

Universidad de San Carlos de Guatemala
Centro Universitario de Occidente CUNOC
Ingeniería en Ciencias y Sistemas
Laboratorio de Lenguajes Formales y de Programación.

Practica 1

Jorge Daniel Sacor Barrios 201931582

Presentación.

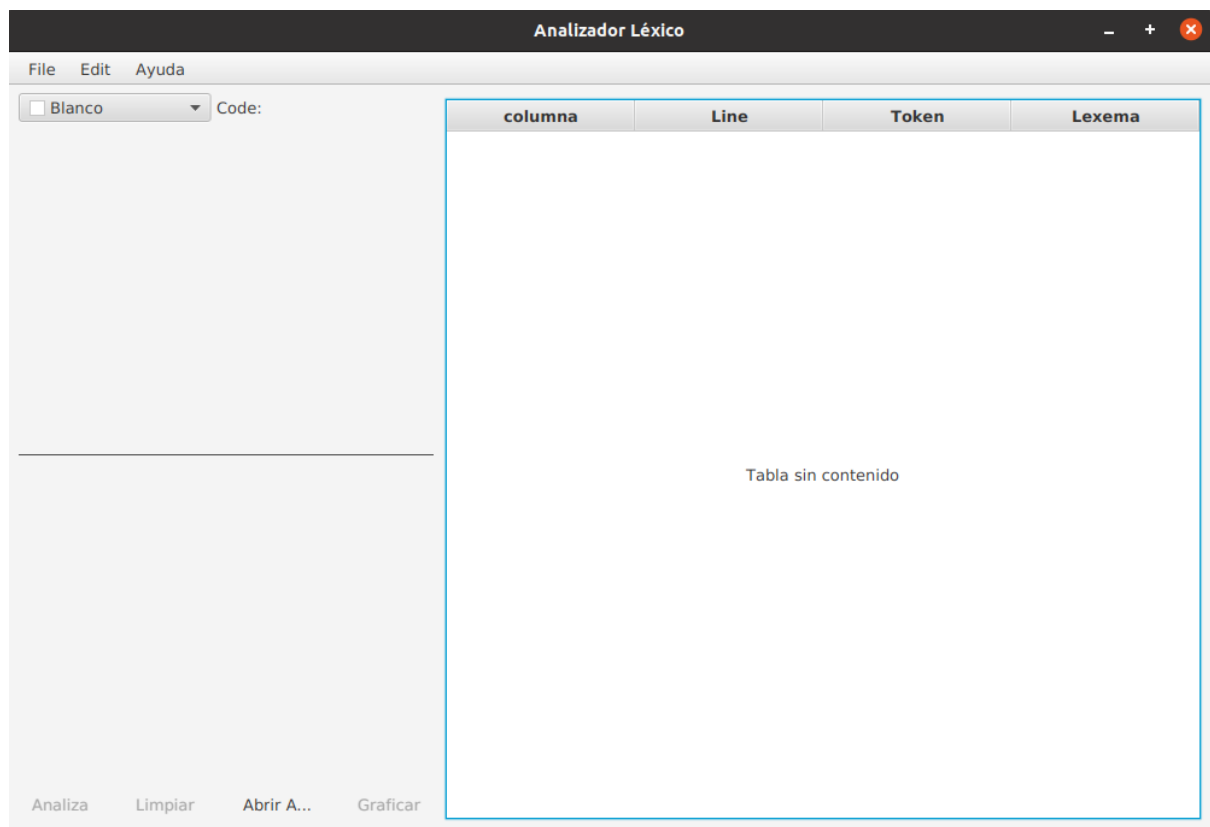
Analizador lexico en el entorno de desarrollo de python

Esta es una herramienta en su primera versión que contiene como inicio un analizador léxico que analiza de primera mano el lenguaje de programación python. Como un compilador está compuesto por analizador léxico y sintáctico lo cual la segunda parte es el analizador sintáctico que nos asemejara lo más posible a nuestro objetivo que es el lenguaje de programación python y como lenguaje entender la sintaxis y orden.

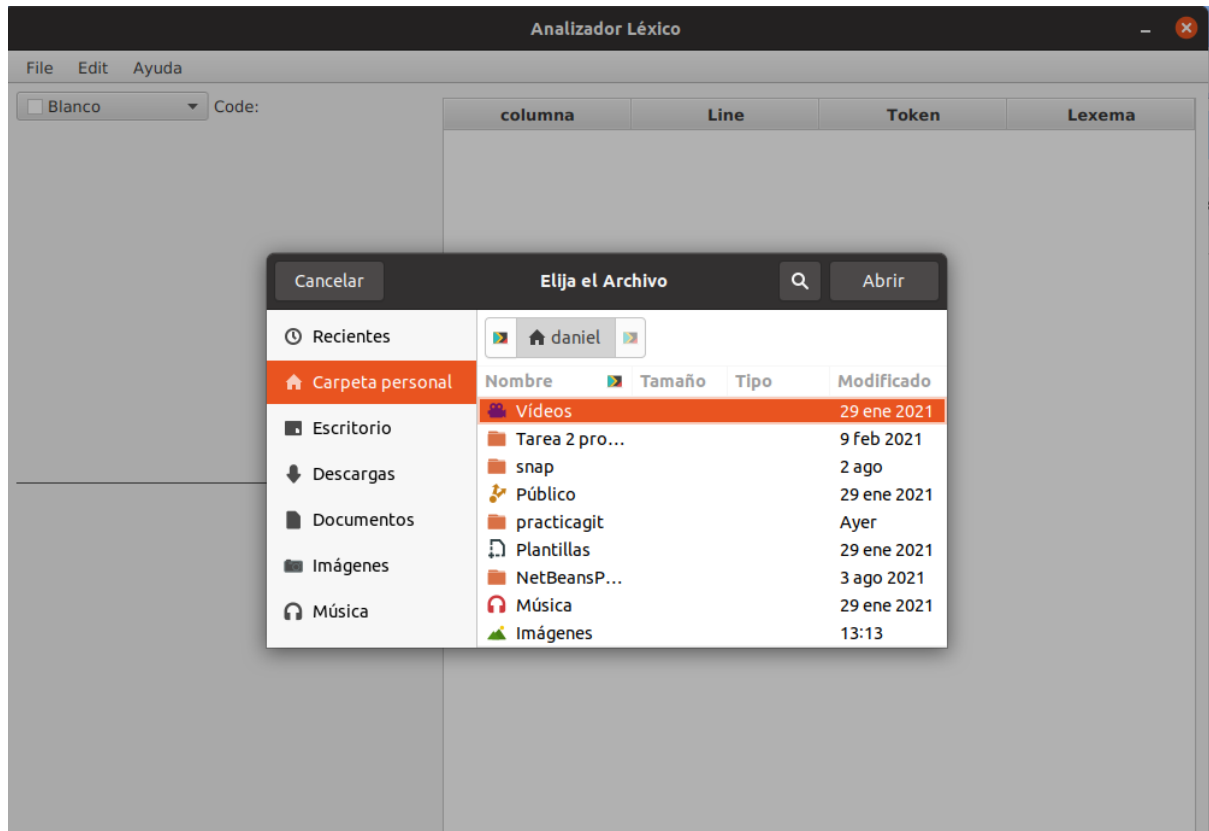
Uso:

Este es un sistema muy fácil e intuitivo que no necesita mayor requerimiento para entender lo que está pasando en tiempo de ejecución.

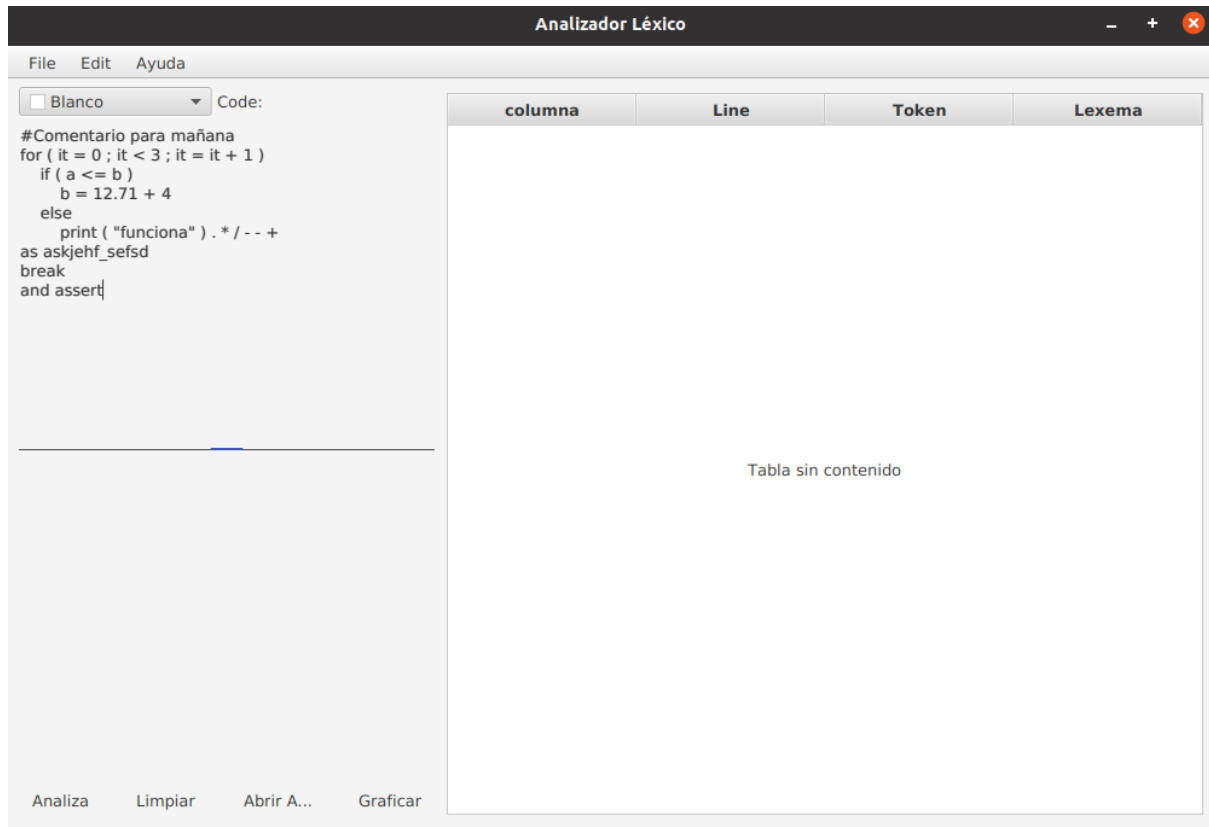
Comenzamos con el inicio de la aplicación en donde nos muestra un panel con diferentes características y apartados que nos indica que es lo que pasará con el proceso.



Como siguiente punto nosotros podemos ver que en la parte inferior tenemos un apartado de botones en los cuales tenemos 4 botones y solamente 1 está habilitado. El cual es “Abrir Archivo” si nosotros le damos click nos abrirá una pestaña nueva en la cual podremos seleccionar nuestro documento.



También podemos ver que podemos escribir directamente en el apartado donde nos muestra el lado superior izquierdo en donde podremos escribir directamente el lenguaje que necesitemos.



En la parte inferior podemos ver que se nos habilitan nuestros 3 botones. El que nos interesa es “Analizar” y es ese en el cual nosotros podemos ver que se hará todo el análisis que separara los tokens por tipos.

Como podemos ver si nosotros le damos click podemos observar que nuestro analizador léxico está funcionando correctamente separando los tipos de tokens y almacenarlos en una tabla que sería nuestro reporte y coloreando las palabras según el orden de la tabla

Analizador Léxico

File Edit Ayuda

Blanco Code:

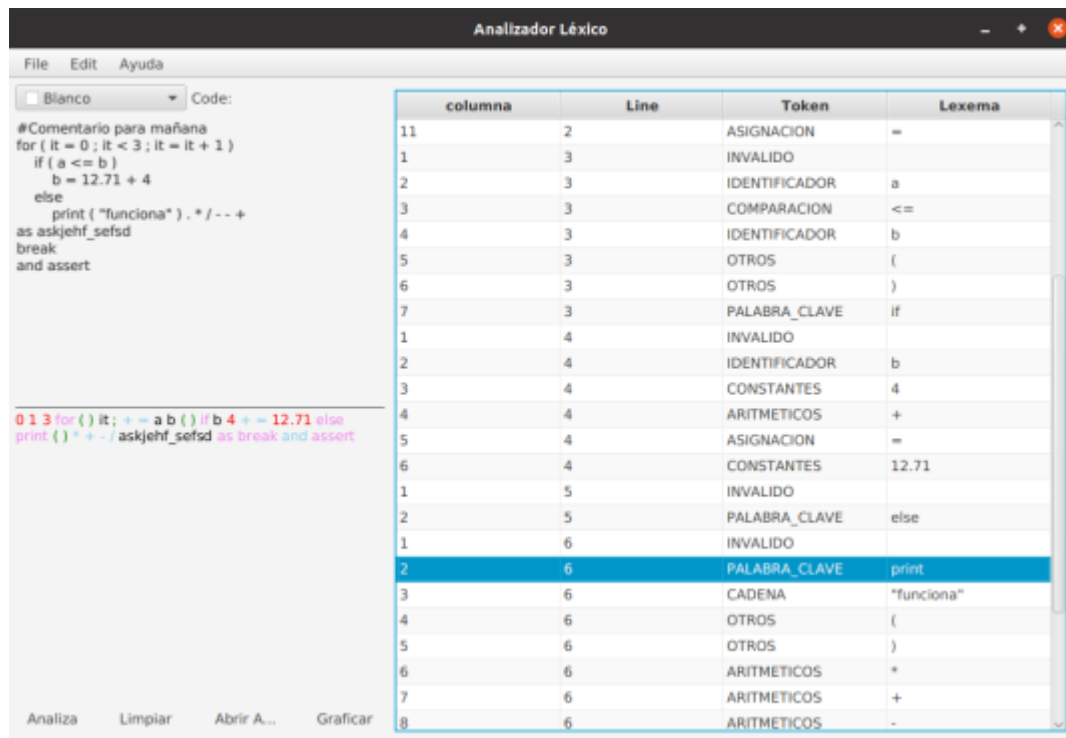
```
#Comentario para mañana
for ( it = 0 ; it < 3 ; it = it + 1 )
    if ( a <= b )
        b = 12.71 + 4
    else
        print ( "funciona" ) . * / - - +
as askjehf_sefsd
break
and assert
```

columna	Line	Token	Lexema
1	2	CONSTANTES	0
2	2	CONSTANTES	1
3	2	CONSTANTES	3
4	2	PALABRA_CLAVE	for
5	2	OTROS	(
6	2	OTROS)
7	2	IDENTIFICADOR	it
8	2	OTROS	;
9	2	ARITMETICOS	+
10	2	COMPARACION	<
11	2	ASIGNACION	=
1	3	INVALIDO	
2	3	IDENTIFICADOR	a
3	3	COMPARACION	<=
4	3	IDENTIFICADOR	b
5	3	OTROS	(
6	3	OTROS)
7	3	PALABRA_CLAVE	if
1	4	INVALIDO	
2	4	IDENTIFICADOR	b
3	4	CONSTANTES	4
4	4	ARITMETICOS	+
5	4	ASIGNACION	=
6	4	CONSTANTES	12.71

Analiza Limpiar Abrir A... Graficar

¿Cómo graficamos?

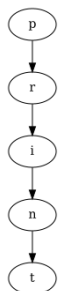
Pues es sencillo ya que podemos graficar cualquier token que nosotros seleccionemos en nuestra tabla la cual está separada con los tipos que se encontraron.



columna	Line	Token	Lexema
11	2	ASIGNACION	=
1	3	INVALIDO	
2	3	IDENTIFICADOR	a
3	3	COMPARACION	<=
4	3	IDENTIFICADOR	b
5	3	OTROS	(
6	3	OTROS)
7	3	PALABRA_CLAVE	if
1	4	INVALIDO	
2	4	IDENTIFICADOR	b
3	4	CONSTANTES	4
4	4	ARITMETICOS	+
5	4	ASIGNACION	=
6	4	CONSTANTES	12.71
1	5	INVALIDO	
2	5	PALABRA_CLAVE	else
1	6	INVALIDO	
2	6	PALABRA_CLAVE	print
3	6	CADENA	"funciona"
4	6	OTROS	(
5	6	OTROS)
6	6	ARITMETICOS	*
7	6	ARITMETICOS	+
8	6	ARITMETICOS	-

Si le damos click a nuestro botón de graficar nos abrirá una pestaña en html en donde nos mostrará el resultado de nuestro lexema graficado por nodos.

Grafo Generado

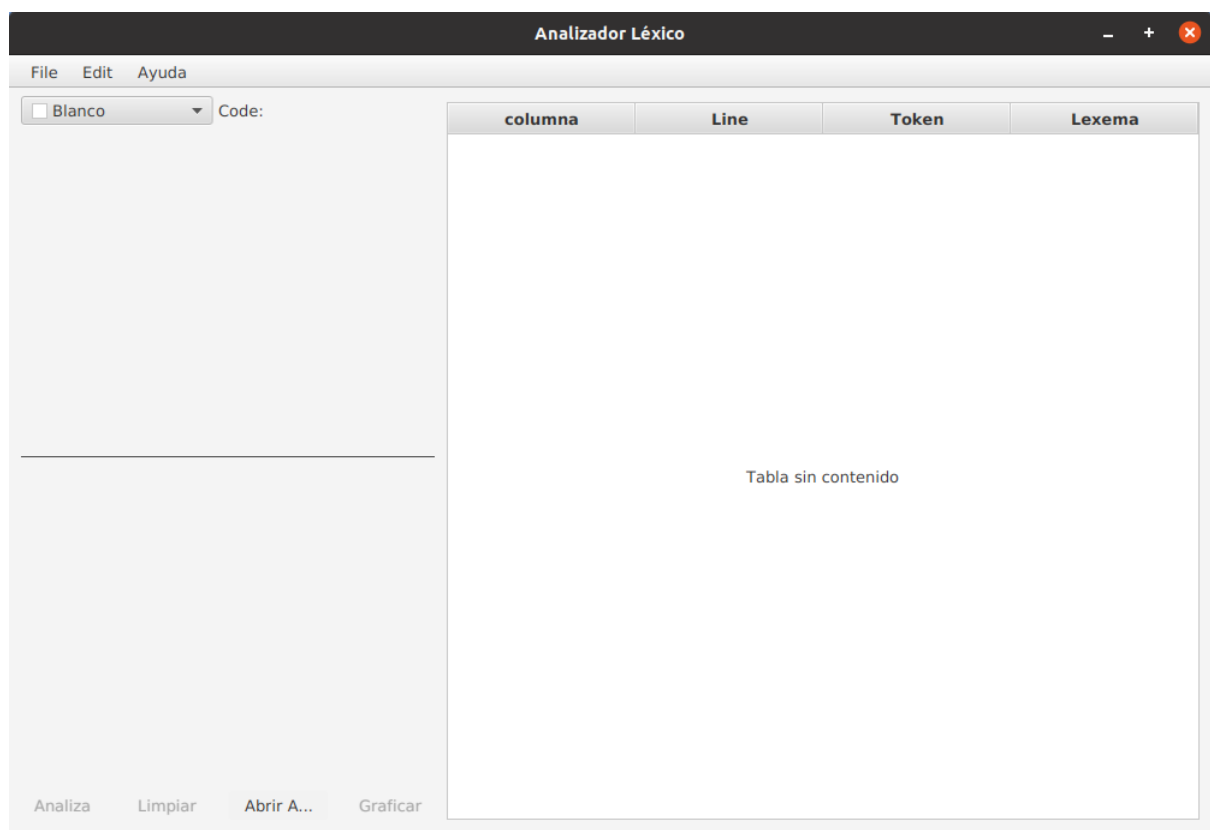


Podemos ver en la parte inferior izquierda que nos aparece nuestro código pintado de colores como se muestra en la tabla de colores correspondiente a nuestro documento guía.

```
0 1 3 for ( ) it ; + = a b ( ) if b 4 + = 12.71 else
print ( ) * + - / askjehf_sefsd as break and assert
```

Analiza Limpiar Abrir A... Graficar

Si nosotros le damos click en el botón de limpiar podemos ver que limpia todos los espacios para poder volver a usarlos con otro texto y nos damos cuenta que es asertivo.



Ejemplos de archivo de entrada tiene que ser con la terminacion .txt

```
#Comentario para mañana
for ( it = 0 ; it < 3 ; it = it + 1 )
    if ( a <= b )
        b = 12.71 + 4
    else
        print ( "funciona" ) . * / - - +
as askjehf_sefsd
break
and assert
```

```
def if "cadena" : ( 4 + 5 )
mi_id_
#comentario
while
for
>=
<

#un error se puede presentar como la siguiente linea

"mi-cadena-que-no_cierra
```

```
def if "cadena" : ( 4 + 5 )
mi_id_
#comentario
while
for
>=
<
!<
==
else
print ( "funciona" ) . * / - - +

#un error se puede presentar como la siguiente linea

"mi-cadena-que-no_cierra
```



```

def if "cadena" : ( 4 + 5 )
mi_id_
#comentario
while
for
>=
<
!<
==
else
print ( "funciona" ) . * / - - +
as break

```

```

def if "cadena" : ( 4 + 5 )
mi_id_
#comentario
while
for
>=
<
!<
==
else
print ( "funciona" ) . * / - - +
as break
12
12.5 / * +
while ( a + b ) = 100.15
{
strong
}
asbbd [ 1 + 2 + 2 + asea_asea ]
_Mivvariable
_sease_
ajsebkfabse + {
}

```

