

# Analizador Léxico para a Linguagem C- utilizando Máquina de Moore

João Vitor Moraski Lunkes

<sup>1</sup>Universidade Tecnológica Federal Do Paraná(UTFPR)  
R. Rosalina Maria Ferreira, 1233 - Vila Carolo, Campo Mourão - PR, 87301-899

<sup>2</sup>Departamento Acadêmico de Computação (DACOM) - UTFPR - CM

joaolunkes@alunos.utfpr.edu.br

**Abstract.** *This document presents the technical report of the first practical work on the subject of Formal Languages, Automata and Computability, taught by Professor Rogério Aparecido Gonçalves, the work asks for the implementation of a lexical analyzer in some language using an automaton.*

**Resumo.** *Este documento apresenta o relatório técnico do primeiro trabalho prático da matéria de Linguagens Formais, Autômatos E Computabilidade, ministrada pelo professor Rogério Aparecido Gonçalves, o trabalho pede a implementação de um analisador léxico em alguma linguagem utilizando um automato.*

## 1. Descrição do trabalho

O trabalho da matéria de Linguagens Formais, Autômatos e Computabilidade é para que seja implementado um analisador léxico utilizando qualquer linguagem, nesse caso foi utilizado PHP, para que ele reconheça os tokens da linguagem C-.

### 1.1. Linguagem C-

A escolha dessa linguagem foi pelo fato de grande parte das disciplinas introdutórias à programação, como Algoritmos dos cursos de Computação serem ministradas utilizando C, assim teríamos a vantagem de ser uma linguagem conhecida pelos alunos.

### 1.2. Análise Léxica

A Análise Léxica, ou tokenização é o processo que é feito para converter uma sequência de caracteres em um arquivo em uma sequência de tokens. Um programa que realiza isso é conhecido como lexer, tokenizador ou scanner, porém scanner também pode ser o nome do primeiro estágio de um lexer. O lexer faz a varredura do programa caractere por caractere e traduz isso em uma sequência de símbolos ou tokens. Por conta disso que cada linguagem possui suas palavras reservadas, constantes e identificadores.

## 2. Implementação

Linguagem: PHP 7.4.3

IDE: PHP Storm - JetBrains Student License

O arquivo do projeto possui um arquivo chamado README.md onde é possível encontrar tutoriais de instalação e como rodar o código.

O código está separado em mais de um arquivo.

- entrada.txt Onde ficam os casos de teste.
- Estados.php Classe que possui todos os estados e as lógicas para passar de um para o outro.
- Token.php Entidade do token que possui várias variáveis estáticas para salvar os valores como constantes na entidade para ser mais fácil utilizar nos estados.
- main.php Arquivo principal para rodar o código, lê um arquivo e após isso executa as verificações linha a linha.

## 3. Casos de teste

### 3.1. Caso 01

Primeiro caso de teste

```
# Primeiro caso teste
int main(void){
    return(0);
}
```

Figure 1. Primeiro caso de teste

```
0:52 ~/faculdade/lfac/lexical-analyzer master x
$ php main.php
Tamanho do vetor de tokens: 12
Token 0: INT
Token 1: ID
Token 2: LPAREN
Token 3: VOID
Token 4: RPAREN
Token 5: LBRACES
Token 6: RETURN
Token 7: LPAREN
Token 8: NUMBER
Token 9: RPAREN
Token 10: SEMICOLON
Token 11: RBRACES
```

Figure 2. Resposta do primeiro caso de teste

### 3.2. Caso 02

Segundo caso de teste

```
# Segundo caso teste
int main(void) {
    if(1 < 2){
        return 0;
    }
    while (1){
        void;
    }
    printf(,);
    int numero;
    + - ** / <
    <=
    >
    >=
    ==
    !=
    (
    [
    {}
    =
    ;
}
```

Figure 3. Segundo caso de teste

```
0:00 ~/faculdade/lfac/lexical-analyzer master x
$ php main.php
Tamanho do vetor de tokens: 58
Token 0: INT          Token 1: ID          Token 2: LPAREN       Token 3: VOID         Token 4: RPAREN
Token 5: LBRACES      Token 6: IF          Token 7: LPAREN       Token 8: NUMBER       Token 9: LESS
Token 10: NUMBER      Token 11: RPAREN     Token 12: LBRACES     Token 13: RETURN      Token 14: NUMBER
Token 15: SEMICOLON   Token 16: RBRACES    Token 17: WHILE       Token 18: LPAREN      Token 19: NUMBER
Token 20: RPAREN      Token 21: LBRACES    Token 22: ID          Token 23: SEMICOLON   Token 24: RBRACES
Token 25: ID          Token 26: LPAREN     Token 27: COMMA       Token 28: SEMICOLON   Token 29: RPAREN
Token 30: SEMICOLON   Token 31: INT        Token 32: ID          Token 33: SEMICOLON   Token 34: PLUS
Token 35: MINUS       Token 36: TIMES      Token 37: TIMES       Token 38: DIVIDE      Token 39: LESS
Token 40: LESS EQUAL  Token 41: ATTRIBUTION Token 42: GREATER      Token 43: GREATER EQUAL Token 44: ATTRIBUTION
Token 45: EQUALS      Token 46: ATTRIBUTION Token 47: DIFFERENT    Token 48: ATTRIBUTION  Token 49: LPAREN
Token 50: RPAREN      Token 51: LBRACKETS  Token 52: RBRACKETS   Token 53: LBRACES     Token 54: RBRACES
Token 55: ATTRIBUTION Token 56: SEMICOLON   Token 57: RBRACES
0:00 ~/faculdade/lfac/lexical-analyzer master x
$
```

Figure 4. Resposta do primeiro caso de teste

### 3.3. Caso 03

Terceiro caso de teste

```
## Terceiro caso teste
if (candidato1 >= candidato2){
    if (candidato1 > candidato3){
        if (candidato1 > candidato4){
            candidato1+=branco;
        } else{
            candidato4 += branco;
        }
    } else if (candidato3 > candidato4){
        candidato3+=branco;
    } else{
        candidato4+=branco;
    }
} else{
    candidato2 += branco;
}
```

Figure 5. Terceiro caso de teste

```
--9:00 ~/faculdade/lfac/lexical-analyzer master x
└─$ php main.php
Tamanho do vetor de tokens: 58
Token 0: INT          Token 1: ID          Token 2: LPAREN       Token 3: VOID         Token 4: RPAREN
Token 5: LBRACES      Token 6: IF           Token 7: LPAREN       Token 8: NUMBER       Token 9: LESS
Token 10: NUMBER      Token 11: RPAREN      Token 12: LBRACES     Token 13: RETURN      Token 14: NUMBER
Token 15: SEMICOLON   Token 16: RBRACES     Token 17: WHILE       Token 18: LPAREN      Token 19: NUMBER
Token 20: RPAREN      Token 21: LBRACES     Token 22: ID          Token 23: SEMICOLON   Token 24: RBRACES
Token 25: ID          Token 26: LPAREN      Token 27: COMMA       Token 28: SEMICOLON   Token 29: RPAREN
Token 30: SEMICOLON   Token 31: INT         Token 32: ID          Token 33: SEMICOLON   Token 34: PLUS
Token 35: MINUS       Token 36: TIMES       Token 37: TIMES       Token 38: DIVIDE      Token 39: LESS
Token 40: LESS EQUAL  Token 41: ATTRIBUTION Token 42: GREATER      Token 43: GREATER EQUAL Token 44: ATTRIBUTION
Token 45: EQUALS      Token 46: ATTRIBUTION Token 47: DIFFERENT    Token 48: ATTRIBUTION Token 49: LPAREN
Token 50: RPAREN      Token 51: LBRACKETS   Token 52: RBRACKETS   Token 53: LBRACES     Token 54: RBRACES
Token 55: ATTRIBUTION Token 56: SEMICOLON   Token 57: RBRACES
--9:00 ~/faculdade/lfac/lexical-analyzer master x
└─$
```

Figure 6. Resposta do primeiro caso de teste