



CONSTRUÇÃO DE UM ANALISADOR LÉXICO PARA A DISCIPLINA DE COMPILADORES

Vinicius Gulartt¹, Patricia Mariotto Mozzaquatro Chicon², Luis Filipe Durigon De Melo¹,
Rodrigo Luiz Antoniazzi³

Palavras-chave: Compilador, Lexema, *Token*, *Scanner*, Análise Léxica.

1 INTRODUÇÃO

Nos dias atuais, a informática vem se mostrando cada vez mais presente, onde o desenvolvimento de *software* e a prática da programação não é algo mais exclusivo somente a programadores experientes e diplomados, pois com o grande número de Ambientes de Desenvolvimento Integrados (ADI) grátis e com o auxílio da maior rede do mundo, a *internet*, a atividade de programação de um *software* vem se tornando cada vez mais praticada (SEBESTA, 2018).

Assim, entender como um ADI efetua a compilação do código pode facilitar o entendimento e melhorar a descoberta de erros que o programa possa ter. A Análise Léxica é uma das primeiras fases, com o objetivo de identificar cada sequência de caracteres que constituem uma unidade léxica (*tokens*). O analisador léxico faz a leitura do código-fonte caractere por caractere, e faz a verificação do mesmo comparando com o alfabeto da linguagem. Em seguida, agrupa-os formando os *tokens* e, com isso, ele transforma o programa-fonte com uma representação mais conveniente para o analisador sintático dar continuidade ao processo de compilação. Para entender melhor o processo é necessário entender o conceito de linguagem.

De acordo com ALEXANDRE (2019) uma linguagem é formada por um conjunto de palavras, que por sua vez é formada por um conjunto de caracteres, denominado de alfabeto da linguagem. Esse alfabeto definirá os caracteres aceitos pela linguagem e qualquer caractere fora desse alfabeto ocasionará um desentendimento, como exemplo, a linguagem utilizada

¹ Discente do curso de Ciência da Computação, da Universidade de Cruz Alta - Unicruz, Cruz Alta, Brasil. E-mail: gulartt94@gmail.com, lfilipe271197@gmail.com

² Docente do curso de Ciência da Computação, da Universidade de Cruz Alta - Unicruz, Cruz Alta, Brasil. E-mail: pmozzaquatro@unicruz.edu.br

³ Pesquisador do Grupo de Engenharias e Tecnologias - ENGETEC, Docente da Universidade de Cruz Alta - Unicruz, Cruz Alta, Brasil. E-mail: rantoniazzi@unicruz.edu.br



nesse trabalho foi a Java, que por sua vez possui seu próprio alfabeto e conjunto de lexemas, onde qualquer caractere inválido para a linguagem ocasionará um erro.

Assim, linguagens computacionais funcionam de maneiras diferentes, a linguagem que mais se aproxima da linguagem usada pelo processador é chamada de linguagem de baixo nível. Pela sua complexidade e grande nível de detalhes, torna-se muito complicado o seu uso na programação. Deste modo, surgiram os compiladores, que possuem a finalidade de traduzir linguagens mais fáceis e dinâmicas, chamadas de linguagens de alto nível, para linguagens de baixo nível. O compilador possui diversas fases de operações, sendo elas, a análise léxica, análise sintática, análise semântica e a geração de código.

No presente trabalho será explicado o funcionamento da análise léxica.

Quanto à metodologia do presente trabalho, trata-se de uma pesquisa quantitativa, a qual apresenta caráter explicativo e contou com pesquisas bibliográficas para enriquecê-lo.

2 MATERIAIS E MÉTODOS

Em relação à metodologia do presente trabalho, este refere-se a um estudo quantitativo, que segundo Mezzaroba e Monteiro (2001, pg 109):

Quantidade representa tudo aquilo que pode ser medido, o mensurável. Então, se o objeto de sua pesquisa se prestar a qualquer tipo de medição e está, evidentemente, for interessante para o resultado final da investigação a que você se propôs, a adoção de procedimento de quantificação pode lhe ser útil.

A pesquisa apresenta caráter explicativo. De acordo com Gil (2008, p. 28) este tipo de pesquisa “[...] têm como preocupação central identificar os fatores que determinam ou que contribuem para a ocorrência dos fenômenos. Este é o tipo de pesquisa que mais aprofunda o conhecimento da realidade, porque explica a razão, o porquê das coisas. [...]”. O estudo também contou com pesquisas bibliográficas para enriquecê-lo.

A análise léxica é a primeira etapa de um compilador, efetuando o recebimento da entrada realizada pelo usuário, e seu primeiro passo normalmente é produzir uma separação de caracteres que podem ser lidos. Logo após, ela executa um agrupamento dos caracteres individuais em *tokens*, que são a menor unidade que possui significado no programa-fonte, praticamente uma “palavra”. (AHO et al, 2015).

Os *tokens* se enquadram em classes específicas, as quais serão mostradas a seguir:

- Números, podendo ser inteiros, reais ou complexos;



- Operadores, sendo aritméticos, relacionais ou lógicos;
- Parentisadores, incluindo delimitadores, como aspas, parênteses, colchetes ou chaves;
- Sinais de pontuação, como ponto, vírgula ou ponto e vírgula;
- Identificadores, que são variáveis atribuídas pelo usuário, palavras reservadas da linguagem e funções;

Quando executado o processo de análise léxica são desconsiderados caracteres não significativos, como espaços em brancos ou comentários.

Logo após, a separação dos *tokens*, é efetuada uma verificação da classe dos mesmos, comparando-os com uma tabela desenvolvida para a linguagem utilizada e, assim, salvando os mesmos em outra tabela, na qual deve conter a linha em que se encontra, o lexema e a classe do *token*, segundo LOUDEN (2004).

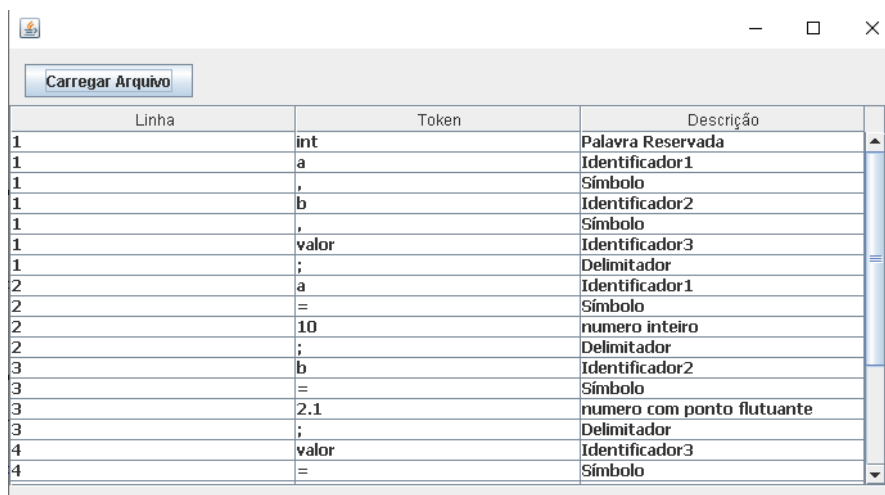
3 RESULTADOS E DISCUSSÕES

O programa desenvolvido efetua os mesmos passos de um analisador léxico, começando pela captação do código escrito pelo usuário, e a separação de cada caractere, efetuando o agrupamento dos lexemas, pegando por limite os espaços em branco, delimitadores, e sinais de pontuação. São desconsiderados os espaços e comentários.

Após é executado uma comparação desses lexemas, com a tabela que possui a classe e a descrição de cada *token*. Se o lexema não se encaixar em nenhuma das classes da tabela, o mesmo é atribuído como um identificador e é numerado, para que se ocorra uma outra aparição o mesmo já esteja identificado. Assim todos os dados são atribuídos a uma tabela, mostrada na Figura 1.



Figura 1 – Tela de exibição dos tokens



Linha	Token	Descrição
1	int	Palavra Reservada
1	a	Identificador1
1	,	Símbolo
1	b	Identificador2
1	,	Símbolo
1	valor	Identificador3
1	;	Delimitador
2	a	Identificador1
2	=	Símbolo
2	10	numero inteiro
2	;	Delimitador
3	b	Identificador2
3	=	Símbolo
3	2.1	numero com ponto flutuante
3	;	Delimitador
4	valor	Identificador3
4	=	Símbolo

Fonte: Elaborado pelo Autor

Assim, tem-se por resultado que o programa desenvolvido obteve êxito em seu objetivo, qual seja, explicar o funcionamento da análise léxica de um compilador.

4 CONCLUSÃO

O presente trabalho teve por objetivo explicar e simular o funcionamento da análise léxica, a qual é a primeira etapa do processo de um compilador.

Conclui-se assim que o presente trabalho trouxe maior facilidade na compreensão da análise léxica e o funcionamento de um compilador por meio do desenvolvimento de um algoritmo simulando esse processo.

REFERÊNCIAS

ALEXANDRE, Eduardo UFBP. **Introdução aos Compiladores**. Disponível em <<http://producao.virtual.ufpb.br/books/tautologico/compiladores-livro/livro/livro.pdf>>. Acesso em: 18 ago 2019.

SEBESTA, Robert W. Conceitos de linguagem de programação. 11.ed. Porto Alegre: Bookman, 2018.

AHO, Alfred V.; SETHI, Ravi; ULLMAN, Jeffrey D. Compiladores - Princípios , Técnicas e Ferramentas. 2 ed. São Paulo: Person, 2015.

LOUDEN, Kenneth C. Compiladores : princípios e práticas. São Paulo: Cengage Learning, 2004.