

Informe Etapa 1

Compiladores e Interpretes

Nahuel Ignacio Fuentes

August 24, 2023

Tokens y sus Expresiones regulares asociadas

Por simplificación asociaremos a estas expresiones regulares las siguientes abreviaciones, con el fin de hacer la expresión resultante más legible :

```
Digito = [0..9]
Letra = [a..z] | [A..Z]
Letra_Mayus = [A..Z]
Letra_Minus = [a..z]
Caracter = [Letra | Digito | , | . | ; | ( | .. | ] ]
```

- Identificador de clase

```
idClase = Letra_Mayus [ Letra | Digito | _ ]*
```

- Identificador de método y variable

```
idMetVar = Letra_Minus [ Letra | Digito | _ ]*
```

- Enteros

```
intLiteral = Digito ^ n , 1 <= n <= 9
```

- Floats

```
floatLiteral = intLiteral . intLiteral
```

- Caracteres

```
charLiteral = ' [[Character - { \ , ' } ] | [ \ Character ] ] '
```

- String

```
stringLiteral = " [ Character - { \ , ' , \n } | [ \" ] ] * "
```

- Puntuación

```
openPar = (  
closePar = )  
openCurl = {  
closeCurl = }  
period = .  
comma = ,  
semiColon = ;
```

- Operadores

```
opAdd = +  
opSub = -  
opProd = *  
opIntDiv = %  
opDiv = /  
opLess = <  
opLessEq = <=  
opEq ==  
opGreater = >  
opGreaterEq = >=  
opAnd = &&  
opOr = ||  
opNot = !  
opNotEq = !=
```

- Asignacion

```
assign = =  
assignAdd = +=  
assignSub = -=
```