

Módulo 1: Práctica – TP 1: Introducción a Java

Datos Generales

- **Título del trabajo:** Introducción a las Bases de Datos
 - **Alumno:** Renzo Calcatelli - rcalcatelli@gmail.com
 - **Materia:** Programación II
 - **Profesor/a:** Ariel Enferrel
 - **Fecha de Entrega:** N/A
-

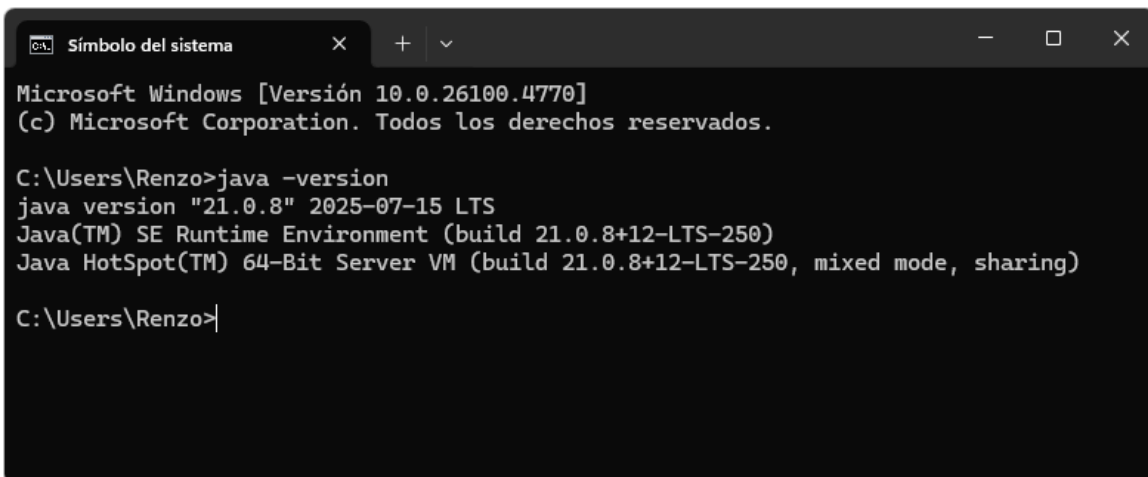
Índice

1. **Ejercicio 1: Verificación de instalación y configuración**
 - 1.1. Verificar instalación de Java JDK con comando `java -version`
 - 1.2. Configuración de NetBeans y modo oscuro
 - 1.3. Captura de pantalla del entorno configurado
2. **Ejercicio 2: Programa básico en Java: HolaMundo**
 - 2.1. Creación de la clase `HolaMundo`
 - 2.2. Código que imprime "¡Hola, Java!"
 - 2.3. Captura de pantalla del resultado en consola
3. **Ejercicio 3: Declaración de variables y mostrar valores**
 - 3.1. Declaración de variables `String`, `int`, `double` y `boolean`
 - 3.2. Impresión de valores por consola
4. **Ejercicio 4: Entrada de datos con Scanner**
 - 4.1. Solicitar nombre y edad al usuario
 - 4.2. Mostrar datos ingresados en pantalla
5. **Ejercicio 5: Operaciones aritméticas básicas**
 - 5.1. Ingreso de dos números enteros
 - 5.2. Operaciones: suma, resta, multiplicación y división
 - 5.3. Mostrar resultados y validar división por cero
6. **Ejercicio 6: Mensaje con caracteres de escape**
 - 6.1. Uso de `\n` y `\"` para formatear salida
 - 6.2. Mostrar mensaje formateado en consola
7. **Ejercicio 7: Expresiones vs instrucciones**
 - 7.1. Análisis de código dado
 - 7.2. Definición y diferencia entre expresiones e instrucciones
8. **Ejercicio 8: Conversiones de tipo y división**
 - 8.1. División con números enteros
 - 8.2. División usando `double` para resultados decimales
 - 8.3. Comparación entre ambos resultados

9. **Ejercicio 9: Corrección de código con error**
 - 9.1. Identificación del error de tipo en lectura con Scanner
 - 9.2. Corrección para lectura correcta de texto
 10. **Ejercicio 10: Prueba de escritorio**
 - 10.1. Análisis paso a paso del código proporcionado
 - 10.2. Valor final de variables y explicación del resultado
 11. **Conclusiones**
 - Aprendizajes y reflexiones sobre el TP
-

1. Verificación de instalación y configuración

Primero verifiqué que tengo instalado Java JDK en mi computadora. Para esto abrí la terminal (en Windows uso PowerShell) y escribí el siguiente comando: **java -version**. Al ejecutarlo, me apareció la versión instalada de Java, lo que confirmó que Java JDK está correctamente instalado.



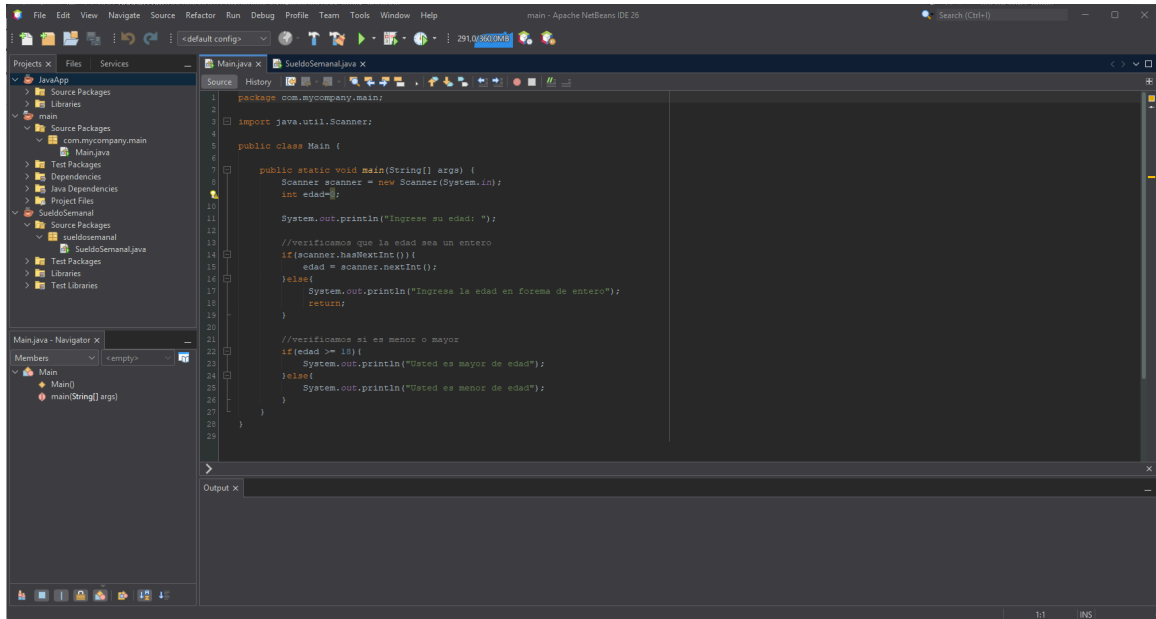
```
Microsoft Windows [Versión 10.0.26100.4770]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Renzo>java -version
java version "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)

C:\Users\Renzo>
```

Después abrí NetBeans, que es el entorno de desarrollo que vamos a usar para programar en Java. Creé un nuevo proyecto y configuré el modo oscuro porque me resulta más cómodo para trabajar.

Adjunto una captura de pantalla del entorno NetBeans configurado en modo oscuro:



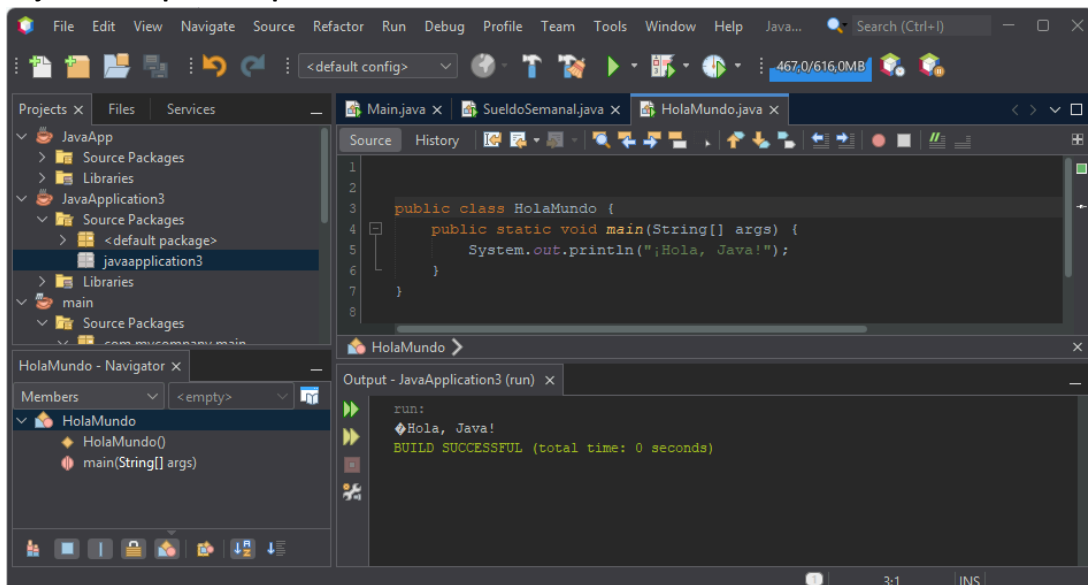
2. Programa básico en Java: HolaMundo

Creé una clase llamada **HolaMundo** que contiene un programa simple que imprime el mensaje:

```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("¡Hola, Java!");  
    }  
}
```

Ejecuté el programa en NetBeans y en la consola apareció el mensaje correctamente.

Adjunto la captura de pantalla del resultado en la consola:

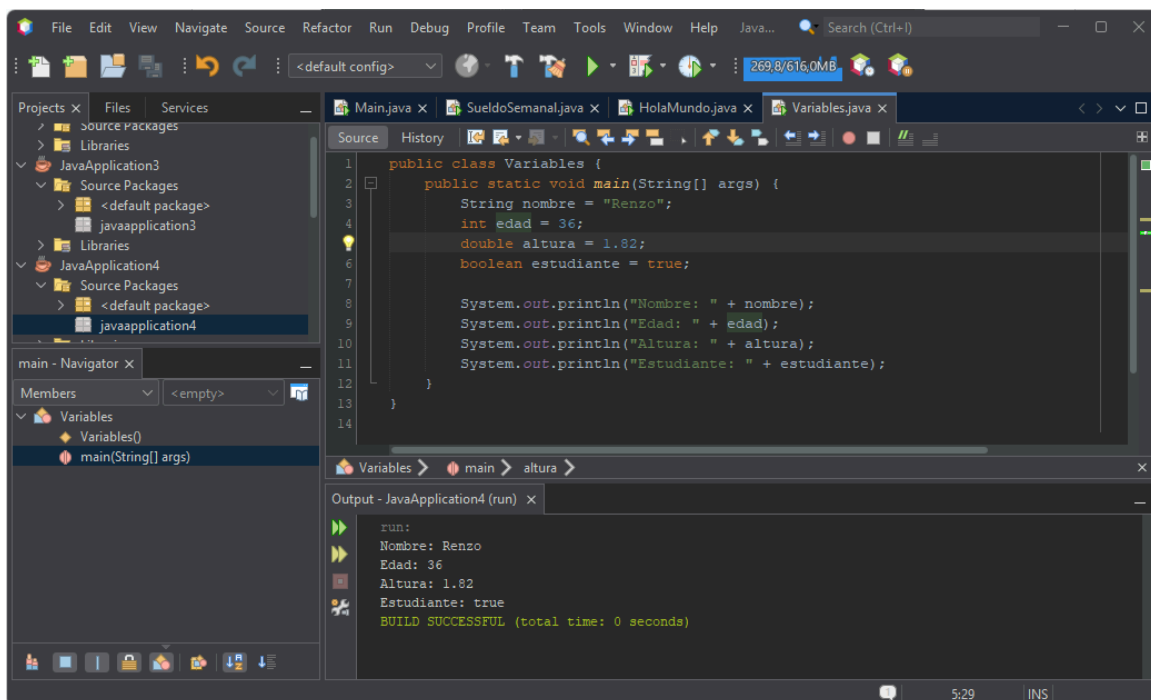


3. Declaración de variables y mostrar valores

Escribí un programa que declara variables con valores asignados de distintos tipos:

```
public class Variables {  
    public static void main(String[] args) {  
        String nombre = "Carlos";  
        int edad = 22;  
        double altura = 1.75;  
        boolean estudiante = true;  
  
        System.out.println("Nombre: " + nombre);  
        System.out.println("Edad: " + edad);  
        System.out.println("Altura: " + altura);  
        System.out.println("Estudiante: " + estudiante);  
    }  
}
```

Al ejecutar, imprime en consola cada valor de las variables declaradas.



4. Entrada de datos con Scanner

Ahora hice un programa que solicita al usuario ingresar su nombre y edad, y luego los muestra en pantalla. Usé `Scanner` para leer la entrada desde teclado.

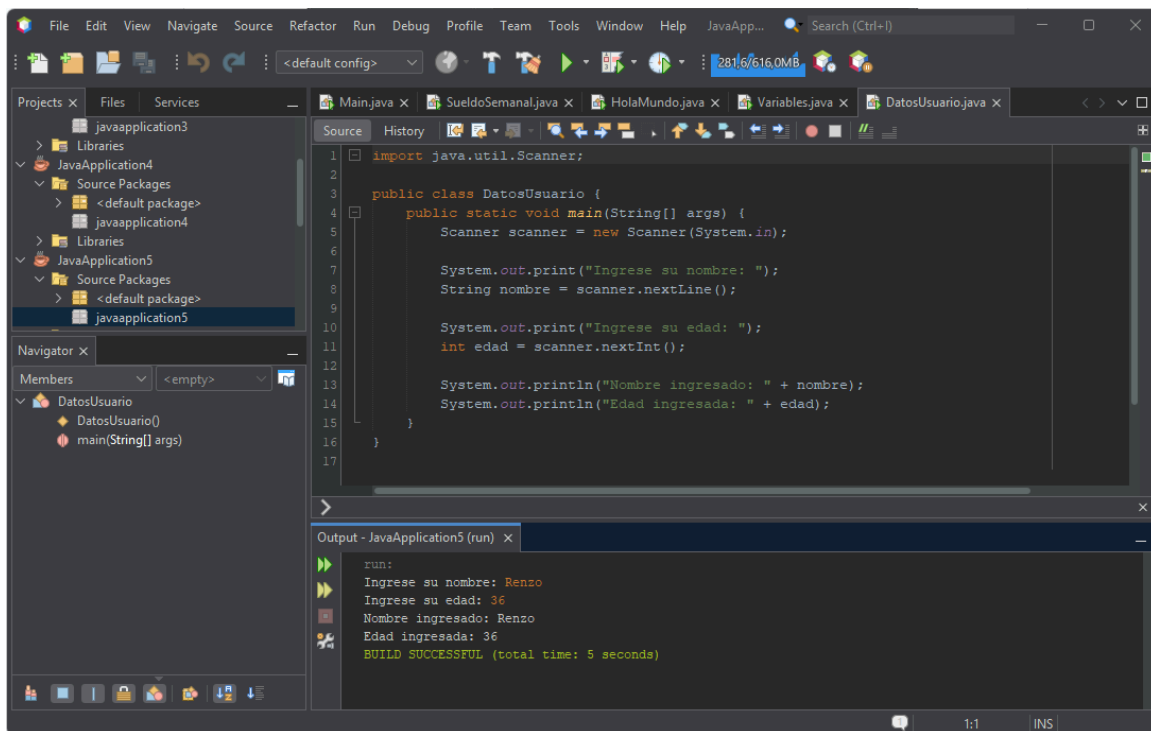
```
import java.util.Scanner;

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

        System.out.print("Ingrese su nombre: ");
        String nombre = scanner.nextLine();

        System.out.print("Ingrese su edad: ");
        int edad = scanner.nextInt();

        System.out.println("Nombre ingresado: " + nombre);
        System.out.println("Edad ingresada: " + edad);
    }
}
```



5. Operaciones aritméticas básicas

El siguiente programa pide dos números enteros y realiza suma, resta, multiplicación y división, mostrando los resultados:

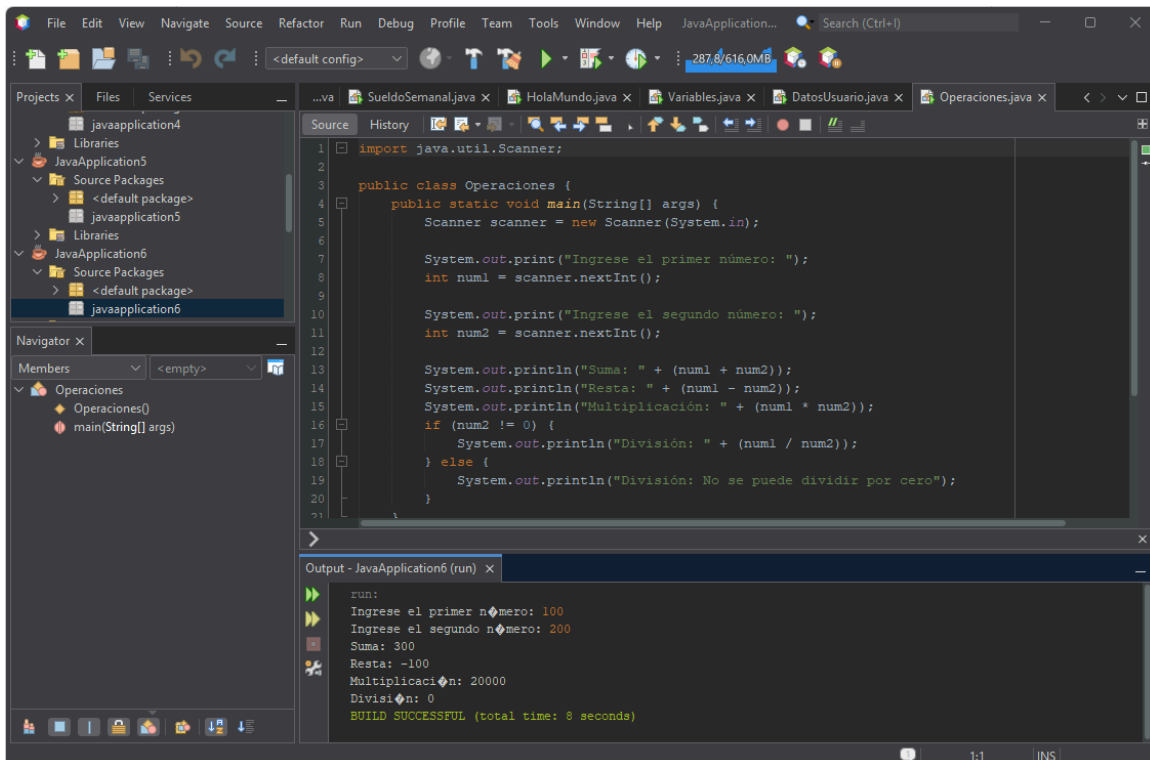
```
import java.util.Scanner;

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

        System.out.print("Ingrese el primer número: ");
        int num1 = scanner.nextInt();

        System.out.print("Ingrese el segundo número: ");
        int num2 = scanner.nextInt();

        System.out.println("Suma: " + (num1 + num2));
        System.out.println("Resta: " + (num1 - num2));
        System.out.println("Multiplicación: " + (num1 * num2));
        if (num2 != 0) {
            System.out.println("División: " + (num1 / num2));
        } else {
            System.out.println("División: No se puede dividir por
cero");
        }
    }
}
```



The screenshot shows an IDE window with the following components:

- Source Editor:** Displays the Java code for the `Operaciones` class, including imports, class definition, and the `main` method with arithmetic operations and conditional division.
- Navigator:** Shows the project structure with `Operaciones` and its `main` method.
- Output Console:** Shows the execution results for `JavaApplication6 (run)`, including user input (100 and 200) and calculated results (Suma: 300, Resta: -100, Multiplicación: 20000, División: 0).

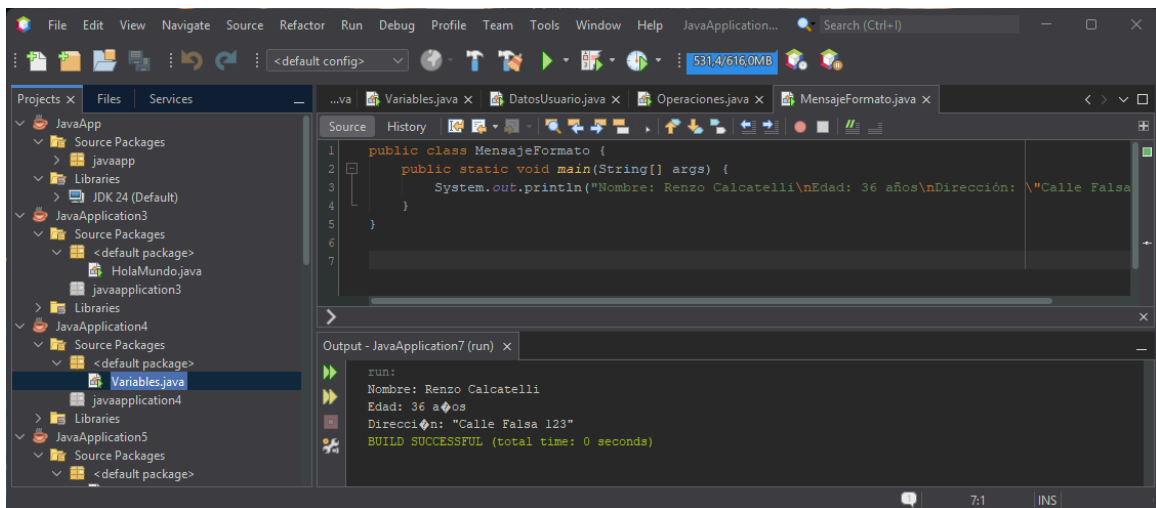
```
run:
Ingrese el primer número: 100
Ingrese el segundo número: 200
Suma: 300
Resta: -100
Multiplicación: 20000
División: 0
BUILD SUCCESSFUL (total time: 8 seconds)
```

6. Mensaje con caracteres de escape

Para mostrar un mensaje con formato, usé caracteres especiales como `\n` para salto de línea y `\"` para comillas dobles:

```
public class MensajeFormato {  
    public static void main(String[] args) {  
        System.out.println("Nombre: Renzo Calcatelli\nEdad: 36  
años\nDirección: \"Calle Falsa 123\"");  
    }  
}
```

La consola muestra el texto con el formato solicitado.



7. Expresiones vs Instrucciones

El código analizado es:

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

- **Expresiones:** Son fragmentos de código que producen un valor. En este caso, `x + 5` es una expresión porque calcula un valor (15).
- **Instrucciones:** Son órdenes que realiza el programa. Por ejemplo, la línea `x = x + 5`; es una instrucción porque asigna un valor a la variable. También `int x = 10`; y `System.out.println(x)`; son instrucciones porque definen y ejecutan acciones.

En resumen, las expresiones calculan valores y las instrucciones ejecutan acciones o cambios en el programa.

8. Conversiones de tipo y división

a) División con números enteros:

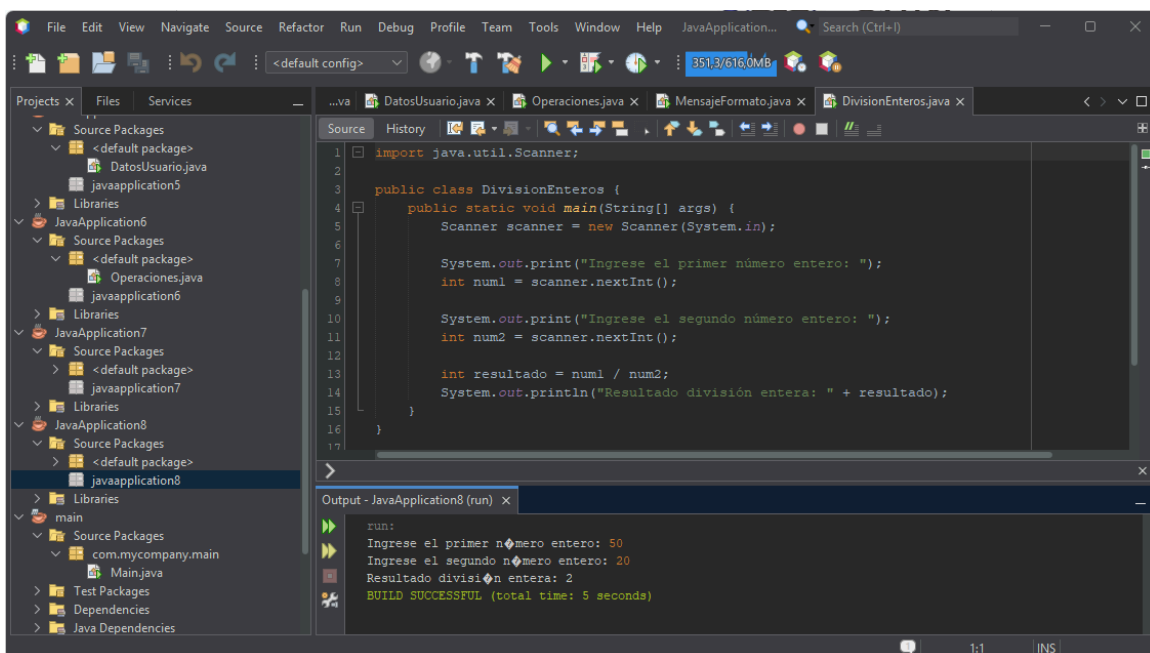
```
import java.util.Scanner;

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

        System.out.print("Ingrese el primer número entero: ");
        int num1 = scanner.nextInt();

        System.out.print("Ingrese el segundo número entero: ");
        int num2 = scanner.nextInt();

        int resultado = num1 / num2;
        System.out.println("Resultado división entera: " + resultado);
    }
}
```



b) Modificación para usar **double** y obtener resultado con decimales:

```
import java.util.Scanner;

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

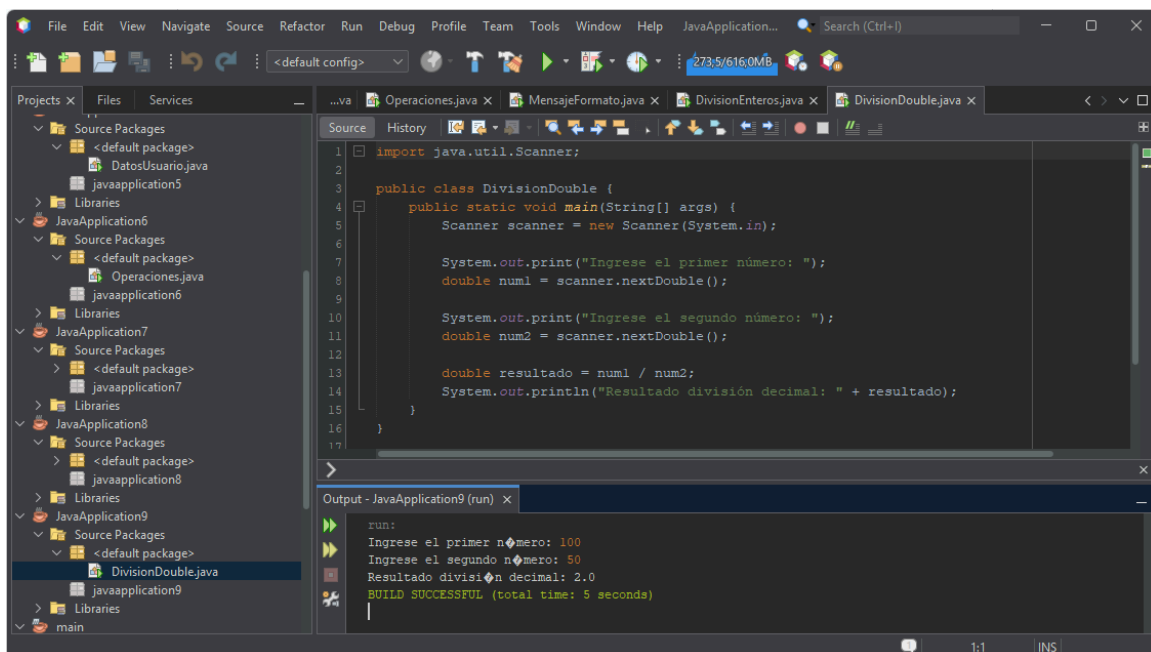
        System.out.print("Ingrese el primer número: ");
        double num1 = scanner.nextDouble();

        System.out.print("Ingrese el segundo número: ");
        double num2 = scanner.nextDouble();

        double resultado = num1 / num2;
        System.out.println("Resultado división decimal: " +
resultado);
    }
}
```

Comparación:

Con enteros la división trunca el resultado (no muestra decimales). Con **double** se obtienen los decimales reales de la división.

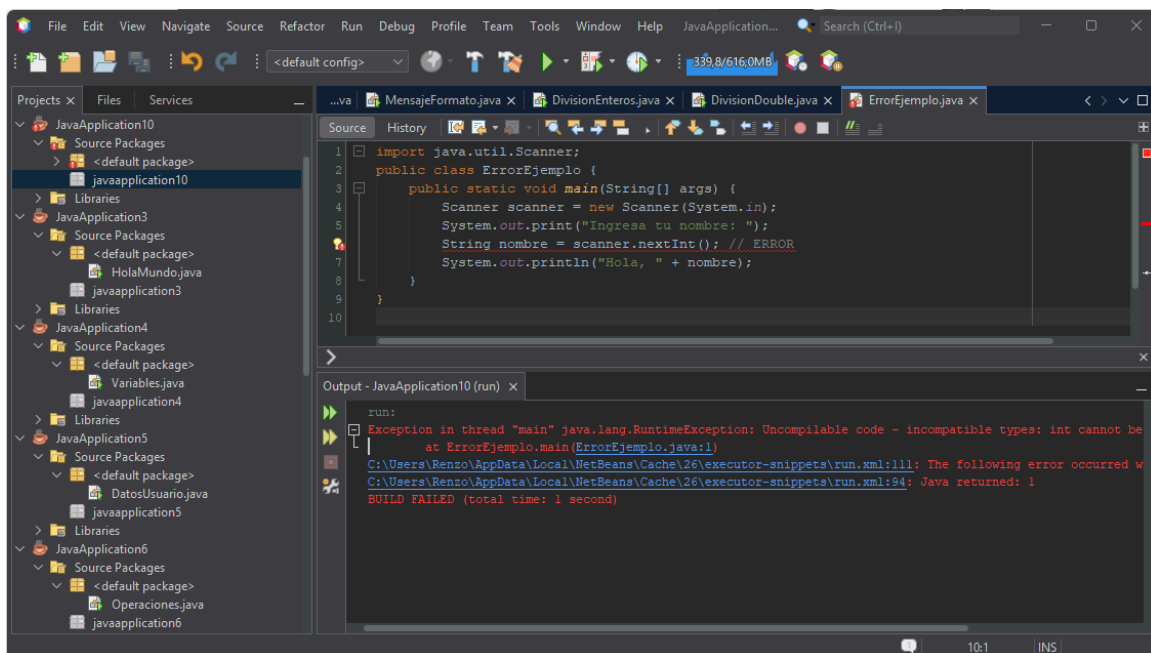


9. Corrección de código con error

El 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);
    }
}
```

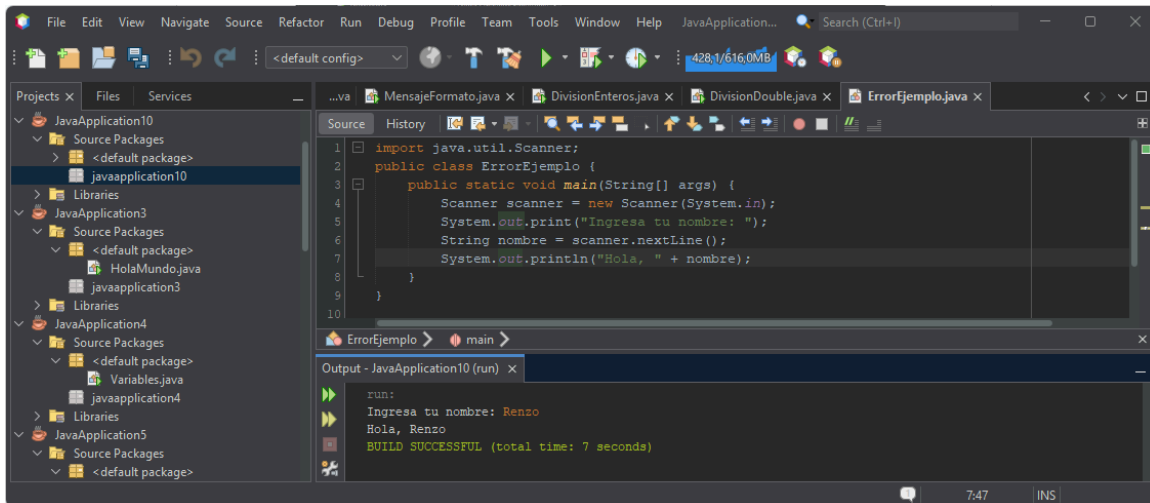
Error: Se usó `scanner.nextInt()` para leer un nombre, pero `nextInt()` es para números enteros, no para texto.



Corrección:

```
String nombre = scanner.nextLine();
```

Así se lee una línea de texto correctamente.



10. Prueba de escritorio

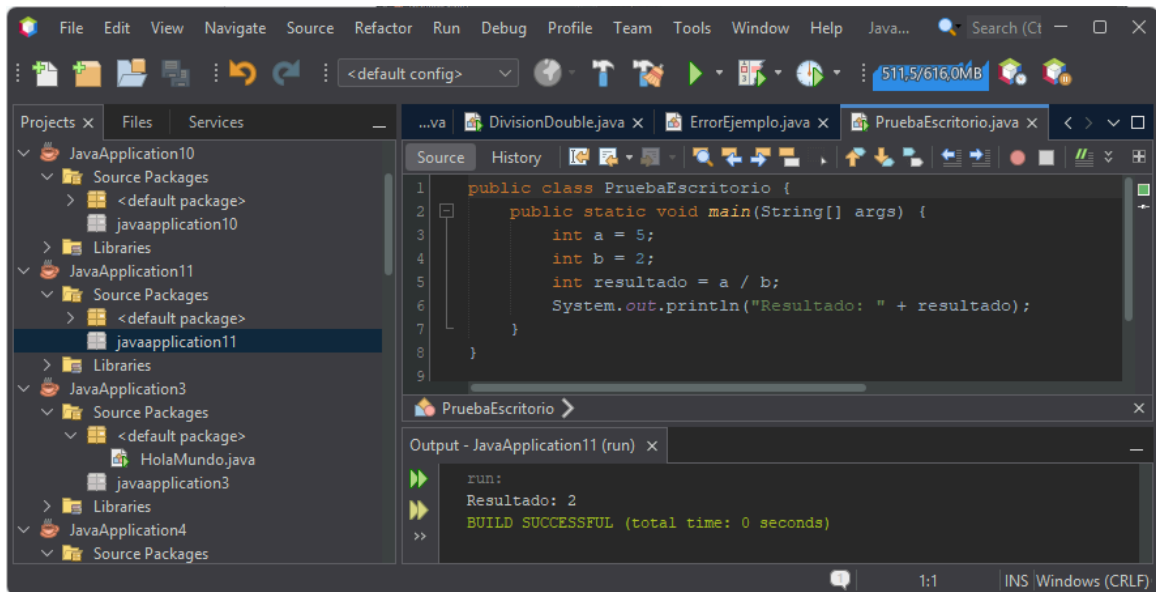
Código:

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

Variable	Valor
a	5
b	2
resultado	2

Explicación:

La división $5 / 2$ entre enteros trunca la parte decimal, resultando en 2. Por eso **resultado** vale 2 y no 2.5.



Conclusiones

Este trabajo me ayudó a comprender conceptos básicos de Java y a familiarizarme con NetBeans. Aprendí cómo declarar variables, leer datos con Scanner, usar operadores, y corregir errores comunes. También comprendí la diferencia entre expresiones e instrucciones, y la importancia de los tipos de datos y conversiones para obtener resultados correctos.

Además, practicar con la consola y capturar pantallas me permitió ver el funcionamiento real de los programas y mejorar mi capacidad para depurar y entender el código paso a paso.