

**Universidade Eduardo Mondlane**

**Faculdade de Ciências**

**Departamento de Matemática e Informática**

**Curso de Informática**

**Disciplina**: Linguagens de Programação e Compiladores **Nível**: I I I **Regime**: Pós-laboral

**Discentes**:

Donaldo Domingos Mariquele

Naum Armindo Tinga

Soares Magaia Júnior

**Docentes:** Seugénio Zeferino

Alfredo Muchanga

Maputo, julho de 2021

Índice

[Ficha Técnica 3](#_Toc80808211)

[Registo de alterações 3](#_Toc80808212)

[1. Introdução 4](#_Toc80808213)

[2. Objectivo 4](#_Toc80808214)

[3. Analisador Léxico 5](#_Toc80808215)

[4. Analisador Léxico Mini Pascal 5](#_Toc80808216)

[5. Linguagem Pascal 5](#_Toc80808217)

[5.1. Palavras reservadas 5](#_Toc80808218)

[5.2. Símbolos 5](#_Toc80808219)

[5.3. Funções aritméticas 5](#_Toc80808220)

[5.4. Funções de transferência 5](#_Toc80808221)

[5.5. Funções ordinais 5](#_Toc80808222)

[5.6. Funções booleanas 5](#_Toc80808223)

[5.7. Procedimentos 5](#_Toc80808224)

[6. Diagrama de Classes 6](#_Toc80808225)

[7. Interface de Utilizador 7](#_Toc80808226)

# Ficha Técnica

**Nome**: Analisador Léxico Mini-Pascal

**Ferramenta de Implementação**: NetBeans IDE 12.3

**Linguagem de Programação**: Java (Versão Java 5 ou posterior)

# Registo de alterações

|  |  |  |
| --- | --- | --- |
| Versão | Conteúdo da alteração | Data |
|  |  |  |
| 1.0.0 | Versão inicial | Junho 2021 |

# Introdução

Linguagem de programação é um método padronizado, formado por um conjunto de regras de implementação de um código fonte que pode ser transformado em um conjunto de instruções que serão executadas pelo computador. O processo de transformação ou tradução do código fonte em instruções para o computador e denominado compilação.

Compiladores convertem programas escritos em uma linguagem de alto nível para programas equivalentes em linguagem de máquina. O Processo de compilação ocorre em numa sequência de etapas a descrever: Análise léxica, Analise Sintática, Analise Semântica, Geração de Código intermediário, Otimização do Código e Geração do Código Objecto.

# Objectivo

O Presente trabalho tem como objectivo desenvolver um Analisador Léxico para um subconjunto da linguagem de Programação Pascal denominado mini-Pascal.

# Analisador Léxico

O analisador léxico (scanner) é a parte do compilador responsável por ler caracteres do programa fonte e transformá-los em uma representação conveniente para o analisador sintático. A analise léxica é a primeira etapa do compilador, essa fase recebe como *input* um arquivo de código fonte e agrupa os caracteres individuais em forma de **tokens** que servirão de entrada para a fase subsequente da compilação. O arquivo de entrada é geralmente armazenado em forma de sequência de caracteres individuais que podem ser lidos e entendidos pelo humano.

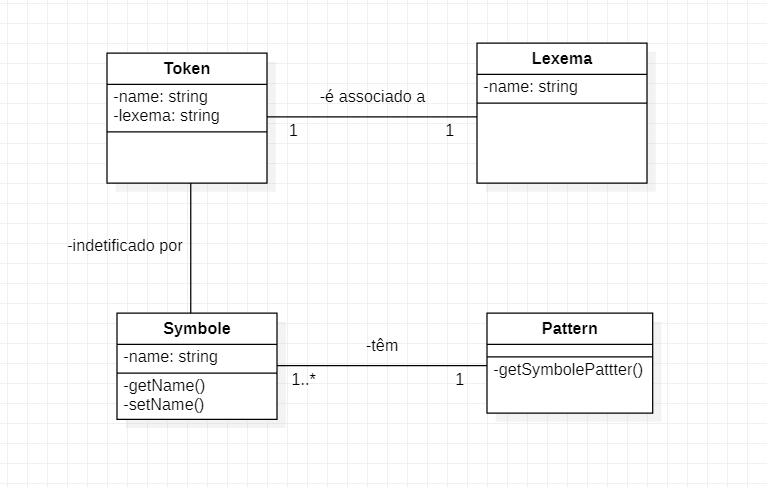
# Analisador Léxico Mini Pascal

O Analisador Léxico Mini Pascal é um programa responsável por realizar a analise léxica do código fonte escrito na linguagem de programação pascal.

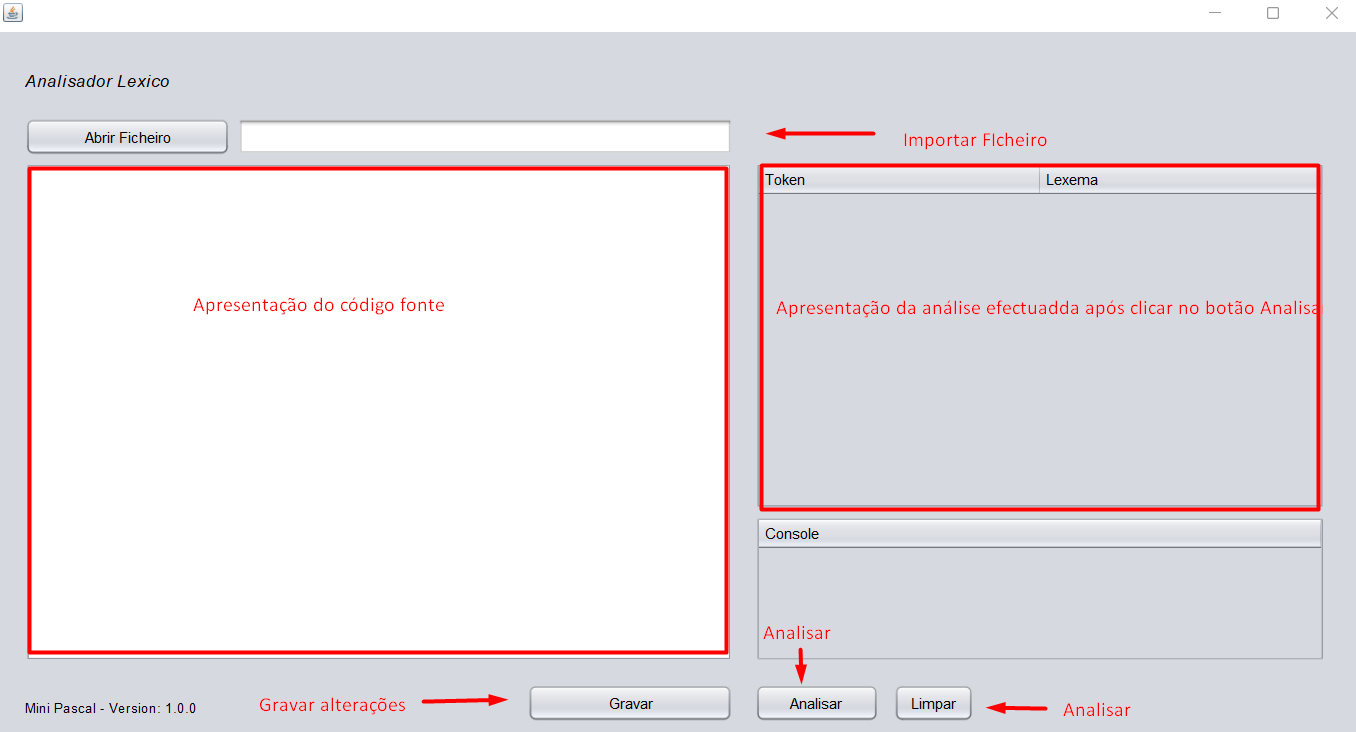
# Linguagem Pascal

|  |  |
| --- | --- |
| Tipo | Exemplo |
| Palavras reservadas | And, array, begin, case, const, div, do, downto, else, end, file, for, function, goto, if, in, label, mod, nil, not, of, or, packed, procedure, program, record, repeat, set, then, to, type, until, var, while, with |
| Símbolos | + - \* / = < > [ ] . , : ; ( ) <> <= >= := .. |
| Funções aritméticas | abs(), arctan(), cos(), exp(), ln(), sin(), sqr(), sqrt() |
| Funções de transferência | round(), trunc() |
| Funções ordinais | chr(), ord(), pred(), succ() |
| Funções booleanas | eof(), eoln(), odd() |
| Procedimentos | get(), page(), put(),read, readln, reset(), rewrite(),write, writeln, dispose(), new(), pack(), unpack() |

# Diagrama de Classes



# Interface de Utilizador

s