



DISCIPLINA: ÉTICA E CIDADANIA

CODICRED: 1501A-04

CARGA HORÁRIA: 60h/a

ANO/SEMESTRE: 2020/1

EMENTA

Abordagem de problemas e conceitos filosóficos relacionados à moralidade e ao exercício da cidadania. Introdução aos principais temas e problemas de Ética, presentes na História da Filosofia. Reflexão filosófica sobre o ideal democrático, a cidadania e os direitos humanos.

OBJETIVOS

Introduzir os problemas e conceitos fundamentais da filosofia, assumindo a Filosofia Moral como ponto de partida. Estabelecer a relação entre as principais posições éticas e políticas contemporâneas e sua origem histórica.

Compreender os conceitos e argumentos fundamentais das principais teorias éticas e políticas contemporâneas. Compreender como as teorias éticas e políticas são aplicadas sobre algumas questões fundamentais da existência humana.

CONTEÚDO PROGRAMÁTICO

I) Introdução à Filosofia

- 1) O que é a Filosofia?
- 2) A História da Filosofia
- 3) A Ética como disciplina filosófica

II) Algumas concepções da Ética

- 1) A Ética socrática e platônica. Virtude e conhecimento.
- 2) A Ética aristotélica. Virtude e felicidade.
- 3) O Hedonismo. A importância do prazer.
- 4) O Utilitarismo. A máxima utilidade das decisões.
- 5) O Formalismo kantiano. O que eu devo fazer?
- 6) Outras Escolas. Filosofia da existência e Filosofia da alteridade

III) O Conceito de Cidadania

- 1) Introdução: O ideal democrático
- 2) Teoria da Cidadania
- 3) Cidadania e Direitos Humanos
- 4) Cidadania e globalização

IV) Tópicos de Ética aplicada e cidadania

METODOLOGIA

Aulas expositivo-dialogadas, trabalhos em grupo, exploração de textos e discussões dirigidas que visam à introdução e desenvolvimento dos tópicos do programa.

AVALIAÇÃO

Rigor na análise das teorias; coerência lógica; utilização precisa da terminologia filosófica; articulação verbal e postura na apresentação; capacidade de mobilização da turma e promoção de discussão no grupo. Serão consideradas, ainda, as habilidades para estabelecer relações, fazer comparações e detectar contradições. Produção escrita (60%)
2 Provas (P1 e P2) Prática (30%) Apresentação de Seminários em Grupo (T1)
Cálculo de G1: $P1 \cdot 0,3 + P2 \cdot 0,3 + T1 \cdot 0,4$.

BIBLIOGRAFIA BÁSICA



KYMLICKA, W. Filosofia Política Contemporânea. São Paulo: Martins Fontes, 2006.

SINGER, P. Ética Prática. São Paulo: Martins Fontes, 2002.

VÁZQUEZ, Adolfo Sánchez. Ética. Rio de Janeiro: Civilização Brasileira, 1999.

BIBLIOGRAFIA COMPLEMENTAR

ARISTÓTELES. Ética a Nicômaco. São Paulo: EDIPRO, 2002.

BIELEFELDT, H. Filosofia dos Direitos Humanos. São Leopoldo: UNISINOS, 2003.

CUNNINGHAM, F. Teorias da Democracia. Porto Alegre: ARTMED, 2009.

RACHELS, J. Os Elementos de Filosofia Moral. Barueri: Manole, 2006.

WOLFF, J. Introdução à Filosofia Política. Lisboa: Gradiva, 2004.

OUTRA BIBLIOGRAFIA

SANDEL, M.J. Justiça: o que é fazer a coisa certa. Rio de Janeiro: Civilização Brasileira, 2011.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

MATEMATICA DISCRETA (CC)

CODCRED	CARGA HORÁRIA	MÓDULO
4115G-04	60	60

EMENTA:

Teoria dos Conjuntos. Técnicas Básicas de Demonstração. Funções Totais, Parciais e Relações. Relações de Ordem e Equivalência. Conjuntos Indutivos e Funções Recursivas. Indução sobre os Naturais. Princípios de Análise Combinatória.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

- entendimento da notação formal em matemática;
- clareza e objetividade de raciocínio;
- capacidade de desenvolver raciocínio dedutivo e indutivo.

CONTEÚDO:

1. TEORIA DE CONJUNTOS E TÉCNICAS BÁSICAS DE DEMONSTRAÇÃO

Introdução e conceitos básicos: conjunto, elemento e relação de pertinência. Formas de representação de conjuntos: por extensão, por compreensão, por gráficos e por diagrama de Venn. Conjunto vazio e conjunto universo. Intervalos. Relações entre conjuntos: inclusão, inclusão estrita e igualdade. Operações entre conjuntos: união, intersecção, diferença e complementação. Propriedades das operações. Leis de De Morgan. Diferença simétrica. Sequências ordenadas de elementos. Produto cartesiano de dois conjuntos: propriedades.

2. FUNÇÕES TOTAIS, PARCIAIS E RELAÇÕES

Relação de A em B. Domínio e imagem de uma relação. Operações com relações. Funções totais e parciais. Domínio, imagem e contradomínio de uma função. Funções reais e funções de variável real. Maior domínio de uma função. Determinação de imagem de uma função. Gráfico de uma função. Funções inversíveis. Funções injetoras. Funções sobrejetoras. Funções bijetoras.



3. RELAÇÕES DE ORDEM E EQUIVALÊNCIA

Relações em A. Propriedades das relações em A: reflexividade, irreflexividade, transitividade, simetria, assimetria e anti-simetria. Relações de equivalência em A. Classes de equivalência. Relações de ordem em A. Ordem total e ordem parcial.

4. CONJUNTOS INDUTIVOS, FUNÇÕES RECURSIVAS E INDUÇÃO SOBRE OS NATURAIS

Construção de Leis Gerais: recursão, definições indutivas, conjuntos fechados. Indução matemática: princípio de indução finita, etapas da prova por indução matemática, base de indução (BI), passo da indução, conclusão. Justificativa da validade do princípio de indução matemática. Justificativa da necessidade do passo de indução. Justificativa da necessidade da base da indução. Somatórios: propriedades válidas em somatórios, indução em somatórios, indução em fórmulas de recorrência.

5. PRINCÍPIOS DE ANÁLISE COMBINATÓRIA

Análise combinatória: permutação, arranjo e combinação.

BIBLIOGRAFIA BÁSICA:

GERSTING, Judith L. Fundamentos matemáticos para a ciência da computação. 4. ed. Rio de Janeiro: LTC, 2001. 538 p.

MENEZES, Paulo Blauth. Matemática discreta: para computação e informática. Porto Alegre: Sagra Luzzato, 2005. 258 p.

MENEZES, Paulo Blauth; TOSCANI, Laira Vieira; LOPES, Javier García. Aprendendo matemática discreta com exercícios. Porto Alegre: Bookman, 2009. 356 p.

BIBLIOGRAFIA COMPLEMENTAR:

BALAKRISHNAN, V. K. Introductory discrete mathematics. New York (NY): Dover Publications, 2014. 236 p.

EPP, Susanna S. Discrete mathematics with applications. 4. ed. Boston: Books/Cole, 2011. ca900 p.

LIPSCHUTZ, Seymour. Theory and problems of essential computer mathematics. New York (NY): MacGraw-Hill, 1987. 357 p.

STEIN, Clifford; DRYSDALE, Robert; BOGART, Kenneth. Matemática discreta para ciência da computação. São Paulo: Pearson, 2013.

STOLL, Robert. Set theory and logic. New York, NY: Dover, 1979. 474 p.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

FUNDAMENTOS DE PROGRAMACAO

CODCRED	CARGA HORÁRIA	MÓDULO
4611C-06	90	30

EMENTA:

Discussão sobre algoritmos como ferramenta para a descrição da solução de problemas e os fundamentos necessários para sua implementação usando linguagens de programação. Exame das noções de variáveis e tipos. Estudo de programas sequenciais, comandos de entrada e saída e o processo de compilação. Construção de programas iterativos e uso do conceito de subprograma. Estudo da noção de estrutura de dados usando arranjos e matrizes e conceitos básicos de programação orientada a objetos.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Conhecer e utilizar de forma precisa conceitos e termos relacionados a algoritmos, linguagens de programação e a construção, teste e depuração de programas.
2. Desenvolver as competências e habilidades para a formular problemas, pensar sobre soluções e expressar estas soluções na forma de um programa em uma linguagem de programação.
3. Descrever algoritmos utilizando diagramas e pseudocódigo.
4. Construir abstrações para procedimentos, funções e tipos abstratos de dados utilizando os conceitos de classes e objetos.
5. Compreender e utilizar as técnicas fundamentais de teste e depuração de programas.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Algoritmos, Linguagens de Programação e Programas

- 1.1. Conceitos de programa de computador e organização de computadores
- 1.2. Definição de compilador e de interpretador
- 1.3. Projeto de algoritmos



Nº DA UNIDADE: 02

CONTEÚDO: Variáveis e tipos

- 2.1. Tipos de dados
 - 2.1.1 Tipos primitivos
 - 2.1.2 Strings
- 2.2. Variável e constante
- 2.3. Expressões aritméticas
 - 2.3.1. Operadores aritméticos
 - 2.3.2. Prioridades
 - 2.3.3. Funções matemáticas
 - 2.3.4. Conversão de tipos numéricos
- 2.4. Comandos
 - 2.4.1. Atribuição
 - 2.4.2. Entrada via console
 - 2.4.3. Saída via console

Nº DA UNIDADE: 03

CONTEÚDO: Comandos condicionais

- 3.1. Expressões lógicas
 - 3.1.1. Operadores Relacionais
 - 3.1.2. Operadores lógicos
 - 3.1.3. Tabela verdade
- 3.2. Seleção simples (comando if)
- 3.3. Seleção composta (comando if-else)
- 3.4. Seleção aninhada
- 3.5. Seleção múltipla (comando switch-case)

Nº DA UNIDADE: 04

CONTEÚDO: Iteração

- 4.1. Repetição simples
 - 4.1.1. Comando while
 - 4.1.2. Inicialização, contadores e acumuladores
 - 4.1.3. Comando for, break, continue
 - 4.1.4. Comando do-while
- 4.2. Repetição aninhada

Nº DA UNIDADE: 05

CONTEÚDO: Classes e objetos



- 5.1. Tipos abstratos de dados
- 5.2. Programação orientada a objetos
- 5.3. Conceito de classe, abstração e encapsulamento
- 5.4. Conceito de objeto
- 5.5. Definição de atributos
- 5.6. Métodos
 - 5.6.1. Definição, Parâmetros e Retorno
 - 5.6.2. Método Construtor
 - 5.6.3. Escopo de variáveis
 - 5.6.4. Recursão
 - 5.6.5. Teste Unitário
- 5.7. Referências para objetos

Nº DA UNIDADE: 06

CONTEÚDO: Arquivos

- 6.1. Arquivos: conceitos
- 6.2. Arquivos texto
- 6.3. Tratamento de exceções de entrada e saída
- 6.4. Argumentos de linha de comando

Nº DA UNIDADE: 07

CONTEÚDO: Arranjos

- 7.1. Unidimensionais
 - 7.1.1. Conceito
 - 7.1.2. Declaração e uso
 - 7.1.3. Passagem por parâmetro
 - 7.1.4. Algoritmos de preenchimento, soma e média, máximo e mínimo, pesquisa linear, remoção de elemento, inserção de elemento, troca de posição de elementos, cópia, inversão, inserção ordenada, pesquisa binária, ordenação, união, intersecção, diferença, etc.)
- 7.2. Bidimensionais
 - 7.2.1. Conceito
 - 7.2.2. Declaração e uso
 - 7.2.3. Passagem por parâmetro
 - 7.2.4. Algoritmos de operações matemáticas sobre matrizes
- 7.3. De arranjos
 - 7.2.1. Conceito
 - 7.2.2. Declaração e uso
 - 7.2.3. Passagem por parâmetro

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:



AVALIAÇÃO:

$$G1 = (P1 + 2*P2 + 3*P3 + MT) / 7$$

Onde:

P1 – conteúdos até a unidade 4

P2 – conteúdos até a unidade 6

P3 – conteúdos até a unidade 7

MT – média ponderada de tarefas realizadas ao longo do semestre (pesos das tarefas deve seguir os pesos das provas)

Observações:

No que tange as tarefas, sugere-se que o grau de complexidade das mesmas acompanhe a evolução dos conteúdos abordados ao longo da disciplina e que sejam todas realizadas individualmente.

BIBLIOGRAFIA BÁSICA:

1. □ FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3ª ed. São Paulo: Prentice Hall, 2005. 218 p.
2. □ HORSTMANN, C. Java for everyone: late object. 2nd edition. New Jersey: Wiley, 2011. 624 p.
3. □ HUBBARD, J. R. Teorias e problemas de programação com Java. 2ª ed. Porto Alegre: Bookman, 2006. 328 p.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ AGUILAR, L. J. Fundamentos de programação: algoritmos, estruturas de dados e objetos. 3ª ed. Porto Alegre: McGraw-Hill, 2008. 720 p.
2. □ AHO, A. V. Foundations of computer science. New York: Computer Science Press, 1998. 786 p.
3. □ BARNES, D.; KOLLING, M. Programação orientada a objetos com Java: uma introdução prática usando o BlueJ. 4ª ed. São Paulo: Pearson, 2009. 455 p.
4. □ CORMEN, T. H. Desmistificando algoritmos. Rio de Janeiro: Campus, 2013. 200 p.
5. □ HORSTMANN, C. Conceitos de computação com Java. 5ª ed. Porto Alegre: Bookman, 2009. 720 p.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

LOGICA PARA COMPUTACAO

CODCRED	CARGA HORÁRIA	MÓDULO
4611E-04	60	60

EMENTA:

Estudos principais conceitos da Lógica Matemática e de sua aplicação na Ciência da Computação. Exame e discussão dos conceitos de sintaxe e semântica, equivalência, relação de consequência lógica, relação de dedutibilidade, cálculo de dedução natural e outros sistemas de dedução tanto para a lógica proposicional como para a lógica de predicados. Teorias de primeira ordem e provas por indução. Fundamentação e aplicação dos conceitos subjacentes à utilização do raciocínio automático na computação.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Mostrar a evolução da lógica a partir dos sistemas axiomáticos da geometria e da lógica aristotélica, passando pelos trabalhos de Leibniz, Hilbert e Gödel, até culminar com o nascimento da ciência da computação através de Alan Turing.
2. Manipular os principais conceitos da lógica proposicional, em particular as noções de sintaxe, semântica, relação de consequência lógica, prova e teorema;
3. Utilizar os principais conceitos da lógica de primeira ordem, em particular as noções de sintaxe, semântica, relação de consequência lógica, prova e teorema.
4. Compreender os princípios básicos do raciocínio automático e suas aplicações, que podem incluir: especificação de sistemas, verificação de programas, programação lógica e verificação de modelos.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Matemática, Lógica e Computação

- 1.1. Lógica Aristotélica
- 1.2. Lógica no Século XIX (Boole, Leibniz, Peano e Frege)
- 1.3. Lógica no Século XX (Hilbert e Gödel)
- 1.4. Alan Turing e o nascimento da Computação



Nº DA UNIDADE: 02

CONTEÚDO: Lógica Proposicional

- 2.1. Introdução: as noções de argumento, validade e suas aplicações
- 2.2. Sintaxe de linguagens proposicionais
- 2.3. Semântica das Linguagens Proposicionais
- 2.4. Tautologias, Contradições e Fórmulas Satisfatíveis.
- 2.5. Conjecturas e Consequência Semântica
- 2.6. Sistema de Dedução Natural
- 2.7. Outros Sistemas de Dedução
- 2.8. Completude e Correção dos Sistemas de Dedução
- 2.9. Raciocínio automático em Lógica Proposicional

Nº DA UNIDADE: 03

CONTEÚDO: Lógica de Predicados (Primeira Ordem)*

- 3.1. Sintaxe de linguagens da Lógica de Predicados
- 3.2. Semântica modelo-teórica ou de Tarski
- 3.3. Satisfatibilidade, validade, interpretação, modelo
- 3.4. Conjecturas e Consequência Lógica
- 3.5. Sistema de Dedução Natural
- 3.6. Completude e Correção dos Sistemas de Dedução
- 3.7. Teorias de Primeira Ordem e Teorias Indutivas

Nº DA UNIDADE: 04

CONTEÚDO: Aplicações da Lógica na Computação

- 4.1. Lógica como Formalismo para Especificação de Sistemas
- 4.2. Verificação de Programas
- 4.3. Verificação de Modelos
- 4.4. Fundamentos do Raciocínio Automático*,**
 - 4.4.1. Cláusulas e Formas Clausais
 - 4.4.2. Normalização e Skolemização
 - 4.4.3. Unificação e Resolução
 - 4.4.4. Programação Lógica

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 (SI, ES) = (P1 + P2 + P3 + T) /4$$

$$G1(CC) = (P1 + 2*P2 + T) /4$$

Onde:



P1 – unidades 1 e 2;

P2 – para os cursos de Sistemas de Informação e Engenharia de Software, a unidade 3; para o Curso de Ciência da Computação 3 e 4.

P3 – para os cursos de Sistemas de Informação e Engenharia de Software unidade 4;

T – média aritmética ou ponderada das tarefas aplicadas durante o semestre.

BIBLIOGRAFIA BÁSICA:

1. □ HUTH, M. R. A; Ryan, M. D. Lógica em Ciência da Computação: Modelagem e Argumentação sobre Sistemas. 2ª ed. LTC, 2008. 322p.
2. □ MORTARI, C. A. Introdução à Lógica. Ed. UNESP, 2001
3. □ SOUZA, J.N. Lógica para Ciência da Computação: Uma Introdução Concisa. 2ª ed. Campus, 2008.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ GABBAY, D. M. Elementary Logics: a Procedural Perspective. Prentice-Hall, 1998.
2. □ GENESERETH, M. R.; Nilsson, N. J. Logical Foundations of Artificial Intelligence. Palo Alto, Morgan Kaufmann Publishers, 1987.
3. □ GERSTING, Judith L. Fundamentos Matemáticos para a Ciência da Computação, 3ª Edição, LTC Editora, 1995.
4. □ MENDELSON, B. Introduction to Mathematical Logic. Princeton, NJ, Van Nostrand, 1964.
5. □ NOLT, J.; Rohatyn, D. Lógica. Coleção Schaum, Mcgraw-Hill, Inc., 1991.

□ SOFTWARE DE APOIO:

1. □ Analisador Alloy
2. □ Jape (Just Another Proof Editor)



PROGRAMA DA DISCIPLINA

DISCIPLINA:

PROGRAMAÇÃO ORIENTADA A OBJETOS

CODCRED	CARGA HORÁRIA	MÓDULO
4611F-04	60	30

EMENTA:

Desenvolvimento de sistemas de software de complexidade média utilizando o paradigma de orientação a objetos. Estudo e análise dos conceitos de herança e polimorfismo. Discussão de aspectos avançados sobre classes. Discussão sobre empacotamento e distribuição de aplicações. Uso de bibliotecas de coleções. Projeto de sistemas orientados a objetos.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

- Conhecer e utilizar de forma precisa conceitos e termos relacionados ao paradigma de orientação a objetos.
- Desenvolver as competências e habilidades para a criação de sistemas de complexidade média, formado por múltiplos componentes, e expressar estas soluções na forma de um sistema de classes em uma linguagem de programação.
- Descrever sistemas utilizando código e diagramas de classe UML.
- Construir abstrações para tipos de dados, usando os conceitos de classe, objeto mensagem, herança e interface.
- Compreender e utilizar as técnicas fundamentais de teste e depuração de programas.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Objetos e Classes

- 1.1. O ambiente de programação (*)
 - 1.1.1 Compilação por linha de comando
 - 1.1.2 Argumentos do "main"



- 1.1.3 Recursos do sistema operacional
- 1.2. Revisão de conceitos
 - 1.2.1. Classes e objetos
 - 1.2.2. Atributos e métodos: classe e instância
 - 1.2.3. Tipos de dados: referência e valor
 - 1.2.4 Visibilidade de atributos e métodos
 - 1.2.5 Construtores e destrutores
 - 1.2.6 Sobrecarga de métodos
- 1.3. Modelagem orientada a objetos
 - 1.3.1 Associações e dependências
 - 1.3.2 Diagramas de classe UML
 - 1.3.3 Problemas de modelagem
- 1.4. Gerenciamento de memória (*)
- 1.5. Introdução ao teste unitário
- 1.6. Empacotamento e distribuição de aplicações

Nº DA UNIDADE: 02

CONTEÚDO: Herança e Polimorfismo

CONTEÚDO: Herança e Interfaces

- 2.1. Relacionamentos de generalizaçãoespecialização
 - 2.1.1 Hierarquias de herança
 - 2.1.2 O princípio da substituição
 - 2.1.3 Sobrescrita de métodos
- 2.2 Polimorfismo
 - 2.2.1 Explorando polimorfismo
 - 2.2.2 Classes abstratas
 - 2.2.3 A classe "Object"
 - 2.2.4 O operador "instanceof"
 - 2.2.5 Interfaces
 - 2.2.1. Hierarquia de interfaces
- 2.4. Parametrização de tipos
- 2.5. Problemas de modelagem

Nº DA UNIDADE: 03

CONTEÚDO: Entrada e saída e tratamento de exceções

- 3.1. Tratamento de exceções
 - 3.1.1 Lançamento de exceções
 - 3.1.2 Captura de exceções
 - 3.1.3 Entendendo a hierarquia de exceções padrão



- 3.1.4 Modelando exceções
- 3.2. Leitura e escrita de arquivos texto
 - 3.2.1 Leitura de dados
 - 3.2.2 Gravação de dados
 - 3.2.3 Formatação de dados
 - 3.2.4 Armazenamento e recuperação de objetos

Nº DA UNIDADE: 04

CONTEÚDO: Projeto de Sistemas Orientados a Objetos

- 4.1. Princípios de Projeto Orientado a Objetos
 - 4.1.1 Encapsulamento, coesão, acoplamento
 - 4.1.2 Heurísticas de projeto
- 4.2. Introdução à reutilização
 - 4.2.1 Padrões de projeto
 - 4.2.2 Frameworks
 - 4.2.3 Componentes

Nº DA UNIDADE: 05

CONTEÚDO: A API de Coleções

- 3.1. O framework de coleções
- 5.1. Coleções
 - 5.1.1 A estrutura da API de coleções
 - 5.1.2 Listas, filas e pilhas
 - 5.1.3 Conjuntos
 - 5.1.4 Mapas
 - 5.1.5 Resolução de problemas
- 5.2. Expressões lambda e predicados

Nº DA UNIDADE: 06

CONTEÚDO: Interface de Usuário Gráfica

- 6.1. O framework de interface de usuário
- 6.2. Hierarquia de componentes
- 6.3. Eventos e ações
- 6.4. Janelas e painéis
- 6.5. Menus



PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = P1 + 2.P2 + 2.T + ME / 6$$

Onde:

P1 – conteúdos até a unidade 3.

P2 – conteúdos até a unidade 6.

T – Desenvolvimento de estudo de caso.

ME – média de exercícios realizados ao longo do semestre.

BIBLIOGRAFIA BÁSICA:

1. □ HORSTMANN, C. Java for everyone: late objects. 2nd ed. Danvers: Wiley, 2013. 589 p.
2. □ RAMNATH, S.; DATHAN, B. Object-oriented analysis and design. New York: Springer, 2010. 440 p.
3. □ WEISFELD, M. The object-oriented thought process. 4th ed. Addison Wesley, 2013. 36 p.
4. □ MEYER, B; Object Oriented Software Construction. 2a ed. Prentice-Hall, 1997. 1254 p.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ BOOCH, G.; MAKSIMCHUK, R. A.; ENGLE, M. W.; YOUNG, B. J.; CONALLEN J.; HOUSTON, K. A. Object-oriented analysis and design with applications. 3rd ed. Addison Wesley, 2007. 720 p.
2. □ DEITEL, P.; DEITEL, H. Java: how to program. 9th ed. Prentice Hall, 2011. 1496 p.
3. □ FARRELL, J. An object-oriented approach to programming logic and design. 4th ed. Cengage Learning, 2012. 560 p.
4. □ GAMMA, E.; JOHNSON, R.; VLISSIDES, J.M.; HELM, R.; FOWLER, M. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2002. 364 p.
5. □ GOODRICH, M. T.; TAMASSIA, R. Estruturas de dados e algoritmos em Java. 4ª ed. Porto Alegre: Bookman, 2012. 600 p.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

ALGORITMOS E ESTRUTURA DE DADOS I

CODCRED	CARGA HORÁRIA	MÓDULO
4645G-04	60	60

EMENTA:

Construção e raciocínio sobre diferentes algoritmos e implementações para estruturas de dados lineares e hierárquicas: listas, filas, pilhas e árvores. Exame da adequação destes algoritmos na solução de diversas classes de problemas. Construção de algoritmos e implementações para problemas de ordenação e pesquisa. Discussão, análise e raciocínio sobre a complexidade de algoritmos e implementações correspondentes.

OBJETIVOS:

O aluno ao término da disciplina deverá ser capaz de:

1. Conhecer e utilizar as técnicas fundamentais para avaliar a complexidade de algoritmos.
2. Conhecer e diferenciar as estruturas de dados: listas, filas, pilhas, conjuntos, árvores.
3. Manipular estas estruturas de dados por meio de algoritmos.
4. Selecionar e construir estruturas de dados adequadas para aplicações específicas, bem como modelar estas aplicações.
5. Aplicar algoritmos de ordenação e de pesquisa na solução de problemas.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Desempenho de algoritmos

1. Complexidade de algoritmos
 - 1.1. Contagem de operações
 - 1.2. Notações O, Omega e Theta

Nº DA UNIDADE: 02

CONTEÚDO: Estruturas lineares

- 2.1. Estruturas contíguas X Estruturas encadeadas
- 2.2. Coleções e suas operações de acesso



- 2.2.1. Listas
- 2.2.2. Pilhas
- 2.2.3. Filas
- 2.3. Conjuntos

Nº DA UNIDADE: 03

CONTEÚDO: Classificação e pesquisa

- 3.1. Pesquisa sequencial X pesquisa binária
- 3.2. Classificação de dados
 - 3.2.1. Insertion Sort
 - 3.2.2. Mergesort
 - 3.2.3. Quicksort
 - 3.2.4. Limite inferior para ordenação por comparações $O(n \log(n))$
 - 3.2.5. Algoritmos que não são baseados em comparação

Nº DA UNIDADE: 04

CONTEÚDO: Árvores

- 4.1. Definições e representação
- 4.2. Árvores genéricas
- 4.3. Árvores binárias de pesquisa
- 4.4. Operações: caminhamento, pesquisa, inserção, remoção
- 4.5. Árvores balanceadas e sua eficiência
- 4.6. Tries

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = (P1 + 2*P2 + 2*MT) / 5$$

Onde:

P1: compreende as unidades 1 e 2;

P2: compreende todas as unidades;

MT: média dos trabalhos do semestre, podendo permitir pesos diferentes entre os trabalhos;

BIBLIOGRAFIA BÁSICA:

- 1. CORMEN, T. H. Algoritmos – teoria e prática. 3 ed., Rio de Janeiro: Elsevier-Campus, 2012.
- 2. ELLIS, H.; SAHNI, S.; RAJASEKARAN, S. Computer algorithms. Silicon Press, 2007.
- 3. GOODRICH, M. T.; TAMASSIA, R. Estruturas de dados e algoritmos em Java. 5 ed., Porto Alegre: Bookman, 2013.

BIBLIOGRAFIA COMPLEMENTAR:

- 1. AHO, A. V. Foundations of computer science. New York: Computer Science Press, 1998.



2. □ GERSTING, J. L. Fundamentos matemáticos para ciência da computação: um tratamento moderno de matemática discreta. 5. ed., Rio de Janeiro: LTC, 2004.
3. □ LEVITIN, A. Introduction to the design and analysis of algorithms. Boston: Addison-Wesley, 2003.
4. □ MCALLISTER, W. Data structures and algorithms using Java. 1 ed., Boston: Jones and Bartlett, 2009.
5. □ RAWLINS, G. J. E. Compared to what? An introduction to the analysis of algorithms. New York: Computer Science Press, 1992.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

ENGENHARIA DE SOFTWARE

CODCRED	CARGA HORÁRIA	MÓDULO
4645U-04	60	60

EMENTA:

Introdução aos conceitos fundamentais de Engenharia de Software. Estudo de métodos, processos e ferramentas aplicados no desenvolvimento e manutenção de software. Desenvolvimento de conhecimento teórico-prático na análise de requisitos, projeto, implementação e teste de software.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Desenvolver uma visão geral da Engenharia de Software através das atividades, técnicas, métodos e ferramentas que auxiliam o processo de desenvolvimento de software em equipe;
2. Identificar, compreender e comparar diferentes processos e ciclos de vida de software;
3. Relacionar as principais técnicas de modelagem às atividades do ciclo de vida de um software, incluindo o uso de restrições em Linguagem de Restrições sobre Objetos (OCL);
4. Compreender as diferenças de complexidade intrínsecas de diferentes categorias de software;
5. Identificar técnicas para abordagem de problemas de alta complexidade e para o trabalho em equipe.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Tipos de Sistemas

- 1.1. Categorias de Sistemas de Software
 - 1.1.1. Negócios
 - 1.1.2. Tempo Real
 - 1.1.3. Científico e Matemático
 - 1.1.4. Sistemas Críticos, de Infraestrutura e de Alta Confiabilidade
- 1.2. Ciclo de Vida de Software
 - 1.2.1. Mudanças e incertezas (modelagem de requisitos)
 - 1.2.2. Controle de escopo, orçamento e prazo (gerência de projetos)
 - 1.2.3. Confiabilidade (desenvolvimento de software)



- 1.2.4. Verificação e Validação (qualidade de software)
- 1.3. Evolução de Software
 - 1.3.1. A Crise de Software
 - 1.3.2. Mudança contínua (Lei de Lehman)
 - 1.3.3. Complexidade crescente (Lei de Lehman)
 - 1.3.4. Influência da comunicação na estrutura do produto (Lei de Conway)
 - 1.3.5. Custos crescentes (Lei de Moore)
- 1.4. Estudos de casos

Nº DA UNIDADE: 02

CONTEÚDO: Processos de Desenvolvimento de Software

- 2.1. Estrutura de um Processo
 - 2.1.1. Etapas
 - 2.1.2. Disciplinas
- 2.2. Modelos de Processos
 - 2.2.1. Cascata
 - 2.2.2. Iterativo
 - 2.2.3. Incremental
 - 2.2.4. Espiral
 - 2.2.5. Evolucionário
- 2.3. Estudos de casos

Nº DA UNIDADE: 03

CONTEÚDO: Métodos de Desenvolvimento de Software

- 3.1. Estrutura de um Método
 - 3.1.1. Papéis e Responsabilidades
 - 3.1.2. Cerimônias, Atividades e Tarefas
 - 3.1.3. Artefatos
- 3.2. Métodos Empíricos de Pesquisa
 - 3.2.1. Experimentos Controlados
 - 3.2.2. Estudos de Caso
 - 3.2.3. Revisão de Literatura
- 3.3. Métodos Planificados
 - 3.3.1. Atividades e tarefas do Processo Unificado
 - 3.3.2. Atividades e tarefas do UML Components
- 3.4. Métodos Ágeis
 - 3.4.1. Cerimônias e artefatos do Scrum
 - 3.4.2. Práticas da Extreme Programming
- 3.5. Estudos de casos

Nº DA UNIDADE: 04

CONTEÚDO: Modelagem de Software

- 4.1. Diagramas, Modelos e Meta-Modelos



- 4.1.1. Linguagem de Modelagem Unificada (UML)
 - 4.1.2. Diagramas de Estrutura e de Comportamento na UML
 - 4.2. Restrições, Invariantes, Condições
 - 4.2.1. Linguagem de Restrições sobre Objetos (OCL)
 - 4.2.2. Restrições sobre classes e operações em OCL
 - 4.2.3. Contratos em OCL
 - 4.3. Modelagem Precisa
 - 4.3.1. Restrição de domínios de variáveis
 - 4.3.2. Remoção de ambiguidades
 - 4.3.3. Verificação de estados e de consistência entre variáveis
 - 4.4. Modelos Executáveis
 - 4.4.1. Teste de modelos
 - 4.4.2. Verificação de modelos
 - 4.5. Estudos de casos
-

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = (P1 + P2 + T + E) / 4$$

Onde:

P1 – Prova de avaliação sobre as unidades 1 e 2.

P2 – Prova de avaliação sobre as unidades 3 e 4.

T – Média dos trabalhos desenvolvidos ao longo do semestre. Recomenda-se um trabalho para cada unidade.

E – Média dos exercícios ou desenvolvidos ao longo do semestre. Recomenda-se dois ou três exercícios a serem realizados em aula, para cada unidade.

BIBLIOGRAFIA BÁSICA:

1. □ SOMMERVILLE, Ian. Software Engineering. 9 ed., New York: Addison-Wesley, 2011. ISBN 9780137035151.
2. □ MEYER, Bertrand. Agile! The good, the hype and the ugly. Basel: Springer International, 2014. ISBN 9783319051543.
3. □ MILLS, Bruce. Practical Formal Software Engineering. Cambridge: Cambridge University, 2009. ISBN 9780521879033.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ ARLOW, Jim; NEUSTADT, Ila. UML 2 and the Unified Process. 2 ed., Upper Saddle River: Addison Wesley, 2005. ISBN 9780321321275.
2. □ PRESSMAN, Roger. S. Engenharia de Software: uma abordagem profissional. 7 ed., Porto Alegre: AMGH, 2011. ISBN 9788563308337.
3. □ PFLEEGER, Shari L. Engenharia de Software: teoria e prática. 4 ed., São Paulo: Prentice Hall, 2004. ISBN 8587918311.
4. □ BEATTY, Joy; WIEGERS, Karl. Software Requirements. 3 ed., Redmond: Microsoft: 2013. ISBN



9780735679665.

5. □ LARMAN, Craig. Agile and Iterative Development: a manager's guide. Boston: Addison Wesley, 2010. ISBN 0131111558.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

FUNDAMENTOS DE BANCO DE DADOS

CODCRED	CARGA HORÁRIA	MÓDULO
4645Y-04	60	60

EMENTA:

Visão geral da abordagem banco de dados nos aspectos de modelagem conceitual e projeto de bancos de dados. Uso de dependências funcionais, formas normais e de restrições de integridade para a definição e manipulação de banco de dados e, em especial, SQL. Solução de implementação de bancos de dados: modelo lógico e físico. Mapeamento da modelagem conceitual para o modelo de implementação.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Compreender os conceitos básicos de bancos de dados, bem como as funções, arquitetura típica e os diferentes modelos de dados empregados em sistemas de gerência de bancos de dados (SGBDs);
2. Compreender e praticar o processo de análise e projeto de bancos de dados, em suas abstrações típicas e formalismos, nos diferentes níveis de modelagem: conceitual, lógica e física;
3. Empregar SQL como linguagem de definição de dados, e praticar a implementação de bancos de dados, utilizando SGBDs baseados em modelos pós-relacionais;
4. Compreender e praticar a manipulação de banco de dados com SQL, explorando seus aspectos básicos e avançados, e a otimização de consultas e atualizações;
5. Ter uma visão crítica da gestão de projetos de software empregando bancos de dados.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Introdução aos SGBDs

- 1.1. Conceitos de Bancos de Dados (BDs) e SGBDs e sua evolução histórica
- 1.2. Noções gerais de um sistema de BDs
 - 1.2.1. Abstração de dados
 - 1.2.2. Arquitetura típica de SGBDs
 - 1.2.3. Funções básicas de SGBDs



- 1.2.4. Usuários de BDs
- 1.3. Principais serviços oferecidos por um SGBD
 - 1.3.1. Persistência de dados
 - 1.3.2. Acesso otimizado a dados
 - 1.3.3. Acesso disciplinado a dados
- 1.4. Formas de uso de SGBDs
 - 1.4.1. SGBDs orientados ao processamento de transações
 - 1.4.2. SGBDs orientados ao processamento analítico de dados
 - 1.4.3. SGBDs orientados a consultas de dados não convencionais
- 1.5. Processo de projeto e implementação de BDs
 - 1.5.1. Modelo conceitual
 - 1.5.2. Modelo lógico
 - 1.5.3. Modelo físico
- 1.6. Gestão de projetos de BDs

Nº DA UNIDADE: 02

CONTEÚDO: Modelos conceituais de dados

- 2.1. Conceitos básicos de modelos conceituais
 - 2.1.1. Modelos orientados ao processamento de transações
 - 2.1.1. Modelos orientados ao processamento analítico de dados
 - 2.1.1. Modelos orientados a consultas de dados não convencionais
- 2.2. Mecanismos de abstração
 - 2.2.1. Abstrações básicas: entidades/classes, instâncias/objetos, atributos, relações e papéis
 - 2.2.2. Classificação/instanciação
 - 2.2.3. Generalização/especialização
 - 2.2.4. Agregação/composição
- 2.3. Modelos de Dados
 - 2.3.1. Modelo de dados Orientados a Objetos (OO)
 - 2.3.2. Modelo de dados Entidade Relacionamento (ER)
 - 2.3.3. Semelhanças e diferenças entre modelos conceituais

Nº DA UNIDADE: 03

CONTEÚDO: Projeto de BDs

- 3.1. Modelos lógicos e físicos de BDs
- 3.2. Abordagem relacional para BDs
 - 3.2.1. Conceitos básicos: relações, domínios e atributos
 - 3.2.2. Chaves candidatas
 - 3.2.3. Dependências funcionais
 - 3.2.4. Formas normais
- 3.3. Esquemas de BDs relacionais
 - 3.3.1. Linguagem de Definição de Dados (DDL)
 - 3.3.2. Restrições de integridade declarativas
- 3.4. Transformações de diagramas conceituais para modelos de BDs



Nº DA UNIDADE: 04

CONTEÚDO: Implementação de BDs

- 4.1. Linguagens de BDs
 - 4.1.1. Álgebra relacional
 - 4.1.2. SQL
- 4.2. Consultas simples, com junções e com agregação de dados
- 4.3. Consultas com subconsultas e com operações de conjuntos
- 4.4. Atualizações de dados
 - 4.4.1. Conceito de transação, e segurança e integridade de transações
- 4.5. Visões
 - 4.5.1. Definição e consultas através de visões
 - 4.5.2. Atualização por visões
- 4.6. Otimização de consultas e atualizações
- 4.7. Restrições de integridade procedurais
- 4.8. Conexão com SGBDs

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = P1 + P2 + T/3$$

Onde:

P1 – prova 1, abrange as unidades 1, 2 e parte da 3;

P2 – prova 2, abrange as unidades 3, 4 e 5;

T - corresponde ao conjunto de trabalhos sobre projeto e implementação de bases de dados, e abrange as unidades 2, 3 e 4.

BIBLIOGRAFIA BÁSICA:

1. HEUSER, C. A. Projeto de banco de dados. 6ed., Porto Alegre: Sagra-DC Luzzatto, 2009.
2. ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6ed., São Paulo: Pearson Brasil, 2012.
3. PUGA, S.; FRANÇA, E.; GOYA, M. Banco de Dados: implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson, 2014. 332p. Acesso on-line.

BIBLIOGRAFIA COMPLEMENTAR:

1. BOWMAN, J.; EMERSON, S.; DARNOVSKY, M. The practical SQL handbook: using SQL variants. 4ed., Boston: Addison-Wesley, 2001.
2. MEDEIROS, L.F; Banco de dados: princípios e prática. Intersaberes, 2013. 188p. Acesso on-line.
3. RAMAKRISHNAN, R. Database management systems. 3ed., Boston: McGraw-Hill, 2008.
4. SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de bancos de dados. 6ed., Rio de Janeiro: Campus, 2012.



5. □ ULLMAN, J.; WIDOM, J. A first course in database systems. 3ed., New Jersey: Prentice Hall, 2007.

• □ OUTRAS REFERÊNCIAS:

1. □ Manuais de Fabricantes de SGBDs.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

FUNDAMENTOS DE SISTEMAS DIGITAIS

CODCRED	CARGA HORÁRIA	MÓDULO
4646B-04	60	30

EMENTA:

Estudo de operações aritméticas usando sistemas de numeração binário e hexadecimal. Estudo de álgebra booleana e formas canônicas de expressões algébricas. Estudo análise e aplicação da lógica combinacional e sequencial. Descrição e simulação de circuitos combinacionais e sequenciais com linguagens de descrição de hardware.

OBJETIVOS:

Ao final da disciplina os alunos devem:

- Demonstrar habilidade de entender e realizar operações usando sistemas de numeração binário e hexadecimal.
- Demonstrar habilidade de entender e realizar operações utilizando lógica Booleana.
- Dominar conceitos fundamentais de estrutura, representação, projeto de sistemas digitais combinacionais e sequenciais, e modelagem usando uma linguagem de descrição de hardware (HDL).
- Compreender e avaliar formas de representação funcionais e temporais associadas a descrições HDL de blocos básicos de um processador.
- Projetar blocos básicos de um computador, quais sejam: operadores lógicos e aritméticos, registradores, contadores, memórias e máquinas de estados.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Sistemas de Numeração e Operações aritméticas

- 1.1. Sistemas de numeração
 - 1.1.1. Conversões entre bases decimal e binário;
 - 1.1.2. Conversões entre bases decimal e hexadecimal;
 - 1.1.3. Conversões entre bases hexadecimal e binário;
- 1.2. Operações aritméticas;
 - 1.2.1. Soma e subtração em binário;



1.2.2. Representação de sinal em complemento de 2;

Nº DA UNIDADE: 02

CONTEÚDO: Circuitos Digitais Combinacionais e sua Representação em HDL

2.1. Álgebra booleana e otimização lógica

2.1.1. Operadores lógicos;

2.1.2. Tabela verdade;

2.1.3. Expressões e equações Booleanas;

2.1.4. Soma de produtos e produto de somas;

2.1.5. Teoremas para álgebra Booleana;

2.1.6. Relações de equivalência;

2.1.7. Mapas de Karnaugh e Quine–McCluskey;

2.2. Introdução a sistemas digitais combinacionais;

2.2.1. O fluxo de projeto de sistemas digitais;

2.2.2. Representação de circuitos digitais em diferentes níveis de abstração;

2.2.3. Portas lógicas elementares;

2.3. Introdução a uma linguagem de descrição de hardware (HDL);

2.3.1. Componentes combinacionais primitivos: not, and, or, xor, etc;

2.3.2. Aritmética binária: soma e subtração, com e sem sinal;

2.3.3. Modelagem de componentes combinacionais compostos: operadores aritméticos, multiplexadores, somadores, subtratores, comparadores, codificadores, decodificadores, ULAs;

2.3.4. Simulação de circuitos e o projeto de testbenches;

2.3.5. Simulação de circuitos combinacionais com atrasos.

Nº DA UNIDADE: 03

CONTEÚDO: Circuitos Digitais Sequenciais e sua Representação em HDL

3.1. Introdução a circuitos digitais sequenciais;

3.1.1. Circuitos digitais: combinacionais versus sequenciais;

3.1.2. A realimentação de saídas e a memorização dinâmica de informações;

3.2. Modelagem de circuitos sequenciais síncronos;

3.2.1. Representações de circuitos sequenciais síncronos;

3.2.2. Modelagem sequencial em HDL: processos;

3.2.3. Elementos de memória unitários: latches, flip-flops, registradores;

3.2.4. Outros elementos de memória: contadores, temporizadores, e registradores deslocamento;

3.2.5. Elementos de memória bidimensionais: bancos de registradores e memórias (RAM, ROM, etc.)

3.2.6. Máquinas de estado finitas: Moore e Mealy;

3.3. Modelagem de sistemas digitais completos: trabalho integrador.

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:



$$G1 = (P1 + P2 + T) / 3$$

Onde:

P1 – Prova compreendendo os conteúdos da Unidade 1

P2 – Prova compreendendo os conteúdos da Unidade 2.

T – Média das tarefas e atividades aplicadas durante o semestre.

BIBLIOGRAFIA BÁSICA:

1. □ FLOYD, T. L. Sistemas digitais: fundamentos e aplicações. 9a. Ed. Editora Bookman. 2007. 888 p.
2. □ D'AMORE, R. VHDL: Descrição e Síntese de Circuitos Digitais. 2ª. Ed. Editora LTC. 2012. 259 p.
3. □ GREGG, J. R. Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets. Editora Willey. 1998. 296 p.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ MEALY, B.; TAPPERO, F. Free Range VHDL. 2013. 190 p.
http://www.freerangefactory.org/dl/free_range_vhdl.pdf
2. □ HARRIS, D.; HARRIS, S. Digital Design and Computer Architecture. 2a edição. Editora Morgan Kaufmann. 2012.
3. □ PEDRONI, V. A. Eletrônica Digital Moderna e VHDL. Editora Elsevier. 2010. 619 p.
4. □ BROWN, S. D. Fundamentals of digital logic with VHDL design. 3a edição. Editora McGraw-Hill. 2008. 960 p.
5. □ VAHID, F. Sistemas digitais : projeto, otimização e HDLs. Editora Bookman. 2010.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

FUNDAMENTOS DO RACIOCÍNIO LÓGICO

CODCRED	CARGA HORÁRIA	MÓDULO
4646C-04	60	60

EMENTA:

Lógica, estrutura e validade dos argumentos. Conceitos básicos de sintaxe e semântica da lógica formal proposicional e de predicados. Lógica e silogismos categóricos. Sistemas dedutivos e construção de provas em dedução natural. Formalização e Verificação da Validade de Argumentos.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Compreender e formalizar a estrutura dos argumentos verbais
2. Compreender a noção de validade de argumentos e sua relação com os sistemas dedutivos da Lógica Proposicional e de Predicados
3. Computar o valor lógico de fórmulas da Lógica Proposicional e de Predicados
4. Construir provas formais na Lógica Proposicional e de Predicados
5. Verificar Argumentos formalizados na Linguagem Proposicional e de Predicados

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Lógica e a Estrutura dos Argumentos

- 1.1. O que é Lógica?
- 1.2. O que é um Argumento
- 1.3. Diagramação de Argumentos
- 1.4. Validade e Consistência
- 1.5. Padrões de Inferência
- 1.6. Técnica de Contra-Exemplos
- 1.7. Validade de Argumentos
- 1.8. Uso menção

Nº DA UNIDADE: 02

CONTEÚDO: Lógica Proposicional



- 2.1. Declarações, Tautologias e Argumentos
 - 2.1.1. Conectivos e Valores Verdade
 - 2.1.2. Tautologias
 - 2.1.3. Formalização de Argumentos
- 2.2. Argumentos e Regras de Derivação
 - 2.2.1. Sequência de Derivação
 - 2.2.2. Regras de Derivação
 - 2.2.3. Método da Dedução e Outras Regras
 - 2.2.4. Verificação de Argumentos

Nº DA UNIDADE: 03

CONTEÚDO: Lógica dos Enunciados Categóricos

- 3.1. Enunciados Categóricos
- 3.2. Diagramas de Venn
- 3.3. Inferências Imediatas
- 3.4. Silogismos Categóricos

Nº DA UNIDADE: 04

CONTEÚDO: Lógica de Predicados

- 4.1. Quantificadores, Predicados e Validade
 - 4.1.1. Predicados
 - 4.2.2. Quantificadores
 - 4.2.3. Tradução de Argumentos
 - 4.2.4. Validade de Argumentos
- 4.2. Regras de Derivação
 - 4.2.1. Instanciação Universal
 - 4.2.2. Generalização Existencial
 - 4.2.3. Generalização Universal
 - 4.2.4. Instanciação Existencial
- 4.3. Verificação de Argumentos

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = (P1 + 2 \cdot P2 + T) / \square 4$$

Onde: \square

P1 - Unidades 1 e 2

P2 - Unidades 3 e 4

T - Média aritmética ou ponderada das diversos tarefas e trabalhos publicados durante o semestre.

BIBLIOGRAFIA BÁSICA:

- 1. GERSTING, J. Fundamentos Matemáticos Para a Ciência da Computação. LTC, 7a. edição, 2016.



2. NOLT, J., ROHATYN, Dennis., VARZIN, A. Schaum´s Outline of Logic, Second Edition (Schaum´s Outlines). McGraw-Hill, 2nd edition, 2011.
3. HAIGHT, M. A Serpente e a Raposa: Uma Introdução à Lógica. Edições Loyola, 2003.

BIBLIOGRAFIA COMPLEMENTAR:

1. MORTARI, C. A. Introdução à Lógica. Ed. UNESP, 2001
2. SOUZA, J.N. Lógica para Ciência da Computação: Uma Introdução Concisa. 2ª ed. Campus, 2008.
3. SMITH, P. An Introduction to Formal Logic. CUP, 2003.
4. HEIN, J. Discrete structures, logic, and computability. Jones and Bartlett Publishers, 2009
5. BASSHAM, G. et al. Critical Thinking: A Student´s Introduction. McGraw-Hill, 2011.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

INTRODUÇÃO A CIÊNCIA DA COMPUTAÇÃO

CODCRED	CARGA HORÁRIA	MÓDULO
4646F-04	60	60

EMENTA:

Domínio das técnicas de pensamento computacional para a solução de problemas computacionais: abstração, reconhecimento e generalização de padrões e projeto de algoritmos. Conceitos básicos de programação, estratégias e técnicas para a solução de problemas computacionais, teste e depuração de programas, complexidade de algoritmos, visualização de dados.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Dominar as técnicas de pensamento computacional para a solução de problemas: abstração, reconhecimento e generalização de padrões, e projeto de algoritmos
2. Expressar computações através de uma linguagem de programação
3. Utilizar uma abordagem sistemática para organizar, descrever, testar e depurar algoritmos
4. Compreender de maneira introdutória o conceito de complexidade computacional
5. Aprender a utilizar ferramentas computacionais para visualizar e compreender dados

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Introdução à solução algorítmica de problemas

- 1.1. Computação e Cálculo
- 1.2. Conhecimento Declarativo e Imperativo
- 1.3. Algoritmos
 - 1.3.1. Descrições de Computações
 - 1.3.2. Processos Mecânicos
 - 1.3.3. Máquinas Computacionais
 - 1.3.4. Máquinas Universais e a Tese de Church-Turing
- 1.4. Conceitos Básicos sobre Linguagens de Programação
- 1.5. Uma estratégia geral para solução de problemas



Nº DA UNIDADE: 02

CONTEÚDO: Fundamentos de programação

- 2.1. Elementos básicos de um programa
 - 2.1.1. Objetos, expressões e tipos numéricos
 - 2.1.2. Variáveis e atribuição
 - 2.1.3. Entrada e saída
- 2.2. Representação de textos: strings
- 2.3. Estruturas de controle de fluxo
 - 2.3.1. Comandos condicionais
 - 2.3.2. Iteração

□

Nº DA UNIDADE: 03

CONTEÚDO: Problemas de aproximação e convergência

- 3.1. Enumeração exaustiva
 - 3.1.1. O método de enumeração exaustiva
 - 3.1.2. Iteração avançada com o comando "for" de Python
- 3.2. Técnicas de aproximação
 - 3.2.1. Ordens totais
 - 3.2.2. O método da bissecção
 - 3.2.3. Representação de valores em ponto flutuante
 - 3.2.4. Aproximações sucessivas: o método de Newton-Raphson

Nº DA UNIDADE: 04

CONTEÚDO: Abstração de computações

- 4.1. Subprogramas
 - 4.1.1. Representação de funções em Python
 - 4.1.1.1. Estrutura e sintaxe
 - 4.1.1.2. Parâmetros
 - 4.1.1.3. Escopo
 - 4.2. Abordagens "top-down" e "botton-up"
 - 4.3. Recursão
 - 4.3.1. Fundamentos da Recursão
 - 4.3.2. Funções Recursivas
 - 4.3.3. Exemplos de solução de Problemas usando Recursão
- 4.5. Outro nível de abstração: módulos
- 4.6. Arquivos

Nº DA UNIDADE: 05

CONTEÚDO: Especificação, teste e depuração de programas

- 5.1. Especificação de funções
 - 5.1.1. Pré e pós-condições



- 5.1.2. Noção de verificação de programas
- 5.2. Teste
 - 5.2.1. Tipos e níveis de testes
 - 5.2.2. Projeto e execução de casos de testes
- 5.3. Depuração
 - 5.3.1. Projeto dos experimentos
 - 5.3.2. Condução dos experimentos

Nº DA UNIDADE: 06

CONTEÚDO: Abstração de dados e visualização

- 6.1. Abstração de dados
 - 6.1.1. Tuplas
 - 6.1.2. Intervalos (ranges)
 - 6.1.3. Listas
- 6.2. Visualização de dados
 - 6.2.1. Desenhando gráficos usando "PyLab"
 - 6.2.2. Estudo de caso: análise dos custos de uma hipoteca

Nº DA UNIDADE: 07

CONTEÚDO: Introdução a complexidade de algoritmos

- 7.1. Entendendo a noção de complexidade
- 7.2. Notação assintótica
- 7.3. Classes de complexidade relevantes
 - 7.3.1. Complexidade constante
 - 7.3.2. Complexidade logarítmica
 - 7.3.3. Complexidade linear
 - 7.3.4. Complexidade log-linear
 - 7.3.5. Complexidade polinomial
 - 7.3.6. Complexidade exponencial
 - 7.3.7. Comparando as classes de complexidade

Nº DA UNIDADE: 08

CONTEÚDO: Algoritmos de busca e ordenação

- 8.1. Algoritmos de busca
 - 8.1.1. Busca linear
 - 8.1.2. Busca binária
- 8.2. Algoritmos de ordenação
 - 8.2.1. "Selection-sort"
 - 8.2.2. "Merge Sort"

Nº DA UNIDADE: 9

CONTEÚDO: Modelagem de processos estocásticos



- 9.1. Introdução: importância dos modelos estocásticos
- 9.2. Estudos de caso: caminhamentos aleatórios
 - 9.2.1. "O Passo do Bêbado"
 - 9.2.2. Caminhamentos tendenciosos
 - 9.2.3. "Campos Traiçoeiros"

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = (P1 + P2 + T)/3$$

Onde:

P1 – Prova 1;

P2 – Prova 2;

T – Média aritmética ou ponderada das tarefas e trabalhos publicados durante o semestre.

BIBLIOGRAFIA BÁSICA:

1. □GUTTAG, John V. Introduction to Computation and Programming Using Python with Application to Understanding Data. 2 ed. The MIT Press, 2016.
2. □FOROUZAN, B.; MOSHARRAF, F. Fundamentos da ciência da computação. Cengage Learning, 2011.
3. □ANTON SPRUAL, V. Think Like a Programmer: An Introduction to Creative Problem Solving. 1 ed., No Starch Press, 2012.

BIBLIOGRAFIA COMPLEMENTAR:

1. □BIERMANN, W. Alan. Great Ideas in Computer Science: a gentle introduction. 2 ed. The MIT Press, 2016. (livro eletrônico)
2. □SCHNEIDER, G. Michael; GERSTING, Judith L. An Invitation to Computer Science. 6 ed., Cengage Learning, 2009
3. □GLENN, B. J. Ciência da computação: Uma visão abrangente. 11 ed., Bookman, 2013.
4. □EVANS, D. Introduction to Computing: Exploration in Language, Logic and Machines. Creative Commons, 2011.
5. □SEDGEWICK, R.; WAYNE, K.; DONDERO, R. Introduction to Programming in Python: An Interdisciplinary Approach. 1 ed. Addison-Wesley Professional, 2015.
6. □DIERBACH, C. Introduction to computer science using python: A computational problem-solving focus. 1 ed., Wiley, 2012.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

LINGUAGENS FORMAIS E AUTOMATOS

CODCRED	CARGA HORÁRIA	MÓDULO
4646J-04	60	60

EMENTA:

Introdução aos conceitos fundamentais de linguagens e gramáticas formais. Estudo de máquinas como modelos de computação: autômatos finitos determinísticos, autômatos finitos não-determinísticos, autômatos de pilha e máquinas de Turing. Estudo de linguagens como modelos de especificação de computações: linguagens regulares, linguagens livres de contexto, linguagens sensíveis ao contexto. Discussão e análise das equivalências entre os modelos baseados em máquinas e linguagens e da hierarquia sobre estes modelos.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Descrever linguagens de modo formal;
2. Entender uma linguagem a partir de sua representação formal;
3. Descobrir e provar propriedades de linguagens;
4. Diferenciar classes de linguagens, suas propriedades e relações hierárquicas;
5. Compreender e implementar algoritmos de manipulação de representações formais e a geração de reconhecedores a partir de representações formais de linguagens.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Introdução

- 1.1. Motivação ao estudo de linguagens formais
- 1.2. Definições básicas de linguagens: símbolos, alfabeto, cadeia: prefixos, sufixos, tamanho, concatenação, linguagens, concatenação e fechamento de linguagens
- 1.3. Formalismos para definição de linguagens: gramáticas (formalismos geradores) e autômatos finitos (reconhecedores)
- 1.4. Linguagens como modelos de especificação de computações

Nº DA UNIDADE: 02

CONTEÚDO: Linguagens Regulares



- 2.1. Autômatos finitos (máquinas de estado finitas)
 - 2.1.1. Autômato finito determinístico
 - 2.1.2. Autômato finito não determinístico
- 2.2. Expressões regulares
 - 2.2.1. Álgebra de expressões regulares
- 2.3. Equivalência e conversão entre os modelos de autômato finito e expressões regulares
- 2.4. Minimização de autômatos finitos determinísticos
- 2.5. Gramáticas regulares
- 2.6. Propriedades das linguagens regulares
 - 2.6.1. Propriedades de fechamento
 - 2.6.2. Técnicas para decidir se uma linguagem é regular
 - 2.6.3. Decidibilidade e algoritmos de decisão
- 2.7. Aplicações dos formalismos para linguagens regulares

Nº DA UNIDADE: 03

CONTEÚDO: Linguagens Livres de Contexto

- 3.1. Gramáticas livres de contexto
 - 3.1.1. Definições básicas: gramática, derivação, derivações mais à esquerda e mais à direita, linguagem gerada por uma gramática
 - 3.1.2. Árvores de derivação, gramáticas ambíguas
 - 3.1.3. Algoritmos de transformação de gramáticas e formas normais
- 3.2. Autômato de Pilha
 - 3.2.1. Autômato de pilha determinístico e não determinístico
- 3.3. Relações de equivalência entre gramáticas e autômatos
- 3.4. Propriedades das linguagens livres de contexto
 - 3.4.1. Técnicas para decidir se uma linguagem é livre de contexto
 - 3.4.2. Propriedades de fechamento
 - 3.4.3. Decidibilidade e algoritmos de decisão
- 3.5. Aplicações dos formalismos para linguagens livres de contexto.

Nº DA UNIDADE: 04

CONTEÚDO: Linguagens Sensíveis ao Contexto e Irrestritas

- 4.1. Gramáticas irrestritas e conjuntos recursivamente enumeráveis
- 4.2. Gramáticas sensíveis ao contexto
- 4.3. Máquinas de Turing como reconhecedores de linguagens recursivamente enumeráveis
- 4.4. Hierarquia de Chomsky

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:



$$G1 = P1 + P2 + P3 + NT / 4$$

Onde: □

P1 – Prova 1, abrange as unidades 1 e 2

P2 – Prova 2, abrange a unidade 3

P3 – Prova 3, abrange a unidade 4

NT – Nota de trabalho(s) práticos

BIBLIOGRAFIA BÁSICA:

1. □ HOPCROFT, John E.; MOTWANI, Rajeev; ULLMAN, Jeffrey D. Introdução à teoria de autômatos, linguagens e computação. Rio de Janeiro: Elsevier, 2003. 560 p.
2. □ RAMOS, Marcus V.M.; NETO, João J.; VEGA, Ítalo S. Linguagens Formais: Teoria, Modelagem e Implementação. Porto Alegre: Bookman, 2009. 656 p.
3. □ PARKES, A. Introduction to Languages, Machine and Logic. Computable Languages, Abstract Machines and Formal Logic. Springer, 1 ed., 2002.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ MENEZES, Paulo B. Linguagens Formais e Autômatos. 4 ed. Porto Alegre: Sagra-Luzzatto, 2001. 165p.
2. □ HOPCROFT, John E.; ULLMAN, Jeffrey D. Introduction to automata theory, languages, and computation. Reading: Addison-Wesley, 1979.
3. □ RÉVÉSZ, György E. Introduction to Formal Languages. New York: Dover, 1991. 199 p.
4. □ HEIN, J. Discrete Structures, Logic and Computability. Jones & Bartlett Publishers (1 de janeiro de 2016).
5. □ VIEIRA, José N. Introdução aos Fundamentos da Computação: Linguagens e Máquinas. São Paulo: Pioneira Thomson Learning, 2006. 319 p.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

MATEMATICA CONCRETA

CODCRED	CARGA HORÁRIA	MÓDULO
4646K-04	60	60

EMENTA:

Estudo de técnicas matemáticas fundamentais para a análise e solução de problemas algorítmicos. Exame de técnicas elementares de contagem. Introdução às leis fundamentais de manipulações de somas finitas, funções inteiras, funções binomiais, recorrências. Fundamentação e exame dos conceitos de notação assintótica e taxas de crescimento.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Compreender e aplicar o raciocínio recursivo e iterativo.
2. Provar por indução matemática a validade de relações de recorrência.
3. Saber expressar matematicamente o resultado de problemas de contagem.
4. Compreender e aplicar o cálculo de comportamentos assintóticos (notação O).

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Manipulação de somas

- 1.1. Representação e manipulação algébrica
- 1.2. Prova por indução
- 1.3. Uso em modelagem e solução de problemas
- 1.4. Somas Infinitas

Nº DA UNIDADE: 02

CONTEÚDO: Coeficientes Binomiais e Estruturas Combinatórias

- 2.1. Permutações, Arranjos e Combinações
 - 2.1.1. Propriedades
 - 2.1.2. Geração recursiva
- 2.2. Coeficientes Binomiais



- 2.2.1. Identidades e manipulação
- 2.2.2. Uso em modelagem e solução de problemas

Nº DA UNIDADE: 03

CONTEÚDO: Equações recorrentes

- 3.1. Representação e manipulação algébrica
- 3.2. Uso em modelagem e solução de problemas
- 3.3. Implementação recursiva e não recursiva
- 3.4. Solução algébrica

Nº DA UNIDADE: 04

CONTEÚDO: Funções Inteiras e Teoria dos Números

- 4.1. Funções Inteiras
 - 4.1.1. Pisos e Tetos
- 4.2. Teoria dos Números
 - 4.2.1. Divisibilidade e Primalidade
 - 4.2.2. Congruência

Nº DA UNIDADE: 05

CONTEÚDO: Comportamento Assintótico

- 5.1. Comportamentos Assintóticos
 - 5.1.1. A Notação O
 - 5.1.2. Propriedades da notação O
 - 5.1.3 Relação com programas e algoritmos

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

(P1 + 2P2 + 2T) / 5

Onde: □

P1 = conteúdos das unidades 1 e 2

P2 = conteúdos das unidades 3 a 5

T = média dos trabalhos práticos

BIBLIOGRAFIA BÁSICA:

1. □ GRAHAM, Ronald; KNUTH, Donald; PATASHNIK, Oren. Matemática concreta: Fundamentos para ciência da computação. 2 ed., LTC, 1995.
2. □ BOGART, Kenneth P.; DRYSDALE, Scot; STEIN, Cliff. Matemática discreta para ciência da computação. Pearson, 2013.
3. □ GERSTING, Judith. Fundamentos matemáticos para ciência da computação. 5 ed., LTC, 2004.



BIBLIOGRAFIA COMPLEMENTAR:

1. □LOVÁSZ, L.; PELIKÁN, J.; VESZTERGOMBI, K. Discrete mathematics: Elementary and beyond (undergraduate texts in mathematics). Springer, 2003.
2. □LEHMAN, E., LEIGHTON, F. T., MEYER, A.: Mathematics for Computer Science. Disponível online em <https://courses.csail.mit.edu/6.042> .
3. □JENKYNS, Tom; STEPHENSON, Ben. Fundamentals of discrete math for computer science: A problem-solving primer. Springer, 2013.
4. □PATRICK, David. Introduction to counting and probability. The Art of Problem Solving Inc., 2005.
5. □HEIN, J. Discrete structures, logic, and computability. Jones and Bartlett Publishers, 2009.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

METODOLOGIA CIENTIFICA E ESCRITA PARA COMPUTACAO

CODCRED	CARGA HORÁRIA	MÓDULO
4646L-04	60	60

EMENTA:

Fundamentos do conhecimento científico e seus métodos. Pesquisa bibliográfica e resumos. Estudos dos conceitos de fatos, leis, teorias e hipóteses. Estilo de escrita: diretrizes gerais e específicas. Apresentação de algoritmos e textos matemáticos. Declaração, desenvolvimento e defesa de hipóteses. Construção de um projeto científico. Ética na escrita. Diretrizes para a construção de apresentações.

OBJETIVOS:

No contexto da Ciência da Computação, o cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Utilizar métodos, conceitos e definições na elaboração de projetos científicos.
2. Criar conexões entre metodologia científica, educação através das artes liberais e o desenvolvimento da ciência e da tecnologia.
3. Compreender o método científico enquanto processo de sistematização para o desenvolvimento da ciência.
4. Utilizar a metodologia científica no desenvolvimento de projetos científicos.
5. Capacitar para a escrita científica pela análise de textos científicos da literatura e daqueles preparados pelos próprios alunos.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Fundamentos do conhecimento científico

- 1.1. Conceito de método
- 1.2. Método indutivo
- 1.3. Método dedutivo
- 1.4. Método hipotético-dedutivo
- 1.5. Método dialético
- 1.6. Metodologia científica e o pensamento de Descartes



Nº DA UNIDADE: 02

CONTEÚDO: Escrita científica

- 2.1. Estilo de escrita
- 2.2. Tipos de publicação científica: artigo, resumo, relatório

Nº DA UNIDADE: 03

CONTEÚDO: Preparação de texto escrito

- 3.1. Construção de gráficos, figuras, tabelas
- 3.2. Apresentações matemáticas e de algoritmos
- 3.3. Processo de referência bibliográfica
- 3.4. Organização da informação: artigo, resumo, relatório
- 3.5. Edição da informação: escolha do processador de texto, consistência, estilo
- 3.6. Agradecimentos
- 3.7. Ética

Nº DA UNIDADE: 04

CONTEÚDO: Apresentação de resultados

- 4.1. Conteúdo, organização
- 4.2. Introdução, conclusão
- 4.3. Sobreposição de textos e slides
- 4.4. Figuras, gráficos e tabelas
- 4.5. Apresentação oral, audiência e tempo para perguntas

Nº DA UNIDADE: 05

CONTEÚDO: Projeto científico

- 5.1. Desenvolvimento de hipóteses
- 5.2. Condução e apresentação de experimentos
- 5.3. Pesquisa bibliográfica

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

O grau G1 é o resultado da média aritmética das notas P1, P2, T, ou seja,

$$G1 = (P1+P2+T)/3.$$

P1 e P2 são provas regulares e T a nota de trabalho. A prova P1 é parcial e trata dos conteúdos das unidades 1, 2 e 3. A prova P2 é parcial e trata dos conteúdos das unidades de 3, 4 e 5.



Além das provas regulares existe uma prova de substituição, prova PS, que é realizada somente quando, pelo menos, uma das provas regulares não é realizada. A prova PS é cumulativa.

BIBLIOGRAFIA BÁSICA:

1. □ ZOBEL, Justin. Writing for computer science. 2 ed., Springer, 2004.
2. □ WILSON JR, E. Bright. An introduction to science research. Dover Publications, 1991.
3. □ MARCONI, Marina de Andrade; LAKATOS, Eva Maria. Fundamentos da metodologia científica. 7 ed., Atlas, 2010.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ WAZLAWICK, Raul S. Metodologia de pesquisa para ciência da computação. Campus, 2009.
2. □ DAWKINGS, Richard. The Oxford book of modern science writing. Oxford University Press, 2009.
3. □ GLASMAN-DEAL, Hilary. Science research writing: A guide for non-native speakers of english. 1 ed., Imperial College Press, 2009.
4. □ COHEN, Morris F. An introduction to logic and scientific method. Hughes Press, 2008.
5. □ KRAMER, Stephen P. How to think like a scientist: Answering questions by the scientific method. HarperCollins, 1987.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

CODCRED	CARGA HORÁRIA	MÓDULO
4646Q-04	60	60

EMENTA:

Apresentação da organização básica de um processador. Estudo das principais unidades, fases de operação, fluxo de dados e instruções e principais variações arquiteturais dos processadores. Discussão do formato das instruções e a interação entre o processador e a memória. Utilização de linguagens de descrição de hardware e simuladores. Estudo e construção de programas em linguagem de montagem. Análise e discussão de casos reais.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Identificar as estruturas fundamentais de conjuntos de instruções e linguagens de montagem de arquiteturas RISC, sendo capaz de empregar estes conceitos em programação destas arquiteturas.
2. Reconhecer as relações fundamentais existentes entre o hardware e o software em arquiteturas de computadores modernas em suas abstrações mais relevantes, organização do hardware, linguagem de montagem e linguagem de alto nível.
3. Identificar a relação entre linguagens de programação de alto nível e as estruturas de hardware em arquiteturas de computadores.
4. Utilizar e ter noções de como implementar programas básicos empregados na tradução e execução de programas escritos em linguagem de montagem, tais como montadores, ligadores e carregadores.
5. Identificar as principais formas de elaborar organizações de computadores que seguem o paradigma RISC: implementação monociclo, multiciclo e pipeline.

CONTEÚDO:

- 3.3.1. Modificações necessárias no bloco de dados e bloco de controle
- 3.3.2. Hazards estruturais, de dados e de controle

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:



$$G1 = 0.4 * P1 + 0.4 * P2 + 0.2 * T$$

Onde:

P1 – Prova compreendendo os conteúdos das unidades 1 e 2

P2 – Prova compreendendo os conteúdos da unidade 3

T – Média das atividades e tarefas aplicadas ao longo do semestre □

BIBLIOGRAFIA BÁSICA:

1. □ PATTERSON, D. A.; HENNESSY, J. L. Computer Organization and Design: the hardware and software interface. 5 ed., Oxford: Morgan Kaufmann, 2014.
2. □ STALLINGS, W. Arquitetura e organizaçã#771;o de computadores. 8 ed., Pearson, 2010.
3. □ MONTEIRO, M. A. Introduçã#807;a#771;o a#768; organizacã#807;a#771;o de computadores. 5 ed., LTC, 2007.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ PATTERSON, D. A.; HENNESSY, J. L. Computer Organization and Design: the hardware and software interface. 5 ed., Oxford: Morgan Kaufmann, 2014.
2. □ STALLINGS, W. Arquitetura e organizaçã#771;o de computadores. 8 ed., Pearson, 2010.
3. □ MONTEIRO, M. A. Introduçã#807;a#771;o a#768; organizacã#807;a#771;o de computadores. 5 ed., LTC, 2007.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

ORGANIZACAO E ARQUITETURA DE COMPUTADORES II

CODCRED	CARGA HORÁRIA	MÓDULO
4646R-04	60	60

EMENTA:

Estudo de métodos de exploração de paralelismos em computadores mono e multiprocessados. Discussão sobre os principais métodos de programação paralela e sua relação com a arquitetura/organização do hardware e a organização da memória. Estudo sobre a evolução das topologias de comunicação intrachip. Compreensão do papel da memória cache e métodos de acesso a periféricos no desempenho do sistema. Estudo de métodos de programação visando melhorar a eficiência das memórias cache. Utilização de benchmarks e análise comparativa de desempenho.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Conhecer diferentes tipos de sistemas de entrada e saída, bem como os mecanismos que existem para acessar dispositivos externos ao computador.
2. Compreender como computação, comunicação e armazenamento afetam o desempenho de um sistema computacional, bem como entender métricas usadas para avaliar o desempenho deste sistema computacional.
3. Conhecer a hierarquia de memória, os fundamentos, mecanismos e políticas para transferência de informação entre níveis e os mecanismos de coerência de cache para sistemas multiprocessados.
4. Compreender conceitos básicos de arquiteturas de computação paralela intrachip.
5. Conhecer topologias de comunicação intrachip e saber como empregá-las para atender aos requisitos de computação e comunicação.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Sistemas de Entrada e Saída (E/S)

- 1.1. Interfaces de comunicação
 - 1.1.1. Síncrona
 - 1.1.1. Assíncrona



- 1.1.1. Semi-síncrona
- 1.2. Mapeamento em portas e mapeamento em memória
- 1.3. Interface entre programa e dispositivo
 - 1.3.1. E/S programada
 - 1.3.1.1. Bloqueado
 - 1.3.1.2. Polling
 - 1.3.1.3. Interjeição
 - 1.3.2. E/S não programada
 - 1.3.2.1. Interrupção interna ao processador
 - 1.3.2.2. Interrupção externa ao processador
 - 1.3.3. Acesso direto à memória (DMA)

Nº DA UNIDADE: 02

CONTEÚDO: Arquiteturas de Computação Paralela Intrachip

- 2.1. Introdução aos sistemas paralelos intrachip
 - 2.1.1. Motivações para exploração de paralelismo
 - 2.1.2. Níveis de Paralelismo / Grau de Paralelismo
 - 2.1.3. Complicadores, limitações
 - 2.1.4. Medidas básicas de desempenho (speed-up, eficiência)
- 2.2. Classificação de sistemas paralelos
 - 2.2.1. Classificação de Flynn (SISD, MISD, SIMD, MIMD)
 - 2.2.2. Classificação segundo modelo de memória (UMA, NUMA, NORMA, ...)
 - 2.2.2.1. Memória compartilhada versus troca de mensagem
- 2.3. Paralelismo em um único núcleo de processamento
 - 2.3.1. Pipelines avançados
 - 2.3.1.1. Pipeline superescalar
 - 2.3.1.2. Superpipeline
 - 2.3.1.3. Pipeline super-super
 - 2.3.2. Processadores vetoriais
 - 2.3.3. Arquitetura VLIW
 - 2.3.4. Multithreading
- 2.4. Paralelismo com múltiplos núcleos de processamento

Nº DA UNIDADE: 03

CONTEÚDO: Sistemas de Armazenamento

- 3.1. Hierarquia de memória
 - 3.1.1. Fundamentos, conceitos e características
 - 3.1.2. Arquiteturas e organização
- 3.1. Memória cache
 - 3.1.1. Conceitos, características e organização
 - 3.1.2. Mapeamento de endereços
 - 3.1.3. Políticas de atualização
- 3.2. Memória principal e virtual
 - 3.2.1. Conceitos, características e organização
 - 3.2.2. Sistemas paginados, segmentados e segmento-paginados
- 3.3. TLB



3.4. Coerência de cache em sistemas multiprocessados

Nº DA UNIDADE: 04

CONTEÚDO: Arquiteturas de Comunicação Intrachip

4.1. Introdução às arquiteturas de comunicação intrachip

4.1.1. Conceitos e fundamentos

4.1.2. Classificação

4.1.3. Medidas básicas de desempenho (latência, vazão)

4.2. Estratégias de Comunicação

4.2.1. Estratégias de Roteamento

4.2.2. Estratégias de Conexão

4.2.3. Estratégias de Compartilhamento

4.3. Paradigmas de comunicação

4.3.1. Memória compartilhada

4.3.2. Troca de mensagens

4.4. Topologias de comunicação intrachip (barramentos, conexão dedicada, ...)

PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

AVALIAÇÃO:

$$G1 = (0,35 \times P1) + (0,35 \times P2) + (0,3 \times T)$$

Onde:

P1 – Prova compreendendo os conteúdos das unidades 1 e 2

P2 – Prova compreendendo os conteúdos das unidades 3 e 4

T – Média aritmética dos trabalhos aplicados ao longo do semestre

BIBLIOGRAFIA BÁSICA:

1. □ STALLINGS, W. Arquitetura e organização de computadores. 8a ed., Pearson, 2010, 625p.
2. □ BAER, J.-L. Arquitetura de Microprocessadores – do Simple Pipeline ao Multiprocessador em Chip. 1a ed., LTC, 2013, 326p.
3. □ HENNESSY, J; PATTERSON, D. Arquitetura de computadores: uma abordagem quantitativa. 5a ed., Campus, 2014, 744p.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ TANENBAUM, A.; AUSTIN, T. Organização estruturada de computadores. 6a ed., Capítulo 8 - Arquitetura de Computadores Paralelos, pp. 436-518, Pearson, 2013.
2. □ PATTERSON, D.; HENNESSY, J. Organização e projeto de computadores: a interface hardware/software. 4a ed., Elsevier, 2014, 709p.
3. □ MONTEIRO, M. Introdução à organização de computadores. 5a ed., LTC, 2012, 686p.
4. □ PEDRONI, V. Eletrônica digital moderna e VHDL. 1a ed., Campus, 2010, 648p.
5. □ PARHAMI, B. Computer Architecture: From Microprocessors to Supercomputers. Oxford University Press, 2014, 576p.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

TECNICAS DE PROGRAMACAO

CODCRED	CARGA HORÁRIA	MÓDULO
4647E-04	60	30

EMENTA:

Discussão de técnicas e padrões para o desenvolvimento de software com qualidade. Reconhecimento, seleção e aplicação de padrões de projeto no desenvolvimento de sistemas de porte médio. Estudo e uso de contratos na especificação e projeto orientado a objetos. Estudo e aplicação de técnicas de teste unitário. Estudo e aplicação de técnicas de verificação de software.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Dominar as técnicas da abordagem de projeto baseado em contratos no desenvolvimento de sistemas de software.
2. Utilizar uma abordagem sistemática para verificar, testar e depurar sistemas de software.
3. Aplicar de forma objetiva padrões de projeto orientados a objetos no desenvolvimento de sistemas de software.
4. Desenvolver as competências e habilidades para a criação de sistemas de complexidade média dentro do paradigma de orientação a objetos.

CONTEÚDO:

Nº DA UNIDADE: 01

CONTEÚDO: Projeto Baseado em Contratos

- 1.1. Especificação de contratos
 - 1.1.1. Fundamentação: triplas de Hoare e asserções
 - 1.1.2. Pré-condições
 - 1.1.3. Pós-condições
 - 1.1.4. Invariantes
 - 1.1.4.1. De classe
 - 1.1.4.2. De laço
 - 1.1.5. Contrato de exceções
- 1.2. Especificação formal de contratos



- 1.3. Introdução à análise e verificação de contratos
 - 1.3.1. Lógica de primeira-ordem, lógica dinâmica e obrigações de prova
 - 1.3.2. Análise estática
 - 1.3.3. Verificação em tempo de execução
 - 1.3.4. Execução simbólica
 - 1.3.5. Provedores de teoremas e SAT/SMT-solvers

Nº DA UNIDADE: 02

CONTEÚDO: Projeto Orientado a Objetos Baseado em Contratos

- 2.1. Modularização de contratos
 - 2.1.1. Herança
 - 2.1.2. Especificação abstrata
 - 2.1.3. Encapsulamento (campos, variáveis e métodos de modelos)
- 2.2. Elementos auxiliares de especificação
 - 2.2.1. Frames
 - 2.2.2. Contratos de blocos de código

Nº DA UNIDADE: 03

CONTEÚDO: Teste e Depuração de Software

- 3.1. Introdução ao teste de software
 - 3.1.1. Tipos e níveis de testes
- 3.2. Introdução ao desenvolvimento guiado por testes (TDD)
- 3.3. Teste de unidade
- 3.4. Critérios de teste
 - 3.4.1. Domínio de entrada
 - 3.4.2. Grafos
 - 3.4.3. Expressões lógicas
 - 3.4.4. Estruturas sintáticas
- 3.5. Automatização de testes
 - 3.5.1. Dublês
 - 3.5.2. Geração de casos de teste

Nº DA UNIDADE: 04

CONTEÚDO: Padrões de Projeto para Sistemas Orientados a Objetos

- 4.1. Projeto orientado a objetos
 - 4.1.1. Critérios de qualidade (encapsulamento, coesão, acoplamento)
 - 4.1.2. Heurísticas de projeto de objetos
- 4.2. Conceitos de padrões de projeto
 - 4.2.1. Visão geral
 - 4.2.2. Categorização de padrões
 - 4.2.2.1. Padrões criacionais
 - 4.2.2.2. Padrões estruturais
 - 4.2.2.3. Padrões comportamentais



PROCEDIMENTOS E CRITÉRIOS DE AVALIAÇÃO:

$$G1 = P1 + P2 + 2*T + ME / 5$$

Onde: □

P1 – conteúdo da unidade 1 e 2;

P2 – conteúdo das unidades 3 e 4;

T – trabalho final;

ME – média de exercícios realizados ao longo do semestre.

BIBLIOGRAFIA BÁSICA:

1. □ LISKOV, B.; GUTTAG, J. Program Development in Java: abstraction, specification, and object-oriented design. Addison-Wesley, 2000
2. □ AHRENDT, W. et al. Deductive Software Verification – The KeY Book: from theory to practice. Lecture Notes in Computer Science, vol. 10001. Springer, 2016.
3. □ AMMANN, P.; OFFUTT, J. Introduction to Software Testing. 2 ed. Cambridge University Press, 2016.

BIBLIOGRAFIA COMPLEMENTAR:

1. □ GAMMA, E. et al. Padrões de projeto: Soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000.
2. □ FOWLER, M.; RICE, D.; FOEMMEL, M.; HIEATT, E.; MEE, R.; STAFFORD, R. Padrões de arquitetura de aplicações corporativas. Porto Alegre: Bookman, 2008.
3. □ BECK, K. TDD desenvolvimento guiado por testes. Porto Alegre: Bookman, 2010.
4. □ MEYER, B. Object-oriented software construction. 2 ed. Upper Saddle River: Prentice Hall, 1997.
5. □ TENNENT, R. D. Specifying Software: a hands-on introduction. Cambridge University Press, 2002.



PROGRAMA DA DISCIPLINA

DISCIPLINA:

CALCULO A (CC)

CODCRED	CARGA HORÁRIA	MÓDULO
41153-04	60	60

EMENTA:

- Estudo e caracterização das funções básicas de uma variável real. Definição e apresentação das propriedades de limites e continuidade. Definição e apresentação das propriedades de derivadas. Aplicações de derivadas e diferenciais. Definição e apresentação das propriedades da integral indefinida. Estudo de técnicas de integração. Aplicação de integral indefinida aos problemas clássicos de equações diferenciais.

OBJETIVOS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

- Manipular com habilidade a linguagem simbólica relacionada aos conceitos de limite e derivada.
- Compreender e aplicar o conceito de derivada como taxa de variação e inclinação de uma curva, tanto a problemas quanto à análise de funções.
- Compreender o conceito de integral, manipular antiderivadas, compreender a linguagem simbólica associada e as aplicações básicas à resolução de equações diferenciais.
- Conhecer os instrumentos básicos de funções e gráficos necessários para a modelagem matemática e resolução de problemas.
- Conhecer softwares científicos utilizados como ferramenta computacional.

CONTEÚDO:

1. FUNÇÕES

Introdução: definição, domínio, imagem, gráficos, problemas aplicados. Funções algébricas e famílias de funções. Funções definidas por partes. Funções exponencial e logarítmica. Novas funções a partir de antigas e operações com funções.

2. LIMITES E CONTINUIDADE

Definição, limites laterais, noção intuitiva e gráficos. Cálculo de Limites: limites básicos, propriedades, polinômios, funções racionais, funções por partes. Limites infinitos e no infinito. Continuidade de uma função.



3. DERIVADAS

Retas tangentes, velocidades e taxas de variação. Função derivada. Técnicas de derivação: propriedades básicas, derivadas de ordem superior, regra do produto, regra do quociente, regra da cadeia, derivação implícita. Funções trigonométricas: gráficos, propriedades, limites, derivadas. Análise de funções: crescimento/decrescimento, extremos locais, concavidade, assíntotas, gráficos, problemas de modelagem. Regra de L'Hôpital. Diferenciais: definição, interpretação geométrica, problemas de modelagem.

4. INTEGRAIS

Integral Indefinida: definição, propriedades, significado geométrico. Estudo de Técnicas de Integração: integração por substituição, integração por partes. Aplicação de integral a problemas de equações diferenciais.

BIBLIOGRAFIA BÁSICA:

ANTON, Howard. Cálculo: um novo horizonte. 6. ed. Porto Alegre: Bookman, 1999-2005. v.1.
STEWART, James. Cálculo. São Paulo: CENGAGE Learning, 2010. v.1.
THOMAS JUNIOR, George B. Cálculo. 10 ed. São Paulo: Pearson Education, 2005. v. 1.

BIBLIOGRAFIA COMPLEMENTAR:

BOULOS, Paulo. Introdução ao cálculo. 2. ed.rev.ampl. São Paulo: Edgar Blücher, 1978-1997, v.1: Cálculo diferencial.
EDWARDS, Charles Henry; PENNEY, David E. Cálculo com geometria analítica. 4. ed. Rio de Janeiro: Prentice-Hall do Brasil, 1997. v.1.
FLEMMING, Diva Marília; GONÇALVES, Miriam Buss. 6. ed.rev.ampl. Cálculo A: funções, limite, derivação, integração. São Paulo: Pearson, 2012. 448 p.
HOFFMANN, Laurence D. Cálculo: um curso moderno e suas aplicações. 2. ed. Rio de Janeiro: LTC, 1990-1996. v.1.
SWOKOWSKI, Earl W. Cálculo com geometria analítica. 2. ed. São Paulo: Makron Books, 1995. v.1.