Universidad de San Carlos de Guatemala

Ingeniería en Ciencias y Sistemas

Lenguajes formales de programación

PROYECTO FINAL

MANUAL DE USUARIO: ANALISIS SINTACTICO

Cristian Giovanni Estrada Ramirez

Carnet 202006413

30 de abril del 2023

OBJETIVO:

El objetivo del software es proporcionar una herramienta de análisis y visualización de archivos de entrada. En primer lugar, el software analizará el archivo de entrada y verificará si cumple con el modelo estándar indicado en el proyecto. Se realizará una verificación de errores de sintaxis para asegurarse de que la estructura del archivo sea coherente y cumpla con los requisitos del modelo estándar. En caso de encontrar algún error, el software informará al usuario y proporcionará detalles sobre los errores detectados. Si no se detectan errores, el software procederá a procesar los datos del archivo de entrada.

Una vez que los datos del archivo de entrada se hayan verificado y validado, el software generará una visualización de las operaciones incluidas en el archivo. Esto puede ser en forma de gráficos, diagramas, tablas o cualquier otro formato visual que sea relevante para los datos procesados. La visualización permitirá al usuario entender mejor los datos incluidos en el archivo y analizarlos de manera más efectiva.

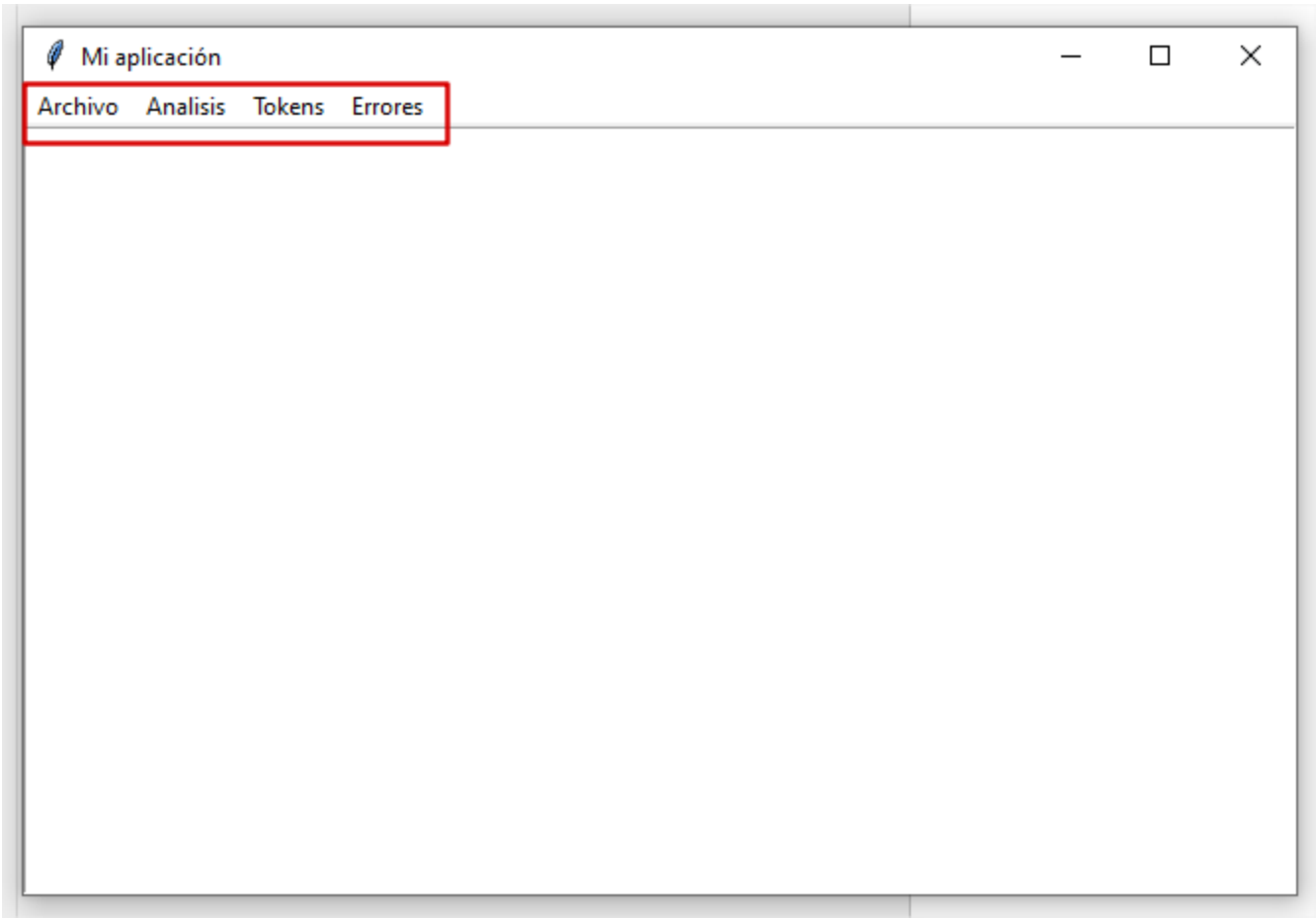
ESPECIFICACIONES TECNICAS:

Las especificaciones técnicas mediante las que fue elaborado el sistema son las siguientes:

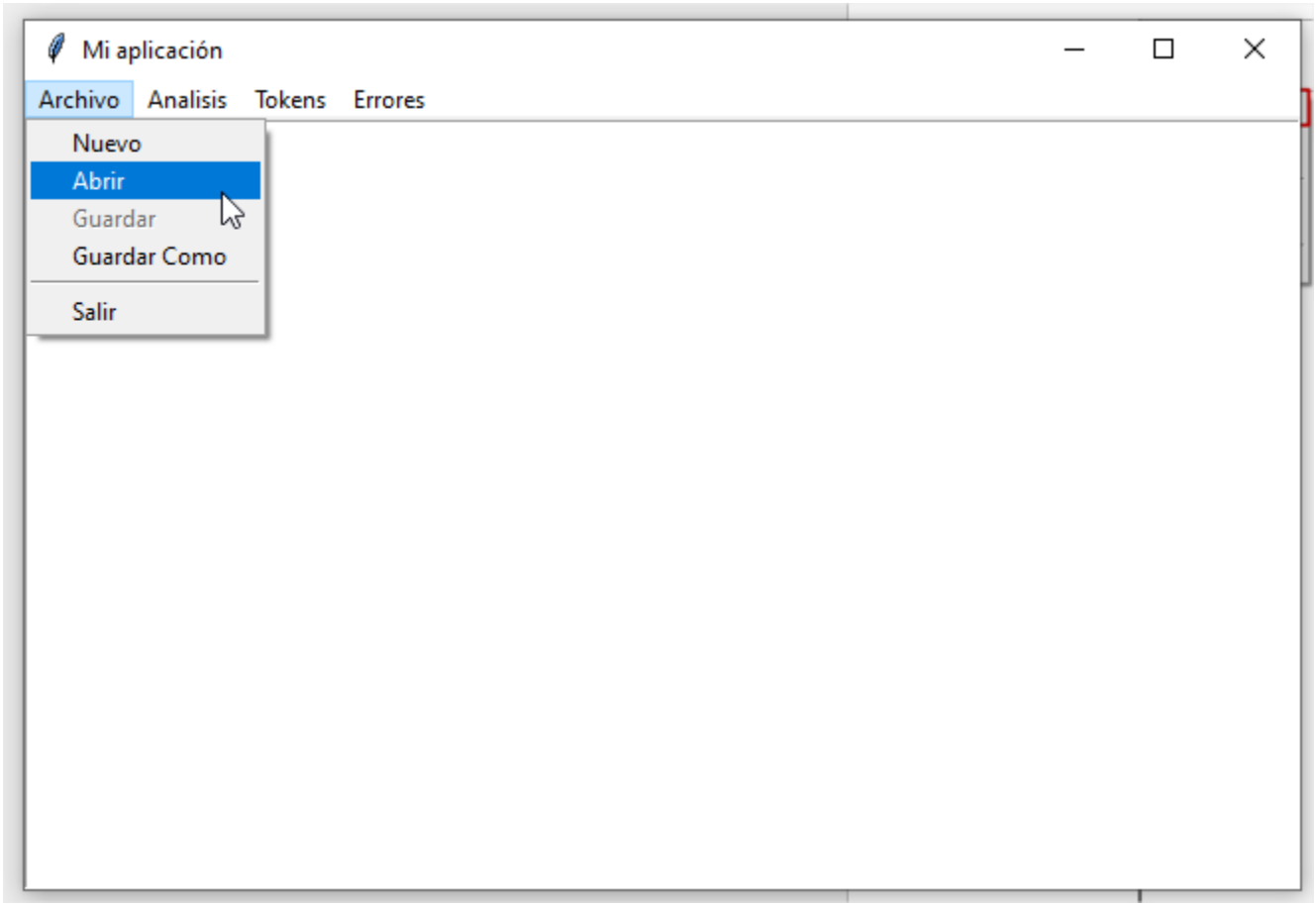
- Hardware:
  - Procesador Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz 3.00 GHz
  - 8 GB de Ram
  - No se requirió el uso de ningún tipo de recursos gráficos extra
- Software:
  - Visual Studio Code
  - Sistema operativo: Windows 10 Pro
  - Sistema operativo de 64 bits

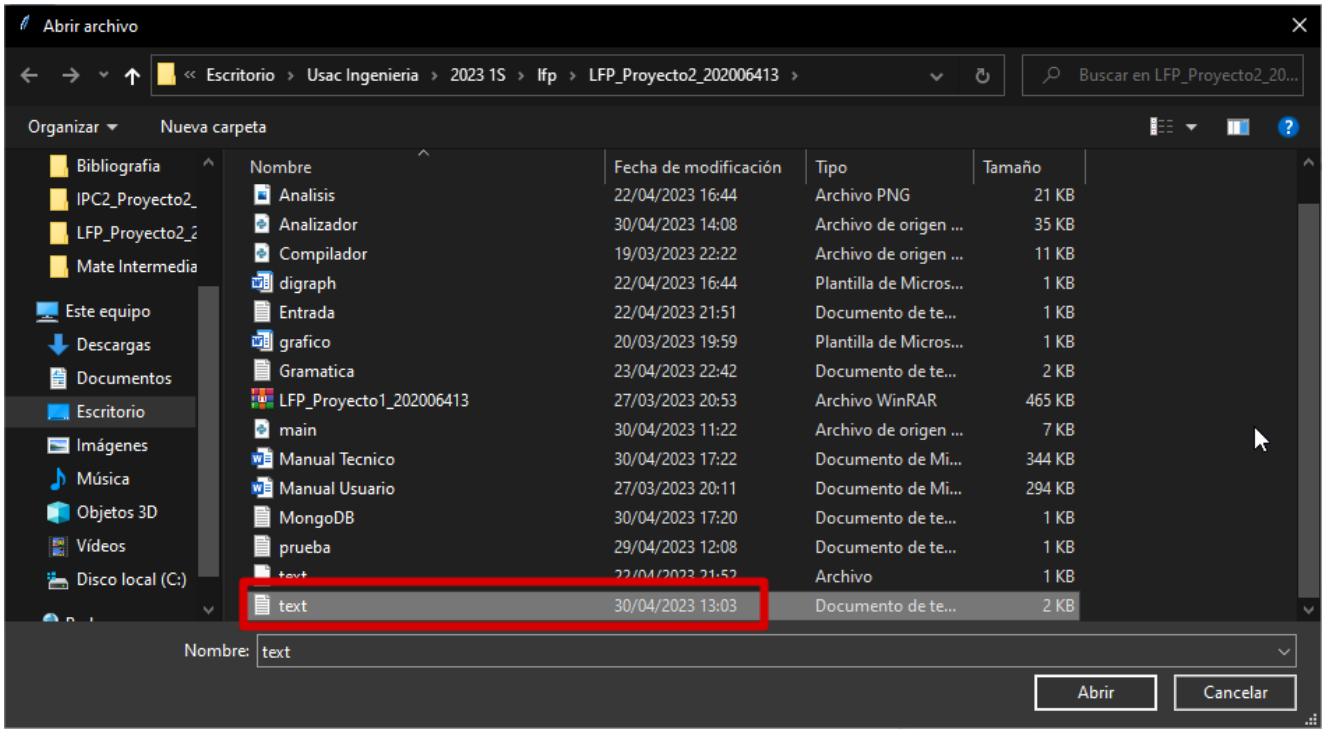
FLUJO DEL PROGRAMA:

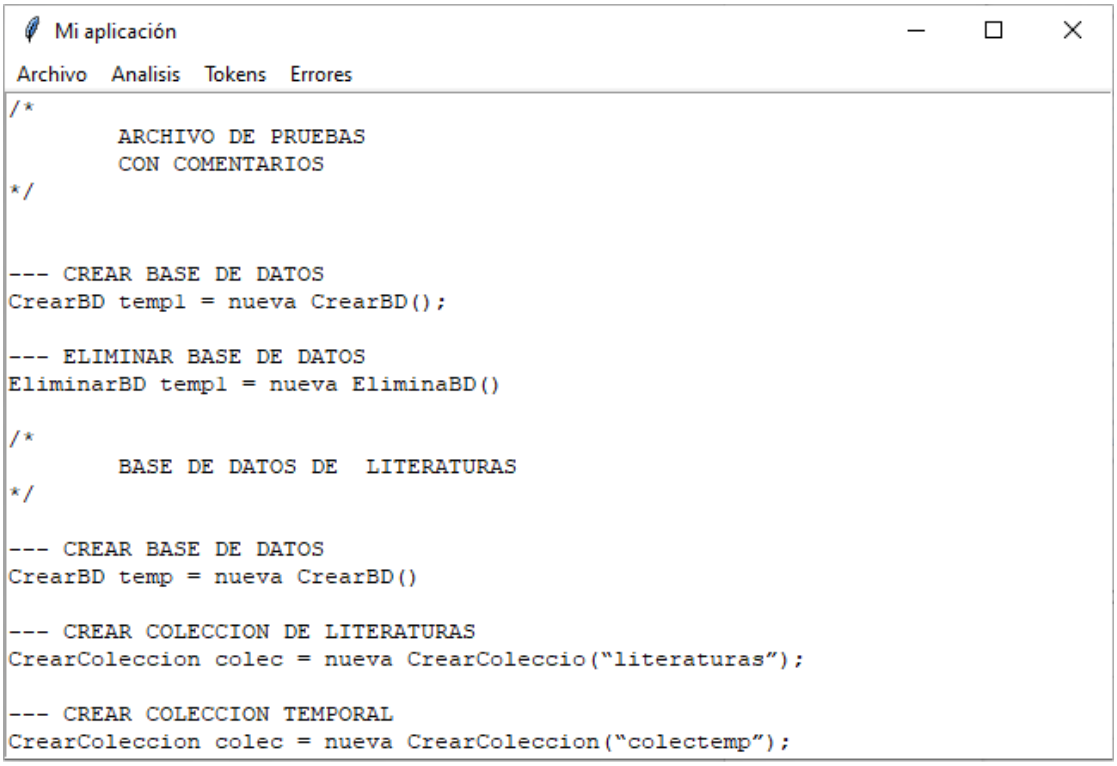
Al iniciar el sistema abre una ventana en la cual se podrá editar texto, además tendremos las pestañas de Archivo, Análisis, Tokens y Errores con las cuales realizaremos todas las acciones:



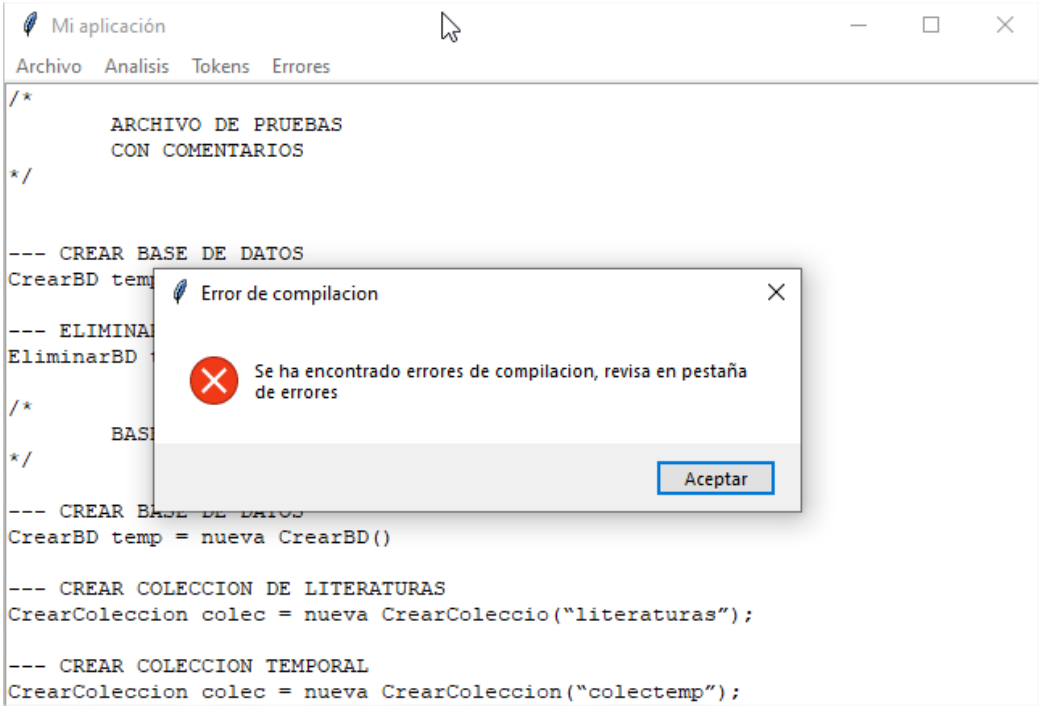
Seleccionaremos las opciones que nos ofrece “Archivo”, las cuales son Nuevo, Abrir, Guardar, Guardar Como y Salir, en este caso seleccionaremos la opción de abrir lo cual desplegará el explorador de archivos donde vamos a buscar el archivo de entrada:



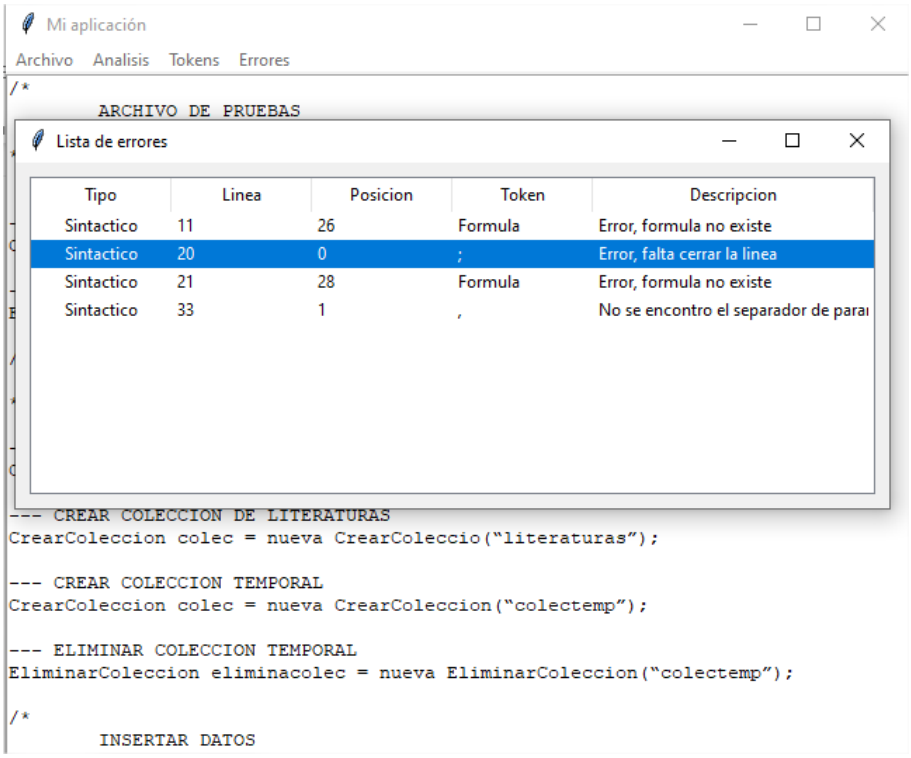




El archivo seleccionado se podrá en pantalla para que el usuario pueda hacer las modificaciones necesarias, en este caso no haremos ninguna modificación, lo que haremos es acceder a la pestaña de Análisis en la cual realizaremos un análisis sobre el código impreso en pantalla:



Como observamos el programa nos genera una alerta, la cual nos indica que nuestro código tiene errores los cuales deben ser solucionados para poder generar nuestras formulas de mongoBD, para poder ver los errores de compilación tendremos que ir a la pestaña de Errores y ver que nos indica:

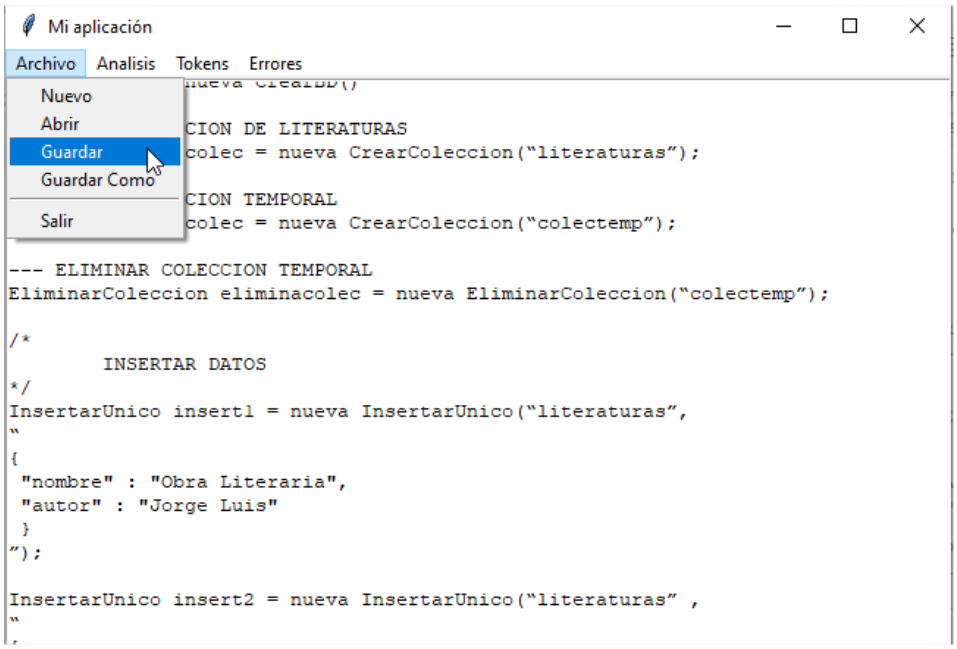


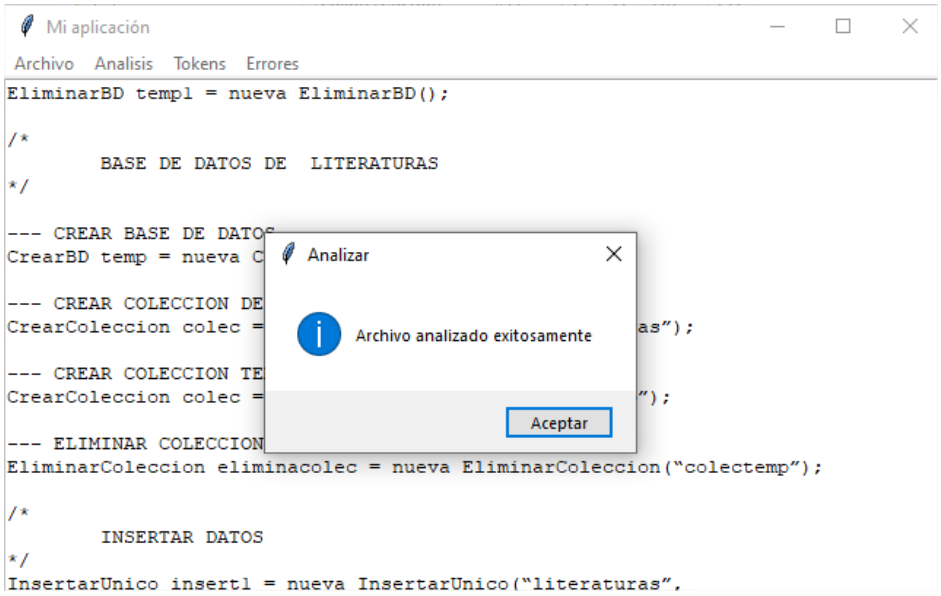
Ahora realizaremos los cambios que nos están afectando la compilación, luego guardaremos los cambios y por ultimo volveremos a analizar:

```
--- CREAR BASE DE DATOS
CrearBD temp = nueva CrearBD();

--- CREAR COLECCION DE LITERATURAS
CrearColeccion colec = nueva CrearColeccion("literaturas");

--- CREAR COLECCION TEMPORAL
```

```
--- ELIMINAR BASE DE DATOS
EliminarBD temp1 = nueva EliminarBD();

/*
        BASE DE DATOS DE  LITERATURAS
*/

--- CREAR BASE DE DATOS
CrearBD temp = nueva CrearBD()

--- CREAR COLECCION DE LITERATURAS
CrearColeccion colec = nueva CrearColeccion "literaturas");
```

```
/*
        INSERTAR DATOS
*/
InsertarUnico insert1 = nueva InsertarUnico("literaturas",
"
{
 "nombre" : "Obra Literaria",
 "autor" : "Jorge Luis"
 }
");
```

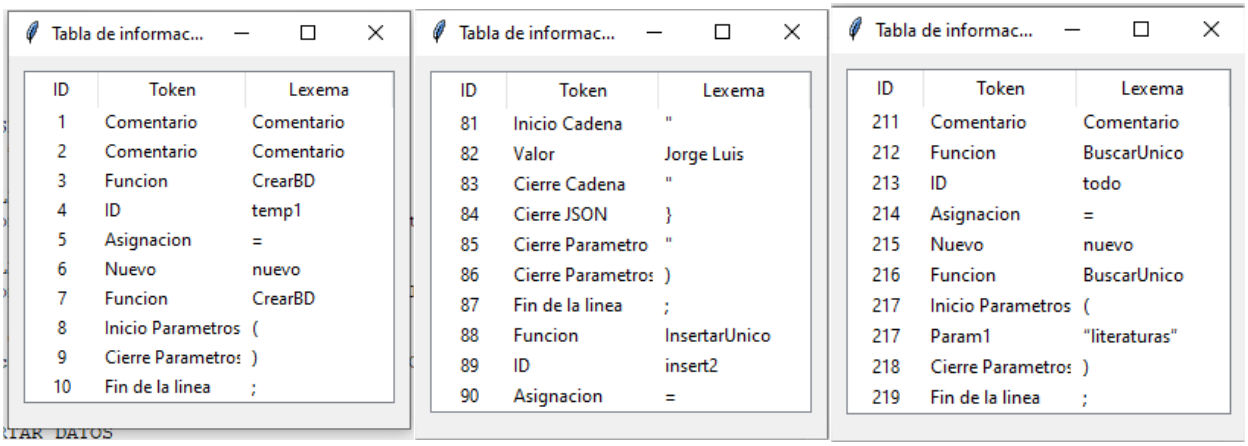




Una vez analizado el programa generará un archivo el cual incluye todas las funciones insertadas en el archivo de entrada pero con el formato especifico para el manejo de bases de datos de MongoDB:

```
use('temp1');

temp1.dropDatabase();

use('temp');

dbo.createCollection(�literaturas�);

dbo.createCollection(�colectemp�);

dbo.colectemp.drop();

dbo.literaturas.InsertOne("
 {
     "nombre" : "Obra Literaria",
     "autor" : "Jorge Luis"
 }
");

dbo.literaturas.InsertOne("
 {
     "nombre" : "El Principito",
     "autor" : "Antoine de Saint"
 }
");

dbo.literaturas.InsertOne("
 {
     "nombre" : "Moldavita. Un Visitante Amigable",
     "autor" : "Norma Mu�oz Ledo"
 }
");

dbo.literaturas.InsertOne("
 {
     "nombre" : "Obra Literaria"
 }
");

dbo.literaturas.Find();

dbo.literaturas.InsertOne();

```

Por ultimo revisaremos que se hayan registrado los tokens en el programa, esto nos servirá para ver que pasos utilizó el programa para poder leer la estructura del archivo de entrada:



Para finalizar revisaremos en la pestaña de errores que ahora se encuentra limpia, eso debido a que los errores que fueron encontrados fueron solucionados: