

PROGRAMACIÓN II Trabajo Práctico 2: Programación Estructurada

Nombre: Farias, Gustavo

Comisión: M2025-13

Matrícula: 101662

Repositorio GitHub:

https://github.com/Lucenear/UTN-TUPaD-TPs/tree/main/Programacion/Programacion_II

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.

```
package trabajo_practico_ii;

import java.util.Scanner;

public class consigna_1 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.print("Ingrese un año: ");
        int anio = scan.nextInt();

        boolean esBisiesto = false;

        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
            esBisiesto = true;
        }

        if (esBisiesto) {
            System.out.println("El año " + anio + " es bisiesto.");
        } else {
            System.out.println("El año " + anio + " no es bisiesto.");
        }
    }
}
```

```

7   package trabajo_practico_ii;
8
9   import java.util.Scanner;
10
11  public class consigna_1 {
12      public static void main(String[] args) {
13          Scanner scan = new Scanner(System.in);
14
15          System.out.print("Ingrese un año: ");
16          int anio = scan.nextInt();
17
18          boolean esBisiesto = false;
19
20          if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
21              esBisiesto = true;
22          }
23
24          if (esBisiesto) {
25              System.out.println("El año " + anio + " es bisiesto.");
26          } else {
27              System.out.println("El año " + anio + " no es bisiesto.");
28          }
29      }
30  }

```

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.

```

package trabajo_practico_ii;

import java.util.Scanner;

public class consigna_2 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.print("Ingrese el primer número: ");
        int num1 = Integer.parseInt(scan.nextLine());

        System.out.print("Ingrese el segundo número: ");
        int num2 = Integer.parseInt(scan.nextLine());

        System.out.print("Ingrese el tercer número: ");
        int num3 = Integer.parseInt(scan.nextLine());

        int mayor = num1;

        if (num2 > mayor) {
            mayor = num2;
        }
    }
}

```

```

        if (num3 > mayor) {
            mayor = num3;
        }

        System.out.println("El número mayor es: " + mayor);
    }
}

```



```

7      package trabajo_practico_ii;
8
9      import java.util.Scanner;
10
11     public class consigna_2 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14
15         System.out.print("Ingrese el primer número: ");
16         int num1 = Integer.parseInt(scan.nextLine());
17
18         System.out.print("Ingrese el segundo número: ");
19         int num2 = Integer.parseInt(scan.nextLine());
20
21         System.out.print("Ingrese el tercer número: ");
22         int num3 = Integer.parseInt(scan.nextLine());
23
24         int mayor = num1;
25
26         if (num2 > mayor) {
27             mayor = num2;
28         }
29
30         if (num3 > mayor) {
31             mayor = num3;
32         }
33
34         System.out.println("El número mayor es: " + mayor);
35     }
36 }

```

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"

```

package trabajo_practico_ii;

import java.util.Scanner;

public class consigna_3 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.print("Ingrese su edad: ");
        int edad = Integer.parseInt(scan.nextLine());

        String etapa;

        if (edad < 12) {
            etapa = "Niño";
        } else if (edad >= 12 && edad <= 17) {
            etapa = "Adolescente";
        } else if (edad >= 18 && edad <= 59) {
            etapa = "Adulto";
        } else if (edad >= 60) {
            etapa = "Adulto mayor";
        } else {
            etapa = "Edad no valida";
        }

        if (edad >= 0) {
            System.out.println("Usted es: " + etapa);
        } else {
            System.out.println("Error: La edad no puede ser negativa.");
        }
    }
}

```

```

7  package trabajo_practico_ii;
8
9  import java.util.Scanner;
10
11 public class consigna_3 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14
15         System.out.print("Ingrese su edad: ");
16         int edad = Integer.parseInt(scan.nextLine());
17
18         String etapa;
19
20         if (edad < 12) {
21             etapa = "Niño";
22         } else if (edad >= 12 && edad <= 17) {
23             etapa = "Adolescente";
24         } else if (edad >= 18 && edad <= 59) {
25             etapa = "Adulto";
26         } else if (edad >= 60) {
27             etapa = "Adulto mayor";
28         } else {
29             etapa = "Edad no valida";
30         }
31
32         if (edad >= 0) {
33             System.out.println("Usted es: " + etapa);
34         } else {
35             System.out.println("Error: La edad no puede ser negativa.");
36         }
37     }
38 }

```

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.

```

7 package trabajo_practico_ii;
8
9 import java.util.Scanner;
10
11 public class consigna_4 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14
15         System.out.print("Ingrese el precio del producto: ");
16         double precioOriginal = Double.parseDouble(scan.nextLine());
17
18         System.out.print("Ingrese la categoria (A, B o C): ");
19         String categoria = scan.nextLine().trim().toUpperCase();
20
21         if (!categoria.equals("A") && !categoria.equals("B") && !categoria.equals("C")) {
22             System.out.println("Error: Categoria no valida. Debe ser A, B o C");
23             return;
24         }
25
26         double porcentajeDescuento;
27
28         switch (categoria) {
29             case "A":
30                 porcentajeDescuento = 10;
31                 break;
32             case "B":
33                 porcentajeDescuento = 15;
34                 break;
35             case "C":
36                 porcentajeDescuento = 20;
37                 break;
38             default:
39                 porcentajeDescuento = 0;
40         }
41
42         double descuento = precioOriginal * (porcentajeDescuento / 100);
43         double precioFinal = precioOriginal - descuento;
44
45         System.out.println("Precio original: $" + String.format("%.2f", precioOriginal));
46         System.out.println("Descuento aplicado: " + porcentajeDescuento + "% ($" + String.format("%.2f", descuento) + ")");
47         System.out.println("Precio final: $" + String.format("%.2f", precioFinal));
48     }
49 }

```

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.

```

7   package trabajo_practico_ii;
8
9   import java.util.Scanner;
10
11  public class consigna_5 {
12      public static void main(String[] args) {
13          Scanner scan = new Scanner(System.in);
14          int numero;
15          int sumaPares = 0;
16
17          System.out.println("Ingrese numeros enteros. Ingrese 0 para terminar.");
18
19          while (true) {
20              System.out.print("Ingrese un numero: ");
21              numero = Integer.parseInt(scan.nextLine());
22
23              if (numero == 0) {
24                  break;
25              }
26
27              if (numero % 2 == 0) {
28                  sumaPares += numero;
29              }
30          }
31
32          System.out.println("La suma de los numeros pares ingresados es: " + sumaPares);
33      }
34  }

```

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.


```

7   package trabajo_practico_ii;
8
9   import java.util.Scanner;
10
11  public class consigna_6 {
12      public static void main(String[] args) {
13          Scanner scan = new Scanner(System.in);
14
15          int positivos = 0;
16          int negativos = 0;
17          int ceros = 0;
18
19          for (int i = 1; i <= 10; i++) {
20              System.out.print("Ingrese el numero " + i + ": ");
21              int numero = Integer.parseInt(scan.nextLine());
22
23              if (numero > 0) {
24                  positivos++;
25              } else if (numero < 0) {
26                  negativos++;
27              } else {
28                  ceros++;
29              }
30          }
31
32          System.out.println("Resultados:");
33          System.out.println("Positivos: " + positivos);
34          System.out.println("Negativos: " + negativos);
35          System.out.println("Ceros: " + ceros);
36      }
37  }

```

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.

```

7  package trabajo_practico_ii;
8
9  import java.util.Scanner;
10
11 public class consigna_7 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14         int nota;
15
16         do {
17             System.out.print("Ingrese una nota (0-10): ");
18             nota = Integer.parseInt(scan.nextLine());
19
20             if (nota < 0 || nota > 10) {
21                 System.out.println("Error: Nota invalida. Ingrese una nota entre 0 y 10.");
22             }
23         } while (nota < 0 || nota > 10);
24
25         System.out.println("Nota guardada correctamente.");
26     }
27 }

```

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.

```

7 package trabajo_practico_ii;
8
9 import java.util.Scanner;
10
11 public class consigna_8 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14
15         System.out.print("Ingrese el precio base del producto: ");
16         double precioBase = Double.parseDouble(scan.nextLine());
17
18         System.out.print("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): ");
19         double impuestoPorcentaje = Double.parseDouble(scan.nextLine());
20
21         System.out.print("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): ");
22         double descuentoPorcentaje = Double.parseDouble(scan.nextLine());
23
24         double precioFinal = calcularPrecioFinal(precioBase, impuestoPorcentaje, descuentoPorcentaje);
25
26         System.out.println("El precio final del producto es: " + String.format("%.2f", precioFinal));
27     }
28
29     /**
30      * Calcula el precio final de un producto aplicando impuesto y descuento.
31      *
32      * @param precioBase Precio original del producto
33      * @param impuesto Porcentaje de impuesto (ej: 10 para 10%)
34      * @param descuento Porcentaje de descuento (ej: 5 para 5%)
35      * @return Precio final después de impuesto y descuento
36      */
37     public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
38         double montoImpuesto = precioBase * (impuesto / 100);
39         double montoDescuento = precioBase * (descuento / 100);
40
41         double precioFinal = precioBase + montoImpuesto - montoDescuento;
42
43         return precioFinal;
44     }
45 }

```

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.

```

7 package trabajo_practico_ii;
8
9 import java.util.Scanner;
10
11 public class consigna_9 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14
15         System.out.print("Ingrese el precio del producto: ");
16         double precioProducto = Double.parseDouble(scan.nextLine());
17
18         System.out.print("Ingrese el peso del paquete en kg: ");
19         double peso = Double.parseDouble(scan.nextLine());
20
21         System.out.print("Ingrese la zona de envio (Nacional/Internacional): ");
22         String zona = scan.nextLine().trim().toUpperCase();
23
24         double costoEnvio = calcularCostoEnvio(peso, zona);
25
26         if (costoEnvio == -1) {
27             System.out.println("Error: Zona de envio no valida. Ingrese 'Nacional' o 'Internacional'.");
28             scan.close();
29             return;
30         }
31
32         double total = calcularTotalCompra(precioProducto, costoEnvio);
33
34         System.out.println("El costo de envio es: " + String.format("%.2f", costoEnvio));
35         System.out.println("El total a pagar es: " + String.format("%.2f", total));
36     }
37
38     /**
39      * Calcula el costo de envio segun el peso y la zona.
40      *
41      * @param peso Peso del paquete en kg
42      * @param zona Zona de envio: "NACIONAL" o "INTERNACIONAL"
43      * @return Costo de envio o -1 si la zona es invalida
44      */
45     public static double calcularCostoEnvio(double peso, String zona) {
46         if (zona.equals("NACIONAL")) {
47             return peso * 5;
48         } else if (zona.equals("INTERNACIONAL")) {
49             return peso * 10;
50         } else {
51             return -1;
52         }
53     }

```

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.

```

7 package trabajo_practico_ii;
8
9 import java.util.Scanner;
10
11 public class consigna_10 {
12     public static void main(String[] args) {
13         Scanner scan = new Scanner(System.in);
14
15         System.out.print("Ingrese el stock actual del producto: ");
16         int stockActual = Integer.parseInt(scan.nextLine());
17
18         System.out.print("Ingrese la cantidad vendida: ");
19         int cantidadVendida = Integer.parseInt(scan.nextLine());
20
21         System.out.print("Ingrese la cantidad recibida: ");
22         int cantidadRecibida = Integer.parseInt(scan.nextLine());
23
24         int nuevoStock = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
25
26         System.out.println("El nuevo stock del producto es: " + nuevoStock);
27     }
28
29     /**
30      * Actualiza el stock del producto.
31      *
32      * @param stockActual Stock actual del producto
33      * @param cantidadVendida Cantidad de unidades vendidas
34      * @param cantidadRecibida Cantidad de unidades recibidas
35      * @return Nuevo stock: stockActual - vendidos + recibidos
36      */
37     public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
38         return stockActual - cantidadVendida + cantidadRecibida;
39     }
40 }

```

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.

```

7 package trabajo_practico_ii;
8
9 import java.util.Scanner;
10
11 public class consigna_11 {
12
13     private static final double DESCUENTO_ESPECIAL = 0.10;
14
15     public static void main(String[] args) {
16         Scanner scan = new Scanner(System.in);
17
18         System.out.print("Ingrese el precio del producto: ");
19         double precio = Double.parseDouble(scan.nextLine());
20
21         calcularDescuentoEspecial(precio);
22     }
23
24     /**
25      * Calcula y muestra el descuento especial usando la variable estatica.
26      *
27      * @param precio Precio original del producto
28      */
29     public static void calcularDescuentoEspecial(double precio) {
30         double descuentoAplicado = precio * DESCUENTO_ESPECIAL;
31
32         double precioFinal = precio - descuentoAplicado;
33
34         System.out.println("El descuento especial aplicado es: " + descuentoAplicado);
35         System.out.println("El precio final con descuento es: " + precioFinal);
36     }
37 }

```

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.

```

7   package trabajo_practico_ii;
8
9   public class consigna_12 {
10      public static void main(String[] args) {
11
12          double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};
13
14          System.out.println("Precios originales:");
15          mostrarPrecios(precios);
16
17          precios[2] = 129.99;
18
19          System.out.println("Precios modificados:");
20          mostrarPrecios(precios);
21      }
22
23      /**
24       * Muestra los precios del array.
25       *
26       * @param precios Array de precios a mostrar
27       */
28      public static void mostrarPrecios(double[] precios) {
29          for (double precio : precios) {
30              System.out.println("Precio: $" + String.format("%.2f", precio));
31          }
32      }
33  }

```

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.

```

7 package trabajo_practico_ii;
8
9 public class consigna_13 {
10     public static void main(String[] args) {
11
12         double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};
13
14         System.out.println("Precios originales:");
15         imprimirPreciosRecursoivo(precios, 0);
16
17         precios[2] = 129.99;
18
19         System.out.println("Precios modificados:");
20         imprimirPreciosRecursoivo(precios, 0);
21     }
22
23     /**
24      * Funcion recursiva que imprime los precios del array.
25      *
26      * @param precios Array de precios a mostrar
27      * @param indice Indice actual del recorrido
28      */
29     public static void imprimirPreciosRecursoivo(double[] precios, int indice) {
30         if (indice >= precios.length) {
31             return;
32         }
33
34         System.out.println("Precio: $" + String.format("%.2f", precios[indice]));
35
36         imprimirPreciosRecursoivo(precios, indice + 1);
37     }
38 }

```