

EXA869 MI Proc. de Linguagem de Programação
Problema 1: Quem fez fez quem não fez não faz mais!
Parte 3.

Cronograma

Sessão	Data	Assunto
1	30/abr	Apresentação do Problema 3
	02/mai	Suspensão atividades acadêmicas
2	07/mai	Problema 3
3	09/mai	Problema 3
4	14/mai	Problema 3
5	16/mai	Problema 3
6	21/mai	Problema 3
7	23/mai	Problema 3
8	28/mai	Problema 3

Pronto! Gramática enviada, disse João para Maria, e ela disse para João: “Antes eu estava com dúvidas entre erros léxicos e sintáticos, agora eu estou com dúvidas entre erros sintáticos e semânticos! Kkkk”. João comentou: “É como o professor comentou, em cada etapa do compilador é realizada um tipo de análise, e como estamos aprendendo por etapas, é normal essas dúvidas”. Maria disse: “Sim, pouco a pouco as coisas vão se encaixando e vamos entendendo melhor o que cada analisador faz.”

Problema

Desenvolver um Analisador Sintático Descendente Preditivo Recursivo para a gramática definida no problema 2. Além disso, o analisador sintático deverá ser capaz de identificar TODOS os erros de sintaxe presentes, alertando ao usuário o tipo do erro e onde ele ocorreu no código.

Produto

O analisador sintático poderá ser desenvolvido em dupla ou individualmente. **Esta dupla deve ser a mesma do problema 1.** O analisador sintático deverá ser implementado na linguagem Java ou Python. O código-fonte do analisador deve ser enviado para o seu tutor, pelo classroom da disciplina, até as 23h59m do dia **28/05/2025**. Será descontado **02 (dois)** pontos por um dia de atraso. Após esse prazo, o código não será mais aceito.

Observações

1. A entrada para este analisador é um conjunto de arquivos texto. Estes arquivos de entrada deverão estar em uma pasta na raiz do projeto chamada **files**.
2. A saída do analisador sintático deverá ser através de um conjunto de arquivos de saída (um para cada arquivo de entrada), denominados **X-saida.txt**, onde **X** é o nome do arquivo de entrada. Estes arquivos de saída deverão estar em uma pasta na raiz do projeto chamada **files**.
3. Os arquivos de saída deverão apresentar a lista de erros sintáticos. Se não houver erros, uma mensagem

de sucesso deve ser exibida. O formato dos arquivos de saída será definido pelo tutor e todos devem segui-lo.

Recursos para aprendizagem

- AHO, A. V.; SETHI, S. & ULLMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. Rio de Janeiro: LTC, 1995.
- LOUDEN, K. C. **Compiladores – Princípios e Práticas**. São Paulo, Thomson, 2004.
- HOPCROFT, J. E. *et al.* **Introdução à Teoria dos Autômatos, Linguagens e Computação**. 1ª edição, Editora Campus, 2002.
- MENEZES, P. F. B. **Linguagens Formais e Autômatos**. 5ª edição, Editora Sagra-Luzzatto, 2005.