



UADY

**FACULTAD DE
MATEMÁTICAS**

"Luz, Ciencia y Verdad"

UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS

LICENCIATURA EN ING. DE SOFTWARE

3er. SEMESTRE

PROYECTO FINAL:

Analizador Léxico y Sintáctico

PROFESOR:

DR. LUIS FERNANDO CURI QUINTAL

10 /DICIEMBRE /2019



Gabriel Islas Peraza

Emilio M. Lara Bermúdez

Víctor M. Lavalle Cantón

Eberth F. Mezeta Xool

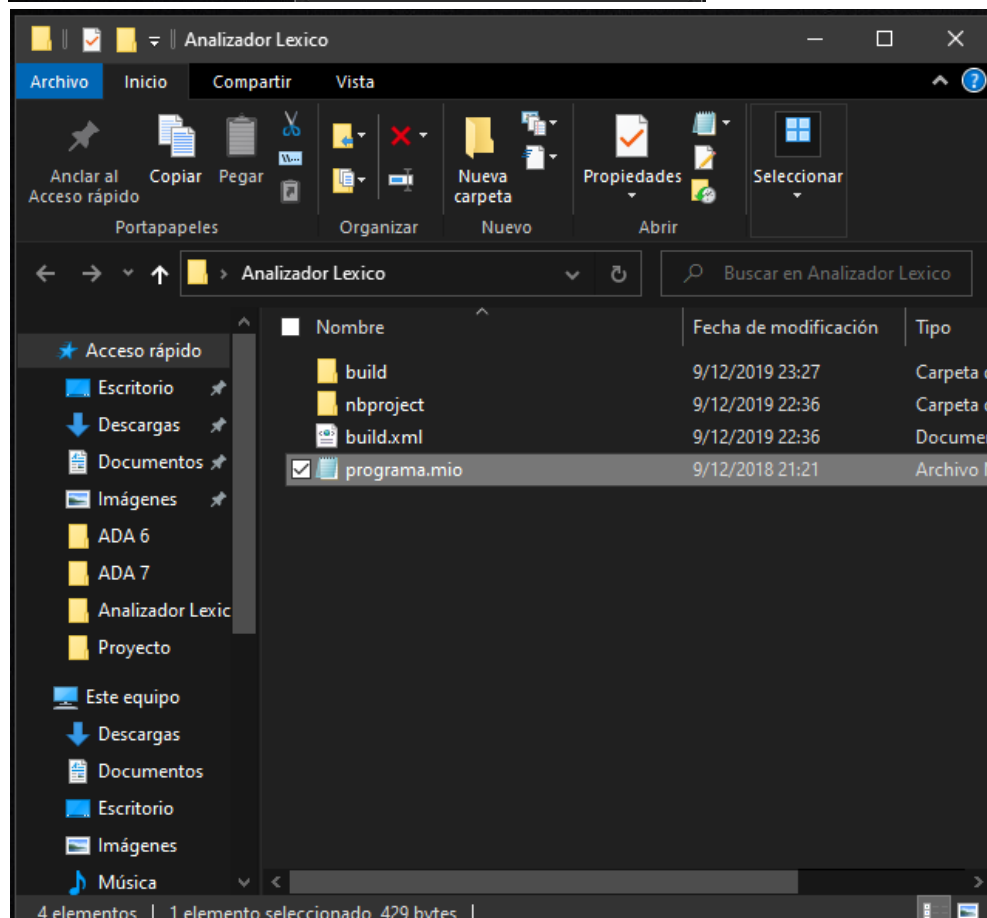
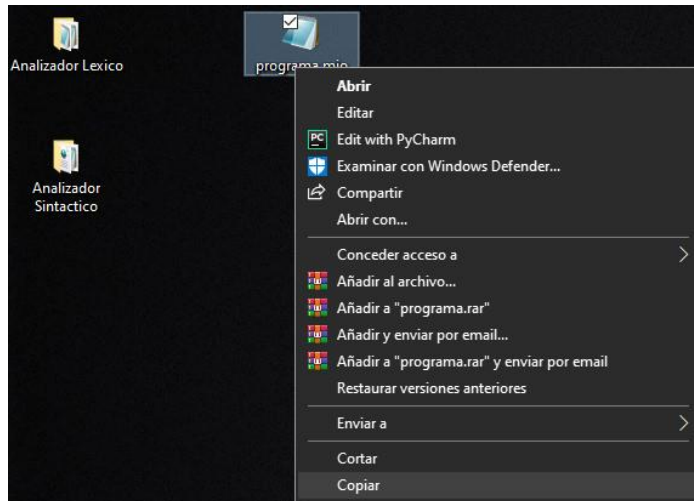
Manual de Usuario

2

Restricciones

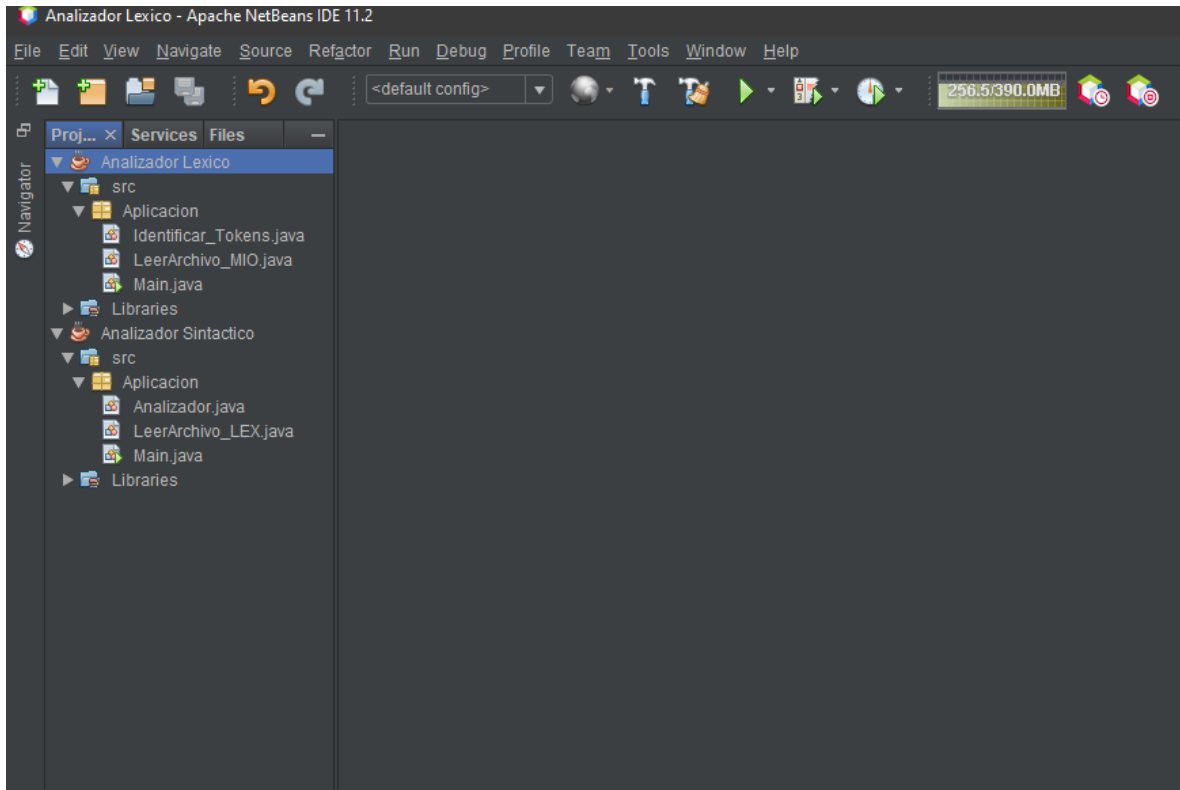
- Tener un IDE de Java (puede ser eclipse o Netbeans 8.2 o Posterior).
- Tener instalado JDK 8 en adelante.

1. Ingresamos un archivo con la extensión ".mio" en la carpeta del analizador léxico.

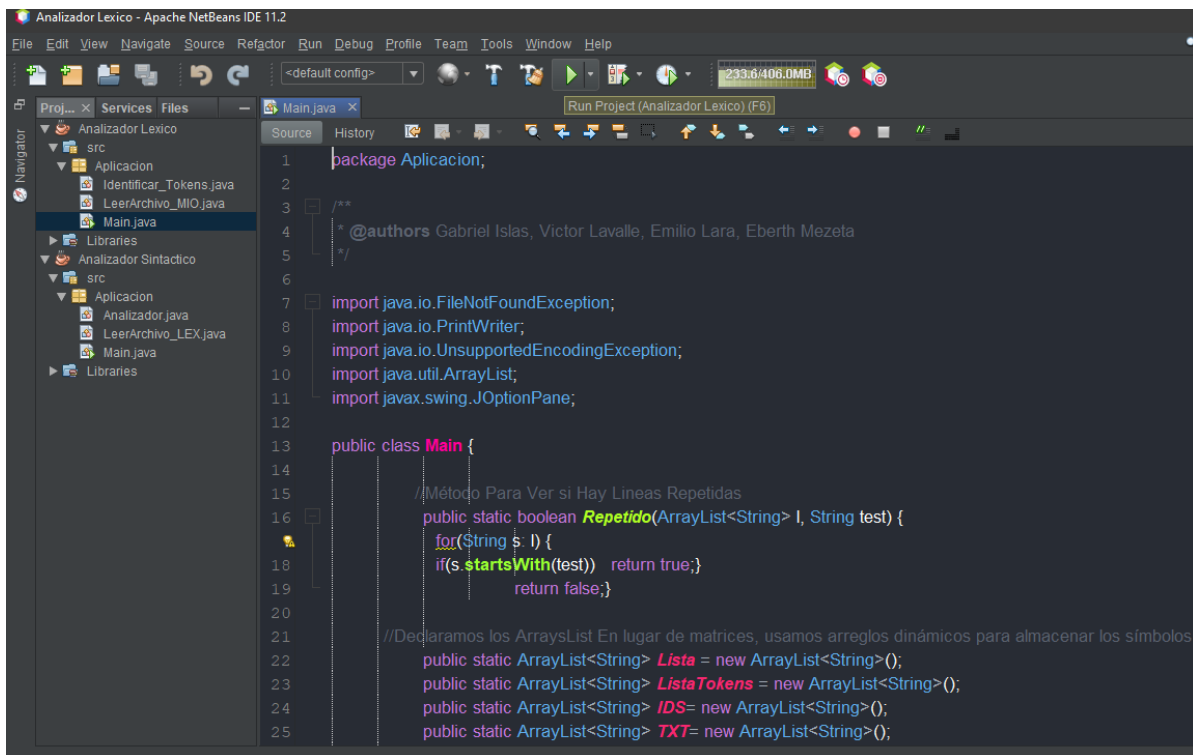


2. Abrimos los proyectos en nuestro IDE.

3

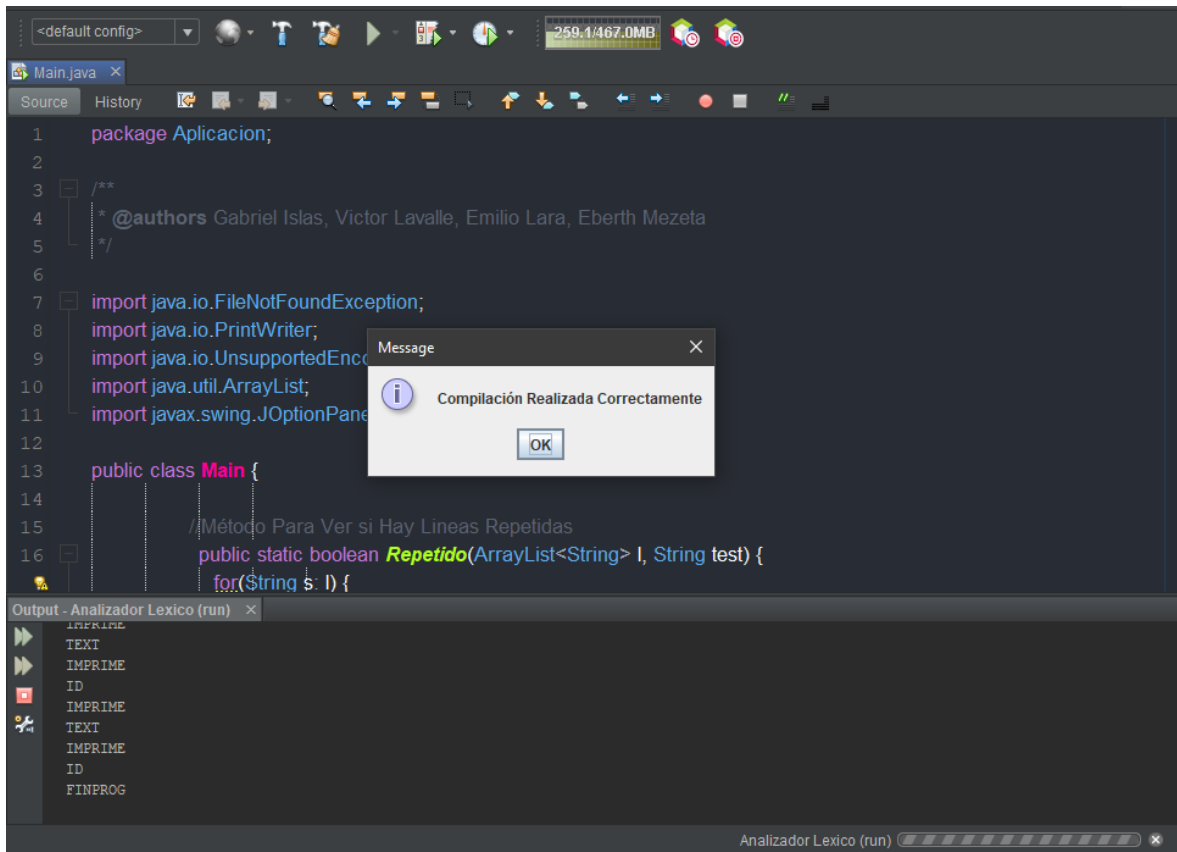


3. Abrimos la clase "Main" y la ejecutamos.

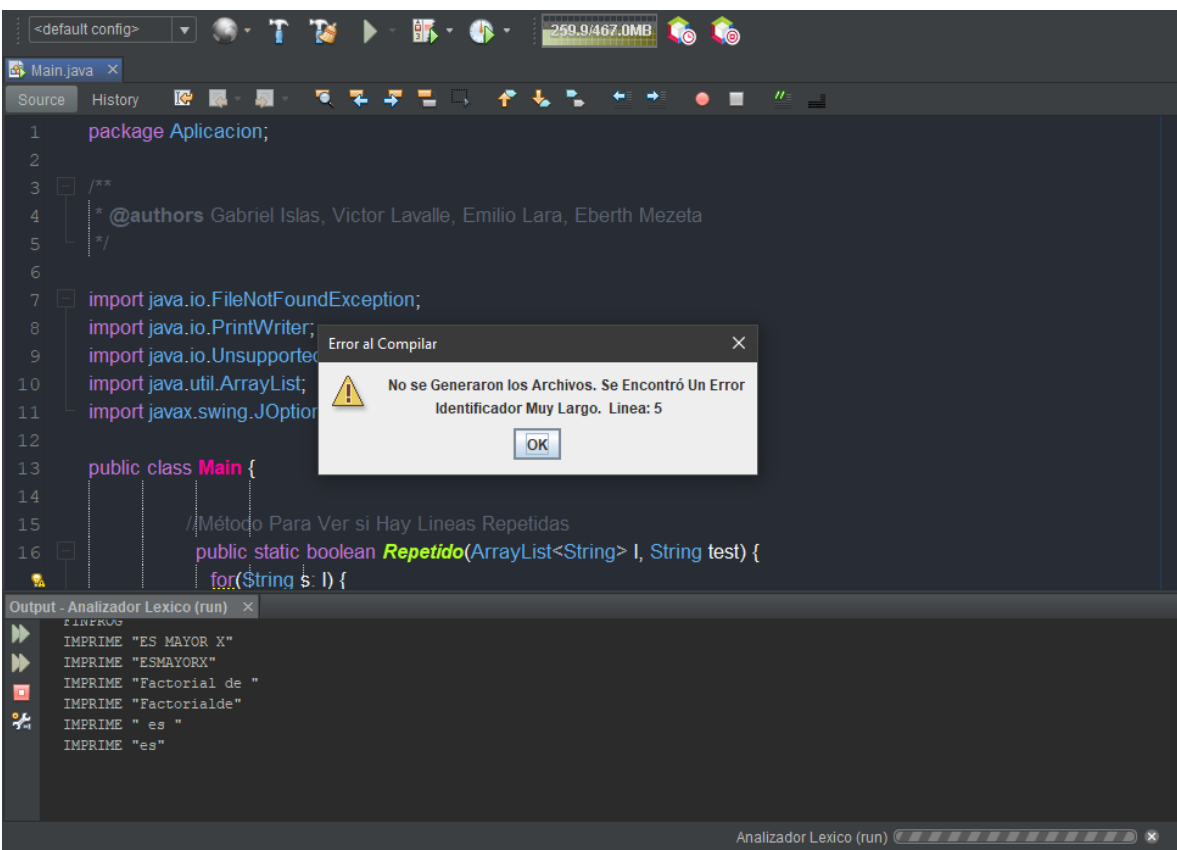


3.1 Si el código en "programa.mio" contiene algún error mostrará una advertencia, de lo contrario, muestra un mensaje de "Compilación Realizada Correctamente".

4



The screenshot shows an IDE window with a Java file named "Main.java". The code includes package declarations, imports for Java I/O, utility classes, and Swing, and a public class "Main" with a static method "Repetido". A message dialog box is displayed in the center, stating "Compilación Realizada Correctamente" (Compilation Completed Successfully) with an "OK" button. Below the code editor, the "Output - Analizador Lexico (run)" window shows a list of tokens: IMPRIME, TEXT, IMPRIME, ID, IMPRIME, TEXT, IMPRIME, ID, and FINPROG. The status bar at the bottom indicates "Analizador Lexico (run)".



The screenshot shows the same IDE window with the same Java code. However, a message dialog box is displayed, stating "Error al Compilar" (Compilation Error) with the message "No se Generaron los Archivos. Se Encontró Un Error Identificador Muy Largo. Línea: 5" (Files were not generated. An error was found: Identifier too long. Line: 5). Below the code editor, the "Output - Analizador Lexico (run)" window shows a list of tokens: IMPRIME, TEXT, IMPRIME, ID, IMPRIME, TEXT, IMPRIME, ID, and FINPROG. The status bar at the bottom indicates "Analizador Lexico (run)".

En consola, en caso de compilar se muestra el código del "programa.mio" y se muestra la secuencia de tokens identificados.

5

Si se encontró un error, de igual manera se muestra el código y el error encontrado.

```

Output - Analizador Lexico (run) x
run:
INICIAL
# Programa que calcula el factorial de un numero
PROGRAMA factorial
# VarX acumula los productos por iteracion
VarX = 0x1
# VarY contiene el iterador del factor
VarY = 0x0
LEE Num
# Aplica Num! = 1 * 2 * 3 * ... * Num
REPITE Num VECES
VarY = VarY + 0x1
VarX = VarX * VarY
SI COMPARA VarX > VarY ENTONCES
IMPRIME "ES MAYOR X"
FINSI
FINREP
IMPRIME "Factorial de "
IMPRIME Num
IMPRIME " es "
IMPRIME VarX
FINPROG
IMPRIME "ES MAYOR X"
IMPRIME "ESMAYORX"
IMPRIME "Factorial de "
IMPRIME "Factorialde"
IMPRIME " es "
IMPRIME "es"

PROGRAMA
ID
ID
IGUAL
VAL
ID

```

```

Output - Analizador Lexico (run) x
run:
INICIAL
# Programa que calcula el factorial de un numero
PROGRAMA factorial
# VarX acumula los productos por iteracion
VarX = 0x1
# VarY contiene el iterador del factor
VarY454v524tf53chch6c 6 = 0x0
LEE Num
# Aplica Num! = 1 * 2 * 3 * ... * Num
REPITE Num VECES
VarY = VarY + 0x1
VarX = VarX * VarY
SI COMPARA VarX > VarY ENTONCES
IMPRIME "ES MAYOR X"
FINSI
FINREP
IMPRIME "Factorial de "
IMPRIME Num
IMPRIME " es "
IMPRIME VarX
FINPROG
IMPRIME "ES MAYOR X"
IMPRIME "ESMAYORX"
IMPRIME "Factorial de "
IMPRIME "Factorialde"
IMPRIME " es "
IMPRIME "es"

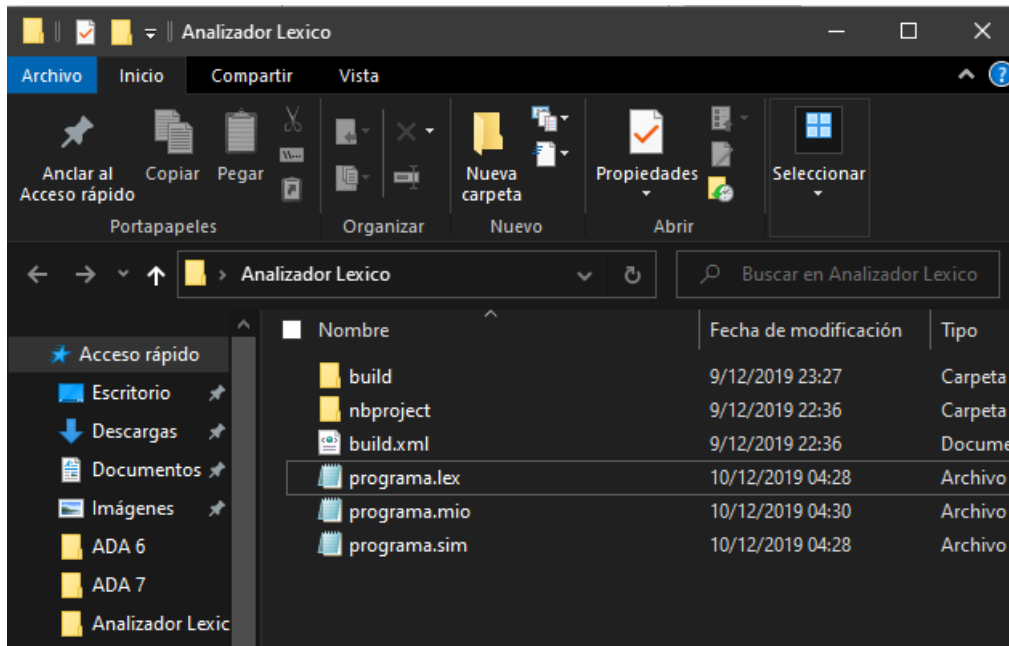
No se Generaron los Archivos, Se Encontró Un Error
Identificador Muy Largo. Linea: 5
BUILD SUCCESSFUL (total time: 9 minutes 39 seconds)

```

3.2 Si la compilación es exitosa, se generan dos archivos "programa.lex" y "programa.sim" en la misma carpeta del analizador léxico.

6

- "programa.lex" contiene la secuencia de tokens
- El archivo .sim contiene los arreglos de símbolos.



```

programa.lex: Bloc de notas
Archivo Edición Formato Ver Ayuda
PROGRAMA
ID
ID
IGUAL
VAL
ID
IGUAL
VAL
LEE
ID
REPITE
ID
VECES
ID
IGUAL
ID
OPARIT
VAL
ID
IGUAL
ID
OPARIT
ID
ST

```

Ln 1, Col 1

```

programa.sim: Bloc de notas
Archivo Edición Formato Ver Ayuda
IDS
factorial, ID1
VarX, ID2
VarY, ID3
Num, ID4
COMPARA, ID5

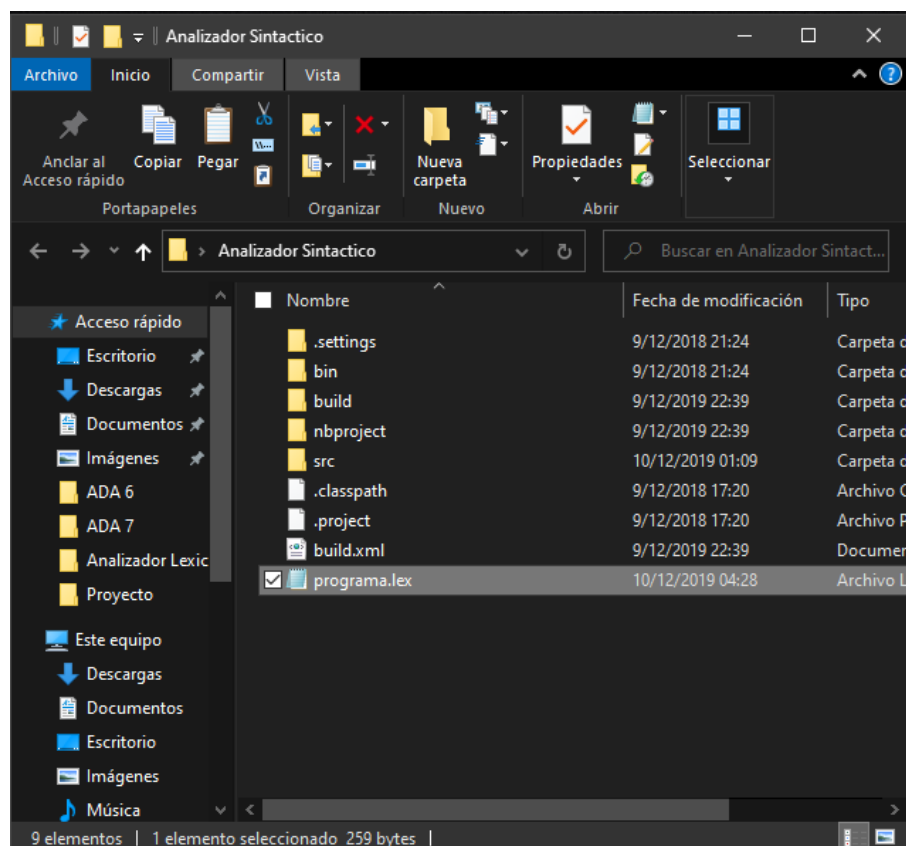
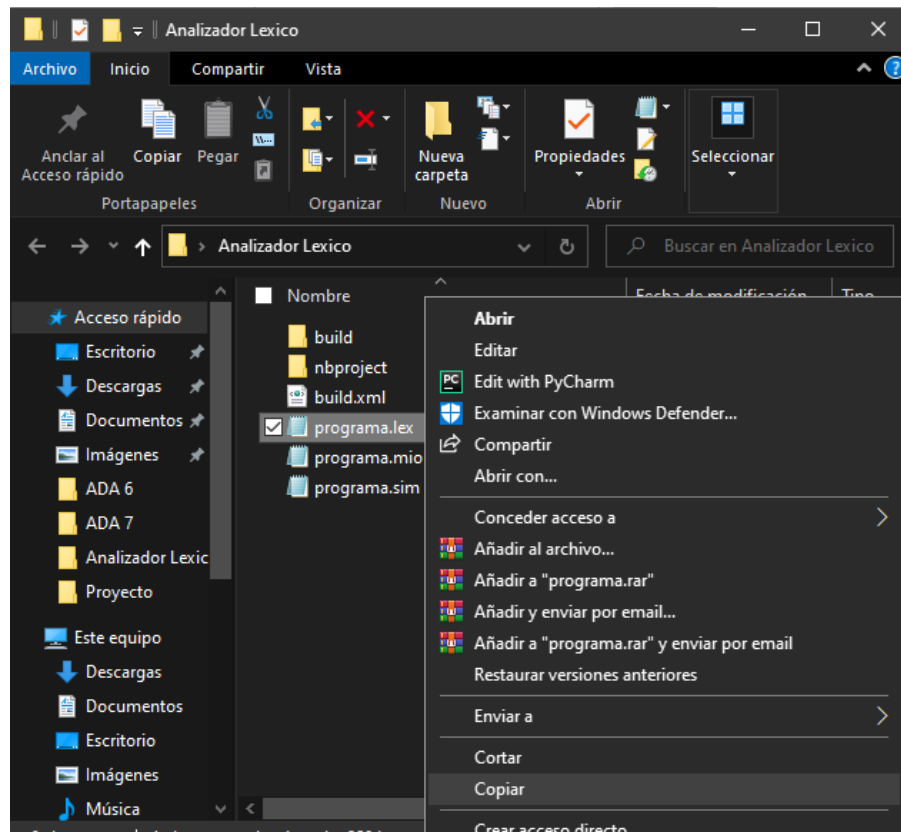
TXT
"ESMAYORX", TX1
"Factorialde", TX2
"es", TX3

VAL
0x1, 1
0x0, 0

```

Ln 1, Col 1

4. Posterior a la creación del archivo “programa.lex” es necesario copiarlo a la carpeta del proyecto “Analizador Sintactico”.



5.

5. En el IDE abrimos la clase "Main" del proyecto "Analizador Sintactico" y lo ejecutamos.



```

1 package Aplicacion;
2
3
4 import java.util.Stack;
5 import javax.swing.JOptionPane;
6
7 public class Main {
8
9     /*Stack=Pila: Estructura de datos de tipo LIFO
10     (Last in First Out, último en entrar, primero en salir)
11     permite almacenar y recuperar datos.*/
12
13
14     public static Stack<String> StackInicial = new Stack<>();
15     public static Stack<String> CambiarStack = new Stack<>();
16     public static Stack<String> StackActual = new Stack<>();
17
18     /*Metodo Para Mostrar Cuadro de Advertencia al No encontrar los archivos necesarios
19     public static void error(String x) {
20         JOptionPane.showMessageDialog(null, " Se Encontró Un Error\n", "Error al Cor
21
22     /*Main
23     public static void main(String[] args) {
24         /*Stack.push (elemento) insertar un elemento en la pila.
25         El elemento se empuja hacia la parte superior de la pila.*/

```

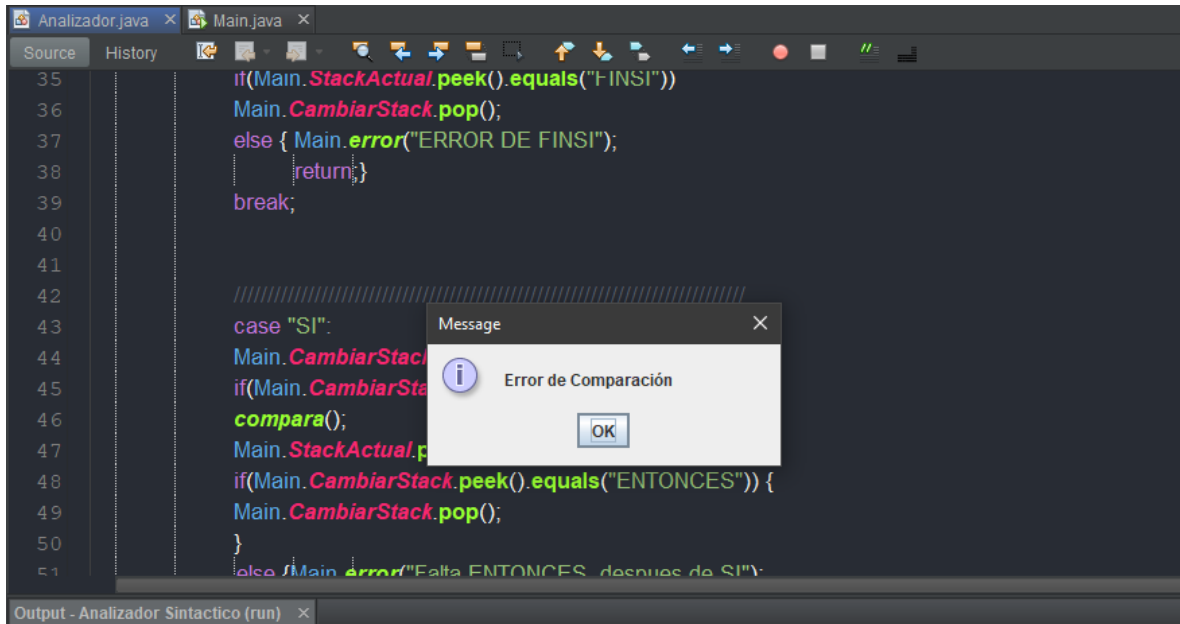
5.1 Se arroja un mensaje de que se ha compilado correctamente. Y El contenido del archivo .lex se Imprimirá en consola

```

run:
$
FINPROG
ID
IMPRIME
TEXT
IMPRIME
ID
IMPRIME
TEXT
IMPRIME
FINREP
FINSI
TEXT
IMPRIME
ENTONCES
ID
OPRELAC
ID
ID
SI
ID
OPARIT
ID
IGUAL
ID
ID
OPARIT
ID
IGUAL
ID
VECES
ID
REPITE
ID

```


5.2. En caso de haber algún tipo error durante compilación se enviara un mensaje de error.



```

35 if(Main.StackActual.peek().equals("FINSI"))
36 Main.CambiarStack.pop();
37 else { Main.error("ERROR DE FINSI");
38       return;}
39 break;
40
41
42 ///////////////////////////////////////////////////
43 case "SI":
44 Main.CambiarStack.compara();
45 if(Main.CambiarStack.compara());
46 Main.StackActual.pop();
47 if(Main.CambiarStack.peek().equals("ENTONCES")) {
48 Main.CambiarStack.pop();
49 }
50 else {Main.error("Falta ENTONCES despues de SI");
51

```

Message

Error de Comparación

OK

Output - Analizador Sintactico (run)

Caso de Usos

