

**Universidade Federal de Alagoas**  
Instituto de Computação  
Curso de Ciência da Computação

**Jvlal**  
**Tabela de Símbolos - Tokens**

João Victor Ribeiro Ferro  
Lucas Albuquerque Lisboa

Maceió  
2020.1

<b>Linguagem para desenvolvimento</b>	<b>3</b>
<b>Enumeração com as categorias de Tokens</b>	<b>3</b>
<b>Expressões Regulares Auxiliares</b>	<b>4</b>
<b>Expressões Regulares dos Lexemas</b>	<b>4</b>

# Linguagem para desenvolvimento

A linguagem jv1al foi desenvolvida na linguagem de programação Java os seus analisadores léxicos e sintáticos. Sendo utilizado a versão Java Development Kit na versão 11.0.10

## Enumeração com as categorias de Tokens

```
public enum TokenCategory {  
    ID,  
    CT_FLOAT,  
    CT_INT,  
    CT_CHAR,  
    CT_STRING,  
    OP_ATR,  
    OP_REL,  
    OP_RELNOT,  
    OP_AD,  
    OP_SUB,  
    OP_MULT,  
    OP_DIV,  
    OP_MOD,  
    OP_GREATER,  
    OP_LESS,  
    OP_GREATEREQ,  
    OP_LESSEQ,  
    OP_NOT,  
    OP_AND,  
    OP_OR,  
    OP_CONC,  
    RW_FUNCTION,  
    RW_RETURN,  
    RW_IF,  
    RW_ELSE,  
    RW_WHILE,  
    RW_FOR,  
    RW_INT,  
    RW_FLOAT,  
}
```

```

RW_STRING,
RW_TOSTRING,
RW_CHAR,
RW_BOOL,
RW_READ,
RW_PRINT,
RW_PRINTLN,
RW_TRUE,
RW_FALSE,
RW_NULL,
ON_PAR,
OFF_PAR,
ON_BRACE,
OFF_BRACE,
RW_PROC,
RW_MAIN,
TERMINAL,
SEP;}

```

## Expressões Regulares Auxiliares

letras\_maiusculas = ['A'-'Z']

letras\_minusculas = ['a'-'z']

dígitos = [0-9]

alfanumerico = letras minúsculas | letras\_maisculas | digito;

## Expressões Regulares dos Lexemas

Identificadores	
ID	(“letter”)(‘digit’   ‘letter’)*

Palavras reservadas	
RW_MAIN	‘main’
RW_PROC	‘proc’

RW_FUN	'fun'
PR_IF	'if'
PR_ELSE	'else'
PR_WHILE	'while'
PR_FOR	'for'
RW_FLOAT	'float'
RW_INT	'int'
RW_CHAR	'char'
RW_STRING	'string'
RW_READ	'read'
RW_PRINTLN	'println'
RW_PRINT	'print'
RW_BOOL	'bool'
RW_TRUE	'True'
RW_FALSE	'False'
RW_TOSTRING	'toString'
RW_VOID	'void'
RW_RETURN	'return'
RW_NULL	'Null'

Operadores	
OP_ATR	'='
OP_REL	'=='
OP_RELNOT	'!='
OP_AD	'+'
OP_SUB	'-'
OP_MULT	'*'

OP_DIV	'/'
OP_MOD	'%'
OP_GREATER	'>'
OP_LESS	'<'
OP_GREATEREQ	'>='
OP_LESSEQ	'<='
OP_NOT	'!'
OP_AND	'&'
OP_OR	' '
OP_CONC	'#'

Delimitadores	
ON_PAR	'('
OFF_PAR	')'
ON_BRACES	'{'
OFF_BRACES	'}'
TERMINAL	','
COLON	':'
SEP	','

Constante Literal	
CT_FLOAT	('digit')*( '.' )(digit)*
CT_INT	('digit')*
CT_STRING	(")('letter')*(")
CT_CHAR	(')('letter')(')