

## 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

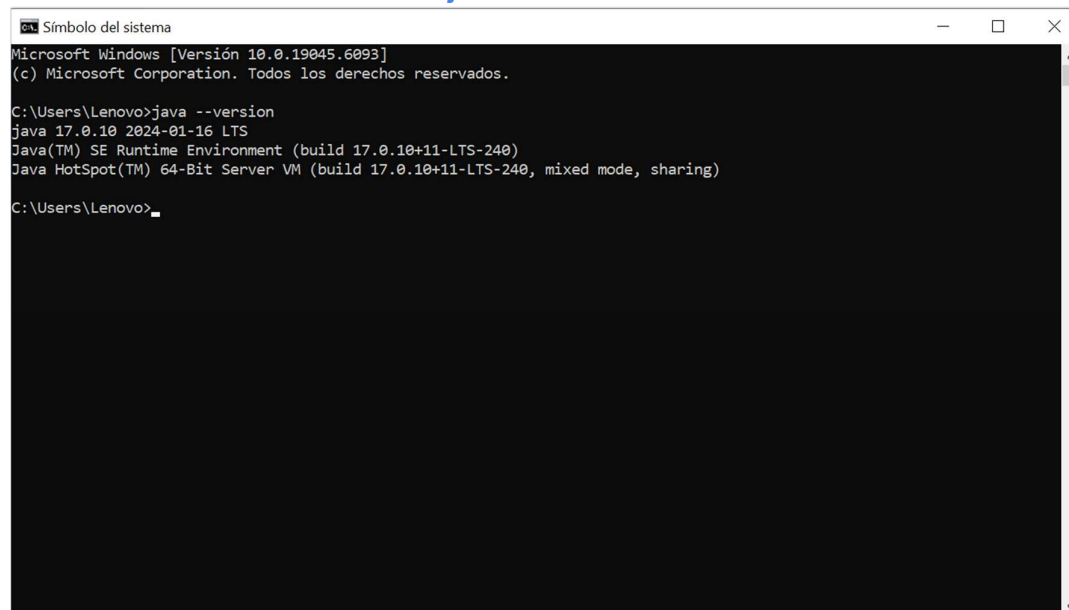
Concepto	Aplicación en el proyecto
Instalación y entorno	Almacenan el conjunto de países
Variables y tipos de datos	Representan los datos de cada país (nombre, población, superficie, etc.)
Entrada y salida	Separan las operaciones: carga, búsqueda, estadísticas, ordenamientos
Operadores aritméticos	Aplican 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
  - a. Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal: `java -version`



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.6093]
(c) Microsoft Corporation. Todos los derechos reservados.

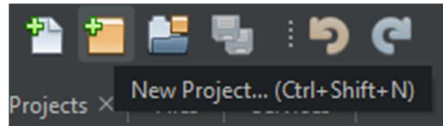
C:\Users\Lenovo>java --version
java 17.0.10 2024-01-16 LTS
Java(TM) SE Runtime Environment (build 17.0.10+11-LTS-240)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.10+11-LTS-240, mixed mode, sharing)

C:\Users\Lenovo>
```

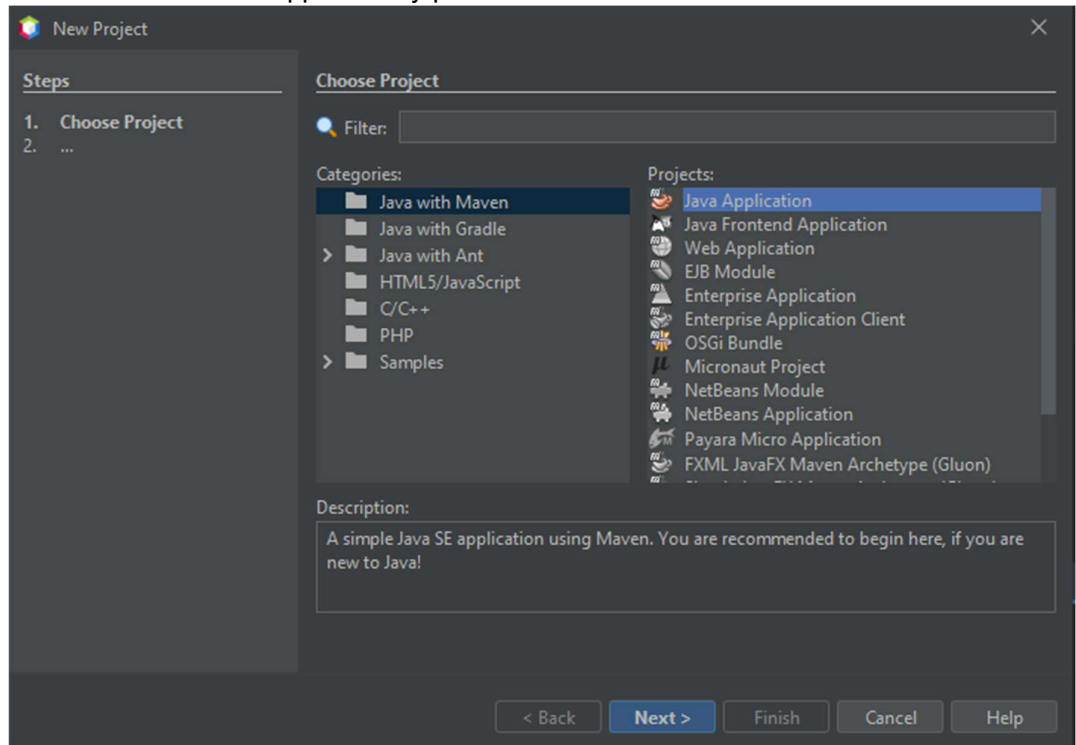
De esta manera puedo verificar que tengo instalado correctamente java y también puedo ver su versión.

- 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.

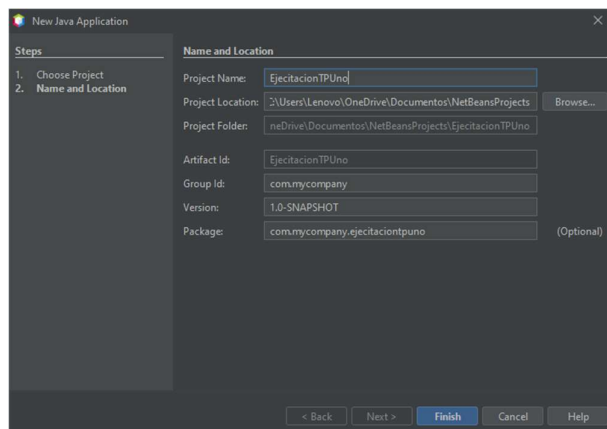
Paso 1: Creamos un nuevo proyecto



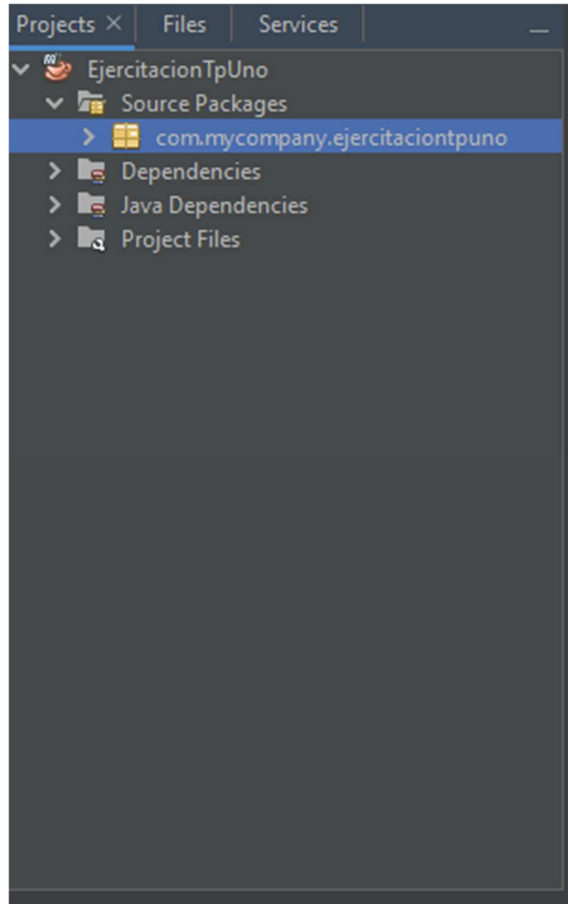
Seleccionamos Java Application y presionamos Next.



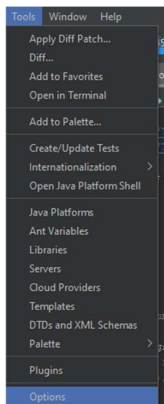
Le asignamos un nombre y presionamos finish.



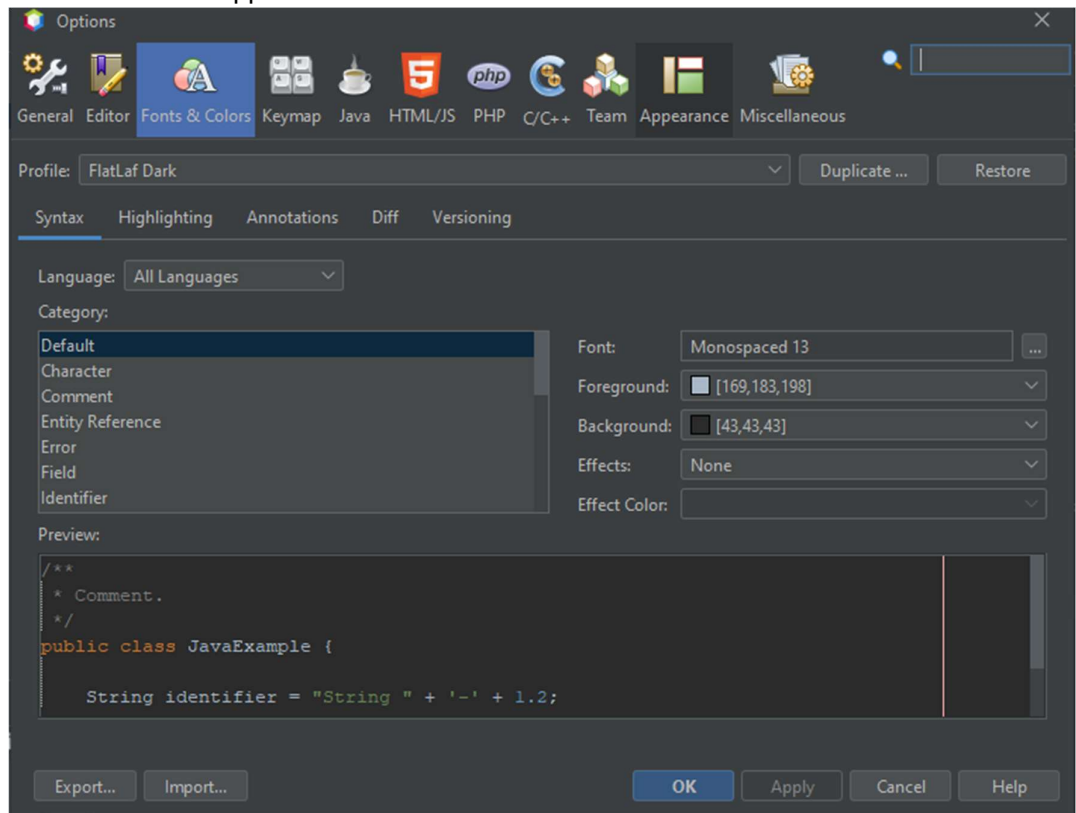
En el lado izquierdo de la pantalla podemos ver que nuestro proyecto se creó correctamente.



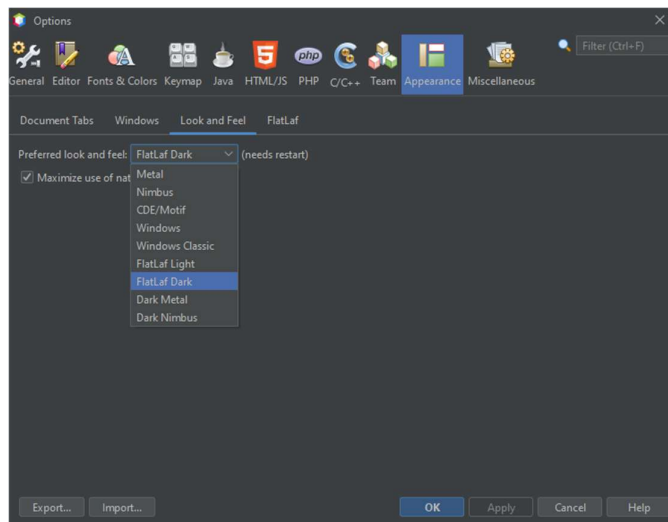
Paso 2: Configurar el modo oscuro.  
Presionamos en tools y luego en options.



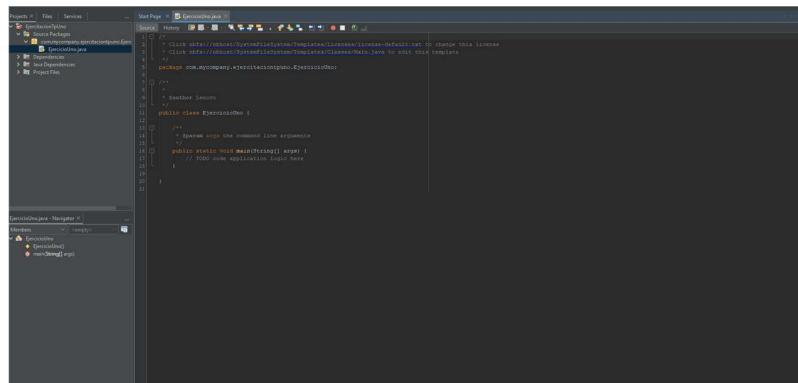
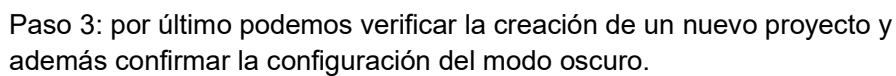
Presionamos en Appearance.



Luego en look and feel, seleccionamos el modo “FlatLaf Dark” y presionamos Apply, nos va a pedir reiniciar NetBeans para reflejar los cambios.

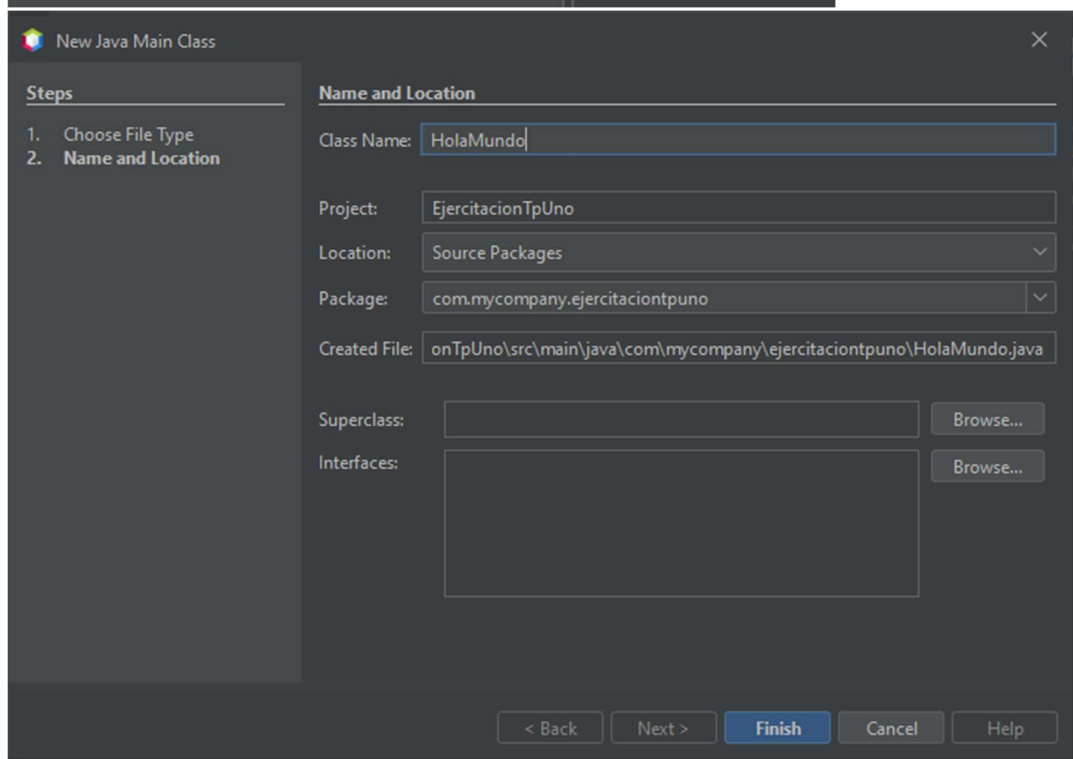
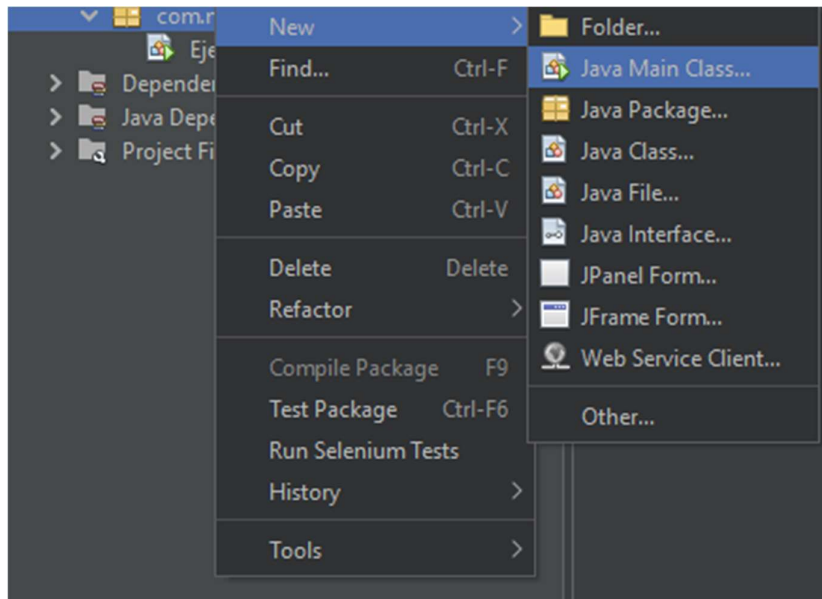


Entonces: Tools -> Options -> Fonts & Colors -> Profile -> FlatLaf Dark -> Apply.



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.

Paso 1: creamos una clase llamada HolaMundo



Paso 2: declaramos la instrucción para imprimir el mensaje “¡Hola, Java!”

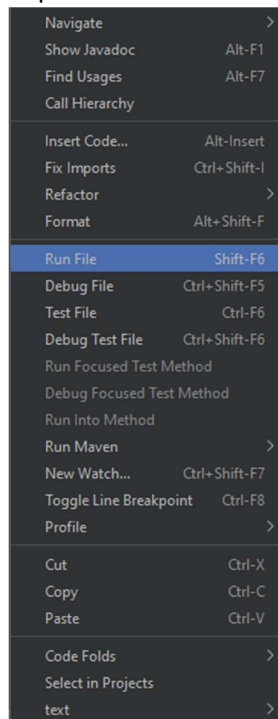
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package com.mycompany.ejercitaciontpuno;

/**
 *
 * @author Lenovo
 */
public class HolaMundo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Esta instruccion va a imprimir el mensaje "¡Hola, Java!" al ejecutar el programa
        System.out.println("¡Hola, Java!");
    }

}
```

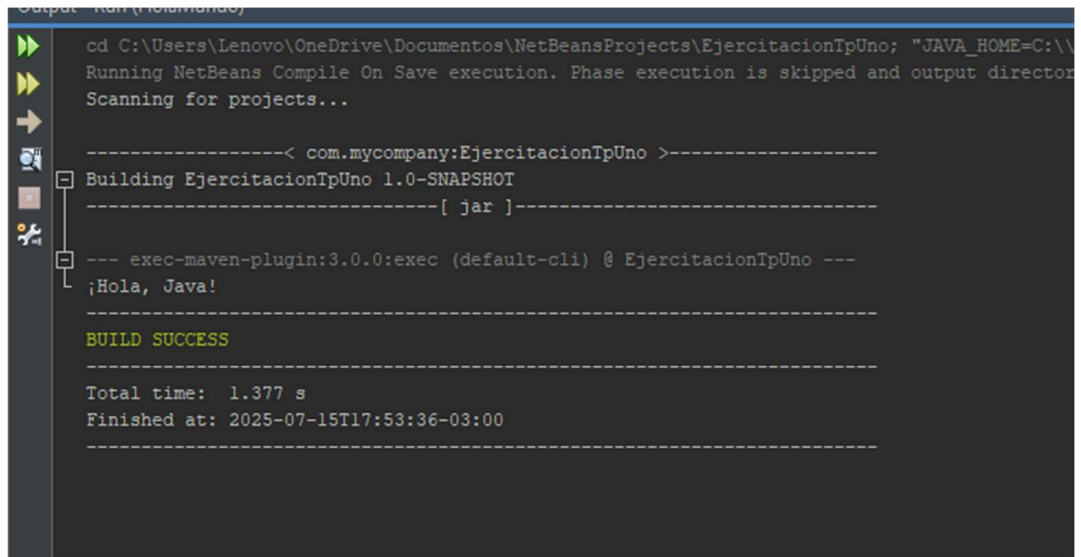
Paso 3: ejecutamos el programa y verificamos que nuestro mensaje se imprima correctamente en la consola



Click derecho sobre el archivo y presionamos Run

File





```
cd C:\Users\Lenovo\OneDrive\Documentos\NetBeansProjects\EjercitacionTpUno; "JAVA_HOME=C:\
Running NetBeans Compile On Save execution. Phase execution is skipped and output director
Scanning for projects...

-----< com.mycompany:EjercitacionTpUno >-----
Building EjercitacionTpUno 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpUno ---
¡Hola, Java!

BUILD SUCCESS

Total time: 1.377 s
Finished at: 2025-07-15T17:53:36-03:00
```

Verificamos que se logra mostrar el mensaje por pantalla.

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()`.

Paso 1: Declarar e imprimir por consola las variables nombre, edad, altura, estudiante

```
package com.mycompany.ejercitaciontpuno;

public class EjercicioTres {

    public static void main(String[] args) {
        // Declarmos variables nombre, edad, altura, estudiante
        String nombre = "Jeronimo Cortez";
        int edad = 22;
        double altura = 1.83;
        boolean estudiante = false;

        // Imprimimos las variables declaradas por consola
        System.out.println(nombre);
        System.out.println(edad);
        System.out.println(altura);
        System.out.println(estudiante);
    }
}
```

Paso 2: Ejecutar el programa

```
cd C:\Users\Lenovo\OneDrive\Documentos\NetBeansProjects\EjercitacionTpUno; "JAVA_HOME=C:
Running NetBeans Compile On Save execution. Phase execution is skipped and output direct
Scanning for projects...

-----< com.mycompany:EjercitacionTpUno >-----
Building EjercitacionTpUno 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpUno ---
Jeronimo Cortez
22
1.83
false

BUILD SUCCESS

Total time: 1.456 s
Finished at: 2025-07-15T16:39:12-03:00
|
```

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.

Paso 1: Declaramos variables, leemos los datos del usuario y los mostramos por pantalla

```
package com.mycompany.ejercitaciontpuno;

import java.util.Scanner;

public class EjercicioCuatro {

    public static void main(String[] args) {
        // Declaramos input de tipo Scanner para poder leer datos del usuario
        Scanner input = new Scanner(System.in);

        // Declaramos variables
        String nombre;
        int edad;

        // Pedimos al usuario ingresar los datos
        System.out.println("Ingrese su nombre");
        nombre = input.nextLine();
        System.out.println("Ingrese su edad");
        edad = Integer.parseInt(input.nextLine());

        // Mostramos los datos por pantalla
        System.out.println("Nombre: " + nombre);
        System.out.println("Edad: " + edad);
    }
}
```

Paso 2: ejecutamos el programa

```
cd C:\Users\Lenovo\OneDrive\Documentos\NetBeansProjects\EjercitacionIpUno; "JAVA_HOME=C:\Program Files\Java\jdk-11.0.2\bin" java -jar EjercitacionIpUno.jar
Running NetBeans Compile On Save execution. Phase execution is skipped and output direct
Scanning for projects...

-----< com.mycompany:EjercitacionIpUno >-----
Building EjercitacionIpUno 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionIpUno ---
Ingrese su nombre
Jeronimo
Ingrese su edad
22
Nombre: Jeronimo
Edad: 22

BUILD SUCCESS

Total time: 6.131 s
Finished at: 2025-07-15T17:15:11-03:00
```

5. Escribe un programa que solicite dos números enteros y realice las siguientes operaciones:
- Suma
  - Resta
  - Multiplicación
  - División

Muestra los resultados en la consola.

Paso 1: Declaramos variables, leemos los datos del usuario y mostramos los resultados por pantalla

```
package com.mycompany.ejercitaciontpuno;

import java.util.Scanner;

public class EjercicioCinco {

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

        // Declaramos variables num1, num2
        int num1;
        int num2;

        // Leemos los datos del usuario
        System.out.println("Ingrese dos numeros enteros");
        System.out.println("Numero 1: ");
        num1 = Integer.parseInt(input.nextLine());
        System.out.println("Numero 2: ");
        num2 = Integer.parseInt(input.nextLine());

        // Mostramos los datos por pantalla
        System.out.println(num1 + " + " + num2 + " = " + (num1 + num2));
        System.out.println(num1 + " - " + num2 + " = " + (num1 - num2));
        System.out.println(num1 + " * " + num2 + " = " + (num1 * num2));
        System.out.println(num1 + " / " + num2 + " = " + (num1 / num2));
    }
}
```

Paso 2: ejecutamos el programa

```
out - Run (EjercicioCinco)

cd C:\Users\Lenovo\OneDrive\Documentos\NetBeansProjects\EjercitacionTpUno; "JAVA_HOME=C:\
Running NetBeans Compile On Save execution. Phase execution is skipped and output directo
Scanning for projects...

-----< com.mycompany:EjercitacionTpUno >-----
Building EjercitacionTpUno 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpUno ---
Ingrese dos numeros enteros
Numero 1:
4
Numero 2:
2
4 + 2 = 6
4 - 2 = 2
4 * 2 = 8
4 / 2 = 2

BUILD SUCCESS

Total time: 3.482 s
Finished at: 2025-07-15T17:23:10-03:00
```

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()`.

Paso 1: escribimos los datos `System.out.println()` utilizando `\n` para escribir el nombre de la calle entre comillas y `\n` para poder realizar los saltos de línea

```
package com.mycompany.ejercitaciontpuno;

public class EjercicioSeis {

    public static void main(String[] args) {
        System.out.println("Nombre: Juan Pérez \n" +
                           "Edad: 30 años \n" +
                           "Dirección: \"Calle Falsa 123\"");
    }
}
```

Paso 2: ejecutar el programa

```
Output - Run (EjercicioSeis)

cd C:\Users\Lenovo\OneDrive\Documentos\NetBeansProjects\EjercitacionTpUno; "JAVA_F
Running NetBeans Compile On Save execution. Phase execution is skipped and output
Scanning for projects...

-----< com.mycompany:EjercitacionTpUno >-----
Building EjercitacionTpUno 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpUno ---
Nombre: Juan Pérez
Edad: 30 años
Dirección: "Calle Falsa 123"

BUILD SUCCESS

Total time: 1.464 s
Finished at: 2025-07-15T17:27:47-03:00
```

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 y la línea 2 son instrucciones de asignación, ya que le asignan un valor a una variable. Ambas contienen expresiones en el lado derecho del igual, como 10 o  $x + 5$ , que se evalúan para luego asignarse. La línea 3 también es una instrucción, ya que ejecuta una acción (mostrar en pantalla). En resumen, una expresión produce un valor; una instrucción realiza una acción en el programa.

8. 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.



Paso 1: declaramos variables, le pedimos los números al usuario y mostramos el resultado de la división convertido en double por pantalla

```
package com.mycompany.ejercitaciontpuno;

import java.util.Scanner;

public class Ejercicio7 {

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

        // Declaramos variables
        int num1, num2;

        // Pedimos los datos al usuario
        System.out.println("Ingrese dos numeros enteros");
        System.out.println("Numero 1: ");
        num1 = Integer.parseInt(input.nextLine());
        System.out.println("Numero 2: ");
        num2 = Integer.parseInt(input.nextLine());

        // Mostramos el resultado por pantalla
        System.out.println(num1 + " / " + num2 + " = " + ((double) num1 / num2));
    }
}
```

Paso 2: ejecutamos el programa

```
cd C:\Users\Lenovo\OneDrive\Documentos\NetBeansProjects\EjercitacionTpUno; "JAVA_HOME=C:\P...
Running NetBeans Compile On Save execution. Phase execution is skipped and output directorie
Scanning for projects...

-----< com.mycompany:EjercitacionTpUno >-----
Building EjercitacionTpUno 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpUno ---
Ingrese dos numeros enteros
Numero 1:
1
Numero 2:
2
1 / 2 = 0.5

BUILD SUCCESS

Total time: 3.731 s
Finished at: 2025-07-15T17:37:41-03:00
```

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);  
}  
}
```

```
package com.mycompany.ejercitaciontpuno;  
  
import java.util.Scanner;  
  
public class EjercicioNueve {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Ingresa tu nombre: ");  
        String nombre = scanner.nextLine(); // Línea corregida  
        System.out.println("Hola, " + nombre);  
    }  
}
```

El error en el código original era que se usaba `scanner.nextInt()` para leer un nombre, pero este método solo sirve para ingresar números enteros. Como queríamos ingresar texto (un nombre), lo correcto es usar `scanner.nextLine()`, que permite leer toda una línea de texto. Al hacer este cambio, el programa funciona correctamente y muestra el saludo con el nombre ingresado.



10. 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);  
}  
}
```

Línea de código	Valor de a	Valor de b	Resultado
Int a = 5	5	-	-
Int b = 2	5	2	-
Int resultado = a / b	5	2	2
System.out.println("Resultado: " + resultado);	-	-	Muestra: "Resultado: 2"

El valor de resultado es 2 ya que el programa realiza una división entre 5 y 2 y esto nos da un número decimal "2.5" pero como nuestra variable está declarada como int (entero) el resultado que devuelve es 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.