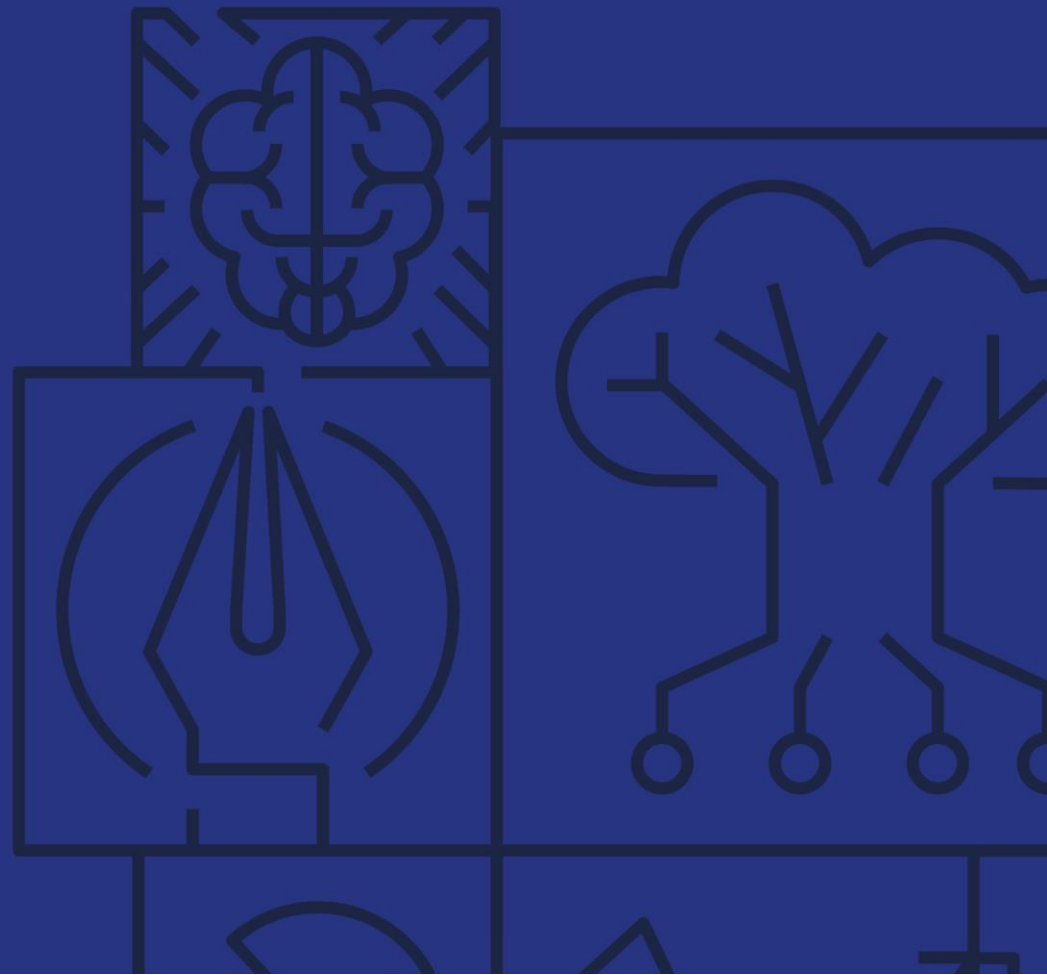
GG20210524

fidélitas
Universidad

SC-202 Introducción a la Programación

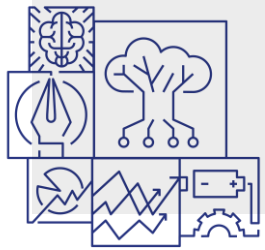


Semana 3

Estructuras de Repetición

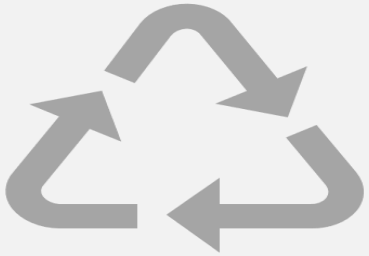
Concepto de estructura de repetición

Una estructura de repetición o ciclo es una estructura de control que permite repetir una o varias sentencias un determinado número de veces, mientras una condición se cumpla.



Ciclos en Java

For



While

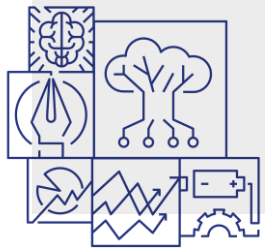
Do while

fidélitas Universidad

Ciclo **for**

El ciclo **for** permite ejecutar un bloque de instrucciones una determinada cantidad de veces. Se requiere:

- Conocer el número de iteraciones que realizará el ciclo, puede usarse una variable cuyo valor sea proporcionado por el usuario.
- Una variable de control.
- Indicar el incremento o decremento de la variable en cada iteración.



Sintaxis

```
for (inicialización; expresión_booleana; actualización){
        bloque de código;
}
```

La inicialización es una parte de la sentencia que asigna por única vez un valor inicial a la variable de control.

La expresión booleana se evalúa en cada inicio del ciclo.

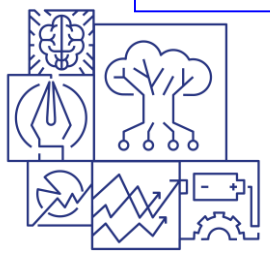
La actualización se ejecuta antes de realizar una nueva prueba de la expresión.

Ciclo for

- El código se repite utilizando un contador en el ciclo para controlar la cantidad de iteraciones.

Inicialización | Condicipon | Actualización

```
for (int i = 0; i < 5; i++){
    System.out.println("I vale: "+i);
}
```



```
I vale: 0
I vale: 1
I vale: 2
I vale: 3
I vale: 4
```

Ejecución

Ejemplo01

```
for (int i=0; i<5; i++) {
    System.out.println("El valor de i es: "+i);
}
```

```
El valor de i es: 0
El valor de i es: 1
El valor de i es: 2
El valor de i es: 3
El valor de i es: 4
```

Ejecución

Ejemplo02

```
for (int i=10; i>0; i--) {
   System.out.println(i);
}
System.out.println("Feliz año nuevo!!!");
```

```
10
9
8
7
6
5
4
3
2
1
Feliz año nuevo!!!
```

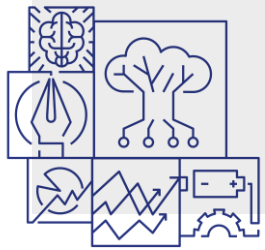
Ejecución

Ciclo **while**

El ciclo **while** permite repetir un bloque de instrucciones mientras una expresión se mantenga verdadera.

Se puede utilizar cuando se conoce el número de veces que itera el ciclo y también cuando se desconoce esa información.

Nota: Debe incluirse dentro del ciclo una sentencia que convierta en falsa la expresión lógica de la condición.



Sintaxis

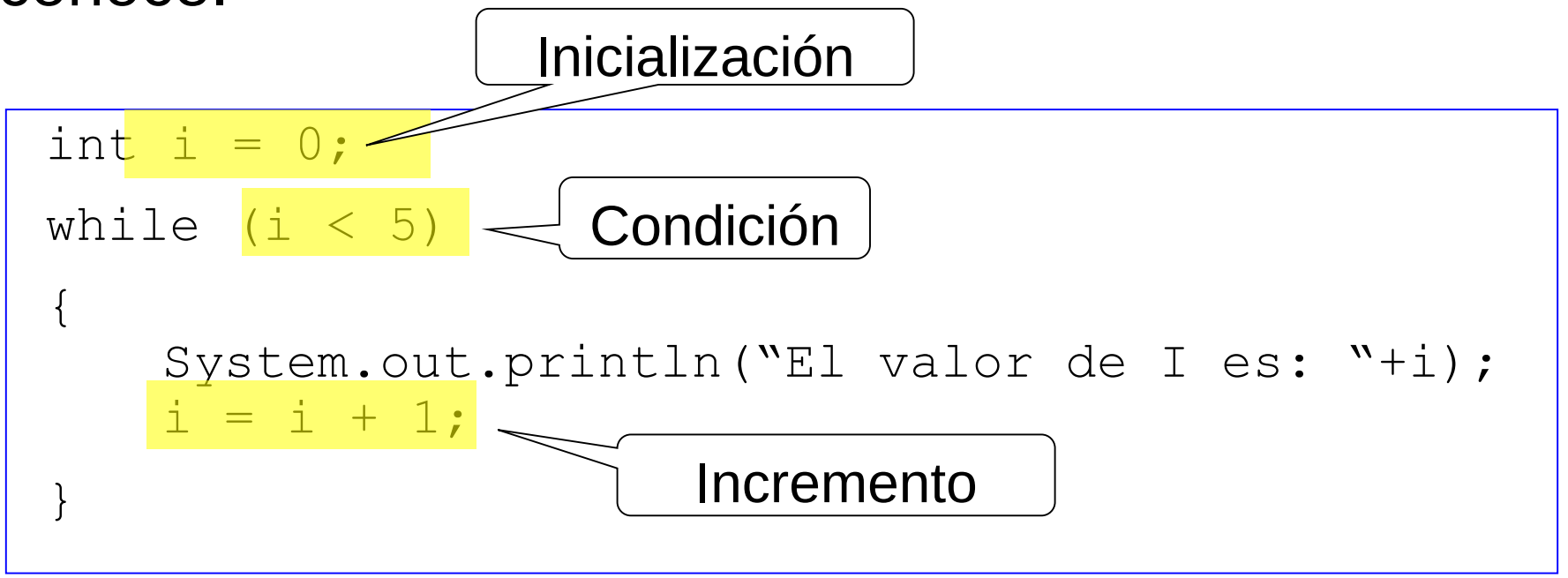
```
while (exp) {
    bloque de código;
}
```

El ciclo while tiene los siguientes componentes:

- exp es una expresión que se verifica falso o verdadero antes de cada iteración del ciclo.
- El bloque de código representa las instrucciones que se ejecutan cada vez que se itera el ciclo.
- Como se indicó anteriormente, dentro del bloque de código, debe incluirse alguna instrucción que permita que la condición (exp) se deje de cumplir en algún momento.

Ciclo while

- El código se repite mientras la condición sea verdadera.
- La condición es evaluada antes de ejecutar el código.
- Puede ser utilizada cuando la cantidad de iteraciones no se conoce.



Ejemplo03

Sumatoria:

```
int i=1, suma=0;
while (i<=10) {
   suma +=i;
   ++i;
}
System.out.println("La sumatoria es: "+suma);
```

El ciclo while se ejecuta de 1 a 10 veces, la expresión booleana, parte del ciclo, se evalúa antes de ingresar al ciclo y si esta es falsa, no se ejecuta el cuerpo del while.

Ejercicio 01

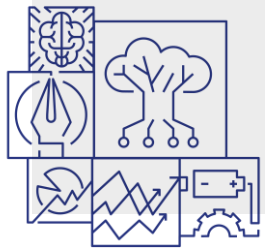
Desarrolle un nuevo proyecto que imprima un rectángulo de 10 símbolos “@” en 5 líneas.

20 minutos

@@@@@@@@@@
@@@@@@@@@@
@@@@@@@@@@
@@@@@@@@@@
@@@@@@@@@@

Ciclo do while

El ciclo do / while es utilizado para una a muchas iteraciones, la condición se encuentra al final del cuerpo del ciclo lo que provoca que al menos se ejecute una vez las instrucciones contenidas en el cuerpo del ciclo.



Sintaxis

```
do {
        bloque de código;
} while (exp);
```

La expresión booleana se evalúa al final de cada iteración. El ciclo do while siempre repetirá al menos una vez las sentencias.

Al igual que en el ciclo while, dentro del bloque de código, debe incluirse alguna instrucción que permita que la condición (exp) se deje de cumplir en algún momento.

Ejemplo04

```
int i=0;
   do{
      System.out.println("Hola, i vale: “+i);
     ++i;
   } while (i<5);
```

```
Hola, i vale 0
Hola, i vale 1
Hola, i vale 2
Hola, i vale 3
Hola, i vale 4
```

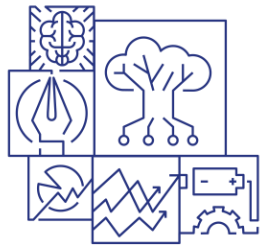
Ejecución

Ejemplo05: Contando de 1 a 10 con las tres estructuras de repetición (ejemplos)

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

```
int i = 1;
while (i <= 10){
   System.out.println("while "+i);
   i++;
}
```

```
i=1;
do {
  System.out.println("do "+i)
   i++;
} while (i <= 10);
```

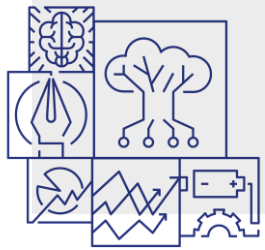


fidélitas Universidad

Error de ámbito del ciclo for

```
for(int i = 0; i < 5; i++) {
    System.out.println(i);
}
System.out.println(i);
```

En este fragmento de código se produce un error, pues al estar declarada la variable en el mismo ciclo, no se puede utilizar fuera de él.



Ciclos Anidados

Podemos anidar ciclos para funciones complejas. Debemos seguir las siguientes reglas:

1. Crear contadores individuales para cada ciclo (i, j, k)
2. Cada ciclo debe contener la estructura completa de ciclos, a saber:
   a. Declaración e inicialización de contador (int i=0;)
   b. Evaluar la expresión booleana (i<valor)
   c. Actualización del contador, incremento o decremento(i--, ++i)
3. Cuando se termina el ciclo interno debe inicializar el contador interno, generar el código del ciclo externo y actualizar el contador externo.

Ejemplo06

```
int i=0;
while (i<7){
  int j=0;
   System.out.print(“Ejecutando ciclo externo i=”+i);
  while (j<4){
    System.out.printf(“Ejecutando ciclo interno i= “+i+”j=“+j);
    j++;
  }
  i++;
}
```

Ejercicio 02

Cree un nuevo proyecto que imprima un rectángulo de 10 símbolos en 5 líneas, pero solo puede imprimir una @ a la vez.

20 minutos

```
@@@@@@@@@@
@@@@@@@@@@
@@@@@@@@@@
@@@@@@@@@@
@@@@@@@@@@
```

Ejercicio 03

Desarrolle un nuevo proyecto que le solicite al usuario un número y muestre la tabla de multiplicar de dicho número.

20 minutos

- 5 X 1 = 1
- 5 X 2 = 10
- 5 X 3 = 15
- 5 X 4 = 20
- 5 X 5 = 25
- 5 X 6 = 30
- 5 X 7 = 35
- 5 X 8 = 40
- 5 X 9 = 45
- 5 X 10 = 50

fidélitas Universidad

Ejercicio 04

Desarrolle un programa que le solicite al usuario un valor y dibuje un triángulo con ese número de filas, por ejemplo si el número es 5.

20 minutos

```
*
**
***
****
*****
```

Ejercicio 05

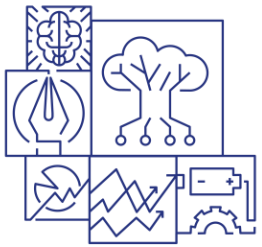
Desarrolle un programa que le muestre al usuario los valores entre 20 y 30 (incluidos) y su correspondiente valor al cuadrado.

20 minutos

Ejercicio 06

Desarrolle un programa que le solicite al usuario una cantidad de estudiantes y posteriormente que le solicite la nota para cada uno de ellos. Además debe mostrar al final la nota promedio, la nota mayor, la nota menor y la cantidad de aprobados. Se aprueba al menos con 70.

30 minutos

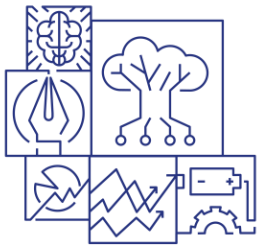


fidélitas Universidad

Ejercicio 07

Modifique el ejercicio anterior para que no se tenga que definir la cantidad de estudiantes, sino que, cuando la nota ingresada sea negativa, se debe de solicitar información.

30 minutos



fidélitas Universidad

¡Muchas gracias !

fidélitas
Universidad