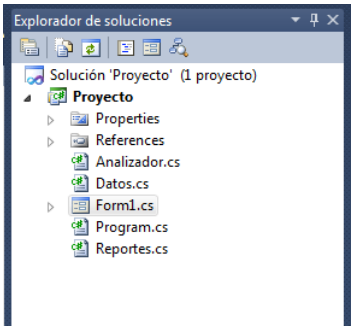
MANUAL TECNICO

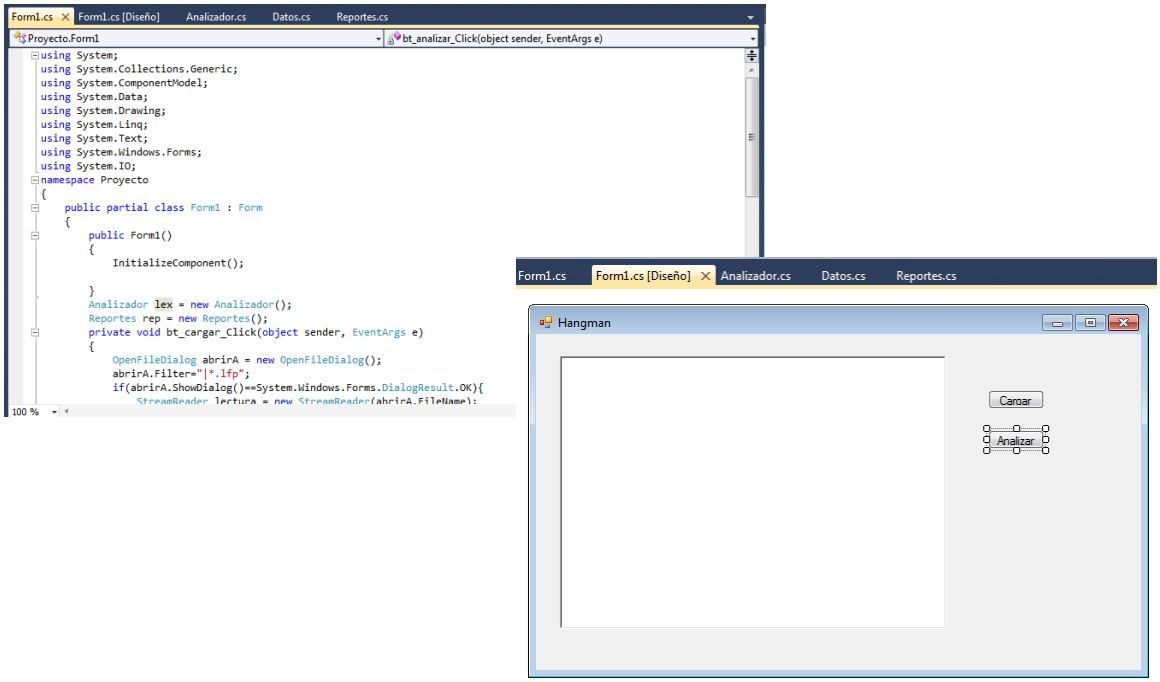
Composición básica de la aplicación

La aplicación de escritorio esta distribuida de la siguiente forma:



Un paquete principal que contiene todas las clases que se requieren para implementar la solución. Esta a su vez contiene cuatro clases que se detallaran a continuación:

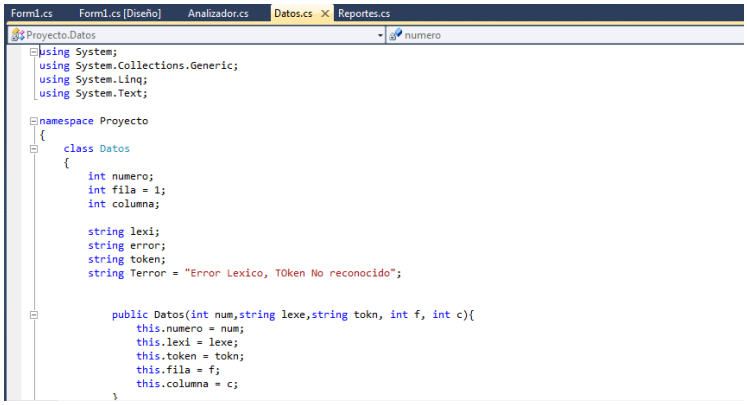
- Form1.cs: en esta se encuentra el método main y en esta se cargan la interfaz de la aplicación

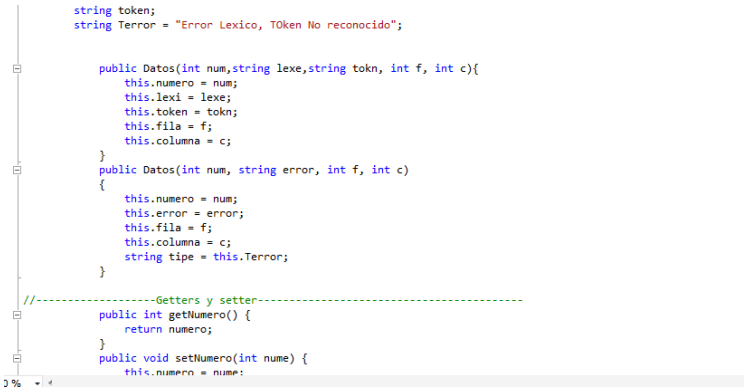


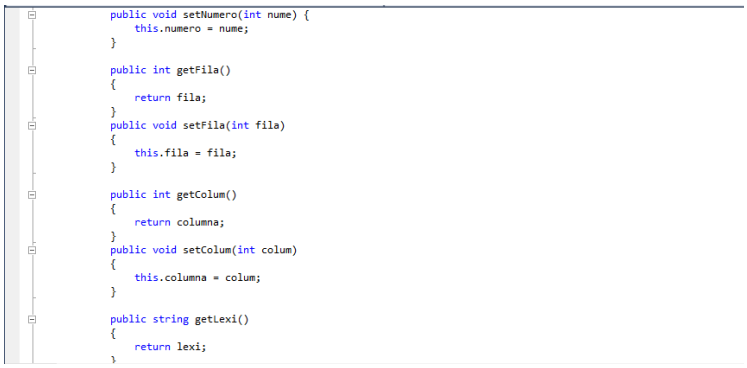
- Analizador.cs: En esta se encuentran todas las instrucciones para poder generar el anaiisis léxico, además esta contiene una serie de métodos adicionales.

```
Form1.cs   Form1.cs [Diseño]   Analizador.cs   Datos.cs   Reportes.cs
Proyecto.Analizador                                   lexico(string entrada)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Collections;

namespace Proyecto
{
    class Analizador
    {
        int state = 0;
        string lexema = "";
        int num = 0;
        int numerr = 0;
        int fila = 1;
        int columna=0;

        public List<Datos> arregloToken = new List<Datos>();
        public List<Datos> arregloError = new List<Datos>();


        public List<Datos> getArrT() {
            return arregloToken;
        }
100 %

//-----------------------------Metodo analizar-----------------------------------
        public string lexico(string entrada)
        {

            string texto = entrada;
            Char[] cadena = texto.ToCharArray();
            for (int i = 0; i < cadena.Length; i++)
            {
                if (cadena[i] == 10)
                {
                    fila++;
                    columna = 0;
                }
                else
                {
                    columna++;
                }

                switch (state){

            //----------------Estado 0-----------------------------------------------------------------

                    case 0:
                        if (Char.IsLetter(cadena[i]))
100 %
                    break;
                //---------------Estado 1-----------------------------------------------------------------
                    case 1:
                        if (Char.IsLetter(cadena[i]))
                        {
                            lexema = lexema + cadena[i];
                            state = 11;

                        }
                        else if (Char.IsDigit(cadena[i]))
                        {
                            lexema = lexema + cadena[i];
                            state = 11;

                        }
                        else if (cadena[i]==58)
                        {
                            lexema = lexema + cadena[i];
                            state = 4;

                        }
                        else if (cadena[i] == 32 || cadena[i] == 10 || cadena[i] == 9)
                        {

                            num++;
100 %
Proyecto.Analizador                                   lexico(string entrada)
                            lexema = lexema + cadena[i];
                            state = 100;
                        }
                        break;
                    //---------------Estado 11-----------------------------------------------------------------
                        case 11:
                        if (Char.IsLetter(cadena[i]))
                        {
                            lexema = lexema + cadena[i];
                            state = 11;

                        }
                        else if (Char.IsDigit(cadena[i]))
                        {
                            lexema = lexema + cadena[i];
                            state = 11;

                        }
                        else if (cadena[i] == 32 || cadena[i] == 10 || cadena[i] == 9)
                        {

                            num++;
                            arregloToken.Add(new Datos(num, lexema, tipoT(lexema), fila, columna));
                            lexema = "";
                            state = 0;

            TextWriter archivo = new StreamWriter(nomA);
            archivo.WriteLine(cod);
            archivo.Close();
        }

//-----------------------------Metodo Verificar tokens-----------------------------------
        public string tipoT(string lexema) {
            string tiptoken;

            switch(lexema){
                case "Configuracion":
                    tiptoken = "Token_Configuracion";
                break;

                case "configuracion":
                    tiptoken = "Token_Configuracion";
                break;

                case "Juego":
                tiptoken = "Token_Juego";
                break;

                case "Niveles":
                tiptoken = "Token_Niveles";
                break;
```

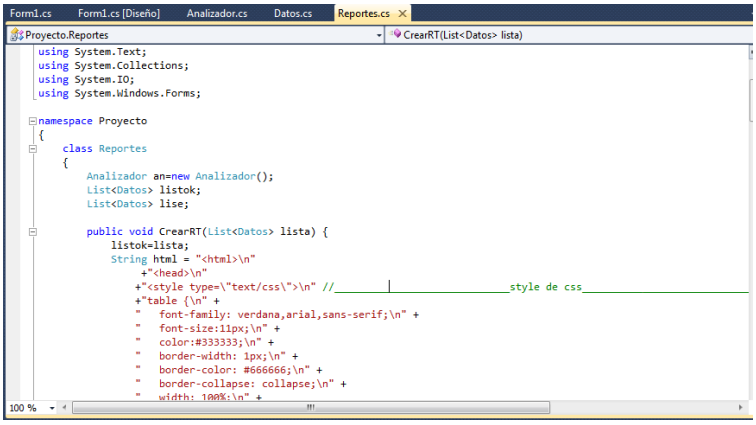
- Datos.cs: Aquí se crean los constructores que almacenan los distintos atributos que contendrá nuestra lista de objetos.
  También consta de sus respectivos métodos get y set de cada atributo.

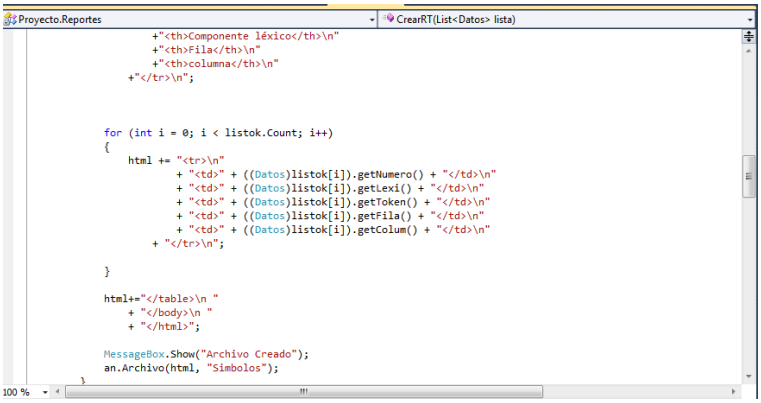






- Reportes.cs: clase encargada de crear un archivo html, para luego llenarlo con los datos de la lista de objetos estos archivos corresponden a la tabla de símbolos y errores.





Árbol Léxico

A continuación se muestra la implementación del método del árbol para poder generar el autómata que se utilizó para la programación del analizador.

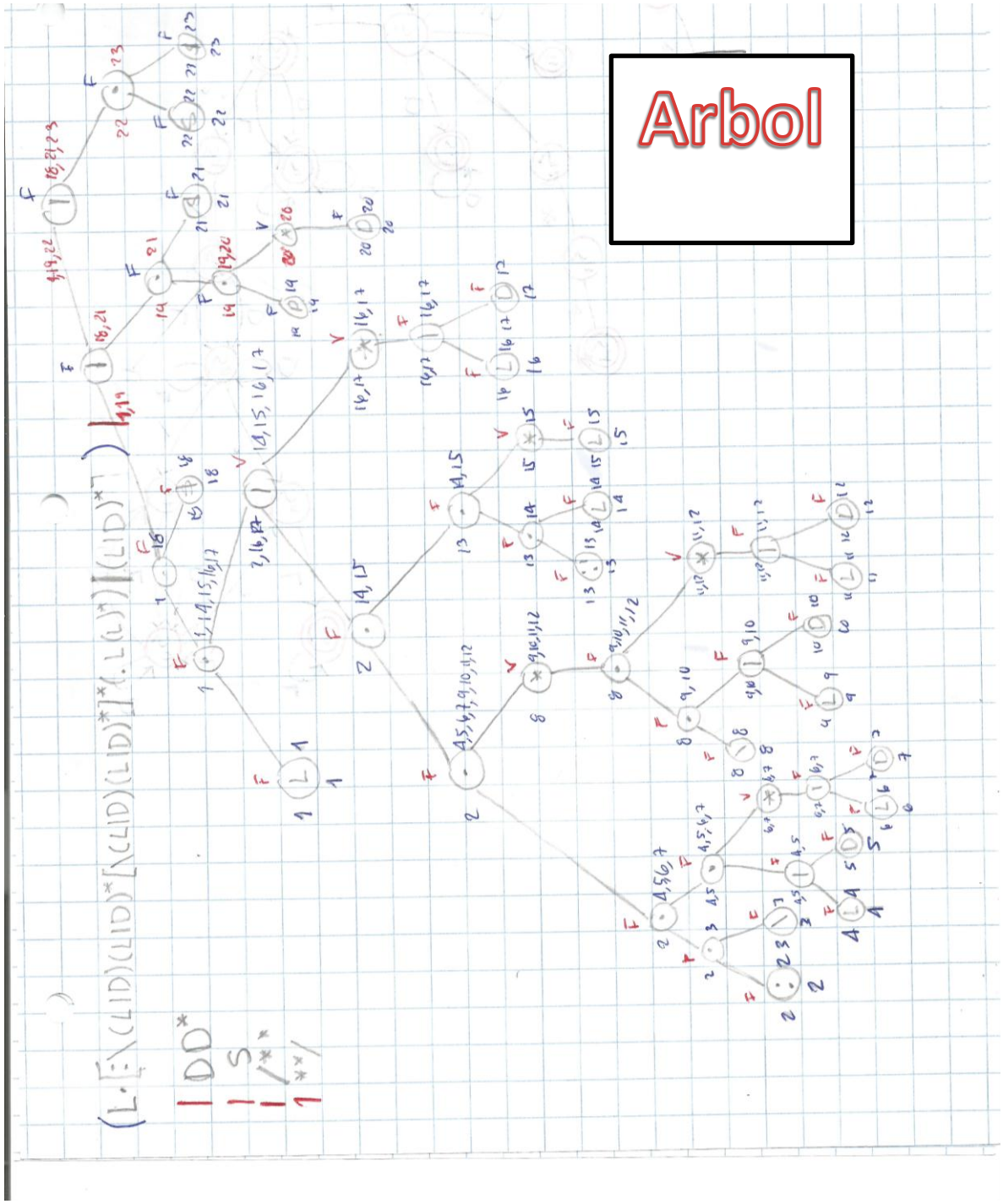
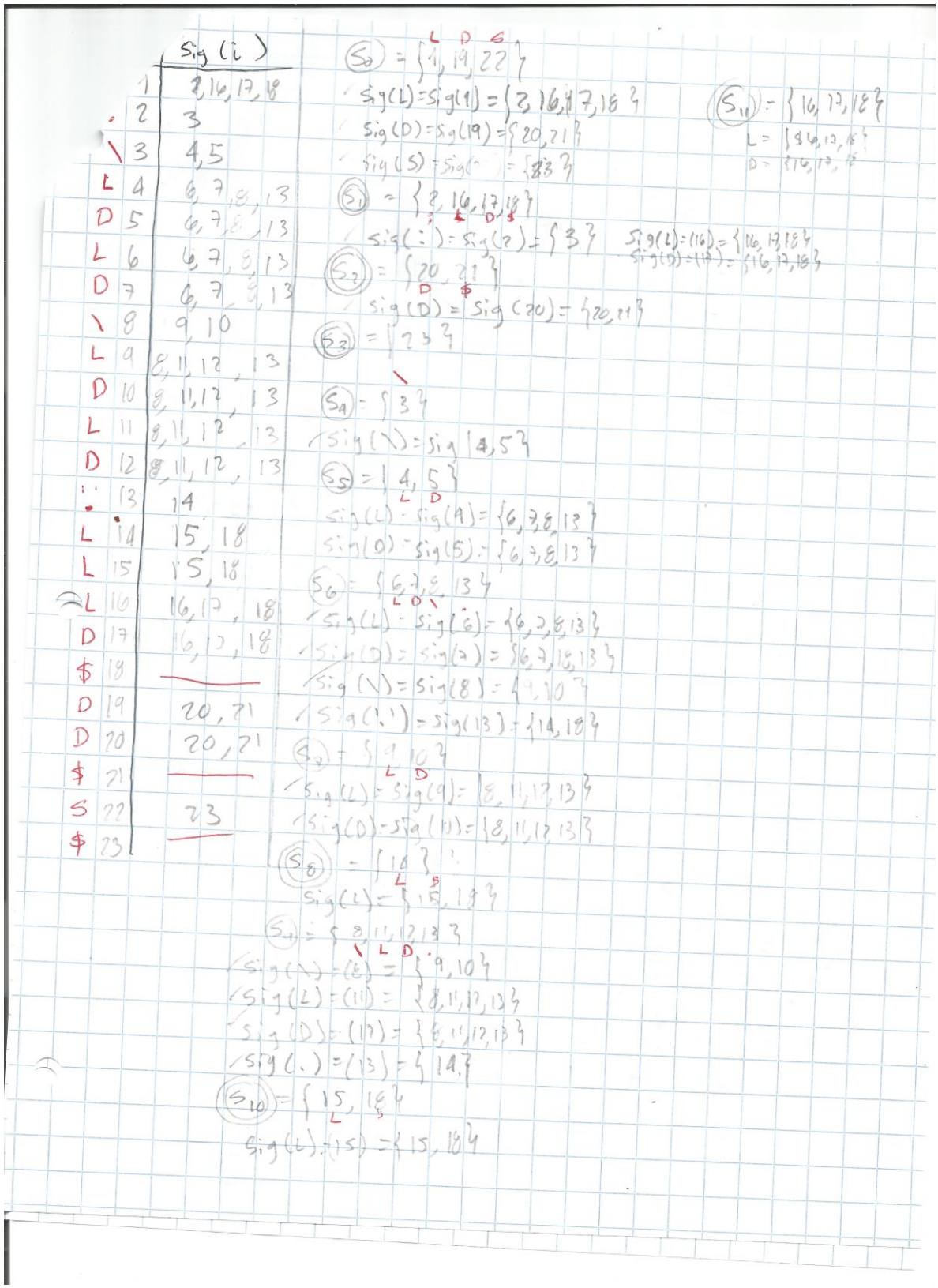


Tabla de siguientes



Autómata Determinístico