



TRABAJO PRÁCTICO 1: INTRODUCCIÓN A JAVA



Resolución de los ejercicios:

1.a. Verificación de la instalación del JDK

Para comprobar que Java JDK está correctamente instalado en el sistema, se ejecutó el siguiente comando en la terminal:

```
java -version
```

Este comando permite verificar la versión del JDK instalada y confirmar que Java está configurado en el entorno del sistema. La salida esperada debe mostrar la versión actual de Java.

```
Microsoft Windows [Versión 10.0.26100.1742]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>java -version
java version "11.0.28" 2025-07-15 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.28+12-LTS-279)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.28+12-LTS-279, mixed mode)

C:\Users\Usuario>
```

1.b y c. Apertura de NetBeans y creación de nuevo proyecto y configuración de modo oscuro.

Se abrió el entorno de desarrollo NetBeans y se creó un nuevo proyecto Java siguiendo estos pasos:

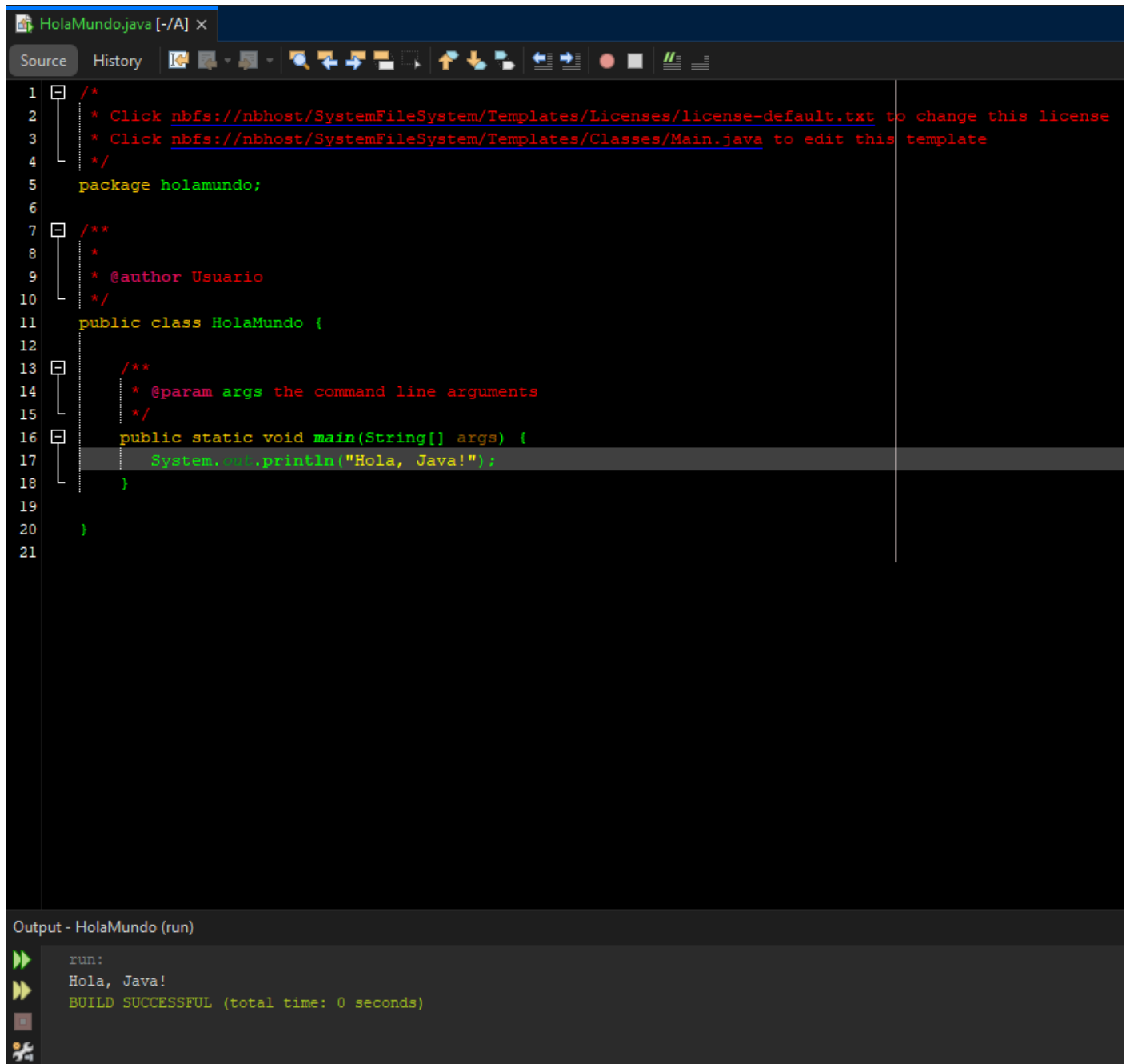
1. Ir a File > New Project.
2. Seleccionar Java > Java Application.
3. Asignar el nombre del proyecto.
4. Finalizar con el botón **Finish**.

Para mejorar la visualización, se activó el modo oscuro desde el menú Tools > Options > Appearance > Look and Feel. Esta configuración no es obligatoria pero puede ser útil para trabajar de forma más cómoda.

2. Escribir y ejecutar un programa básico en Java.

Se creó una nueva clase en Java llamada HolaMundo, cuyo objetivo es imprimir un mensaje en pantalla. Este ejercicio permite familiarizarse con la estructura básica de un programa Java: uso de clases, método main y salida por consola.

El resultado fue el siguiente:



```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package holamundo;
6
7  /**
8   *
9   * @author Usuario
10  */
11  public class HolaMundo {
12
13      /**
14       * @param args the command line arguments
15      */
16      public static void main(String[] args) {
17          System.out.println("Hola, Java!");
18      }
19
20  }
21
```

Output - HolaMundo (run)

```
run:
Hola, Java!
BUILD SUCCESSFUL (total time: 0 seconds)
```

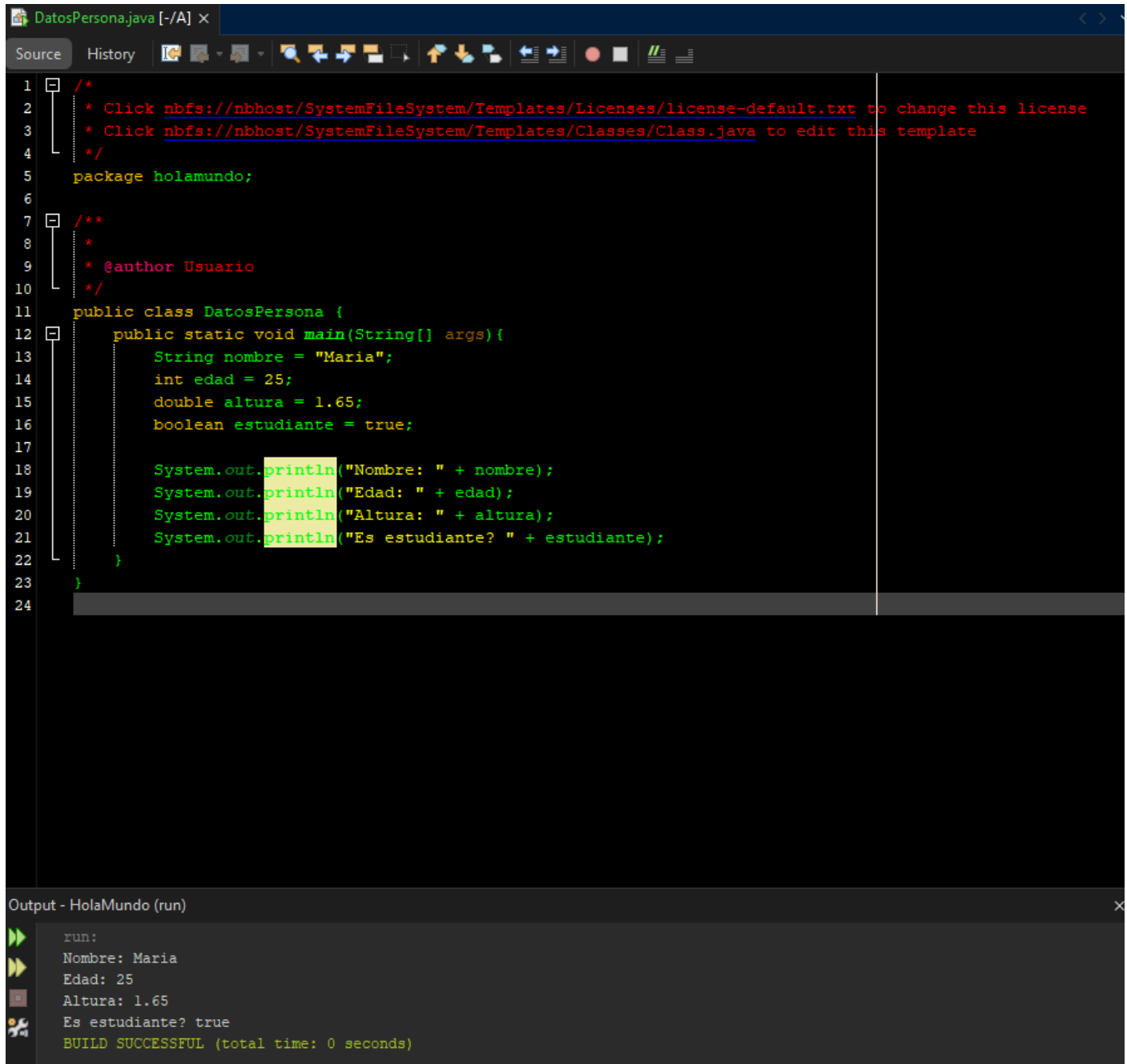
3. Declaración de variables y salida por consola

En este ejercicio se trabajó con **declaración de variables** de distintos tipos en Java, que es fundamental para representar información. Se usaron los siguientes tipos:

- String para texto (nombre)

- int para números enteros (edad)
- double para números con decimales (altura)
- boolean para valores lógicos verdadero/falso (estudiante)

El código utilizado fue:



The screenshot shows an IDE window titled 'DatosPersona.java [-/A] x'. The code is as follows:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package holamundo;
6
7  /**
8   *
9   * @author Usuario
10  */
11  public class DatosPersona {
12      public static void main(String[] args){
13          String nombre = "Maria";
14          int edad = 25;
15          double altura = 1.65;
16          boolean estudiante = true;
17
18          System.out.println("Nombre: " + nombre);
19          System.out.println("Edad: " + edad);
20          System.out.println("Altura: " + altura);
21          System.out.println("Es estudiante? " + estudiante);
22      }
23  }
24
```

Below the code editor is the 'Output - HolaMundo (run)' window, which displays the following output:

```
run:
Nombre: Maria
Edad: 25
Altura: 1.65
Es estudiante? true
BUILD SUCCESSFUL (total time: 0 seconds)
```

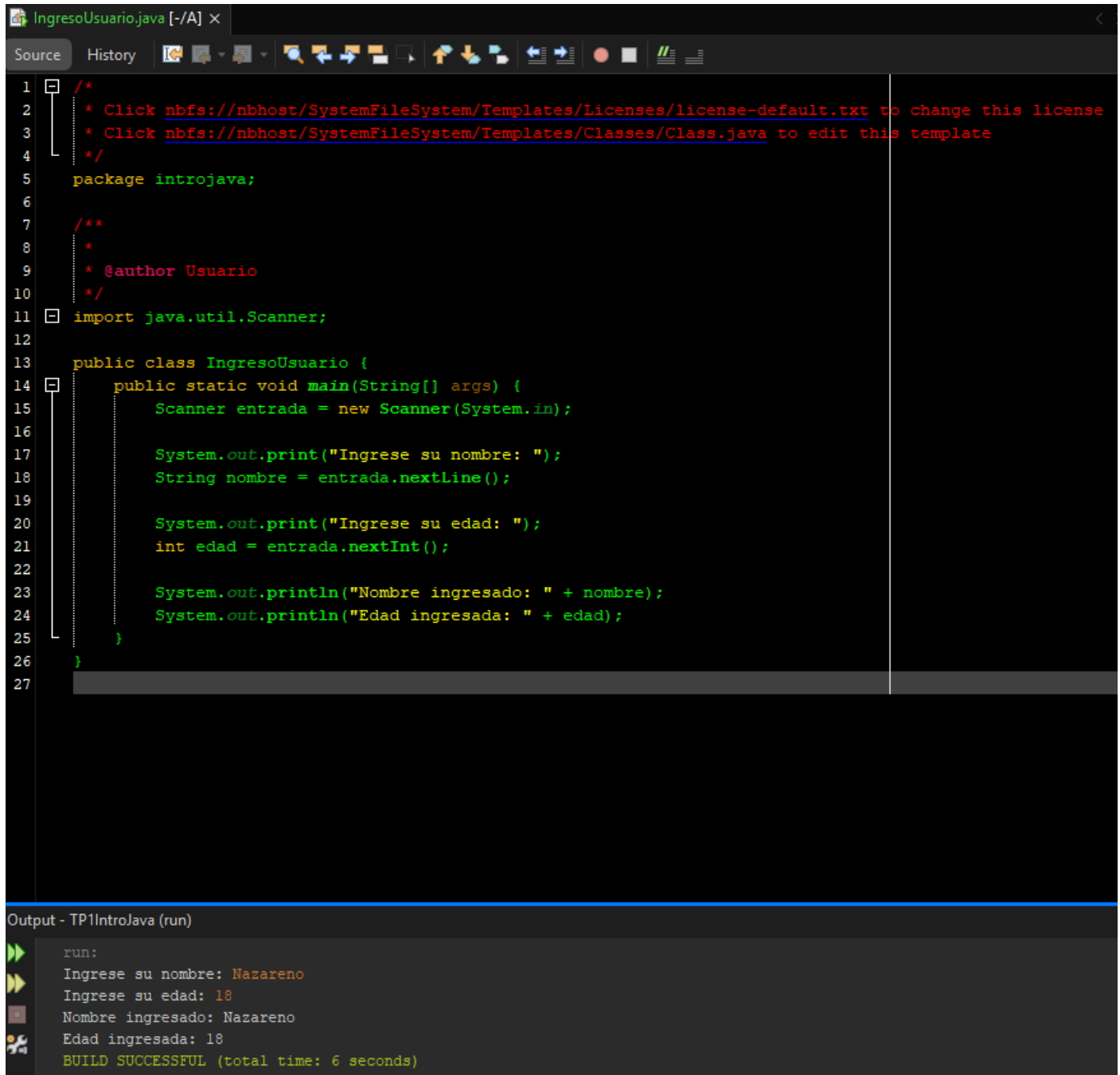
4. Captura de datos con Scanner

En este ejercicio se utilizó la clase Scanner de Java para **leer datos ingresados por el usuario** desde el teclado. Esta técnica es muy útil cuando se quiere interactuar con el usuario durante la ejecución del programa.

Se pidieron dos datos:

- El **nombre** (tipo String)
- La **edad** (tipo int)

El código fue:



```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package introjava;
6
7  /**
8   *
9   * @author Usuario
10  */
11  import java.util.Scanner;
12
13  public class IngresoUsuario {
14      public static void main(String[] args) {
15          Scanner entrada = new Scanner(System.in);
16
17          System.out.print("Ingrese su nombre: ");
18          String nombre = entrada.nextLine();
19
20          System.out.print("Ingrese su edad: ");
21          int edad = entrada.nextInt();
22
23          System.out.println("Nombre ingresado: " + nombre);
24          System.out.println("Edad ingresada: " + edad);
25      }
26  }
27
```

Output - TP1IntroJava (run)

```
run:
Ingrese su nombre: Nazareno
Ingrese su edad: 18
Nombre ingresado: Nazareno
Edad ingresada: 18
BUILD SUCCESSFUL (total time: 6 seconds)
```

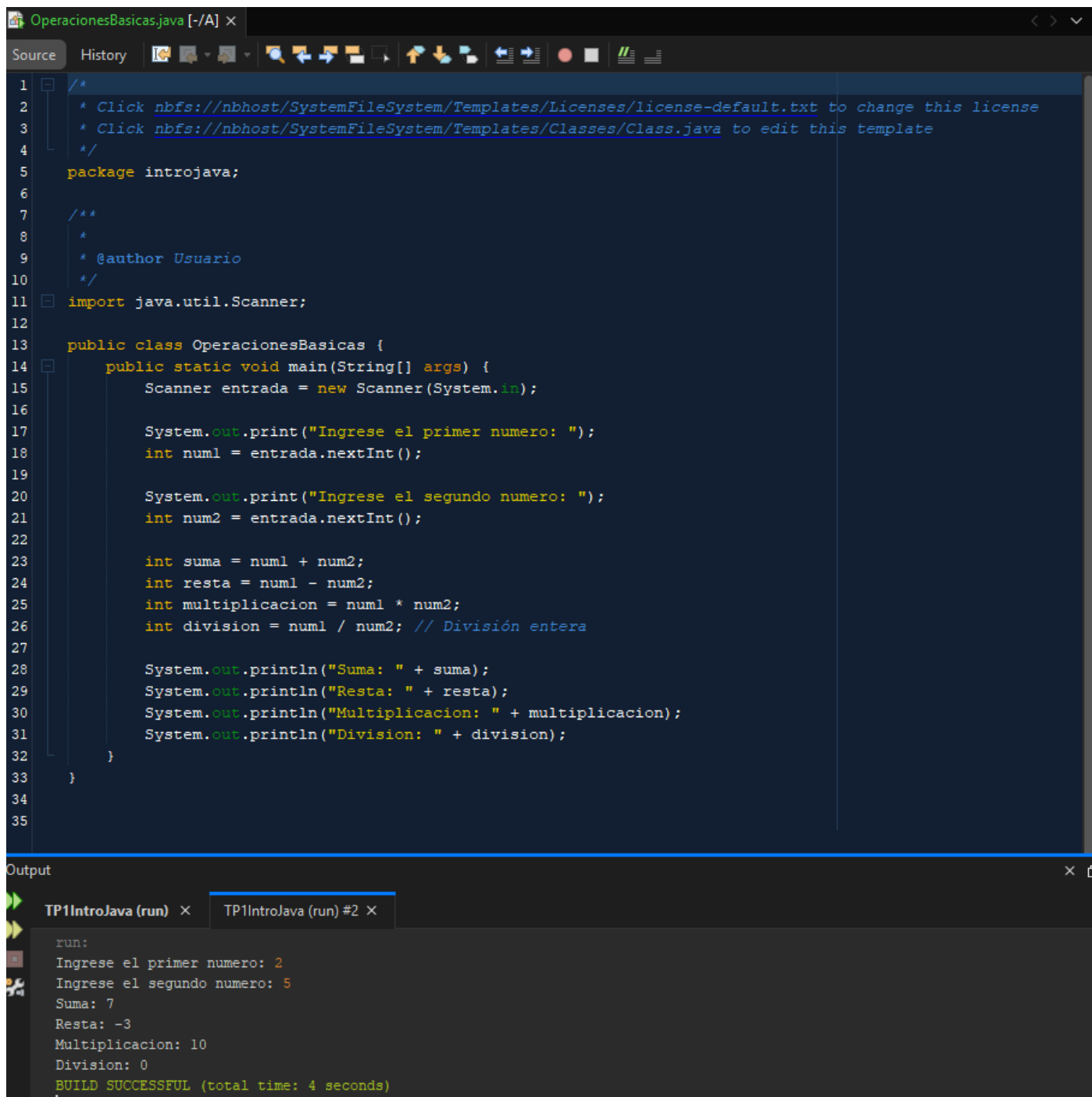
5. Operaciones aritméticas básicas

Este ejercicio introduce el uso de **operadores matemáticos** en Java. El programa solicita dos números enteros y calcula:

- La suma (+)

- La resta (-)
- La multiplicación (*)
- La división (/)

El código fue el siguiente:



The screenshot shows an IDE window titled 'OperacionesBasicas.java [-/A] x'. The code is as follows:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package introjava;
6
7  /**
8   *
9   * @author Usuario
10  */
11  import java.util.Scanner;
12
13  public class OperacionesBasicas {
14      public static void main(String[] args) {
15          Scanner entrada = new Scanner(System.in);
16
17          System.out.print("Ingrese el primer numero: ");
18          int num1 = entrada.nextInt();
19
20          System.out.print("Ingrese el segundo numero: ");
21          int num2 = entrada.nextInt();
22
23          int suma = num1 + num2;
24          int resta = num1 - num2;
25          int multiplicacion = num1 * num2;
26          int division = num1 / num2; // División entera
27
28          System.out.println("Suma: " + suma);
29          System.out.println("Resta: " + resta);
30          System.out.println("Multiplicacion: " + multiplicacion);
31          System.out.println("Division: " + division);
32      }
33  }
```

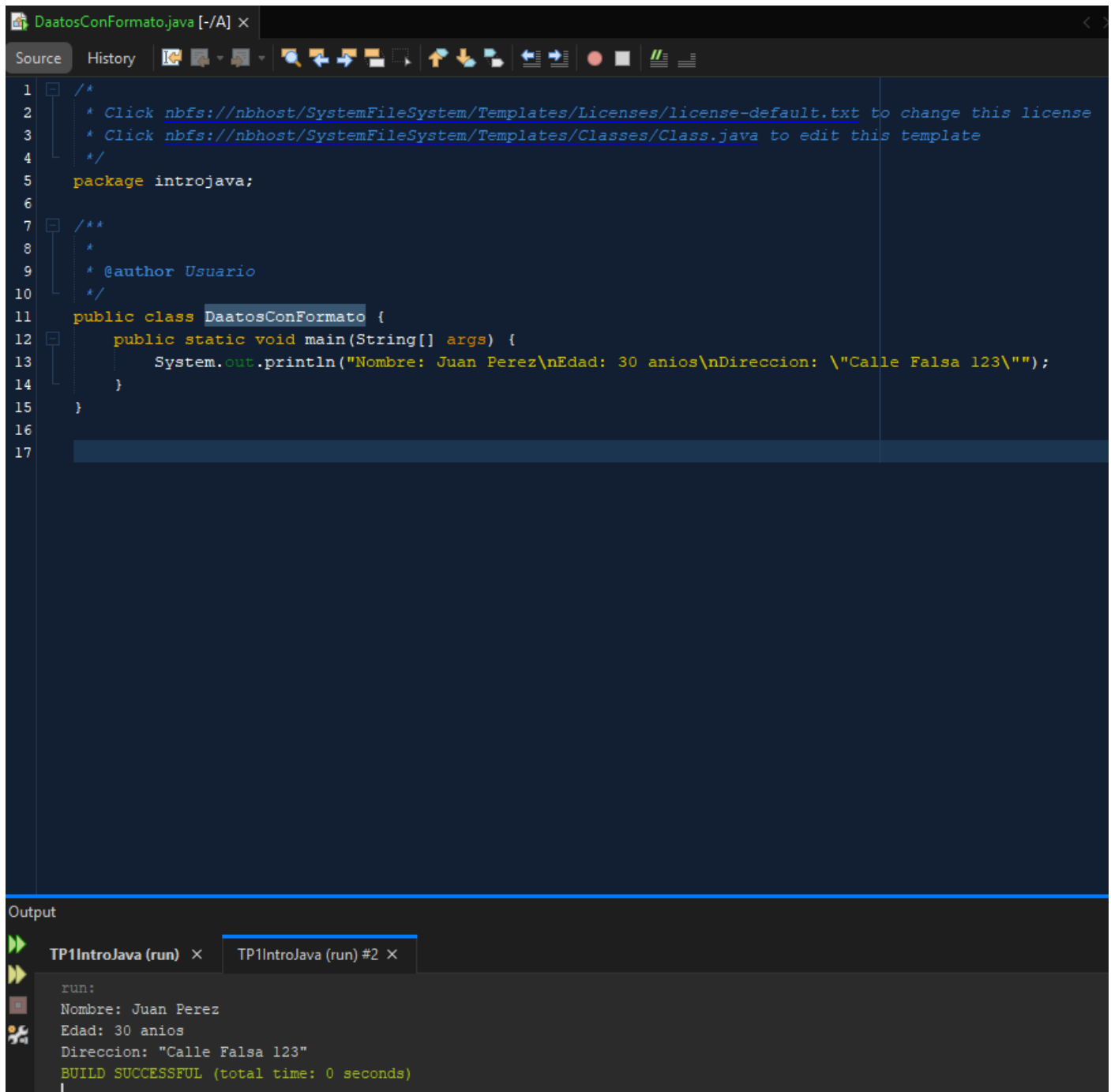
The Output window shows the execution results for 'TP1IntroJava (run) #2':

```
run:
Ingrese el primer numero: 2
Ingrese el segundo numero: 5
Suma: 7
Resta: -3
Multiplicacion: 10
Division: 0
BUILD SUCCESSFUL (total time: 4 seconds)
```

6. Formato de texto con caracteres especiales

Este ejercicio muestra cómo usar **caracteres de escape** para dar formato al texto que se imprime en consola. Se utilizó `\n` para saltos de línea y `\"` para incluir comillas dobles.

El código fue:



The screenshot shows an IDE window titled 'DaatosConFormato.java [-/A] x'. The editor displays the following Java code:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package introjava;
6
7   /**
8    *
9    * @author Usuario
10   */
11  public class DaatosConFormato {
12      public static void main(String[] args) {
13          System.out.println("Nombre: Juan Perez\nEdad: 30 anios\nDireccion: \"Calle Falsa 123\"");
14      }
15  }
```

Below the editor, the 'Output' window shows the execution results for 'TP1IntroJava (run) #2':

```
run:
Nombre: Juan Perez
Edad: 30 anios
Direccion: "Calle Falsa 123"
BUILD SUCCESSFUL (total time: 0 seconds)
```

7. Diferencia entre expresiones e instrucciones

Se analizó el siguiente fragmento de código:

```
int x = 10;           // Línea 1
x = x + 5;           // Línea 2
System.out.println(x); // Línea 3
```

Análisis:

- En la **Línea 1**, 10 es una **expresión** que devuelve un valor. Toda la línea es una **instrucción** que asigna ese valor a x.

- En la **Línea 2**, $x + 5$ es una **expresión** que también genera un valor, y la instrucción completa asigna ese valor a x .
- En la **Línea 3**, se ejecuta una **instrucción** que imprime el contenido de x . No contiene expresiones aritméticas, pero sí una evaluación del valor.

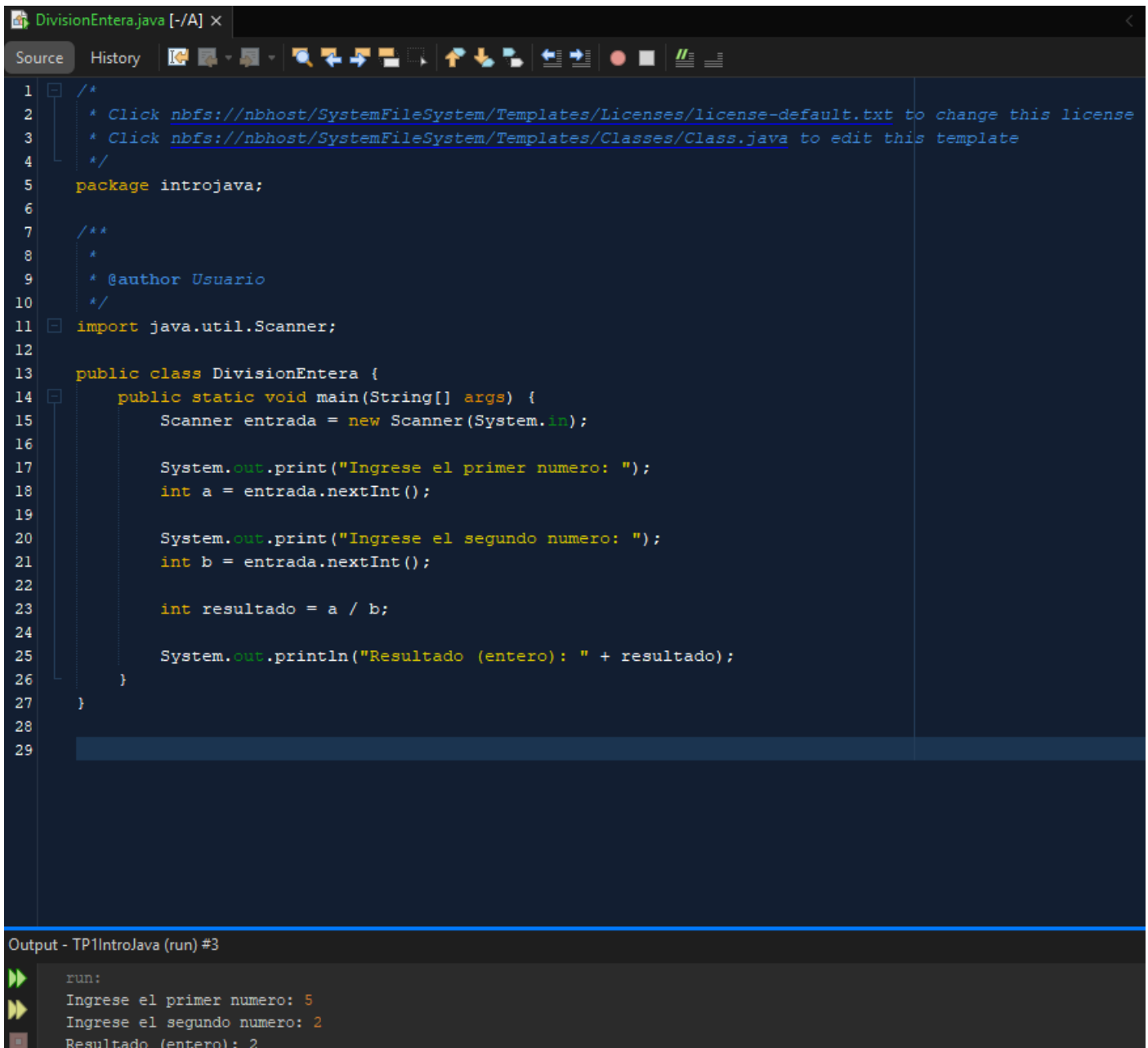
Conclusión:

- Una **expresión** es todo fragmento de código que devuelve un valor.
- Una **instrucción** es una acción completa que puede incluir expresiones.
- Toda línea que termina con `;` es una instrucción, aunque internamente puede contener expresiones.

8. Manejo de conversiones de tipo y precisión en Java

Este ejercicio permite analizar el comportamiento de Java al realizar divisiones con variables de distintos tipos (`int` y `double`).

A: División con `int`



The image shows a screenshot of an IDE with a Java file named `DivisionEntera.java`. The code is as follows:

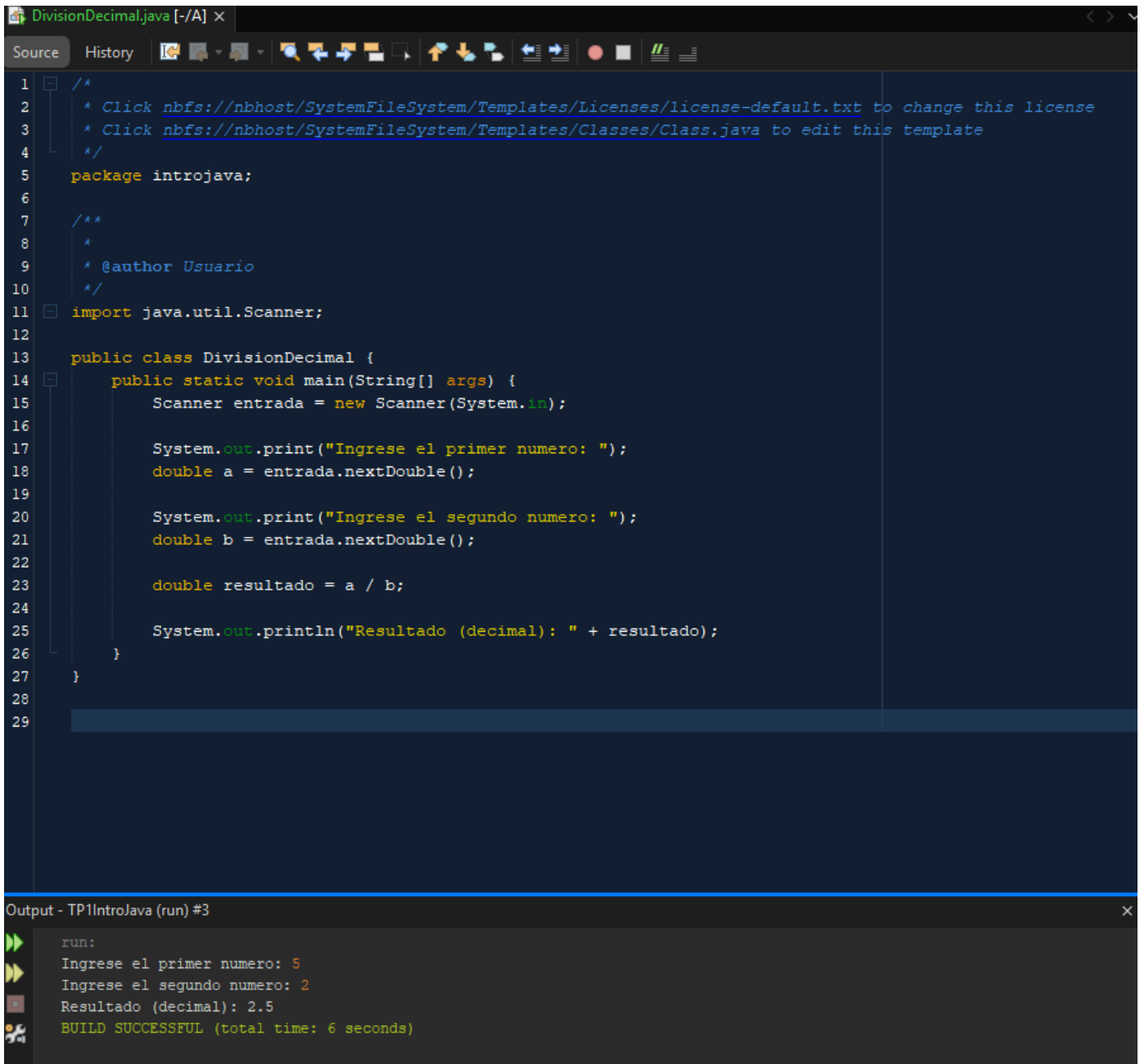
```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package introjava;
6
7  /**
8   *
9   * @author Usuario
10  */
11  import java.util.Scanner;
12
13  public class DivisionEntera {
14      public static void main(String[] args) {
15          Scanner entrada = new Scanner(System.in);
16
17          System.out.print("Ingrese el primer numero: ");
18          int a = entrada.nextInt();
19
20          System.out.print("Ingrese el segundo numero: ");
21          int b = entrada.nextInt();
22
23          int resultado = a / b;
24
25          System.out.println("Resultado (entero): " + resultado);
26      }
27  }
```

Below the code editor, the output window shows the execution results:

```
Output - TP1IntroJava (run) #3
run:
Ingrese el primer numero: 5
Ingrese el segundo numero: 2
Resultado (entero): 2
```

Resultado esperado: Si se ingresan 5 y 2, la salida será 2 porque los decimales se descartan automáticamente.

B: División con double



The screenshot shows an IDE with a file named `DivisionDecimal.java`. The code is as follows:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package introjava;
6
7  /**
8   *
9   * @author Usuario
10  */
11 import java.util.Scanner;
12
13 public class DivisionDecimal {
14     public static void main(String[] args) {
15         Scanner entrada = new Scanner(System.in);
16
17         System.out.print("Ingrese el primer numero: ");
18         double a = entrada.nextDouble();
19
20         System.out.print("Ingrese el segundo numero: ");
21         double b = entrada.nextDouble();
22
23         double resultado = a / b;
24
25         System.out.println("Resultado (decimal): " + resultado);
26     }
27 }
28
29
```

Below the code editor, the output window shows the following text:

```
Output - TP1IntroJava (run) #3
run:
Ingrese el primer numero: 5
Ingrese el segundo numero: 2
Resultado (decimal): 2.5
BUILD SUCCESSFUL (total time: 6 seconds)
```

Resultado esperado: Si se ingresan 5 y 2, la salida será 2.5, ya que ahora se conservan los decimales.

9. Corrección de errores de tipo

Se partió del siguiente código con error:

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

```
public class ErrorEjemplo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingresa tu nombre: ");
```

```

String nombre = scanner.nextInt(); // ERROR
System.out.println("Hola, " + nombre);
}
}

```

Explicación del error:

El método `nextInt()` sirve para leer **números enteros**, pero en este caso se quería capturar un **nombre**, que es una cadena de texto. Usar `nextInt()` provocaba un error cuando el usuario ingresaba texto.

Solución:

Se reemplazó `nextInt()` por `nextLine()`, que es el método correcto para leer cadenas de texto:

The screenshot shows an IDE window titled 'ErrorEjemplo.java [-/A] x'. The code editor displays the following Java code:

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package introjava;
6
7  /**
8   *
9   * @author Usuario
10  */
11  import java.util.Scanner;
12
13  public class ErrorEjemplo {
14      public static void main(String[] args) {
15          Scanner scanner = new Scanner(System.in);
16          System.out.print("Ingresa tu nombre: ");
17          String nombre = scanner.nextLine(); // CORREGIDO
18          System.out.println("Hola, " + nombre);
19      }
20  }
21

```

Below the code editor, the 'Output - TP1IntroJava (run) #3' window shows the execution results:

```

run:
Ingresa tu nombre: Nazareno
Hola, Nazareno
BUILD SUCCESSFUL (total time: 4 seconds)

```

10. Prueba de escritorio

Se realizó una prueba de escritorio sobre el siguiente código:

```

public class PruebaEscritorio {
    public static void main(String[] args) {
        int a = 5;
        int b = 2;
    }
}

```

```

    int resultado = a / b;
    System.out.println("Resultado: " + resultado);
}
}

```

Análisis línea por línea:

Línea de código	Explicación	Valor de variables
int a = 5;	Se declara a con valor 5	a = 5
int b = 2;	Se declara b con valor 2	b = 2
int resultado = a / b;	División entera: 5 / 2 → trunca → 2	resultado = 2
System.out.println("Resultado: " + resultado);	Se imprime el resultado por consola	Se muestra Resultado: 2

Se simula la ejecución línea por línea:

1. a toma el valor 5.
2. b toma el valor 2.
3. La operación a / b es una **división entre enteros**, por lo tanto el resultado se **trunca**, y no incluye decimales. 5 / 2 = 2.
4. Se imprime en pantalla: Resultado: 2.

Cuando se trabaja con enteros (int), Java realiza divisiones truncadas. Para obtener resultados con decimales, al menos uno de los operandos debe ser de tipo double.