PROGRAMACIÓN

UT3: Estructuras básicas de control (II)

1. Estructuras de repetición. FOR

Las estructuras de repetición se utilizan cuando queremos ejecutar la misma instrucción (línea/s de código) múltiples veces.

Por ejemplo, si queremos actualizar una tabla, o si queremos crear 20 alumnos, coches, ruedas, …

Aunque muchas veces son intercambiables entre sí, debemos tener en cuenta ciertos aspectos para elegir la mejor opción en cada ocasión.

En el caso de que sepamos de antemano cuántas veces se va a repetir la instrucción, elegiremos la estructura FOR.

La estructura en Java es la siguiente:

```
for (<variable_control>;<condición_parada>;<incremento_variable>)
{
…
}
```

Y en código real quedaría de la siguiente forma.

Necesitamos una variable *contador* que nos indique cuantas veces se va a ejecutar el bucle.

Dentro del propio **for** se va a crear la variable de control, en este caso se llamará **auxiliar**, que irá contando el número de vueltas.

Finalmente, con **auxiliar++** iremos añadiendo una vuelta a cada ejecución.

```
int contador = 5;
for (int auxiliar = 0; auxiliar < contador; auxiliar++)
{
     // Código del bucle
}
```

En este caso, el bucle se ejecutará 5 veces, y se irá evaluando la condición hasta que esta sea falta.

EJECUCIÓN 1

Contador = 5

Auxiliar = 0

¿auxilar < contador? (0 < 5) VERDADERO

- ➔ Se ejecuta el bucle
- ➔ Auxiliar++ (auxiliar = 1)

EJECUCIÓN 2

Contador = 5

Auxiliar = 1

¿auxilar < contador? (1 < 5) VERDADERO

- ➔ Se ejecuta el bucle
- ➔ Auxiliar++ (auxiliar = 2)

…

EJECUCIÓN 5

Contador = 5

Auxiliar = 4

¿auxilar < contador? (4 < 5) VERDADERO

- ➔ Se ejecuta el bucle
- ➔ Auxiliar++ (auxiliar = 5)

EJECUCIÓN 6

Contador = 5

Auxiliar = 5

¿auxilar < contador? (5 < 5) **FALSO**

- ➔ NO se ejecuta el bucle
- ➔ Se continua con el código debajo del **for**

NOTAS:

En condición podemos poner cualquier pregunta, es decir, <, >, <=, >=, ==, !=, …

En incremento podemos poner cualquier incremento o decremento, es decir, ++, --, +1, -2, +5, *2, …

2. Estructuras de repetición. WHILE

Con la estructura WHILE, las instrucciones dentro de ella se ejecutarán repetidamente mientras la condición impuesta al principio sea cierta.

ESTA CONDICIÓN SE EVALUA ANTES DE ENRTAR, es decir, si no se cumple la primera vez, el bucle no se ejecutará.

La estructura en Java es la siguiente:

while (<condición>)

{

…

}

Y en código real quedaría de la siguiente forma.

```
boolean condicion = true;
while (condicion){

    // Codigo del bucle

    condicion = false;
}


int contador = 1;
while (contador < 5){

    // Codigo del bucle

    contador++;
}
```

En ambos casos, es necesario cambiar la condición o aumentar (o disminuir) el contador dentro del bucle, ya que a diferencia del **for** en este caso no se hace por defecto, ¡SINO PROVOCAREMOS UN BUCLE INFINITO!

3. Estructuras de repetición. REPEAT (DO-WHILE)

Con la estructura REPEAT (do – while), las instrucciones dentro de ella se ejecutarán repetidamente mientras la condición impuesta al final sea cierta.

ESTA CONDICIÓN SE EVALUA AL FINAL DEL BUCLE, es decir, aunque no se cumpla por defecto, el bucle se ejecutará al menos 1 vez.

La estructura en Java es la siguiente:

do {

…

}

while (<condición>) ;

Y en código Java quedaría:

```
int contador = 1;
do {
    // Código del bucle
    contador++;
}
while(contador < 5);
```

Al igual que en el bucle while, es necesario cambiar la condición o aumentar (o disminuir) el contador dentro del bucle, ya que a diferencia del **for** en este caso no se hace por defecto, ¡SINO PROVOCAREMOS UN BUCLE INFINITO!