

# Plano de Ensino para Construção de Compiladores 1

## Tópicos:

1. Introdução - 4 horas
2. Análise Léxica - 8 horas
3. Análise Sintática - Introdução - 4 horas
4. Análise Sintática Descendente - 8 horas
5. Análise Sintática Ascendente - 1 hora
6. Análise Sintática Ascendente LR - 7 horas
7. Análise Semântica - 4 horas
8. Geração e Otimização de Código - 8 horas
9. Aplicações - 4 horas

Primeira Avaliação (tópicos 1, 2 e 3) - 4 horas

Segunda Avaliação (tópicos 4, 5 e 6) - 4 horas

Terceira Avaliação (tópicos 7 e 8) - 4 horas

## Objetivos específicos:

Ao final da disciplina, os alunos devem ser capazes de compreender e manipular gramáticas livres de contexto, além de conseguir construir compiladores e interpretadores para diferentes tipos de linguagens.

## Estratégias de ensino:

Em todos os tópicos:

- aulas expositivas;
- apresentação e discussão de exemplos;
- projetos realizados em sala de aula, envolvendo o professor e todos os alunos, em forma de debate;
- resolução de exercícios.

## **Atividades dos alunos:**

- participação nas aulas expositivas e nos projetos realizados em sala de aula;
- resolução dos exercícios propostos;
- resolução e entrega dos trabalhos práticos.

## **Atividades em EAD:**

## **Recursos a serem utilizados:**

- página da disciplina como repositório de material didático e envio de tarefas por parte dos alunos;
- sala de aula teórica;
- projetor e notebook para projeção de slides;
- laboratórios do DC para resolução de exercícios e trabalhos extra-aula.

## **Procedimentos de avaliação do aluno:**

Durante o semestre, a avaliação será realizada com base em:

- três provas (P1, P2 e P3) usadas para cálculo da média aritmética MP, cobrindo os seguintes tópicos: P1 (Tópicos 1, 2, 3), P2 (Tópicos 4, 5, 6), P3 (Tópicos 7, 8). Não existe prova substitutiva.
- dois trabalhos práticos (T1 e T2), a serem realizados em grupos, cuja média aritmética é MT, cobrindo os seguintes tópicos: T1 (Tópicos 1 a 4), T2 (Tópicos 7 e 8).

A média final MF será calculada da seguinte forma:

- se  $MP \geq 6,0$  então  $MF = 0,75*MP + 0,25*MT$
- senão  $MF = 0,75*MP + 0,15*MT$

**IMPORTANTE:** Em caso de detecção de plágio (de outro colega ou da internet), em qualquer um dos trabalhos ou provas, a média final da disciplina (MF) será zero!

A AVALIAÇÃO COMPLEMENTAR será uma única prova a respeito de todo o conteúdo do curso (Tópicos 1 a 8). A nova média final (NMF) após a realização da Avaliação Complementar (AC) será calculada da seguinte forma:  $NMF = 0,86*AC$

Se NMF for maior do que MF, ela substituirá MF obtida no semestre. Caso contrário, a média final obtida durante o semestre permanece inalterada.

## **Bibliografia básica:**

- ALFRED V. AHO. et al. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo: Pearson Addison-Wesley, 2007. x, 634 ISBN 9788588639249 - disponível na BCo - UFSCar
- LOUDEN, Kenneth C. Compiladores: princípios e práticas. São Paulo: Pioneira Thomson Learning, 2004. 569 p. ISBN 85-221-0422-0 - disponível na BCo - UFSCar
- COOPER, Keith D. ; Torczon, Linda. Engineering a compiler. 2nd. ed. Amsterdam: Elsevier, 2012. xxiii, 800 p. : il., tabs. ISBN 9780120884780 - disponível na BCo - UFSCar

## **Bibliografia complementar:**

- PARR, Terence. The Definitive ANTLR 4 Reference. IN: The Pragmatic Bookshelf, 2013. 328 p. ISBN 9781934356999.
- DELAMARO, Márcio E. Como Construir um Compilador Utilizando Ferramentas Java. IN: Novatec, 2004. 308p. ISBN 8575220551.
- MAK, Ronald. Writing compilers and interpreters: a modern software engineering approach using Java. 3rd. ed. Indianapolis, IN: Wiley Publishing, 2009. xxiii, 840 p. : il., tabs. ISBN 9780470177075.
- NETO, João José. Introdução à Compilação. 2a. ed. Rio de Janeiro: Elsevier, 2016. 307 p. ISBN 9788535278101.

## **Observações:**

Neste semestre esta disciplina é oferecida nas Turmas A e B.

## Tópicos detalhados:

### 1. Introdução - 4 horas

- O que é um compilador
- Compilador vs. Interpretador
- Outros programas relacionados (pré-processador, montador, editor de ligação)
- Linguagens Formais e Autômatos (LFA) e Compiladores
- Estrutura de um compilador (fases de front-end/análise e back-end/síntese)
- Manipulação de erros
- Aplicações
- Apresentação da Primeira Lista de Exercícios

### 2. Análise Léxica - 8 horas

- O que é a análise léxica
- Como a análise léxica é feita na teoria
- Como um analisador léxico é implementado na prática
- Expressões regulares e Autômatos Finitos aplicados à análise léxica
- Geradores automáticos de analisadores léxicos
- Apresentação da Segunda Lista de Exercícios

### 3. Análise Sintática - Introdução - 4 horas

- O que é a análise sintática
- Tipos de gramáticas (LFA) e Compiladores
- Gramáticas livres de contexto (GLCs)
- Árvores de análise sintática
- Características e processamentos de GLC: ambiguidade, associatividade e precedência, recursividade à esquerda, fatoração à esquerda
- EBNF e diagramas sintáticos
- Apresentação da Terceira Lista de Exercícios

### 4. Análise Sintática Descendente - 8 horas

- O que é a análise sintática descendente
- Como é feita a análise sintática descendente: estratégias e exemplos
- Análise sintática descendente preditiva recursiva
- Análise sintática descendente preditiva sem recursividade
- Recuperação de erros
- Gramáticas LL e suas extensões
- ANTLR
- Apresentação da Quarta Lista de Exercícios

### 5. Análise Sintática Ascendente - 1 hora

- O que é a análise sintática ascendente
- Tipos de analisadores sintáticos ascendentes

### 6. Análise Sintática Ascendente LR - 7 horas

- O que é a análise sintática ascendente LR
- Tabela de análise LR

- Algoritmo de análise LR
- Tabela SLR: estratégia de construção
- Resolução de conflitos na gramática
- Recuperação de erros
- Análise LR e suas extensões: LR(1) canônica, LALR
- Análise sintática descendente vs. análise sintática ascendente
- Apresentação da Quinta Lista de Exercícios

#### 7. Análise Semântica - 4 horas

- O que é a análise semântica
- Como a análise semântica é implementada “à mão”
- Análise semântica dirigida pela sintaxe
- Esquemas de tradução dirigida pela sintaxe
- Tabela de símbolos
- Escopo
- Verificação de tipos
- Apresentação da Sexta Lista de Exercícios

#### 8. Geração e Otimização de Código - 8 horas

- O que é a geração de código
- O que é a geração de código intermediário
- Exemplos de código intermediário (P-código e código de três endereços)
- Como é feita a geração de código intermediário
- O que é a otimização de código
- Níveis de otimização de código
- Apresentação da Sétima Lista de Exercícios

#### 9. Aplicações - 4 horas

- Exemplos de compiladores

Primeira Avaliação (tópicos 1, 2 e 3) - 4 horas

Segunda Avaliação (tópicos 4, 5 e 6) - 4 horas

Terceira Avaliação (tópicos 7 e 8) - 4 horas