

## Trabajo Práctico 2: Programación Estructurada.

**Materia:** Programación 2

**Estudiante:** Etchegoyen Gabriel

**Enlace a repositorio en GitHub:** <https://github.com/Gabriel071185/UTN-TUPaD-P2>

### Objetivo General:

Desarrollar habilidades en programación estructurada en Java, abordando desde conceptos básicos como operadores y estructuras de control hasta temas avanzados como funciones, recursividad y estructuras de datos. Se busca fortalecer la capacidad de análisis y solución de problemas mediante un enfoque práctico,

### Caso Práctico:

Desarrollar los siguientes ejercicios en Java utilizando el paradigma de programación estructurada. Agrupados según el tipo de estructuras o conceptos aplicados.

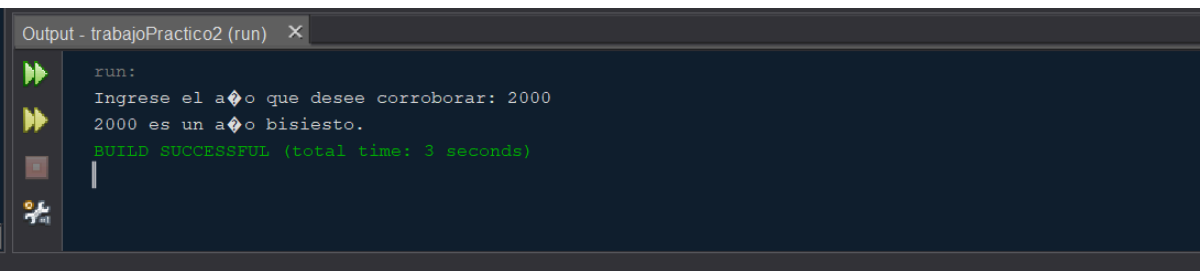
**Aclaración:** Para capturar el código de los ejercicios, se utilizó la función *Print to HTML* de NetBeans, que exporta el código con formato y resaltado de sintaxis para facilitar su presentación en este documento.

## 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.

```
7 package trabajopractico2;
8 import java.util.Scanner;
9 /*
10 Estructuras Condicionales:
11 1. Verificación de Año Bisiesto.
12 Escribe un programa en Java que solicite al usuario un año y determine si es
13 bisiesto. Un año es bisiesto si es divisible por 4, pero no por 100, salvo que sea
14 divisible por 400.
15 */
16 public class ejercicio1 {
17
18     public static void main(String[] args) {
19         Scanner entrada = new Scanner(System.in);
20         System.out.print("Ingrese el año que desee corroborar: ");
21         int anio = entrada.nextInt();
22         boolean esBisiesto = false;
23
24         if (anio % 4 == 0) {
25             if (anio % 100 != 0) {
26                 esBisiesto = true;
27             } else {
28                 if (anio % 400 == 0) {
29                     esBisiesto = true;
30                 }
31             }
32         }
33
34         if (esBisiesto) {
35             System.out.println(anio + " es un año bisiesto.");
36         } else {
37             System.out.println(anio + " no es un año bisiesto.");
38         }
39     }
40 }
```



```
Output - trabajoPractico2 (run) X
run:
Ingrese el año que desee corroborar: 2000
2000 es un año 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.

```
7 package trabajopractico2;
8 import java.util.Scanner;
9 /*
10 2. Determinar el Mayor de Tres Números.
11 Escribe un programa en Java que pida al usuario tres números enteros y
12 determine cuál es el mayor.
13 */
14 public class ejercicio2 {
15     public static void main(String[] args) {
16         Scanner entrada = new Scanner(System.in);
17         System.out.println("Ingrese el primer numero: ");
18         int uno = entrada.nextInt();
19         System.out.println("Ingrese el segundo numero: ");
20         int dos = entrada.nextInt();
21         System.out.println("Ingrese el tercer numero: ");
22         int tres = entrada.nextInt();
23
24         if (uno >= dos && uno >= tres) {
25             System.out.println("El mayor es: "+ uno);
26         } else if (dos >= uno && dos >= tres) {
27             System.out.println("El mayor es: "+ dos);
28         } else {
29             System.out.println("El mayor es: "+ tres);
30         }
31     }
32 }
33 }
```

Output - trabajoPractico2 (run) X

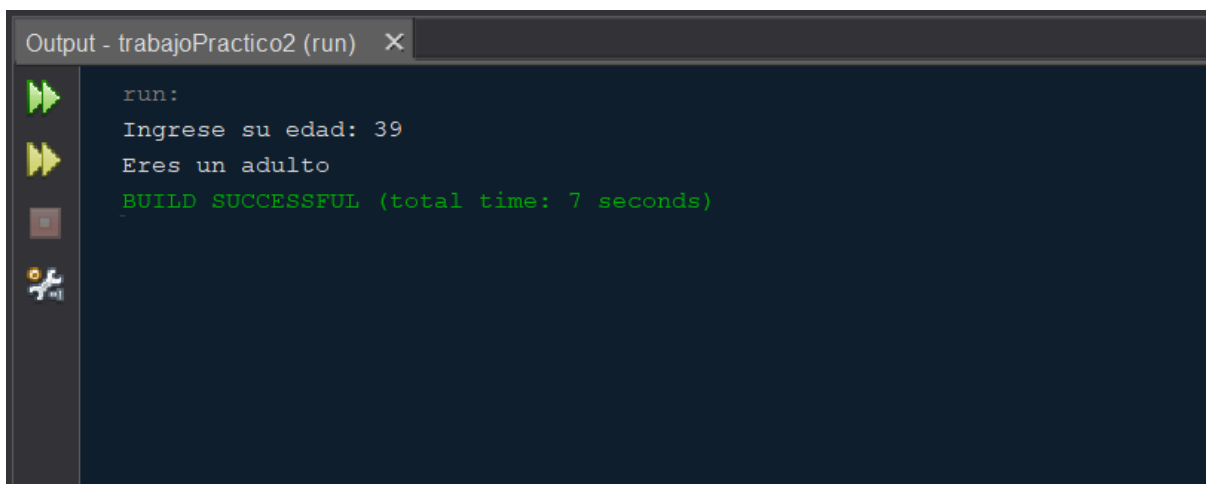
```
run:
Ingrese el primer numero:
1
Ingrese el segundo numero:
2
Ingrese el tercer numero:
3
El mayor es: 3
BUILD SUCCESSFUL (total time: 8 seconds)
```

### 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:

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

```
7 package trabajopractico2;
8 import java.util.Scanner;
9 /*3. Clasificación de Edad.
10 Escribe un programa en Java que solicite al usuario su edad y clasifique su
11 etapa de vida según la siguiente tabla:
12 */
13 public class ejercicio3 {
14     public static void main(String[] args) {
15         Scanner entrada = new Scanner(System.in);
16         System.out.print("Ingrese su edad: ");
17         int edad = entrada.nextInt();
18         if (edad < 0) {
19             System.out.println("Edad no válida. Por favor ingrese un número positivo.");
20         } else if (edad <= 12) {
21             System.out.println("Eres un niño");
22         } else if (edad > 12 && edad <=18) {
23             System.out.println("Eres un adolescente");
24         } else if (edad > 18 && edad <59 ) {
25             System.out.println("Eres un adulto");
26         } else if (edad >= 60) {
27             System.out.println("Eres un adulto mayor");
28         }
29     }
30 }
31 }
```



```
run:
Ingrese su edad: 39
Eres un adulto
BUILD SUCCESSFUL (total time: 7 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

```
7 package trabajoPractico2;
8 import java.util.Scanner;
9
10 /*4. Calculadora de Descuento según categoría.
11 Escribe un programa que solicite al usuario el precio de un producto y
12 su categoría (A, B o C).
13 Luego, aplique los siguientes descuentos:
14 Categoría A: 10% de descuento
15 Categoría B: 15% de descuento
16 Categoría C: 20% de descuento
17 El programa debe mostrar el precio original, el descuento aplicado y el
18 precio final
19 */
20 public class ejercicio4 {
21
22     static final double DESCUENTO_A = 0.10;
23     static final double DESCUENTO_B = 0.15;
24     static final double DESCUENTO_C = 0.20;
25
26
27     public static void main(String[] args) {
28         Scanner entrada = new Scanner(System.in);
29         System.out.print("Ingrese el precio del producto: ");
30         int precio = entrada.nextInt();
31         entrada.nextLine();
32         System.out.print("Ingrese la categoría del producto ( A, B, o C): ");
33         String categoria = entrada.nextLine().toUpperCase();
34         double precioFinal = 0;
35
36         switch( categoria ){
37             case "A":
38                 precioFinal = precio * (1 - DESCUENTO_A);
39                 System.out.println("El valor de su producto es de: $" + precio + " y su descuento es del %" + (int)(DESCUENTO_A*100) + ". El precio final es de: " + precioFinal);
40                 break;
41             case "B":
42                 precioFinal = precio * (1 - DESCUENTO_B);
43                 System.out.println("El valor de su producto es de: " + precio + " y su descuento es del %" + (int)(DESCUENTO_B*100) + ". El precio final es de: " + precioFinal);
44                 break;
45             case "C":
46                 precioFinal = precio * (1 - DESCUENTO_C);
47                 System.out.println("El valor de su producto es de: " + precio + " y su descuento es del %" + (int)(DESCUENTO_C*100) + ". El precio final es de: $" + precioFinal );
48                 break;
49             default:
50                 System.out.println("La categoría no existe! ");
51         }
52     }
53 }
```

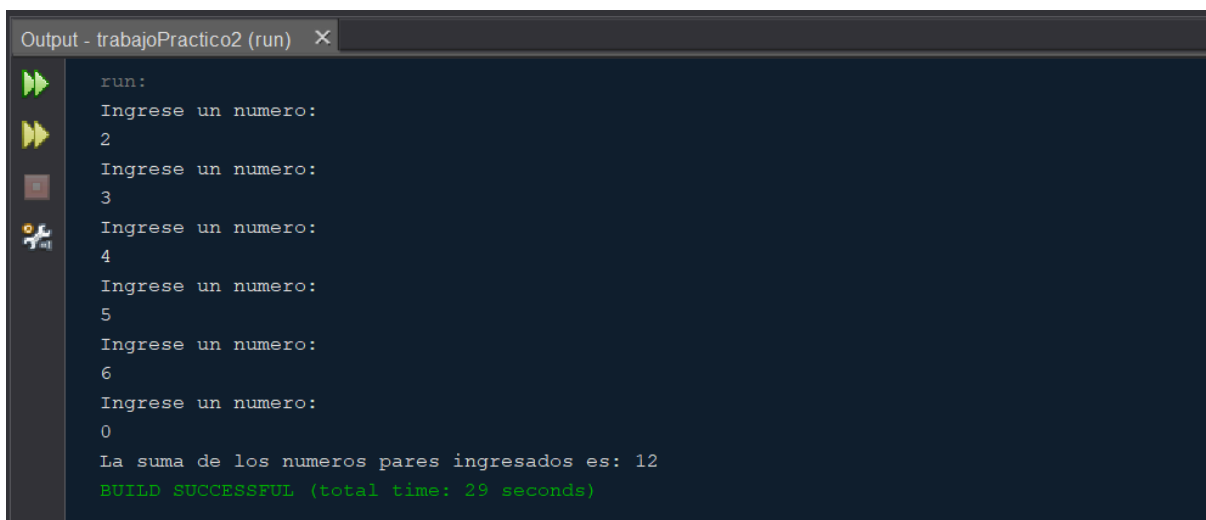
```
Output - trabajoPractico2 (run) X
run:
Ingrese el precio del producto: 1000
Ingrese la categoría del producto ( A, B, o C): a
El valor de su producto es de: $1000 y su descuento es del %10. El precio final es de: 900.0
BUILD SUCCESSFUL (total time: 7 seconds)
```

## 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.

```
7 package trabajopractico2;
8
9 import java.util.Scanner;
10
11 /*
12 5. Suma de Números Pares (while).
13 Escribe un programa que solicite números al usuario y sume solo los
14 números pares. El ciclo debe continuar hasta que el usuario ingrese el número
15 0, momento en el que se debe mostrar la suma total de los pares ingresados.
16
17 */
18 public class ejercicio5 {
19
20     public static void main(String[] args) {
21         boolean bandera = true;
22         int suma = 0;
23         Scanner entrada = new Scanner(System.in);
24
25         while (bandera) {
26             System.out.println("Ingrese un numero: ");
27             int numero = entrada.nextInt();
28             if (numero == 0) {
29                 bandera = false;
30             } else if (numero % 2 == 0) {
31                 suma += numero;
32             }
33         }
34         System.out.println("La suma de los numeros pares ingresados es: " + suma);
35     }
36 }
```

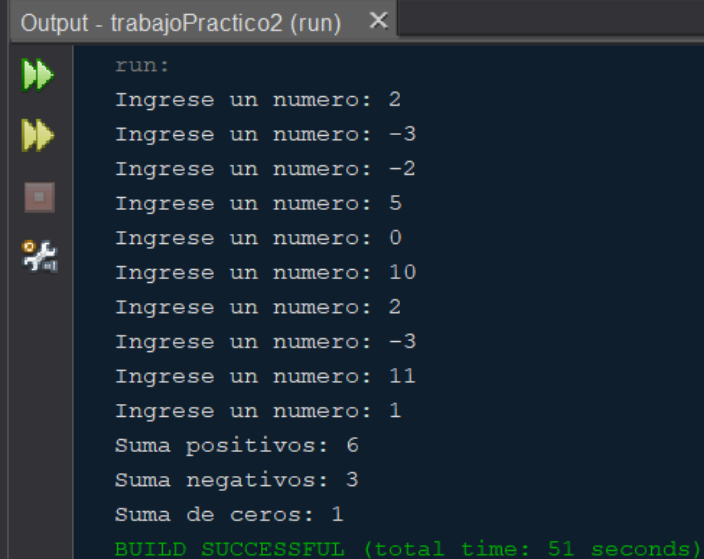


```
run:
Ingrese un numero:
2
Ingrese un numero:
3
Ingrese un numero:
4
Ingrese un numero:
5
Ingrese un numero:
6
Ingrese un numero:
0
La suma de los numeros pares ingresados es: 12
BUILD SUCCESSFUL (total time: 29 seconds)
```

## 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.

```
7 package trabajopractico2;
8 import java.util.Scanner;
9
10 /*
11 6. Contador de Positivos, Negativos y Ceros (for).
12 Escribe un programa que pida al usuario ingresar 10 números enteros y
13 cuente cuántos son positivos, negativos y cuántos son ceros.
14 */
15 */
16
17 public class ejercicio6 {
18     public static void main(String[] args) {
19         Scanner entrada = new Scanner(System.in);
20         int sumPos = 0, sumNeg = 0, sumCeros = 0;
21         for (int i = 0; i < 10; i++) {
22             System.out.print("Ingrese un numero: ");
23             int numero = entrada.nextInt();
24             if (numero > 0) {
25                 sumPos++;
26             } else if (numero < 0) {
27                 sumNeg++;
28             } else {
29                 sumCeros++;
30             }
31         }
32
33         System.out.println("Suma positivos: "+ sumPos);
34         System.out.println("Suma negativos: "+ sumNeg);
35         System.out.println("Suma de ceros: "+ sumCeros);
36
37     }
38 }
39 }
```

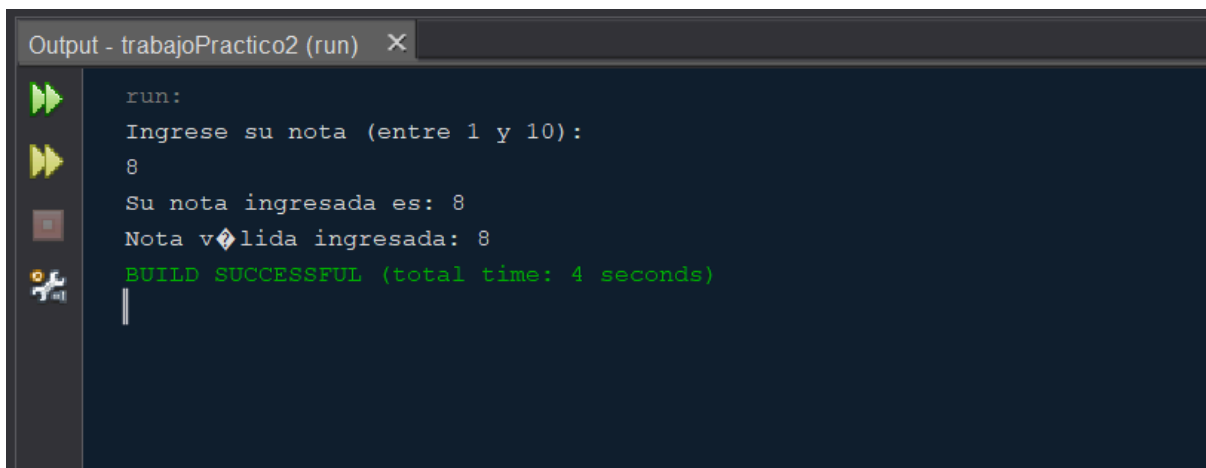


```
Output - trabajoPractico2 (run) X
run:
Ingrese un numero: 2
Ingrese un numero: -3
Ingrese un numero: -2
Ingrese un numero: 5
Ingrese un numero: 0
Ingrese un numero: 10
Ingrese un numero: 2
Ingrese un numero: -3
Ingrese un numero: 11
Ingrese un numero: 1
Suma positivos: 6
Suma negativos: 3
Suma de ceros: 1
BUILD SUCCESSFUL (total time: 51 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.

```
7 package trabajopractico2;
8 import java.util.Scanner;
9 /*
10 7. Validación de Nota entre 0 y 10 (do-while).
11 Escribe un programa que solicite al usuario una nota entre 0 y 10. Si el
12 usuario ingresa un número fuera de este rango, debe seguir pidiéndole la nota
13 hasta que ingrese un valor válido.
14
15 */
16
17 public class ejercicio7 {
18
19     public static void main(String[] args) {
20         Scanner entrada = new Scanner(System.in);
21         int nota;
22         do {
23             System.out.println("Ingrese su nota (entre 1 y 10): ");
24             nota = entrada.nextInt();
25             System.out.println("Su nota ingresada es: " + nota);
26
27             if (nota < 1 || nota > 10) {
28                 System.out.println("Nota inválida. Intente nuevamente.");
29             }
30         } while (nota < 1 || nota > 10);
31
32         System.out.println("Nota válida ingresada: " + nota);
33     }
34 }
35 }
```



```
Output - trabajoPractico2 (run) X
run:
Ingrese su nota (entre 1 y 10):
8
Su nota ingresada es: 8
Nota válida ingresada: 8
BUILD SUCCESSFUL (total time: 4 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:

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

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.

```
7 package trabajopractico2;
8
9 import java.util.Scanner;
10
11 /*
12 8. Cálculo del Precio Final con impuesto y descuento.
13 Crea un método calcularPrecioFinal(double impuesto, double
14 descuento) que calcule el precio final de un producto en un e-commerce. La
15 fórmula es:
16 PrecioFinal = PrecioBase + (PrecioBase×Impuesto) - (PrecioBase×Descuento)
17 PrecioFinal = PrecioBase + (PrecioBase \times Impuesto) - (PrecioBase \times
18 Descuento)
19 Desde main(), solicita el precio base del producto, el porcentaje de
20 impuesto y el porcentaje de descuento, llama al método y muestra el precio
21 final.
22 */
23 public class ejercicio8 {
24
25     public static void main(String[] args) {
26         Scanner entrada = new Scanner(System.in);
27         System.out.println("Ingrese el precio base de su producto: ");
28         double precioBase = entrada.nextDouble();
29         System.out.println("Ingrese el porcentaje de impuestos (Ejemplo: 10 para 10%): ");
30         double porcImpuesto = entrada.nextDouble() / 100;
31         System.out.println("Ingrese el porcentaje de descuento (Ejemplo: 5 para 5%): ");
32         double porcDescuento = entrada.nextDouble() / 100;
33
34         double resultado = calcularPrecioFinal(precioBase, porcDescuento, porcImpuesto);
35         System.out.println("Precio final: $" + (int) resultado);
36     }
37
38     static Double calcularPrecioFinal(double precioBase, double porcImpuesto, double porcDescuento) {
39         double impuesto = precioBase * porcImpuesto;
40         double descuento = precioBase * porcDescuento;
41         double precioFinal = precioBase + impuesto - descuento;
42         return precioFinal;
43     }
44 }
45
46 }
```

```
Output - trabajoPractico2 (run) X
run:
Ingrese el precio base de su producto:
1250
Ingrese el porcentaje de impuestos (Ejemplo: 10 para 10%):
10
Ingrese el porcentaje de descuento (Ejemplo: 5 para 5%):
5
Precio final: $1187
BUILD SUCCESSFUL (total time: 18 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.

```
7 package trabajopractico2;
8
9 import java.util.Scanner;
10
11 /*9. Composición de funciones para calcular costo de envío y total de compra.
12 a. calcularCostoEnvio(double peso, String zona): Calcula el costo de
13 envío basado en la zona de envío (Nacional o Internacional) y el peso del
14 paquete.
15 Nacional: $5 por kg
16 Internacional: $10 por kg
17 b. calcularTotalCompra(double precioProducto, double
18 costoEnvio): Usa calcularCostoEnvio para sumar el costo del producto con
19 el costo de envío.
20 Desde main(), solicita el peso del paquete, la zona de envío y el precio
21 del producto. Luego, muestra el total a pagar.
22 */
23 public class ejercicio9 {
24
25     public static void main(String[] args) {
26         Scanner entrada = new Scanner(System.in);
27         System.out.println("Ingrese el precio del producto: ");
28         double precioProducto = entrada.nextDouble();
29         System.out.println("Ingreso el peso del paquete en kg(Ejemplo: 50): ");
30         double peso = entrada.nextDouble();
31         entrada.nextLine();
32         System.out.println("Ingrese la zona de envío (Nacional/Internacional): ");
33         String zona = entrada.nextLine();
34         double costoEnvio = calcularCostoEnvio(peso, zona);
35         double total = calcularTotalCompra(precioProducto, costoEnvio);
36         System.out.println("El costo de envío es: $" + (int)costoEnvio);
37         System.out.println("Total a pagar: $" + (int) total);
38     }
39
40     public static double calcularCostoEnvio(double peso, String zona) {
41         double costoEnvio = 0;
42         if (zona.equalsIgnoreCase("Nacional")) {
43             costoEnvio = peso * 5;
44         } else if (zona.equalsIgnoreCase("Internacional")) {
45             costoEnvio = peso * 10;
46         } else {
47             System.out.println("Zona no válida. Use 'Nacional' o 'Internacional'.");
48         }
49         return costoEnvio;
50     }
51
52     public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
53         double costoTotal = precioProducto + costoEnvio;
54         return costoTotal;
55     }
56
57 }
```

```
Output - trabajoPractico2 (run) ×
run:
Ingrese el precio del producto:
5000
Ingreso el peso del paquete en kg(Ejemplo: 50):
15
Ingrese la zona de envio (Nacional/Internacional):
nacional
El costo de envio es: $75
Total a pagar: $5075
BUILD SUCCESSFUL (total time: 11 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

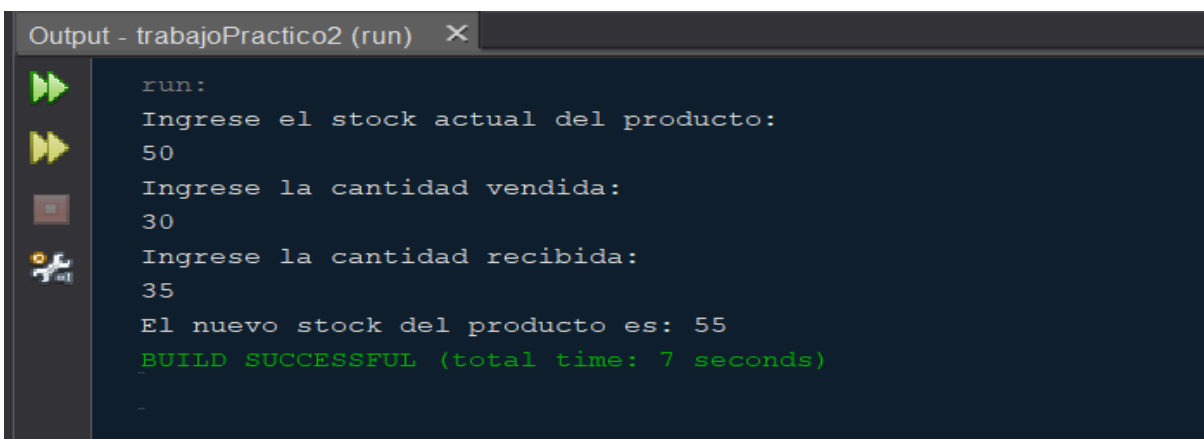
de productos:

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

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

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

```
7 package trabajopractico2;
8 import java.util.Scanner;
9
10 /*
11 10. Actualización de stock a partir de venta y recepción de productos.
12 Crea un método actualizarStock(int stockActual, int cantidadVendida,
13 int cantidadRecibida), que calcule el nuevo stock después de una venta y
14 recepción
15 de productos:
16 NuevoStock = StockActual - CantidadVendida + CantidadRecibida
17 NuevoStock = CantidadVendida + CantidadRecibida
18 Desde main(), solicita al usuario el stock actual, la cantidad vendida y la
19 cantidad recibida, y muestra el stock actualizado.
20 */
21 public class ejercicio10 {
22
23     public static void main(String[] args) {
24         Scanner entrada = new Scanner(System.in);
25         System.out.println("Ingrese el stock actual del producto: ");
26         int stockActual = entrada.nextInt();
27         System.out.println("Ingrese la cantidad vendida: ");
28         int cantidadVendida = entrada.nextInt();
29         System.out.println("Ingrese la cantidad recibida: ");
30         int cantidadRecibida = entrada.nextInt();
31
32         int stockActualizado = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
33         System.out.println("El nuevo stock del producto es: " + stockActualizado);
34     }
35
36     public static int actualizarStock(int x, int y, int z) {
37         int nuevoStock = x - y + z;
38         return nuevoStock;
39     }
40 }
```



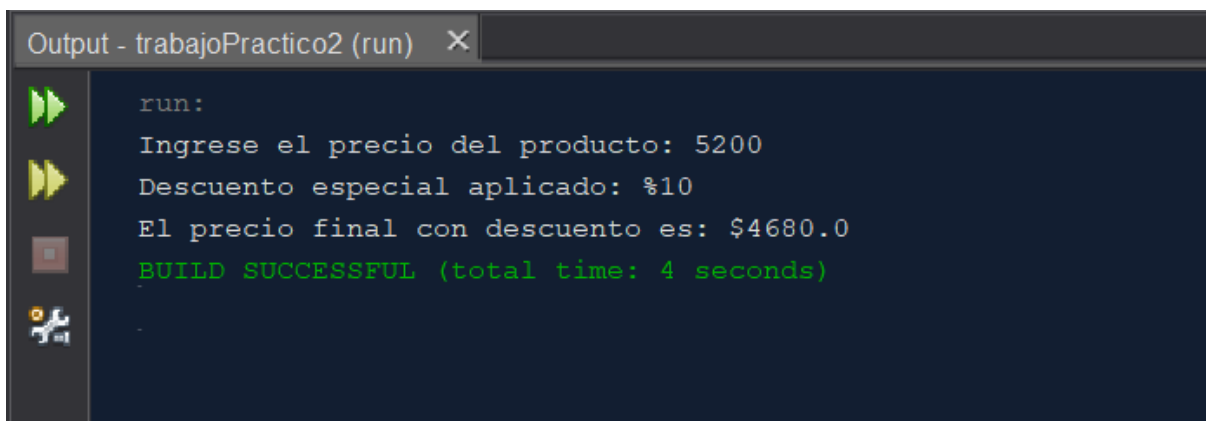
```
Output - trabajoPractico2 (run) X
run:
Ingrese el stock actual del producto:
50
Ingrese la cantidad vendida:
30
Ingrese la cantidad recibida:
35
El nuevo stock del producto es: 55
BUILD SUCCESSFUL (total time: 7 seconds)
```

### 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.

```
7 package trabajopractico2;
8 import java.util.Scanner;
9 /*
10
11 11. Cálculo de descuento especial usando variable global.
12 Declara una variable global Ejemplo de entrada/salida: = 0.10. Luego, crea un
13 método calcularDescuentoEspecial(double precio) que use la variable global para
14 calcular el descuento especial del 10%.
15 Dentro del método, declara una variable local descuentoAplicado, almacena
16 el valor del descuento y muestra el precio final con descuento.
17
18
19 */
20
21 public class ejercicio11 {
22
23     static final double DESCUENTO_ESPECIAL = 0.10;
24
25     public static void main(String[] args) {
26         Scanner entrada = new Scanner(System.in);
27         System.out.print("Ingrese el precio del producto: ");
28         double precio = entrada.nextDouble();
29         calcularDescuentoEspecial(precio);
30
31     }
32
33     public static void calcularDescuentoEspecial(double precio){
34         double descuentoAplicado = DESCUENTO_ESPECIAL * 100;
35         double precioFinal = precio - (precio * DESCUENTO_ESPECIAL);
36         System.out.println("Descuento especial aplicado: %" + (int) descuentoAplicado);
37         System.out.println("El precio final con descuento es: $" + precioFinal);
38     }
39 }
40 }
```



```
Output - trabajoPractico2 (run) X
run:
Ingrese el precio del producto: 5200
Descuento especial aplicado: %10
El precio final con descuento es: $4680.0
BUILD SUCCESSFUL (total time: 4 seconds)
```

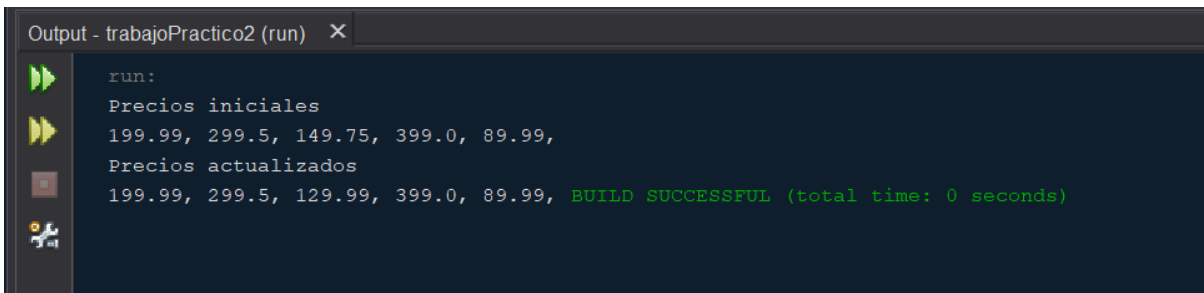
## Arrays y Recursividad:

### 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.

```
7 package trabajopractico2;
8
9 /*
10 12. Modificación de un array de precios y visualización de resultados.
11 Crea un programa que:
12 a. Declare e inicialice un array con los precios de algunos productos.
13 b. Muestre los valores originales de los precios.
14 c. Modifique el precio de un producto específico.
15 d. Muestre los valores modificados.
16 */
17 public class ejercicio12 {
18
19     public static void main(String[] args) {
20         double precios[] = {199.99, 299.5, 149.75, 399.0, 89.99};
21         System.out.println("Precios iniciales");
22         for (double precio : precios) {
23             System.out.print(precio + ", ");
24         }
25         System.out.println("");
26         precios[2] = 129.99;
27         System.out.println("Precios actualizados");
28         for (double precio : precios) {
29             System.out.print(precio + ", ");
30         }
31     }
32 }
33 }
```



```
Output - trabajoPractico2 (run) X
run:
Precios iniciales
199.99, 299.5, 149.75, 399.0, 89.99,
Precios actualizados
199.99, 299.5, 129.99, 399.0, 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.

```
7 package trabajopractico2;
8
9 /*
10
11 13. Impresión recursiva de arrays antes y después de modificar un elemento.
12 Crea un programa que:
13 a. Declare e inicialice un array con los precios de algunos productos.
14 b. Use una función recursiva para mostrar los precios originales.
15 c. Modifique el precio de un producto específico.
16 d. Use otra función recursiva para mostrar los valores modificados.
17
18 */
19 public class ejercicio13 {
20     public static void main(String[] args) {
21         double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
22         System.out.println("Precios Originales");
23         mostrarPreciosRecursivo(precios, 0);
24         precios[2]= 129.99;
25         System.out.println("");
26         System.out.println("Precios Modificados");
27         mostrarPreciosRecursivo(precios, 0);
28     }
29 }
30
31
32     public static void mostrarPreciosRecursivo(double[] precios, int indice) {
33         if (indice < precios.length) {
34
35             System.out.println("Precio " +(indice+1)+": " + precios[indice]);
36             mostrarPreciosRecursivo(precios, indice + 1);
37         }
38     }
```