

Manual de Usuario

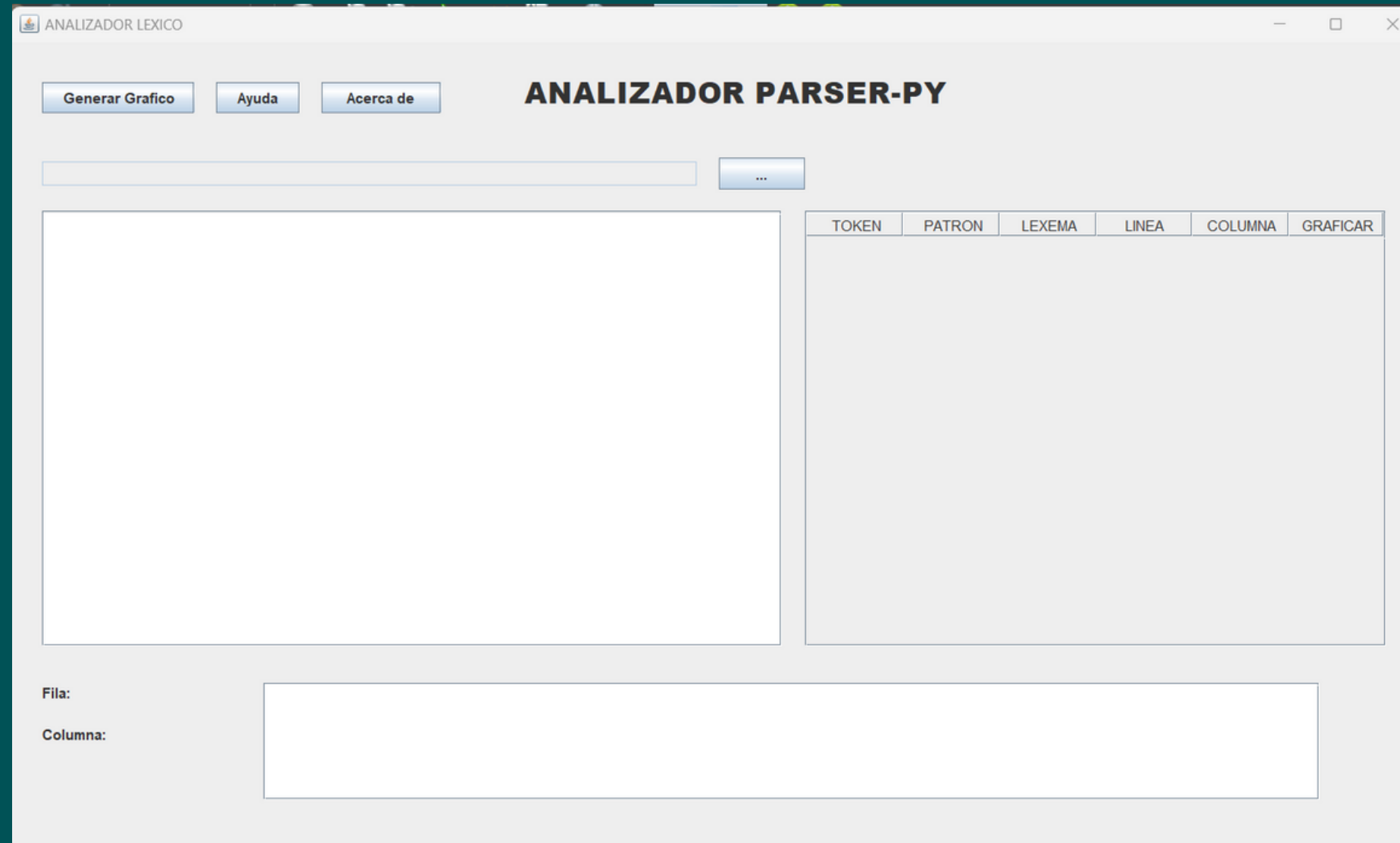
Analizador Lexico -
Lenguajes Formales y de
Programacion



REQUISITOS:

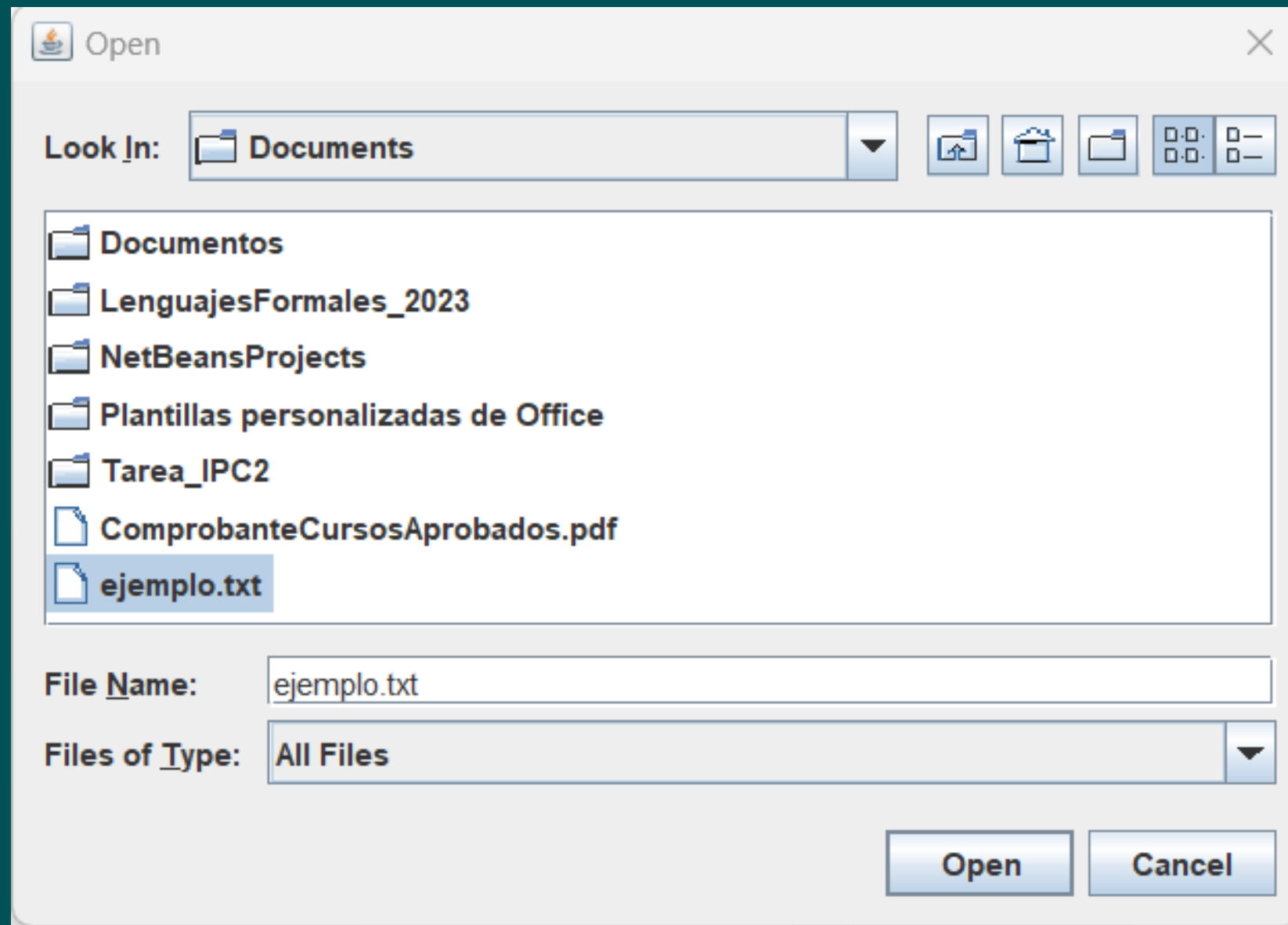
- JDK 17 instalado
- Intel celeron o superior
- 2gb de ram
- 50MB de espacio

Interfaz Grafica



Este es nuestro menú principal tenemos un cuadro de texto es el blanco de la izquierda, podemos ingresar texto y nuestro analizador actuara de forma automatica, de igual manera tenemos un JFileChooser esta abajo del titulo principal hay un boton con tres puntos ahi podemos cargar nuestro archivo

JFileChooser



En el JFileChooser podemos seleccionar nuestro archivo que se analizara. El programa por si mismo buscara la ruta no hay necesidad de nada más y se cargara automaticamente al bloque de texto

Analizador

ANALIZADOR LEXICO

Generar Grafico

Ayuda

Acerca de

ANALIZADOR PARSE-PY

...

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

#un
```

Token	Patron	Lexema	Linea	Columna
PALABRA_RES...	DEF	DEF	0	0
IDENTIFICADOR	IDENTIFICADOR	"CADENA"	0	0
IDENTIFICADOR	IDENTIFICADOR	"	0	0
SIMBOLO	DOS_PUNTOS	:	0	0
SIMBOLO	PARENTESISA...	(0	0
NUMERO	CUATRO	4	0	0
ARITMETICO	SUMA	+	0	0
NUMERO	CINCO	5	0	0
SIMBOLO	PARENTESISISC...)	0	0
COMENTARIO	COMENTARIO	#COMENTARIO	0	0
PALABRA_RES...	WHILE	WHILE	0	0
PALABRA_RES...	FOR	FOR	0	0
COMPARACION	MAYOR_O_IGU...	>=	0	0
COMPARACION	MENOR_QUE	<	0	0
IDENTIFICADOR	IDENTIFICADOR	'CADENA	0	0

Fila: 9

Columna: 3

Token{gramatica=WHILE, lexema=WHILE, fila=0, columna=0}
Token{gramatica=FOR, lexema=FOR, fila=0, columna=0}
Token{gramatica=MAYOR_O_IGUAL_QUE, lexema=>=, fila=0, columna=0}
Token{gramatica=MENOR_QUE, lexema=<, fila=0, columna=0}
Token{gramatica=IDENTIFICADOR, lexema='CADENA, fila=0, columna=0}

Automaticamente el analizador generara los tokens de las palabras ingresadas abajo a la izquierda tenemos el conteo de filas y columnas y abajo tenemos un TextArea de la informacion de tokens

Analizador

ANALIZADOR LEXICO

Generar Grafico

Ayuda

Acerca de

ANALIZADOR PARSE-PY

...

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

#un
```

Token	Patron	Lexema	Linea	Columna
PALABRA RES...	DEF	DEF	0	0
IDENTIFICADOR	IDENTIFICADOR	"CADENA"	0	0
IDENTIFICADOR	IDENTIFICADOR	"	0	0
SIMBOLO	DOS_PUNTOS	:	0	0
SIMBOLO	PARENTESISA...	(0	0
NUMERO	CUATRO	4	0	0
ARITMETICO	SUMA	+	0	0
NUMERO	CINCO	5	0	0
SIMBOLO	PARENTESISC...)	0	0
COMENTARIO	COMENTARIO	#COMENTARIO	0	0
PALABRA RES...	WHILE	WHILE	0	0
PALABRA RES...	FOR	FOR	0	0
COMPARACION	MAYOR_O_IGU...	>=	0	0
COMPARACION	MENOR_QUE	<	0	0
IDENTIFICADOR	IDENTIFICADOR	'CADENA	0	0

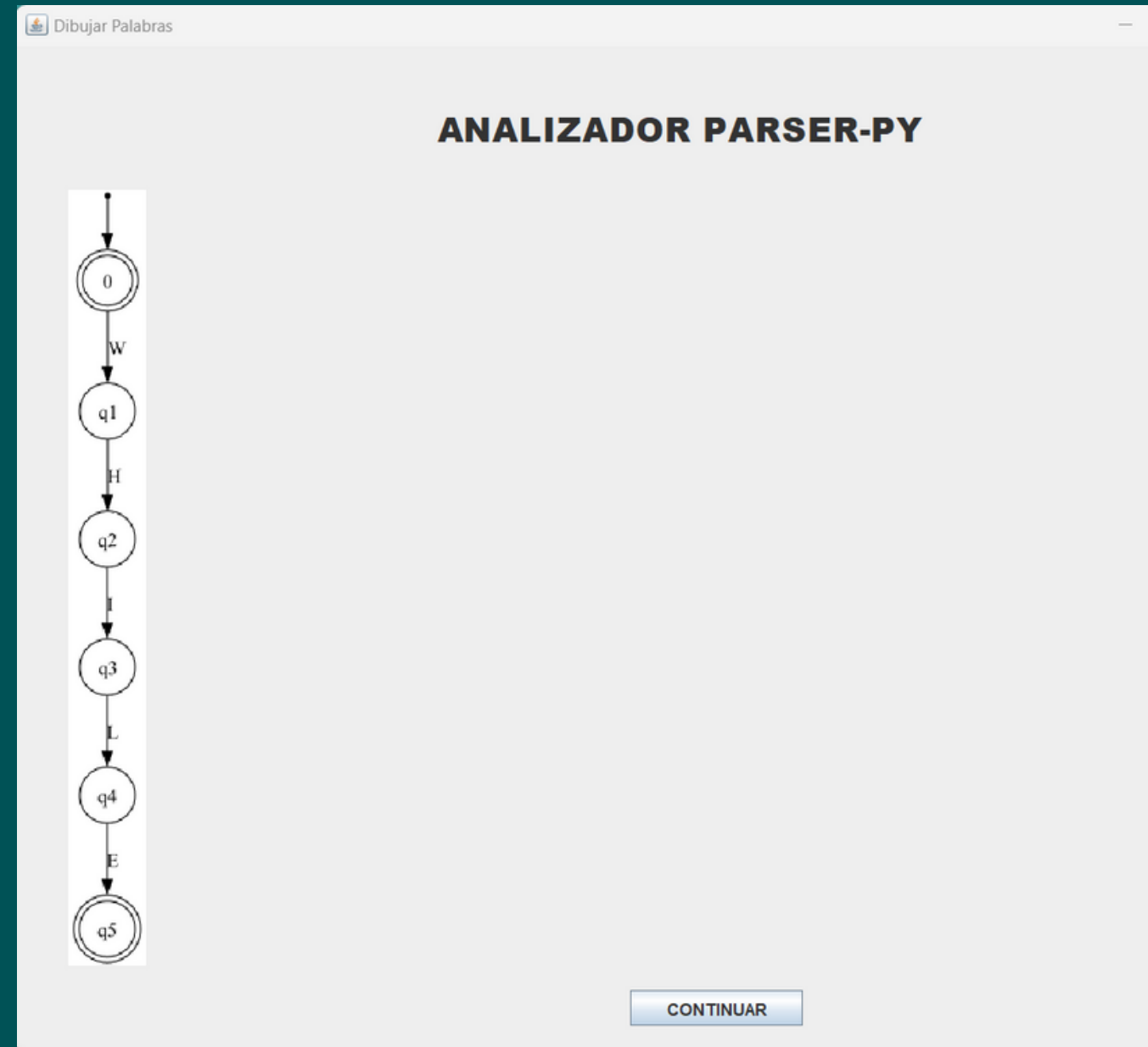
Fila: 9

Columna: 3

Token{gramatica=WHILE, lexema=WHILE, fila=0, columna=0}
Token{gramatica=FOR, lexema=FOR, fila=0, columna=0}
Token{gramatica=MAYOR_O_IGUAL_QUE, lexema=>=, fila=0, columna=0}
Token{gramatica=MENOR_QUE, lexema=<, fila=0, columna=0}
Token{gramatica=IDENTIFICADOR, lexema='CADENA, fila=0, columna=0}

Aca podemos generar el grafico que necesitamos para eso desde la jTable seleccionamos el lexema que queremos generar posterior a eso se generara el grafico de forma automatica.

Analizador



Nos saldra la ventana del grafico que seleccionamos. Para salir le damos continuar y cerrar.

MODULO DE INFORMACION

