

# Trabajo Practico N°1 de Programación II

Estudiante: Tobias Leiva

Unidad: Introducción a Java

Universidad Tecnológica Nacional

### **Estructuras Condicionales:**

#### 1. Verificación de Año Bisiesto.

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.

```
public class TP1 {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        int anio;  
  
        System.out.print("Ingrese el año: ");  
        anio = input.nextInt();  
  
        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)){  
            System.out.println("El " + anio + " es bisiesto");  
        } else {  
            System.out.println("El " + anio + " no es bisiesto");  
        }  
    }  
}
```

#### 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.

```
public class TP2 {  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        int num1, num2, num3;  
  
        System.out.print("Ingrese el Primer numero entero: ");  
        num1 = Integer.parseInt(input.nextLine());
```

```

        System.out.print("Ingrese el Segundo numero entero: ");
        num2 = Integer.parseInt(input.nextLine());

        System.out.print("Ingrese el Tercer numero entero: ");
        num3 = Integer.parseInt(input.nextLine());

        int mayor = num1;

        if (num2 > mayor) {
            mayor = num2;
        }
        if (num3 > mayor){
            mayor = num3;
        }
        System.out.println("El numero mayor de los tres es: " +
        mayor);
    }
}

```

3. Clasificación de Edad. 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"

```

public class TP3 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int edad;

        System.out.print("Ingrese su edad: ");
        edad = input.nextInt();
    }
}

```

```

        if (edad < 0) {
            System.out.println("Edad no valida");
        } else if (edad < 12){
            System.out.println("Niño");
        } else if (edad <= 17 ){
            System.out.println("Adolescente");
        } else if (edad <= 59 ){
            System.out.println("Adulto");
        } else {
            System.out.println("Adulto mayor");
        }
    }
}

```

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

```

public class TP4 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        double precio, resultado, descuento;
        char categoria;

        System.out.print("Ingrese el precio del producto: ");
        precio = Double.parseDouble(input.nextLine());

        System.out.println();
    }
}

```

```

System.out.println("Categoria A: 10% de descuento");
System.out.println("Categoria B: 15% de descuento");
System.out.println("Categoria C: 20% de descuento");

System.out.println();

System.out.print("Ingrese La categoria del producto: ");
categoria = input.nextLine().charAt(0);

switch (categoria) {
    case 'A', 'a' -> {
        descuento = ((double)10 / 100) * precio;
        resultado = precio - descuento;
        System.out.println("Precio original: "+ precio);
        System.out.println("Descuento aplicado, 10%: "+
descuento);
        System.out.println("Su precio final es de: "+
resultado);
    }
    case 'B', 'b' ->{

        descuento = ((double)15 / 100) * precio;
        resultado = precio - descuento;
        System.out.println("Precio original: "+ precio);
        System.out.println("Descuento aplicado, 15%: "+
descuento);
        System.out.println("Su precio final es de: "+
resultado);
    }
    case 'C', 'c' -> {
        descuento = ((double)20 / 100) * precio;
        resultado = precio - descuento;
        System.out.println("Precio original: "+ precio);

```

```

        System.out.println("Descuento aplicado, 20%: "+
descuento);

        System.out.println("Su precio final es de: "+
resultado);
    }

    default -> System.out.println("No se realizo ningun
descuento.");
}

}

}

```

### **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.

```

public class TP5 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int num = 1, acum = 0;

        while (num != 0) {

            System.out.print("Ingrese un numero(0 para terminar): ");
            num = input.nextInt();

            if (num == 0) {
                break;
            }

            if (num % 2 == 0) {
                acum += num;
            }

        }

        System.out.println("La suma de los numeros pares 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.

```
public class TP6 {  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        int positivo = 0, negativo = 0, cero = 0, num;  
  
        for (int i = 1; i < 11; i++) {  
            System.out.print("Ingrese el numero " + i + ": ");  
            num = Integer.parseInt(input.nextLine());  
  
            if (num > 0) {  
                positivo = positivo + 1;  
            } else if (num < 0) {  
                negativo++;  
            } else {  
                cero++;  
            }  
        }  
  
        System.out.println("Positivos: "+ positivo);  
        System.out.println("Negativos: "+ negativo);  
        System.out.println("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.

```
package tp7;

import java.util.Scanner;

public class TP7 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int nota;

        do {
            System.out.print("Ingrese una nota(0-10): ");
            nota = input.nextInt();
            if (nota >= 0 && nota <= 10) {
                System.out.println("Nota guardada");
            } else {
                System.out.println("Error, Ingrese una nota entre 0 y 10");
            }
        } 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:

$$\text{PrecioFinal} = \text{PrecioBase} + (\text{PrecioBase} \times \text{Impuesto}) - (\text{PrecioBase} \times \text{Descuento})$$



$\text{PrecioFinal} = \text{PrecioBase} + (\text{PrecioBase} \times \text{Impuesto}) - (\text{PrecioBase} \times \text{Descuento})$

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.

```
package tp8;

import java.util.Scanner;

public class TP8 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double precioBase, impuesto, descuento;

        System.out.print("Ingrese el precio base del producto: ");
        precioBase = Double.parseDouble(input.nextLine());

        System.out.print("Ingrese el impuesto en porcentaje: ");
        impuesto = Double.parseDouble(input.nextLine());

        System.out.print("Ingrese el descuento en porcentaje: ");
        descuento = Double.parseDouble(input.nextLine());

        double precioFinal = calcularPrecioFinal(impuesto, descuento,
precioBase);

        System.out.println(precioFinal);
    } //Cierre del main

    static double calcularPrecioFinal(double impuesto, double
descuento, double precioBase){
        double precioFinal;

        double impuestoD = impuesto / 100, descuentoD = descuento /
100;

        precioFinal = precioBase + (precioBase * impuestoD) -
(precioBase * descuentoD);
    }
}
```

```

        return precioFinal;
    }
} // Cierre de la clase

```

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.

```

import java.util.Scanner;

public class TP9 {

    static double calcularCostoEnvio(double peso, String zona){
        double costoEnvio = 0;

        if (zona.equalsIgnoreCase("Nacional")) {
            costoEnvio = 5 * peso;
        } else if (zona.equalsIgnoreCase("Internacional")) {
            costoEnvio = 10 * peso;
        }

        return costoEnvio;
    }

    static double calcularTotalCompra(double precio, double peso,
String zona){
        return precio + calcularCostoEnvio(peso, zona);
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
    }
}

```

```

        double peso, precio;
        String zona;

        System.out.print("Ingrese el peso del paquete en kg: ");
        peso = Double.parseDouble(input.nextLine());

        System.out.print("Ingrese la zona de envio(Nacional o
Internacional): ");
        zona = input.nextLine();

        System.out.print("Ingrese el precio del producto: ");
        precio = Double.parseDouble(input.nextLine());

        double total = calcularTotalCompra(precio, peso, zona);
        System.out.println(total);
    }//Cierre de main

```

```

} //Cierre de clase

```

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:

$$\text{NuevoStock} = \text{StockActual} - \text{CantidadVendida} + \text{CantidadRecibida}$$

$$\text{NuevoStock} = \text{CantidadVendida} + \text{CantidadRecibida}$$

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

```

import java.util.Scanner;

public class TP10 {
    static int actualizarStock(int stockActual, int vendida, int
recibida){
        int nuevoStock;
        nuevoStock = stockActual - vendida + recibida;
        return nuevoStock;
    }
}

```

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int stockActual, recibida, vendida;

    System.out.print("Ingrese el stock actual: ");
    stockActual = Integer.parseInt(input.nextLine());

    System.out.print("Ingrese la cantidad recibida: ");
    recibida = Integer.parseInt(input.nextLine());

    System.out.print("Ingrese la cantidad vendida: ");
    vendida = Integer.parseInt(input.nextLine());

    int stockActualizado = actualizarStock(stockActual, vendida,
recibida);

    System.out.println("El nuevo stock del producto es: "+
stockActualizado);
}
}

```

#### 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.

```

import java.util.Scanner;

public class TP11 {

    static final double DESCUENTO = 0.10;

    static double calcularDescuentoEspecial(double precio){
        double descuentoAplicado = DESCUENTO * precio;

        System.out.println("el descuento especial es: "+
descuentoAplicado);
    }
}

```

```

        return precio - descuentoAplicado;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingrese el precio del producto: ");
        double precio = input.nextInt();

        double precioFinal = calcularDescuentoEspecial(precio);
        System.out.println("El precio final con descuento es: "+
precioFinal);
    }
}

```

### **Arrays y Recursividad:**

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

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

```

package tp12;

public class TP12 {

    public static void main(String[] args) {
        double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
        System.out.println("Precios originales");
        for (double precio : precios) {
            System.out.println("Precio: $" + precio);
        }
        System.out.println();
        precios[1] = 10.9;
    }
}

```

```

        System.out.println("Precios Modificados");
        for (double precio : precios) {
            System.out.println("Precio: $" + precio);
        }
    }
}

```

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

Crea un programa que:

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

```

package tp13;

public class TP13 {

    static void mostrarPrecios(double[] precios){
        for (double precio : precios) {
            System.out.println("Precio: $" + precio);
        }
    }

    public static void main(String[] args) {

        double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

        System.out.println("Precios originales");
        mostrarPrecios(precios);

        System.out.println();

        precios[3] = 20.5;
    }
}

```

```
        System.out.println("Precios Modificados");  
        mostrarPrecios(precios);  
    }  
  
}
```

GitHub: <https://github.com/Tobias-L7/Trabajos-Practicos-P2.git>