

Manual de Usuario

Copy Analyzer es un software utilizado para comparar el código JavaScript entre dos proyectos. Para ejecutar instrucciones, Copy Analyzer utiliza un lenguaje propio en el cual se le puede decir al software realizar distintas tareas. El programa puede ejecutar varios archivos con extensión fca en donde hay escrito código del lenguaje de Copy Analyzer. La interfaz gráfica permite abrir y cerrar pestañas tanto como el usuario quiera, además de seleccionar el directorio en donde se quiere guardar el archivo fca. Además, Copy Analyzer puede generar distintos tipos de reportes dependiendo de los resultados del análisis de copias.

Reporte de Errores

El reporte de errores crea un listado de todos los errores encontrados, tanto en el archivo fca como en los archivos JavaScript analizados en los proyectos a comparar.

Reporte de Tokens

El reporte de Tokens crea un listado con todos los tokens encontrados, junto con el lexema, línea y columna, tanto en el archivo fca como en los archivos JavaScript analizados en la comparación de proyectos

Reporte Estadístico

El reporte estadístico crea una tabla de todos los archivos JavaScript comparados en donde se muestra la cantidad de clases, métodos, variables y comentarios encontrados en cada archivo. Además se muestran todas las graficas creadas en el archivo fca.

Reporte Json

El reporte json crea un archivo json con información relacionada con todos los puntajes de los archivos JavaScript comparados. Se muestra el valor del puntaje general de la comparación y un arreglo de puntajes específicos de cada clase, método, variable y comentario encontrados por el analizador

Lenguaje de Reporteria

▼ GenerarReporteEstadistico

Este método se utiliza para escribir todas las instrucciones a realizar por el analizador. Dentro de este método se generan las graficas, se declaran las variables y se indica las rutas a comparar.

```
GenerarReporteEstadistico{  
    // Aqui dentro van todas las instrucciones  
}
```

▼ Compare

Este método se utiliza para indicar las rutas de los dos proyectos que se pretenden comparar.

```
GenerarReporteEstadistico{  
    compare("ruta1", "ruta2"); // ruta1 y ruta2 son las rutas  
    definirGlobales{
```

▼ DefinirGlobales

Este método se utiliza para definir variables que se utilizaran en el proyecto. Se pueden definir dos tipos de variables: string para cadenas de texto y double para

números. Además se pueden asignar puntajes generales y específicos a las variables.

```
definirglobales{
    string reporteResumen = "Reporte de Archivo file_1.js de

    ## variables para Reportes de Barras

    Double pr1 = ${PuntajeEspecifico,"file1.js","variable","t
    Double pr2 = ${PuntajeEspecifico,"file1.js","variable","d
    Double pr3 = ${PuntajeEspecifico,"file1.js","variable","s
    Double pr4 = ${PuntajeEspecifico,"file1.js","variable","d
    Double pr5 = ${PuntajeEspecifico,"file1.js","variable","d
    Double pr6 = ${PuntajeEspecifico,"file1.js","variable","d
    Double pr9 = ${PuntajeEspecifico,"file1.js","variable","d
    Double pr12 = ${PuntajeEspecifico,"file1.js","variable","
    Double pr15 = ${PuntajeEspecifico,"file1.js","variable","
    Double pr16 = ${PuntajeEspecifico,"file1.js","variable","
    String s1 = "hanoi_generator";
    String s2 = "fact_generator";|
    Double pe1 = 0;
    Double pe2 = 1;
    Double pe3 = 0;
    Double pe4 = 1;
    Double pe5 = 1;
    Double pe6 = 1;
```

▼ GraficaBarras

Este método se utiliza para generar una gráfica de barras en donde se le declaran el titulo, titulo del eje x, titulo del eje y, valores para el eje x y valores para el eje y.

```
graficaBarras{
  Valores: [4, 5, 6, 7];
  TituloX: "Eso tilinX";
  TituloY: "eso tiliny";
  EjeX: ["ejex1", nom, "ejex3", variable];
  Titulo: "El titulo de la grafica 2";
}
```

▼ GraficaPie

Este método se utiliza para generar una gráfica pie en donde se le debe de especificar el titulo, los valores y el eje x.

```
GraficaPie {
  Titulo: "EL PIE";
  EjeX: ["valorPie1", nombrePie1, "valorPIE3"];
  Valores: [1.4, 2.4, 3.1];
}
```

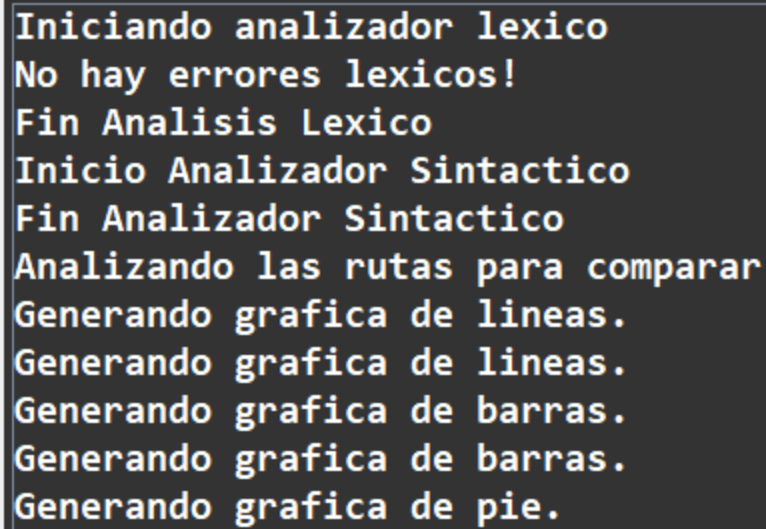
▼ GraficaLineas

Este método se utiliza para generar una gráfica de líneas en donde solo se debe especificar el titulo de la grafica y el nombre del archivo al que se le desea hacer la grafica de líneas.

```
GraficaLineas {
  Titulo: nombre2Lineas;
  Archivo: "file1.js";
}
```

Consola

La consola sirve para saber qué es lo que se esta ejecutando al momento y ayuda a dar una retroalimentación al usuario para saber si el analizador falló en algún lugar o si todo se ejecutó con éxito.

A screenshot of a terminal window with a dark background and a light blue title bar. The terminal displays the following text in a monospaced font:

```
Iniciando analizador lexico
No hay errores lexicos!
Fin Analisis Lexico
Inicio Analizador Sintactico
Fin Analizador Sintactico
Analizando las rutas para comparar
Generando grafica de lineas.
Generando grafica de lineas.
Generando grafica de barras.
Generando grafica de barras.
Generando grafica de pie.
```