

**Universidad de San Carlos de Guatemala**  
**Facultad de Ingeniería**  
**Escuela de Ciencias y Sistemas**  
**Lenguajes Formales y de Programación**



**MANUAL TÉCNICO**

**Nombre: Christopher Alexander Acajabon Gudiel**  
**Carnet: 201404278**

## **Métodos:**

- **Public Sub** metodo\_analizar()

En este método se encuentra la lectura de los componentes cargados o escritos que se introduzcan para luego operarlos.

- **Public Sub** metodoErrores()

Este método va agregando los tokens a la tabla.

- **Public Sub** metodoErrores2()

Método encargado del manejo de errores de caracteres no aceptados otorgando posición de fila y columna asignados y el carácter error.

- **Public Sub** metodoSimbolosTerminales()

Este método me comprueba si hay números en un terminal, si es así agrega el error a la tabla.

- **Public Sub** metodoSimbolosNoTerminales()

Este método me comprueba si existen errores en los no terminales, basándonos en la expresión regular dada.

- **Public Sub** metodoTransiciones()

Método que me agrega las transiciones a la tabla dgvTrans.

- **Public Sub** metodoestadoInicial()

Me separa cual es el estado inicial y me lo agrega a una tabla distinta.

- **Public Sub** terminalesTrancisiones()

Para saber en que transición de usan los símbolos terminales.

- **Public Sub** noTerminalesTrancisiones()

Para saber en que transición de usan los símbolos no terminales.

- **Public Function** contadorcorchetes() As **Integer**

Para contar cuantos corchetes de apertura hay.

- `Public Function` contadorcorchetes2() As `Integer`

Para contar cuantos corchetes de cierre hay.

- `Public Sub` cadenaGrafo()

Para saber cuantas comas tiene una transición

- `Public Sub` cadenaGrafo2

Para cuando hay dos comas en una transición

- `Public Sub` cadenaGrafo3

Para cuando hay una coma en una transición

- `Public Sub` cadenaGrafo4()

Para verificar si los símbolos (terminal o no terminal) de las transiciones, existen en la tabla de símbolos

- `Public Sub` cadenaGrafo5()

Para cuando una transición tiene una coma, esto quiere decir que es el estado de Epsilon.

- `Public Sub` graficar()

Método para graficar

- `Public Function` generarGrafo() As `String`

Me va leyendo y concatenando las cadenas para generar el grafo.

- `Public Sub` abrirGrafo()

Método que me abre el grafo ya generado.

- `Public Sub` metodoLimpiarTodo()

Método que me limpia todas las tablas

- `Public Sub` inicioAnalizar()

Método que me inicia el análisis

- `Public Sub` MetodoHTML\_simbolos()

En este método se manejan el listado de símbolos para así mostrarlos en una página html.

- `Public Sub MetodoHTML_Errores_Lexemas()`

En este método se manejan el listado de errores y de tokens para así mostrarlos en una página html.

- `Public Sub metodo_abrir()`

Nos abre un archivo de texto para hacer analizado.

- `Public Sub metodo_guardar()`

Nos guarda los cambios hechos en el archivo que abrimos.

- `Public Sub metodo_guardarcomo()`

Nos guarda el archivo en una ruta específica.

- `Public Sub metodo_salir()`

Nos saca del analizador léxico.

### **Descripción de las clases:**

En la clase Form1 En la que se trabaja directamente la interfaz gráfica se manejan también los diferentes métodos léxicos los cuales son utilizados para la agrupación de caracteres.

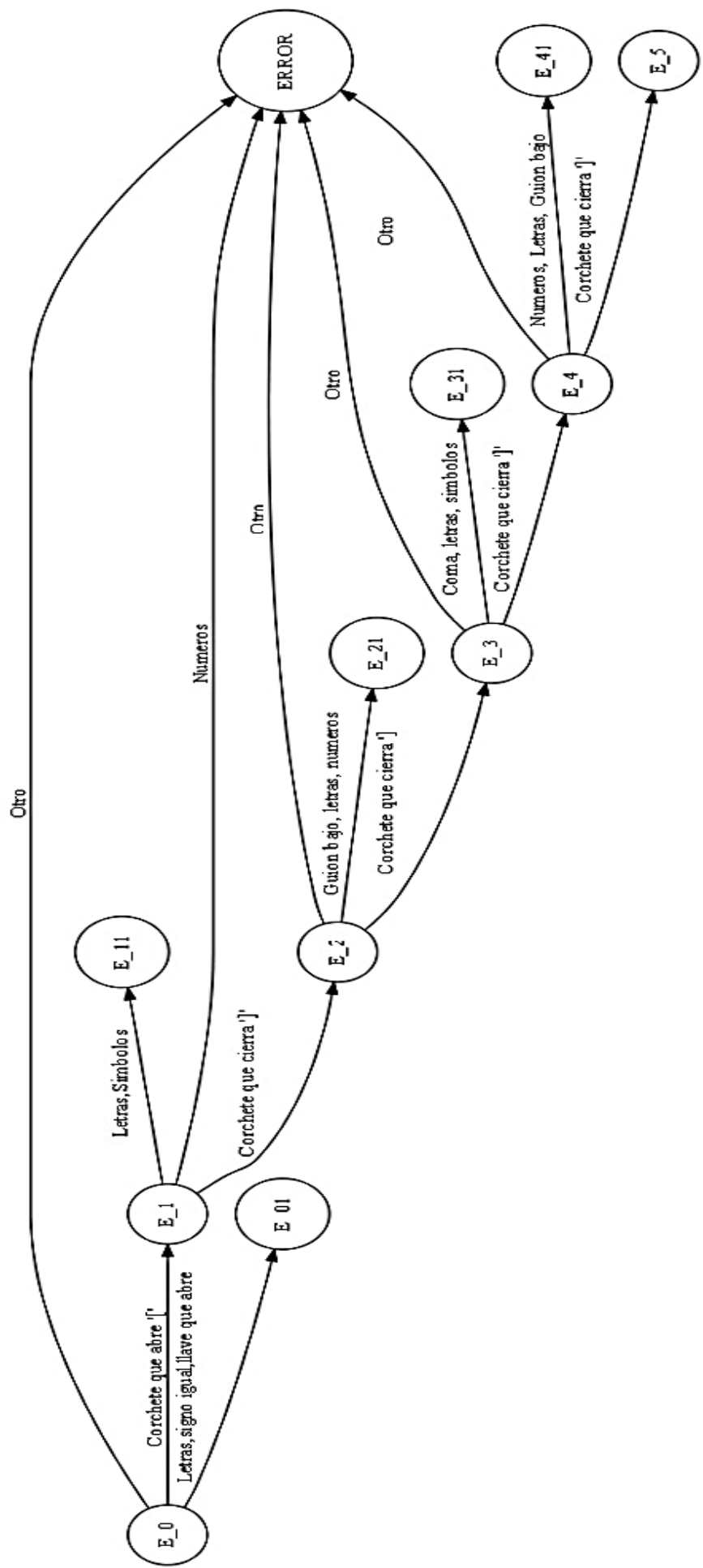
### **Descripción de métodos y funciones principales**

Los métodos principales en el programa son los métodos "Public Sub metodo\_analizar()" en este método se encuentra el corazón del programa pues es el encargado de tomar cada carácter del texto de uno a uno y manejarlos, para luego ya agrupados compararlos si coinciden con algún patrón léxico establecido, eso a su vez se encarga de los procesos de almacenaje y operación siendo los principales métodos.

### **Tabla de Gramática:**

Definición: $G = \{\Sigma, N, P, S_0\}$	
$\Sigma$	Representa el alfabeto (o "terminales") de la gramática
N	: Representará los "No Terminales" de las gramáticas
P	Identificara las reglas de producción (o bien dicho las transiciones) de la gramática
S <sub>0</sub>	Sera quien se define como estado inicial de la gramática

Automata:



## **Explicacion de Automata:**

E\_0 es nuestro estado inicial.

E\_0 Solo acepta corchete que abre, letras, signo igual y corchete que abre.

Si en el estado inicial viene el signo "[" entonces pasa al estado E\_1.

E\_1 solo acepta letras y cualquier símbolo, excepto corchetes, coma.

Si en el estado E\_1 viene el signo "]" entonces pasa al estado E\_2.

E\_2 solo acepta guion bajo, letras y números.

Si en el estado E\_2 viene el signo "]" entonces pasa al estado E\_3.

E\_3 solo acepta coma, letra, símbolos.

Si en el estado E\_3 viene el signo "]" entonces pasa al estado E\_4.

E\_4 solo acepta letras, números y guion bajo.

Si en el estado E\_4 viene el signo "]" entonces pasa al estado E\_5.

E\_5 es cuando cierra el ultimo corchete de la gramática (termina la gramática).