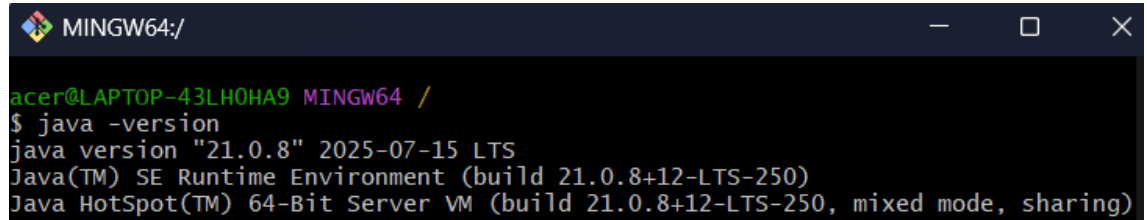


UTN TUPaD - Programación II

Trabajo Práctico 1 - Introducción a Java

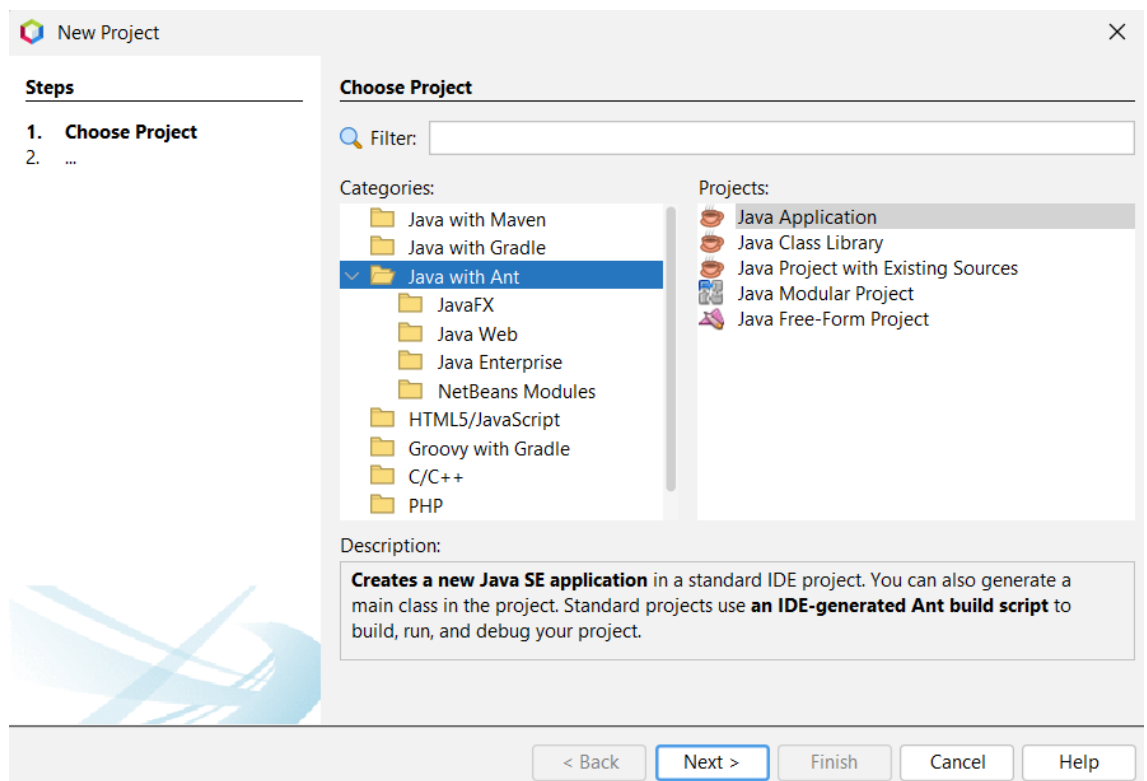
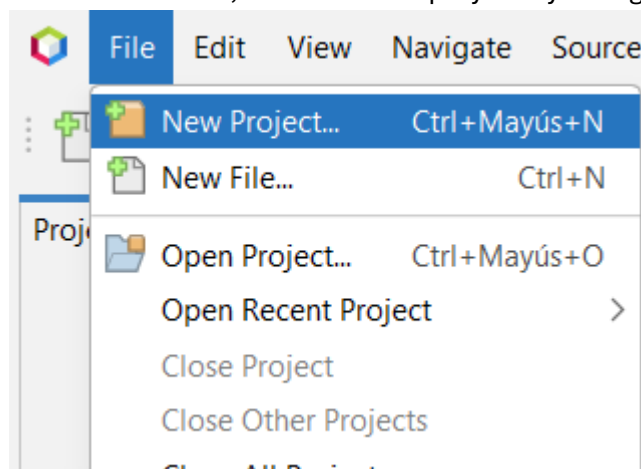
1. Verificar que tienes instalado Java JDK y NetBeans

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



```
MINGW64:/
acer@LAPTOP-43LH0HA9 MINGW64 /
$ 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)
```

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



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

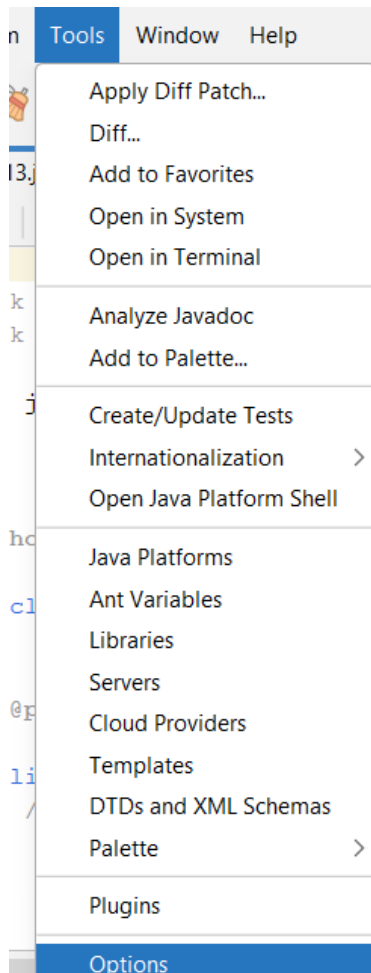
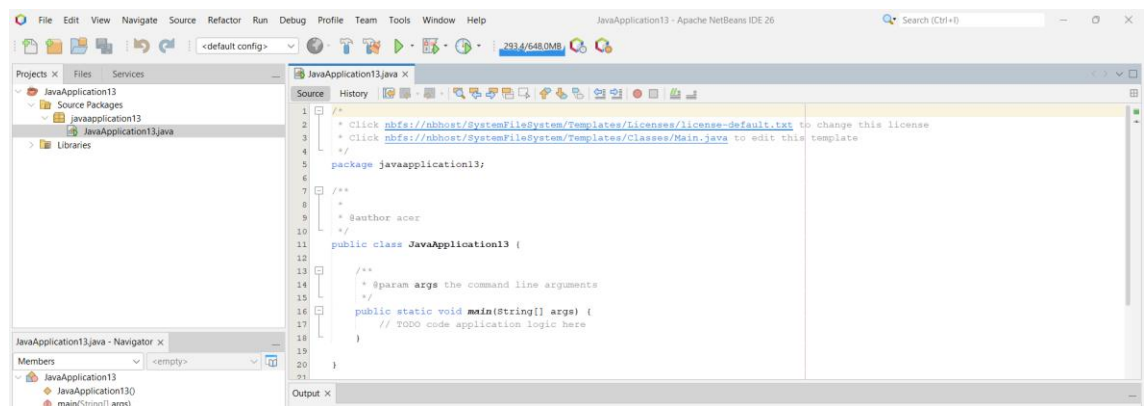
Project Folder:

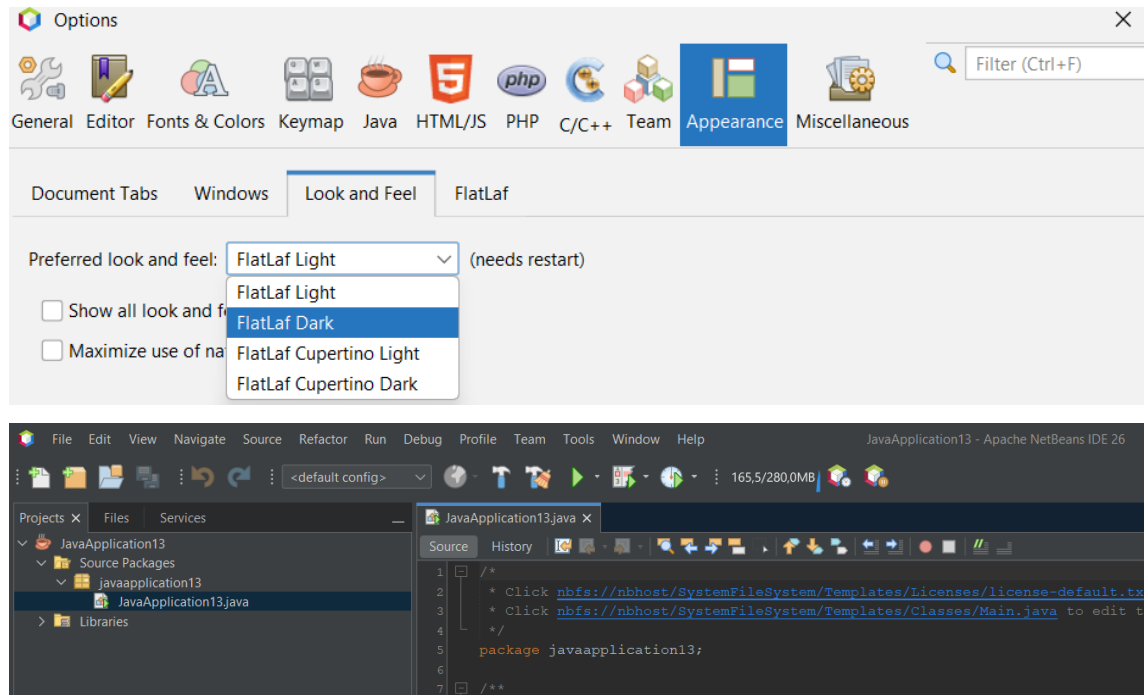
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

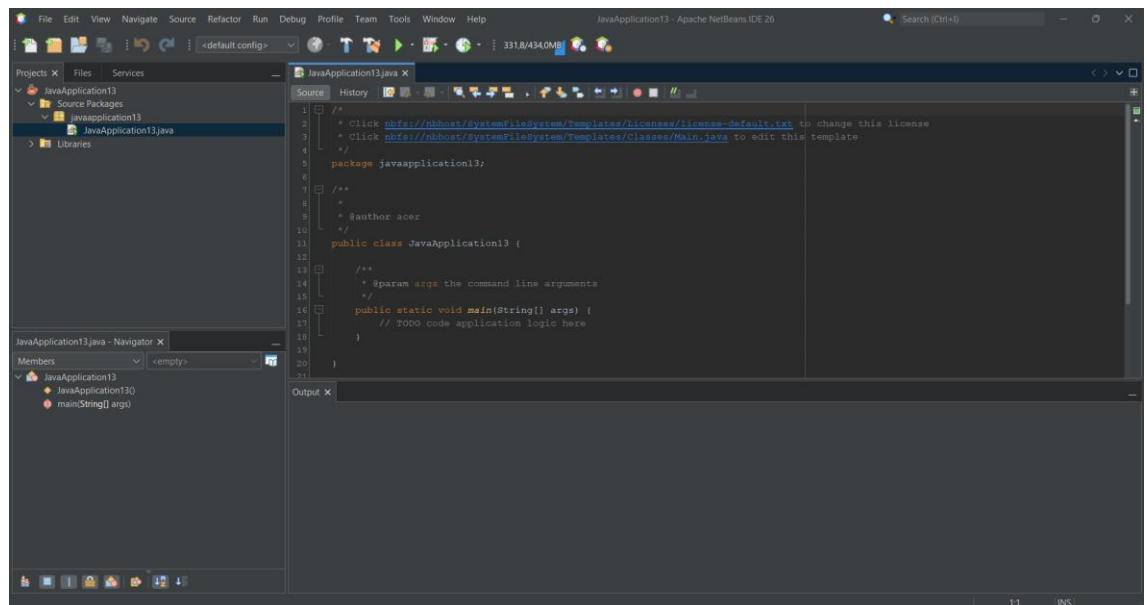
Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

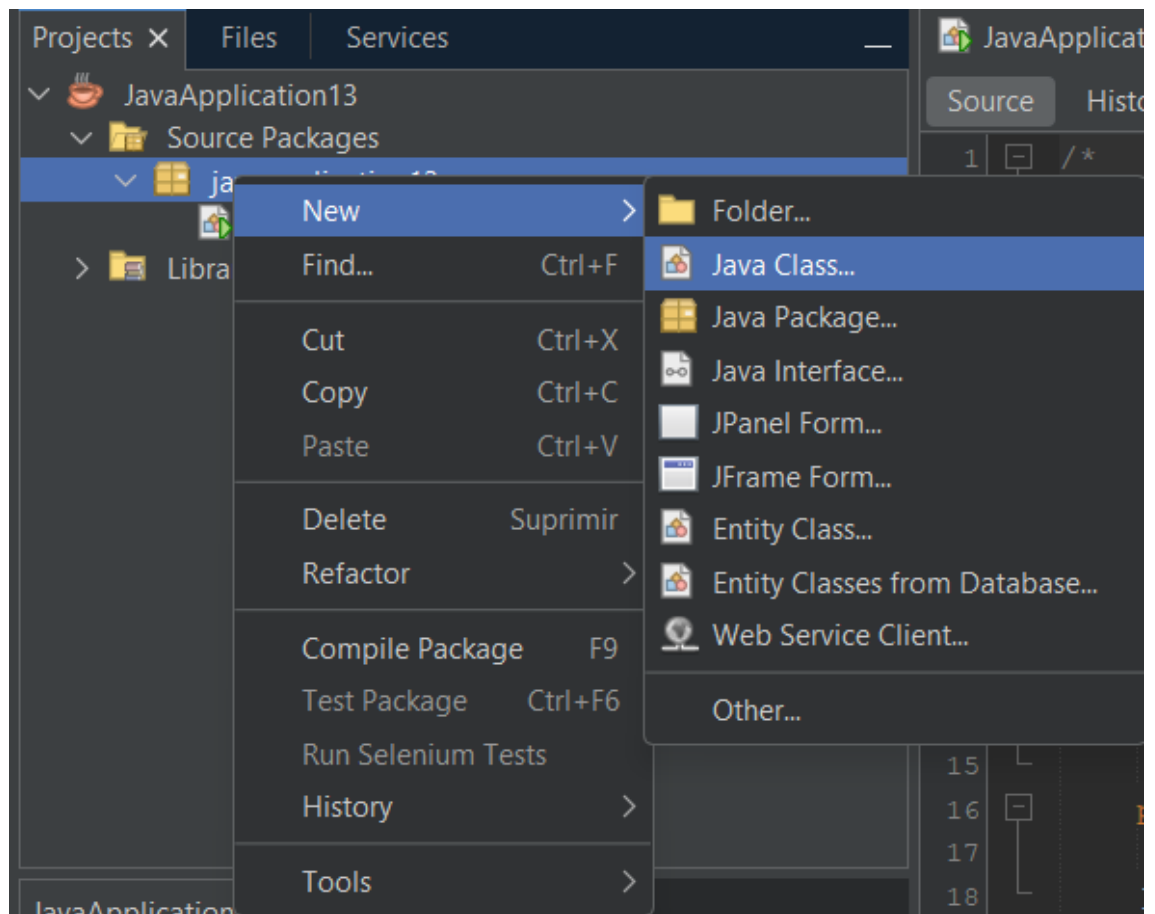




c. Toma una captura de pantalla del entorno configurado y agrégala a tu entrega.



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



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

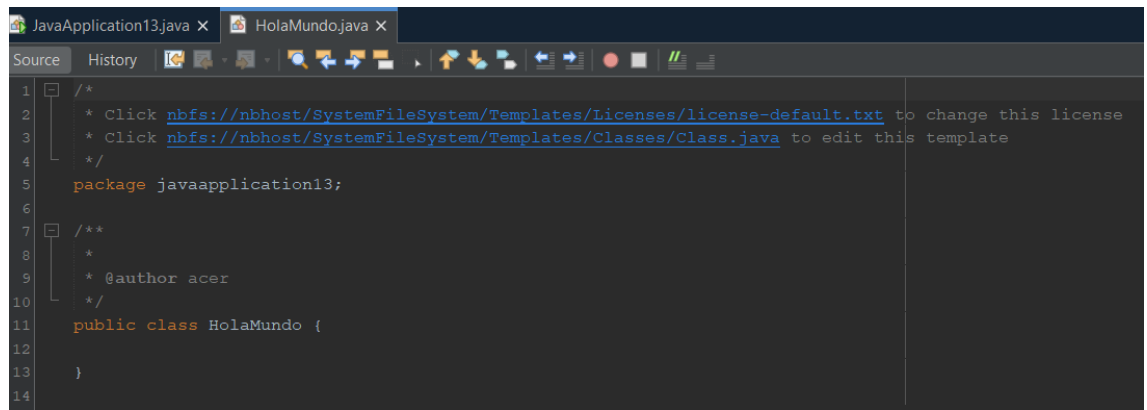
Location:

Package:

Created File:

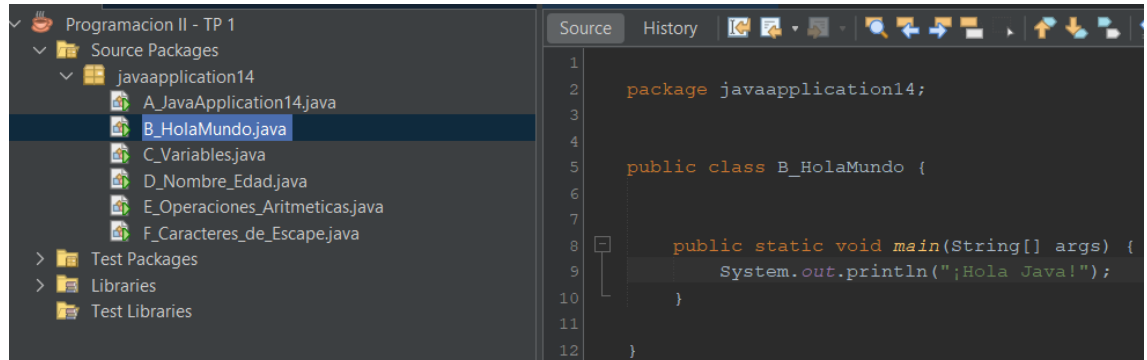
Superclass:

Interfaces:



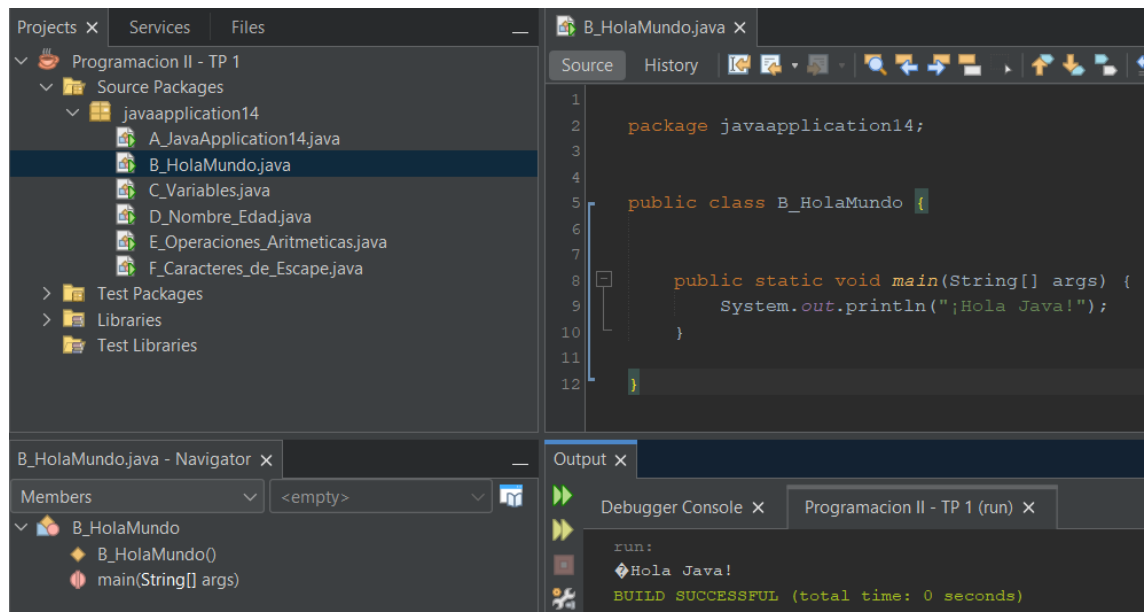
```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package javaapplication13;
6
7   /**
8    *
9    * @author acer
10   */
11   public class HolaMundo {
12
13   }
14
```

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



```
1
2   package javaapplication14;
3
4
5   public class B_HolaMundo {
6
7
8       public static void main(String[] args) {
9           System.out.println("¡Hola Java!");
10      }
11
12  }
```

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



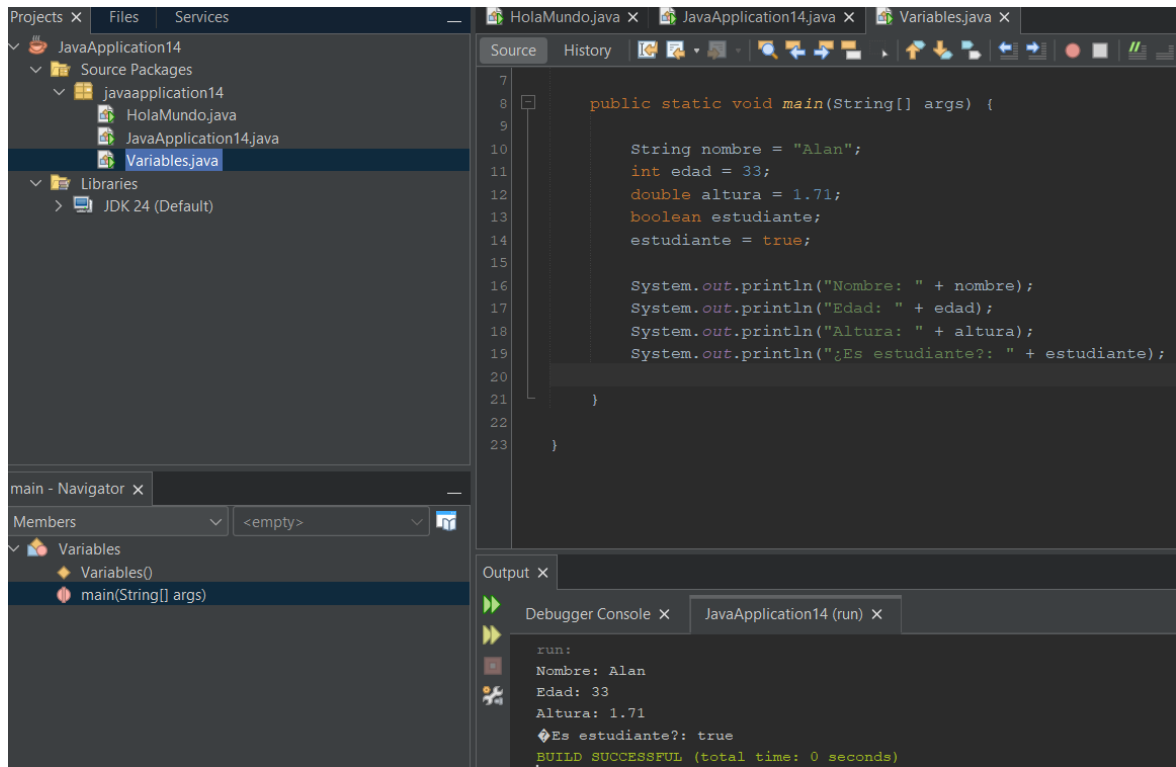
```
B_HolaMundo.java - Navigator
Members
B_HolaMundo
  B_HolaMundo()
  main(String[] args)

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

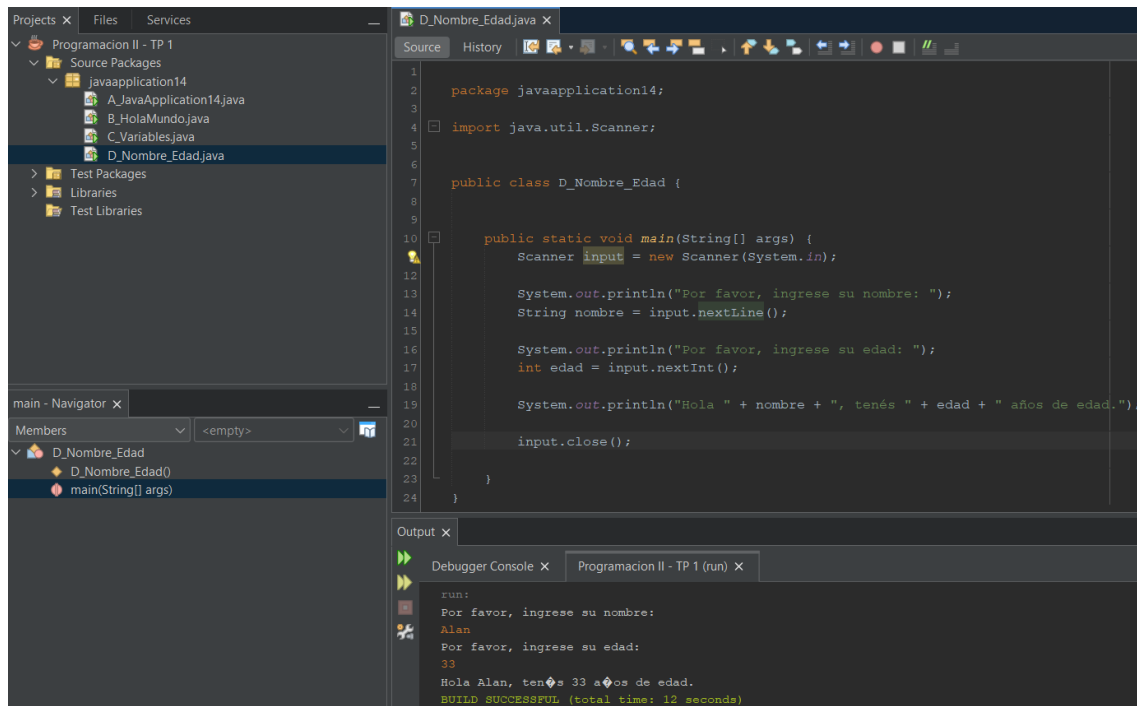
3. Crea un programa que declare las siguientes variables con valores asignados:

- a. String nombre
- b. int edad
- c. double altura
- d. boolean estudiante

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



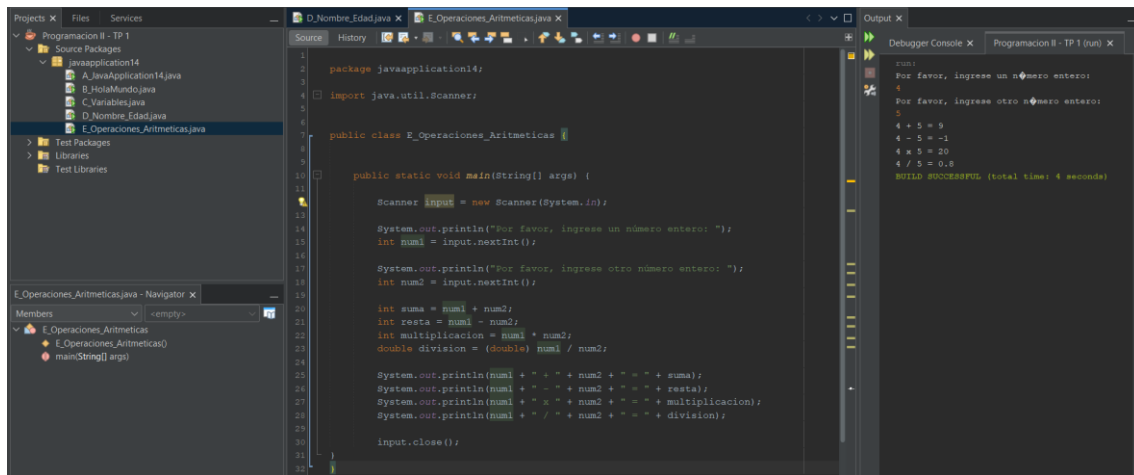
4. Escribe un programa que solicite al usuario ingresar su nombre y edad, y luego los muestre en pantalla. Usa **Scanner** para capturar los datos.



5. Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:

- Suma
- Resta
- Multipliación
- División

Muestra los resultados en la consola.



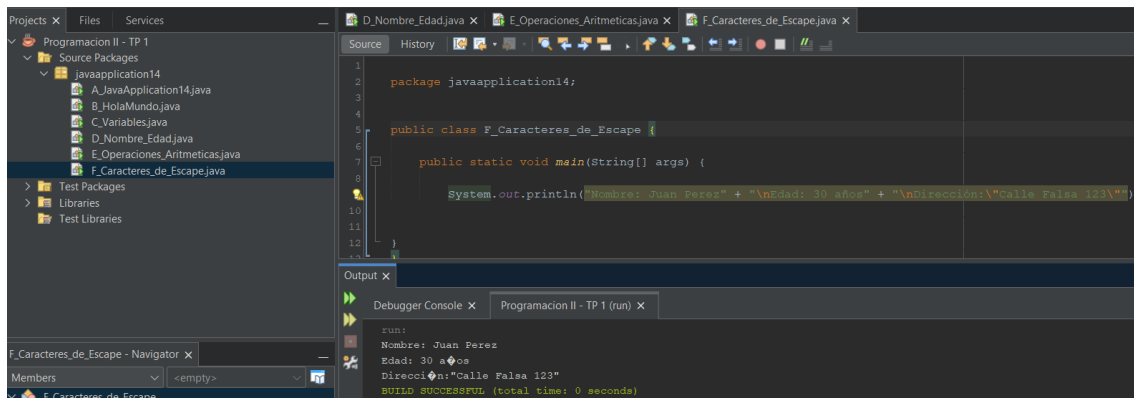
6. Escribe un programa que muestre el siguiente mensaje en consola:

Nombre: Juan Pérez

Edad: 30 años

Dirección: "Calle Falsa 123"

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



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

Rta: Instrucción (Se declara e inicializa la variable x).

x = x + 5; // Línea 2

Rta: Instrucción (Se asigna a x el valor dado por el resultado de x + 5, es decir, se actualiza el valor de x).

System.out.println(x); // Línea 3

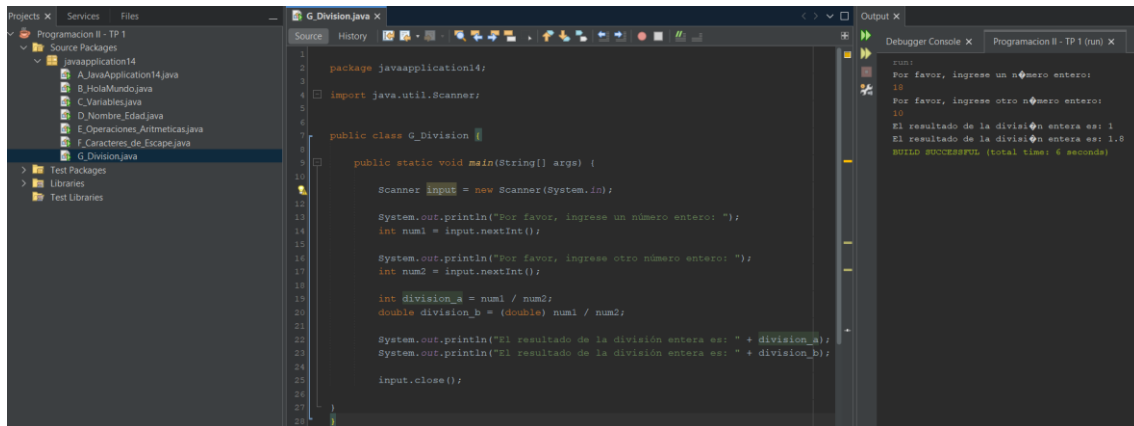
Rta: Instrucción (Se imprime un valor en la consola).

Expresión: Es cualquier combinación de valores, variables, operadores y métodos que se evalúa para producir un único valor. Una expresión por sí sola no realiza una acción, solo calcula un resultado.

Instrucción: Es una línea de código que realiza una acción. Una instrucción termina con un punto y coma (;). A menudo, una instrucción puede contener una o más expresiones.

8. Manejar conversiones de tipo y división en Java.

- Escribe un programa que divida dos números enteros ingresados por el usuario.
- Modifica el código para usar **double** en lugar de **int** y compara los resultados.



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);

    }

}
```

El error se debe a que la instrucción **scanner.nextInt()** no se puede utilizar con una variable String, por lo que la reemplacé por **scanner.nextLine()** para que el programa pudiera leer la cadena de caracteres ingresadas por el usuario (su nombre).

