

**Construção de Compiladores**  
**Daniel Lucrédio**  
**Especificação e Critérios de notas do Trabalho 1**  
**(Última revisão: dez/2021)**

O trabalho 1 (T1) da disciplina consiste em implementar um analisador léxico para a linguagem LA (Linguagem Algorítmica) desenvolvida pelo prof. Jander, no âmbito do DC/UFSCar. O analisador léxico deve ler um programa-fonte e produzir uma lista de tokens identificados. Exemplo:

**Entrada:**

```
{ leitura de nome e idade com escrita de mensagem usando estes dados }
algoritmo
    declare
        nome: literal
    declare
        idade: inteiro

    { leitura de nome e idade do teclado }
    leia(nome)
    leia(idade)

    { saída da mensagem na tela }
    escreva(nome, " tem ", idade, " anos.")
fim_algoritmo
```

**Saída produzida:**

```
<'algoritmo','algoritmo'>
<'declare','declare'>
<'nome',IDENT>
<':',':'>
<'literal','literal'>
<'declare','declare'>
<'idade',IDENT>
<':',':'>
<'inteiro','inteiro'>
<'leia','leia'>
<'(','('>
<'nome',IDENT>
<')',')'>
<'leia','leia'>
<'(','('>
<'idade',IDENT>
<')',')'>
<'escreva','escreva'>
<'(','('>
<'nome',IDENT>
<','',','>
<' " tem " ',CADEIA>
<','',','>
<'idade',IDENT>
<','',','>
<' " anos." ',CADEIA>
<')',')'>
<'fim_algoritmo','fim_algoritmo'>
```

Espaços em branco e comentários devem ser ignorados.

Em caso de erro léxico, o programa deve interromper a execução e reportar o erro, indicando o primeiro símbolo onde o erro ocorre. Exemplo:

**Entrada:**

```

{ leitura de nome e idade com escrita de mensagem usando estes dados }

algoritmo
    declare
        nome~ literal
    declare
        idade: inteiro

    { leitura de nome e idade do teclado }
    leia(nome)
    leia(idade)

    { saída da mensagem na tela }
    escreva(nome, " tem ", idade, " anos.")

fim_algoritmo

```

#### Saída produzida:

```

<'algoritmo','algoritmo'>
<'declare','declare'>
<'nome',IDENT>
Linha 5: ~ - simbolo nao identificado

```

Outros erros a serem reportados são: comentário não fechado na mesma linha e cadeia não fechada na mesma linha (ver casos de teste).

O analisador deve poder ser executado em linha de comando (windows, mac ou linux), com DOIS ARGUMENTOS OBRIGATORIAMENTE:

Argumento 1: arquivo de entrada (caminho completo)

Argumento 2: arquivo de saída (caminho completo)

Exemplo de como seu analisador deve rodar:

```

c:\java      -jar      c:\compilador\meu-compilador.jar      c:\casos-de-teste\arquivo1.txt
c:\temp\saida.txt

```

Como resultado, seu compilador deve ler a entrada de c:\casos-de-teste\arquivo1.txt e salvar a saída no arquivo c:\temp\saida.txt

NÃO SERÃO ACEITOS programas que imprimem a saída no terminal. É obrigatório salvar no arquivo.

### **Critérios de avaliação do Trabalho 1 e descontos nas notas**

O trabalho 1 deve ser desenvolvido em **grupos de até 3 estudantes** (máximo). A nota do trabalho 1 será composta de 3 parcelas, cada uma valendo de 0 a 10, e com os pesos assim distribuídos:

DE - Documentação externa: 10%

DI - Documentação interna: 10%

CT1 - Casos de teste léxico: 80%

**DE - Documentação externa:** deve ser fornecido um arquivo de ajuda, para possibilitar que qualquer pessoa consiga compilar e executar seu trabalho. Deve incluir programas que precisam ser instalados, suas respectivas versões, configurações necessárias, e os passos de execução.

Exemplos de erros comuns na documentação externa e que causarão desconto na nota:

- Só diz como executar, mas não como compilar o programa, ou vice-versa
- Nada foi dito sobre como compilar/interpretar o programa e executá-lo
- Ausência da documentação externa

**DI - Documentação interna:** o código-fonte (gramática + demais arquivos) devem ser documentados a ponto de possibilitar seu entendimento por parte de outros programadores olhando seu código. Insira comentários explicativos em todos os pontos relevantes do seu código. Nomes de variáveis e funções também fazem parte da documentação interna.

Exemplos de erros comuns na documentação interna e que causarão desconto na nota:

- Pouco documentado (sem explicação do propósito das regras léxicas, sintáticas, semânticas e geração de código, entrada, saída, descrição das variáveis, etc.)
- Descuido nos comentários ou nomes de funções/variáveis pouco indicativos
- Ausência de comentários sobre o processo de compilação
- Nenhuma linha de documentação relevante

**CT1 - Casos de teste léxico:** a correção será feita automaticamente, com base em um conjunto de casos de teste fornecidos. Por isso, é obrigatório que a saída produzida seja idêntica à contida nos casos de teste. Será fornecido também um corretor automático (o mesmo a ser utilizado pelo professor), para que o estudante saiba exatamente qual será a sua nota neste critério. São **37 casos de teste** nesta categoria.