

Manual Técnico Proyecto 1
Organización Lenguajes Compiladores 1.

Índice

Introducción	2
Objetivos.....	2
Conocimientos Previos	2
Requerimientos.....	2
Descripción de clases y funcionalidad.....	3
.....	

Introducción

En este documento se describe los aspectos técnicos informáticos de la aplicación de escritorio, de manera que cualquier técnico informático pueda entender y comprender la lógica dentro del programa, para así poder darle mantenimiento y actualizarla si es necesario.

Objetivos

Instruir al lector para el uso adecuado del código del programa, así mismo dar a conocer la lógica detrás del programa, de manera que se le facilite al lector la manipulación del código del programa.

Conocimientos Previos

Los conocimientos que deberán tener las personas que manejen el programa son:

- Conocimiento de java
- Manejo de gramaticas
- Manejo de JFrame
- Manejo de archivos Jflex.
- Manejo de archivos Cup.
- Fases de un compilador.

Requerimientos

- Netbeans 8.2 o superior
- 2gb de ram
- Intel Core 2 Duo 2
- windows xp, 7,8,10

Descripción de clases y funcionalidad

Paquete	Clase	descripción
Codigo	Árbol.java	Contiene el nodo raíz que da estructura al árbol AST, se grafica con los métodos GraficarSintactico y GenerarDot, Reportar. GraficarSintactico contiene las variables necesarias se construye conforme se van recorriendo los nodos, con el return de la grafica por medio de un fichero y File se genera el .dot que es para Graphiz.
	FrmPrincipal.java	Encargado de mostrar la parte grafica al usuario, contiene paneles, botones y labels para dar sentido a la interfaz grafica.
	Lexer.flex	Encargado de almacenar los tokens que se utilizarán en la gramática.
	Lexer.java	Por medio del Lexer.flex se genera el Lexer.java con los tokens pero en una sintaxis para que Java la pueda reconocer y utilizar.
	LexerCup.flex	Se encarga de la parte sintáctica, encargado de recoger errores en la gramática.
	LexerCup.java	Extension generada por el LexerCup.flex que puede reconocer Java para poder utilizarla.
	Nodo.java	Clase que contiene token, lexema, línea y columna encargado de darle construcción al árbol AST.
	Principal.java	Clase Main que contiene las rutas de las librerías que estamos utilizando, se crean 3 variables string donde se pasan las rutas para ser reconocidas por Java.
Codigo	Sintax.cup	En esta clase se crean las gramáticas para poder obtener la lectura deseada, también se generan las partes del árbol AST según la estructura deseada.
Codigo	Sintax.java	Clase generada por Sintax.Cup para que Java pueda reconocer todo lo que se necesita y pueda implementarlo a la salida deseada.
	Tokens	Clase donde se definen todos los tokens utilizados para el LexerCup y el SintaxCUP.
	Sym.java	Clase generada al momento de ejecutar la clase Principal.java para poder utilizar lo que se tiene en las clases Lexer.flex y LexerCup.flex

