

# Projeto de Implementação de um Compilador para a Linguagem **T++**

*Análise Léxica (Trabalho – 1ª parte)*

**Prof. Rogério Aparecido Gonçalves<sup>1</sup>**

<sup>1</sup>*Universidade Tecnológica Federal do Paraná (UTFPR)*

*Departamento de Computação (DACOM)*

[rogerioag@utfpr.edu.br](mailto:rogerioag@utfpr.edu.br)

**12 de agosto de 2019**

## **Resumo**

Este documento apresenta a especificação da 1ª parte do trabalho de implementação da disciplina. O objetivo nessa etapa é projetar e implementar a fase de *Análise Léxica* do compilador para a linguagem **T++**.

## **Sumário**

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Análise Léxica</b>                                    | <b>2</b> |
| 1.1      | Instruções Gerais . . . . .                              | 2        |
| 1.2      | Implementação . . . . .                                  | 2        |
| 1.3      | Linguagens de programação para a implementação . . . . . | 3        |
| 1.4      | Testes . . . . .   | 4        |
| 1.5      | Documentação . . . . .                                   | 4        |
| 1.6      | Avaliação . . . . .                                      | 4        |
| 1.7      | Entrega e Apresentação . . . . .                         | 5        |
| 1.8      | Referências . . . . .                                    | 6        |

# 1 Análise Léxica

## 1.1 Instruções Gerais

1. Faça download do arquivo do modelo de estrutura do trabalho e relatório disponível na página da disciplina no moodle. Descompacte e trabalhe nos arquivos e estrutura fornecida, pois será a mesma estrutura que deverá ser entregue ao final do projeto.
2. Siga a estrutura fornecida para desenvolver o trabalho.
3. O relatório deve ter a descrição do trabalho e dos programas, o código fonte dos programas, uma explicação sobre o funcionamento do programa, o processo de tradução com exemplos de instruções dos três formatos e um exemplo de execução do seu programa reproduzindo a saída gerada.
4. Deverão ser entregues:
  - a) O código fonte dos programas.
  - b) Relatório em **pdf** que pode ser feito no formato do LibreOffice ou no Latex.
5. O projeto deve seguir a estrutura de diretórios e arquivos, disponível no formato. A estrutura do projeto é apresentada na Figura 1.

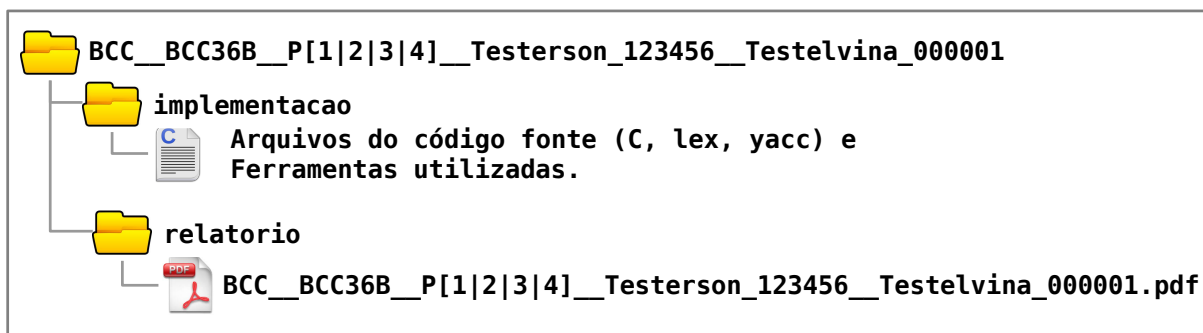


Figura 1: Formato de Entrega

## 1.2 Implementação

A linguagem de programação que será implementada é a **T++**.

Para implementar o sistema de varredura (scanner) da linguagem, é necessário tomar nota das classes de **tokens** apresentadas na Tabela 1.

Tabela 1: **Tokens** da linguagem **T++**

| palavras reservadas | símbolos        |
|---------------------|-----------------|
| <b>se</b>           | + soma          |
| <b>então</b>        | - subtração     |
| <b>senão</b>        | * multiplicação |
| <b>fim</b>          | / divisão       |
| <b>repita</b>       | = igualdade     |
| <b>flutuante</b>    | , vírgula       |
| <b>retorna</b>      | := atribuição   |
| <b>até</b>          | < menor         |
| <b>leia</b>         | > maior         |
| <b>escreva</b>      | <= menor-igual  |
| <b>inteiro</b>      | >= maior-igual  |
|                     | ( abre-par      |
|                     | ( fecha-par     |
|                     | : dois-pontos   |
|                     | [ abre-col      |
|                     | ] fecha-col     |
|                     | && e-logico     |
|                     | ou-logico       |
|                     | ! negação       |

Ainda podem ser definidos os **tokens**:

- **número**: 1 ou mais dígitos que podem ser *inteiro* ou *flutuante* (representação em notação científica ou não);
- **identificador**: começa com uma letra e precede com  $N$  letras e números sem limite de tamanho;
- **comentários**: cercados de chaves da seguinte forma: {...}. É possível inclusive comentários que cubram múltiplas linhas.

**Obs.:** Para a construção da linguagem, utilize palavras reservadas em português brasileiro como indicado na tabela.

### 1.3 Linguagens de programação para a implementação

Para a implementação do compilador, pode ser utilizado qualquer linguagem de programação<sup>1</sup>. É recomendado que seja utilizado uma linguagem que dê suporte à estruturas de dados de alto nível e preferencialmente que exista bibliotecas para a construção da varredura e gramática. Algumas LPs/ferramentas conhecidas são:

<sup>1</sup>Que tenha suporte às ferramentas de especificação do sistema de varredura e da gramática e que também tenha suporte ao [LLVM](#).

- C/C++ - Flex/Bison [3][4]
- Python - PLY (que possui ferramenta léxica e sintática) [5]
- Java - JFlex/Jacc [6][7]

**Obs.:** A implementação de referência será apresentada na linguagem **C**.

Foi adicionado também um vídeo no moodle da disciplina sobre varredura na linguagem de programação **Go**. Ao longo da disciplina serão adicionados exemplos, tutoriais e outras informações para auxiliar o aluno na disciplina.

## 1.4 Testes

Alguns casos de testes estão disponíveis no moodle institucional junto com essa especificação. Serão executados esses testes e outros testes que o professor julgar necessário durante a avaliação desta parte do trabalho.

## 1.5 Documentação

Durante toda a disciplina o aluno criará uma documentação formal da implementação do compilador para a linguagem. Sendo os relatórios com conteúdo acumulativo, isto é, as fases subsequentes irão complementar o conteúdo existente das fases anteriores. Para a fase de Análise Léxica, a documentação deve apresentar:

- Especificação da linguagem de programação **T++**;
- Especificação formal dos autômatos para a formação de cada classe de **token** da linguagem;
- Detalhes da implementação da varredura na LP e ferramenta (e/ou bibliotecas) escolhidas pelo projetista;
- Exemplos de saída do sistema de varredura (lista de *tokens*) para exemplos de entrada (código fonte).
- Implemente uma função que imprima a lista de **tokens**, não utilize a saída padrão da ferramenta de implementação de **Analísadores Léxicos**.

Utilize o formato de artigo da SBC [8] para fazer o relatório.

## 1.6 Avaliação

Será avaliado o funcionamento da varredura para a linguagem de programação **T++**.

- Varredura: programa de exemplo **T++** de **entrada** na linha de comando.
- **Saída** será o conjunto de pares (valor:token).

Para a avaliação inicial será considerado então (obrigatoriamente):

- Utilizar palavras reservadas em português-BR;
- Construção da Análise Léxica;
- Inserção de comentários (para adicionar explicações futuras no código);
- Levantamento de erros (sugerindo classes de erros). Para isso: Contabilizar linhas (`\n`), colunas e lexema atual (as ferramentas fazem isso);
- Serão avaliados, dentre outros critérios:
  - a) Da implementação:
    - O funcionamento do programa.
    - O capricho e a organização na elaboração do projeto.
    - A corretude da implementação em relação ao que foi pedido no trabalho.
    - A colocação em prática dos conceitos que foram discutidos em sala de aula de forma correta.
    - A qualidade do projeto e da implementação (descrição e elaboração do projeto e o passo a passo da implementação).
  - b) Do relatório:
    - O conteúdo e a forma que foi apresentado, se o formato é o mesmo solicitado.
    - Organização das ideias e do processo de tradução.
    - O capricho na elaboração e na formatação do texto, bem como o conteúdo do texto.
- Não serão avaliados os trabalhos:
  - a) Que cheguem fora do prazo.
  - b) Que não forem feitos nas ferramentas solicitadas.
  - c) Que não estejam no formato especificado.
  - d) Que não foram compactados em um só arquivo.
  - e) Que não tiverem identificação (nome e matrícula).
  - f) Que forem cópias de outros trabalhos ou materiais da internet.
  - g) Que não seguirem todas estas instruções.
- Não se esqueça que o trabalho contribui com **1,0** ponto da nota.

## 1.7 Entrega e Apresentação

O trabalho será **individual** e deverá ser entregue até o dia **05/09/2019** no moodle da disciplina em um pacote compactado. A estrutura do projeto com os arquivos do projeto (fonte e relatório) deve ser compactada (zipados) e o arquivo compactado deve ser enviado pelo moodle utilizando a opção de submissão “**Trabalho 1a. parte - Análise Léxica**”, o nome do arquivo compactado deve seguir o padrão de nomes do formato.

Deverá ser especificado na entrega o mecanismo de execução da varredura para a realização da correção.

**Obs.:** Favor utilizar ZIP como forma de compactação. O RELATÓRIO DEVE SER TAMBÉM ENTREGUE IMPRESSO, NO HORÁRIO DA AULA, PARA O PROFESSOR.

## 1.8 Referências

- [1] LOUDEN, Kenneth C. Compiladores: princípios e práticas. São Paulo, SP: Thomson, c2004. xiv, 569 p. ISBN 8522104220.
- [2] <http://web.cecs.pdx.edu/~mpj/jacc/>
- [3] <http://flex.sourceforge.net/>
- [4] <http://www.gnu.org/software/bison/>
- [5] <http://www.dabeaz.com/ply/>
- [6] <http://jflex.de/>
- [7] <http://web.cecs.pdx.edu/~mpj/jacc/>
- [8] Formato para publicação de artigos da SBC