

Trabajo Práctico n° 1

Introducción a Java

Comision 16

Macarena Marinoni <*marinonimacarena@gmail.com*>

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional

Programación II

Profesor

Cortez, Alberto

Tutor

Bianchi, Neyén

16/08/2025

ÍNDICE

Objetivo general.....	3
Marco teórico.....	4
Caso práctico.....	5
Verificar instalación de Java JDK y NetBeans.....	6
Programa básico en Java.....	7
Declaración variables.....	8
Nombre y Edad.....	9
Enteros y operaciones.....	10
Datos Usuario.....	11
Expresiones e Instrucciones.....	12
Manejar conversiones de tipo y división en Java.....	12
Corrección de código.....	15
Prueba de escritorio.....	16

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.

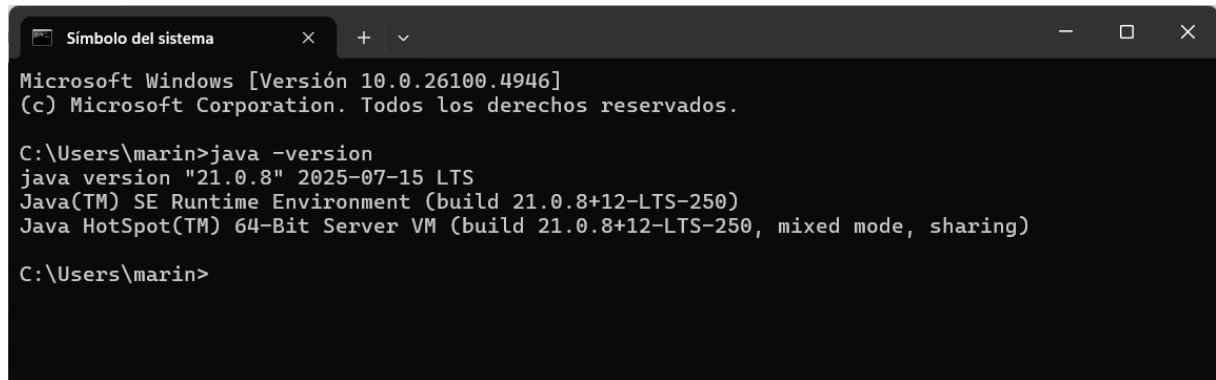
Caso práctico

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.

Verificar instalación de Java JDK y NetBeans

Para confirmar instalación de Java JDK se ejecuta el comando `java -version` en terminal obteniéndose lo siguiente:



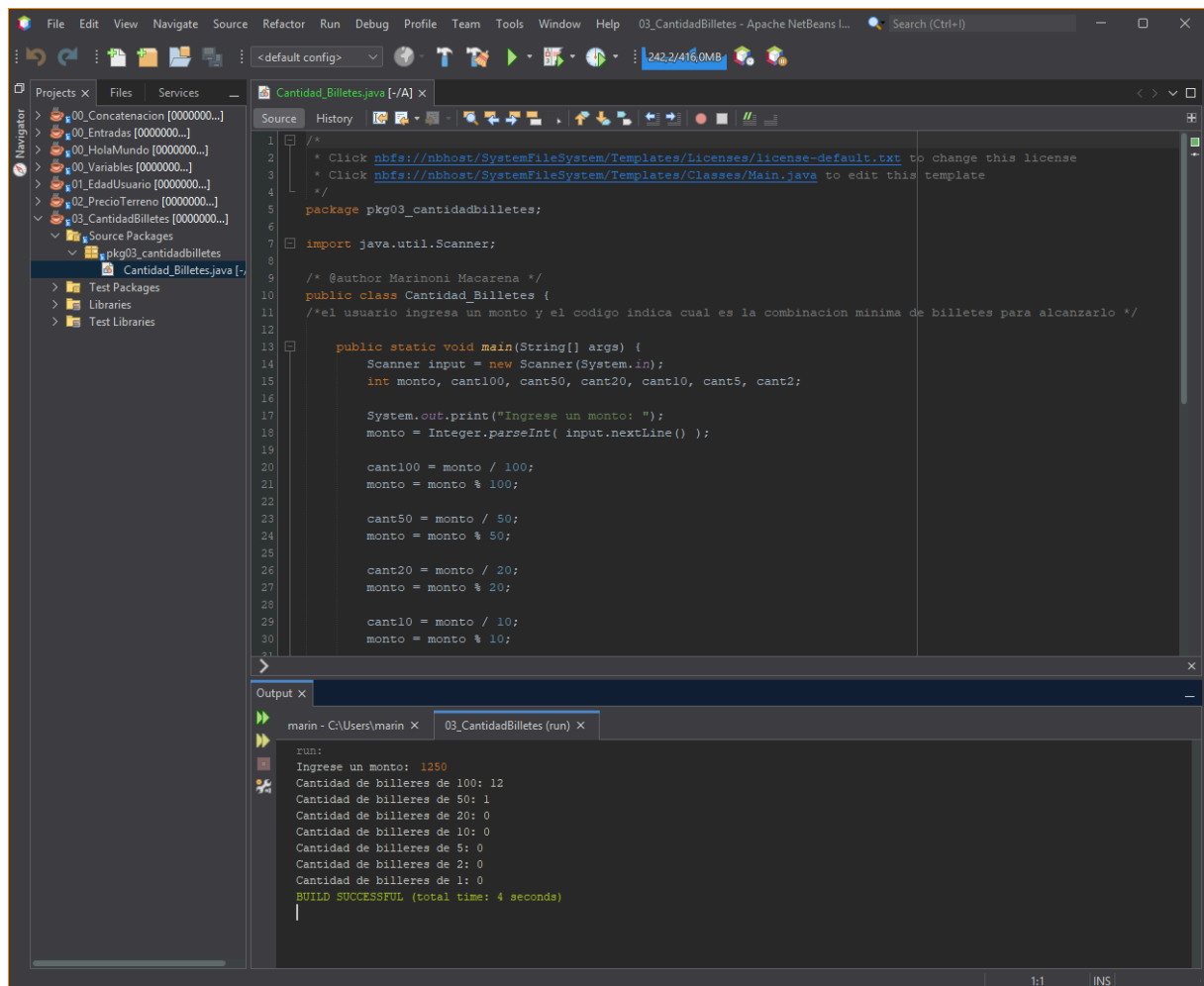
```

Microsoft Windows [Versión 10.0.26100.4946]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\marin>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\marin>
  
```

Para confirmar la instalación de Abre NetBeans, se procede a crear un nuevo proyecto y configurar el modo oscuro, obteniéndose la siguiente visualización a final:



```

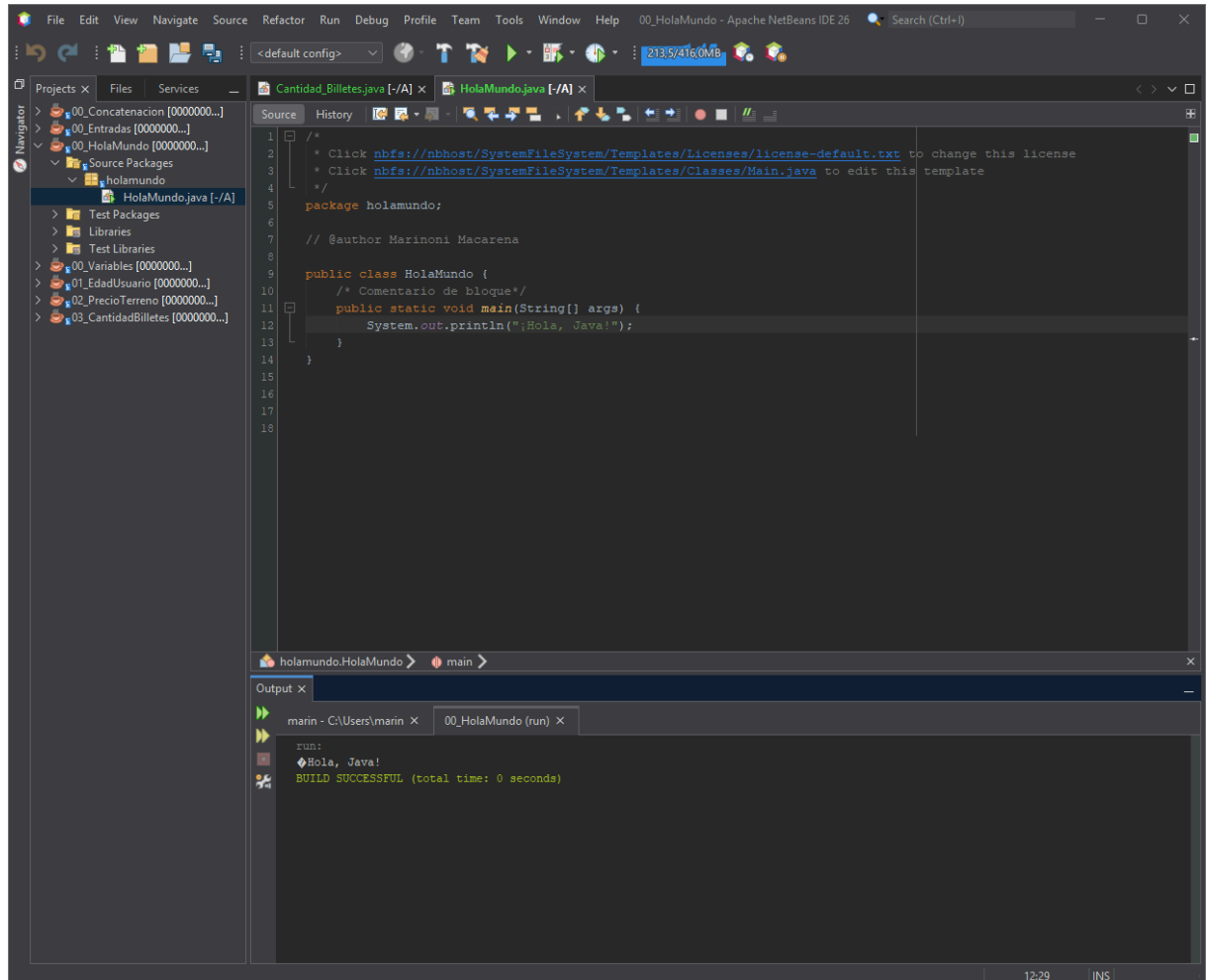
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package pkg03_cantidadbilletes;
6
7  import java.util.Scanner;
8
9  /* @author Marloni Macarena */
10 public class Cantidad_Billetes {
11     /*el usuario ingresa un monto y el codigo indica cual es la combinacion minima de billetes para alcanzarlo */
12
13     public static void main(String[] args) {
14         Scanner input = new Scanner(System.in);
15         int monto, cant100, cant50, cant20, cant10, cant5, cant2;
16
17         System.out.print("Ingrese un monto: ");
18         monto = Integer.parseInt(input.nextLine());
19
20         cant100 = monto / 100;
21         monto = monto % 100;
22
23         cant50 = monto / 50;
24         monto = monto % 50;
25
26         cant20 = monto / 20;
27         monto = monto % 20;
28
29         cant10 = monto / 10;
30         monto = monto % 10;
31     }
32 }
  
```

```

run:
Ingrese un monto: 1250
Cantidad de billetes de 100: 12
Cantidad de billetes de 50: 1
Cantidad de billetes de 20: 0
Cantidad de billetes de 10: 0
Cantidad de billetes de 5: 0
Cantidad de billetes de 2: 0
Cantidad de billetes de 1: 0
BUILD SUCCESSFUL (total time: 4 seconds)
  
```

Programa básico en Java.

Se crea una clase llamada HolaMundo, se escribe un programa que imprima el mensaje: *¡Hola, Java!* y se ejecuta este programa en NetBeans obteniéndose el siguiente resultado en consola:

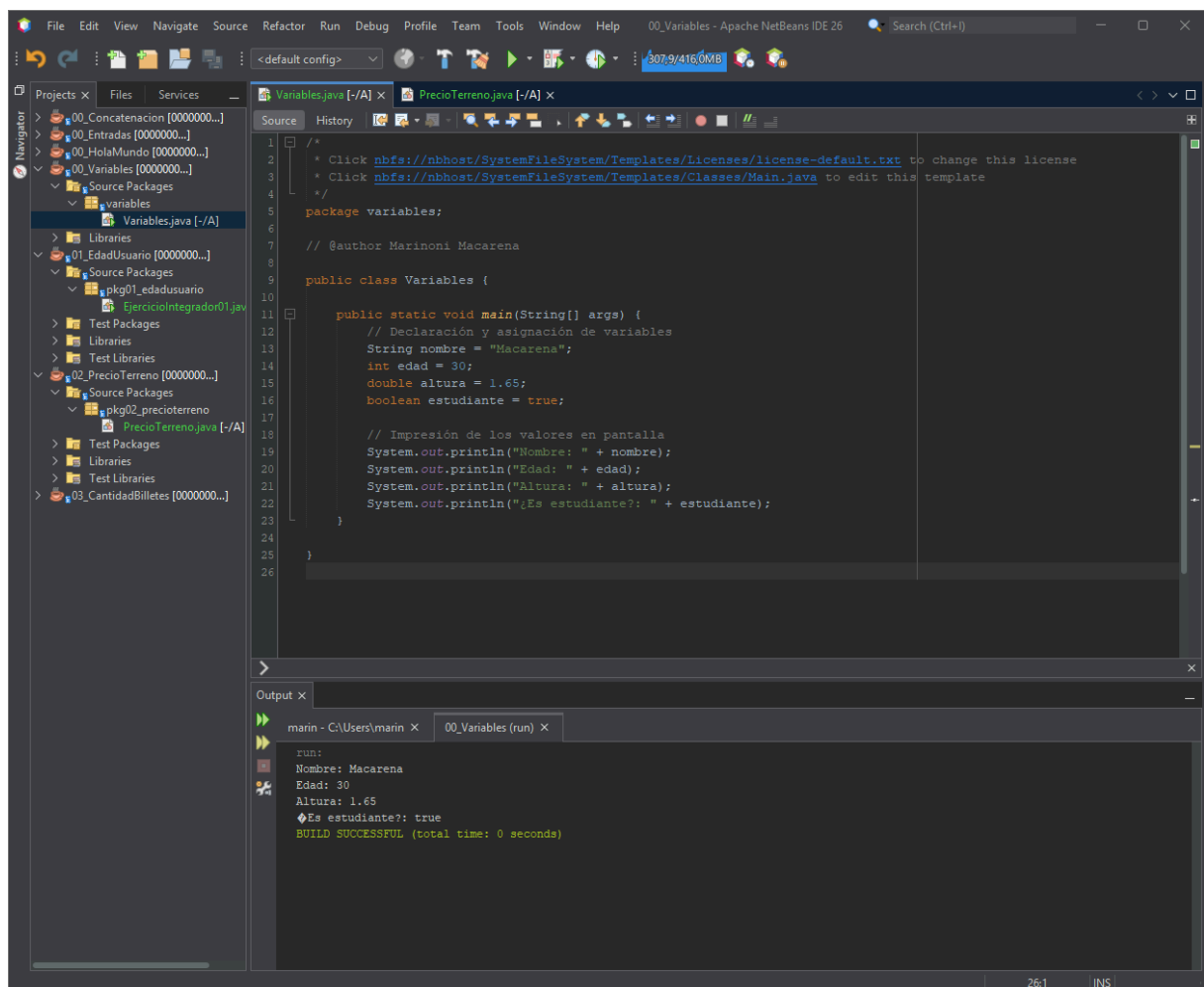


Declaración variables

Se crear un programa que declara las siguientes variables con valores asignados:

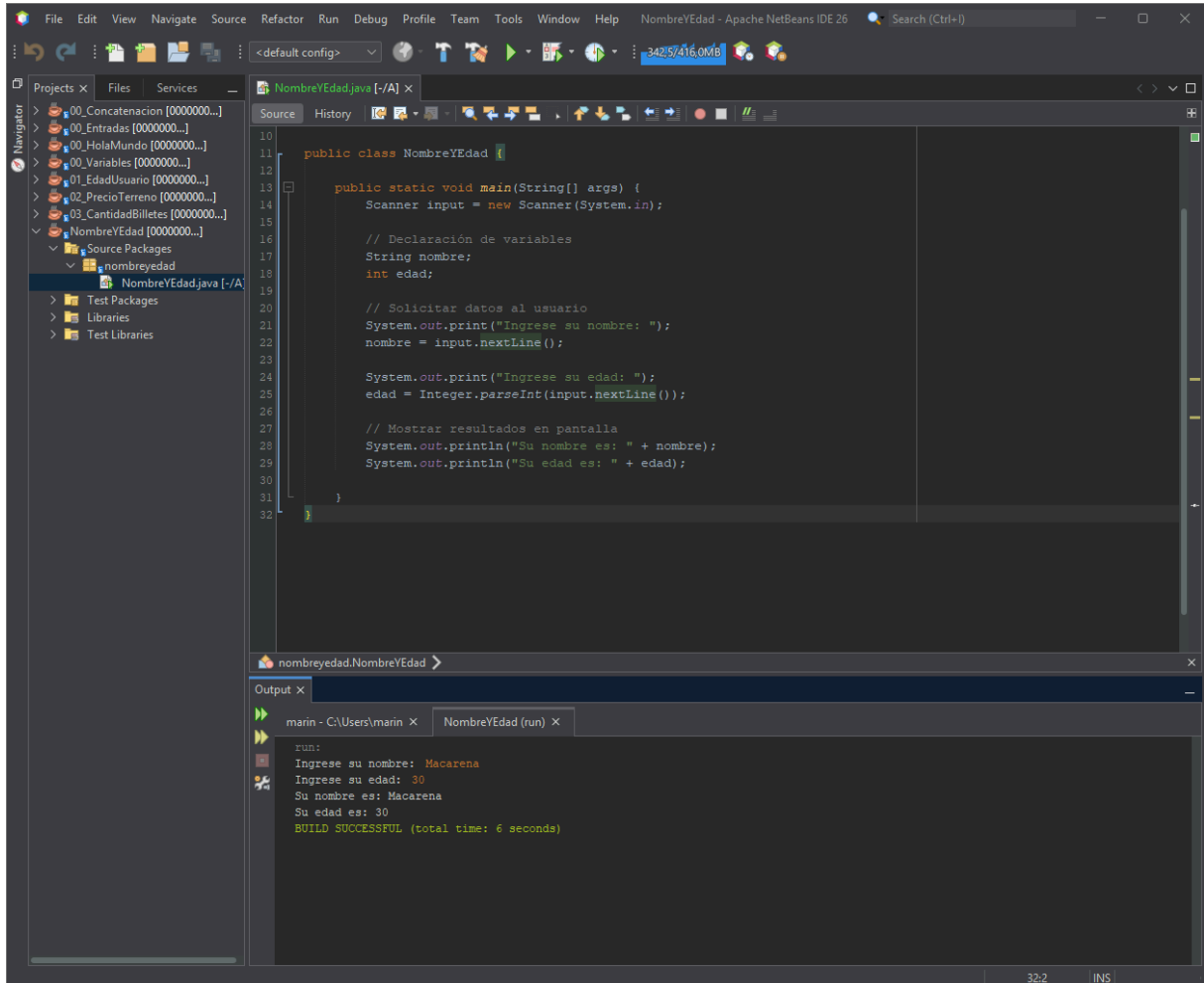
1. String nombre
2. int edad
3. double altura
4. boolean estudiante

Además, se imprimen los valores en pantalla usando `System.out.println()`.



Nombre y Edad

Se escribe un programa que solicite al usuario ingresar su nombre y edad, y luego los muestre en pantalla. Se usa Scanner para capturar los datos.



The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the source code for `NombreYEdad.java`. The code is as follows:

```
10 public class NombreYEdad {
11
12     public static void main(String[] args) {
13         Scanner input = new Scanner(System.in);
14
15         // Declaración de variables
16         String nombre;
17         int edad;
18
19         // Solicitar datos al usuario
20         System.out.print("Ingrese su nombre: ");
21         nombre = input.nextLine();
22
23         System.out.print("Ingrese su edad: ");
24         edad = Integer.parseInt(input.nextLine());
25
26         // Mostrar resultados en pantalla
27         System.out.println("Su nombre es: " + nombre);
28         System.out.println("Su edad es: " + edad);
29     }
30 }
31
32 }
```

The left sidebar shows the project structure with the following items:

- Projects x
- Files
- Services
- 00_Concatenacion [00000000...]
- 00_Entradas [00000000...]
- 00_HolaMundo [00000000...]
- 00_Variables [00000000...]
- 01_EdadUsuario [00000000...]
- 02_PrecioTerreno [00000000...]
- 03_CantidadBilletes [00000000...]
- NombreYEdad [00000000...]
- Source Packages
 - nombreyedad
 - NombreYEdad.java [-/A]
- Test Packages
- Libraries
- Test Libraries

The bottom output window shows the execution results:

```
run:
Ingreso su nombre: Macarena
Ingreso su edad: 30
Su nombre es: Macarena
Su edad es: 30
BUILD SUCCESSFUL (total time: 6 seconds)
```

Enteros y operaciones

Se escribe un programa que solicite dos números enteros y realice las siguientes operaciones: suma, resta, multiplicación, división

Luego, muestra los resultados en la consola.

The screenshot shows an IDE window titled "OperacionesBasicas - Apache NetBeans L...". The main editor displays the source code for "OperacionesBasicas.java". The code defines a package, imports Scanner, and creates a public class with a main method. The main method uses Scanner to get two integers, performs addition, subtraction, multiplication, and division (with a check for division by zero), and prints the results. The output window at the bottom shows the execution results for the input values 5 and 3.

```

1  ...4 lines
2  package operacionesbasicas;
3  import java.util.Scanner;
4  /* @author Marinoni, Macarena */
5  public class OperacionesBasicas {
6
7      public static void main(String[] args) {
8          Scanner input = new Scanner(System.in);
9          // Declaración de variables
10         int num1, num2;
11         int suma, resta, multiplicacion;
12         double division;
13         // Solicitar números al usuario
14         System.out.print("Ingrese el primer número entero: ");
15         num1 = Integer.parseInt(input.nextLine());
16         System.out.print("Ingrese el segundo número entero: ");
17         num2 = Integer.parseInt(input.nextLine());
18         // Operaciones
19         suma = num1 + num2;
20         resta = num1 - num2;
21         multiplicacion = num1 * num2;
22         if (num2 != 0) {
23             division = (double) num1 / num2;
24         } else {
25             division = Double.NaN; // No es un número (cuando se divide por 0)
26         }
27         // Mostrar resultados
28         System.out.println("Suma: " + suma);
29         System.out.println("Resta: " + resta);
30         System.out.println("Multiplicación: " + multiplicacion);
31         if (num2 != 0) {
32             System.out.println("División: " + division);
33         } else {
34             System.out.println("División: No se puede dividir por 0");
35         }
36     }
37 }

```

Output:

```

run:
Ingrese el primer número entero: 5
Ingrese el segundo número entero: 3
Suma: 8
Resta: 2
Multiplicación: 15
División: 1.6666666666666667
BUILD SUCCESSFUL (total time: 4 seconds)

```

Datos Usuario

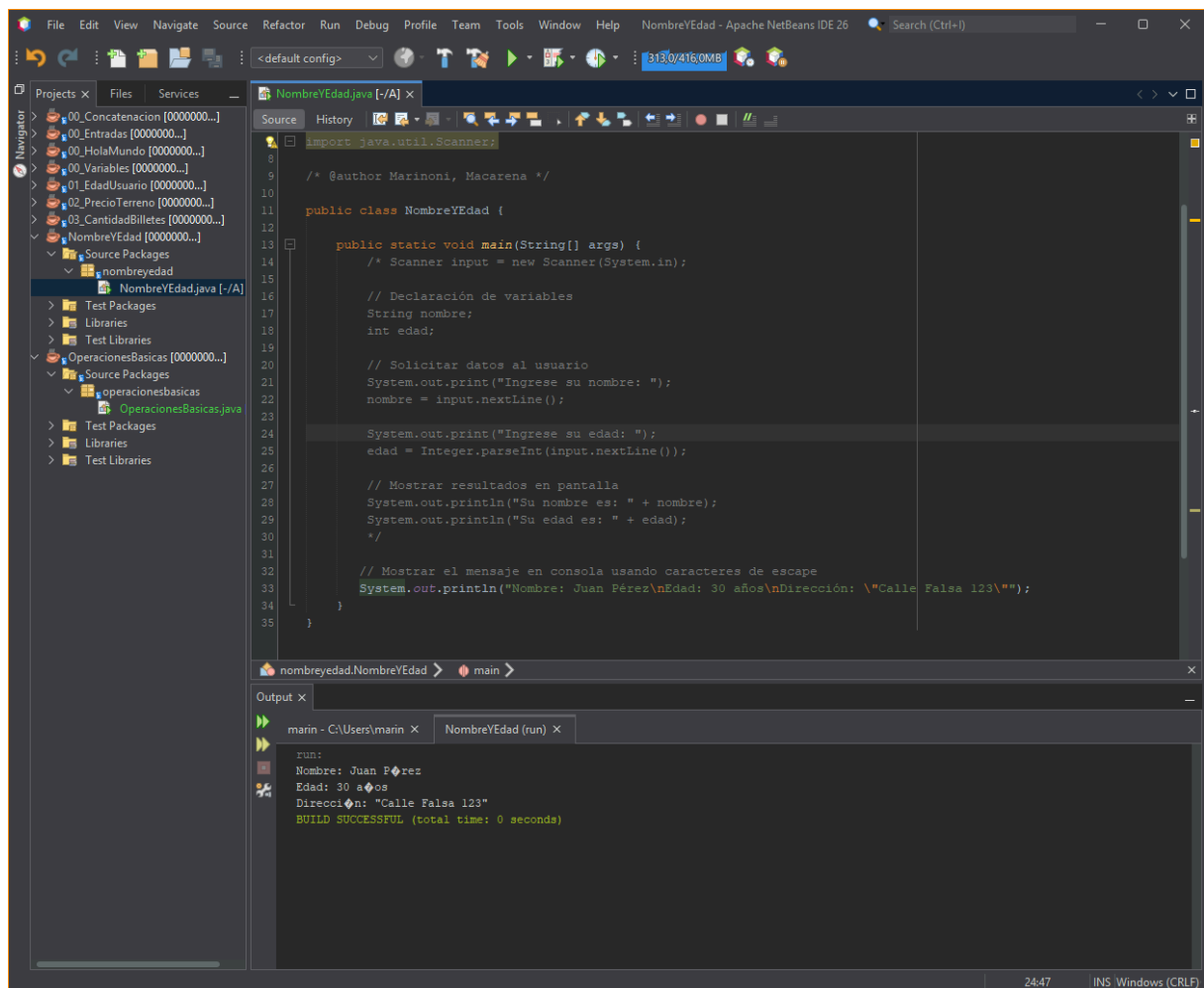
Se escribe un programa que muestra el siguiente mensaje en consola:

Nombre: Juan Pérez

Edad: 30 años

Dirección: "Calle Falsa 123"

Se utilizan caracteres de escape (\n, \") en System.out.println().



The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the source code of a Java file named `NombreYEdad.java`. The code imports `java.util.Scanner` and defines a `public class NombreYEdad` with a `main` method. The `main` method uses a `Scanner` to read user input for name and age, and then prints the results using `System.out.println`. The output window at the bottom shows the execution results, including the user input and the program's output.

```
import java.util.Scanner;

/* @author Marinoni, Macarena */

public class NombreYEdad {

    public static void main(String[] args) {

        /* Scanner input = new Scanner(System.in);

        // Declaración de variables
        String nombre;
        int edad;

        // Solicitar datos al usuario
        System.out.print("Ingrese su nombre: ");
        nombre = input.nextLine();

        System.out.print("Ingrese su edad: ");
        edad = Integer.parseInt(input.nextLine());

        // Mostrar resultados en pantalla
        System.out.println("Su nombre es: " + nombre);
        System.out.println("Su edad es: " + edad);
        */

        // Mostrar el mensaje en consola usando caracteres de escape
        System.out.println("Nombre: Juan Pérez\nEdad: 30 años\nDirección: \"Calle Falsa 123\"");

    }

}
```

Output:

```
run
Nombre: Juan Pérez
Edad: 30 años
Dirección: "Calle Falsa 123"
BUILD SUCCESSFUL (total time: 0 seconds)
```

Expresiones e Instrucciones

En el el siguiente código:

```
int x = 10; // Línea 1
x = x + 5; // Línea 2
System.out.println(x); // Línea 3
```

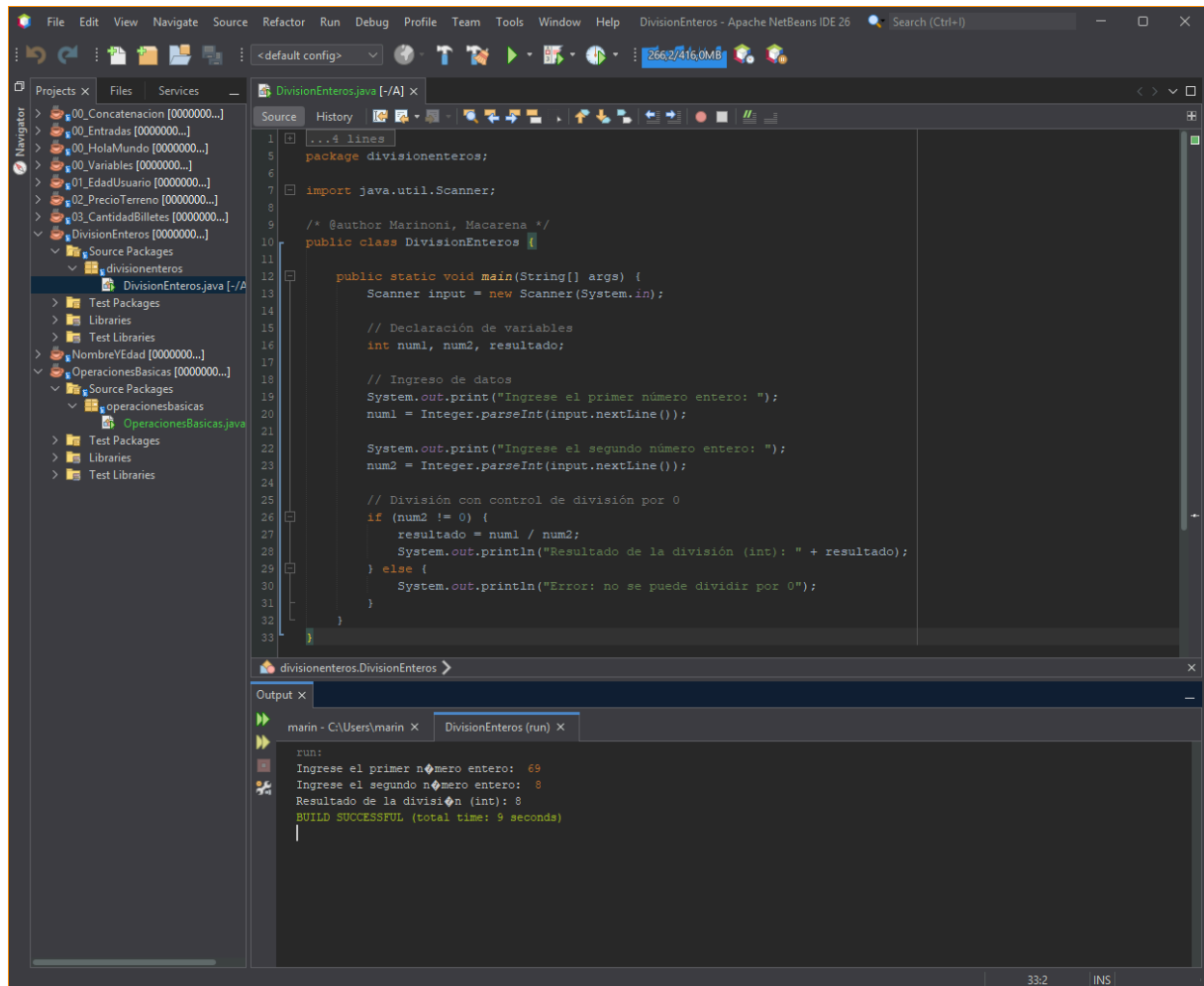
Se identifica que tanto la línea 1 como la línea 2 corresponden a Instrucciones con una Expresión a su derecha En el caso de la línea 3 sólo se observa una Instrucción.

Hay que tener presente que las Expresiones son fragmentos de código que se evalúan y producen un valor., mientras que las Instrucciones son sentencias completas que realizan una acción.

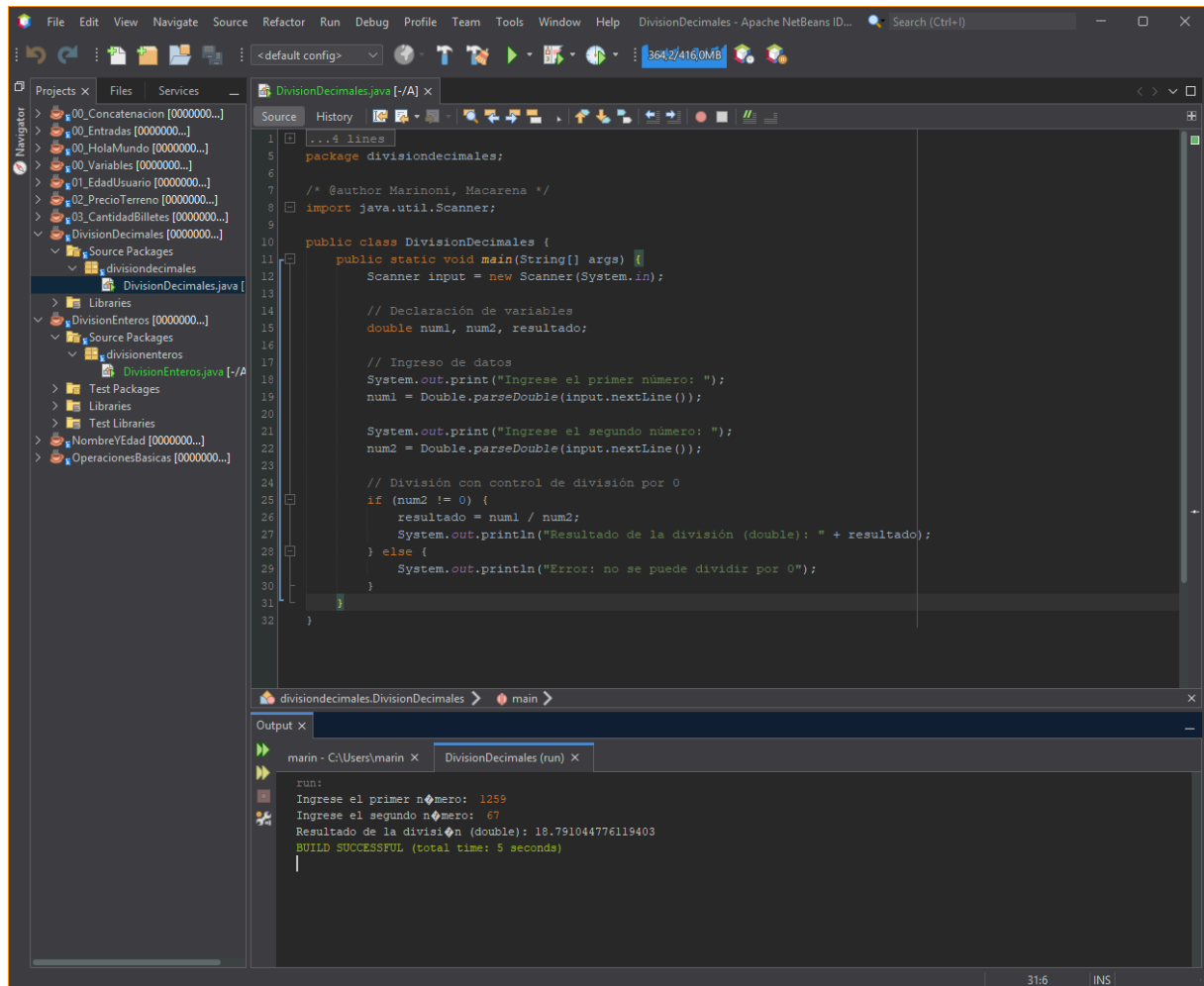
En concreto, la diferencia entre ellas es que una Expresión devuelve un valor (por ejemplo, $x + 5$ da como resultado un número), mientras que una Instrucción indica al programa que realice una acción (declarar, asignar, imprimir, etc.) y no necesariamente devuelve un valor.

Manejar conversiones de tipo y división en Java.

Se escribe un programa que divida dos números enteros ingresados por el usuario de la siguiente forma:



Se modifica el código anterior para usar double en lugar de int:



Si se comparan los dos códigos anteriores, puede interpretarse que con int, la división descarta la parte decimal mientras que con double, la división conserva la parte decimal

Corrección de código

Se analiza el siguiente código que presenta errores:

```
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);  
    }  
}
```

Se determina que `nextInt()` devuelve un entero (`int`), pero se está guardando en una variable de tipo `String` (`nombre`) lo cual genera un error de incompatibilidad. Esto mismo puede observarse en la consola

```
incompatible types: int cannot be converted to String  
May split declaration into a declaration and assignment  
----  
(Alt-Enter shows hints)  
String nombre = scanner.nextInt();
```

La solución es utilizar `nextLine()` en lugar de `nextInt()`

Prueba de escritorio

Se presenta a continuación la tabla de prueba de escritorio para el siguiente fragmento de código:

```

1  ...4 lines
5  package pruebaescritorio;
6
7  /**...4 lines */
11 public class PruebaEscritorio {
12
13     public static void main(String[] args) {
14         int a = 5;
15         int b = 2;
16         int resultado = a / b;
17         System.out.println("Resultado: " + resultado);
18     }
19
20 }
21

```

Línea	a	b	resultado
13	<sin_definir>	<sin_definir>	<sin_definir>
14	<sin_inicializar>	<sin_inicializar>	<sin_inicializar>
15	5	<sin_inicializar>	<sin_inicializar>
16	5	2	<sin_inicializar>
17	5	2	2
18	5	2	2

El valor de la variable *resultado* es 2 debido a que la variable está definida como int, lo cual genera que descarte a parte decimal del resultado de la división.