

	<p style="text-align: center;"><b>UNIVERSIDAD DON BOSCO</b>  <b>FACULTAD DE INGENIERÍA</b>  <b>ESCUELA DE COMPUTACIÓN</b>  <b>TÉCNICO EN DESARROLLO DE APLICACIONES MÓVILES</b></p>
<p style="text-align: center;"><b>Ciclo I</b></p>	<p style="text-align: center;"><b>Programación Orientada a Objetos</b>  <b>Guía de Laboratorio No. 4</b>  <b>Estructuras repetitivas (iterativas o bucles)</b></p>

## I. OBJETIVOS.

- Introducir al alumno en el desarrollo de aplicaciones en JAVA.
- Que el alumno conozca los principios básicos de la programación estructurada a partir de los siguientes conceptos:
  - Estructuras repetitivas
    - Bucle While
    - Bucle do... While
    - Bucle for

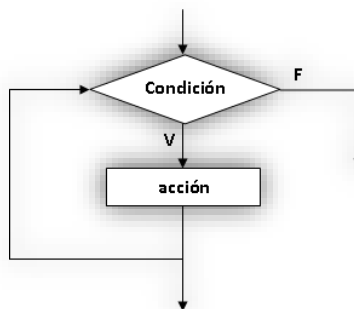
## II. INTRODUCCIÓN.

### Estructuras repetitivas

También conocidas como estructuras iterativas o bucles, permiten ejecutar un conjunto de instrucciones (cuerpo del bucle) mientras se cumpla una determinada condición; cada repetición de secuencias de instrucciones se conoce como iteración.

### Estructura de repetición WHILE:

Esta estructura primero verifica la condición y luego ejecuta la acción. La acción puede ser una acción simple o una acción compuesta ( bloque de instrucciones encerradas entre llaves ).



SINTAXIS:

Para una sola acción:

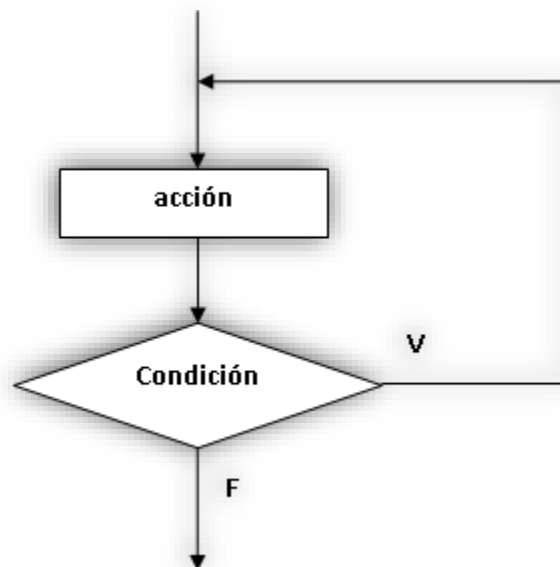
```
while ( condición )  
    acción;
```

Para varias acciones:

```
while ( condición ) {  
    acción 1;  
    acción 2;  
    .  
    .  
    .  
    acción n;  
}
```

### Estructura de repetición DO WHILE:

Este tipo de estructura primero ejecuta la acción y luego verifica la condición. La acción puede ser simple o compuesta.



### SINTAXIS:

Para una sola acción:

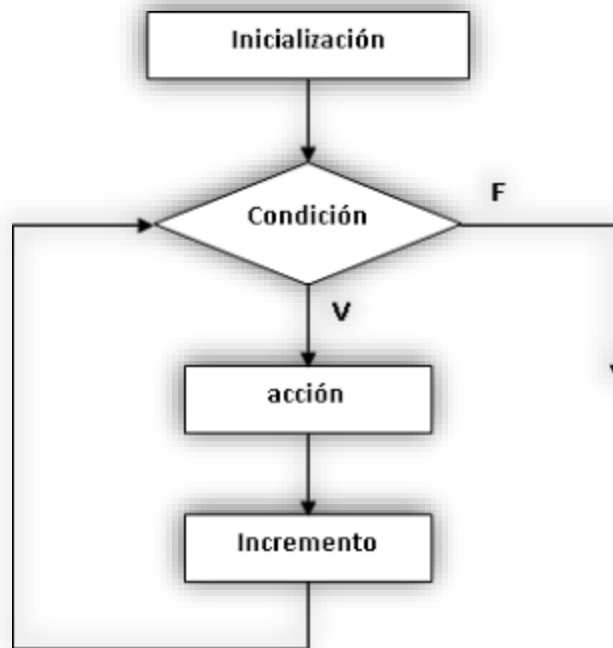
```
do
    acción;
while ( condición );
```

Para varias acciones:

```
do {
    acción 1;
    acción 2;
    .
    .
    .
    acción n;
} while ( condición );
```

### Estructura de repetición FOR:

Este tipo de estructuras incluye un contador como parte de su estructura, lo cual, quiere decir que se conoce el número de veces que se repetirá el cuerpo del bucle.



SINTAXIS:

Para una sola acción:

```
for ( inicio; condición; incremento )  
    acción;
```

Para varias acciones:

```
for ( inicio; condición; incremento ) {  
    acción 1;  
    acción 2;  
    .  
    .  
    acción n;  
}
```

**NOTA:** El contador del for puede estar declarado dentro de él mismo, en este caso el contador se considera como variable local al for, siendo inaccesible fuera de él. Por lo tanto, dos o más for no anidados, pueden declarar contadores con el mismo nombre.

### III. Material y equipo necesario.

Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Requerimiento	Cantidad
1	Guía de Laboratorio #3 de POO	1
2	Computadora con Eclipse IDE	1

### III. Procedimiento.

Crear el proyecto Guia3

Ejemplo 1:

**Estructura repetitiva do...while**

```
import java.util.Scanner;
```

```

public class Ejemplo1 {

    public static void main(String args[]) {
        double a=0;
        double b=0;

        String continuar="y";

        String op;

        Scanner reader=new Scanner(System.in);

        do{    //Estructura repetitiva do while.
              //se iniciará el bucle y
              //se repetirá si la condición dentro del while
              //es verdadera

            System.out.print("Ingrese el primer valor: ");
            a=reader.nextInt();
            System.out.print("Ingrese el segundo valor: ");
            b=reader.nextInt();
            System.out.print("Ingrese la operacion: ");

            op = reader.next();

            System.out.print("El resultado es: ");

            switch (op) {
                case "+":
                    System.out.println(a + b);
                    break;
                case "-":
                    System.out.println(a - b);
                    break;
                case "*":
                    System.out.println(a * b);
                    break;
                case "/":
                    System.out.println(a / b);
                    break;
                default:
                    System.out.println("error");
                    break;
            }
        }
    }
}

```

```

        System.out.print("Continuar?... (y/n): ");

        continuar=reader.next();

    } while( continuar.equals("y")); //Fin del do...while

}

```

## Ejemplo 2 Estructura repetitiva while:

```

import java.util.Scanner;

public class Ejemplo2 {

    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);

        String continuar = "y";

        int contador = 0;

        while (continuar.equals("y")) {

            contador++;

            System.out.println("Repeticion #: " + contador);

            System.out.print(" Continuar?... (y/n): ");

            continuar = reader.next();

        }

    }

}

```

## Ejemplo 3:

### Uso de do...while con contador

---

```

public class Ejemplo3 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int contador=10;

        System.out.println("Cuenta regresiva");

        do{
            System.out.println("Numero: " + contador);

            contador--;
        } while(contador>=0);

    }

}

```

#### Ejemplo 4:

##### Uso de while con contador

```

public class Ejemplo4 {

    public static void main(String[] args) {
        int contador=0;
        while(contador<=10){

            System.out.println("Numero: " + contador);
            contador++;

        }

    }

}

```

#### Ejemplo 5:

##### Uso de for

```

public class Ejemplo5 {

    public static void main(String[] args) {
        //sumar los primeros 10 números enteros
        int suma=0;

        for(int i=1; i<=10; i++){
            suma=suma+i;
        }

        System.out.println("La suma de los primeros números enteros es: "+suma);
    }

}

```

**Ejemplo 6:**

**Otro ejemplo de for:**

```

public class Ejemplo6 {

    public static void main(String [] args){
        //suma de numeros pares hasta 20
        int suma=0;
        for(int i=0; i<=20; i +=2 ){
            suma=suma+i;
        }

        System.out.println("La suma de los primeros 20 números enteros pares es:
"+suma);

    }

}

```

**Ejemplo 7:**

**Calcular factorial con ciclo for:**

```

import java.util.Scanner;

public class Ejemplo7 {

    //calculo de numero factorial

```



```

public static void main(String [] args){
    Scanner reader= new Scanner(System.in);
    int factorial=1;

    System.out.println("Ingresa el valor al que se le calculará el factorial: ");
    int contador=reader.nextInt();

    if( contador==0 ){
        //no se hace nada, 0!=1 el valor de factorial para 0 es 1
    }else{

        for( int i=contador; i>0; i-- ){
            factorial= factorial*i;
        }

        System.out.println("El número factorial es: " + factorial);
    }
}

```

### Ejemplo 8: for anidados

```

public class Ejemplo08 {
    public static void main(String[] args) {

        for(int i=0; i <= 8; i++){

            for(int j=0; j<=8; j++){
                if(i != 3 && i!=4 && i!=5 ){
                    if((j<=2 || j>=6)){
                        if(j==8){
                            System.out.print(" \n");
                        }else{
                            System.out.print(" ");
                        }
                    }else{
                        System.out.print("*");
                    }
                }
            }
        }
    }
}

```

Ejemplo 09:

#### IV. EJERCICIOS COMPLEMENTARIOS.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377...

La sucesión comienza con los números 1 y 1, y a partir de estos, «cada término es la suma de los dos anteriores», es la relación de recurrencia que la define.

2) Se solicita a usuario que ingrese un número entero positivo, y luego se le indicará si el número ingresado es o no PRIMO. Si no es primo, muéstrele por qué no es primo. Recuerde que un número entero es primo cuando es divisible únicamente por el mismo y por la unidad.

Ejemplos de números primos: 1, 5, 13, 37, etc. porque cualquiera de ellos, es divisible por uno (1) y el mismo.

Ejemplo de número no primo: 42, porque además de ser divisible por 42 y por 1, también es divisible por 2, divisible por 3, divisible por 6, divisible por 7, divisible por 14, y divisible por 21.

3) Realice un programa que permita hacer deducciones sobre un sueldo neto. Las deducciones que se aplican son isss (3%), afp (6.25%), renta (7.1%).

El resultado debe aparecer como se muestra a continuación.

Sueldo: 450

Renta: 31.95

AFP: 28.125

Isss: 13.5

Total a pagar: 376.425

El programa debe realizar los cálculos que el usuario desee. Para ello utilice la estructura repetitiva do...while, y debe preguntar si desea continuar.

4) Crear una aplicación que permita encontrar los divisores de un número ingresado en consola por el usuario.

Debe utilizar el ciclo repetitivo for.

## **VI. REFERENCIA BIBLIOGRAFICA.**

<http://inprojava.blogspot.com/2010/12/estructuras-repetitivas.html>

Java : Cómo Programar ./ Paul J. Deitel y Harvey M. Deitel Pauld Deitel  
MEXICO, MEXICO : PEARSON, 2012  
9a. ed. 2012