



Plano de Ensino para o Ano Letivo de 2020

IDENTIFICAÇÃO		
Disciplina: Linguagens Formais, Autômatos e Compiladores		Código da Disciplina: ECM253
Course: Formal Languages, Automata and Compilers		
Materia: Lenguajes Formales, Autómatas y Compiladores		
Periodicidade: Anual	Carga horária total: 160	Carga horária semanal: 02 - 00 - 02
Curso/Habilitação/Ênfase: Engenharia de Computação	Série: 3	Período: Diurno
Professor Responsável: Marco Antonio Furlan de Souza	Titulação - Graduação Engenheiro Eletricista	Pós-Graduação Mestre
Professores: Marco Antonio Furlan de Souza	Titulação - Graduação Engenheiro Eletricista	Pós-Graduação Mestre
OBJETIVOS - Conhecimentos, Habilidades, e Atitudes		
<p>Conhecimentos:</p> <p>C1. Compreender os fundamentos de Matemática Discreta necessários ao aprendizado de Linguagens Formais, Autômatos e Compiladores;</p> <p>C2. Entender o embasamento teórico necessário para o projeto e implementação de linguagens de programação;</p> <p>C3. Projetar e implementar linguagens de programação utilizando ferramentas de alto nível;</p> <p>C4. Identificar características necessárias às linguagens de programação, de acordo com sua finalidade e sistema hospedeiro;</p> <p>C5. Compreender os estágios típicos no projeto de um interpretador / compilador: varredura, análise sintática, análise semântica, otimizador e gerador de código.</p> <p>Habilidades:</p> <p>H1. Aplicar conhecimentos da teoria de Matemática Discreta na solução de problemas associados à linguagens formais, autômatos e compiladores;</p> <p>H2. Implementar, com alguma linguagem de programação, as principais estruturas da teoria de Linguagens Formais, Autômatos e Compiladores;</p> <p>H3. Praticar com ferramentas de software específicas para a construção de interpretadores e compiladores;</p> <p>H4. Implementar interpretadores / compiladores destinados à pequenas linguagens de programação ou conjunto de expressões;</p> <p>H5. Reconhecer a aplicabilidade da teoria de de Linguagens Formais, Autômatos e Compiladores em diversos segmentos da Computação (Internet, editores de texto, sistemas embarcados, linguagens específicas a um determinado domínio de aplicação etc).</p> <p>Atitudes:</p> <p>A1. Desenvoltura na criação de softwares básicos como aqueles relacionados à</p>		



disciplina (reconhecedores, compiladores, interpretadores, carregadores, montadores, pré-processadores, depuradores, perfiladores);

A2. Reconhecer os requisitos necessários para qualquer sistema de software que necessite de algum tipo de interpretação ou compilação.

EMENTA

ELEMENTOS DE MATEMÁTICA DISCRETA: Conjuntos e Relações, Funções, Relações de Equivalência, Indução Matemática, Lógica Proposicional e de Predicados, Aplicações da Lógica de Primeira Ordem, Revisão de Grafos, Dígrafos e Árvores. ELEMENTOS DE ANÁLISE DE ALGORITMOS: aplicações de conceitos de Matemática Discreta à Análise de Algoritmos. INTRODUÇÃO AOS MODELOS TEÓRICOS DE COMPUTAÇÃO: Autômatos Finitos, Máquinas de Turing, Computabilidade, Problemas de Decisão, Problema da Parada, Complexidade Computacional, Problemas Intratáveis e Completude NP. GRAMÁTICAS E LINGUAGENS, Relação entre linguagens, Autômatos e Máquinas de Turing, Classificação das Gramáticas: a hierarquia de Chomsky, Gramáticas Regulares, Gramáticas Livres de Contexto. ELEMENTOS DA TEORIA DOS COMPILADORES: Varredura de Código, Análise Sintática, Metalinguagem EBNF, Análise Semântica, Gramáticas de Atributos, Tabela de Símbolos, Tipos e Verificação de Tipos, Ambientes de Execução, Organização da Memória, Mecanismos de Passagem de Parâmetros, Técnicas de Geração de Código, Otimizações de Código.

SYLLABUS

ELEMENTS OF DISCRETE MATHEMATICS: Sets and Relations, Functions, Equivalence Relations, Mathematical Induction, Propositional and Predicate Logic, Applications of First-Order Logic, Review of Graphs, Digraphs and Trees. ELEMENTS OF ALGORITHM ANALYSIS: Applications of Discrete Mathematics to Algorithm Analysis. INTRODUCTION TO THEORETICAL COMPUTING MODELS: Finite Automata, Turing Machines, Computability, Decision Problems, Halt Problem, Computational Complexity, Intractable Problems and NP Completeness. GRAMMARS AND LANGUAGES, Relationship among languages, automata and Turing Machines, Grammar Classification: Chomsky Hierarchy, Regular Grammars, Context Free Grammars. ELEMENTS OF COMPILER THEORY: Code scanning, Parsing, EBNF Metalanguage, Semantic Analysis, Attribute Grammars, Symbol Tables, Types and Type Checking, Execution Environments, Memory Organization, Parameter passing mechanisms, Code generation techniques, Code optimization.

TEMARIO

ELEMENTOS DE MATEMÁTICA DISCRETA: Conjuntos y relaciones, funciones, relaciones de equivalencia, inducción matemática, lógica proposicional y de predicados, Aplicaciones de la lógica de primer orden, Revisión de gráficos, dígrafos y árboles. ELEMENTOS DE ANÁLISIS DE ALGORITMOS: Aplicaciones de la Matemática Discreta al Análisis de Algoritmos. INTRODUCCIÓN A LOS MODELOS TEÓRICOS DE COMPUTACIÓN - autómatas finitos, Máquinas de Turing, computabilidad, Problemas de Decisión, El Problema de la Parada, Complejidad Computacional, Problemas Intratables y NP-completitud. GRAMÁTICAS Y LENGUAJES - Relación entre las lenguajes, autómatas y Máquinas de Turing, Clasificación de las Gramáticas: Jerarquía de Chomsky, Gramáticas Regulares, Gramáticas libres de Contexto. ELEMENTOS DE LA TEORÍA DE LOS COMPILADORES: processo de barrido, análisis



sintático, metalenguaje EBNF, análisis semántico, tablas de símbolos, Gramáticas de atributos, Tipos y verificación de tipos, ambientes de ejecución, organización de la memoria, mecanismos de transmisión de parámetros, técnicas de generación de código, optimización de código.

ESTRATÉGIAS ATIVAS PARA APRENDIZAGEM - EAA

Aulas de Teoria - Sim

Aulas de Laboratório - Sim

LISTA DE ESTRATÉGIAS ATIVAS PARA APRENDIZAGEM

- Peer Instruction (Ensino por pares)
- Sala de aula invertida
- Project Based Learning
- Problem Based Learning

METODOLOGIA DIDÁTICA

O curso será ministrado em um laboratório de computação com máquinas em que tenham sido instaladas ferramentas de software específicas para a disciplina, tais como ambientes integrados de programação, compiladores, geradores de varredura e geradores de compiladores. Para a apresentação dos tópicos das aulas será necessário uma máquina para o instrutor e um projetor eletrônico.

CONHECIMENTOS PRÉVIOS NECESSÁRIOS PARA O ACOMPANHAMENTO DA DISCIPLINA

É necessário o conhecimento de alguma linguagem de programação tais como Pascal, C, C++, Java ou Python, bem como o conhecimento básico de algoritmos e de estruturas de dados.

CONTRIBUIÇÃO DA DISCIPLINA

A teoria e a prática envolvidas neste curso permitirão ao engenheiro de computação propor soluções para problemas computacionais correlatos com diferentes graus de complexidade, de interpretação de expressões regulares ao projeto de pequenas linguagens de programação, que poderão ser compiladas ou interpretadas. Além disso, a disciplina possui relacionamentos com outras disciplinas do currículo da Engenharia de Computação, principalmente com Arquitetura de Computadores e Sistemas Operacionais, oferecendo ao aluno uma oportunidade de aplicar os conceitos aprendidos nessas disciplinas na solução de problemas associados à tradução, interpretação e compilação de linguagens de programação criando, assim, uma oportunidade de integrar seus conhecimentos com os conhecimentos aprendidos nessas disciplinas.

BIBLIOGRAFIA

Bibliografia Básica:

AHO, Alfred V. Compiladores: princípios, técnicas e ferramentas. VIEIRA, Daniel (Trad.). 2. ed. São Paulo: Pearson Addison-Wesley, c2008. 634 p. ISBN 978858639249.

COOPER, Keith D.; TORCZON, Linda. Construindo compiladores. VIEIRA, Daniel (Trad.). 2. ed. Rio de Janeiro: Elsevier, c2014. 656 p. ISBN 9788535255645.



GERSTING, Judith L. Fundamentos matemáticos para a ciência da computação. 4. ed. Rio de Janeiro, RJ: LTC, 2001. 538 p. ISBN 85-216-1263-X.

Bibliografia Complementar:

DOS REIS, Anthony J. Compiler construction using Java, Java CC, and Yacc. Hoboken, N. J: John Wiley & Sons, c2012. 635 p. ISBN 9780470949597.

HOLUB, Allen I. Compiler design in C. Englewood Cliffs, NJ: Prentice-Hall, c1990. 924 p. ISBN 0131550454.

LOUDEN, Kenneth C. Compiladores: princípios e práticas. SILVA, Flávio Soares Corrêa (Trad.). São Paulo: Pioneira Thomson Learning, c2004. 569 p. ISBN 8522104220.

AValiação (conforme Resolução RN CEPE 16/2014)

Disciplina anual, com trabalhos e provas (quatro e duas substitutivas).

Pesos dos trabalhos:

k_1 : 1,0 k_2 : 1,0

Peso de MP(k_p): 0,6

Peso de MT(k_T): 0,4

INFORMAÇÕES SOBRE PROVAS E TRABALHOS

Os trabalhos consistirão de pequenos projetos de programação em linguagem a ser definida com os alunos e com auxílio de ferramentas de construção de compiladores apropriadas. Serão propostos pelo menos quatro trabalhos que deverão ser desenvolvidos, preferencialmente, em sala de aula.

**OUTRAS INFORMAÇÕES**

Esta disciplina empregará as técnicas de Problem/Project Based Learning e Peer Instruction. Desse modo, após uma discussão dos conceitos envolvidos, os alunos deverão resolver problemas propostos e também participar em um projeto multidisciplinar a ser desenvolvido durante as aulas.

**SOFTWARES NECESSÁRIOS PARA A DISCIPLINA**

- Java JDK 12:
<https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html>
- Apache NetBeans: <https://netbeans.apache.org/download/nb112/nb112.html>
- Python 3: <https://www.python.org/downloads/>
- Interpretador SWI Prolog: <http://www.swi-prolog.org/download/stable>
- Visual Studio Code: <https://code.visualstudio.com/>
- No Visual Studio Code, instalar a extensão "Java Studio Pack"
- No Visual Studio Code, instalar a extensão "VSC-Prolog"



APROVAÇÕES

Prof.(a) Marco Antonio Furlan de Souza
Responsável pela Disciplina

Prof.(a) Angelo Sebastiao Zanini
Coordenador do Curso de Engenharia de Computação

Data de Aprovação:



PROGRAMA DA DISCIPLINA		
Nº da semana	Conteúdo	EAA
1 T	Recepção e Integração dos Calouros	0
1 L	Recepção e Integração dos Calouros	0
2 T	Apresentação/Lógica Proposicional	41% a 60%
2 L	Lógica Proposicional	41% a 60%
3 T	Lógica de Predicados	41% a 60%
3 L	Lógica de Predicados	41% a 60%
4 T	Lógica e Prolog	41% a 60%
4 L	Lógica e Prolog	41% a 60%
5 T	Lógica e Prolog	41% a 60%
5 L	Lógica e Prolog	41% a 60%
6 L	Conjuntos, relações e funções	41% a 60%
6 T	Conjuntos, relações e funções	41% a 60%
7 T	Conjuntos, relações e funções	41% a 60%
7 L	Conjuntos, relações e funções	41% a 60%
8 L	Período de Provas P1	0
8 T	Período de Provas P1	0
9 L	Feriado - Paixão de Cristo	0
9 T	Dia não letivo	0
10 T	Autômatos finitos	41% a 60%
10 L	Autômatos finitos	61% a 90%
11 T	Autômatos finitos	41% a 60%
11 L	Autômatos finitos	61% a 90%
12 L	Feriado - Dia do Trabalhador	0
12 T	Autômatos de pilha	41% a 60%
13 T	Autômatos de pilha	41% a 60%
13 L	Autômatos de pilha	61% a 90%
14 T	Máquinas de Turing	41% a 60%
14 L	Máquinas de Turing	61% a 90%
15 T	Linguagens e Gramáticas	41% a 60%
15 L	Linguagens e Gramáticas	61% a 90%
16 T	Sistema de varredura	41% a 60%
16 L	Sistema de varredura	61% a 90%
17 T	Sistema de varredura	41% a 60%
17 L	Sistema de varredura	61% a 90%
18 T	Feriado - Corpus Christi	0
18 L	Revisão	0
19 L	Período de provas P2	0
19 T	Período de provas P2	0
20 L	Entrega de trabalhos	0
20 T	Entrega de trabalhos	0
21 L	Férias	0
21 T	Férias	0
22 L	Férias	0
22 T	Férias	0



23 L	Período de provas PS	0
23 T	Período de provas PS	0
24 T	Análise sintática descendente	41% a 60%
24 L	Análise sintática descendente	61% a 90%
25 T	Análise sintática descendente	41% a 60%
25 L	Análise sintática descendente	61% a 90%
26 T	Análise sintática ascendente	41% a 60%
26 L	Análise sintática ascendente	61% a 90%
27 T	Análise sintática ascendente	41% a 60%
27 L	Análise sintática ascendente	61% a 90%
28 T	Análise semântica	41% a 60%
28 L	Análise semântica	61% a 90%
29 T	Análise semântica	41% a 60%
29 L	Análise semântica	61% a 90%
30 L	Período de provas P3	0
30 T	Período de provas P3	0
31 T	Ambientes de execução	41% a 60%
31 L	Ambientes de execução	61% a 90%
32 T	Ambientes de execução	41% a 60%
32 L	Ambientes de execução	61% a 90%
33 T	Geração de código	41% a 60%
33 L	Geração de código	61% a 90%
34 T	Geração de código	41% a 60%
34 L	Geração de código	61% a 90%
35 T	Geração de código	41% a 60%
35 L	Geração de código	61% a 90%
36 T	Geração de código	41% a 60%
36 L	Geração de código	61% a 90%
37 T	Revisão	0
37 L	Revisão	0
38 T	Período de provas P4	0
38 L	Feriado - Consciência Negra / Vestibular Mauá	0
39 L	Período de provas P4	0
39 T	Período de provas P4	0
40 L	Atividades de atendimento e orientação	0
40 T	Atividades de atendimento e orientação	0
41 T	Atividades de atendimento e orientação	0
41 L	Período de provas PS2	0
Legenda: T = Teoria, E = Exercício, L = Laboratório		