

Analizador Léxico Sintáctico

Manual Técnico para Usuarios

Desarrollador



Juan Cardona



**UNIVERSIDAD
DE ANTIOQUIA**

1 8 0 3

Facultad de Ingeniería

Departamento de ingeniería de sistemas

Teoría de Lenguajes

2021-1

Aspectos Técnicos

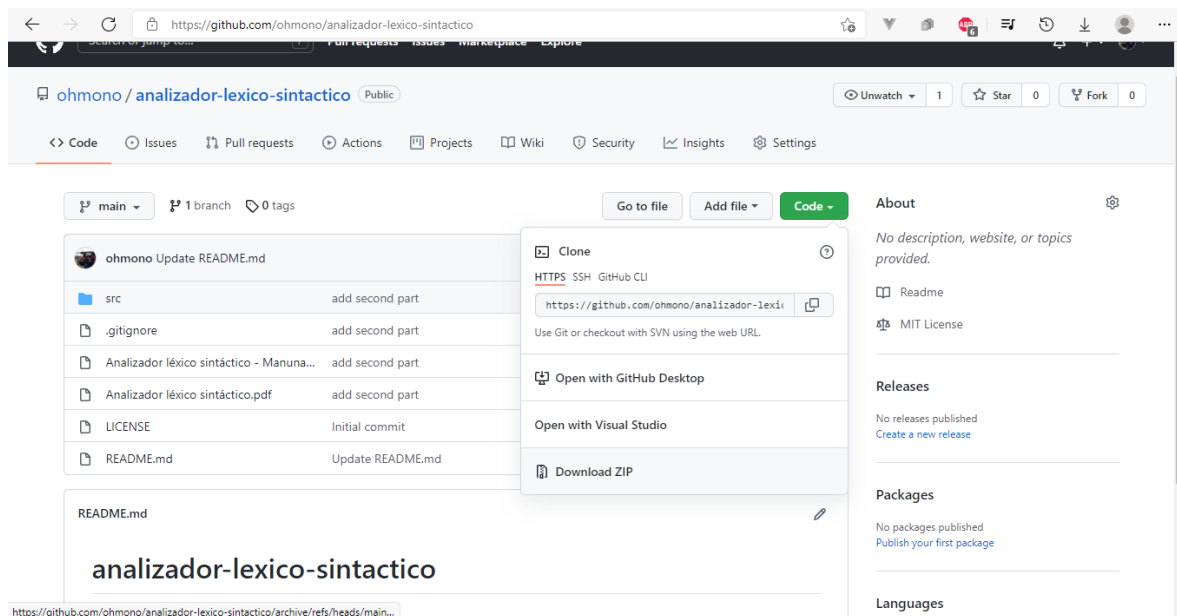
Este documento es el manual de técnico del **analizador léxico sintáctico** desarrollado por Juan Cardona.

El lenguaje de programación que utilice fue **Python 3.8.5** en el ambiente de desarrollo **Visual Studio Code** importando las librerías **PyQt5** (diseño de ambiente gráfico), **webbrowser** (dirección de url), y **re** (manejo para expresiones regulares).

No es necesario tener ningún programa instalado previo a la ejecución del **analizador léxico sintáctico**, pero si se quiere hacer alguna modificación es necesario tener instalado los programas y librerías anteriormente listados.

Obtener el Analizador Léxico Sintáctico

Para obtener el **Analizador Léxico Sintáctico** ingrese al repositorio de GitHub:
<https://github.com/ohmono/analizador-lexico>



En el botón Code, al dar clic, seleccione la opción Download ZIP, para obtener el archivo con el programa.

Descomprima el archivo e ingrese a la carpeta analizador-léxico

analizador-lexico-sintactico				
Nombre	Fecha de modificación	Tipo	Tamaño	
.git	8/10/2021 4:55 a. m.	Carpeta de archivos		
src	8/10/2021 4:47 a. m.	Carpeta de archivos		
.gitignore	8/10/2021 4:54 a. m.	Documento de te...	1 KB	
Analizador léxico sintáctico - Manual té...	8/10/2021 4:57 a. m.	Microsoft Edge P...	429 KB	
Analizador léxico sintáctico.pdf	8/10/2021 4:52 a. m.	Microsoft Edge P...	529 KB	
LICENSE	9/05/2021 8:39 p. m.	Archivo	2 KB	
README.md	8/10/2021 4:55 a. m.	Archivo de origen ...	1 KB	

Para acceder a el archivo ejecutable, ingrese en la carpeta **src/dist**

analizador-lexico-sintactico > src > dist				
Nombre	Fecha de modificación	Tipo	Tamaño	
main.exe	8/10/2021 4:52 a. m.	Aplicación	35.469 KB	
prueba.txt	8/10/2021 4:51 a. m.	Documento de te...	1 KB	

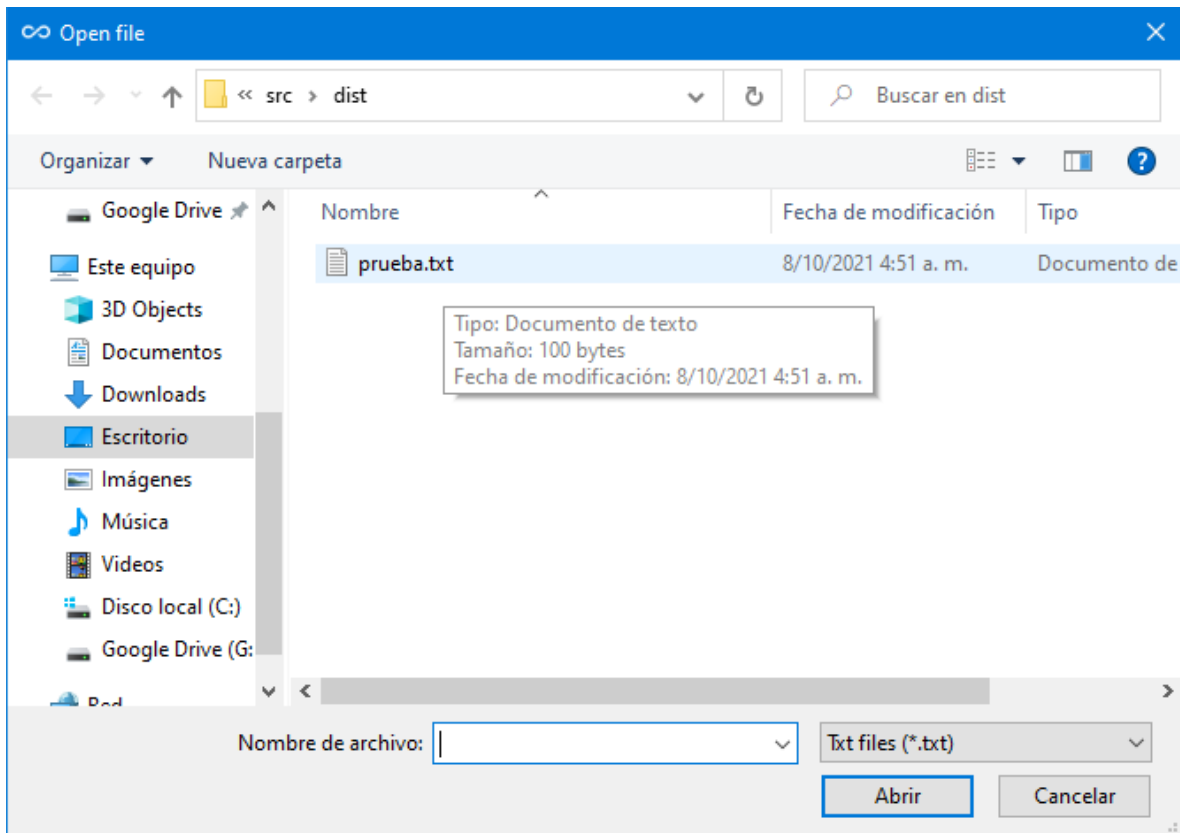
El archivo **main.exe** corresponde al **analizador léxico sintáctico**.

Usar el Analizador Léxico Sintáctico

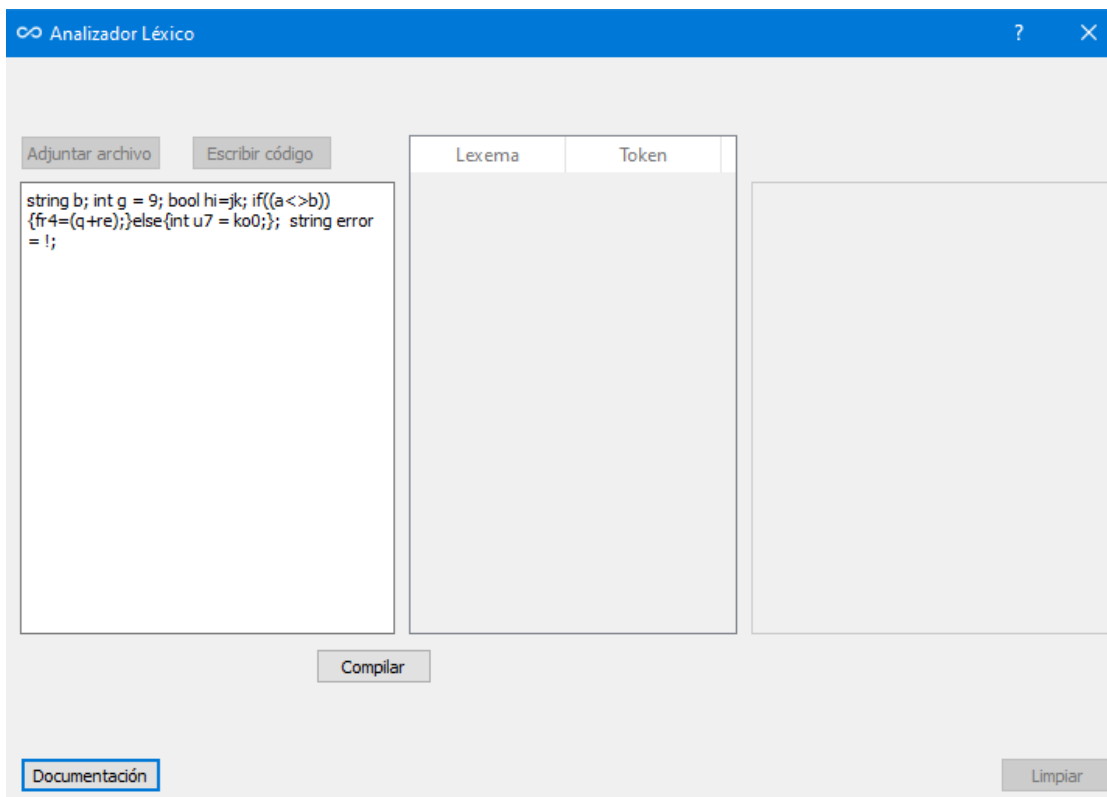
The screenshot shows a web application titled 'Analizador Léxico'. It features a blue header bar with the title and navigation icons. The main interface is divided into three vertical sections. The left section contains a large text area for input, with buttons 'Adjuntar archivo' and 'Escribir código' above it. The middle section is a table with two columns, 'Lexema' and 'Token', and a large empty area below the header. The right section contains another large text area. At the bottom, there is a 'Compilar' button in the center, a 'Documentación' link on the left, and a 'Limpiar' button on the right.

Lexema	Token
--------	-------

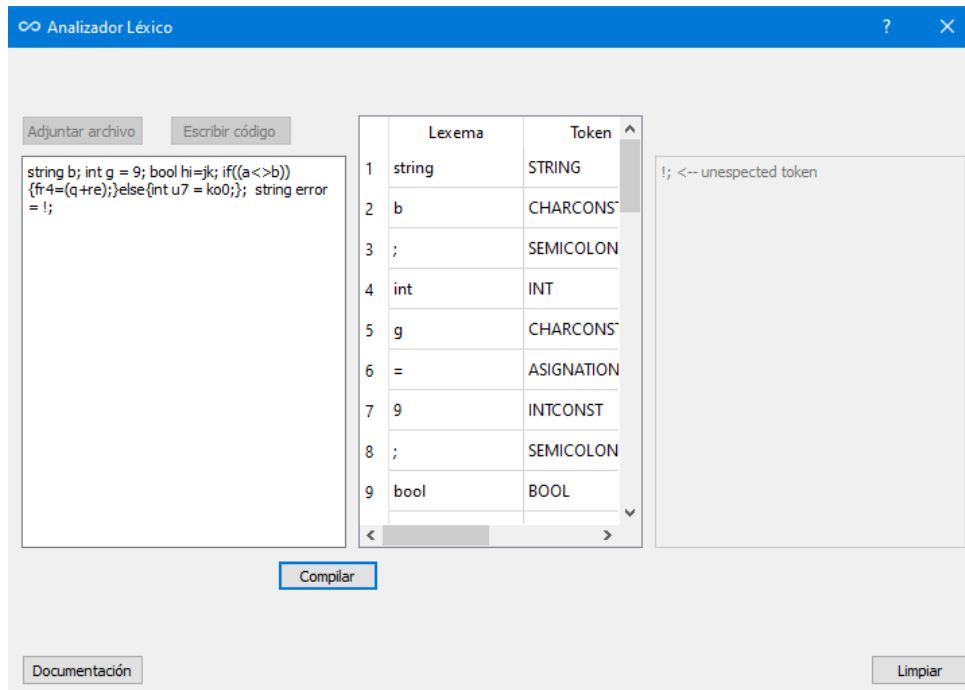
Con el botón **Adjuntar archivo** podemos adjuntar un archivo .txt para analizar



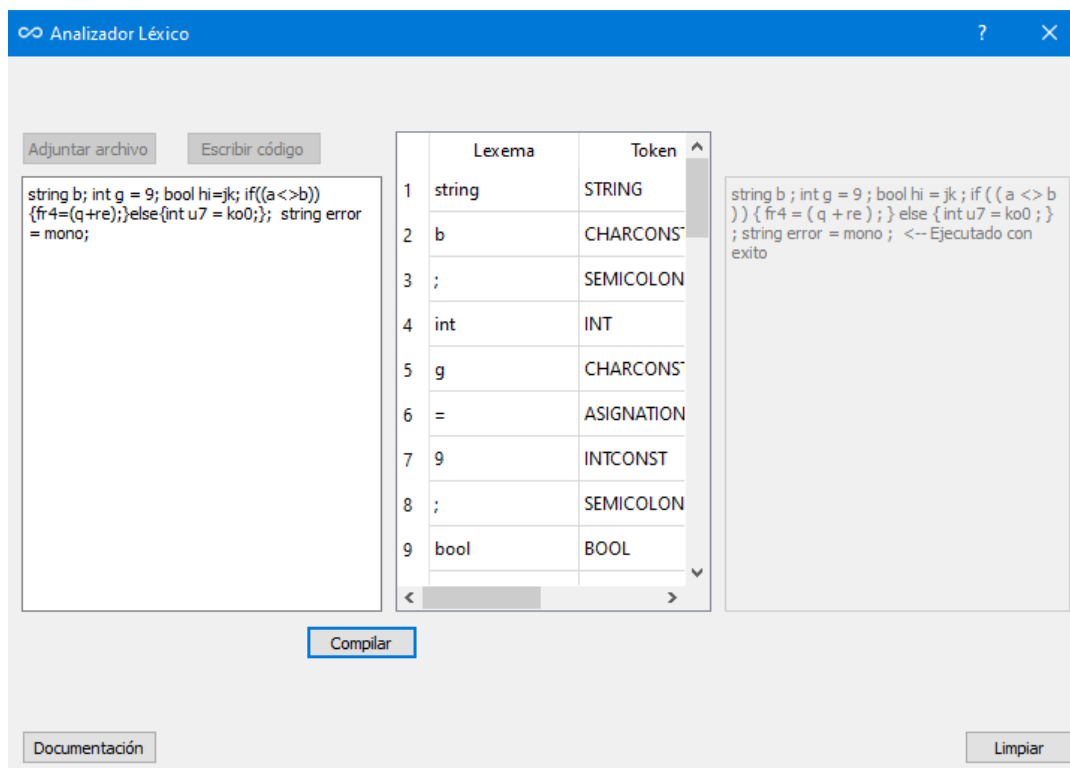
Luego de seleccionar un archivo .txt vemos que se despliega su contenido en el programa



Y al darle en el botón **compilar** se nos mostrara una lista con los lexemas y sus tokens correspondientes y los correspondientes errores de sintaxis.

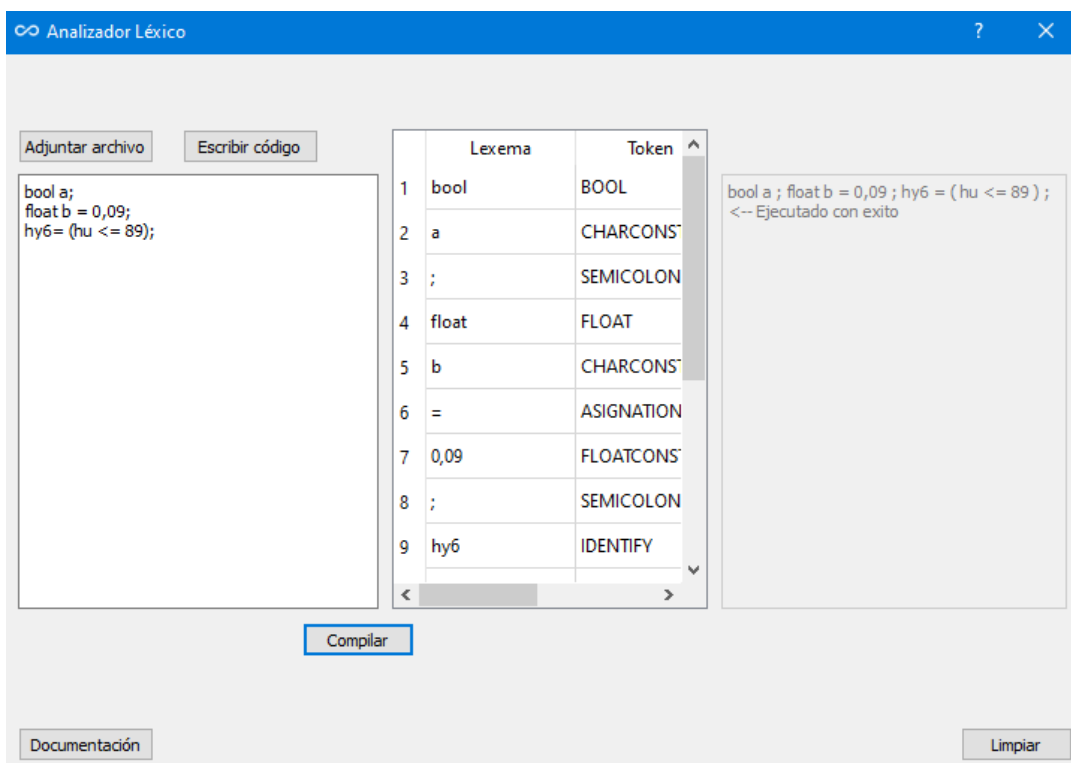
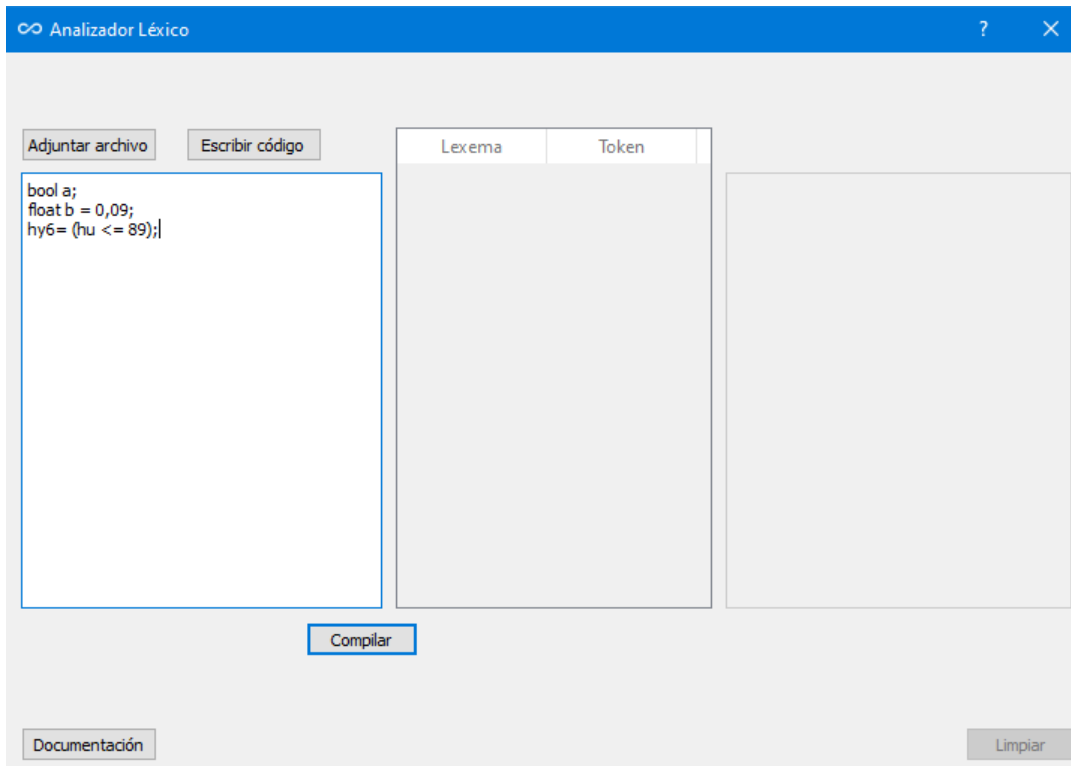


Podemos editar el código directamente en la aplicación y corregir el token '!' que no pertenece a nuestro lenguaje.



El botón **Limpiar** nos permite reiniciar la interfaz para ser usada de nuevo.

Con el botón **Escribir código** podemos ingresar manualmente el código a analizar



Por último, el botón **Documentación** nos direcciona al repositorio de GitHub donde se aloja el **Analizador Léxico Sintáctico**.

Lista de lexemas, tokens y terminales que reconoce y asigna el **Analizador Léxico Sintáctico**:

Lexema	Token	Terminales
esle	ELSE	ELSE
if	IF	IF
int	INT	T
float	FLOAT	
bool	BOOL	
string	STRING	
L+	CHARCONST	V
L+(L+D)*	IDENTIFY	
D+	INTCONST	D
D+,D+	FLOATCONST	
+	PLUS	O
-	MINUS	
*	MULTIPLICATION	
/	DIVISION	
=	ASIGNATION	=
==	BOOLCOPARISON	B
<	MINOR	
<=	MINOREQUAL	
<>	DIFFERENT	
>	HIGHER	
>=	HIGHEREQUAL	
(OPENPARENTHESIS	(
)	CLOSEPARENTHESIS)
{	OPENBRACKETS	{
}	CLOSEBRACKETS	}
;	SEMICOLON	;

- D (cualquier carácter numérico o dígito)
- L (cualquier carácter alfabético, en minúscula o mayúscula, o letra)

Gramática con la que se identifica si las secuencias de tokens identificadas son válidas:

1. $\langle S \rangle \rightarrow TV\langle A \rangle; \langle S \rangle$
2. $\langle S \rangle \rightarrow V = \langle X \rangle; \langle S \rangle$
3. $\langle S \rangle \rightarrow IF(\langle X \rangle B \langle X \rangle) \{ \langle S \rangle \} ELSE \{ \langle S \rangle \}; \langle S \rangle$
4. $\langle S \rangle \rightarrow \lambda$
5. $\langle A \rangle \rightarrow = \langle B \rangle$
6. $\langle A \rangle \rightarrow \lambda$
7. $\langle B \rangle \rightarrow V$
8. $\langle B \rangle \rightarrow D$
9. $\langle X \rangle \rightarrow V$
10. $\langle X \rangle \rightarrow D$
11. $\langle X \rangle \rightarrow (\langle X \rangle \langle Z \rangle \langle X \rangle)$
12. $\langle Z \rangle \rightarrow O$
13. $\langle Z \rangle \rightarrow B$

Bibliografía

- Flórez Rueda, R. (2010). Introducción a los compiladores. Medellín, Colombia: Universidad de Antioquia
- Regular Expression Quick Reference. Retrieved May 10, 2021, from http://regexrenamer.sourceforge.net/help/regex_quickref.html
- PyQt5 - Qt for Python. Retrieved May 10, 2021, from <https://doc.qt.io/qtforpython/>