

COMPILADORES

Prof. Celso Olivete Júnior

Relatório Analisador Léxico

**Alunos Responsáveis:**

Arthur Reis da Silva RA:151953579

Matheus Palmeira G. Santos RA:151256359

Presidente Prudente

Agosto/2018

**Sumário**

[**Sumário** 2](#_Toc515982843)

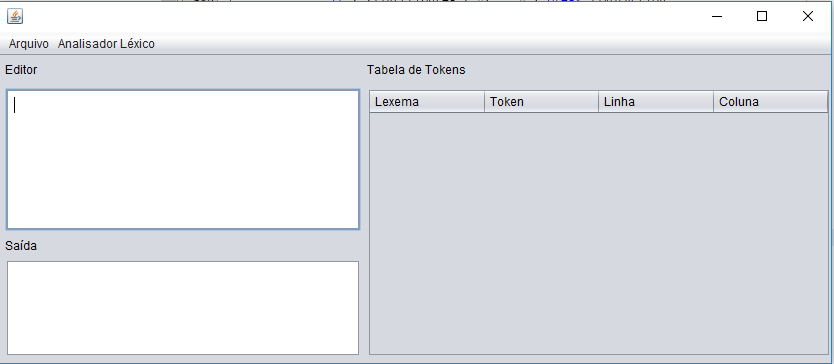
[1. Aplicação e Interface 3](#_Toc515982844)

[2. Implementação 7](#_Toc515982845)

[2.1.1. Método lógico Busca em Largura 7](#_Toc515982846)

[**3.** **Conclusão** 8](#_Toc515982847)

# Aplicação e Interface

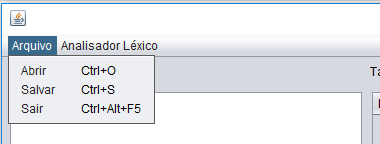
A tela inicial da aplicação pode ser vista conforme imagem abaixo. O uso desta é bem simples e didática:

*Figura 1 – Interface Inicial da Aplicação*

Clicando na opção de menu “Arquivo” disponível no cabeçalho da janela, podemos encontrar as funções:

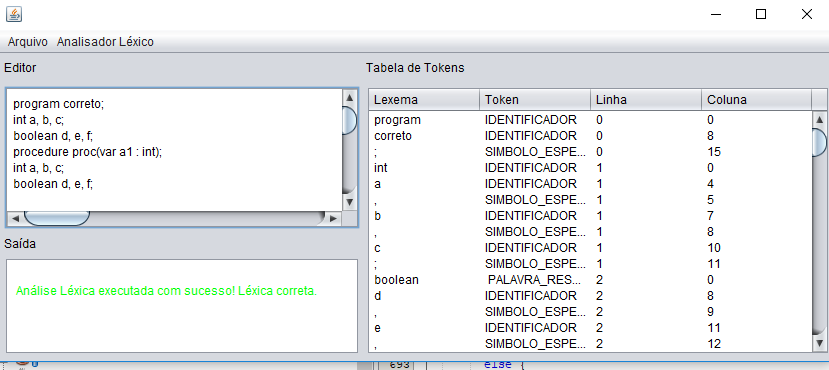
1. Abrir: Permitirá o usuário abrir um arquivo de extensão texto “.txt em seguida exibirá o texto na área Editor.
2. Salvar: Permite o usuário salvar um programa em texto, inserido na área de edição.
3. Sair: Permite o usuário sair da aplicação.

Clicando na opção de menu “Analisador Léxico”, o usuário poderá executar a análise léxica no texto que foi aberto via arquivo ou que foi digitado. O mesmo pode ser feito com o atalho de tecla F8.



*Figura 2 – Opção de menu: “Arquivo”*

Exemplo de execução do programa abaixo:



*Figura 3 – Executado a opção de menu: “Analisador Léxico”*

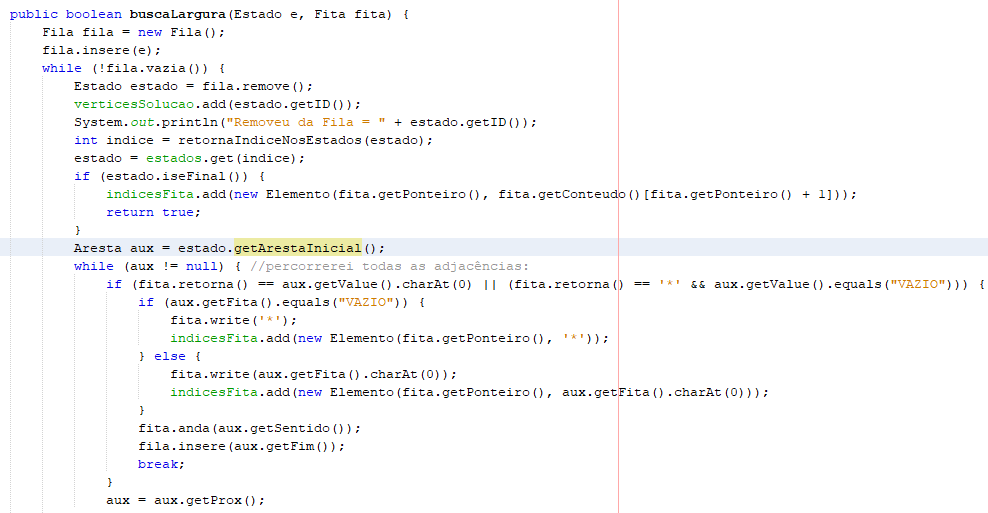
# Implementação

Apresentando as classes utilizadas na implementação. A aplicação foi desenvolvida com base nas seguintes classes:

* Controlador: Classe que representa uma aresta de transição na parte gráfica, possui atributos como: Cor, Vertex inicial, Vertex Final, booleano selecionada, sentido, fita, e transição.
* Gerador: Uma classe auxiliar que representa um estado, ela possui sua aresta inicial, boolean incial e final, e uma identificação.
* LexycalAnalyzer: É a classe que representa a fita, contendo um array de String e um ponteiro. A fita é finita. Tanto para esquerda, quanto para a direita;
* Simbolo: É a classe controladora e uma das mais importantes para a lógica da Máquina de Turing, ela possui vários métodos que controlam os estados, arestas, vértices criados, a fita e transições.
* InterfaceIO: A classe que representa uma transição, ela contém uma String label, e fita, sentido e um ponto (x,y).

## 2.1.1. Método lógico Busca em Largura

O método principal que opera sobre a fita, estados e transições é o método Busca em Largura. Nele encontramos outras sub-rotinas que são chamadas para auxiliar na execução, mas resumidamente o método percorre todos os vértices (estados) do autômato, buscando verificar se a entrada da fita, índice a índice, corresponde as regras apresentadas no autômato da Máquina. Observemos a codificação na imagem abaixo:



*Figura 8 – Código Busca em Largura*

1. **Conclusão**

Conclui-se que a Máquina de Turing foi implementada com sucesso. Os resultados foram conforme o esperado, isto é, a partir de entradas e testes o algoritmo correspondeu com resultados corretos. A implementação gráfica também foi bem executada, apesar da sua complexidade.