**Relatorio do analizador Lexico**

**Nome:**Mariane Luiza Mariano **RA:**11510564

**Nome:**Marcos vinicius ferreira mol **RA:**11520615

Este documento especifica com detalhes e o código do projeto de um analisador léxico para a disciplina de compiladores feito em java utilizando à IDE NetBeans .

Estrutura dos arquivos

ArquivoJavazin.txt

LexerAluno.java​

Tag.java

Token.java

TS.java

Todo o analisador está dentro da pasta Lexer.

**ArquivoJavazin.txt:** Local onde o codigo *javazin* de ser inserido

**LexerAluno.java​ :** Arquivo principal, nele se encontra todo o processo de análise léxica, juntamente com o tratamento de erros.

*public class LexerAluno {...}* Descrição: ​Construtor da classe.

*public LexerAluno(String input\_data)*

Descrição: ​Este método tem como objetivo abrir o arquivo para ser analisado. Caso dê algum erro na abertura do arquivo, um erro é retornado.

*public void fechaArquivo()*

Descrição:​ Este método tem como objetivo fechar o arquivo que estava aberto para a análise.

*public void sinalizaErroLexico(String mensagem)*

Descrição: ​Este método tem como objetivo escrever no arquivo de texto os erros retornados durante a análise.

*public void retornaPonteiro()*

Descrição: ​Este método tem como objetivo voltar à posição anterio do buffer de leitura

*public Token proxToken()*

Descrição: ​Este método tem como objetivo obter o proximo token seguindo o caminho demonstrado pela AFD

*public static void main(String[] args)*

Descrição: ​Este método tem como objetivo de receber o programa fonte para ser lido é imprimir o token é à linha onde o mesmo se encontra além de imprimir à tabela de símbolos

**TS.java** Arquivo onde se encontra à tabela de simbolos.

**Token.java** Contém o modelo de Token padronizado.

*public Token(Tag nome, String lexema, int linha, int coluna)*

*nome* :nome do token que foi definido no Enum de tags.

*lexema*: string contendo o lexema encontrado ao analisar o arquivo.

*linha* : linha onde se encontra o token.

*coluna*: coluna onde se encontra o token.

**Tag.java** Neste aqrquivo se envontara, as tags do analisador léxico que possuem a função de identificar os tokens recebidos

// fim de arquivo

EOF,

//Operadores

RELOP\_LT, // <

RELOP\_OR, //||

RELOP\_AND, //&&

RELOP\_LE, // <=

RELOP\_GT, // >

RELOP\_GE, // >=

RELOP\_EQ, // ==

RELOP\_NE, // !=

RELOP\_ASSIGN, // =

RELOP\_SUM, // +

RELOP\_MINUS, // -

RELOP\_MULT, // \*

RELOP\_DIV, // /

RELOP\_UNNE, // -

RELOP\_UNESC, //!

//Simbolos

SMB\_OP, // (

SMB\_CP, // )

SMB\_OB, // {

SMB\_CB, // }

SMB\_COMMA, // ,

SMB\_SEMICOLON, // ;

SMB\_NULL, // zerar auxnegativo

//identificador

ID,

//numeros

INTEGER,

FLOAT,

//strings

STRING,

// palavra reservada

KW;

//

**AFD:**

