

Universidad San Carlos De Guatemala
Centro Universitario De Occidente
División De Ciencias De La Ingeniería
Lenguajes Formales y de Programación
Auxiliar Julio Fernando



Manual de Usuario

202132299, G3rman Alex Ram3rez Limatuj

Quetzaltenango, Octubre de 2023

Manual de usuario

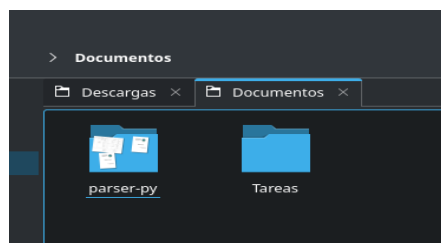
Analizador Sintactico

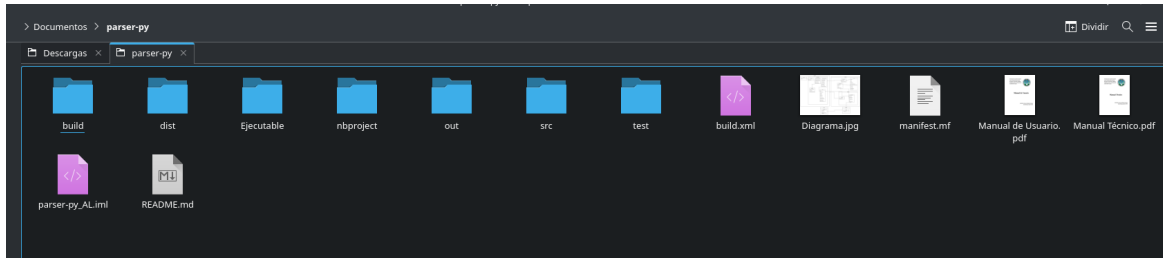
Parse Py

Bienvenido al manual de usuario del programa Parse-py (Analizador Léxico y Sintactico), a continuación, se le darán generalidades del programa para que pueda darle el mejor uso posible a la app:

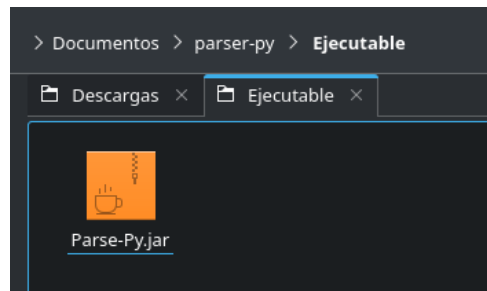
¿Cómo inicio (ejecuto) la app?

Debes ingresar en el directorio (foder) del programa

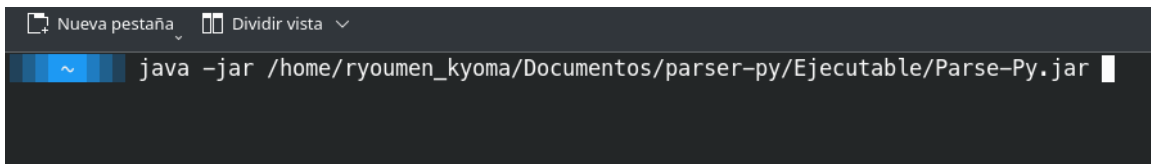




Ingresa a la carpeta de Ejecutable, ya que en ella se encuentra el ejecutable (.jar)

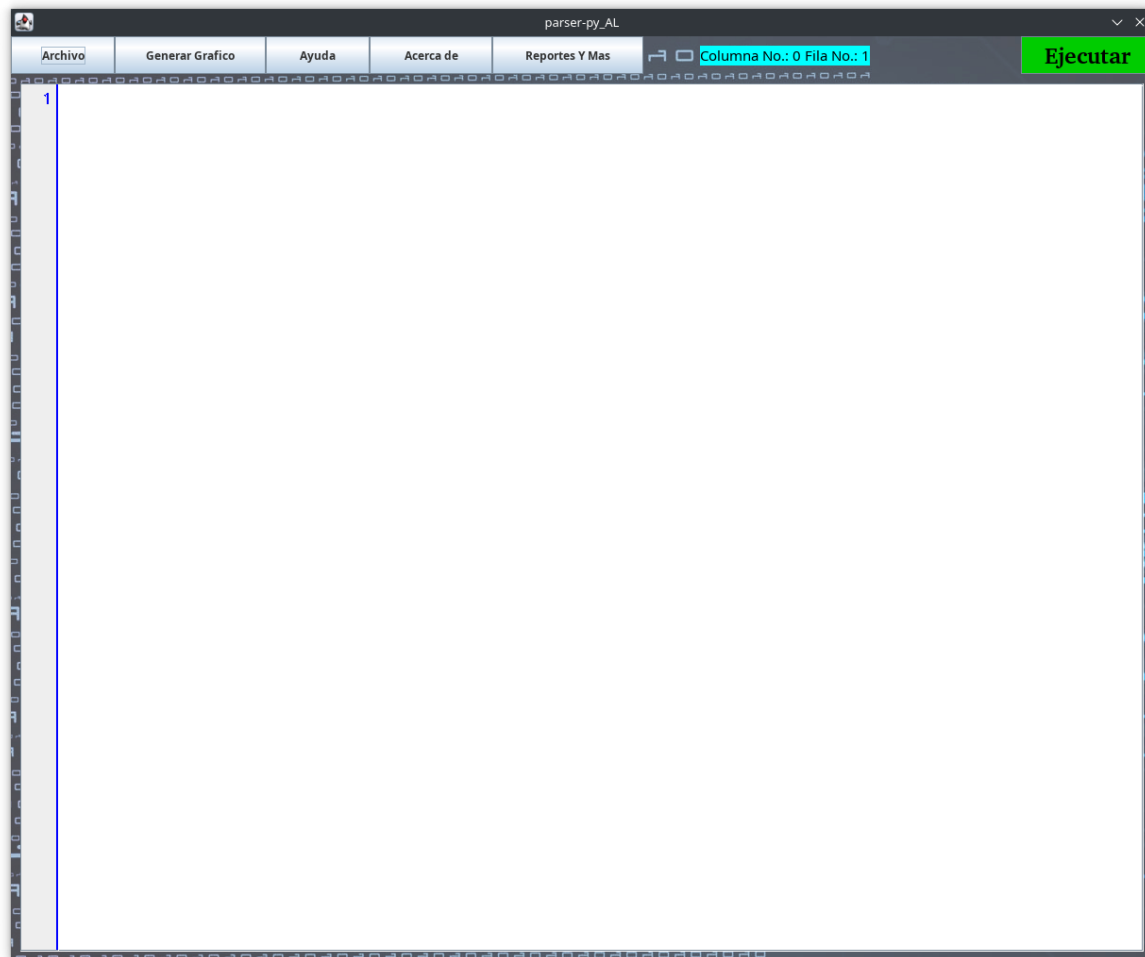


Tras esto se ejecutará el archivo del programa, debemos abrirlo por medio de código en la terminal, si en dado caso el sistema operativo no permite la ejecución. Para esto tendremos que hacer lo siguiente:



¿Qué veré tras ejecutarlo?

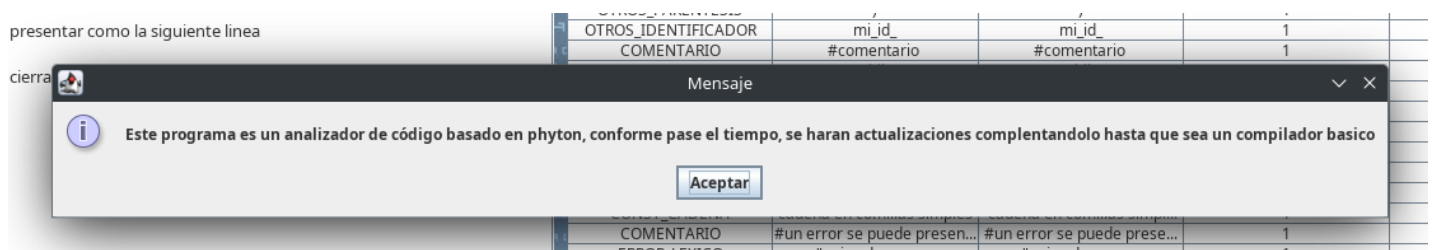
Se le presentará el menú principal de la siguiente manera:



Se tienen diversos botones, en los que se estará mostrando cierta información o cuadro de dialogo, las opciones son:

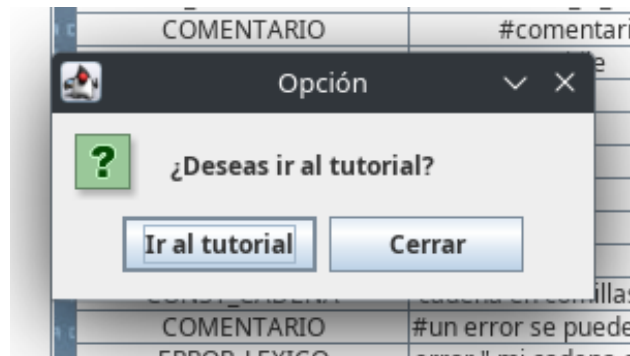
-Si escoges el botón “Acerca de”:

Se te mostrará el siguiente cuadro de dialogo:



-Si escoges el botón “Ayuda”:

Se te mostrará el siguiente cuadro de dialogo:



Si tus sistema operativo lo permite, se te redireccinará al enlace de YouTube en donde hay una breve explicación del programa :)

Si no, se te mostrará el siguiente cuadro de dialogo:

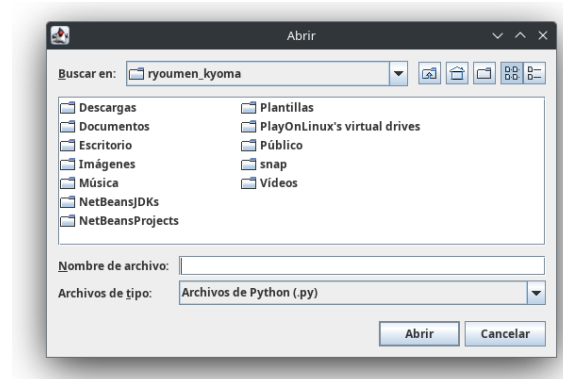


-Si escoges el botón “Generar Gráfico”

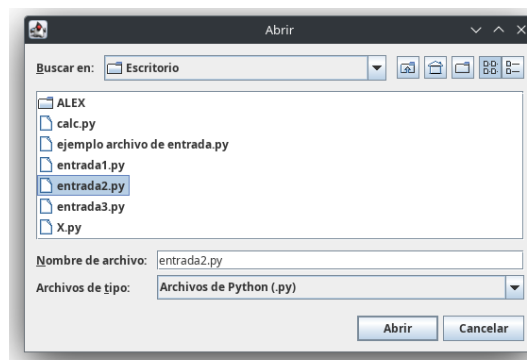
No se realizará nada ya que el botón aun está en fase de desarrollo (aun :()

-Si escoges el botón “Archivo”

Nos mostrará la siguiente ventana donde escogemos en que carpeta tenemos el archivo que deseamos abrir



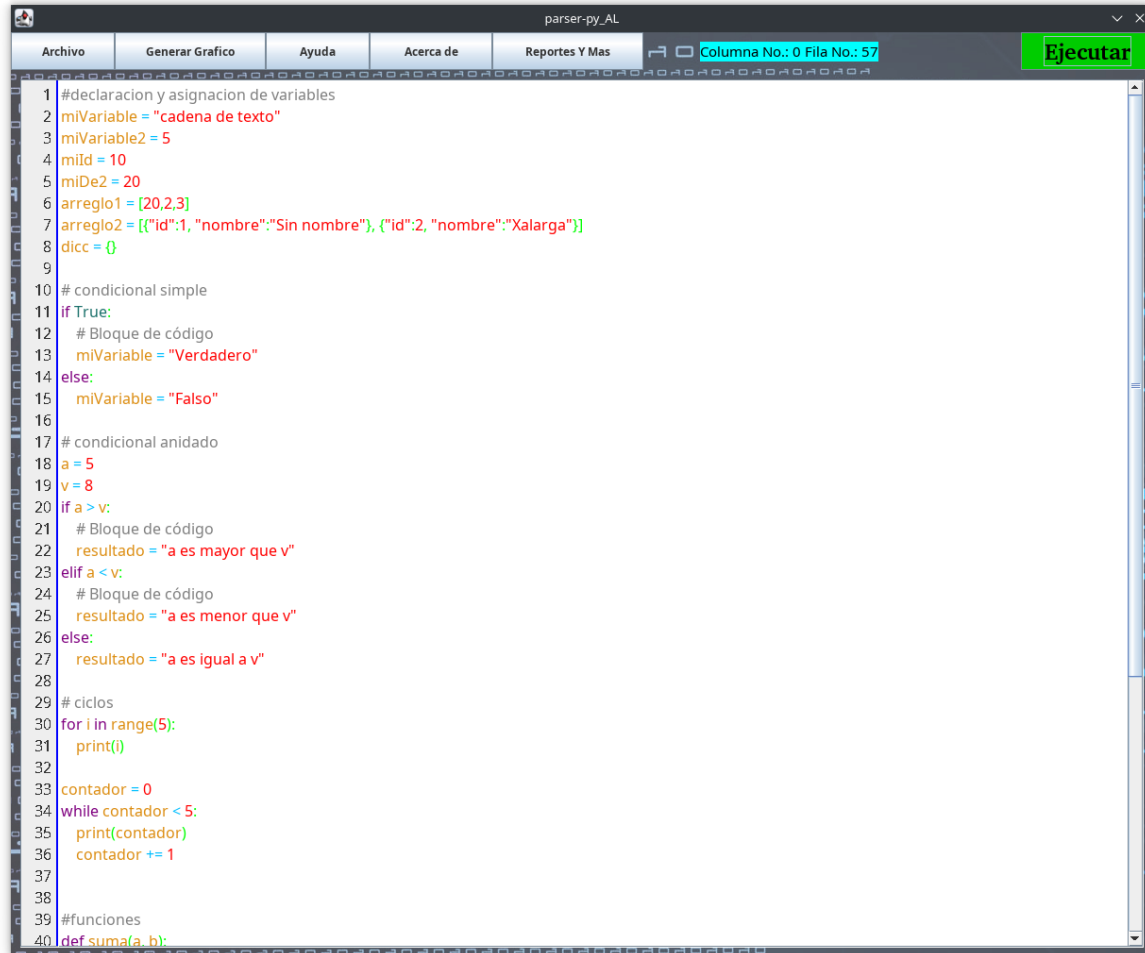
En nuestro caso, abriremos la carpeta de Escritorio, en tu caso es dependiendo la carpeta en la que tengas el archivo a abrir



Al abrirlo, nos imprimirá el contenido del archivo txt en el panel disponible

```
1 #declaracion y asignacion de variables
2 miVariable = "cadena de texto"
3 miVariable2 = 5
4 mild = 10
5 miDe2 = 20
6 arreglo1 = [20,2,3]
7 arreglo2 = [{"id":1, "nombre":"Sin nombre"}, {"id":2, "nombre":"Xalarga"}]
8 dicc = {}
9
10 # condicional simple
11 if True:
12     # Bloque de código
13     miVariable = "Verdadero"
14 else:
15     miVariable = "Falso"
16
17 # condicional anidado
18 a = 5
19 v = 8
20 if a > v:
21     # Bloque de código
22     resultado = "a es mayor que v"
23 elif a < v:
24     # Bloque de código
25     resultado = "a es menor que v"
26 else:
27     resultado = "a es igual a v"
28
29 # ciclos
30 for i in range(5):
31     print(i)
32
33 contador = 0
34 while contador < 5:
35     print(contador)
36     contador += 1
37
38 #funciones
39 def suma(a, b):
```

Al pulsar en ejecutar, nos pintará cada token dependiendo del tipo, como un IDE tipico



```
1 #declaracion y asignacion de variables
2 miVariable = "cadena de texto"
3 miVariable2 = 5
4 miId = 10
5 miDe2 = 20
6 arreglo1 = [20,2,3]
7 arreglo2 = [{"id":1, "nombre":"Sin nombre"}, {"id":2, "nombre":"Xalarga"}]
8 dicc = {}
9
10 # condicional simple
11 if True:
12     # Bloque de código
13     miVariable = "Verdadero"
14 else:
15     miVariable = "Falso"
16
17 # condicional anidado
18 a = 5
19 v = 8
20 if a > v:
21     # Bloque de código
22     resultado = "a es mayor que v"
23 elif a < v:
24     # Bloque de código
25     resultado = "a es menor que v"
26 else:
27     resultado = "a es igual a v"
28
29 # ciclos
30 for i in range(5):
31     print(i)
32
33 contador = 0
34 while contador < 5:
35     print(contador)
36     contador += 1
37
38
39 #funciones
40 def suma(a, b):
```

-Si escoges el botón “Reportes Y Mas”

Se te mostrará la siguiente pestaña



--Si escoges Tokens Identificados

Se te mostrará la siguiente ventana que presenta la tabla de los tokens encontrados, asi como mas informacion importante de el, en la parte superior tienes botones que sirven para filtrar por tipo los tokens identificados

Tokens Identificados						
Mostrar Todo	Identificador	Palabra Reservada	Aritmetico	Asignacion	Booleanas	Comentario
Comparacion	Constante	Logico	Otro			
TIPO	TOKEN	PATRON	LEXEMA	FILA	COLUMNA	
TypeComentario	COMENTARIO	#(\[a-z]*[0-9]*\')	#declaracion y asignacion de ...	1	39	
TypeIdentificador	IDENTIFICADOR	(\w _)(\w \d)*	miVariable	2	10	
TypeAsignacion	ASIGNACION	=	=	2	12	
TypeConstante	CADENA	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	"cadena de texto"	2	30	
TypeIdentificador	IDENTIFICADOR	(\w _)(\w \d)*	miVariable2	3	11	
TypeAsignacion	ASIGNACION	=	=	3	13	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	5	3	15	
TypeIdentificador	IDENTIFICADOR	(\w _)(\w \d)*	mid	4	4	
TypeAsignacion	ASIGNACION	=	=	4	6	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	10	4	9	
TypeIdentificador	IDENTIFICADOR	(\w _)(\w \d)*	miDe2	5	5	
TypeAsignacion	ASIGNACION	=	=	5	7	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	20	5	10	
TypeIdentificador	IDENTIFICADOR	(\w _)(\w \d)*	arreglo1	6	8	
TypeAsignacion	ASIGNACION	=	=	6	10	
TypeOtro	CORCHETES	[]	[6	12	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	20	6	14	
TypeOtro	COMA	,	,	6	15	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	2	6	16	
TypeOtro	COMA	,	,	6	17	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	3	6	18	
TypeOtro	CORCHETES	[]]	6	19	
TypeIdentificador	IDENTIFICADOR	(\w _)(\w \d)*	arreglo2	7	8	
TypeAsignacion	ASIGNACION	=	=	7	10	
TypeOtro	CORCHETES	[]	[7	12	
TypeOtro	LLAVES	{ }	{	7	13	
TypeConstante	CADENA	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	"id"	7	17	
TypeOtro	DOS_PUNTOS	:	:	7	18	
TypeConstante	ENTERO	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	1	7	19	
TypeOtro	COMA	,	,	7	20	
TypeConstante	CADENA	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	"nombre"	7	29	
TypeOtro	DOS_PUNTOS	:	:	7	30	
TypeConstante	CADENA	((\[a-z]*[0-9]*\') (\[A-Z]*[0-9]*))	"Sin nombre"	7	42	
TypeOtro	LLAVES	[]	}	7	43	
TypeOtro	COMA	,	,	7	44	

--Si escoges errores lexicos

Se te mostrará una tabla similar a la anterior, pero en esta van los errores lexicos, que en este caso no hay ninguno, por lo que sale vacia

Errores Lexicos				
TOKENS	PATRON	LEXEMA	FILA	COLUMNA

--Si escoges Tabla de Simbolos Globales

Se te muestra una tabla donde en este caso van todos los bloques de código registrados por el lenguaje en el código ingresado, también cuenta con botones en la parte superior para filtrarlos por tipos

Tabla de Símbolos Global, Cantidad de Funciones y No. de Llamadas						
Mostrar Todo	Operadores Ternarios	Declaración de Variables	Ciclo For	Ciclo While	Definición de Función	Comentarios
DESCRIPCION						
Comentario #1 en línea 1, columna 39: #declaracion y asignacion de variables						
Declaración de Variable #1: Identificador: miVariable Operador de Asignación: = Valor: "cadena de texto" en línea 2, columna 10						
Declaración de Variable #2: Identificador: miId Operador de Asignación: = Valor: 10 en línea 4, columna 4						
Declaración de Variable #3: Identificador: arreglo1 Operador de Asignación: = Valor: [en línea 6, columna 8						
Declaración de Variable #4: Identificador: dicc Operador de Asignación: = Valor: { en línea 8, columna 4						
Comentario #2 en línea 10, columna 20: # condicional simple						
Comentario #3 en línea 12, columna 22: # Bloque de código						
Declaración de Variable #5: Identificador: miVariable Operador de Asignación: = Valor: "Verdadero" en línea 13, columna 14						
Comentario #4 en línea 17: # condicional anidado						
Declaración de Variable #6: Identificador: miVariable Operador de Asignación: = Valor: "Falso" en línea 15, columna 14						
Declaración de Variable #7: Identificador: a Operador de Asignación: = Valor: 5 en línea 18, columna 1						
Declaración de Variable #8: Identificador: a Operador de Asignación: > Valor: v en línea 20, columna 4						
Comentario #5 en línea 21, columna 22: # Bloque de código						
Declaración de Variable #9: Identificador: resultado Operador de Asignación: = Valor: "a es mayor que v" en línea 22, columna 13						
Declaración de Variable #10: Identificador: a Operador de Asignación: < Valor: v en línea 23, columna 6						
Comentario #6 en línea 24, columna 22: # Bloque de código						
Declaración de Variable #11: Identificador: resultado Operador de Asignación: = Valor: "a es menor que v" en línea 25, columna 13						
Comentario #7 en línea 29: # ciclos						
Declaración de Variable #12: Identificador: resultado Operador de Asignación: = Valor: "a es igual a v" en línea 27, columna 13						
Ciclo For #1: Token 'for': for Variable: i Token 'in': in Token 'range': range Token ':': (en línea 30, columna 3						
Declaración de Variable #13: Identificador: print Operador de Asignación: (Valor: i en línea 31, columna 9						
Declaración de Variable #14: Identificador: contador Operador de Asignación: = Valor: 0 en línea 33, columna 8						
Declaración de Variable #15: Identificador: contador Operador de Asignación: < Valor: 5 en línea 34, columna 14						
Declaración de Variable #16: Identificador: print Operador de Asignación: (Valor: contador en línea 35, columna 9						
Comentario #8 en línea 39: #funciones						
Declaración de Variable #17: Identificador: contador Operador de Asignación: += Valor: 1 en línea 36, columna 12						
Definición de Función #1: Token 'def': def Nombre de la función: suma Token '(': (Parámetro: a Token ')': , Token ':': b en línea 40, columna 3						
Declaración de Variable #18: Identificador: a Operador de Asignación: + Valor: b en línea 41, columna 12						
Declaración de Variable #19: Identificador: suma Operador de Asignación: (Valor: 3 en línea 43, columna 16						
Definición de Función #2: Token 'def': def Nombre de la función: funcion_compleja Token '(': (Parámetro: a Token ')': , Token ':': b en línea 46, columna 3						
Declaración de Variable #20: Identificador: a Operador de Asignación: > Valor: b en línea 47, columna 8						
Ciclo For #2: Token 'for': for Variable: i Token 'in': in Token 'range': range Token ':': (en línea 48, columna 11						
Declaración de Variable #21: Identificador: a Operador de Asignación:) Valor: : en línea 48, columna 27						
Declaración de Variable #22: Identificador: i Operador de Asignación:) Valor: else en línea 49, columna 19						
Ciclo While #1: Token 'while': while Condición: a Token ':': < en línea 51, columna 13						

-Para salir, simplemente pulsamos el botón de la esquina superior derecha, así mismo para minimizar la pestaña.

