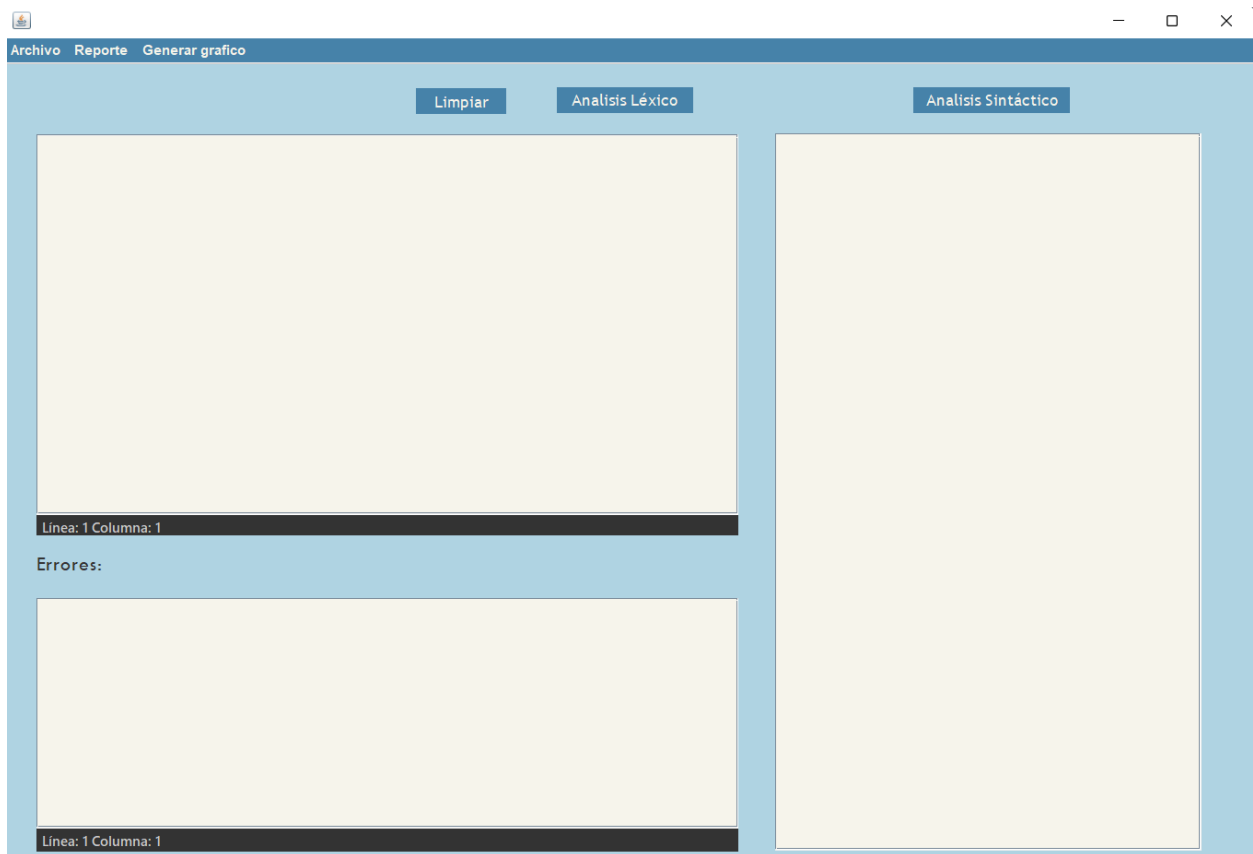


Manual de Usuario

Interface principal del software



Esta diseñada con la librería swing de java y está compuesta principalmente por 3 JTextPane, el que esta en la parte superior izquierda nos permite ya sea escribir o cargar código del lenguaje Python, eso incluye editarlo.

En el que se encuentra abajo se mostraran los resultados del análisis léxico de dicho código y para eso debemos hablar de la funcionalidad de los botones:

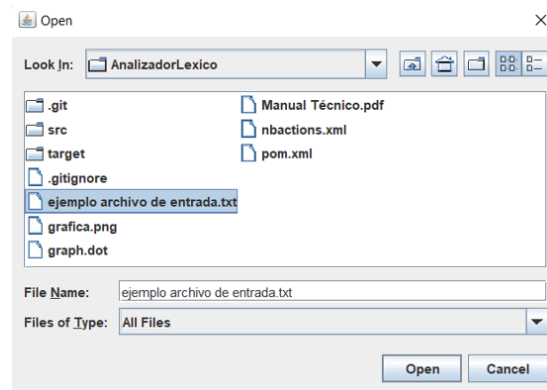


Limpiar: nos permite borrar cualquier texto ingresado al editor, así como también los datos resultantes de los análisis en los otros JTextpane.

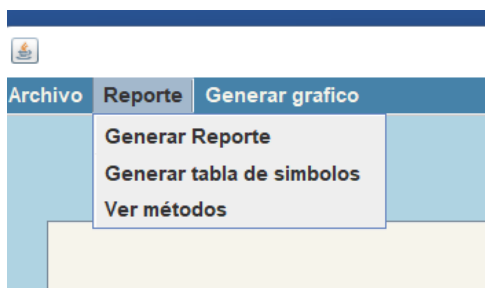
Análisis Léxico: Se encarga de realizar el análisis del texto ingresado en el editor.

Análisis Sintáctico: realiza el análisis sintáctico de los tokens obtenidos del análisis léxico, dicho eso se necesita de un análisis léxico previo o de lo contrario no realizara ningún análisis y saltara un aviso confirmándolo.

En la barra de menú encontramos en el menú desplegable archivo la opción de cargar un archivo de código Python.



Apartado de reportes

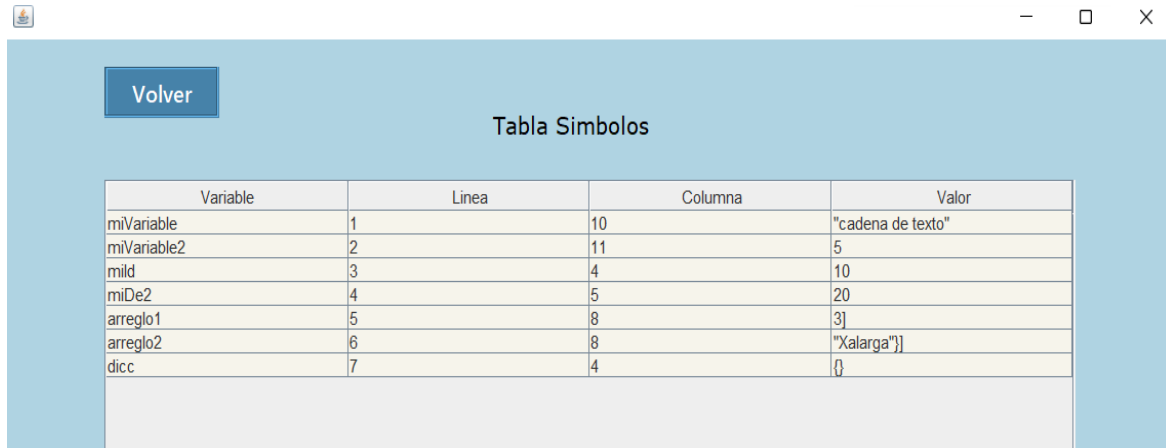


Generar reportes: nos mostrara en forma detallada los tokens obtenidos incluyendo línea, columna, patrón y lexema de cada token, también si se encontraron errores léxicos.

[Volver](#)

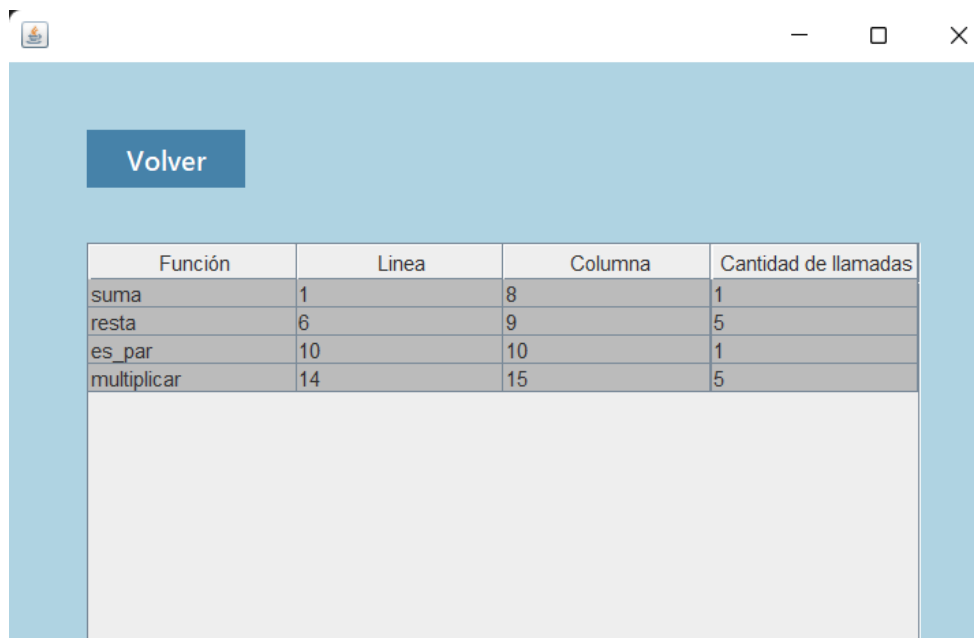
Token	Patron	Lexema	Linea	Columna
PALABRA RESERVADA	def	def	1	4
PALABRA RESERVADA	if	if	1	7
CADENA	\'[^\']*\'	"cadena "	1	17
OTROS_OPERADORES	:	:	1	19
OTROS_OPERADORES	((1	21
ENTERO	[0-9]+	4	1	23
OPERADOR_ARITMETICO	+	+	1	24
ENTERO	[0-9]+	5	1	27
OTROS_OPERADORES))	1	29
IDENTIFICADOR	mi_id	mi_id	2	6
COMENTARIO	^!s*#[^#]*\$	#comentario	4	1
PALABRA RESERVADA	while	while	5	6
PALABRA RESERVADA	for	for	6	4
OPERADOR_COMPARACION	>=	>=	7	2
OPERADOR_COMPARACION	<	<	8	69
CADENA	\'[^\']*\'	'cadena en comillas simples'	9	29
COMENTARIO	^!s*#[^#]*\$	#un error se puede presentar ..	12	1
ERROR_LEXICO	" mi cadena que no cierra	" mi cadena que no cierra	14	26
PALABRA RESERVADA	def	def	1	4
PALABRA RESERVADA	if	if	1	7
CADENA	\'[^\']*\'	"cadena "	1	17
OTROS_OPERADORES	:	:	1	19
OTROS_OPERADORES	((1	21
ENTERO	[0-9]+	4	1	23
OPERADOR_ARITMETICO	+	+	1	24
ENTERO	[0-9]+	5	1	27
OTROS_OPERADORES))	1	29
IDENTIFICADOR	mi_id	mi_id	2	6
COMENTARIO	^!s*#[^#]*\$	#comentario	4	1
PALABRA RESERVADA	while	while	5	6
PALABRA RESERVADA	for	for	6	4
OPERADOR_COMPARACION	>=	>=	7	2
OPERADOR_COMPARACION	<	<	8	69
CADENA	\'[^\']*\'	'cadena en comillas simples'	9	29
COMENTARIO	^!s*#[^#]*\$	#un error se puede presentar ..	12	1
ERROR_LEXICO	" mi cadena que no cierra	" mi cadena que no cierra	14	26

Generar tabla de símbolos: acá se mostrarán las variables definidas y verificadas por el analizador sintáctico.



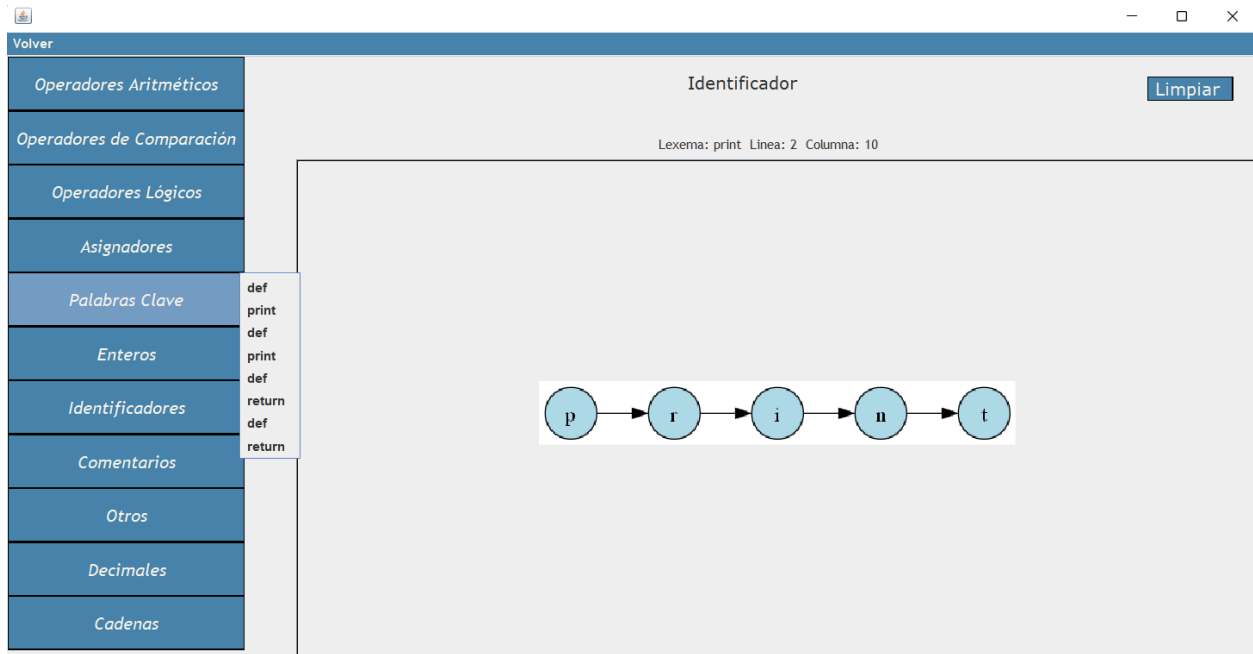
Variable	Linea	Columna	Valor
miVariable	1	10	"cadena de texto"
miVariable2	2	11	5
mild	3	4	10
miDe2	4	5	20
arreglo1	5	8	3]
arreglo2	6	8	"Xalarga"]]
dicc	7	4	{}

Ver métodos: Mostrara una tabla con el nombre de cada método declarado indicando línea y columna y cuantas llamadas de dicho método se han realizado.



Función	Linea	Columna	Cantidad de llamadas
suma	1	8	1
resta	6	9	5
es_par	10	10	1
multiplicar	14	15	5

Generar gráfico:



Por último, en este apartado tendremos una barra vertical de los tipos de tokens y al seleccionar cualquier token se encargará de graficar el lexema mostrando la línea y columna del token, también cuenta con el botón limpiar para cambiar entre graficas.

Cada JFrame tiene un botón de volver para regresar al editor de texto y realizar cualquier otra acción.