

# Manual Tecnico

14/09/2020

---

Carlos Benjamín Pac Flores Carnet. 201931012  
Centro Universitario de Occidente (CUNOC)

<b>Definición de Lenguaje</b>	<b>2</b>
Signos Aceptados Por el lenguaje:	2
Letras:	2
Números:	2
Signos:	2
Palabras Reservadas:	3
Datos Primitivos:	3
Sentencias:	3
Operadores Aritméticos:	3
Operadores Relacionales:	3
Operadores Lógicos:	3
Signos De Agrupación:	4
Finalización de sentencia:	4
<b>Definición de Autómatas</b>	<b>4</b>
ID (referencia de variable):	5
Número Entero:	6
Número Decimal:	7
Cadena:	8
Carácter:	9
Comentario:	10
<b>Diagrama de Clases:</b>	<b>11</b>

## Definición de Lenguaje

El lenguaje manejado se basa en palabras reservadas, letras números, signos aritméticos, signos de agrupación, operadores relacionales y operadores lógicos que a continuación se detallan los mismos.

### Signos Aceptados Por el lenguaje:

#### 1. Letras:

a	b	c	d	e	f	g	h	i
j	k	l	m	n	ñ	o	p	q
r	s	t	u	v	w	x	y	z

A	B	C	D	E	F	G	H	I
J	K	L	M	N	Ñ	O	P	Q
R	S	T	U	V	W	X	Y	Z

#### 2. Números:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

#### 3. Signos:

	!	"	&	/	(	)	+	*
-	;	>	<	.				

## Palabras Reservadas:

### 1. Datos Primitivos:

- 1) entero → 34,2
- 2) decimal → 3.4, 678.9, 0.45
- 3) cadena → "esta es una cadena"
- 4) booleano → verdadero,falso
- 5) carácter → s,S,a,A

### 2. Sentencias:

- 1) SI
- 2) SINO
- 3) SINO\_SI
- 4) MIENTRAS
- 5) HACER
- 6) DESDE
- 7) HASTA
- 8) INCREMENTO

## Operadores Aritméticos:

- 1) +
- 2) -
- 3) \*
- 4) /
- 5) ++
- 6) --
- 7) = (este símbolo también cumple como de asignación de valor)

## Operadores Relacionales:

- 1) >
- 2) <
- 3) >=
- 4) <=
- 5) ==
- 6) !=

## Operadores Lógicos:

- 1) ||

- 2) &&
- 3) !

### Signos De Agrupación:

- 1) (
- 2) )

### Finalización de sentencia:

- 1) ;

También es de mencionar que se puede admitir el uso de comentarios en el lenguaje con el siguiente formato:

- //Esto es un comentario
- /\* Esto también es un comentario \*/

## Definición de Autómatas

Los autómatas son utilizados para el reconocimiento de cadenas así saber al tipo de expresión a la cual pertenece, partiendo desde esta premisa la definición de los autómatas utilizados se basan en expresiones regulares las cuales son:

Referencia	Expresión
ID (referencia a variable)	$[L] + [L] \{[N]   [ \_ ]   ([L])\}^*$
Número entero	$[0\dots9]^+$
Número decimal	$[0\dots9]^+ \cdot [.] [0\dots9]^+$
Cadena	$(\text{"}(\Sigma)^*\text{"})$
Carácter	$([A\dots Z]   [a\dots z])$
Comentario	$(/\cdot(\Sigma)^*\cdot(\backslash n)/\cdot'^*\cdot(\Sigma)^*\cdot'^*/)$

En referencia ID se toma "L" como cualquier letra mayúscula o minúscula, "N" como cualquier número y "\_" solamente como guión bajo.

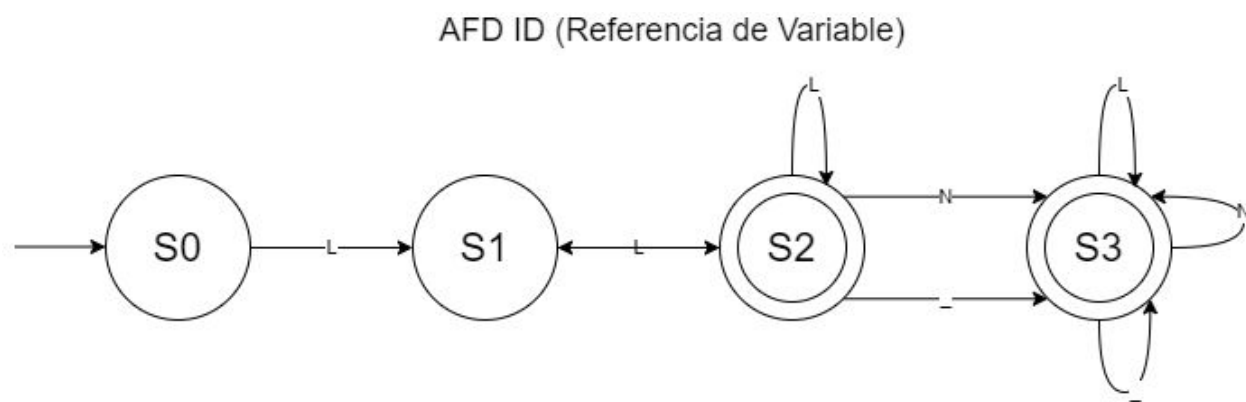
Para los estados booleanos su expresión es considerada como una palabra reservada.

Las palabras reservadas como su nombre lo indica no tienen una expresión regular como tal, sino que son reconocidas automáticamente por jerarquía.

Los autómatas fueron generados por medio del método de Thomson resolviendo las expresiones regulares planteadas anteriormente y algunos de ellos pasaron por una simplificación para llegar al autómata óptimo.

A continuación se presentan los dibujos de los autómatas:

ID (referencia de variable):



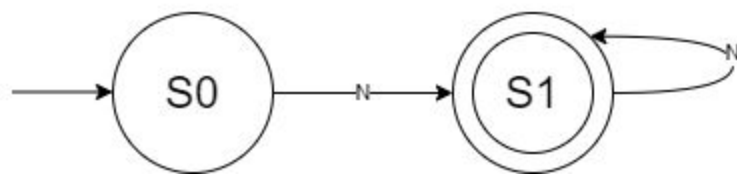
Definición Formal	
Q	S0,S1,S2,S3
Estado Inicial	S0
Alfabeto	{L,N,_} = (letras, números, guión bajo)
Estados De Aceptación	S2,S3

Tabla de Transiciones				
Lenguaje	Estados			
	S0	S1	S2	S3
L	S1	S2	S2	S3

N	$\emptyset$	$\emptyset$	S3	S3
-	$\emptyset$	$\emptyset$	S3	S3

## Número Entero:

### ADF Numero Entero

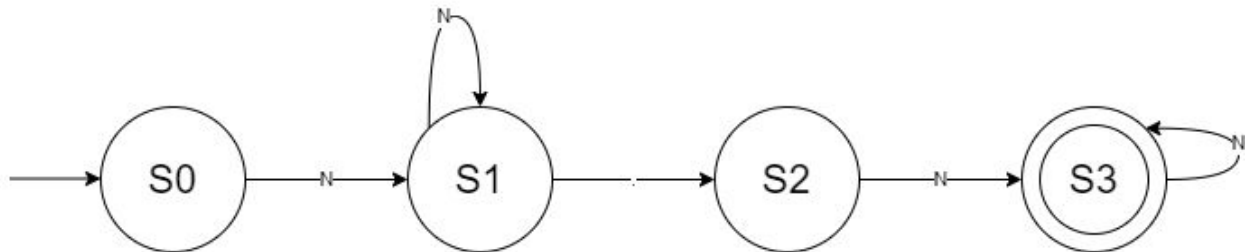


Definición Formal	
<b>Q</b>	S0,S1
<b>Estado Inicial</b>	S0
<b>Alfabeto</b>	{N} (números)
<b>Estados De Aceptación</b>	S1

Tabla de Transiciones		
Lenguaje	Estados	
	S0	S1
N	S1	S1

## Número Decimal:

AFD Numero Decimal



Definición Formal

Definición Formal	
<b>Q</b>	S0,S1,S2,S3
<b>Estado Inicial</b>	S0
<b>Alfabeto</b>	{N, "."} (números, punto)
<b>Estados De Aceptación</b>	S3

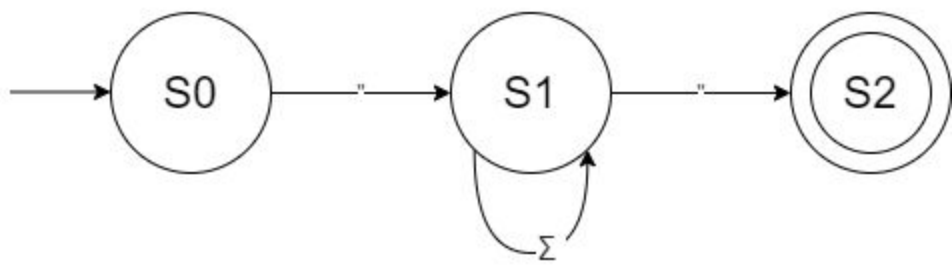
Tabla de Transiciones

Tabla de Transiciones				
Lenguaje	Estados			
	S0	S1	S2	S3
N	S1	S1	S3	S3
.	∅	S2	∅	∅



Cadena:

AFD Cadena de Texto

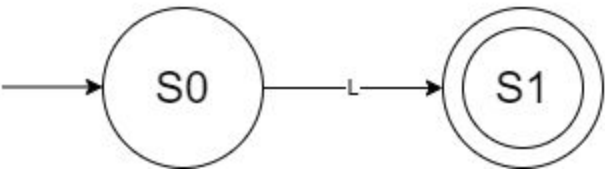


Definición Formal	
Q	S0,S1,S2
Estado Inicial	S0
Alfabeto	{",Σ} (comillas, cualquier carácter)
Estados De Aceptación	S2

Tabla de Transiciones			
Lenguaje	Estados		
	S0	S1	S2
"	S1	S2	∅
Σ	∅	S1	∅

Carácter:

AFD Carcter

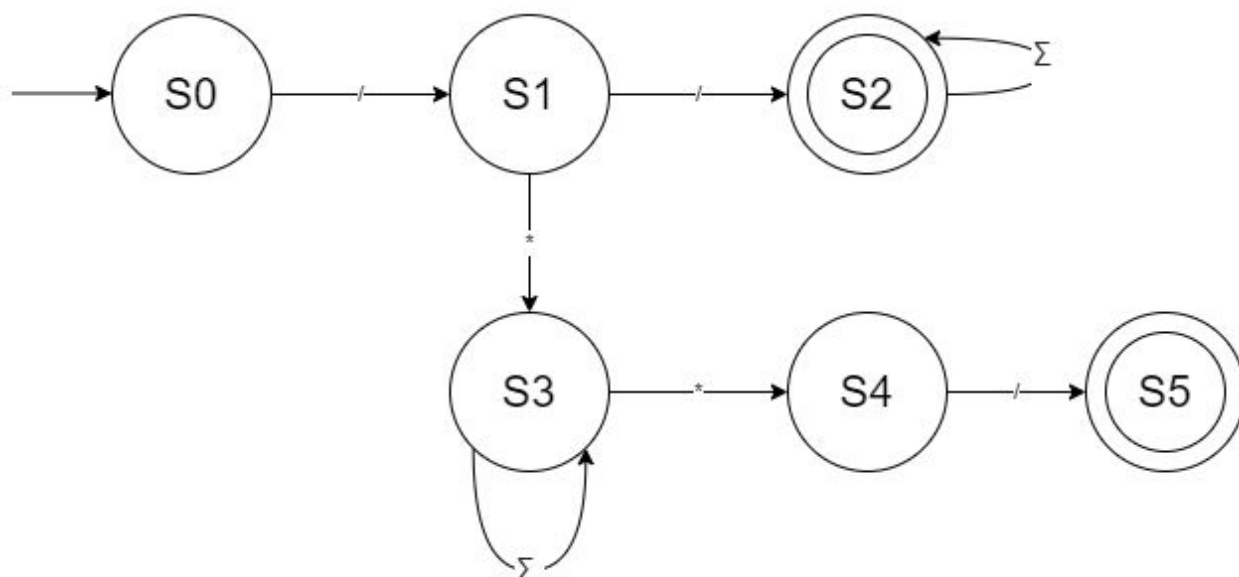


Definición Formal	
Q	S0,S1
Estado Inicial	S0
Alfabeto	{L} (letras)
Estados De Aceptación	S1

Tabla de Transiciones		
Lenguaje	Estados	
	S0	S1
L	S1	∅

Comentario:

AFD Comentario



Definición Formal	
<b>Q</b>	S0,S1,S2,S3,S4,S5
<b>Estado Inicial</b>	S0
<b>Alfabeto</b>	{/,*,Σ} (barra, asterisco, cualquier carácter)
<b>Estados De Aceptación</b>	S2,S5

Tabla de Transiciones						
Lenguaje	Estados					
	S0	S1	S2	S3	S4	S5
Σ	S1	∅	S2	S3	∅	∅
/	∅	S2	∅	∅	S5	∅

[illegible]

Se adjunta el diagrama de clases del proyecto adicionalmente en la carpeta contenedora se encontrara el mismo archivo en formato .png para su visualización.

A continuación se presenta la serie de producciones que fueron generadas para el análisis del lenguaje.

Las expresiones en **negrita** son los que nos representan nuestros estados terminales y las de color naranja son los estados no terminales

### Definición de producciones:

$E \rightarrow \text{PRINCIPAL } ( ) \{ \text{EXP} \}$

$\text{EXP} \rightarrow$   
ASIG\_VAR  
| DECLA\_VAR  
| PRINT  
| READ  
| ES\_SI  
| ES\_MIENTRAS  
| ES\_DESDE  
|  $\epsilon$

$\text{PRINT} \rightarrow \text{IMPRIMIR } ( \text{OPE} );$

$\text{OPE} \rightarrow$   
ASIG\_E' IDENT  
| ASIG\_D' IDENT  
| ASIG\_S' IDENT

$\text{IDENT} \rightarrow$   
+ OPE  
|  $\epsilon$

$\text{READ} \rightarrow \text{LEER } ( \text{ID\_CADENA} );$

$\text{ASIG\_VAR} \rightarrow$   
ID\_ENTERO ASIG\_VAR\_E  
| ID\_DECIMAL ASIG\_VAR\_D  
| ID\_BOOLEANO = ASIG\_B';  
| ID\_CARACTER = ASIG\_C';

ASIG\_VAR\_E → ++;  
| --;  
| = ASIG\_E';

ASIG\_VAR\_D → ++;  
| --;  
| = ASIG\_D';

DECLA\_VAR → ENTERO ASIG\_E  
| BOOLEANO ASIG\_B  
| DECIMAL ASIG\_D  
| CADENA ASIG\_S  
| CARACTER ASIG\_C

ASIG\_E → = ASIG\_E';  
| , ID\_ENTERO ASIG\_E  
| ID\_ENTERO ASIG\_E  
| ;

ASIG\_E' → - ASIG\_E"  
| ASIG\_E"

ASIG\_E" → NUMERO\_E  
| ID\_ENTERO

ASIG\_D → = ASIG\_D';  
| , ID\_DECIMAL ASIG\_D  
| ID\_DECIMAL ASIG\_D  
| ;

ASIG\_D' → - ASIG\_D"  
| ASIG\_D"

ASIG\_D" →      **NUMERO\_D**  
                  | **ID\_DECIMAL**  
                  | **ID\_ENTERO**  
                  | **NUMERO\_E**

ASIG\_B →        = ASIG\_B' ;  
                  | , ID\_BOOLEANO ASIG\_B  
                  | ID\_BOOLEANO ASIG\_B  
                  | ;

ASIG\_B' →        **VERDADERO**  
                  | **FALSO**  
                  | **ID\_BOOLEANO**

ASIG\_S →        = ASIG\_S' ;  
                  | , ID\_CADENA ASIG\_S  
                  | ID\_CADENA ASIG\_S  
                  | ;

ASIG\_S' →        **ID\_CADENA**  
                  | **CAD\_TEXTO**

ASIG\_C →        = ASIG\_C' ;  
                  | , ID\_CARACTER ASIG\_C  
                  | ID\_CARACTER ASIG\_C  
                  | ;

ASIG\_C' →        **CARACTER**  
                  | **ID\_CARACTER**

ES\_MIENTRAS → **MIENTRAS ( LOGICA ) { EXP }**

ES\_SI → **SI ( LOGICA ) { EXP } ES\_SI'**

ES\_SI' → SINO { EXP }  
 | SINO\_SI ( LOGICA ) { EXP } ES\_SI'  
 | ε

ES\_HACER → HACER { EXP } MIENTRAS ( LOGICA )

ES\_DESDE → DESDE ASIG\_DES HASTA LOGICA INCREMENTO ASIG\_DES" { EXP }

ASIG\_DES → ID\_ENTERO ASIG\_DES'  
 | ENTERO ID\_ENTERO ASIG\_DES'

ASIG\_DES' → = ASIG\_DES"

ASIG\_DES" → ID\_ENTERO  
 | NUMERO\_E

LOGICA → ASIG\_D" OP\_RELACIONAL ASIG\_D" LOGICA'  
 | ASIG\_B' LOGICA'  
 | ! LOGICA LOGICA'

LOGICA' → && LOGICA  
 | || LOGICA  
 | ε