

PROGRAMACIÓN II

Trabajo Práctico 1: Introducción a Java

Alumna: Astrid Ayelen Añazco

Link repositorio GitHub: [Link](#)

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:

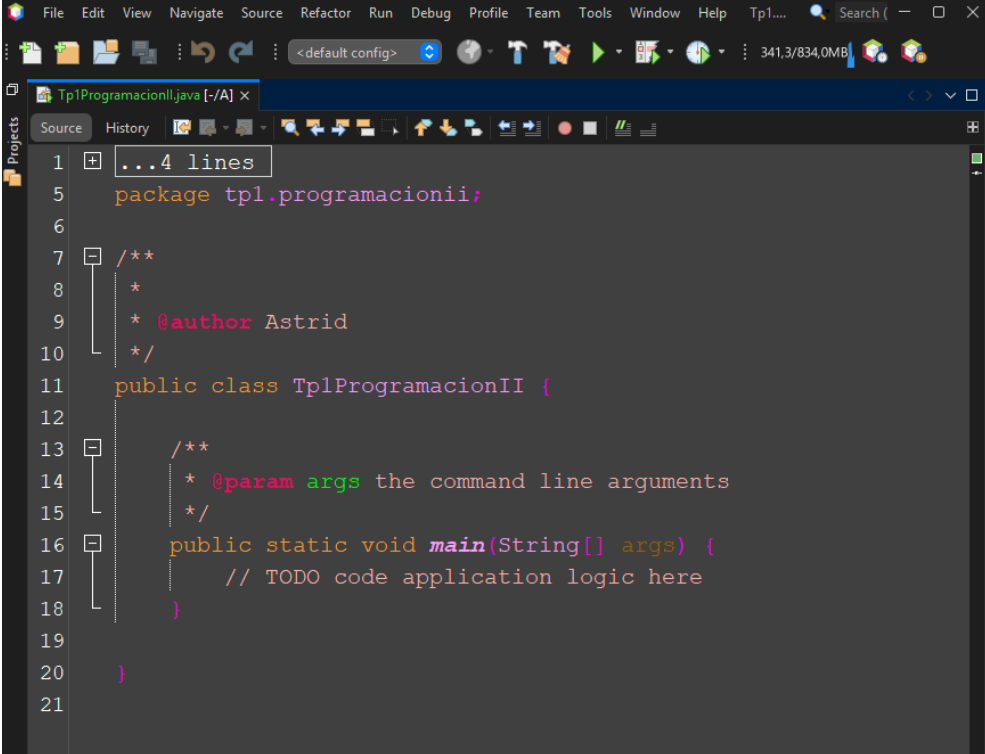
- Configurar correctamente el entorno de desarrollo (Java JDK y NetBeans).
- Crear programas básicos que imprimen mensajes en consola.
- Declarar variables de distintos tipos y manipular sus valores. • Leer datos ingresados por el usuario usando **Scanner**.
- Realizar operaciones aritméticas básicas.
- 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**

```
PS C:\Users\Astrid> java -version
java version "24.0.2" 2025-07-15
Java(TM) SE Runtime Environment (build 24.0.2+12-54)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode, sharing)
```

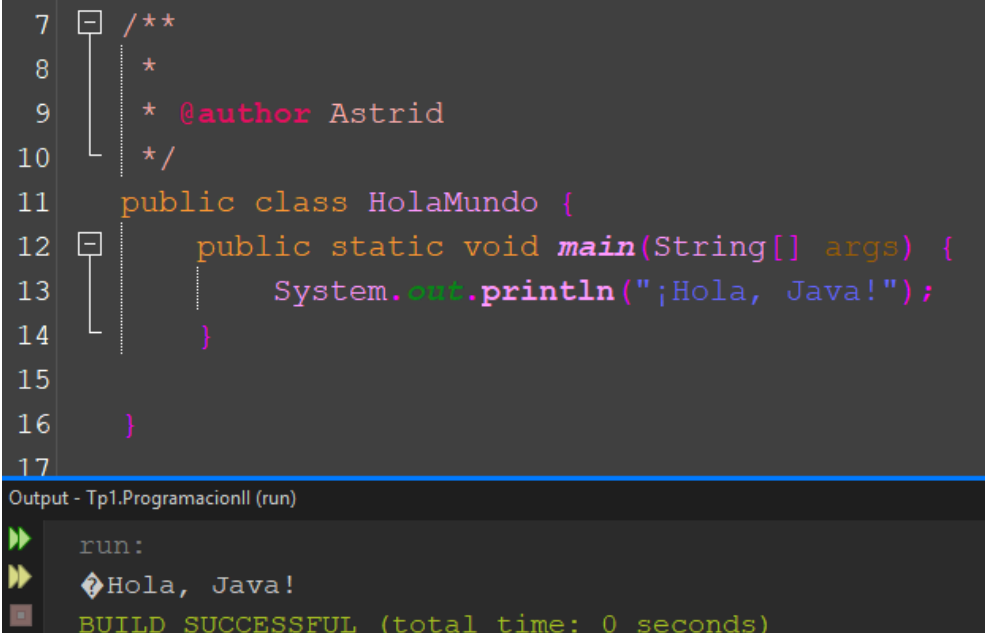
- b. Abre NetBeans, crea un nuevo proyecto y configura el modo oscuro.
- c. Toma una captura de pantalla del entorno configurado y agrégala a tu entrega.



```
1  ...4 lines
5  package tp1.programacionii;
6
7  /**
8   *
9   * @author Astrid
10  */
11  public class Tp1ProgramacionII {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18      }
19
20  }
21
```

Hol

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.



```
7  /**
8   *
9   * @author Astrid
10  */
11  public class HolaMundo {
12      public static void main(String[] args) {
13          System.out.println("¡Hola, Java!");
14      }
15
16  }
17
```

Output - Tp1.ProgramacionII (run)

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

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()`.

```
11 public class Tp1ProgramacionII {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         String nombre = "Astrid";
18         int edad = 27;
19         double altura = 1.60;
20         boolean estudiante = true;
21
22         System.out.println("Mi nombre es: " + nombre);
23         System.out.println("Tengo " + edad + " años");
24         System.out.println("Mi altura en metros es: " + altura);
25         System.out.println("Soy estudiante? " + estudiante);
26
27     }
28 }
```

Output - Tp1.ProgramacionII (run)

```
>> run:
>> Mi nombre es: Astrid
>> Tengo 27 años
>> Mi altura en metros es: 1.6
>> Soy estudiante? true
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

```
13 public class Tp1ProgramacionII {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner input = new Scanner(System.in);
20
21         System.out.print("Ingrese su nombre: ");
22         String nombre = input.nextLine();
23
24         System.out.print("Ingrese su edad: ");
25         int edad = Integer.parseInt(input.nextLine());
26
27         System.out.println("Nombre: " + nombre);
28         System.out.println("Edad: " + edad);
29     }
30 }
```

Output - Tp1.ProgramacionII (run)

```
>> run:
>> Ingrese su nombre: Pepe
>> Ingrese su edad: 55
>> Nombre: Pepe
>> Edad: 55
BUILD SUCCESSFUL (total time: 12 seconds)
```

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

Muestra los resultados en la consola.

```
Source History
19 Scanner input = new Scanner(System.in);
20 int num1, num2, suma, resta, multiplicacion, division;
21
22 System.out.print("Ingrese el primer numero entero: ");
23 num1 = input.nextInt();
24
25 System.out.print("Ingrese el segundo numero entero: ");
26 num2 = input.nextInt();
27
28 suma = num1 + num2;
29 resta = num1 - num2;
30 multiplicacion = num1 * num2;
31 division = num1 / num2;
32
33 System.out.println("Suma : " + suma);
34 System.out.println("Resta : " + resta);
35 System.out.println("Multiplicacion : " + multiplicacion);
36 System.out.println("Division : " + division);
37
38
Output - Tp1.ProgramacionII (run)
run:
Ingrese el primer numero entero: 30
Ingrese el segundo numero entero: 5
Suma : 35
Resta : 25
Multiplicacion : 150
Division : 6
BUILD SUCCESSFUL (total time: 6 seconds)
```

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()**.

**System.out.println("Nombre: Juan Perez \nEdad: 30 años
\nDireccion: \"Calle Falsa 123\"");**

```
16 //
17 public static void main(String[] args) {
18     System.out.println("Nombre: Juan Perez \nEdad: 30 años \nDireccion: \"Calle Falsa 123\"");
19 }
Output
run:
Nombre: Juan Perez
Edad: 30 años
Direccion: "Calle Falsa 123"
BUILD SUCCESSFUL (total time: 0 seconds)
```

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`

Una instrucción es una línea completa de código que realiza una acción, como declarar una variable (Línea 1), asignarle un valor (Línea 1 y Línea 2), o llamar un método (Línea 3).

Una expresión es una parte del código que produce un valor, como una operación matemática ($x + 5$) o un literal (10). Las expresiones pueden usarse dentro de instrucciones, pero las instrucciones son las que organizan la lógica y controlan el flujo del programa. La expresión por sí sola en Java, va a dar un error en el compilador.

8. Manejar conversiones de tipo y división en Java.

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

```
19 public static void main(String[] args) {
20     Scanner input = new Scanner(System.in);
21     int num1, num2, division;
22     System.out.print("Ingrese el primer numero entero: ");
23     num1 = input.nextInt();
24     System.out.print("Ingrese el segundo numero entero: ");
25     num2 = input.nextInt();
26
27     division = num1 / num2;
28 }
```

Output

Astrid - C:\Users\Astrid X Tp1.ProgramacionII (run) X

run:
Ingrese el primer numero entero: 100
Ingrese el segundo numero entero: 3
El resultado de la division es: 33
BUILD SUCCESSFUL (total time: 9 seconds)

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

```
19 public static void main(String[] args) {
20     Scanner input = new Scanner(System.in);
21     double num1, num2, division;
22     System.out.print("Ingrese el primer numero entero: ");
23     num1 = input.nextDouble();
24     System.out.print("Ingrese el segundo numero entero: ");
25     num2 = input.nextDouble();
26
27     division = num1 / num2;
28 }
```

Output

Astrid - C:\Users\Astrid X Tp1.ProgramacionII (run) X

run:
Ingrese el primer numero entero: 100
Ingrese el segundo numero entero: 3
El resultado de la division es: 33.333333333333336
BUILD SUCCESSFUL (total time: 4 seconds)

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);  
  
}  
  
}
```

El error dice que hay una “incompatibilidad de tipos”, se declaró que nombre es String y cuando se quiere guardar el dato de entrada lo hace con “.nextInt()”, que corresponde para un numero entero. En este caso se debe usar:

String nombre = scanner.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);  
}  
}
```

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

El valor de resultado es 2, porque su tipo es int (entero) entonces descarta el numero decimal de la división y solo coloca el valor entero. Para que el valor final sea correcto hay que definir el resultado y una o ambas variables como double.

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.