

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

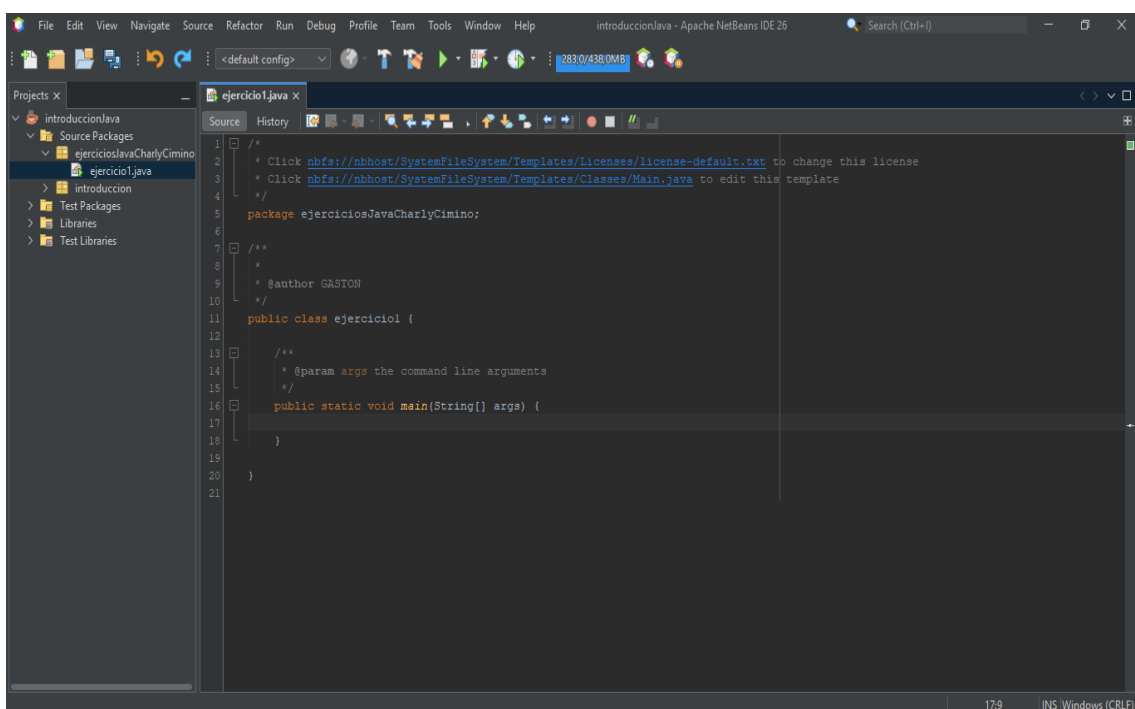
CASO PRÁCTICO

El trabajo consiste en resolver una serie de ejercicios introductorios en Java que permitan:

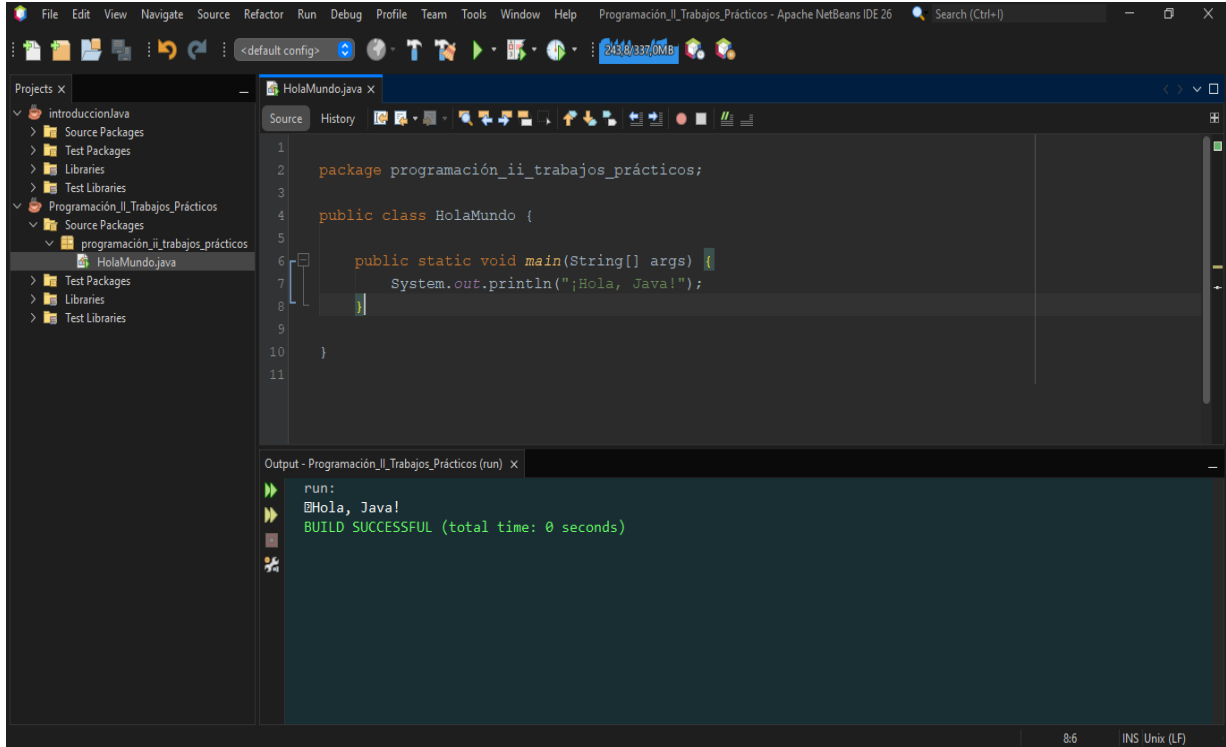
- Configurar correctamente el entorno de desarrollo (Java JDK y NetBeans).
- Crear programas básicos que imprimen mensajes en consola.
- Declarar variables de distintos tipos y manipular sus valores.
- Leer datos ingresados por el usuario usando **Scanner**.
- Realizar operaciones aritméticas básicas.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



```
1 package programación_ii_trabajos_prácticos;
2
3
4 public class HolaMundo {
5
6     public static void main(String[] args) {
7         System.out.println("¡Hola, Java!");
8     }
9
10 }
11
```

Output - Programación_II_Trabajos_Prácticos (run) x

```
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 estudianteImprime 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

- **Análisis:**

- **Línea 1: int x = 10;**

Es una **instrucción** que declara una variable y le asigna un valor.

El valor 10 es una **expresión**, que se evalúa y se asigna a x.

- **Línea 2: x = x + 5;**

Es una **instrucción**: asigna el resultado de una operación a x.

La **expresión** es x + 5, donde se evalúa primero y devuelve un valor.

- **Línea 3: System.out.println(x);**

Es una **instrucción**, que ejecuta una acción (mostrar en consola).

La variable x es una **expresión**, que devuelve el valor que se imprimirá.

En Java, una expresión es cualquier fragmento de código que devuelve un valor, como x + 5 o 10. Una instrucción es una unidad completa de código que realiza una acción, como declarar una variable, asignar un valor o imprimir algo. Las instrucciones pueden contener expresiones, pero su propósito es ejecutar una acción, no solo devolver un valor.



8. Manejar conversiones de tipo y división en Java.
 - a. Escribe un programa que divida dos números enteros ingresados por el usuario.
 - b. Modifica el código para usar **double** en lugar de **int** y compara los resultados.
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);
    }
}
```

- 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);
    }
}
```

¿Cuál era el error?

En esta línea del código original: `String nombre = scanner.nextInt(); // ERROR`

Se está intentando guardar un número entero (int) en una variable de tipo `String`, lo cual provoca un error de compilación porque los tipos de datos no coinciden.

¿Cómo se solucionó?

Se reemplazó `scanner.nextInt()` por `scanner.nextLine()`, ya que:

- `nextInt()` → lee números enteros.
- `nextLine()` → lee líneas completas de texto, que es lo que se necesita para leer el nombre de una persona.

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	a	b	resultado
1	Sin inicializar	Sin inicializar	Sin inicializar
2	Sin inicializar	Sin inicializar	Sin inicializar
3	Sin inicializar	Sin inicializar	Sin inicializar
4	5	Sin inicializar	Sin inicializar
5	5	2	Sin inicializar
6	5	2	2

El resultado de la división en este ejercicio es 2 (sin decimales).

- ¿Por qué es 2 y no 2.5?

Porque a y b son enteros (int), y en Java cuando dividimos dos enteros, el resultado también es un entero. La parte decimal se descarta, no se redondea.

Si quisiéramos lograr un resultado con decimales, al menos uno de los operandos debe ser double:

```
double resultado = (double) a / b;
```

Esto daría como resultado: 2.5

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.