

1. Introducción

Analizador lexico que crea imagenes a partir de tokens generados de un lenguaje inspirado en Visual Basic (Sintaxis en la ultima seccion).

2. Descripción General

2.1. Funcionalidad

El analizador léxico realiza las siguientes tareas:

- **Análisis Léxico:** Identifica tokens en un texto de entrada basándose en expresiones regulares.
- **Generación de Autómatas:** Crea autómatas a partir de tokens.
- **Visualización:** Muestra una cuadrícula con colores correspondientes a cada token.

2.2. Requisitos del Sistema

- **Graphviz:** Herramienta externa para la generación de gráficos

3. Instalación y Configuración

3.1. Instalación de Dependencias

3.1.1. Java

Se debe tener instalado Java JDK preferiblemente la version 17.

3.1.2. Graphviz

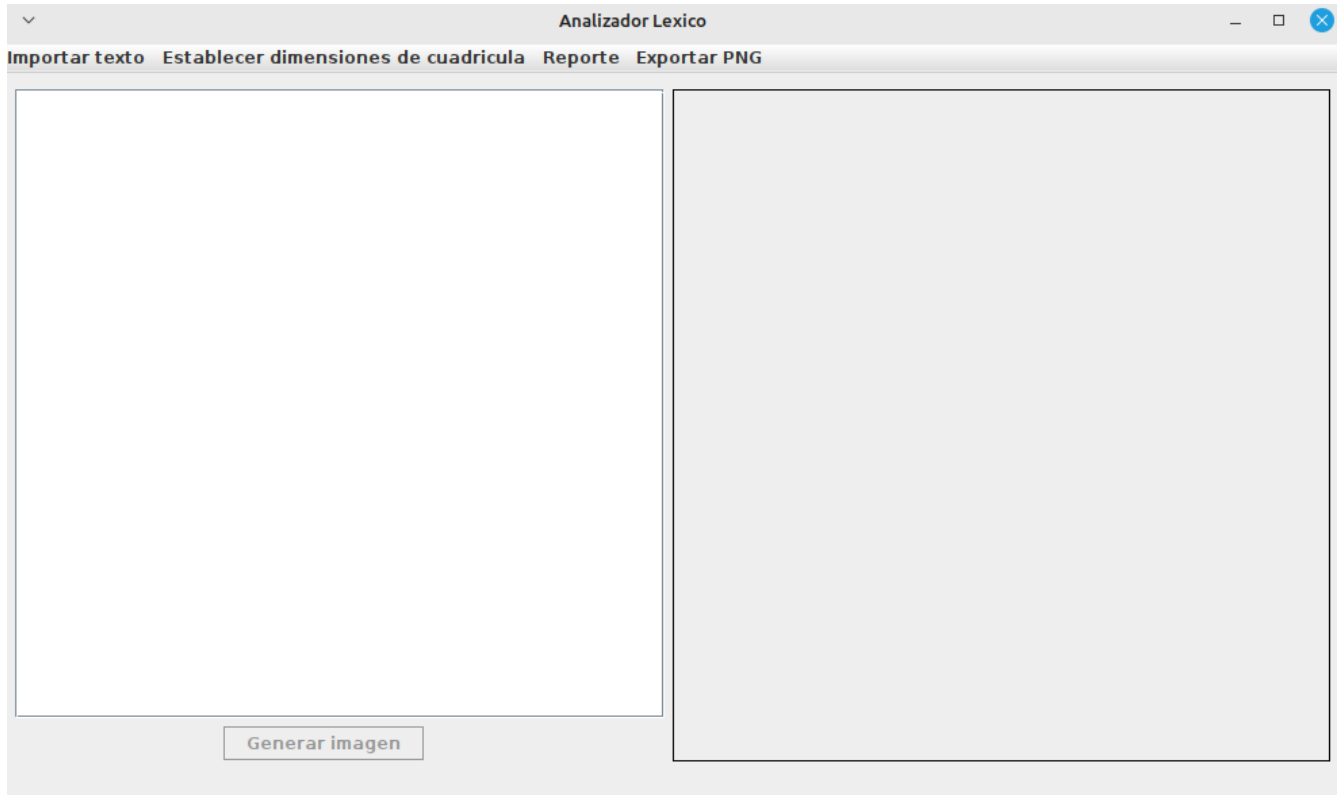
Instalar Graphviz ('sudo apt install graphviz' desde SO derivados de Ubuntu).

4. Uso del Programa

4.1. Ejecutar el Programa

1. Compila el programa Java utilizando tu IDE o desde la línea de comandos (java -jar ejecutable.jar).

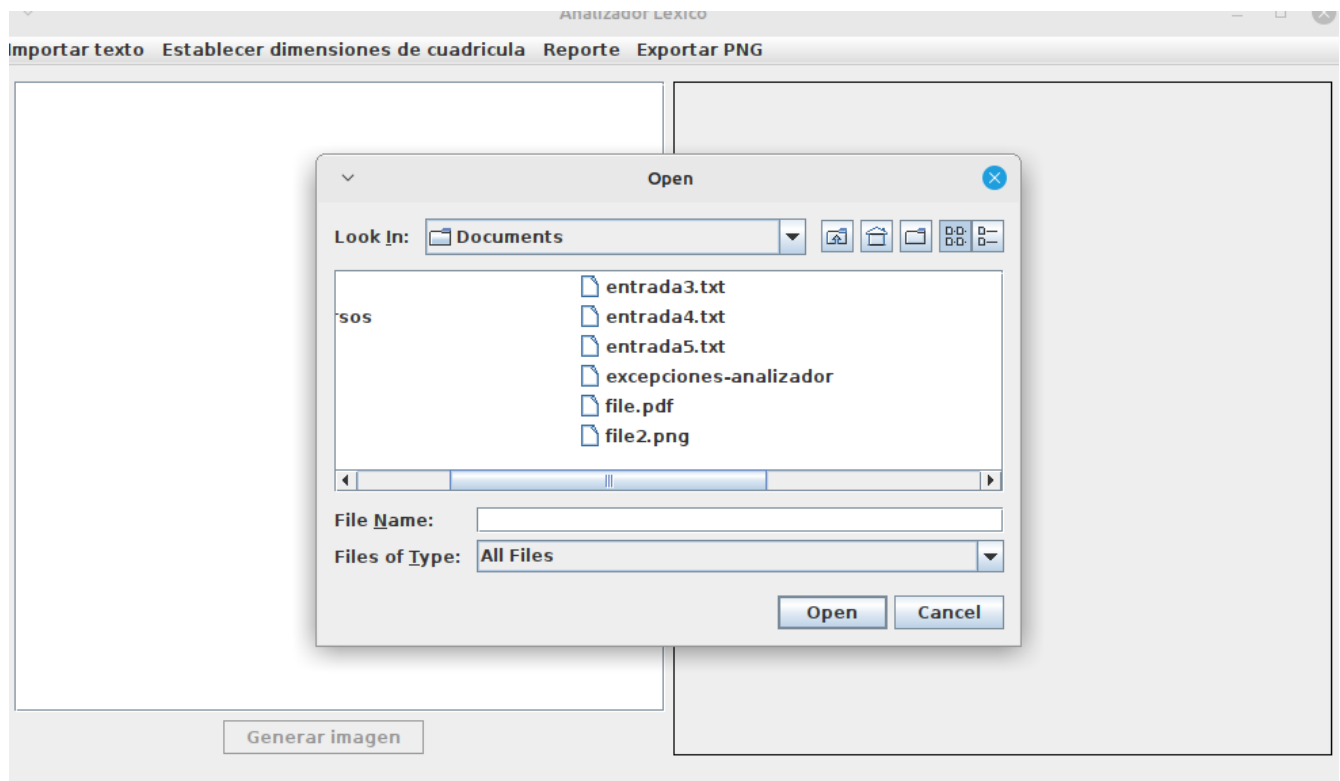
4.2. Pantalla de Inicio



4.3. Editor de Texto

El usuario puede escribir directamente en el editor de texto o importar un archivo.

Se debe tomar en cuenta que los tokens especiales sobrescribieran cualquier otro token que se encuentre en la fila y columna indicada en los parametros.



4.4. Cuadrícula

El usuario debe especificar las dimensiones de la cuadrícula antes de generar la imagen.

La imagen generada se podrá exportar en formato PNG, esta se guardará en la carpeta del programa.

Analizador Lexico

Importar texto Establecer dimensiones de cuadrícula Reporte Exportar PNG

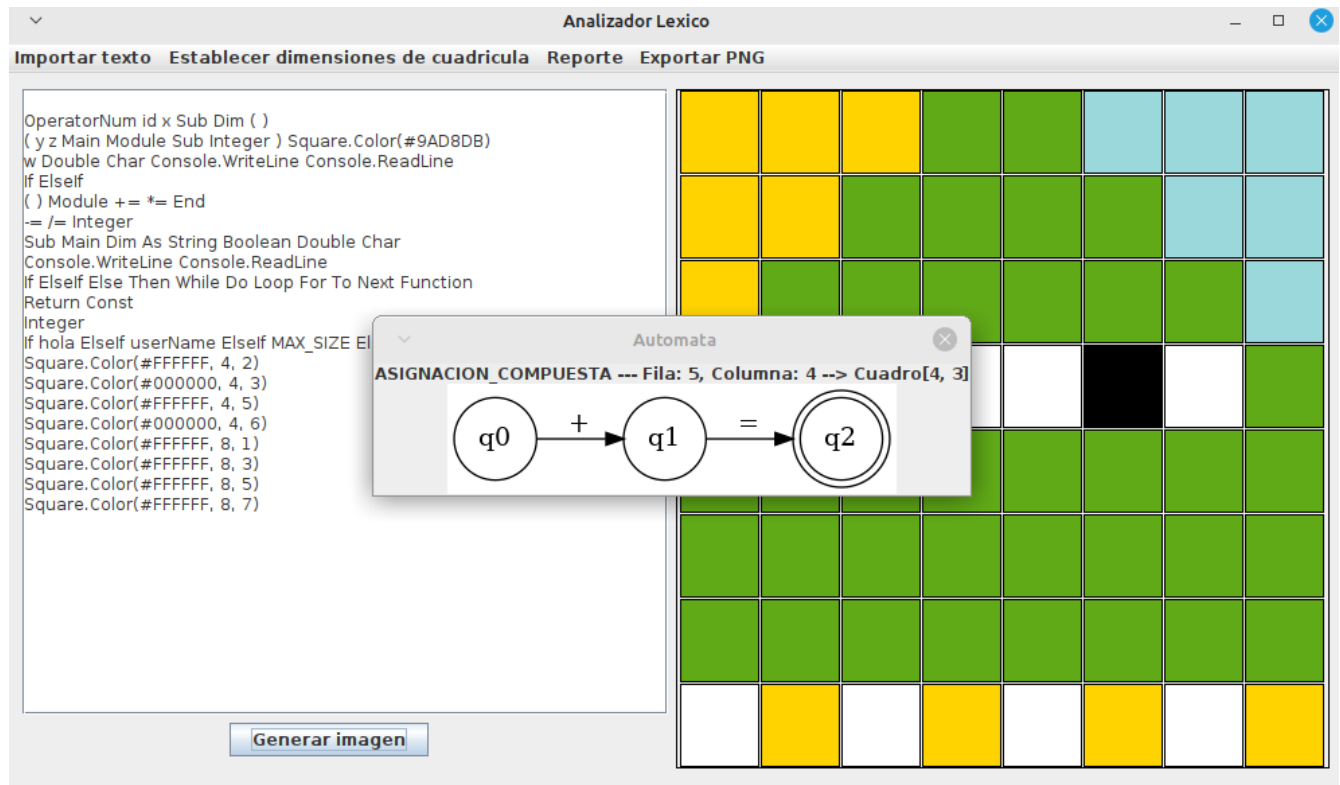
```
OperatorNum id x Sub Dim ( )
( y z Main Module Sub Integer ) Square.Color(#9AD8DB)
w Double Char Console.WriteLine Console.ReadLine
If ElseIf
( ) Module += *= End
-= /= Integer
Sub Main Dim As String Boolean Double Char
Console.WriteLine Console.ReadLine
If ElseIf Else Then While Do Loop For To Next Function
Return Const
Integer
If hola ElseIf userName ElseIf MAX_SIZE Else MaxUsers
Square.Color(#FFFFFF, 4, 2)
Square.Color(#000000, 4, 3)
Square.Color(#FFFFFF, 4, 5)
Square.Color(#000000, 4, 6)
Square.Color(#FFFFFF, 8, 1)
Square.Color(#FFFFFF, 8, 3)
Square.Color(#FFFFFF, 8, 5)
Square.Color(#FFFFFF, 8, 7)
```

Generar imagen

Reporte						
Token	Lexema	Linea	Columna	Fila Cuadrícula	Columna Cuadrícula	Color
IDENTIFICADOR	OperatorNum	1	1	1	1	#FFD300
IDENTIFICADOR	id	1	2	1	2	#FFD300
IDENTIFICADOR	x	1	3	1	3	#FFD300
PALABRA_RESERVADA	Sub	1	4	1	4	#60A917
PALABRA_RESERVADA	Dim	1	5	1	5	#60A917
PARENTESIS	(1	6	1	6	#9AD8DB
PARENTESIS)	1	7	1	7	#9AD8DB
PARENTESIS	(2	1	1	8	#9AD8DB
IDENTIFICADOR	y	2	2	2	1	#FFD300
IDENTIFICADOR	z	2	3	2	2	#FFD300
PALABRA_RESERVADA	Main	2	4	2	3	#60A917
PALABRA_RESERVADA	Module	2	5	2	4	#60A917
PALABRA_RESERVADA	Sub	2	6	2	5	#60A917
PALABRA_RESERVADA	Integer	2	7	2	6	#60A917
PARENTESIS)	2	8	2	7	#9AD8DB
SQUARE2	Square.Color(#9AD...	2	9	2	8	#9AD8DB
IDENTIFICADOR	w	3	1	3	1	#FFD300
PALABRA_RESERVADA	Double	3	2	3	2	#60A917
PALABRA_RESERVADA	Char	3	3	3	3	#60A917
PALABRA_RESERVADA	Console.WriteLine	3	4	3	4	#60A917
PALABRA_RESERVADA	Console.ReadLine	3	5	3	5	#60A917
PALABRA_RESERVADA	If	4	1	3	6	#60A917
PALABRA_RESERVADA	Elseif	4	2	3	7	#60A917
PARENTESIS	(5	1	3	8	#9AD8DB
PARENTESIS)	5	2	4	1	#9AD8DB
PALABRA_RESERVADA	Module	5	3	4	2	#60A917
ASIGNACION_COMPU...	+=	5	4	4	3	#FFFFFF
ASIGNACION_COMPU...	*=	5	5	4	4	#FFFFFF
PALABRA_RESERVADA	End	5	6	4	5	#60A917
ASIGNACION_COMPU...	-=	6	1	4	6	#FFFFFF
ASIGNACION_COMPU...	/=	6	2	4	7	#FFFFFF
PALABRA_RESERVADA	Integer	6	3	4	8	#60A917
PALABRA_RESERVADA	Sub	7	1	5	1	#60A917
PALABRA_RESERVADA	Main	7	2	5	2	#60A917

4.6. Automatas

Se generara una descripcion del token correspondiente al hacer click en un cuadro de la cuadrícula.



5. Sintaxis del Lenguaje

Identificadores

- Los identificadores deben comenzar con una letra (A-Z o a-z).
- Pueden contener letras, dígitos (0-9) y el carácter de subrayado (_).
- Los identificadores no pueden comenzar con un dígito.







Color del Identificador:









Operadores

Los operadores son símbolos que son utilizados para realizar algún tipo de operación aritmética, Racional, Lógica.



Aritméticos

Nombre	Símbolo	Cuadro
Suma	+	 #FF33FF
Resta	-	 #C19A6B
Exponente	^	 #FCD0B4
División	/	 #B4D941
Módulo	Mod	 #D9AB41
Multiplicación	*	 #D80073


Relacionales o Comparación

Nombre	Símbolo	Cuadro
Igual	==	 #6A00FF
Diferente	<>	 #3F2212
Mayor que	>	 #D9D441
Menor que	<	 #D94A41
Mayor o Igual que	>=	 #E3C800
Menor o Igual que	<=	 #F0A30A

Asignación

Nombre	Símbolo	Cuadro
Asignación Simple	=	 #41D9D4
Asignación Compuesta	+=, -=, *=, /=	 #FFFFFF

Palabras Reservadas

Nombre	Símbolo	Cuadro
Palabra Reservada	Module End Sub Main Dim As Integer String Boolean Double Char Console.WriteLine Console.ReadLine If Elseif Else Then While Do Loop For To Next Function Return Const	 #60A917

Tipos de Datos

Nombre	Símbolo	Cuadro
Entero	Cualquier número entero: 0, 1, 2, 3,...n	 #1BA1E2
Decimal	2.5, 32.02, 0.001	 #FFFF88
Cadena	"cadena", "Nombre", "123"	 #E51400
Booleano	True, False	 #FA6800
Carácter	'A', '2', '\$'	 #0050EF

Square.Color

Este será un token especial el cual tendrá que realizar una función especial a diferencia de los demás tokens.

La estructura es la siguiente:




Este token pintará el cuadro según el código de color asignado en el primer parámetro, este tipo de color es [hexadecimal](#). Solo se usará la notación hexadecimal.

La Fila y Columna corresponde a la posición o cuadro que pintará del lienzo, a diferencia de los demás tokens a este se le puede especificar qué cuadro es el que se pintará.




También se podrá aceptar el token especial simple, donde tendrá la misma acción que el anterior, con la diferencia que pintará el cuadro según sea su secuencia dentro del código fuente, y el color que pintará será el especificado en el parámetro, el cual es el siguiente



Square.Color(#4fbad4)

Comentario

Nombre	Símbolo	Cuadro
Comentario	' este es un comentario	 #B3B3B3

Signos y Símbolos

Nombre	Símbolo	Cuadro
Paréntesis	()	 #9AD8DB
Llaves	{ }	 #DBD29A
Corchetes	[]	 #DBA49A

Coma	,	 #B79ADB
Punto	.	 #9ADBA6