

TRABAJO PRÁCTICO 1 - INTRODUCCIÓN A JAVA

Alumnos: Montero Morinigo, Ramón Armando

Materia: Programación II

Profesor/a:

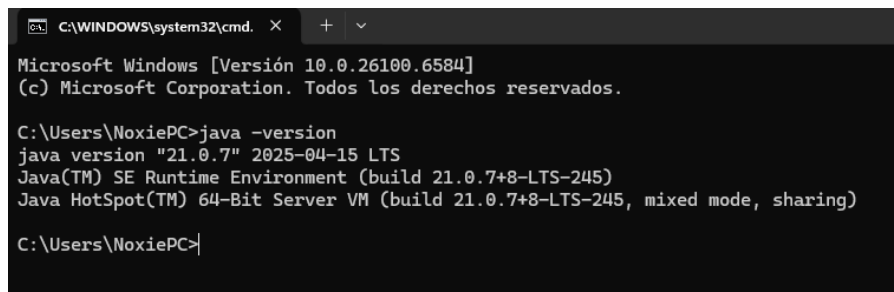
Tutor:

COMISIÓN 11

1. Verificar que tienes instalado Java JDK y NetBeans

- Confirma que tienes Java JDK instalado ejecutando el siguiente comando en la terminal:
`java -version`
- Abre NetBeans, crea un nuevo proyecto y configura el modo oscuro.
- Toma una captura de pantalla del entorno configurado y agrégala a tu entrega.

a.

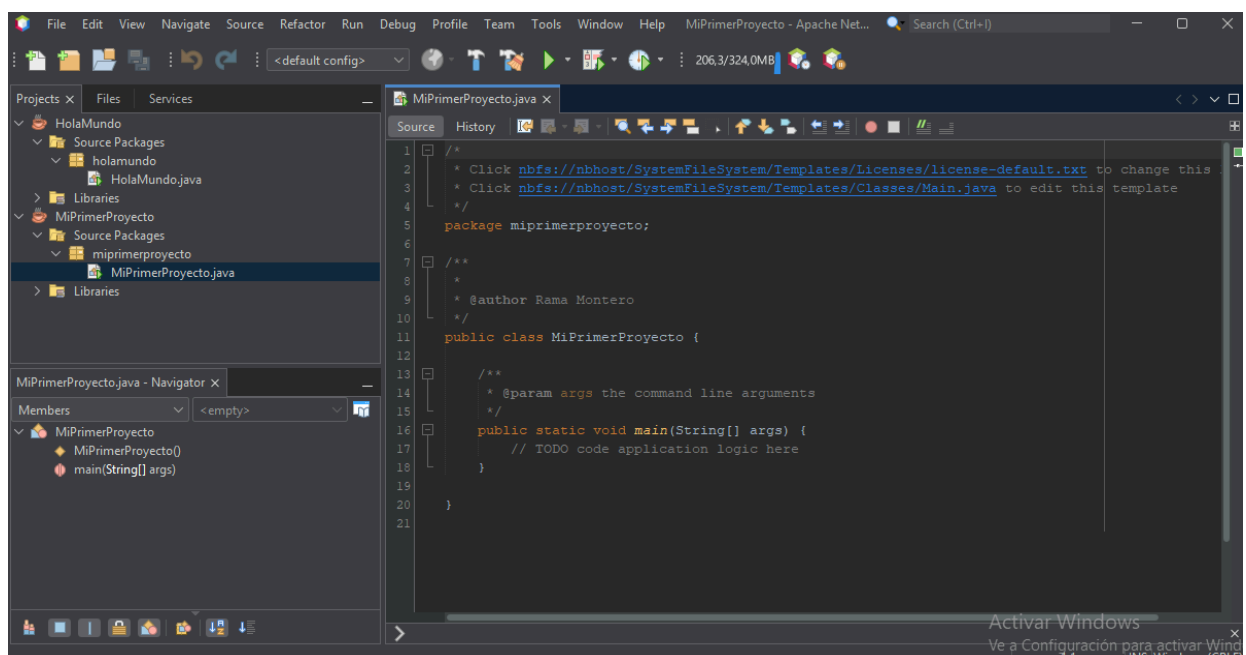


```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Versión 10.0.26100.6584]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\NoxiePC>java -version
java version "21.0.7" 2025-04-15 LTS
Java(TM) SE Runtime Environment (build 21.0.7+8-LTS-245)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.7+8-LTS-245, mixed mode, sharing)

C:\Users\NoxiePC|
```

b.

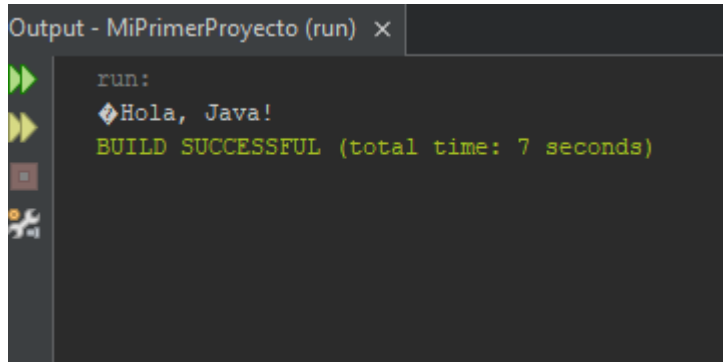


```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package miprimerproyecto;
6
7  /**
8   *
9   * @author Rama Montero
10  */
11  public class MiPrimerProyecto {
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  }
```

2. Escribir y ejecutar un programa básico en Java.

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

c.



```
Output - MiPrimerProyecto (run) x
run:
  Hola, Java!
BUILD SUCCESSFUL (total time: 7 seconds)
```

3. Crea un programa que declare las siguientes variables con valores asignados:

- String nombre
- int edad
- double altura
- boolean estudiante

Imprime los valores en pantalla usando `System.out.println()`.

```
8      * @author Rama Montero
9      */
10     public class DatosPersonales {
11         public static void main(String[] args) {
12             String nombre = "Ramón";
13             int edad = 28;
14             double altura = 1.75;
15             boolean estudiante = true;
16
17             System.out.println("Nombre: " + nombre);
18             System.out.println("Edad: " + edad);
19             System.out.println("Altura: " + altura);
20             System.out.println("Estudiante: " + estudiante);
21         }
22     }
```

```
Output - MiPrimerProyecto (run) x
run:
Nombre: Ramon
Edad: 28
Altura: 1.75
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.

```
11  * @author Rama Montero
12  */
13  public class EntradaUsuario {
14      public static void main(String[] args) {
15
16          try (Scanner scanner = new Scanner(System.in)
17              ) {
18
19              System.out.print("Ingresa tu nombre: ");
20              String nombre = scanner.nextLine();
21              System.out.print("Ingresa tu edad: ");
22              int edad = scanner.nextInt();
23              System.out.println("Tu nombre es: " + nombre);
24              System.out.println("Tu edad es: " + edad);
25
26          }
27      }
28  }
```

```
Output - MiPrimerProyecto (run) #2 x
run:
Ingresa tu nombre: Ramon
Ingresa tu edad: 28
Tu nombre es: Ramon
Tu edad es: 28
BUILD SUCCESSFUL (total time: 7 seconds)
```

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.

```
11  * @author Rama Montero
12  */
13  public class OperacionesBasicas {
14      public static void main(String[] args) {
15
16          try (Scanner scanner = new Scanner(System.in)) {
17
18              System.out.print("Ingresa el primer número entero: ");
19              int numero1 = scanner.nextInt();
20
21              System.out.print("Ingresa el segundo número entero: ");
22              int numero2 = scanner.nextInt();
23
24              int suma = numero1 + numero2;
25              int resta = numero1 - numero2;
26              int multiplicacion = numero1 * numero2;
27
28              double division = 0;
29              if (numero2 != 0) {
30                  division = (double) numero1 / numero2;
31              }
32
33              System.out.println("Suma: " + suma);
34              System.out.println("Resta: " + resta);
35              System.out.println("Multiplicación: " + multiplicacion);
36
37              if (numero2 != 0) {
38                  System.out.println("División: " + division);
39              } else {
40                  System.out.println("División: No se puede dividir por cero");
41              }
42          }
```

```
Output - MiPrimerProyecto (run) #2 x
run:
Ingresa el primer número entero: 12
Ingresa el segundo número entero: 6
Suma: 18
Resta: 6
Multiplicación: 72
División: 2.0
BUILD SUCCESSFUL (total time: 4 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()`.

```
8  * @author Rama Montero
9  */
10 public class CaracteresEscape {
11     public static void main(String[] args) {
12         System.out.println("Nombre: Juan Pérez\nEdad: 30 años\nDirección: \"Calle Falsa 123\"");
13     }
14 }
```

Output - MiPrimerProyecto (run) #3 x

```
run:
Nombre: Juan Pérez
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
x = x + 5; // Línea 2
System.out.println(x); // Línea 3
```

En el código dado, todas las líneas son instrucciones porque realizan acciones completas: declarar y asignar una variable (línea 1), actualizar su valor (línea 2) e imprimirlo (línea 3). Dentro de estas instrucciones, también hay expresiones, como 10, $x + 5$ y x , que son fragmentos de código que producen valores. La diferencia es que una expresión genera un valor, mientras que una instrucción ejecuta una acción completa.

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.

```
13 public class DivisionEnteros {
14     public static void main(String[] args) {
15
16         try (Scanner scanner = new Scanner(System.in)) {
17
18             System.out.print("Ingresa el primer número entero: ");
19             int num1 = scanner.nextInt();
20
21             System.out.print("Ingresa el segundo número entero: ");
22             int num2 = scanner.nextInt();
23
24             if (num2 != 0) {
25                 int resultado = num1 / num2;
26                 System.out.println("Resultado (entero): " + resultado);
27             } else {
28                 System.out.println("Error: No se puede dividir por cero.");
29             }
30         }
31     }
32 }
```

```
Output - MiPrimerProyecto (run) #3 x
run:
Ingresa el primer número entero: 13
Ingresa el segundo número entero: 2
Resultado (entero): 6
BUILD SUCCESSFUL (total time: 7 seconds)
```

```
13 public class DivisionEnteros {
14     public static void main(String[] args) {
15
16         try (Scanner scanner = new Scanner(System.in)) {
17
18             System.out.print("Ingresa el primer número: ");
19             double num1 = scanner.nextDouble();
20
21             System.out.print("Ingresa el segundo número: ");
22             double num2 = scanner.nextDouble();
23
24             if (num2 != 0) {
25                 double resultado = num1 / num2;
26                 System.out.println("Resultado (decimal): " + resultado);
27             } else {
28                 System.out.println("Error: No se puede dividir por cero.");
29             }
30         }
31     }
32 }
```

```
Output - MiPrimerProyecto (run) #3 x
run:
Ingresa el primer número: 13
Ingresa el segundo número: 2
Resultado (decimal): 6.5
BUILD SUCCESSFUL (total time: 4 seconds)
```

Al usar `int`, la división entre dos enteros descarta los decimales, mostrando solo la parte entera. En cambio, con `double`, el resultado conserva los decimales, lo que permite mayor 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);  
}  
}
```

Corrección:

```
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();  
        System.out.println("Hola, " + nombre);  
        scanner.close();  
    }  
}
```

El error estaba en que se usó `scanner.nextInt()` para leer un nombre, pero ese método solo sirve para leer números enteros. Como el nombre es texto, debía usarse `scanner.nextLine()` o `scanner.next()` para capturarlo correctamente.

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	Descripción	Valor de variables
<code>int a = 5;</code>	Se declara a y se le asigna 5	<code>a = 5</code>
<code>int b = 2;</code>	Se declara b y se le asigna 2	<code>b = 2</code>
<code>int resultado = a / b;</code>	Se divide 5 entre 2 (enteros)	<code>resultado = 2</code> (no 2.5, porque es int)
<code>System.out.println(...)</code>	Se imprime el resultado	Se muestra: Resultado: 2

El valor de resultado es 2 porque se realiza una división entre dos variables enteras (int). En Java, cuando se divide un entero por otro entero, el resultado también es entero, y cualquier parte decimal se descarta.