



TP N°1 : INTRODUCCIÓN A JAVA

Programación II

- Alumnos: Pérez Lucio Gabriel – perezluciogabriel@gmail.com
 - Profesor/a: Ariel Enferrel
 - Tutor: Marcos Vega
- Fecha de Entrega: 07/08/2025

Lucio Gabriel Pérez
perezluciogabriel@gmail.com

Índice:

Introducción y Ejercicio N°1.....	2
Ejercicio N°2.....	3
Ejercicio N°3.....	4
Ejercicio N°4.....	5
Ejercicio N°5.....	6
Ejercicio N°6.....	8
Ejercicio N°7.....	9
Ejercicio N°8.....	10
Ejercicio N°9.....	14
Ejercicio N°10 y link al repositorio de GitHub.....	16
Conclusión.....	17

1. Introducción:

En el presente trabajo se abordaran los ejercicios dados en la unidad 1 de la materia Programación II de la carrera Tecnicatura Universitaria en Programación a Distancia con el objetivo de consolidar los siguientes aspectos referentes a la programación en general y al lenguaje Java en específico:

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.

En la siguiente sección, se presentan los ejercicios resueltos en el lenguaje Java utilizando NetBeans, junto con sus correspondientes consignas.

2. Ejercicios:

Ejercicio 1:

Consigna:

Verificar que tienes instalado Java JDK y NetBeans. Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal: `java -versión`. Abre NetBeans, crea un nuevo proyecto y configura el modo oscuro. Toma una captura de pantalla del entorno configurado y agrégala a tu entrega.

Respuesta:

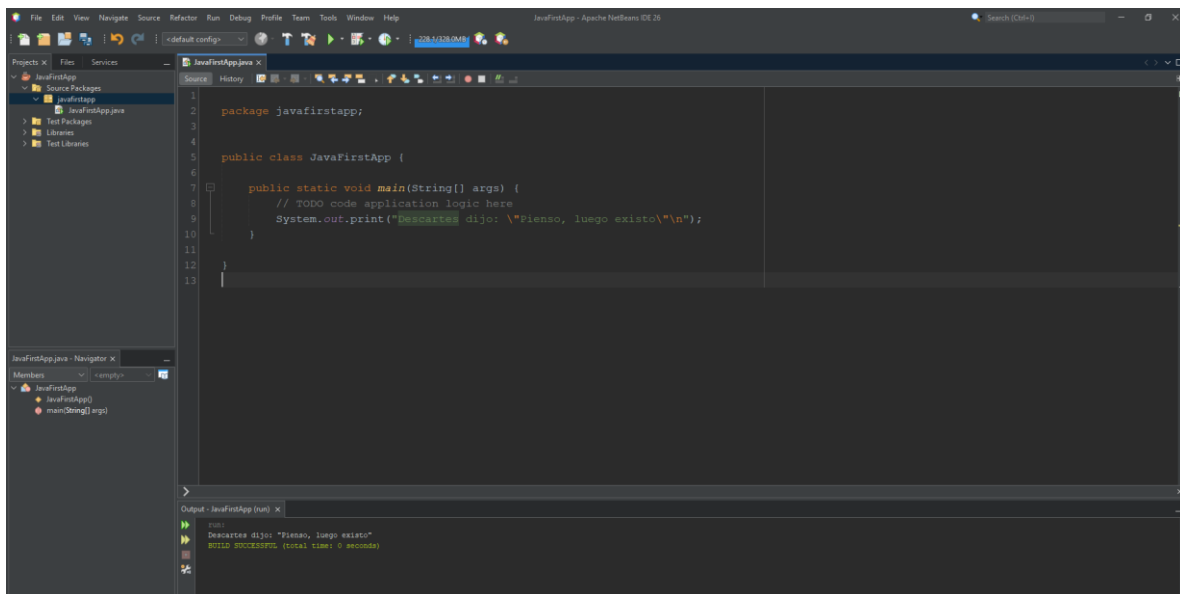
Se verifica que Java JDK esté instalado correctamente en el sistema:

```
Command Prompt
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lgp_2>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\lgp_2>
```

Se configuró el entorno de desarrollo en modo oscuro:

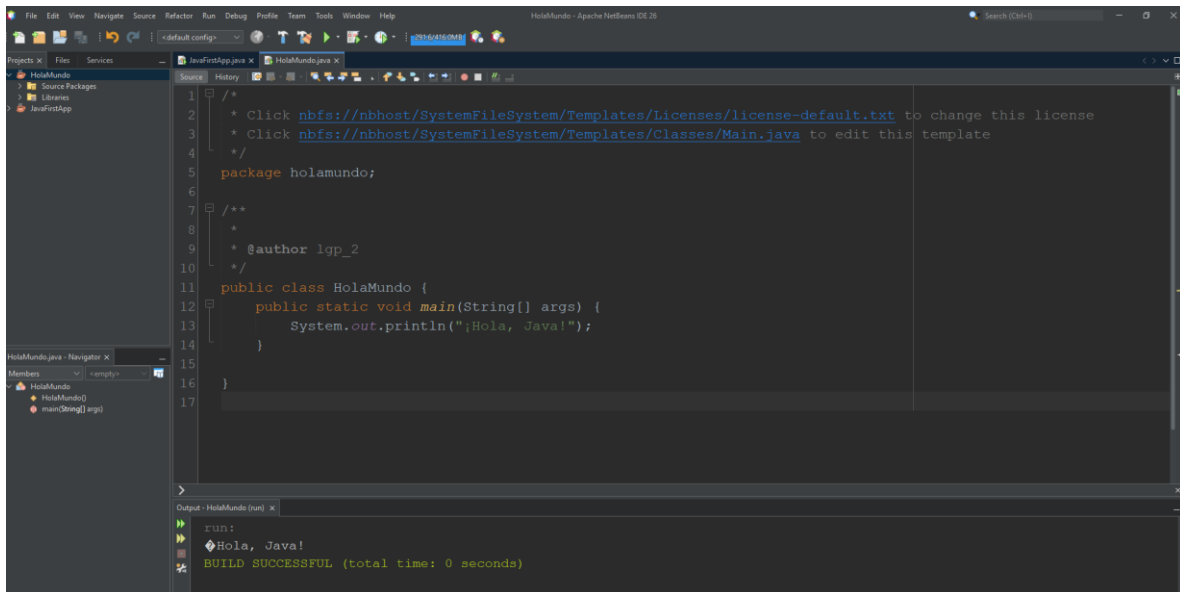


Ejercicio 2:

Consigna:

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

Captura:



Código Java:

```
package holamundo;

public class HolaMundo {
    public static void main(String[] args) {
        System.out.println("¡Hola, Java!");
    }
}
```

Salida por consola:

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

Ejercicio 3:

Consigna:

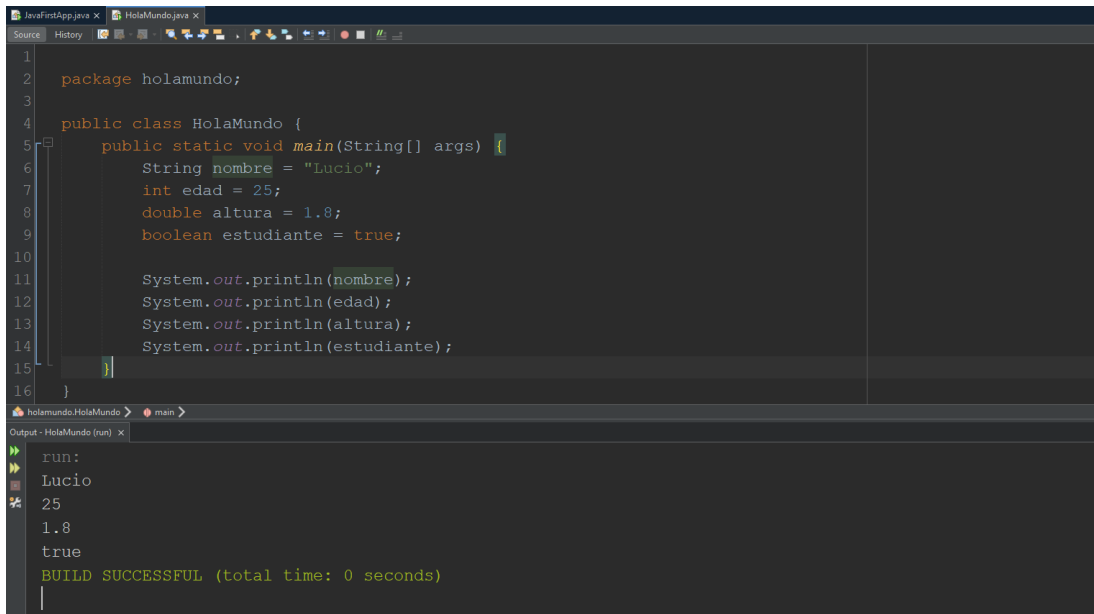
Crea un programa que declare las siguientes variables con valores asignados:

- a. String nombre
- b. int edad
- c. double altura

d. boolean estudiante

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

Captura:

A screenshot of an IDE window. The top pane shows a Java file named 'HolaMundo.java' with the following code:

```
1 package holamundo;
2
3
4 public class HolaMundo {
5     public static void main(String[] args) {
6         String nombre = "Lucio";
7         int edad = 25;
8         double altura = 1.8;
9         boolean estudiante = true;
10
11         System.out.println(nombre);
12         System.out.println(edad);
13         System.out.println(altura);
14         System.out.println(estudiante);
15     }
16 }
```

The bottom pane shows the output of running the program:

```
run:
Lucio
25
1.8
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Código Java:

```
package holamundo;

public class HolaMundo {
    public static void main(String[] args) {
        String nombre = "Lucio";
        int edad = 25;
        double altura = 1.8;
        boolean estudiante = true;

        System.out.println(nombre);
        System.out.println(edad);
        System.out.println(altura);
        System.out.println(estudiante);
    }
}
```

Salida por consola:

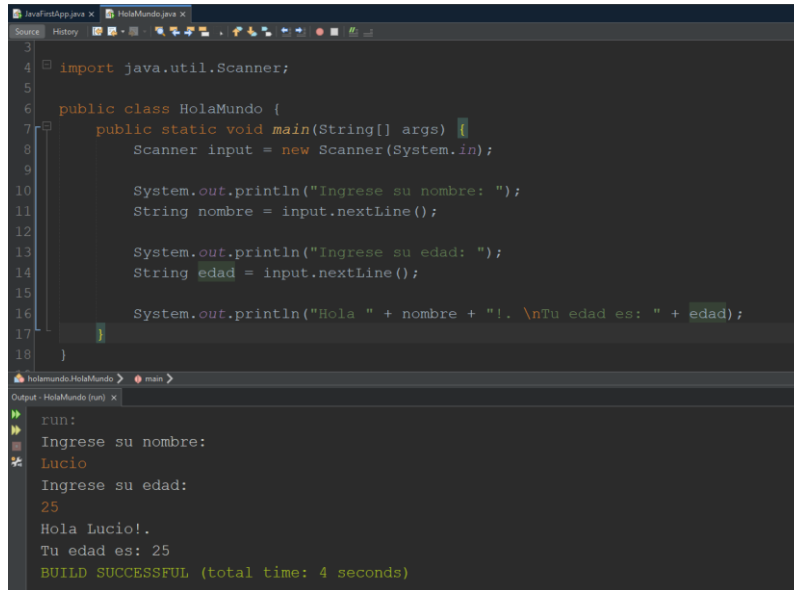
```
run:
Lucio
25
1.8
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio 4:

Consigna:

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

Captura:

A screenshot of an IDE window. The top pane shows the source code for a Java class named 'HolaMundo'. The code imports 'java.util.Scanner', defines a 'main' method, creates a 'Scanner' object for 'System.in', prompts the user for their name and age, and then prints the name and age. The bottom pane shows the output of the program, which matches the prompts and user input shown in the code: 'Ingrese su nombre:', 'Lucio', 'Ingrese su edad:', '25', and the final output 'Hola Lucio!. \nTu edad es: 25'. The IDE also shows a 'BUILD SUCCESSFUL' message at the bottom.

Código Java:

```
package holamundo;
import java.util.Scanner;
public class HolaMundo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Ingrese su nombre: ");
        String nombre = input.nextLine();

        System.out.println("Ingrese su edad: ");
        String edad = input.nextLine();

        System.out.println("Hola " + nombre + "!. \nTu edad es: " +
edad);
    }
}
```

Salida por consola:

```
run:
Ingrese su nombre:
Lucio
Ingrese su edad:
25
```

```
Hola Lucio!.  
Tu edad es: 25  
BUILD SUCCESSFUL (total time: 4 seconds)
```

Ejercicio 5:

Consigna:

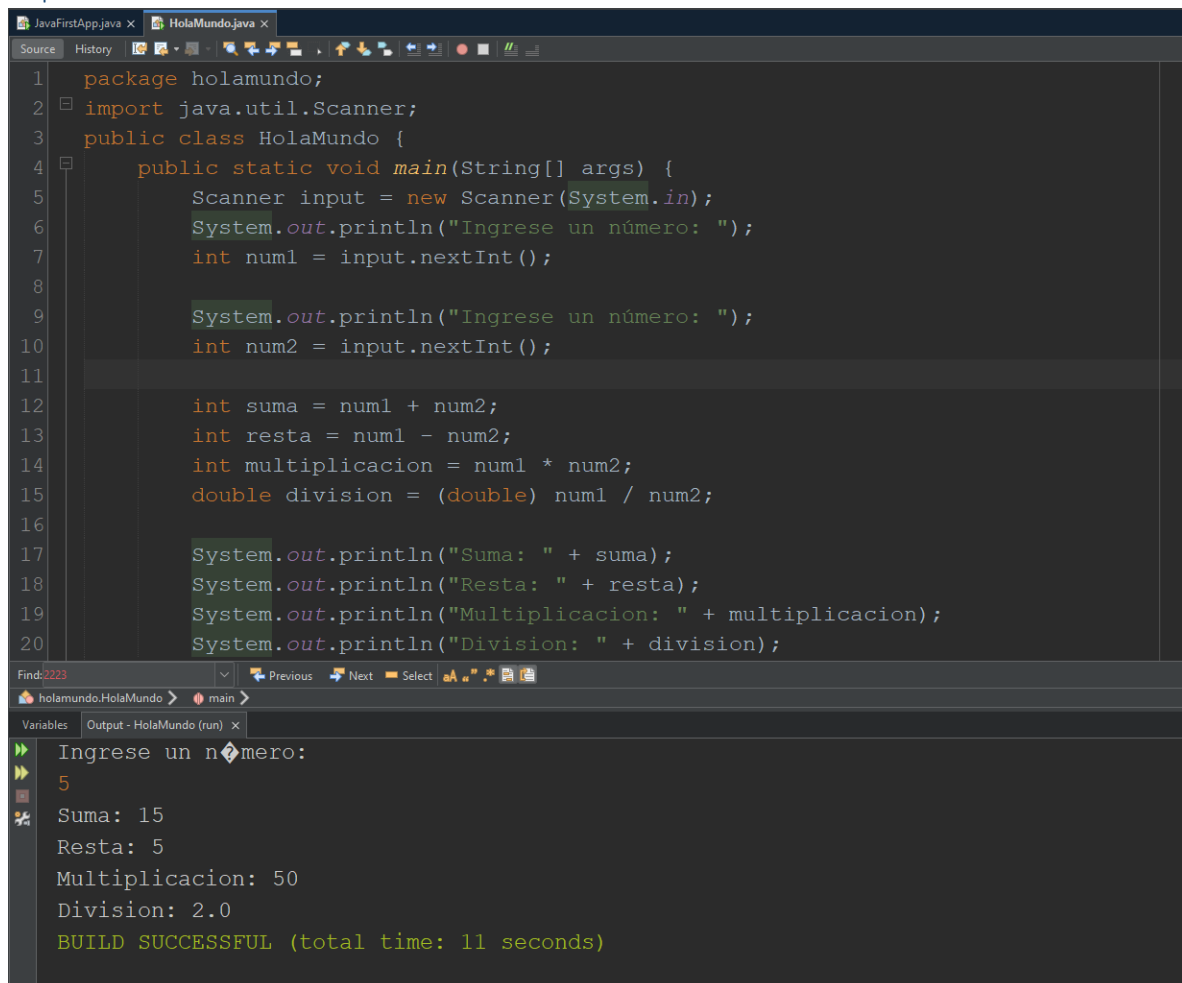
Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:

- a. Suma
- b. Resta
- c. Multiplicación
- d. División

Muestra los resultados en la consola.

Respuesta:

Captura:



```
1 package holamundo;  
2 import java.util.Scanner;  
3 public class HolaMundo {  
4     public static void main(String[] args) {  
5         Scanner input = new Scanner(System.in);  
6         System.out.println("Ingrese un número: ");  
7         int num1 = input.nextInt();  
8  
9         System.out.println("Ingrese un número: ");  
10        int num2 = input.nextInt();  
11  
12        int suma = num1 + num2;  
13        int resta = num1 - num2;  
14        int multiplicacion = num1 * num2;  
15        double division = (double) num1 / num2;  
16  
17        System.out.println("Suma: " + suma);  
18        System.out.println("Resta: " + resta);  
19        System.out.println("Multiplicacion: " + multiplicacion);  
20        System.out.println("Division: " + division);  
21    }  
22 }  
23
```

Find: 2223

holamundo.HolaMundo > main >

Variables Output - HolaMundo (run) x

```
Ingrese un número:  
5  
Suma: 15  
Resta: 5  
Multiplicacion: 50  
Division: 2.0  
BUILD SUCCESSFUL (total time: 11 seconds)
```


Código Java:

```
package holamundo;
import java.util.Scanner;
public class HolaMundo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Ingrese un número: ");
        int num1 = input.nextInt();

        System.out.println("Ingrese un número: ");
        int num2 = input.nextInt();

        int suma = num1 + num2;
        int resta = num1 - num2;
        int multiplicacion = num1 * num2;
        double division = (double) num1 / num2;

        System.out.println("Suma: " + suma);
        System.out.println("Resta: " + resta);
        System.out.println("Multiplicacion: " + multiplicacion);
        System.out.println("Division: " + division);
    }
}
```

Salida por consola:

```
run:
Ingrese un número:
10
Ingrese un número:
5
Suma: 15
Resta: 5
Multiplicacion: 50
Division: 2.0
BUILD SUCCESSFUL (total time: 11 seconds)
```

Ejercicio 6:

Consigna:

Escribe un programa que muestre el siguiente mensaje en consola:

Nombre: Juan Pérez

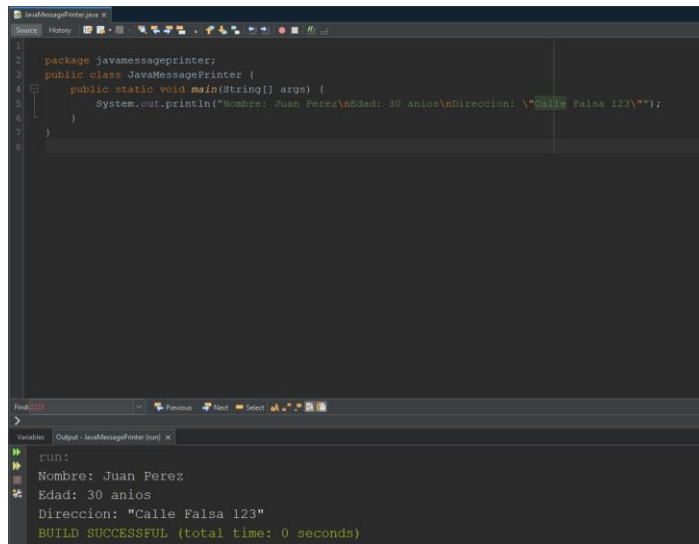
Edad: 30 años

Dirección: "Calle Falsa 123"

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

Respuesta:

Captura:

A screenshot of an IDE window titled 'JavaMessagePrinter.java'. The editor shows the following Java code:

```
1 package javamessageprinter;  
2  
3 public class JavaMessagePrinter {  
4     public static void main(String[] args) {  
5         System.out.println("Nombre: Juan Perez\nEdad: 30 anios\nDireccion: \"Calle Falsa 123\"");  
6     }  
7 }  
8
```

The bottom panel shows the 'Output' window with the following text:

```
run:  
Nombre: Juan Perez  
Edad: 30 anios  
Direccion: "Calle Falsa 123"  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Código Java:

```
package javamessageprinter;  
public class JavaMessagePrinter {  
    public static void main(String[] args) {  
        System.out.println("Nombre: Juan Perez\nEdad: 30  
anios\nDireccion: \"Calle Falsa 123\"");  
    }  
}
```

Salida por consola:

```
run:  
Nombre: Juan Perez  
Edad: 30 anios  
Direccion: "Calle Falsa 123"  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio 7:

Consigna:

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
```

Respuesta:

Primero comencemos definiendo los términos *expresión* e *instrucción*. Una **instrucción** es una *unidad completa de código que realiza una acción* (en Java siempre deben terminar en “;”) mientras que una **expresión** es un *fragmento de código que evalúa un valor*.

Todas las líneas en el código dado **son instrucciones que contienen una expresión**. Expliquémoslas paso a paso:

La línea 1 da la instrucción de *declarar e inicializar la variable X con el valor de 10*. La instrucción es toda la unidad de código: “**int x = 10;**” mientras que la expresión es “x = 10” ya que se asigna el valor de 10 a la variable en cuestión.

La línea 2 da la orden de asignar a X *el valor del resultado de hacer la operación: “x + 5”*. La instrucción es toda la unidad de código: “**x = x + 5;**”, mientras que la expresión es el fragmento de código: “x + 5”.

La línea 3 llama al método println() evaluando X. La instrucción es la unidad de código: “**System.out.println(x);**”, mientras que la expresión es la simple evaluación de “x” dentro de la llamada al método println().

Ejercicio 8:

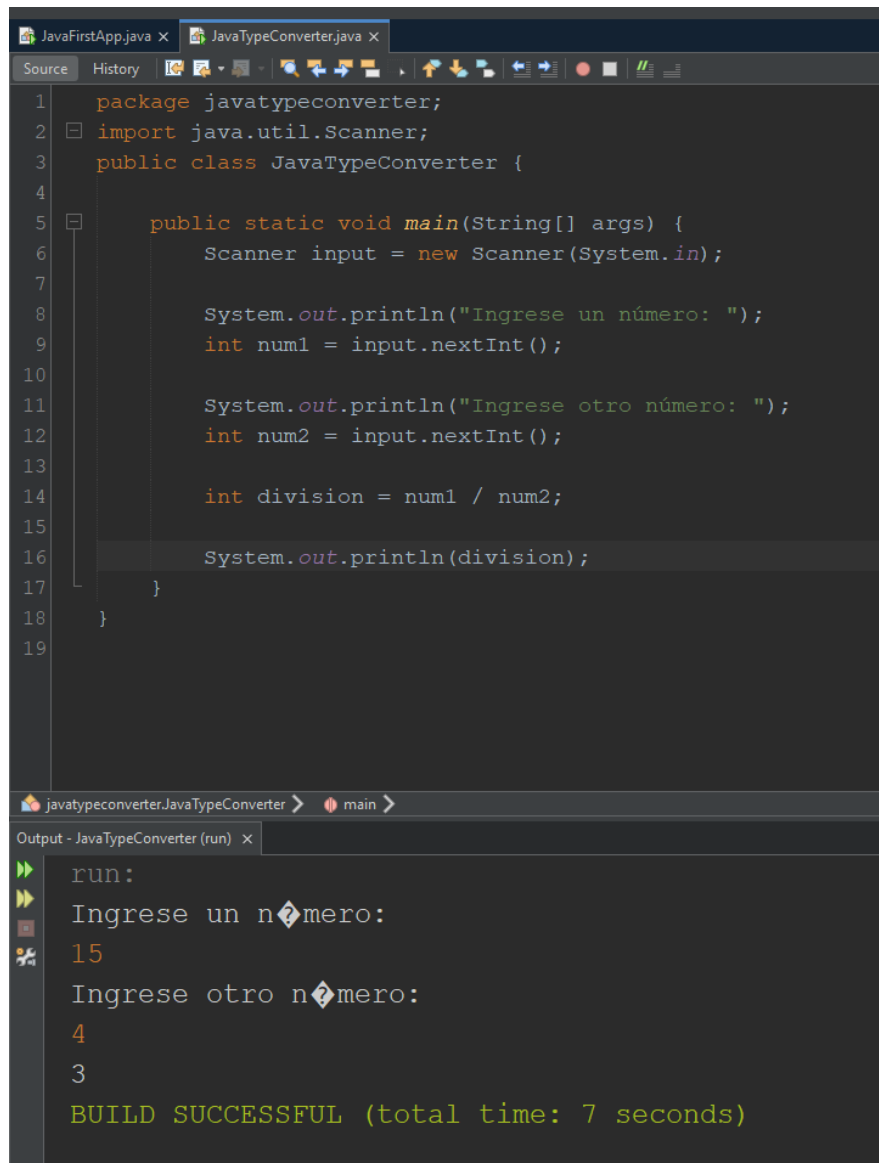
Consigna:

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.

Respuesta:

Captura:



The screenshot shows an IDE with two tabs: 'JavaFirstApp.java' and 'JavaTypeConverter.java'. The 'JavaTypeConverter.java' tab is active, displaying the following code:

```
1 package javatypeconverter;
2 import java.util.Scanner;
3 public class JavaTypeConverter {
4
5     public static void main(String[] args) {
6         Scanner input = new Scanner(System.in);
7
8         System.out.println("Ingrese un número: ");
9         int num1 = input.nextInt();
10
11         System.out.println("Ingrese otro número: ");
12         int num2 = input.nextInt();
13
14         int division = num1 / num2;
15
16         System.out.println(division);
17     }
18 }
19
```

Below the code editor, the 'Output - JavaTypeConverter (run)' window shows the execution results:

```
run:
Ingrese un número:
15
Ingrese otro número:
4
3
BUILD SUCCESSFUL (total time: 7 seconds)
```

Código Java:

```
package javatypeconverter;
import java.util.Scanner;
public class JavaTypeConverter {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Ingrese un número: ");
        int num1 = input.nextInt();
```

```

        System.out.println("Ingrese otro número: ");
        int num2 = input.nextInt();

        int division = num1 / num2;

        System.out.println(division);
    }
}

```

Salida por consola:

```

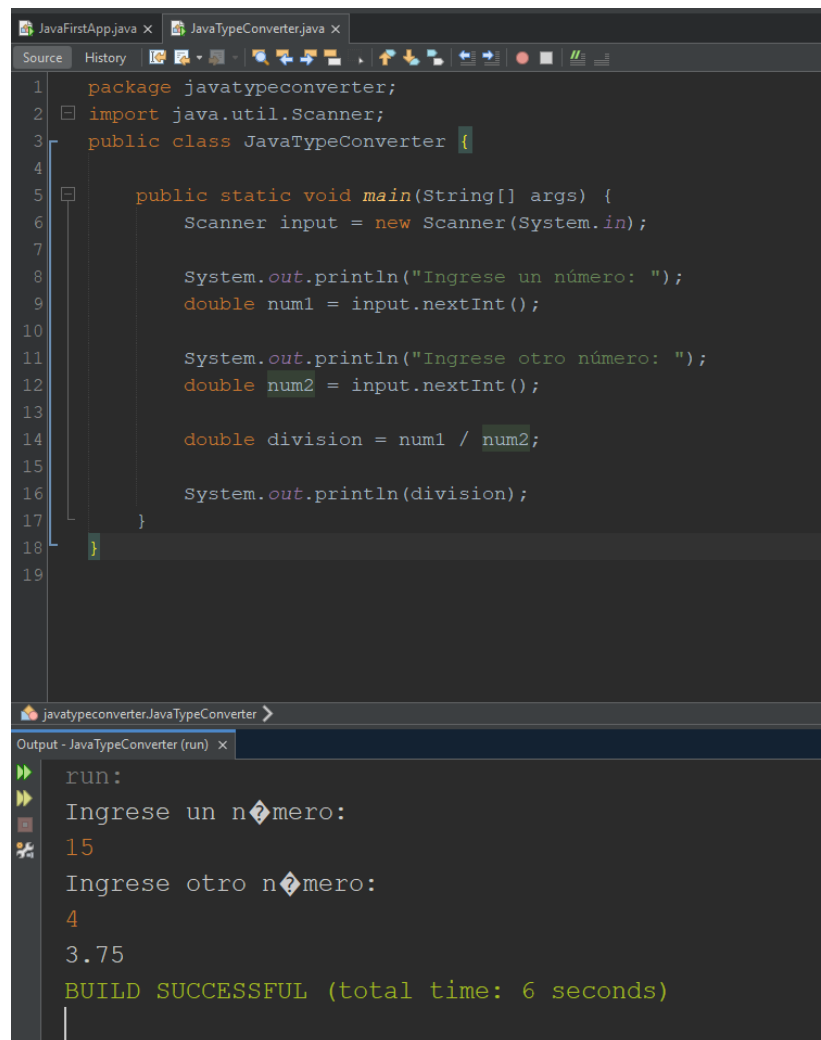
run:
Ingrese un número:
15
Ingrese otro número:
4
3
BUILD SUCCESSFUL (total time: 7 seconds)

```

Este código *actualmente declara e inicializa tres variables de tipo int* “num1”, “num2” y “división” y crea un objeto Scanner que se almacena en “input”, para pedir que el usuario ingrese números enteros. En la variable “num1” se almacena el número ingresado por el usuario en la primera ocasión, al igual que “num2”, pero esta vez se trata del número ingresado en la segunda ocasión. En el caso de “división”, almacena el resultado de la división entre “num1” y “num2”. En este caso, al declarar “división” como entero (int) el resultado que se almacene será un entero y no se tendrá en cuenta el decimal, por tal motivo es que el resultado que nos imprime por pantalla es 3 y no 3.75, como debería ser. Esto se llama ***división entera***.

A continuación se modificará este código y se presentará una versión en la que las variables “num1”, “num2”, y “división” sean definidas con el tipo de dato ***double***.

Captura:



The screenshot shows an IDE with two tabs: 'JavaFirstApp.java' and 'JavaTypeConverter.java'. The 'JavaTypeConverter.java' tab is active, displaying the following code:

```
1 package javatypeconverter;
2 import java.util.Scanner;
3 public class JavaTypeConverter {
4
5     public static void main(String[] args) {
6         Scanner input = new Scanner(System.in);
7
8         System.out.println("Ingrese un número: ");
9         double num1 = input.nextInt();
10
11         System.out.println("Ingrese otro número: ");
12         double num2 = input.nextInt();
13
14         double division = num1 / num2;
15
16         System.out.println(division);
17     }
18 }
19
```

Below the code editor, the 'Output - JavaTypeConverter (run)' window shows the execution results:

```
run:
Ingrese un número:
15
Ingrese otro número:
4
3.75
BUILD SUCCESSFUL (total time: 6 seconds)
```

Código Java:

```
package javatypeconverter;
import java.util.Scanner;
public class JavaTypeConverter {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Ingrese un número: ");
        double num1 = input.nextInt();

        System.out.println("Ingrese otro número: ");
        double num2 = input.nextInt();

        double division = num1 / num2;
```

```

        System.out.println(division);
    }
}

```

Salida por consola:

```

run:
Ingrese un número:
15
Ingrese otro número:
4
3.75
BUILD SUCCESSFUL (total time: 6 seconds)

```

En este caso, todas las variables numéricas fueron declaradas con el tipo de dato double. Las divisiones en Java son enteras por defecto, es decir, que si en los dividendos no existe al menos un valor que sea del tipo double, la división retornara un valor entero. En este código ahora modificado, ambos dividendos son del tipo double por lo que la división retorna double e imprime por consola “3.75”, que es el resultado real de dividir 15 por 4.

Ejercicio 9:

Consigna:

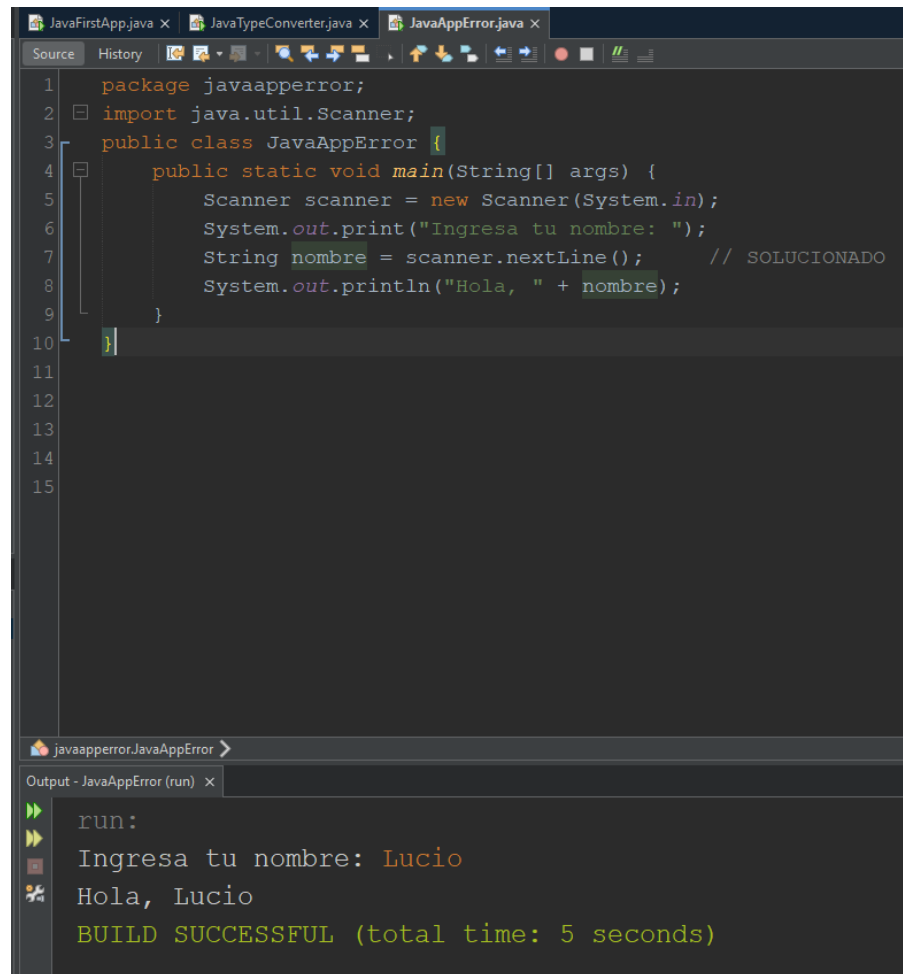
Corrige el siguiente código para que funcione correctamente. Explica qué error tenía y cómo lo solucionaste.

```

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 presenta este código es que intenta convertir a String un dato de tipo int. Concretamente en la línea: “String nombre = scanner.nextInt();” se intenta llamar al método Scanner para pedir al usuario un dato de tipo “int” y almacenarlo en una variable que se definió de tipo String, lo cual no es posible y devuelve error de compilación. El primer paso para solucionar este error es comprender la finalidad con la que se escribió este código. Teniendo en cuenta que la línea anterior lanza un mensaje por consola comunicándole al usuario que ingrese su nombre, y que la línea siguiente, intenta imprimir por consola “Hola” + nombre, deduzco que lo que la intención con esa línea era capturar el nombre del usuario para luego saludarlo, por lo cuál infiero que el error está en que el método llamado (“scanner.nextInt()”) no es el correcto para este caso, sino que se debería utilizar scanner.nextLine(). Haciendo esa corrección, el código debería funcionar de la forma esperada.

A screenshot of an IDE window with three tabs: 'JavaFirstApp.java', 'JavaTypeConverter.java', and 'JavaAppError.java'. The 'JavaAppError.java' tab is active, showing the following code:

```
1 package javaapperror;
2 import java.util.Scanner;
3 public class JavaAppError {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Ingresa tu nombre: ");
7         String nombre = scanner.nextLine(); // SOLUCIONADO
8         System.out.println("Hola, " + nombre);
9     }
10 }
11
12
13
14
15
```

Below the code editor is a console window titled 'Output - JavaAppError (run)'. It shows the execution output:

```
run:
Ingresa tu nombre: Lucio
Hola, Lucio
BUILD SUCCESSFUL (total time: 5 seconds)
```

Código Java:

```
package javaapperror;
import java.util.Scanner;
public class JavaAppError {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingresa tu nombre: ");
        String nombre = scanner.nextLine(); // SOLUCIONADO
        System.out.println("Hola, " + nombre);
    }
}
```

Salida por consola:

```
run:
Ingresa tu nombre: Lucio
Hola, Lucio
BUILD SUCCESSFUL (total time: 5 seconds)
```


Como se puede ver en la captura, en el código Java y la salida por consola, efectivamente, se le dio solución al error presentado y el programa funcionó de la manera esperada acorde a la finalidad intencionada. El programa pide el nombre al usuario y lo imprime por consola en un saludo.

Ejercicio 10:

Consigna:

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);  
}  
}
```

Tabla de prueba de escritorio:

Línea	A	b	resultado
1	Sin definir	Sin definir	Sin definir
2	5	Sin definir	Sin definir
3	5	2	Sin definir
4	5	2	2
5	5	2	2

En la tabla de escritorio de arriba, correspondiente al código citado en el enunciado del presente ejercicio, donde cabe aclarar que, la línea N°1 es aquella en la que se define la clase PruebaEscritorio y desde allí es que se comienza el conteo de las siguientes líneas para la tabla, podemos ver que las líneas 2, 3 y 4 se encargan de definir como tipo de dato "int", los valores de las variables "a", "b" y "resultado". El valor de la variable "resultado" viene dado del resultado de la división de los valores de las variables "a" y "b". Dado que tanto "resultado", como así también las variables "a" y "b" son valores del tipo "int", el resultante será también un valor de tipo int a causa del funcionamiento interno del lenguaje Java, es por eso que el valor de "resultado" es "2" y no "2.5", como hubiera sido en el caso en el que los tipos de dato hubieran sido "double".

Link al repositorio de GitHub: https://github.com/lucioperez01/TUPaD-Programacion-II_Trabajos-Practicos/tree/main/TP-1Introduccion-a-Java

3. Conclusión:

Este trabajo me sirvió para comprender los fundamentos del lenguaje Java y reforzar varios conceptos clave como la declaración de variables, entrada de datos, operaciones básicas y diferencias entre expresiones e instrucciones. También entendí mejor cómo funciona el sistema de tipos en Java y la importancia de probar el código y leer bien los errores para corregirlos. En resumen, fue una buena base para arrancar más firme en la materia.