

## PROGRAMACIÓN II

### Trabajo Práctico 2: Programación Estructurada

**Alumno:** Angelelli, Rodrigo Martin

**Comisión:** N.º 7

**Link a GitHub:** [https://github.com/roangelelli/Programacion2\\_TUPAD\\_Angelelli\\_Rodrigo](https://github.com/roangelelli/Programacion2_TUPAD_Angelelli_Rodrigo)

### CASO PRACTICO:

- 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.

```
Ejercicio1.java [-/A] x
Source History
1 package TP2_Ejercicios;
2
3 import java.util.Scanner;
4
5 public class Ejercicio1 {
6     public static void main(String[] args) {
7         Scanner input = new Scanner (System.in);
8         System.out.println("Ejercicio 1 - Verificacion de año Bisiesto");
9         System.out.println("Ingrese un año:");
10        int anioUsuario = input.nextInt();
11
12        if ((anioUsuario % 4==0 && anioUsuario%100!=0 )|| (anioUsuario % 400==0)) {
13            System.out.println("El año es bisiesto.");
14        } else {
15            System.out.println("El año NO es bisiesto.");
16        }
17    }
18 }

Output - TP2_Programacion2_AngelelliRodrigoM (run) x
run:
Ejercicio 1 - Verificacion de año Bisiesto
Ingrese un año:
2024
El año es bisiesto.
BUILD SUCCESSFUL (total time: 9 seconds)

Output - TP2_Programacion2_AngelelliRodrigoM (run) x
run:
Ejercicio 1 - Verificacion de año Bisiesto
Ingrese un año:
1929
El año NO 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  package TP2_Ejercicios;
2  import java.util.Scanner;
3  public class Ejercicio2 {
4      public static void main(String[] args) {
5          Scanner input = new Scanner (System.in);
6          System.out.println("Ejercicio 2 – Mayor de 3 números");
7          System.out.println("Por favor ingrese un número entero:");
8          int num1 = input.nextInt();
9          System.out.println("Por favor ingrese otro número entero:");
10         int num2 = input.nextInt();
11         System.out.println("Por favor ingrese otro número entero:");
12         int num3 = input.nextInt();
13
14         if (num1>=num2 && num1>=num3){
15             System.out.println("El número mayor es: "+num1);
16         } else if (num2>=num1 && num2>=num3){
17             System.out.println("El número mayor es: "+num2);
18         } else {
19             System.out.println("El número mayor es: "+num3);
20         }
21     }
22 }
```

Output - TP2\_Programacion2\_AngellelliRodrigoM (run) x

```
run:
Ejercicio 2 – Mayor de 3 números
Por favor ingrese un número entero:
12
Por favor ingrese otro número entero:
90
Por favor ingrese otro número entero:
36
El número mayor es: 90
BUILD SUCCESSFUL (total time: 8 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:

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

```
1 package TP2_Ejercicios;
2 import java.util.Scanner;
3 public class Ejercicio3 {
4     public static void main(String[] args) {
5         Scanner input = new Scanner (System.in);
6         System.out.println("Ejercicio 3 – Clasificación de Edad");
7         System.out.println("Por favor ingrese su edad:");
8         int edad = input.nextInt();
9
10
11         if (edad<12){
12             System.out.println("Eres un Niño");
13         } else if (edad>=12 && edad<=17){
14             System.out.println("Eres un Adolescente");
15         } else if (edad>=18 && edad<=59){
16             System.out.println("Eres un Adulto");
17         } else
18             System.out.println("Eres un Adulto mayor");
19     }
20 }
```

Output - TP2\_Programacion2\_AngellelliRodrigoM (run) ×

```
run:
Ejercicio 3 – Clasificación de Edad
Por favor ingrese su edad:
14
Eres un Adolescente
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 package TP2_Ejercicios;
2 import java.util.Scanner;
3 public class Ejercicio4 {
4
5     static double descuentoDiez = 10.0;
6     static double descuentoVeinte = 20.0;
7     static double descuentoTreinta = 30.0;
8
9     public static void main(String[] args) {
10         Scanner input = new Scanner (System.in);
11         System.out.println("Ejercicio 4 - Calculadora de Descuentos");
12         System.out.println("Ingrese el precio del producto:");
13         double precioOriginal = input.nextDouble();
14         System.out.println("Ingrese la categoria del producto (A,B o C)");
15         char categoria = input.next().charAt(0);
16
17         if (categoria == 'A'){
18             double precioFinal = precioOriginal - (precioOriginal*descuentoDiez/100);
19             System.out.println("Precio Original: $" + precioOriginal);
20             System.out.println("Porcentaje de descuento: " + descuentoDiez + " %.");
21             System.out.println("Precio final : $" + precioFinal);
22         } else if (categoria == 'B'){
23             double precioFinal = precioOriginal - (precioOriginal*descuentoVeinte/100);
24             System.out.println("Precio Original: $" + precioOriginal);
25             System.out.println("Porcentaje de descuento: " + descuentoVeinte + " %.");
26             System.out.println("Precio final : $" + precioFinal);
27         } else if (categoria == 'C'){
28             double precioFinal = precioOriginal - (precioOriginal*descuentoTreinta/100);
29             System.out.println("Precio Original: $" + precioOriginal);
30             System.out.println("Porcentaje de descuento: " + descuentoTreinta + " %.");
31             System.out.println("Precio final : $" + precioFinal);
32         } else
33             System.out.println("Opcion Invalida");
34     }
35 }
```

```
Ejercicio 4 - Calculadora de Descuentos
Ingrese el precio del producto:
1200
Ingrese la categoria del producto (A,B o C)
A
Precio Original: $1200.0
Porcentaje de descuento: 10.0 %.
Precio final : $1080.0
BUILD SUCCESSFUL (total time: 13 seconds)
```

```
Ejercicio 4 - Calculadora de Descuentos
Ingrese el precio del producto:
1200
Ingrese la categoria del producto (A,B o C)
B
Precio Original: $1200.0
Porcentaje de descuento: 20.0 %.
Precio final : $960.0
BUILD SUCCESSFUL (total time: 7 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 package TP2_Ejercicios;
2 import java.util.Scanner;
3 public class Ejercicio5 {
4
5     public static void main(String[] args) {
6         System.out.println("Ejercicio 5 - Suma de números pares");
7         Scanner input = new Scanner (System.in);
8         int pares = 0;
9         int impares = 0;
10
11         System.out.println("Ingrese un número (0 para terminar): ");
12         int numBase = input.nextInt();
13         while (numBase != 0) {
14             if (numBase % 2 == 0){
15                 pares++;
16             } else {impares++;
17             }
18             System.out.println("Ingrese otro número (0 para terminar):");
19             numBase = input.nextInt();
20         }
21         System.out.println("La cantidad de numeros pares fueron: "+pares);
22     }
23 }
```

```
run:
Ejercicio 5 - Suma de números pares
Ingrese un número (0 para terminar):
12
Ingrese otro número (0 para terminar):
15
Ingrese otro número (0 para terminar):
13
Ingrese otro número (0 para terminar):
16
Ingrese otro número (0 para terminar):
0
La cantidad de numeros pares fueron: 2
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.

```
1 package TP2_Ejercicios;
2 import java.util.Scanner;
3 public class Ejercicio6 {
4
5     public static void main(String[] args) {
6         System.out.println("Ejercicio 6 - Contador de Positivos, Negativos y Ceros");
7         Scanner input = new Scanner (System.in);
8         int numPositivos = 0, numNegativos=0, numCero=0;
9
10        for (int cont = 1; cont < 11; cont++){
11            System.out.println("Ingrese un número: ");
12            int numUsuario = input.nextInt();
13
14            if (numUsuario>0)
15                numPositivos++;
16            else if (numUsuario==0)
17                numCero++;
18            else
19                numNegativos++;
20        }
21        System.out.println("Resultados: ");
22        System.out.println("Números positivos: "+numPositivos+ ".");
23        System.out.println("Números negativos: "+numNegativos+ ".");
24        System.out.println("Ceros: "+numCero+ ".");
25    }
26 }
27 }
```

```
run:
Ejercicio 6 - Contador de Positivos, Negativos y Ceros
Ingrese un número:
12
Ingrese un número:
14
Ingrese un número:
14
Ingrese un número:
-2
Ingrese un número:
-4
Ingrese un número:
0
Ingrese un número:
0
Ingrese un número:
9
Ingrese un número:
-6
Ingrese un número:
-101
Resultados:
Números positivos: 4.
Números negativos: 4.
Ceros: 2.
BUILD SUCCESSFUL (total time: 26 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 package TP2_Ejercicios;
2 import java.util.Scanner;
3
4 public class Ejercicio7 {
5     public static void main(String[] args) {
6         System.out.println("Ejercicio 7 - Validacion de nota");
7         Scanner input = new Scanner (System.in);
8         int notaUsuario = 0;
9
10        do {
11            System.out.print("Ingrese una nota (0-10): ");
12            notaUsuario = input.nextInt();
13            if (notaUsuario < 0 || notaUsuario > 10) {
14                System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y 10.");
15            }
16        } while (notaUsuario < 0 || notaUsuario > 10);
17
18        System.out.println("Nota guardada correctamente.");
19    }
20 }
```

```
run:
Ejercicio 7 - Validacion de nota
Ingrese una nota (0-10): 12
Error: Nota inválida. Ingrese una nota entre 0 y 10.
Ingrese una nota (0-10): -5
Error: Nota inválida. Ingrese una nota entre 0 y 10.
Ingrese una nota (0-10): 8
Nota guardada correctamente.
BUILD SUCCESSFUL (total time: 18 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.

- a. 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 package TP2_Ejercicios;
2 import java.util.Scanner;
3
4 public class Ejercicio8 {
5     public static void main(String[] args) {
6         System.out.println("Ejercicio 8 - Precio Final");
7         Scanner input = new Scanner (System.in);
8         System.out.println("Ingrese el precio base del producto: ");
9         double precioBase = input.nextDouble();
10        System.out.println("Ingrese el impuesto en porcentaje (Ej. 10 para 10%: ");
11        double impuesto = input.nextDouble();
12        System.out.println("Ingrese el descuento en porcentaje (Ej. 5 para 5%: ");
13        double descuento = input.nextDouble();
14
15        System.out.println("El precio final de su producto es: " + calcularPrecioFinal(impuesto,descuento,precioBase));
16
17    } //Cierre de main
18
19    static double calcularPrecioFinal(double impuesto, double descuento, double precioBase) {
20        double imp = impuesto / 100.0;
21        double desc = descuento / 100.0;
22        return precioBase + (precioBase*imp)-(precioBase*desc);
23    }
24 } //Cierre de funcion
25 }
```

```
run:
Ejercicio 8 - Precio Final
Ingrese el precio base del producto:
2800
Ingrese el impuesto en porcentaje (Ej. 10 para 10%):
20
Ingrese el descuento en porcentaje (Ej. 5 para 5%):
9
El precio final de su producto es: 3108.0
BUILD SUCCESSFUL (total time: 9 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.
  - i. Nacional: \$5 por kg
  - ii. 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.

```

1 package TP2_Ejercicios;
2 import java.util.Scanner;
3 public class Ejercicio9 {
4
5     public static void main(String[] args) {
6         System.out.println("Ejercicio 9 - Costo envio y total compra");
7         Scanner input = new Scanner (System.in);
8         System.out.println("Ingrese el precio del producto: ");
9         double precioProducto = input.nextDouble();
10        System.out.println("Ingrese el peso del producto: ");
11        double pesoProducto = input.nextDouble();
12        System.out.println("Ingrese la zona de envio (Nacional/Internacional): ");
13        String zonaEnvio = input.next();
14
15        double totalEnvio = costoEnvio(pesoProducto,zonaEnvio);
16        double precioFinal = costoTotal(precioProducto,totalEnvio);
17        System.out.println("El costo de su envio es: $ " +totalEnvio);
18        System.out.println("El total a pagar es: $" +precioFinal);
19
20    } // Cierre del main
21    static double costoEnvio(double pesoProducto, String zonaEnvio) {
22        double costo;
23        if ("Nacional".equalsIgnoreCase(zonaEnvio)){
24            costo = pesoProducto*5;
25        }else if ("Internacional".equalsIgnoreCase(zonaEnvio)){
26            costo =pesoProducto*10;
27        }else {
28            costo = 0;
29        }
30        return costo;
31    }
32    //Cierre funcion
33
34    static double costoTotal (double precioProducto, double costoEnvio){
35        double costoTotal;
36        return costoTotal = precioProducto+costoEnvio;
37    }
38
39 } //Cierre de class

```

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:

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

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

```

1 package TP2_Ejercicios;
2 import java.util.Scanner;
3 public class Ejercicio10 {
4     public static void main(String[] args) {
5         System.out.println("Ejercicio 10 - Stock");
6         Scanner input = new Scanner (System.in);
7
8         System.out.println("Ingrese el stock actual: ");
9         int stockActual = input.nextInt();
10        System.out.println("Ingrese la cantidad vendida: ");
11        int cantVendida = input.nextInt();
12        System.out.println("Ingrese la cantidad recibida: ");
13        int cantRecibida = input.nextInt();
14        int nuevoStock = actualizarStock(stockActual,cantVendida,cantRecibida);
15
16        System.out.println(" El nuevo stock de su producto es: "+nuevoStock);
17
18    } //cierre main
19
20    static int actualizarStock (int stockActual, int cantVendida, int cantRecibida) {
21        int nuevoStock = (stockActual - cantVendida)+cantRecibida;
22        return nuevoStock;
23    } // cierre de funcion
24
25 } //cierre class

```

```

run:
Ejercicio 10 - Stock
Ingrese el stock actual:
19000
Ingrese la cantidad vendida:
3500
Ingrese la cantidad recibida:
6780
El nuevo stock de su producto es: 22280
BUILD SUCCESSFUL (total time: 19 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  package TP2_Ejercicios;
2  import java.util.Scanner;
3  public class Ejercicio11 {
4      final static double PORCENTAJE_DESCUENTO = 0.10;
5
6      public static void main(String[] args) {
7          System.out.println("Ejercicio 11 - Descuento");
8          Scanner input = new Scanner (System.in);
9          System.out.println("Ingrese el precio del producto:");
10         double precioOriginal = input.nextDouble();
11         descuentoEspecial (precioOriginal);
12     }
13     static void descuentoEspecial (double precioOriginal){
14         double descuento = precioOriginal*PORCENTAJE_DESCUENTO;
15         double precioFinal = precioOriginal - descuento;
16         System.out.println("El descuento especial aplicado es: "+descuento);
17         System.out.println("El precio final con descuento es: $ "+precioFinal);
18     }
19 }

```

Output x

TP2\_Programacion2\_AngellelliRodrigoM (run) x TP2\_Programacion2\_AngellelliRodrigoM (run) #2 x

```

run:
Ejercicio 11 - Descuento
Ingrese el precio del producto:
12500
El descuento especial aplicado es: 1250.0
El precio final con descuento es: $ 11250.0
BUILD SUCCESSFUL (total time: 18 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 package TP2_Ejercicios;
2
3 public class Ejercicio12 {
4
5     public static void main(String[] args) {
6         System.out.println("Ejercicio 12 - Precios");
7         double [] precios = {199.99, 299.5, 149.76, 399.0, 89.88};
8         System.out.println("Precios originales: ");
9         for (double p:precios){
10             System.out.println("$" +p);
11         }
12
13         precios[2] = 129.99;
14
15         System.out.println("Precios modificados: ");
16         for (double p:precios){
17             System.out.println("$" +p);
18         }
19     }
20 }
21 }
```

Output - TP2\_Programacion2\_AngellelliRodrigoM (run) #2 x

```
run:
Ejercicio 12 - Precios
Precios originales:
$199.99
$299.5
$149.76
$399.0
$89.88
Precios modificados:
$199.99
$299.5
$129.99
$399.0
$89.88
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  package TP2_Ejercicios;
2  public class Ejercicio13 {
3      public static void main(String[] args) {
4      double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
5
6          System.out.println("Precios originales:");
7          imprimirRecursivo(precios, 0);
8
9          precios[2] = 129.99;
10
11         System.out.println("Precios modificados:");
12         imprimirRecursivo(precios, 0);
13     }
14
15     private static void imprimirRecursivo(double[] precios, int indice) {
16         if (indice == precios.length) {
17             return;
18         }
19         System.out.println("Precio: $" + precios[indice]);
20         imprimirRecursivo(precios, indice + 1);
21     }

```

Output - TP2\_Programacion2\_AngellelliRodrigoM (run) #2 ×

```

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)

```