

A continuación se modela un grupo de expresiones regulares para poder identificar un patrón de comportamiento de sus respectivos tokens.

Empezando por los identificadores, estos empiezan por una letra o un guión bajo seguido de al menos un dígito o una letra, a ambas posibilidades le puede seguir cero o muchas veces un guión bajo, una letra o un número.

## Identificadores

```
((_+([a-zA-Z]|[0-9])+|[a-zA-Z])(_|[a-zA-Z]|[0-9]))*)
```

El decimal puede o no traer un guion medio que en es interpretado como un signo de negatividad, seguido de cualquier dígito entre uno y nueve seguido de ninguno o varios dígitos entre cero y nueve, sinó, un solo cero; continuando con un punto y al menos un dígito entre cero y nueve.

## Decimal

```
(((-?[1-9][0-9]*)|0)(\.([0-9]+)))
```

El entero puede o no traer un guion medio que en es interpretado como un signo de negatividad, seguido de cualquier dígito entre uno y nueve seguido de ningún o varios dígitos entre cero y nueve; sinó, un solo cero.

## Enteros

$$((-?[1-9][0-9]*)|0)$$

Las cadenas aceptan una comilla doble o simple, seguido de cualquier carácter que no sea una secuencia de escape, ya sea nueva línea o retorno de carro. Finaliza con otra comilla simple o doble.

Cadenas

```
("([^\n\\]|\\.)*"|'([^'\n\\]|\\.)*')
```

Los comentarios reciben un símbolo de numeral, seguido de cualquier carácter que no sea las secuencias de escape de retorno de carro o nueva línea con los cuales termina el patrón.

## Comentarios

$$(\#([^\backslash r \backslash n]^*) \backslash n)$$

Los aritmeticos son una secuencia de Or's para cada uno de ellos.

# Aritmeticos

(\+|\-|\\*\\*|\\*|\/|\/|\/|\/|%)

Para comparacion se aplica la misma idea que en los aritmeticos.

## Comparacion

$$(\backslash = \backslash = | \backslash ! \backslash = | \backslash < \backslash = | \backslash > \backslash = | \backslash < | \backslash >)$$

Asignación repite lo mismo de los dos casos anteriores.

## Asignacion

(\+=|\-=|\\\*\\\*=|\\\*|=|\\/\\/\\/=|\\/\\/=|\\%\\=|\\=)

Y es 'otros' quienes finalizan este procedimiento llenos de or's

## Otros

(\(|\)|\{|\}\| |\,|\;|\:)

Las palabras claves para no tener que hacer lo mismo que con los ultimos cuatro casos, se aprovechó que estas cumplen la expresion de los identificadores por lo que para hacer mas escalable el desarrollo, cuando se vaya a realizar el registro de un identificador este se verificará si está en la lista de palabras reservadas, si es el caso se hacen los cambios correspondientes y se guarda el token.



Automata:

