

Marcelo Oviedo

## PROGRAMACIÓN II

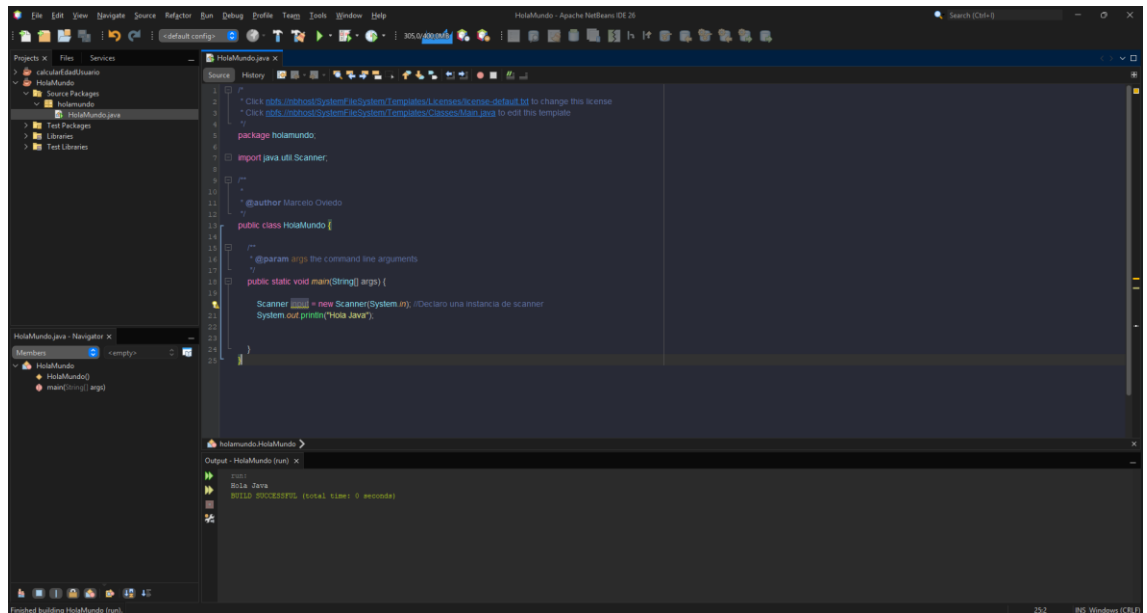
### Trabajo Práctico 1: Introducción a Java

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.

The first screenshot shows a Windows Command Prompt window titled 'Símbolo del sistema'. It displays the output of the command `java -version`, which is: `java version "24.0.2" 2025-07-15`, `Java(TM) SE Runtime Environment (build 24.0.2+12-54)`, and `Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode, sharing)`. The second screenshot shows the NetBeans IDE interface with a dark theme. The 'Projects' pane on the left shows a project named 'HolaMundo'. The 'Sources' pane shows the package 'holamundo' and the class 'HolaMundo'. The 'Output' pane at the bottom shows the output of the program: `print: Hola Java` and `BUILD SUCCESSFUL (total time: 0 seconds)`.

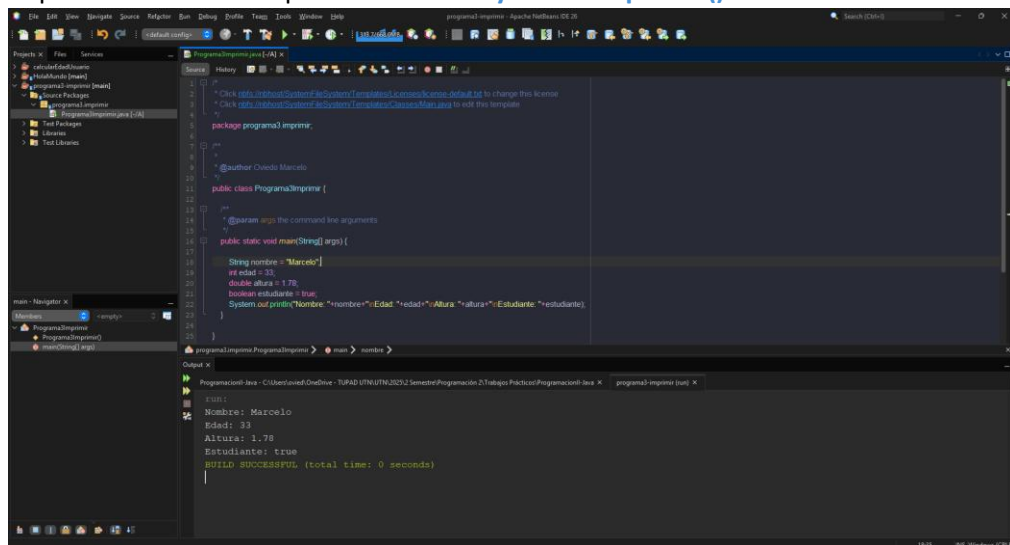
Marcelo Oviedo

2. Escribir y ejecutar un programa básico en Java.
  - a. Creá una clase llamada HolaMundo.
  - b. Escrib 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()`.



Marcelo Oviedo

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.

```

7 import java.util.Scanner;
8
9 /**
10  *
11  * @author oviedo
12  */
13 public class SolicitarDatosAlUsuario {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner scanner = new Scanner(System.in);
20         String nombre;
21         int edad;
22         System.out.println("Ingrese su nombre:");
23         nombre = scanner.nextLine();
24         System.out.println("Ingrese su edad:");
25         edad = Integer.parseInt(scanner.nextLine());
26         System.out.println("Su edad nombre es " + nombre + " y su edad es de " + edad + " años.");
27     }
28 }
29
30

```

Output - solicitarDatosAlUsuario (run) x

```

run:
Ingrese su nombre:
Marcelo
Ingrese su edad:
33
Su edad nombre es Marcelo y su edad es de 33 años
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.

```

14 /**
15  *
16  * @param args the command line arguments
17  */
18 public static void main(String[] args) {
19     //Inicializo el Scanner
20     Scanner scanner = new Scanner(System.in);
21     int num1, num2;
22     System.out.println("Bienvenido el programa para realizar las operaciones básicas entre 2 números");
23     System.out.println("Ingrese el primer número:");
24     num1 = Integer.parseInt(scanner.nextLine());
25     System.out.println("Ingrese el segundo número:");
26     num2 = Integer.parseInt(scanner.nextLine());
27
28     //Operaciones
29     System.out.println("Resultados:");
30     System.out.println("Suma: " + (num1 + num2));
31     System.out.println("Resta: " + (num1 - num2));
32     System.out.println("Multiplicación: " + (num1 * num2));
33
34     if (num2 != 0) {
35         System.out.println("División: " + ((double) num1 / num2)); //Es importante castear 1 número a double para que la división sea correcta
36     } else {
37         System.out.println("División: No es posible dividir por cero.");
38     }
39 }
40

```

Output - OperacionesMatematicas (run) x

```

Bienvenido el programa para realizar las operaciones básicas entre 2 números
Ingrese el primer número:
10
Ingrese el segundo número:
5
Resultados:
Suma: 15
Resta: 5
Multiplicación: 50
División: 2.0
BUILD SUCCESSFUL (total time: 5 seconds)

```

Marcelo Oviedo

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().

The screenshot shows an IDE window with a Java file named `MostrarMensajeConEscape.java`. The code is as follows:

```
1 //  
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  
3 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template  
4  
5 package mostrarmensajeconescape;  
6  
7 /**  
8  *  
9  * @author ovi  
10  */  
11 public class MostrarMensajeConEscape {  
12  
13     /**  
14     * @param args the command line arguments  
15     */  
16     public static void main(String[] args) {  
17         System.out.println("Nombre: Juan Perez\nEdad: 30 años\nDirección: \"Calle Falsa 123\"");  
18     }  
19  
20 }  
21
```

The IDE also shows a console output window with the following text:

```
run:  
Nombre: Juan Perez  
Edad: 30 años  
Dirección: "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` → Es una instrucción, ya que hay una declaración de variable y una asignación de valor a esa misma variables declarada.

`x = x + 5; // Línea 2` → Es una instrucción, ya que esta asignado un valor a la variable x. Dentro de esta instrucción hay una expresión, que corresponde a la suma de x+5.

`System.out.println(x); // Línea 3` → Es una instrucción, ya que es la llamada a un método. Pero dentro de la misma hay una expresión que es la variable en sí que viaja como parámetro.

Una expresión es cualquier fragmento de código que se evalúa un produce un valor, por ejemplo 10 o X+5. Pero una instrucción es una orden completa que ejecuta un programa (declarar una variable, asignar un valor, llamar a un método, etc).

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

Marcelo Oviedo

resultados. Al realizar la división e ingreso como double, se obtienen resultados más precisos, ya que si la división no es entera muestra el número incluyendo la parte decimal. Aunque si es entero el resultado, muestra igualmente la parte decimal.

```

public class DivisionDeNumeros {
    public static void main(String[] args) {
        //Declaración de variables y scanner
        int num1, num2;
        double num1d, num2d;
        Scanner scanner = new Scanner(System.in);
        //Se solicitan los valores
        System.out.println("Ingrese el primer número Int.");
        num1 = Integer.parseInt(scanner.nextLine());
        System.out.println("Ingrese el segundo número Int.");
        num2 = Integer.parseInt(scanner.nextLine());

        //Se divide con int
        System.out.println("División de " + num1 + "/" + num2 + " como int = " + num1 / num2);

        //Se solicitan los valores doubles
        System.out.println("Ingrese el primer número Int.");
        num1d = Double.parseDouble(scanner.nextLine());
        System.out.println("Ingrese el segundo número Int.");
        num2d = Double.parseDouble(scanner.nextLine());

        //Se divide con double
        System.out.println("División de " + num1d + "/" + num2d + " como double = " + (double) num1 / num2);
    }
}

```

Output - Run (DivisionDeNumeros) x

```

--- exec:3.1.0:exec (default-cli) @ DivisionDeNumeros ---
Ingrese el primer número Int:
80
Ingrese el segundo número Int:
3
División de 80/3 como int = 26
Ingrese el primer número Int:
80
Ingrese el segundo número Int:
3
División de 80/3 como double = 26.666666666666668

```

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);

    }

}

```

Se resuelve cambiando el nextInt (que se usa para recibir un entero por consola, por un nextLine, que recibe un string por consola.

```

import java.util.Scanner;

public class ErrorEjemplo {

    public static void main(String[] args) {

```

Marcelo Oviedo

```
Scanner scanner = new Scanner(System.in);

System.out.print("Ingresa tu nombre: ");

String nombre = scanner.nextLine(); // ERROR

System.out.println("Hola, " + nombre);

}

}
```

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	Variable A	Variable B	Variable resultado	Salida en consola
1	5	Sin declarar	Sin declarar	-
2	5	2	Sin declarar	-
3	5	2	2	-
4	5	2	2	Resultado: 2

El resultado final es 2, ya que al dividir 2 ints siempre el resultado va a ser sin parte decimal, solo mostrando la parte entera.