



UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS DA NATUREZA
DEPARTAMENTO DE COMPUTAÇÃO

PLANO DE ENSINO

Disciplina:	Compiladores
Código:	DC/CCN042
Crédito:	2.2.0 (curso: 2020-5)
Pré-requisito:	Programação Orientada a Objetos Teoria da Computação
Carga Horária:	60h
Curso:	Bacharelado em Ciência da Computação
Professor:	Dr. Raimundo Santos Moura

Ementa

Conceitos básicos (compilação e interpretação). Estrutura de um Compilador. Análise léxica. Análise sintática descendente e ascendente. Análise semântica. Tabela de símbolos. Representações intermediárias de código. Geração de código e ambientes de execução. Noções de otimização de código. Ferramentas para construção de compiladores. Projeto de um compilador simplificado.

Objetivos

Fornecer ao aluno(a) conhecimentos para compreender e aplicar conceitos sobre implementação de linguagens de programação, com ênfase na solução de problemas encontrados no projeto de um tradutor de linguagem, independente da máquina fonte ou alvo. Desta forma, o(a) aluno(a) poderá criar mecanismos e formalismos através dos quais seja possível descrever, analisar e sintetizar linguagens de programação.

CRONOGRAMA E CONTEÚDO PROGRAMÁTICO

AULA	ASSUNTO	MÉTODO
Aula 1	Apresentação e discussão do plano de curso Introdução e Conceitos Básicos	T
Aulas 2 e 3	Um compilador Simples	
Aula 4	Análise Léxica	
Aula 5	Análise Léxica – Uso do ANTLR	P

Aulas 6 e 7	Análise Sintática	T
Aula 8	Análise Sintática – Uso do ANTLR	P
Aula 9	Integração entre Análise Léxica e Sintática	P
Aulas 10 e 11	Tradução Dirigida por Sintaxe	T
Aula 12	<i>Abstract Syntax Tree</i> – Uso do ANTLR	P
1ª AVALIAÇÃO ESCRITA		
Aula 13	Trabalho Final: Especificação	P
Aula 14	Análise semântica: Verificação de tipos	T
Aula 15	Verificação de tipos – Aula Prática	P
Aula 16	Geração de Código Intermediário	T
Aula 17	Geração de Código Intermediário – Aula Prática	P
Aulas 18 e 19	Ambiente de Execução OO	T
Aula 20	Ambiente de Execução – Aula Prática	P
Aula 21	Acompanhamento de Projeto – parte I	P
Aulas 22 e 23	Geração de Código	T
Aula 24	Geração de Código – Aula Prática	P
Aulas 25 a 28	Otimização de Código	T
2ª AVALIAÇÃO ESCRITA		
Aula 29	Acompanhamento de Projeto – parte II	P
Aula 30	Conclusão: Apresentação do Trabalho Final e discussão das notas	P

Procedimento de ensino

- Aulas expositivas virtuais síncronas e assíncronas (T)
- Aulas práticas e exercícios (P)
- Elaboração de trabalhos práticos com temas atuais

Sistemática de avaliação

Para efeito de avaliação será observada a Resolução 043/95-CEPEX que regulamenta a Verificação do Rendimento Escolar dos cursos de graduação da Universidade Federal do Piauí

Serão registradas três notas, sendo duas avaliações escritas envolvendo os conteúdos apresentados e discutidos nas aulas e uma avaliação prática (Projeto Final). Será considerado(a) aprovado(a) na disciplina o(a) aluno(a) que:

- Obter frequência igual ou superior a 75% da carga horária da disciplina;
- Obter média aritmética das três avaliações maior ou igual a 7 (sete), ou média aritmética igual ou superior a 6 (seis), resultante da média aritmética das avaliações e da nota do exame final.

O(A) aluno(a) que obtiver média aritmética das 3 avaliações inferior a 4 (quatro) será considerado(a) reprovado(a) e não realizará avaliação final. A prova final versará sobre todo o conteúdo da disciplina.

O(A) aluno(a) que não comparecer às avaliações e/ou exame final terá o direito de requerer a oportunidade de realizá-los em segunda chamada.

O(A) candidato(a) a exame de segunda chamada poderá requerê-lo por si ou por procurador legalmente constituído, mediante envio de *e-mail* ao professor da disciplina, em um prazo de 3 dias úteis, justificando através de documento o motivo da ausência.

Recursos de ensino

Serão utilizados no processo ensino-aprendizagem: quadro de acrílico, projetor multimídia, além do uso de laboratório para desenvolvimento de atividades e projetos práticos. Para a criação automática de analisadores léxicos e sintáticos, será utilizada a ferramenta ANTLR (*ANother Tool for Language Recognition*), disponível em: <https://www.antlr.org/index.html>.

Bibliografia Básica

- AHO, Alfred V.; ULLMAN, Jeffrey D.; SETHI, R. Compiladores: princípios, técnicas e ferramentas. Rio de Janeiro: LTC, 1995.

Bibliografia Complementar

- APPEL, Andrew. Modern Compiler Implementation in Java. Cambridge University Press, 1998.
- GRUNE, Dick; BAL, Henri; JACOBS, Criel; LANGENDOEN, Koen. Projeto Moderno de Compiladores. Editora Campus, 2001.
- PARR, Terence. The Definitive ANTLR Reference – Building Domain-Specific Languages. The Pragmatic Bookshelf, 2009.