

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.

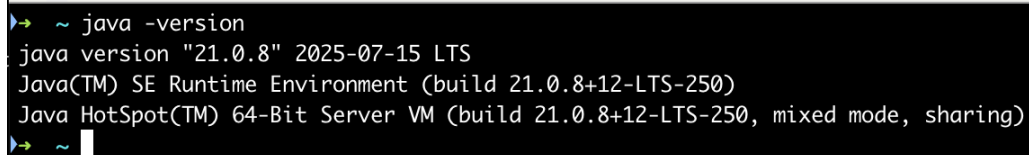
CASO PRÁCTICO

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

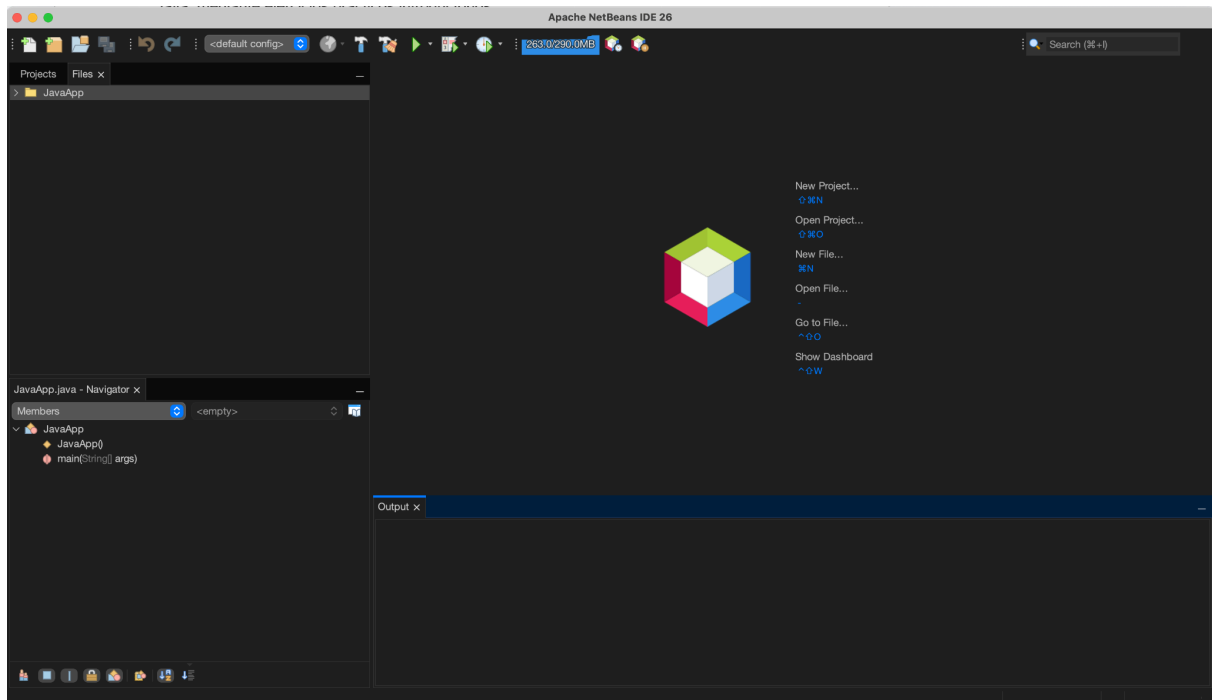
a. Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal: `java -version`



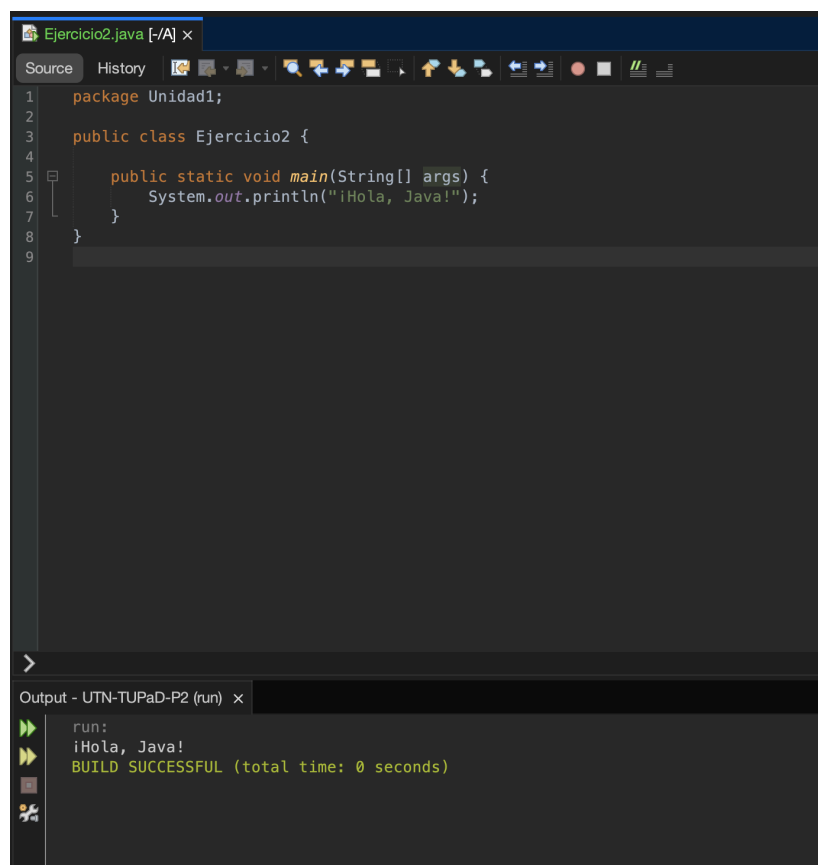
```
> ~ java -version
java version "21.0.8" 2025-07-15 LTS
Java(TM) SE Runtime Environment (build 21.0.8+12-LTS-250)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.8+12-LTS-250, mixed mode, sharing)
> ~
```

b. Abre NetBeans, crea un nuevo proyecto y configura el modo oscuro.

c. Toma una captura de pantalla del entorno configurado y agrégala a tu entrega.



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



3. 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()`.

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

5. 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.

6. 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()`.

7. 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`

La línea 1 es una instrucción ya que está definiendo una variable y asignándole un valor.

La línea 2 es una instrucción ya que está reasignando un valor nuevo a la variable.

La línea 3 es una instrucción ya que está imprimiendo por consola el valor de x.

8. Manejar conversiones de tipo y división en Java.

a. Escribe un programa que divida dos números enteros ingresados por el usuario.

```
run:
Ingresa un número:
3
Ingresa otro número:
4
El resultado es 0
BUILD SUCCESSFUL (total time: 5 seconds)
```

b. Modifica el código para usar `double` en lugar de `int` y compara los resultados.

```
run:
Ingresa un número:
3
Ingresa otro número:
4
El resultado es 0.75
BUILD SUCCESSFUL (total time: 2 seconds)
```

El resultado muestra que cuando se utiliza int para números enteros la división no devuelve un resultado correcto cuando el número tiene decimales, porque se pierde el resto. Para divisiones es mejor utilizar double ya que tiene más precisión.

9. 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);
    }
}
```

```
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(); // ERROR
        System.out.println("Hola, " + nombre);
    }
}
```

El error se encuentra en la línea: `String nombre = scanner.nextInt();`. No se estaba utilizando el método correcto, ya que `nextInt()` devuelve un número y no se puede guardar en una variable de tipo String.

Para este caso utilizar: `nextLine();`

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

```
1 public class PruebaEscritorio {
2     public static void main(String[] args) {
3         int a = 5;
4         int b = 2;
```

```
5  int resultado = a / b;  
6  System.out.println("Resultado: " + resultado);  
7  }  
8 }
```

Línea	a	b	resultado
1	<Sin definir>	<Sin definir>	<Sin definir>
2	<Sin definir>	<Sin definir>	<Sin definir>
3	5	<Sin definir>	<Sin definir>
4	5	2	<Sin definir>
5	5	2	2
6	5	2	2
7	5	2	2
8	5	2	2