



PROGRAMACIÓN II

Trabajo Práctico 2: Programación Estructurada

Alumno: Marianela Guerrero.

Fecha de entrega: 18/08/2025

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.

```
1  /*
2  3  public static void main(String[] args) {
4  5      // TODO code application logic here
6  7      Scanner sc = new Scanner(System.in);
7  8
8  9      System.out.print("Ingresa el año: ");
9 10      int year = sc.nextInt();
10 11
11 12      if ((year%4 == 0 && year%100 != 0) || year%100 == 0) {
12 13          System.out.print("El año " + year + " es bisiesto.");
13 14      } else {
14 15          System.out.print("El año " + year + " no es bisiesto.");
15 16      }
16 17
17 18      sc.close();
18 19  }
19 20  }
```

holamundo.HolaMundo >

tput - HolaMundo (run) x

```
run:
Ingresa el año: 2024
El año 2024 es bisiesto.BUILD SUCCESSFUL (total time: 3 seconds)
```

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.

```
1  public static void main(String[] args) {
2  3      // TODO code application logic here
3  4      Scanner sc = new Scanner(System.in);
4  5
5  6      System.out.print("Ingresa el primer numero: ");
6  7      int num1 = sc.nextInt();
7  8      System.out.print("Ingresa el segundo numero: ");
8  9      int num2 = sc.nextInt();
9 10      System.out.print("Ingresa el tercer numero: ");
10 11      int num3 = sc.nextInt();
11 12
12 13      if (num1 > num2 && num1 > num3) {
13 14          System.out.print("El numero " + num1 + " es el mayor.");
14 15      } else if (num2 > num1 && num2 > num3) {
15 16          System.out.print("El numero " + num2 + " es el mayor.");
16 17      } else {
17 18          System.out.print("El numero " + num3 + " es el mayor.");
18 19      }
19 20
20 21      sc.close();
21 22  }
```

holamundo.HolaMundo > main > if (num1 > num2 && num1 > num3) else if (num2 > num1 && num2 > num3) else >

tput - HolaMundo (run) x

```
run:
Ingresa el primer numero: 3
Ingresa el segundo numero: 2
Ingresa el tercer numero: 1
El numero 3 es el mayor.BUILD SUCCESSFUL (total time: 3 seconds)
```

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:

```

1
2      System.out.print("Ingresa la edad: ");
3      int edad = sc.nextInt();
4
5      if (edad < 12 ) {
6          System.out.print("Niño.");
7      } else if (edad <= 17 ){
8          System.out.print("Adolescente");
9      } else if (edad <= 59 ){
10         System.out.print("Adulto");
11     } else {
12         System.out.print("Adulto mayor");
13     }
14
15     sc.close();
16 }
17
18 }

```

holamundo.HolaMundo > main > if (edad < 12) else if (edad <= 17) else if (edad <= 59) >

Input - HolaMundo (run) X

```

run:
Ingresa la edad: 25
AdultoBUILD SUCCESSFUL (total time: 3 seconds)

```

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

```

1      System.out.print("Ingresa precio: ");
2      double precio = sc.nextDouble();
3      System.out.print("Ingresa categoria: ");
4      char categoria = sc.next().toUpperCase().charAt(0);
5      double total = 0;
6
7      if (categoria == 'A') {
8          total = precio - (precio*0.1);
9          System.out.print("Precio original: " + precio + "\nDescuento: " + (precio*0.1) + "\nFinal: " + total );
10     } else if (categoria == 'B') {
11         total = precio - (precio*0.15);
12         System.out.print("Precio original: " + precio + "\nDescuento: " + (precio*0.15) + "\nFinal: " + total );
13     } else {
14         total = precio - (precio*0.2);
15         System.out.print("Precio original: " + precio + "\nDescuento: " + (precio*0.2) + "\nFinal: " + total );
16     }
17
18     sc.close();
19 }

```

lamundo.HolaMundo > main > if (categoria == 'A') else if (categoria == 'B') >

- HolaMundo (run) X

```

run:
Ingresa precio: 100
Ingresa categoria: c
Precio original: 100.0
Descuento: 20.0
Final: 80.0BUILD SUCCESSFUL (total time: 5 seconds)

```

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.print("Ingresa num: ");
        int valorIng = sc.nextInt();
        int valorTot = 0;

        while (valorIng != 0){
            if(valorIng % 2 == 0){
                valorTot += valorIng ;
            }
            System.out.print("Ingresa num: ");
            valorIng = sc.nextInt();
        }
        System.out.println("La suma de los num es : " + valorTot);
        sc.close();
    }
}

```

```

lamundo.HolaMundo > main > while (valorIng != 0) > if (valorIng % 2 == 0) >
t - HolaMundo (run) x
run:
Ingresa num: 2
Ingresa num: 4
Ingresa num: 5
Ingresa num: 1
Ingresa num: 2
Ingresa num: 0
La suma de los num es : 8
BUILD SUCCESSFUL (total time: 10 seconds)

```

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.

```

        int numPos = 0;
        int numNeg = 0;
        int numCero = 0;

        for(int i = 0; i < 10; i++){
            System.out.print("Ingrese un numero");
            int numeroIngresado = sc.nextInt();
            if(numeroIngresado == 0) {
                numCero += 1;
            } else if (numeroIngresado > 0){
                numPos += 1;
            } else {
                numNeg += 1;
            }
        }

        System.out.print("Se cargaron:\n" + numPos + " positivos\n" + numNeg + " negativos\n" + numCero + " ceros\n");
    }
}

```

```

lamundo.HolaMundo > main >
t - HolaMundo (run) x
run:
Ingrese un numero1
Ingrese un numero3
Ingrese un numero5
Ingrese un numero-1
Ingrese un numero-3
Ingrese un numero-6
Ingrese un numero0
Ingrese un numero0
Ingrese un numero0
Ingrese un numero3
Se cargaron:
4 positivos
3 negativos
3 ceros

```

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.

```

3 // TODO code application logic here
Scanner sc = new Scanner(System.in);
int nota;

do {
    System.out.print("Ingresa nota: ");
    nota = sc.nextInt();
    if (nota < 0 || nota > 10) {
        System.out.println("Nota invalida");
    }
} while (nota < 0 || nota > 10);

System.out.println("Nota valida: " + nota);

sc.close();
}

```

holamundo.HolaMundo > main >

tput - HolaMundo (run) x

```

run:
Ingresa nota: 12
Nota invalida
Ingresa nota: -5
Nota invalida
Ingresa nota: 6
Nota valida: 6
BUILD SUCCESSFUL (total time: 8 seconds)

```

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:

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

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.

```

1 public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
    double imp = impuesto / 100;
    double desc = descuento / 100;
    double precioFinal = precioBase + (precioBase * imp) - (precioBase * desc);
    return precioFinal;
}

1 public static void main(String[] args) {
    // TODO code application logic here
    Scanner sc = new Scanner(System.in);

    System.out.print("Precio: ");
    double precioBase = sc.nextDouble();
    System.out.print("Impuesto: ");
    double impuesto = sc.nextDouble();
    System.out.print("Descuento: ");
    double descuento = sc.nextDouble();

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

    System.out.println("Precio base: $" + precioBase);
    System.out.println("Impuesto aplicado: " + impuesto + "%");
    System.out.println("Descuento aplicado: " + descuento + "%");
    System.out.println("Precio final: $" + precioFinal);
}

```

lamundo.HolaMundo > calcularPrecioFinal >

- HolaMundo (run) x

```

run:
Precio: 100
Impuesto: 21
Descuento: 10
Precio base: $100.0
Impuesto aplicado: 21.0%
Descuento aplicado: 10.0%
Precio final: $111.0
BUILD SUCCESSFUL (total time: 11 seconds)

```

9. Composición de funciones para calcular costo de envío y total de compra.

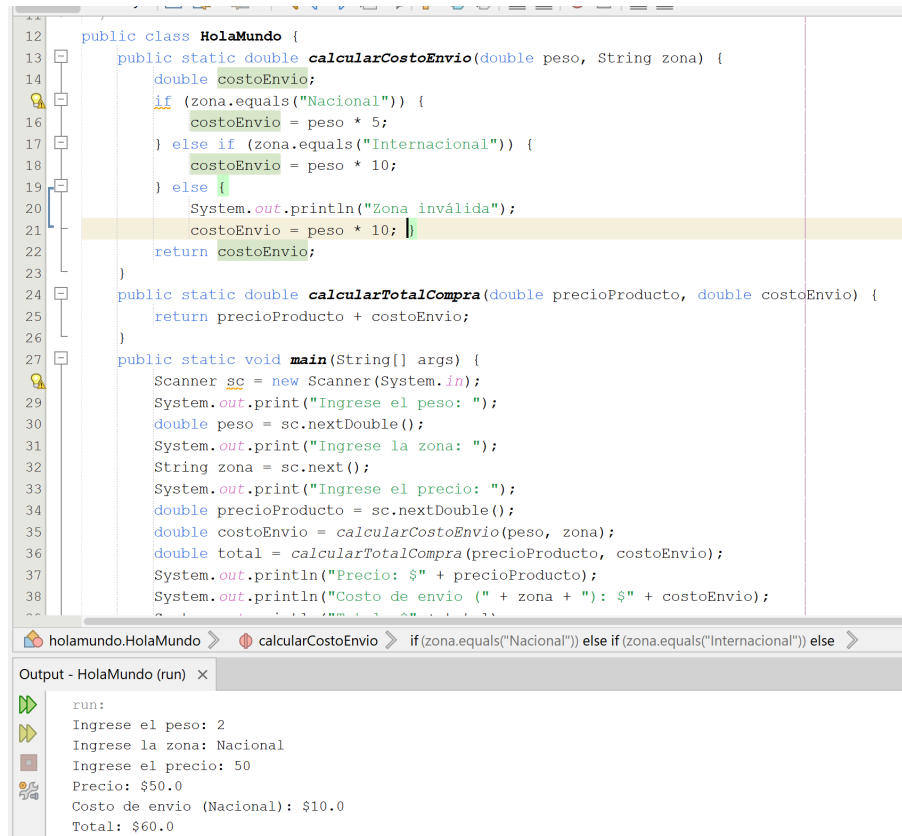
a. `calcularCostoEnvio(double peso, String zona)`: Calcula el costo de envío basado en la zona de envío (Nacional o Internacional) y el peso del paquete.

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.



```
12 public class HolaMundo {
13     public static double calcularCostoEnvio(double peso, String zona) {
14         double costoEnvio;
15         if (zona.equals("Nacional")) {
16             costoEnvio = peso * 5;
17         } else if (zona.equals("Internacional")) {
18             costoEnvio = peso * 10;
19         } else {
20             System.out.println("Zona inválida");
21             costoEnvio = peso * 10;
22         }
23         return costoEnvio;
24     }
25     public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
26         return precioProducto + costoEnvio;
27     }
28     public static void main(String[] args) {
29         Scanner sc = new Scanner(System.in);
30         System.out.print("Ingrese el peso: ");
31         double peso = sc.nextDouble();
32         System.out.print("Ingrese la zona: ");
33         String zona = sc.next();
34         System.out.print("Ingrese el precio: ");
35         double precioProducto = sc.nextDouble();
36         double costoEnvio = calcularCostoEnvio(peso, zona);
37         double total = calcularTotalCompra(precioProducto, costoEnvio);
38         System.out.println("Precio: $" + precioProducto);
39         System.out.println("Costo de envio (" + zona + "): $" + costoEnvio);
40         System.out.println("Total: $" + total);
41     }
42 }
```

Output - HolaMundo (run)

```
run:
Ingrese el peso: 2
Ingrese la zona: Nacional
Ingrese el precio: 50
Precio: $50.0
Costo de envio (Nacional): $10.0
Total: $60.0
```

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:

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

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

```

public class HolaMundo {
    public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
        int nuevoStock = stockActual - cantidadVendida + cantidadRecibida;
        return nuevoStock;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el stock: ");
        int stockActual = sc.nextInt();
        System.out.print("Ingrese la cantidad vendida: ");
        int cantidadVendida = sc.nextInt();
        System.out.print("Ingrese la cantidad recibida: ");
        int cantidadRecibida = sc.nextInt();

        int stockFinal = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);

        System.out.println("Stock actual: " + stockActual);
        System.out.println("Cantidad vendida: " + cantidadVendida);
        System.out.println("Cantidad recibida: " + cantidadRecibida);
        System.out.println("Stock actualizado: " + stockFinal);

        sc.close();
    }
}

```

holamundo.HolaMundo > main >

put - HolaMundo (run) x

```

run:
Ingrese el stock: 50
Ingrese la cantidad vendida: 20
Ingrese la cantidad recibida: 30
Stock actual: 50
Cantidad vendida: 20
Cantidad recibida: 30
Stock actualizado: 60
BUILD SUCCESSFUL (total time: 8 seconds)

```

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.

```

public class HolaMundo {
    public static double descuento = 0.2;

    public static void calcularDescuentoEspecial(double precio) {
        double descuentoAplicado = precio * descuento;
        double precioFinal = precio - descuentoAplicado;

        System.out.println("Precio original: $" + precio);
        System.out.println("Descuento: $" + descuentoAplicado);
        System.out.println("Precio final: $" + precioFinal);
    }

    public static void main(String[] args) {
        double precioProducto = 200.0;
        calcularDescuentoEspecial(precioProducto);
    }
}

```

holamundo.HolaMundo > descuento >

put - HolaMundo (run) x

```

run:
Precio original: $200.0
Descuento: $40.0
Precio final: $160.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

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

```
public static void main(String[] args) {
    double[] precios = {199.99, 299.5, 149.75, 300.0, 399.0, 89.99};

    System.out.println("Precios originales:");
    for (int i = 0; i < precios.length; i++) {
        System.out.println("Producto " + (i + 1) + ": $" + precios[i]);
    }
    precios[2] = 90.0;
    System.out.println("\nPrecios modificados:");
    for (int i = 0; i < precios.length; i++) {
        System.out.println("Producto " + (i + 1) + ": $" + precios[i]);
    }
}

holamundo.HolaMundo > main >

out - HolaMundo (run) x
run:
Precios originales:
Producto 1: $199.99
Producto 2: $299.5
Producto 3: $149.75
Producto 4: $300.0
Producto 5: $399.0
Producto 6: $89.99

Precios modificados:
Producto 1: $199.99
Producto 2: $299.5
Producto 3: $90.0
Producto 4: $300.0
Producto 5: $399.0
Producto 6: $89.99
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

```
public static void mostrarPrecios(double[] precios, int indice) {
    if (indice >= precios.length) {
        return;
    }
    System.out.println("Precio: $" + precios[indice]);
    mostrarPrecios(precios, indice + 1);
}

public static void main(String[] args) {
    double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
    System.out.println("Precios:");
    mostrarPrecios(precios, 0);
    precios[2] = 129.99;
    System.out.println("Precios modificados:");
    mostrarPrecios(precios, 0);
}

holamundo.HolaMundo > main >

Output - HolaMundo (run) x
run:
Precios:
Precio: $199.99
Precio: $299.5
Precio: $149.75
Precio: $399.0
Precio: $89.99
Precios modificados:
Precio: $199.99
Precio: $299.5
Precio: $129.99
Precio: $399.0
Precio: $89.99
BUILD SUCCESSFUL (total time: 0 seconds)
```