

MANUAL DE USUARIO

Este manual tiene como finalidad dar a conocer los conceptos básicos, características, funcionalidades y fundamentos básicos para el uso de la aplicación “Solo Análisis”. Solo Análisis es una aplicación que consiste en un analizador léxico – sintáctico que permite la entrada de un archivo de texto con cierta estructura para ser analizado y generar una lista de tokens, esta lista de tokens posteriormente es enviada a un analizador sintáctico que se encarga de verificar que los tokens de entrada tengan la sintaxis correcta, si todo esto se cumple posteriormente procede a diagramar el archivo de entrada bajo las especificaciones de UML.

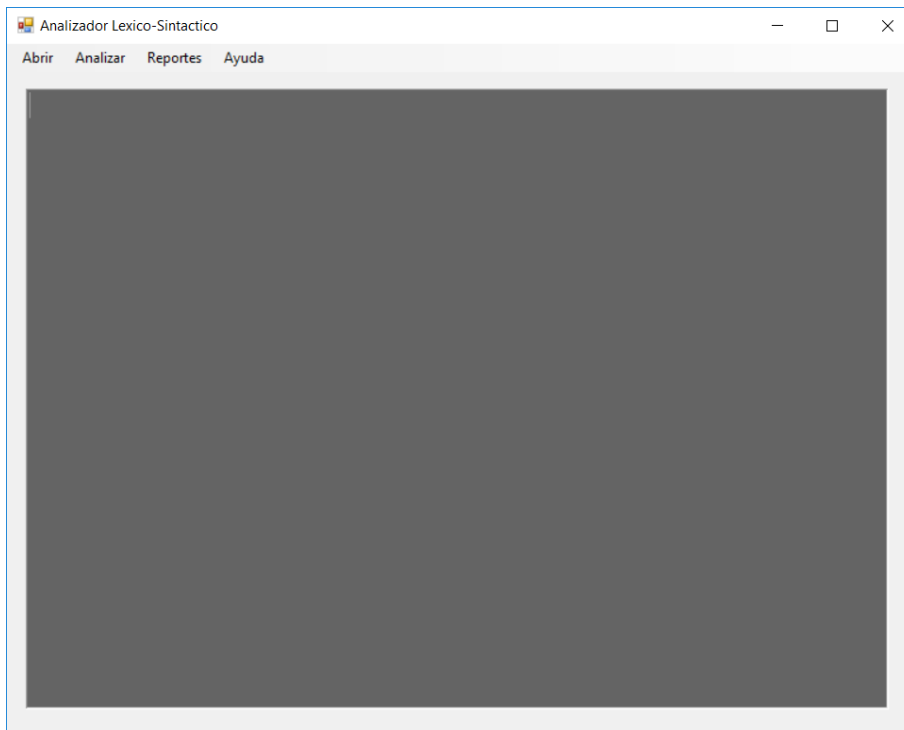
Requerimientos mínimos

- Windows 7 o superior
- Tener instalado Graphviz para la generación del diagrama

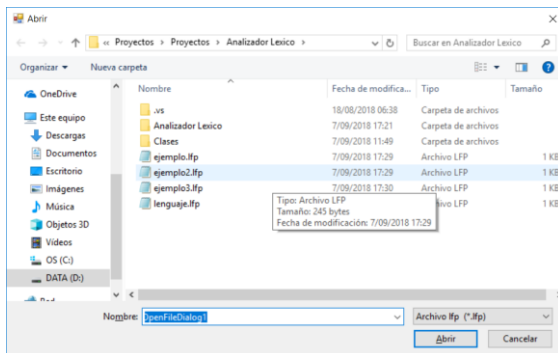
Guía / Tutorial

Al iniciar el programa se muestra una ventana que contiene un menú en la parte superior y un panel de fondo negro, este panel será el editor de texto, donde podremos abrir los archivos o escribir cierto código a analizar.

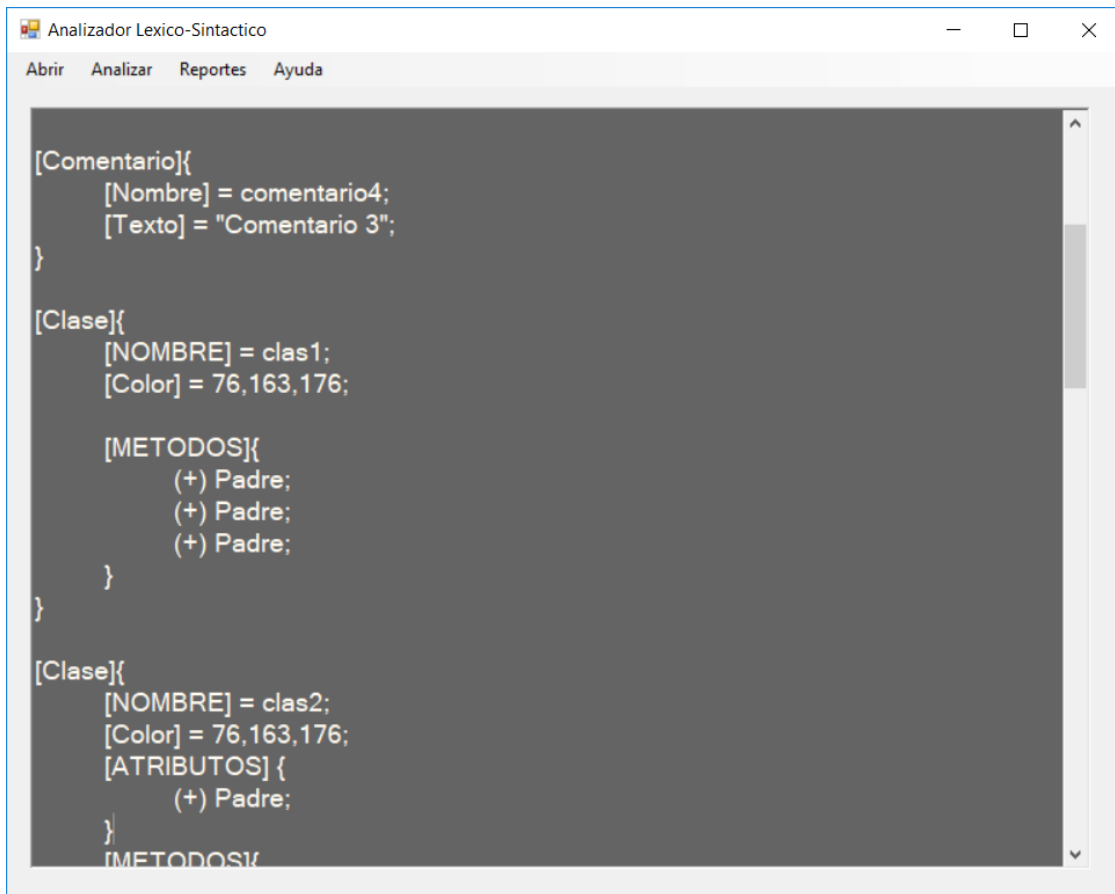
El menú consta de 4 items, subdivididos cada uno. En el menú Archivo tenemos las opciones de Abrir un archivo, Guardar archivo nuevo, Guardar archivo previamente abierto y Salir de la aplicación. La pestaña “Analizar” analiza el texto ingresado en el editor, es importante realizar esta operación antes de crear Reportes o Diagramar. En la pestaña Reportes podemos crear Reporte de Tokens de no existir error o Diagramar las clases ingresadas. En el menú Ayuda se encuentra el “Manual Técnico” y este manual, además el nombre del creador de la aplicación.



Podemos escribir directamente en el editor o cargar un archivo ya creado a través del menú Archivo – Abrir.



Elegimos un archivo previamente guardado con terminación “lfp” y lo abrimos.



```
[Comentario]{
  [Nombre] = comentario4;
  [Texto] = "Comentario 3";
}

[Clase]{
  [NOMBRE] = clas1;
  [Color] = 76,163,176;

  [METODOS]{
    (+) Padre;
    (+) Padre;
    (+) Padre;
  }
}

[Clase]{
  [NOMBRE] = clas2;
  [Color] = 76,163,176;
  [ATRIBUTOS] {
    (+) Padre;
  }
  [METODOS]{
```

Ya abierto o cuando ya tengamos texto escrito en el editor procedemos a analizarlo. El análisis consta de varias fases, la primera es verificar que todos los atributos de entrada pertenezcan al lenguaje y realizar la lista de entradas. Posteriormente se ejecuta el analizador sintáctico y al finalizar se pinta de colores el archivo de entrada de acuerdo con la siguiente tabla:

Palabras Reservadas	Amarillo
Corchetes, llaves, paréntesis	Rojo
Puntos, dos puntos, punto y coma	Negro
Visibilidades	Verde
Cadenas y “	Morado
Errores	Azul

Quedando el archivo de entrada de la siguiente manera:

```

error:Herencia:clas1;
clas6:composicion:clas5;
clas5:AsociacionSimple:clas3;
clas5:comentario4;
}

{Comentario{
    {Nombre} = comentario4;
    {Texto} = "Comentario 3";
}

{Clase{
    {NOMBRE} = clas1;
    {Color} = 76,163,176;

    {METODOS{
        (+) Padre;
        (+) Padre;
        (+) Padre;
    }
}

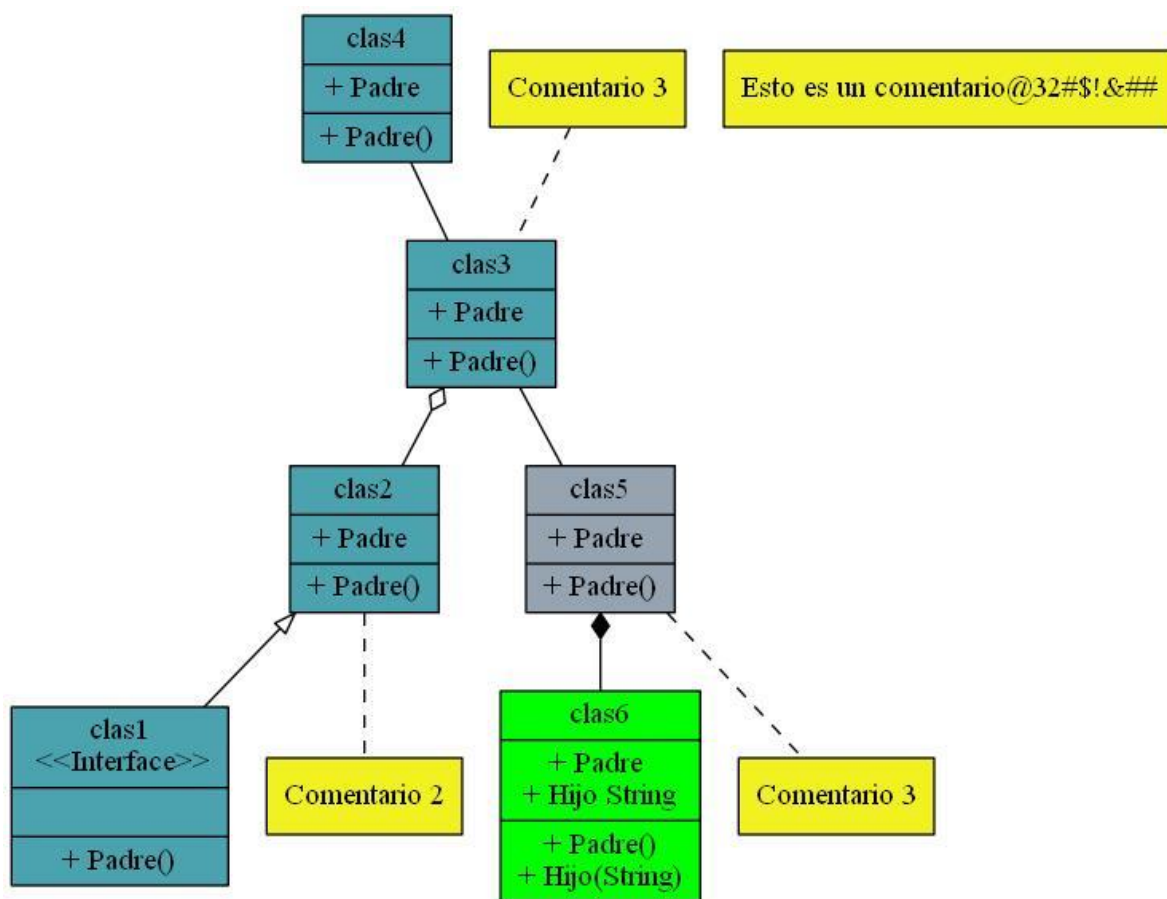
{Clase}

```

Ejemplo Diagrama de Tokens:

No	Lexema	Tipo	Columna	Fila	Token
1	[[0	1	3
2	Asociacion	Asociacion	1	1	5000
3]]	11	1	4
4	{	{	12	1	1
5	clas1	Identificador	1	2	200
6	:	:	6	2	8
7	Herencia	Herencia	7	2	6000
8	:	:	15	2	8
9	clas2	Identificador	16	2	200
10	:	:	21	2	7
11	clas2	Identificador	1	3	200
12	:	:	6	3	8
13	Agregacion	Agregacion	7	3	7000
14	:	:	17	3	8
15	clas3	Identificador	18	3	200
16	:	:	23	3	7
17	clas3	Identificador	1	4	200
18	:	:	6	4	8
19	AsociacionSimple	AsociacionSimple	7	4	9000
20	:	:	23	4	8
21	clas4	Identificador	24	4	200
22	:	:	29	4	7
23	comentario3	Identificador	1	5	200
24	:	:	12	5	8
25	clas3	Identificador	13	5	200
26	:	:	18	5	7
27	clas2	Identificador	1	6	200
28	:	:	6	6	8
29	comentario2	Identificador	7	6	200

Ejemplo Digrama de Clases:



ESTRUCTURA DEL ARCHIVO DE ENTRADA

1. BLOQUE CLASE

En el bloque clase es necesario definir el nombre, que será el atributo que definirá que clase es. En el sub-bloque color se define el color que será pintada la clase, bajo 3 números que representan RGB, estos tres siempre deben venir. El sub-bloque Atributos define los atributos de la clase, este bloque puede venir o no. El sub-bloque Metodos define los métodos que representan a la clase y este bloque siempre debe venir. El orden en que estos deben venir no importa, se pueden ordenar de forma arbitraria. Este bloque puede venir las veces que se deseen crear nuevas clases.

```
[Clase]{  
    [Nombre] = nombreclase ;  
    [Color] = num1,num2,num3;  
    [Atributos]{  
        (visibilidad) identificador: tipoRetorno;  
        (visibilidad) identificador;  
    }  
    [Metodos]{  
        (visibilidad) identificador: tipoRetorno;  
        (visibilidad) identificador;  
    }  
}
```

2. BLOQUE COMENTARIO

El bloque comentario únicamente consta de su nombre que puede ser cualquier identificador y de Texto, texto es la parte donde viene lo que se desea comentar, cabe resaltar que en este puede venir cualquier símbolo de entrada sin restricciones del lenguaje. Este bloque puede venir las veces que se deseen crear nuevos comentarios. Solo se permite un comentario por bloque.

```
[Comentario]{  
    [Nombre] = identificador;  
    [Texto] = "Aqui va el comentario";  
}
```

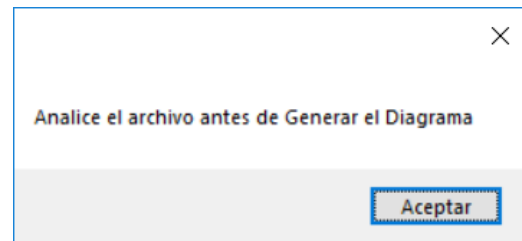
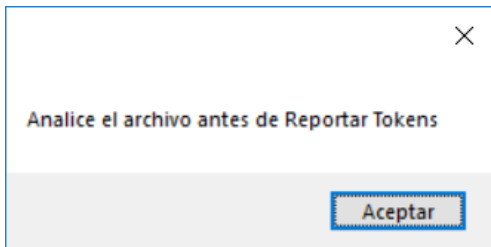
3. BLOQUE ASOCIACION

En el bloque asociación se pueden definir dos tipos de asociación, de clase a clase y de clase a comentario. Bajo la estructura de abajo. Este bloque únicamente puede venir una sola vez y aquí se definen todas las asociaciones deseadas.

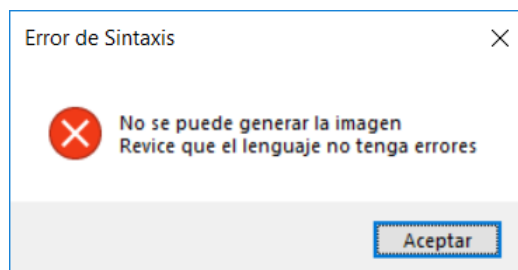
```
[Asociacion]{  
    clase1:Herencia:clase2;  
    clase1:comentario1;  
    comentario1:clase1;  
}
```

POSIBLES ERRORES

- Intentar crear un Reporte de Tokens o generar el diagrama de clases sin haber analizado el texto previamente.



- Diagramar un conjunto de clases con errores léxicos dentro del código, cree un reporte de Tokens y le mostrará automáticamente el Reporte de Errores, especificando donde se encuentra el error.



Corrija el código y vuelva a ejecutar el análisis, si no se encuentra ningún error el programa abrirá el diagrama.

- Ejemplo de un reporte de Errores

No	Error	Columna	Fila	Tipo
1	@	21	4	Lexico
2	@	0	6	Lexico
3	Se esperaba identificador	1	3	Sintactico