

IT Bootcamp

Estructuras de Control

DigitalHouse>

Índice

- 1. Estructuras Selectivas (Condicionales)**
- 2. Estructuras Repetitivas**
- 3. Ejemplos**

1

Estructuras Selectivas



Estructura IF

La **estructura de control “if”** permite decidir entre dos opciones resultantes de la evaluación de una sentencia.

En **Java** se implementa mediante la siguiente sintaxis:

```
Java    if (condición){  
        ...  
        ...  
    }
```



Estructura IF + ELSE

El **else** es la extensión de la sentencia IF. Significa “de lo contrario” y permite la ejecución de un bloque de código **SOLO** si la condición de la sentencia IF fue **falsa**.

En **Java** se implementa mediante la siguiente sintaxis:

```
Java    if (condición){  
        ...  
    }  
    else {  
        ...  
    }
```



Ejemplo IF + else

Java

```
int num;  
if (num>5){  
    System.out.println ("El número es mayor a 5")  
}  
else {  
    System.out.println ("El número es menor a 5")  
}
```



Estructura IF + ELSE Múltiples

Las **estructuras if y else** permiten **anidamientos**, es decir, unir varias condiciones if y else en caso de que una primera condición evaluada no se cumpla.

En **Java** se implementa mediante la siguiente sintaxis:

Java

```
if (condición){  
    ...  
}  
else {  
    if (condición){  
        ...  
    }  
    else {  
        ...  
    }  
}
```



Estructura Switch

La **estructura switch** permite múltiples caminos posibles a partir de la evaluación de una sola expresión/condición.

Cada comienzo de camino se expresa mediante la palabra reservada **case**, mientras que cada final mediante la palabra **break**.

Para tener en cuenta las condiciones que no hayan sido contempladas por los case, se puede implementar un **default**.

Java

```
int tipoResul = 3;
String mensaje;

switch (tipoResul)
{
    case 1: mensaje = "Aprobado";
            break;
    case 2: mensaje = "Desaprobado";
            break;
    case 3: mensaje = "Recuperatorio";
            break;
    default: mensaje = "Examen no dado";
            break;
}

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


2 | Estructuras Repetitivas



Estructuras Repetitivas



Permiten ejecutar un bloque de código una “n” cantidad de veces.



Cada repetición o ejecución repetitiva de un bloque de código se conoce como **bucle**.



Hay bucles:

- Controlados por un **contador**
- Controlados por un **centinela** o **bandera**



do While

Es una **estructura repetitiva** de **Java** que permite la ejecución de un bloque de código la cantidad de veces que sean necesarias dependiendo de una condición.

Tiene las siguientes características:

- La **condición** se evalúa **al final** del bloque de código.
- Como la condición se evalúa al final, se ejecuta **siempre** por lo menos **una vez**.

```
Java    do{  
        ...  
        ...  
    } while (condicion)
```



While

Es una **estructura repetitiva** de **Java** que permite la ejecución de un bloque de código **mientras** se cumpla una determinada condición.

Tiene las siguientes características:

- La **condición** se evalúa **al principio** del bloque de código.
- Como la condición se evalúa al principio, se ejecuta **solo** si la condición **se cumple**.

```
Java    while (condicion){  
        ...  
        ...  
    }
```



For

El **for** se considera una **estructura controlada por contador** por excelencia en Java. Su principal característica es que posee una **variable contadora propia** que se encarga de controlar la cantidad de repeticiones que tendrá un bloque de código.

Está compuesto por **3 partes**:

Indica el valor en el que se inicializa el contador. Se ejecuta solo una vez.

Indica en cuánto incrementará el contador en cada ejecución del bucle.

Java

```
for (inicializacion; condicion; incremento)
```

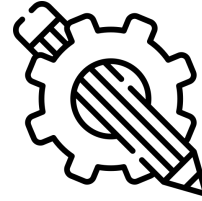
Indica la condición que se tiene que cumplir para que se detenga el bucle.



3 | Ejemplos



Ejemplos



```
//do While
int cont = 0;
do {
    ...
    cont = cont + 1;
} while (cont<=10)
```

```
//while
int cont = 0;
while (cont<=10) {
    ...
    cont = cont + 1;
}
```

```
//for
for (int c=0; c<=10; c++) {
    ...
}
```



Material extra

Material extra /bibliografía complementaria



Links de lectura y videos de interés

Lectura: [Do While Loop in Java](#)

Lectura: [While Loop in Java](#)

Video: [Estructura repetitiva While en Java](#)

Lectura: [For Loop in Java](#)

Video: [Estructura For en Java](#)

DigitalHouse>