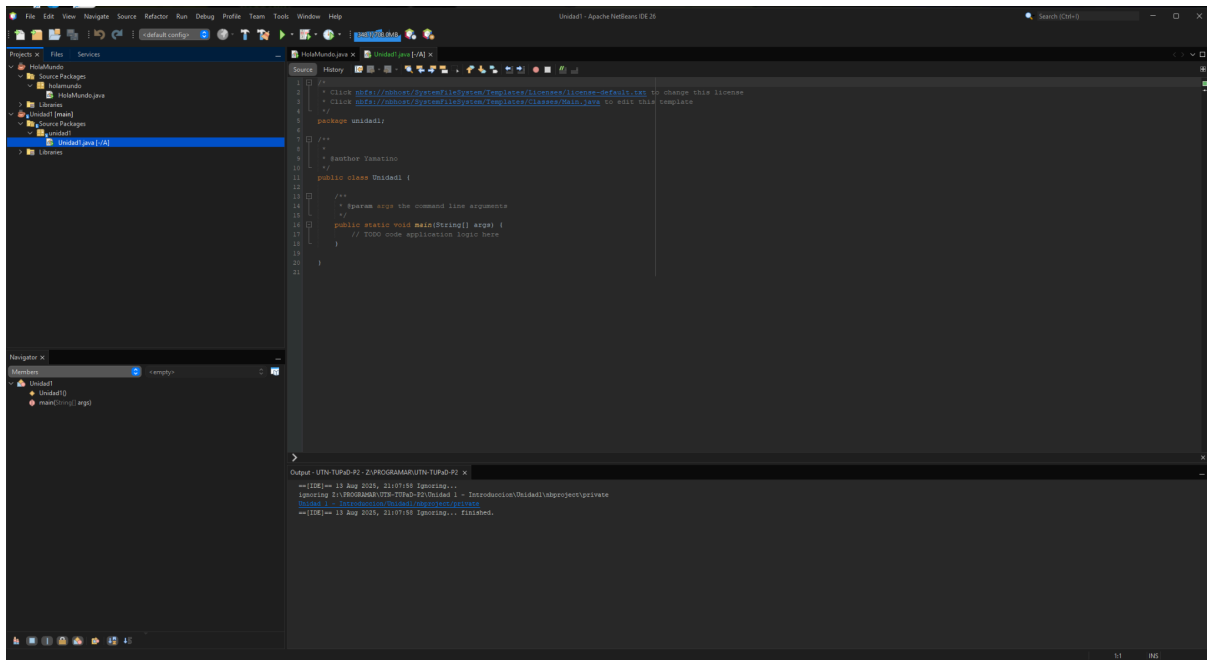


Link a github: <https://github.com/Yamatino/UTN-TUPaD-P2>

1)



2)

```
10  */
11  public class HolaMundo {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          System.out.println("¡Hola, Java!");
18      }
19
20  }
21
```

unidad1.HolaMundo >

Output x

UTN-TUPaD-P2 - Z:\PROGRAMAR\UTN-TUPaD-P2 x    Unidad1 (run) x

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

7)

int x = 10; (Instrucción)

x = x + 5; (Instrucción)

System.out.println(x); (Instrucción)

una instrucción es como una oración completa que le dice al programa qué hacer, mientras que una expresión es como una frase que se evalúa para obtener un valor.

9)

El problema en el código original es que se intenta leer un String (el nombre) usando el método `scanner.nextInt()`, que está diseñado para leer únicamente números enteros.

Error: El método `nextInt()` no puede procesar caracteres de texto, lo que provoca un error de tipo `InputMismatchException` en tiempo de ejecución.

Solución: Se debe usar el método `scanner.nextLine()` o `scanner.next()` para leer una cadena de texto. `nextLine()` lee toda la línea de texto, mientras que `next()` lee hasta el primer espacio

10)

Línea de Código	Valor de a	Valor de b	Valor de resultado
<code>int a = 5;</code>	5	(sin inic.)	(sin inic.)
<code>int b = 2;</code>	5	2	(sin inic.)
<code>int resultado = a / b;</code>	5	2	2

El valor final de resultado es 2. Esto se debe a que las variables `a` y `b` son de tipo entero (`int`). En Java, cuando se realiza una operación de división entre dos números enteros, el resultado también es un entero. La parte decimal del resultado se trunca (se elimina), no se redondea.