

PROGRAMACIÓN II

Trabajo Práctico 1: Introducción a Java

NOGUEIRA DAMIÁN IGNACIO

OBJETIVO GENERAL

Aplicar los conocimientos adquiridos sobre la instalación y configuración del entorno de desarrollo, manipulación de datos, operadores matemáticos y depuración de código en Java, mediante ejercicios prácticos introductorios.

MARCO TEÓRICO

Concepto	Aplicación en el proyecto
Instalación y entorno	Almacenan el conjunto de países
Variables y tipos de datos	Representan los datos de cada país (nombre, población, superficie, etc.)
Entrada y salida	Separan las operaciones: carga, búsqueda, estadísticas, ordenamientos
Operadores aritméticos	Aplican filtros y validaciones según criterios
Caracteres especiales	Permite ordenar países por población, nombre, superficie, etc.
Expresiones e instrucciones	Permiten obtener indicadores clave del dataset
Tipos de datos y conversiones	Lectura del dataset desde un archivo CSV
Debugging y errores comunes	Identificación y corrección de errores de compilación.

Pruebas de escritorio

Análisis paso a paso de ejecución de código.

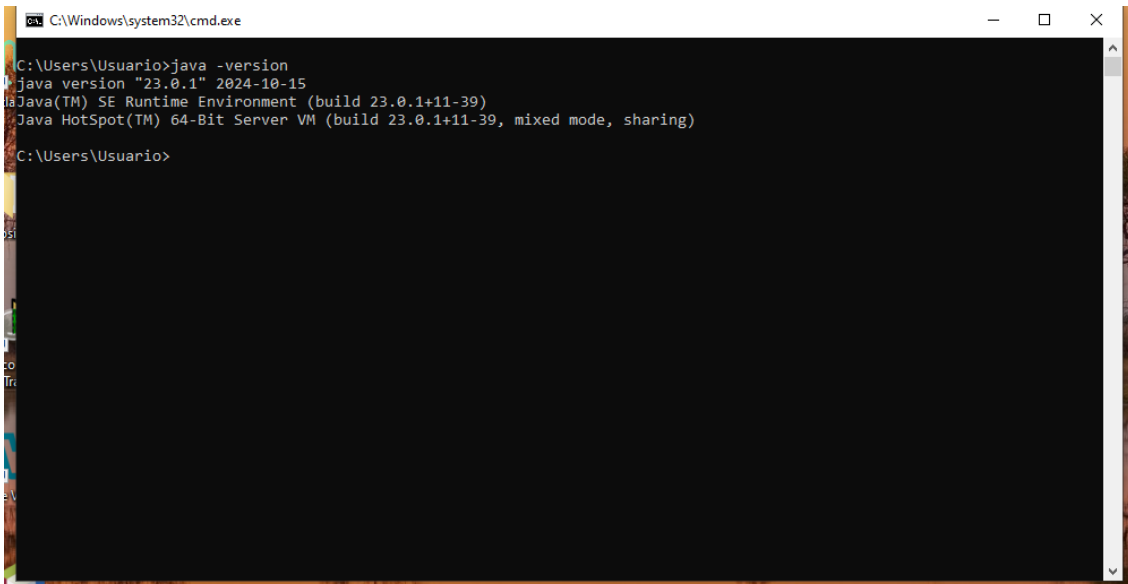
CASO PRÁCTICO

El trabajo consiste en resolver una serie de ejercicios introductorios en Java que permitan:

- Configurar correctamente el entorno de desarrollo (Java JDK y NetBeans).
- Crear programas básicos que imprimen mensajes en consola.
- Declarar variables de distintos tipos y manipular sus valores. • Leer datos ingresados por el usuario usando **Scanner**.
- Realizar operaciones aritméticas básicas.
- Aplicar caracteres de escape para dar formato a la salida.
- Analizar diferencias entre expresiones e instrucciones.
- Detectar y corregir errores simples en el código.
- Comprender el comportamiento del lenguaje mediante pruebas de escritorio.

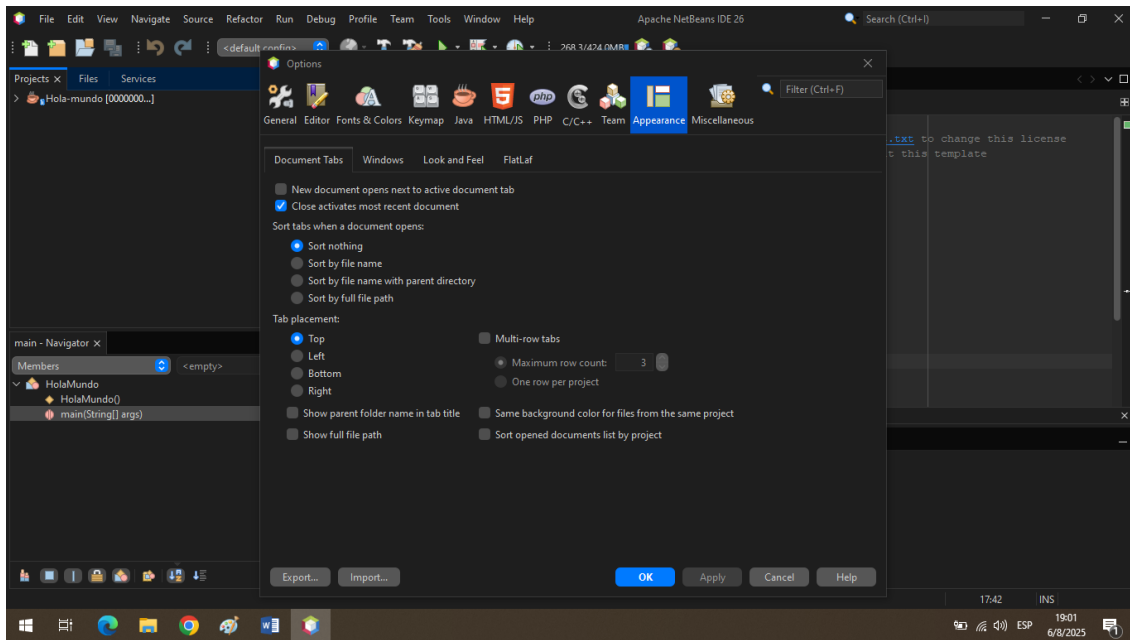
1. Verificar que tienes instalado Java JDK y NetBeans

- a. Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal: **java -version**



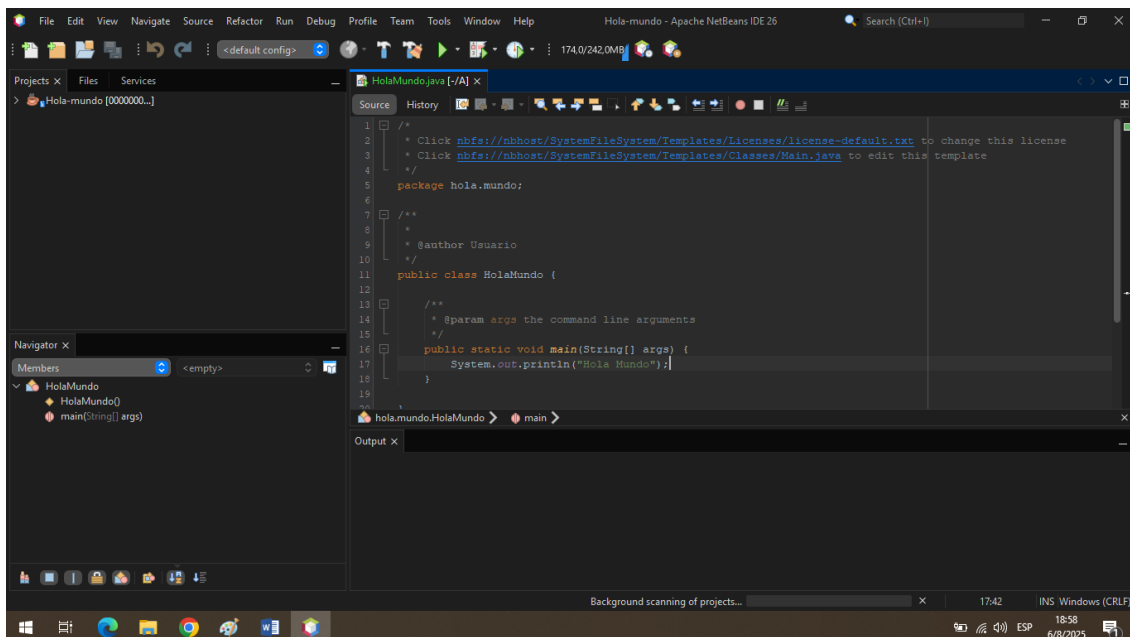
```
C:\Windows\system32\cmd.exe
C:\Users\Usuario>java -version
java version "23.0.1" 2024-10-15
Java(TM) SE Runtime Environment (build 23.0.1+11-39)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.1+11-39, mixed mode, sharing)
C:\Users\Usuario>
```

b. Abre NetBeans, crea un nuevo proyecto y configura el modo oscuro.

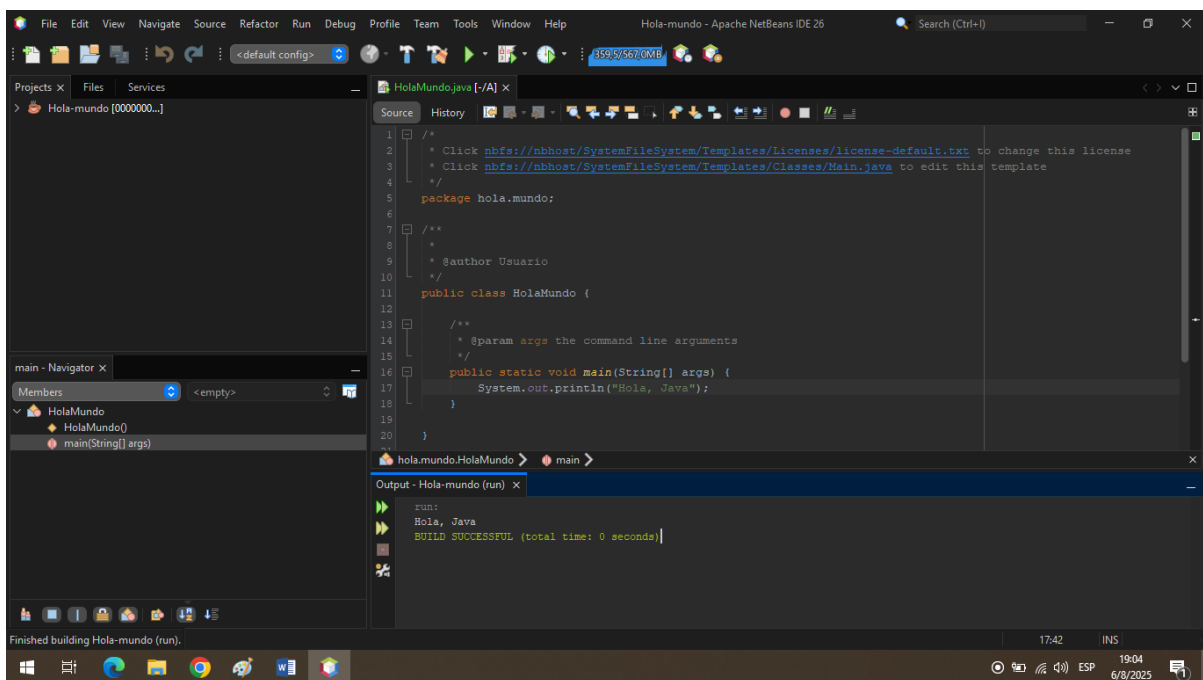


c. Toma una captura de pantalla del entorno configurado y agrégala a tu entrega. (captura anterior)

2. Escribir y ejecutar un programa básico en Java.
 - a. Creá una clase llamada **HolaMundo**.



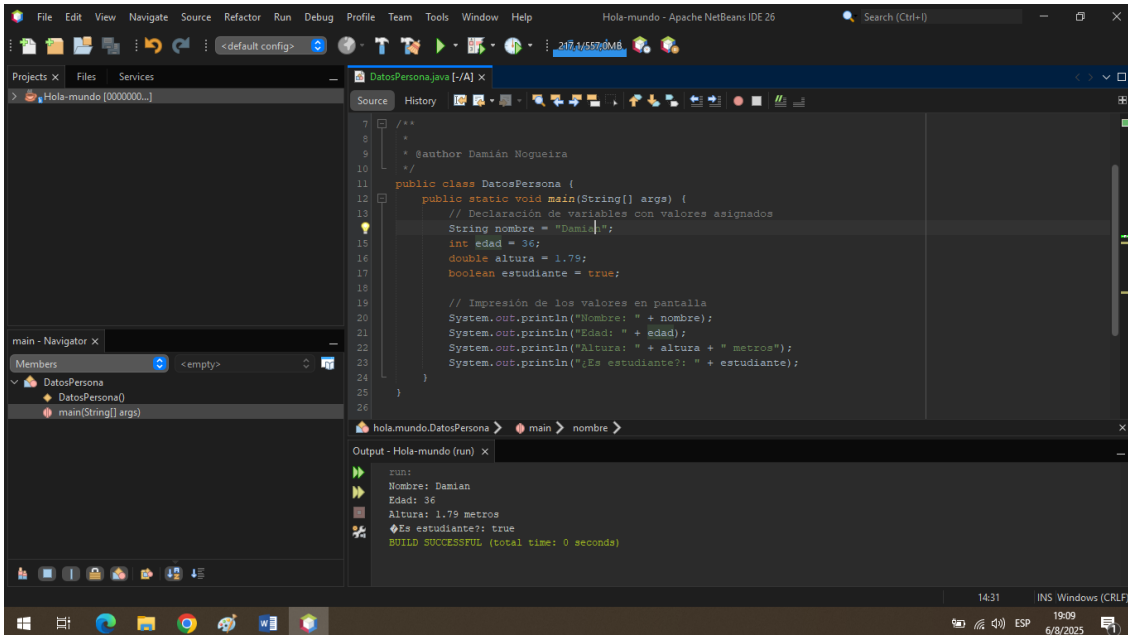
- b. Escribe un programa que imprima el mensaje: **¡Hola, Java!**



- c. Ejecuta el programa en NetBeans y adjunta una captura del resultado en la consola. (Captura anterior)

3. Crea un programa que declare las siguientes variables con valores asignados:
- String nombre
 - int edad
 - double altura
 - boolean estudiante

Imprime los valores en pantalla usando `System.out.println()`.

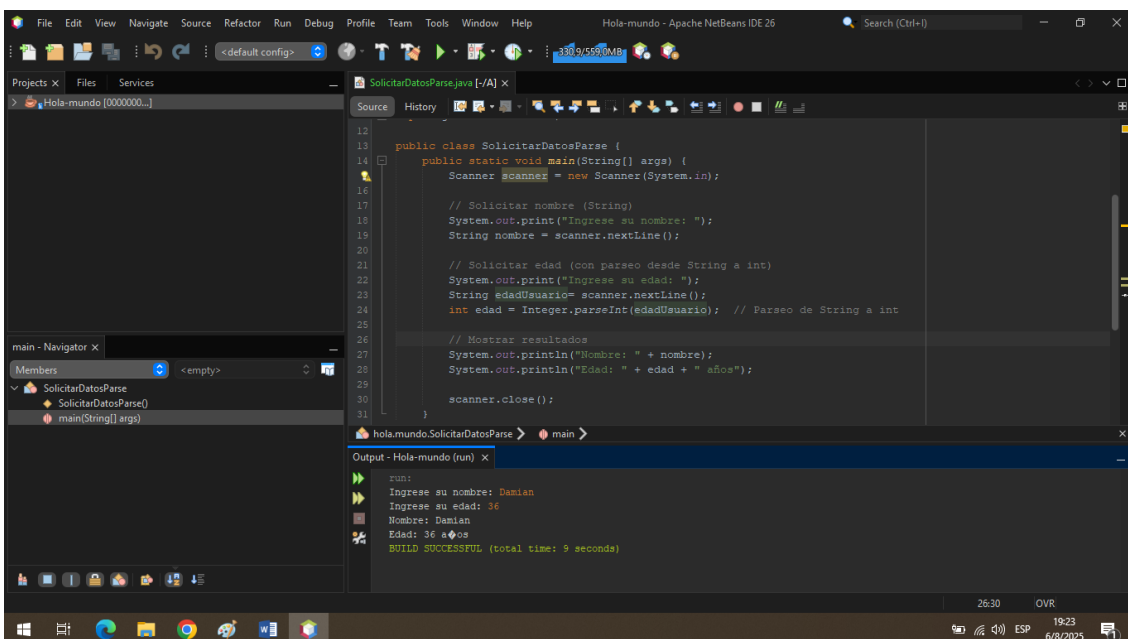


```
7  /**
8   *
9   * @author Damián Nogueira
10  */
11  public class DatosPersona {
12      public static void main(String[] args) {
13          // Declaración de variables con valores asignados
14          String nombre = "Damián";
15          int edad = 36;
16          double altura = 1.79;
17          boolean estudiante = true;
18
19          // Impresión de los valores en pantalla
20          System.out.println("Nombre: " + nombre);
21          System.out.println("Edad: " + edad);
22          System.out.println("Altura: " + altura + " metros");
23          System.out.println("¿Es estudiante?: " + estudiante);
24      }
25  }
26  }
```

Output - Hola-mundo (run) x

```
run:
Nombre: Damian
Edad: 36
Altura: 1.79 metros
¿Es estudiante?: true
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Escribe un programa que solicite al usuario ingresar su nombre y edad, y luego los muestre en pantalla. Usa `Scanner` para capturar los datos.



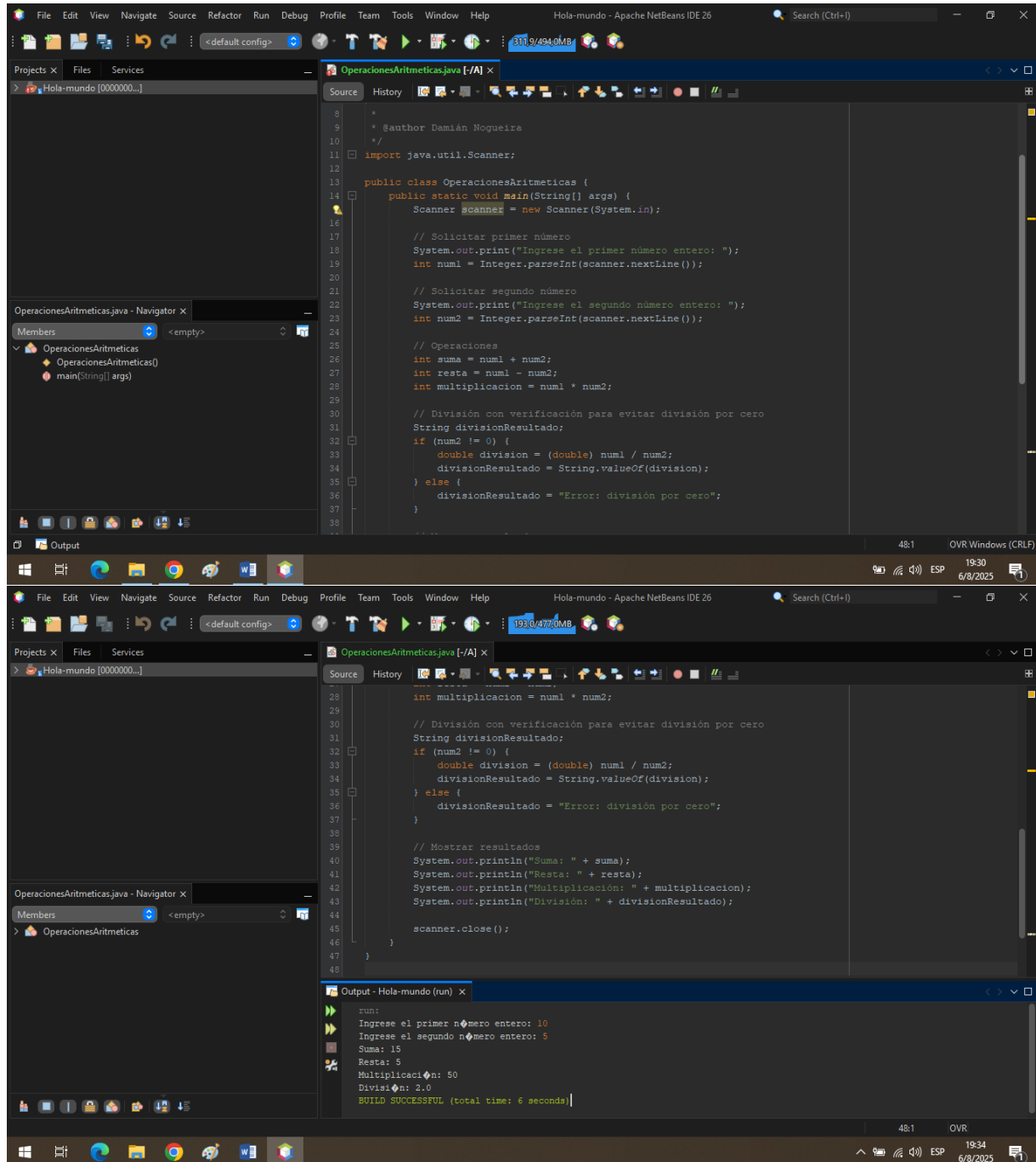
```
12  public class SolicitarDatosParse {
13      public static void main(String[] args) {
14          Scanner scanner = new Scanner(System.in);
15
16          // Solicitar nombre (String)
17          System.out.print("Ingrese su nombre: ");
18          String nombre = scanner.nextLine();
19
20          // Solicitar edad (con parseo desde String a int)
21          System.out.print("Ingrese su edad: ");
22          String edadUsuario = scanner.nextLine();
23          int edad = Integer.parseInt(edadUsuario); // Parseo de String a int
24
25          // Mostrar resultados
26          System.out.println("Nombre: " + nombre);
27          System.out.println("Edad: " + edad + " años");
28
29          scanner.close();
30      }
31  }
```

Output - Hola-mundo (run) x

```
run:
Ingrese su nombre: Damian
Ingrese su edad: 36
Nombre: Damian
Edad: 36 años
BUILD SUCCESSFUL (total time: 9 seconds)
```

5. Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:
- Suma
 - Resta
 - Multiplicación
 - División

Muestra los resultados en la consola.



The screenshot displays the Apache NetBeans IDE interface. The top window shows the source code for `OperacionesAritmeticas.java`. The code imports `java.util.Scanner` and defines a `main` method that prompts the user for two integers, `num1` and `num2`. It then performs arithmetic operations: addition, subtraction, multiplication, and division (with a check for division by zero). The results are printed to the console. The bottom window shows the output of the program, where the user has entered 10 and 5, resulting in Suma: 15, Resta: 5, Multiplicación: 50, and División: 2.0. The IDE status bar indicates a successful build.

```
8
9
10
11 import java.util.Scanner;
12
13 public class OperacionesAritmeticas {
14     public static void main(String[] args) {
15         Scanner scanner = new Scanner(System.in);
16
17         // Solicitar primer número
18         System.out.print("Ingrese el primer número entero: ");
19         int num1 = Integer.parseInt(scanner.nextLine());
20
21         // Solicitar segundo número
22         System.out.print("Ingrese el segundo número entero: ");
23         int num2 = Integer.parseInt(scanner.nextLine());
24
25         // Operaciones
26         int suma = num1 + num2;
27         int resta = num1 - num2;
28         int multiplicacion = num1 * num2;
29
30         // División con verificación para evitar división por cero
31         String divisionResultado;
32         if (num2 != 0) {
33             double division = (double) num1 / num2;
34             divisionResultado = String.valueOf(division);
35         } else {
36             divisionResultado = "Error: división por cero";
37         }
38
39         // Mostrar resultados
40         System.out.println("Suma: " + suma);
41         System.out.println("Resta: " + resta);
42         System.out.println("Multiplicación: " + multiplicacion);
43         System.out.println("División: " + divisionResultado);
44
45         scanner.close();
46     }
47 }
48
```

Output - Hola-mundo (run)

```
sum:
Ingrese el primer número entero: 10
Ingrese el segundo número entero: 5
Suma: 15
Resta: 5
Multiplicación: 50
División: 2.0
BUILD SUCCESSFUL (total time: 6 seconds)
```

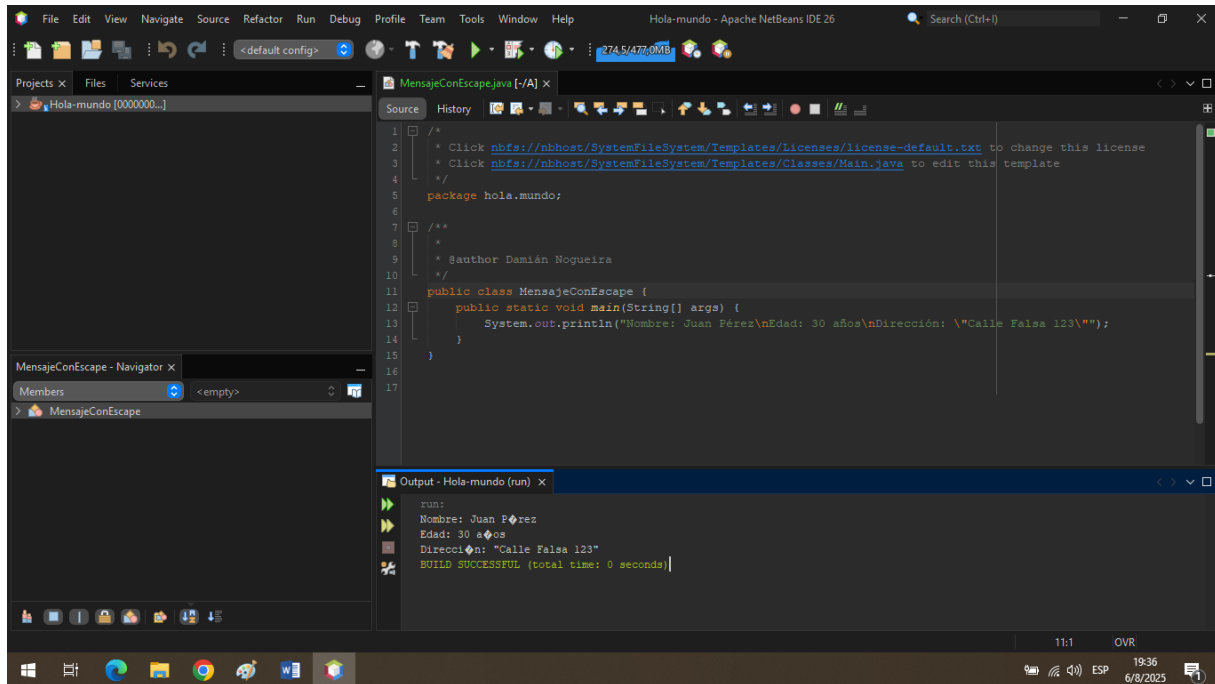
6. Escribe un programa que muestre el siguiente mensaje en consola:

Nombre: Juan Pérez

Edad: 30 años

Dirección: "Calle Falsa 123"

Usa caracteres de escape (\n, \") en `System.out.println()`.



The screenshot shows the NetBeans IDE interface. The main editor window displays a Java file named `MensajeConEscape.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/Main.java to edit this template
4  */
5  package hola.mundo;
6
7  /**
8   *
9   * @author Damián Nogueira
10  */
11  public class MensajeConEscape {
12      public static void main(String[] args) {
13          System.out.println("Nombre: Juan Pérez\nEdad: 30 años\nDirección: \"Calle Falsa 123\"");
14      }
15  }
```

The bottom-right pane shows the output of the program:

```
run:
Nombre: Juan Pérez
Edad: 30 años
Dirección: "Calle Falsa 123"
BUILD SUCCESSFUL (total time: 0 seconds)
```

7. Analiza el siguiente código y responde: ¿Cuáles son expresiones y cuáles son instrucciones? Explica la diferencia en un breve párrafo.

`int x = 10; // Línea 1`

Es una instrucción
ya que le estamos
asignando un valor
entero a una variable.

`x = x + 5; // Línea 2`

Es una expresión
dentro de una
instrucción ya que

$x+5$ es una expresión

que se evalúa, y

luego esa expresión

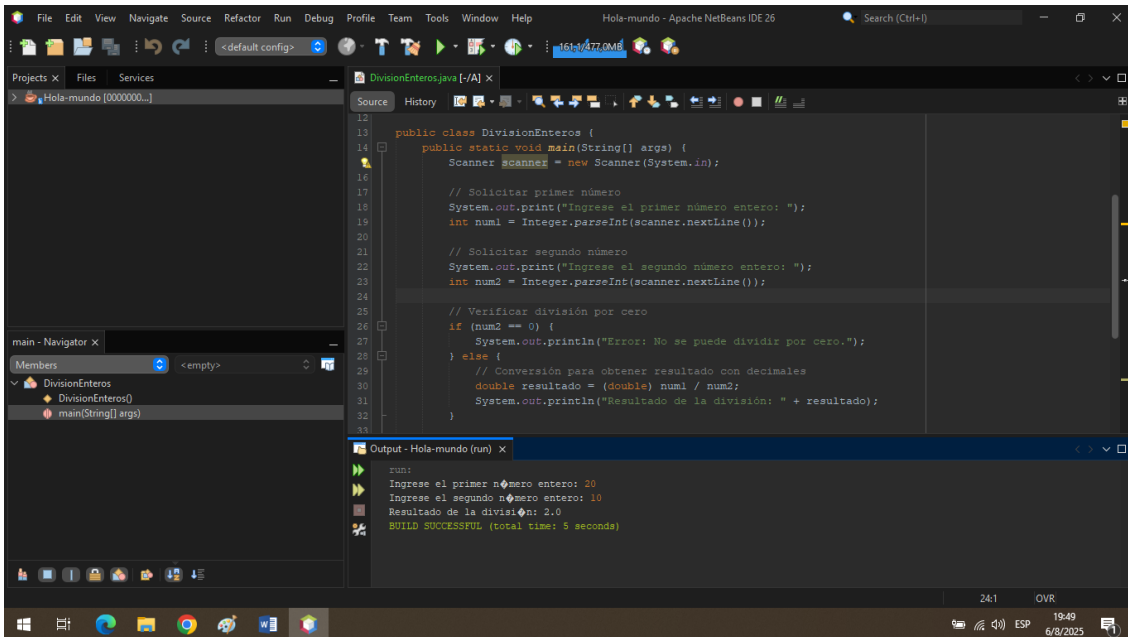
es asignada a una

variable.

`System.out.println(x);` // Línea 3

Es una instrucción ya que llama a un método

8. Manejar conversiones de tipo y división en Java.
 - a. Escribe un programa que divida dos números enteros ingresados por el usuario.

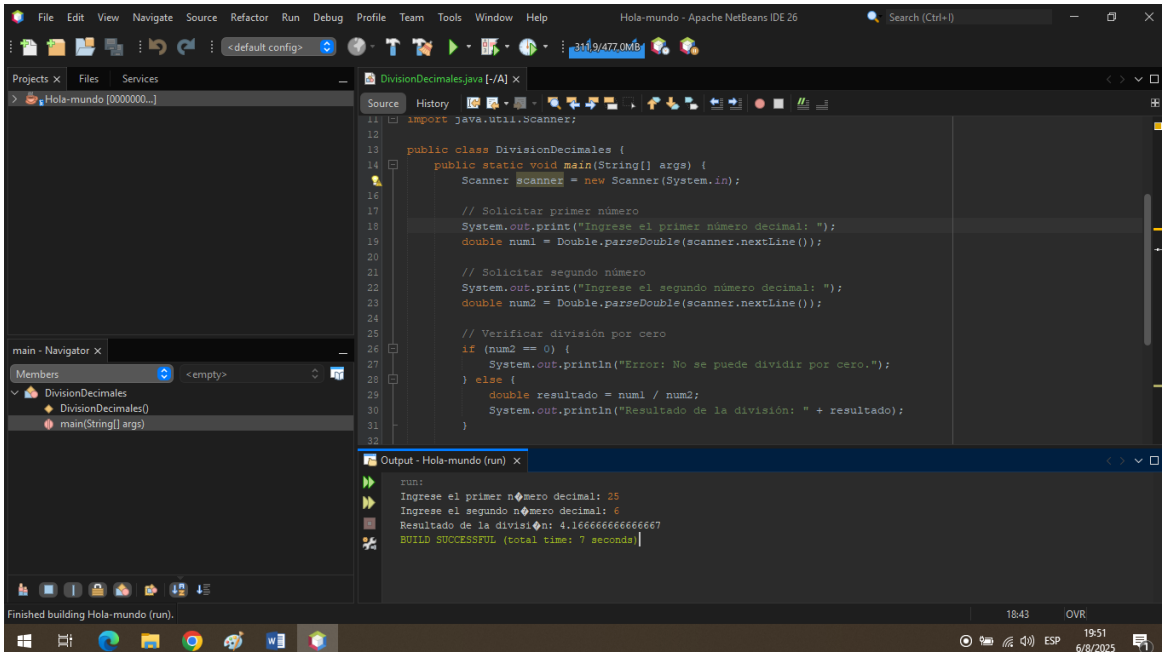


```
12
13 public class DivisionEnteros {
14     public static void main(String[] args) {
15         Scanner scanner = new Scanner(System.in);
16
17         // Solicitar primer número
18         System.out.print("Ingrese el primer número entero: ");
19         int num1 = Integer.parseInt(scanner.nextLine());
20
21         // Solicitar segundo número
22         System.out.print("Ingrese el segundo número entero: ");
23         int num2 = Integer.parseInt(scanner.nextLine());
24
25         // Verificar división por cero
26         if (num2 == 0) {
27             System.out.println("Error: No se puede dividir por cero.");
28         } else {
29             // Conversión para obtener resultado con decimales
30             double resultado = (double) num1 / num2;
31             System.out.println("Resultado de la división: " + resultado);
32         }
33     }
34 }
```

Output - Hola-mundo (run) x

```
main
  Ingrese el primer número entero: 20
  Ingrese el segundo número entero: 10
  Resultado de la división: 2.0
BUILD SUCCESSFUL (total time: 5 seconds)
```


- b. Modifica el código para usar **double** en lugar de **int** y compara los resultados.



The screenshot shows the Apache NetBeans IDE interface. The main editor displays the `DivisionDecimales.java` file. The code imports `java.util.Scanner` and defines a `main` method that prompts the user for two decimal numbers, parses them as `double`, and performs a division check. The output window shows the execution results, including the input values and the calculated result.

```
import java.util.Scanner;

public class DivisionDecimales {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Solicitar primer número
        System.out.print("Ingrese el primer número decimal: ");
        double num1 = Double.parseDouble(scanner.nextLine());

        // Solicitar segundo número
        System.out.print("Ingrese el segundo número decimal: ");
        double num2 = Double.parseDouble(scanner.nextLine());

        // Verificar división por cero
        if (num2 == 0) {
            System.out.println("Error: No se puede dividir por cero.");
        } else {
            double resultado = num1 / num2;
            System.out.println("Resultado de la división: " + resultado);
        }
    }
}
```

Output - Hola-mundo (run) X

```
Run:
Ingrese el primer número decimal: 25
Ingrese el segundo número decimal: 6
Resultado de la división: 4.166666666666667
BUILD SUCCESSFUL (total time: 7 seconds)
```

9. Corrige el siguiente código para que funcione correctamente. Explica qué error tenía y cómo lo solucionaste.

```
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);
}
}
```

`nextInt()` se usa para leer un número entero (`int`), pero acá estamos tratando de guardarlo en una variable `String`, lo cual no es compatible.
La solución es usar `scanner.nextLine()` para leer texto como `String`.

10. Completa la tabla de prueba de escritorio para el siguiente código. ¿Cuál es el valor de **resultado** y por qué?

```
public class PruebaEscritorio { public static void main(String[] args) { int  
a = 5; int b = 2; int resultado = a / b;  
System.out.println("Resultado: " + resultado);  
}  
}
```

LINEA	a	b	resultado
1	SIN DEFINIR	SIN DEFINIR	SIN DEFINIR
2	5	SIN DEFINIR	SIN DEFINIR
3	5	2	2
4	5	2	2
5	5	2	2

CONCLUSIONES ESPERADAS

- Reforzar los conceptos fundamentales del lenguaje Java.
- Familiarizarse con la estructura básica de un programa en Java.
- Aprender a depurar errores comunes.
- Comprender la importancia de las conversiones de tipo y expresiones.
- Adquirir habilidades prácticas para manipular entradas/salidas y variables.
- Aplicar el uso de herramientas como NetBeans y prácticas de depuración.