PROGRAMACIÓN II

Trabajo Práctico 1: Introducción a Java

**Alumno: Massazza Sergio**

**Comisión: 8**

**Enlace a github: https://github.com/sermass/UTN\_TUPaD\_ProgII/TP1.git**

# 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

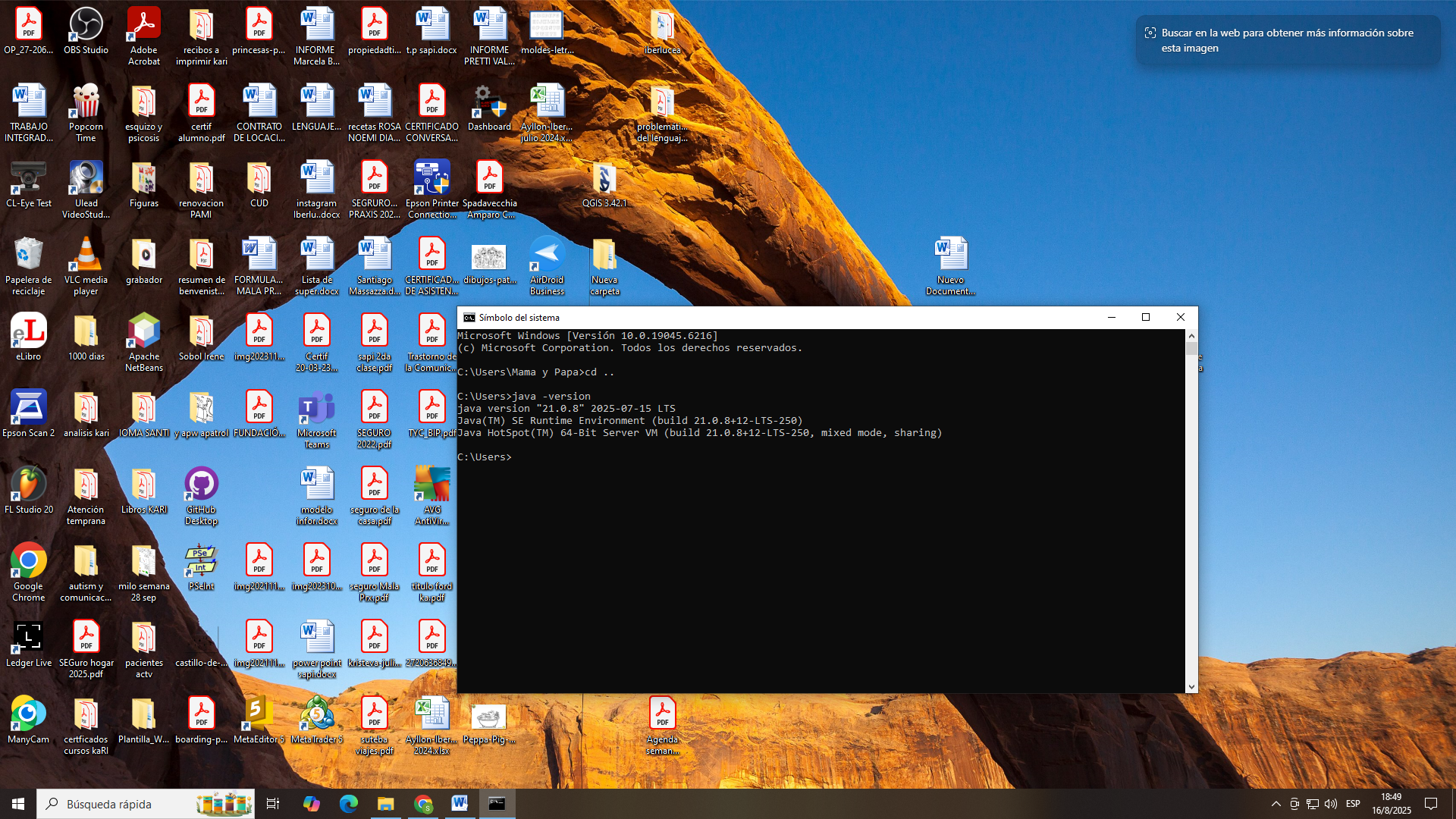
|  |  |
| --- | --- |
| **Concepto** | **Aplicación en el proyecto** |
| Instalación y entorno | Almacenan el conjunto de países |
| Variables y tipos de datos | Representan los datos de cada país (nombre, población, superficie, etc.) |
| Entrada y salida | Separan las operaciones: carga, búsqueda, estadísticas, 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**

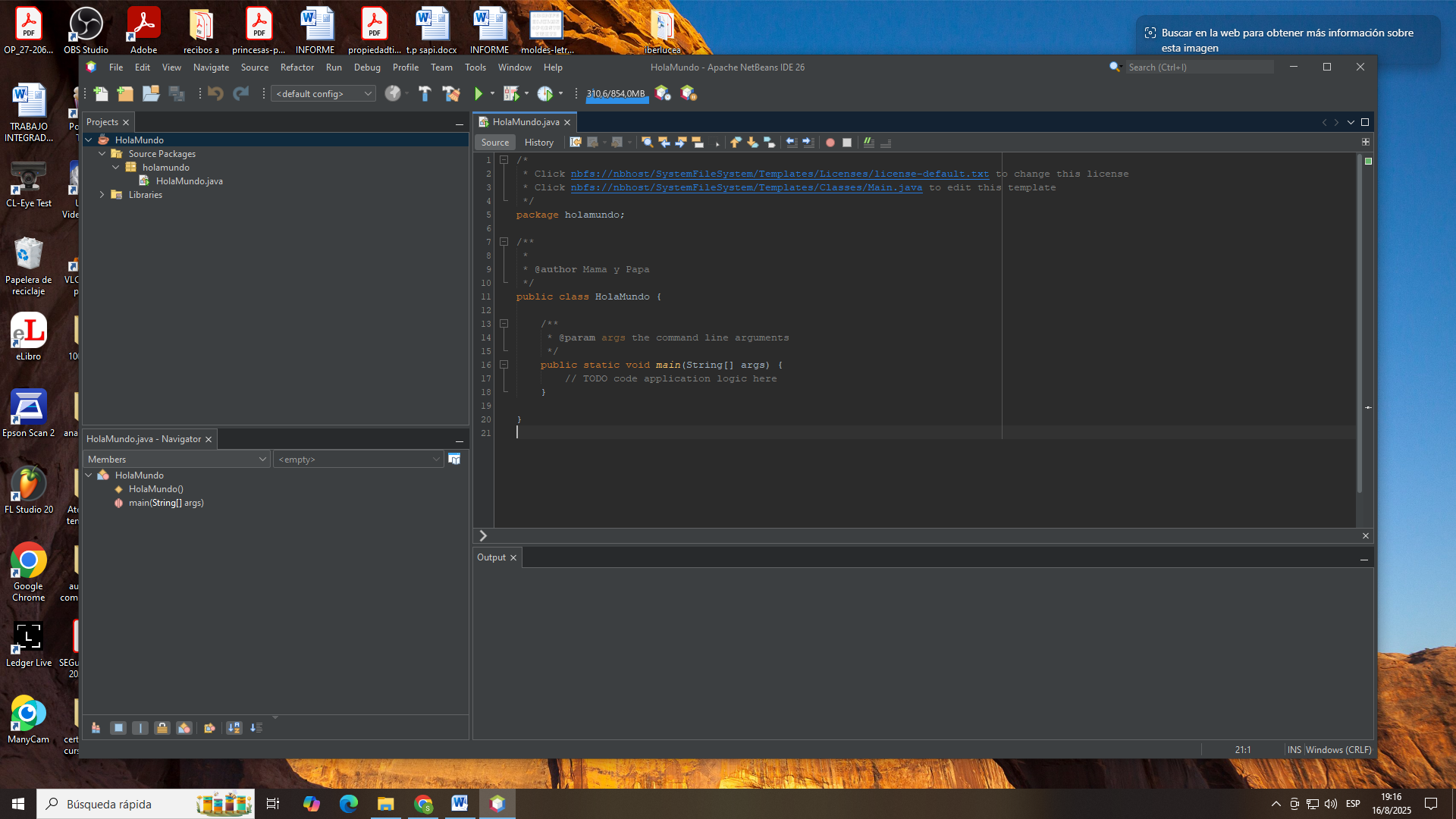
El trabajo consiste en resolver una serie de ejercicios introductorios en Java que permitan:

* 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.

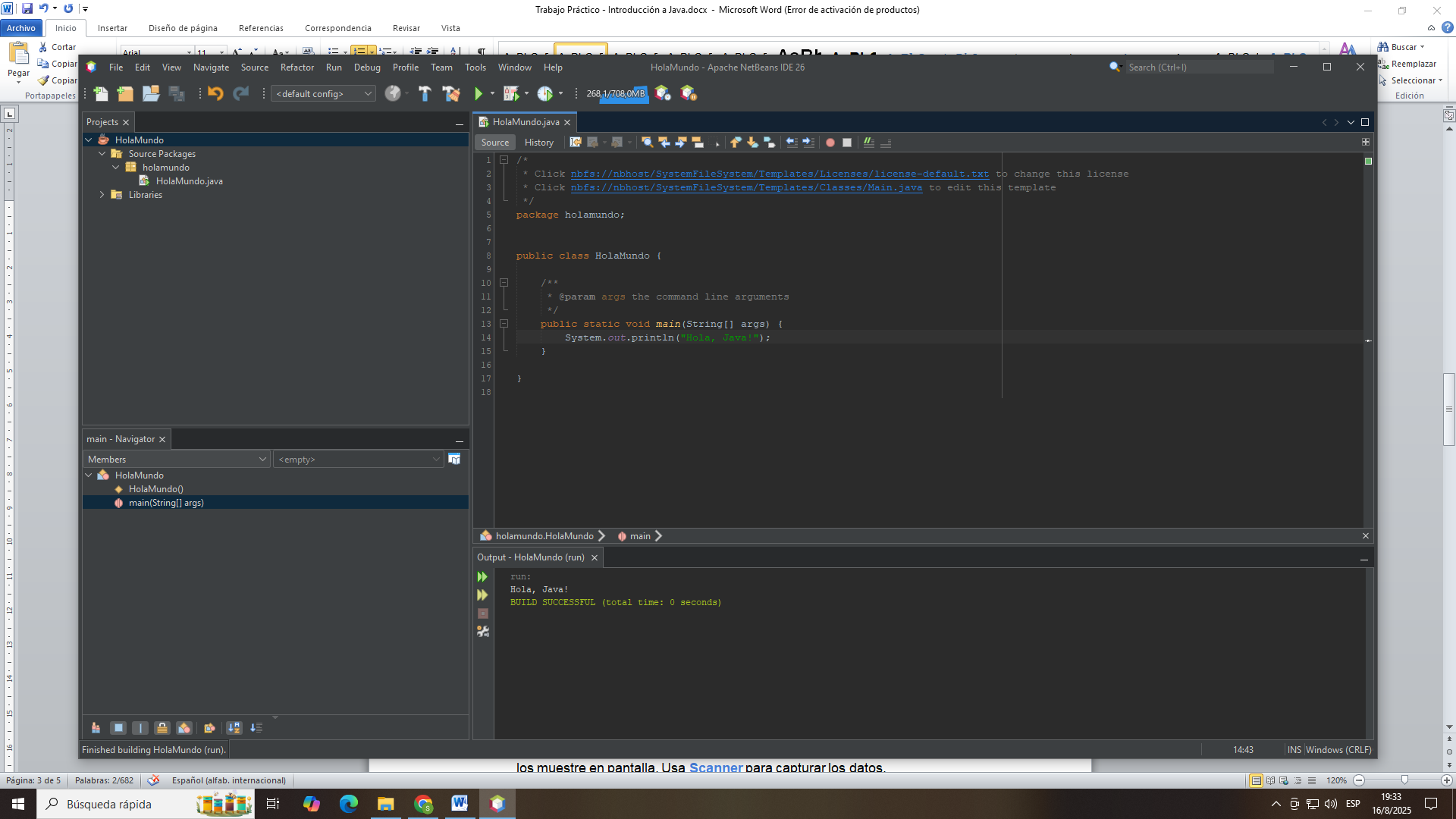
1. Verificar que tienes instalado Java JDK y NetBeans
   1. Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal: **java –versión**



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

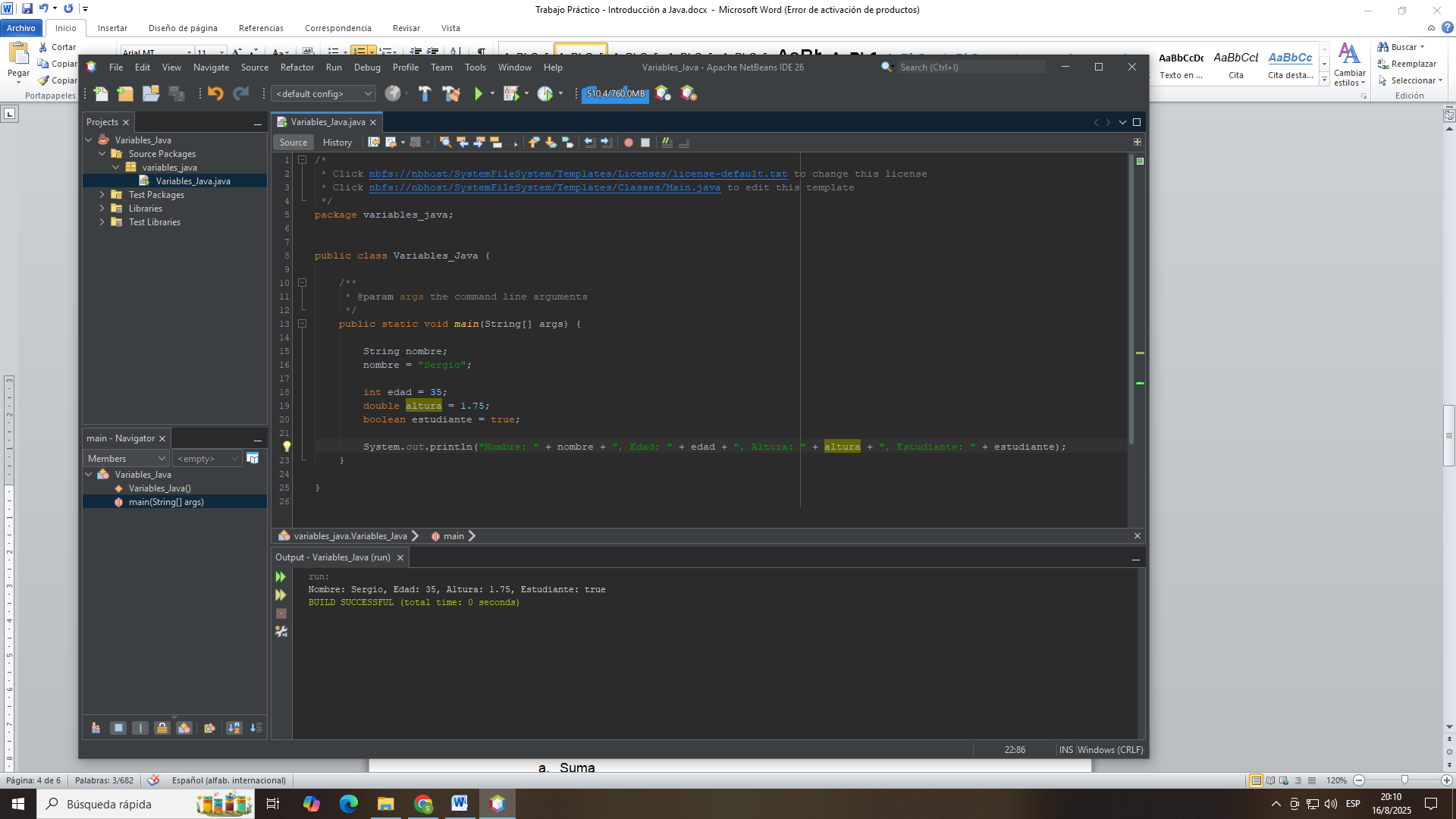


1. Escribir y ejecutar un programa básico en Java.
   1. Creá una clase llamada **HolaMundo**.
   2. Escribe un programa que imprima el mensaje: **¡Hola, Java!**
   3. Ejecuta el programa en NetBeans y adjunta una captura del resultado en la consola.



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

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

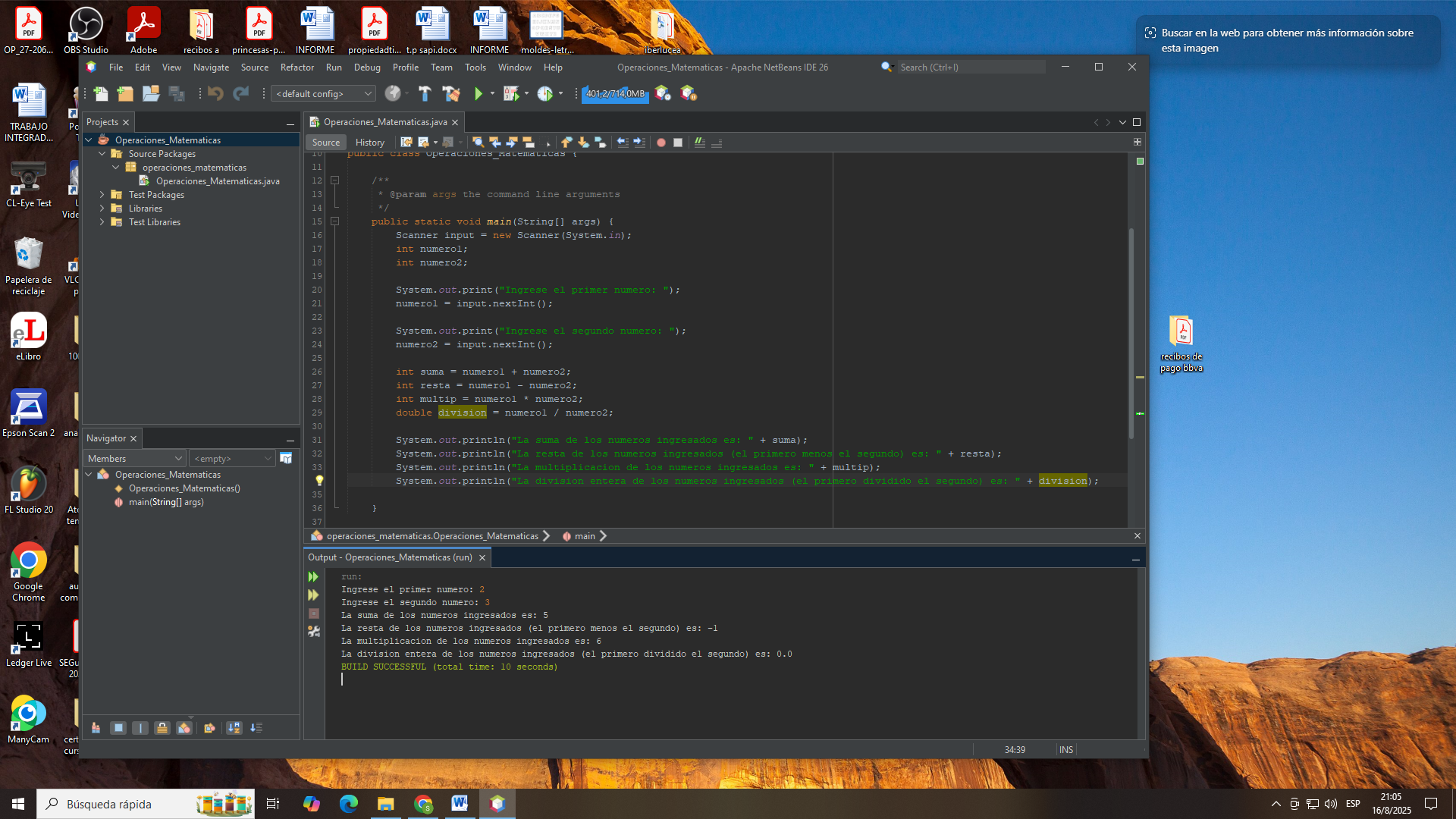


1. Escribe un programa que solicite al usuario ingresar su nombre y edad, y luego los muestre en pantalla. Usa **Scanner** para capturar los datos.



1. Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:
   1. Suma
   2. Resta
   3. Multiplicación
   4. División

Muestra los resultados en la consola.

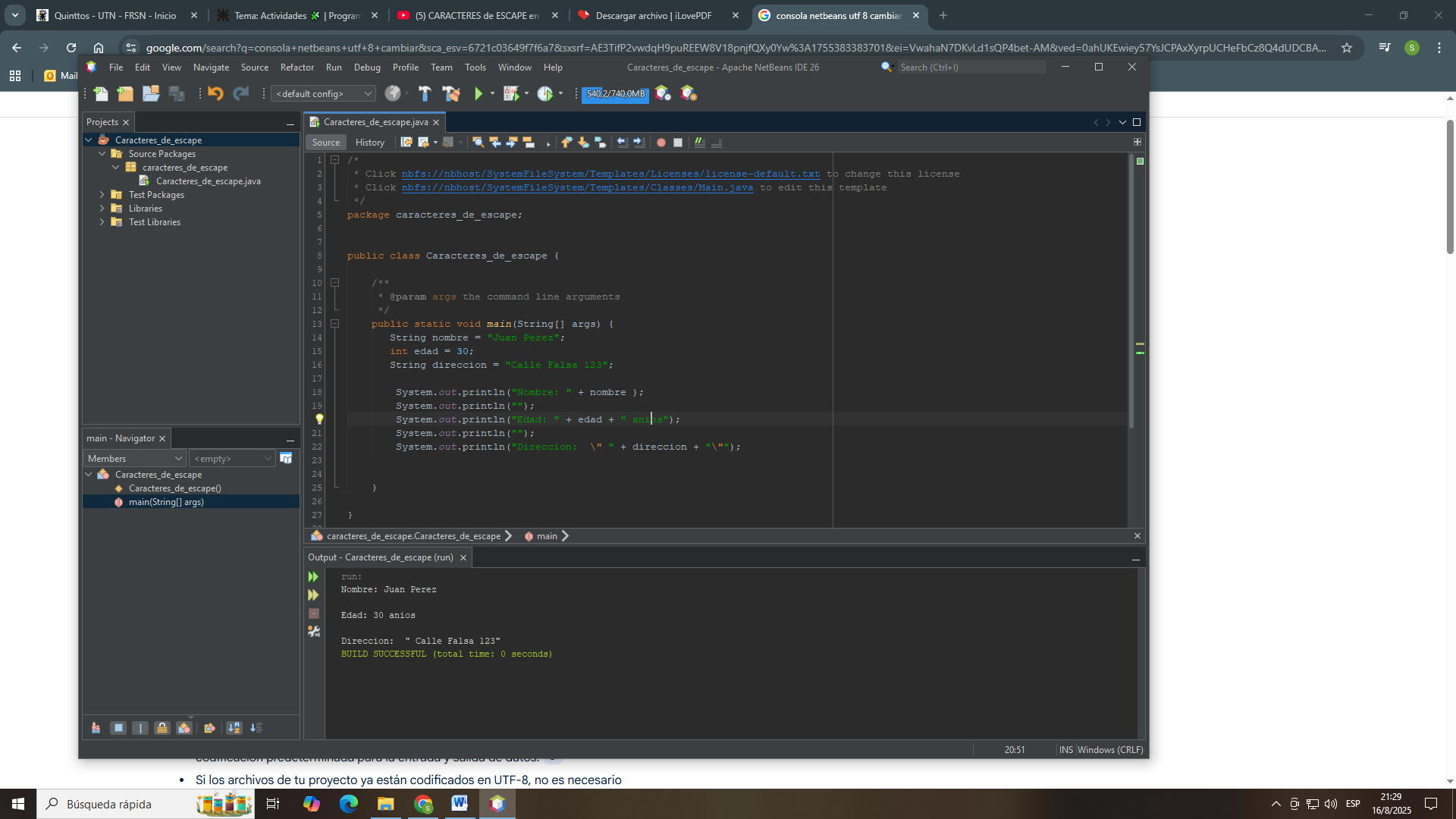


1. Escribe un programa que muestre el siguiente mensaje en consola:

**Nombre: Juan Pérez Edad: 30 años**

**Dirección: "Calle Falsa 123"**

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



1. 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**

Las 3 lineas son instrucciones ya que producen una acción:

**Línea 1**: asigna a la variable entera x el valor 10.

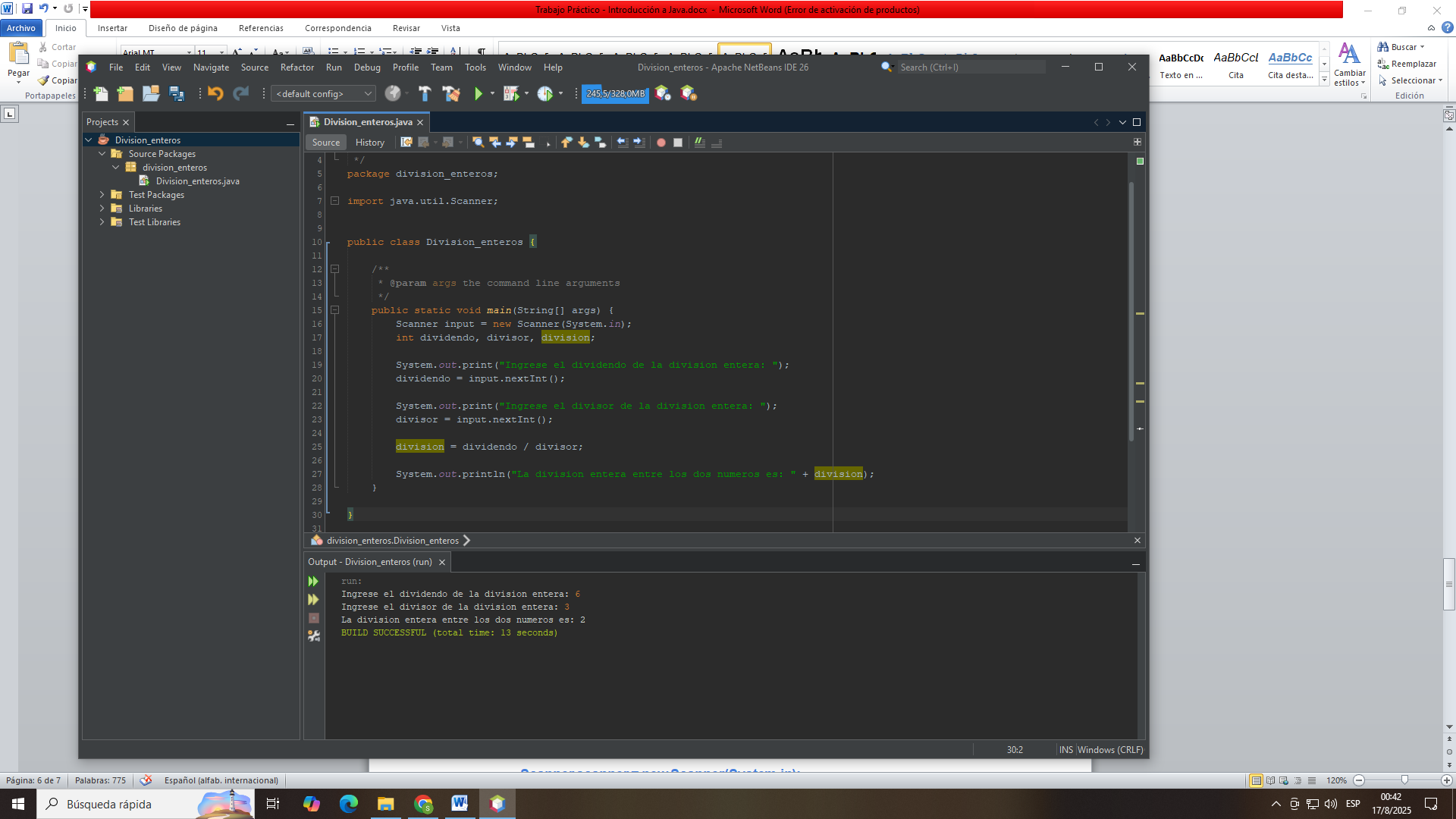
**Línea 2**: asigna a la misma variable x la suma de su valor (10) + 5

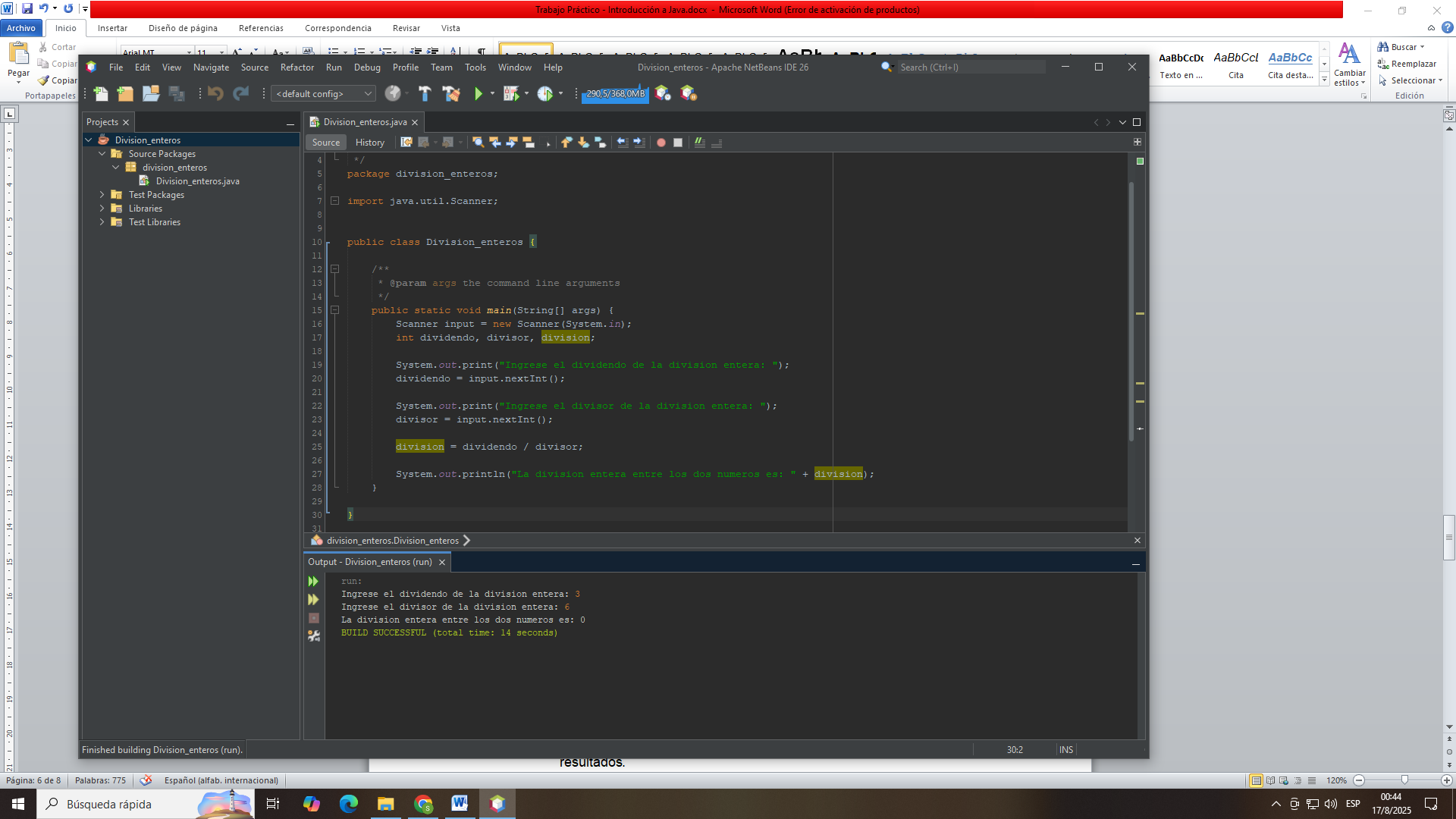
**Línea 3**: imprime en consola el valor de la variable x

Una expresión es un conjunto de valores y operadores que al ser evaluados devuelven un valor mientras que una instrucción es código que produce una acción.

En la línea 2 “x + 5” es una expresión pero “x = x + 5” es una instrucción

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

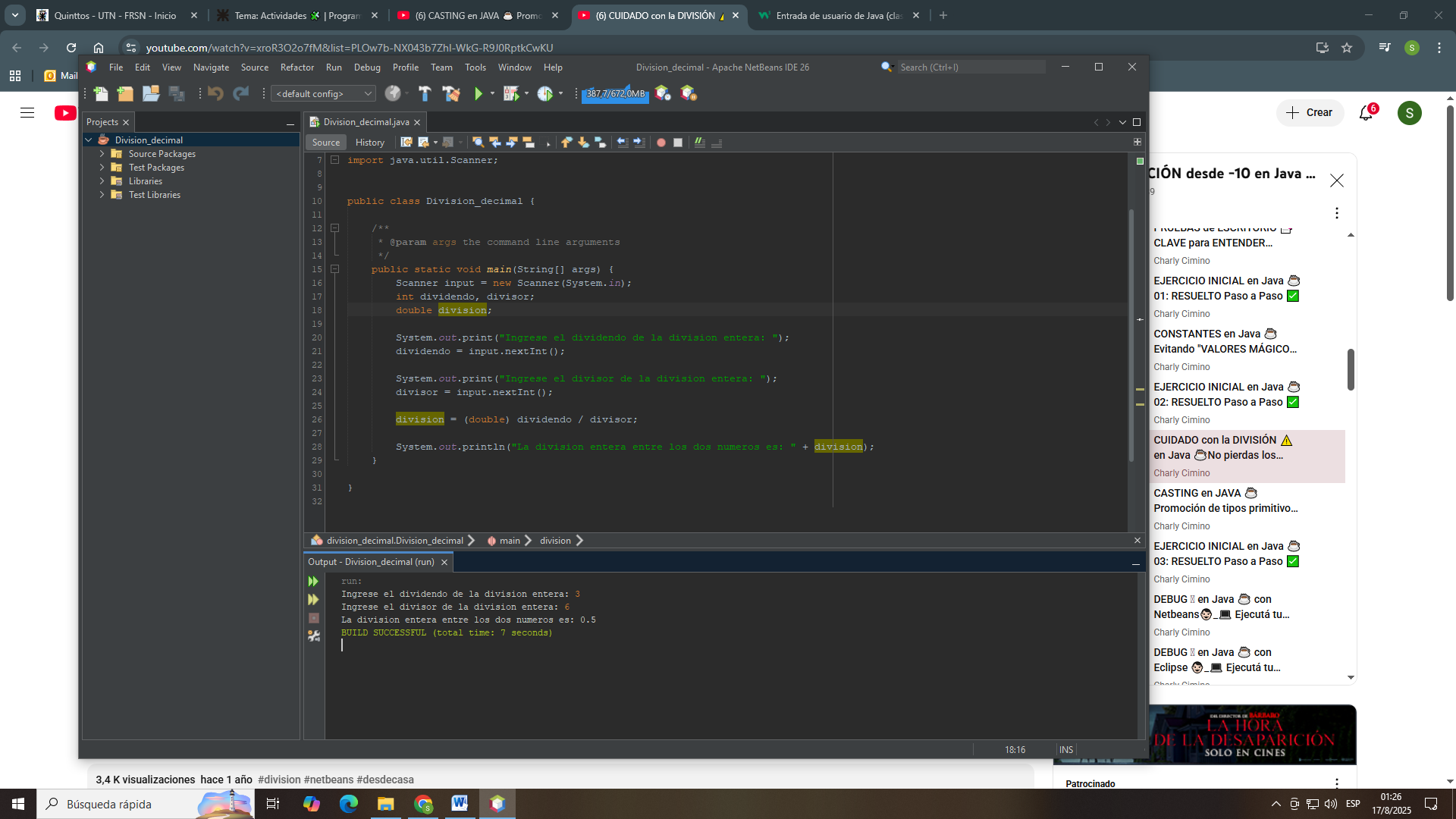




El resultado de dividir 2 enteros en java, siempre es entero y descarta los decimales.

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

Para que java dé como resultado un valor con decimal en una división debe estar o el dividendo o el divisor al menos definido como tipo doublé. Para ello se puede multiplicar por 1.0 alguno de los dos o realizar casting de uno de ellos.



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

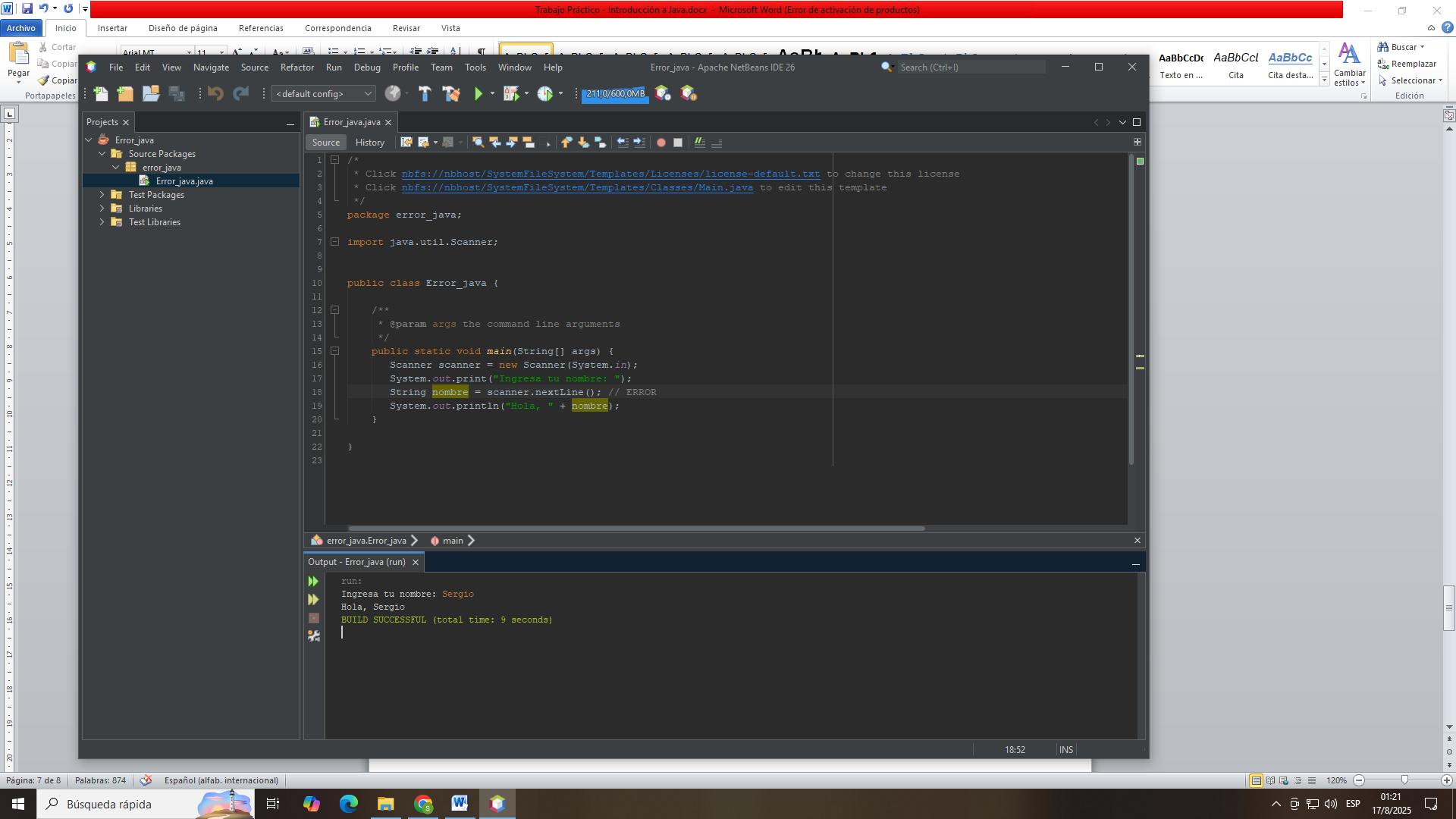
**import java.util.Scanner; public class ErrorEjemplo {**

**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);**

**}**

**}**

En la línea que tiene como comentario ERROR (**String nombre = scanner.nextInt**()) hay incompatibilidad de tipos: nombre es una variable tipo string y le esta asignando un valor tipo entero . Para solucionarlo reemplazararia el método nextInt() por nextLine()



1. 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** | **Variable a** | **Variable b** | **Resultado** |
| **1** | **Sin definir** | **Sin definir** | **Sin definir** |
| **2** | **Sin definir** | **Sin definir** | **Sin definir** |
| **3** | **5** | **Sin definir** | **Sin definir** |
| **4** | **5** | **2** | **Sin definir** |
| **5** | **5** | **2** | **2** |
| **6** | **Consola imprime: “Resultado: 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.