



2022.

Grupo PATC

Tema a tratar:

Tipos de Ciclos

CURN

Integrantes:

Angel David Ariz Lambraño
Luis Miguel Uparela Cuellar
Juan David Fuentes Miranda
Juan David Duarte Mendoza
David Joé Patiño Leal.

Introducción.

Somos estudiantes de la Corporación Universitaria Rafael Nuñez, por medio de este trabajo estaremos dando a conocer nuestro conocimiento sobre los tipos de ciclos, de acuerdo a nuestro conocimiento, todo lo investigado y a todo lo aprendido. Estarán viendo los concepto de cada uno de los tipos de ciclos, anexados de un ejemplo y un video explicativo el cual estará de guía para un mejor aprendizaje.

Tabla de Contenido Preparación del Documento

| | |
|----------------------|---|
| Ciclo For..... | 3 |
| Ciclo While..... | 4 |
| Ciclo Do While | 6 |

Tipos de Ciclos

En los ciclos existen diferentes tipos los cuales tienen un uso específico, y depende de cada uno de nuestros conocimientos y destrezas saber cual utilizar para cada situación dada.

Tenemos los siguientes tipos de ciclos en java :

Ciclo For (JAVA).

Ciclo Do While (JAVA).

Ciclo While (JAVA).

Ciclo For.

Son estructuras cíclicas, es decir, que se repiten cada cierto tiempo, y que nos permiten ejecutar una o varias líneas de código de manera repetitiva , pero teniendo conocimiento sobre las iteraciones. Es necesario tener un valor inicial y uno final. También podemos hacer uso del tamaño de los pasos entre cada iteración del ciclo.

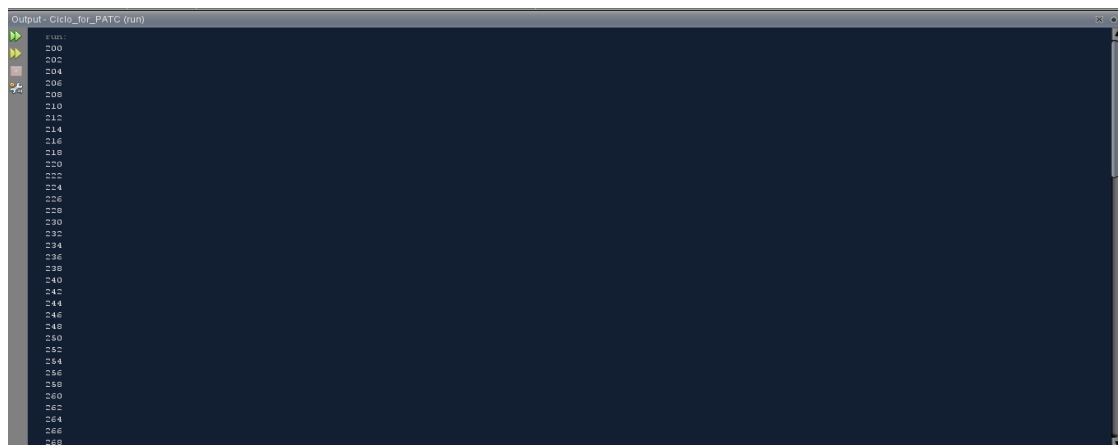
Aquí veremos a continuación la sintaxis de un ciclo for:

```
for(int i=Valor inicial; i<=Valor final; i+=paso){  
instrucciones  
}
```

Con esta sintaxis podríamos hacer un ejemplo sencillo que sería. Vamos a suponer que queremos mostrar los números pares que existen entre 200 y 400, en este caso el Valor inicial sería 200 y el Valor final sería 400, y el tamaño de paso que es 2(pares).

```
package ciclo_for_patc;  
  
public class Ciclo_for_PATC {  
  
    public static void main(String[] args) {  
  
        for(int i=200; i<=400; i+=2){  
            System.out.println(i);  
        }  
  
    }  
  
}
```

Y lo que nos debe imprimir este programa son los números pares entre 200 y 400 como se muestra en su ejecución.



```
Output - Ciclo_for_PATC (run)  
200  
202  
204  
206  
208  
210  
212  
214  
216  
218  
220  
222  
224  
226  
228  
230  
232  
234  
236  
238  
240  
242  
244  
246  
248  
250  
252  
254  
256  
258  
260  
262  
264  
266  
268
```

Ciclo While.

Al igual que el ciclo for, este ciclo tiene una estructura cíclica, que nos permite ejecutar una o varias líneas de código, en este no es necesario tener un Valor inicial, Incluso a veces no se conoce cuando se va a dar el valor final, estos ciclos no tienen una dependencia total de valores numéricos, sino de booleanos, es decir, depende se si una condición es verdadera o falsa. Este ciclo es más efectivo con condiciones indeterminadas.

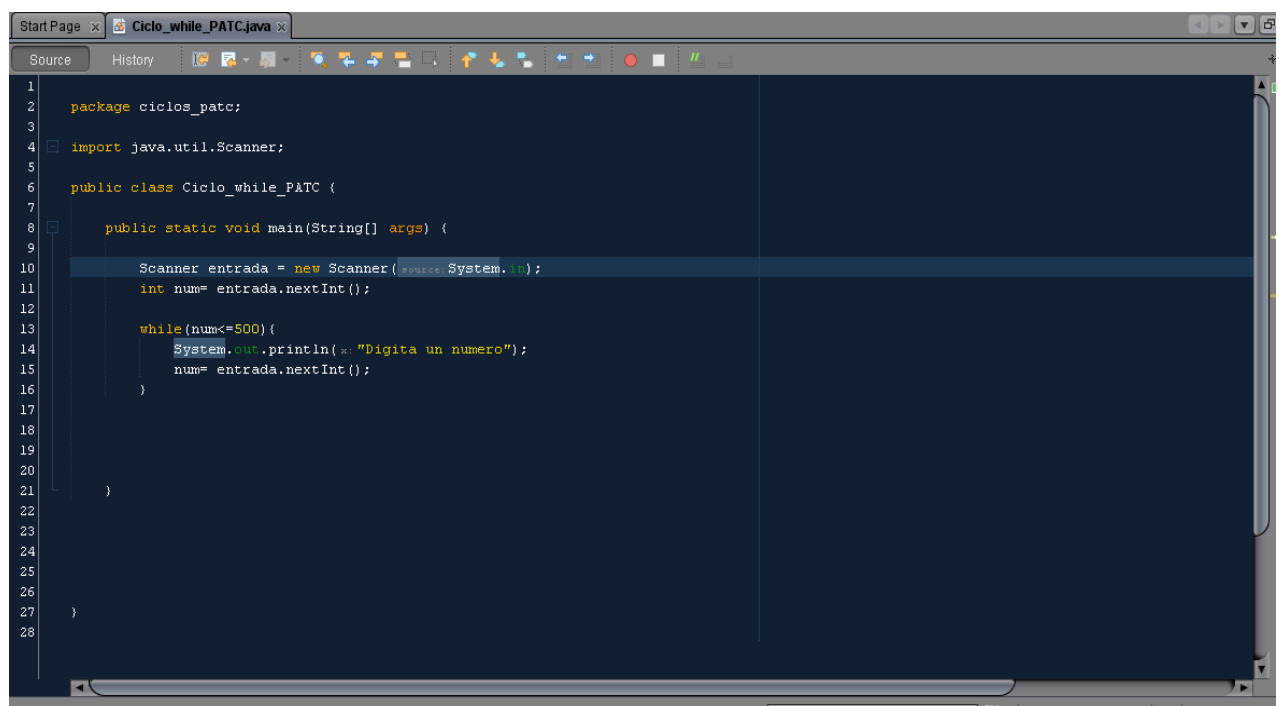
Pongamos un ejemplo de el ciclo while.

Si queremos que queramos pedirle a un usuario una serie de números y que termine de hacerlo cuando el USUARIO digite un número mayor a 300. este programa se hace con este ciclo porque es algo indeterminado y el ciclo while nos permite ejecutar una acción de manera ("Infinita") hasta que se cumpla una condición específica, que en este ejemplo sería que digite un número mayor a 300.

A continuación está la sintaxis de un ciclo While.

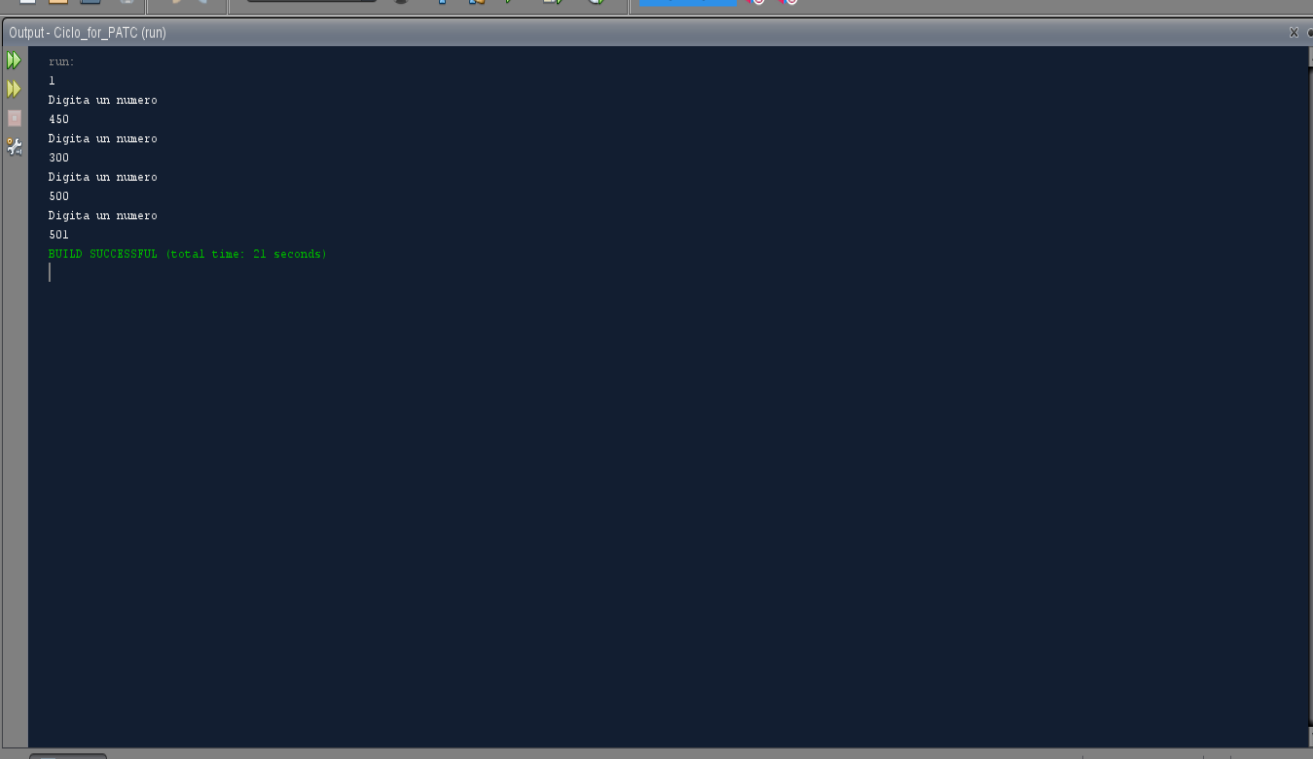
```
while(condición de finalización ){  
  
    instrucciones  
  
}
```

Dada la sintaxis vamos a realizar el programa



```
1 package ciclos_patc;  
2  
3  
4 import java.util.Scanner;  
5  
6 public class Ciclo_while_PATC {  
7  
8     public static void main(String[] args) {  
9  
10         Scanner entrada = new Scanner(System.in);  
11         int num= entrada.nextInt();  
12  
13         while(num<=500) {  
14             System.out.println("Digita un numero");  
15             num= entrada.nextInt();  
16         }  
17  
18  
19  
20  
21     }  
22  
23  
24  
25  
26  
27 }  
28
```

Y la ejecución nos muestra que podemos ingresar los números que sean menores o iguales a 500 pero desde que se ingresa un número mayor a 500 se cierra el programa.



```
run:
1
Digita un numero
450
Digita un numero
300
Digita un numero
500
Digita un numero
501
BUILD SUCCESSFUL (total time: 21 seconds)
```

Ciclo Do-While.

El ciclo do while en su definición es semejante al ciclo while ya que al igual que el anterior este ciclo tiene una estructura cíclica, que nos permite ejecutar una o varias líneas de código, en este no es necesario tener un Valor inicial, Incluso a veces no se conoce cuando se va a dar el valor final. El ciclo do while nos permite ejecutar primero el bloque de instrucciones, antes de pasar a evaluar la condición necesaria, son muy efectivos para

algunas situaciones específicas, en conclusión este ciclo evalúa la condición después de haber ejecutado el bloque de instrucciones.

Para dar un ejemplo de esto también podríamos utilizar la situación del ciclo while.

Si queremos que queramos pedirle a un usuario una serie de números y que termine de hacerlo cuando el USUARIO digite un número mayor a 300. Este programa se hace con este ciclo porque es algo indeterminado y el ciclo while nos permite ejecutar una acción de manera ("Infinita") hasta que se cumpla una condición específica, que en este ejemplo sería que digite un número mayor a 300.

Aquí estaremos viendo la sintaxis de un ciclo do-while.

```
do{
```

```
    instrucciones
```

```
}while(condición de finalización).
```

Luego de ver la sintaxis vamos a la realización del programa.

```
package javaapplication1;

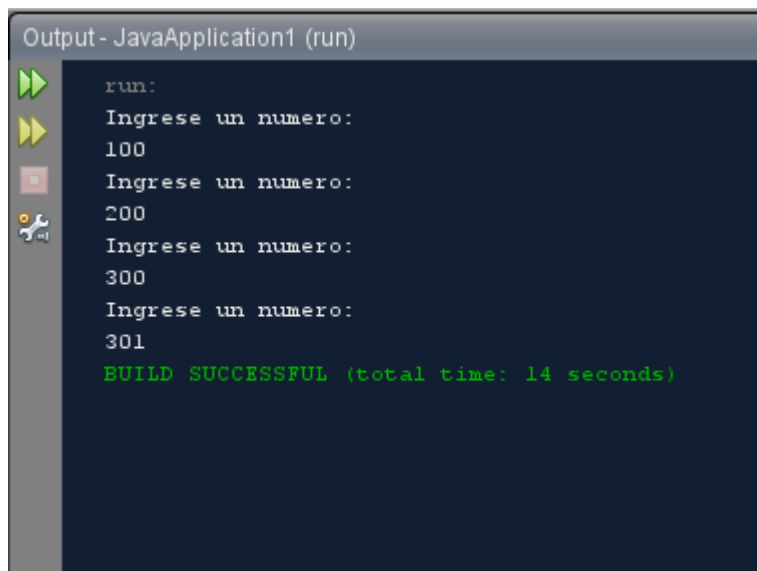
import java.util.Scanner;

public class JavaApplication1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;

        do
        {
            System.out.println("Ingrese un numero: ");
            num = sc.nextInt();
        }
        while(num <= 300);
    }
}
```

Y luego vamos a la ejecución.



```
Output - JavaApplication1 (run)

run:
Ingrese un numero:
100
Ingrese un numero:
200
Ingrese un numero:
300
Ingrese un numero:
301
BUILD SUCCESSFUL (total time: 14 seconds)
```

Cuando se pasa el número del límite que es 300 se finaliza el programa.

