Universidad San Carlos de Guatemala

Facultad de Ingenieria

Escuela de Ciencias y Sistemas

Ing Mario Bautista

Aux Fernando Paz

MANUAL TECNICO

Bryan Alexander Portillo Alvarado

201602880

INTRODUCCION

En el presente documento fue elaborado con la intención, de hacer que el lector pueda conocer a detalle el funcionamiento de la aplicación; Este contiene una breve explicación de las clases utilizadas en el proyecto, como también su explicación de las Clases empleadas y los Requerimientos mínimos para la funcionalidad optima de la misma, entre otras cosas más que se encuentran indicadas en el índice, esperamos que sea de su utilidad

INDICE

Descripcion Paginas

OBJETIVOS DEL MANUAL

GENERAL**:**

El objetivo de este manual es proporcionar información acerca del desarrollo de la aplicación, para aquel que desee aportar al mejoramiento de la aplicación.

ESPECIFICO:

Brindar al lector una concepción técnica de la funcionalidad de los principales procesos del sistema, con la ayuda del ide VISUAL STUDIO y el lenguaje de programación C#.

OBJETIVOS Y ALCANCES DE LA APLICACION

Mediante la realización de este proyecto, en el lenguaje de programación C#. se abarcará parte de los conocimientos del análisis léxico y del análisis Sintáctico, adquiridos durante el curso de Lenguajes formales y por ende parte de este curso de compiladores 1. Este proyecto es una versión mucho mas simple de un lenguaje de Consultas como lo es SQL, que a pesar de comprender parte de su sintaxis como tal no abarco completamente el concepto del anterior, además cabe recalcar que este proyecto resulta que su idioma principal es el español, por lo tanto se adaptó hacia esa necesidad.

DESCRIPCION GENERAL DE LA APLICACION

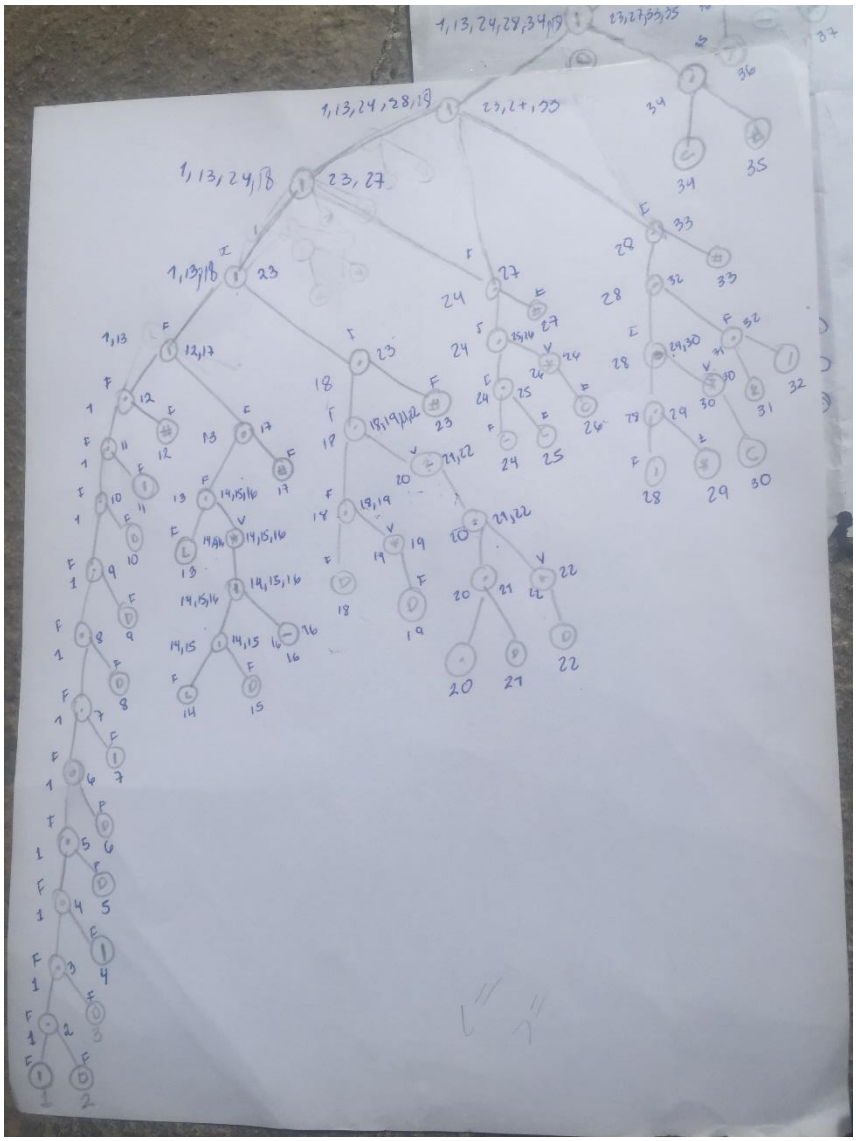
La aplicación cuenta con único modulo el cual permite cargar dos tipos de archivos: un archivo el cual contiene las informaciones que serán almacenadas en memorias por medio del concepto de tablas. Esta aplicación al tomar como base SQL, podemos cargar un segundo archivo para ejecutar nuestras consultas, en ellas podemos realizar las operaciones básicas, tales como eliminar, actualizar y seleccionar. Esta utilización de comandos es permitida, gracias a la aplicación del de los análisis lexico como sintactico, las funciones y métodos correspondientes se detallaran a lo largo del manual

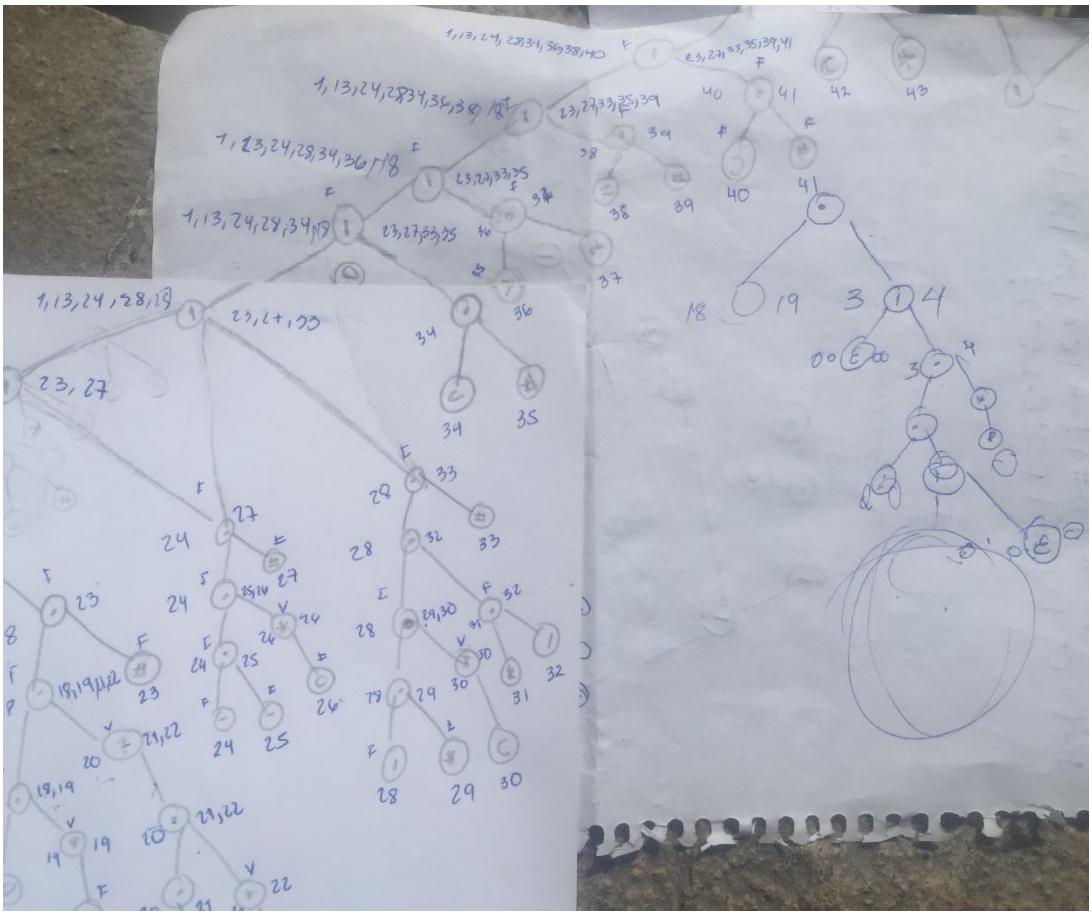
CARACTERISTICAS TECNICAS

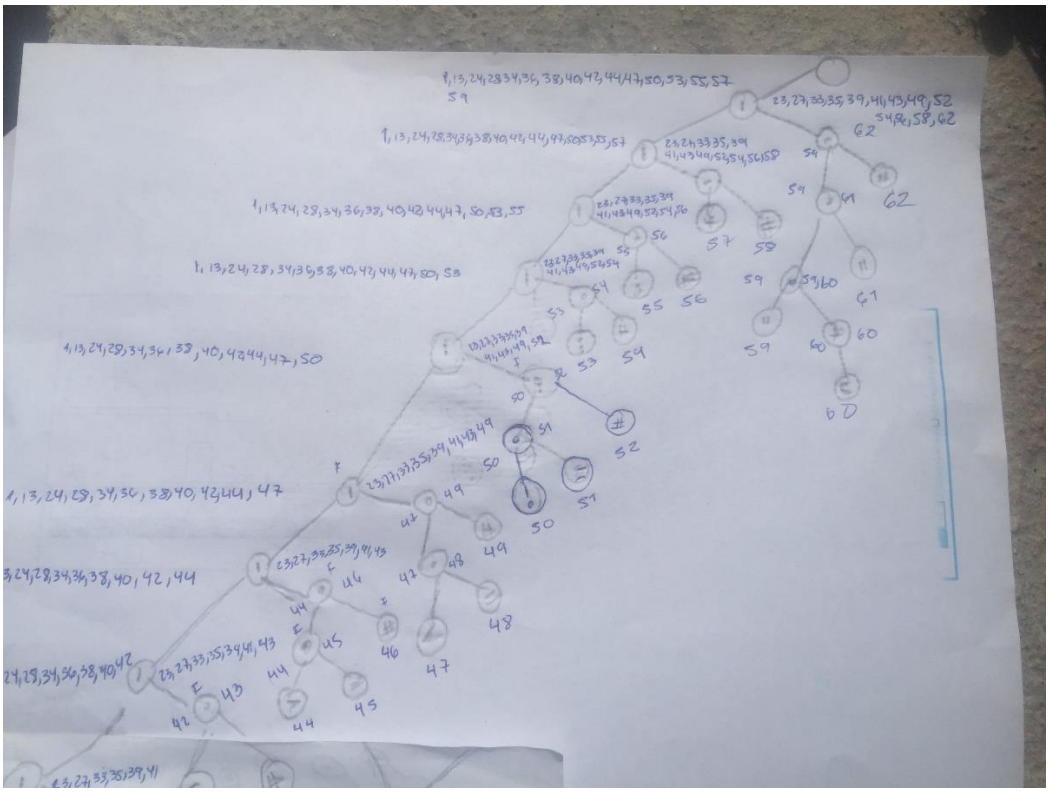
RECOMENDABLE:

- Versión SO: Windows Vista/7/8/10
- Procesador: Amd Turión o Superior, Intel Dual Core o Superior .
- Velocidad del Procesador: 1.5Ghz o superior
- Video: 256mb o Superior;

ANALISIS LEXICO







i	siguientes
1	2
2	3
3	4
4	5
5	6
6	7
~~7~~	8
8	9
9	10
10	11
11	12
12	~~XXXX~~
13	14,15,16
14	14,15,16,17
15	14,15,16,17
16	14,15,16,17
17	~~XXXX~~
18	19,20,23
19	19,20,23
20	21
21	22,23
22	22,23
23	~~XXXXXX~~
24	25
25	26,27
26	26,27
27	~~XXXXX~~

i	siguientes
28	29
29	30,31
30	30,31
31	32
32	33
33	~~XXX~~
34	35
35	~~XXX~~
36	37
37	~~XXX~~
38	39
39	~~XX~~
40	41
41	~~XXX~~
42	43
43	~~XXX~~
44	45
45	46
46	~~XXX~~
47	48
48	49
49	~~XXX~~
50	51
51	52
52	~~XXX~~
53	54
54	~~XXX~~
55	56
56	~~XXX~~
57	58
58	~~XXX~~
59	60,61
60	60,61
61	~~XXX~~

Sig(13) = {14, 15, 16} = S2

Sig(24) = {25} = S3

Sig(28) = {29} = S4

Sig(34) ∪ Sig(47) = {35, 48} = S5

Sig(36) ∪ Sig(44) = {37, 45} = S6

Sig(38) = {39} = S7

Sig(40) = {41} = S8

Sig(42) = {43} = S9

Sig(50) = {51} = S10

Sig(53) = {54} = S11

Sig(55) = {56} = S12

Sig(57) = {58} = S13

Sig(59) = {60, 61} = S14

Sig(53) = Acepta

Sig(56) = Acepta

Sig(58) = Acepta

Sig{60} = S14

Sig{61} = {6} = S30

Sig(62) = Acepta

Sig(S1) = {D 3} = S15

Sig(S15) = {D 4} = S16

Sig(S16) = {D 5} = S17

Sig(S17) = {D 6} = S18

Sig(S18) = {7} = S19

Sig(S19) = {D 8} = S20

Sig(S20) = {D 9} = S21

Sig(S21) = {D 10} = S22

Sig(S22) = {11} = S23

Sig(S23) = {# 12}

Sig(14) = {14, 15, 16, # 17} = S24

Sig(15) = S24

Sig(16) = S24

Sig(25) = {26, # 27} = S25

Sig(26) = S25

Sig(29) = {30, 31} = S26

Sig(30) = {S26}

Sig(31) = {32} = S27

Sig(32) = {#}

Sig(35) = # Acepta

Sig(48) = {49} = S28

Sig(49) = # Acepta

Sig(37) = Acepta

Sig(45) = {46} = S2

Sig(46) = Acepta

Sig(39) = Acepta

Sig(44) = Acepta

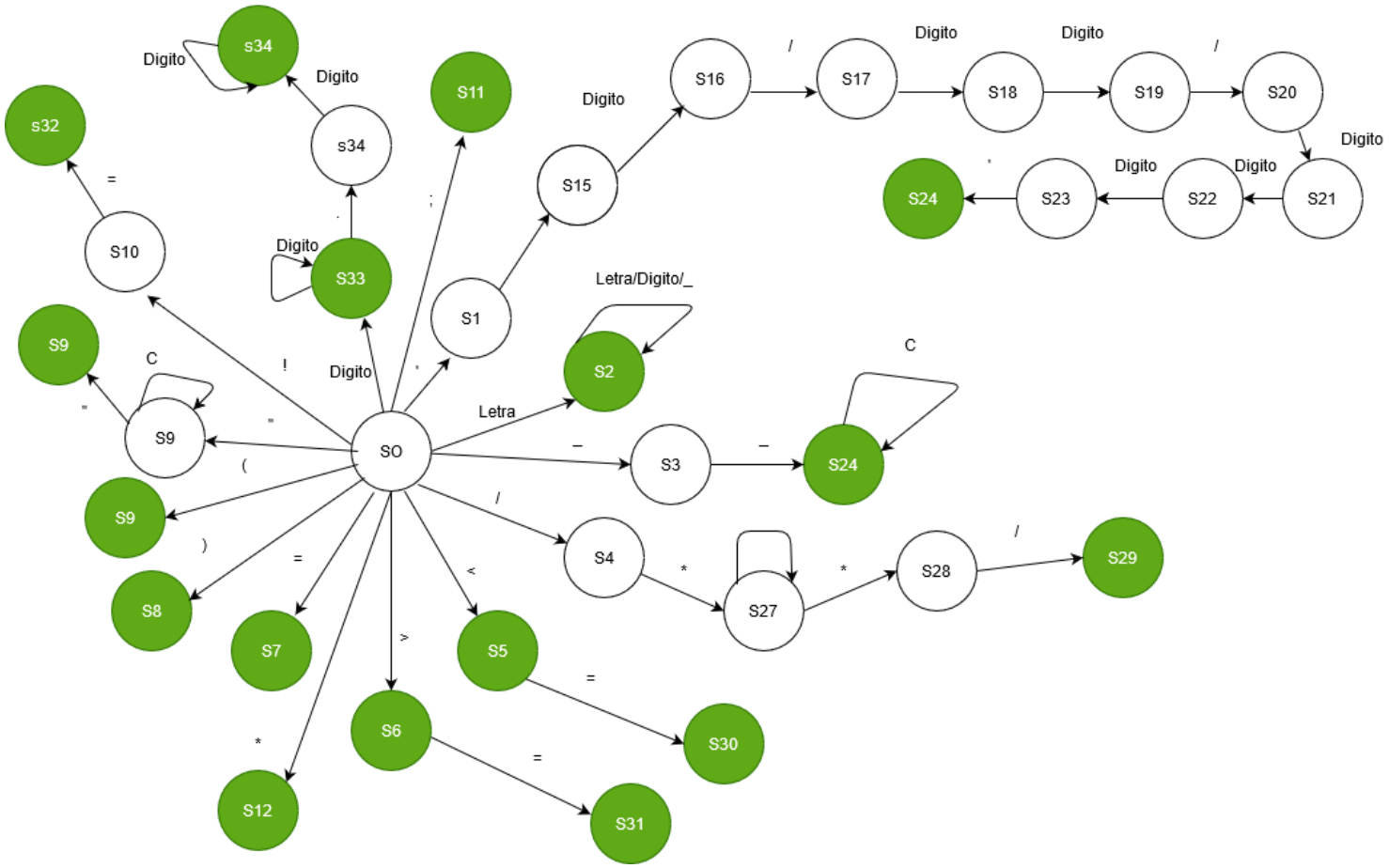
Sig(43) = {Ac

Sig(50) = {51}

Sig(51) = {52} =

Sig(52) = Acepta

AUTOMATA EQUIVALENTE



ANALISIS SINTACTICO

GRAMATICA:

Para los comandos que se requirieron para la aplicación se realización las siguientes gramáticas.

*/*gramática para la creación de tablas */*

SO-> CREAR TABLA Identificador (**SENTENCIA SENTENCIAR**);

SENTENCIA-> Identificador **TIPO**

SENTENCIA R->, **SENTENCIA SENTENCIAR_R** | épsilon

TIPO-> FECHA| ENTERO| CADENA| FLOTANTE

*/*gramática para insertar datos a tablas*/*

S1->**INSERTAR** EN Identificador Valores (**INGRESO**);

INGRESO-> , **VALOR INGRESO** | épsilon

VALOR-> Fecha | Entero| Cadena| Flotante

*/*Gramatica para eliminar datos de las tablas */*

S2 -> ELIMINAR DE Identificador **CUERPO**;

CUERPO-> DONDE **SUB_CUERPO**
| epsilon

SUB_CUERPO-> Identificador **CONDICION** VALOR **SUB_OPERADOR**

SUB_OPERADOR->Y **SUBCUERPO SUB_OPERADOR**
|O S**UBCUERPO SUB_OPERADOR**
| epsilon

CONDICION->Igual| Diferente| Mayor| Menor| Mayor Igual| Menor Igual

VALOR-> Fecha | Entero| Cadena| Flotante

*/*GRAMATICA DE ACTUALIZAR*/*

S3-> ACTUALIZAR Identificador ESTABLECER (Identificador Igual **VALOR VALOR_U**) **CLAUSULA_U**;

VALOR_U->,Identificador Igual **VALOR VALOR_U**
| epsilon

CLAUSULA_U->DONDE Identificador Igual VALOR **CLAUSULA_AUX**

CLAUSULA_AUX-> Y Identificador **CONDICION VALOR**
| O Identificador **CONDICION VALOR**
|epsilon

/*Gramatica de Seleccionar*/

S4-> SELECCIONAR **ORIGEN** DE Identificador **SUB_TABLA LOCALIDAD**;

ORIGEN-> Identificador . Identificador ALIAS SUB_ORIGEN
| *

SUB_ORIGEN-> ,Identificador . Identificado **ALIAS SUB_ORIGEN**
| epsilon

ALIAS-> COMO Identificador
| epsilon

SUB_TABLA->,Identificador **SUB_TABLA**
|epsilon

LOCALIDAD-> DONDE **REGLA**
| epsilon

REGLA-> Identificador **CONDICION VALOR REGLA_AUX**

REGLA_AUX-> Y Identificador **CONDICION VALOR REGLA_AUX**
| O Identificador **CONDICION VALOR REGLA_AUX**
|épsilon

*Nota: Para la utilización del análisis Sintactico se emplearon las anteriores gramatacias, sin embargo se agrego un estado inicial para unirlas.

START-> CREAR **S0** START
| INSERTAR **S1** START
| ELIMINAR **S2** START
| ACTUALIZAR **S3** START
| SELECCIONAR **S4** START
| epsilon

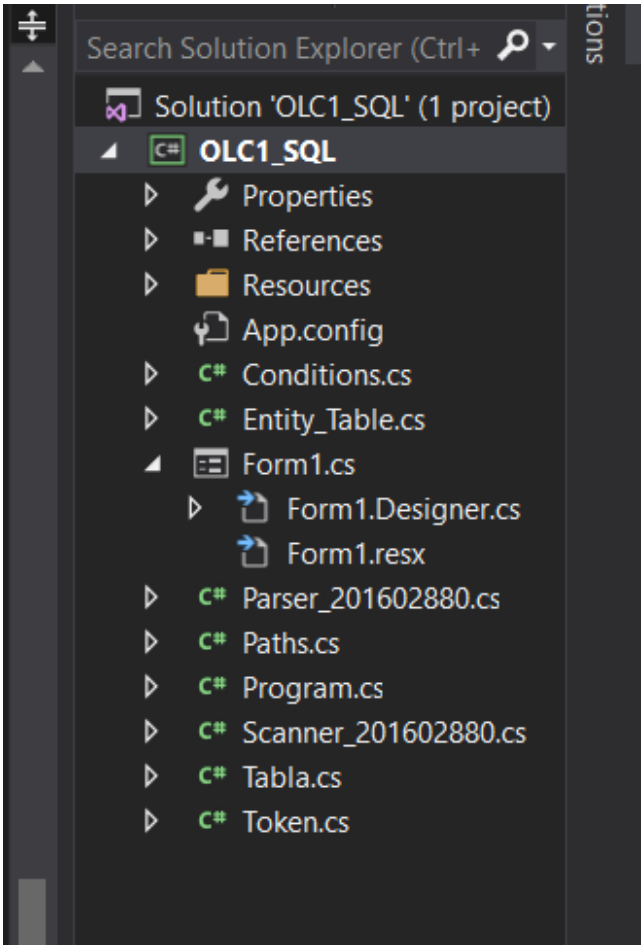
RECUPERACION DE ERRORES

Para la recuperación de errores sintácticos se aplicó el método de pánico, el cual consumirá tokens hasta llegar a un token designado y poder continuar así el análisis, pese haber encontrado uno anteriormente.

```
public void PanicMode(int counter)
{
    if (counter < Token_List.Count() - 1)
    {
        Token temp = post_analisis;

        temp = Token_List.ElementAt(counter);
        while (counter < Token_List.Count)//ultimo
        {


            if (temp.GetCorr() == 9)
            {
                post_analisis = temp;
                Numb_post_analisis = counter;
                match(9);
                START();
                break;
            }
            counter += 1;
            temp = Token_List.ElementAt(counter);

        }
    }
```

El token que fue designado para este método, fue el punto y coma, que tiene como asignado el correlativo #9

CLASES



Nombre	Descripcion
Conditions.cs	Esta clase contiene el obteto de tipo condición, donde funciona como un beans de la misma.

Entity_Table.cs	Esta clase también contiene el objeto de tipo columna que almacena, las tablas, de nuestro objeto tabla.
Form.cs	Esta clase contiene el modulo principal de la aplicación, acciones de botones y llamadas hacia los analizadores.
Parser_201602880.cs	En esta clase se realiza el análisis sintactico, también posee el método de recuperación, como también las acciones para manejar el almacenamiento de datos.
Paths.cs	En esta clase se contiene el objeto ruta, aplicado en las acciones Save and Save as.
Program.cs	Clase que invoca el modulo principal de la aplicación.
Scanner_201602880.cs	En esta clase se realiza el análisis Lexico, donde

	también, se encuentran los métodos para realizar reportes y almacenamiento de Tokens.
Tabla.cs	Esta clase contiene la descripción del objeto Tabla que se emplea para el almacenamiento de información, en el objeto respectivo.
Token.cs	En esta clase contiene la descripción del objeto Token, el cual es usado en para el almacenamiento de la lista de tokens y utilizado en los respectivos analizadores.