

Trabajo Práctico N° 1: Introducción a Java

Carrera: Tecnicatura Universitaria en Programación

Materia: Programación II

Profesora: Cinthia Rigoni

Tutor: Luciano Chirolí

Estudiante: Natalia Córdoba

Resolución del Caso Práctico

Este trabajo adjunta las capturas de pantalla y las respuestas teóricas solicitadas para la resolución de algunos ejercicios junto con el link al repositorio de GitHub donde se encuentra el código realizado para el resto de los puntos del trabajo.

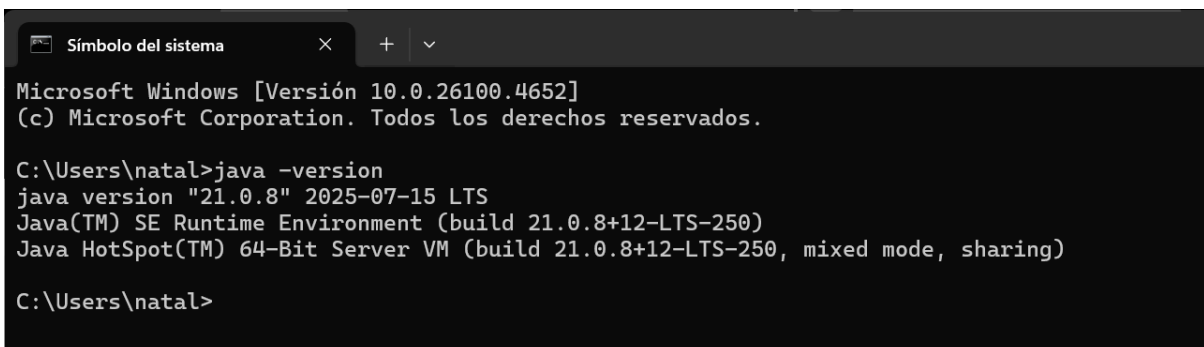
Se creó un repositorio general para la entrega de todos los trabajos prácticos de esta materia. Este trabajo se encuentra dentro de la carpeta **01 Introducción a Java**. Cada ejercicio se creó como una clase dentro del mismo proyecto de modo que la entrega sea más organizada.

→ [Link al repositorio de GitHub](#)

Capturas de pantalla

1. Verificación de la instalación de JDK y NetBeans

a. Captura comprobación de la instalación con el comando **java -version**:

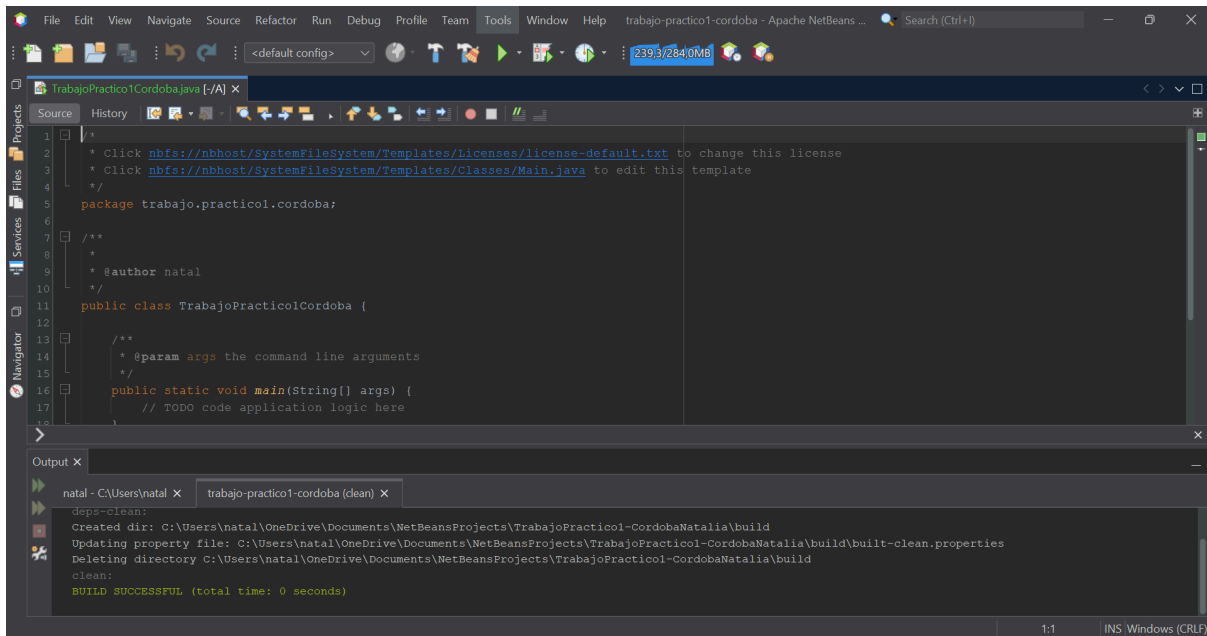


```
Símbolo del sistema
Microsoft Windows [Versión 10.0.26100.4652]
(c) Microsoft Corporation. Todos los derechos reservados.

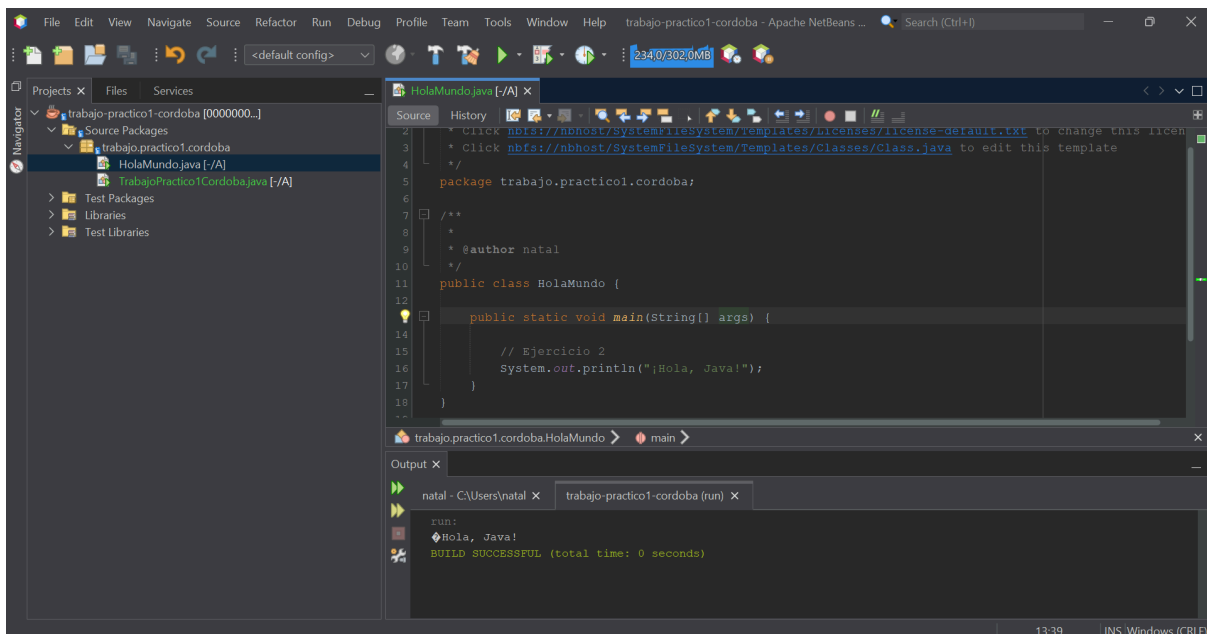
C:\Users\natal>java -version
java version "21.0.8" 2025-07-15 LTS
Java(TM) SE Runtime Environment (build 21.0.8+12-LTS-250)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.8+12-LTS-250, mixed mode, sharing)

C:\Users\natal>
```

- b. Captura de la creación de un nuevo proyecto y la configuración del modo oscuro:



2. Captura ejecución ejercicio 2:

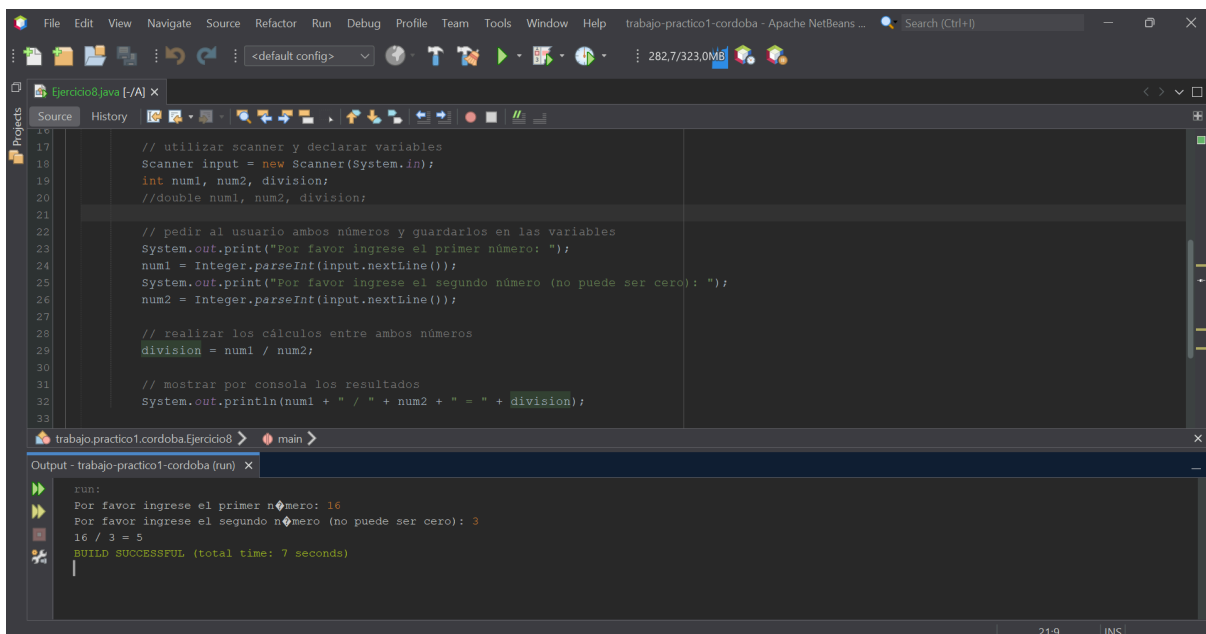


7. Como vimos en los videos una expresión es un fragmento de código que evalúa un valor, mientras que una instrucción es una unidad completa de código que realiza una acción.

Por lo tanto, como las expresiones son fragmentos de código, en el ejemplo dado las tres líneas de código son instrucciones, aunque algunas de ellas contienen dentro una expresión: la línea 1 es una instrucción de declaración y asignación que incluye la expresión **10**, la línea 2 es una instrucción de asignación que contiene la expresión **x + 5** y la línea 3 es una instrucción de salida.

8. Las divisiones en Java muestran diferentes resultados por consola dependiendo del tipo de variables que utilicemos. Por ejemplo, si todas las variables son enteros, obtendremos como resultado solo la parte entera de la división, es decir que se descartan los decimales. Para obtener la parte decimal de la división al menos uno de los operadores (divisor o dividendo) debe ser de tipo **double o float**.

Si utilizamos en el programa todas variables de tipo **int** (num1, num2, division), obtenemos como resultado de la división 16/3 el valor 5.

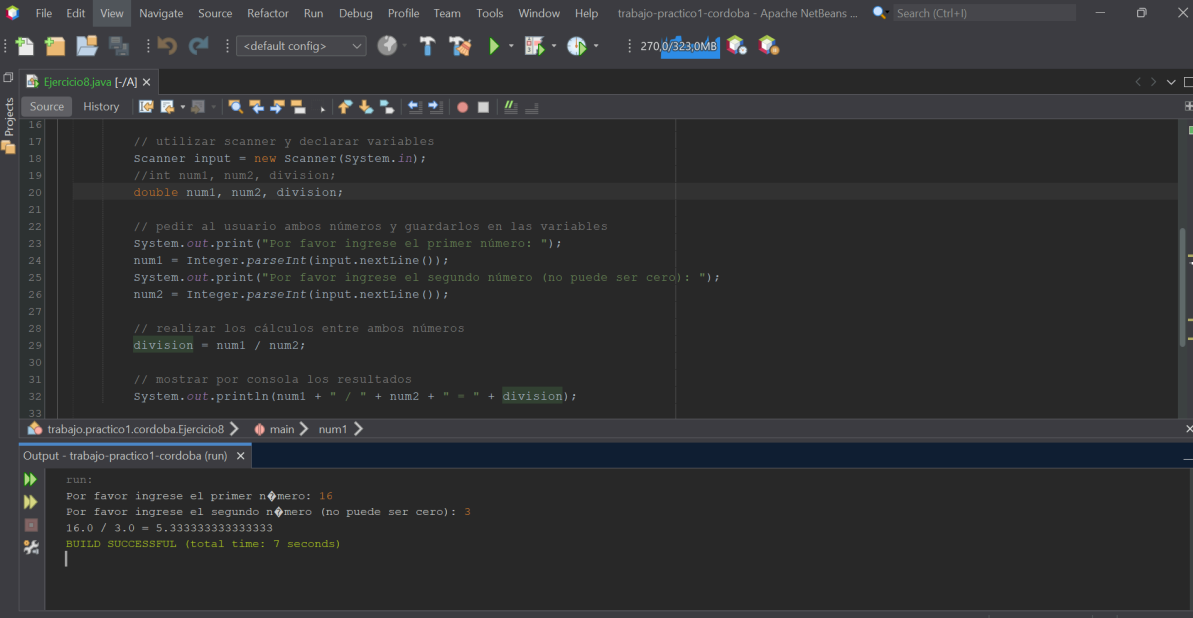


```
16 // utilizar scanner y declarar variables
17 Scanner input = new Scanner(System.in);
18 int num1, num2, division;
19 //double num1, num2, division;
20
21
22 // pedir al usuario ambos números y guardarlos en las variables
23 System.out.print("Por favor ingrese el primer número: ");
24 num1 = Integer.parseInt(input.nextLine());
25 System.out.print("Por favor ingrese el segundo número (no puede ser cero): ");
26 num2 = Integer.parseInt(input.nextLine());
27
28 // realizar los cálculos entre ambos números
29 division = num1 / num2;
30
31 // mostrar por consola los resultados
32 System.out.println(num1 + " / " + num2 + " = " + division);
33
```

Output - trabajo-practico1-cordoba (run)

```
run:
Por favor ingrese el primer número: 16
Por favor ingrese el segundo número (no puede ser cero): 3
16 / 3 = 5
BUILD SUCCESSFUL (total time: 7 seconds)
```

Si modificamos el código y determinamos que todas las variables sean de tipo **double** en lugar de **int**, podemos ver que en este caso Java si devuelve la división con decimales dando como resultado 5,33333...

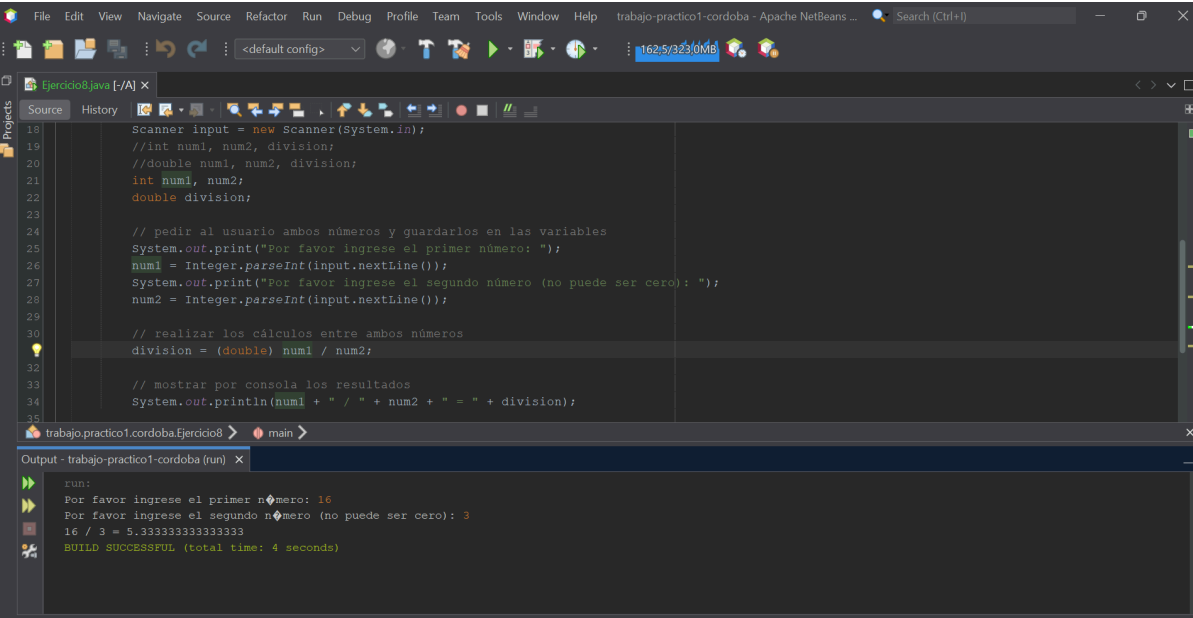


```
16
17 // utilizar scanner y declarar variables
18 Scanner input = new Scanner(System.in);
19 //int num1, num2, division;
20 double num1, num2, division;
21
22 // pedir al usuario ambos números y guardarlos en las variables
23 System.out.print("Por favor ingrese el primer número: ");
24 num1 = Integer.parseInt(input.nextLine());
25 System.out.print("Por favor ingrese el segundo número (no puede ser cero): ");
26 num2 = Integer.parseInt(input.nextLine());
27
28 // realizar los cálculos entre ambos números
29 division = num1 / num2;
30
31 // mostrar por consola los resultados
32 System.out.println(num1 + " / " + num2 + " = " + division);
33
```

Output - trabajo-practico1-cordoba (run)

```
run:
Por favor ingrese el primer número: 16
Por favor ingrese el segundo número (no puede ser cero): 3
16.0 / 3.0 = 5.333333333333333
BUILD SUCCESSFUL (total time: 7 seconds)
```

En el caso de que no quisiéramos modificar el tipo de variable de los números ingresados, una opción para observar correctamente los resultados de la división podría ser: dejar las variables num1 y num2 como **int**, cambiar la variable división a **double** para que acepte valores con decimales y, por último utilizar la herramienta **casting** que brinda java para modificar el valor de uno de los números implicados en la división pero solo en el momento de realizar el cálculo.



The screenshot shows the NetBeans IDE with a Java file named 'Ejercicio8.java'. The code uses a Scanner to read two integers from the user and calculates their division. The output window shows the program running successfully with the input values 16 and 3, resulting in a division of 5.333333333333333.

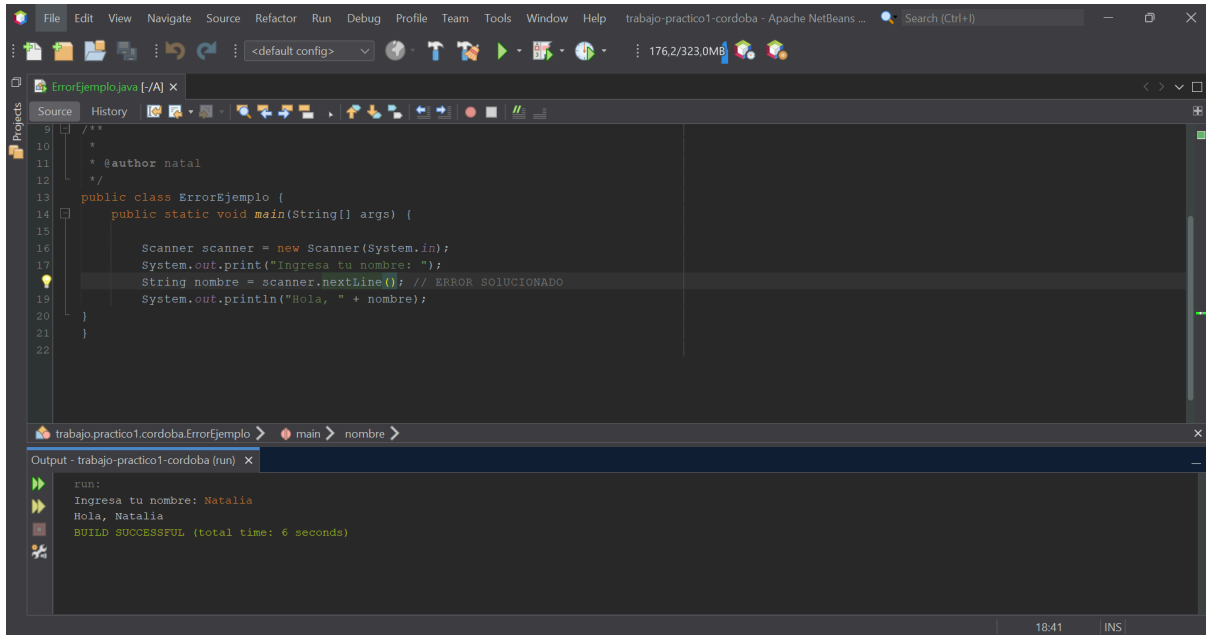
```
18 Scanner input = new Scanner(System.in);
19 //int num1, num2, division;
20 //double num1, num2, division;
21 int num1, num2;
22 double division;
23
24 // pedir al usuario ambos números y guardarlos en las variables
25 System.out.print("Por favor ingrese el primer número: ");
26 num1 = Integer.parseInt(input.nextLine());
27 System.out.print("Por favor ingrese el segundo número (no puede ser cero): ");
28 num2 = Integer.parseInt(input.nextLine());
29
30 // realizar los cálculos entre ambos números
31 division = (double) num1 / num2;
32
33 // mostrar por consola los resultados
34 System.out.println(num1 + " / " + num2 + " = " + division);
35
```

Output - trabajo-practico1-cordoba (run)

```
run:
Por favor ingrese el primer número: 16
Por favor ingrese el segundo número (no puede ser cero): 3
16 / 3 = 5.333333333333333
BUILD SUCCESSFUL (total time: 4 seconds)
```

9. El error del código está en el uso del método `nextInt()` para leer el valor de la variable nombre que es ingresada por el usuario. La variable nombre está correctamente declarada de tipo `String`, pero se estaba utilizando para almacenar y leer su valor el método `nextInt()`, que permite leer números enteros. Este método está esperando que el usuario ingrese números en lugar de caracteres, por eso sucede un error de incompatibilidad en los tipos de datos.

La solución sería usar el método `nextLine()`, que permite almacenar el valor ingresado por el usuario de tipo `String`; por lo tanto se espera que el usuario ingrese caracteres y funciona correctamente.



```

9  /**
10  *
11  * @author natal
12  */
13  public class ErrorEjemplo {
14      public static void main(String[] args) {
15
16          Scanner scanner = new Scanner(System.in);
17          System.out.print("Ingresa tu nombre: ");
18          String nombre = scanner.nextLine(); // ERROR SOLUCIONADO
19          System.out.println("Hola, " + nombre);
20      }
21  }
22

```

Output - trabajo-practico1-cordoba (run)

```

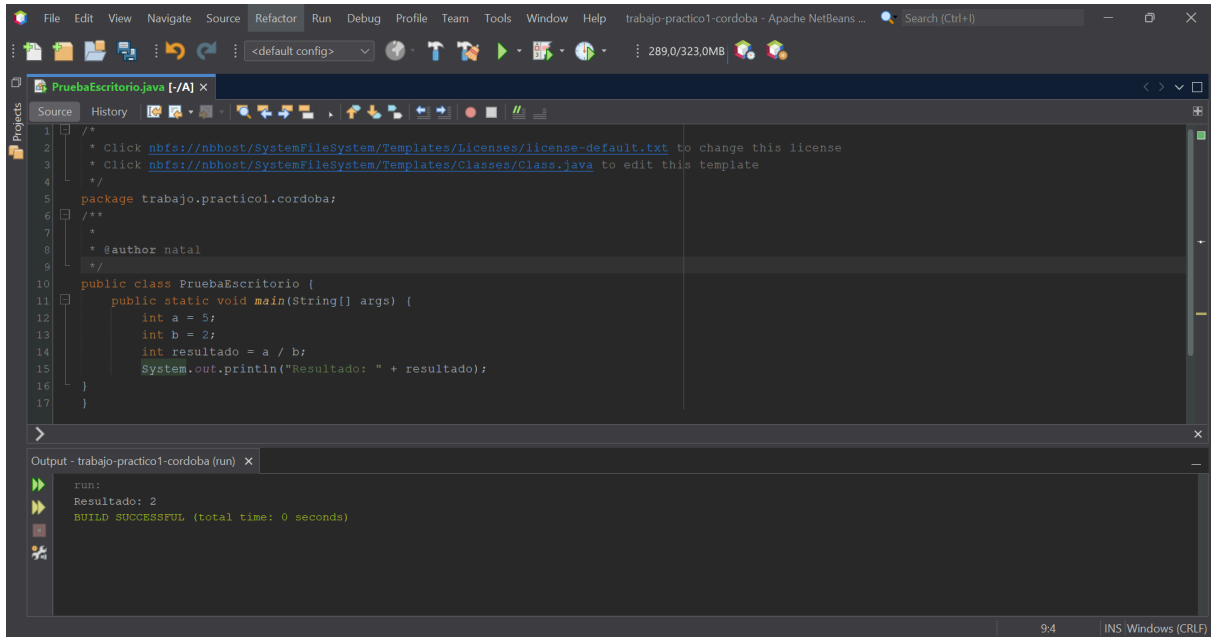
run:
Ingresa tu nombre: Natalia
Hola, Natalia
BUILD SUCCESSFUL (total time: 6 seconds)

```

10. Prueba de Escritorio

Línea	a	b	resultado	Observaciones
1 a 11	-	-	-	Código previo (comentarios, paquete, etc.) sin variables declaradas ni inicializadas
12	5	-	-	Se declara e inicializa la variable a con el valor 5
13	5	2	-	Se declara e inicializa la variable b con el valor 2
14	5	2	2	Se declara la variable resultado con el valor $a / b = 2$
15	5	2	2	Se imprime el valor de la variable resultado por consola ("Resultado: 2")
16 a 17	5	2	2	Fin del programa

El valor de la variable resultado es 2, por lo que se mencionó en el ejercicio anterior, como todas las variables son de tipo entero (a, b, resultado) obtenemos solo la parte entera de la división, descartando los decimales.



The screenshot shows the NetBeans IDE interface. The main editor window displays the file `PruebaEscritorio.java` with the following 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 trabajo.practico1.cordoba;  
6   /**  
7    *  
8    * @author natal  
9    */  
10  public class PruebaEscritorio {  
11      public static void main(String[] args) {  
12          int a = 5;  
13          int b = 2;  
14          int resultado = a / b;  
15          System.out.println("Resultado: " + resultado);  
16      }  
17  }
```

Below the editor, the Output window shows the results of running the program:

```
run:  
Resultado: 2  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The status bar at the bottom indicates the file is at line 9, column 4, and the encoding is INS Windows (CRLF).