

# **MANUAL TECNICO**



**SOFTWARE DE ANALISIS DE ARCHIVOS**  
**OSCAR ARMIN CRISOSTOMO RUIZ**  
**201709140**

# OBJETIVOS

## - **Objetivo general**

Elaboración de un software capaz analizar expresiones regulares generando a su salida un autómata finito determinista con la validación de lexemas. Su funcionalidad principal será el interpretar expresiones regulares permitidas, por medio del análisis de un archivo de entrada, el cual contendrá expresiones regulares permitidas por el lenguaje.

## - **Objetivos específicos**

1. Definir claramente el procedimiento de instalación del aplicativo
2. Detallar la especificación de los requerimientos de Hardware y Software
3. Describir las herramientas utilizadas para el diseño y desarrollo del prototipo

# **REQUERIMIENTOS TECNICOS**

## **REQUERIMIENTOS MÍNIMOS DE HARDWARE**

- Procesador: Core
- Memoria RAM Mínimo : 1 Gigabytes (GB)
- Disco Duro: 500Gb.

## **REQUERIMIENTOS MÍNIMOS DE SOFTWARE**

- Sistema operativo: Windows /2000/2003/XP/10
- Conocimientos de extensiones de archivos

## **CARACTERISTICAS DE CREACION DE SOFTWARE**

- Sistema Operativo Windows 10 Home
- Procesador Intel Core i 5
- Visual Studio 2017
- .NET Framework4.6.1

# DISEÑO

## **Análisis Léxico**

AUTOMATA FINITO DETERMINISTA

- Abreviaturas

### LETRAS

$L = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$

### DIGITOS

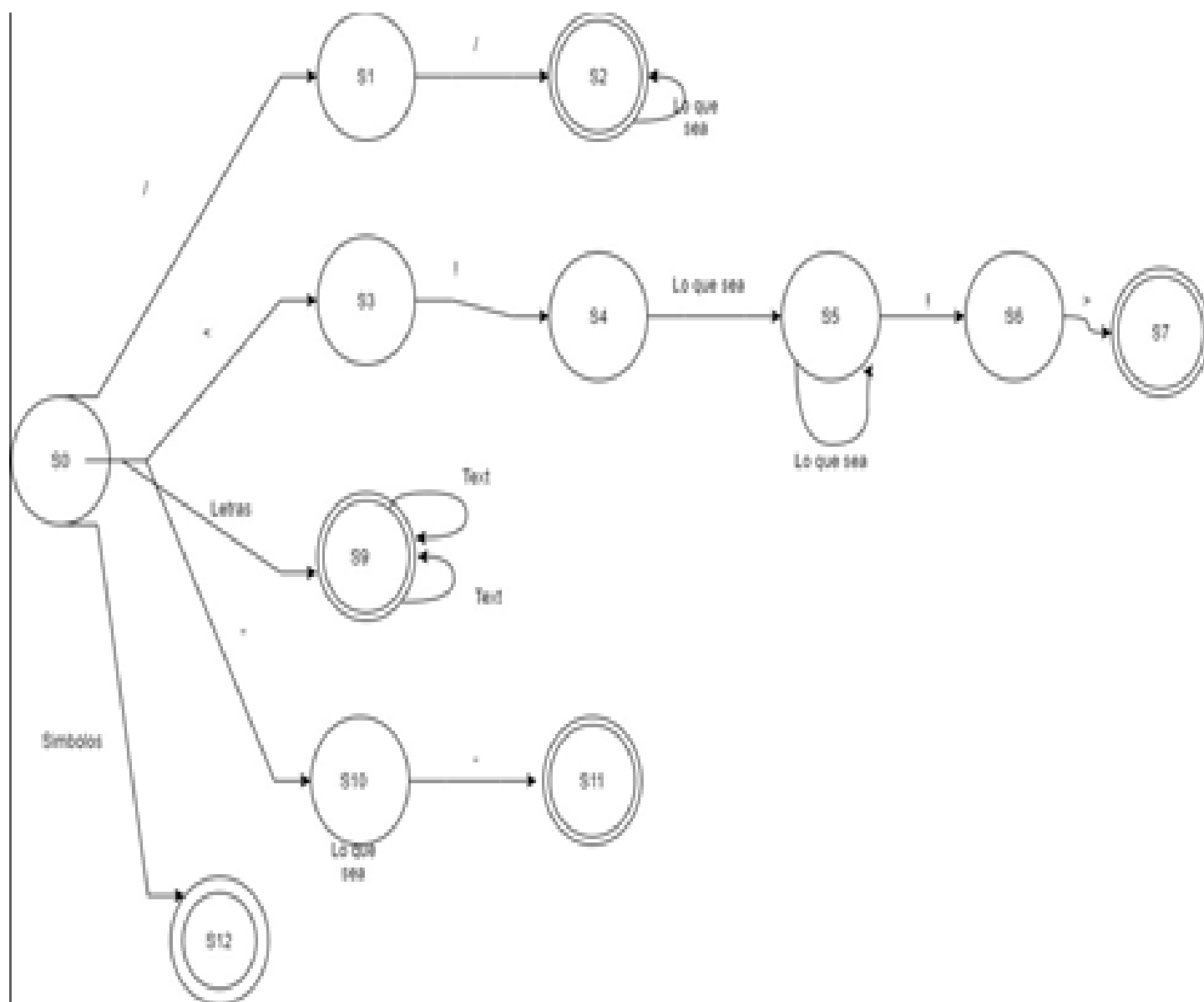
$D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

### SIMBOLOS

$S = \{\{, \}, [, ], (, ), :, ,, \}$

### CARACTERES

$C = \{\text{Cualquier carácter menos comillas}\}$



# TOKEN ASIGNADO

Token	Expresion	Descripcion
1	//.* /n	comentario una linea
2	<!. * !>	Comentario multilina
3	{	llave abre
4	}	llave cierre
5	L (L  D)*	identificador
6	->	flecha
7	L	letras
8	%	porcentaje
9	:	Dos puntos
10	;	punto y coma
11	" . * "	cadena/lexema
12	,	coma
13	~	virgulilla
14	D	digitos
15	*	asterisco
16	.	punto
17		pleca
18	+	signo mas
19	?	interrogacion
20	CONJ	Palabra reservada
21	ascii 32-127	simbolos
22	#	Numeral

Estructura básica de cada archivo de código fuente:

```
{  
ID1 -> Expresion_Regular ;  
ID2 -> Expresion_Regular ;  
ID3 -> Expresion_Regular ;  
//Mas sentencias.....  
  
%%  
%%  
  
IDA: "Lexema de entrada" ;  
IDB: "Lexema de entrada" ;  
IDC: "Lexema de entrada" ;  
//Mas sentencias  
}
```