

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA, INGENIERÍA EN CIENCIAS Y SISTEMAS

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1 SECCIÓN N

MANUAL DE USUARIO

PROYECTO #1

CARLOS JAVIER MARTINEZ POLANCO

CARNÉ: 201709282

05/09/2021

INTRODUCCIÓN

El siguiente programa consiste en una herramienta para detección de copias en los proyectos de programación de los estudiantes de la universidad. Se desarrolló en el lenguaje java, esta aplicación cuenta con una amplia funcionalidad ya que es capaz de reconocer el lenguaje FCA. Este nos permite por medio de comandos determinados realizar la comparación de ambos proyectos y verificar en base a un punteo establecido el porcentaje de copia que hay entre ellos, además de poder realizar graficas de barras, pie o líneas con diferentes configuraciones para poder reportar el porcentaje de copias con más precisión. A continuación, se presenta información detallada sobre el lenguaje FCA.

LENGUAJE FCA

El lenguaje FCA posee una configuración de 6 funciones o comandos principales con los cuales daremos la información necesaria al sistema para que este se encargue de generar las gráficas solicitadas, una de las facilidades que da el lenguaje es que no es sensible a los tamaños de letra, lo cual permite el reconocimiento de instrucciones tanto con minúsculas como con mayúsculas.

COMENTARIOS: Para el desarrollador es muy importante la implementación de los comentarios de una o varias líneas para poder ordenar de mejor manera su código y poder dejar anotaciones de manera que sea mas entendible el desarrollo para otros desarrolladores. Para los comentarios de una línea se utiliza el delimitador **##**, seguido de este delimitador puede venir cualquier carácter hasta que encuentra el salto de línea. Para hacer comentarios multilínea se utiliza el delimitador de apertura **#*** seguido de este puede venir cualquier carácter, incluyendo saltos de línea, hasta que encuentra el delimitador de cierre ***#** para darle fin al comentario. Es importante mencionar que se puede hacer un comentario en cualquier parte del archivo.

INICIO DE LA INSTRUCCIÓN: Para dar inicio al bloque de instrucciones el lenguaje FCA encapsula todas las instrucciones en un bloque principal que llamado con la palabra reservada **GenerarReporteEstadistico**, con esta reservada y encerrando las instrucciones en llaves podemos definir nuestro bloque de código a analizar. Para finalizar las instrucciones internas de cada una de las instrucciones principales se utiliza como delimitador de fin de línea el **;**, que reconocerá el fin de la instrucción.

COMPARACION DE ARCHIVOS: Para comparar los proyectos se utiliza la instrucción que esta definida con la palabra reservada **Compare** seguida de paréntesis que contiene separado por una coma 2 cadenas o identificadores con las rutas de los directorios donde se encuentran los archivos del proyecto, esta instrucción se encargara de realizar los respectivos cálculos de las copias entre los proyectos.

VARIABLES GLOBALES: El lenguaje FCA implementa las variables globales con el fin de ordenar de mejor manera las instrucciones y poder hacerlas no solo mas legibles si mas entendibles para los desarrolladores. Con estas variables podemos asignar un valor que puede ser utilizado en cualquier parte de nuestro archivo. La instrucción que encapsula todas nuestras variables se declara con la palabra reservada **DefinirGlobales**. También es posible la obtención de los punteos y su utilización como variable global.

INSTRUCCIONES PARA LA ELABORACIÓN DE GRÁFICAS: Las instrucciones de las gráficas cuentan con una sintaxis común para cada una de ellas. El patrón que se debe seguir es el mismo para identificar la instrucción por medio de lo que solicitan y el valor que puede venir como decimal, una cadena de texto, una variable, el punteo general o específico, incluso arreglos. Se declara una instrucción de gráfica como: <CARACTERÍSTICA> : <VALOR>. Dentro de las características se pueden utilizar las palabras reservadas:

Título: que recibe como valor una cadena o identificador.

Ejex: que recibe como valor un arreglo que puede estar formado cadenas o identificadores.

Valores: que recibe como valor un arreglo de decimales o identificadores.

Archivo: que recibe como valor una cadena o un identificador.

TítuloX: que recibe como valor una cadena o un identificador.

TítuloY: que recibe como valor una cadena o un identificador.

TIPO DE GRÁFICA: Para definir el tipo de gráfica es necesario el uso de las palabras reservadas: **GraficaBarras**, **GraficaPie**, **GraficaLineas**. Estas instrucciones encapsularán entre llaves las características y valores que conforman la estructura de estas mismas.

PUNTAJE GENERAL: Para obtener el puntaje general se utiliza la siguiente sintaxis.

`\${PuntajeGeneral}`, esto debe ser almacenado en una variable global para utilizarlo.

PUNTAJE ESPECIFICO: Para obtener el puntaje especifico se utiliza la siguiente sintaxis.

`\${PuntajeEspecifico, Nombre del archivo, nombre del objetivo del análisis, nombre de la clase}`

Este puede ser utilizado como valor en las graficas y se puede declarar en las propiedades.

EJEMPLO DE IMPLEMENTACIÓN FCA:

##ARCHIVO DE PRUEBA FCA

GenerarReporteEstadistico{

Compare("C:\\Users\\CarlosMartinez\\Desktop\\P1_OLC1_201709282\\archivosPrueba\\Prueba1.js","C:\\Users\\CarlosMartinez\\Desktop\\P1_OLC1_201709282\\archivosPrueba\\Prueba2.js");

DefinirGlobales{

string TituloGbarra = "GRAFICA DE BARRAS";

string TituloGpie = "GRAFICA DE PIE";

string TituloGlineas = "GRAFICA DE LINEAS";

string var1 = "Clase2";

double var2 = \${PuntajeEspecifico, "archivo1.js", "clases", "clasespuntos"};

double val1 = 3.3;

double val2 = 9;

double val3 = 1.9;

}

GraficaBarras{

Titulo: TituloGbarra;

Ejex: ["Clase1",var1,"Clase3"];

TituloX: "VARIABLEX";

TituloY: "VARIABLEY";

Valores: [3.5, 5, var2];

}

ESTE ES UN COMENTARIO MULTILINEA

PUEDA VENIR EN CUALQUIER PARTE DEL ARCHIVO *#

GraficaPie{

Titulo: TituloGpie;

Ejex: ["Clase1",var1,"Clase3"];

Valores: [val1, 5,\${PuntajeEspecifico, "archivo1.js", "clases", "clasespuntos"}];

}

GraficaLineas{

Titulo: TituloGlineas;

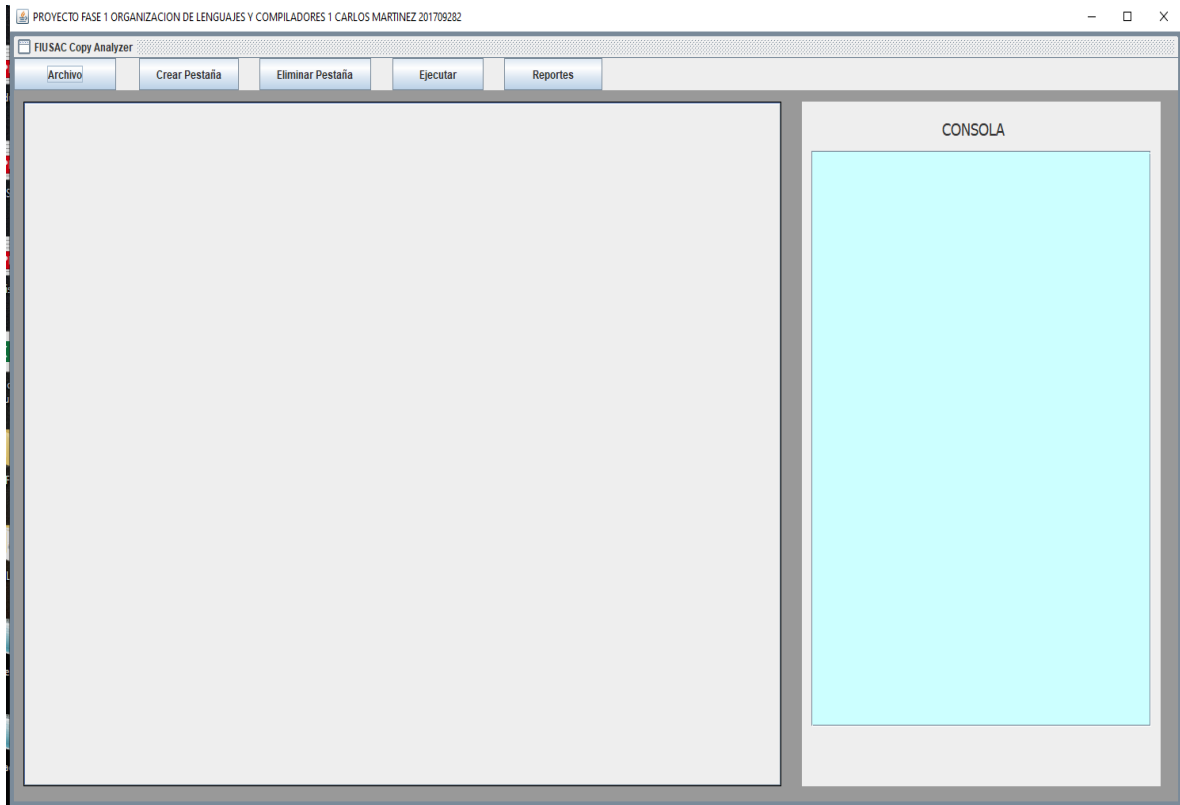
Archivo: "archivo1";

}

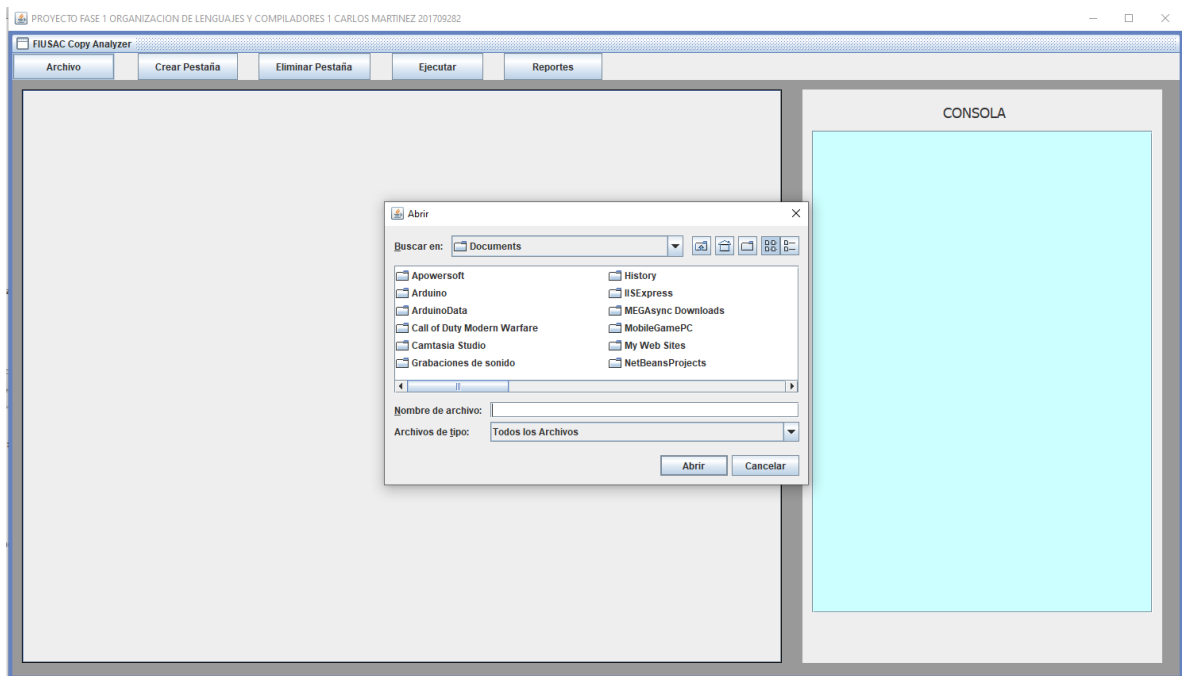
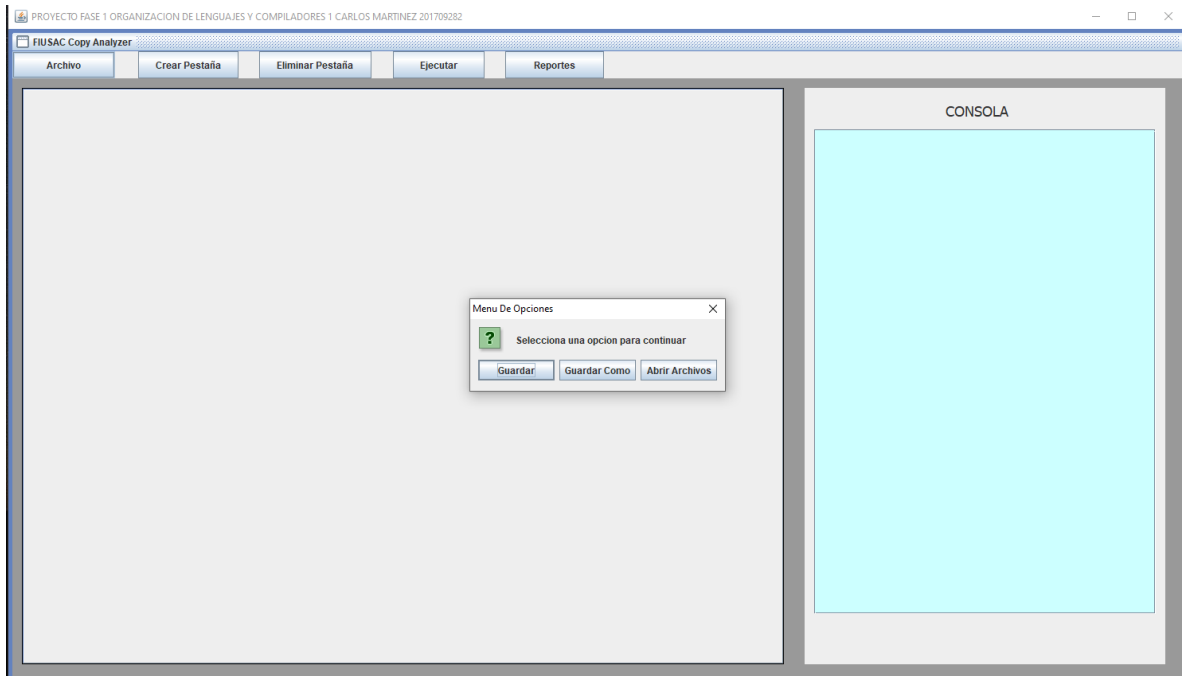
}

INTERFAZ GRÁFICA

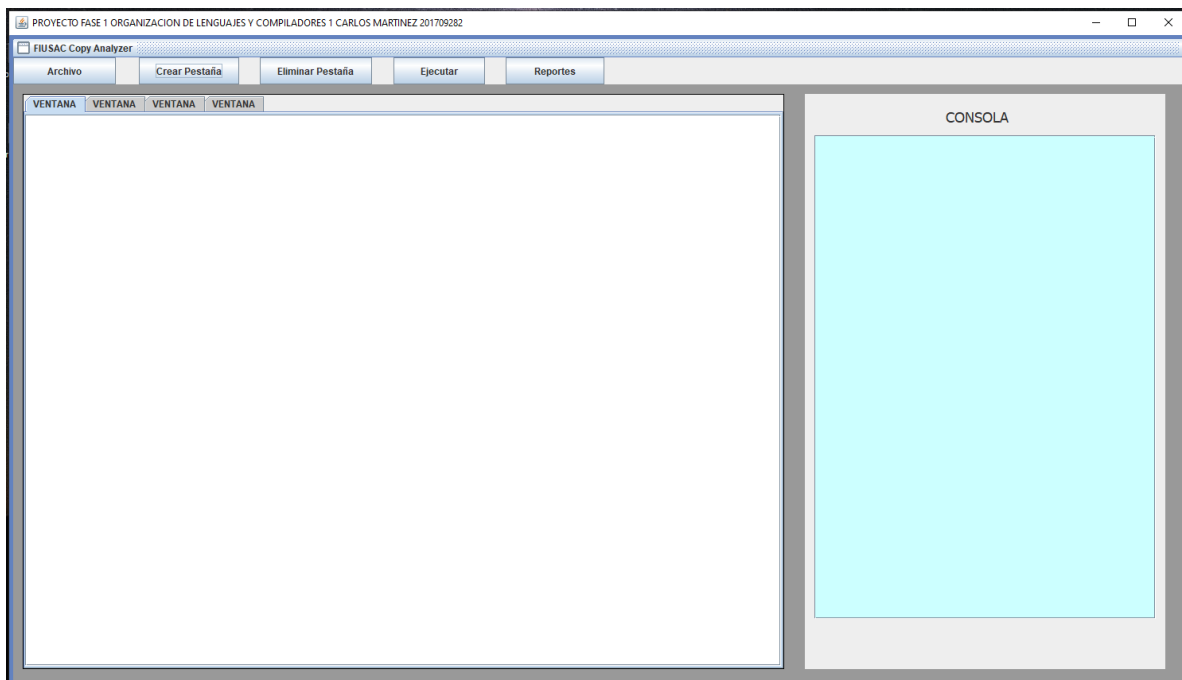
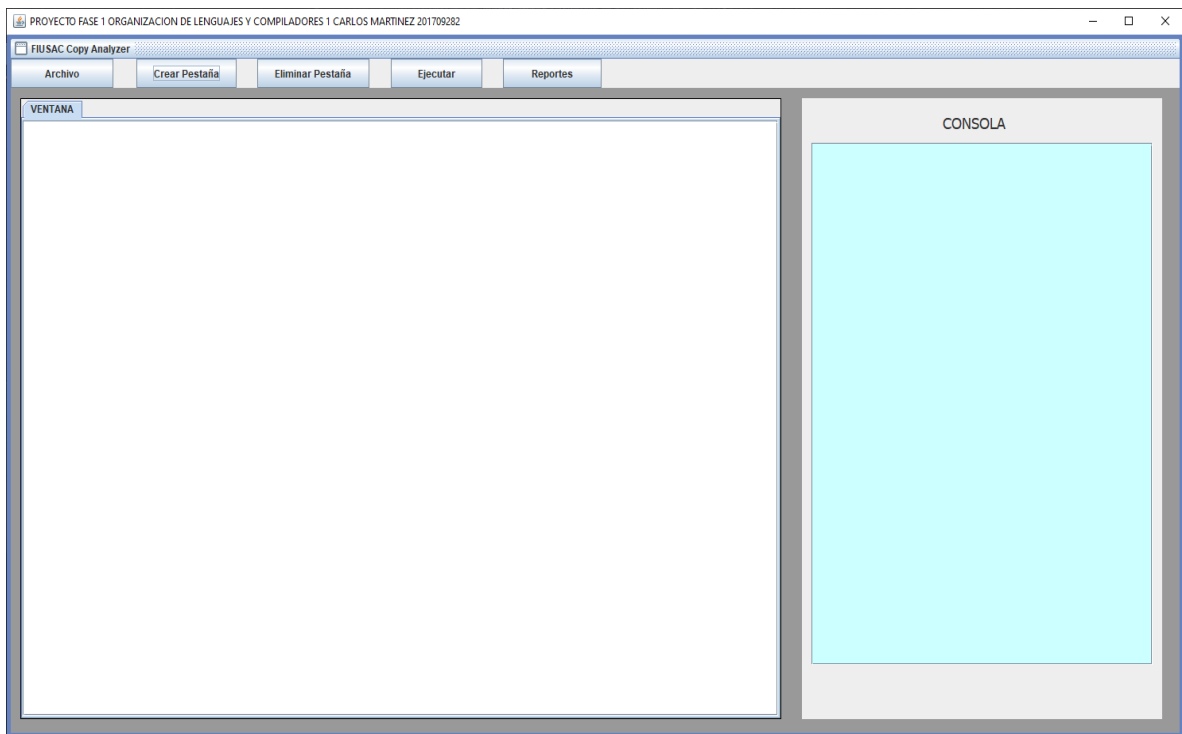
INICIO DEL SISTEMA: Se presenta la interfaz de diseño clásico de un editor de texto.



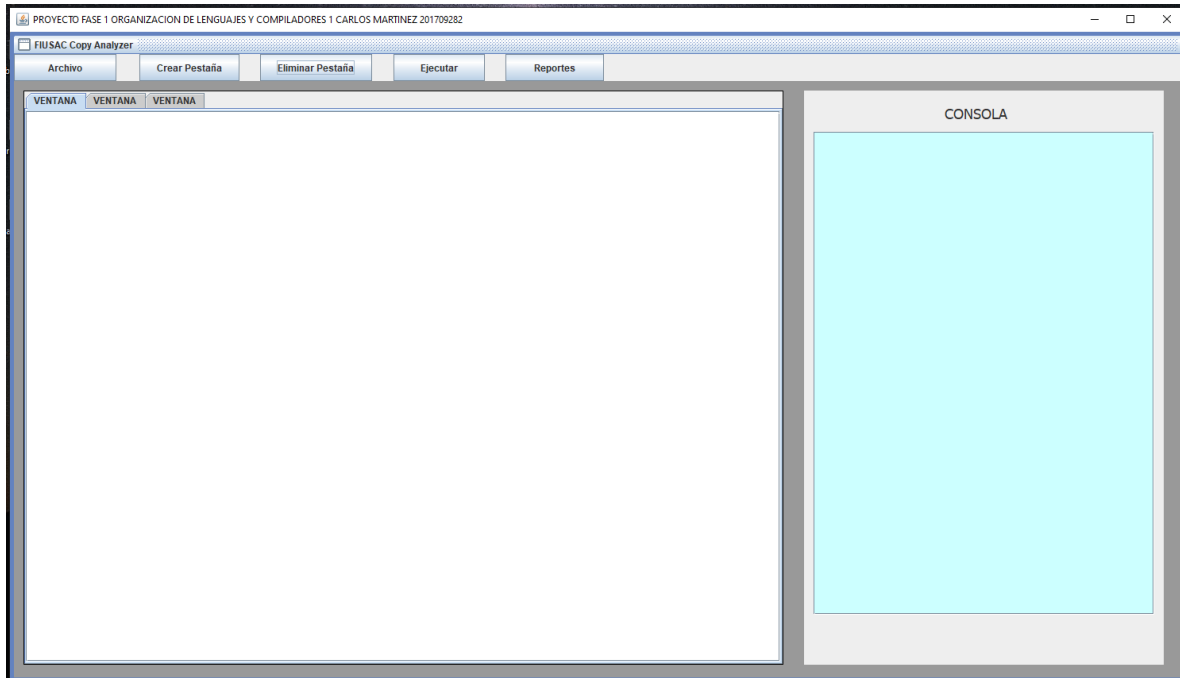
ARCHIVO: Despliega un input option pane para poder elegir entre Abrir un archivo, guardar como o guardar en la carpeta principal. Para guardar como y abrir archivos se utiliza el file chooser y para el de guardar solo se solicita el nombre del archivo por medio de un input dialog, para esta ultima opción la pestaña debe de estar creada.



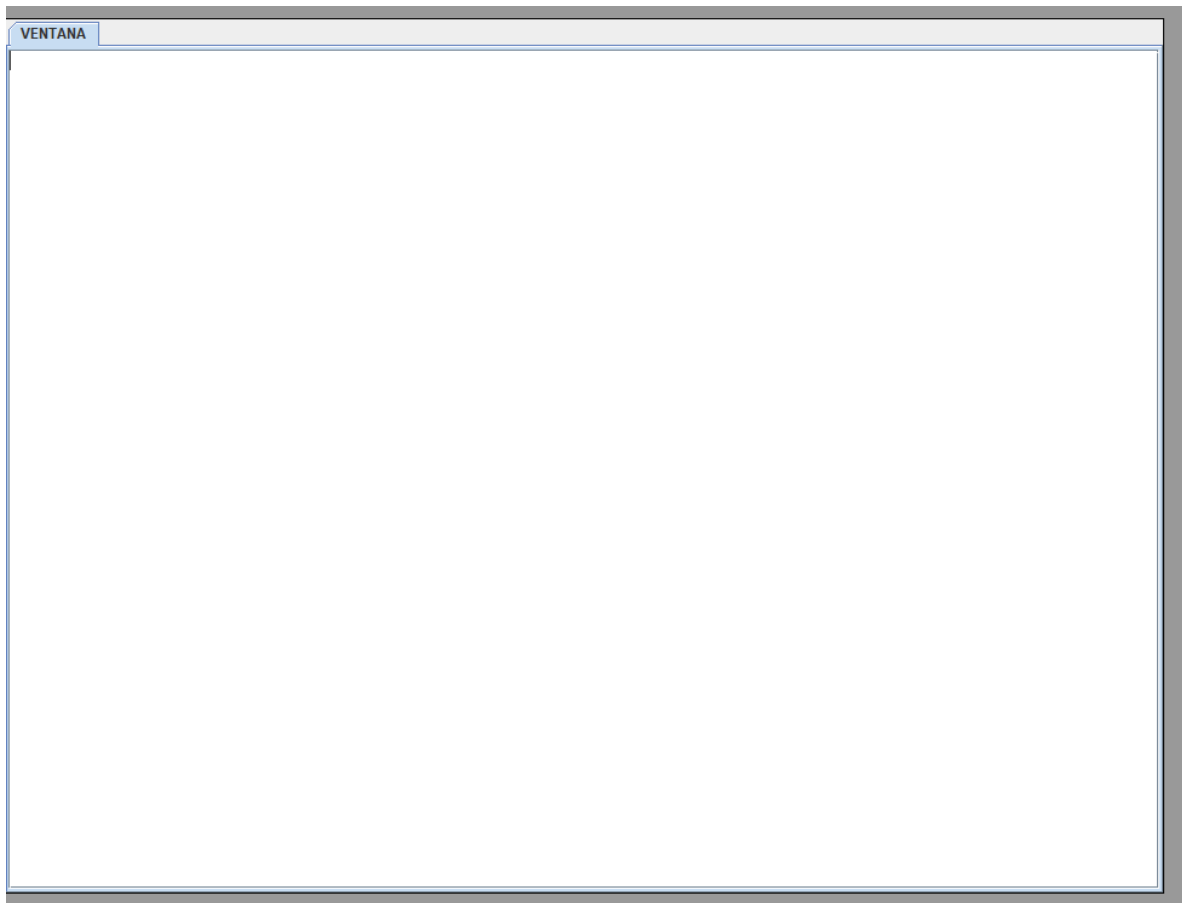
CREAR PESTAÑAS: Para poder crear un archivo primero se debe iniciar una nueva pestaña, se puede agregar la cantidad de pestañas que desee.



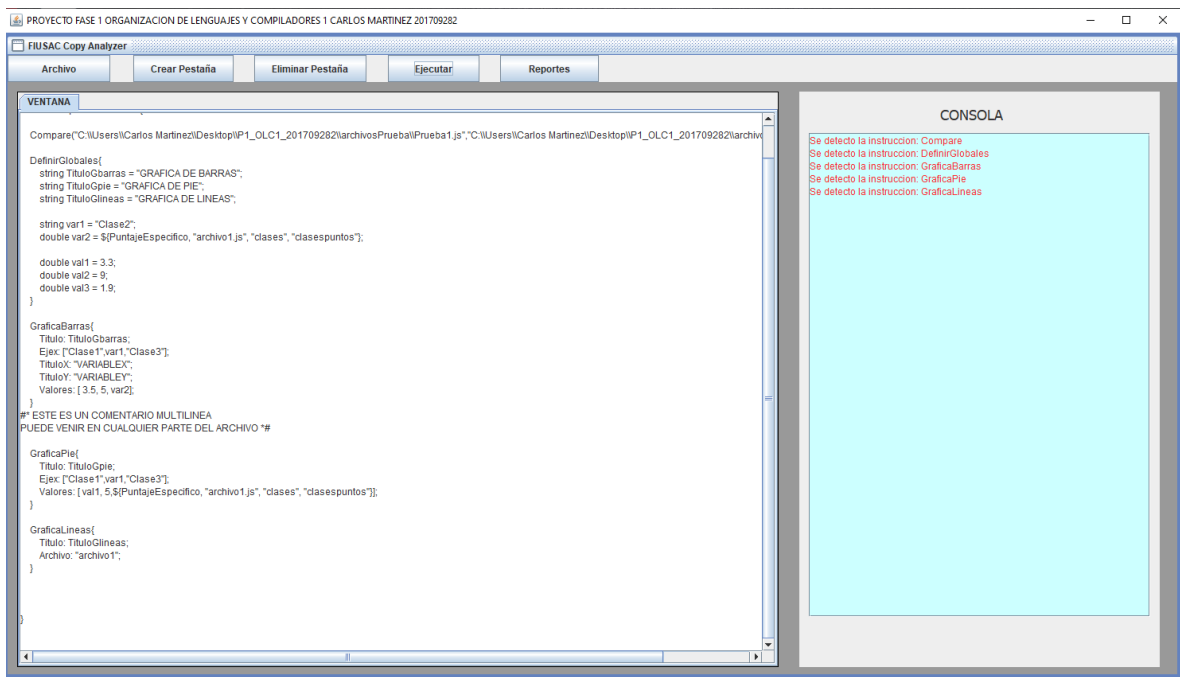
ELIMINAR PESTAÑAS: Con el botón de eliminar pestaña es posible realizar dicha acción, el documento no se guardará automáticamente antes de cerrarlo, este quedará tal y como se guardó por última vez.



EDITOR DE TEXTO: El editor de texto será la entrada principal del sistema para poder ejecutar los comandos, estas son cada una de las pestañas agregadas. Estas se trabajan de forma independiente como que cada una de ellas fuese un archivo.



EJECUTAR: El botón de ejecutar realiza el análisis del texto que se encuentre en la pestaña seleccionada en el editor.



CONSOLA: En la consola podemos ver cualquier información que el analizador nos proporcione referente a la ejecución del código que hay en el editor de texto.



ANALIZADOR

El analizador FCA implementa un analizador de la sintaxis de javascript, lenguaje en el cual estarán escritos los archivos a evaluar en los directorios de los proyectos. Con este analizador se obtienen los puntajes generales y específicos del análisis según lo que solicite el evaluador.