

Trabajo Práctico – Programación Estructurada

Materia: Programación II

Alumno: Fermin Alliot

Carrera: Tecnicatura Universitaria en Programación (a distancia)

Comisión: "3"

GitHub: <https://github.com/Falliot00/UTN-TUPaD-P2/tree/main/2.%20Programaci%C3%B3n%20Estructurada>

1) Verificación de Año Bisiesto

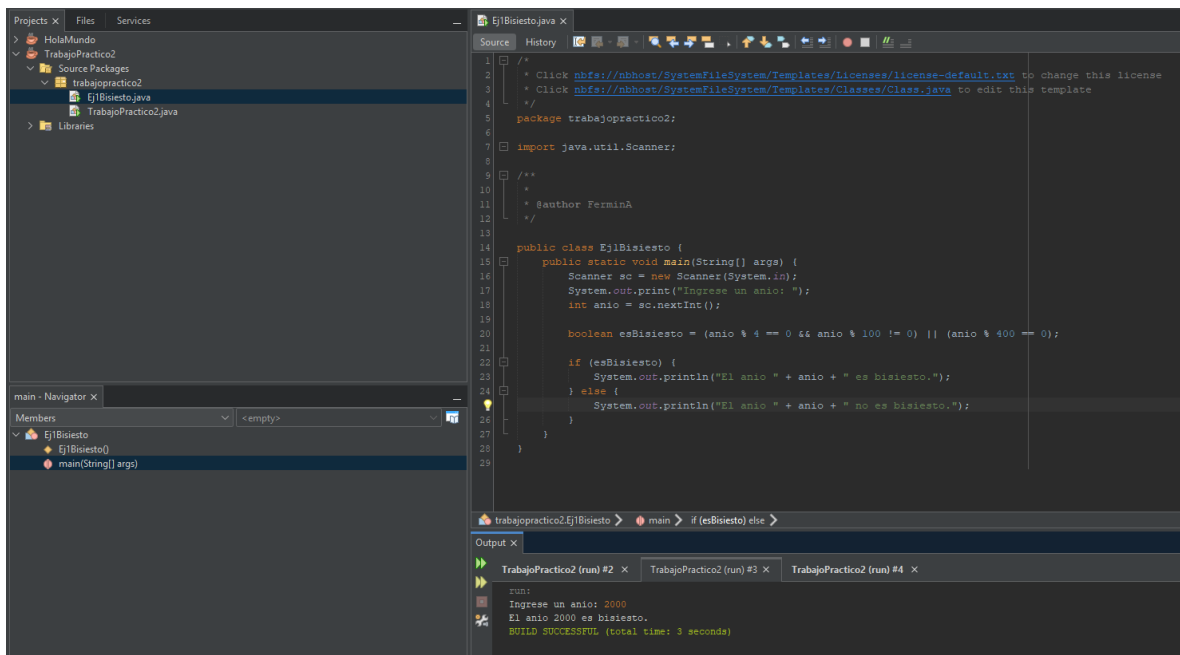
Un año es bisiesto si es divisible por 4 y no por 100, o si es divisible por 400.

```
import java.util.Scanner;

public class Ej1Bisiesto {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese un año: ");
        int anio = sc.nextInt();

        boolean esBisiesto = (anio % 4 == 0 && anio % 100 != 0) || (anio
% 400 == 0);

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



2) Determinar el Mayor de Tres Números

Leo tres enteros y comparo con condicionales. También podría usarse `Math.max` anidado.

```
import java.util.Scanner;
```

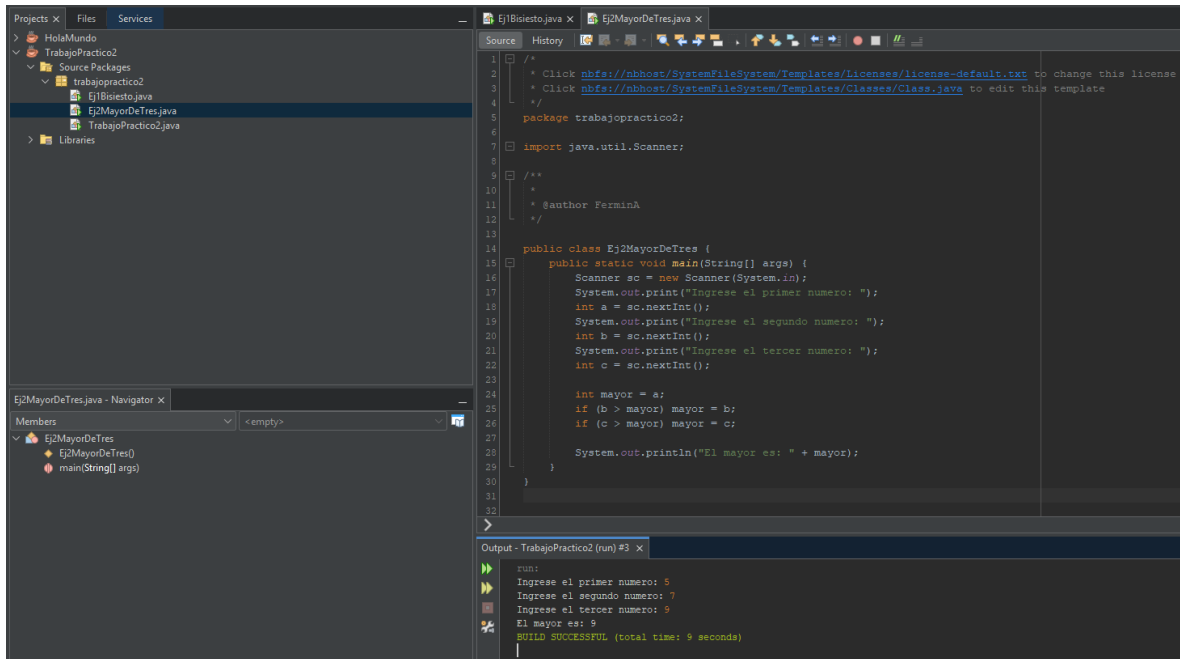
```

public class Ej2MayorDeTres {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el primer número: ");
        int a = sc.nextInt();
        System.out.print("Ingrese el segundo número: ");
        int b = sc.nextInt();
        System.out.print("Ingrese el tercer número: ");
        int c = sc.nextInt();

        int mayor = a;
        if (b > mayor) mayor = b;
        if (c > mayor) mayor = c;

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

```

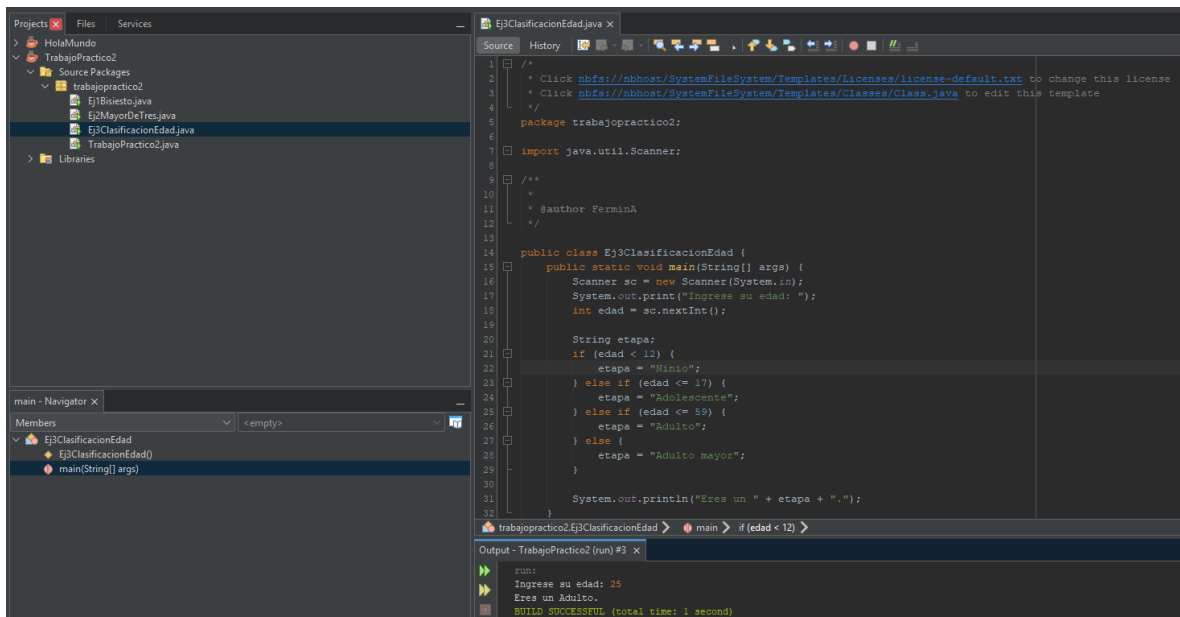


3) Clasificación de Edad

Usar un if-else if-else con los rangos indicados.

```
import java.util.Scanner;
```

```
public class Ej3ClasificacionEdad {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Ingrese su edad: ");  
        int edad = sc.nextInt();  
  
        String etapa;  
        if (edad < 12) {  
            etapa = "Niño";  
        } else if (edad <= 17) {  
            etapa = "Adolescente";  
        } else if (edad <= 59) {  
            etapa = "Adulto";  
        } else {  
            etapa = "Adulto mayor";  
        }  
  
        System.out.println("Eres un " + etapa + ".");  
    }  
}
```



4) Calculadora de Descuento según categoría

Normalizo la categoría a mayúsculas y aplico porcentaje según A/B/C.

```
import java.util.Scanner;
```

```
public class Ej4DescuentoPorCategoria {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el precio del producto: ");
        double precio = sc.nextDouble();
        System.out.print("Ingrese la categoría del producto (A, B o C): ");
        char cat = sc.next().trim().toUpperCase().charAt(0);

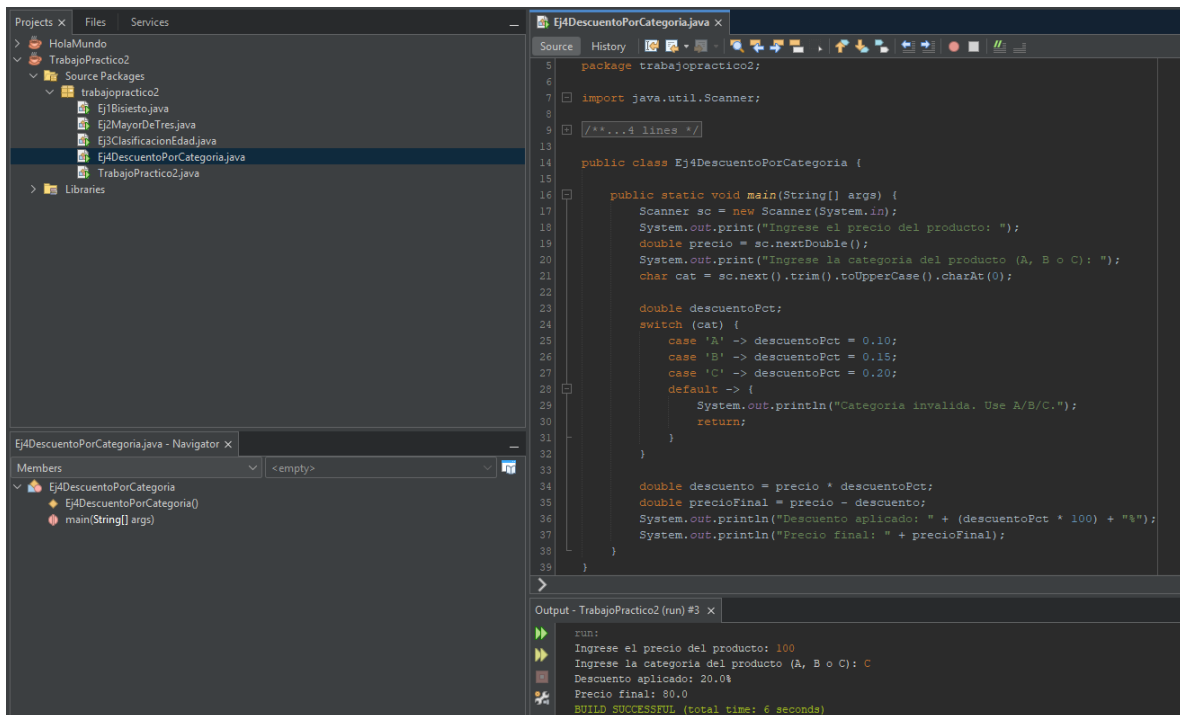
        double descuentoPct;
        switch (cat) {
            case 'A': descuentoPct = 0.10; break;
            case 'B': descuentoPct = 0.15; break;
            case 'C': descuentoPct = 0.20; break;
            default:
                System.out.println("Categoría inválida. Use A/B/C.");
                return;
        }

        double descuento = precio * descuentoPct;
        double precioFinal = precio - descuento;
        System.out.println("Descuento aplicado: " + (descuentoPct * 100)
            + "%");
    }
}
```

```

        System.out.println("Precio final: " + precioFinal);
    }
}

```



5) Suma de Números Pares (while)

Leo hasta ingresar 0, acumulo solo si el número es par.

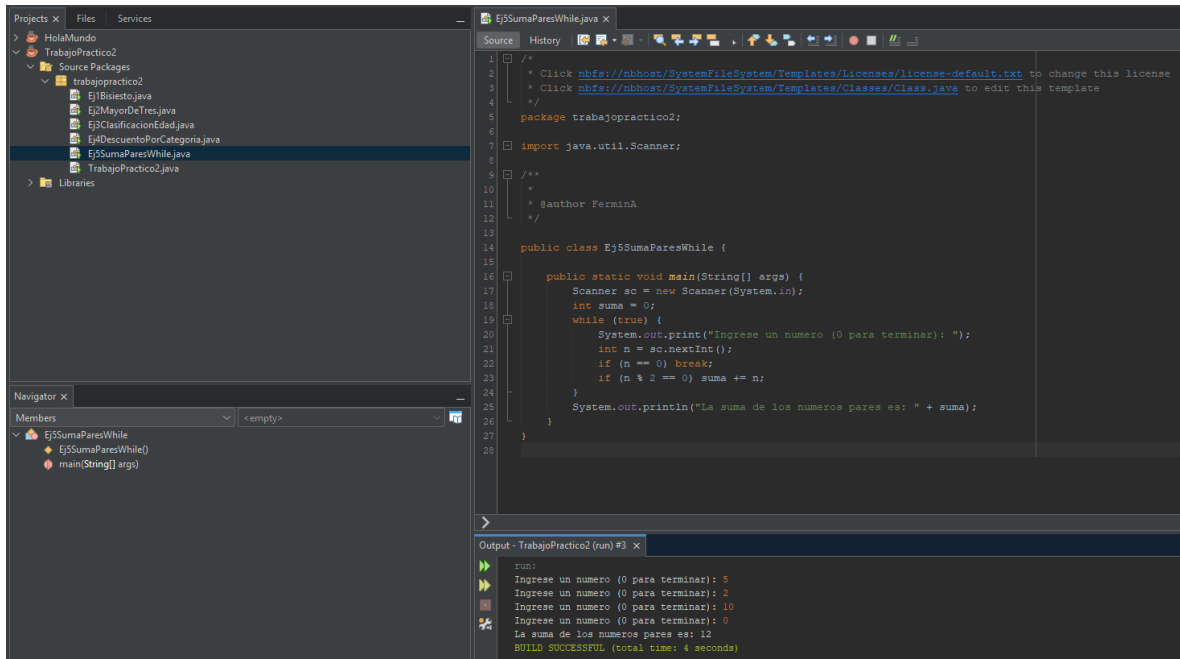
Nota: no sumo el 0 final, aunque sea par, ya que es el centinela de fin.

```

import java.util.Scanner;

public class Ej5SumaParesWhile {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int suma = 0;
        while (true) {
            System.out.print("Ingrese un número (0 para terminar): ");
            int n = sc.nextInt();
            if (n == 0) break;
            if (n % 2 == 0) suma += n;
        }
        System.out.println("La suma de los números pares es: " + suma);
    }
}

```

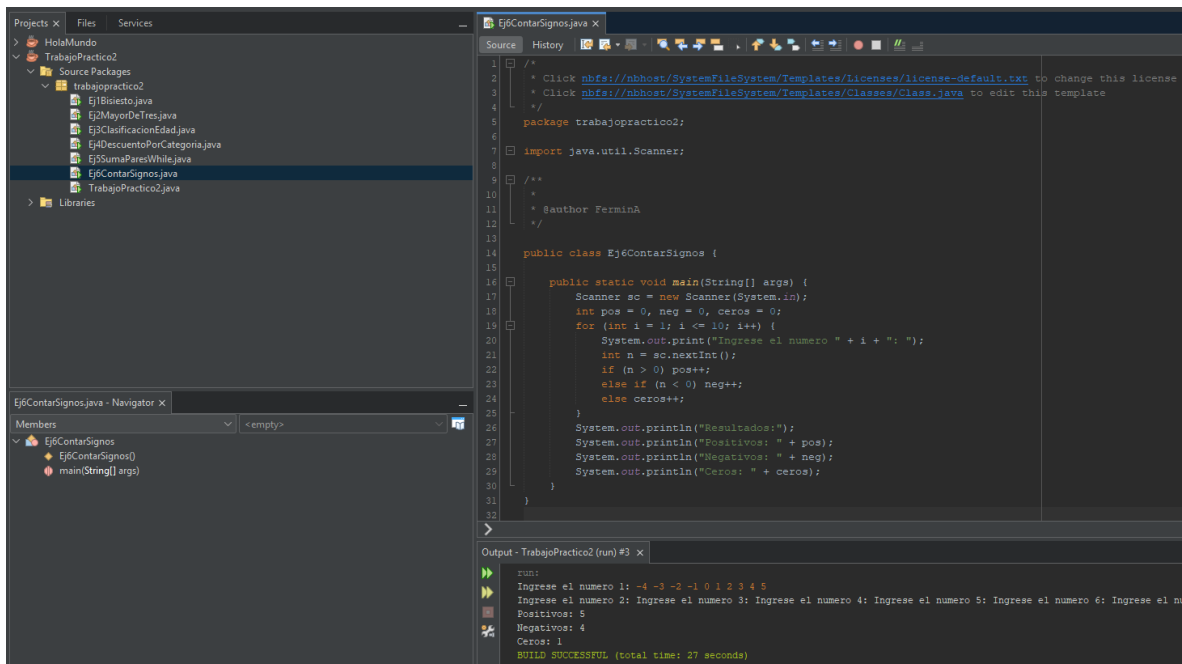


6) Contador de Positivos, Negativos y Ceros (for)

Pido 10 enteros y uso contadores.

```
import java.util.Scanner;
```

```
public class Ej6ContarSignos {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int pos = 0, neg = 0, ceros = 0;  
        for (int i = 1; i <= 10; i++) {  
            System.out.print("Ingrese el número " + i + ": ");  
            int n = sc.nextInt();  
            if (n > 0) pos++;  
            else if (n < 0) neg++;  
            else ceros++;  
        }  
        System.out.println("Resultados:");  
        System.out.println("Positivos: " + pos);  
        System.out.println("Negativos: " + neg);  
        System.out.println("Ceros: " + ceros);  
    }  
}
```

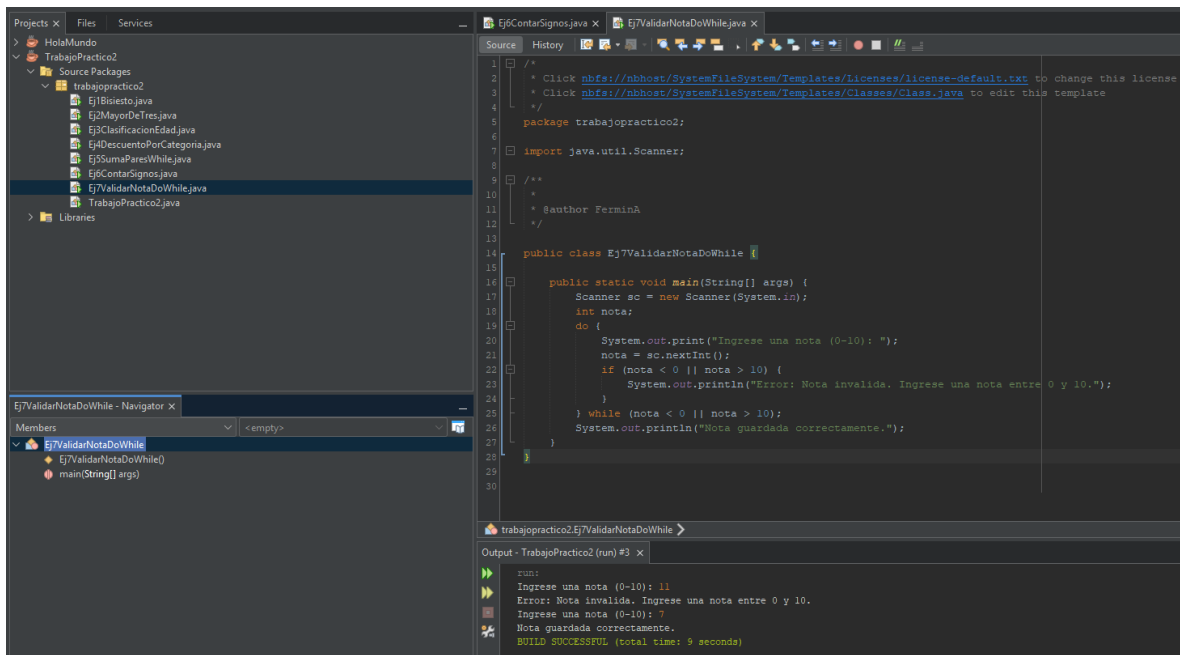


7) Validación de Nota entre 0 y 10 (do-while)

Con do-while garantizo al menos una lectura y repito hasta un valor válido.

```
import java.util.Scanner;
```

```
public class Ej7ValidarNotaDowhile {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nota;
        do {
            System.out.print("Ingrese una nota (0-10): ");
            nota = sc.nextInt();
            if (nota < 0 || nota > 10) {
                System.out.println("Error: Nota inválida. Ingrese una
                nota entre 0 y 10.");
            }
        } while (nota < 0 || nota > 10);
        System.out.println("Nota guardada correctamente.");
    }
}
```



8) Cálculo del Precio Final con impuesto y descuento (función)

La consigna propone calcular `PrecioFinal(double impuesto, double descuento)` sin `precioBase` como parámetro. Para respetar esto, defino un **campo estático** `precioBase` que cargo desde `main`. Alternativamente, otra versión es pasar `precioBase` como parámetro.

```
import java.util.Scanner;

public class Ej8PrecioFinal {
    // Contexto compartido para respetar la firma pedida
    private static double precioBase;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el precio base del producto: ");
        precioBase = sc.nextDouble();
        System.out.print("Ingrese el impuesto en porcentaje (Ej: 10 para 10%): ");
        double impPct = sc.nextDouble();
        System.out.print("Ingrese el descuento en porcentaje (Ej: 5 para 5%): ");
        double descPct = sc.nextDouble();

        double precioFinal = calcularPrecioFinal(impPct, descPct);
        System.out.println("El precio final del producto es: " + precioFinal);
    }
}
```

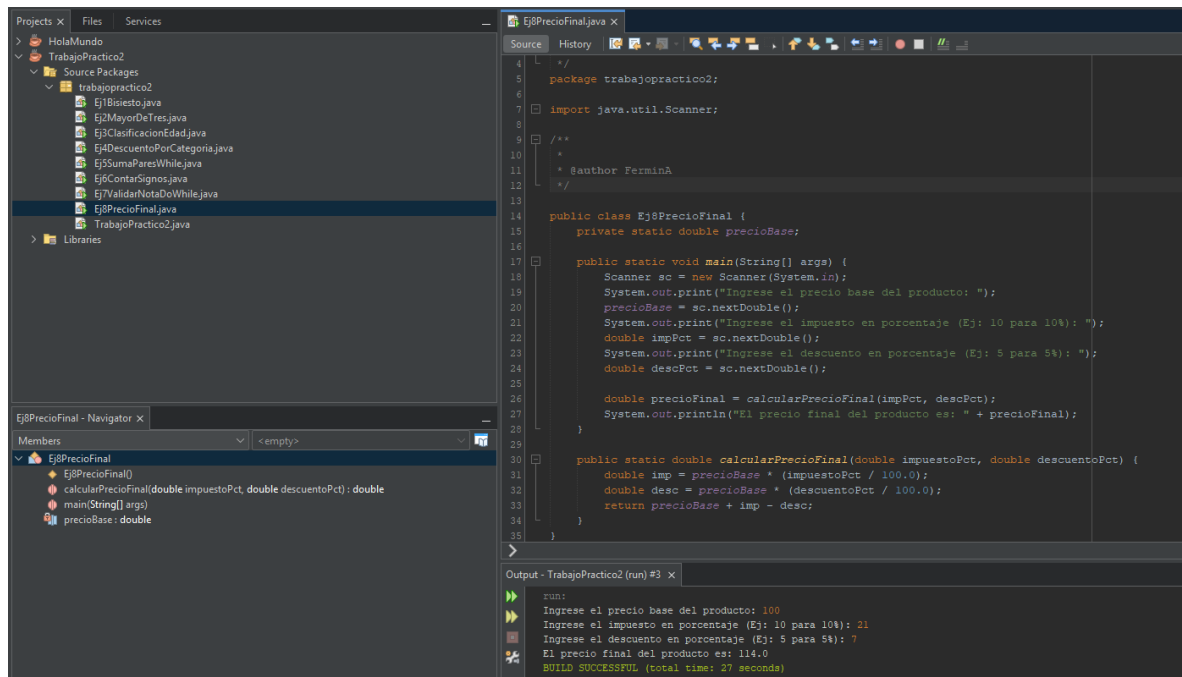


```

    }

    public static double calcularPrecioFinal(double impuestoPct, double
descuentoPct) {
        double imp = precioBase * (impuestoPct / 100.0);
        double desc = precioBase * (descuentoPct / 100.0);
        return precioBase + imp - desc;
    }
}

```



9) Composición de funciones: costo de envío y total de compra

Implemento dos funciones, la primera calcula el envío según zona/peso, la segunda suma al precio del producto.

```
import java.util.Scanner;
```

```

public class Ej9EnvioYTotal {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el precio del producto: ");
        double precioProducto = sc.nextDouble();
        System.out.print("Ingrese el peso del paquete en kg: ");
        double peso = sc.nextDouble();
        System.out.print("Ingrese la zona de envío
(Nacional/Internacional): ");
    }
}

```

```

String zona = sc.next();

double costoEnvio = calcularCostoEnvio(peso, zona);
double total = calcularTotalCompra(precioProducto, costoEnvio);

System.out.println("El costo de envío es: " + costoEnvio);
System.out.println("El total a pagar es: " + total);
}

public static double calcularCostoEnvio(double peso, String zona) {
    double tarifaPorKg;
    if (zona.equalsIgnoreCase("Nacional")) {
        tarifaPorKg = 5.0;
    } else if (zona.equalsIgnoreCase("Internacional")) {
        tarifaPorKg = 10.0;
    } else {
        throw new IllegalArgumentException("Zona inválida. Use Nacional/Internacional.");
    }
    return tarifaPorKg * peso;
}

public static double calcularTotalCompra(double precioProducto,
double costoEnvio) {
    return precioProducto + costoEnvio;
}
}

```

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'TrabajoPractico2' with a package 'trabajopractico2' containing several Java files, including 'Ej8EnvioYTotal.java'.
- Source Editor:** Displays the code for 'Ej8EnvioYTotal.java', which includes a 'main' method and two static methods: 'calcularCostoEnvio' and 'calcularTotalCompra'.
- Output Console:** Shows the execution results of the program. The user input is '100' for the price, '5' for the weight, and 'internacional' for the zone. The output shows a shipping cost of 50.0 and a total payment of 150.0.

```

run:
Ingrese el precio del producto: 100
Ingrese el peso del paquete en kg: 5
Ingrese la zona de envio (Nacional/Internacional): internacional
El costo de envio es: 50.0
El total a pagar es: 150.0
BUILD SUCCESSFUL (total time: 6 seconds)

```

10) Actualización de stock

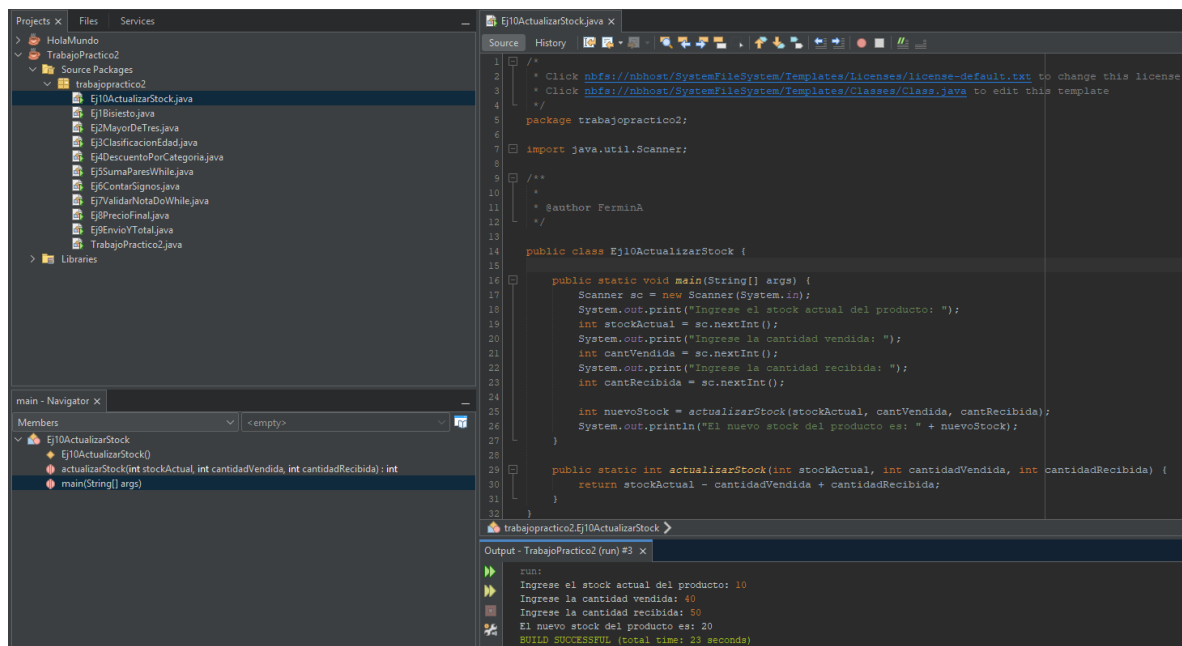
Aplico la fórmula correcta: $\text{nuevo} = \text{stockActual} - \text{cantidadVendida} + \text{cantidadRecibida}$.

```
import java.util.Scanner;
```

```
public class Ej10ActualizarStock {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el stock actual del producto: ");
        int stockActual = sc.nextInt();
        System.out.print("Ingrese la cantidad vendida: ");
        int cantVendida = sc.nextInt();
        System.out.print("Ingrese la cantidad recibida: ");
        int cantRecibida = sc.nextInt();

        int nuevoStock = actualizarStock(stockActual, cantVendida,
cantRecibida);
        System.out.println("El nuevo stock del producto es: " +
nuevoStock);
    }

    public static int actualizarStock(int stockActual, int
cantidadVendida, int cantidadRecibida) {
        return stockActual - cantidadVendida + cantidadRecibida;
    }
}
```



11) Descuento especial con variable global

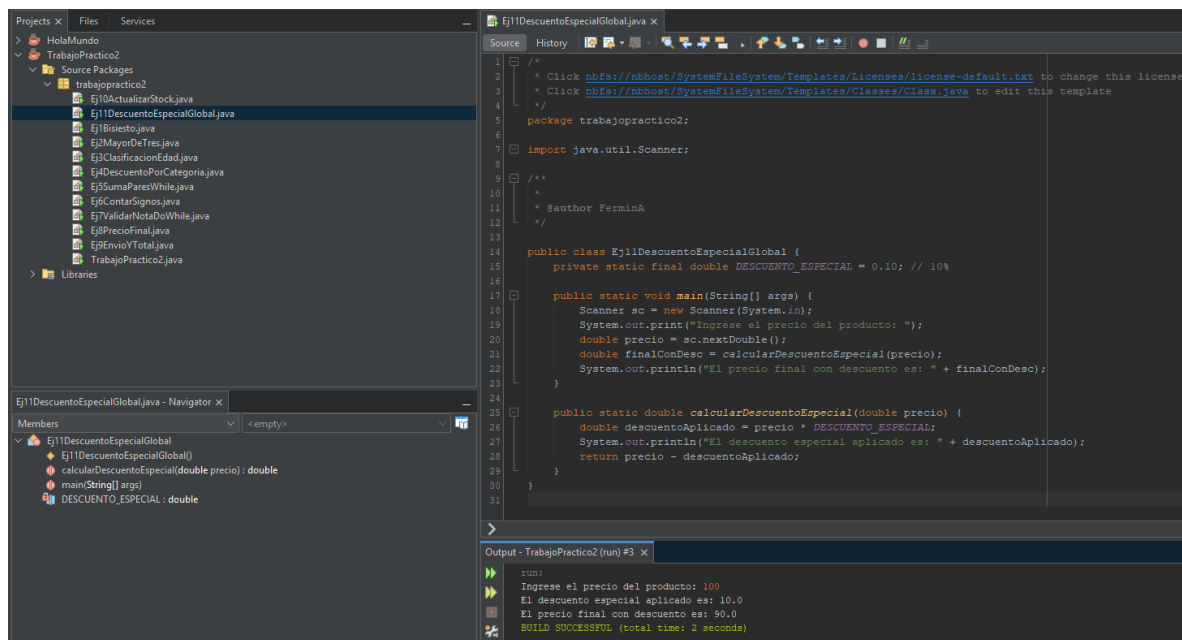
Defino una **variable de clase** (estática) con el porcentaje del descuento. Dentro del método uso una variable local `descuentoAplicado`.

```
import java.util.Scanner;
```

```
public class Ej11DescuentoEspecialGlobal {
    private static final double DESCUENTO_ESPECIAL = 0.10; // 10%

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el precio del producto: ");
        double precio = sc.nextDouble();
        double finalConDesc = calcularDescuentoEspecial(precio);
        System.out.println("El precio final con descuento es: " +
finalConDesc);
    }

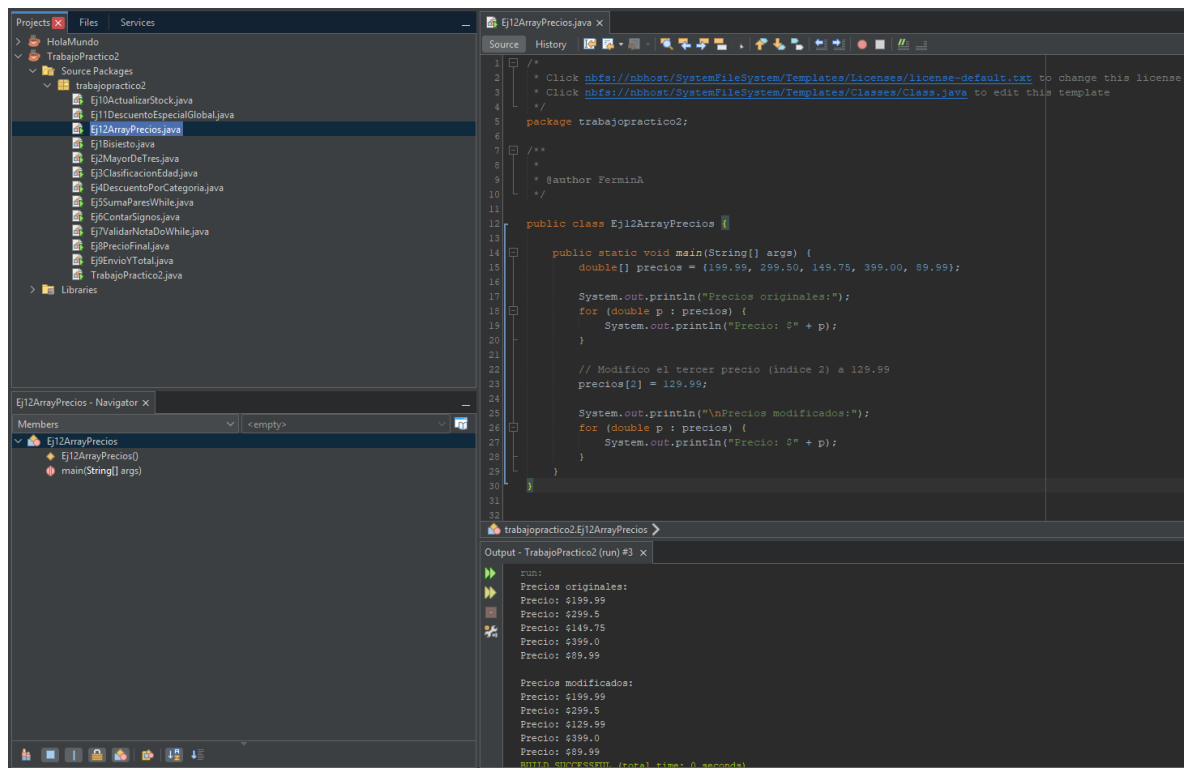
    public static double calcularDescuentoEspecial(double precio) {
        double descuentoAplicado = precio * DESCUENTO_ESPECIAL;
        System.out.println("El descuento especial aplicado es: " +
descuentoAplicado);
        return precio - descuentoAplicado;
    }
}
```



12) Modificación de un array de precios y visualización

Inicializo un `double[]`, muestro con `for-each`, modifico un índice y vuelvo a mostrar.

```
public class Ej12ArrayPrecios {  
    public static void main(String[] args) {  
        double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};  
  
        System.out.println("Precios originales:");  
        for (double p : precios) {  
            System.out.println("Precio: $" + p);  
        }  
  
        // Modifico el tercer precio (índice 2) a 129.99  
        precios[2] = 129.99;  
  
        System.out.println("\nPrecios modificados:");  
        for (double p : precios) {  
            System.out.println("Precio: $" + p);  
        }  
    }  
}
```



13) Impresión recursiva de arrays antes y después de modificar

Uso dos funciones recursivas (o la misma) que imprimen desde un índice *i* hasta el final.

```
public class Ej13ArrayRecursivo {
    public static void main(String[] args) {
        double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};

        System.out.println("Precios originales:");
        imprimirRecursivo(precios, 0);

        // Modificar el tercer precio (índice 2)
        precios[2] = 129.99;

        System.out.println("\nPrecios modificados:");
        imprimirRecursivo(precios, 0);
    }

    private static void imprimirRecursivo(double[] arr, int i) {
        if (i == arr.length) {
            return; // caso base
        }
        System.out.println("Precio: $" + arr[i]);
        imprimirRecursivo(arr, i + 1); // paso recursivo
    }
}
```

Projects x | Files | Services

HolaMundo

TrabajoPractico2

Source Packages

trabajopractico2

- Ej10ActualizarStock.java
- Ej11DescuentoEspecialGlobal.java
- Ej12ArrayPrecios.java
- Ej13ArrayRecursivo.java
- Ej16Bisesto.java
- Ej2MayorDeTres.java
- Ej3ClasificacionEdad.java
- Ej4DescuentoPorCategoria.java
- Ej5SumaParesWhile.java
- Ej6ContarSignos.java
- Ej7ValidarNotaDoWhile.java
- Ej8PrecioFinal.java
- Ej9EnvioYTotal.java
- TrabajoPractico2.java

Libraries

Ej13ArrayRecursivo.java - Navigator x

Members

<empty>

- Ej13ArrayRecursivo
- imprimirRecursivo(double[] arr, int i)
- main(String[] args)

Ej13ArrayRecursivo.java x

```
1  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
3
4  package trabajopractico2;
5
6
7  /**
8   *
9   * @author FerminA
10  */
11
12  public class Ej13ArrayRecursivo {
13      public static void main(String[] args) {
14          double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};
15
16          System.out.println("Precios originales:");
17          imprimirRecursivo(precios, 0);
18
19          // Modificar el tercer precio (indice 2)
20          precios[2] = 129.99;
21
22          System.out.println("\nPrecios modificados:");
23          imprimirRecursivo(precios, 0);
24      }
25
26      private static void imprimirRecursivo(double[] arr, int i) {
27          if (i == arr.length) {
28              return; // caso base
29          }
30          System.out.println("Precio: " + arr[i]);
31          imprimirRecursivo(arr, i + 1); // paso recursivo
32      }
33  }
```

Output - TrabajoPractico2 (run) #3 x

```
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)
```