



PROYECTO 2



MANUAL TÉCNICO

# PARSER LFP

---

**Realizado por:**

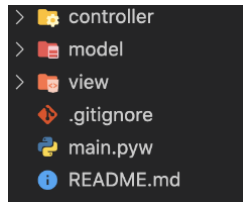
Daniel Estuardo Cuque Ruíz  
Lenguajes formales y de programación  
Sección A-

# INDICE

<i>Lógica del programa .....</i>	<i>3</i>
<i>Diseño del AFD .....</i>	<i>4</i>
<i>Método del árbol .....</i>	<i>4</i>
<i>Tipos de tokens .....</i>	<i>5</i>
<i>Gramática .....</i>	<i>6</i>

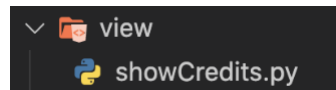
# Lógica del programa

El programa se divide de la siguiente manera:

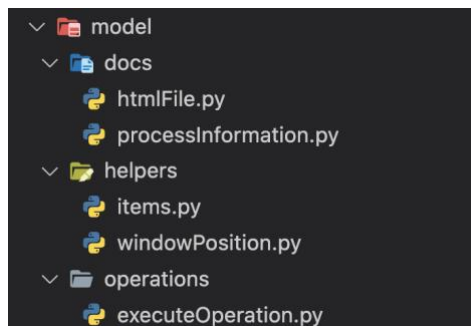


- **Vista:** en esta capa se guardarán todas las interfaces para que el usuario pueda interactuar con el programa.

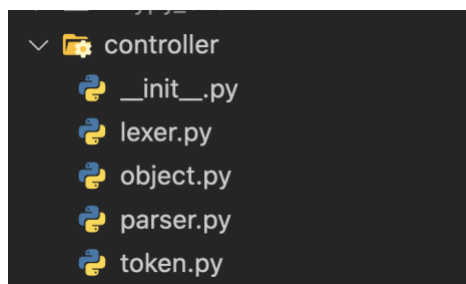
Para nuestra aplicación, solo contamos con una única vista, que nos servirá para mostrar los créditos.



- **Modelo:** en esta capa se guardarán todas las clases que ayudarán a construir la lógica de nuestro programa, lógica, funciones extra, etc.

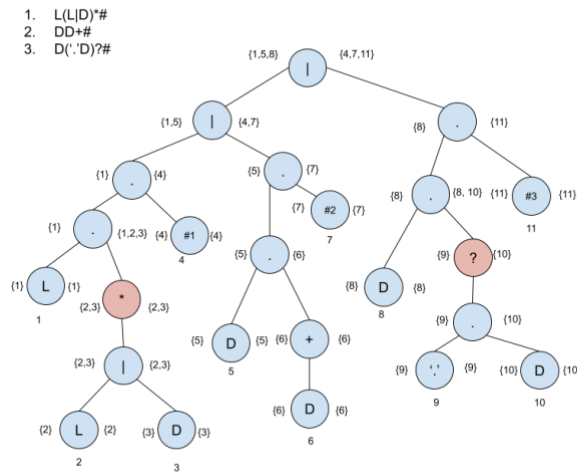


- **Controlador:** en esta capa, se almacenarán todas las clases base para la creación del lexer.



# Diseño del AFD

## Método del árbol



Elemento	Follow
1-L	2,3,4
2-L	2,3,4
3-D	2,3,4
4-#	
5-D	6
6-D	7
7-#	
8-D	9,11
9-'	10
10-D	9,11
11-#	

Daniel Estuardo Cuque Ruiz  
 202112145

Existirán 10 estados de aceptación

- Números
- Etiqueta de apertura
- Etiqueta de cierre
- Texto
- Punto y coma (;)
- Coma (,)
- Paréntesis izquierdo
- Paréntesis derecho

La expresión regular para los casos anteriores está dada por:

- $D+(\backslash.D)^*$
- $(\langle L+\rangle)(\langle /L+\rangle)(\langle L((L+=D+)|(L+=L+))+/\rangle)$
- $L+$

Su estado inicial:

$S = q_0$

Los estados de transición son los siguientes:

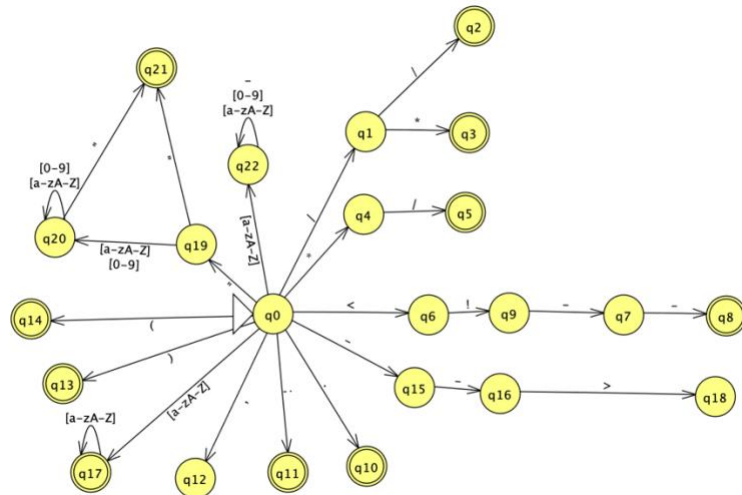
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, \dots, q_{22}\}$

Los estados de aceptación son los siguientes:

$F = \{q_2, q_3, q_5, q_8, q_{10}, q_{11}, q_{17}, q_{13}, q_{14}, q_{21}\}$

$$\begin{aligned}(q_0, D) &= q_1 \\ (q_0, L) &= q_2 \\ (q_0, <) &= q_3\end{aligned}$$
$$\begin{aligned}(q1, D) &= q1 \\ (q1, \backslash.) &= q8 \\ (q8, D) &= q1\end{aligned}$$
$$(q_2, L) = q_2$$
$$\begin{aligned}(q3, /) &= q6 \\ (q3, L) &= q4\end{aligned}$$

(q4, /) = q6  
(q4, =) = q7  
(q4, >) = q5

$$\begin{aligned}(q6, >) &= q5 \\ (q6, L) &= q4\end{aligned}$$
$$\begin{aligned}(q7, L) &= q4 \\ (q7, D) &= q4\end{aligned}$$


## Tipos de tokens

Todos los tokens, guardarán su posición (fila y columna) dentro de la cadena de entrada.

Tipo	Patrón
BOOLEAN	('true' 'false')
COMMA	','
CONTROL	'Boton' 'Etiqueta' '..'' 'AreaTexto'
CLOSE_TAG	'→'
DOT	'.'
FUNCTION	'set'L+
IDENT	L+(D+ L+)*
INT	D+
LPAREN	'('
OPEN_TAG	'<!--'
SEMICOLON	','
STRING	'\"(D* L*)* \"'
RPAREN	)'
WRAPPER	'Controles'   'Propiedades'   'Colocacion'

# Gramática

## No terminales

**<START>** ::= <OPEN\_TAG> <WRAPPER> <LET\_STATEMENT> <WRAPPER> <CLOSE\_TAG>  
<OPEN\_TAG> <WRAPPER> <CALL\_FUNCTION> <WRAPPER> <CLOSE\_TAG> <OPEN\_TAG>  
<WRAPPER> <CALL\_FUNCTION> <WRAPPER> <CLOSE\_TAG>

**<LET\_STATEMENT>** ::= <CONTROL> <IDENT> <SEMICOLON> <LET\_STATEMENT> |  $\epsilon$

**<CALL\_FUNCTION>** ::= <IDENT> <DOT> <FUNCTION> <LPAREN> <PARAMETERS> <RPAREN>  
<SEMICOLON> <CALL\_FUNCTION> |  $\epsilon$

**<PARAMETERS>** ::= <BOOLEAN> <PARAMETER> | <INT> <PARAMETER> | <STRING> <PARAMETER>  
| <JUSTIFY> <PARAMETER> | <IDENT> <PARAMETER>

**<PARAMETER>** ::= <COMMA> <PARAMETERS> |  $\epsilon$

## Terminales

**<BOOLEAN>** ::= 'true' | 'false'

**<COMMA>** ::= ',' |  $\epsilon$

**<CONTROL>** ::= 'Boton' | 'Etiqueta'

**<CLOSE\_TAG>** ::= '—>'

**<DOT>** ::= '.'

**<FUNCTION>** ::= 'set'L+

**<IDENT>** ::= L+DL

**<INT>** ::= D+

**<LPAREN>** ::= '('

**<OPEN\_TAG>** ::= <!--

**<SEMICOLON>** ::= ';'

**<STRING>** ::= "\"L+\""

**<RPAREN>** ::= ')'

**<WRAPPER>** ::= 'Controles' | 'Propiedades' | 'Colocacion'