

Instituto Tecnológico de Costa Rica

Unidad de Ingeniería en Computación

Compiladores e intérpretes

Proyecto 1

Profesor

Allan Gabriel Rodríguez Davila

Estudiante

Yeudi Martínez Sánchez

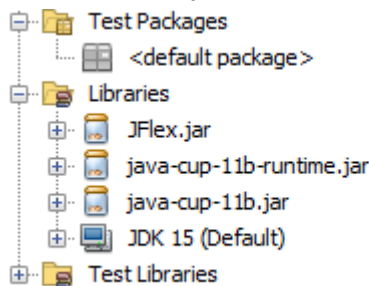
I semestre, 2022

Manual de usuario:

Paso 1 -

Descargar y descomprimir y abrir en un IDE el proyecto

(Importante verificar tener las librerías, en caso de no aparecer click derecho en libraries add JAR/Folder y seleccionar las librerías, estan ubicadas en la misma carpeta del proyecto)



Paso 2-

En la clase llamada Principal se deben cambiar las rutas

```
// C:/Users/jeudy/Music/Analizado_CopiaMediaNoche/src/codigo/Sintax.cup

String ruta1 = "C:/Users/jeudy/OneDrive/Imágenes/Proyecto-_Compiladores-main/Analizado_CopiaMediaNoche/Analizado_CopiaMediaNoche/src/codigo/Lexer.flex";
String ruta2 = "C:/Users/jeudy/OneDrive/Imágenes/Proyecto-_Compiladores-main/Analizado_CopiaMediaNoche/Analizado_CopiaMediaNoche/src/codigo/LexerCup.flex";

// C:/Users/jeudy/Downloads/Proyecto-_Compiladores-main/Proyecto-_Compiladores-main/AnalizadorLexico/AnalizadorLexico/src/codigo/Sintax.cup
String[] rutaS = {"-parser", "Sintax", "C:/Users/jeudy/OneDrive/Imágenes/Proyecto-_Compiladores-main/Analizado_CopiaMediaNoche/Analizado_CopiaMediaNoche/src/codigo/Sintax.cup"};
generar(ruta1, ruta2, rutaS);
```

String ruta1 = "C:/Users/jeudy/Music/Analizado_CopiaMediaNoche/src/codigo/Lexer.flex";

Por ejemplo para esta ruta se debe cambiar lo marcado en verde por la ruta donde se guarde el proyecto, click derecho en Sintax.cup >> propiedades y sustituir la base de la ruta por la base de ruta donde está ubicado Sintax.cup

Paso 3-

Click derecho en clase Principal >> File Run

Paso 4-

Ejecutar la clase FrmPrincipal >> File Run

Pruebas de funcionalidad:

A continuación se adjunta evidencia de la funcionalidad del analizador sintáctico y léxico

Analizador Lexico

Abrir txt

```
main()
{
    if(var < 5)
    {
        while( var > 12 ){

            if(variante == 16){
                switch(varX){
                    case 777:
                        int multi = 6 *15#
                }
            }else{
                int cont = (cont + 12)#
            }
        }
    }else
    {
        // Comentario 1
        string imprimir = "Algun Texto"#
    }
}
```

Analisis Lexico

<Llave de cierre>	}
<Reservada else>	else
Line --- 18	
<Llave de apertura>	{
Line --- 19	
Line --- 20	
<Tipo de dato>	string
<Identificador>	imprimir
<Operador igual>	=
<Comillas>	"
<Identificador>	Algun
<Identificador>	Texto
<Comillas>	"
<Caracter numeral>	#
Line --- 21	
Line --- 22	
<Llave de cierre>	}
Line --- 23	
Line --- 24	
Line --- 25	
Line --- 26	
<Llave de cierre>	}

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

```
main({  
  if(xvar <= 19)  
  {  
    int numerito = 25#  
  }  
}
```

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main(){  
    if(variable == 5)  
    {  
        int x = 3#  
    }  
}
```

Analisis Lexico

<Reservada main>	main
<Parentesis de apertura>	(
<Parentesis de cierre>)
<Llave de apertura>	{
-- 2	
-- 3	
<Reservada if>	if
<Parentesis de apertura>	(
<Identificador>	variable
<Operador relacional>	==
<Numero>	5
<Parentesis de cierre>)
-- 4	
<Llave de apertura>	{
-- 5	
<Tipo de dato>	int
<Identificador>	x
<Operador igual>	=
<Numero>	3
<Punto y coma >	#
-- 6	
<Llave de cierre>	}
-- 7	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main(){  
  switch(variable){  
    case 7:  
      int x = 3#  
  }  
}
```

Analisis Lexico

```
<Llave de apertura> {  
- 2  
<Condicional > sv  
<Parentesis de apertura> (  
<Identificador> variable  
<Parentesis de cierre> )  
<Llave de apertura> {  
- 3  
<Palabra reservada de condicional>  
<Numero> 7  
<Tipo de dato> :  
- 4  
<Tipo de dato> int  
<Identificador> x  
<Operador igual> =  
<Numero> 3  
<Punto y coma > #  
- 5  
<Llave de cierre> }  
- 6  
- 7  
<Llave de cierre> }
```

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Abrir txt

```
main(){
  switch(variable){
    case 7:
      int x = 3#
  }
}
```

Analisis Lexico

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main({
while(c < 5){
    int x = 5#
}
```

Analisis Lexico

line	Encontrado
1	
<Reservada main>	main
<Parentesis de apertura>	(
<Parentesis de cierre>)
<Llave de apertura>	{
2	
<Reservada while>	while
<Parentesis de apertura>	(
<Identificador>	c
<Operador relacional>	<
<Numero>	5
<Parentesis de cierre>)
<Llave de apertura>	{
3	
<Tipo de dato>	int
<Identificador>	x
<Operador igual>	=
<Numero>	5
<Punto y coma >	#
4	
<Llave de cierre>	}
5	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main()
{

float y = 25.3#
}
```

Analisis Lexico

line		Encontrado
1	<Reservada main>	main
	<Parentesis de apertura>	(
	<Parentesis de cierre>)
2		
	<Llave de apertura>	{
3		
4		
5		
	<> float >	
	<Identificador>	y
	<Operador igual>	=
	<Numero>	25
	<Caracter >	.
	<Numero>	3
	<Punto y coma >	#
6		
7		
	<Llave de cierre>	}

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main()
{
int x = 5#
}
```

Analisis Lexico

line 1	<Reservada main>	Encontrado	main
	<Parentesis de apertura>	(
	<Parentesis de cierre>)	
- 2	<Llave de apertura>	{	
- 3			
- 4	<Tipo de dato>		int
	<Identificador>	x	
	<Operador igual>	=	
	<Numero>	5	
	<Punto y coma >	#	
- 5			
- 6	<Llave de cierre>	}	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

char otro#

I

line 1

<> char >

<Identificador>

<Punto y coma >

Encontrado

otro

#

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

int dde#

line 1

	Encontrado	
<Tipo de dato>		int
<Identificador>	dde	
<Punto y coma >	#	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

int numeroEntero = 55#

Analisis Lexico

line 1

	Encontrado	
<Tipo de dato>		int
<Identificador>	numeroEntero	
<Operador igual>	=	
<Numero>	55	
<Punto y coma >	#	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

float var = 2.8985#

I

Analisis Lexico

line 1

	Encontrado
<> float >	
<Identificador>	var
<Operador igual>	=
<Numero>	2
<Caracter >	.
<Numero>	8985
<Punto y coma >	#

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

char caracter = "s"#

line 1

Encontrado

<> char >	
<Identificador>	caracter
<Operador igual>	=
<Comillas>	"
<Identificador>	s
<Comillas>	"
<Punto y coma >	#

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

string miVar = "cadena" #

I

Analisis Lexico

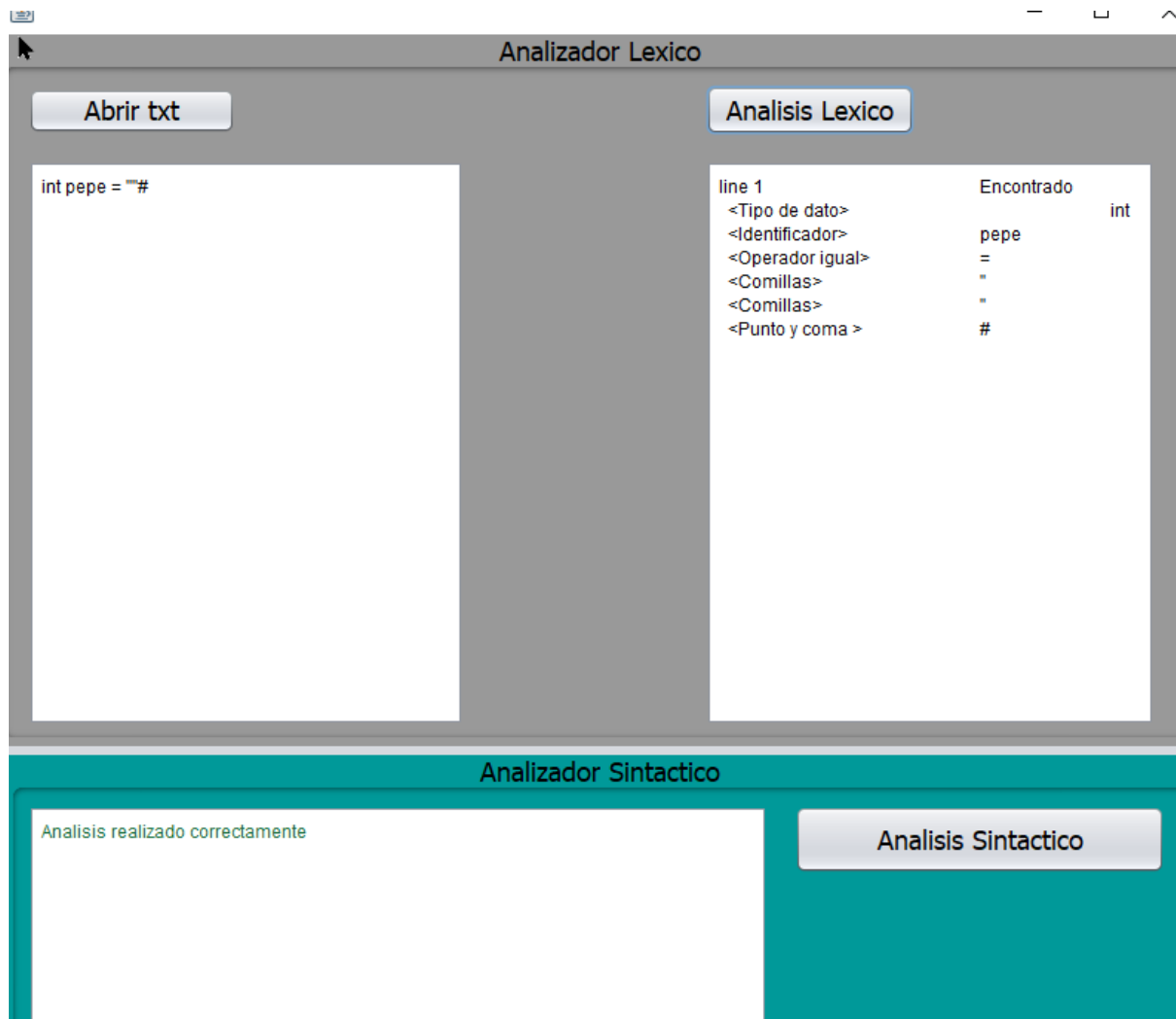
line 1

<Tipo de dato>	Encontrado
<Identificador>	string
<Operador igual>	miVar
<Comillas>	=
<Identificador>	"
<Comillas>	cadena
<Punto y coma >	"
	#

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico



Analizador Lexico

Abrir txt

Analisis Lexico

```
main(){  
    if(xvar <= 19)  
    {  
        int numerito = 25#  
    }  
    else{  
        numerito = 255#  
    }  
}
```

line	Encontrado
1	<Reservada main> main
	<Parentesis de apertura> (
	<Parentesis de cierre>)
	<Llave de apertura> {
-- 2	
-- 3	<Reservada if> if
	<Parentesis de apertura> (
	<Identificador> xvar
	<Operador relacional> <=
	<Numero> 19
	<Parentesis de cierre>)
-- 4	
	<Llave de apertura> {
-- 5	
	<Tipo de dato> int
	<Identificador> numerito
	<Operador igual> =
	<Numero> 25
	<Punto y coma > #
-- 6	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Abrir txt

```
main(){
// aqui un comentario
if(xvar <= 19)
{
    int numerito = 25#
}
else{
    int pepe = 45#
}
}
```

Analisis Lexico

<Llave de apertura>	{	
- 5		
<Tipo de dato>		int
<Identificador>	numerito	
<Operador igual>	=	
<Numero>	25	
<Punto y coma >	#	
- 6		
<Llave de cierre>	}	
- 7		
<Reservada else>	else	
<Llave de apertura>	{	
- 8		
<Tipo de dato>		int
<Identificador>	pepe	
<Operador igual>	=	
<Numero>	45	
<Punto y coma >	#	
- 9		
<Llave de cierre>	}	
- 10		
<Llave de cierre>	}	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main()
{
    if(var < 5)
    {
        float pi = 3.14#
    }else
    {
        int miFuncion = 12#
    }
}
```

Analisis Lexico

Line	Encontrado
Line 1	<Reservada main> main
	<Parentesis de apertura> (
	<Parentesis de cierre>)
Line --- 2	<Llave de apertura> {
Line --- 3	
Line --- 4	<Condicional> if
	<Parentesis de apertura> (
	<Identificador> var
	<Operador relacional> <
	<Numero> 5
	<Parentesis de cierre>)
Line --- 5	<Llave de apertura> {
Line --- 6	
	<-----> float >
	<Identificador> pi
	<Operador igual> =
	<Numero> 3
	<Caracter > .

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main()
{
  int miFuncion()
  {
    if(var < 5)
    {
      while( var > var 20){
        float pi = 3.14#
      }
    }else
    {
      int pi = 3 #
    }
  }
}
```

Analisis Lexico

Line	Encontrado
Line 1	<Reservada main> main
	<Parentesis de apertura> (
	<Parentesis de cierre>)
Line --- 2	<Llave de apertura> {
Line --- 3	<Tipo de dato> int
	<Identificador> miFuncion
	<Parentesis de apertura> (
	<Parentesis de cierre>)
Line --- 4	<Llave de apertura> {
Line --- 5	<Condicional> if
	<Parentesis de apertura> (
	<Identificador> var
	<Operador relacional> <
	<Numero> 5
	<Parentesis de cierre>)
Line --- 6	<Llave de apertura> {

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

```
main()
{
  if(var < 5)
  {
    while( var > 12 ){
      float pi = 3.14#
    }
  }else
  {
    int pi = 3 #
  }
}
```

<Parentesis de cierre>

)

<Llave de apertura>

{

Line --- 6

<-----> float >

<Identificador>

pi

<Operador igual>

=

<Numero>

3

<Caracter >

.

<Numero>

14

<Caracter numeral>

#

Line --- 7

<Llave de cierre>

}

Line --- 8

<Llave de cierre>

}

<Reservada else>

else

Line --- 9

<Llave de apertura>

{

Line --- 10

<Tipo de dato>

int

<Identificador>

pi

<Operador igual>

=

<Numero>

3

<Caracter numeral>

#

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

```
main()
{
    if(var < 5)
    {
        while( var > 12 ){
            float pi = 3.14#
        }
    }else
    {
        int pi = 3 * 2 #
    }
}
```

```
<Caracter> .
<Numero> 14
Line --- 7
<Llave de cierre> }
Line --- 8
<Llave de cierre> }
<Reservada else> else
Line --- 9
<Llave de apertura> {
Line --- 10
<Tipo de dato> int
<Identificador> pi
<Operador igual> =
<Numero> 3
<Operador multiplicacion> *
<Numero> 2
<Caracter numeral> #
Line --- 11
<Llave de cierre> }
Line --- 12
Line --- 13
<Llave de cierre> }
```

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

```
main()
{
    if(var < 5)
    {
        while( var > 12 ){
            float pi = 3.14 #
        }
    }else
    {
        int pi = 3 + 2#
    }
}
```

Line	Encontrado
Line 1	<Reservada main> main
	<Parentesis de apertura> (
	<Parentesis de cierre>)
Line --- 2	<Llave de apertura> {
Line --- 3	<Condicional> if
	<Parentesis de apertura> (
	<Identificador> var
	<Operador relacional> <
	<Numero> 5
	<Parentesis de cierre>)
Line --- 4	<Llave de apertura> {
Line --- 5	<Reservada while> while
	<Parentesis de apertura> (
	<Identificador> var
	<Operador relacional> >
	<Numero> 12
	<Parentesis de cierre>)

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

```
main()
{
    if(var < 5)
    {
        while( var > 12 )
        {
            string pi = "Mi texto" #
        }
    }else
    {
        if(var < 5)
        {
            while( var > 12 ){
                float pi = 3.14 #
            }
        }else
        {
            while(stop == 99){
                pi = 12 +5 #
            }
        }
    }
}
```

```
<Numero> 12
<Parentesis de cierre> )
Line --- 6
<Llave de apertura> {
Line --- 7
<Tipo de dato> string
<Identificador> pi
<Operador igual> =
<Comillas> "
<Identificador> Mi
<Identificador> texto
<Comillas> "
<Caracter numeral> #
Line --- 8
<Llave de cierre> }
Line --- 9
<Llave de cierre> }
<Reservada else> else
Line --- 10
<Llave de apertura> {
Line --- 11
<Condicional> if
<Parentesis de apertura> (
<Identificador> var
```

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

Analisis Lexico

```
main()
{
    while(y <= 56){
        int particula = 555#
    }
}
```

Line 1	<Reservada main>	Encontrado	main
	<Parentesis de apertura>		(
	<Parentesis de cierre>)
Line --- 2	<Llave de apertura>		{
Line --- 3			
Line --- 4	<Reservada while>		while
	<Parentesis de apertura>		(
	<Identificador>		y
	<Operador relacional>		<=
	<Numero>		56
	<Parentesis de cierre>)
	<Llave de apertura>		{
Line --- 5			
	<Tipo de dato>		int
	<Identificador>		particula
	<Operador igual>		=
	<Numero>		555
	<Caracter numeral>		#
Line --- 6			

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico

Analizador Lexico

Abrir txt

```
main()
{
    if(var < 5)
    {
        while( var > 12 ){
            float variablita = 312.14#
        }
    }else
    {
        int pi =(12 + numero) #
    }
}
```

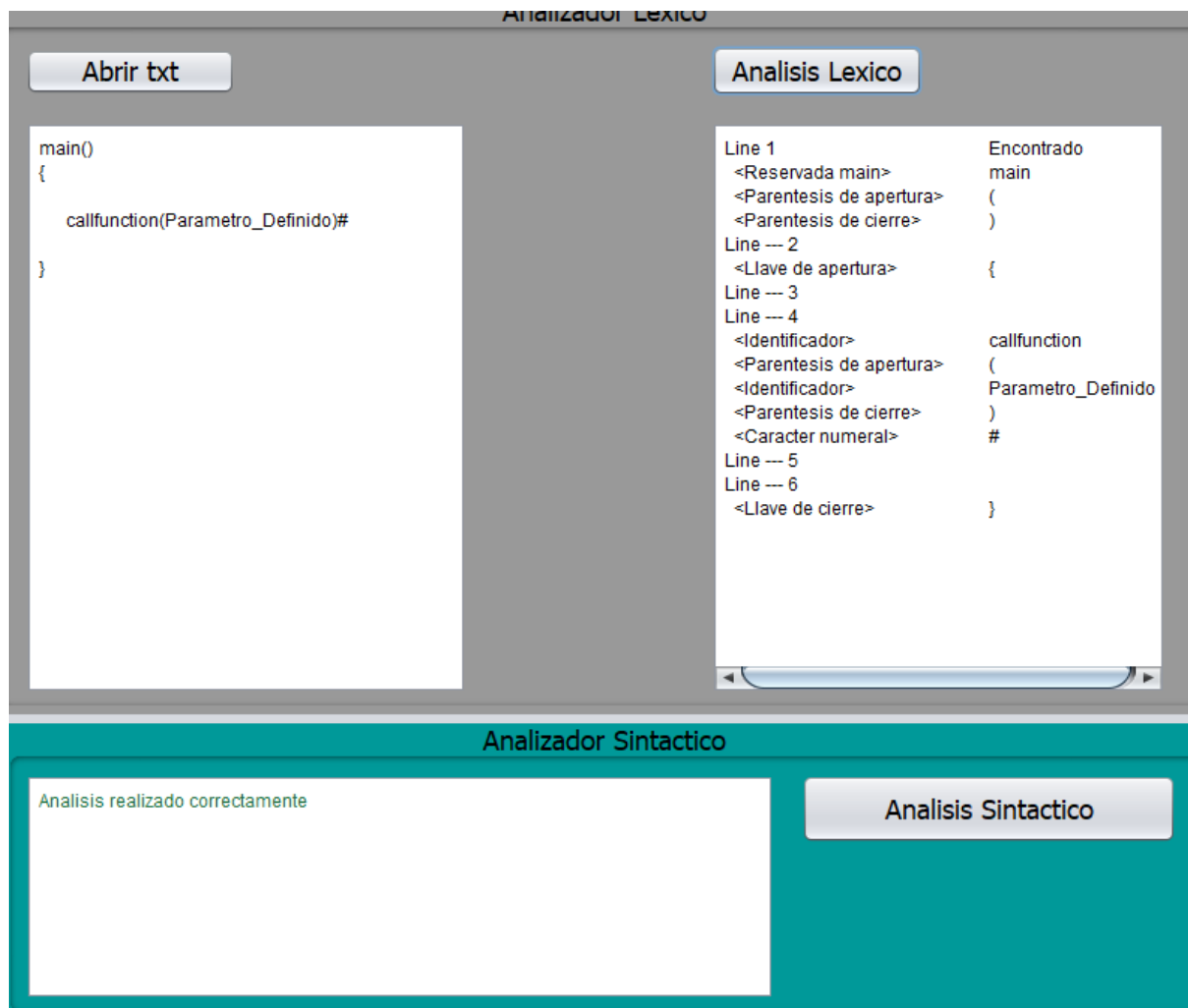
Analisis Lexico

<Numero>	14
<Caracter numeral>	#
Line --- 7	
<Llave de cierre>	}
Line --- 8	
<Llave de cierre>	}
<Reservada else>	else
Line --- 9	
<Llave de apertura>	{
Line --- 10	
<Tipo de dato>	int
<Identificador>	pi
<Operador igual>	=
<Parentesis de apertura>	(
<Numero>	12
<Operador suma>	+
<Identificador>	numero
<Parentesis de cierre>)
<Caracter numeral>	#
Line --- 11	
<Llave de cierre>	}
Line --- 12	

Analizador Sintactico

Analisis realizado correctamente

Analisis Sintactico



Descripción del problema.

Realizar un analizador léxico y sintáctico utilizando Java con las librerías JFlex y Cup. El mismo debe permitir lecturas de archivos de texto

Diseño del programa

Se utiliza una IU básica para ser visualmente más amigable para las pruebas, usando tester de EBNF primero realice la gramática para luego empezar con java.

Librerías usadas

Para la creación del proyecto se utilizaron java-cup11b.jar y JFlex.jar

Análisis de resultados:

Se alcanzaron parcialmente todos los objetivos, tuve problemas para retornar números o identificadores en las funciones, existe el token sin embargo cuando hago el no terminal (RETURN ::= Return Identificador P_coma) no reconoce el no terminal pero cada token por individual si.

Cometí el error de empezar a realizar toda la gramática las no terminales y producciones en un tester web de EBNF y cuando hice el cambio a jflex en jcup varía el sintaxis y tuve que modificar y simplificar muchas reglas porque no podía usarlas como originalmente está en la gramática por lo que invertí mucho tiempo innecesariamente.

Detecta errores de sintaxis pero no tiene la función de modo pánico.