

PROJETO LMP (LINGUAGEM MINI PASCAL)

Desenvolver o Analisador Léxico e Sintático da Mini-Linguagem descrita a seguir.

Descrição da Mini-Linguagem PASCAL SIMPLIFICADA

inicio = **'begin'** comando **'end.'**

comando = condicional | iterativo | atribuição | ϵ

condicional = **'if'** condição **'then'** (atribuição) * comando `;`

iterativo = **'while'** condição **'do'** (atribuição) * comando `;`

atribuição = identificador **':='** (expressão | identificador) (operador (expressão | identificador)) * **';'**

condição = identificador lógica identificador

lógica = **'>'** | **'<'** | **'=='** | **'<='** | **'>='**

identificador = letra | letra (letra | dígito)*

letra = **'a'|'b'|'c'|'d'|'e'|'f'|'g'|'h'|'i'|'j'|'k'|'l'|'m'|'n'|'o'|'p'|'q'|'r'|'s'|'t'|'u'|'v'|'x'|'y'|'w'|'z'|'A'|'B'|'C'|'D'|'E'|'F'|'G'|'H'|'I'|'J'|'K'|'L'|'M'|'N'|'O'|'P'|'Q'|'R'|'S'|'T'|'U'|'V'|'X'|'Y'|'W'|'Z'|**

dígito = **'0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'**

expressão = número | **'('** expressão operador expressão **)'**

operador = **'+'** | **'*'** | **'/'** | **'MOD'**

número = dígito⁺ | **'.'** dígito⁺ | dígito⁺ **'.'** dígito⁺

MOD: operador módulo.

Desenvolver:

1. Escreva as ERs (Expressões Regulares) completas.
2. Escreva as ERs (Expressões Regulares) completas em BNF e EBNF.
4. Construa um ou mais autômatos para reconhecer a Mini Linguagem.
5. Construa as duas primeiras fases de um processo de compilação em Java ou C que simule a Mini-Linguagem apresentada, contendo os respectivos Analisadores

Léxico e Sintático. A técnica para implementação do Analisador Sintático poderá ser qualquer uma das técnicas vistas na disciplina Construção de Compiladores.

5. Poderá ser utilizada na implementação nenhuma, uma ou mais ferramentas de automatização em um processo de compilação, a escolha do grupo, ou então, desenvolver todo o código em uma linguagem de programação, sendo as opções:

1) Linguagem C e ferramentas:

- FLEX e B/YACC.

2) Linguagem Java e ferramentas:

- JFLEX e B/YACC/J.
- JFLEX e CUP
- JAVACC

3) Implementação integral em linguagem C ou Java

6. Faça uma avaliação final de seu projeto sugerindo e realizando testes de reconhecimento, contendo pelo menos 4 programas fonte com grau de complexidade crescente (maiores) a sua escolha. Cada um destes programas fonte deverá conter o máximo de símbolos terminais da Mini Linguagem apresentada, o Ideia que entre os 4 programas cubram todos os símbolos terminais. Como produto dos testes deverão ser apresentados os resultados da Análise Léxica e Sintática para os 4 programas, ou seja:

- Extração e classificação dos tokens.
- Identificação de palavras reservadas.
- Tratamento de Erros.
- Tabela de Símbolos.
- Árvore Sintática Abstrata (AST).

7. Elabore um relatório final do projeto contendo todo o escopo do projeto elaborado, ou seja:

- As ERs, BNF e EBNF criadas.
- Os autômatos criados.
- Código fonte dos analisadores léxico sintático (o máximo comentado, utilize técnicas de documentação vistas em Engenharia de Software), seja através de desenvolvimento ou das linguagens das ferramentas de automatização utilizadas.
- Descrição da técnica de reconhecimento sintático, LR(K)/LL(K) ou outra qualquer que utilize.
- Todos os testes realizados e documentados.
- Conclusões e comentários do projeto realizado, tanto em termos de aprendizagem, restrições, adoções e suposições assumidas no projeto.

Entregar impresso e em mídia.

REGRAS PARA A AVALIAÇÃO DOS LABORATÓRIOS:

- Cada grupo deverá entregar o relatório ao professor até no máximo a última aula antes da P4. **NÃO SERÃO DE NENHUMA FORMA ACEITAS ENTREGAS FORA DE PRAZO !**
- Deverá ser apresentada obrigatoriamente a simulação dos 4 programas criados ao professor. O professor poderá também solicitar alguma simulação individual. A demonstração da execução faz parte e tem grande peso na avaliação da atividade.
- Cada grupo deverá ter no máximo 3 componentes.
- **NÃO FAÇA PLÁGIO DE NENHUMA MÍNIMA PARTE DE RELATÓRIOS DE OUTROS COMPONENTES, SE IDENTIFICADO SERÁ ATRIBUÍDA NOTA ZERO A TODOS, INDEPENDENTE DE QUEM TENHA SIDO A FONTE DO PLÁGIO OU DE QUE GRUPO TENHA COPIADO !**

EXEMPLO DIDÁTICO (não obrigatoriamente seu projeto deverá apresentar este formato)

Exemplo de trecho de um programa em Mini Pascal:

```
1  begin
2    if a > 10 then b:=b+1;
3  end.
```



Analizador Léxico



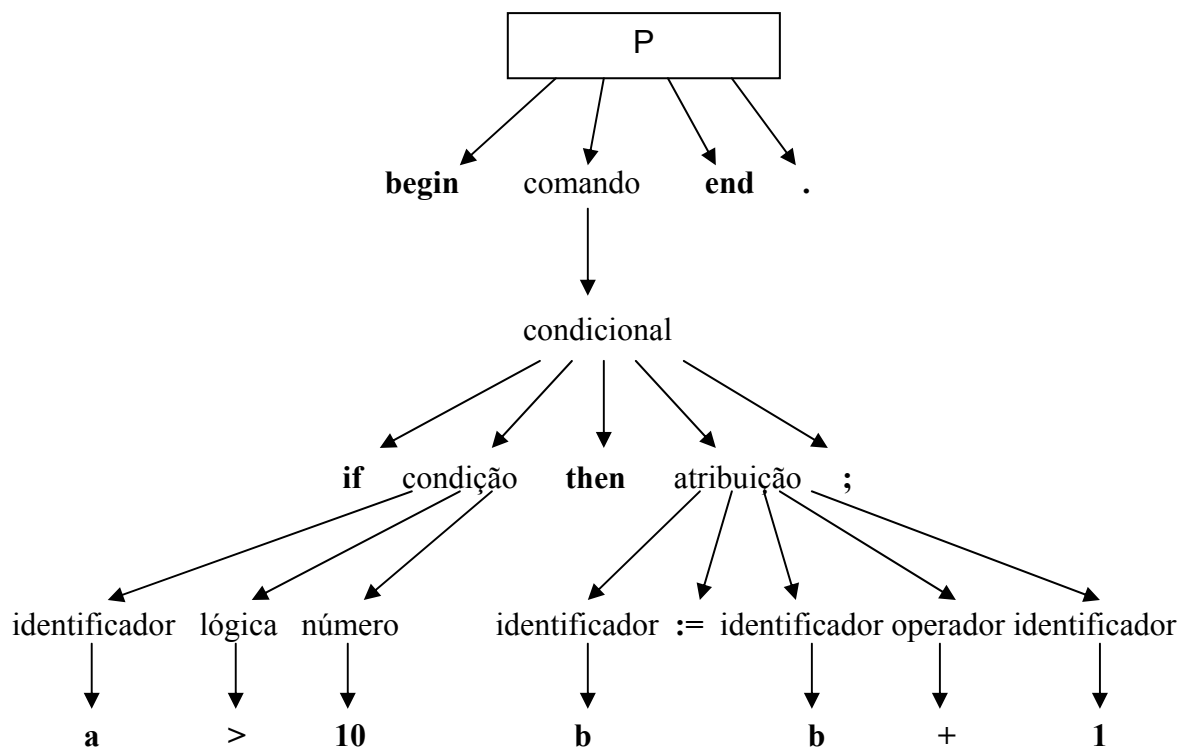
Lista de tokens

| token | Linha |
|----------------------------|-------|
| (begin, palavra reservada) | 1 |
| (if, palavra reservada) | 2 |
| (a, identificador) | 2 |
| (then, palavra reservada) | 2 |
| (b, identificador) | 2 |
| (;, simbolo especial) | 2 |
| (end, palavra reservada) | 3 |
| (., símbolo especial) | 3 |
| (...) | |



Analizador Sintático





Analizador Semântico



Tabela de Símbolos

| Identificador | Categoria |
|---------------|------------------|
| a | variável inteira |
| b | variável inteira |
| (...) | |

Estrutura de dicionário de dados: Listas Lineares, Árvores de Busca, Tabelas Hash
 Todas as etapas de análise fazem interface com esta estrutura.

Dicas:

- 1) Realize uma avaliação das opções de implementação quanto ao escopo do projeto solicitado, facilidade de utilização, documentação disponível e facilidade de aprendizado das ferramentas disponíveis.
- 2) Em qualquer opção selecionada realize antes alguns testes ou desenvolvimentos com escopo menor do que o projeto selecionado, com o objetivo de certificar-se que conseguirá atingir o objetivo final do projeto.