

PROGRAMACIÓN II

Trabajo Práctico 2: Programación Estructurada

20/08/2025

Alumno:

- Federico Garcia Bengolea - feddericogarciaa@gmail.com
- Comisión: M2025-14
- Link GitHub: <https://github.com/FeddericoGarcia/utn-tupad-p2>

Profesores:

- **Profesor:** Alberto Cortez
- **Tutor:** Ramiro Hualpa

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 package javaapp.TP2;
3 import java.util.Scanner;
4
5 public class ejercicio1 {
6
7     public static void ejercicio() {
8         Scanner scanner = new Scanner(System.in);
9         System.out.println("-- AÑO BISIESTO --");
10        System.out.println("Ingresá el número de algun año: ");
11        int num = scanner.nextInt();
12        if (num%4==0 && num%100!=0 || num%400==0) {
13            System.out.println("¡El año es bisiesto!");
14        } else {
15            System.out.println("El año NO es bisiesto.");
16        }
17    }
18
19 }
```

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 javaapp.TP2;
2
3 import java.util.Scanner;
4
5 public class ejercicio2 {
6
7     public static void ejercicio(){
8         Scanner scanner = new Scanner(System.in);
9         System.out.println("-- EL MAYOR DE TRES --");
10        System.out.println("Ingresá el primer número:");
11        int num1 = scanner.nextInt();
12        System.out.println("Ingresá el segundo número:");
13        int num2 = scanner.nextInt();
14        System.out.println("Ingresá el tercer número:");
15        int num3 = scanner.nextInt();
16        int mayor = (num1 > num2) ? ((num1 > num3) ? num1 : num3) :
17                        ((num2 > num3) ? num2 : num3);
18        System.out.println("El número mayor es el "+ mayor );
19    }
20 }
```

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
2 package javaapp.TP2;
3 import java.util.Scanner;
4
5 public class ejercicio3 {
6
7     public static void ejercicio() {
8         Scanner scanner = new Scanner(System.in);
9         System.out.println("-- CLASIFICACIÓN DE EDAD --");
10        System.out.println("Ingresa tu edad: ");
11        int edad = scanner.nextInt();
12        if (edad > 0 && edad < 12) {
13            System.out.println("Eres un/a Niño/a");
14        } else if (edad >= 12 && edad <= 17) {
15            System.out.println("Eres Adolescente");
16        } else if (edad >= 18 && edad <= 59) {
17            System.out.println("Sos Adulto/a");
18        } else {
19            System.out.println("Eres un/a Adulto/a mayor");
20        }
21    }
22 }
```

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 javaapp.TP2;
2 import java.util.Scanner;
3
4 public class ejercicio4 {
5
6     public static void ejercicio(){
7         Scanner scanner = new Scanner(System.in);
8         double productPrice, productDiscount;
9         String categoryOption;
10        System.out.println("-- CALCULADORA DE DESCUENTO x CATEGORÍA --");
11        System.out.println("Ingresa el precio del producto: ");
12        productPrice = scanner.nextDouble();
13        scanner.nextLine();
14        System.out.println("Ingresa la categoría (puede ser A, B o C): ");
15        categoryOption = scanner.nextLine();
16        System.out.println("-----");
17        productDiscount = categoryOption.equalsIgnoreCase("A") ?
18            0.10 : (categoryOption.equalsIgnoreCase("B") ?
19            0.15 : (categoryOption.equalsIgnoreCase("C") ? 0.20 : 0));
20        System.out.println("- El precio del producto es $" + productPrice);
21        System.out.println("- El descuento aplicado es " + (productDiscount*100) + "%");
22        System.out.println("- El precio total es $" + (productPrice -
23            (productPrice * productDiscount)));
24    }
25 }
26 }
```

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.

```
1 package javaapp.TP2;
2 import java.util.Scanner;
3
4 public class ejercicio5 {
5
6     public static void ejercicio(){
7         Scanner scanner = new Scanner(System.in);
8         int num, amount = 0;
9         System.out.println("-- SUMA DE NÚMEROS PARES --");
10        System.out.println("Ingresa un número, o '0' para finalizar: ");
11        num = scanner.nextInt();
12        while(num != 0){
13            amount += (num % 2 == 0) ? num : 0;
14            System.out.println("Ingresa un número");
15            num = scanner.nextInt();
16        }
17        System.out.println("- La suma total de los pares es: " + amount);
18    }
19 }
20 }
```

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
2 package javaapp.TP2;
3 import java.util.Scanner;
4
5 public class ejercicio6 {
6
7     public static void ejercicio(){
8         Scanner scanner = new Scanner(System.in);
9         int num, count = 5, countPos = 0, countNeg = 0, countZero = 0;
10        System.out.println("-- CONTADOR POSITIVO, NEGATIVOS Y CEROS --");
11        System.out.println("Ingresa 10 números al azar: ");
12        while(count != 0){
13            System.out.println("Te quedan "+ count + " intentos");
14            num = scanner.nextInt();
15            countPos += (num > 0) ? num : 0;
16            countNeg += (num < 0) ? num : 0;
17            countZero += (num == 0) ? 1 : 0;
18            count--;
19        }
20        System.out.println("- Suma de números pares: "+ countPos);
21        System.out.println("- Suma de números impares: "+ countNeg);
22        System.out.println("- Suma de números ceros: "+ countZero);
23    }
24 }
```

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 javaapp.TP2;
2
3 import java.util.Scanner;
4
5 public class ejercicio7 {
6
7     public static void ejercicio(){
8         Scanner scanner = new Scanner(System.in);
9         int nota;
10        System.out.println("-- VALIDACIÓN DE NOTA --");
11        do{
12            System.out.println("Ingresa una nota entre 0 y 10: ");
13            nota = scanner.nextInt();
14            if (nota < 0 || nota > 10){
15                System.out.println("Error: Nota inválida.");
16            }
17        } while (nota < 0 || nota > 10);
18        System.out.println("Nota guardada");
19    }
20 }
```

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.

```
1
2 package javaapp.TP2;
3 import java.util.Scanner;
4
5 public class ejercicio8 {
6
7     public static void ejercicio(){
8         Scanner scanner = new Scanner(System.in);
9         double precio, impuesto, descuento;
10        System.out.println("-- CALCULAR PRECIO FINAL CON IMP. Y DESC. --");
11        System.out.println("Ingresa el precio del producto: ");
12        precio = scanner.nextDouble();
13        System.out.println("Ingresa el impuesto: ");
14        impuesto = scanner.nextDouble();
15        System.out.println("Ingresa el descuento: ");
16        descuento = scanner.nextDouble();
17        System.out.println("Precio final del producto es de $" +
18            calcularPrecioFinal(precio, impuesto, descuento));
19    }
20
21    public static double calcularPrecioFinal(
22        double precio,
23        double impuesto,
24        double descuento){
25        return precio + (precio*(impuesto/100)) - (precio*(descuento/100));
26    }
27 }
```

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.


```
1 package javaapp.TP2;
2
3 import java.util.Scanner;
4
5 public class ejercicio9 {
6
7     public static void ejercicio(){
8         Scanner scanner = new Scanner(System.in);
9         double peso, precio;
10        String zona;
11        System.out.println("-- LOGÍSTICA DE PAQUETES --");
12        System.out.println("Ingresa el precio del producto: ");
13        precio = scanner.nextDouble();
14        System.out.println("Ingresa el peso del producto en kg: ");
15        peso = scanner.nextDouble();
16        System.out.println("Ingresa la zona del envío (Nacional / Internacional): ");
17        scanner.nextLine();
18        zona = scanner.nextLine();
19        calcularTotalCompra(precio, peso, zona);
20    }
21
22    public static double calcularCostoEnvio(double peso, String zona){
23        double costoEnvio = zona.equalsIgnoreCase("nacional") ? 5 : 10;
24        return costoEnvio * peso;
25    }
26
27    public static void calcularTotalCompra(double precio, double peso, String zona){
28        double costoEnvio = calcularCostoEnvio(peso, zona);
29        System.out.println("El costo del envío es $" + (costoEnvio));
30        System.out.println("El precio total es $" + (costoEnvio+precio));
31    }
32 }
```

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.

```
1
2 package javaapp.TP2;
3 import java.util.Scanner;
4
5 public class ejercicio10 {
6
7     public static void ejercicio(){
8         Scanner scanner = new Scanner(System.in);
9         int stockActual, cantidadVendida, cantidadRecibida;
10        System.out.println("-- ACTUALIZAR STOCK --");
11        System.out.println("Ingresa el stock actual: ");
12        stockActual = scanner.nextInt();
13        System.out.println("Ingresa la cantidad vendida: ");
14        cantidadVendida = scanner.nextInt();
15        System.out.println("Ingresa la cantidad recibida: ");
16        cantidadRecibida = scanner.nextInt();
17        System.out.println("El nuevo stock del producto es: "+ actualizarStock(
18            stockActual, cantidadVendida, cantidadRecibida));
19    }
20
21    public static int actualizarStock(
22        int stockActual,
23        int cantidadVendida,
24        int cantidadRecibida){
25        return stockActual - cantidadVendida + cantidadRecibida;
26    }
27 }
```

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
2 package javaapp.TP2;
3 import java.util.Scanner;
4
5 public class ejerciciol1 {
6     static double descuento = 0.10;
7
8     public static void ejercicio(){
9         Scanner scanner = new Scanner(System.in);
10        double precio;
11        System.out.println("-- VARIABLE GLOBAL --");
12        System.out.println("Ingresa el precio del producto: ");
13        precio = scanner.nextDouble();
14        System.out.println("El precio con descuento es $" +
15            (precio - calcularDescuentoEspecial(precio)));
16    }
17
18    public static double calcularDescuentoEspecial(double precio){
19        double descuentoAplicado = precio * descuento;
20        System.out.println("El descuento aplicado es del " + descuentoAplicado + "%");
21        return descuentoAplicado;
22    }
23 }
```

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
2 package javaapp.TP2;
3
4 public class ejerciciol2 {
5
6     public static void ejercicio(){
7         System.out.println("-- LISTA DE PRECIOS --");
8         double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
9         System.out.println("Lista de precios: ");
10        mostrarPrecios(precios);
11        precios[2] = 129.99;
12        System.out.println("Lista de precios modificados: ");
13        mostrarPrecios(precios);
14    }
15
16    public static void mostrarPrecios(double[] precios) {
17        for (double precio : precios) {
18            System.out.println("Precio: $" + precio);
19        }
20    }
21 }
```

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
2 package javaapp.TP2;
3
4 public class ejercicio13 {
5
6     public static void ejercicio(){
7         System.out.println("-- LISTA DE PRECIOS --");
8         double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
9         System.out.println("Lista de precios: ");
10        imprimirRecursivo(precios, 0);
11        precios[2] = 129.99;
12        System.out.println("Lista de precios modificados: ");
13        imprimirRecursivo(precios, 0);
14    }
15
16    public static void imprimirRecursivo(double[] precios, int indice) {
17        if (indice < precios.length) {
18            System.out.println("Precio: $" + precios[indice]);
19            imprimirRecursivo(precios, indice + 1);
20        }
21    }
22 }
```