

PROGRAMACIÓN II

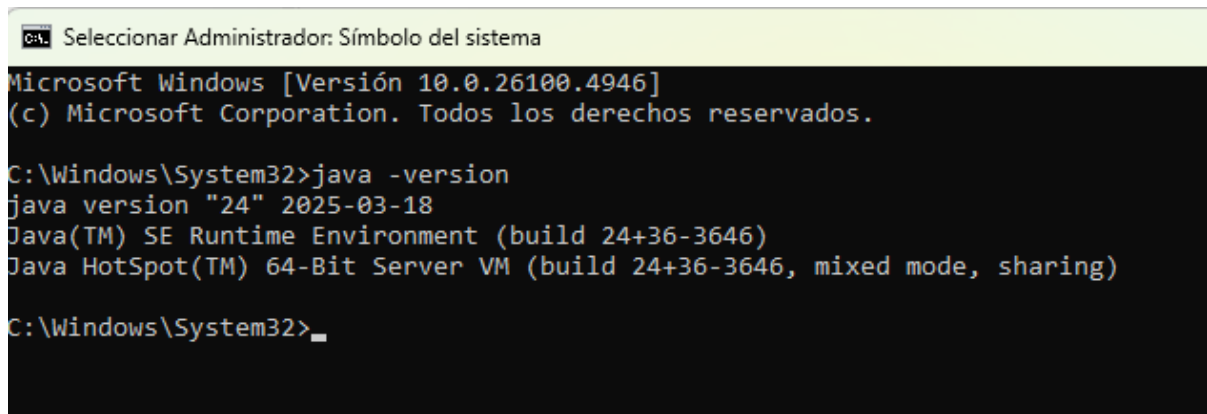
Trabajo Práctico 1: Introducción a Java

Alumno Santiago Raúl Salinas

CASO PRACTICO

1)

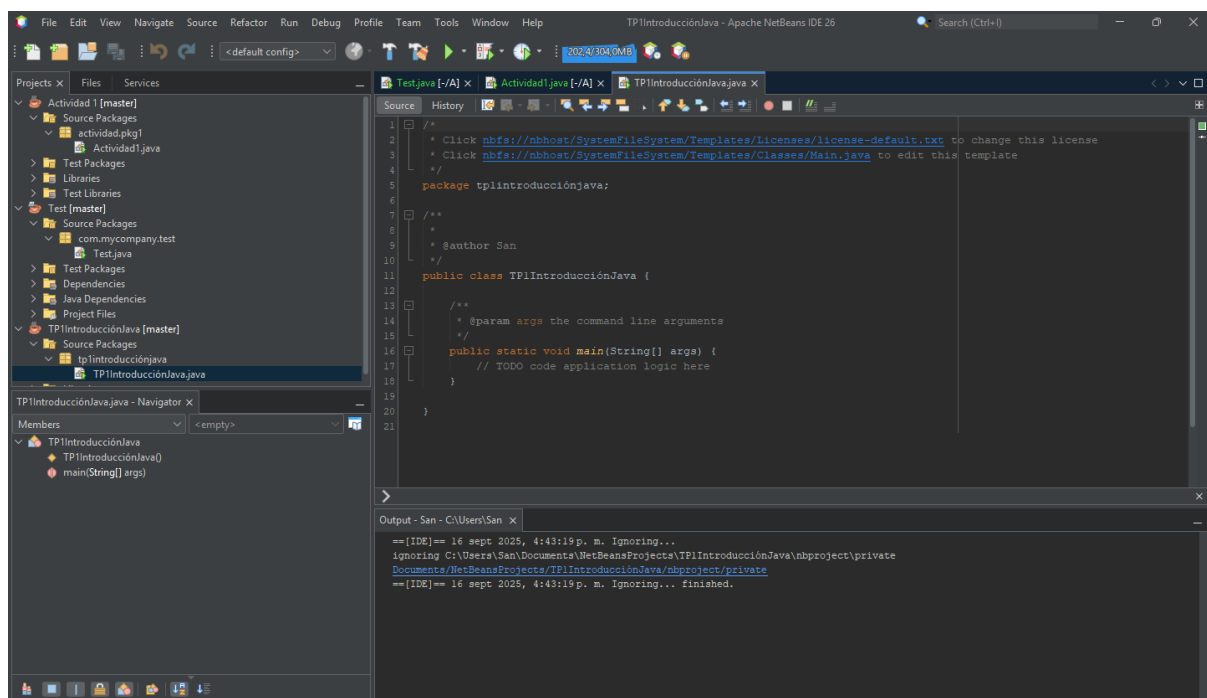
a.



```
C:\Windows\System32>java -version
java version "24" 2025-03-18
Java(TM) SE Runtime Environment (build 24+36-3646)
Java HotSpot(TM) 64-Bit Server VM (build 24+36-3646, mixed mode, sharing)

C:\Windows\System32>
```

b.

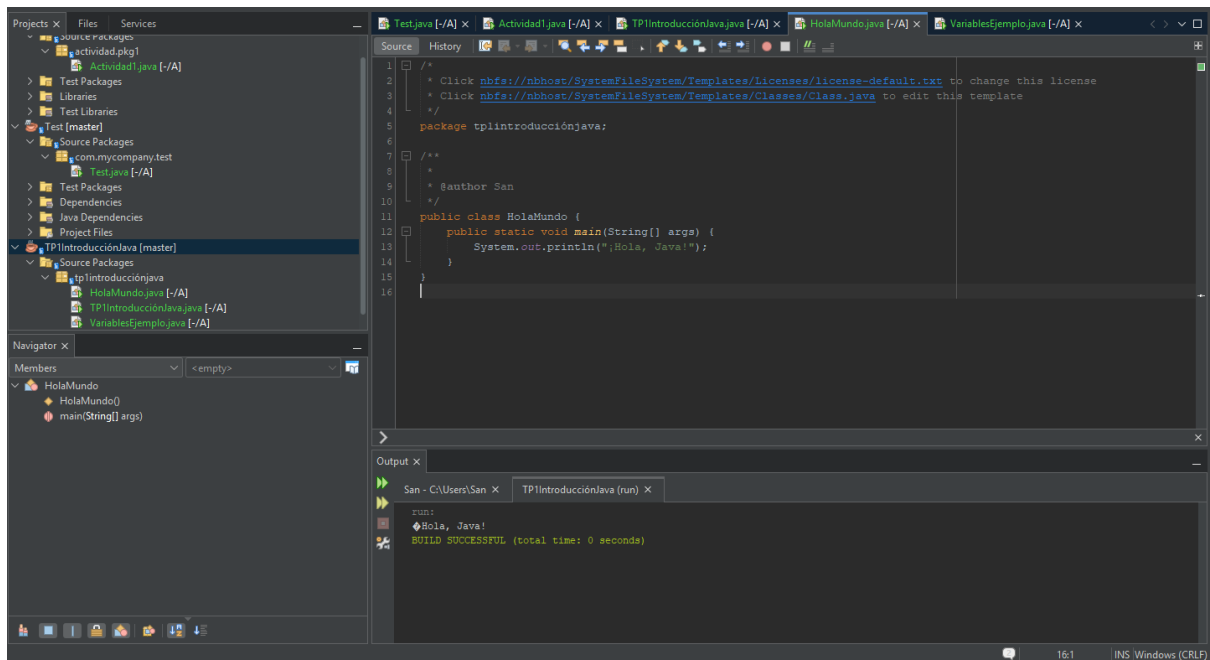


```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package tp1introducciónjava;
6
7  /**
8   *
9   * @author San
10  */
11  public class TP1IntroducciónJava {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18      }
19
20  }
21
```

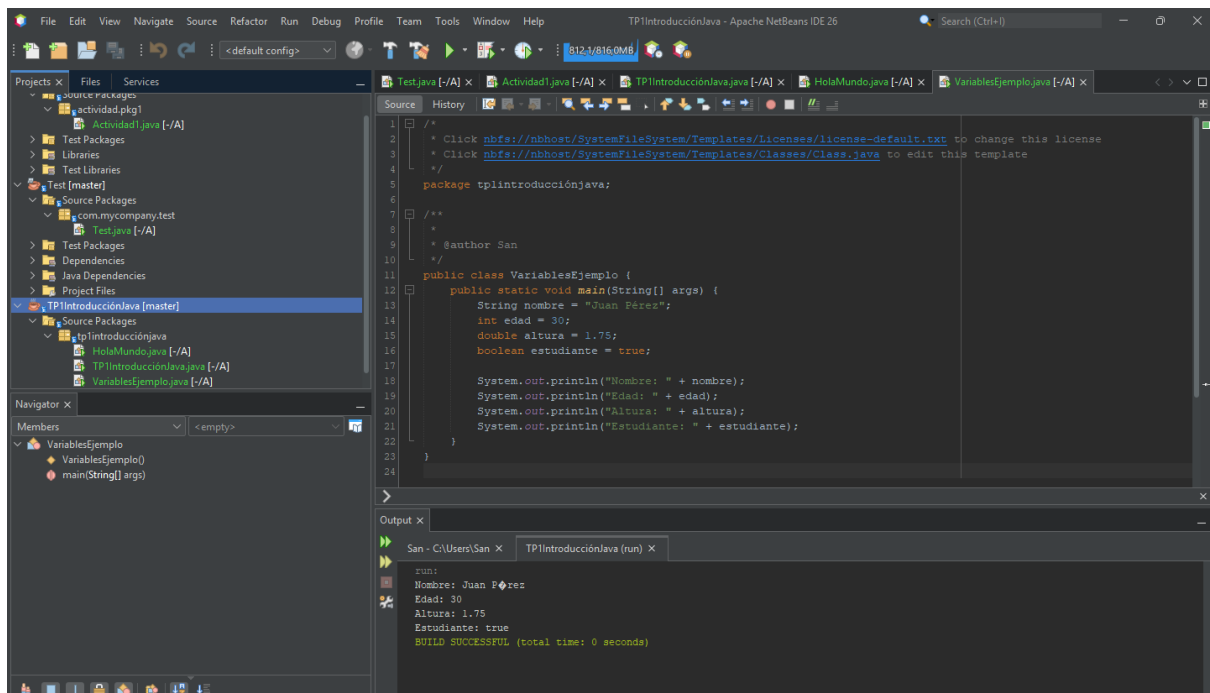
Output - San - C:\Users\San X

```
==[IDE]== 16 sept 2025, 4:43:19p. m. Ignoring...
Ignoring C:\Users\San\Documents\NetBeansProjects\TP1IntroducciónJava\nbproject\private
Documents\NetBeansProjects\TP1IntroducciónJava\nbproject\private
==[IDE]== 16 sept 2025, 4:43:19p. m. Ignoring... finished.
```

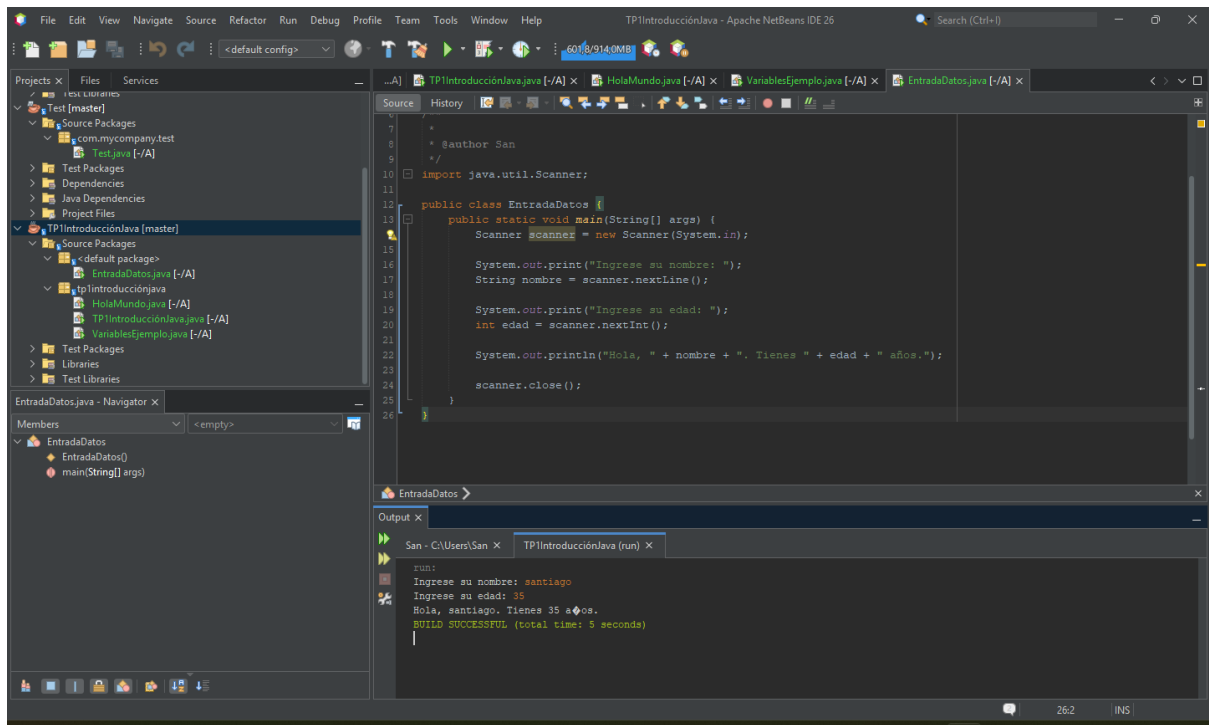
2)



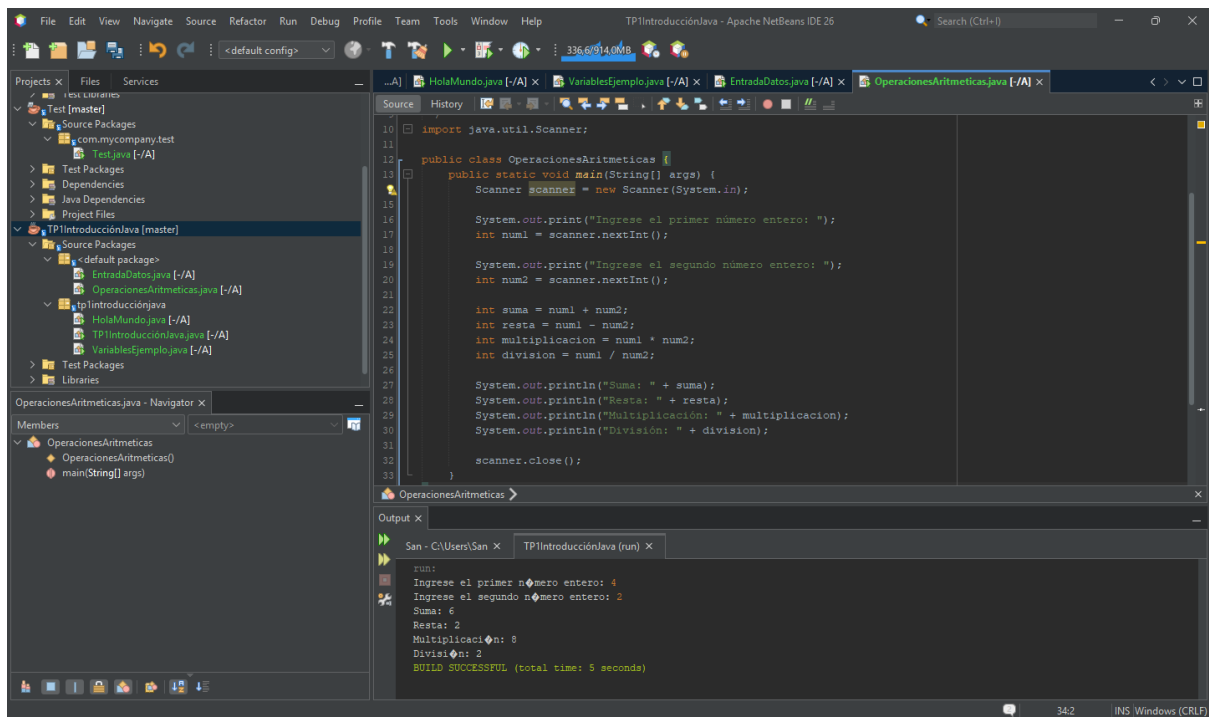
3)



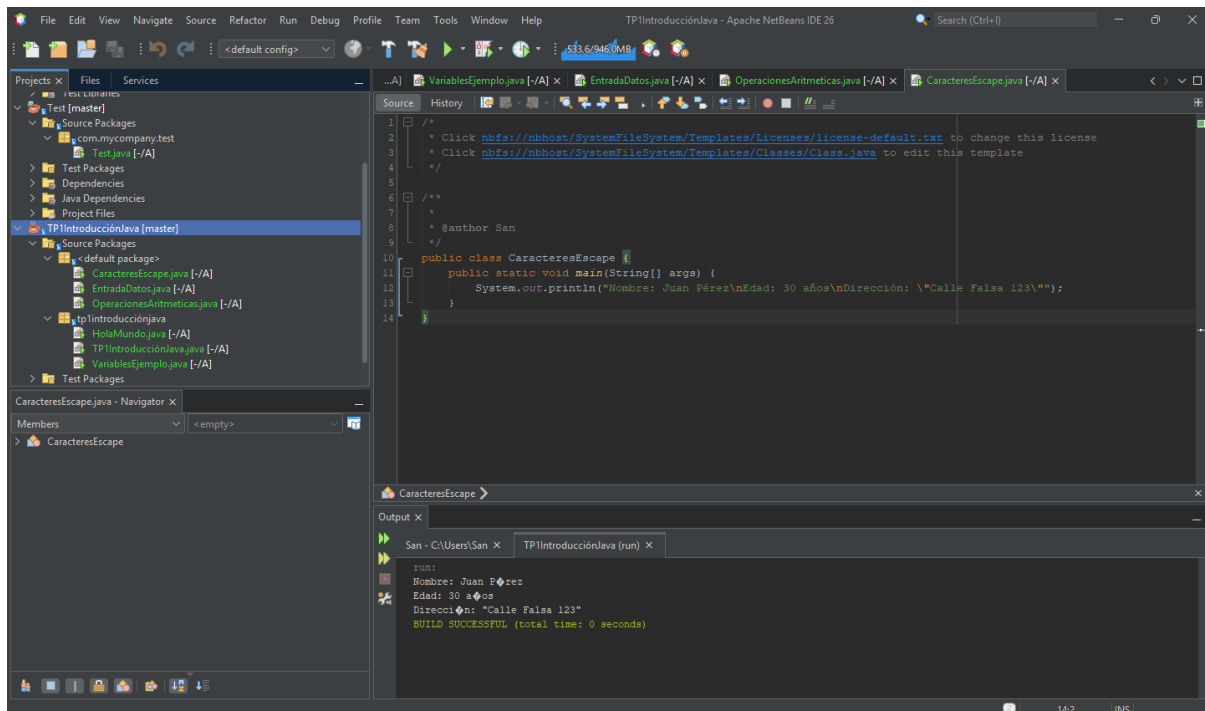
4)



5)



6)



7)

Código dado:

`int x = 10; // Línea 1`

`x = x + 5; // Línea 2`

`System.out.println(x); // Línea 3`

Expresiones: Son partes que se evalúan a un valor.

- En Línea 1: "10" (valor literal).
- En Línea 2: "x + 5" (se evalúa a 15).

Instrucciones: Son líneas completas que realizan acciones y terminan en ";".

- Línea 1: Declaración e inicialización de variable (instrucción).
- Línea 2: Asignación (instrucción, contiene una expresión).
- Línea 3: Llamada a método para imprimir (instrucción).

Explicación breve: Una expresión calcula un valor (como una operación matemática) pero no hace nada por sí sola. Una instrucción es una unidad ejecutable que realiza una acción, como asignar un valor o imprimir, y puede incluir expresiones. La diferencia es que las expresiones producen resultados temporales, mientras que las instrucciones cambian el estado del programa.

8)

a.

The screenshot shows the NetBeans IDE with a project named 'TP1IntroducciónJava'. The 'Ejercicio8.java' file is open, containing the following code:

```
5
6  /**
7   *
8   * @author San
9   */
10 import java.util.Scanner;
11
12 public class Ejercicio8 {
13     public static void main(String[] args) {
14         Scanner scanner = new Scanner(System.in);
15
16         System.out.print("Ingrese el primer número entero: ");
17         int num1 = scanner.nextInt();
18
19         System.out.print("Ingrese el segundo número entero: ");
20         int num2 = scanner.nextInt();
21
22         int resultado = num1 + num2;
23         System.out.println("Resultado (entero): " + resultado);
24
25         scanner.close();
26     }
27 }
```

The output window shows the following execution results:

```
run:
Ingrese el primer número entero: 5
Ingrese el segundo número entero: 2
Resultado (entero): 7
BUILD SUCCESSFUL (total time: 4 seconds)
```

b.

The screenshot shows the NetBeans IDE with the same project. The 'Ejercicio8.java' file is open, containing the following code:

```
5
6  /**
7   *
8   * @author San
9   */
10 import java.util.Scanner;
11
12 public class Ejercicio8 {
13     public static void main(String[] args) {
14         Scanner scanner = new Scanner(System.in);
15
16         System.out.print("Ingrese el primer número entero: ");
17         double num1 = scanner.nextDouble(); // Usa double desde el inicio
18
19         System.out.print("Ingrese el segundo número entero: ");
20         double num2 = scanner.nextDouble();
21
22         double resultado = num1 + num2;
23         System.out.println("Resultado (decimal): " + resultado);
24
25         scanner.close();
26     }
27 }
```

The output window shows the following execution results:

```
run:
Ingrese el primer número entero: 5
Ingrese el segundo número entero: 2
Resultado (decimal): 7.0
BUILD SUCCESSFUL (total time: 3 seconds)
```

9)

Codigo

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: nextInt() espera un entero, pero el nombre es texto (String). Causa excepción si
 ingresa texto.

Codigo

corregido:

```
import java.util.Scanner;

public class ErrorEjemploCorregido {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingresa tu nombre: ");
        String nombre = scanner.nextLine(); // Cambiado a nextLine() para leer texto
        System.out.println("Hola, " + nombre);
        scanner.close();
    }
}
```

Explicación: Cambié nextInt() a nextLine() porque el input es una cadena de texto, no un número. Ahora lee correctamente cualquier nombre.

10)

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

Tabla de Prueba de Escritorio

Línea	a	b	resultado	Salida
int a = 5;	5	-	-	-
int b = 2;	5	2	-	-
int resultado = a / b;	5	2	2	-
System.out.println("Resultado: " + resultado);	5	2	2	Resultado: 2

Valor de resultado y por qué: 2. Porque es división entera entre ints ($5 / 2 = 2.5$, pero Java trunca la parte decimal a 2). Para obtener 2.5, usa double o casting.