



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Faculdade de Computação

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902

Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> facom@ufu.br



PLANO DE ENSINO

1. IDENTIFICAÇÃO

Componente Curricular:	Construção de Compiladores						
Unidade Ofertante:	Faculdade de Computação (FACOM)						
Código:	GBC071	Período/Série:	7o Período	Turma:			
Carga Horária:				Natureza:			
Teórica:	60h (72h/a)	Prática:	-	Total:	60h (72h/a)	Obrigatória: (X)	Optativa: ()
Professor(A):	Luiz Gustavo Almeida Martins				Ano/Semestre:	2021-2	
Observações:							

2. EMENTA

Gramáticas. Autômatos. Análise léxica. Análise sintática. Geração de código. Recuperação de erros. Meta-compiladores.

3. JUSTIFICATIVA

O compilador é um importante componente do sistema computacional, pois está intimamente ligado com a arquitetura do computador e seus sistemas (ex: sistema operacional), bem como com a metodologia de programação e o projeto das linguagens de programação. A construção de compiladores envolve ideias de diferentes áreas de ciência da computação, tais como: inteligência artificial, linguagens formais e autômatos, algoritmos e estruturas de dados, organização e arquitetura de computadores, engenharia de software, entre outras. Portanto, o seu estudo possibilita a aplicação prática de muitos conceitos teóricos adquiridos ao longo do curso. Além disso, as técnicas empregadas em compiladores também podem ser aplicadas em outros contextos computacionais, como: editores de textos, tratamento de conteúdos dinâmicos (ex: XML), entre outras aplicações que tenham entradas com formato semelhante a linguagens.

4. OBJETIVO

Objetivo Geral:

Ao final do curso, o aluno deve ser capaz de definir gramáticas concretas e implementar uma linguagem de programação simples contendo declarações de variáveis globais e de procedimentos e funções com contexto local usando as técnicas estabelecidas de construção de compiladores.

Objetivos Específicos:

- Conhecer a estrutura básica de um compilador e o funcionamento de suas etapas;
- Compreender e implementar algoritmos de análise léxica, sintática e semântica;
- Gerar representações intermediárias a partir do código fonte;
- Compreender e implementar técnicas de tratamento de erros nas diferentes etapas da compilação;
- Conhecer os ambientes de execução e seu funcionamento; e

- Compreender as técnicas de otimização e geração de código objeto.

5. PROGRAMA

1. Introdução a compiladores

a. Conceitos básicos

b. Estrutura básica de um compilador

2. Gramáticas regulares, expressões regulares e análise Léxica

a. Expressões regulares

b. Conversão AFND para AFD

c. Diagramas de transição

d. Formas de implementação de um analisador léxico

e. Tratamento de erros léxicos

f. Gerador automático de um analisador léxico (FLEX)

3. Gramáticas livres de contexto e análise sintática

a. Transformações em gramáticas

b. Algumas propriedades importantes de gramáticas: *First* e *Follow*

c. Análise *Top-Down*

- Método LL(1)

- Método com descida recursiva

d. Análise *Bottom-Up* (ou *Shift-Reduce*)

- Método LR(0)

- Método LR simples (SLR)

- Métodos LR(1) (CLR e LALR)

e. Tratamento de erros sintáticos

f. Gerador automático de um analisador sintático (Yacc)

4. Tradução dirigida por sintaxe e análise semântica

a. Construção de árvore de sintaxe abstrata

b. Construção da tabela de símbolos

c. Análise de escopo

d. Verificação de tipos

5. Ambientes de execução e geração de código

a. Pilha de execução

b. Máquinas de execução

- Máquinas de registradores

- Máquinas de pilha (máquina TAM)

c. Geração de código intermediário

d. Geração de código

6. Tópicos especiais

a. Alocação de registradores

b. Otimização de código

6. **METODOLOGIA**

O conteúdo da disciplina será ministrado através de atividades presenciais e assíncronas. A parte presencial (62 horas-aula divididas em 4 horas-aula semanais) envolve aulas expositivas e apresentação de exemplos que ilustram o uso das técnicas. Essas aulas serão ministradas nas **segundas e terças das 10:40 às 12:20**. Conforme apresentado no cronograma a seguir, a carga horária será complementada com 08 horas-aula assíncronas, destinadas ao desenvolvimento do projeto da disciplina; e 02 horas-aula presenciais extras na 15ª semana para a vista da 2ª prova.

O quadro abaixo apresenta o cronograma das atividades presenciais e assíncronas previstas e sua distribuição em quantidade de horas-aulas durante o semestre letivo:

Semana	Módulos	Atividades Presenciais Previstas	H/A Presenciais	Data/Horário Presencial	Atividades Assíncronas Previstas	H/A Assíncr.
02/05/2022	INÍCIO DO SEMESTRE	-	-	-	-	-
1ª	1	Apresentação do plano e aula expositiva	4	02 e 03/05 – 10:40 às 12:20	-	-

2ª	1 e 2	Aula expositiva	4	09 e 10/05 – 10:40 às 12:20	-	-
3ª	2	Aula expositiva	4	16 e 17/05 – 10:40 às 12:20	-	-
4ª	2	Aula expositiva	4	23 e 24/05 – 10:40 às 12:20	Desenvolvimento do projeto	2
5ª	2	Aula expositiva	4	30 e 31/05 – 10:40 às 12:20	-	-
6ª	3	Aula expositiva	4	06 e 07/06 – 10:40 às 12:20	-	-
7ª	3	Aula expositiva	4	13 e 14/06 – 10:40 às 12:20	-	-
8ª	-	Aula expositiva	4	20 e 21/06 – 10:40 às 12:20	Desenvolvimento do projeto	2
9ª	3	Aula expositiva	4	27 e 28/06 – 10:40 às 12:20	-	-
10ª	3	Aula expositiva	4	04 e 05/07 – 10:40 às 12:20	-	-
11ª	4	Aula expositiva	4	11 e 12/07 – 10:40 às 12:20	Desenvolvimento do projeto	2
12ª	4	Aula	4	18 e 19/07 –	-	-

		expositiva		10:40 às 12:20		
13 ^a	5 e 6	Aula expositiva	4	25 e 26/07 – 10:40 às 12:20	-	-
14 ^a	-	Revisão e prova	4	01 e 02/08 – 10:40 às 12:20	Desenvolvimento do projeto	2
15 ^a	-	Apresentação do projeto	4	08 e 09/08 – 10:40 às 12:20	-	-
16 ^a	-	Prova de recuperação	2	16/08 – 10:40 às 12:20	-	-
20/08/2022	TÉRMINO SEMESTRE		Total presenciais: 64 h/a			Total assíncr.: 8h/a
Carga horária total (síncrona + assíncrona)						72 h/a

TÉCNICAS DE ENSINO E FERRAMENTAS TECNOLÓGICAS

O conteúdo programático da disciplina será desenvolvido por meio de aulas expositivas, nas quais o sistema de audiovisual e o quadro negro serão utilizados como instrumentos para a disseminação do conteúdo da matéria. As aulas serão complementadas com exercícios teóricos e práticos para fixação do conhecimento ensinado.

Também pretende-se utilizar os ambientes virtuais de aprendizagem MS Teams e Moodle como ferramentas de apoio à disciplina. Uma equipe será criada no MS Teams para facilitar a troca de informação entre o professor e aluno, tanto por meio de postagens quanto via chat, e a entrega das atividades avaliativas. Para ter acesso ao MS Teams, o aluno deve possuir uma conta acadêmica criada a partir de seu e-mail institucional (ufu.br). O link de acesso à equipe da disciplina será enviado por e-mail para os alunos matriculados até a véspera da primeira aula.

O aluno também deve ter uma conta de acesso ao Moodle da UFU, pois todo material de apoio (vídeo-aulas, slides, etc.) será disponibilizado na página da disciplina nesse ambiente virtual, a qual pode ser acessada pelo link:

[Curso: Construção de Compiladores \(ufu.br\)](#)

OBS: a senha de acesso será disponibilizada em uma postagem na equipe do MS Teams.

ATENDIMENTO E COMUNICAÇÃO COM OS DISCENTES

O atendimento aos alunos ocorrerá por meio de um horário fixo extra-classe (**segundas das 09:00 às 10:30**) para esclarecimento de dúvidas que possam surgir durante a realização dos trabalhos ou em qualquer outro momento. A troca de mensagens via postagens na equipe do MS Teams também será utilizado, a fim de promover um ambiente colaborativo e coletivo de aprendizagem.

A comunicação com a turma será por meio de postagens na equipe do MS Teams. Excepcionalmente, quando se tratar de assuntos que envolvam apenas um ou um grupo de alunos, o professor pode enviar mensagens via chat ou para o e-mail cadastrado no sistema da UFU.

7. AVALIAÇÃO

O método de avaliação pretendido prevê a realização das seguintes atividades avaliativas:

- **Provas:** o plano de avaliação prevê a realização de duas provas individuais de 25 pontos (**50 pontos** no total), as quais serão realizadas em sala de aula e consistem em questões acerca do conteúdo apresentado nas aulas e desenvolvidos durante as demais atividades avaliativas. A avaliação será realizada pelo professor a partir da verificação do uso correto e eficiente das técnicas solicitadas em cada questão.

- **Trabalho:** ao longo da disciplina estão previstas 04 (quatro) entregas referentes ao projeto de um compilador para uma linguagem simplificada previamente estabelecida. Essas etapas envolvem a especificação da gramática da linguagem e dos respectivos tokens, bem como o projeto e implementação dos analisadores presentes no *front-end* do compilador. A descrição da linguagem e das diversas entregas previstas para o projeto serão disponibilizadas no ambiente virtual de acordo com o avanço da matéria. As entregas devem ser realizadas através do envio de arquivos com a resolução dentro do prazo especificado nas tarefas. A avaliação do projeto consiste na verificação do uso correto dos conceitos discutidos na disciplina, nas especificações e implementações apresentadas, bem como no correto funcionamento dos módulos do compilador quando aplicado aos casos de testes. Além disso, o aluno será submetido a uma arguição oral individual sobre seu projeto, em um horário previamente agendado com o professor na penúltima semana de aula. Essa arguição será realizada presencialmente, somente com a presença do aluno e do professor. A nota da arguição será utilizada como um fator de ponderação (*FP*), entre 0 e 100%, que será aplicado sobre a nota do projeto (*NP*) para obter a nota final do trabalho (*NT*). Isto é, a nota do aluno nesta atividade será definida por: $NT = NP \times FP$. A pontuação máxima que pode ser alcançada nesta atividade avaliativa é **50 pontos**.

- **Recuperação:** o plano de avaliação prevê a realização de uma prova individual de **100 pontos** como mecanismo de recuperação de aprendizagem aos alunos com frequência mínima de 75% (setenta e cinco por cento) na disciplina e que não atingiram a pontuação necessária para a aprovação, conforme previsto no Art. 141 da Resolução CONGRAD No 46/2022. A prova será realizada em sala de aula, englobará todo o conteúdo da disciplina e sua avaliação será pautada pelo uso correto e eficiente das técnicas de compilação solicitadas em cada questão. A nota final do aluno será a média aritmética entre a nota da prova de recuperação e aquela obtida pelo aluno ao longo do semestre.

A fim de destacar a distribuição dessas atividades ao longo do semestre letivo, é apresentado a seguir o cronograma de avaliação:

Nro	Data de Entrega	Hora	Descrição	Pontos
1	21/06/2022	10:40 – 12:20	1ª prova	25
2	02/08/2022	10:40 –		25

		12:20	2ª prova	
3	Entregas das 4 etapas do projeto: 30/05; 26/06; 17/07; e 07/08/2022 Apresentação: 08 e 09/08/2022	Até 23:59 10:40 – 12:20	Trabalho (Projeto)	50
			TOTAL:	100
4	16/08/2022	10:40 – 12:20	Recuperação	100

A assiduidade nas atividades assíncronas será avaliada mediante a entrega das etapas do projeto.

8. BIBLIOGRAFIA

Básica

- AHO, A.; et al. Compiladores: Princípios, Técnicas e Ferramentas, 2a. ed. Pearson, 2007.
- LOUDEN, K. Compiladores: Princípios e Práticas, Thomson Pioneira, 2004.
- GRUNE, D.; et al. Projeto Modemo de Compiladores: implementação e aplicações, Campus, 2001.

Complementar

- APPEL, A. W. Modem Compiler Implementation in Java, 2a ed. Cambridge Univ. Press, 2002.
- WATT, D.; BROWN, D. Programming Language Processors in Java: compilers and interpreters, Prentice Hall, 2000.
- DELAMARO, M. Como construir um compilador utilizando ferramentas Java, Novatec, 2004.
- COOPER, K.D.; TORCZON, L. Construindo compiladores, 2a. ed. Elsevier, 2012.
- RICARTE, I. Introdução à Compilação, Elsevier, 2008.
- dos SANTOS, P. R.; LAGLOIS, T. Compiladores: da teoria à prática, LTC, 2018.

Material de apoio com acesso remoto:

- Torben Ægidius Mogensen, Basics of Compiler Design (livro disponibilizado on-line pelo autor em http://hjemmesider.diku.dk/~torbenm/Basics/basics_lulu2.pdf), acesso em 10/12/2020.
- TEUFEL, B.; SCHMIDT, S.; TEUFEL, T. C² Compiler Concepts, Springer-Verlag, 1993. (disponível pela plataforma Cafe/CAPES), acesso em 27/09/2020.

- WAITE, W. M.; GOOS, G. Compiler Construction, Springer-Verlag, 1984. (disponível pela plataforma Cafe/CAPES), acesso em 27/09/2020.
- Ruslan's Blog - Let's Build A Simple Interpreter. (<https://ruslanspivak.com/lbasi-part1/>), acesso em 10/12/2020.

9. DIREITOS AUTORAIS

Todo o material produzido e divulgado pelo docente, como vídeos, textos, arquivos de voz, etc., está protegido pela Lei de Direitos Autorais, a saber, a lei nº 9.610, de 19 de fevereiro de 1998, pela qual fica vetado o uso indevido e a reprodução não autorizada de material autoral por terceiros. Parágrafo Único: responsáveis pela reprodução ou uso indevido do material de autoria dos docentes ficam sujeitos às sanções administrativas e as dispostas na Lei de Direitos Autorais.

10. APROVAÇÃO

Aprovado em reunião do Colegiado realizada em: ____/____/____

Coordenação do Curso de Graduação: _____