

“Manual de Usuario, Analizador Léxico”

Presentación

El objetivo de este manual es dar al usuario una guía para que pueda manipular y utilizar la aplicación del analizador léxico. Poder dar asistencia a los usuarios y enseñarle la funcionalidad de la respectiva Aplicación.

Este programa consiste en enviar un archivo de texto que contenga información o caracteres la cual pasara por una tabla de estados que determinara que tipo entrada se está recibiendo y la catalogara respecto a su gramática léxico-funcional en su clasificación lingüística. Se solicitarán tokens uno a uno para ir teniendo un reporte individual de las expresiones analizadas y al final un listado del resultado de la operación.

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.
- Desarrollar diagramas de clase como parte del análisis del problema.
- Desarrollar de manual técnico.

Instrucciones de Programa

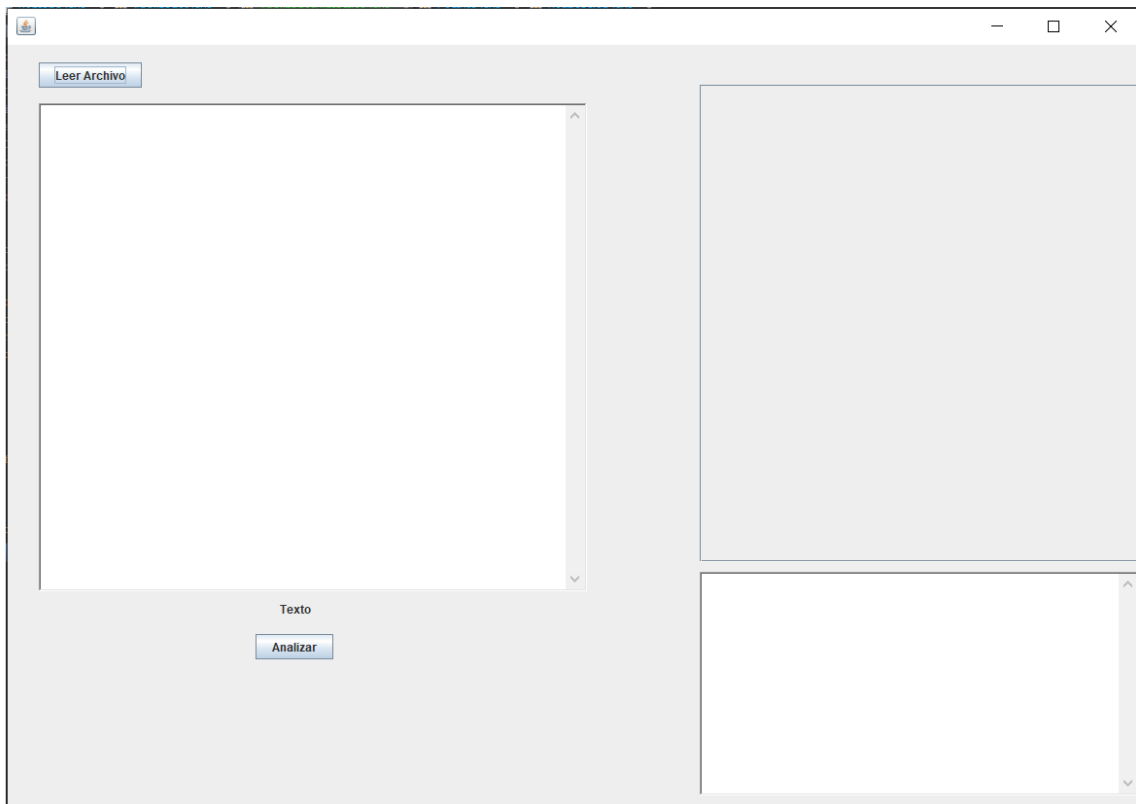
1. Instalar JDK.
2. Ejecutar JAR (ejecutable).
3. Declarar función Principal
4. Buscar Archivo de Texto a Analizar.
5. Analizar Token
6. Pedir Token
7. Verificar análisis léxico-funcional
8. Generar pila de tokens
9. Ver Resumen de actividad
10. Salir



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.
- Experiencia con aplicaciones Java (funcionamiento).

Analizador



Gramática:

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

	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							+ -* / % = == < > > = <=		

Alfabeto:

Autómata: Un autómata es un modelo matemático para una máquina de estado finito (FSM sus siglas en inglés). Una FSM es una máquina que, dada una entrada de símbolos, "salta" a través de una serie de estados de acuerdo a una función de transición (que puede ser expresada como una tabla).

NetBeans: es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

JavaScript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, 3 basado en prototipos, imperativo, débilmente tipado y dinámico.

Node: Node es un programa de servidor.

JAVA: es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable }

Pila y cola: Las pilas y colas son estructuras de datos que se utilizan generalmente para simplificar ciertas operaciones de programación. Estas estructuras pueden implementarse mediante arrays o mediante listas enlazadas.

Gramática: Gramática (autómata) Una gramática ("G") desde el punto de vista de la teoría de autómatas es un conjunto finito de reglas que describen toda la secuencia de símbolos pertenecientes a un lenguaje específico L.

CSS: Las siglas CSS (Cascading Style Sheets) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc...) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma masiva.