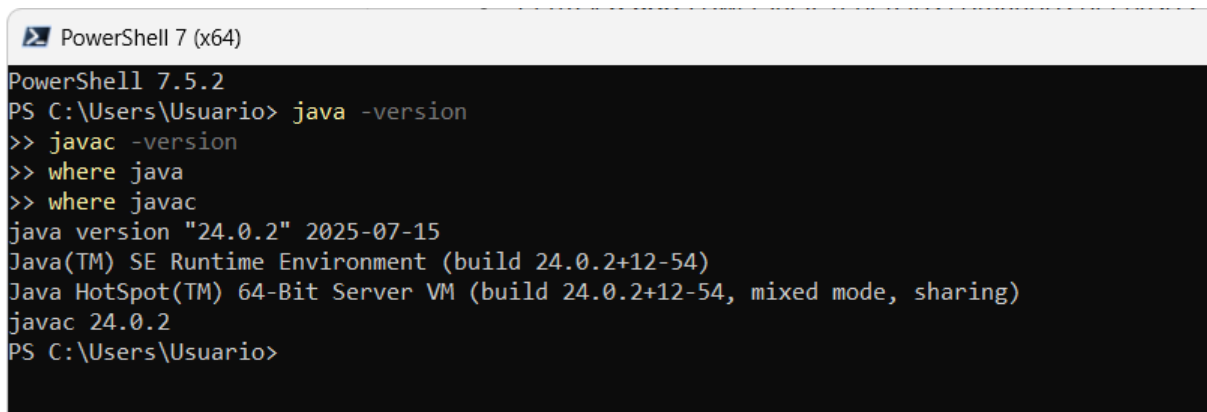
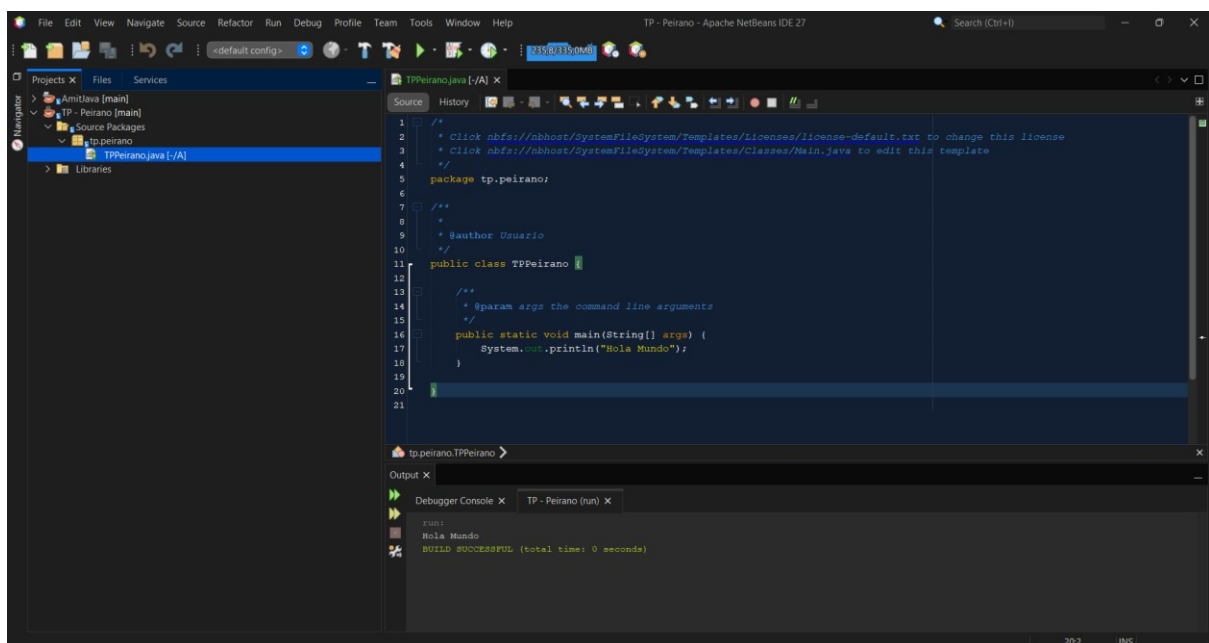


1. Verificar que tienes instalado Java JDK y NetBeans
 - a) Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal: `java -versión`

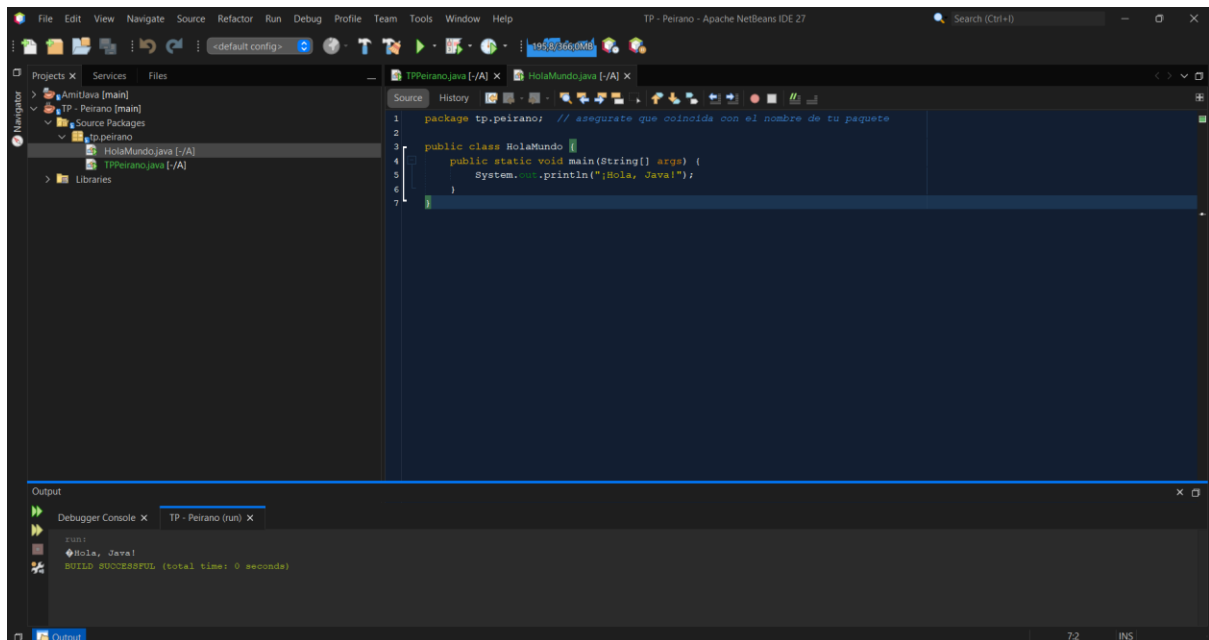


```
PowerShell 7 (x64)
PowerShell 7.5.2
PS C:\Users\Usuario> java -version
>> javac -version
>> where java
>> where javac
java version "24.0.2" 2025-07-15
Java(TM) SE Runtime Environment (build 24.0.2+12-54)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode, sharing)
javac 24.0.2
PS C:\Users\Usuario>
```

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

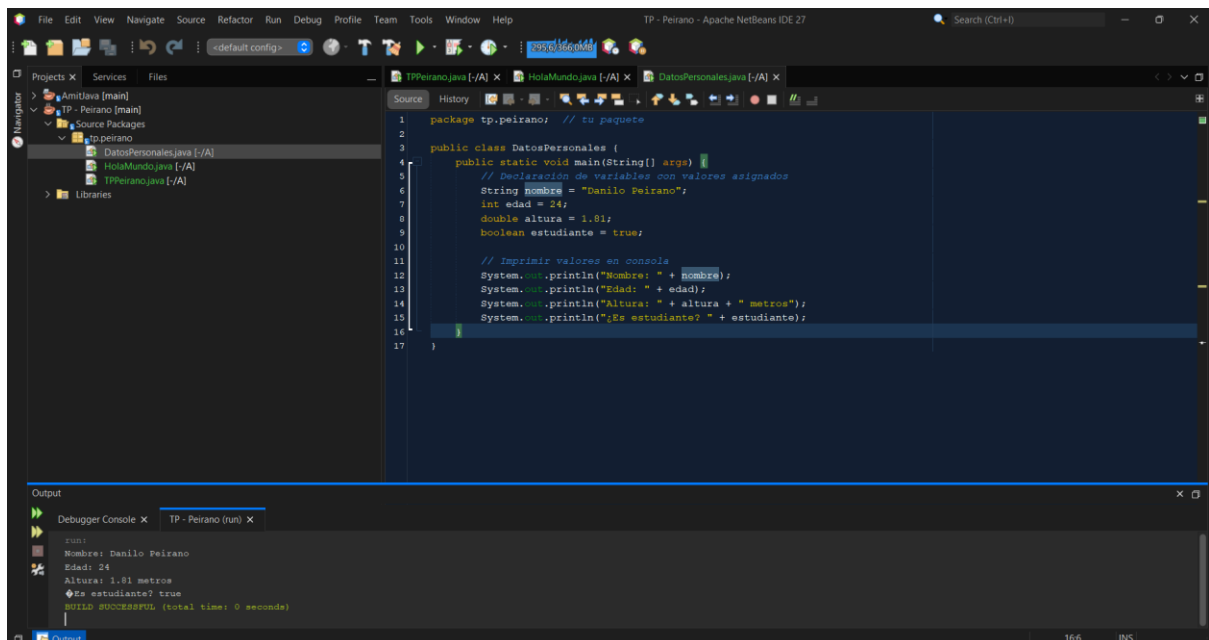


2. Escribir y ejecutar un programa básico en Java.
 - a Crea una clase llamada HolaMundo
 - b Escribe un programa que imprima el mensaje: ¡Hola, Java!
 - c Ejecuta el programa en NetBeans y adjunta una captura del resultado en la consola.

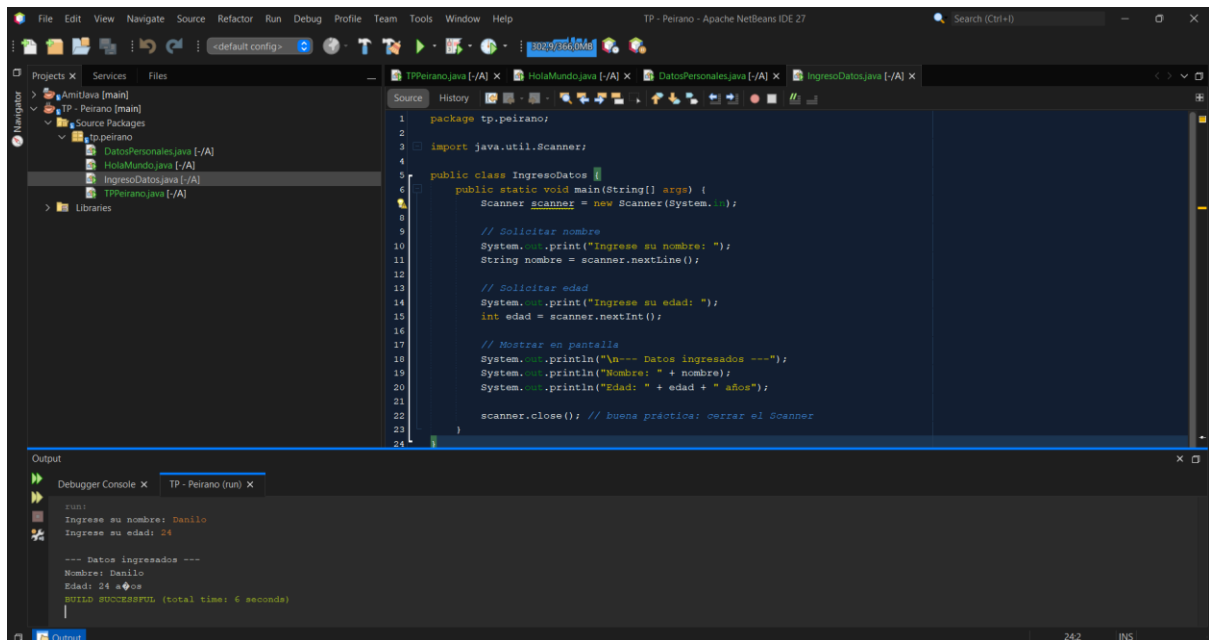


3. Crea un programa que declare las siguientes variables con valores asignados:
 - a) String nombre
 - b) int edad
 - c) double altura
 - d) boolean estudiante

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

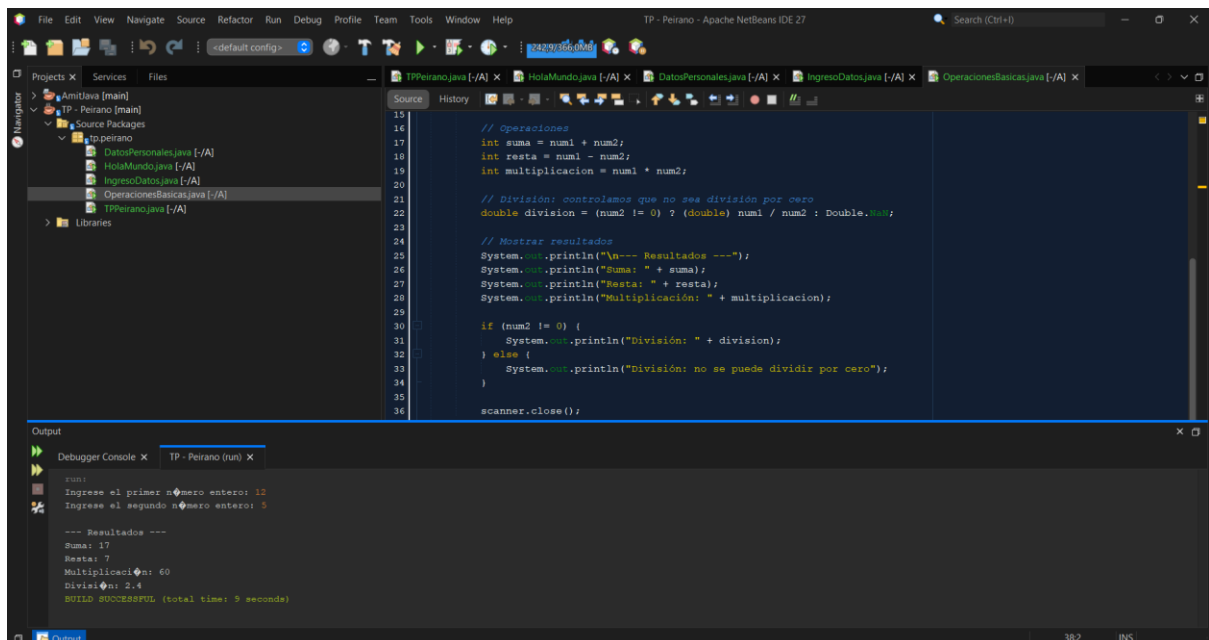


4. Escribe un programa que solicite al usuario ingresar su nombre y edad, y luego los muestre en pantalla. Usa Scanner para capturar los datos.



5. Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:
 - a. Suma
 - b. Resta
 - c. Multiplicación
 - d. División

Muestra los resultados en la consola.



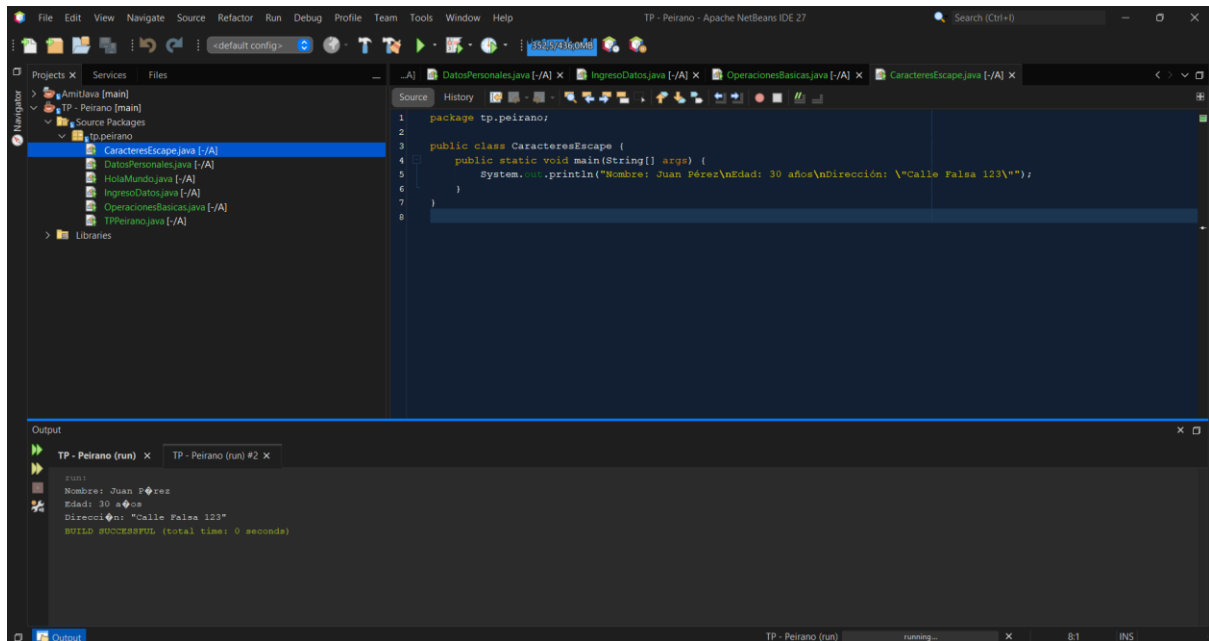
6. Escribe un programa que muestre el siguiente mensaje en consola:

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

x = x + 5; // Línea 2

System.out.println(x); // Línea 3

Expresiones (producen un valor):

- 10 (valor literal).
- x + 5 (operación aritmética que devuelve un valor).
- x (acceso a la variable).

Instrucciones (acciones que ejecuta el programa):

- int x = 10; (declaración e inicialización de variable).
- x = x + 5; (asignación del resultado de la expresión a la variable).
- System.out.println(x); (instrucción que imprime en pantalla el valor de x).

Diferencia:

En Java, **una expresión** es cualquier fragmento de código que se puede evaluar y produce un **valor** (por ejemplo $x + 5$). En cambio, **una instrucción** es una orden completa que el programa ejecuta (como declarar una variable, asignar un valor o imprimir en consola). Las instrucciones pueden contener expresiones en su interior, pero no al revés.

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.

The screenshot shows the Apache NetBeans IDE with a project named 'TP - Peirano'. The 'Sources' window on the left lists several Java files, including 'DivisionEnteros.java'. The 'Main' window displays the code for 'DivisionEnteros.java', which uses the Scanner class to read two integers from the user and performs integer division. The 'Output' window at the bottom shows the program's execution, where the user enters 10 and 6, resulting in a division of 1. The status bar at the bottom indicates the program is running successfully.

```
package tp.peirano;

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();

        if (num2 != 0) {
            int division = num1 / num2; // división entera
            System.out.println("Resultado de la división (int): " + division);
        } else {
            System.out.println("No se puede dividir por cero.");
        }

        scanner.close();
    }
}
```

Output:

```
TP - Peirano (run) X TP - Peirano (run) #2 X
run:
Ingrese el primer número entero: 10
Ingrese el segundo número entero: 6
Resultado de la división (int): 1
BUILD SUCCESSFUL (total time: 8 seconds)
```

The screenshot shows the Apache NetBeans IDE with the same project. The 'Sources' window now lists 'DivisionDouble.java' as the active file. The 'Main' window displays the code for 'DivisionDouble.java', which is identical to the previous one but uses double-precision floating-point numbers instead of integers. The 'Output' window shows the program's execution, where the user enters 10.0 and 3.0, resulting in a division of 3.3333333333333335. The status bar at the bottom indicates the program is running successfully.

```
package tp.peirano;

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): ");
        double num1 = scanner.nextDouble();

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

        if (num2 != 0) {
            double division = num1 / num2; // división con decimales
            System.out.println("Resultado de la división (double): " + division);
        } else {
            System.out.println("No se puede dividir por cero.");
        }

        scanner.close();
    }
}
```

Output:

```
TP - Peirano (run) X TP - Peirano (run) #2 X
run:
Ingrese el primer número (double): 10
Ingrese el segundo número (double): 3
Resultado de la división (double): 3.3333333333333335
BUILD SUCCESSFUL (total time: 10 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);

    }

}
```

Problema detectado:

- Se intenta guardar el **nombre** en una variable String usando scanner.nextInt().
- El método nextInt() sirve para leer **números enteros**, no texto.
- Por eso el compilador marca error de tipo: *“incompatible types: int cannot be converted to String”*.

Corrección:

Debemos usar scanner.nextLine() para leer una **cadena de texto (String)**.

```
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.nextLine(); // CORREGIDO → nextLine() lee texto

        System.out.println("Hola, " + nombre);

    }

}
```

```

        scanner.close();
    }
}

```

**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);

    }

}

```

Pasó a paso

Línea de código	Acción realizada	Valor de las variables
Int a = 5;	Se declara a y se asigna 5	a = 5, b = (sin valor), resultado = (sin valor)
Int b = 2;	Se declara b y se asigna 2	a = 5, b = 2, resultado = (sin valor)
Int resultado = a / b;	Se divide 5 / 2 (división entera)	a = 5, b = 2, resultado = 2
System.out.println(...)	Imprime el valor de resultado	Consola: Resultado: 2

El valor de `resultado` es **2**, no 2.5, porque tanto `a` como `b` son variables de tipo `int`. En Java, cuando se dividen dos enteros, el resultado es **una división entera**, que descarta la parte decimal (trunca). Para obtener decimales debería usarse `double`
`resultado = (double) a / b;`