**PROGRAMACIÓN II**

**Trabajo Práctico 1: Introducción a Java**

# 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

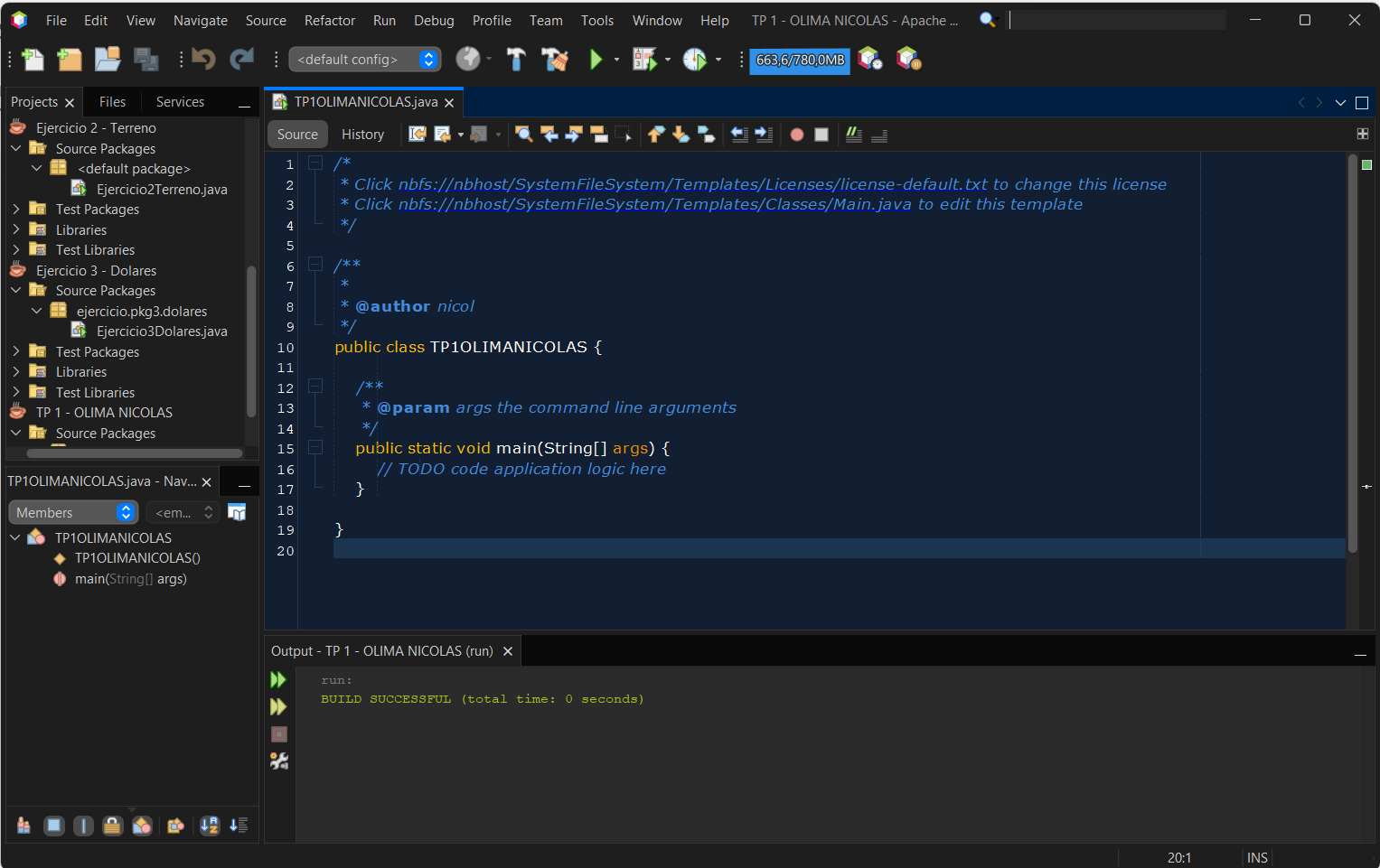
|  |  |
| --- | --- |
| **Concepto** | **Aplicación en el proyecto** |
| Instalación y entorno | Almacenan el conjunto de países |
| Variables y tipos de datos | Representan los datos de cada país (nombre, población, superficie, etc.) |
| Entrada y salida | Separan las operaciones: carga, búsqueda, estadísticas,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

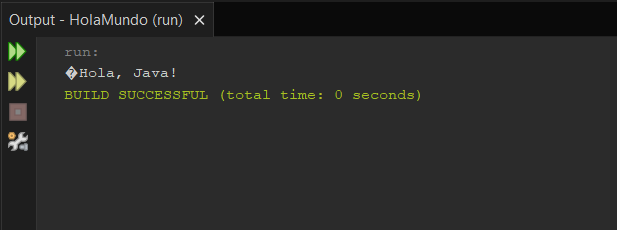
El trabajo consiste en resolver una serie de ejercicios introductorios en Java que permitan:

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

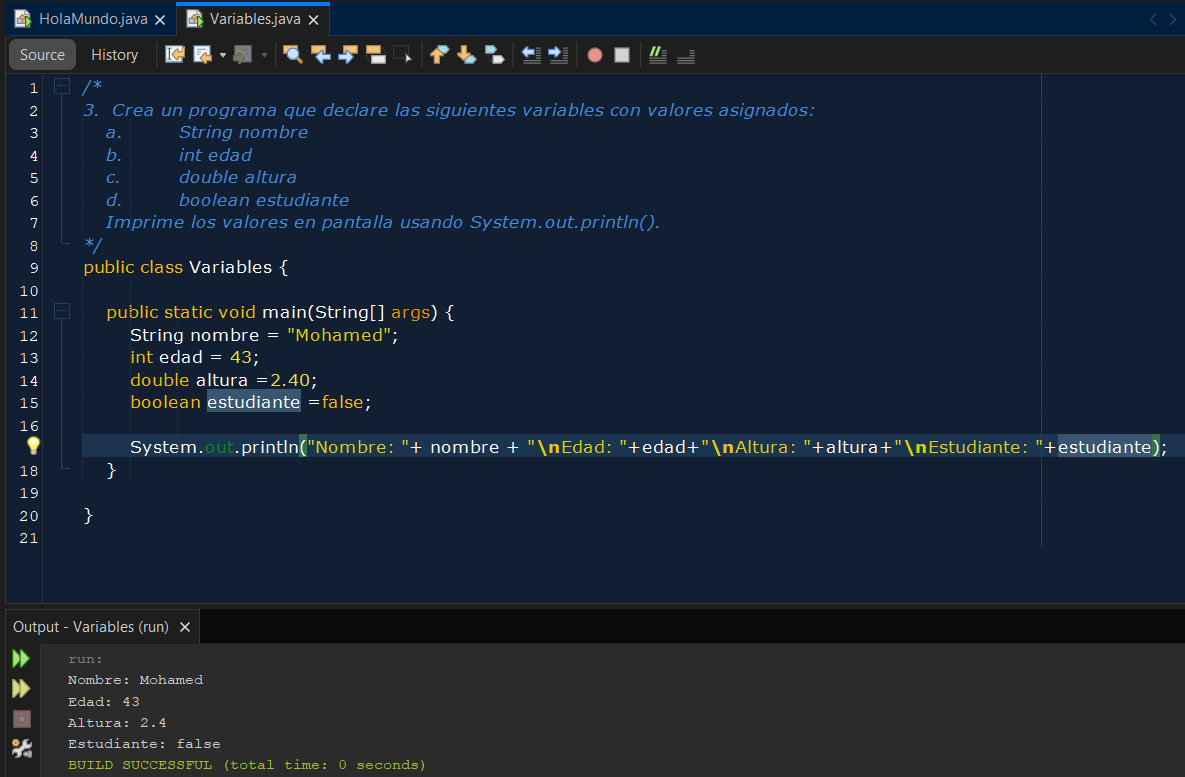


1. Escribir y ejecutar un programa básico en Java.
   * 1. Creá una clase llamada **HolaMundo**.
     2. Escribe un programa que imprima el mensaje: **¡Hola, Java!**
     3. Ejecuta el programa en NetBeans y adjunta una captura del resultado en la consola.

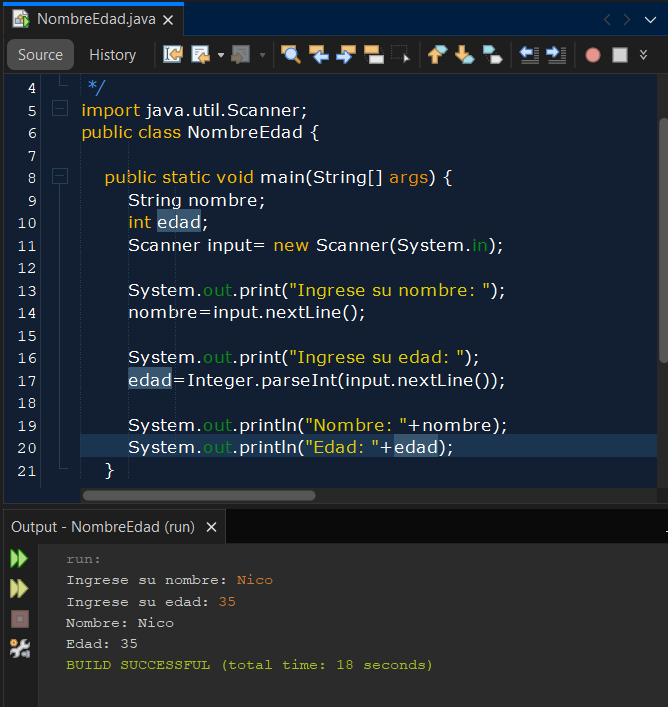


1. Crea un programa que declare las siguientes variables con valores asignados:
   * 1. String nombre
     2. int edad
     3. double altura
     4. boolean estudiante

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

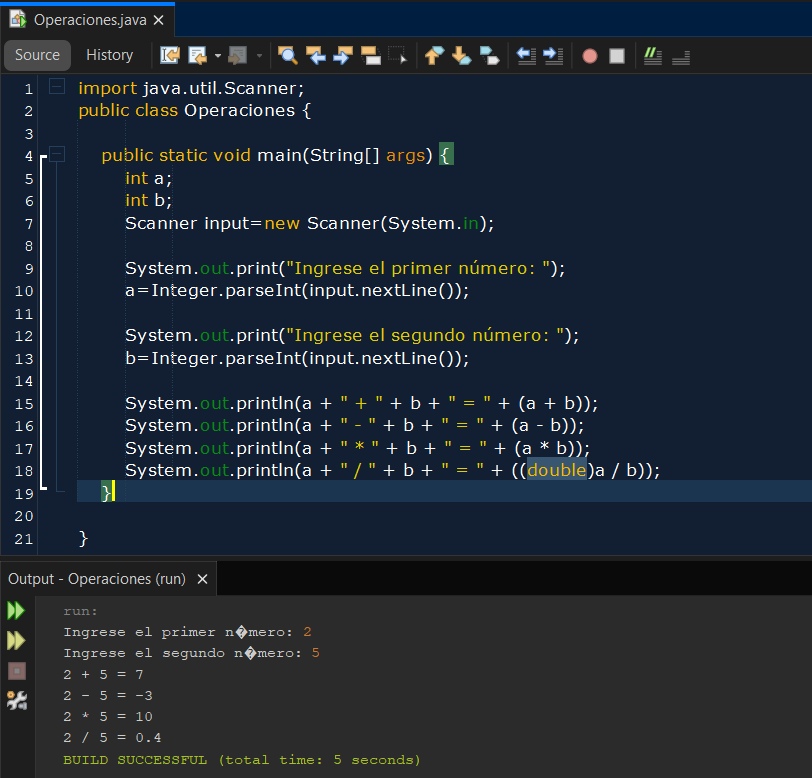
****

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.



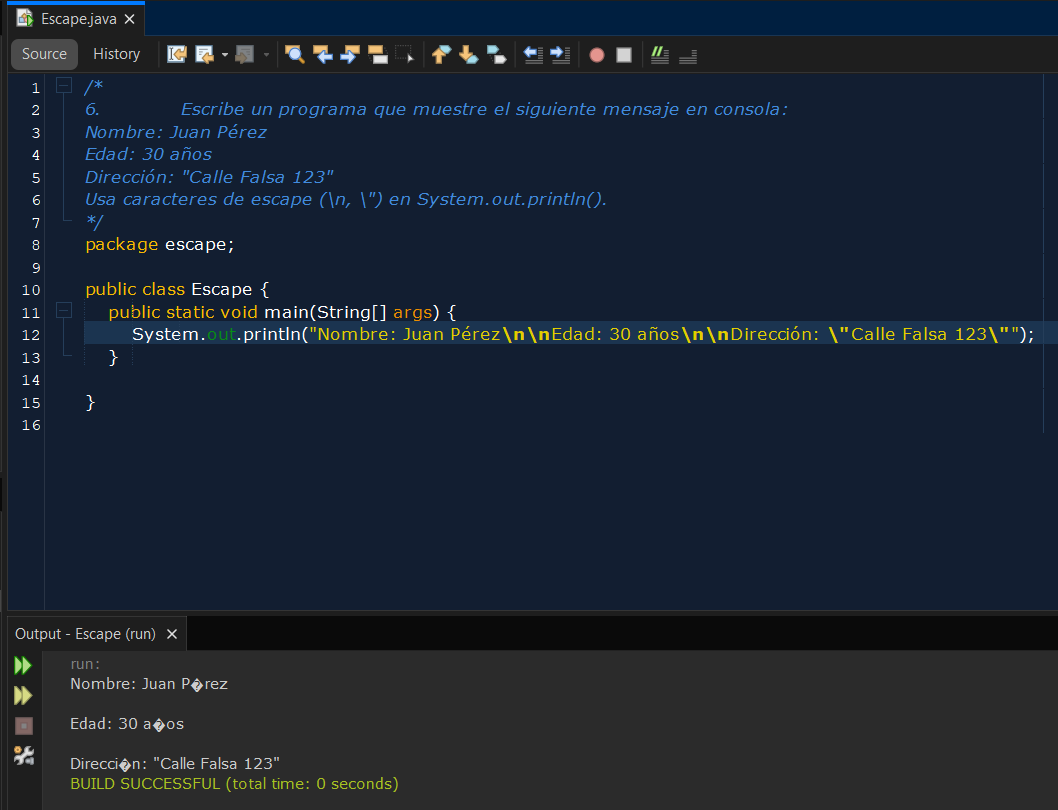
1. Escribe un programa que muestre el siguiente mensaje en consola:

**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.

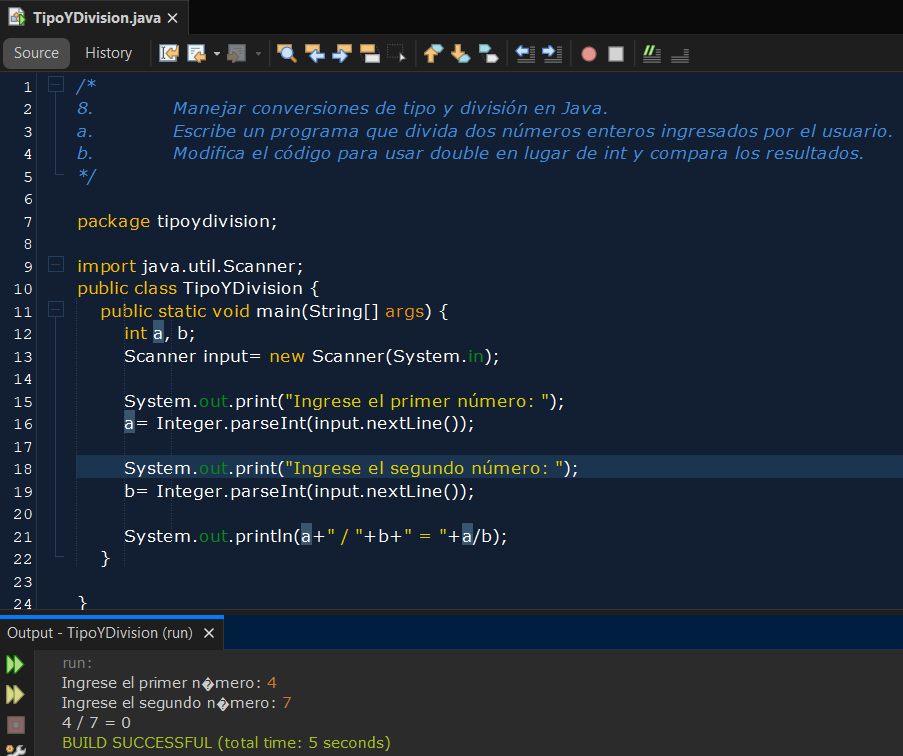
**int x = 10; // Línea 1 x = x + 5; // Línea 2**

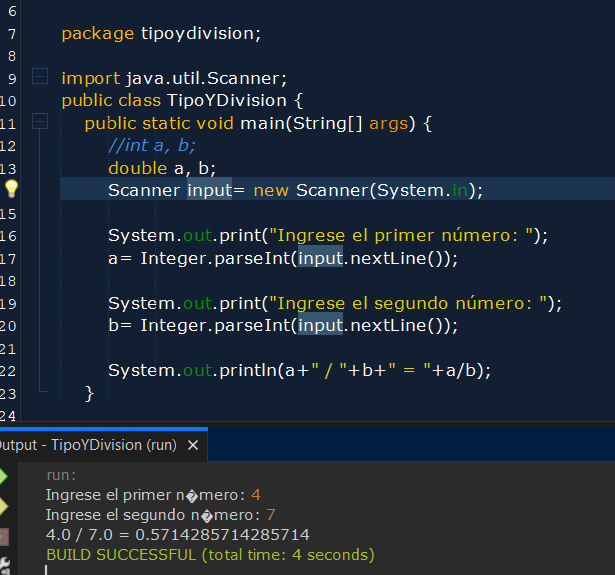
**System.out.println(x); // Línea 3**

Las líneas 1 y 3 son instrucciones, y la línea 2 es una expresión.

La diferencia es que una expresión es un conjunto de valores, variables y operandos, que siempre devuelve un valor. Mientras que las instrucciones son acciones a realizar que producen un efecto interno en el programa, controlan su flujo y pueden incluir una expresión.

1. Manejar conversiones de tipo y división en Java.
   1. Escribe un programa que divida dos números enteros ingresados por el usuario.
   2. Modifica el código para usar **double** en lugar de **int** y compara los resultados.





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);**

**}**

**}**

El error que tiene el código es que en la línea comentada con el texto “ERROR”, utiliza el método nextInt(), que se utiliza para ingresar valores numéricos enteros, para almacenar el valor ingresado por el usuario en la variable ‘nombre’, que es de tipo String.

La solución sería utilizar el método nextLine(), que permite leer líneas de texto completas.

1. Completa la tabla de prueba de escritorio para el siguiente código. ¿Cuál es el valor de **resultado** y por qué?

|  |
| --- |
| Linea 1 |
| Linea 2 |
| Linea 3 |
| Linea 4 |
| Linea 5 |
| Linea 6 |

**public class PruebaEscritorio {**

**public static void main(String[] args) {**

**int a = 5;**

**int b = 2;**

**int resultado = a / b;**

**System.out.println("Resultado: " + resultado);**

**}**

**}**

|  |  |  |  |
| --- | --- | --- | --- |
| variables  línea | a | b | resultado |
| 1 | Sin definir | Sin definir | Sin definir |
| 2 | Sin definir | Sin definir | Sin definir |
| 3 | Sin definir | Sin definir | Sin definir |
| 4 | 5 | Sin definir | Sin definir |
| 5 | 5 | 2 | Sin definir |
| 6 | 5 | 2 | 2 |

El valor de **resultado** es 2 porque la división se realiza entre valores de tipo int, por lo cual la parte decimal del resultado es descartada. Así, siendo el resultado de la división 2.5, sólo se considera la parte entera (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.