

# MANUAL TÉCNICO

## Información del sistema:

- OS: Ubuntu
- CPU: Intel core i5 2.50 GHz
- RAM: 8 GB
- Versión de java: 17.0.8
- IDE: IntelliJ idea
- Control de versiones: git 2.34.1
- Github: <https://github.com/Hatsune02/LenguajesFormales1.git>

## Descripción del proyecto:

Una de las tareas de un analizador léxico es la detección de símbolos que no forman parte de un lenguaje y también el reconocimiento de tokens válidos para el lenguaje en cuestión. Sin embargo el analizador léxico debe reaccionar correctamente ante los posibles errores que se encuentren, indicando la posición del error y una descripción del mismo.

## Alfabeto:

- L = Letras (a-z)(A-Z) (no se cuenta la ñ).
- N = Dígitos del 0 - 9.
- \_ = Guión bajo.
- S = Signos (+,-,/,\*,%).
- C = Caracteres Comparativos (=,!,>,<).
- Co = Caracter para comentario (#).
- O = Otros caracteres (“[, ], {, }, (, ), ;, :, ,”).
- V = Vacíos (espacio, salto de línea, tabulación).
- Ca = Caracter de cadena (Comillas simples y dobles).
- D = Delimitadores que contienen algunos caracteres ya establecidos.  
(S,C,Co,O,Ca,V).
- P = punto (.).
- E = No pertenece al alfabeto.

## Tokens:

- Identificador: Palabras que inician con un caracter de letra o guión bajo, y después tiene que ser cualquier cantidad de letras, guiones bajos y números.
- Aritmético: Ser un signo aritmético.
- Comparación: Ser un signo de comparación (==,!,>,<,!>,>=,<=).
- Lógico: Son palabras reservadas para propósito lógico:
  - and
  - or
  - not

- True
- False
- Asignación: Ser un símbolo para asignar (=) o combinarse con cualquier signo aritmético y luego el símbolo para asignar (+=, -=, \*=, /=, %=).
- Palabra clave: Son palabras con un patrón fijo y estrico.
  - and
  - as
  - assert
  - break
  - class
  - continue
  - def
  - del
  - elif
  - else
  - except
  - False
  - finally
  - for
  - from
  - global
  - if
  - import
  - in
  - is
  - lambda
  - None
  - nonlocal
  - not
  - or
  - pass
  - raise
  - return
  - True
  - try
  - while
  - with
  - yield
- Número: Son palabras que contienen 1 o más caracteres de tipo dígito.
- Decimal: Son palabras que contienen 1 o más caracteres de tipo dígito, seguidos de un punto y luego 1 o más caracteres de tipo dígito.
- Cadena: Es una cadena de caracteres que estén encerrados entre unas comillas simples o dobles.
- Comentario: Es una cadena de caracteres que empiecen con el símbolo de comentario (#).
- Otros: Símbolos que no entran en alguna descripción anterior (“[, ], {, }, (, ), ::, :, ,”).

## Expresiones regulares:

`_?[a-zA-z]+_?\w*`

RegEx identificadores.

`(\w)?(\*|\*?)?|[-+\w%]?`

RegEx aritmético.

`[!><]=?(==)`

RegEx comparación.

`(and)|(or)|(not)|(True)|(False)`

RegEx lógicos.

`=|[-+\w]=`

RegEx asignación.

`[0-9]+`

RegEx Número.

`[0-9]+\.[0-9]+`

RegEx Decimal.

`".[^\"]+\"|.^[^']+'`

RegEx Cadenas.

`#.*`

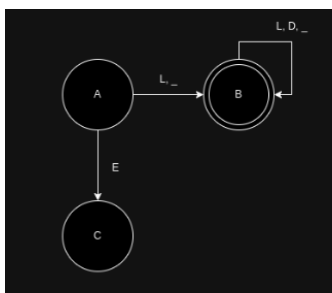
RegEx Comentarios.

`[\[\]\(\)\{\}\.\,\;\:]`

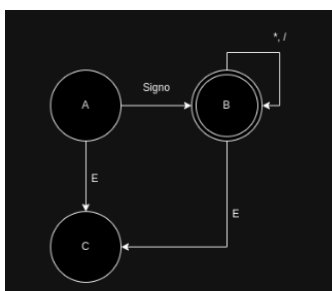
RegEx Otros.

## Autómatas:

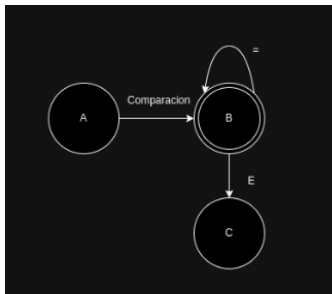
- **Identificador:**



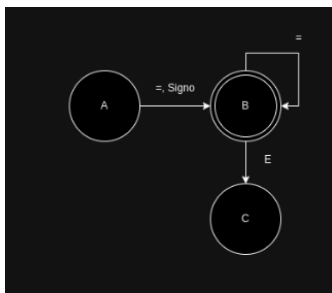
- **Aritmético:**



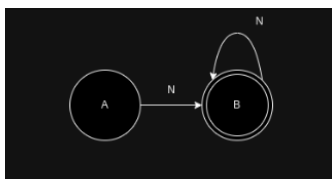
- **Comparación:**



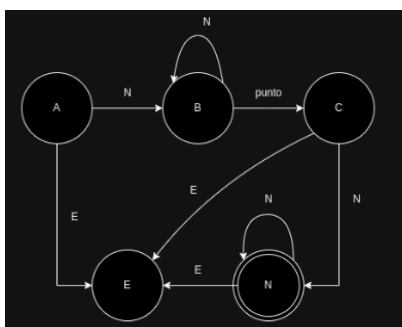
- **Asignación:**



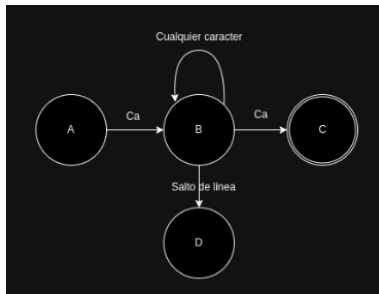
- **Número:**



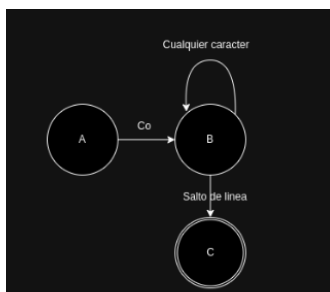
- **Decimal:**



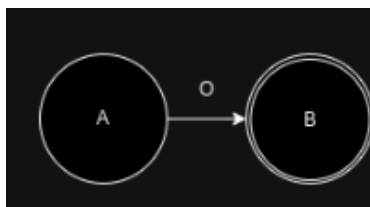
- **Cadena:**



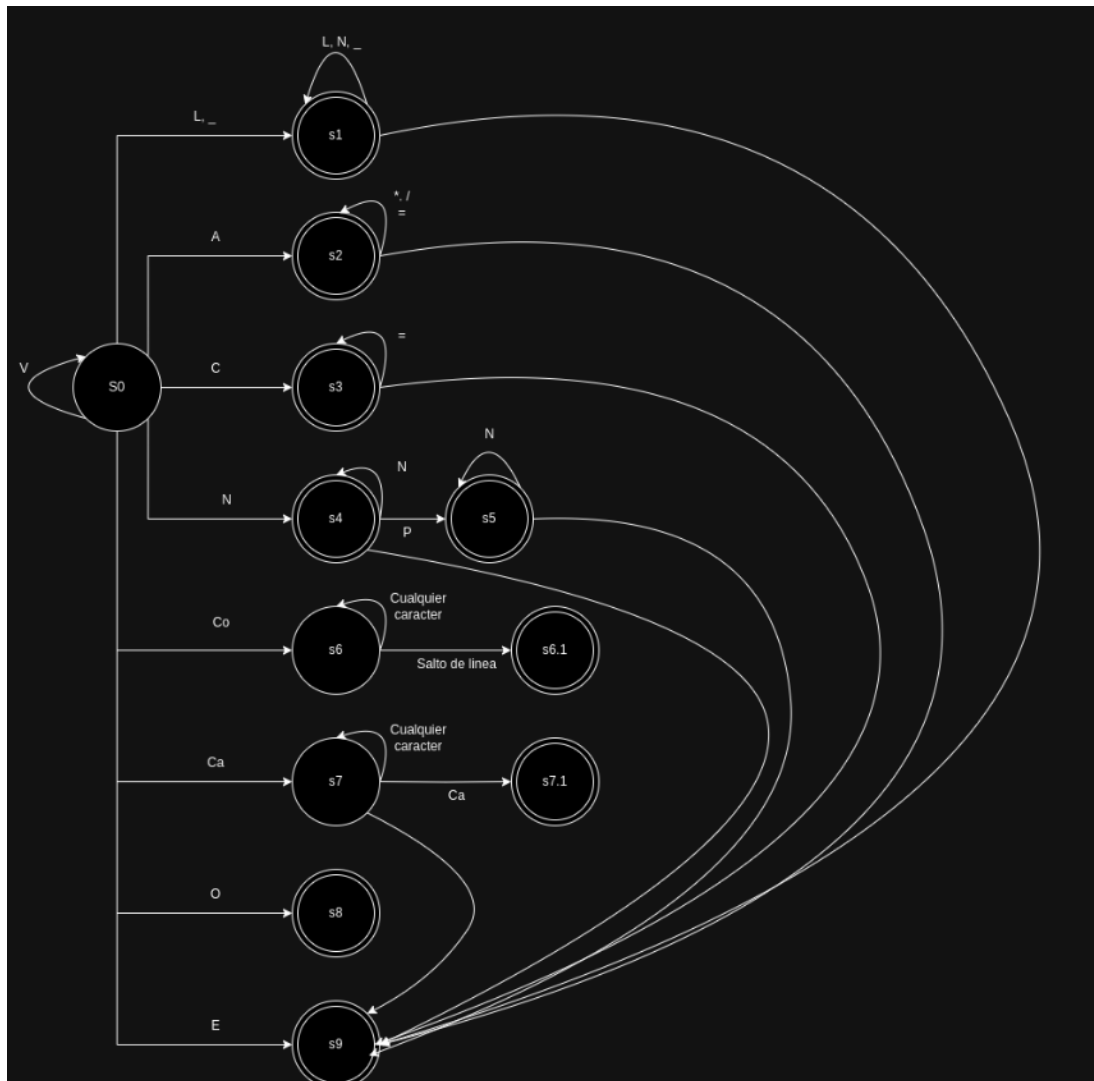
- **Comentario:**



- **Otros:**



## Autómata general:



## Definición formal:

$$\begin{aligned}
 I &= (Q, \Sigma, \delta, q_0, F) \\
 Q &= \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\} \\
 \Sigma &= \{L, N, \_, S, C, Co, O, V, Ca, D, P, E\} \\
 F &= \{s_1, s_2, s_4, s_5, s_6, s_7\} \\
 q_0 &= s_0
 \end{aligned}$$

## Función de transición $\delta(Q, \Sigma)$ :

[illegible]