

**UNIVERSIDADE FEDERAL DE ALAGOAS**

Instituto de Computação

# **Compiladores**

Especificações dos Tokens **Valcode**

Luís Antônio da Silva Nascimento, Vinicius Monteiro Pontes

Maceió, AL - 2020.1

# 1. Linguagem de desenvolvimento

Valcode usa Python (3.8.5) para implementar seus analisadores léxico e sintático.

## 2. Enum de categoria de Tokens

```
class TokenCategory(Enum):  
    IDENTIFIER = auto()  
    RW_INIT = auto()  
    OP_PAR = auto()  
    CL_PAR = auto()  
    OP_CBRA = auto()  
    CL_CBRA = auto()  
    OP_BRA = auto()  
    CL_BRA = auto()  
    RW_FN = auto()  
    RW_INT = auto()  
    RW_FLOAT = auto()  
    RW_BOOL = auto()  
    RW_CHAR = auto()  
    RW_NULL = auto()  
    RW_VOID = auto()  
    RW_STR = auto()  
    RW_RTN = auto()  
    RW_IF = auto()  
    RW_ELSE = auto()  
    RW_WHILE = auto()  
    RW_FOR = auto()  
    RW_VAR = auto()  
    RW_AND = auto()  
    RW_OR = auto()  
    RW_NOT = auto()  
    RW_INPUT = auto()  
    RW_PRINT = auto()  
    SB_COLON = auto()  
    SB_SCOLON = auto()  
    SB_COMMA = auto()  
    SB_ASGN = auto()  
    OPR_ADD = auto()  
    OPR_SUB = auto()  
    OPR_MUL = auto()  
    OPR_DIV = auto()  
    OPR_UNG = auto()  
    OPR_UPS = auto()  
    OPR_EQUAL = auto()  
    OPR_GT = auto()  
    OPR_LT = auto()
```

```

OPR_GTE = auto()
OPR_LTE = auto()
CT_INT = auto()
CT_FLOAT = auto()
CT_BOOL = auto()
CT_CHAR = auto()
CT_STR = auto()

```

### 3. Expressões regulares auxiliares

Constantes literais de int: `[+-]?[0-9]+$`  
 Constantes literais de float: `[+-]?[0-9]\.[0-9]+`  
 Constantes literais de bool: `true|false`  
 Constantes literais de char: `^'.'*$`  
 Constantes literais de strings: `^".*" $`  
 Símbolos especiais: `^[A-Za-z0-9_]`  
 Identificadores: `^[A-Za-z][A-Za-z0-9_]*`

### 3. Expressões regulares dos lexemas

Identificadores	
IDENTIFIER	{Identificadores}

Palavras reservadas	
RW_FN	'function'
RW_RTN	'return'
RW_IF	'if'
RW_ELSE	'else'
RW_WHILE	'while'
RW_FOR	'for'
RW_INT	'int'
RW_FLOAT	'float'
RW_CHAR	'char'

RW_STR	'string'
RW_BOOL	'bool'
RW_INPUT	'input'
RW_PRINT	'print'
RW_NULL	'null'
RW_VOID	'void'
RW_INIT	'init'

Operadores	
OPR_ASGN	'='
OPR_EQ	'=='
OPR_DIFF	'!=='
OPR_ADD	'+'
OPR_SUB	'_'
OPR_MUL	'*'
OPR_DIV	'/'
OPR_MOD	'%'
OPR_GT	'>'
OPR_LT	'<'
OPR_GTE	'>='
OPR_LTE	'<='
OPR_AND	'and'
OPR_OR	'or'
OPR_NOT	'not'
OPR_UNG	'ung'
OPR_UPS	'ups'

Delimitadores	
OP_CBRA	{
CL_CBRA	}
OP_PAR	(
CL_PAR	)
OP_BRA	[
CL_BRA	]
SB_SCOLON	;
SB_COMMA	,

Constantes literais	
CT_BOOL	{ Constantes literais de bool }
CT_INT	{ Constantes literais de int }
CT_FLOAT	{ Constantes literais de float }
CT_CHAR	{ Constantes literais de char }
CT_STRING	{ Constantes literais de string }