

---

**Universidad de San Carlos de Guatemala**  
**Facultad de Ingeniería**  
**Organización de Lenguajes y Compiladores 1 N**  
**Aux Emely García**  
**Maynor Octavio Piló Tuy**  
**201531166**

# FIUSAC Copy Analyzer

## Manual Técnico

### Descripción

FIUSAC Copy Analyzer es una herramienta que sirve 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.

El flujo del programa funciona con una interfaz grafica que permite la carga de archivos con extensión “.fca”, estos archivos contienen toda la información para el desarrollo de los reportes, así mismo contienen la ruta de los proyectos a analizar.

Este programa fue desarrollado en lenguaje java.

### Requerimientos mínimos del sistema

El sistema debe de tener los siguientes requerimientos instalados:

- JDK 8.1 o superior
- Librería Jflex
- Librería Jcup
- Librería JfreeChart
- Windows 7,8,10

## 1. Funcionamiento de la aplicación:

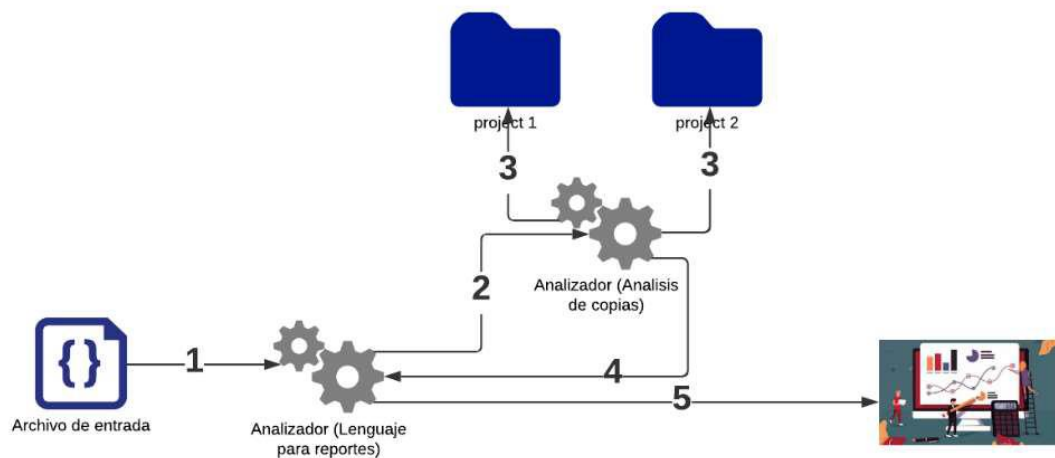
la aplicación tendrá la funcionalidad de analizar archivos de tipo JavaScript con el mismo nombre de dos proyectos diferentes. Estos serán analizados y se tendrá como producto reportes estadístico de los datos analizados.

El contenido de los reportes serán especificados dentro del archivo “.fca”, este tendrá de la misma forma la ruta de los proyectos a analizar.

Lo siguientes reportes son:

- Reporte general: dicho reporte contiene una tabla de resumen del los proyectos ingresados con la cantidad de clases, métodos, variables, comentarios.
- Reporte de errores: contiene todos los errores encontrados durante el análisis de los archivos cargados del proyecto.
- Reporte de Tokens: contiene todos los tokens reconocidos durante el análisis de cada archivo cargado.
- Reporte JSON: este reporte contiene todas las clases, métodos, comentarios, variables con cierto grado de repitencia, ya que contiene el punteo de cada uno de los mismos en formato .json.

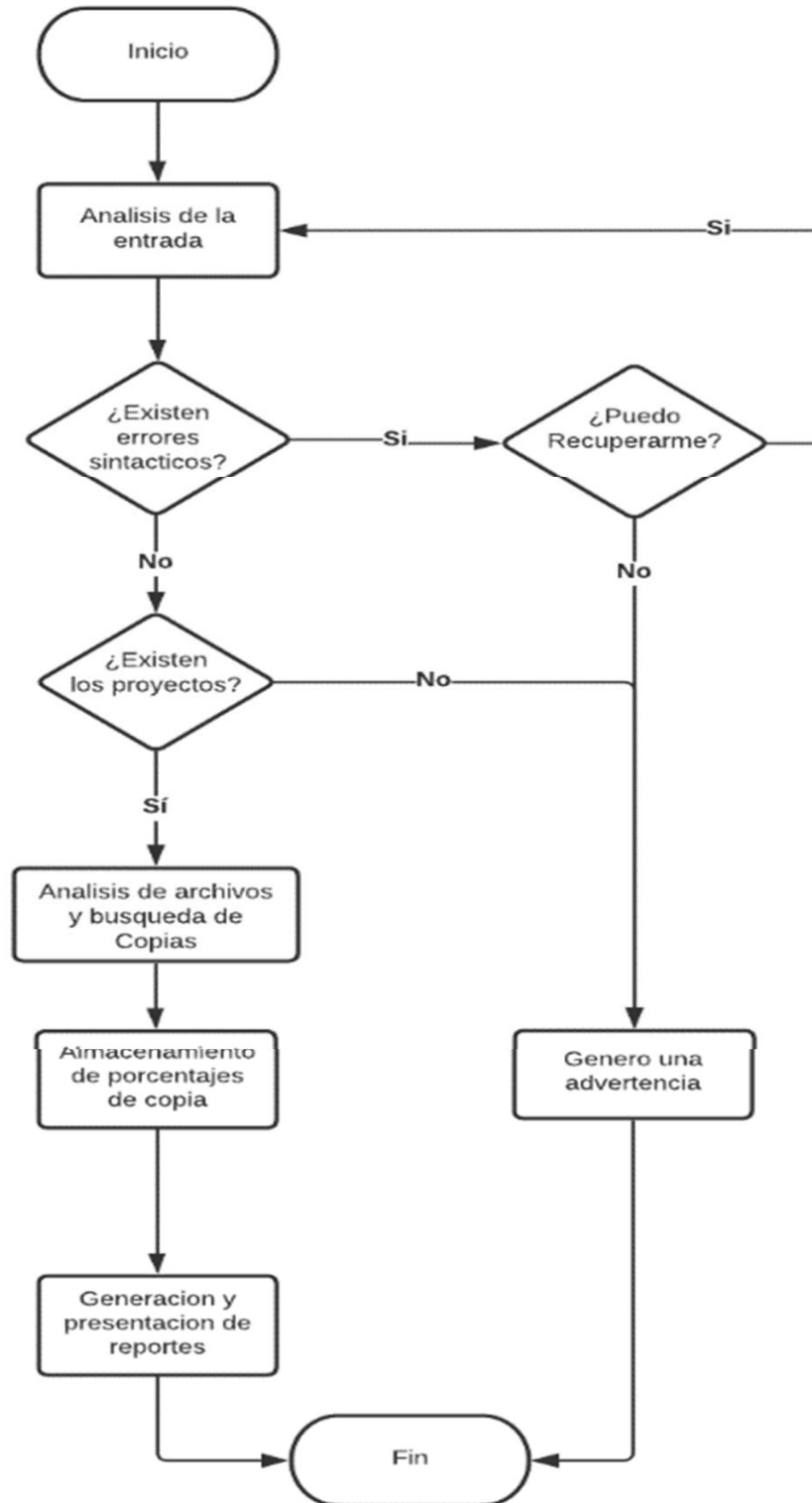
Fig. 1 funcionamiento de la aplicación.



Fuente: Enunciado Proyecto 1, Organización de Lenguajes y Compiladores 1 2s-2021

## 2. Diagrama de Flujo Fiusac Copy Analyzer

El diagrama de flujo que mejor describe el programa es el siguiente:



Fuente: Enunciado Proyecto 1, Organización de Lenguajes y Compiladores 1 2s-2021

### 3. Principales Clases, funciones y métodos de la aplicación.

Paquete:		Analisis
Paquete utilizado para el análisis léxico y sintáctico de los archivos .fca		
Clases	Generador. java	Clase generada para la compilación de los archivos lexico.jflex y sintactico.cup
	Parser.java	Clase generada a partir de la compilacion del archivo sintactico.cup
	Scanner.java	Clase generada a partir del la compilacion del archivo lexico.jflex
	Sym.java	Clase auxiliar generada con el archivo sintactio.cup

Paquete:		Analisis_Reporte
Paquete utilizado para el análisis léxico y sintáctico de los JavaScript		
Clases	Generador. java	Clase generada para la compilación de los archivos lexico.jflex y sintactico.cup
	Parser.java	Clase generada a partir de la compilacion del archivo sintactico.cup
	Scanner.java	Clase generada a partir del la compilacion del archivo lexico.jflex
	Sym.java	Clase auxiliar generada con el archivo sintactio.cup

Paquete:		OLC_P1_201531166
Paquete utilizado que contiene la función principal de la aplicación, así como la interfaz		
Clases	OLC_P1_201531166.java	Clase principal que contiene el funcionamiento inicial de la aplicación.
	Interfaz.java	Clase JFrame, contiene la interfaz de la aplicación.
	Iniciar_Analisis.java	Clase sirve de comunicación entre la interfaz y las Data para la ejecución de análisis y reportes.
	Tabs.java	Clase auxiliar de la interfaz para crear, eliminar y verificar las pestañas que se crean o eliminan de la interfaz.

Paquete:		Data
Paquete que contiene las clases y objetos para el análisis y procesamiento de datos que se cargan con los archivos .fca y los archivos .js		
<b>Clases</b>	Barras.java	Clase que contiene al objeto de tipo barras, este objeto contiene todas las instrucciones de las graficas de tipo barras a generarse en el reporte.
	Comentario.java	Clase que contiene al objeto comentario, dicho objeto guarda todos los comentarios que cargaron de los archivos.js
	Data.java	Clase orquestadora de las funciones de análisis de los archivos js, procesamiento de datos y de la generación de reportes del proyecto.
	Error.java	Clase que contiene al objeto Error, dicho objeto guarda todos los errores de tipo léxico y sintáctico que se cargaron de los archivos .fca
	Lineas.java	Clase que contiene al objeto de tipo Líneas, este objeto contiene todas las instrucciones de las graficas de tipo Líneas a generarse en el reporte.
	Metodos.java	Clase que contiene al objeto Método, dicho objeto guarda todos los Métodos que cargaron de los archivos.js
	Nombre_clases.java	Clase que contiene al objeto nombre, dicho objeto guarda todos los nombres de las clases que se cargaron de los archivos.js
	variables.java	Clase que contiene al objeto variable, dicho objeto guarda todas las variables que se cargaron de los archivos.js
	Pe_Clase.java	Clase que contiene todas las clases con sus respectivos punteos específicos obtenidos durante el procesamiento de datos.
	Pe_Comentario.java	Clase que contiene todas los comentarios con sus respectivos punteos específicos obtenidos durante el procesamiento de datos.
	Pe_metodo.java	Clase que contiene todos los métodos con sus respectivos punteos específicos obtenidos durante el procesamiento de datos.
	Pe_variables.java	Clase que contiene todas las variables con sus respectivos punteos específicos obtenidos durante el procesamiento de datos.

Método/ función	Tipo	Parámetros	Descripción
Clase Generador			
generarCompilador()	Static void	--	Analiza los archivos .fca de entrada
Clase Data			
Cargar_archivo()	Static void	--	Analiza las rutas reconocidas y guarda todos los archivos.js en cada ruta
Comparar_archivos()	Static void	--	Compara cada archivo encontrado, si son iguales llama a la función Analizar archivos para que sean analizados.
Analizar_archivos()	Static void	1 String	Analiza al ruta que recibe como parámetro y llama a los analizadores para el respectivo análisis del archivo.js
Nombre_clases()	Static void	1 String 2 int	Función que se llama desde el analizador sintáctico, recibe los parámetros del nombre de la clase, las líneas iniciales y finales de la clase.
Variables_clase()	Static void	2 String	Función que se llama desde el analizador sintáctico, recibe los parámetros del nombre de la clase y la variables correspondientes de la clase.
Metodos_clase()	Static void	3 String 2 int	Función que se llama desde el analizador sintáctico, recibe los parámetros del nombre de la clase , el nombre del método, los parámetros del método, las líneas iniciales y finales del método.
Comentarios()	Static void	2 String	Función que se llama desde el analizador sintáctico, recibe los parámetros del nombre de la clase , y el comentario encontrado de la clase

Pe_clase()	Static void	--	Función que analiza las clases que se cargaron en los archivos y obtiene el puntaje específico de repitencia.
Pe_metodo()	Static void	--	Función que analiza los métodos de cada clase que se cargaron en los archivos y obtiene el puntaje específico de repitencia.
Pe_variables()	Static void	--	Función que analiza las variables de cada clase que se cargaron en los archivos y obtiene el puntaje específico de repitencia.
Pe_comentarios()	Static void	--	Función que analiza los comentarios que se cargaron en los archivos y obtiene el puntaje específico de repitencia.
Puntaje_general()	Static void	--	Función correspondiente a la generación del puntaje general del análisis de los proyectos.
Generar_gbarras()	Static void	--	Función que se encarga de generar las graficas de barra en formato png. Para los reportes.
Generar_gpie()	Static void	--	Función que se encarga de generar las graficas tipo Pie en formato png. Para los reportes.
Generar_glinea()	Static void	--	Función que se encarga de generar las graficas de linea en formato png. Para los reportes.
Reporte_general()	Static void	--	Función que se encarga de generar el reporte general del análisis
generar_reporte_errores()	Static void	--	Función que se encarga de generar el reporte de errores en el análisis léxico y sintáctico de lo archivos.
generar_reporte_token()	Static void	--	Función que se encarga de generar el reporte de tokens encontrados en el análisis léxico y sintáctico de lo archivos.

---

generar_json()	Static void	--	Función que se encarga de generar el reporte en formato Json. De los puntajes específicos.
----------------	-------------	----	--