

# Programación II - TP2: Programación Estructurada

*Alumno: Romero, Abel Tomás (Comisión 5)*

## 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 import java.util.Scanner;
2
3 public class EjercicioUno {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         int year;
7
8         do {
9             System.out.print("Ingrese un año: ");
10            year = Integer.parseInt(scanner.nextLine());
11            if (year <= 0) System.out.println("ERROR: Año ingresado inválido.");
12        } while (year <= 0);
13
14        boolean esBisiesto = (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
15        System.out.println("El año " + year + (esBisiesto ? " es" : " NO es") + " bisiesto.");
16    }
```

```
run:
Ingrese un año: 2024
El año 2024 es bisiesto.
BUILD SUCCESSFUL (total time: 4 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 import java.util.Scanner;
2
3 public class EjercicioDos {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         System.out.print("Ingrese el primer número: ");
8         int num1 = Integer.parseInt(scanner.nextLine());
9         System.out.print("Ingrese el segundo número: ");
10        int num2 = Integer.parseInt(scanner.nextLine());
11        System.out.print("Ingrese el tercer número: ");
12        int num3 = Integer.parseInt(scanner.nextLine());
13
14        int max = mayor(num1, num2, num3);
15        if (num1 == num2 && num2 == num3) {
16            System.out.println("Los tres números son iguales.");
17        } else {
18            System.out.println("El mayor es: " + max);
19        }
20    }
21
22    public static int mayor(int a, int b, int c) {
23        if (a > b && a > c) return a;
24        if (b > a && b > c) return b;
25        return c;
26    }
27 }
```

```

run:
Ingrese el primer número: 8
Ingrese el segundo número: 12
Ingrese el tercer número: 5
El mayor es: 12
BUILD SUCCESSFUL (total time: 5 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:

- Menor de 12 años: "Niño".
- Entre 12 y 17 años: "Adolescente".
- Entre 18 y 59 años: "Adulto".
- 60 años o más: "Adulto mayor".

```

1  import java.util.Scanner;
2
3  public class EjercicioTres {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          int age;
7
8          do {
9              System.out.print("Ingrese su edad: ");
10             age = Integer.parseInt(scanner.nextLine());
11             if (age <= 0) System.out.println("La edad debe ser mayor a cero.");
12         } while (age <= 0);
13
14         if (age < 12) {
15             System.out.println("Eres un Niño.");
16         } else if (age >= 12 && age <= 17) {
17             System.out.println("Eres un Adolescente.");
18         } else if (age >= 18 && age <= 59) {
19             System.out.println("Eres un Adulto.");
20         } else {
21             System.out.println("Eres un Adulto mayor.");
22         }
23     }

```

```

run:
Ingrese su edad: 26
Eres un Adulto.
BUILD 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  import java.util.Scanner;
2
3  public class EjercicioCuatro {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          double precio, precioFinal = 0;
7          char categoria;
8
9          do {
10             System.out.print("Ingrese el precio del producto: ");
11             precio = Double.parseDouble(scanner.nextLine());
12             if (precio <= 0) {
13                 System.out.println("ERROR: EL precio debe ser mayor a cero.");
14             }
15         } while (precio <= 0);
16
17         do {
18             System.out.print("Ingrese la categoría del producto (A, B o C): ");
19             categoria = Character.toUpperCase(scanner.nextLine().charAt(0));
20             if (categoria != 'A' && categoria != 'B' && categoria != 'C') {
21                 System.out.println("ERROR: Categoría inválida.");
22             }
23         } while (categoria != 'A' && categoria != 'B' && categoria != 'C');
24
25         switch (categoria) {
26             case 'A' -> {
27                 System.out.println("Descuento aplicado: 10%");
28                 precioFinal = precio * 0.90;
29             }
30             case 'B' -> {
31                 System.out.println("Descuento aplicado: 15%");
32                 precioFinal = precio * 0.85;
33             }
34             case 'C' -> {
35                 System.out.println("Descuento aplicado: 20%");
36                 precioFinal = precio * 0.80;
37             }
38         }
39         System.out.println("Precio final: $" + precioFinal);
40     }
41 }

```

```

run:
Ingrese el precio del producto: 1000
Ingrese la categoría del producto (A, B o C): B
Descuento aplicado: 15%
Precio final: $850.0
BUILD SUCCESSFUL (total time: 5 seconds)

```

## 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.

```

1  import java.util.Scanner;
2
3  public class EjercicioCinco {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          int nums, sumaPares = 0;
7          boolean seguir = true;
8
9          while (seguir) {
10             System.out.print("Ingrese un número (0 para terminar): ");
11             nums = Integer.parseInt(scanner.nextLine());
12
13             if (nums == 0) {
14                 seguir = false;
15             } else if (nums % 2 == 0) {
16                 sumaPares += nums;
17             }
18         }
19         System.out.println("La suma de los números pares es: " + sumaPares);
20     }
21 }

```

```

run:
Ingrese un número (0 para terminar): 4
Ingrese un número (0 para terminar): 7
Ingrese un número (0 para terminar): 2
Ingrese un número (0 para terminar): 0
La suma de los números pares es: 6
BUILD SUCCESSFUL (total time: 5 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.

```

1  import java.util.Scanner;
2
3  public class EjercicioSeis {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          int nums, cantPositivos = 0, cantNegativos = 0, cantCeros = 0;
7
8          for (int i = 1; i <= 10; i++) {
9              System.out.print("Ingrese el número " + i + ": ");
10             nums = Integer.parseInt(scanner.nextLine());
11
12             if (nums > 0) {
13                 cantPositivos += 1;
14             } else if (nums < 0) {
15                 cantNegativos += 1;
16             } else {
17                 cantCeros += 1;
18             }
19         }
20
21         System.out.println("Resultados:");
22         System.out.println("Positivos: " + cantPositivos);
23         System.out.println("Negativos: " + cantNegativos);
24         System.out.println("Ceros: " + cantCeros);
25     }
26 }

```

```

run:
Ingrese el número 1: -5
Ingrese el número 2: 3
Ingrese el número 3: 0
Ingrese el número 4: -1
Ingrese el número 5: 6
Ingrese el número 6: 0
Ingrese el número 7: 9
Ingrese el número 8: -3
Ingrese el número 9: 4
Ingrese el número 10: -8
Resultados:
Positivos: 4
Negativos: 4
Ceros: 2
BUILD SUCCESSFUL (total time: 31 seconds)

```

## 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.

```

1  import java.util.Scanner;
2
3  public class EjercicioSiete {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          int nota;
7
8          do {
9              System.out.print("Ingrese una nota (0-10): ");
10             nota = Integer.parseInt(scanner.nextLine());
11             if (nota < 0 || nota > 10) {
12                 System.out.println("ERROR: Nota inválida.");
13             } else {
14                 System.out.println("Nota guardada correctamente.");
15             }
16         } while (nota < 0 || nota > 10);
17     }

```

```

run:
Ingrese una nota (0-10): 15
ERROR: Nota inválida.
Ingrese una nota (0-10): -2
ERROR: Nota inválida.
Ingrese una nota (0-10): 8
Nota guardada correctamente.
BUILD SUCCESSFUL (total time: 9 seconds)

```

## 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)**

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 import java.util.Scanner;
2
3 public class EjercicioOcho {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         double precioBase, impuesto, descuento, precioFinal;
7
8         do {
9             System.out.print("Ingrese el precio base del producto: $");
10            precioBase = scanner.nextDouble();
11            if (precioBase <= 0) System.out.println("ERROR: Precio base inválido.");
12        } while (precioBase <= 0);
13
14        do {
15            System.out.print("Ingrese el impuesto en porcentaje (ej: 10 para 10%): ");
16            impuesto = scanner.nextDouble();
17            if (impuesto <= 0) System.out.println("ERROR: Impuesto inválido.");
18        } while (impuesto <= 0);
19
20        do {
21            System.out.print("Ingrese el descuento por porcentaje (ej: 5 para 5%): ");
22            descuento = scanner.nextDouble();
23            if (descuento <= 0) System.out.println("ERROR: Descuento inválido.");
24        } while (descuento <= 0);
25
26        precioFinal = calcularPrecioFinal(precioBase, impuesto, descuento);
27        System.out.printf("El precio final del producto es: $%.1f%n", precioFinal);
28    }
29
30    public static double calcularPrecioFinal(double precioBase, double impu, double desc) {
31        double impuesto = impu / 100;
32        double descuento = desc / 100;
33
34        double precioFinal = precioBase + (precioBase * impuesto) - (precioBase * descuento);
35        return precioFinal;
36    }
37 }

```

```

run:
Ingrese el precio base del producto: $100
Ingrese el impuesto en porcentaje (ej: 10 para 10%): 10
Ingrese el descuento por porcentaje (ej: 5 para 5%): 5
El precio final del producto es: $105,0
BUILD SUCCESSFUL (total time: 4 seconds)

```

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

- `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.

- `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.

```

3 public class EjercicioNueve {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         double precioProducto, peso;
7         String zona;
8
9         do {
10             System.out.print("Ingrese el precio del producto: ");
11             precioProducto = Double.parseDouble(scanner.nextLine());
12             if (precioProducto <= 0) System.out.println("ERROR: Precio inválido.");
13         } while (precioProducto <= 0);
14
15         do {
16             System.out.print("Ingrese el peso del paquete en kg: ");
17             peso = Double.parseDouble(scanner.nextLine());
18             if (peso <= 0) System.out.println("ERROR: Peso inválido.");
19         } while (peso <= 0);
20
21         do {
22             System.out.print("Ingrese la zona de envío (Nacional/Internacional): ");
23             zona = scanner.nextLine().toUpperCase();
24             if (!zona.equals("NACIONAL") && !zona.equals("INTERNACIONAL")) {
25                 System.out.println("ERROR: Zona inválida.");
26             }
27         } while (!zona.equals("NACIONAL") && !zona.equals("INTERNACIONAL"));
28
29         double costoEnvio = calcularCostoEnvio(peso, zona);
30         System.out.printf("El costo de envío es: %.1f%n", costoEnvio);
31
32         double total = calcularTotalCompra(precioProducto, costoEnvio);
33         System.out.printf("El total a pagar es: %.1f%n", total);
34     }
35
36     public static double calcularCostoEnvio(double peso, String zona) {
37         final int ZONA_NACIONAL = 5, ZONA_INTERNACIONAL = 10;
38         return zona.equals("NACIONAL") ? peso * ZONA_NACIONAL : peso * ZONA_INTERNACIONAL;
39     }
40
41     public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
42         return precioProducto + costoEnvio;
43     }
44 }

```

```

run:
Ingrese el precio del producto: 50
Ingrese el peso del paquete en kg: 2
Ingrese la zona de envío (Nacional/Internacional): Nacional
El costo de envío es: 10,0
El total a pagar es: 60,0
BUILD SUCCESSFUL (total time: 8 seconds)

```

## 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**

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

```

1  import java.util.Scanner;
2
3  public class EjercicioDiez {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          System.out.print("Ingrese el stock actual del producto: ");
8          int stockActual = Integer.parseInt(scanner.nextLine());
9
10         System.out.print("Ingrese la cantidad vendida: ");
11         int cantidadVendida = Integer.parseInt(scanner.nextLine());
12
13         System.out.print("Ingrese la cantidad recibida: ");
14         int cantidadRecibida = Integer.parseInt(scanner.nextLine());
15
16         int nuevoStock = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
17         System.out.println("El nuevo stock del producto es: " + nuevoStock);
18     }
19
20     public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
21         return stockActual - cantidadVendida + cantidadRecibida;
22     }
23 }

```

```

run:
Ingrese el stock actual del producto: 50
Ingrese la cantidad vendida: 20
Ingrese la cantidad recibida: 30
El nuevo stock del producto es: 60
BUILD SUCCESSFUL (total time: 5 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.

```

1  import java.util.Scanner;
2
3  public class EjercicioOnce {
4      private static final double DESCUENTO_ESPECIAL = 0.10;
5
6      public static void main(String[] args) {
7          Scanner scanner = new Scanner(System.in);
8          double precio;
9
10         do {
11             System.out.print("Ingrese el precio del producto: ");
12             precio = Double.parseDouble(scanner.nextLine());
13             if (precio <= 0) System.out.println("ERROR: Precio inválido.");
14         } while (precio <= 0);
15
16         calcularDescuentoEspecial(precio);
17     }
18
19     public static void calcularDescuentoEspecial(double precio) {
20         double descuentoAplicado = precio * DESCUENTO_ESPECIAL;
21         double precioFinal = precio - descuentoAplicado;
22
23         System.out.println("El descuento especial aplicado es: " + descuentoAplicado);
24         System.out.println("El precio final con descuento es: " + precioFinal);
25     }
26 }

```



```
run:
Ingrese el precio del producto: 200
El descuento especial aplicado es: 20.0
El precio final con descuento es: 180.0
BUILD SUCCESSFUL (total time: 3 seconds)
```

## 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.

```
1 public class EjercicioDoce {
2     public static void main(String[] args) {
3         double[] precios = {199.99, 299.5, 149.75, 399, 89.99};
4
5         System.out.println("Precios originales:");
6         for (int i = 0; i < 2; i++) {
7             for (int j = 0; j < precios.length; j++) {
8                 System.out.println("Precio: $" + precios[j]);
9             }
10            precios[2] = 129.99;
11            if (i == 0) System.out.println("Precios modificados:");
12        }
13    }
```

```
run:
Precios originales:
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)
```

## 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.

```

1 public class EjercicioTrece {
2     public static void main(String[] args) {
3         double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
4
5         System.out.println("Precios originales:");
6         mostrarPreciosRecursoivo(precios, 0);
7
8         precios[2] = 129.99;
9
10        System.out.println("Precios modificados:");
11        mostrarPreciosRecursoivo(precios, 0);
12    }
13
14    public static void mostrarPreciosRecursoivo(double[] precios, int indice) {
15        if (indice >= precios.length) return;
16
17        System.out.println("Precio: $" + precios[indice]);
18
19        mostrarPreciosRecursoivo(precios, indice + 1);
20    }
21 }

```

```

run:
Precios originales:
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)

```

### Link del repo de GitHub:

<https://github.com/Tomu98/UTN-TUPaD-P2-TPs/tree/main/02%20Programaci%C3%B3n%20Estructurada>