

Trabajo Práctico 1

Introducción a Java

Programación II

Alumno

Alex Pedro Dauria

Fecha de Entrega

23 de Agosto de 2025

Índice

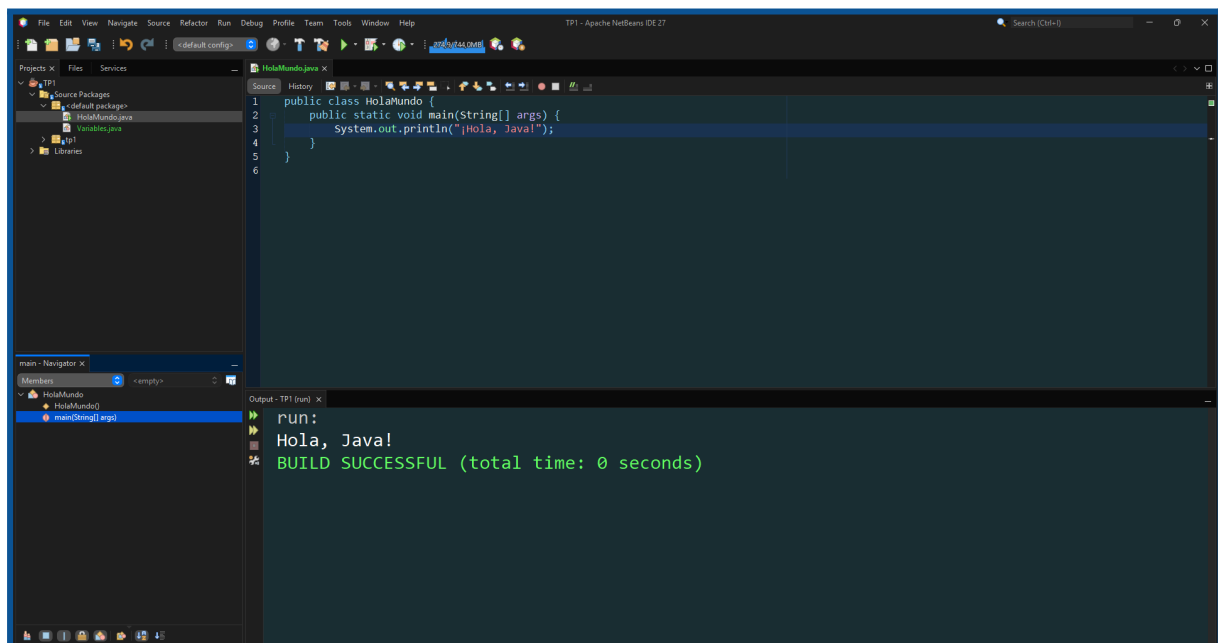
Ejercicio 1: <i>Configuración de entorno</i>	2
Ejercicio 2: <i>Programa básico que imprime “¡Hola, Java!”</i>	2
Ejercicio 3: <i>Declarar variables e imprimir sus valores</i>	3
Ejercicio 4: <i>Solicitar nombre y edad con Scanner</i>	3
Ejercicio 5: <i>Realizar operaciones aritméticas</i>	4
Ejercicio 6: Usar caracteres de escape para formato de salida	4
Ejercicio 7: <i>Análisis de código - Expresiones vs Instrucciones</i>	5
Ejercicio 8: <i>Manejar conversiones de tipo y división</i>	6
Ejercicio 9: <i>Corrección de código para leer nombre</i>	7
Ejercicio 10: <i>Prueba de escritorio</i>	8
Link al Repositorio en GitHub	8

Ejercicio 1: Configuración de entorno

Confirmación de instalación de Java JDK

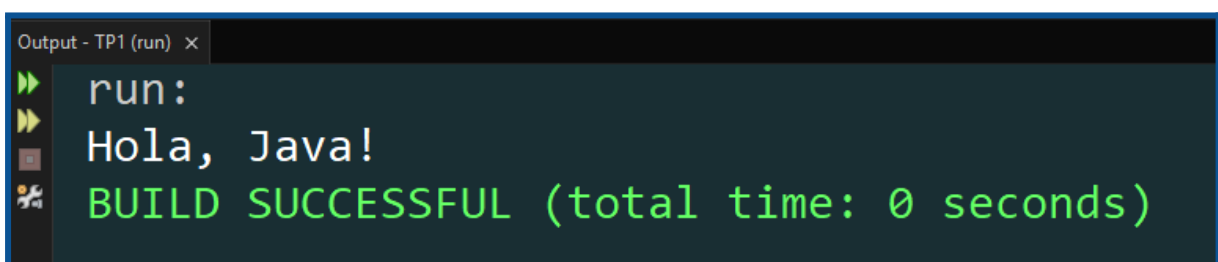
```
PS C:\Users\A_P_D> java -version
java version "21.0.8" 2025-07-15 LTS
Java(TM) SE Runtime Environment (build 21.0.8+12-LTS-250)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.8+12-LTS-250, mixed mode, sharing)
PS C:\Users\A_P_D> |
```

Modo oscuro configurado correctamente:



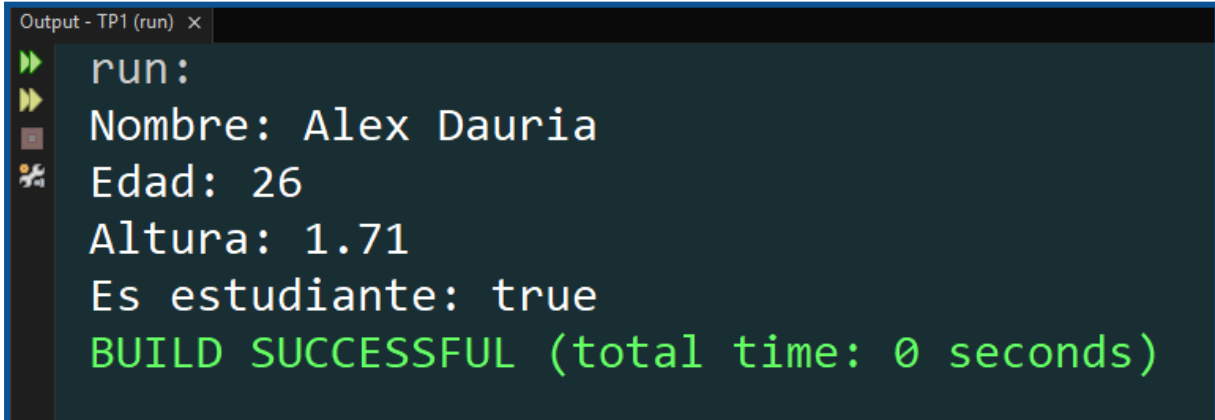
Ejercicio 2: Programa básico que imprime “¡Hola, Java!”

Resultado en la consola:



Ejercicio 3: Declarar variables e imprimir sus valores

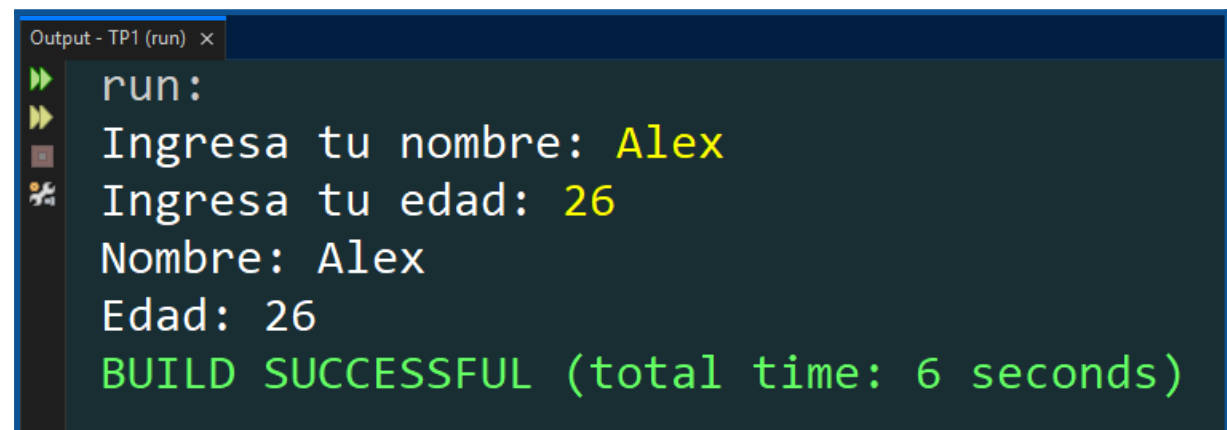
Resultado en la consola:



```
run:
Nombre: Alex Dauria
Edad: 26
Altura: 1.71
Es estudiante: true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio 4: Solicitar nombre y edad con Scanner

Resultado en la consola:



```
run:
Ingresa tu nombre: Alex
Ingresa tu edad: 26
Nombre: Alex
Edad: 26
BUILD SUCCESSFUL (total time: 6 seconds)
```

Ejercicio 5: Realizar operaciones aritméticas

Resultado en la consola:

```
Output - TP1 (run) x
run:
Ingresa el primer número entero: 10
Ingresa el segundo número entero: 2
Suma: 12
Resta: 8
Multiplicacion: 20
Division: 5.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

Ejercicio 6: Usar caracteres de escape para formato de salida

Resultado en la consola:

```
Output - TP1 (run) x
run:
Nombre: Juan Perez
Edad: 30 anos
Direccion: "Calle Falsa 123"
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio 7: Análisis de código - Expresiones vs Instrucciones

Código a analizar:

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

Línea 1: `int x = 10;` → **instrucción** (crea la variable x y le asigna un valor).
La **expresión** dentro es 10.

Línea 2: `x = x + 5;` → **instrucción** (actualiza el valor de x).
La **expresión** dentro es `x + 5`.

Línea 3: `System.out.println(x);` → **instrucción** (muestra el valor de x).
La **expresión** dentro es x.

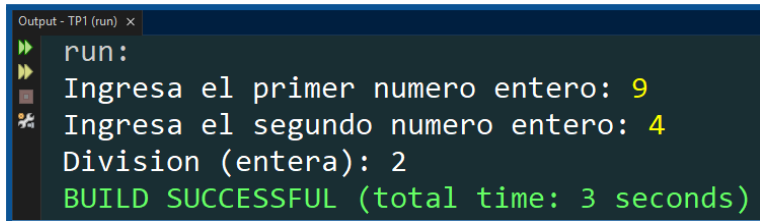
Diferencia entre expresiones e instrucciones:

Una **expresión** es una parte del código que da un valor, como 10, `x + 5` o x. Una **instrucción** es una acción completa que hace algo en el programa, como crear una variable, cambiar su valor o mostrarlo en pantalla. Es decir, las *expresiones* calculan valores y las *instrucciones* hacen que el programa haga cosas con esos valores.

Ejercicio 8: Manejar conversiones de tipo y división

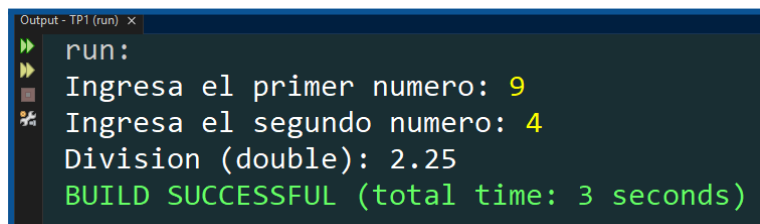
Resultados en consola:

División con int:



```
Output - TP1 (run) x
>> run:
>> Ingresar el primer numero entero: 9
>> Ingresar el segundo numero entero: 4
>> Division (entera): 2
BUILD SUCCESSFUL (total time: 3 seconds)
```

División con double:



```
Output - TP1 (run) x
>> run:
>> Ingresar el primer numero: 9
>> Ingresar el segundo numero: 4
>> Division (double): 2.25
BUILD SUCCESSFUL (total time: 3 seconds)
```

Conclusión

La división depende del tipo de dato usado. Con **int**, se pierde la parte decimal y solo queda la parte entera del resultado. Con **double**, se conserva la parte decimal, mostrando el resultado exacto. Por eso, si necesitamos precisión en los cálculos, debemos usar *double* en lugar de *int*.

Ejercicio 9: Corrección de código para leer nombre

Código corregido:

```
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
        System.out.println("Hola, " + nombre);
        scanner.close();
    }
}
```

Explicación del error

El error estaba en la línea `String nombre = scanner.nextInt();`

Se intentaba leer un `String` usando `nextInt()`, que solo sirve para números enteros. Esto genera un error si el usuario ingresa texto.

Solución

Se reemplazó `nextInt()` por `nextLine()`, que permite leer correctamente una cadena de texto ingresada por el usuario.

Ejercicio 10: Prueba de escritorio

Código:

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

Prueba de escritorio:

Línea	Código	a	b	resultado	Salida en consola
1	<code>int a = 5;</code>	5	-	-	-
2	<code>int b = 2;</code>	5	2	-	-
3	<code>int resultado = a / b;</code>	5	2	2	-
4	<code>System.out.println("Resultado: " + resultado);</code>	5	2	2	Resultado: 2

Explicación del resultado:

El valor de resultado es 2 porque a y b son enteros (int).

En Java, la división entre enteros descarta la parte decimal, por eso 5 / 2 no da 2.5, sino solo la parte entera: 2.

Link al repositorio en GitHub

<https://github.com/Alex-Dauria/UTN-TUPaD-P2>