

ANÁLISIS LÉXICO

-

DISEÑO DE COMPILADORES

Juan Daniel Salazar Coronado

Matricula: 17289193

Profesor: Ernesto Liñán García



```
resource_id' => $role_details['resource_id'],  
);  
if ( $this->rule_exists( $resource_details['id'], $role_details['role_id'] ) ) {  
    if ( $access == false ) {  
        // Remove the rule as there is currently no need for it  
        $details['access'] = !$access;  
        $this->_sql->delete( 'acl_rules', $details );  
    } else {  
        // Update the rule with the new access value  
        $this->_sql->update( 'acl_rules', array( 'access' => $access ) );  
    }  
} else {  
    foreach( $this->rules as $key=>$rule ) {  
        if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {  
            if ( $access == false ) {  
                unset( $this->rules[ $key ] );  
            } else {  
                $this->rules[ $key ]['access'] = $access;  
            }  
        }  
    }  
}
```

EXPRESIONES REGULARES:

- Números enteros: $\sim \text{digito}^+$
- Números enteros con notación científica: $\sim \text{digito}^+ \text{edigito}^+$
- Números decimales: $\sim \text{digito}^+ . \text{digito}^+$
- Números decimales con notación científica: $\sim \text{digito}^+ . \text{digito}^+ \text{edigito}^+$
- Identificadores: $\$ \text{letra} (\text{letra} | \text{digito} | _ | -)^*$
- Incremento: $< +$
- Decremento: $< -$
- Multiplicación: $\$ *$
- División: $@ /$
- Resta: $-$
- Suma: $+$
- Llave abierta: $@ \{$
- Llave cerrada: $@ \}$
- Paréntesis abierto: $@ ($
- Paréntesis cerrado: $@)$
- Operador Or: $@ |$
- Operador and: $@ \&$
- Corchete abierto: $@ [$
- Corchete cerrado: $@]$
- Igual(asignación): $< :-$
- Mayor: $@ >$
- Mayor igual: $@ > =$
- Menor: $@ <$
- Menor igual: $@ < =$

EXPRESIONES REGULARES QUE LLEVAN TODOS LOS PROYECTOS:

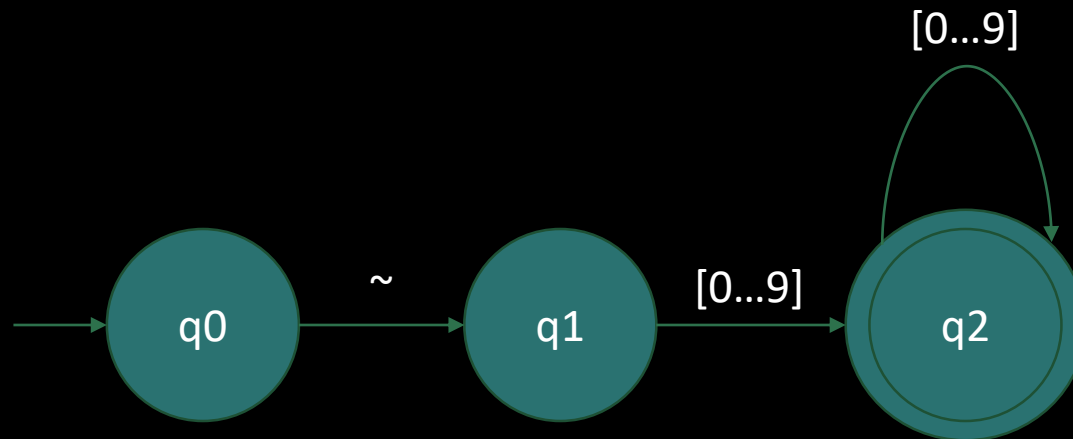
- | | |
|-----------------------|--|
| • Doble comillas: “ | • Dos puntos: : |
| • Punto y coma: ; | • Negación: ! |
| • Punto: . | • Brinco de linea: \n |
| • Signo \$: \$ | • Igual comparación: %=% |
| • Arroba: @ | • Tabulador |
| • Diagonal inversa: \ | • Comentario de una línea: //(letra digito espacio)* |
| • Espacio | • Comentario de varias líneas: /*(letra digito espacio)**/ |
| • Coma: , | • Texto forma 1: “(letra digito espacio)*” |
| • Comilla simple: ‘ | • Texto forma 2: ‘(letra digito espacio)*’ |

ALFABETO DEL LENGUAJE:

[0-9]	espacio	tabulador	e	.	~
letra	\$	*	<	+	-
_	@	/	{	}	(
)		&	[]	:
>	=	“	;	\	,
'	\n	!	%		

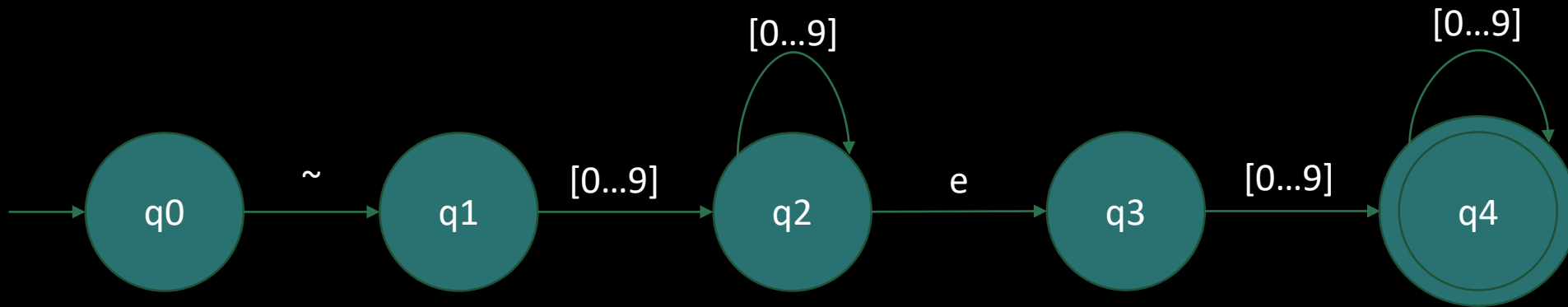
NÚMEROS ENTEROS

- Expresión regular: $\sim\text{digito}^+$



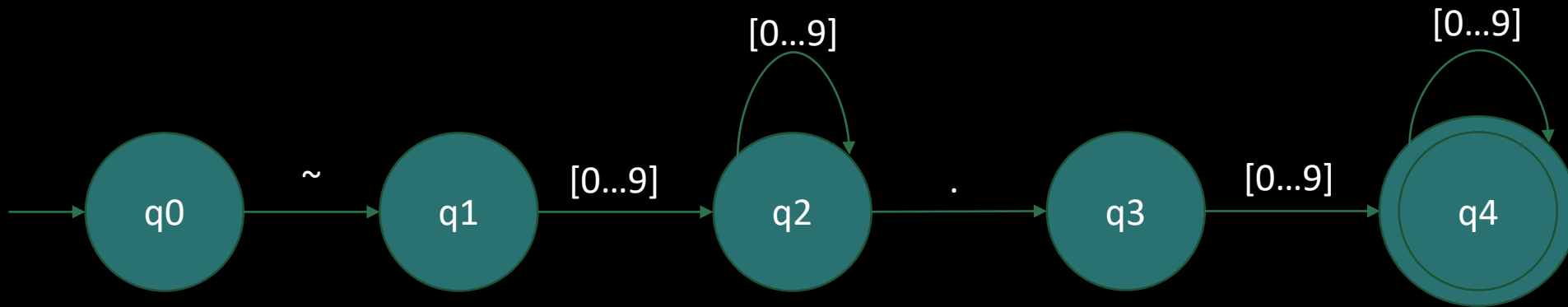
NÚMEROS ENTEROS CON NOTACIÓN CIENTÍFICA

- Expresión regular: $\sim\text{digito}^+\text{edigito}^+$



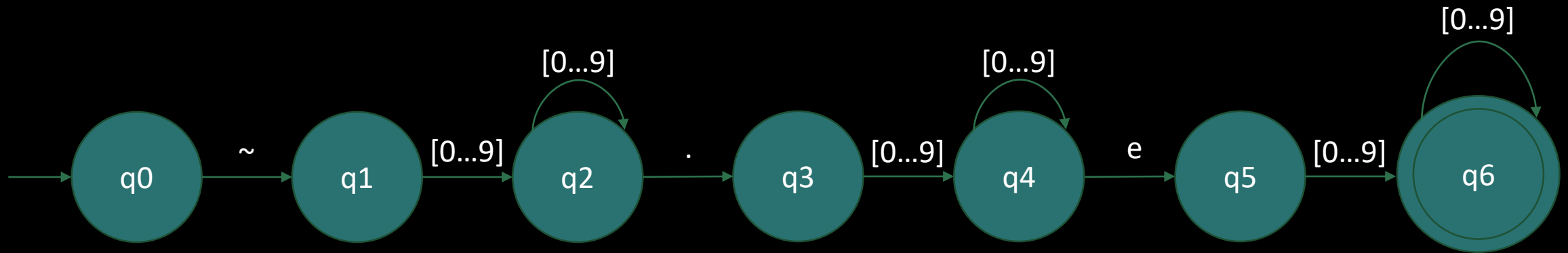
NÚMEROS DECIMALES

- Expresión regular: $\sim\text{digito}^+.\text{digito}^+$



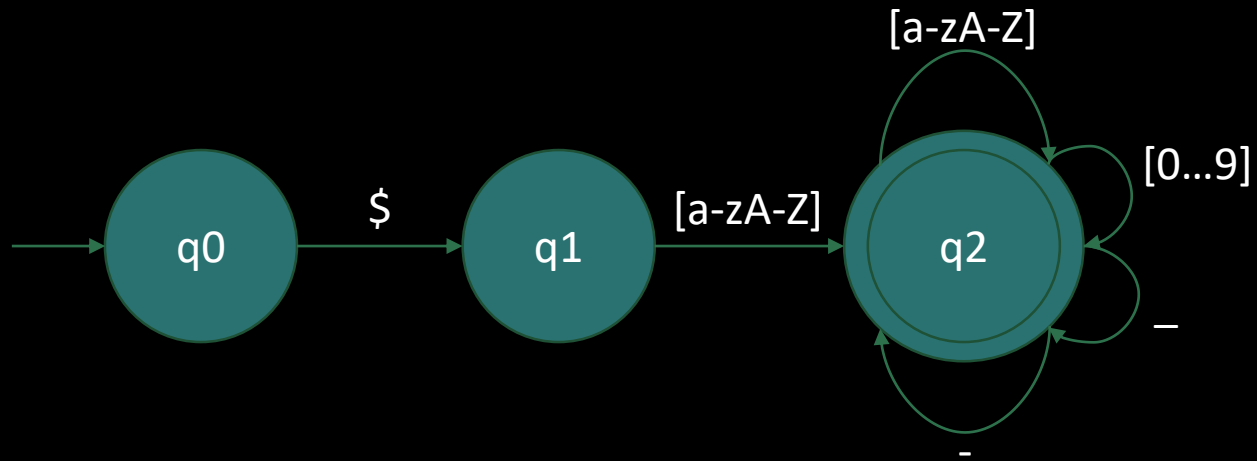
NÚMEROS DECIMALES CON NOTACIÓN CIENTÍFICA

- Expresión regular: $\sim \text{digito}^+ . \text{digito}^+ e \text{digito}^+$



IDENTIFICADORES

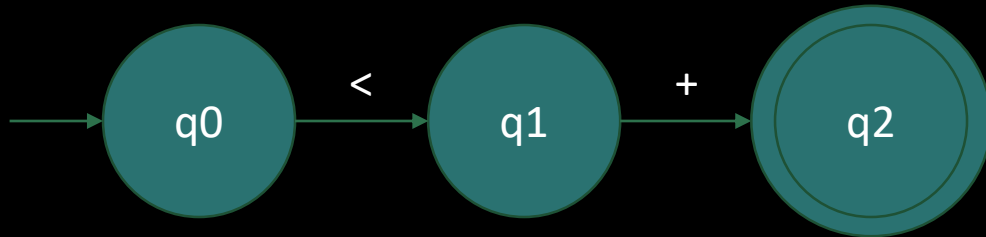
- Expresión regular: $\$letra(letra|digito|_|-)^*$



SÍMBOLOS ESPECIALES:

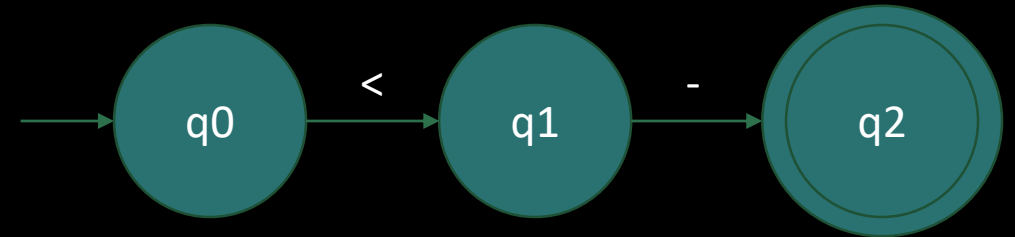
Incremento

Expresión regular: <+



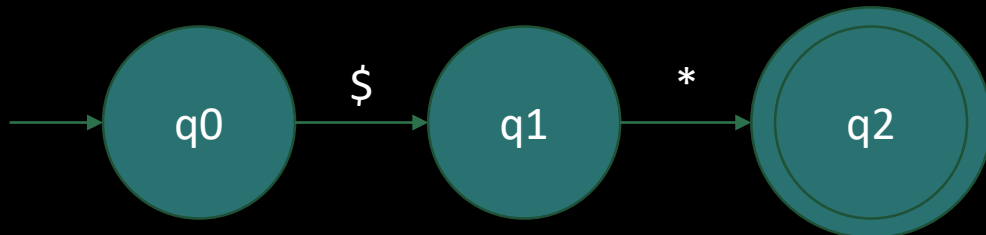
Decremento

Expresión regular: <-



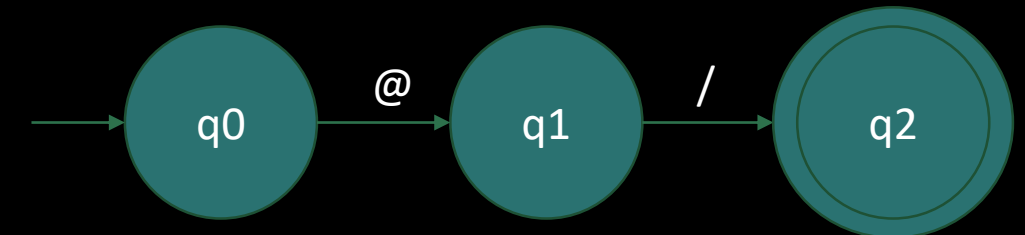
Multipliación

Expresión regular: \$*



División

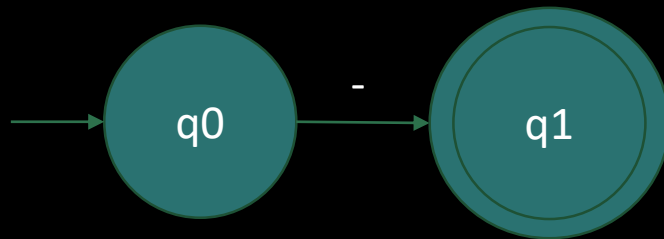
Expresión regular: @/



SÍMBOLOS ESPECIALES:

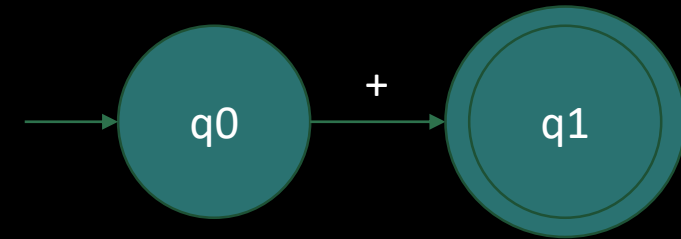
Resta

Expresión regular: -



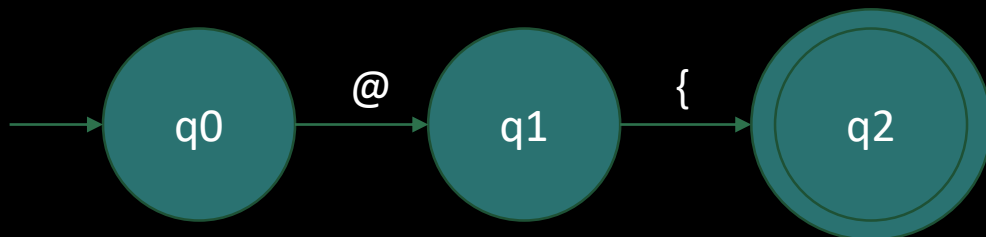
Suma

Expresión regular: +



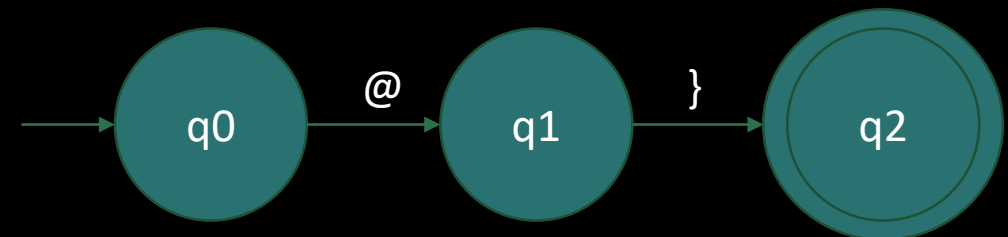
Llave abierta

Expresión regular: @{



Llave cerrada

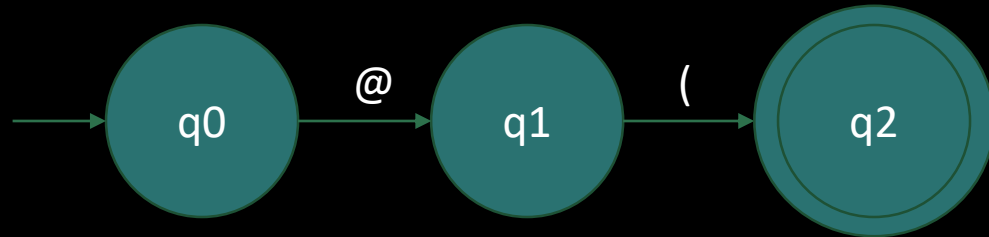
Expresión regular: @}



SÍMBOLOS ESPECIALES

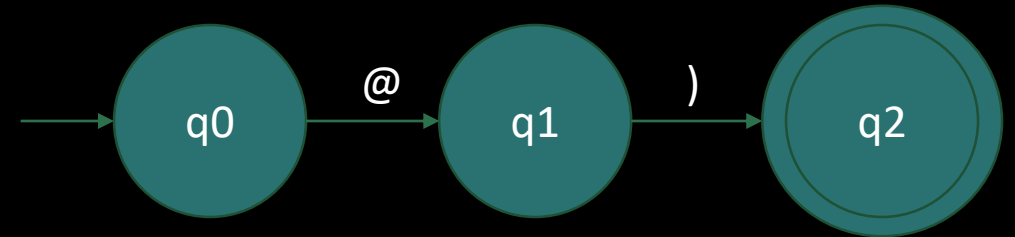
Paréntesis abierto

Expresión regular: @(



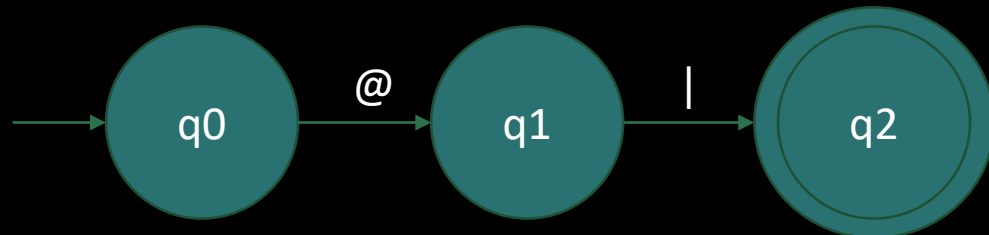
Paréntesis cerrado

Expresión regular: @)



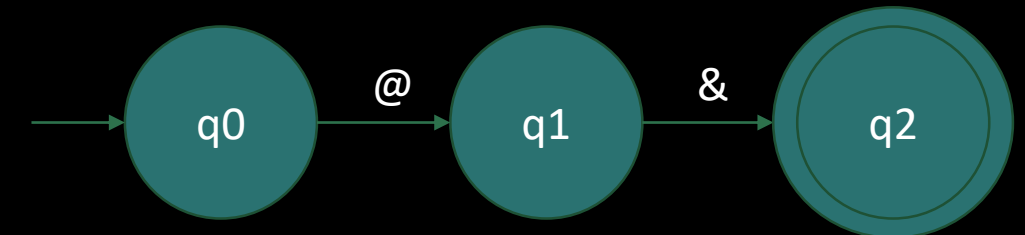
Operador Or

Expresión regular: @|



Operador and

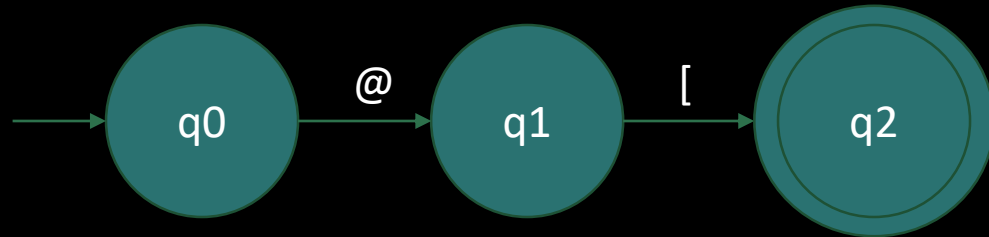
Expresión regular: @&



SÍMBOLOS ESPECIALES

Corchete abierto

Expresión regular: @[



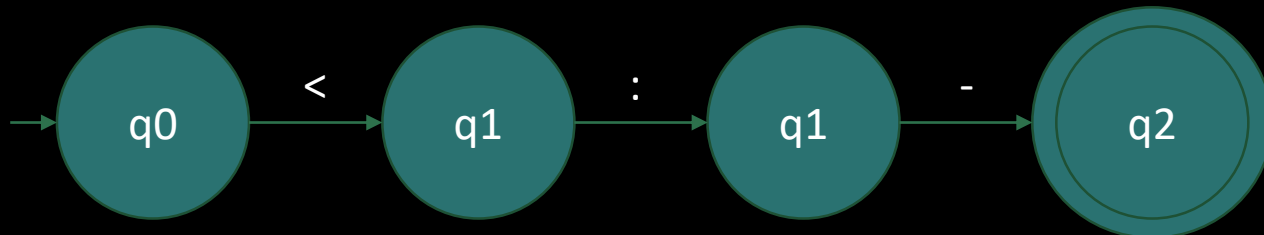
Corchete cerrado

Expresión regular: @]



Igual (asignación)

Expresión regular: <:-



Mayor

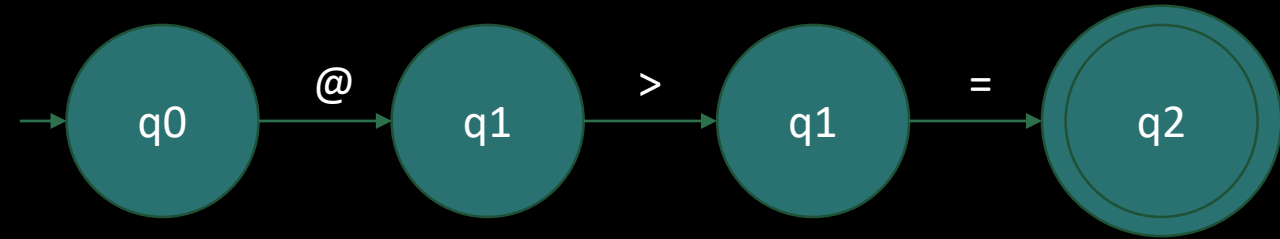
Expresión regular: @>



SÍMBOLOS ESPECIALES:

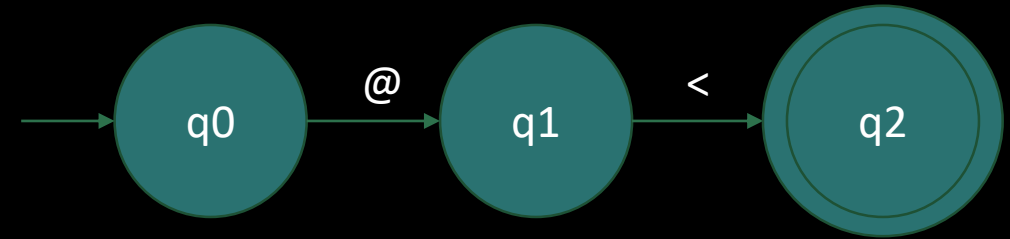
Mayor igual

Expresión regular: @>=



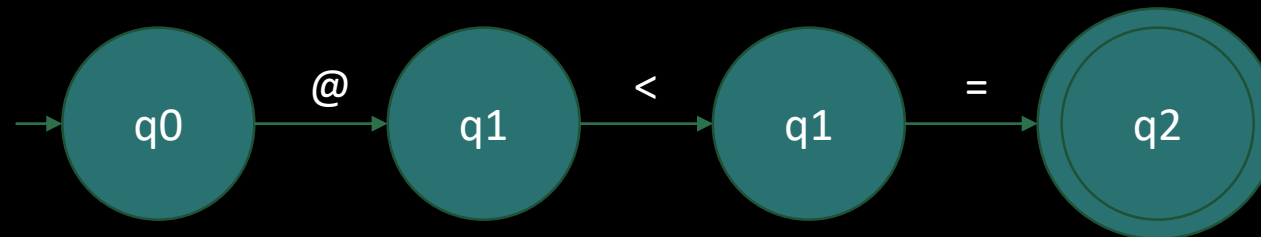
Menor

Expresión regular: @<



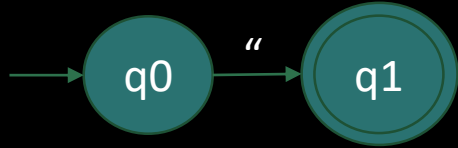
Menor igual

Expresión regular: @<=

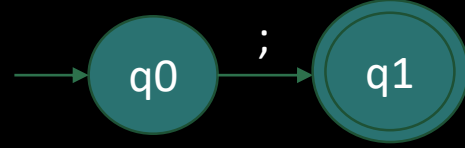


SÍMBOLOS PARA TODOS LOS PROYECTOS

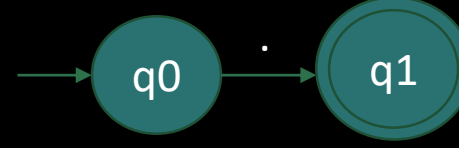
Doble comillas: "



Punto y coma: ;



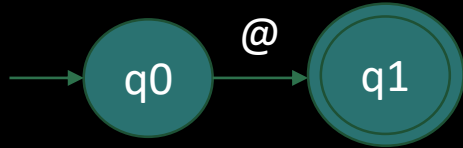
Punto: .



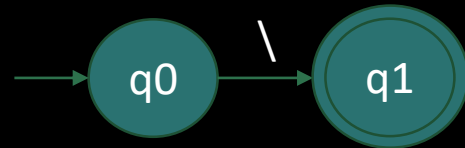
Signo \$: \$



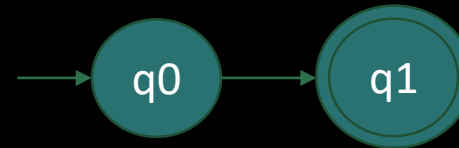
Arroba: @



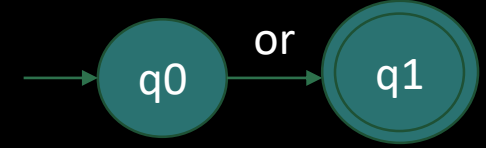
*Diagonal inversa: *



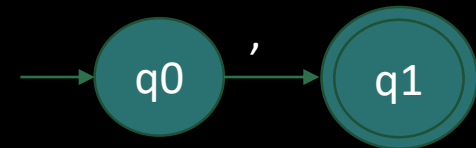
Espacio:



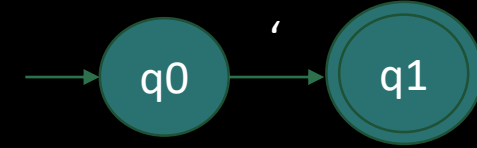
or



Coma: ,



Comilla simple: '



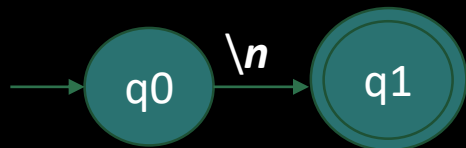
Dos puntos: :



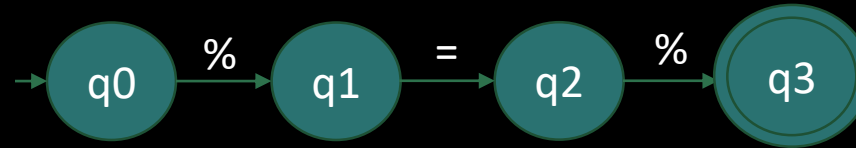
Negación: !



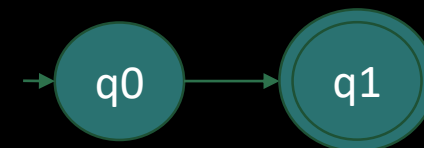
Brinco de línea: \n



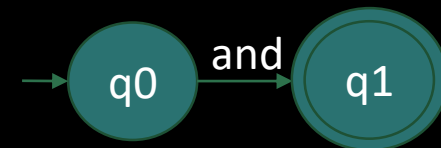
Igual comparación: %=%



Tabulador:

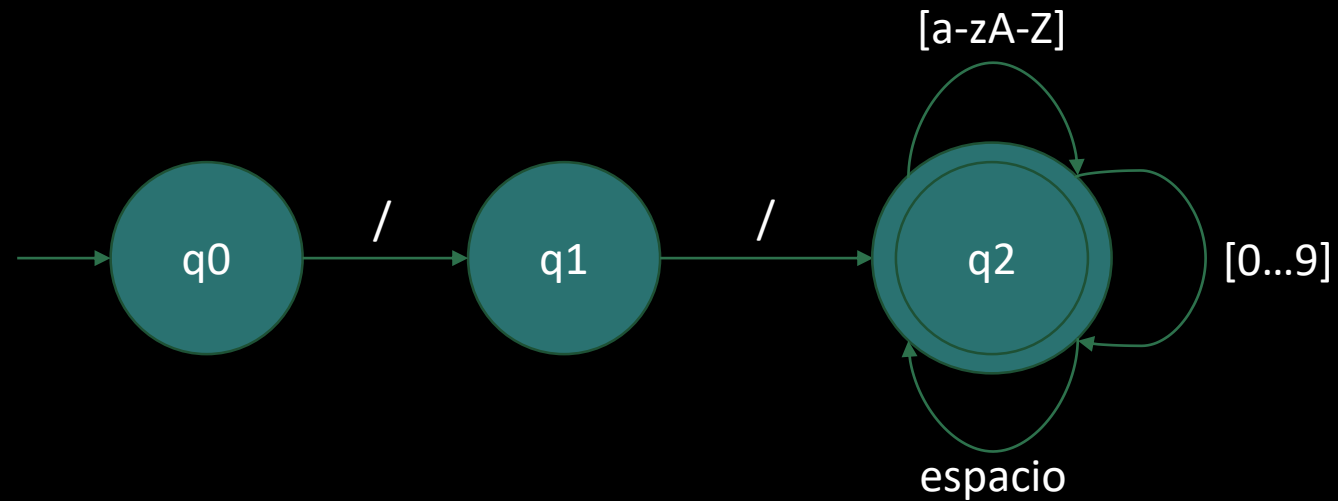


and

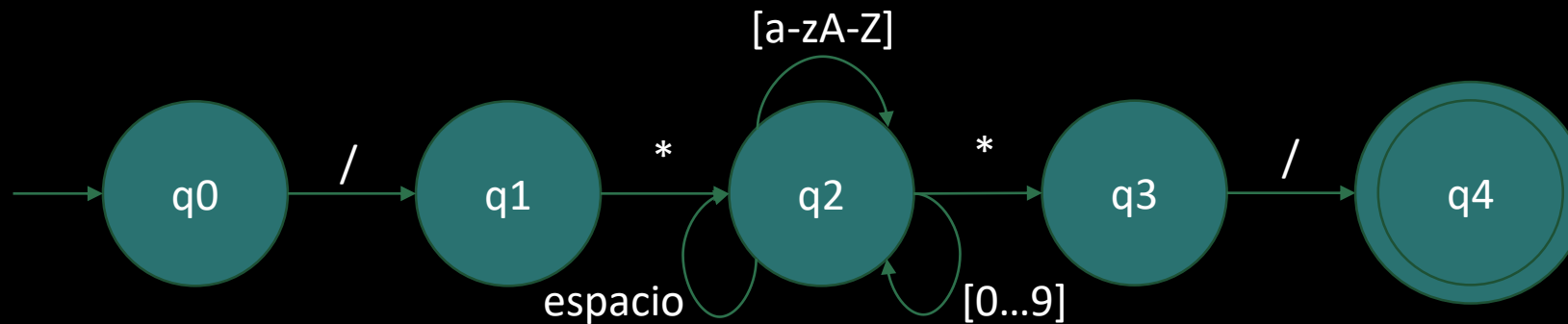


COMENTARIOS

- Para una línea. Expresión regular: `/(letra | digito | espacio)*`

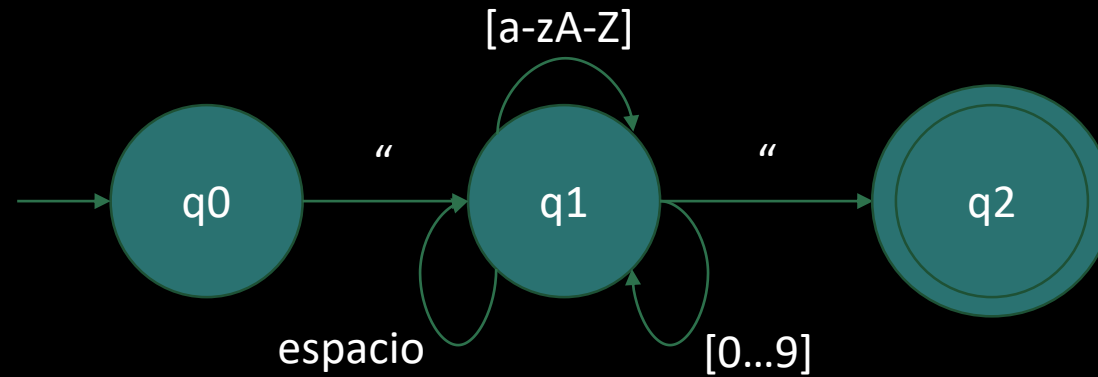


- Para varias líneas. Expresión regular: `/*(letra | digito | espacio)**/`

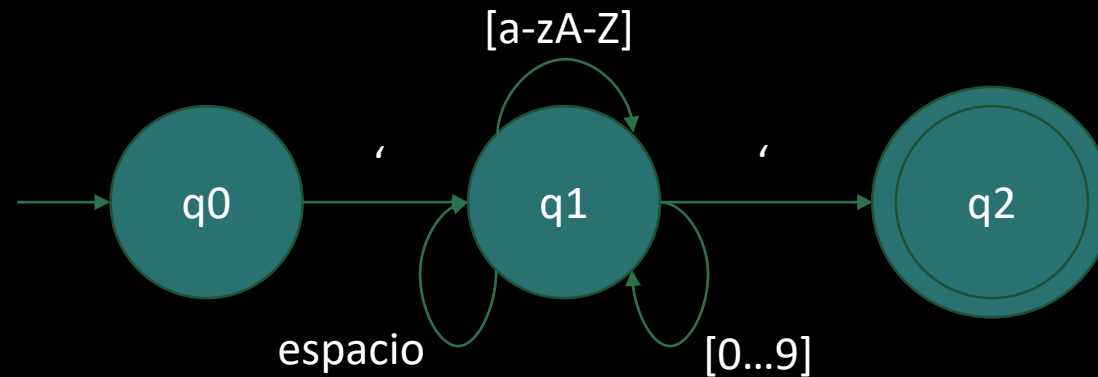


TEXTO

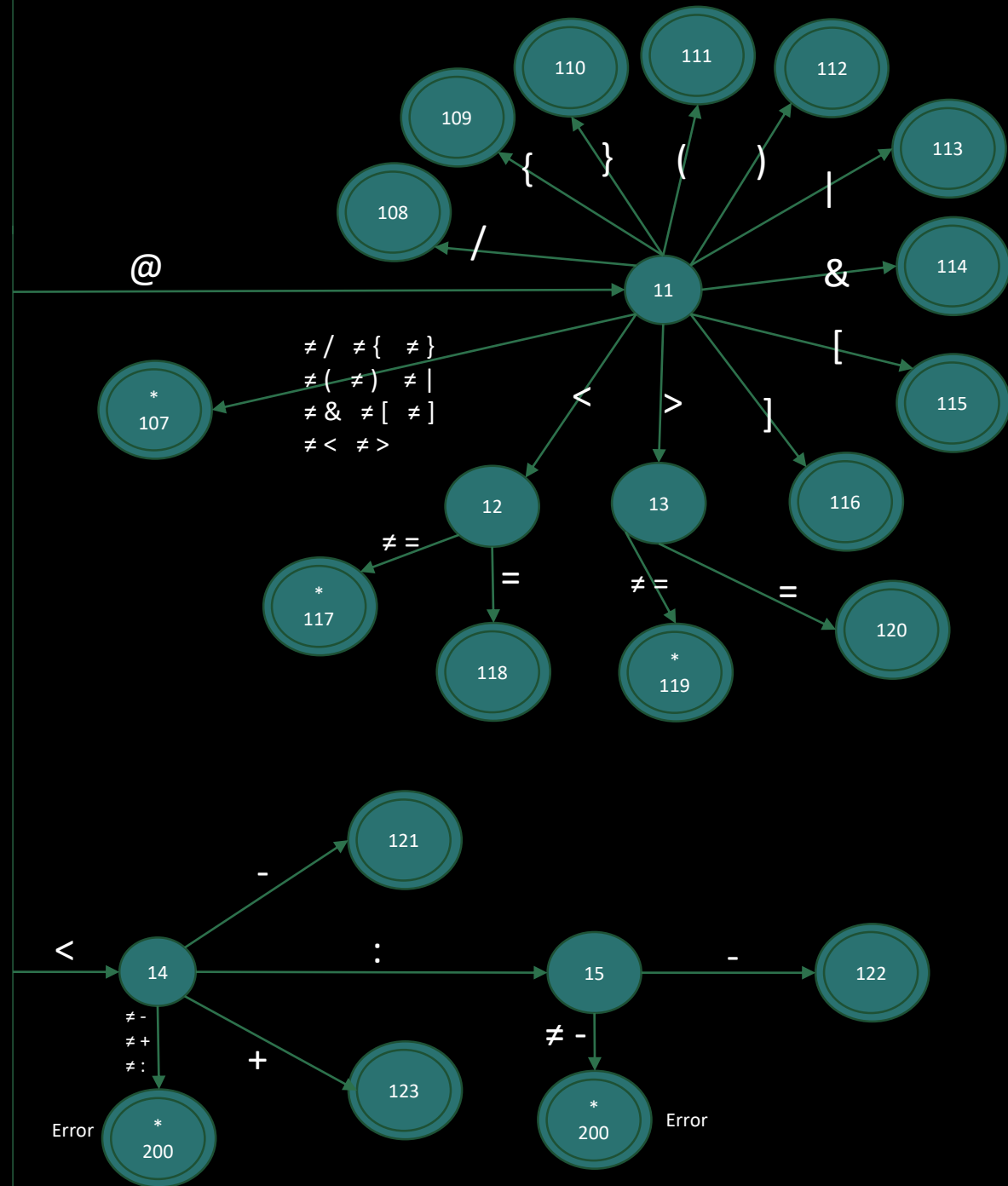
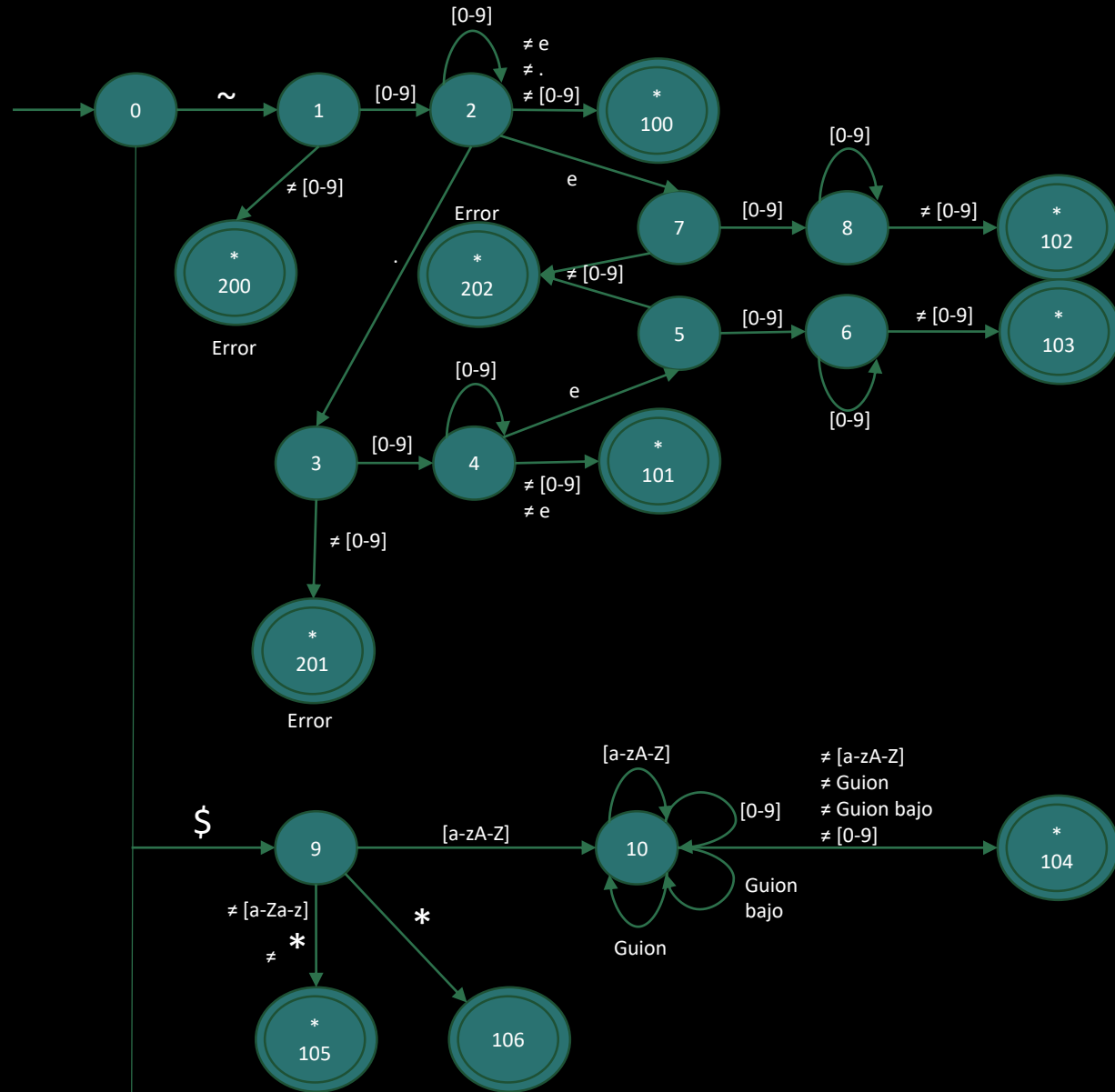
- Forma de escribir texto 1: Expresión regular: `"(letra | digito | espacio)*"`

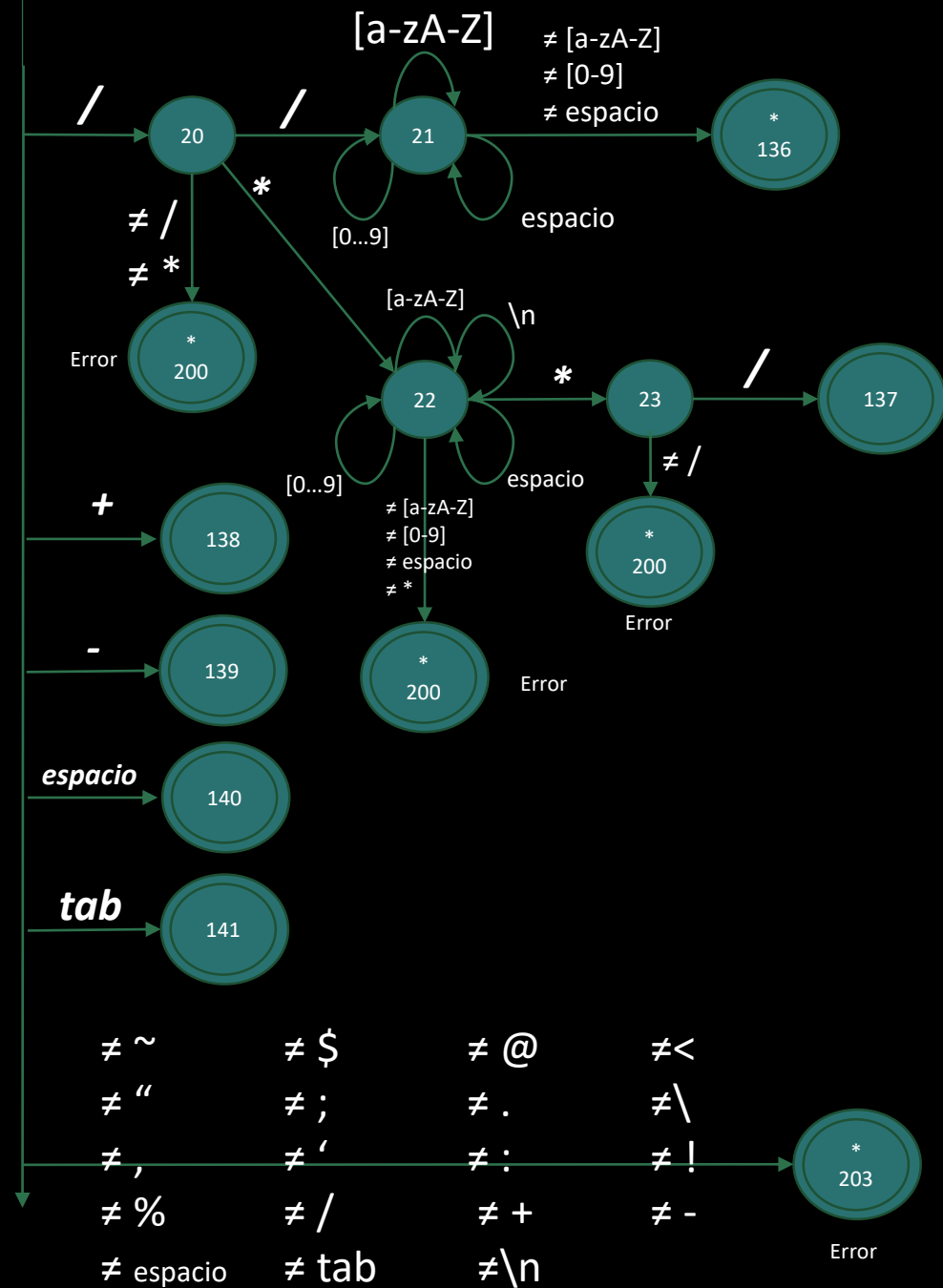
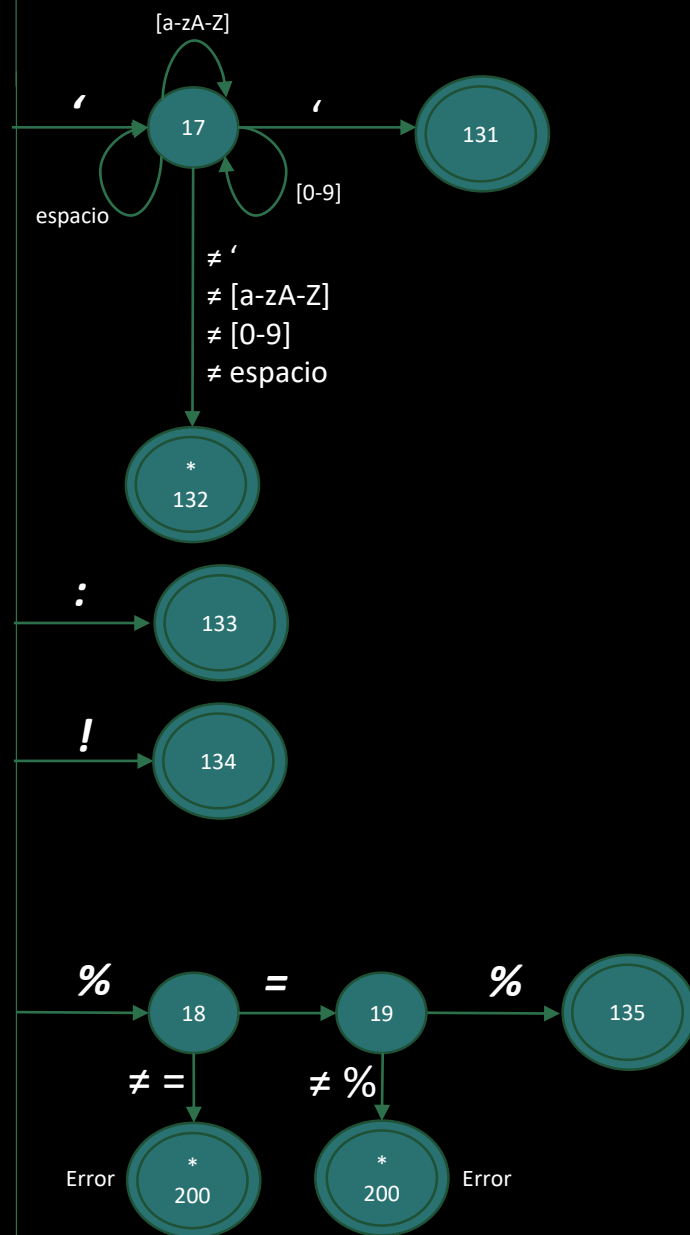
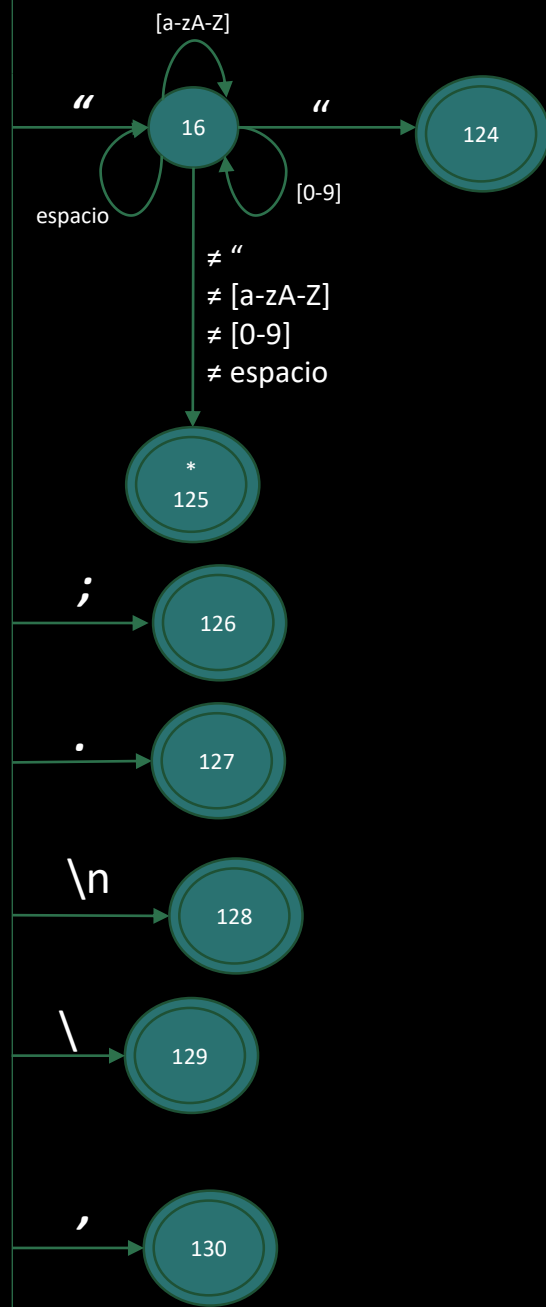


- Forma de escribir texto 2: Expresión regular: `'(letra | digito | espacio)*'`



AUTÓMATA INTEGRAL





LISTA DE PALABRAS RESERVADAS:

- | | |
|------------------------|------------------------|
| • Condición: Suppose | Ciclo inicio: enter |
| • Ciclo while: while | Ciclo final: Finishing |
| • Ciclo Repeat: Repeat | Impresión: Out |
| • Ciclo for: for | Lectura: Get |

[illegible]

DESCRIPCIÓN DE ESTADOS FINALES Y ESTADOS ERROR:

- 100: Se encuentra un numero natural
- 101: Se encuentra un numero decimal
- 102: Se encuentra un numero natural con notación científica
- 103: Se encuentra el numero decimal con notación científica
- 104: Se encuentra una variable o palabra reservada
- 105: Se encuentra el símbolo \$
- 106: Se encuentra una multiplicación
- 107: Se encuentra un arroba

- 108: Se encuentra una división
- 109: Se encuentra una llave que abre
- 110: Se encuentra una llave que cierra
- 111: Se encuentra un paréntesis que abre
- 112: Se encuentra un paréntesis que cierra
- 113: Se encuentra un condicional OR
- 114: Se encuentra un condicional AND
- 115: Se encuentra un corchete que abre
- 116: Se encuentra un corchete que cierra
- 117: Se encuentra un menor que
- 118: Se encuentra un menor o igual que
- 119: Se encuentra un mayor que
- 120: Se encuentra un mayor o igual que
- 121: Se encuentra un decremento
- 122: Se encuentra una asignación(igual)
- 123: Se encuentra un incremento
- 124: Se encuentra un texto en su primera forma
- 125: Se encuentran unas dobles comillas
- 126: Se encuentra un punto y coma
- 127: Se encuentra un punto
- 128: Se encuentra un salto de linea

- 129: Se encuentra una diagonal inversa
- 130: Se encuentra una coma
- 131: Se encuentra un texto en su segunda forma
- 132: Se encuentra una comilla simple
- 133: Se encuentran dos puntos
- 134: Se encuentra una negación
- 135: Se encuentra una comparación igual
- 136: Se encuentra un comentario de una línea
- 137: Se encuentra un comentario de varias líneas

- 138: Se encuentra una suma
- 139: Se encuentra una resta
- 140: Se encuentra un espacio
- 141: Se encuentra un tabulador

ESTADOS ERROR

- 200: Se encuentra un carácter o conjunto de caracteres que no es reconocido por el lenguaje
- 201: Se encuentra un numero decimal mal escrito
- 202: Se encuentra un numero natural o decimal con notación científica mal escrito
- 203: Se encuentra un carácter o conjunto de caracteres que no es reconocido por el lenguaje, pero este estado solo puede aparecer cuando se viene del estado 0.

DOCUMENTACIÓN DEL PROGRAMA:

```
1 import re#libreria para utilizar expresiones regulares
2
3 matrizdeestados = [[203, 2, 2, 4, 4, 6, 6, 8, 8, 105, 10, 107, 117, 119, 200, 200, 16, 17, 200, 200, 200, 21, 22, 200],
4 [140, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 16, 17, 200, 200, 200, 21, 22, 200],
5 [141, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
6 [203, 200, 7, 201, 5, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
7 [127, 200, 3, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
8 [1, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
9 [203, 200, 100, 201, 101, 202, 103, 202, 102, 10, 10, 107, 117, 119, 200, 200, 16, 17, 200, 200, 200, 21, 22, 200],
10 [9, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
11 [203, 200, 100, 201, 101, 202, 103, 202, 102, 106, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 22, 136, 23, 200],
12 [14, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 12, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
13 [138, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 123, 200, 125, 132, 200, 200, 200, 136, 200, 200],
14 [139, 200, 100, 201, 101, 202, 103, 202, 102, 105, 10, 107, 117, 119, 121, 122, 125, 132, 200, 200, 200, 136, 200, 200],
15 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 10, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
16 [11, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
17 [20, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 108, 117, 119, 200, 200, 125, 132, 200, 200, 21, 136, 200, 137],
18 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 109, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
19 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 110, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
20 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 111, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
21 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 112, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
22 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 113, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
23 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 114, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
24 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 115, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
25 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 116, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
26 [133, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 15, 200, 125, 132, 200, 200, 200, 136, 200, 200],
27 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 13, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
28 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 118, 120, 200, 200, 125, 132, 19, 200, 200, 136, 200, 200],
29 [16, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 124, 132, 200, 200, 200, 136, 200, 200],
30 [126, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
31 [129, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
32 [130, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
33 [17, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 131, 200, 200, 200, 136, 200, 200],
34 [128, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 22, 200],
35 [134, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200],
36 [18, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 135, 200, 136, 200, 200],
37 [203, 200, 100, 201, 101, 202, 103, 202, 102, 105, 104, 107, 117, 119, 200, 200, 125, 132, 200, 200, 200, 136, 200, 200]]
38 #Está es la matriz de estados, cada lista es una columna, por ejemplo la primera columna son los digitos
```

Linea 1: Se importa una librería que te permite utilizar expresiones regulares.

Linea 3: Se establece la lista de listas que guarda la matriz de estados, está lista de listas funciona así: Cada lista contiene todos los estados de una columna en la matriz de estados, por ejemplo, la primera lista contiene todos los estados de los dígitos, o como dice en la matriz de estados de [0-9], si se revista su columna cada estado coincide, y así cada lista con su columna correspondiente.

FUNCIÓN ABRIRYGUARDAR():

```
39  
40 def abriryguardar():#Esta función se u  
41     f = open('codigofuente.txt', 'r')  
42     codigofuente = f.read()  
43     f.close()  
44     return codigofuente  
45
```

Linea 40: En esta linea comienza la función “abriryguardar()” esta función se encarga de abrir el archivó de texto, después guarda su contenido en la variable “codigofuente” y cierra el archivo para que cualquier otro proceso pueda usar el archivo, al final solo regresa la variable “codigofuente” que es donde tenemos todo el contenido del Archivo de texto.

FUNCIÓN OBTENERCOLUMNA(CARÁCTER, ESTADOANTERIOR):

Línea 46: En esta línea comienza la función “obtenercolumna(caracter, estadoanterior)”, esta función tiene la tarea de brindarnos a qué lista pertenece cada carácter, funciona de forma muy simple, si el carácter coincide con uno de los if, o con alguna de las dos expresiones regulares dentro de la función te retornará la lista que le pertenece.

Existe una situación especial, esta es la de el carácter “e” pues puede ser reconocido como una letra o como indicio de notación científica, para solucionar los problemas con esta situación el if principal solo reconoce letras, y si acaso el estado anterior es uno donde la “e” tiene importancia y puede indicar algo especial se retorna la lista que pertenece a la e, si no es así se retorna la lista asignada a las letras como si fuera cualquier otra letra.

Si el carácter no coincide con ninguno de los if se pasa a la lista de los símbolos no reconocidos.

```
46 def obtenercolumna(caracter, estadoanterior):#Recibe u
47     #Esta función nos ayuda a saber a qué columna pert
48     #dependiendo del carácter te llevará a una columna
49     #si el carácter no coincide con ninguno de los ind
50     #columna de los caracteres no identificados
51     if re.match("[0-9]", caracter):
52         return 0
53     if caracter == " ":
54         return 1
55     if caracter == " ":
56         return 2
57     if caracter == ".":
58         return 4
59     if caracter == "~":
60         return 5
61     if re.match("[a-zA-Z]", caracter):#la e es especia
62     #Entonces si el estado anterior es uno donde l
63     #si no se pasa como una letra más.
64     if caracter == "e" and estadoanterior == 2:
65         return 3
66     elif caracter == "e" and estadoanterior == 4:
67         return 3
68     return 6
69
70 if caracter == "$":
71     return 7
72 if caracter == "*":
73     return 8
74 if caracter == "<":
75     return 9
76 if caracter == "+":
77     return 10
78 if caracter == "-":
79     return 11
80 if caracter == "_":
81     return 12
82 if caracter == "@":
83     return 13
84 if caracter == "/":
85     return 14
86 if caracter == "{":
87     return 15
88 if caracter == "}":
89     return 16
90 if caracter == "(":
91     return 17
92 if caracter == ")":
93     return 18
94 if caracter == "|":
95     return 19
96 if caracter == "&":
97     return 20
98 if caracter == "[":
99     return 21
100 if caracter == "]":
101     return 22
102 if caracter == ":":
103     return 23
104 if caracter == ">":
105     return 24
106 if caracter == "=":
107     return 25
108 if caracter == "'":
109     return 26
110 if caracter == ";":
111     return 27
112 if caracter == "\\":
113     return 28
114 if caracter == ",":
115     return 29
116 if caracter == "'":
117     return 30
118 if caracter == "\n":
119     return 31
120 if caracter == "!":
121     return 32
122 if caracter == "%":
123     return 33
124     return 34
```

FUNCIÓN

ANALIZARCODIGOFUENTE

E(INFTXT,

MATRIZDEESTADOS):

```
125
126 def analizarcodigofuente(inftxt, matrizdeestados):
127
128     estado = 0#esta variable nos indicara en que es
129     contador = 0#nos deja saber en que caracter de
130     estadoanterior = 0#nos deja saber cual fue el e
131     inftxt += " "#añade un espacio final para acaba
132     numcaracteres = len(inftxt)#nos da el numero de
133
134     loqueseencuentra = ""#nos ayuda a guardar los c
135
```

Linea 126: En esta linea comienza la función “analizarcodigofuente(inftxt, matrizdeestados)”, esta función tiene el objetivo de analizar todo el código fuente y decirnos todo lo que encuentra, ya sea algo que si pertenece a nuestro lenguaje o un error.

“estado” nos ayuda para saber en que estado estamos.

“contador” nos ayuda a saber en que carácter estamos.

“estadoanterior” nos ayuda a saber cual fue el estado anterior.

La linea 131 añade un espacio al final de todo el contenido, para que nada quede sin analizar.

“numcaracteres” nos dice cuantos caracteres tiene el archivo que se lee.

“loqueseencuentra” va juntando los caracteres analizados y los imprime cuando se llega a un estado final.

```

Go Run Terminal Help
• ProyectoLexico2.py - ProyectoAnal

ProyectoLexico2.py • ProyectoLexico.py codigofuente.txt
ProyectoLexico2.py > analizarcodigofuente

135
136 while contador != numcaracteres:#El for itera sobre todos los caracteres de e
137
138     caracter = inftxt[contador]
139
140     columna = obtenercolumna(caracter, estadoanterior)#se obtiene la columna
141
142     loqueseencuentra += caracter#se va sumando el caracter a una cadena strin
143
144     estado = matrizdeestados[columna][estado]#dependiendo la matriz de estado
145
146     if estado >= 100:
147         #si el estado en el que estamos pertenece a un estado final entra a u
148         #en cada if se indica que es lo que se encontro, se imprime y se regr
149         #si el estado es uno que tenia asterisco en el automata integral se r
150         #si el estado es 200 o mayor se indica a que tipo de error pertenece,
151         if estado == 100:
152             contador-=1
153             estado = 0
154             cadenaencontrada = loqueseencuentra.strip()
155             print(f"Se encontro el numero natural: {cadenaencontrada}")
156             loqueseencuentra = ""
157         if estado == 101:
158             contador-=1
159             estado = 0
160             cadenaencontrada = loqueseencuentra.strip()
161             print(f"Se encontro el numero decimal: {cadenaencontrada[:-1]}")
162             loqueseencuentra = ""

```

El ciclo while en la linea 136 no se terminará hasta que se hayan analizado todos los caracteres en el archivo fuente, después se obtiene el carácter en la linea 138, luego se hace uso de la función “obtenercolumna” para saber a que columna pertenece el carácter tomado, continua añadiendo el carácter a la cadena lo “loqueseencuentra” y después se obtiene el nuevo estado, dependiendo de la columna y su estado en la matriz de estados.

Si el nuevo estado resulta ser un estado final se indicara que se encontró y se imprimirá, la cadena “loqueseencuentra” se vacía, dependiendo del estado final puede que se regrese a un carácter anterior o no, y se regresa al estado 0.

```

if estado == 104: #si se llega a este estado puede que sea una palabra reservada c
    #por eso se checa si coincide con una de las palabras reservadas, en caso de
    #si no se pasa como una variable normal
    contador -= 1
    estado = 0
    cadenaencontrada = loqueseencuentra[:-1].strip()
    if cadenaencontrada == "$suppose":
        print(f"Se encontro una indicación de condición: {cadenaencontrada}")
    elif cadenaencontrada == "$while":
        print(f"Se encontro una indicación de ciclo while: {cadenaencontrada}")
    elif cadenaencontrada == "$Repeat":
        print(f"Se encontro una indicación de ciclo Repeat: {cadenaencontrada}")
    elif cadenaencontrada == "$for":
        print(f"Se encontro una indicación de ciclo for: {cadenaencontrada}")
    elif cadenaencontrada == "$enter":
        print(f"Se encontro un inicio de ciclo: {cadenaencontrada}")
    elif cadenaencontrada == "$Finishing":
        print(f"Se encontro el final de un ciclo: {cadenaencontrada}")
    elif cadenaencontrada == "$Out":
        print(f"Se encontro una impresión: {cadenaencontrada}")
    elif cadenaencontrada == "$Get":
        print(f"Se encontro una lectura: {cadenaencontrada}")
    else:
        print(f"Se encontro la variable: {cadenaencontrada}")
    loqueseencuentra = ""

```

El estado 104 es un estado especial, por que quiere decir que se encontró un identificador, si el identificador encontrado coincide con una de las palabras reservadas se indica que se encontró esa palabra reservada y lo que significa. Si el identificador encontrado no coincide con ninguna de las palabras reservadas se toma como una variable.

```

392         if estado == 200:
393             contador-=1
394             estado = 0
395             print(f'ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: {loqueseencuentra}')
396             loqueseencuentra = ""
397         if estado == 201:
398             contador-=1
399             estado = 0
400             print(f'ERROR: Un numero decimal no puede quedar asi: {loqueseencuentra}, debe continuar con otro digito')
401             loqueseencuentra = ""
402         if estado == 202:
403             contador-=1
404             estado = 0
405             print(f'ERROR: Un numero natural o decimal con notacion cientifica no puede quedar asi: {loqueseencuentra[1:-1]}, debe seguir con un otro digito')
406             loqueseencuentra = ""
407         if estado == 203:#Este error se repite ya que cuando viene del estado 0 no necesita regresar al caracter anterior, pero en los otros casos si es neces
408             estado = 0
409             print(f'ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: {loqueseencuentra}')
410             loqueseencuentra = ""
411
412     estadoanterior = estado
413     contador+=1

```

Para finalizar la función se tienen los estados de error, estos indican el tipo de error que se encontró, el estado 200 y el estado 203 son iguales por que uno solo sale cuando se viene del estado 0 y el otro es dentro de cualquier estado, si se viene del estado 0 no se regresa al carácter anterior o se crea un bucle interminable, por eso mismo se hace la separación.

En la linea 412 se obtiene el ultimo estado que será el estado anterior en la siguiente iteración del ciclo y después se incrementa el contador para tomar el siguiente carácter.

LÍNEAS FINALES:

```
416     codigofuente = abriryguardar()#Lee el codigo fuente  
417     #print(codigofuente)imprime el codigo fuente para sa  
418     analizarcodigofuente(codigofuente, matrizdeestados)#
```

Para finalizar el código entero se tienen la línea 416 y 418, en la línea 416 se usa la función “abriryguardar()” para guardar toda la información de el archivo que se analizar en una variable.

En la línea 418 solamente se utiliza la función “analizarcodigofuente(codigofuente, matrizdeestados)” para realizar todo el proceso de análisis del archivo.

PRUEBAS

RESULTADO:

- El archivo "codigofuente.txt" contiene cada uno de los caracteres y símbolos reconocidos por el lenguaje para comprobar que los reconoce correctamente y funciona sin problemas.

```
codigofuente: Bloc de no
Archivo Edición Formato
~16
~15e42
~12.89
~10.57e45
$indentifi4-cador
$suppose
$while
$Repeat
$for
$enter
$Finishing
$Out
$Get
<+
<-
$*
@/
-
+
@{
@}
@ (
@)
@|
<
```

```
@&
@[
@]
<:-
@>
@>=
@<
@<=
"
;
.
$
@
\
,
:
!
%= %
//esto es un comentario
/*esto es un comentario de
varias lineas*/
"esto es un texto en primera forma"
'esto es un texto en segunda forma'
(
~15e
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\User\Desktop\Repositorio perso\Diseño de compiladores\
Diseño de compiladores\Proyecto AnalisisLexico-DanielSalazar-172891
Se encontro el numero natural: ~16
- Se encontro un salto de linea -
Se encontro el numero natural con notación científica: ~15e42
- Se encontro un salto de linea -
Se encontro el numero decimal: ~12.89
- Se encontro un salto de linea -
Se encontro el numero decimal con notación científica: ~10.57e45
- Se encontro un salto de linea -
Se encontro la variable: $indentifi4-cador
- Se encontro un salto de linea -
Se encontro una indicación de condición: $suppose
- Se encontro un salto de linea -
Se encontro una indicación de ciclo while: $while
- Se encontro un salto de linea -
Se encontro una indicación de ciclo Repeat: $Repeat
- Se encontro un salto de linea -
Se encontro una indicación de ciclo for: $for
- Se encontro un salto de linea -
Se encontro un inicio de ciclo: $enter
- Se encontro un salto de linea -
Se encontro el final de un ciclo: $Finishing
- Se encontro un salto de linea -
Se encontro una impresión: $Out
- Se encontro un salto de linea -
Se encontro una lectura: $Get
- Se encontro un salto de linea -
Se encontro un incremento: <+
- Se encontro un salto de linea -
Se encontro un decremento: <-
- Se encontro un salto de linea -
Se encontro una multiplicacion: $*
- Se encontro un salto de linea -
Se encontro una indicación de división: @/
- Se encontro un salto de linea -
Se encontro una resta: -
- Se encontro un salto de linea -
Se encontro una suma: +
- Se encontro un salto de linea -
Se encontro una llave abierta: @{
- Se encontro un salto de linea -
Se encontro una llave cerrada: @}
- Se encontro un salto de linea -
Se encontro un parentesis abierto: @(
- Se encontro un salto de linea -
Se encontro un parentesis cerrado: @)
- Se encontro un salto de linea -
Se encontro un OR: @|
- Se encontro un salto de linea -
Se encontro un AND: @&
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Se encontro un mayor que: @>
- Se encontro un salto de linea -
Se encontro un mayor o igual que: @>=
- Se encontro un salto de linea -
Se encontro un menor que: @<
- Se encontro un salto de linea -
Se encontro un menor o igual que: @<=
- Se encontro un salto de linea -
Se encontraron dobles comillas:
- Se encontro un salto de linea -
Se encontro un punto y coma: ;
- Se encontro un salto de linea -
Se encontro un punto: .
- Se encontro un salto de linea -
Se encontro el simbolo $: $
- Se encontro un salto de linea -
Se encontro un arroba: @
- Se encontro un salto de linea -
Se encontro una diagonal inversa: \
- Se encontro un salto de linea -
Se encontro un espacio -
- Se encontro un salto de linea -
Se encontro una coma: ,
- Se encontro un salto de linea -
Se encontro una comilla simple:
- Se encontro un salto de linea -
Se encontro un dos puntos: :
- Se encontro un salto de linea -
Se encontro una negación o símbolo de exclamación: !
- Se encontro un salto de linea -
Se encontro una comparación de iguales: %=%
- Se encontro un salto de linea -
Se encontro un tabulador -
- Se encontro un salto de linea -
Se encontro un comentario de una linea: //esto es un comentario
- Se encontro un salto de linea -
Se encontro un comentario de varias lineas: /*esto es un comentario de
varias lineas*/
- Se encontro un salto de linea -
Se encontro el texto (forma1): "esto es un texto en primera forma"
- Se encontro un salto de linea -
Se encontro el texto (forma2): 'esto es un texto en segunda forma'
- Se encontro un salto de linea -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: (
- Se encontro un salto de linea -
ERROR: Un numero natural o decimal con notacion cientifica no puede quedar asi: ~15e
, debe seguir con un otro dígito
- Se encontro un salto de linea -
ERROR: Un numero decimal no puede quedar asi: ~12.
, debe continuar con otro dígito
- Se encontro un salto de linea -
```

RESULTADO:

- El archivo “codigofuente2.txt” contiene un pequeño programa escrito en nuestro lenguaje para comprobar el funcionamiento y que cosas son admitidas y que da error, funcionando como debe ser.

```
codigofuente2: Bloc de notas
Archivo Edición Formato Ver Ayuda
$Out@("Programa para sumar restar dividir o multiplicar 2 numeros"@)

$n1 <:- $Get@("Introduce el primer numero: "@)
$n2 <:- $Get@("Introduce el segundo numero: "@)
$opcion <:- $Get@("Que quieres hacer con ellos? @(1 suma, 2 resta, 3 dividir, 4 multiplicar@): "@)

$suppose $opcion %=% 1:
    $n3 <:- $n1 + $n2
$suppose $opcion %=% 2:
    $n3 <:- $n1 - $n2
$suppose $opcion %=% 3:
    $n3 <:- $n1 @/ $n2
$suppose $opcion %=% 4:
    $n3 <:- $n1 $* $n2

$Out@("Tu operacion da como resultado: " $n3 @)
```

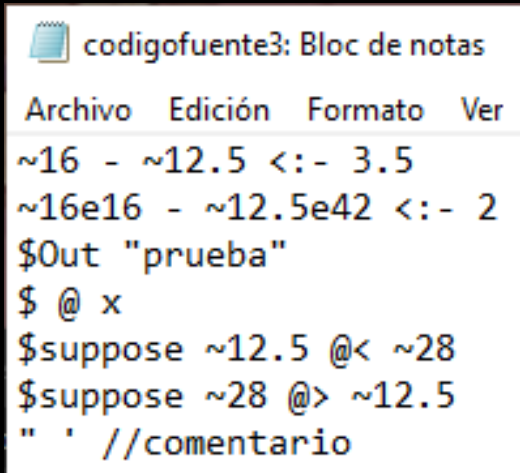
```
Se encontro una impresión: $Out
Se encontro un parentesis abierto: @(
Se encontro el texto (format1): "Programa para sumar restar dividir o multiplicar 2 numeros"
Se encontro un parentesis cerrado: @)
- Se encontro un salto de linea -
- Se encontro un salto de linea -
Se encontro la variable: $n1
- Se encontro un espacio -
Se encontro una asignación/igual: <:-
- Se encontro un espacio -
Se encontro una lectura: $Get
Se encontro un parentesis abierto: @(
Se encontraron dobles comillas: "Introduce el primer numero
Se encontro un dos puntos: :
- Se encontro un espacio -
Se encontraron dobles comillas: "
Se encontro un parentesis cerrado: @)
- Se encontro un salto de linea -
Se encontro la variable: $n2
- Se encontro un espacio -
Se encontro una asignación/igual: <:-
- Se encontro un espacio -
Se encontro una lectura: $Get
Se encontro un parentesis abierto: @(
Se encontraron dobles comillas: "Introduce el segundo numero
Se encontro un dos puntos: :
- Se encontro un espacio -
Se encontraron dobles comillas: "
Se encontro un parentesis cerrado: @)
- Se encontro un salto de linea -
Se encontro la variable: $opcion
- Se encontro un espacio -
Se encontro una asignación/igual: <:-
- Se encontro un espacio -
Se encontro una lectura: $Get
Se encontro un parentesis abierto: @(
Se encontraron dobles comillas: "Que quieres hacer con ellos
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: ?
- Se encontro un espacio -
Se encontro un parentesis abierto: @(
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 1
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: s
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: u
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: m
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: a
Se encontro una coma: ,
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 2
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: r

ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: e
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: s
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: t
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: a
Se encontro una coma: ,
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 3
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: d
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: i
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: v
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: i
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: d
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: i
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: r
Se encontro una coma: ,
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 4
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: m
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: u
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: l
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: t
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: i
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: p
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: l
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: i
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: c
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: a
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: r
Se encontro un parentesis cerrado: @)
Se encontro un dos puntos: :
- Se encontro un espacio -
Se encontraron dobles comillas: "
Se encontro un parentesis cerrado: @)
- Se encontro un salto de linea -
- Se encontro un salto de linea -
Se encontro una indicación de condición: $suppose
- Se encontro un espacio -
Se encontro la variable: $opcion
- Se encontro un espacio -
Se encontro una comparación de iguales: %=%
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 1
Se encontro un dos puntos: :
- Se encontro un salto de linea -
- Se encontro un tabulador -
Se encontro la variable: $n3
- Se encontro un espacio -
Se encontro una asignación/igual: <:-
- Se encontro un espacio -
```

- Lo que se imprime es mucho, pero en esto ya se puede ver que funciona como se esperaba.

RESULTADO:

El archivo “codigofuente3.txt” se diseña para ver que los estados finales con asterisco si regresen a un carácter anterior y siguen reconociendo los símbolos, números y todo sin problemas.



```
Archivo Edición Formato Ver
~16 - ~12.5 <:- 3.5
~16e16 - ~12.5e42 <:- 2
$Out "prueba"
$ @ x
$suppose ~12.5 @< ~28
$suppose ~28 @> ~12.5
" ' //comentario
```

```
Se encontro el numero natural: ~16
- Se encontro un espacio -
Se encontro una resta: -
- Se encontro un espacio -
Se encontro el numero decimal: ~12.5
- Se encontro un espacio -
Se encontro una asignación/igual: <:-
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 3
Se encontro un punto: .
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 5
- Se encontro un salto de linea -
Se encontro el numero natural con notación científica: ~16e16
- Se encontro un espacio -
Se encontro una resta: -
- Se encontro un espacio -
Se encontro el numero decimal con notación científica: ~12.5e42
- Se encontro un espacio -
Se encontro una asignación/igual: <:-
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: 2
- Se encontro un salto de linea -
Se encontro una impresión: $Out
- Se encontro un espacio -
Se encontro el texto (formal): "prueba"
- Se encontro un salto de linea -
Se encontro el simbolo $: $
- Se encontro un espacio -
Se encontro un arroba: @
- Se encontro un espacio -
ERROR: El siguiente caracter o conjunto de caracteres no se reconoce: x
- Se encontro un salto de linea -
Se encontro una indicación de condición: $suppose
- Se encontro un espacio -
Se encontro el numero decimal: ~12.5
- Se encontro un espacio -
Se encontro un menor que: @<
- Se encontro un espacio -
Se encontro el numero natural: ~28
- Se encontro un salto de linea -
Se encontro una indicación de condición: $suppose
- Se encontro un espacio -
Se encontro el numero natural: ~28
- Se encontro un espacio -
Se encontro un mayor que: @>
- Se encontro un espacio -
Se encontro el numero decimal: ~12.5
- Se encontro un salto de linea -
Se encontraron dobles comillas: "
Se encontro una comilla simple: '
Se encontro un comentario de una linea: //comentario
```