5-9-2021

Manual Técnico

Copy Analyzer FIUSAC - Proyecto 1

Eduardo René Agustin Mendoza 201801627
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1 - N

Requerimientos para la Aplicación.

- Sistema Operativo requerido: Windows 8 o superiores.
- Memoria RAM: 4 GB o Superiores.
- Requerimientos de Ejecución: Tener instalado JDK acorde a la versión del sistema que se tiene (Windows SE Development Kit 8) para Linux y iOS su respectiva versión esta en la página oficial https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html
- Procesador: Intel Celeron o Superiores.
- Tipo de Sistema: Recomendado 64bit.
- Librería JFlex
- Librería Cup.
- Librería JFreeChart.

Clases Utilizadas.

Paquete Analizadores (FCA y JS).

- Léxico
- Sintáctico

Estas clases se generan a partir de un archivo llamado también léxico y sintáctico, pero estos carecen de una extensión ya que en estos se escribirán la estructura de dicho lenguaje, en el léxico se declaran los tokens esperados dentro del archivo y las gramáticas utilizadas y en el sintáctico se declaran todos los terminales, no terminales y las producciones que generan los terminales y no terminales esto para poder tener una estructura posible esperada de los lenguajes ya definidos, los terminales son los tokens definidos anteriormente y en los no terminales se pueden declarar las transiciones que se tienen de un terminal a otro o incluso a otro no terminal.

Paquete Información.

- NodeGraph
  En esta clase se declaran todos los atributos para poder guardar la información de la estructura del archivo FCA.
- ValueNode
  En esta clase se declaran los identificadores y valores que puede tener las variables declaradas dentro del archivo.
- Errores
  En esta clase se declara la lista de errores
- Exceptions
  En esta clase se declara el manejo de la lista de errores.

- JsNode
  En esta clase se describe que tipo de identificador es y que es lo que trae como parámetro.
- JsInformacion
  En esta clase se va almacenando la información recaudada en el archivo Js la cual nos servirá más adelante para la creación de las gráficas de líneas

Paquete Ventanas.

- Ventana Inicial
  Esta clase es un form el cual lleva la interfaz gráfica de la aplicación.

Métodos Utilizados.

- Constructor de la clase NodeGraph
  Se pasan los posibles datos que pueda tener dentro de las gráficas y al igual se realiza uno donde todo este vacío.
- Constructor de la clase ValueNode
  Se pasan los posibles datos que pueda tener las variables dentro del archivo y al igual se realiza uno donde todo este vacío.
- Constructor de la clase Exceptions

Ventana Inicial.

- ParseoFCA.
  En este método se realiza el parseo del archivo ingresado por el usuario mediante la interfaz, posterior a esto pasara a los analizadores léxicos y sintácticos respectivos para poder generar las gráficas y posterior seguir con el analizador de copias
- ParseoJS.
  En este método se realizo el parseo para los archivos javascript, este método es irrelevante ya que solo se utilizo para validar la estructura del archivo y poder unirlo al parseo FCA.
- GraficarBarras.
  En este método se van a iterar las listas creadas dentro del paquete NodeGraph y ValueNode esto para poder ir formando las gráficas de barras.

- GraficaPie.
  En este método se van a iterar las listas creadas dentro del paquete NodeGraph y ValueNode esto para poder ir formando las gráficas de pie.
- buscarRuta.
  En este método se utilizaron los directorios que venían dentro del archivo FCA con el fin de buscar en ambas carpetas archivos con el mismo nombre esto para poder detectar posibles copias, utilizando un contador de métodos/funciones, clases, variables y sentencias.

Resolución del Problema.

Actualmente en algunos cursos de Ingeniería en Ciencias y Sistemas existe una población bastante elevada que generalmente alcanza un promedio de 70 estudiantes por curso, debido a esta problemática la detección de copias entre proyectos se ha convertido en una tarea compleja que requiere de mucho tiempo y nuevas técnicas para agilizar el análisis de proyectos. Como estudiante de Organización de Lenguajes y Compiladores 1 se le solicita crear una herramienta que sirva de apoyo a tutores académicos en el análisis del código fuente de los diferentes proyectos desarrollados por los estudiantes, esto buscando una agilización y eficiencia en el proceso de búsqueda de copias.

En la escuela de Ciencias y Sistemas se nos requirió realizar una aplicación para la detección de dichas copias esto para hacer conciencia a nuestros estudiantes esto para ver la transparencia de sus proyectos. Para la siguiente aplicación se tuvo que realizar 2 analizadores los cuales fueron el analizar léxico y sintáctico los cuales su funcionalidad principal es ver que la estructura de un lenguaje esté de forma correcta, al tener dichos analizadores se creo una interfaz gráfica la cual facilita la carga de archivos en el cual vendrán dos directorios los cuales tendrás las carpetas en donde estarán los archivos. La idea principal de la aplicación es de que busque archivos con el mismo nombre y los lea para poder generar una gráfica por la cual se vera que nivel de copia se puede tener dentro del archivo y esto se hará por medio de un conteo de clases, métodos/funciones, variables y sentencias con esto ya se podría dar un veredicto el cual condicionaría al estudiante si copio de alguien más. Por ultimo se generan las gráficas especificadas dentro del primer archivo dentro de una página web y así mismo se da una lista de posibles tokens y errores que se pueden tener en todo el archivo.

Flujo recomendado del Proyecto.