

Compilador Realizado en

Java 11.0.8

Probado en ubuntu 20.04 lts

Direccion github : <https://github.com/juanrios1307/AnalizadorLexicoGrafico>

Realizado por:

Juan Sebastian Henao Ramirez

Juan Esteban Rios Gallego

Universidad EIA
Lenguajes y compiladores

Primer paso:

Se corre la aplicación y se visualiza la interfaz de menú.



Segundo paso:

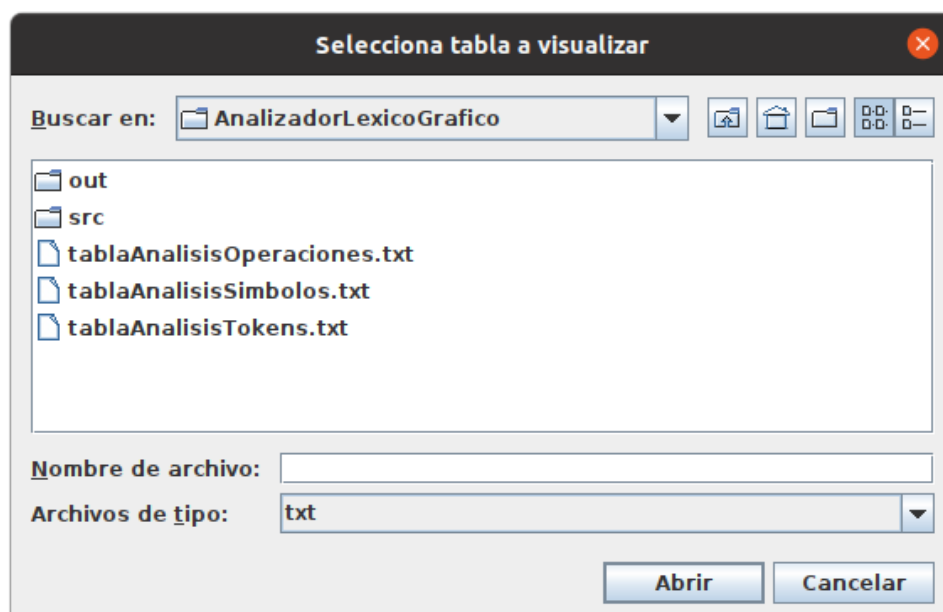
Se hace click al botón de cargar un archivo y se selecciona la ruta del documento a realizar el análisis.



Nota (Abrir Tablas Resultantes):

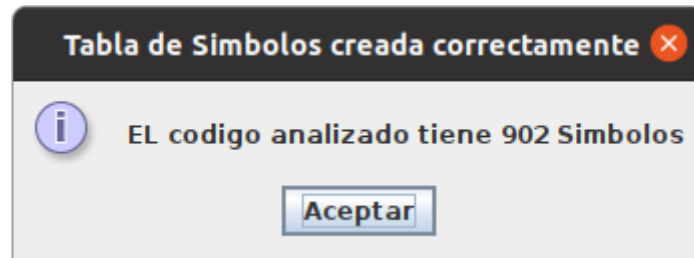
Para abrir las tablas resultantes se debe dar click al botón llamado “Abrir una tabla” después de haber realizado los pasos siguientes. Se abrirá un seleccionador de archivo, donde usted navegará hasta la carpeta principal del analizador y seleccionará la tabla a visualizar.

Se eligió esta alternativa ya que, las tablas se están guardando en una dirección relativa, que dependiendo del sistema operativo se puede o no abrir, lo cual genera problemas al momento de abrir el archivo, deteniendo la ejecución del programa, sin embargo, apoyándonos de las herramientas que ofrece el lenguaje escogido (Java) podemos tomar el archivo con la ayuda del usuario.



Tercer paso:

Ahora se hace click en el botón de “Identificar elementos Tabla de símbolos” y esto nos generará un archivo donde estará la tabla de símbolos encontrados en el documento analizado además de una alerta que muestra la cantidad de símbolos encontrados. Para visualizar esta tabla, abres la carpeta del analizador y allí encontrarás un archivo llamado “tablaAnálisisSimbolos.txt”.

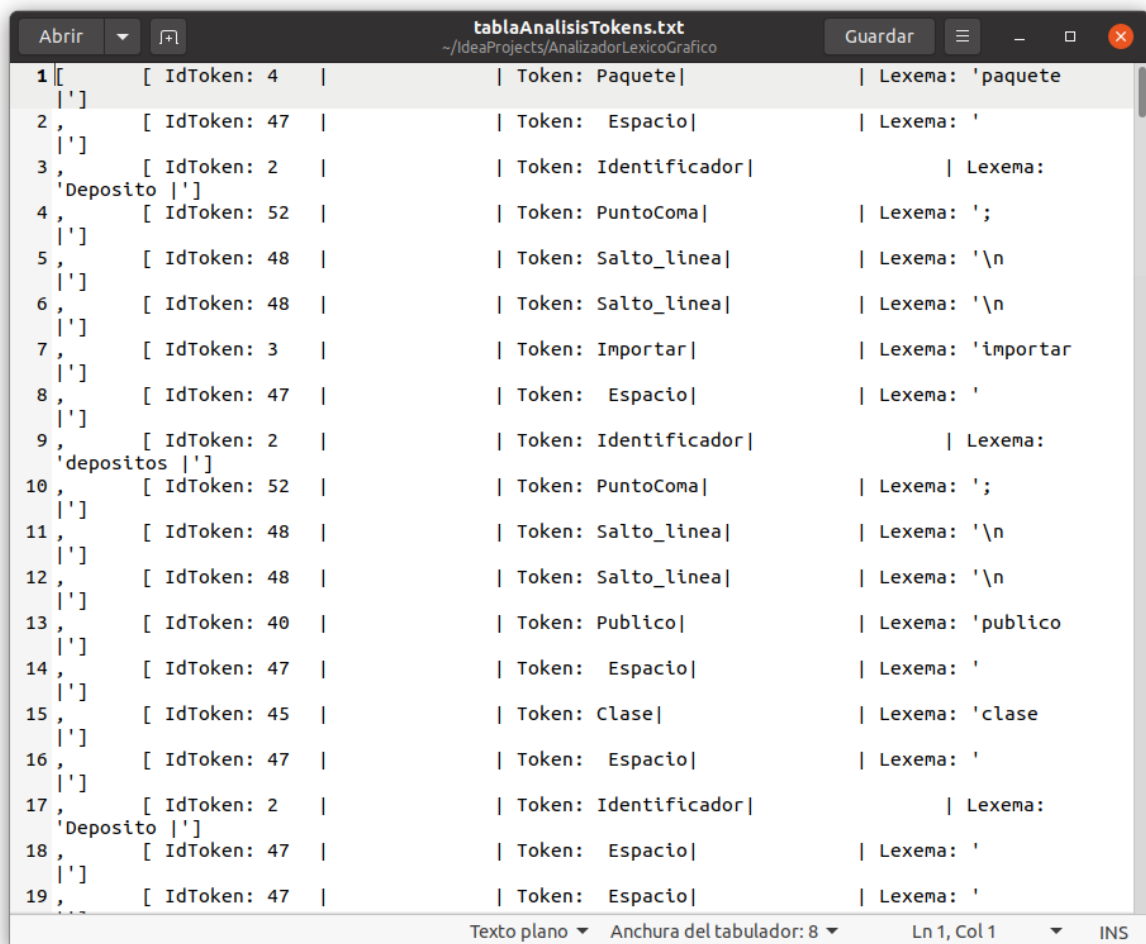
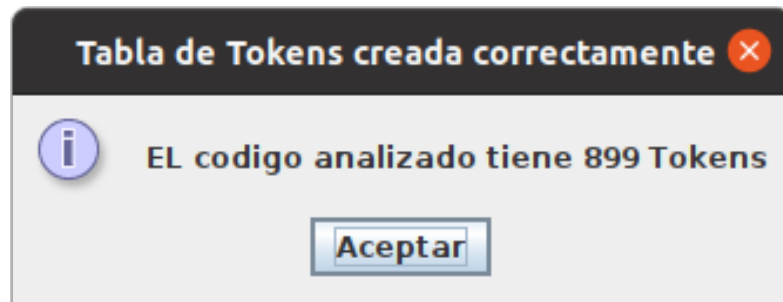
Una captura de pantalla de un editor de texto que muestra el contenido del archivo "tablaAnálisisSimbolos.txt". El archivo contiene una lista de tokens encontrados en un código fuente, con su ubicación y tipo. El editor muestra la barra de título, botones de "Abrir" y "Guardar", y una barra de estado que indica "Texto plano", "Anchura del tabulador: 8", "Ln 1, Col 1" y "INS".

Nº	Simbolo	Ubicacion	Tipos
1	[Simbolo:'paquete'	Ubicacion:(1, 1)	Tipos: palabra
2	reservada]		
3	[Simbolo:' '	Ubicacion:(1, 8)	Tipos: Separador]
4	[Simbolo:'Deposito'	Ubicacion:(1, 9)	Tipos:
5	identificador]		
6	[Simbolo:';'	Ubicacion:(1, 17)	Tipos: Separador
7	Sentencias]		
8	[Simbolo:'\n'	Ubicacion:(1, 18)	Tipos:
9	Separador]		
10	[Simbolo:'\n'	Ubicacion:(2, 1)	Tipos:
11	Separador]		
12	[Simbolo:'importar'	Ubicacion:(3, 1)	Tipos: palabra
13	reservada]		
14	[Simbolo:' '	Ubicacion:(3, 9)	Tipos: Separador]
15	[Simbolo:'depositos'	Ubicacion:(3, 10)	Tipos:
16	identificador]		
17	[Simbolo:';'	Ubicacion:(3, 19)	Tipos: Separador
18	Sentencias]		
19	[Simbolo:'\n'	Ubicacion:(3, 20)	Tipos:
20	Separador]		
21	[Simbolo:'\n'	Ubicacion:(4, 1)	Tipos:
22	Separador]		
23	[Simbolo:'publico'	Ubicacion:(5, 1)	Tipos: palabra
24	reservada]		
25	[Simbolo:' '	Ubicacion:(5, 8)	Tipos: Separador]
26	[Simbolo:'clase'	Ubicacion:(5, 9)	Tipos: palabra
27	reservada]		
28	[Simbolo:' '	Ubicacion:(5, 14)	Tipos: Separador]
29	[Simbolo:'Deposito'	Ubicacion:(5, 15)	Tipos:
30	identificador]		
31	[Simbolo:' '	Ubicacion:(5, 23)	Tipos: Separador]
32	[Simbolo:' '	Ubicacion:(5, 24)	Tipos: Separador]
33	[Simbolo:'{'	Ubicacion:(5, 25)	Tipos: Operador Agrupacion ;
34	Separador]		
35	[Simbolo:'\n'	Ubicacion:(5, 26)	Tipos:
36	Separador]		
37	[Simbolo:'\n'	Ubicacion:(6, 1)	Tipos:

Cuarto paso:

Se hace click en el botón de “Identificar elementos Tabla de Tokens” y esto nos generará un archivo donde estará la tabla de los tokens encontrados en el documento analizado además

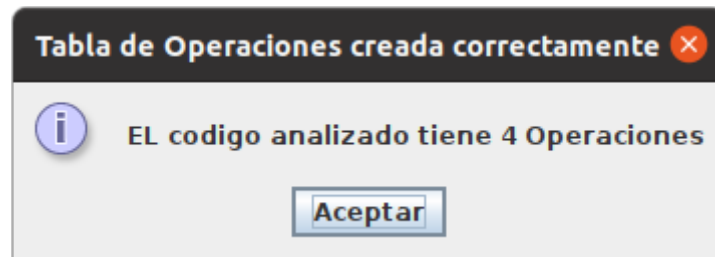
de una alerta que muestra la cantidad de tokens encontrados. Para visualizar esta tabla, abres la carpeta del analizador y allí encontrarás un archivo llamado "tablaAnálisisTokens.txt".

Una captura de pantalla de un editor de texto que muestra el archivo "tablaAnálisisTokens.txt". La interfaz incluye una barra superior con "Abrir", "Guardar", y un menú. El contenido de la tabla se muestra en un formato de texto plano con tabuladores. El status bar en la parte inferior indica "Texto plano", "Anchura del tabulador: 8", "Ln 1, Col 1" y "INS".

1	[[IdToken: 4				Token: Paquete		Lexema: 'paquete
2	,		IdToken: 47				Token: Espacio		Lexema: '
3	,		IdToken: 2				Token: Identificador		Lexema:
4	,		IdToken: 52				Token: PuntoComa		Lexema: ';'
5	,		IdToken: 48				Token: Salto_linea		Lexema: '\n'
6	,		IdToken: 48				Token: Salto_linea		Lexema: '\n'
7	,		IdToken: 3				Token: Importar		Lexema: 'importar'
8	,		IdToken: 47				Token: Espacio		Lexema: ' '
9	,		IdToken: 2				Token: Identificador		Lexema:
10	,		IdToken: 52				Token: PuntoComa		Lexema: ';'
11	,		IdToken: 48				Token: Salto_linea		Lexema: '\n'
12	,		IdToken: 48				Token: Salto_linea		Lexema: '\n'
13	,		IdToken: 40				Token: Publico		Lexema: 'publico'
14	,		IdToken: 47				Token: Espacio		Lexema: ' '
15	,		IdToken: 45				Token: Clase		Lexema: 'clase'
16	,		IdToken: 47				Token: Espacio		Lexema: ' '
17	,		IdToken: 2				Token: Identificador		Lexema:
18	,		IdToken: 47				Token: Espacio		Lexema: ' '
19	,		IdToken: 47				Token: Espacio		Lexema: ' '

Quinto paso:

Se hace click en el botón de “Identificar elementos Expresiones Aritméticas” y esto nos generará un archivo donde estará la tabla de las operaciones aritméticas encontradas en el documento analizado además de una alerta que muestra la cantidad encontrada de estas. Para visualizar esta tabla, abres la carpeta del analizador y allí encontrarás un archivo llamado “tablaAnálisisOperaciones.txt”.



```

1 OperacionAritmetica{
2   [ Simbolo:'(' | Ubicacion:(13, 15) | Tipos: Operador Agrupacion ;
  Separador|]
3   , [ Simbolo:'12' | Ubicacion:(13, 16) | Tipos:
  identificador|]
4   , [ Simbolo:'*' | Ubicacion:(13, 18) | Tipos: Operador Aritmetico ;
  Separador|]
5   , [ Simbolo:'2' | Ubicacion:(13, 19) | Tipos:
  identificador|]
6   , [ Simbolo:')' | Ubicacion:(13, 20) | Tipos: Operador Agrupacion ;
  Separador|]
7   , [ Simbolo:'+' | Ubicacion:(13, 21) | Tipos: Operador Aritmetico ;
  Separador|]
8   , [ Simbolo:'85' | Ubicacion:(13, 22) | Tipos:
  identificador|]
9   , [ Simbolo:'*' | Ubicacion:(13, 24) | Tipos: Operador Aritmetico ;
  Separador|]
10  , [ Simbolo:'diametro' | Ubicacion:(13, 25) | Tipos:
  identificador|]
11  , [ Simbolo:'-' | Ubicacion:(13, 33) | Tipos: Operador Aritmetico ;
  Separador|]
12  , [ Simbolo:'agua' | Ubicacion:(13, 34) | Tipos:
  identificador|]
13  , [ Simbolo:'+' | Ubicacion:(13, 38) | Tipos: Operador Aritmetico ;
  Separador|]
14  , [ Simbolo:'(' | Ubicacion:(13, 39) | Tipos: Operador Agrupacion ;
  Separador|]
15  , [ Simbolo:'volumen' | Ubicacion:(13, 40) | Tipos:
  identificador|]
16  , [ Simbolo:')' | Ubicacion:(13, 47) | Tipos: Operador Agrupacion ;
  Separador|]
17 }
18 Expresión: (12*2)+85*diametro-agua+(volumen)}
19 , OperacionAritmetica{
20   [ Simbolo:'(' | Ubicacion:(15, 16) | Tipos: Operador Agrupacion ;
  Separador|]
21   , [ Simbolo:'(' | Ubicacion:(15, 17) | Tipos: Operador Agrupacion ;

```

Texto plano ▾ Anchura del tabulador: 8 ▾ Ln 1, Col 1 ▾ INS

Sexto paso:

En este paso se le dará click al botón llamado “Verificar 1 Expresión Aritmética”, esto desplegará un menú con todas las operaciones encontradas con un identificador para cada

una ubicado antes de la misma, para evaluar la validez en sintaxis de una de estas expresiones se digita en el input el número correspondiente a la misma que se desea evaluar y despues se da click al botón aceptar, esto desplegará una alerta mostrando si la expresión es válida o contiene un error de sintaxis.

Entrada

?

Operaciones:
[0 : (12*2)+85*diametro-agua+(volumen)
, 1 : ((2+2)-(3+4))*5-(diametro)
, 2 : pi*diametro/2*(diametro/2)*altura
, 3 : (55+2)+hola-diametro
]

AceptarCancelar

Expresion

X

La expresion analizada contiene errores de sintaxis

Aceptar

Expresion

i

La expresion analizada NO contiene errores de sintaxis

Aceptar