



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

PLANO DE ENSINO

1. Identificação

Disciplina: INE5426 - Construção de Compiladores

Nível: Graduação

Carga Horária: 72 horas-aula (Teórica: 20; Prática: 52)

Período Letivo: 2026-1

Turma: 06208 - CIÊNCIAS DA COMPUTAÇÃO (208)
Horário: 5 15:10 2 (CTC203), 6 10:10 2 (LIICT8)
Docente: Alvaro Junio Pereira Franco (alvaro.junio@ufsc.br)

2. Ementa

Projeto de especificação de linguagens de programação. Implementação das etapas que compreendem o processo de compilação: Análise Léxica, Análise Sintática, Análise Semântica, Geração e Otimização de Código. Evolução e tendências da área de compiladores e linguagens de programação.

3. Objetivos

3.1 Objetivo Geral:

Dotar o aluno de conhecimento básico dos conceitos e técnicas necessários para a construção de compiladores, bem como para a compreensão dos conhecimentos envolvidos no projeto de linguagens de programação e o tratamento computacional de linguagens em geral

3.2 Objetivos Específicos:

- a) Compreender os aspectos ligados ao projeto de linguagens de programação
- b) Descrever a organização arquitetural dos compiladores e seu funcionamento
- c) Compreender e implementar os principais algoritmos de análise léxica.
- d) Compreender e implementar os principais algoritmos de análise sintática
- e) Compreender e implementar os processos de análise semântica adotados nos compiladores
- f) Descrever as técnicas de recuperação de erros utilizadas nos compiladores.
- g) Identificar as formas de geração e de representação de código intermediário
- h) Compreender as técnicas de otimização de código e geração de código objeto
- i) Identificar, avaliar e utilizar ferramentas de apoio na construção de compiladores

4. Conteúdo Programático

- 1 A estrutura de um compilador [1 hora-aula]
 - 2 Linguagens de programação [1 horas-aula]
 - 2.1 Características principais
 - 3 Especificação e projeto de uma linguagem [6 horas-aula]
 - 4 Análise léxica [2 horas-aula]
 - 5 Construção de um analisador léxico [8 horas-aula]
 - 6 Análise sintática e correção de erros [6 horas-aula]
 - 7 Construção de um analisador sintático [12 horas-aula]
 - 8 Análise semântica [6 horas-aula]
 - 9 Implementação da análise semântica [12 horas-aula]
 - 10 Geração de código intermediário e otimização [6 horas-aula]
 - 11 Implementação do gerador de código [12 horas-aula]
-

5. Metodologia

Aulas expositivas teóricas e de exercícios e aulas práticas em laboratório para desenvolvimento dos conteúdos teóricos ministrados.

6. Avaliação de Aprendizagem

Teremos quatro atividades avaliativas. Todas as atividades P1, P2, P3 e P4 são valores entre 0 e 10.

A média final (MF) será calculada da seguinte forma:

$$MF = 0,15 * P1 + 0,3 * P2 + 0,25 * P3 + 0,3 * P4.$$

7. Recuperação

Conforme parágrafo 2º do artigo 70 da Resolução 17/CUn/97, o aluno com frequência suficiente (FS) e média final no período (MF) entre 3,0 e 5,5 terá direito a uma nova avaliação ao final do semestre (REC), sendo a nota final (NF) calculada conforme parágrafo 3º do artigo 71 desta resolução, ou seja: $NF = (MF + REC) / 2$.

8. Cronograma

A primeira atividade avaliativa corresponde a uma série de exercícios dados em laboratório. Sendo e1, e2, ... em os exercícios realizados em laboratório, então $P1 = J / (n/2)$ com J sendo igual a soma das (n/2) melhores notas ei. Cada ei varia entre 0 e 10.

A segunda atividade avaliativa, P2, corresponde ao desenvolvimento de um compilador (análises léxica, sintática e semântica, geração de código intermediário, geração de código e otimização). A entrega do compilador será na 15a. semana do calendário.

As terceira e quarta atividades avaliativas, serão avaliações individuais. A P3 será aplicada na 9a. semana e a P4 na 16a. semana.

A 17a. semana será utilizada para aplicar as segundas avaliações para aqueles estudantes que se ausentaram em alguma avaliação individual e que justificaram a ausência.

Haverá uma avaliação de recuperação e individual na 18a. semana. A divulgação das notas ocorrerá a partir do final da 18a. semana.

9. Bibliografia Básica

- [1] MOGENSEN, T. AE. Introduction to Compiler Design. Springer, 2011.

- [2] SU, Y., and YAN, S. Y. Principles of Compilers. Springer, 2011.
 - [3] BORNAT, R. Understanding and Writing Compilers. Springer, 1979.
 - [4] DOS REIS, A. J. Compiler Construction Using Java, JavaCC, and Yacc. Wiley, 2012.
-

10. Bibliografia Complementar

- [1] AHO, A.V.; LAM, M. S.; SETHI, R. ULLMAN, J.D. Compiladores ? Princípios, Técnicas e Ferramentas, Pearson, 2008
- [2] DELAMARO, Márcio Eduardo. Como Construir um acompilador. São Pauo, Novatec, 2004.
- [3] PRICE, Ana Maria de Alencar, TOSCANI, Simão Sirineo. Implementação de Linguagens de Programação: Compiladores. Porto Alegre, Sagra, 2004.
- [4] HOPCROFT, J. E., ULLMAM, J. D. Formal Languages and Their Relations to Automata. Addison-Wesley, 1969..
- [5] HOPCROFT, J. F., ULLMAN, J. D.. Introduction to Automata Theory, Languages and Computation. Ed. Addison-Wesley, 1979
- [6] MENESES, P. B. Linguagens Formais e Autômatos, Ed. Sagra Luzzato, 2. edição, 1998.
- [7] AHO, A. V., ULLMAN, J. D. The Theory of Parsing, Translation, and Compiling. Volume I: Parsing. Ed Prentice-Hall, Inc. 1972, 542p