



# TRABAJO PRACTICO ESTRUCTURAS

## PROGRAMACION II

**ALUMNO: FABIAN IGNACIO CARDOZO**

```

import java.util.Arrays;
import java.util.Scanner;

public class Estructuras {
    public static void main(String[] args) {
/* Estructuras Condicionales:
1. Escribe un programa en Java que solicite al usuario un año y determine si es
bisiesto. Un año es bisiesto si es divisible por 4,
pero no por 100, salvo que sea divisible por 400.
*/

        Scanner input = new Scanner(System.in);
        System.out.println("Ingresa el año que vamos a analizar: ");
        int age = input.nextInt();

        if ((age % 4 == 0 && age % 100 != 0) || (age % 400 == 0)) {
            System.out.println("El año Ingresado es Biciesto");
        } else {
            System.out.println("El año Ingresado no es Biciesto");
        }

/*
2. Determinar el Mayor de Tres Números.
Escribe un programa en Java que pida al usuario tres números enteros y determine
cuál es el mayor.
*/

        System.out.println("Ingresa el primer numero: ");
        int num1 = input.nextInt();
        System.out.println("Ingresa el segundo numero: ");
        int num2 = input.nextInt();
        System.out.println("Ingresa el tercer numero: ");
        int num3 = input.nextInt();
        int numMayor = 0;

        if (num1 > num2 && num1 > num3) {
            numMayor = num1;
        } else if (num2 > num1 && num2 > num3) {
            numMayor = num2;
        } else {
            numMayor = num3;
        }
        System.out.println("De los Numero Ingresados el Mayor es el: " +
numMayor);

/*
3. Escribe un programa en Java que solicite al usuario su edad y clasifique su
etapa de vida según la siguiente tabla:
Menor de 12 años: "Niño"
Entre 12 y 17 años: "Adolescente"
Entre 18 y 59 años: "Adulto"
*/

```

60 años o más: "Adulto mayor"

```
*/
    System.out.println("Ingrese su edad: ");
    int edad = input.nextInt();

    if (edad <= 12 && edad != 0) {
        System.out.println("Eres un Niño");
    } else if (edad > 12 && edad <= 17) {
        System.out.println("Eres un Adolescente");
    } else if (edad > 17 && edad <= 59) {
        System.out.println("Eres un Adulto");
    } else if (edad >= 60 && edad <= 100) {
        System.out.println("Eres un Adulto Mayor");
    } else {
        System.out.println("edad no valida");
    }
}

/*
4. Calculadora de Descuento según categoría.
Escribe un programa que solicite al usuario el precio de un producto y su
categoría (A, B o C).
Luego, aplique los siguientes descuentos:
Categoría A: 10% de descuento
Categoría B: 15% de descuento
Categoría C: 20% de descuento
El programa debe mostrar el precio original, el descuento aplicado y el precio
final
*/

    System.out.println("Ingresa el Precio del Producto: ");
    double precio = input.nextDouble();
    input.nextLine();
    double desc;

    System.out.println("Ahora ingrese la Categoria del Producto: \n
A)\nB)\nC)");
    String categoria = input.nextLine().toUpperCase().trim();
    desc = switch (categoria) {
        case "A" -> precio * 0.10;
        case "B" -> precio * 0.15;
        case "C" -> precio * 0.20;
        default -> -1;
    };

    System.out.println("Categoría inválida, intente de nuevo.");

    double precioFinal = (precio - desc);
    System.out.println("El precio Original es de:" + precio);
    System.out.println("El descuento aplicado es de:" + desc);
    System.out.println("El precio Final es de: " + precioFinal);

/*
```

Estructuras de Repetición:

5. Suma de Números Pares (while).

Escribe un programa que solicite números al usuario y sume solo los números pares.

El ciclo debe continuar hasta que el usuario ingrese el número 0, momento en el que se debe mostrar la suma total de los pares ingresados.\*/

```
System.out.println("Ingrese números (0 para salir):");
int num;
double acum = 0;

while (true) {
    num = input.nextInt();
    if (num == 0) {
        break;
    }
    if (num % 2 == 0) {

        acum += num;
    }
}
System.out.println("La suma de lo numeros Pares Ingresado es: " + acum);
```

/\*

6. Contador de Positivos, Negativos y Ceros (for).

Escribe un programa que pida al usuario ingresar 10 números enteros y cuente cuántos son positivos, negativos y cuántos son ceros.

\*/

```
System.out.println("Ingrese un total de 10 numeros enteros");
int positivos = 0;
int negativos = 0;
int cero = 0;

System.out.println("Numero" + 0 + ": ");
int nume = input.nextInt();

if (nume > 0) {
    positivos++;
} else if (nume < 0) {
    negativos++;
} else {
    cero++;
}

System.out.println("Cantidad de Numeros Positivos:" + positivos);
System.out.println("Cantidad de Numeros:" + negativos);
System.out.println("Cantidad de Numeros que son ceros:" + cero);
```

/\*

7. Validación de Nota entre 0 y 10 (do-while).

Escribe un programa que solicite al usuario una nota entre 0 y 10. Si el usuario ingresa un número fuera de este rango, debe seguir pidiéndole la nota

hasta que ingrese un valor válido.

```
*/

    int nota;
    do {
        System.out.println("Ingrese una Nota entre 0 y 10: ");
        nota = input.nextInt();
        if (nota < 0 || nota > 10) {
            System.out.println("Error Nota " + nota + " esta fuera de
rango.");
            System.out.println("Vuelva a Ingresar una Nota.");
        } else {
            System.out.println("Nota: " + nota);
        }
    } while (nota < 0 || nota > 10);

/*
Funciones:
8. Cálculo del Precio Final con impuesto y descuento.
Crea un método calcularPrecioFinal(double impuesto, double descuento) que
calcule el precio final de un producto en un e-commerce. La fórmula es:
    PrecioFinal = PrecioBase + (PrecioBase×Impuesto) - (PrecioBase×Descuento)
PrecioFinal = PrecioBase + (PrecioBase \times Impuesto) - (PrecioBase \times
Descuento)
Desde main(), solicita el precio base del producto, el porcentaje de impuesto
y el porcentaje de descuento, llama al método y muestra el precio final.
*/

    System.out.println("Ingrese el Precio del Producto: ");
    double PrecioBase = input.nextDouble();
    System.out.println("Ingrese el Procentaje de Impuesto: ");
    double Impuesto = input.nextDouble();
    System.out.println("Ingrese el Porecentaje de Descuento: ");
    double Descuento = input.nextDouble();

    System.out.println("El precio Final del Producto es de: $" +
CalculadoraPrecio(PrecioBase, Impuesto, Descuento));

}
static double CalculadoraPrecio(double PrecioBase, double Impuesto, double
Descuento){
    return PrecioBase + (PrecioBase*(Impuesto / 100)) -
(PrecioBase*(Descuento/100));
}

/*
9. Composición de funciones para calcular costo de envío y total de compra.
a. calcularCostoEnvio(double peso, String zona): Calcula el costo de envío
basado en la zona de envío (Nacional o Internacional) y el peso del paquete.
    Nacional: $5 por kg - Internacional: $10 por kg.
b. calcularTotalCompra(double precioProducto, double costoEnvio): Usa
calcularCostoEnvio para sumar el costo del producto con el costo de envío.
```

Desde main(), solicita el peso del paquete, la zona de envío y el precio del producto. Luego, muestra el total a pagar.

```
*/

System.out.println("Ingresa el Precio del Producto: ");
double precioProducto = input.nextDouble();
System.out.println("Ingresa el peso del Producto: ");
double pesoProducto = input.nextDouble();
input.nextLine();
System.out.println("Indique si el envio es Nacional o Internacional: ");
String zonaEnvio = input.nextLine().toLowerCase().trim();

double total= CalcTotalCompra(pesoProducto,zonaEnvio,precioProducto);
System.out.printf("El costo total del Envio es de: $%.2f\n",total);
}
```

```
static double CalcTotalCompra ( double peso, String zona,double precioProducto){
    double costoEnvio;
    if (zona.equalsIgnoreCase("Nacional")) {
        costoEnvio = peso * 5;
    } else if (zona.equalsIgnoreCase("Internacional")) {
        costoEnvio = peso * 10;
    } else {
        System.out.println("Debe seleccionar una Zona Valida");
        return 0;
    }
    return costoEnvio + precioProducto;
}
```

/\*  
10. Actualización de stock a partir de venta y recepción de productos.  
Crea un método actualizarStock(int stockActual, int cantidadVendida,int  
cantidadRecibida),  
que calcule el nuevo stock después de una venta y recepción de productos:

NuevoStock = StockActual - CantidadVendida + CantidadRecibida  
NuevoStock = CantidadVendida + CantidadRecibida

Desde main(), solicita al usuario el stock actual, la cantidad vendida y la cantidad recibida, y muestra el stock actualizado.

```
*/

int stockActual = input.nextInt();
input.nextLine();

System.out.println("Ingresa el Stock Vendido: ");
int stockVendido = input.nextInt();

System.out.println("Ingresa el Stock Ingresado: ");
int stockIngresado = input.nextInt();

int StockActual = actualizacionStock(stockActual, stockVendido,
stockIngresado);

System.out.printf("El Stock actual es de: %d remeras",StockActual);
```

```

}
static int actualizacionStock(int stockActual, int cantidadVendida,int
cantidadRecibida) {
    return (stockActual - cantidadVendida) + cantidadRecibida;

/*
11. Cálculo de descuento especial usando variable global.

Declara una variable global Ejemplo de entrada/salida: = 0.10. Luego, crea un
método calcularDescuentoEspecial(double precio)
que use la variable global para calcular el descuento especial del 10%.
Dentro del método, declara una variable local descuentoAplicado, almacena el valor
del descuento y muestra el precio final con descuento.
*/
    double precio = 234.45;
    calcularDescuentoEspecial(precio);
}
public static void calcularDescuentoEspecial(double precio){
    double descuentoAplicado = precio * descuentoEspecial;
    double precioFinal = precio - descuentoAplicado;
    System.out.printf("Descuento aplicado: $ %.2f\n", descuentoAplicado);
    System.out.printf("Precio final con descuento: $ %.2f\n", precioFinal);

}

```

/\*Arrays y Recursividad:

12. Modificación de un array de precios y visualización de resultados.  
Crea un programa que:

- Declare e inicialice un array con los precios de algunos productos.
- Muestre los valores originales de los precios.
- Modifique el precio de un producto específico.
- Muestre los valores modificados.

Conceptos Clave Aplicados:

- ✓ Uso de arrays (double[]) para almacenar valores.
- ✓ Recorrido del array con for-each para mostrar valores.
- ✓ Modificación de un valor en un array mediante un índice.
- ✓ Reimpresión del array después de la modificación.

```

*/
    double[] precios= {34.5,23.5,12.4,67.8,89.7};
    Arrays.stream(precios).forEach(System.out::println);

    precios[2]= 456.5;
    System.out.println(precios[2]);
    Arrays.stream(precios).forEach(System.out::println);

```

/\*
13. Impresión recursiva de arrays antes y después de modificar un elemento.

Crea un programa que:

- Declare e inicialice un array con los precios de algunos productos.
- Use una función recursiva para mostrar los precios originales.
- Modifique el precio de un producto específico.
- Use otra función recursiva para mostrar los valores modificados.

Conceptos Clave Aplicados:

- ✓ Uso de arrays (double[]) para almacenar valores.
- ✓ Recorrido del array con una función recursiva en lugar de un bucle.
- ✓ Modificación de un valor en un array mediante un índice.
- ✓ Uso de un índice como parámetro en la recursión para recorrer el array.

```
*/  
  
    double[] monto = {34.5,23.5,12.4,67.8,89.7};  
    Arrays.stream(precios).forEach(System.out::println);  
  
    monto[2] = 456.5;  
  
    recursividad(monto,2);  
} // Cierre main  
  
public static void recursividad (double[]array, int indice) {  
    if (indice < array.length) {  
        System.out.println(array[indice]);  
        recursividad(array, indice + 1);  
    }  
  
}  
  
} // Cierre Clase
```