



# Tema 5

---

## Estructuras de Control



## ESCRITURA DE DATOS EN LA CONSOLA DEL SISTEMA

- El objeto **System.out** de la clase **PrintStream** representa la consola de salida del sistema, y cuenta, entre otros, con los métodos **print** y **println** que permiten enviar cadenas a la consola del sistema. El método **printf** permite como su homónimo en C dar formato a los datos de salida.



# Ejemplo Escritura

```
public class Escritura //Sin lectura
{
    public static void main(String[] args)
    {
        int res;
        int num1=20;
        int num2=40;
        res=num1+num2;
        System.out.print("Vamos a calcular la suma de los numeros: " +num1+" y ");
        System.out.println(num2);
        System.out.print ("El resultado de la suma es: ");
        System.out.println( res);
    }
}
```




## LECTURA DE DATOS A TRAVÉS DE LA CLASE Scanner

- Utilizando la clase Scanner es posible leer los datos introducidos por el usuario desde teclado, para ello debemos utilizar el objeto de entrada por defecto (**System.in**).
- El método **nextInt()** toma lo introducido por el usuario y lo convierte en un número entero (existen varios métodos next... para cada tipo de dato primitivo de Java y para algunos tipos de datos compuestos, como cadena de caracteres)
- Lo primero que tenemos que hacer es declarar un objeto de la siguiente forma:
  - **Scanner teclado=new Scanner(System.in);**



## Método next según dato introducido

- a) Si introducimos un texto
  - `String texto=teclado.nextLine();` (con espacios)
  - `String texto=teclado.next();` (sin espacios)
- b) Si es un entero
  - `int num= teclado.nextInt();`
- c) Si es un double
  - `double num= teclado.nextDouble();`
- d) Si es un carácter
  - `String cadena=teclado.next();`
  - `char carácter=cadena.charAt(0);`
- Importante es no olvidar que es necesaria la clase **java.util** para que todo funcione. Así al principio del programa incluiremos:
  - `Import java.util.Scanner;`



```

import java.io.*;
import java.util.*;
public class Lectura2Scanner {
    public static void main(String[] args) {
        Scanner lector=new Scanner(System.in);
        System.out.println("Introduce un numero:");
        int num=lector.nextInt();
        System.out.println("El numero introducido fue "+num);
        System.out.println("Introduce una cadena de una palabra");
        String cad=lector.next();
        System.out.println("La cadena de una palabra introducida fue: "+cad);
        System.out.println("Introduce otra cadena de una palabra");
        String otraCad=lector.next();
        System.out.println("La cadena introducida fue: "+otraCad);
        System.out.println("Introduce otro numero grande:");
        double otroNum=lector.nextDouble();
        System.out.println("El número introducido fue: "+otroNum);
        System.out.println("Introduce otra cadena de varias palabras");
        String variasCad=lector.nextLine();
        System.out.println("La cadena introducida fue: "+variasCad);
        System.out.println("Introduce otra cadena de varias palabras");
        String masCad=lector.nextLine();
        System.out.println("La cadena digitada fue: "+masCad);

    }
}

```



## Ejemplo

```
import java.util.*;
public class LeerDatosScanner
{
    public static void main(String args[])
    {
        Scanner teclado= new Scanner(System.in);
        System.out.println("Dame un nombre:");
        String texto=teclado.nextLine();
        System.out.println("Dame la edad:");
        int edad=teclado.nextInt();
        System.out.println("Dame la estatura:");
        Double estatura=teclado.nextDouble();
        System.out.println("Nombre "+texto+"\nEdad:"+edad+"\nAltura: "+estatura);
    }
}
```



## Ejemplo

**/\*entrada de caracteres por teclado\*/**

```
import java.io.*;
import java.util.*;
class LeerCarBucleScanner
{
    public static void main(String args[])
    {
        String cad;
        char c,resp;
        Scanner teclado=new Scanner(System.in);
        do
        {
            System.out.println("Introduce un caracter y te dire si es una letra:");
            cad=teclado.next();
            c=cad.charAt(0);
            System.out.println("Se ha pulsado "+c);
            if (c >='a' && c<='z')
                System.out.println("El caracter es una letra minuscula\n En mayuscula: "+(char)(c-32));
            System.out.println("Quiere mas (s/n)");
            cad=teclado.next();
            resp=cad.charAt(0);
        } while (resp!='s');
    }
}
```





## Estructuras de Control

- Las estructuras de control del flujo de ejecución permiten tomar decisiones y realizar un bloque de instrucciones repetidas veces.
- Hay 2 tipos principales:
  - Condicionales: `if`, `switch`
  - Bucles: `for`, `while`, `do while`
- Otras instrucciones que permiten interrumpir el flujo normal de ejecución son **break** y **continue**.



# Sentencias condicionales: if

```
if (expresion)
{
    bloque de sentencias
}
```

```
if (x>18)
{
    System.out.print("x es mayor que 18");
}
```

```
if (expresion)
{
    bloque de sentencias
}
else
{
    bloque de sentencias
}
```

```
if (x>18)
{
    System.out.print("x es mayor que 18");
}
else
{
    System.out.print("x no es mayor que 18");
}
```

**Con una sola instrucción no es obligatorio poner las llaves pero si recomendable**



# Sentencias Condicionales: Switch

```
switch(variable)
{
case valor1:
    sentencia 1:
    .
    sentencia n;
    break;
case valor2:
    sentencia 1:
    .
    sentencia n;
    break;
case .....
default:
    sentencia 1:
    .
    sentencia n;
```

```
switch(numero)
{
case 1:
    System.out.print("EL valor es 1");
    break;
case 2:
    System.out.print("EL valor es 2");
    break;
default:
    System.out.print("ni uno ni dos");
}
```

- Más efectivo que un if anidado.
- No pueden evaluarse expresiones relacionales o lógicas.
- default no es obligatorio pero suele usarse para el caso de que no se cumpla ninguno de los casos anteriores.
- break. Se utiliza para evitar que una vez que se ha entrado en un case se sigan evaluando el resto.



# Ejemplo

```
public class app
{
    public static void main(String[] args)
    {
        int day = 3;

        switch(day) {
            case 0:
                System.out.println("Es domingo.");
                break;
            case 1:
                System.out.println("Es lunes.");
                break;
            case 2:
                System.out.println("Es martes.");
                break;
            case 3:
                System.out.println("Es miércoles.");
                break;
            case 4:
                System.out.println("Es jueves.");
                break;
            case 5:
                System.out.println("Es viernes.");
                break;
            default:
                System.out.println("Debe ser sábado.");
        }
    }
}
```

Este es el resultado:

```
C:\>java app
Es miércoles.
```

# Ejemplo

...cont

```
{  
    int opcion;  
    Scanner teclado = new Scanner(System.in);  
  
    System.out.println("DICHOS:");  
    System.out.println("1. Para flojos");  
    System.out.println("2. Para dar ánimos");  
    System.out.println("3. De experiencia");  
    System.out.println("Opción: ");  
    opcion=teclado.nextInt();  
    switch(opcion){  
        case 1: System.out.println("No por mucho madrugar amanece más  
temprano"); break;  
        case 2: System.out.println("A darle que es mole de olla"); break;  
        case 3: System.out.println("Más sabe el diablo por viejo que por diablo");  
break;  
        default System.out.println("Opcion invalida"); break;  
    }  
}
```





## Bucles: while

**while** (condición)

```
{  
    sentencia 1;  
    .....  
    sentencia n;  
}
```

**Orden de ejecución:**

1. Se evalúa la condición
2. Resultado de la evaluación es verdadero se ejecutan las sentencias de dentro del bucle y se vuelve a evaluar la condición.
3. Resultado es falso las sentencias de dentro del bucle no se ejecutan.



## Ejemplo

```
/*suma numeros de 1 a N */
import java.util.*;
public class BucleMientras
{
    public static void main(String args[])
    {
        int n, contador=1;
        int suma=0;
        Scanner teclado= new Scanner(System.in);
        System.out.println("Dame el valor de n:");
        n=teclado.nextInt();
        while(contador<=n)
        {
            suma+=contador;
            contador++;
        }
        System.out.println("Suma= "+suma);
    }
}
```



# Bucles:do-while

```
do
{
    sentencias;
}while(condición);
```

## Orden de ejecución:

- Se ejecuta el bloque de sentencias.
- Se evalúa la condición.
  - Si es cierta se vuelven a ejecutar las sentencias y luego se evalúa otra vez la condición.
  - Si es falsa se continúa con la ejecución del programa.





## Bucles:for

```
for([v1=e1],[v2=e2]...:[condición];[progresión])
{
    sentencias;
}
```

### **Orden de ejecución:**

- Inicializamos las variables v1, v2, una sólo vez al principio del bucle.
- Se evalúa la condición.
  - Si el resultado es verdadero.
    - Se ejecutan las sentencias internas al bucle for.
    - Se ejecuta la expresión de la progresión y se vuelve a evaluar la condición.
  - Si el resultado es falso se da por finalizada la sentencia for



# Bucles:for

## Observaciones:

- pueden eliminarse uno, dos o los tres campos.
- pueden existir varias inicializaciones y/o varias actualizaciones (separadas por comas).
- cualquier expresión es válida para los campos.

## Ejemplos:

```
for (x=0; x<100;)
    System.out.println(x++);
```

```
for (x=0, y=0; x<100; x++, y+=2)
    System.out.println ("x: "+ x + "y: "+ y);
```

```
for (x=0, System.out.println("Comienzo del bucle\n"); x<100; x++)
    System.out.println (x);
```



# Sentencias de salto

- **break.**
  - Finaliza la ejecución de una sentencia switch, while, do o for en la que aparece.
  - Si está dentro de un bucle que a su vez está anidado en otro, provoca la salida del bloque interno al anterior.
- **continue.**
  - Obliga a ejecutar la siguiente iteración en el bucle correspondiente a la sentencia while, do, for en el que esté contenida.

```
do {  
    System.out.println("Dame el valor de n:");  
    x=teclado.nextInt();  
    if (x<0) continue;  
    System.out.println ( x);  
    System.out.println ("probar");  
} while (X!=100);
```

Visualiza solamente los positivos junto con "probar". Si  $x < 0$  salta a evaluar la condición.