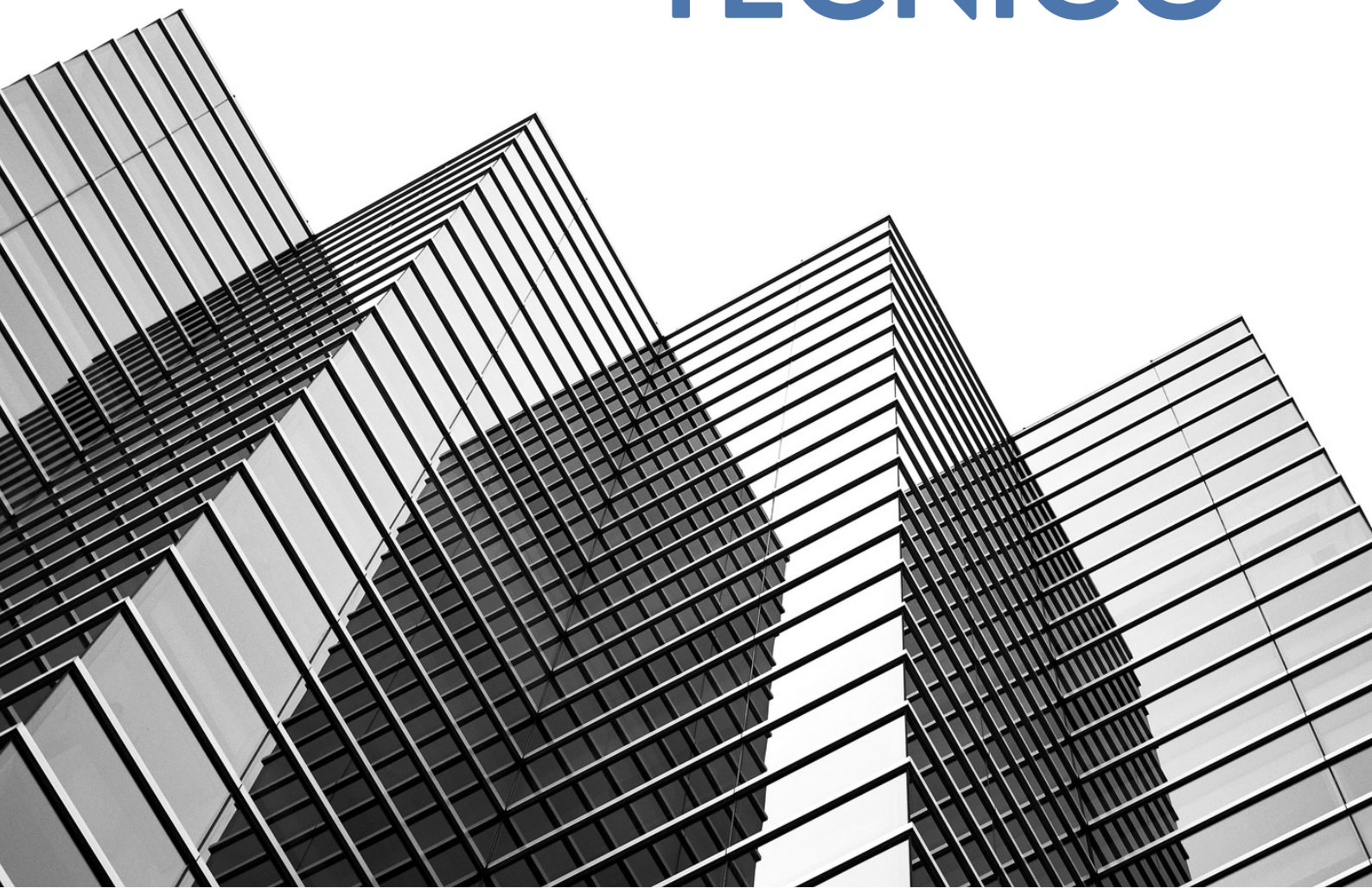




Compiladores y
Ensambladores 1

MANUAL TÉCNICO



Joshua David Osorio Tally
202110773

Acerca del programa

01

Este programa fue desarrollado con el lenguaje de programación JAVA como base principal, utilizando su interfaz. También se usó JFLEX para crear el analizador léxico y CUP para el analizador sintáctico

02

Este programa cuenta con diferentes clases (explicación en las siguientes páginas) y el uso de librerías nativas de JAVA.

03

El objetivo principal del programa es crear un analizador léxico y sintáctico que sea capaz de reconocer una gramática. Posterior a eso, utilizar los datos obtenidos y crear reportes gráficos con Graphviz y archivos JSON.

Clases y Métodos



Clase Interfaz

Esta es la clase principal del archivo, es quien tiene la interfaz gráfica, las funcionalidades de los botones y llama a los métodos de las demás clases.

Archivo lexico.jflex

En este archivo se tienen establecidas las gramáticas a usar en el analizador. Su objetivo es tener las expresiones regulares que se usan en el CUP.

Archivo sintáctico.cup

En este archivo se tiene la estructura a utilizar. Con este archivo generamos el analizador sintáctico.

Clase Generador

Esta clase utiliza los archivos JFLEX y CUP y los genera en un archivo JAVA, para que posteriormente los podamos usar en nuestra clase principal.

Clase arbol

Esta es nuestra clase nodo, recibe los datos y los devuelve al archivo AFD.

Métodos y Clases

Clase AFD

Esta es la clase más importante, ya que acá se hace la mayoría de los métodos para analizar los archivos y generar los reportes, los métodos que se tienen son los siguientes:

asignar(): En este método asignamos los valores que se leen a lo que necesitamos
metodo(): Este método genera las funciones del árbol, los First and Post, el número de hoja y la anulabilidad del nodo.
crear(): A través de este método, creamos el archivo graphviz y este mismo genera el archivo dot y la imagen JPG

Clase Excepcion

Esta clase actúa como un nodo, ya que nos devuelve los errores que generamos en la lectura del archivo



Interface

Acá al ser nuestra clase principal, tenemos el manejo de los botones y las demás funcionalidades, entre las funcionalidades principales está:

`btngenerar()` En esta función llamamos al archivo `lexer` y `sintaxis`, para poder enviar los datos y posteriormente recibirlos y proyectarlos. Este método también analiza los errores y genera el reporte.

`btnanalizar()` En esta función se genera el archivo `JSON` y se evalúa la cadena propuesta

Finalmente, la interfaz se muestra así:



Requerimientos del Sistema

Sistemas operativos compatibles Windows 8.1 (x86 y x64) Windows 8 (x86 y x64) Windows 7 SP1 (x86 y x64) Windows Server 2008 R2 SP1 (x64) Windows Server 2012 (x64) Windows Server 2012 R2 (x64)

Requisitos de hardware Procesador a 1.6 GHz o superior 1 GB (32 bits) o 2 GB (64 bits) de RAM (agregue 512 MB al host si se ejecuta en una máquina virtual) 3 GB de espacio disponible en el disco duro Disco duro de 5400 RPM Tarjeta de video compatible con DirectX 9 con resolución de pantalla de 1024 x 768 o más.