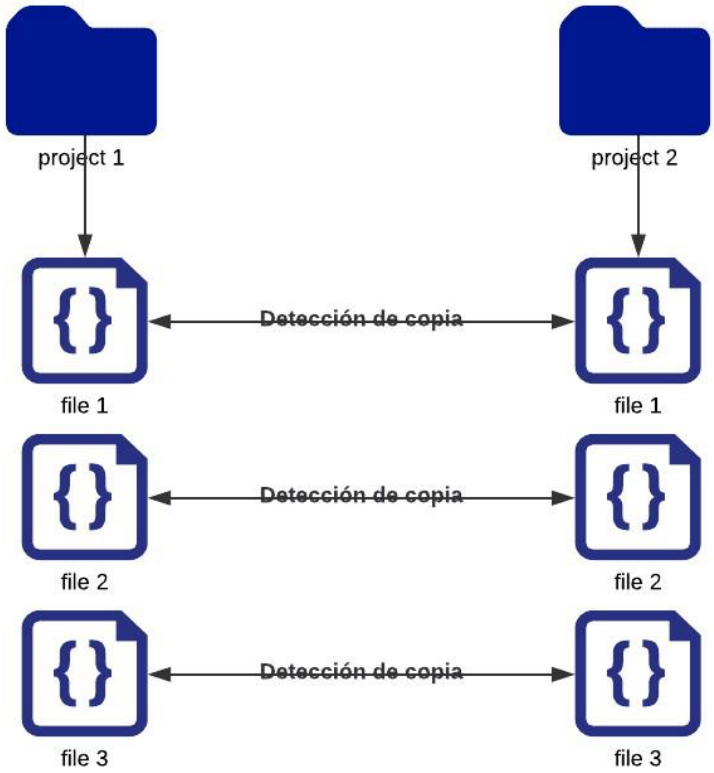
Manual de usuario

Introducción

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 el siguiente documento se darán los pasos para poder usar el programa de forma correcta.

Puntos importantes a entender para el lenguaje FCA

- El lenguaje no distinguirá entre mayúsculas o minúsculas.
- Este tipo de comentario comenzará con ## y deberá terminará con un salto de línea.
- Este tipo de comentario comenzará con #* y deberá terminar con *#
- Para poder ingresar las instrucciones dentro del Lenguaje FCA, se debe seguir la sintaxis del lenguaje.
- Para indicar el final de una instrucción dentro del lenguaje, se hará uso del signo ‘;’.
- En el lenguaje FCA es necesario poder cargar dos proyectos diferentes, con el fin de poder analizar posibles copias entre los estudiantes de un curso.
- Esta sección se utilizará para definir variables globales que podrán ser utilizadas en cualquier sitio dentro del proyecto.
- En el lenguaje de generación de reportes se hará uso de dos tipos de datos para generar una flexibilidad en la cantidad de reportes que podamos manejar.

Ejemplo:

```
##Este es un ejemplo de un archivo de reportería FCA
GenerarReporteEstadistico{
    definirglobales{
        string reporteResumen = "Reporte de Archivo file_1.js de los proyectos";
        ## variables para Reportes de Barras
        Double pr1 = 3;
        Double pr2 = 2;
        Double pr3 = 1;
        Double pr4 = 5;
        Double pr5 = 4;
        Double pr6 = 6;
        Double pr9 = 7;
        Double pr12 = 5;
```

```
        Double pr15 = 4;
        Double pr16 = 3;
        Double pe1 = 0;
        Double pe2 = 1;
        Double pe3 = 0;
        Double pe4 = 1;
        Double pe5 = 1;
        Double pe6 = 1;
        Double pe9 = 0;
        Double pe12 = 0;
        Double pe15 = 0;
        Double pe16 = 0;

        String titulobarrasesperada = "Probabilidades esperadas para variables archivo file_1.js";
        String titulobarrasreal = "Probabilidades obtenidas para variables archivo file_1.js";

        Double p1= 0.2;
        Double p2= 0.8;

        String s1 = "hanoi_generator";
        String s2 = "fact_generator";

        String t1="Comparacion de metodos";
    }
    ##Cargamos los proyectos correspondientes

COMPARE("C:\Users\Userlen\Documents\GitHub\OCL1_P1_2S21_201801155\archivos\ProyectoA
", "C:\Users\Userlen\Documents\GitHub\OCL1_P1_2S21_201801155\archivos\ProyectoB");

    GraficaLineas{
        TiTulO: reporteResumen;
        ArChIvO: "file_1.js";
    }
    graficalineas{
        titulo: "Reporte file_2";
        archivo: "file_2.js";
    }
    ## Agregamos las graficas de barras
```

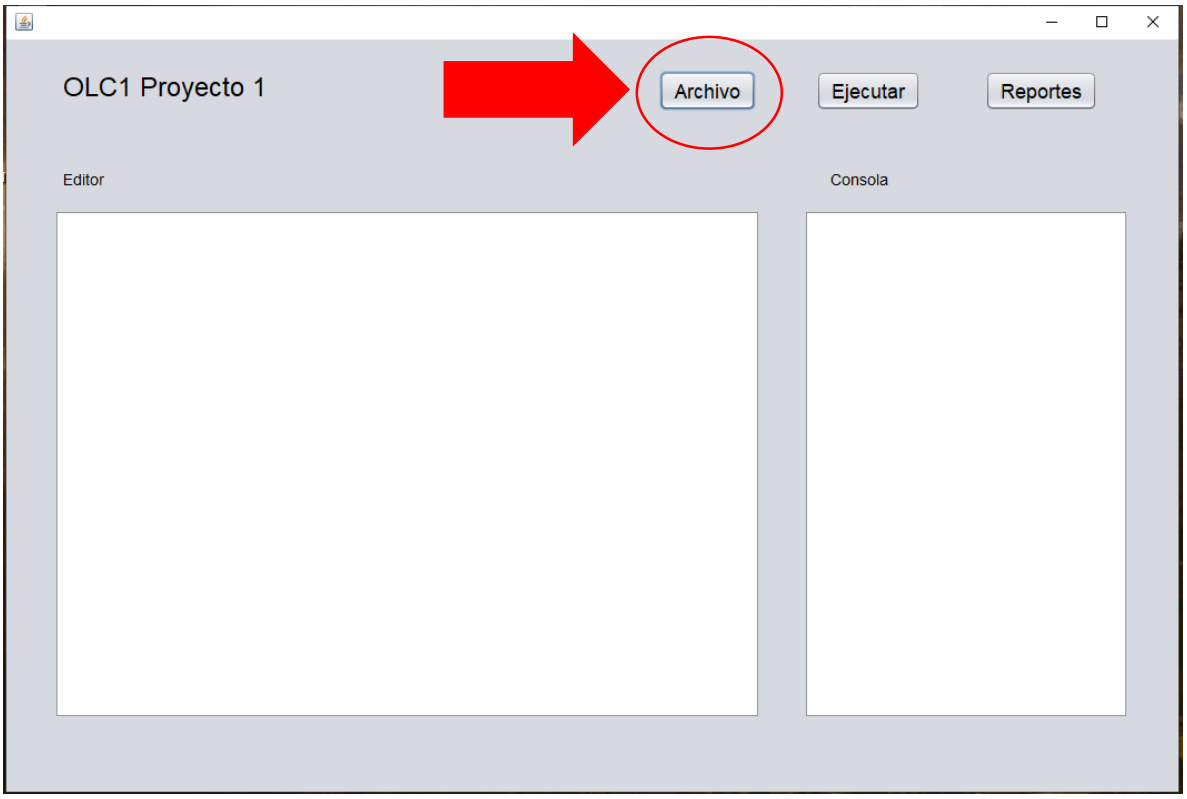
```
    GraficaBarras {
        Titulo: titulobarrasreal;
        EjeX: [ "triangle_draw", "draw_triangle_i", "sq_draw", "draw_square_i", "draw_square_j",
"draw_triangle_j", "draw_triangle_draw", "draw_square_draw", "x", "a" ];
        Valores: [ pr1, pr2, pr3, pr4, pr5, pr6, pr9, pr12, pr15, pr16 ];
        TituloX: "Nombre de las variables";
        TituloY: "Puntaje";
    }
    GraficaBarras {
        Titulo: titulobarrasesperada;
        EjeX: ["triangle_draw","draw_triangle_i","sq_draw","draw_square_i","draw_square_j",
"draw_triangle_j", "draw_triangle_draw", "draw_square_draw", "x", "a" ];
        Valores: [ pe1, pe2, pe3, pe4, pe5, pe6, pe9, pe12, pe15, pe16 ];
        TituloX: "Nombre de las variables";
        TituloY: "Puntaje";
    }
    ## AGREGANdo Graficas de Pie
    GraficaPie{
        Titulo: t1;
        EjeX: [ "fibonacci_generator", "torresDeHanoi", s1, s2 ];
        Valores: [ p1, p2, pr5, pr16 ];
    }
}
```

¿Cómo usar el Analizador de copias?

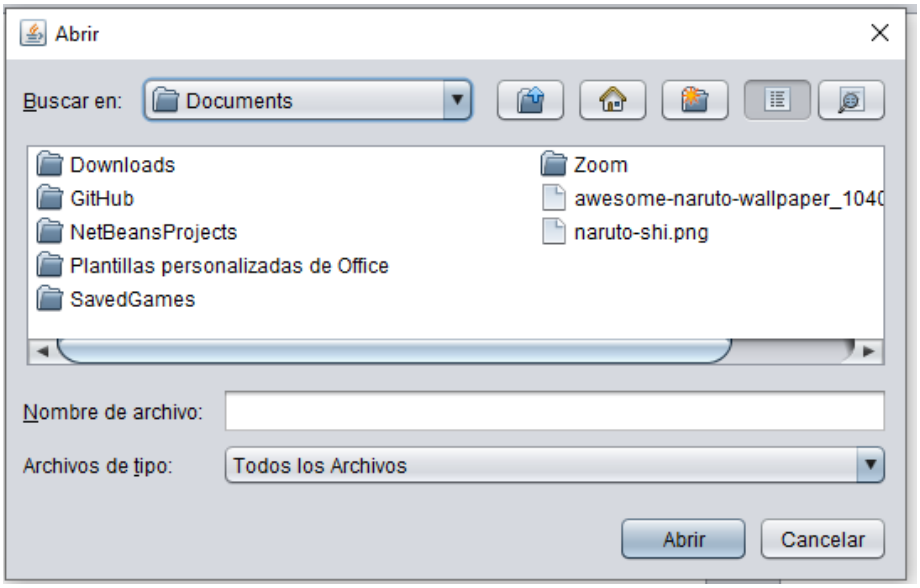
Paso 1:

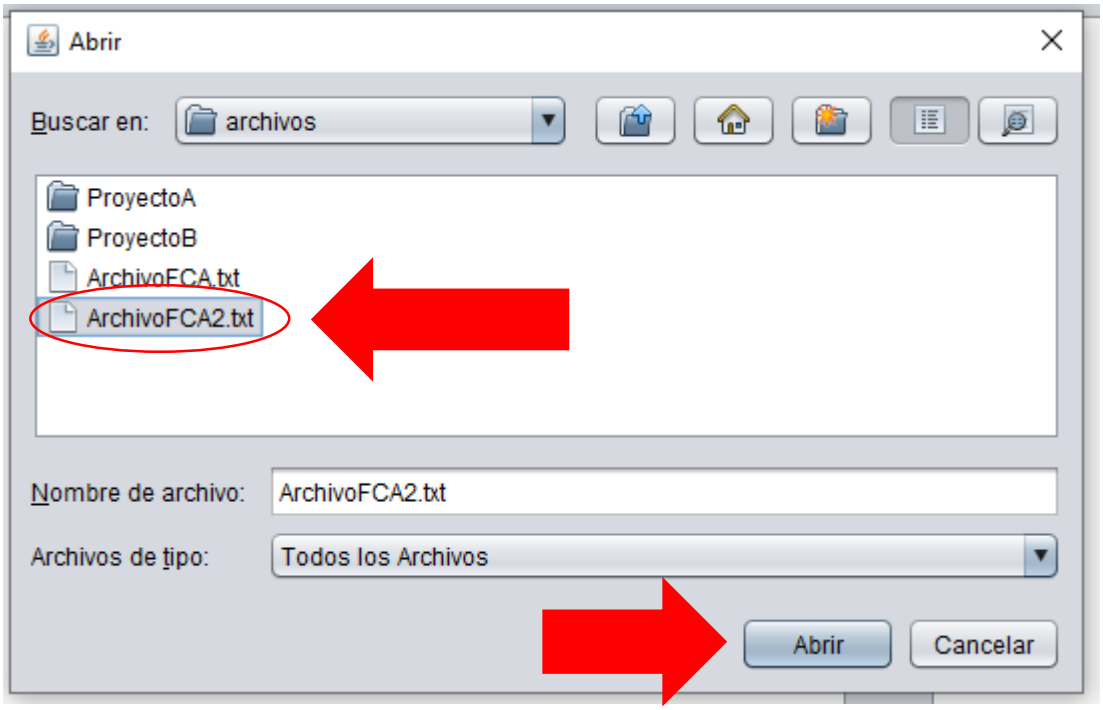
En la siguiente imagen se muestra la interfaz de inicio con la cual se puede manejar los archivos de la gramática FCA con toda la información.

Para poder seleccionar los archivos FCA se debe de dar click en el botón de archivo.



Se desplegara una ventana emergente en el cual podremos navegar en los archivos de nuestro computador para poder seleccionar el archivo que deseemos leer.

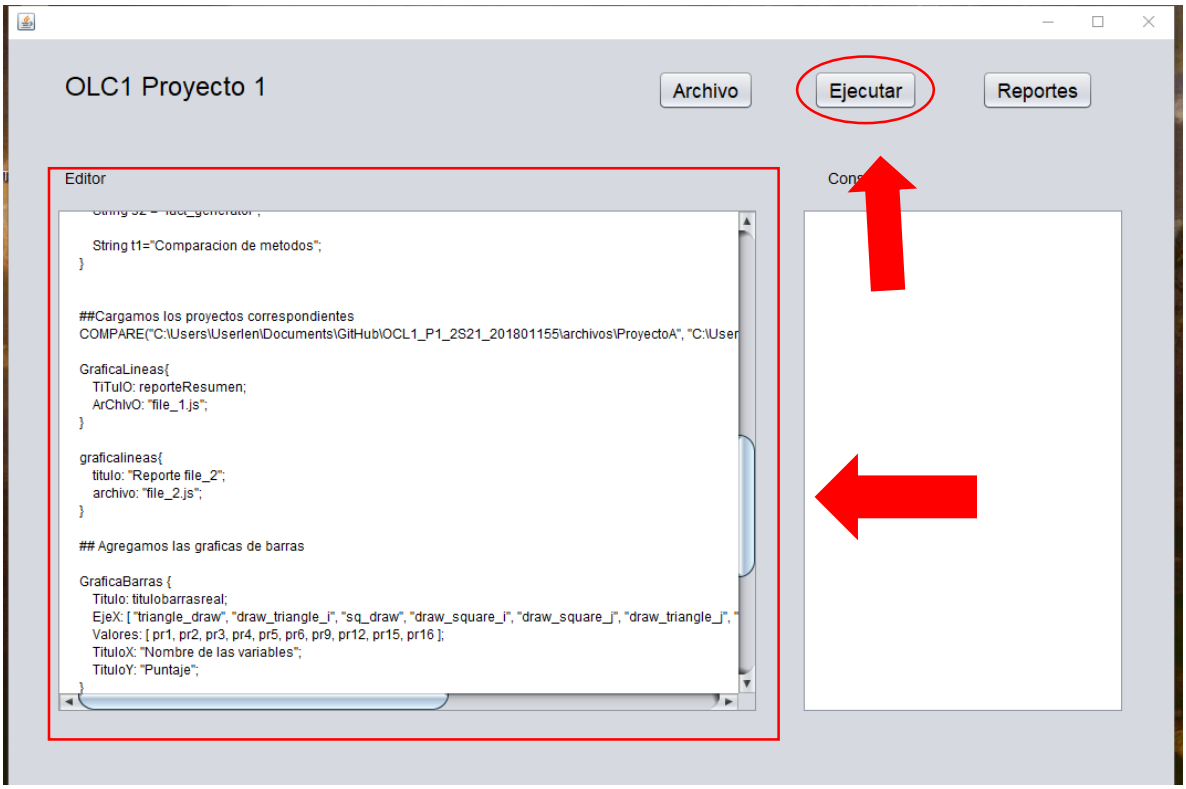




Después de seleccionar el archivo y abrirlo, toda la información que contiene el archivo se mostrara en la pantalla llamada editor, en el cual podremos editar la información del archivo como agregar y eliminar las líneas.

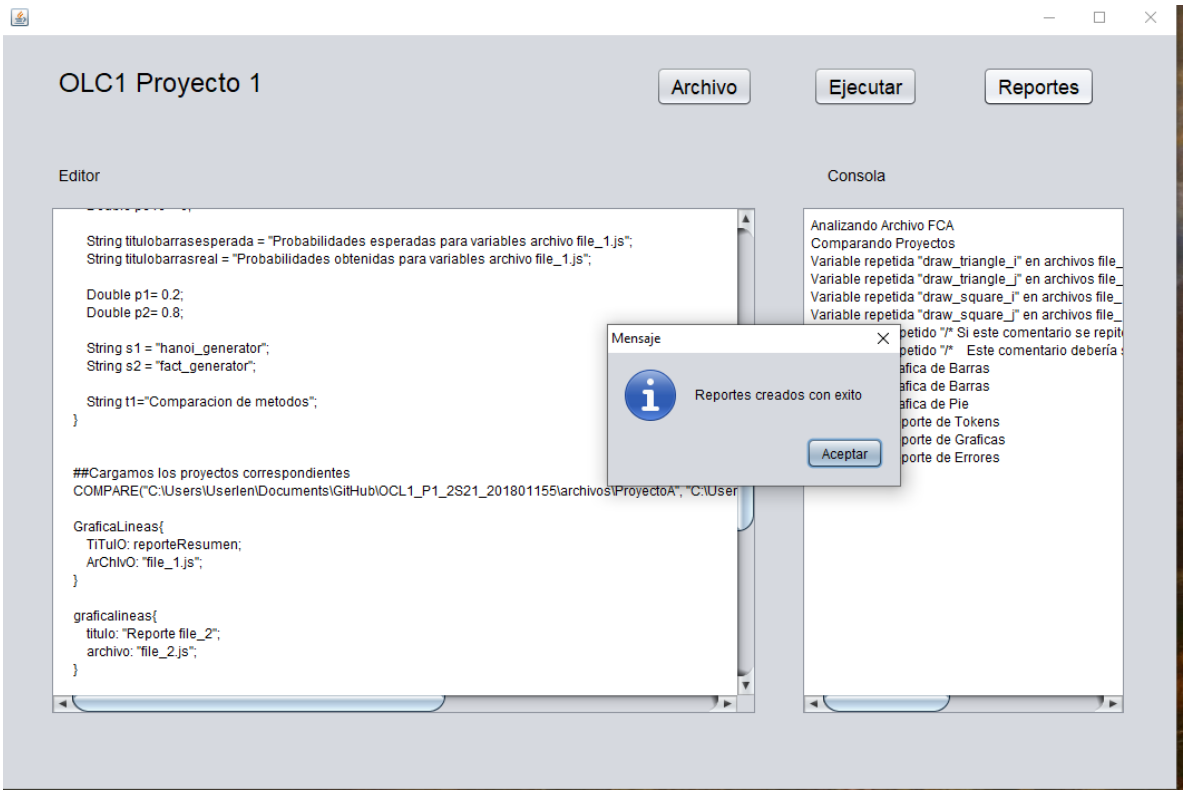
Paso 2:

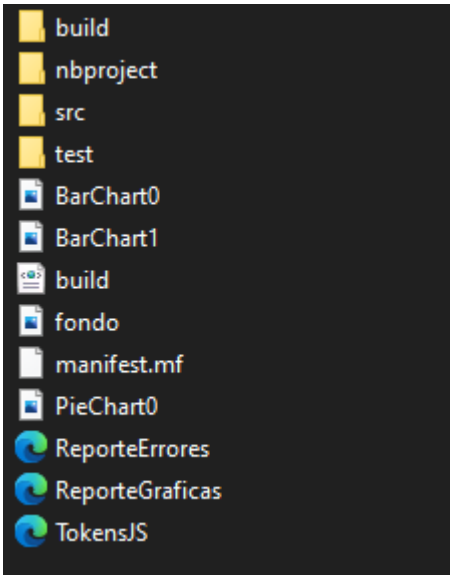
Para poder analizar los proyectos que el archivo FCA, se debe de dar click en el botón ejecutar para poder las instrucciones que contiene el archivo FCA anteriormente seleccionado, al mismo tiempo se generaran las gráficas descritas en espacio de editor.



PASO 3:

Para poder Generar los diferentes tipos de reportes, se debe de dar click en el botón de Reportes, este generara los reportes necesarios los cuales se podrán encontrar en la misma carpeta donde se localiza el programa.





En la parte de consola se mostraran los procesos por los cuales está pasando el programa como la generación de gráficas, reportes y comparación de archivos.


Consola
Analizando Archivo FCA
Comparando Proyectos
Variable repetida "draw_triangle_i" en archivos file_
Variable repetida "draw_triangle_j" en archivos file_
Variable repetida "draw_square_i" en archivos file_
Variable repetida "draw_square_j" en archivos file_
Comentario repetido "/* Si este comentario se repit
Comentario repetido "/* Este comentario debería
Generando Grafica de Barras
Generando Grafica de Barras
Generando Grafica de Pie
Generando Reporte de Tokens
Generando Reporte de Graficas
Generando Reporte de Errores

REPORTES

Archivo | C:/Users/Userlen/Documents/GitHub/OCL1_P1_2S21_201801155/OLC1_P1/TokensJS.html

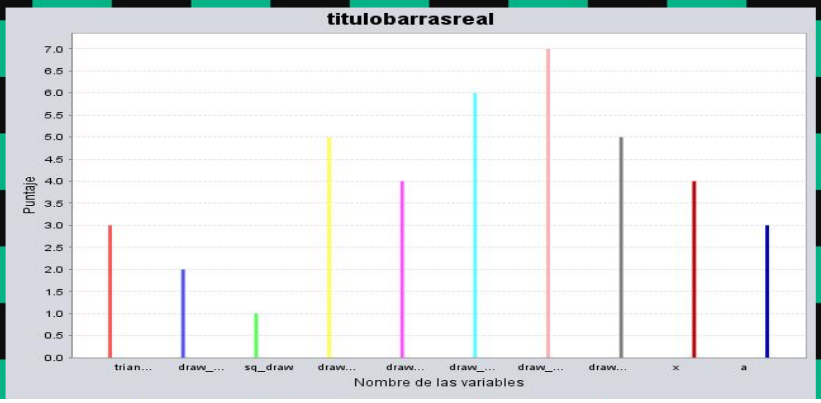
Tokens Archivos JS

Lexema	Token	Line	Columna
ClassJS	class	1	0
identificador	draws	1	6
lla	{	1	12
identificador	tringle_size	3	4
igual	=	3	17
numero	3	3	19
pyc	;	3	20
identificador	square_size	4	4
igual	=	4	16
numero	8	4	18
pyc	;	4	19
identificador	set_size	6	4
para	(	6	12
identificador	tam	6	13

REPORTE ESTADISTICO

RESUMEN

Graficas de Barras



Universidad de San Carlos de Guatemala
Facultad de Ingenieria
Escuela de Ciencias y Sistemas
Reporte de Errores

Organizacion de Lenguajes y Compiladores 1
PROYECTO 1 [FIUSAC Copy Analyzer]
REPORTE DE ERRORES

Tipo	Descripcion	Fila	Columna	Archivo