MANUAL DE USUARIO

Analizador Léxico y Sintáctico Application:

Contenido

I.Introducción

---

1. Objetivo

Brindar asistencia al usuario de este programa informático (Analizador Léxico y Sintáctico App), describiendo las opciones o el funcionamiento del proceso que se muestra en cada una de las pantallas conforme al usuario vaya avanzando en dicho programa.

---

2. Requerimiento

- Sistema Operativo: Windows 7 o superior
- Procesador mínimo Intel Pentium (800MHz Intel Pentium)
- Mínimo 1GB en RAM
- IDE Visual Studio Code, o compatible con Python
- Exploradores: Internet Explorer 9 y superior

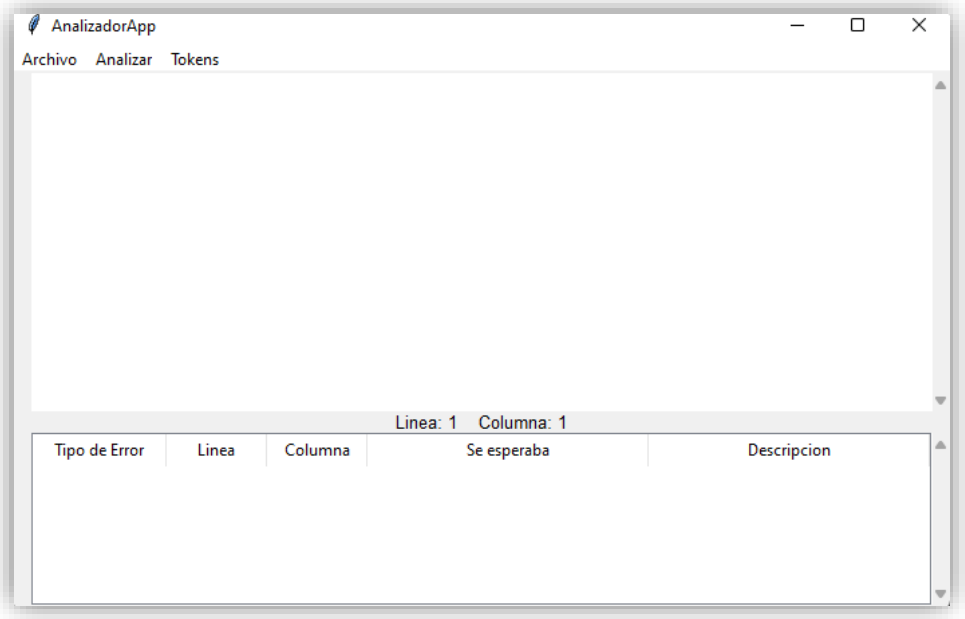
II.Opciones del Sistema

El presente manual está organizado de acuerdo con en el menú de opciones de la siguiente manera:

1. Archivo
2. Analizar
3. Tokens

Inicio

Al momento de ejecutar el programa se mostrará como pantalla principal la siguiente ventana:



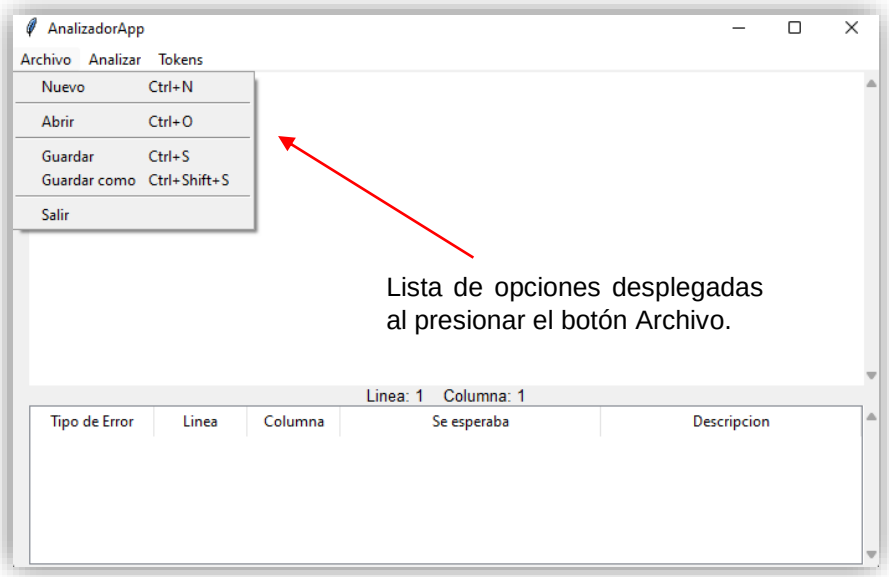
Esta ventana contiene un menú en donde se encuentran tres botones los cuales representan las diferentes acciones que el usuario podrá realizar mientras el programa este ejecutándose

---

1. **Archivo**

   Al presionar este botón se desplegará una lista con todas las opciones que el usuario puede elegir.

   Cada opción es representada por un botón el cual el usuario puede presionar para realizar una acción.



1.1. Nuevo

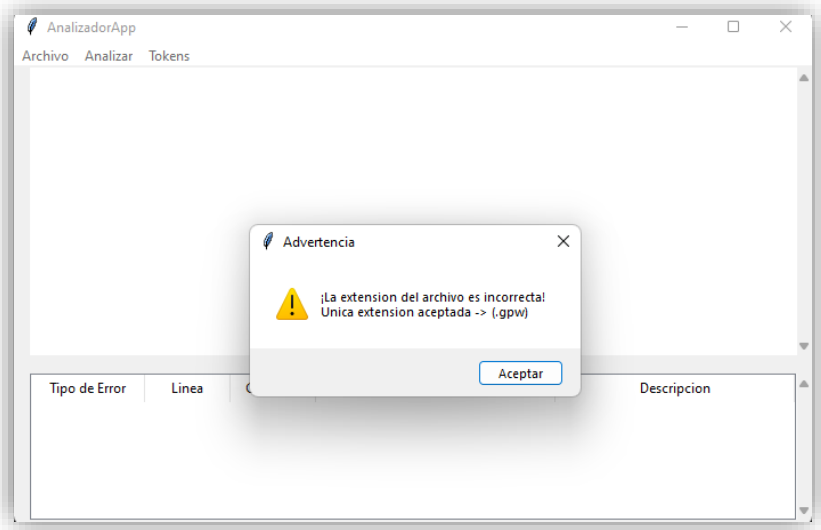
Al presionar el botón “Nuevo” el programa limpiar el área de texto, esto si se encontrase texto dentro del área, de ser así preguntara antes de limpiar si desea guardar los cambios que se realizaron en dicho archivo.

1.2. Abrir

Al hacer clic sobre el botón “Abrir” este mostrara el explorador de archivos para que el usuario seleccione el archivo que desee ejecutar y/o editar.

El programa soporta únicamente archivos con la extensión .gpw y pueden ser abiertos varios archivos en la misma ejecución.

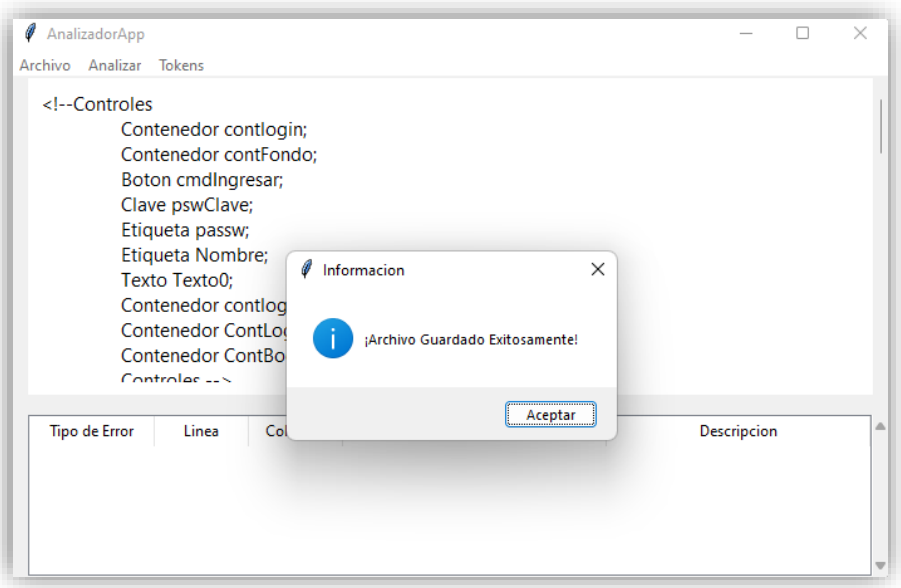
Si se intenta abrir un archivo con la extensión incorrecta el programa mostrará un mensaje de advertencia en pantalla y no leerá el archivo seleccionado.



1.3. Guardar

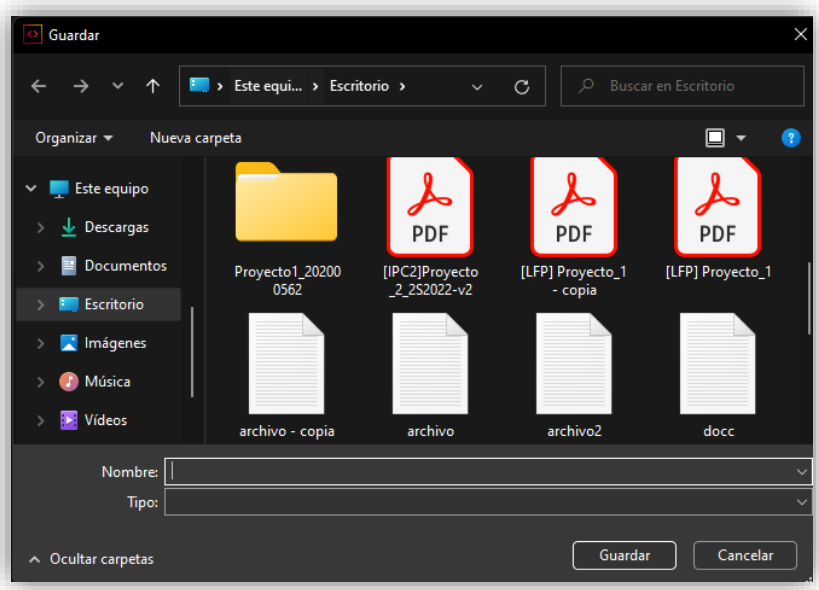
Al presionar el botón “Guardar” el programa guardara el contenido ingresado en el área de texto.

Si existiese un archivo abierto el contenido del área de texto será almacenado en dicho archivo, de no ser así el contenido será almacenado en un archivo nuevo si el usuario así lo desea.



1.4. Guardar Como

Al momento en que el usuario presione el botón “Guarda como” la aplicación mostrara en pantalla el explorador de archivos para que el usuario selección la ubicación en la que desea almacenar el archivo e ingrese el nombre que le desea poner.

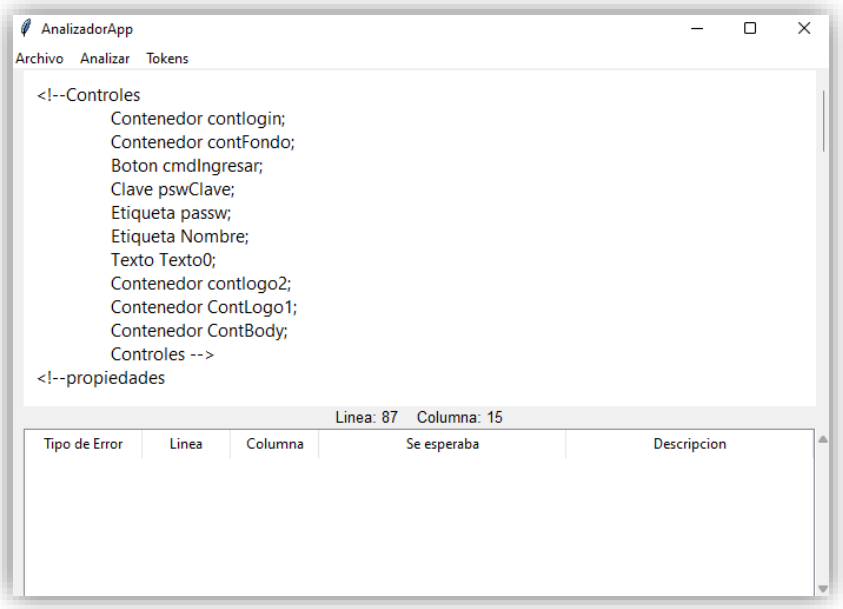


1.5. Salir

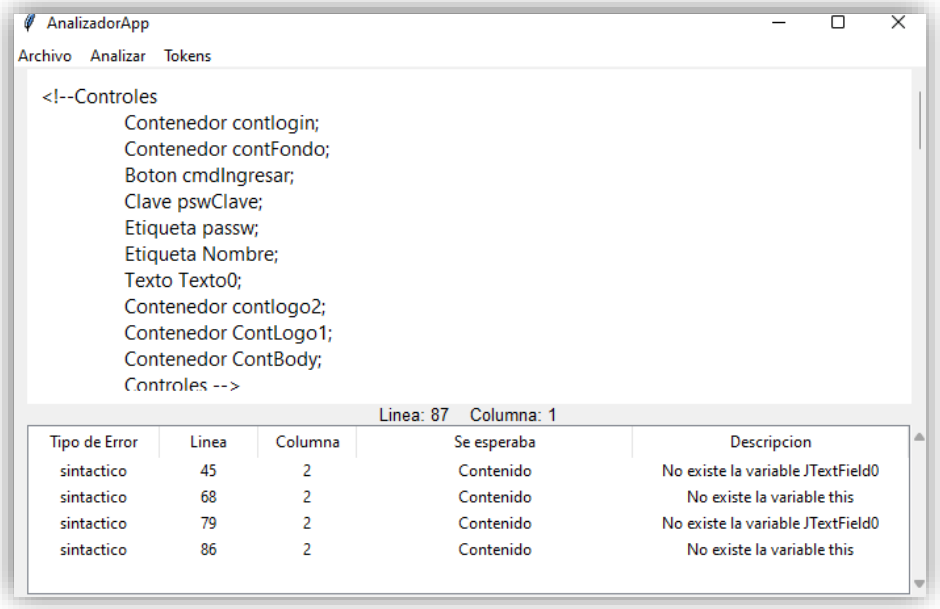
Al presionar el botón “Salir” el programa verifica que el contenido en el área de texto se encuentre almacenado en un archivo, de ser así el programa finaliza con su ejecución.

2. Analizar

Al presionar este botón el programa analizara léxicamente y sintácticamente todo el contenido que se encuentre dentro el área de texto.

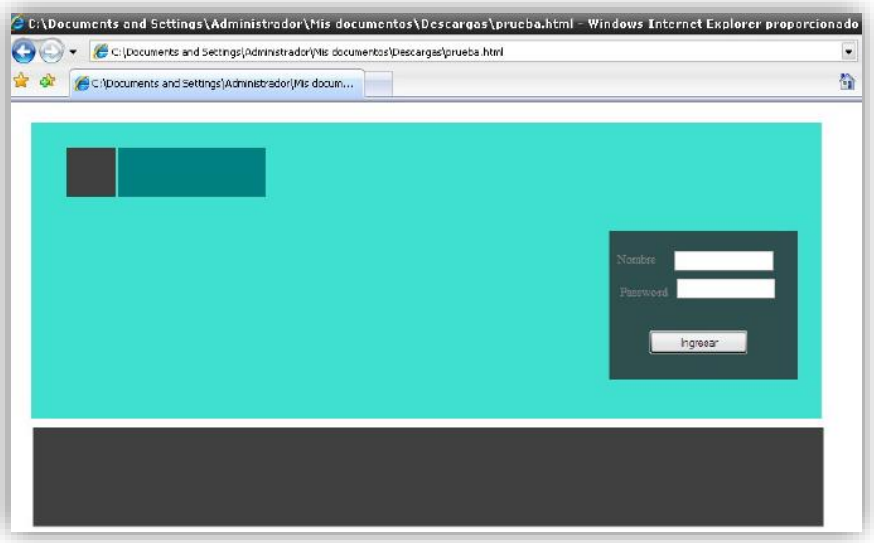


Si al analizar se encuentran errores, el programa mostrara los errores encontrados en al archivo en la tabla de errores que se encuentra en la superficie inferior de la interfaz gráfica del programa.



En caso contrario que al analizar el contenido del área de texto no se encontrasen errores, el programa creara dos archivos finales, los cuales uno es de extensión .css el cual contendrá el diseño, y el otro es de extensión .html el cual contendrá el contenido.

Esto dos archivos mencionados anteriormente componen la página web creada finalmente con todas las instrucciones ingresadas en el área de texto.



En esta imagen se puede observar el resultado final que se obtuvo al leer cada una de las instrucciones contenidas en el área de texto.

---

3. Tokens

Al presionar este botón se mostrará una tabla con todos los tokens que el programa reconoció al momento de analizar el contenido léxicamente.

TABLA DE TOKENS

No.	Lexema/Token	Tipo	Columna	Linea
1	<	Menor que	1	1
2	Tipo	token reservado Tipo	5	1
3	>	Mayor que	6	1
4	<	Menor que	1	2
5	Operacion	token reservado Operacion	10	2
6	=	Igual	11	2
7	SUMA	token reservado SUMA	15	2
8	>	Mayor que	16	2

En esta imagen se puede observar los tokens que fueron reconocidos por el programa al momento de analizar el contenido del área de texto.

III. Archivo

En esta sección se llevará a cabo cual es la estructura adecuada que un archivo de entrada debe tener.

---

1. Estructura de Archivo

Es importante conocer cuál es la estructura correcta que debe tener un archivo de entrada para que el programa no tenga ningún problema al realizar al leer las instrucciones de dicho archivo.

Si el archivo de entrada no contiene las etiquetas necesarias puede que el programa lance un error o simplemente ejecutara todas las etiquetas que encuentre, pero puede que los resultados que se obtengan no sean los correctos o más bien que la página web se muestre incompleta en el navegador.

En la siguiente imagen se puede observar la estructura de un archivo de entrada correcta.

```
<!--Controles
      Contenedor contlogin;
      Contenedor contFondo;
      Boton cmdIngresar;
      Clave pswClave;
      Etiqueta passw;
      Etiqueta Nombre;
      Texto Texto0;
      Contenedor contlogo2;
      Contenedor ContLogo1;
      Contenedor ContBody;
      Controles -->
<!--propiedades
/*
Definicion de propiedades
      */
      //#$inicio de contlogin
      contlogin.setAncho(190);
      contlogin.setAlto(150);
      contlogin.setColorFondo(47,79,79);
      //#$fin de contlogin
      //#$inicio de contFondo
      contFondo.setAncho(800);
      contFondo.setAlto(100);
      contFondo.setColorFondo(64,64,64);
      //#$fin de contFondo
      //#$inicio de cmdIngresar
      cmdIngresar.setTexto("Ingresar");
      //#$fin de cmdIngresar
      //#$inicio de pswClave pswClave.setTexto("");
      //#$fin de pswClave
      //#$inicio de etiqueta passw
      passw.setAncho(53);
      passw.setAlto(13 );
      passw.setColorLetra(128,128,128);
      passw.setTexto("Password");
      //#$fin de passw
      //#$inicio de Nombre
      Nombre.setAncho(44);
      Nombre.setAlto(13);
      Nombre.setColorLetra(128,128,128);
      Nombre.setTexto("Nombre");
      //#$fin de Nombre
      //#$inicio de JTextField0
      JTextField0.setTexto("");
      //#$fin de JTextField0
```

```
        //#$inicio de contlogo2
        contlogo2.setAncho(150);
        contlogo2.setAlto( 50);
        contlogo2.setColorFondo(0,128,128);
        //#$fin de contlogo2
        //#$inicio de ContLogo1
        ContLogo1.setAncho(50);
        ContLogo1.setAlto(50);
        ContLogo1.setColorFondo(64,64,64);
        //#$fin de ContLogo1
        //#$inicio de ContBody
        ContBody.setAncho(800);
        ContBody.setAlto(300);
        ContBody.setColorFondo(64,224,208);
        //#$fin de ContBody
        propiedades -->
<!--Colocacion
        /*
        Posicionamiento de los controles
        */
        contFondo.setPosicion(25,330);
        this.add(contFondo);
        contlogin.setPosicion(586,110);
        ContBody.add(contlogin);
        passw.setPosicion(11,54);
        contlogin.add(passw);
        cmdIngresar.setPosicion(40,100);
        contlogin.add(cmdIngresar);
        pswClave.setPosicion(67,48);
        contlogin.add(pswClave);
        Nombre.setPosicion(8,21);
        contlogin.add(Nombre);
        JTextField0.setPosicion(65,20);
        contlogin.add(JTextField0);
        contlogo2.setPosicion(88,25);
        ContBody.add(contlogo2);
        ContLogo1.setPosicion(36,25);
        ContBody.add(ContLogo1);
        ContBody.setPosicion(23,21);
        this.add(ContBody);
Colocacion -->
```

El resultado obtenido del archivo ingresado en la imagen anterior es el siguiente: