

# Trabajo Práctico 1: Introducción a Java

## 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	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

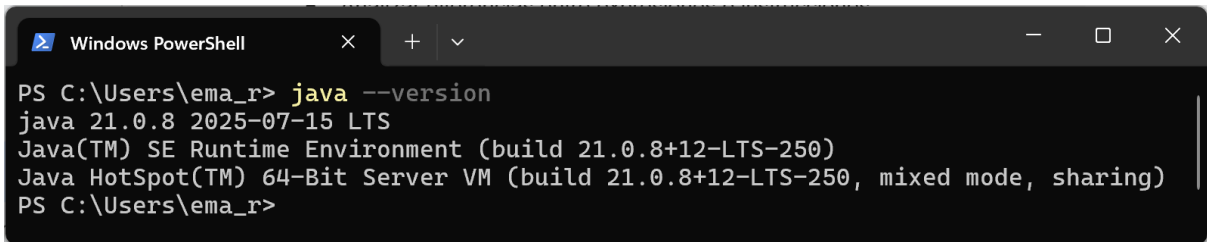
El trabajo consiste en resolver una serie de ejercicios introductorios en Java que permitan:

- 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

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

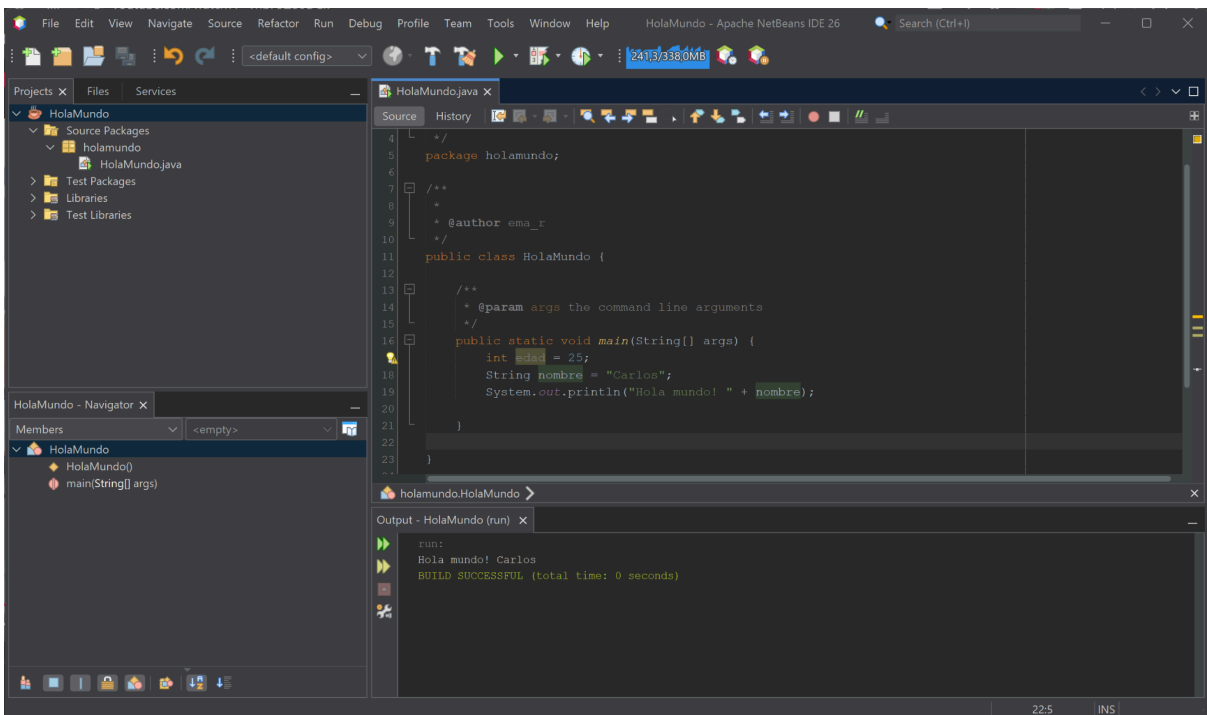


```

PS C:\Users\ema_r> java --version
java 21.0.8 2025-07-15 LTS
Java(TM) SE Runtime Environment (build 21.0.8+12-LTS-250)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.8+12-LTS-250, mixed mode, sharing)
PS C:\Users\ema_r>

```

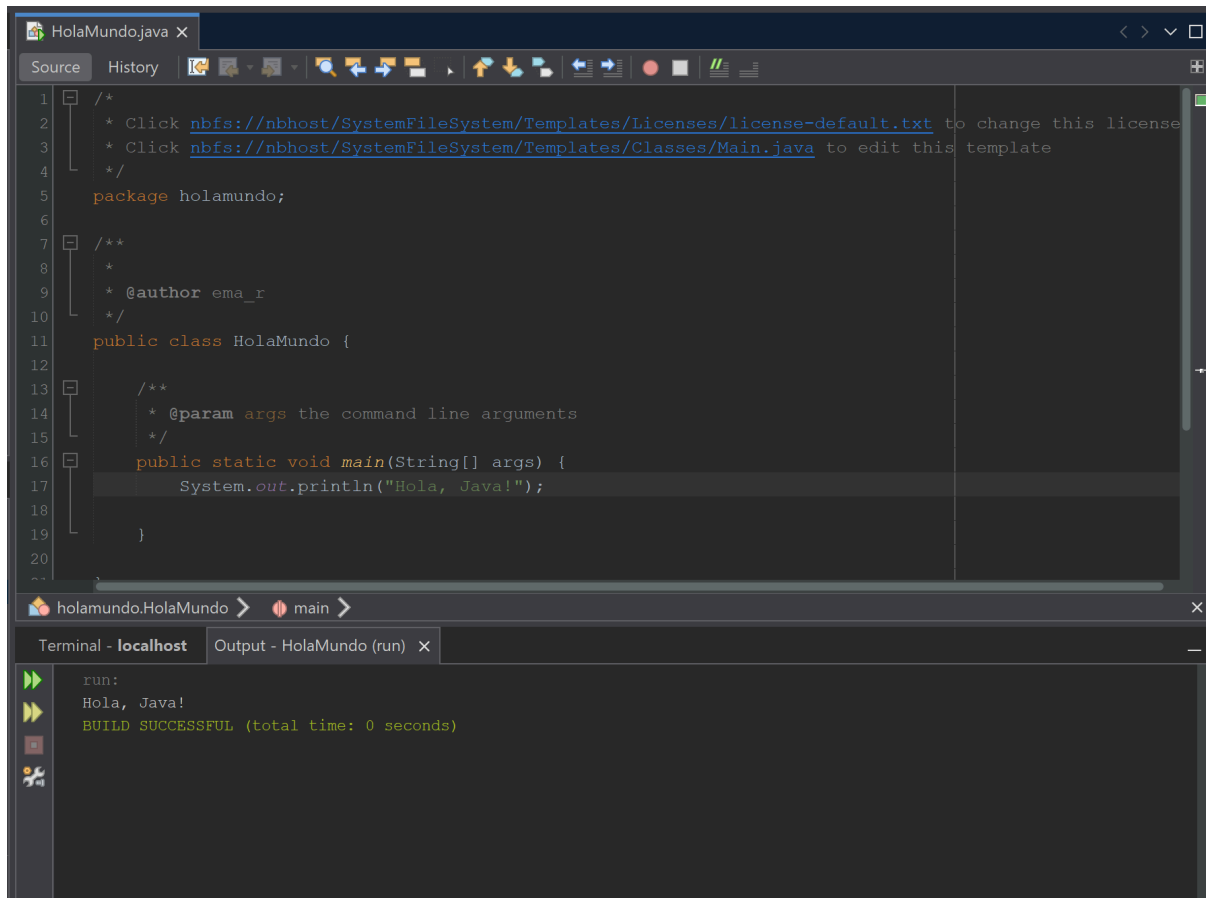
- Abre NetBeans, crea un nuevo proyecto y configura el modo oscuro.
- Toma una captura de pantalla del entorno configurado y agrégala a tu entrega.



2. Escribir y ejecutar un programa básico en Java.

- Creá una clase llamada HolaMundo.
- Escribe un programa que imprima el mensaje: ¡Hola, Java!

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



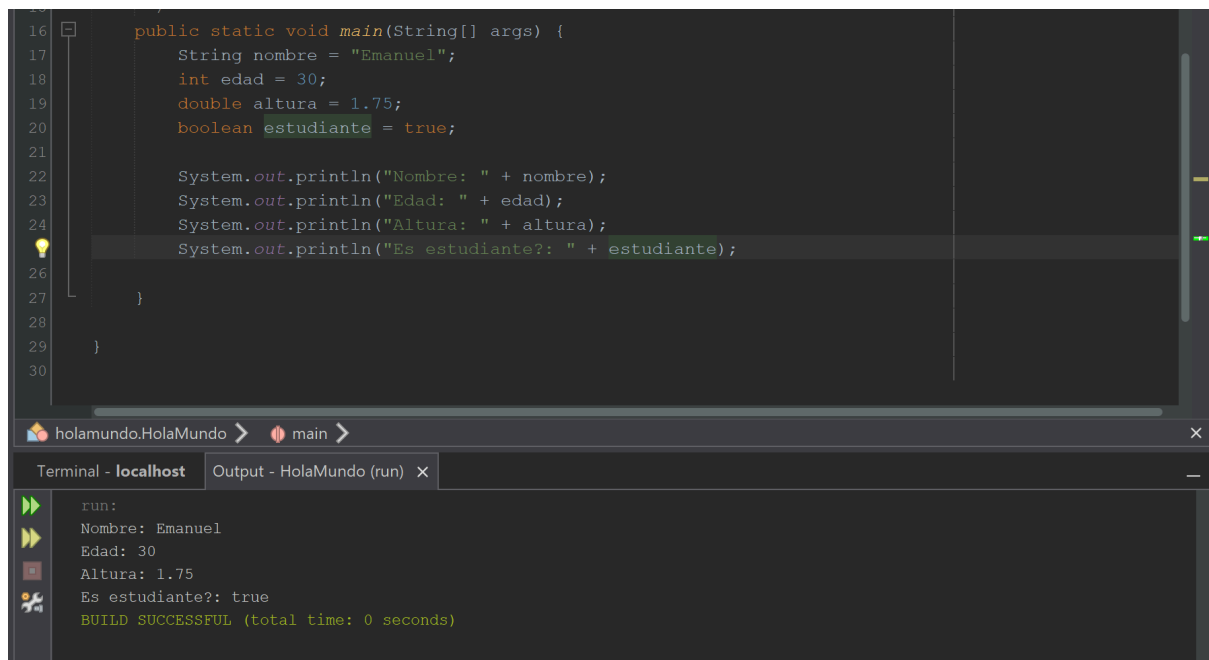
```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package holamundo;
6
7  /**
8   *
9   * @author ema_r
10  */
11  public class HolaMundo {
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  ~
```

holamundo.HolaMundo > main >

Terminal - localhost Output - HolaMundo (run) x

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

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().



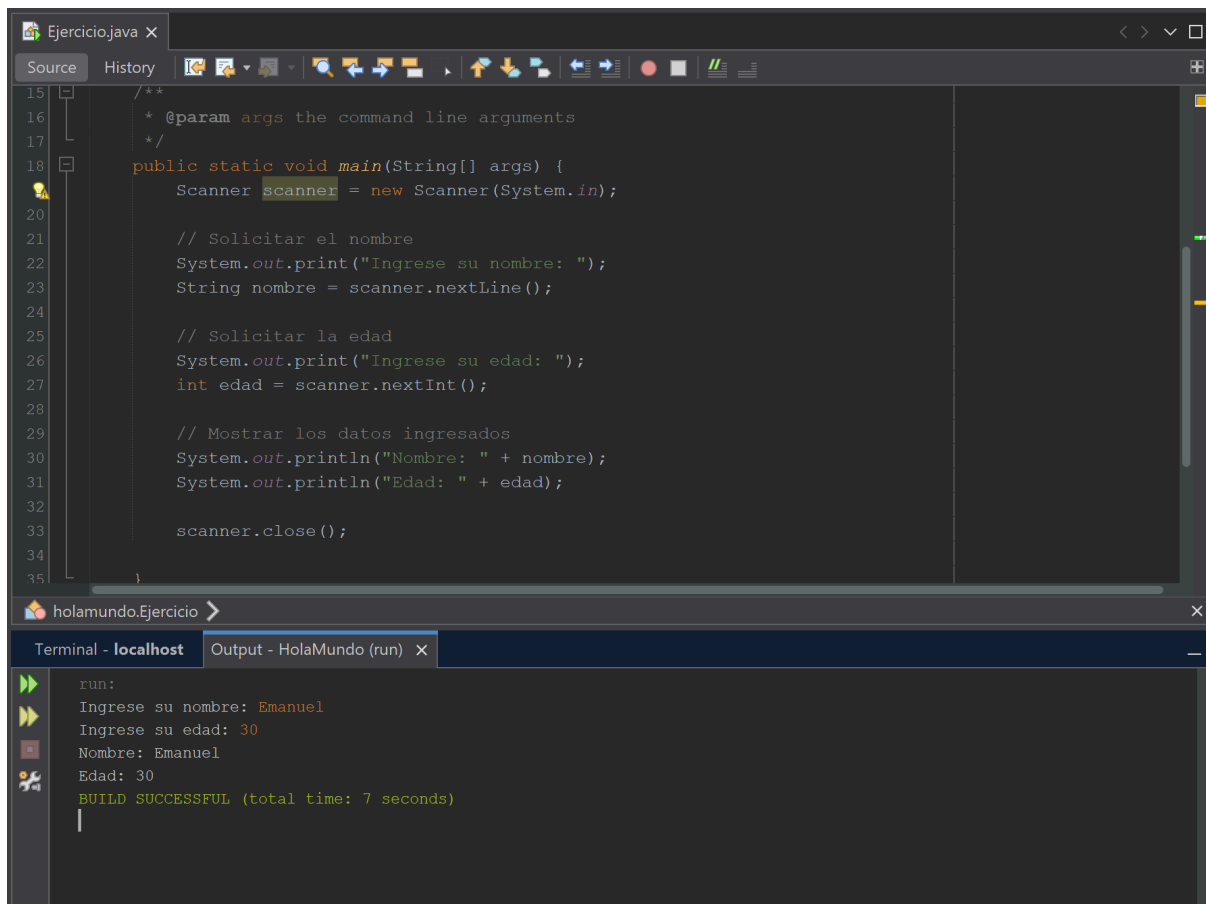
```
16  public static void main(String[] args) {
17      String nombre = "Emanuel";
18      int edad = 30;
19      double altura = 1.75;
20      boolean estudiante = true;
21
22      System.out.println("Nombre: " + nombre);
23      System.out.println("Edad: " + edad);
24      System.out.println("Altura: " + altura);
25      System.out.println("Es estudiante?: " + estudiante);
26  }
27
28
29  }
```

holamundo.HolaMundo > main >

Terminal - localhost Output - HolaMundo (run) x

```
run:
Nombre: Emanuel
Edad: 30
Altura: 1.75
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.



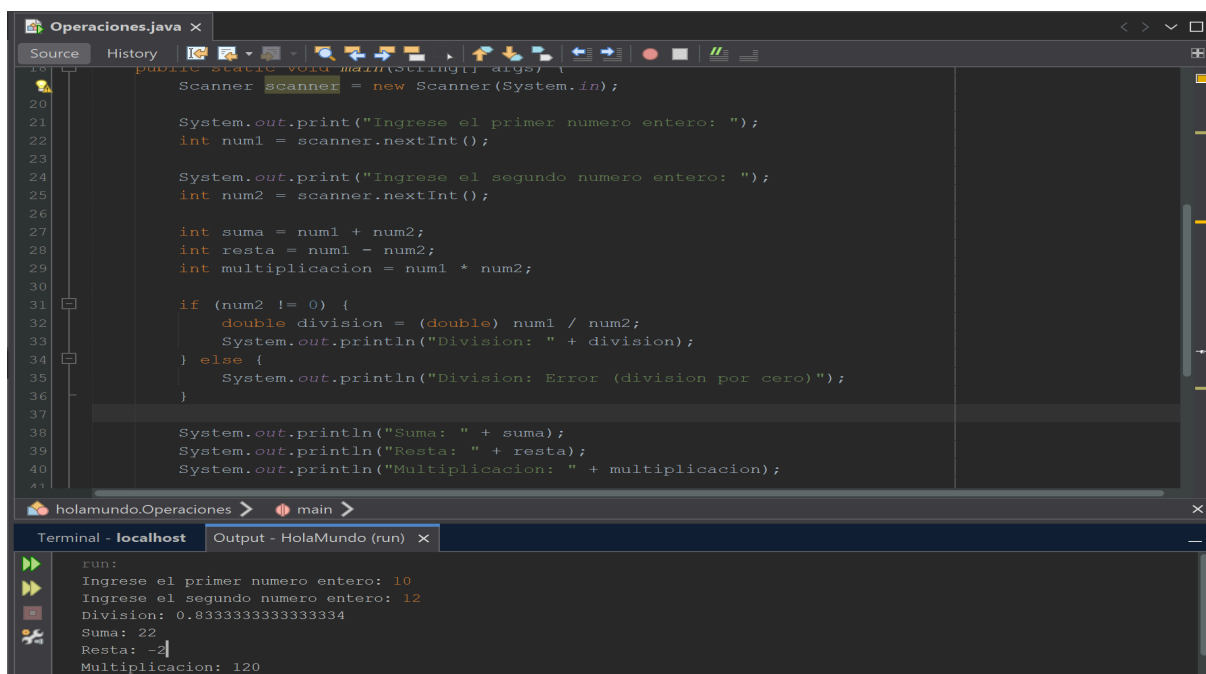
The screenshot shows an IDE with a file named `Ejercicio.java`. The code is as follows:

```
15  /**
16   * @param args the command line arguments
17   */
18  public static void main(String[] args) {
19      Scanner scanner = new Scanner(System.in);
20
21      // Solicitar el nombre
22      System.out.print("Ingrese su nombre: ");
23      String nombre = scanner.nextLine();
24
25      // Solicitar la edad
26      System.out.print("Ingrese su edad: ");
27      int edad = scanner.nextInt();
28
29      // Mostrar los datos ingresados
30      System.out.println("Nombre: " + nombre);
31      System.out.println("Edad: " + edad);
32
33      scanner.close();
34  }
```

Below the code editor, the `Output - HolaMundo (run)` window shows the following execution:

```
run:
Ingrese su nombre: Emanuel
Ingrese su edad: 30
Nombre: Emanuel
Edad: 30
BUILD SUCCESSFUL (total time: 7 seconds)
```

5. Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:
- Suma
  - Resta
  - Multipliación
  - División
- Muestra los resultados en la consola.



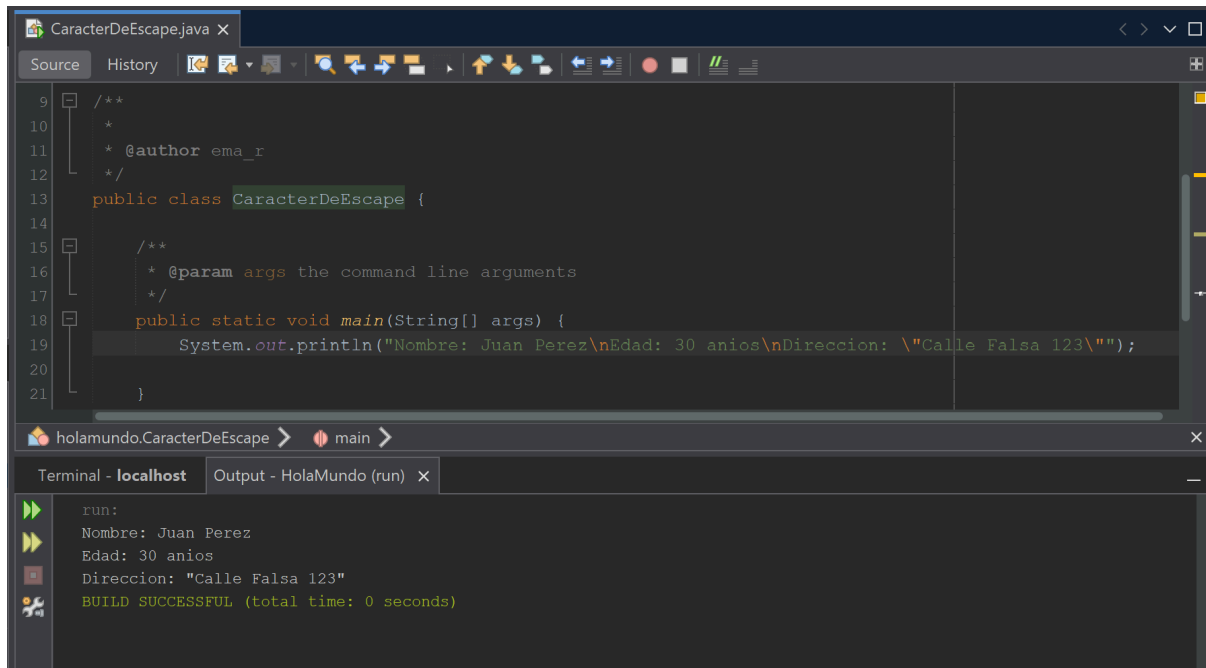
The screenshot shows an IDE with a file named `Operaciones.java`. The code is as follows:

```
18  public static void main(String[] args) {
19      Scanner scanner = new Scanner(System.in);
20
21      System.out.print("Ingrese el primer numero entero: ");
22      int num1 = scanner.nextInt();
23
24      System.out.print("Ingrese el segundo numero entero: ");
25      int num2 = scanner.nextInt();
26
27      int suma = num1 + num2;
28      int resta = num1 - num2;
29      int multiplicacion = num1 * num2;
30
31      if (num2 != 0) {
32          double division = (double) num1 / num2;
33          System.out.println("Division: " + division);
34      } else {
35          System.out.println("Division: Error (division por cero)");
36      }
37
38      System.out.println("Suma: " + suma);
39      System.out.println("Resta: " + resta);
40      System.out.println("Multiplicacion: " + multiplicacion);
41  }
```

Below the code editor, the `Output - HolaMundo (run)` window shows the following execution:

```
run:
Ingrese el primer numero entero: 10
Ingrese el segundo numero entero: 12
Division: 0.8333333333333334
Suma: 22
Resta: -2
Multiplicacion: 120
```

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 an IDE window titled 'CaracterDeEscape.java'. The code is as follows:

```
9  /**
10  *
11  * @author ema_r
12  */
13  public class CaracterDeEscape {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          System.out.println("Nombre: Juan Perez\nEdad: 30 anios\nDireccion: \"Calle Falsa 123\"");
20      }
21  }
```

Below the code editor, the 'Terminal - localhost' tab shows the output of the program:

```
run:
Nombre: Juan Perez
Edad: 30 anios
Direccion: "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
x = x + 5; // Línea 2
System.out.println(x); // Línea 3
```

Línea 1: `int x = 10;` es una instrucción. Es una declaración que crea la variable `x` y le asigna un valor inicial.

Línea 2: `x = x + 5;` es una instrucción. Es una asignación que modifica el valor de `x`.

Línea 3: `System.out.println(x);` también es una instrucción. Es la acción de mostrar el valor de `x` en la consola.

Dentro de estas instrucciones, las expresiones son:

En la línea 2: `x + 5` es una expresión que evalúa a un valor.

En la línea 3: `x` dentro de `println` también es una expresión que evalúa a su valor actual.

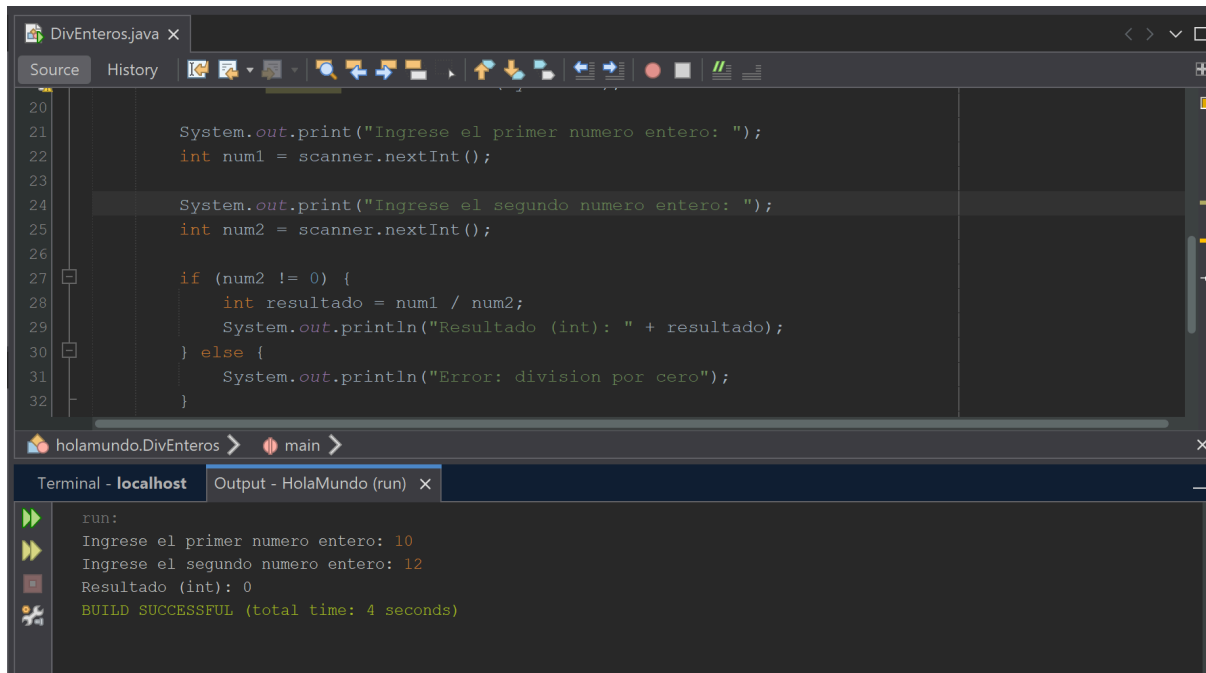
Diferencia:

Una expresión en Java es un fragmento de código que produce un valor, como `x + 5` o `x`. Una instrucción es un comando completo que realiza una acción, como declarar una variable, asignar un valor o mostrar datos en pantalla.

En resumen, las instrucciones pueden contener expresiones, pero las expresiones por sí mismas sólo evalúan a un valor.

8. Manejar conversiones de tipo y división en Java.

- a. Escribe un programa que divida dos números enteros ingresados por el usuario.



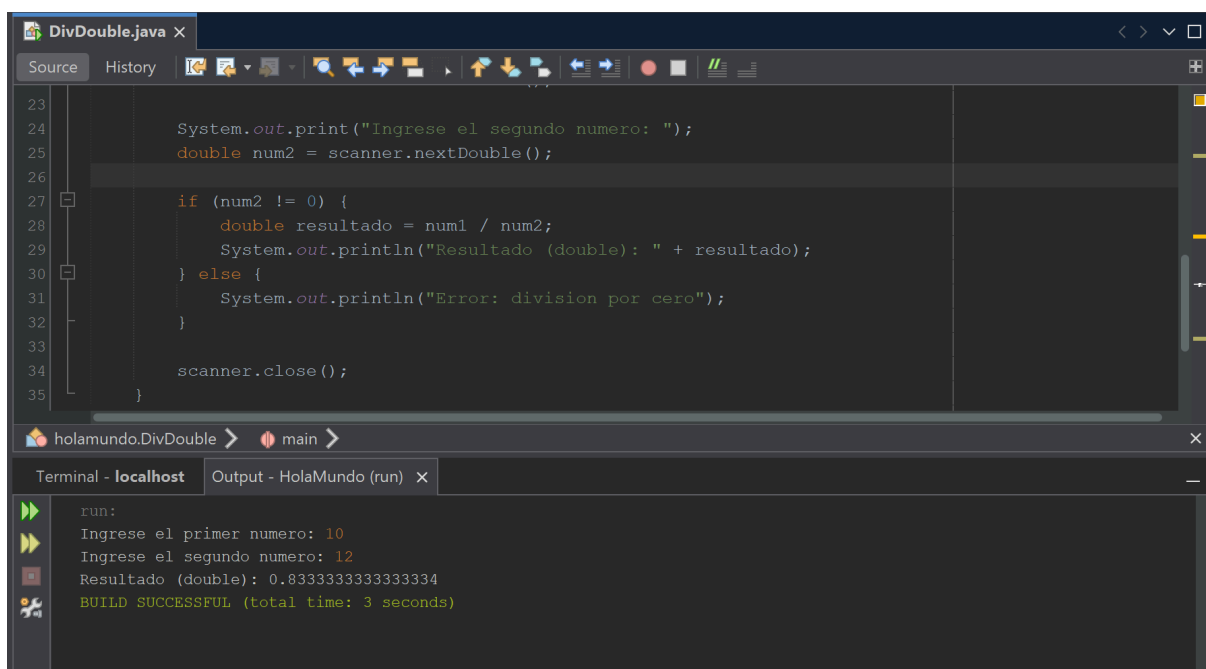
The screenshot shows an IDE window titled 'DivEnteros.java'. The code in the editor is as follows:

```
20
21     System.out.print("Ingrese el primer numero entero: ");
22     int num1 = scanner.nextInt();
23
24     System.out.print("Ingrese el segundo numero entero: ");
25     int num2 = scanner.nextInt();
26
27     if (num2 != 0) {
28         int resultado = num1 / num2;
29         System.out.println("Resultado (int): " + resultado);
30     } else {
31         System.out.println("Error: division por cero");
32     }
33 }
```

Below the editor, the 'Terminal - localhost' tab shows the output of the program:

```
run:
Ingrese el primer numero entero: 10
Ingrese el segundo numero entero: 12
Resultado (int): 0
BUILD SUCCESSFUL (total time: 4 seconds)
```

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



The screenshot shows an IDE window titled 'DivDouble.java'. The code in the editor is as follows:

```
23
24     System.out.print("Ingrese el segundo numero: ");
25     double num2 = scanner.nextDouble();
26
27     if (num2 != 0) {
28         double resultado = num1 / num2;
29         System.out.println("Resultado (double): " + resultado);
30     } else {
31         System.out.println("Error: division por cero");
32     }
33
34     scanner.close();
35 }
```

Below the editor, the 'Terminal - localhost' tab shows the output of the program:

```
run:
Ingrese el primer numero: 10
Ingrese el segundo numero: 12
Resultado (double): 0.8333333333333334
BUILD SUCCESSFUL (total time: 3 seconds)
```

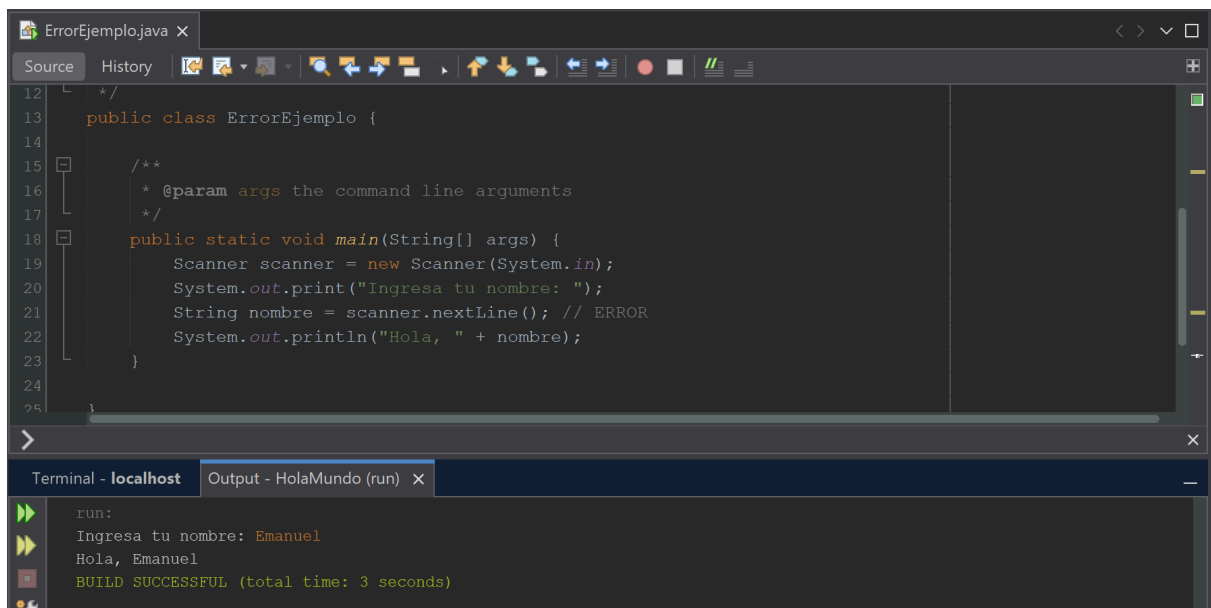
9. Corrige el siguiente código para que funcione correctamente. Explica qué error tenía y cómo lo solucionaste.

```
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); } }
```

Usar `scanner.nextInt()` para leer un texto provoca un error porque `nextInt()` intenta convertir la entrada en un número entero, no en una cadena de texto.

La solución fue usar `scanner.nextLine()` que lee toda la línea ingresada, incluyendo espacios, y devuelve un `String`.



The screenshot shows an IDE window titled "ErrorEjemplo.java". The code in the editor is as follows:

```
12  */  
13  public class ErrorEjemplo {  
14  
15      /**  
16       * @param args the command line arguments  
17       */  
18      public static void main(String[] args) {  
19          Scanner scanner = new Scanner(System.in);  
20          System.out.print("Ingresa tu nombre: ");  
21          String nombre = scanner.nextLine(); // ERROR  
22          System.out.println("Hola, " + nombre);  
23      }  
24  
25  }
```

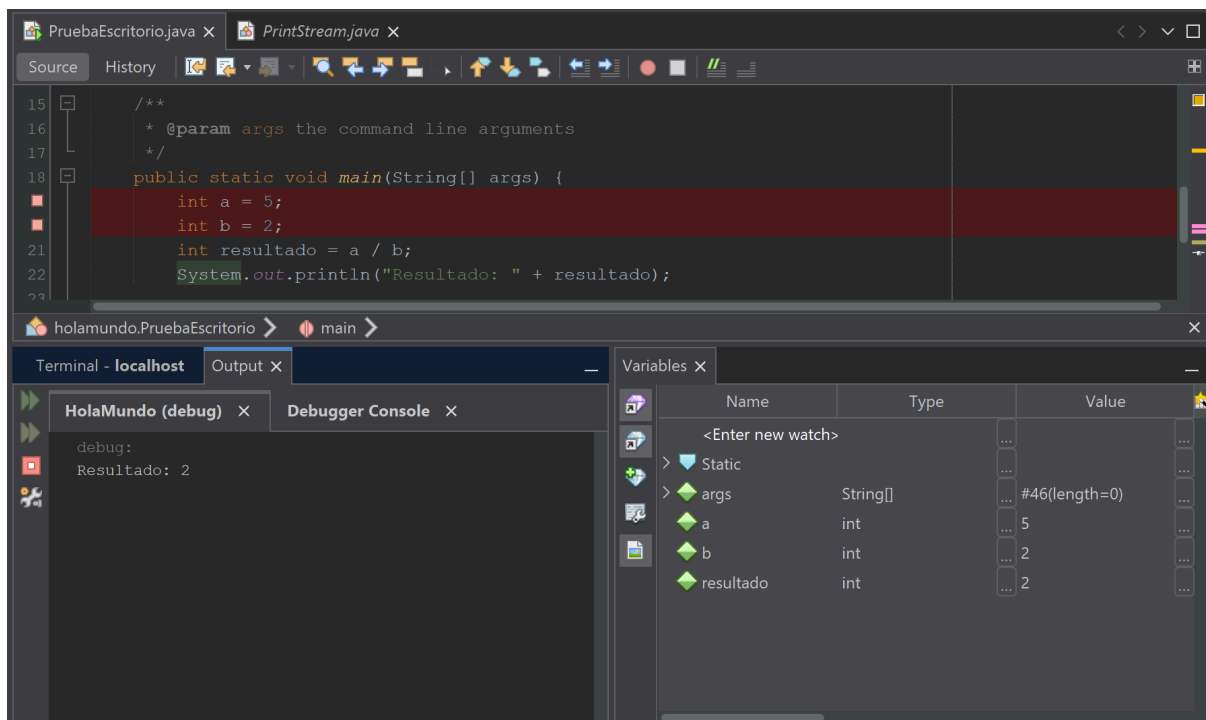
Below the code editor, the "Terminal - localhost" tab is active, showing the output of the program:

```
run:  
Ingresa tu nombre: Emanuel  
Hola, Emanuel  
BUILD SUCCESSFUL (total time: 3 seconds)
```

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);  
    }  
}
```

El valor de resultado es 2. Porque en Java, dividir dos enteros da un resultado entero, descartando la parte decimal. En este caso, 5 / 2 resulta en 2. El resultado real sería 2.5, pero en división de enteros, solo queda la parte entera.



## 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.