

## Trabalho Prático

A construção de um compilador para uma linguagem imperativa simplificada

### Prática No.2 - Implementação do analisador sintático

1. Escreva uma gramática LL(1) com E.R.s para a linguagem L. Verifique cuidadosamente a gramática, que será a base do analisador sintático. Para as expressões, utilize o formato visto em sala de aula.
2. Implemente um procedimento *CasaToken* que deverá receber como parâmetro o token esperado pela gramática e compará-lo com o token corrente (registro montado pelo analisador léxico). Se estes forem iguais, o próximo token do programa será lido (chamada ao analisador léxico), caso contrário, deverá ser emitida uma mensagem com o erro encontrado e a linha do programa onde este ocorreu, encerrando o processo de compilação.
3. Implemente um procedimento para cada símbolo não-terminal da gramática. Em cada procedimento, a escolha da regra de produção a ser utilizada depende do token lido pelo analisador léxico. Cada token do lado direito da regra deverá ser verificado pelo procedimento *CasaToken*. Símbolos não-terminais do lado direito da regra correspondem a chamadas ao procedimento correspondente.
4. O corpo principal do compilador deverá ler o primeiro token do programa-fonte (chamada ao analisador léxico) e em seguida chamar o procedimento correspondente ao símbolo inicial da gramática.
5. Teste o analisador sintático com exemplos de programas-fontes corretos e exemplos de erros sintáticos.

As mensagens devem ter os seguintes formatos (onde *nn* é o número da linha onde o erro foi detectado e *lex* é o lexema encontrado):

*nn:token nao esperado [lex].*

*nn:fim de arquivo não esperado.*

### O que entregar:

- Gramática
- Códigos-fontes

**Obs: Leia as especificações gerais contidas no documento “Descrição do trabalho”.**