

MANUAL JAVA



Contenido

1)	Mi primer programa	4
2)	Navegar en símbolo de sistema	4
3)	Errores de programación más comunes	4
4)	Tipos de datos	5
5)	Operadores aritméticos	6
6)	Condicionales if-else.....	7
7)	Estructuras condicionales anidadadas	8
8)	Introducir datos desde el teclado en un programa.....	9
9)	Condiciones compuestas con operadores lógicos y relacionales 23/10/2022	10
10)	Ejercicio con Condiciones compuestas	11
11)	Condicion con Switch-Case	14
12)	Bucle o Ciclo For (por)	15
13)	Bucle o Ciclo con while (mientras)	15
14)	Bucle: Estructura repetitiva do-while.....	16
15)	Ejercicio con for, while y do while.....	16
16)	Cadenas de caracteres	21
17)	Manipulación de cadenas con los métodos length() & substring()	24
18)	Arreglos unidimensionales (vectores estaticos)	26
19)	Arreglos unidimensionales (vectores dinámicos)	27
20)	Ejercicio vectores dinámicos	28
21)	Arreglos bidimensionales (matrices).....	29
22)	Arreglos bidimensionales (matrices dinamicas).....	31
23)	Interfaces graficas (librería swing)	33
24)	Interfaces gráficas manipulación (swing - JFrame)	35
25)	Interfaces graficas labels	37

1) Mi primer programa

```
public class Hola{  
    public static void main(String args[]){  
        System.out.println("Hola mundo");  
    }  
}
```

Instrucciones de guardado:

- Guardar primero en formato de block de notas
- Después en formato **.java**

2) Navegar en símbolo de sistema

Indicacion	Accion
dir	Ver contenido de carpeta
cd <i>nombre de carpeta</i>	Abrir carpeta
Javac <i>nombre del archivo.java</i>	Crear clase java
Java <i>nombre del archivo</i>	Correr programa

3) Errores de programación más comunes

1. No colocar la primera letra de la clase en mayúscula
2. Que no pongas el mismo nombre al archivo y a la clase
3. No colocar ; después de una operación
4. No definir correctamente las variables
5. No escribir bien el nombre del archivo o acción en el símbolo del sistema
6. No abrir o cerrar llaves

4) Tipos de datos

Primitivos

Enteros

- byte (-128 hasta 127)
- short (-32,768 hasta 32,767)
- int (-2,147,483,648 hasta 2,147,483,647)
- long (muy grande...)

Decimales

- float
- double

Otros

- char (un solo caracter)
- boolean (true | false)

Tipos Objeto

String (cadenas de texto)

Ejemplo:

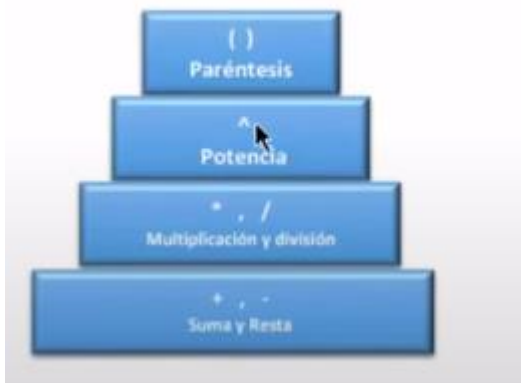
```
public class Suma{  
    public static void main(String args[]){  
        int numero_uno = 5;  
        int numero_dos = 2;  
        int resultado = 0;  
        resultado = numero_uno + numero_dos;  
        System.out.println("El valor de tu suma es:" + resultado);  
    }  
}
```

5) Operadores aritméticos

Operadores aritméticos en programación

SIGNO	NOMBRE DEL SIGNO	TIPO DE OPERACIÓN	CODIGO ASCII
+	Más	Suma	ALT + 43
-	Menos	Resta	ALT + 45
*	Asterisco	Multiplicación	ALT + 42
/	Diagonal o Slash	División	ALT + 47
^	Acento circunflejo	Potencia	ALT + 94
(Paréntesis apertura	Prioridad entre operaciones	ALT + 40
)	Paréntesis cierre	Prioridad entre operaciones	ALT + 41

Prioridad entre operadores



```
public class Operacion{
    public static void main (String args[]){
        int numero_uno = 4 ;
        int numero_dos = 4;
        int resultado = 0;
        resultado = numero_uno + numero_dos / 2;
        System.out.println("El resultado incorrecto es : " + resultado);
        resultado = (numero_uno + numero_dos) / 2;
        System.out.println("El resultado es : " + resultado);

    }
}
```

6) Condicionales if-else

```
public class Condicion{
    public static void main(String args[]){
        int mate=5;
        int bio=8;
        int quim=7;
        int prome=0;
        prome= (mate+bio+quim)/3;
        if(promedio >=6) {
            System.out.println(" El alumno aprobo "+ prome);
        }
        else{ // se debe poner para indicar cuando no se cumple la condición o que pasara si no se cumple
            System.out.println(" El alumno rebrobo "+ prome);
        }
    }
}
```

7) Estructuras condicionales anidadadas

```
public class Condicion2{
    public static void main(String arg[]){
        int op=5;
        int num_uno=8;
        int num_dos=4;
        int res=0;
        if(op==1){
            res=num_uno+num_dos;
            System.out.println("El resultado de la suma es: " + res);
        }
        else if(op==2){ //se debe colocar if else para indicar que sucederá si no se cumple la condición anteri
            res=num_uno-num_dos;
            System.out.println("El resultado de la resta es: " + res);
        }
        else if(op==3){
            res=num_uno*num_dos;
            System.out.println("El resultado de la multiplicacion es: " + res);
        }
        else if(op==4){
            res=num_uno/num_dos;
            System.out.println("El resultado de la division es: " + res);
        }
        else { //cerramos con else para indicar que pasa si ninguna condición se cumple
            System.out.println("la eleccion que elegiste no existe");
        }
    }
}
```


8) Introducir datos desde el teclado en un programa

```
import java.util.Scanner; //llamar librería para escanear o introducir datos

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

        Scanner entra = new Scanner(System.in); //definir variable para escanear
        String nombre = ""; //definir variable para datos
        int numUno = 0, numDos = 0 , resultado = 0; //se puede definir varias variables poniendo coma \(,\)

        System.out.println("Cual es tu nombre?");
        nombre = entra.nextLine(); //nombre de la variable de scanner + .nextLine\(\) para variable tipo String

        System.out.println("Cual es tu primer numero?");
        numUno = entra.nextInt(); //nombre de la variable de scanner + .nextInt\(\) para variable tipo int

        System.out.println("Cual es tu segundo numero?");
        numDos = entra.nextInt();

        resultado= numUno + numDos;

        System.out.println("Hola "+ nombre +" el resultado de la suma es:" + resultado);
    }
}
```

9) Condiciones compuestas con operadores lógicos y relacionales 23/10/2022

Operadores relacionales

Operador	Nombre	Ejemplo	Significado
<	Menor que	5 < 4	5 es menor que 4
>	Mayor que	A > B	A es mayor que B
==	Igual a	5 == 5	5 es igual a 5
!=	No igual a (<i>diferente</i>)	perro != gato	perro no es igual a gato
<=	Menor que o igual a	A <= B	A es menor que o igual a B
>=	Mayor que o igual a	A >= B	A es mayor que o igual a B

Operadores lógicos

Operador	Nombre	Ejemplo	Significado	ASCII
&&	AND	Haz la tarea && lava la ropa	Haz la tarea y lava la ropa	Alt + 38
	OR	Haz la tarea lava la ropa	Haz la tarea o lava la ropa	Alt + 124

10) Ejercicio con Condiciones compuestas

Práctica propuesta:

La empresa **Coca-Cola Company** solicita un sistema que determine los días de vacaciones a los que tiene derecho un trabajador, tomando en cuenta las siguientes características:

Existen tres departamentos dentro de la empresa con sus respectivas claves:

1. Departamento de Atención al cliente. **(Clave 1)**
2. Departamento de Logística. **(Clave 2)**
3. Gerencia. **(Clave 3)**

Trabajadores con **clave 1(Atención al cliente)**:

- Con 1 año de servicio, reciben 6 días de vacaciones.
- Con 2 a 6 años de servicio, reciben 14 días de vacaciones.
- A partir de 7 años de servicio, reciben 20 días de vacaciones.

Trabajadores con **clave 2(Logística)**:

- Con 1 año de servicio, reciben 7 días de vacaciones.
- Con 2 a 6 años de servicio, reciben 15 días de vacaciones.
- A partir de 7 años de servicio, reciben 22 días de vacaciones.

Trabajadores con **clave 3(Gerencia)**:

- Con 1 año de servicio, reciben 10 días de vacaciones.
- Con 2 a 6 años de servicio, reciben 20 días de vacaciones.
- A partir de 7 años de servicio, reciben 30 días de vacaciones.

Nota: El sistema debe de solicitar el “Nombre”, “Clave del departamento” y “Antigüedad” del trabajador, posteriormente mostrar un mensaje que contenga el nombre del trabajador y los días de vacaciones a los que tiene derecho.

```

import java.util.Scanner; //llamar libreria

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

        Scanner entra = new Scanner(System.in); //definir variable para escanear
        String nombre = ""; //definir variable para datos
        int anos = 0, clave =0;
        System.out.println("*****");
        System.out.println("*****");
        System.out.println("*****Bienvenido al sistema de vacaciones*****");
        System.out.println("*****");
        System.out.println("*****");

        System.out.println("Cual es el nombre del trabajador?");
        nombre = entra.nextLine(); //nombre de la variable de scanner + .nextLine\(\) para variable tipo String

        System.out.println("Clasificacion de claves de trabajadores por area");
        System.out.println("-Atencion a clientes=1");
        System.out.println("-Logistica=2");
        System.out.println("-Gerencia=3");
        System.out.println("Escribe la clave del trabajador:");
        clave = entra.nextInt(); //nombre de la variable de scanner + .nextInt\(\) para variable tipo int

        System.out.println("Cuantos son los anos de servicio");
        anos = entra.nextInt(); //nombre de la variable de scanner + .nextInt\(\) para variable tipo int

        if(clave==1){
            if(anos<=1){
                System.out.println("El trabajador" + nombre + " tendra 6 dias de vacaciones");
            } else if(anos>=2 && anos<=6){
                System.out.println("El trabajador" + nombre + " tendra 14 dias de vacaciones");
            } else if(anos>=7){
                System.out.println("El trabajador" + nombre + " tendra 20 dias de vacaciones");
            }
        }else if (clave==2){
            if(anos<=1){
                System.out.println("El trabajador" + nombre + " tendra 7 dias de vacaciones");
            } else if(anos>=2 && anos<=6){
                System.out.println("El trabajador" + nombre + " tendra 15 dias de vacaciones");
            } else if(anos>=7){
                System.out.println("El trabajador" + nombre + " tendra 22 dias de vacaciones");
            }
        }
    }
}

```

```
}  
} else if (clave==3){  
    if(anos<=1){  
        System.out.println("El trabajador" + nombre + " tendra 10 dias de vacaciones");  
    } else if(anos>=2 && anos<=6){  
        System.out.println("El trabajador" + nombre + " tendra 20 dias de vacaciones");  
    } else if(anos>=7){  
        System.out.println("El trabajador" + nombre + " tendra 30 dias de vacaciones");  
    }  
} else{  
    System.out.println("Clave incorrecta");  
}  
}  
}
```

11) Condicion con Switch-Case

```
public class Switch1{
    public static void main(String args[]){
        int num_uno=5, num_dos=3, resultado=0;
        int parametro=10;
        switch(parametro){ // estructura igual a la de if, entre parentesis va el parametro de valores de entrada ( tipo
enteros o char)
// char solo deja usar un solo dato numero o letra
            case 1: resultado =num_uno + num_dos; //pones cada caso con sus 2 puntos, instruccion y coma
                System.out.println("El resultado de la suma es: "+resultado);
                break; //no olvidar de poner break
            case 2: resultado =num_uno - num_dos;
                System.out.println("El resultado de la resta es: "+resultado);
                break;
            case 3: resultado =num_uno * num_dos;
                System.out.println("El resultado de la multiplicacion es: "+resultado);
                break;
            case 4: resultado =num_uno / num_dos;
                System.out.println("El resultado de la division es: "+resultado);
                break;
            default: System.out.println("Error, la opcion no existe");
                break; //default tambien lleva break
        }
    }
}
```

12) Bucle o Ciclo For (por)

```
public class Ciclofor{
public static void main (String args[]) {
    for(int i=1;i<=5; i++){ // primero ponemos la variable y la podemos declarar la mas usada es la letra i
                           // ponemos la condicion y al ultimo la accion que sucedera
                           // para mas 1 i++, para menos i-- , o i+=2 para aumentar de 2 en dos
    System.out.print(i + " , "); // print es para imprimir en una sola linea en horizontal, println para dar un salto
    }
}
}
```

13) Bucle o Ciclo con while (mientras)

```
public class Ciclowhile{
public static void main (String args[]) {
    int i=1; //debes declarar antes la variable a diferencia del for
    while(i<10){ //aqui decimos mientras i sea menor a 10
    System.out.print(i + " , "); // print es para imprimir en una sola linea en horizontal, println para dar un salto
    i+=2; //puedes poner la accion antes o despues dependiendo que es lo que quieras hacer
    }
}
}
```

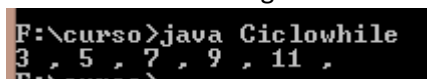
Resultado con el código anterior



```
F:\curso>java Ciclowhile
1 , 3 , 5 , 7 , 9 ,
F:\curso>
```

```
public class Ciclowhile{
public static void main (String args[]) {
    int i=1; //debes declarar antes la variable a diferencia del for
    while(i<10){ //aqui decimos mientras i sea menor a 10
    i+=2; //puedes poner la accion antes o despues dependiendo que es lo que quieras hacer
    System.out.print(i + " , "); // print es para imprimir en una sola linea en horizontal, println para dar un salto
    }
}
}
```

Resultado con el código anterior



```
F:\curso>java Ciclowhile
3 , 5 , 7 , 9 , 11 ,
F:\curso>
```

14) Bucle: Estructura repetitiva do-while

```
public class Dowhile{  
    public static void main (String args[]) {  
        int i=1000; //debes declarar antes la variable  
        do{  
            System.out.print(i + " , "); // print es para imprimir en una sola linea en horizontal, println para dar un salto  
            i-=200; //puedes poner la acción antes o después dependiendo que es lo que quieras hacer  
  
        } while(i>=0); // se repetirá a do mientras i sea mayor o igual a 0  
    }  
}
```

```
F:\curso>java Dowhile  
1000 , 800 , 600 , 400 , 200 , 0 ,
```

15) Ejercicio con for, while y do while

Realizar las siguientes series numéricas utilizando las estructuras repetitivas: **for, while y do-while.**

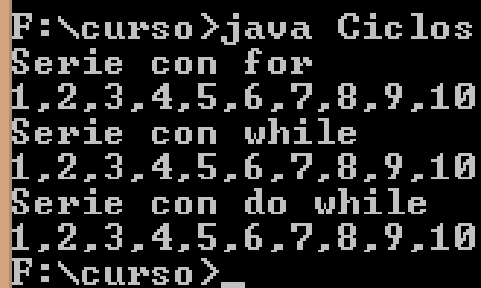
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- 1, 99, 2, 98, 3, 97, 4, 96, 5, 95
- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Primer caso


```

public class Ciclos{
    public static void main (String arg[]){
        int i=0;
        System.out.println("Serie con for");
        for (i=1;i<=10;i++){
            if(i<10){
                System.out.print(i + ",");
            } else{
                System.out.print(i);
            }
        }
        i=1;
        System.out.println("");
        System.out.println("Serie con while");
        while(i<=10){
            if(i<10){
                System.out.print(i + ",");
            } else{
                System.out.print(i);
            }
            i++;
        }
        i=1;
        System.out.println("");
        System.out.println("Serie con do while");
        do{
            if(i<10){
                System.out.print(i + ",");
            } else{
                System.out.print(i);
            }
            i++;
        }while (i<=10);
    }
}

```



```

F:\curso>java Ciclos
Serie con for
1,2,3,4,5,6,7,8,9,10
Serie con while
1,2,3,4,5,6,7,8,9,10
Serie con do while
1,2,3,4,5,6,7,8,9,10
F:\curso>_

```

Caso 2

```
public class Ciclos2{
    public static void main (String arg[]){
        int i=0;
        int j=0;
        System.out.println("Serie con for");
        for (i=1;i<=5;i++){
            if(i<=4){
                System.out.print(i + ",");
                j=100-i;
                System.out.print(j + ",");
            } else{
                System.out.print(i+",");
                System.out.print(j);
            }
        }
        i=1;
        j=0;
        System.out.println("");
        System.out.println("Serie con while");
        while(i<=5){
            if(i<=4){
                System.out.print(i + ",");
                j=100-i;
                System.out.print(j + ",");
            } else{
                System.out.print(i+",");
                System.out.print(j);
            }
            i++;
        }
        i=1;
        System.out.println("");
        System.out.println("Serie con do while");
        do{
            if(i<5){
                System.out.print(i + ",");
                j=100-i;
                System.out.print(j + ",");
            } else{
                System.out.print(i+",");
                System.out.print(j);
            }
            i++;
        }
```

```
}while (i<=5);  
}  
}
```

```
F:\curso>java Ciclos2  
Serie con for  
1,99,2,98,3,97,4,96,5,96  
Serie con while  
1,99,2,98,3,97,4,96,5,96  
Serie con do while  
1,99,2,98,3,97,4,96,5,96  
F:\curso>
```

Caso 3 serie Fibonacci

```

public class Ciclos3{
    public static void main (String arg[]){
        int a=0, b=1, c=0, i=0;

        System.out.println("Serie con for");
        for (i=0;i<10;i++){
            if(i<9){
                System.out.print(a+",");
                c=a+b;
                a=b;
                b=c;
            }else{
                System.out.print(a);
            }
        }
        a=0; b=1; c=0; i=0;
        System.out.println("");
        System.out.println("Serie con while");
        while(i<10){
            if(i<9){
                System.out.print(a+",");
                c=a+b;
                a=b;
                b=c;
            }else{
                System.out.print(a);
            }
            i++;
        }
        a=0; b=1; c=0; i=0;
        System.out.println("");
        System.out.println("Serie con do while");
        do{
            if(i<9){
                System.out.print(a+",");
                c=a+b;
                a=b;
                b=c;
            }else{
                System.out.print(a);
            }
            i++;
        }while(i<10);
    }
}

```

```
}  
}
```

```
F:\curso>java Ciclos3  
Serie con for  
0,1,1,2,3,5,8,13,21,34  
Serie con while  
0,1,1,2,3,5,8,13,21,34  
Serie con do while  
0,1,1,2,3,5,8,13,21,34  
F:\curso>
```

16) Cadenas de caracteres

String permite declarar variables para alojar cadenas de caracteres (texto)

Declarar valores

Int numero=0; [//para números](#)

String nombre=""; [//para alojar texto](#)

Comparar valores

numero==9; [//no se cumple porque arriba la declaramos como 0](#)

nombre=="Juan"; [//se puede comparar de esta forma](#)

nombre.equals("Juan"); [//o con esta forma](#)

Ejercicios:

1. Realizar un programa que compare dos nombres para verificar que son iguales, mostrando un mensaje en pantalla al final que diga: **los nombres son iguales / los nombres son diferentes.**

Nota: Los nombres debe de introducirlos el usuario desde teclado.

2. Realizar un programa que simule un inicio de sesión solicitando el nombre de usuario y la contraseña, para posteriormente comprarlos y mostrar un mensaje en pantalla que diga: **inicio de sesión correcto / nombre de usuario o password incorrectos.**

Ejercicio 1

```
import java.util.Scanner; //llamar librería para escanear o introducir datos

public class Caracteres{
    public static void main(String args[]){
        Scanner entra=new Scanner(System.in); //definimos variable para escanear
        String nombreUno="", nombreDos=""; //definimos variables donde guardaremos lo escaneado

        System.out.println("Cual es el primer nombre?");
        nombreUno = entra.nextLine();

        System.out.println("Cual es el segundo nombre?");
        nombreDos = entra.nextLine();

        if(nombreUno.equals(nombreDos)){ //equals para comparar caracteres
            System.out.println("Los dos nombres son iguales");

        }else{
            System.out.println("Los dos nombres son diferentes");

        }

    }
}
```

Ejercicio 2

```
import java.util.Scanner;

public class Caracteres2{
    public static void main(String args[]){
        Scanner entra=new Scanner(System.in);
        String nombreUno="", contra="", nombreDos="", contraGuar="";
        int i=0;

        System.out.print("Nombre de usuario:");
        nombreUno = entra.nextLine();

        System.out.print("Ingresa codigo de acceso:");
        contra = entra.nextLine();

        System.out.println("*****");
```



```

System.out.println("");
System.out.println("Usuario y/o código incorrectos");
System.out.println("intenta nuevamente");
System.out.println("*****");
System.out.println("");
i=1;
}
}
}
}
}

```

17) Manipulación de cadenas con los métodos length() & substring()

El método **length** indica cuantos caracteres posee una cadena de caracteres

Variable_tipo_int= variable_tipo_string.length();

El método **substring** permite obtener una parte en específico de la cadena de caracteres

variable_tipo_string_substraccion= variable_tipo_string_original.substring (desde,ahora);

Ejercicio:

Realizar un programa que solicite desde teclado una cadena de caracteres, posteriormente el programa debe indicar a través de un mensaje en pantalla, la cantidad de caracteres que posee dicha cadena. Finalmente el programa debe de preguntar al usuario, ¿que parte de la cadena desea obtener? tal y como se vio en el ejemplo pasado.

```

import java.util.Scanner;

public class Caracteres3{
    public static void main(String args[]){
        Scanner entra=new Scanner(System.in);
        String nombreOrig="", nombreSubs="";
        int numCarac=0, desde=0, hasta=0;

        System.out.print("Nombre de usuario:");
        nombreOrig = entra.nextLine();
        numCarac= nombreOrig.length();
        System.out.println("");
    }
}

```



```

System.out.println("La cadena tiene " + numCarac + " caracteres");

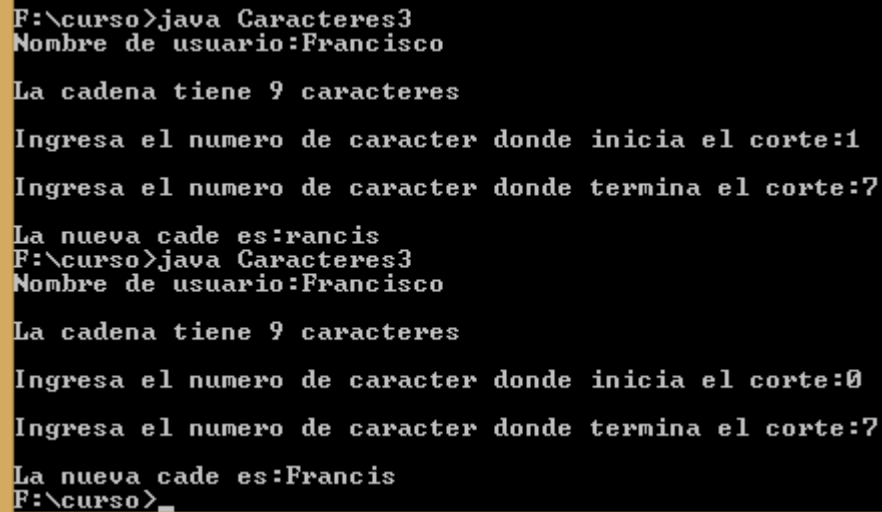
System.out.println("");
System.out.print("Ingresa el numero de caracter donde inicia el corte:");
desde= entra.nextInt();

System.out.println("");
System.out.print("Ingresa el numero de caracter donde termina el corte:");
hasta= entra.nextInt();

nombreSubs= nombreOrig.substring(desde,hasta);
System.out.println("");
System.out.print("La nueva cade es:" + nombreSubs);

}
}

```



```

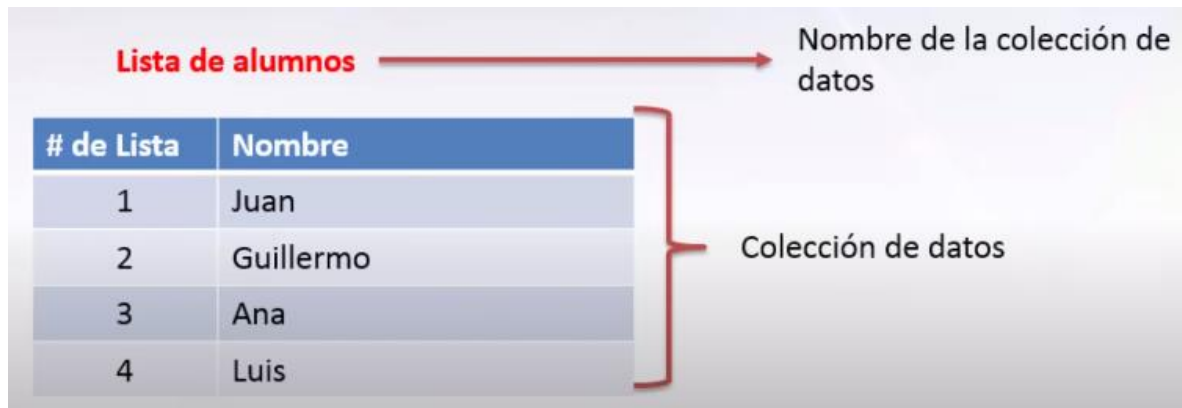
F:\curso>java Caracteres3
Nombre de usuario:Francisco
La cadena tiene 9 caracteres
Ingresa el numero de caracter donde inicia el corte:1
Ingresa el numero de caracter donde termina el corte:7
La nueva cade es:rancis
F:\curso>java Caracteres3
Nombre de usuario:Francisco
La cadena tiene 9 caracteres
Ingresa el numero de caracter donde inicia el corte:0
Ingresa el numero de caracter donde termina el corte:7
La nueva cade es:Francis
F:\curso>_

```

18) Arreglos unidimensionales (vectores estaticos)

Un arreglo es una estructura de datos que almacena bajo el mismo nombre a una colección de datos.

Int nombre[] = new int [num de posiciones]



```
public class Vectores{  
    public static void main(String args[]){  
        int numeros[] = new int[5]; //declaramos variable números tipo int igualamos a new y tipo variable
```

```
        numeros [0] = 5; //la posición 0 es la primera  
        numeros [1] = 220;  
        numeros [2] = 8;  
        numeros [3] = 458;  
        numeros [4] = 22;
```

```
        System.out.println "[" +numeros[0] +"]");  
        System.out.println "[" +numeros[1] +"]");  
        System.out.println "[" +numeros[2] +"]");  
        System.out.println "[" +numeros[3] +"]");  
        System.out.println "[" +numeros[4] +"]");
```

```
    }  
}
```

```
F:\curso>java Vectores  
[5]  
[220]  
[8]  
[458]  
[22]  
F:\curso>
```

19) Arreglos unidimensionales (vectores dinámicos)

A diferencia del vector pasado podemos apoyarnos del método For para asignar los valores de forma automática por ejemplo para crear el vector 10 – 11 - 12 – 13 - 14

```
public class Vectores2{
    public static void main(String args[]){
        int numeros[] = new int[5];

        for (int i=0; i<numeros.length; i++){ //utilizando la variable y el método length contamos el número de
                                                //espacios en el vector que son 5 en el ejemplo

            numeros[i]=i+10;
        }

        System.out.println "[" +numeros[0] +"]");
        System.out.println "[" +numeros[1] +"]");
        System.out.println "[" +numeros[2] +"]");
        System.out.println "[" +numeros[3] +"]");
        System.out.println "[" +numeros[4] +"]");

    }
}
```

- En este caso en la primer vuelta *i* vale 0, al ser menor que 5 cumple la condición, se va a la instrucción: numero [0]=0+10, por lo tanto en la posición 0 tendremos el valor 10, se aplica el incremento **i++**, ahora **i=1**
- En la segunda vuelta *i* vale 1, al ser menor que 5 cumple la condición, se va a la instrucción: numero [1]=1+10, por lo tanto en la posición 1 tendremos el valor 11, se aplica el incremento **i++**, ahora **i=2**
- En la tercer vuelta *i* vale 2, al ser menor que 5 cumple la condición, se va a la instrucción: numero [2]=2+10, por lo tanto en la posición 2 tendremos el valor 12, se aplica el incremento **i++**, ahora **i=3**.
- En la cuarta vuelta *i* vale 3, al ser menor que 5 cumple la condición, se va a la instrucción: numero [3]=3+10, por lo tanto en la posición 3 tendremos el valor 13, se aplica el incremento **i++**, ahora **i=4**.
- En la quinta vuelta *i* vale 4, al ser menor que 5 cumple la condición, se va a la instrucción: numero [4]=4+10, por lo tanto en la posición 4 tendremos el valor 14, se aplica el incremento **i++**, ahora **i=5**.
- En la sexta vuelta *i* vale 5, al ser igual que 5 no cumple la condición y termina el bucle



```
P:\curso>java Vectores2
[10]
[11]
[12]
[13]
[14]
```

20) Ejercicio vectores dinámicos

Ejercicio



Realizar un programa que contenga un vector cuya longitud sea proporcionado por el usuario, posteriormente solicitar al usuario los números necesarios para llenar todas las posiciones del vector, es decir, si el usuario solicitó un arreglo de 10 posiciones, entonces el programa deberá de solicitarle al usuario 10 números, y finalmente se debe de imprimir el arreglo en pantalla de la siguiente manera:

[5] [54] [12] [65] [6] [54] [897]

```
import java.util.Scanner;

public class Vectores3{
    public static void main(String args[]){
        Scanner entra = new Scanner(System.in);
        int valUno=0;
        System.out.print("Escribe el numero de espacios del vector:");
        valUno=entra.nextInt();

        int numeros[] = new int[valUno];
        System.out.println("");

        for(int i=0; i<numeros.length;i++){
            System.out.print("Escribe el numero que corresponde al espacio " + (i+1) + " del vector:");
            numeros[i]=entra.nextInt();
        } //en este primer bucle hacemos que se pida y guarde los valores de cada posición del vector
        System.out.println("");
        for(int i=0; i<numeros.length;i++){
            System.out.print("[ " + numeros[i] + " ]");

        } //en este segundo bucle damos la instrucción de que se impriman los valores, y se imprime la cantidad de
        posiciones
    }
}
```

```

F:\curso>java Vectores3
Escribe el numero de espacios del vector:7

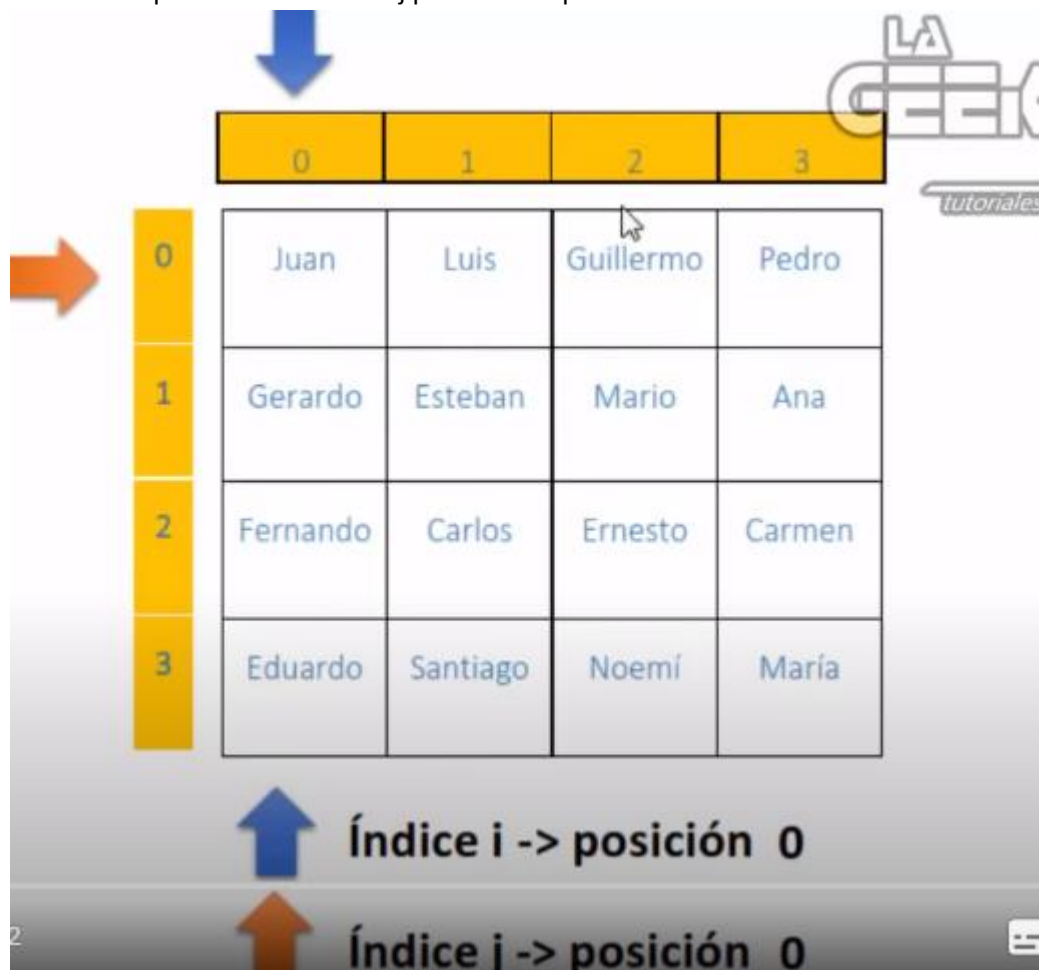
Escribe el numero que corresponde al espacio 1 del vector:5
Escribe el numero que corresponde al espacio 2 del vector:54
Escribe el numero que corresponde al espacio 3 del vector:12
Escribe el numero que corresponde al espacio 4 del vector:65
Escribe el numero que corresponde al espacio 5 del vector:6
Escribe el numero que corresponde al espacio 6 del vector:54
Escribe el numero que corresponde al espacio 7 del vector:897

[ 5 1[ 54 1[ 12 1[ 65 1[ 6 1[ 54 1[ 897 ]
F:\curso>

```

21) Arreglos bidimensionales (matrices)

Vamos a requerir de dos índices j para filas e i para columnas



Ejercicio



Crear un programa que muestre en pantalla una Matriz de 2 Filas y 2 Columnas con un número dentro de cada posición tal y como se muestra a continuación:

```
[5] [2]
[2] [5]
```

```
public class Matrices{
    public static void main(String args[]){
        int numeros [][]= new int [2][2]; //para matriz ponemos 2 corchetes al declarar la variable, primer numero es para filas, segundo columnas
        numeros[0][0]= 5; //fila 1 , columna 1
        numeros[0][1]= 2; //fila 1 , columna 2
        numeros[1][0]= 2; //fila 2 , columna 1
        numeros[1][1]= 5; //fila 2 , columna 2

        System.out.println "["+numeros[0][0]+"]"+"["+numeros[0][1]+"]");
        System.out.println "["+numeros[1][0]+"]"+"["+numeros[1][1]+"]");

    }
}
```

```
D:\curso>java Matrices
[5][2]
[2][5]
```

22) Arreglos bidimensionales (matrices dinamicas)

A diferencia de la matriz pasada podemos apoyarnos del método For para asignar los valores de forma automática

Ejercicio: Crea una matriz dinámica que solicite el número de filas, columnas y llene los espacios de la siguiente forma:

1	2	3
4	5	6
7	8	9

```
import java.util.Scanner;

public class Matrices2{
    public static void main(String args[]){
        Scanner entra = new Scanner(System.in);
        int filas=0,columnas=0,contador=1;

        System.out.println("Escribe el numero de filas:");
        filas = entra.nextInt();

        System.out.println("Escribe el numero de columnas:");
        columnas = entra.nextInt();

        System.out.println("Escribe el intervalo:");
        contador = entra.nextInt();

        int numeros[][]=new int[filas][columnas];

        for (int j=0; j < filas; j++){
            for(int i=0; i < columnas; i++){
                numeros[j][i]=contador;
                contador++;
                System.out.print("["+numeros[j][i]+"]");

            }
            System.out.println("");
        }
    }
}
```

```
F:\curso>java Matrices2
Escribe el numero de filas:
3
Escribe el numero de columnas:
3
Escribe el intervalo:
1
[1][2][3]
[4][5][6]
[7][8][9]
F:\curso>
```

- Asignamos valores, fila =3, columna=3 y contador =1
- 1er vuelta en j, j=0, 1er vuelta en i, i =0, contador = 1
- 1er vuelta en j, j=0, 2da vuelta en i, i =1, contador = 2
- 1er vuelta en j, j=0, 3ra vuelta en i, i =2, contador = 3
- 1er vuelta en j, j=0, 4ta vuelta en i, i =3, termina bucle en i, imprime salto de línea
- 2da vuelta en j, j=1, 1er vuelta en i, i =0, contador = 4
- 2da vuelta en j, j=1, 2da vuelta en i, i =1, contador = 5
- 2da vuelta en j, j=1, 3ra vuelta en i, i =2, contador = 6
- 2da vuelta en j, j=1, 4ta vuelta en i, i =3, termina bucle en i, imprime salto de línea
- 3ra vuelta en j, j=2, 1er vuelta en i, i =0, contador = 4
- 3ra vuelta en j, j=2, 2da vuelta en i, i =1, contador = 5
- 3ra vuelta en j, j=2, 3ra vuelta en i, i =2, contador = 6
- 3ra vuelta en j, j=2, 4ta vuelta en i, i =3, termina bucle en i, imprime salto de línea
- 4ta vuelta en j, j=3, termina bucle en j

23) Interfaces graficas (librería swing)

```
import javax.swing.*; //importamos libreria grafica
```

```
public class Formulario extends JFrame{ //extends para llamar al marco Frame (marco), extends para herencia  
    private JLabel label1; /*declaramos una etiqueta, la declaración privada, para entender su funcionamiento imagina  
el uso de los celulares privados y publicos  
    /*los label deben ser privados
```

```
public Formulario(){ /*declaramos un constructor donde se realiza toda la actividad antes del método main  
    setLayout(null); /*para indicar al programa que debe imprimir la etiqueta, colocamos null para que no lo  
ponga donde quiera  
    label1 = new JLabel("Holo"); /*inicializamos el label  
    label1.setBounds(10,20,200,300); /*setBounds para indicar coordenadas de la etiqueta (x,y,ancho,alto)  
ancho y alto en pixeles  
    add(label1); /*add para indicar en donde se va a colocar  
}
```

```
public static void main(String args[]){ /*iniciamos nuestro metodo void como siempre
```

```
    Formulario Formulario1= new Formulario(); /* declaramos nuestra variable tipo Formulario, todo debe  
ser
```

```
    Formulario1.setBounds(0,0,400,300); /* colocamos coordenadas  
    Formulario1.setVisible(true); /*setVisible para mostrar, ture para que se muestre false para que no  
    Formulario1.setLocationRelativeTo(null); /*para indicar que el formulario salga al centro de la pantalla  
    }  
}
```

Código sin notas

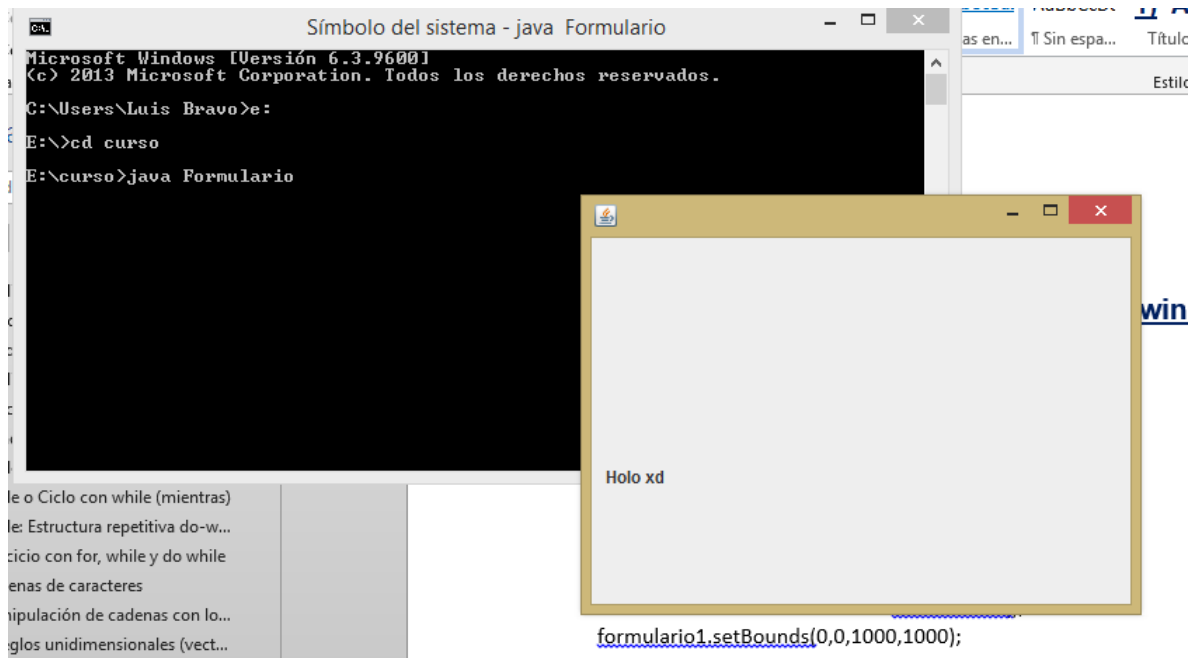
```
import javax.swing.*;  
  
public class Formulario extends JFrame{  
    private JLabel label1;  
    public Formulario(){  
        setLayout(null);  
        label1 = new JLabel("Holo");  
        label1.setBounds(10,20,200,300);  
        add(label1);  
    }  
}
```

```

public static void main(String args[]){
    Formulario Formulario1= new Formulario();
    Formulario1.setBounds(0,0,400,300);
    Formulario1.setVisible(true);
    Formulario1.setLocationRelativeTo(null);

}
}

```



24) Interfaces gráficas manipulación (swing - JFrame)

Primer prueba

```
import javax.swing.*;

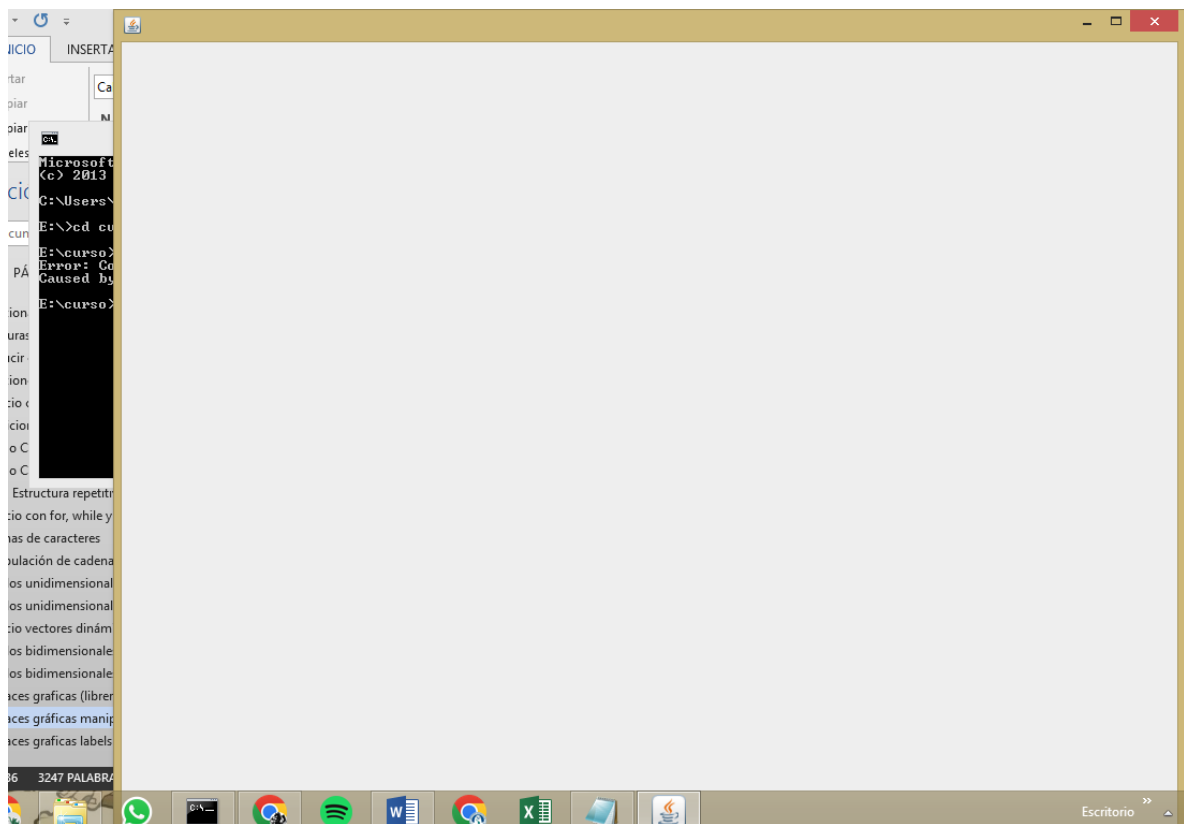
public class Formulario2 extends JFrame{
    public Formulario2(){
        setLayout(null);
    }
    public static void main (String [] args){
        Formulario2 formulario1 = new Formulario2();
        formulario1.setBounds(0,0,1000,1000);
        formulario1.setVisible(true);

    }
}
```

Segunda prueba

```
import javax.swing.*;

public class Formulario2 extends JFrame{
    public Formulario2(){
        setLayout(null);
    }
    public static void main (String [] args){
        Formulario2 formulario1 = new Formulario2();
        formulario1.setBounds(0,0,1000,1000);
        formulario1.setVisible(true); // para indicar que se vea
        formulario1.setLocationRelativeTo(null); // colocar frame al centro de la pantalla
        formulario1.setResizable(true); // activar o desactivar ajuste de frame por el usuario true y false
    }
}
```



25) Interfaces graficas labels

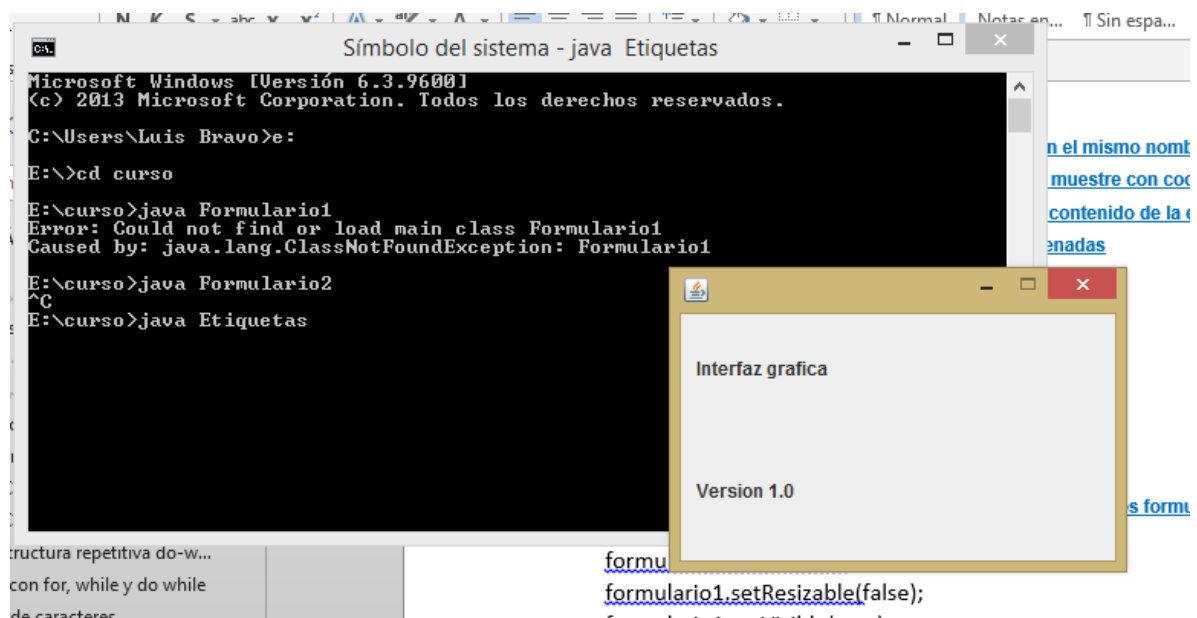
```
import javax.swing.*;

public class Etiquetas extends JFrame{

    private JLabel etiqueta1; //declaramos tipo JLabel privada y ponemos nombre
    private JLabel etiqueta2;

    public Etiquetas () { //declaramos constructor con el mismo nombre de la clase
        setLayout(null); //declarar que el constructor se muestre con coordenadas
        etiqueta1 = new JLabel("Interfaz grafica"); // contenido de la etiqueta
        etiqueta1.setBounds(10,20,300,30); // coordenadas
        add(etiqueta1); // que muestre la etiqueta
        etiqueta2 = new JLabel("Version 1.0");
        etiqueta2.setBounds(10,100,100,30);
        add(etiqueta2);
    }

    public static void main(String[] args){
        Etiquetas formulario1 = new Etiquetas(); // declaramos formulario con el mismo nombre de la clase e
igualamos con objeto con el mismo nombre de la clase
        formulario1.setBounds(0,0,300,200);
        formulario1.setResizable(false);
        formulario1.setVisible(true);
        formulario1.setLocationRelativeTo(null);
    }
}
```



26) Interfaces graficas botones

```
import javax.swing.*;
import java.awt.event.*;
//importar componente para dseho grafico
//importar libreria para controlar eventos

public class Boton extends JFrame implements ActionListener{ //ahora no solo ponemos extends JFrame
// ahora agregamos implements ActionListener para que el programa escuche o ponga atencion de la accion del
//usuario
    JButton boton1; //los comoponentes no se inicializan es decir no ponemos =, solo basta con su nombre

    //primero el diseho del boton, creamos un constructor
    public Boton() { //constructor con mismo nombre de la clase
        setLayout(null); // para indicar que se muestre con coordenadas
        boton1 = new JButton("cerrar"); // contenido del boton
        boton1.setBounds(300,250,100,30);
        add(boton1); //para que aparezca
        boton1.addActionListener(this); // decimos que le agregaremos una accion,
        //this para decir que con este boton al hacer click inicia la accion
    }

    //la accion o evento se debe declarar despues
    public void actionPerformed(ActionEvent e){ //declaramoas un evento publico para crear un espacio en la
//memoria //actionPerformed(Accion realizada) , ActionEvent (evento de accion), //y despues de eso una letra para
//el espacio de memoria en este caso "e", abrimos llaves
        if(e.getSource() == boton1){ //usamos condicion (if) y con getSource(obtener fuente de)
//decimos que cuando se haga click en el boton 1
            System.exit(0); // esta sera la accion a realizar, en este caso salir
        }
    }

    public static void main (String args[]){
        Boton formulario1 = new Boton (); // declaramos formulario con el mismo nombre de la clase e igualamos
//con objeto con el mismo nombre de la clase
        formulario1.setBounds(0,0,450,350);
        formulario1.setVisible(true);
        formulario1.setResizable(false);
        formulario1.setLocationRelativeTo(null);
    }
}
```

