



UNIVERSIDAD

— DE SAN CARLOS DE GUATEMALA —

Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización de Lenguajes y Compiladores 1
Catedrático: Ing. Manuel Castillo
Auxiliar: René Corona
Proyecto 1

Proyecto 1, Manual Técnico “Copy Analyzer”

NOMBRE	CARNÉ
Julio César Eduardo Flores Tubac	2012-12558

Guatemala, miércoles 8 de septiembre del 2021

Introducción

El siguiente documento contiene información sobre el uso adecuado de la aplicación solicitada COPY ANALIZER. El cual cuenta con pasos a seguir para usarla de la forma correcta.

Fue realizada en el lenguaje JAVA, con librerías como JFLEX, JCUP principalmente, y con programación orientada a objetos.

Este documento contiene características detalladas de como se implementó la aplicación COPY ANALIZER.

Objetivos

- Aprender el uso de librería JFLEX
- Aprender el uso de librería JCUP
- Aprender el significado de token, lexema
- Implementar algunas de las fases de un compilador
- Aprender a desarrollar expresiones regulares
- Aprender a crear gramáticas.

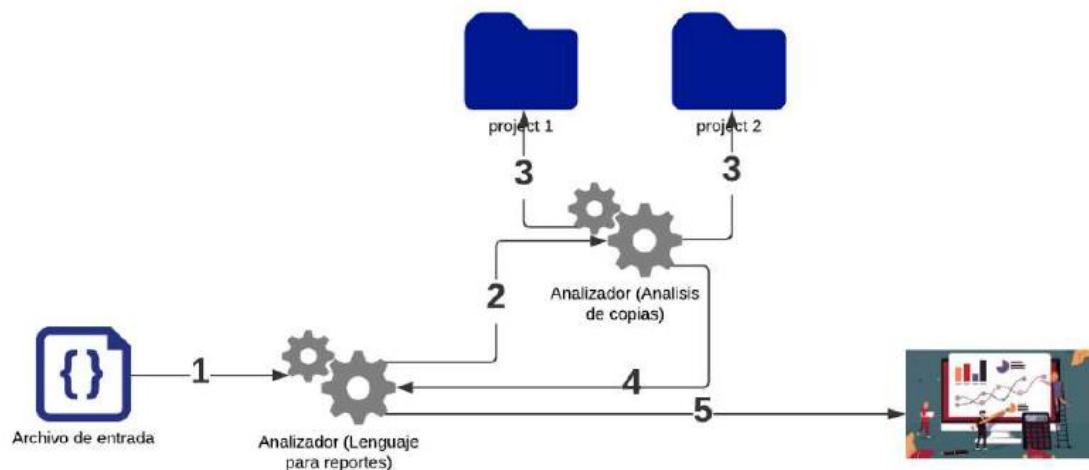
Descripción del problema.

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.

Descripción de la solución.

Se pretende crear un plan piloto para la identificación de copias entre proyectos a través de una aplicación capaz de generar reportes estadísticos sobre distintos aspectos de un proyecto, para dicha aplicación se deberá crear un analizador de archivos Javascript capaz de identificar repitencias entre proyectos y que a su vez genere reportes estadísticos basados en un puntaje de repitencia de copia entre proyectos. Por cuestiones de tiempo el plan piloto realizará la comparación únicamente entre 2 proyectos y entre archivos que posean el mismo nombre y extensión. La aplicación contará con su propio lenguaje de generación de reportes por lo que será necesario crear dos analizadores, un primer analizador el cual servirá para identificar los proyectos que se analizaran y reportes que se deben generar. Adicionalmente necesitará un segundo analizador que identifique las posibles copias entre los proyectos. Para hacer uso de la aplicación el tutor desarrollará un archivo de entrada en el cual indicará la ruta de los proyectos a comparar, adicionalmente desarrollará en dicho archivo los reportes que desea generar basándose en los porcentajes de copia que se obtengan en la comparación de archivos, una vez desarrollado el archivo de entrada el tutor procederá a ejecutar la aplicación y obtendrá resultados que le permitan agilizar el proceso de identificación de copias.

Resumen del funcionamiento



1. El tutor ingresa el archivo de entrada
2. La aplicación analiza el archivo de entrada con el analizador 1
3. La aplicación busca posibles copias haciendo uso del analizador 2
4. El analizador 2 almacena la información sobre las copias detectadas
5. La aplicación genera los reportes finales en base a la información detectada por el analizador 1.

Fuente de desarrollo

- Sistema operativo Windows 10
- Sistema Buffer de 64 bits
- IDE NetBeans
- Lenguaje Java
- Librerías JFLEX, CUP, GSON

Requerimientos del usuario para este documento

- Conocimientos básicos sobre programación orientada a objetos
- Conocimientos básicos sobre uso de interfaz gráfica (JFrame)
- Conocimientos sobre el lenguaje de programación Java
- Conocimientos sobre las librerías JFLX y CUP

Descripción de la aplicación.

- Clases
Se crearon distintas clases para poder tener un mejor control sobre la aplicación, existen clases de tipo nodo, de tipo lista, etc.
- Métodos/Funciones
Se crearon tanto funciones como métodos para poder seguir con un flujo correcto de la solución y poder hacer cálculos cuando se requerían.
- Atributos/Variables
Se crearon varios atributos de tipo variables para poder almacenar datos temporalmente.
- Objetos
Se instanciaron varias clases para poder hacer un uso de estas en el momento que se requería y en la clase que se necesitaba.