

PROGRAMACIÓN ORIENTADA A OBJETOS - 22951

Universidad Industrial de Santander
Escuela de Ingeniería de Sistemas e Informática
Programa de Ingeniería de Sistemas



#LaUISqueQueremos





Universidad
Industrial de
Santander

Estructuras de Control



Estructuras de Control

- Tres estructuras de control básicas:
 - Condicionales (o bifurcaciones).
 - Ciclos iterativos.
 - Ciclos condicionales.
- Arreglos:
 - Vectores y matrices.



Universidad
Industrial de
Santander



Estructura Condicional

- Para la estructura de control condicional (o bifurcación) se utilizan las palabras claves *if* y *else*.

```
if (condicion) {  
    // Código a ejecutar SI se cumple la condición  
} else {  
    // Código a ejecutar si NO se cumple la condición  
}
```

La **condición** es una comparación de dos expresiones que produce un resultado booleano.



Universidad
Industrial de
Santander



Estructura condicional - ejemplo

```
public class Condicional1 {  
  
    public static Scanner entrada = new Scanner(System.in);  
  
    public static void main(String[] args) {  
        System.out.println("Ingrese un valor entero");  
        int valor = entrada.nextInt();  
  
        if (valor>5) {  
            System.out.println("El valor ingresado es mayor que 5");  
        } else {  
            System.out.println("El valor ingresado no es mayor que 5");  
        }  
    }  
}
```



Estructura condicional Simple

```
if (condicion) {  
    // Código a ejecutar SI se cumple la condición  
}
```

```
public class Condicional2 {  
  
    public static Scanner entrada = new Scanner(System.in);  
  
    public static void main(String[] args) {  
        System.out.println("Ingrese un valor entero");  
        int valor = entrada.nextInt();  
  
        if (valor>5) {  
            System.out.println("El valor ingresado es mayor que 5");  
        }  
        System.out.println("Aqui termina el programa");  
    }  
}
```



Universidad
Industrial de
Santander



Estructura condicional Simple

- Si el código a ejecutar consta de una sola instrucción se pueden omitir los corchetes.
- Sin embargo, es recomendado siempre ponerlos, esto facilita la lectura del código

```
if (n > 5) {  
    System.out.println("el valor n es mayor que 5");  
}
```

```
if (n > 5)  
    System.out.println("el valor n es mayor que 5");
```



Condición y Operadores de comparación

- La condición está constituida por dos expresiones comparadas con un operador de comparación. El resultado de la comparación es un booleano.

Operador	Descripción
==	Igual que
>	Mayor que
<	Menor que
>=	Mayor igual
<=	Menor igual
!=	Diferente que



Operadores Lógicos

- Dos condiciones pueden ser unidas en una nueva condición a través de operadores lógicos:
 - $(n \geq 5) \ \&\& \ (n \leq 10)$
 - Cada condición es evaluada y luego el operador lógico es aplicado.
 - En este caso (operador **Y**), la condición global solo es verdadera si las dos condiciones parciales son verdaderas.

```
if ((n >= 5) && (n<=10)) {  
    // Solo ingresa si n esta entre 5 y 10  
}
```

Operadores Lógicos

- Para el operador lógico O se utiliza en Java dos barras verticales:
 - $(n \leq 5) \ || \ (n \geq 10)$
 - La condición global es verdadera si una de las condiciones parciales es verdadera.
- Para el operador de negación se utiliza el símbolo de cerrar admiración !
 - El operador de negación es unario.
 - $!(n == 5)$
 - La condición global es verdadera cuando la condición parcial es falsa y viceversa



Condicionales Anidados



Universidad
Industrial de
Santander

```
if (condicion1) {  
    if (condicion2) {  
        // La condicion1 y la condicion2 se cumplen  
    } else {  
        // La condicion1 se cumple. La condicion2 NO se cumple  
    }  
} else {  
    if (condicion3) {  
        // La condicion1 NO se cumple y la condicion3 se cumple  
    } else {  
        // La condicion1 y la condicion3 NO se cumplen  
    }  
}
```



Ejercicio - Condicional

- Realice una aplicación (*Condicional3*) que lea 3 números enteros (x, y, z) desde la consola y determina si los tres son iguales, si hay dos iguales decir cuales lo son o si todos son diferentes.



Universidad
Industrial de
Santander



Solución



Universidad
Industrial de
Santander

```
if (x==y) {  
    if (y==z) {  
        System.out.println("x, y, z son iguales");  
    } else {  
        System.out.println("x, y son iguales");  
    }  
} else {  
    if (x==z) {  
        System.out.println("x, z son iguales");  
    } else {  
        if (y==z) {  
            System.out.println("y, z son iguales");  
        } else {  
            System.out.println("x, y, z son diferentes");  
        }  
    }  
}
```



Estructura de ciclo: For

- La estructura de control *for* permite repetir un número determinado de veces el mismo conjunto de instrucciones.

```
for (int i = 0; i < 5; i++) {  
    System.out.println("El valor de i es: " + i);  
    System.out.println("El cuadrado de i es :" + i * i);  
}
```

```
El valor de i es: 0  
El cuadrado de i es :0  
El valor de i es: 1  
El cuadrado de i es :1  
El valor de i es: 2  
El cuadrado de i es :4  
El valor de i es: 3  
El cuadrado de i es :9  
El valor de i es: 4  
El cuadrado de i es :16
```



Universidad
Industrial de
Santander



El ciclo for



Universidad
Industrial de
Santander



Palabra clave for

Condición

Evaluada a cada iteración. Si la condición es falsa el ciclo termina

```
for (int i = 0; i < 5; i++) {  
    }  
}
```

Declaración e inicialización

Instrucción ejecutada una sola vez al inicio del ciclo

Incremento

Se ejecuta al final del ciclo. Permite aumentar el valor de la variable de control

El ciclo for Simple

- Al igual que en el condicional, los corchetes son opcionales en un for cuando se trata de una sola instrucción.
- Nuevamente, la recomendación es usar siempre corchetes.

```
for (int i = 0; i < 5; i++) {  
    System.out.println("El valor de i es: " + i);  
}
```

```
for (int i = 0; i < 5; i++)  
    System.out.println("El valor de i es: " + i);
```



Sintaxis de la instrucción for

```
for (declaracion_inicializacion; condicion; incremento) {  
    // Cuerpo del ciclo  
}
```

- **Cuidado:** si la condición nunca se evalúa a falso, el ciclo se repite de forma indefinida:

```
for (int i = 0; i > -1; i++) {  
    System.out.println("El valor de i es: " + i);  
}
```



Ejercicio

- Imprimir la tabla de multiplicar indicada por el usuario

Tabla de multiplicar del 4

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$4 \times 4 = 16$$

$$4 \times 5 = 20$$

$$4 \times 6 = 24$$

$$4 \times 7 = 28$$

$$4 \times 8 = 32$$

$$4 \times 9 = 36$$

$$4 \times 10 = 40$$



Universidad
Industrial de
Santander



Qué imprime el siguiente código?

```
for (int i = 0; i < 5; ++i) {  
    System.out.print(i);  
    if (i % 2 == 0) {  
        System.out.print("p");  
    }  
    System.out.print(" ");  
}
```

- A) 0p 1 2p 3p 4
- B) 0p 1 2p 3 4p
- C) 0p 1p 2 3p 4
- D) 0 1 2p 3 4p



Universidad
Industrial de
Santander



Otros ciclos válidos

```
for(int p = 0; p < 10; p += 2) {  
}
```

La variable de control p aumenta de 2 en 2. p toma el valor de 0, 2, 4, 6, 8

```
for(int k = 10; k > 0; k--) {  
}
```

La variable de control k disminuye. k toma el valor 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

```
for(int i = 0; i >= 0; i++) {  
}
```

Condición que nunca se cumple

Errores comunes

```
for (int i = 0; i < 5; i++);  
System.out.println("Hola");
```

- Un punto y coma al final del ciclo supone un cuerpo vacío del ciclo for.
- El ciclo efectivamente se ejecuta, i toma el valor de 1 a 5 y termina, en seguida se ejecuta la instrucción que muestra el mensaje Hola.



Universidad
Industrial de
Santander



Errores comunes

```
for (int i = 0; i < 5; i++) {  
    if (i < 3) {  
        i--;  
    }  
}
```



- Modificar la variable de control dentro del cuerpo del ciclo es una mala práctica.
- Evitar (en lo posible) este tipo de operaciones en un ciclo for, puede inducir a errores.

Ejercicio

- Escriba una aplicación que solicite al usuario un número de notas a promediar, enseguida solicite cada una de las notas y al final muestre el promedio de esas notas

```
Numero de notas a promediar: 4  
3.5  
4.5  
3.8  
4.2  
El promedio las notas es : 4.1
```



Universidad
Industrial de
Santander



Ciclos for anidados

```
for (int i=0; i<10; i++) {  
    System.out.println("El valor de i es : " + i);  
    for (int j=0; j<10; j++) {  
        System.out.println("El valor de j es : " + j);  
    }  
}
```

- Los ciclos *for* pueden ser anidados
- Cada ciclo debe definir su propia variable de control



Ejercicio

- Escriba una aplicación (TablasCompletas) que imprima en la consola las tablas de multiplicar del 2 al 9. Cada tabla debe ser indicada del mayor a menor.

Tabla de multiplicar del 4

$$4 \times 10 = 40$$

$$4 \times 9 = 36$$

$$4 \times 8 = 32$$

$$4 \times 7 = 28$$

$$4 \times 6 = 24$$

$$4 \times 5 = 20$$

$$4 \times 4 = 16$$

$$4 \times 3 = 12$$

$$4 \times 2 = 8$$

$$4 \times 1 = 4$$



Universidad
Industrial de
Santander





Ciclos condicionales : do-while y while

- Un ciclo for se utiliza cuando se conoce apriori el numero de veces que se va a repetir las instrucciones.
- El algunos casos no es posible conocer ese número al avance, en ese caso se usa un ciclo condicional.
- Existen dos tipos de ciclos condicionales en Java **do-while** y **while**



Aplicacion promedio de notas

```
System.out.println("Ingresa el numero de notas");
int numeroDeNotas = entrada.nextInt();
double suma = 0;
if (numeroDeNotas > 0) {
    for (int i = 1; i <= numeroDeNotas; i++) {
        System.out.println("Ingresa la nota numero " + i);
        double nota = entrada.nextDouble();
        suma = suma + nota;
    }
    System.out.println("Promedio = " + suma / numeroDeNotas);
}
```

- Como forzar al usuario a ingresar un numero de notas superior a 0?



Palabra clave for

do

```
{  
    numeroDeNotas = entrada.nextInt();  
    System.out.println("Ingrese el numero de notas");  
} while(numeroDeNotas < 1);
```

Palabra clave while

Condición

Si la condición es verdadera el ciclo se ejecuta nuevamente

Si la condición es falsa el programa continua

Sintaxis de do-while

```
do {  
    // Cuerpo del do-while  
} while(condicion);
```

- Como para el condicional if:
 - La condición puede usar operadores lógicos
 - La condición debe estar entre paréntesis.
- Las instrucciones en el cuerpo del ciclo se ejecutan por lo menos una vez.
- Si la condición nunca es evaluada en falso, las instrucciones se ejecutan indefinidamente.



Universidad
Industrial de
Santander



Promedio con validación

```
int numeroDeNotas = 0;

do {
    numeroDeNotas = entrada.nextInt();
    System.out.println("Ingresa el numero de notas");
} while(numeroDeNotas < 1);

double suma = 0;
if (numeroDeNotas > 0) {
    for (int i = 1; i <= numeroDeNotas; i++) {
        System.out.println("Ingresa la nota numero " + i);
        double nota = entrada.nextDouble();
        suma = suma + nota;
    }
    System.out.println("Promedio = " + suma / numeroDeNotas);
}
```



Sintaxis while

```
while (condicion) {  
    // cuerpo del while  
}
```

- Las mismas características del do-while.
- La diferencia es que la condición es evaluada antes de entrar a ejecutar el cuerpo.
- Si la primera evaluación de la condición es false, el bloque puede no ser ejecutado.



Universidad
Industrial de
Santander



Diferencias do-while y while



Universidad
Industrial de
Santander

```
int i = 100;  
  
do {  
    System.out.println("Hola");  
} while (i < 5);
```

Cual es la diferencia entre los dos bloques de código?

```
int i = 100;  
  
while (i < 5) {  
    System.out.println("Hola");  
}
```



Arreglos : Vectores

Declaración

```
int[] mivector;
```

Inicialización

```
mivector = new int[5];
```

Acceso a elemento

```
int a = mivector[2];  
mivector[3] = 5;
```

El índice de acceso a un arreglo siempre empieza en 0



Universidad
Industrial de
Santander



Ejercicio

- Generar un vector de enteros de 50 posiciones con números aleatorios entre 0 y 19.
- Decir cuántos de esos números son pares.
- Imprimir el máximo, el mínimo, el promedio y la sumatoria de los números.

```
import java.util.Random;

public class Aleatorio {
    public static void main(String[] args) {
        Random random = new Random();
        int randomNumber = random.nextInt(19);
        System.out.println(randomNumber);
    }
}
```

Ejemplo Arreglo



Universidad
Industrial de
Santander

```
int[] mivector = new int[5];
```

```
for (int i = 0; i < mivector.length; i++) {  
    mivector[i] = i * i;  
}
```

```
double suma = 0;
```

```
for (int i = 0; i < mivector.length; i++) {  
    suma = suma + mivector[i];  
}
```

```
System.out.println("El promedio es " + suma / mivector.length);
```



Arreglos : Matriz

Declaración

```
int[][] mimatriz;
```

Inicialización

```
mimatriz = new int[5][8];
```

Acceso a elemento

```
int a = mimatriz[3][4];  
mimatriz[2][3] = 5;
```

El índice de acceso a un arreglo siempre empieza en 0



Universidad
Industrial de
Santander



Funciones

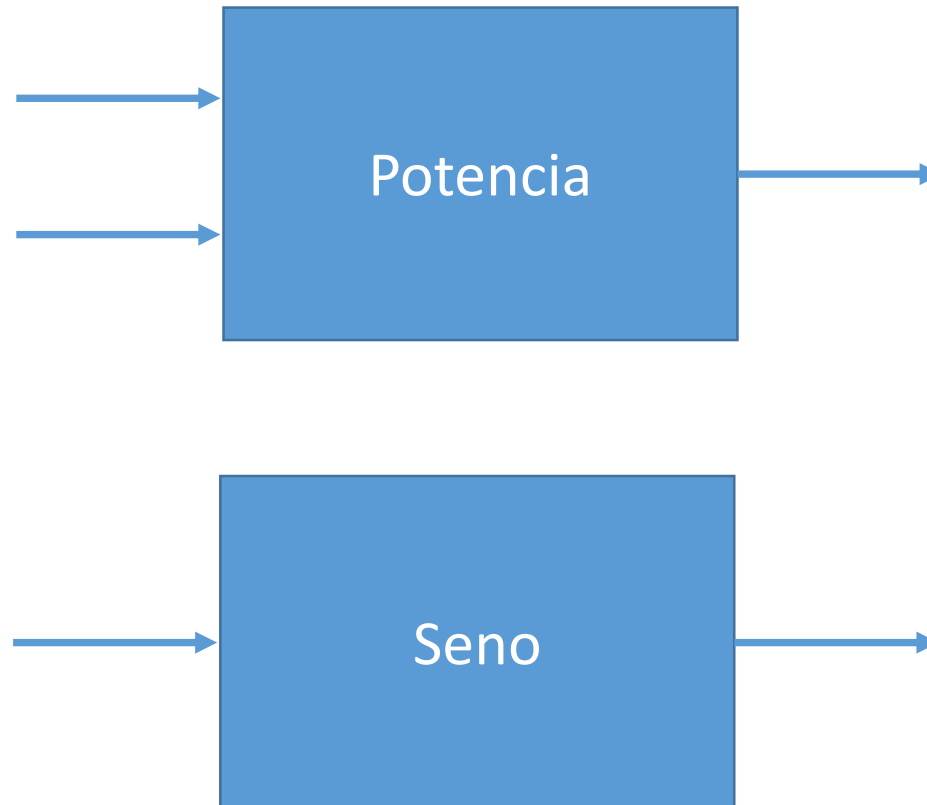
- El software moderno debe hacerse por partes (divide y vencerás)
- Las partes solucionan pequeños problemas y la unión de las partes solucionan problemas más complejos
- Dependiendo del paradigma esas partes toman diferentes nombre
 - Funciones (Procedimental)
 - Clases (Orientado a objetos)
 - Componentes (Basado en Componentes)
 - Modulos (nombre abstracto)



Universidad
Industrial de
Santander



Función



Universidad
Industrial de
Santander



Seno (serie de Taylor)

$$\text{sen}(x) = \frac{x^1}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots$$

$$\text{sen}(x) = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$





Universidad
Industrial de
Santander



#OrgulloESI

iGracias!

#LaUISqueQueremos