

TP 2 Programación Estructurada
PROGRAMACIÓN II
AZCUY NICOLÁS - DNI 33.368.267
LINK GITHUB:

<https://github.com/nazcuy/Programaci-n-II/tree/master/Trabajos%20Pr%C3%A1cticos%20Programaci%C3%B3n%202/TP2>

1. Verificación de Año Bisiesto.

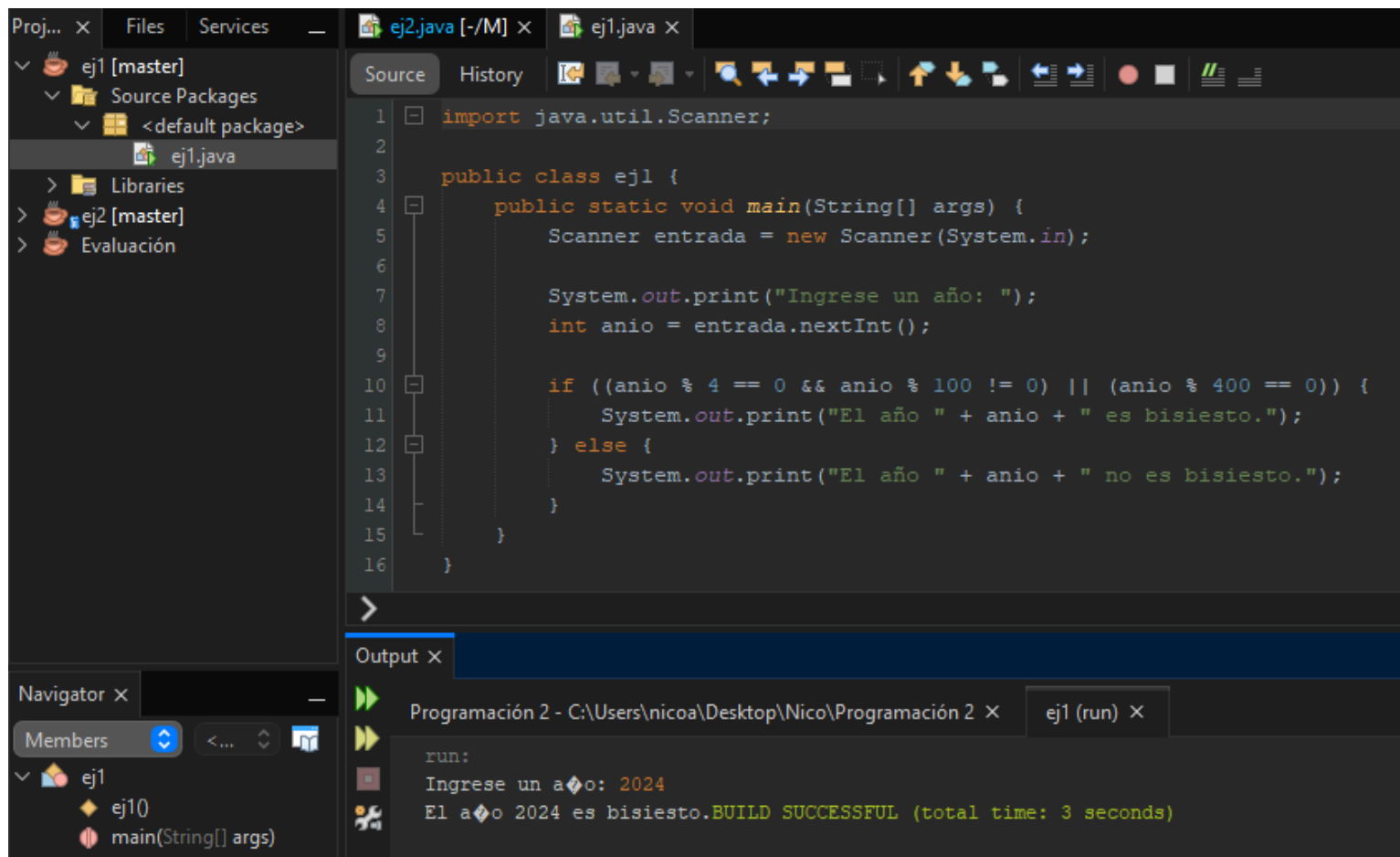
Escribe un programa en Java que solicite al usuario un año y determine si es bisiesto. Un año es bisiesto si es divisible por 4, pero no por 100, salvo que sea divisible por 400. Ejemplo de entrada/salida:

Ingrese un año: 2024

El año 2024 es bisiesto.

Ingrese un año: 1900

El año 1900 no es bisiesto.



```
1 import java.util.Scanner;
2
3 public class ej1 {
4     public static void main(String[] args) {
5         Scanner entrada = new Scanner(System.in);
6
7         System.out.print("Ingrese un año: ");
8         int anio = entrada.nextInt();
9
10        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
11            System.out.print("El año " + anio + " es bisiesto.");
12        } else {
13            System.out.print("El año " + anio + " no es bisiesto.");
14        }
15    }
16 }
```

Output

Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 × ej1 (run) ×

run:
Ingrese un año: 2024
El año 2024 es bisiesto. BUILD SUCCESSFUL (total time: 3 seconds)

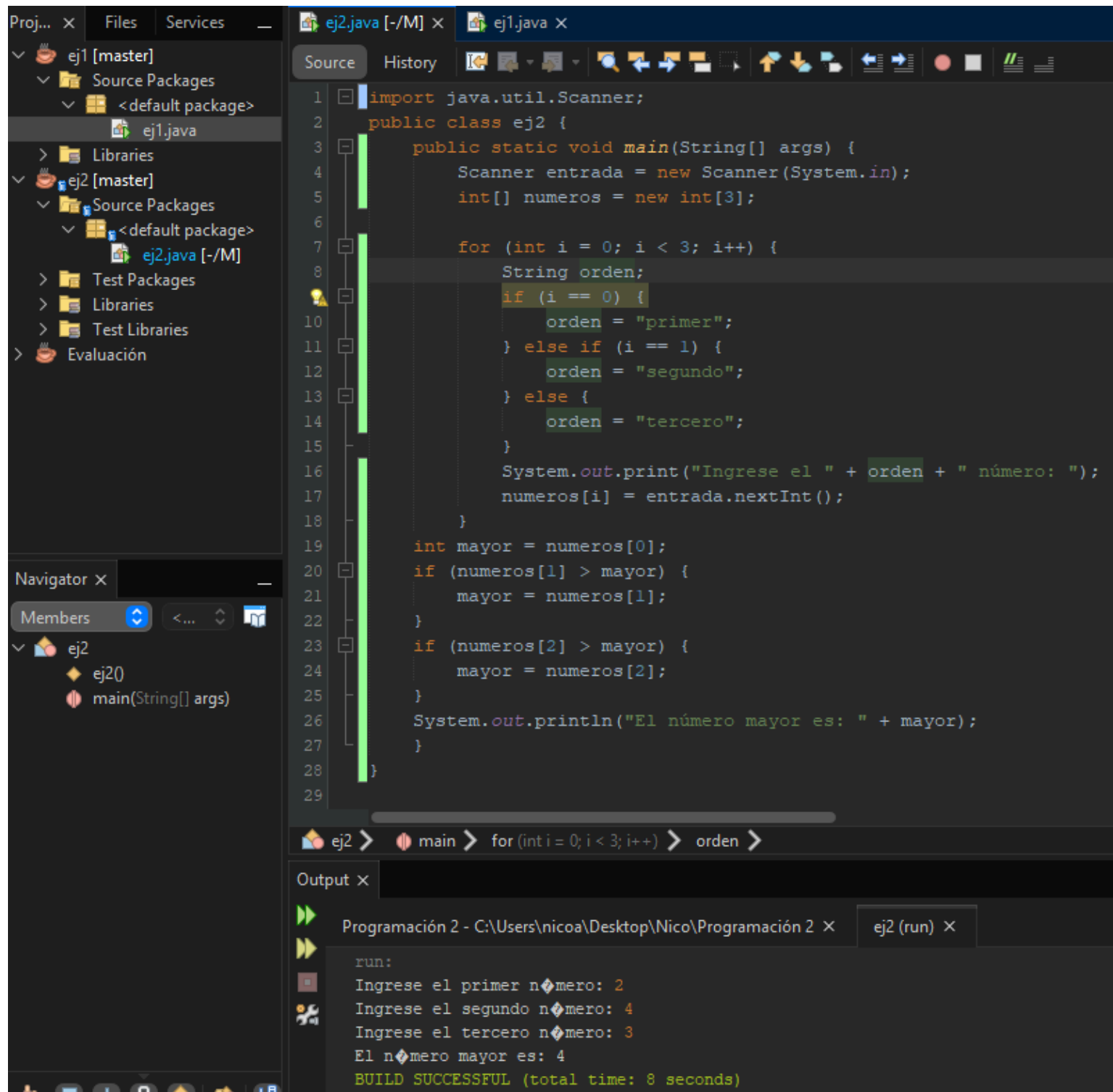
2. Determinar el Mayor de Tres Números. Escribe un programa en Java que pida al usuario tres números enteros y determine cuál es el mayor. Ejemplo de entrada/salida:

Ingrese el primer número: 8

Ingrese el segundo número: 12

Ingrese el tercer número: 5

El mayor es: 12



```
1 import java.util.Scanner;
2 public class ej2 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         int[] numeros = new int[3];
6
7         for (int i = 0; i < 3; i++) {
8             String orden;
9             if (i == 0) {
10                 orden = "primer";
11             } else if (i == 1) {
12                 orden = "segundo";
13             } else {
14                 orden = "tercero";
15             }
16             System.out.print("Ingrese el " + orden + " número: ");
17             numeros[i] = entrada.nextInt();
18         }
19
20         int mayor = numeros[0];
21         if (numeros[1] > mayor) {
22             mayor = numeros[1];
23         }
24         if (numeros[2] > mayor) {
25             mayor = numeros[2];
26         }
27         System.out.println("El número mayor es: " + mayor);
28     }
29 }
```

Output

```
run:
Ingrese el primer número: 2
Ingrese el segundo número: 4
Ingrese el tercero número: 3
El número mayor es: 4
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"

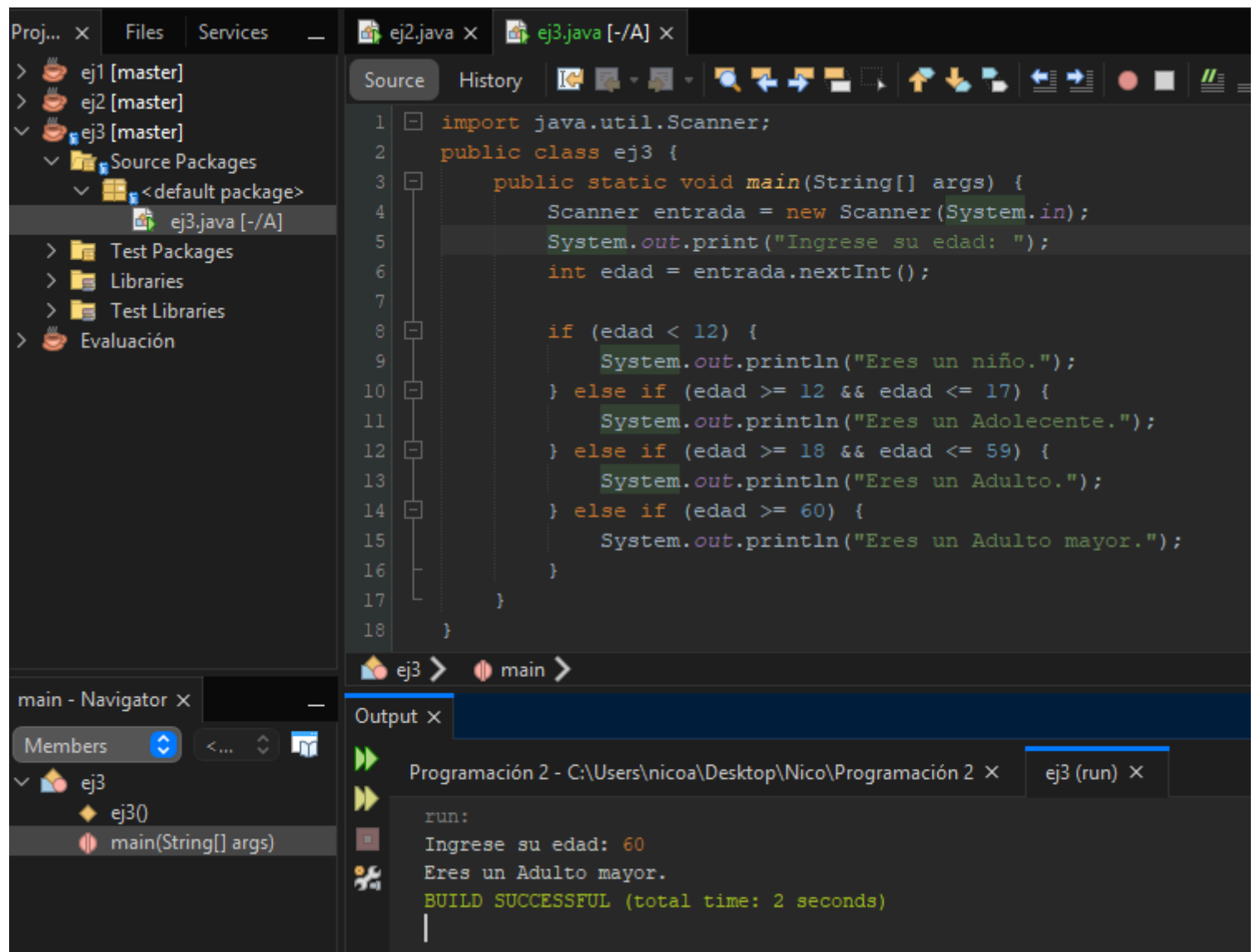
Entre 18 y 59 años: "Adulto"

60 años o más: "Adulto mayor"

Ejemplo de entrada/salida:

Ingrese su edad: 25 Eres un Adulto.

Ingrese su edad: 10 Eres un Niño.



The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'ej3' with a source package containing 'ej3.java [-/A]'. Other packages like 'Test Packages', 'Libraries', and 'Test Libraries' are also visible.
- Source Editor:** Displays the Java code for 'ej3.java'. The code imports 'java.util.Scanner', defines a 'main' method, and uses a 'Scanner' to read user input. It then uses a series of 'if' and 'else if' statements to classify the age into 'Niño', 'Adolescente', 'Adulto', or 'Adulto mayor'.
- Output Console:** Shows the execution of the program. The output is: 'run: Ingrese su edad: 60 Eres un Adulto mayor. BUILD SUCCESSFUL (total time: 2 seconds)'. The console also shows the file path 'Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2' and the command 'ej3 (run)'.

```
1 import java.util.Scanner;
2 public class ej3 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         System.out.print("Ingrese su edad: ");
6         int edad = entrada.nextInt();
7
8         if (edad < 12) {
9             System.out.println("Eres un niño.");
10        } else if (edad >= 12 && edad <= 17) {
11            System.out.println("Eres un Adolescente.");
12        } else if (edad >= 18 && edad <= 59) {
13            System.out.println("Eres un Adulto.");
14        } else if (edad >= 60) {
15            System.out.println("Eres un Adulto mayor.");
16        }
17    }
18 }
```

run:
Ingrese su edad: 60
Eres un Adulto mayor.
BUILD SUCCESSFUL (total time: 2 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

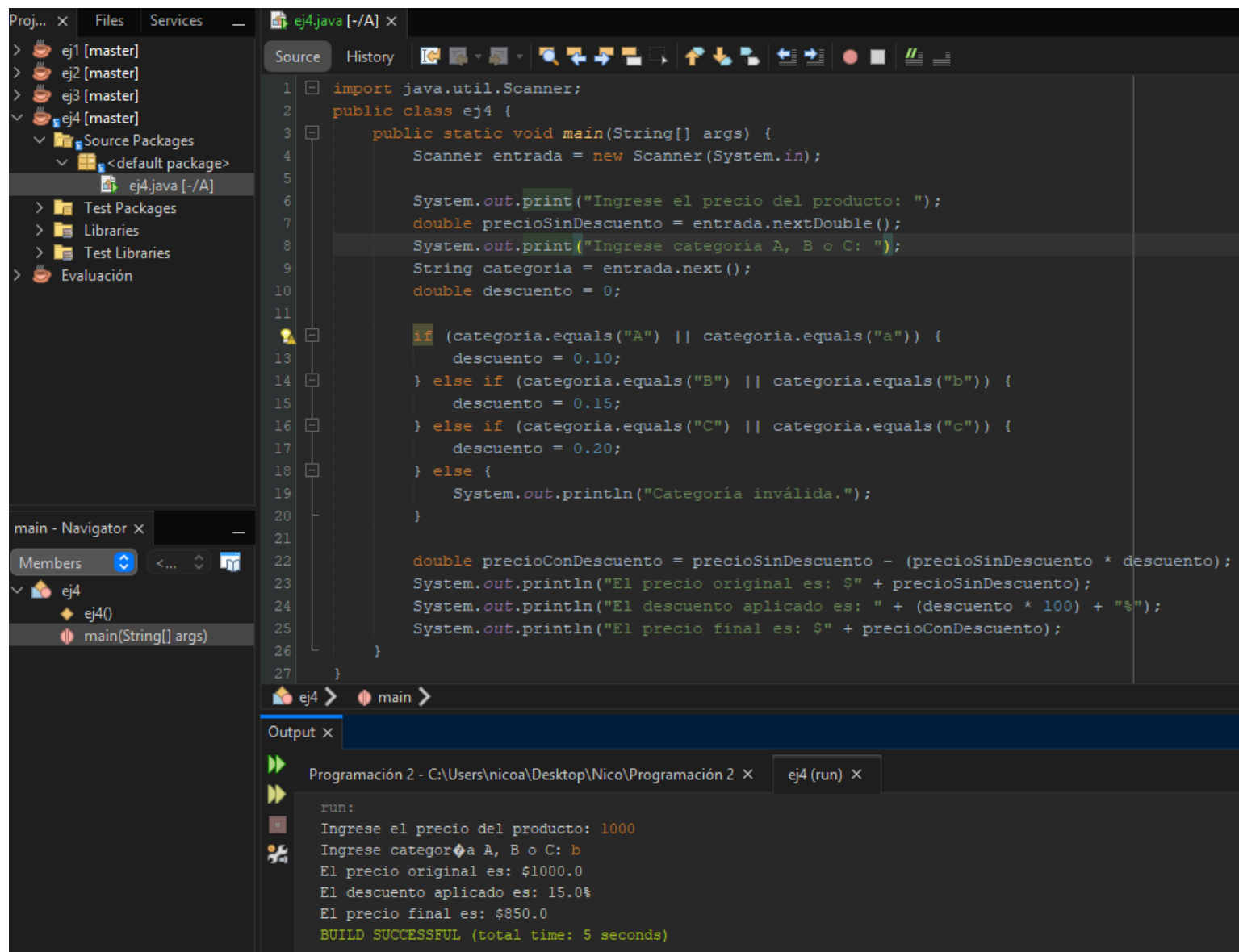
Ejemplo de entrada/salida:

Ingrese el precio del producto: 1000

Ingrese la categoría del producto (A, B o C): B

Descuento aplicado: 15%

Precio final: 850.0



```
1 import java.util.Scanner;
2 public class ej4 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5
6         System.out.print("Ingrese el precio del producto: ");
7         double precioSinDescuento = entrada.nextDouble();
8         System.out.print("Ingrese categoria A, B o C: ");
9         String categoria = entrada.next();
10        double descuento = 0;
11
12        if (categoria.equals("A") || categoria.equals("a")) {
13            descuento = 0.10;
14        } else if (categoria.equals("B") || categoria.equals("b")) {
15            descuento = 0.15;
16        } else if (categoria.equals("C") || categoria.equals("c")) {
17            descuento = 0.20;
18        } else {
19            System.out.println("Categoria inválida.");
20        }
21
22        double precioConDescuento = precioSinDescuento - (precioSinDescuento * descuento);
23        System.out.println("El precio original es: $" + precioSinDescuento);
24        System.out.println("El descuento aplicado es: " + (descuento * 100) + "%");
25        System.out.println("El precio final es: $" + precioConDescuento);
26    }
27 }
```

Output

```
run:
Ingrese el precio del producto: 1000
Ingrese categoria A, B o C: b
El precio original es: $1000.0
El descuento aplicado es: 15.0%
El precio final es: $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.

Ejemplo de entrada/salida:

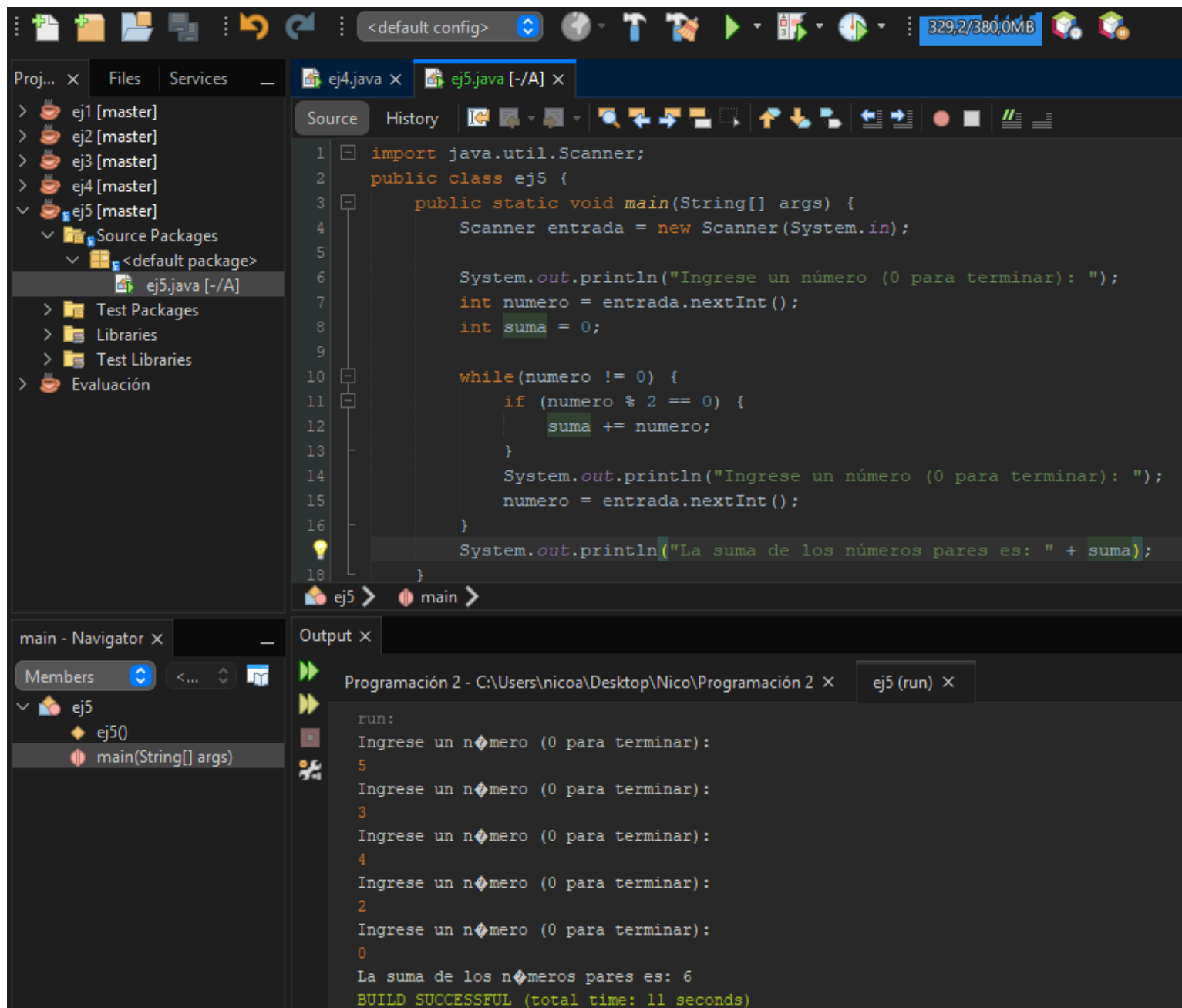
Ingrese un número (0 para terminar): 4

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



```
1 import java.util.Scanner;
2 public class ej5 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5
6         System.out.println("Ingrese un número (0 para terminar): ");
7         int numero = entrada.nextInt();
8         int suma = 0;
9
10        while(numero != 0) {
11            if (numero % 2 == 0) {
12                suma += numero;
13            }
14            System.out.println("Ingrese un número (0 para terminar): ");
15            numero = entrada.nextInt();
16        }
17        System.out.println("La suma de los números pares es: " + suma);
18    }
19 }
```

Output:

```
run:
Ingrese un número (0 para terminar):
5
Ingrese un número (0 para terminar):
3
Ingrese un número (0 para terminar):
4
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: 11 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. Ejemplo de entrada/salida:

Ingrese el número 1: -5

...

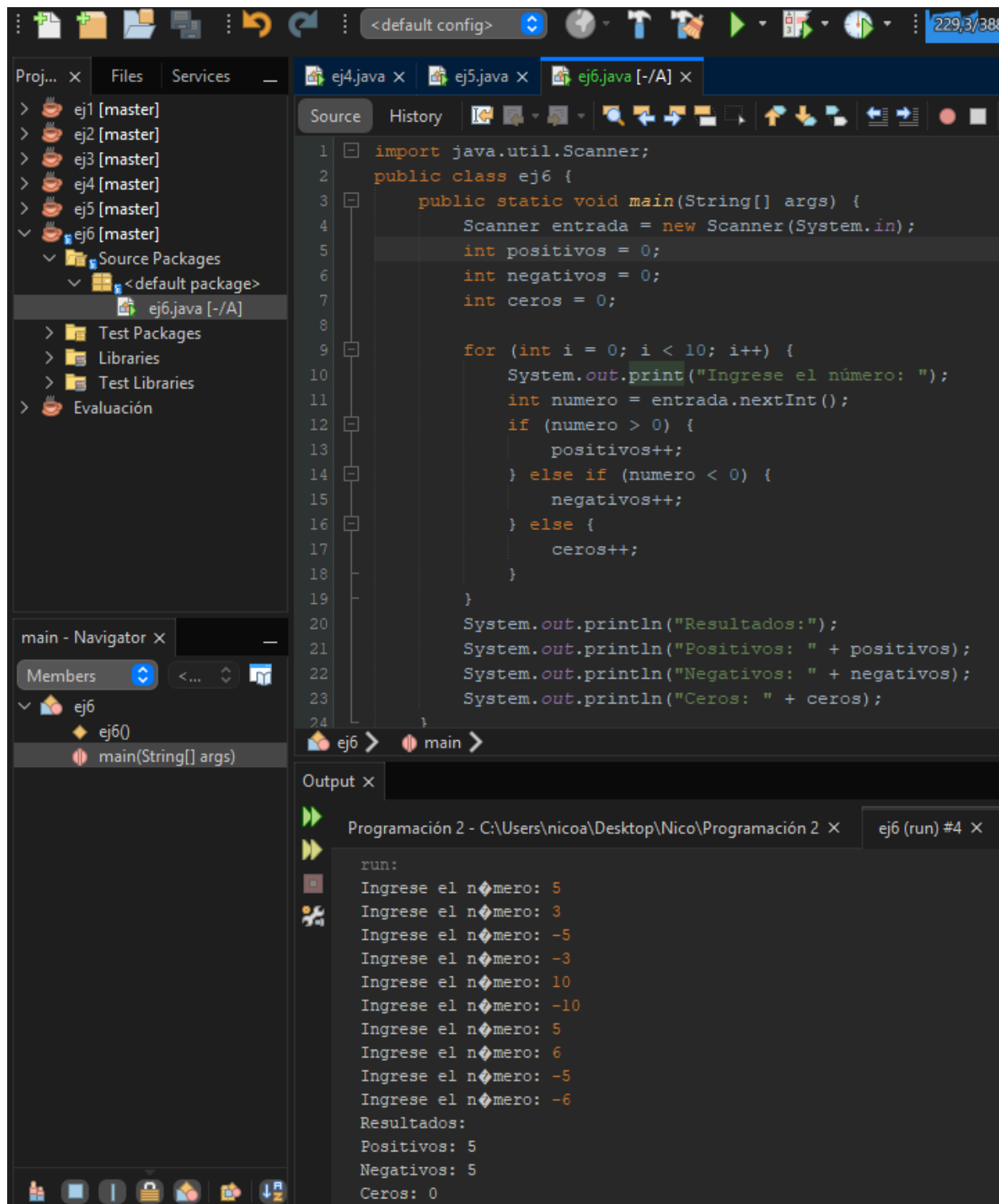
Ingrese el número 10: -8

Resultados:

Positivos: 4

Negativos: 4

Ceros: 2



```
1 import java.util.Scanner;
2 public class ej6 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         int positivos = 0;
6         int negativos = 0;
7         int ceros = 0;
8
9         for (int i = 0; i < 10; i++) {
10             System.out.print("Ingrese el número: ");
11             int numero = entrada.nextInt();
12             if (numero > 0) {
13                 positivos++;
14             } else if (numero < 0) {
15                 negativos++;
16             } else {
17                 ceros++;
18             }
19         }
20         System.out.println("Resultados:");
21         System.out.println("Positivos: " + positivos);
22         System.out.println("Negativos: " + negativos);
23         System.out.println("Ceros: " + ceros);
24     }
25 }
```

Output

```
run:
Ingrese el número: 5
Ingrese el número: 3
Ingrese el número: -5
Ingrese el número: -3
Ingrese el número: 10
Ingrese el número: -10
Ingrese el número: 5
Ingrese el número: 6
Ingrese el número: -5
Ingrese el número: -6
Resultados:
Positivos: 5
Negativos: 5
Ceros: 0
```


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.

Ejemplo de entrada/salida:

Ingrese una nota (0-10): 15

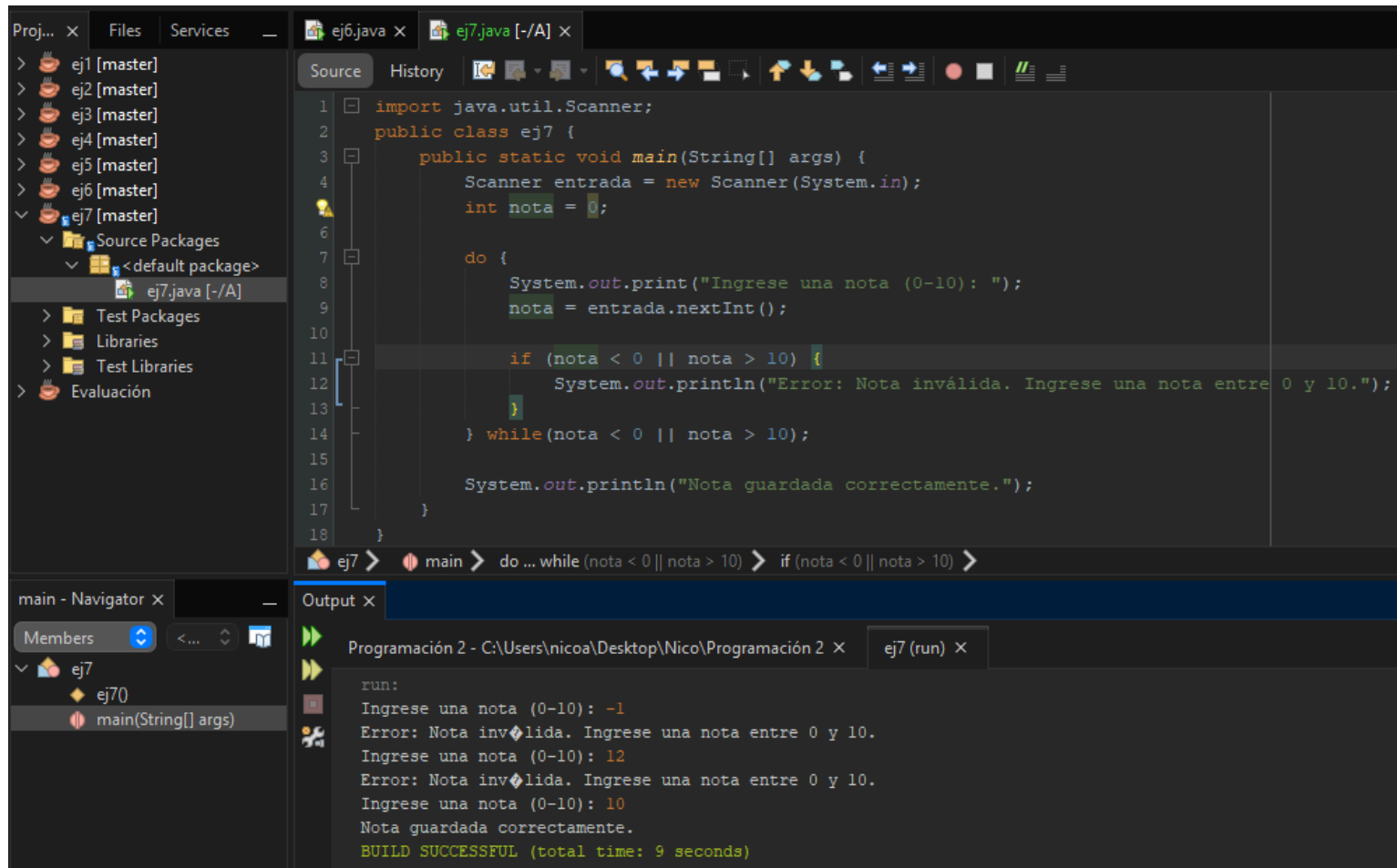
Error: Nota inválida. Ingrese una nota entre 0 y 10.

Ingrese una nota (0-10): -2

Error: Nota inválida. Ingrese una nota entre 0 y 10.

Ingrese una nota (0-10): 8

Nota guardada correctamente.



```
1 import java.util.Scanner;
2 public class ej7 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         int nota = 0;
6
7         do {
8             System.out.print("Ingrese una nota (0-10): ");
9             nota = entrada.nextInt();
10
11             if (nota < 0 || nota > 10) {
12                 System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y 10.");
13             }
14         } while (nota < 0 || nota > 10);
15
16         System.out.println("Nota guardada correctamente.");
17     }
18 }
```

main - Navigator ×

Members

ej7

ej70

main(String[] args)

Output ×

Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 ×

ej7 (run) ×

run:

Ingrese una nota (0-10): -1

Error: Nota inválida. Ingrese una nota entre 0 y 10.

Ingrese una nota (0-10): 12

Error: Nota inválida. Ingrese una nota entre 0 y 10.

Ingrese una nota (0-10): 10

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:

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

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

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.

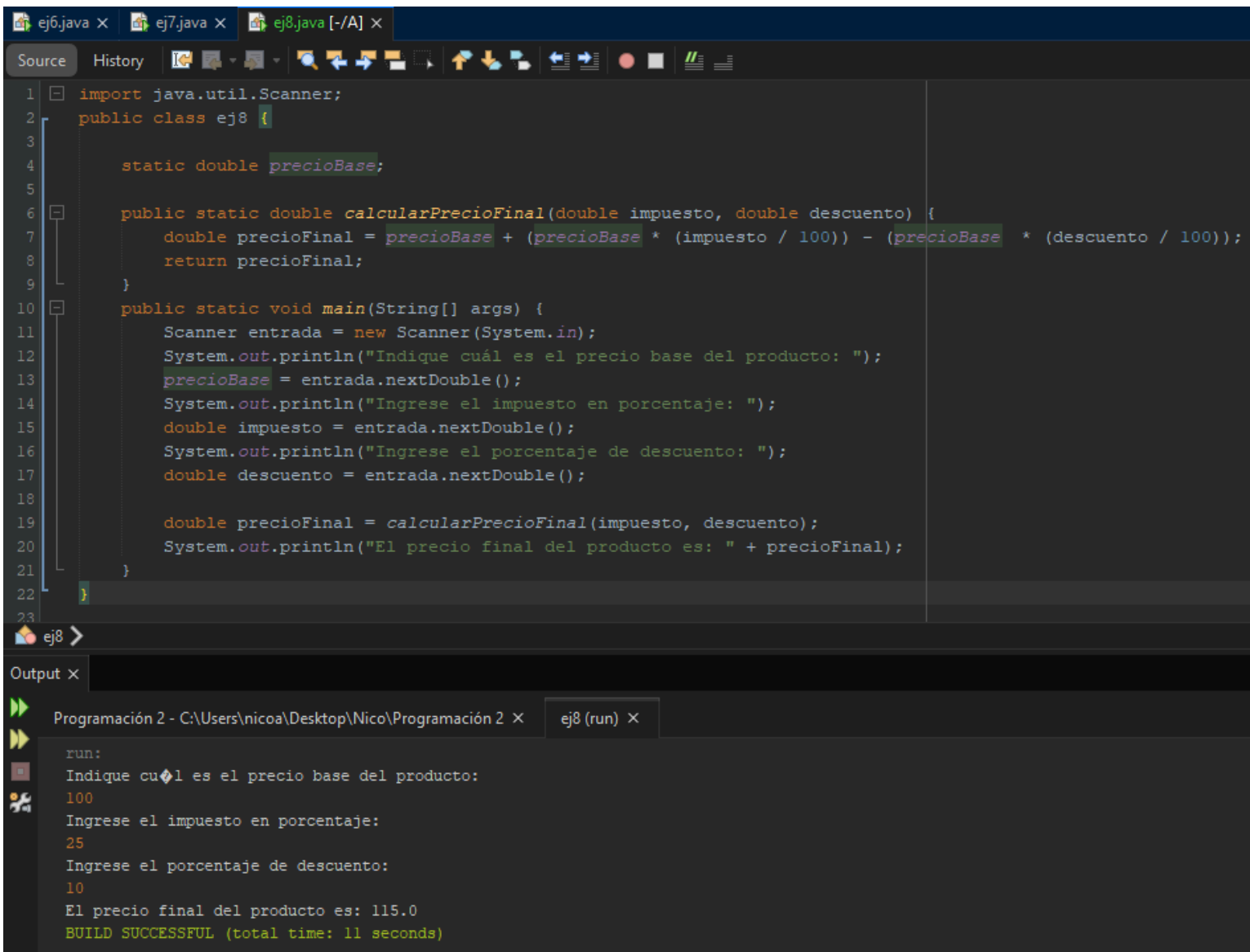
Ejemplo de entrada/salida:

Ingrese el precio base del producto: 100

Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): 10

Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): 5

El precio final del producto es: 105.0



```
1 import java.util.Scanner;
2 public class ej8 {
3
4     static double precioBase;
5
6     public static double calcularPrecioFinal(double impuesto, double descuento) {
7         double precioFinal = precioBase + (precioBase * (impuesto / 100)) - (precioBase * (descuento / 100));
8         return precioFinal;
9     }
10
11     public static void main(String[] args) {
12         Scanner entrada = new Scanner(System.in);
13         System.out.println("Indique cuál es el precio base del producto: ");
14         precioBase = entrada.nextDouble();
15         System.out.println("Ingrese el impuesto en porcentaje: ");
16         double impuesto = entrada.nextDouble();
17         System.out.println("Ingrese el porcentaje de descuento: ");
18         double descuento = entrada.nextDouble();
19
20         double precioFinal = calcularPrecioFinal(impuesto, descuento);
21         System.out.println("El precio final del producto es: " + precioFinal);
22     }
23 }
```

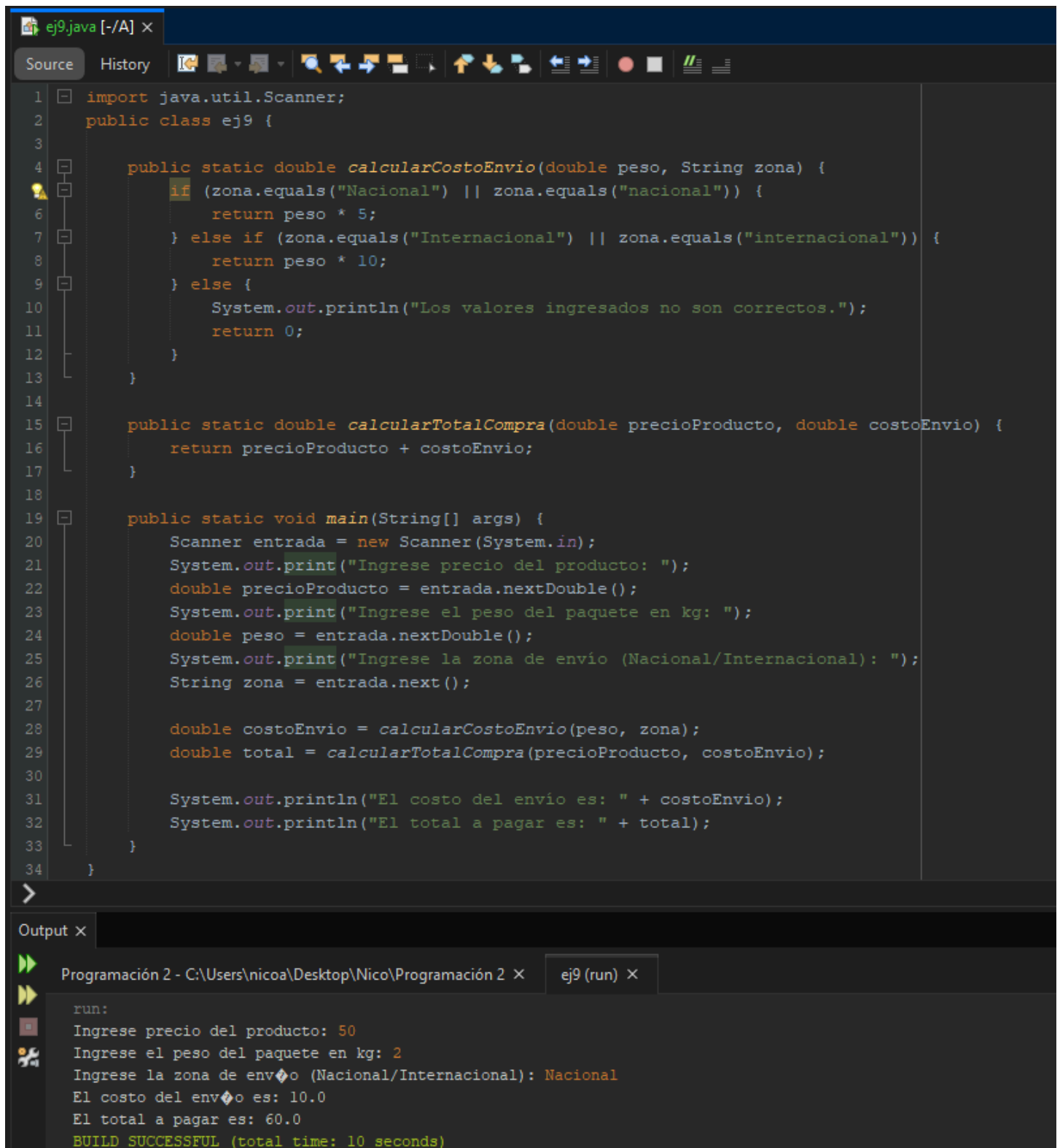
Output ×

```
run:
Indique cuál es el precio base del producto:
100
Ingrese el impuesto en porcentaje:
25
Ingrese el porcentaje de descuento:
10
El precio final del producto es: 115.0
BUILD SUCCESSFUL (total time: 11 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.



```
ej9.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class ej9 {
3
4     public static double calcularCostoEnvio(double peso, String zona) {
5         if (zona.equals("Nacional") || zona.equals("nacional")) {
6             return peso * 5;
7         } else if (zona.equals("Internacional") || zona.equals("internacional")) {
8             return peso * 10;
9         } else {
10             System.out.println("Los valores ingresados no son correctos.");
11             return 0;
12         }
13     }
14
15     public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
16         return precioProducto + costoEnvio;
17     }
18
19     public static void main(String[] args) {
20         Scanner entrada = new Scanner(System.in);
21         System.out.print("Ingrese precio del producto: ");
22         double precioProducto = entrada.nextDouble();
23         System.out.print("Ingrese el peso del paquete en kg: ");
24         double peso = entrada.nextDouble();
25         System.out.print("Ingrese la zona de envío (Nacional/Internacional): ");
26         String zona = entrada.next();
27
28         double costoEnvio = calcularCostoEnvio(peso, zona);
29         double total = calcularTotalCompra(precioProducto, costoEnvio);
30
31         System.out.println("El costo del envío es: " + costoEnvio);
32         System.out.println("El total a pagar es: " + total);
33     }
34 }
```

Output x

Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 x ej9 (run) x

run:
Ingrese precio del producto: 50
Ingrese el peso del paquete en kg: 2
Ingrese la zona de envío (Nacional/Internacional): Nacional
El costo del envío es: 10.0
El total a pagar es: 60.0
BUILD SUCCESSFUL (total time: 10 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.

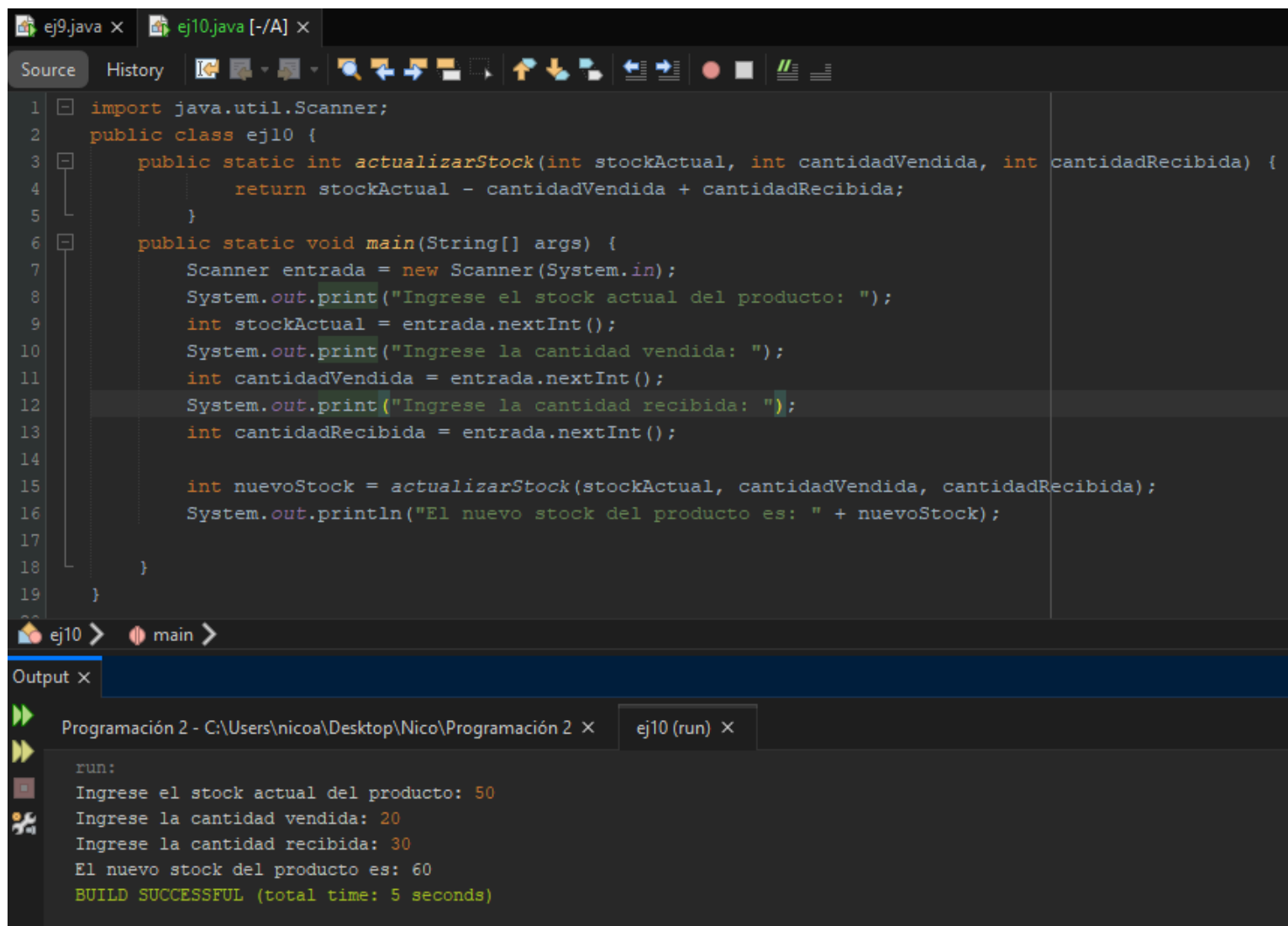
Ejemplo de entrada/salida:

Ingrese el stock actual del producto: 50

Ingrese la cantidad vendida: 20

Ingrese la cantidad recibida: 30

El nuevo stock del producto es: 60



```
ej9.java x ej10.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class ej10 {
3     public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
4         return stockActual - cantidadVendida + cantidadRecibida;
5     }
6     public static void main(String[] args) {
7         Scanner entrada = new Scanner(System.in);
8         System.out.print("Ingrese el stock actual del producto: ");
9         int stockActual = entrada.nextInt();
10        System.out.print("Ingrese la cantidad vendida: ");
11        int cantidadVendida = entrada.nextInt();
12        System.out.print("Ingrese la cantidad recibida: ");
13        int cantidadRecibida = entrada.nextInt();
14
15        int nuevoStock = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
16        System.out.println("El nuevo stock del producto es: " + nuevoStock);
17    }
18 }
19
ej10 > main >
Output x
Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 x ej10 (run) x
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)
```

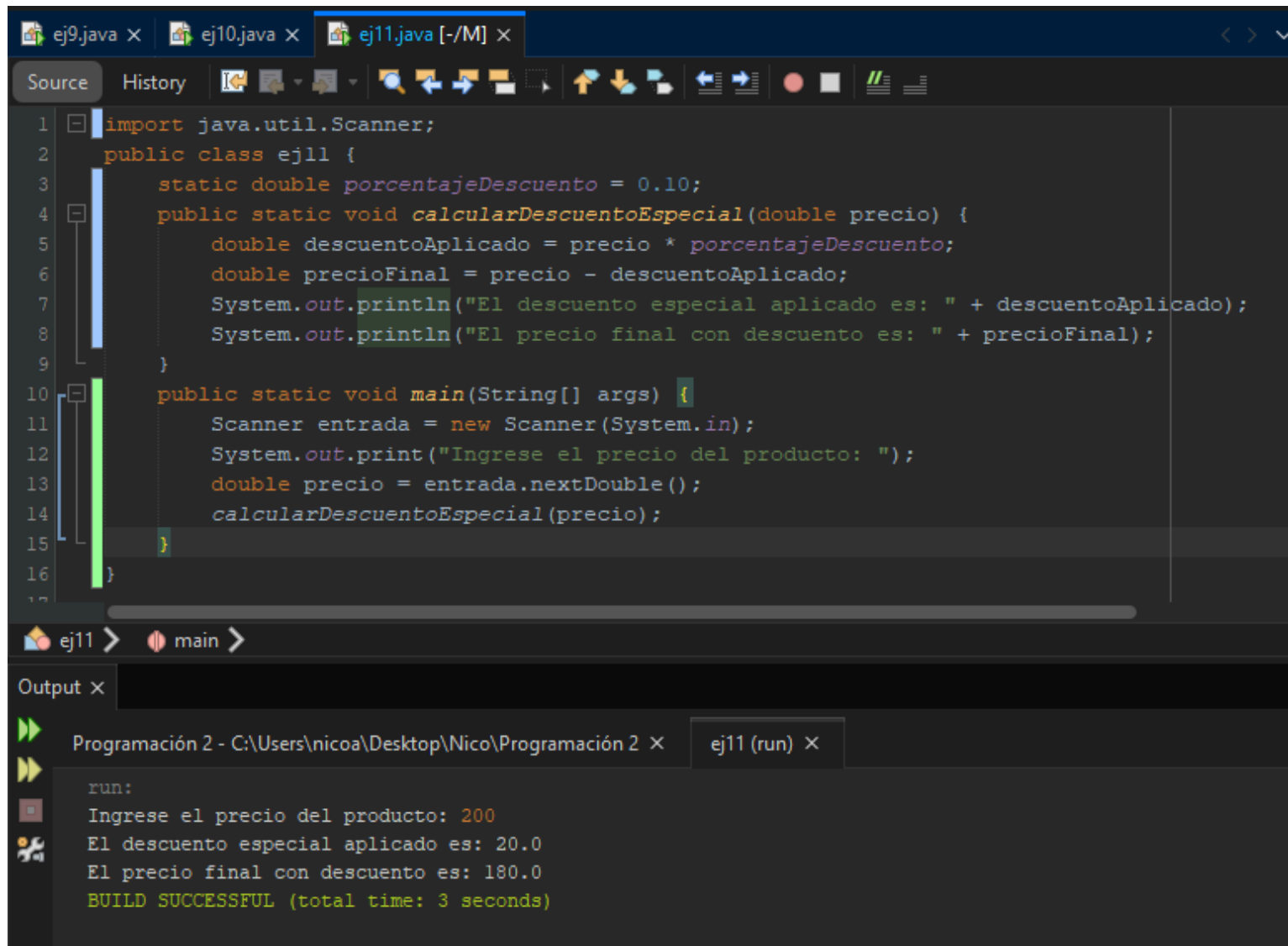
11. Cálculo de descuento especial usando variable global. Declara una variable global Ejemplo de entrada/salida: = 0.10. Luego, crea un método `calcularDescuentoEspecial(double precio)` que use la variable global para calcular el descuento especial del 10%. Dentro del método, declara una variable local `descuentoAplicado`, almacena el valor del descuento y muestra el precio final con descuento.

Ejemplo de entrada/salida:

Ingrese el precio del producto: 200

El descuento especial aplicado es: 20.0

El precio final con descuento es: 180.0



```
1 import java.util.Scanner;
2 public class ej11 {
3     static double porcentajeDescuento = 0.10;
4     public static void calcularDescuentoEspecial(double precio) {
5         double descuentoAplicado = precio * porcentajeDescuento;
6         double precioFinal = precio - descuentoAplicado;
7         System.out.println("El descuento especial aplicado es: " + descuentoAplicado);
8         System.out.println("El precio final con descuento es: " + precioFinal);
9     }
10    public static void main(String[] args) {
11        Scanner entrada = new Scanner(System.in);
12        System.out.print("Ingrese el precio del producto: ");
13        double precio = entrada.nextDouble();
14        calcularDescuentoEspecial(precio);
15    }
16 }
```

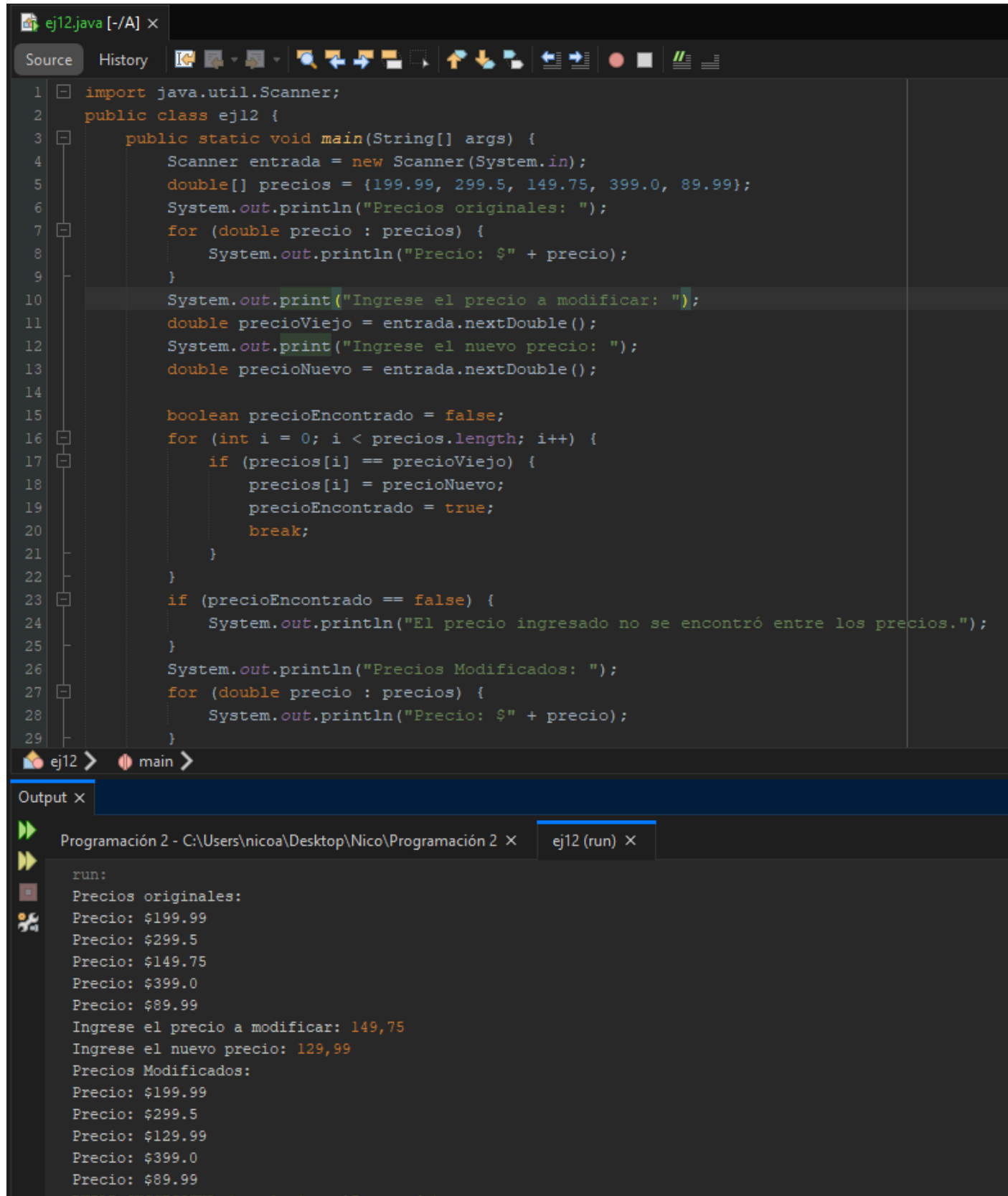
Output ×

Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 × ej11 (run) ×

```
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 import java.util.Scanner;
2 public class ej12 {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
6         System.out.println("Precios originales: ");
7         for (double precio : precios) {
8             System.out.println("Precio: $" + precio);
9         }
10        System.out.print("Ingrese el precio a modificar: ");
11        double precioViejo = entrada.nextDouble();
12        System.out.print("Ingrese el nuevo precio: ");
13        double precioNuevo = entrada.nextDouble();
14
15        boolean precioEncontrado = false;
16        for (int i = 0; i < precios.length; i++) {
17            if (precios[i] == precioViejo) {
18                precios[i] = precioNuevo;
19                precioEncontrado = true;
20                break;
21            }
22        }
23        if (precioEncontrado == false) {
24            System.out.println("El precio ingresado no se encontró entre los precios.");
25        }
26        System.out.println("Precios Modificados: ");
27        for (double precio : precios) {
28            System.out.println("Precio: $" + precio);
29        }
30    }
31}
```

Output ×

Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 × ej12 (run) ×

run:

Precios originales:

Precio: \$199.99

Precio: \$299.5

Precio: \$149.75

Precio: \$399.0

Precio: \$89.99

Ingrese el precio a modificar: 149,75

Ingrese el nuevo precio: 129,99

Precios Modificados:

Precio: \$199.99

Precio: \$299.5

Precio: \$129.99

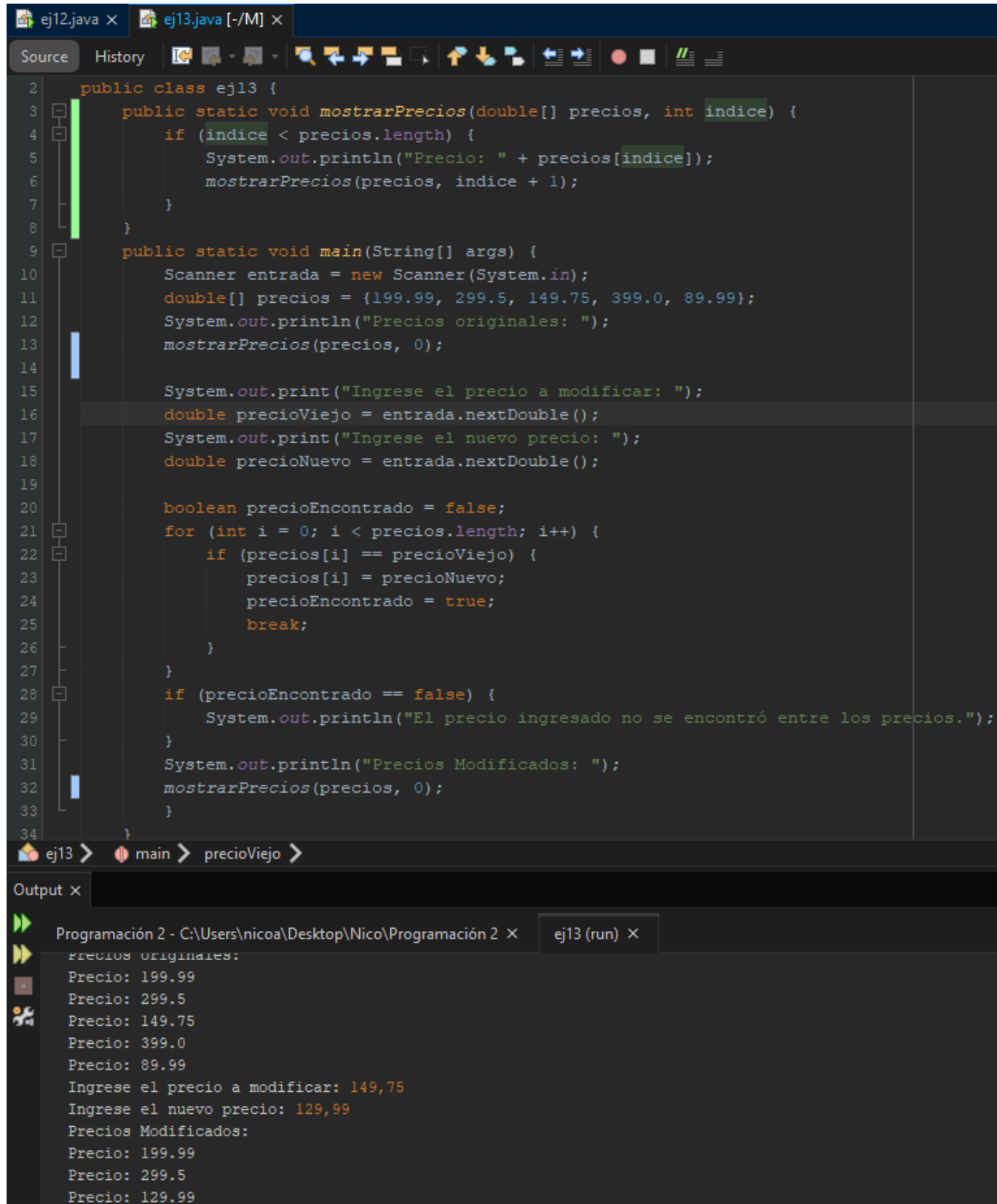
Precio: \$399.0

Precio: \$89.99

BUILD SUCCESSFUL (total time: 17 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.



```
ej12.java x ej13.java [-/M] x
Source History
2 public class ej13 {
3     public static void mostrarPrecios(double[] precios, int indice) {
4         if (indice < precios.length) {
5             System.out.println("Precio: " + precios[indice]);
6             mostrarPrecios(precios, indice + 1);
7         }
8     }
9     public static void main(String[] args) {
10         Scanner entrada = new Scanner(System.in);
11         double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
12         System.out.println("Precios originales: ");
13         mostrarPrecios(precios, 0);
14
15         System.out.print("Ingrese el precio a modificar: ");
16         double precioViejo = entrada.nextDouble();
17         System.out.print("Ingrese el nuevo precio: ");
18         double precioNuevo = entrada.nextDouble();
19
20         boolean precioEncontrado = false;
21         for (int i = 0; i < precios.length; i++) {
22             if (precios[i] == precioViejo) {
23                 precios[i] = precioNuevo;
24                 precioEncontrado = true;
25                 break;
26             }
27         }
28         if (precioEncontrado == false) {
29             System.out.println("El precio ingresado no se encontró entre los precios.");
30         }
31         System.out.println("Precios Modificados: ");
32         mostrarPrecios(precios, 0);
33     }
34 }
ej13 > main > precioViejo >
Output x
Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 x ej13 (run) x
Precios originales:
Precio: 199.99
Precio: 299.5
Precio: 149.75
Precio: 399.0
Precio: 89.99
Ingrese el precio a modificar: 149,75
Ingrese el nuevo precio: 129,99
Precios Modificados:
Precio: 199.99
Precio: 299.5
Precio: 129.99
```