

PROYECTO 2



“Analizador Léxico”

LUIS EDUARDO GUILLÉN RUIZ

UNIVERSIDAD SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE “CUNOC”
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
INGENIERÍA EN CIENCIAS Y SISTEMAS
SEGUNDO SEMESTRE
2,019
Ing. OLIVER ERNESTO SIERRA

“ID Y ENSEÑAD A TODOS”

Contenido del Manual

Contenido del Manual	2
Presentación	3
Objetivos del Programa	3
Generales	3
Específicos	3
Instrucciones de Programa	4
Requerimientos Técnicos	4
Herramientas de Construcción	4
Diseño del Proyecto	5
Como Acceder al Programa	5
Home	5
Analizador	6
Gramática	6
Gramática LL1	8
Primeros	8
Segundos	8
Tabla de Análisis Sintáctico I	9
Tabla de Análisis Sintáctico II	10
Conjunto de Estados del Autómata:	11
Estado Inicial	11
Alfabeto Σ	11
Estados de Aceptación F	11
Función de Transición δ (delta)	11
Tabla de Transición	12
Estado Inicial S0	12

“Manual Técnico, Analizador Léxico”

Presentación

El presente documento es el manual Técnico del analizador léxico implementando la metodología de creación e identificación de gramáticas por medio de autómatas generados previamente en la cual se le dará acceso al cliente de subir archivos para un análisis en la parte lógica simulando un compilador de lenguaje la cual será la encargada de manejar todos los datos que el cliente desee analizar.

La aplicación está diseñada en NetBeans por medio de la sintaxis Java y se incluye un archivo ejecutable para poder dar la respectiva visualización por medio de cualquier interprete que pueda ejecutar JDK, así mismo se incluye el funcionamiento básico del mismo y una guía para su primera ejecución y posteriormente el uso del mismo.

Objetivos del Programa

Generales

- Familiarizar al estudiante con el lenguaje Java.
- Conocer el proceso de análisis léxico realizado por un compilador.
- Aplicar conceptos de creación de Autómatas.
- Aplicar conceptos de aceptación de Gramática.
- Elaborar la lógica para la solución del problema planteado.

Específicos

- Construcción de algoritmos para los requerimientos de la actividad.
- Ampliar el conocimiento de Programación en Java.
- Desarrollar una aplicación utilizando la metodología de un compilador
- Desarrollar toda la lógica del analizador léxico.
- Desarrollar diagramas de Moore.
- Desarrolló de manual técnico.
- Usar de interfaz gráfica en java, utilizando frames, buttons, entre otros.

Instrucciones de Programa

1. Instalar JDK.
2. Ejecutar JAR (ejecutable).
3. Ejecutar primeras funciones.
4. Buscar Archivo de Texto a Analizar.
5. Analizar
6. Verificación de Gramática
7. Solicitar Tokens
8. Refrescar
9. Salir

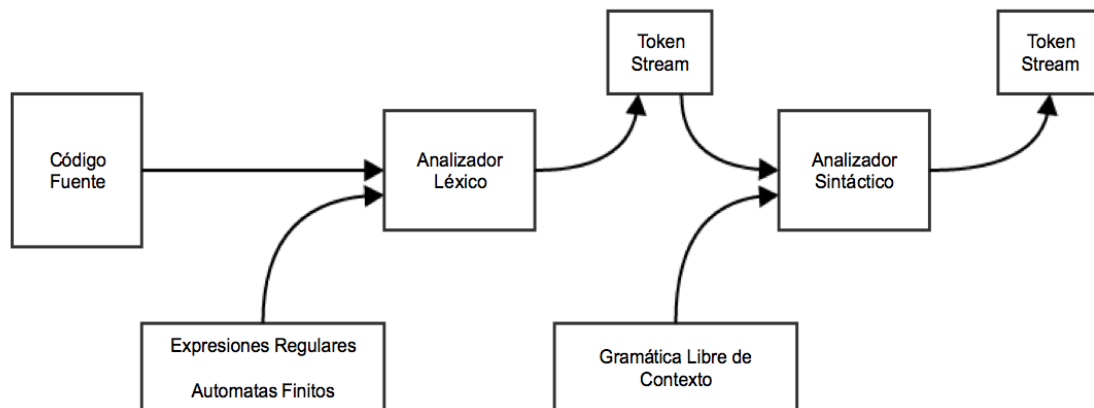
Requerimientos Técnicos

- SO que soporte Java (Ejemp.: Linux, Windows, IOs, Android).
- Java 1.8.0_201 o compatibles.
- NetBeans IDEA (creación: v. 8.2) o cualquier editor de lenguaje JAVA.
- .Jar ejecutable o proyecto completo.
- Entender lenguaje Java (para lógica del programa).
- Experiencia con aplicaciones Java (funcionamiento).

Herramientas de Construcción

- NetBeans IDEA
- Package de JAVA
- Clases de JAVA
- Interfaz de JAVA
- PC funcional
- Conocimientos de JavaScript
- Control de Versiones (GitHub)

Diseño del Proyecto



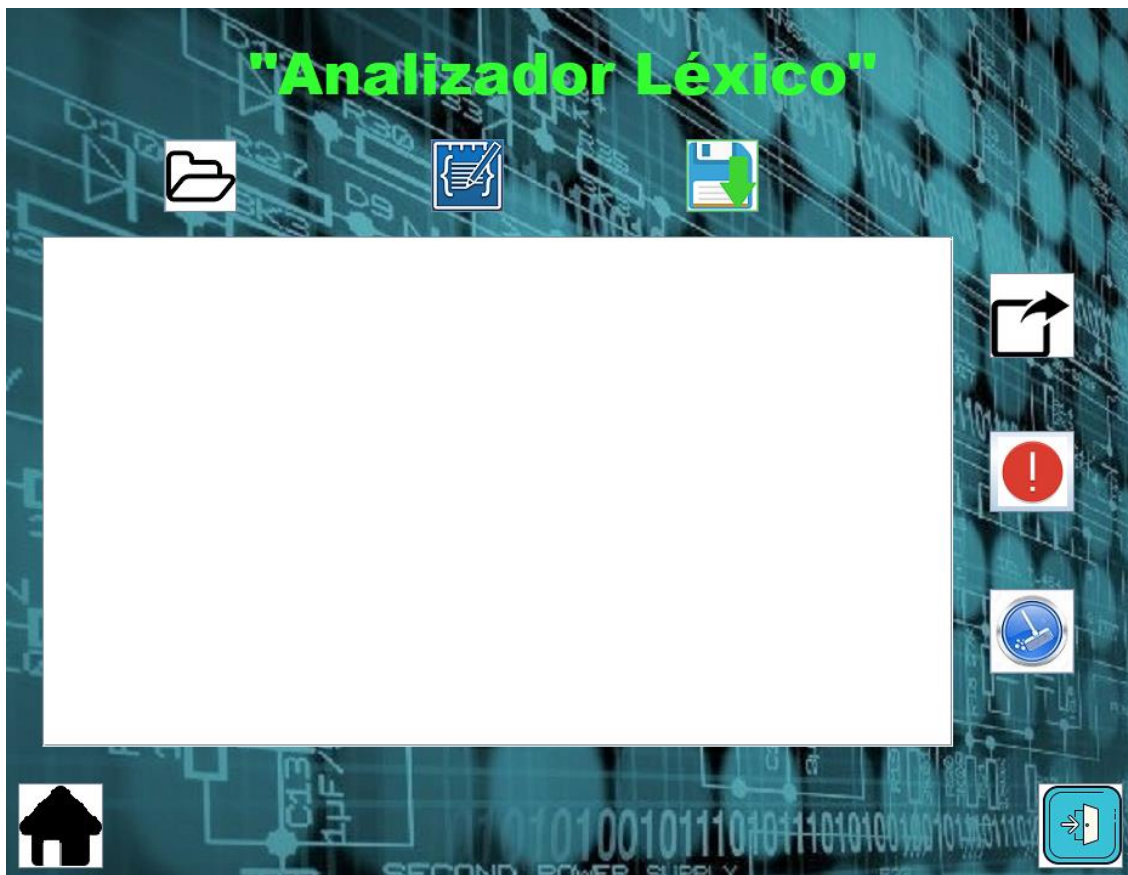
Como Acceder al Programa

Se debe tener el repositorio Online del Analizador descargar por medio de <https://github.com/LGuilln/201744792> Luis Guillen P2. Luego clonar en repositorio local, instalar requerimientos de java para ejecutar el Jar. Analizar archivos de texto.

Home



Analizador



Gramática Libre de Contexto

G = (N, T, P, S);

N = {C, V, D, T, L},

T = {ID, Num, Digito, Operador},

S = {E}

P:

E --> funcion principal { **C** }
funcion ID (**PARAMETRO**) { **C** }

C --> **V** ;
V = **T L** ;
imprimir ();
while() { **C** }

```

if ( ) { C }
for ( V = NUM ; ID < NUM ; ID++ ) { C }
ID ( ID ) ;

```

V --> variable **D** ID

D --> entero
decimal
booleano
cadena
caracter

T --> **NUM**
NUM . NUM
true
false
"cadena"

L --> ID **OPERADOR** ID
e

PARAMETRO --> **D** ID
e

NUM --> **DIGITO DIGITO**
e

DIGITO --> 0|1|2|3|4|5|6|7|8|9
e

OPERADOR --> +|-|*|/|%|=|==|<|>|>|=|<=
e

Comentario --> */ cadena/*

Gramática LL1

Primeros

No terminal	Primeros
E	funcion principal, ID
C	Variable, imprimir, while, if, for, ID
V	Variable
D	entero, decimal, booleano, carácter, cadena
T	0, 1, 2, 3, 4, true, false, cadena
L	Id, e
Parametro	entero, decimal, booleano, carácter, cadena, e
Num	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, e
Dig	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, e
Ope	+ - * / % = == < > >= <= e

Segundos

No terminal	Siguientes
E	\$
C	\$
V	;,
D	id
T	id, e, \$
L	;
Parametro	variable, imprimir, while
Num	;, id, e, \$
Dig	;, id, e, \$
Ope	id

Tabla de Análisis Sintáctico I

[illegible]

Tabla de Análisis Sintáctico II

	booleano	carácter	cadena	true	false	1, 2, 3	+ -* / % = == < > >= <=	;	\$
E									
C									
V									
D	booleano	carácter	cadena						
T			false	NUM.NUM	true	NUM			
L								e	
Parametro	e	e	e						
Num						DIGITO DIGITO			
Digito						0 1 2 3 4 5 6 7 8 9			
Ope							+ -* / % = == < > >= <=		

Tabla de Transición

	[0-9]	[A-Z]	[a-z]	“, ;	< , >	+, -, *, /, %	{, }, (,)	=	.	λ
S0	S1	S4	S4	S5	S6	S8	S9	S10	-	-
S1	S1	-	-	-	-	-	-		S2	-
S2	S3	-	-	-	-	-	-		-	-
S3	S3	-	-	-	-	-	-		-	-
S4	S4	S4	S4	-	-	-	-		-	-
S6	-	-	-	-	-	-	-	S7	-	S8
S10	-	-	-	-	-	-	-	S7	-	S7

Estado Inicial S0

Cadena	[0-9]	[A-Z]	[a-z]	“, ;	< , >	+, -, *, /, %	{, }, (,)	=	.	λ
Estado	S1 S3 S4	S4	S4	S5	S6	S8	S9	S7 S10	S2	S7 S8