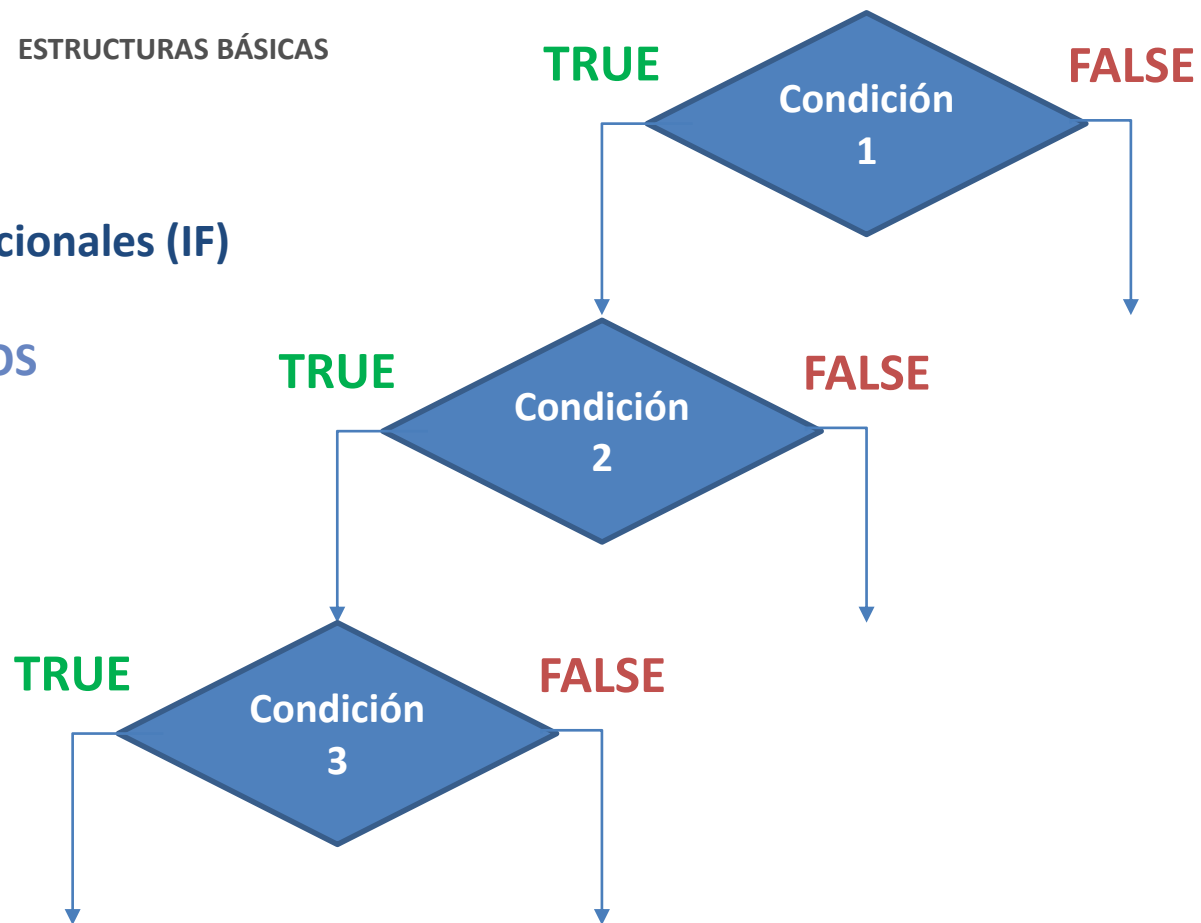
```
int edad = 18;  
  
//Es mayor de edad.  
if (edad >= 18)  
    System.out.println("¡Adelante!");  
    System.out.println("puede votar.");
```



¿Cuál es la salida de este código?

Estructuras Condicionales (IF)

- IF ANIDADOS



Estructuras Condicionales (IF)

- IF ANIDADOS

```
boolean tieneGafas = true;
boolean tieneBarba = false;

if (tieneGafas && tieneBarba) {
    System.out.println("¡Es Juan!");
} else {
    if (tieneGafas) {
        System.out.println("¡Es Laura!");
    } else {
        if (tieneBarba) {
            System.out.println("¡Es Pedro!");
        } else {
            System.out.println("¡Es Raquel!");
        }
    }
}
```



Juan



Laura



Raquel



Pedro

Estructuras Condicionales (IF)

- IF ANIDADOS

```
boolean tieneGafas = true;
boolean tieneBarba = false;

if (tieneGafas && tieneBarba) {
    System.out.println("¡Es Juan!");
} else if (tieneGafas) {
    System.out.println("¡Es Laura!");
} else if (tieneBarba) {
    System.out.println("¡Es Pedro!");
} else {
    System.out.println("¡Es Raquel!");
}
```



Juan



Laura



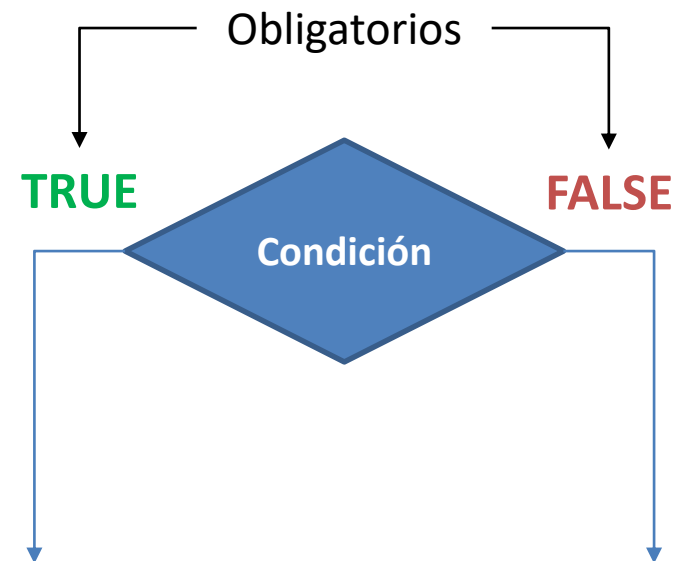
Raquel



Pedro

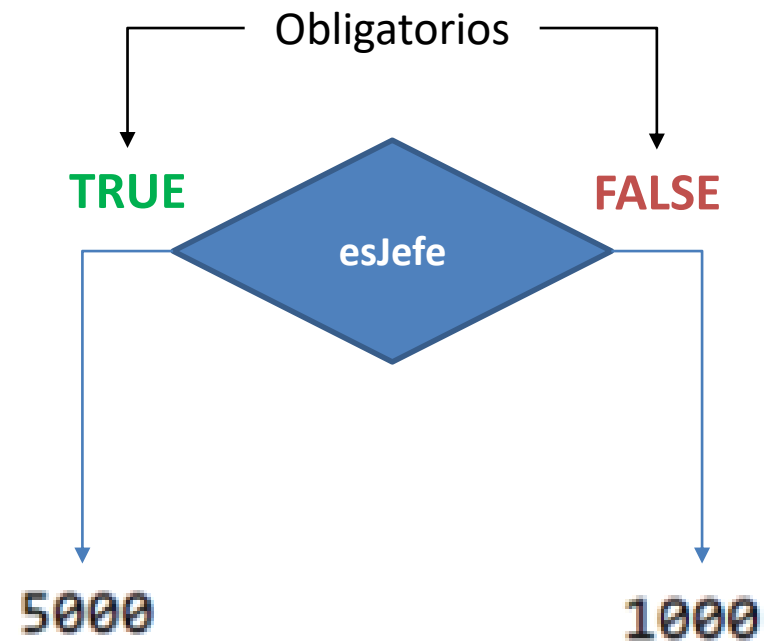
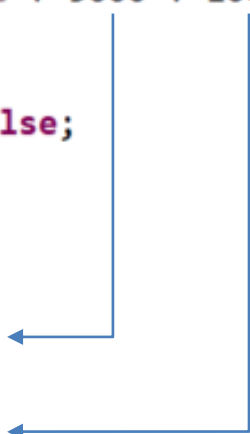
Estructuras Condicionales (OPERADOR TERNARIO)

- Es un operador (?) que toma tres argumentos.
- Funciona de manera similar a una estructura IF ELSE
- Se suele utilizar para asignar valor a una variable ya que puede resultar difícil de mantener.



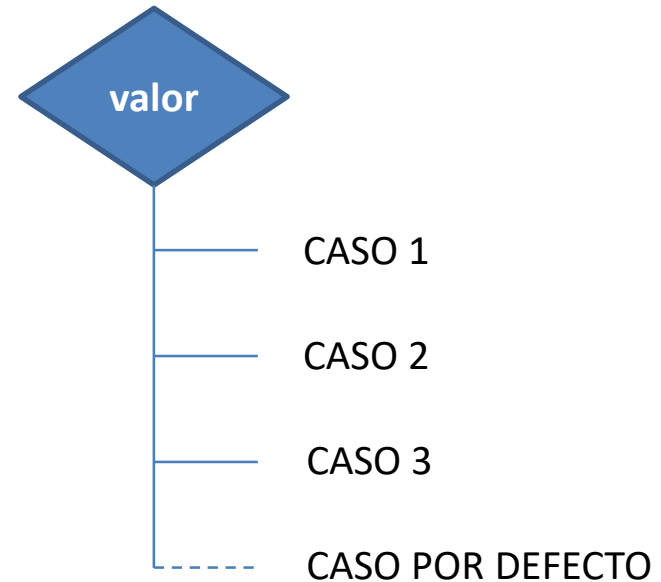
Estructuras Condicionales (OPERADOR TERNARIO)

```
boolean esJefe = false;  
  
int salario = esJefe ? 5000 : 1000;  
  
boolean esJefe = false;  
  
int salario;  
  
if (esJefe) {  
    salario = 5000;  
} else {  
    salario = 1000;  
}
```



Estructuras Condicionales (SWITCH)

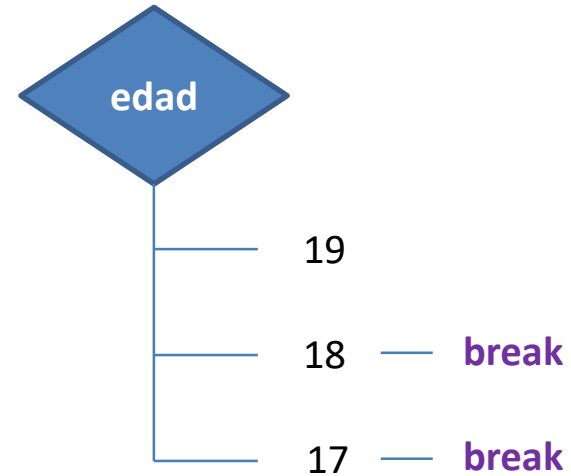
- La estructura condicional **SWITCH** es una instrucción que compara el valor de una variable con los valores de cada caso.
- Aunque haya definidos varios casos, la expresión se evalúa una vez.
- El tipo de dato puede ser tipo de dato primitivo como: byte, short, char e int.
- A partir de JDK7, también funciona con tipos enumerados (Enum), String y las clases Wrapper.



Estructuras Condicionales (SWITCH)

```
boolean mayorEdad = false;
int edad = 19;

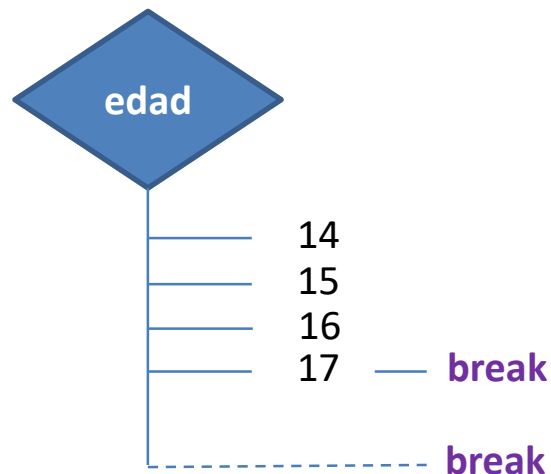
switch (edad) {
case 19:
case 18:
    // Ejecutar si la edad es 18 o 19
    mayorEdad = true;
    break;
case 17:
    // Ejecutar si la edad es 17
    mayorEdad = false;
    break;
}
```



En este caso no hay caso por defecto.
Las palabras clave **break** y **default** son opcionales

Estructuras Condicionales (SWITCH)

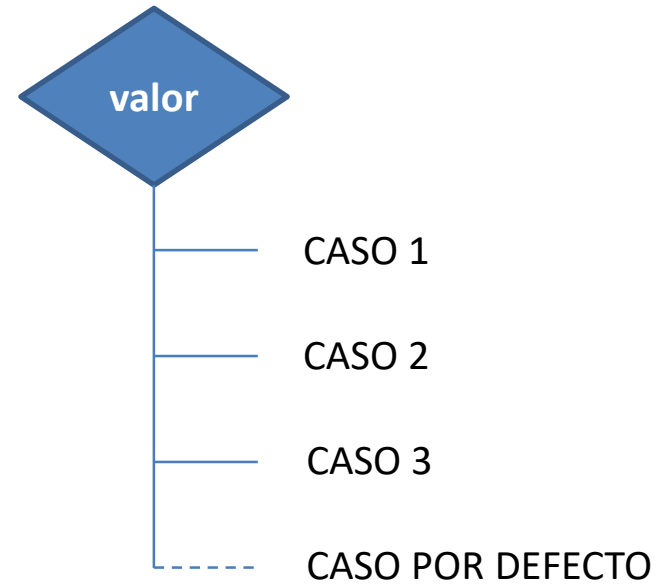
```
int edad = 19;  
  
switch (edad) {  
    case 14:  
    case 15:  
    case 16:  
    case 17:  
        System.out.println("Eres menor de edad");  
        break;  
    default:  
        System.out.println("Eres mayor de edad");  
}
```



Las palabras clave **break** y **default** son opcionales

Estructuras Condicionales (SWITCH)

- Cuando la ejecución llega a un break, esta detendrá la evaluación de más casos de pruebas.
- Cuando un caso se cumple ejecutará el código del mismo y el del resto de casos siguientes hasta encontrarse con un break que finalice el caso.



02

Estructuras Iterativas



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Estructuras Iterativas

- FOR
- FOREACH
- WHILE
- DO WHILE

Estructuras Iterativas

IF (Conozco los elementos o el número de ellos que voy a iterar)

FOR / FOREACH

ELSE IF (Se debe ejecutar al menos una vez)

DO WHILE

ELSE

WHILE

Estructuras Iterativas (FOREACH)

- Esta estructura nos permite recorrer una colección de elementos o array de elementos de una forma sencilla.

```
int[] medidas = {90,60,90};  
  
for(int medida : medidas) {  
    System.out.println(medida);  
}
```

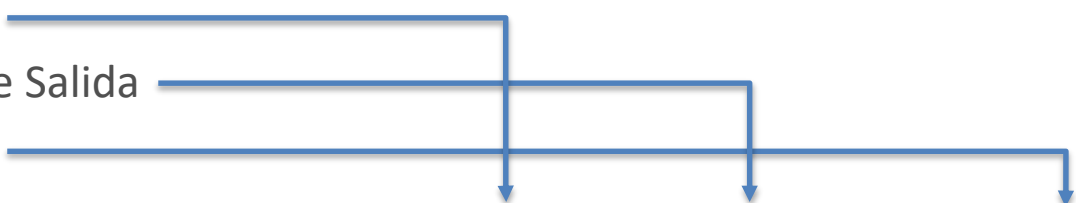
90	60	90
----	----	----

3 Elementos
=
3 Iteraciones

Estructuras Iterativas (FOR)

- Se define:

- Valor Inicial
- Condición de Salida
- Incremento



```
for(int numero = 1; numero <= 10; numero++) {  
    System.out.println(numero);  
}
```

```
for(int numero = 10; numero >= 1; numero--) {  
    System.out.println(numero);  
}
```


Estructuras Iterativas (FOR)

Se puede cambiar el incremento:

```
for(int numero = 1; numero <= 10; numero+=2) {  
    System.out.println(numero);  
}
```

Podemos utilizarlo para recorrer un array:

```
int[] medidas = {90,60,90};  
  
for(int i = 0; i < medidas.length; i++) {  
    System.out.println(medidas[i]);  
}
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Estructuras Iterativas (WHILE)

Es un bucle o sentencia repetitiva con una condición al principio. Se ejecuta una sentencia mientras sea cierta una condición. La sentencia puede que no se ejecute ni una sola vez.

```
int numVeces = 10;
while (numVeces > 0) {
    System.out.println(--numVeces);
}
```

```
int numVeces = 10;
while (numVeces > 10) {
    System.out.println(--numVeces);
}
```

Estructuras Iterativas (DO WHILE)

Es un bucle o sentencia repetitiva con una condición al final. Se ejecuta una sentencia mientras sea cierta una condición. En este caso, la sentencia se ejecuta al menos una vez.

```
int numVeces = 10;  
do {  
    System.out.println(--numVeces);  
} while (numVeces > 0);
```

Estructuras Iterativas (BUCLES INFINITOS)

La mayoría de los bucles poseen una condición de salida para terminar la ejecución pero debemos tener en cuenta que puede que la condición no deje de cumplirse nunca:

```
do {  
    System.out.println("VIRUS");  
} while (true);
```



Estructuras Iterativas (BUCLES INFINITOS)

En cualquier iteración de un bucle es posible “romper” la ejecución de las siguientes maneras:

- **break**: Termina la ejecución del bucle y continúa con la siguiente instrucción.
- **continue**: Termina la iteración actual del bucle y continúa con la siguiente.
- **return***: Sale del método actual.
- **throw***: Lanza una excepción.

* Se explicarán más adelante.

Ejercicio I

Vamos a modificar la clase TipoDatos.java para completar el mensaje que generamos por pantalla:

Hola me llamo **{Nombre} {Apellidos}** y tengo **{Edad}** años.

Nací el **{fecha de nacimiento}** y vivo en **{Ciudad}**

Actualmente **estoy/no estoy trabajando** y **estoy/no estoy estudiando**.

Otro cambio a realizar es que ahora almacenaremos información de 10 personas.

Por último mostraremos el párrafo que se indica en el ejemplo por cada una de las 10 personas.

Ejercicio II

Vamos a modificar la clase Meses.java para y ayudándonos de los operadores vistos, mostraremos los meses del array por pantalla de la siguiente forma.

- Enero es el 1º mes del año y tiene 31 días.
- Febrero es el 2º mes del año y tiene 28 o 29 días.
- ...

Ejercicio III

Vamos a crear la clase Multiplicaciones.java que imprimirá por pantalla las tablas de multiplicar, las tablas del 1 al 10:

TABLA DEL 1

01 x 01 = 01

01 x 02 = 02

...

01 x 10 = 10

TABLA DEL 2

...

Ejercicio IV

Vamos a crear la clase Cuadrado.java que imprimirá por pantalla 3 cuadrados, de la siguiente forma:

3 x 3

###

###

###

4 x 4

####

####

####

####

10 x 10

@@@@@@@@@@

@@@@@@@@@@

...