



**UNIVERSIDADE FEDERAL DE ALAGOAS**  
**INSTITUTO DE COMPUTAÇÃO**

## **ESPECIFICAÇÃO DOS TOKENS DA LINGUAGEM Softy**

**ALUNOS:** Guilherme Volney Mota Amaral, Thiago Tenório Cavalcante Costa

**PROFESSOR:** Alcino Dall Igna Junior

**DISCIPLINA:** Compiladores

# SUMÁRIO

<b>SUMÁRIO</b>	<b>1</b>
<b>ESPECIFICAÇÃO DA LINGUAGEM DE IMPLEMENTAÇÃO</b>	<b>2</b>
<b>ENUMERAÇÃO COM AS CATEGORIAS DOS TOKENS</b>	<b>2</b>
<b>ESPECIFICAÇÃO DOS TOKENS DA LINGUAGEM</b>	<b>2</b>

## 1. LINGUAGEM DE IMPLEMENTAÇÃO

A linguagem de programação escolhida para a implementação dos analisadores léxico e sintático da linguagem Softy é JAVA, mais especificamente, em sua oitava versão.

## 2. ENUMERAÇÃO COM AS CATEGORIAS DOS TOKENS

```
public enum TokenCategory {  
    unknown, funDef, funRet, main, id, paramBeg, paramEnd, commaSep, typeInt,  
    typeFloat, typeBool, typeChar, typeVoid, typeStr, beginScope, endScope,  
    constInt, constFloat, twoPt, constBool, constChar, constStr,  
    opAdd, opSub, opMult, opDiv, arrBegin, arrEnd, opPow, opMod,  
    opGreater, opLesser, opGreq, opLeq, opEquals, opNotEqual,  
    opAnd, opOr, opNot, opConcat, opAttrib,  
    EOL, forLoop, whileLoop, condIf, condElseIf, condElse, scan, print, EOF  
}
```

## 3. ESPECIFICAÇÃO DOS TOKENS DA LINGUAGEM

### - Expressões Regulares Auxiliares:

**letter** = [a - z] | [A - Z];

**digit** = [0 - 9];

**symbol** = ' ' | ';' | ',' | '\.' | '\:' | '\?' | '!' | '\+' | '\-' |  
'\\*' | '\\ ' | '\\/' | '\\\_ ' | '%' | '@' | '&' | '\\#' | '\\\$ ' | '\\< ' |  
'\> ' | '=' | '\\(' | '\\)' | '\\[' | '\\]' | '\\{' | '\\}' | '\\| ' | '\\ ' |  
'\" ' | '\\^ ';

### - Expressões Regulares:

-

#### - Main:

main = 'main';

#### - Identificador:

id = ('letter' | '\_') ('digit' | 'letter' | '\_')\*;

#### - Tipos Primitivos:

typeInt = 'int';

```
typeFloat = 'float';
typeChar = 'char';
typeBool = 'bool';
typeStr = 'string';
typeVoid = 'void';
```

- **Constantes de Tipos:**

```
constInt = (('digit')+);
constFloat = (('digit')+)('.')(('digit')+);
constChar = ('')('letter' | 'symbol' | 'digit' | '\
letter')('');
constBool = ('true' | 'false');
constString = ('''')(('letter' | 'symbol' | 'digit' |
\letter')*)(''');
```

- **Array:**

```
arrayBegin = '['
arrayEnd = ']
```

- **Função:**

```
funDef = 'fun';
funReturn = 'return'
```

- **Parâmetros:**

```
paramBegin = '('
paramEnd = ')'
```

- **Escopo:**

```
escBegin = '{';
escEnd = '}';
```

- **Terminadores:**

```
semicolon = ';';
EOF = final do arquivo
```

- **Separador:**

```
commaSep = ',';
```

- **Palavras reservadas de comando de iteração ou seleção:**

```
condIf = 'if';
condElseIf = 'ceif';
condElse = 'else';
forLoop = 'for';
whileLoop = 'while';
```

- **Instruções de leitura e escrita:**

```
print = 'print';  
scan = 'read';
```

- **Operadores lógicos:**

```
opAnd = '&&'  
opOr = '||';  
opNot = '!';
```

- **Operadores aritméticos:**

```
opAdd = '+';  
opSub = '-';  
opMult = '*';  
opDiv = '/';  
opPow = '^';  
opMod = '%';  
opAttrib = '=';
```

- **Operadores relacionais:**

```
opEquals = '==';  
opGreater = '>';  
opLesser = '<';  
opGreq = '>=';  
opLeq = '<=';  
opNotEqual = '!=';
```

- **Operador de concatenação:**

```
opConcat = '::';
```

- **Declaração parâmetros do contador:**

```
colon = ':';
```

- **Desconhecido:**

Tokens de tipo *unknown* são qualquer token que não faça parte da linguagem, logo não possuem uma expressão regular específica.