

MANUAL TÉCNICO DE LA PRACTICA UNO “ANALIZADOR LEXICO”

<u>INDICE</u>	1
• INTRODUCCIÓN	2
• CLASES UTILIZADAS	3
• AUTOMATA FINITO DETERMINISTA	4
• LOGICA UTILIZADA	5

INTRODUCCIÓN:

El presente proyecto consta de un analizador léxico, con el cual podemos ver las palabras que pertenecen a cierto lenguaje ya definido, generando en un archivo HTML la lista de tokens y errores obtenidos del análisis, además el con el cual se puede organizar una serie de actividades de forma ordenada.

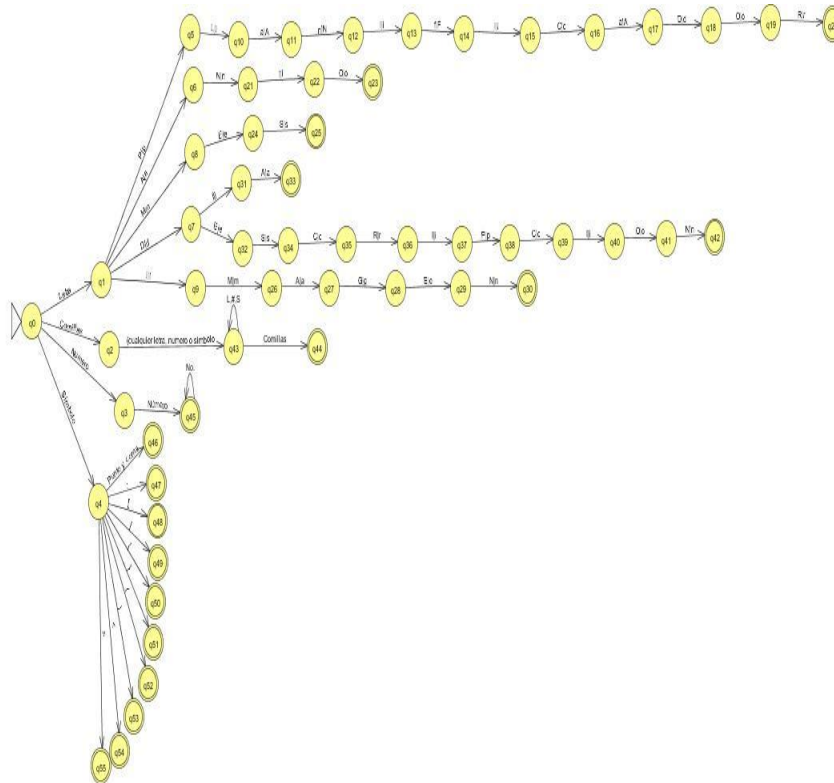
Para la solución del analizador léxico se utilizo un autómata finito determinista el cual se presentará más adelante.

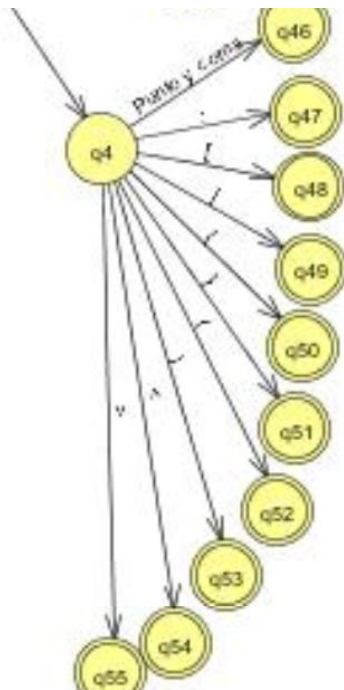
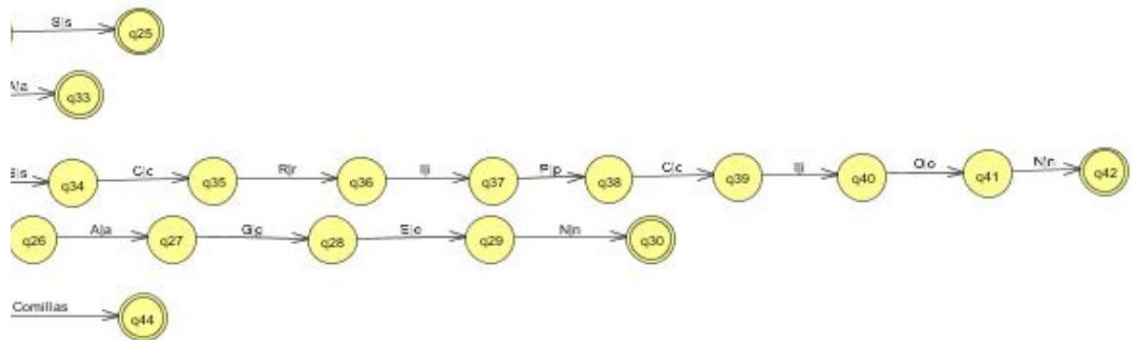
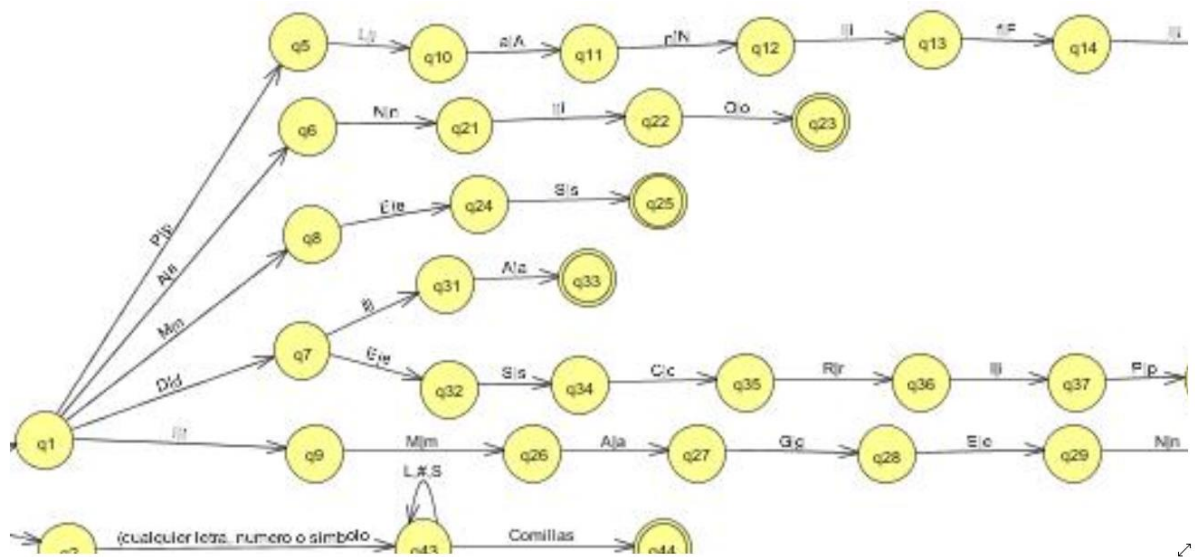
La aplicación fue creada utilizando Windows forms 2015, de c# como lenguaje de programación y utilizando el IDE de visual studio.

CLASES UTILIZADAS:

- TOKEN: En esta clase se creo una variable de tipo entero la cual contaba la línea del token, un String donde se guardaba el valor del token, un entero para el número del token y una variable enumerarle para el tipo del token.
- ANALISADOR LEXICO: En esta clase se crearon una lista de token y errores, donde a través de un método llamado scanner se analizaban la cadena de entrada y guardaba los errores y tokens resultantes
- ERROR: En esta clase se definió una variable string donde se guarda el valor del error, un entero donde se guarda el número de línea, otro entero para el numero de caracteres y un string de descripción del error.
- PRUEBA: En esta clase se crean todos métodos para que organice y guarde las fechas obtenidas del análisis para luego mostrarlas en el treeView y en el calendario.
- FORM2: Esta interfaz se utilizo para guardar los datos del estudiante
- PRINCIPAL: Esta es la interfaz principal del programa donde se muestra todo lo que este posee.
- DIA: Esta clase, posee una variable entera donde se guarda el número del día, un string para la descripción y otra para la ruta de la imagen.
- MES: En esta clase se guarda un entero donde se guardará el numero del mes y una lista de tipo día para guardar varios días
- AÑO: Esta clase cuenta con un entero para guardar el número del año, y una lista de tipo mes para guardar varios meses
- PLANIFICADOR: Esta clase cuenta con una variable de tipo string para guardar el nombre de cada plan y una lista de tipo año para guardar los distintos años.

AUTOMATA FINITO DETERMINISTA:





LOGICA UTILIZADA:

Se creo una clase token, la cual poseía en una variable enumerable todos los posibles valores que los tokens podían tener, además de una string valor donde se guarda el valor del token y un entero línea para mostrar el numero de la línea donde se encuentra el token.

Luego en la clase ANALIZADORLEXICO se creó una linkedList de tipo token llamada salida donde guardaríamos los valores de cada token obtenido.

Se creo un método de tipo linkedList<token> para que nos devolviera la lista de salida en donde por medio de un ciclo que recorría carácter por carácter el contenido del archivo de entrada, en este método por medio de un switch el cual se movía mediante una variable de tipo entero llamada estado, la cual cambiaba dependiendo el estado que tomaba el carácter analizado, estos estados previamente definidos en el AFD mostrado anteriormente.

El primer estado el esto 0, determinaba si el carácter era número, letra, comillas o alguno de los símbolos establecidos.

Cuando el carácter tomaba el valor de uno de los símbolos establecidos se llegaba a un estado de aceptación donde se guardaban los datos del token.

Para contar las líneas se tenia un contador de tipo entero que se incrementaba en 1 cada vez que el carácter era un salto de línea.

También se definió que cuando fuera un espacio en blanco regresara al mismo estado 0 sin guardar ningún token ya que el espacio en blanco no contaba.

A la cadena de entrada se le agregaba un # al final para determinar que cuando se llegase a ese símbolo se terminaba el análisis, y saliera del ciclo.

Cuando en el estado 0 se encontraba un “se sabia que era una cadena de caracteres así que cuando se iba ese estado se guardaban todos los caracteres que vinieran y se permanecía en ese estado hasta que el carácter tomara el valor de “.

Cuando en el estado 0 se encontraba un numero se pasaba al estado de los 4 de los numero y se permanecía y se guardaban los valores hasta que el carácter dejara de ser un número.

Cuando en el estado 0 se encontraba una letra se pasaba al estado 4 para determinar si la letra que venía pertenecía a la primera letra de alguna de las palabras reservadas, al encontrar así una de estas se pasa a un estado por aparte para cada una donde se reconocían solo las letras de esa palabra y si se llegaba ala ultima letra y se completaba se guardaba el token si no se guardaba como error.