

# TALLER DE LENGUAJE DE PROGRAMACIÓN I

## Entorno de CONSOLA y GRÁFICOS

### UNIDAD 1

#### Arreglos – Matrices - ArrayList

JAVA

# OBJETIVO

- Control del Flujo de programas
  - Condicionales
    - If
    - switch
  - Bucles
    - Indeterminados
      - While
      - Do – while
    - Determinados
      - For
      - For - each
- Arreglos
- Matrices
- ArrayList

Trabajo Independiente

- Taller

# Control del flujo de programas

## Condicional flujo **if** y **switch**

```
import java.util.Scanner;
public class condicionalif
{
    public static void main(String[] args)
    {
        // Condicional if
        final int valor_techo = 100;
        Scanner dato = new Scanner(System.in);
        int num1 = 0;
        int num2 = 0;
        int operacion = 0;

        System.out.println("Digite el primer valor");
        num1 = dato.nextInt();

        System.out.println("Digite el segundo valor");
        num2 = dato.nextInt();

        operacion = num1 + num2;
        if(operacion > valor_techo)
        {
            System.out.print("La suma de los valores " + operacion + " superan
            el valor techo." + valor_techo);
        }
        else
        {
            System.out.print("La suma de los valores ingresados es: " +
            operacion);
        }
        dato.close();
    }
}
```

```
import javax.swing.*;
public class condicionalif2
{
    public static void main(String[] args)
    {
        final int valor_techo = 100;

        int num1 = Integer.parseInt(JOptionPane.showInputDialog("Digite
        al primer número"));
        int num2 = Integer.parseInt(JOptionPane.showInputDialog("Digite
        el segundo valor"));
        int operacion = num1 + num2;

        String mensaje = "";

        if(operacion > valor_techo)
        {
            mensaje = "La suma de los valores superan el valor techo." +
            valor_techo;
        }
        else
        {
            mensaje = "La suma de los valores ingresados es: " +
            operacion;
        }
        JOptionPane.showMessageDialog(null, mensaje);
    }
}
```

# Control del flujo de programas, continuación

## Condicional flujo if y **switch**

```
import javax.swing.JOptionPane;
public class condicionalSwitch
{
    public static void main(String[] args)
    {
        int estrato = Integer.parseInt(JOptionPane.showInputDialog("Especifique su Estrato"));
        switch (estrato)
        {
            case 1:
                JOptionPane.showMessageDialog(null, "El Estrato " + estrato + " es veneficiado con 30%");
                break;
            case 2:
                JOptionPane.showMessageDialog(null, "El Estrato " + estrato + " es veneficiado CON 20%");
                break;
            default:
                JOptionPane.showMessageDialog(null, "El Estrato debe ser entre 1 y 6");
        }
    }
}
```

# Control del flujo de programas, continuación

## Condicional Bucles Indeterminados **while** y **Do while**

```
import javax.swing.*;
public class condicionalWhile {
    public static void main(String[] args) {
        String clave = "josebotero";
        String pass = "";
        int sw = 0;
        while (sw == 0) {
            pass = JOptionPane.showInputDialog("Digite su clave");
            if(clave.equals(pass) == false) {
                if(pass == null){
                    JOptionPane.showInternalMessageDialog(null, "Cancelo!,
                    vuelva.");
                    sw = 1;
                }
                else {
                    JOptionPane.showInternalMessageDialog(null, "Clave
                    incorrecta.");
                }
            }
            else {
                JOptionPane.showInternalMessageDialog(null,
                "Despliegue
                del sistema.");
                sw = 1;
            }
        }
    }
}
```

```
import javax.swing.*;
public class condicionalDoWhile {
    public static void main(String[] args) {
        int suma = 0;
        int num = 0;
        int veces;
        int ciclos = 0;
        veces = Integer.parseInt(JOptionPane.showInputDialog("Cantidad
        de Dígitos"));
        do {
            num = Integer.parseInt(JOptionPane.showInputDialog("Digite el
            Número"));
            suma = suma + num;
            ciclos++;
        }while(ciclos < veces);
        JOptionPane.showMessageDialog(null, "Resultado: " + suma);
    }
}
```

# Control del flujo de programas, continuación

## Condicional Bucles **for** y **for each**

```
import javax.swing.JOptionPane;

public class condicionalfor {

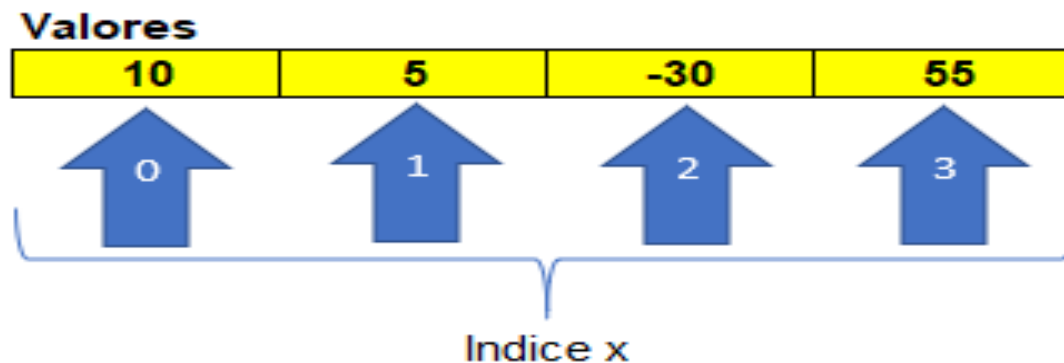
    public static void main(String[] args) {

        String cadena="";
        for (int i=0; i < 5; i++) {
            cadena += JOptionPane.showInputDialog("Digitar
                dato " + (i+1)) + " - ";
        }
        JOptionPane.showMessageDialog(null, cadena);
    }
}
```

La instrucción de bucle (ciclos) **for...each**, se explicará en el tema de arreglos, ya que fue definido específicamente para este propósito.

# Arreglo Unidimensional su Iteración

## Arreglo Unidimensional



# Arreglos y la Iteración

## Arreglos y la Iteración

```
public class arreglos {  
    public static void main(String[] args) {  
        // Manejo de Arreglos
```

```
        int[] arreglo = new int[5];  
        int arreglo2[] = new int[5];  
        int arreglo3[] = {10, 15, 20, 25, 30};
```

Equivalente para la dimensión

```
        arreglo[0] = 10;  
        arreglo[1] = 15;  
        arreglo[2] = 20;  
        arreglo[3] = 25;  
        arreglo[4] = 30;
```

Equivalente para asignación de valor

```
        System.out.println("Posición 1 --> " + arreglo[0]);  
        System.out.println("Posición 2 --> " + arreglo[1]);  
        System.out.println("Posición 3 --> " + arreglo[2]);  
        System.out.println("Posición 4 --> " + arreglo[3]);  
        System.out.println("Posición 5 --> " + arreglo[4]);
```

```
    }  
}
```



# Arreglos y la Iteración

## Arreglos y la Iteración

```
public class arreglosfor
{
    public static void main(String[] args)
    {
        // Manejo de Arreglos

        int[] arreglo = new int [5];

        for (int i=0; i < arreglo.length; i++) {
            arreglo[i] = (int)(Math.random()*100);
            System.out.println(arreglo[i]);
        }
    }
}
```

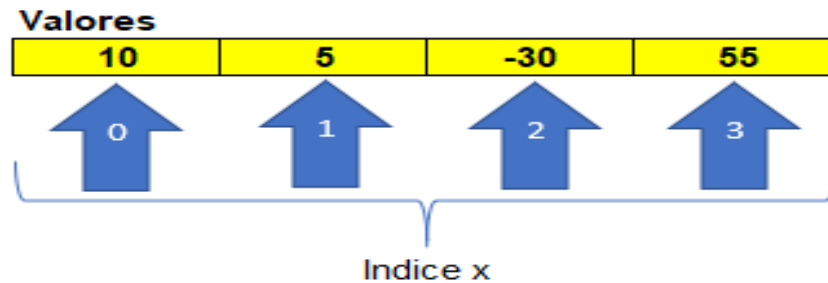
```
public class arreglosForEach
{
    public static void main(String[] args)
    {
        // Manejo de Arreglos

        int[] arreglo = new int [5];

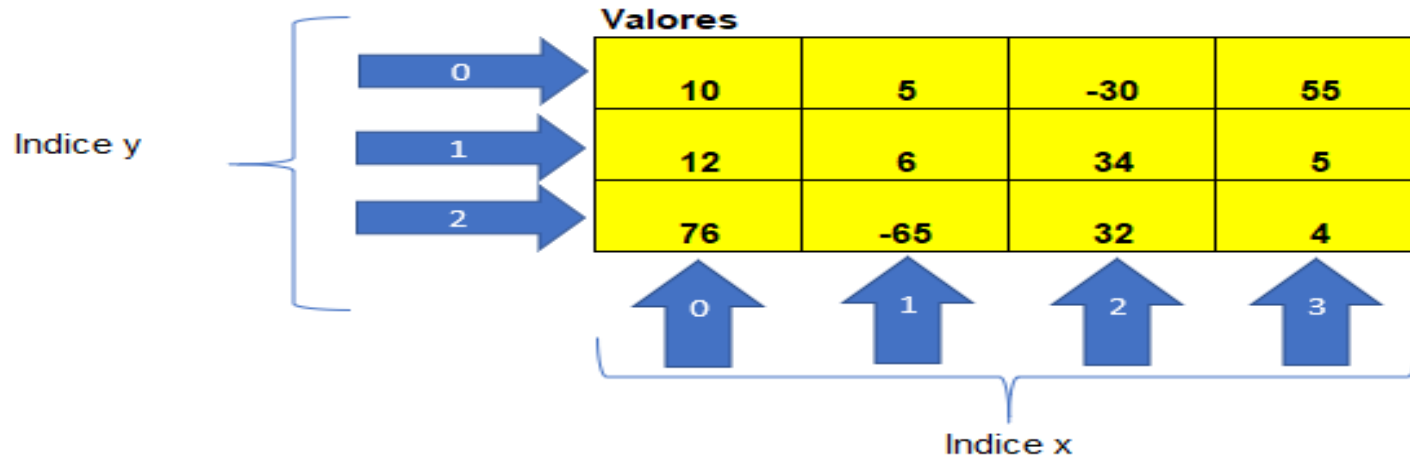
        for (int i=0; i< arreglo.length; i++)
        {
            arreglo[i] = (int)(Math.random()*100);
            System.out.println(arreglo[i]);
        }
        System.out.println("-----");
        for(int i : arreglo)
        {
            System.out.println(i);
        }
    }
}
```

# Arreglo Bidimensional o Matriz y su Iteración

## Arreglo Unidimensional



## Arreglo Bidimensional o Matriz



# Arreglo Bidimensional o Matriz y su Iteración

Definir y recorrer un Arreglo Bidimensional con for

```
public class matriz
{
    public static void main(String[] args)
    {
        // Manejo de Arreglos Bidimensionales
        int[][] matriz = new int[3][4];

        System.out.println(matriz.length);

        for (int i=0;i < matriz.length;i++)
        {
            System.out.println(" ");
            for(int j=0;j < matriz[i].length; j++ )
            {
                System.out.print(matriz[i][j] + " ");
            }
        }
    }
}
```

# Arreglo Bidimensional o Matriz y su Iteración

## Definir y recorre un Arreglo Bidimensional con For ... Each

```
public class matrizForEach
{
    public static void main(String[] args)
    {
        // Manejo de Arreglos Bidimensionales

        int[][] matriz = new int[3][4];

        for (int i=0;i < matriz.length;i++)
        {
            for(int j=0;j < matriz[i].length; j++ )
            {
                matriz[i][j] = (int)(Math.random()*100);
            }
        }

        for (int[] rec_matriz_fila : matriz)
        {
            System.out.println(" ");
            for (int rec_matriz_columna : rec_matriz_fila)
            {
                System.out.print(rec_matriz_columna + " ");
            }
        }
    }
}
```

# ArrayList y su Iteración

La clase ArrayList permite almacenar datos en memoria de forma similar a los Arrays con la ventaja de que el numero de elementos que almacena lo hace de forma dinámica.

- **NO** es necesario declarar su tamaño
- Se puede añadir, eliminar y modificar elementos

```
public class matrizForEach
{
    public static void main(String[] args)
    {
        //Declaración de u ArrayList
        ArrayList<String> listaNombres = new ArrayList<String>();

        //Adicionar un item al ArrayList
        listaNombres.add("Valentina");
        listaNombres.add("Cristina");

        Iterator<String> apuntadorListaNombres = listaNombres.iterator();
        int wciclo = 1;
        while(apuntadorListaNombres.hasNext())
        {
            String itemApuntador = apuntadorListaNombres.next();
            System.out.print("Almacenamiento en el item " + wciclo + " - " + itemApuntador);
            wciclo++;
        }
    }
}
```

# ArrayList y su Iteracción

Algunos de los métodos del ArrayList.

- `// Añade el elemento al ArrayList en la posición 'n'`  
`listaNombres.add(n, "Elemento 2");`
- `// Devuelve el numero de elementos del ArrayList`  
`listaNombres.size();`
- `// Devuelve el elemento que esta en la posición '2' del ArrayList`  
`listaNombres.get(2);`
- `// Comprueba se existe el elemento ('Elemento') que se le pasa como parametro`  
`listaNombres.contains("Elemento");`
- `// Devuelve la posición de la primera ocurrencia ('Elemento') en el ArrayList`  
`listaNombres.indexOf("Elemento");`
- `// Devuelve la posición de la última ocurrencia ('Elemento') en el ArrayList`  
`listaNombres.lastIndexOf("Elemento");`
- `// Borra el elemento de la posición '1' del ArrayList`  
`listaNombres.remove(5);`
- `// Borra todos los elementos de ArrayList`  
`listaNombres.clear();`
- `// Devuelve True si el ArrayList esta vacio. Sino Devuelve False`  
`listaNombres.isEmpty();`
- `// Copiar un ArrayList`  
`ArrayList arrayListCopia = (ArrayList) listaNombres.clone();`
- `// Pasa el ArrayList a un Array`  
`Object[] array = listaNombres.toArray();`

# Trabajo Independiente

- TALLER Mdelamiento.