

Informe Etapa 1

Compiladores e Interpretes

Nahuel Ignacio Fuentes

August 21, 2023

Tokens y sus Expresiones regulares asociadas

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

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

- Identificador de clase

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

- Identificador de metodo y variable

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

- Enteros

```
intLiteral = Digito [ Digito* Digito* Digito*
                    Digito* Digito* Digito* Digito* ]
```

- Caracteres

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

- String

```
stringLiteral = " [ charLiteral | [ \ " ] ]* "
```

- Booleanos

```
boolLiteral = true | false
```

- Nulo

```
nuloLiteral = null
```

- Puntuación

```
abrePar = (  
cierraPar = )  
abreLlave = {  
cierraLlave = }  
punto = .  
coma = ,  
puntoComa = ;
```

- Operadores

```
opSuma = +  
opResta = -  
opMult = *  
opDivisionEntera = %  
opDivision = /  
opMenor = <  
opMenorIgual = <=  
opIgual ==  
opMayor = >  
opMayorIgual = >=  
opIgualBool = &&  
opOrBool = ||  
opNegacion = !  
opDistinto = !=
```

- Asignacion

```
asignar = =  
asignarSum = +=  
asignarRes = -=
```