Table of Contents

# Insper Computer Science

**Ementário**

**Versão 4.7**

\newpage

\newpage

# Vida de Desenvolvedor de Software - Developer Life

Carga Horária: 320

*Semestre 1º período*

## Ementa

Projeto de Software: Conceitos Básicos de Algoritmos; Fundamentos de Programação e Linguagens de Programação (variáveis, expressões, comandos, estruturas de decisão e estruturas de repetição, manipulação de dados estruturados, funções e classes); Resolução Algorítmica de Problemas; Desenvolvimento de Programas; Linguagens de Programação;

Times Ágeis: Processo de desenvolvimento de software; Técnicas de Projeto de Software; Introdução a técnicas de Planejamento e Gerenciamento de Software (Metodologias ágeis); Documentação de projetos de Software; Introdução a ferramentas de controle de versão.

Design Centrado no Usuário: acessibilidade; usabilidade e testes de usabilidade; prototipação e iteração; gráficos e sons; conhecimento do contexto e as pessoas; a natureza da interação com o usuário e ambientes virtuais; interação humano-computador; padrões para interface; definição e métodos de avaliação;

Projeto de Sistemas Complexos: Introdução a arquiteturas de serviços web; Introdução a protocolo HTTP; Introdução a bancos de dados relacionais e operações básica de manipulação de dados (CRUD); Introdução ao desenvolvimento de sistemas Web; Visão estratégica da evolução da tecnologia

## Objetivos

### Programação:

1. Desenvolver programas de computador.
2. Identificar e programar estratégias computacionais de resolução de problemas práticos

### Times Ágeis:

1. Utilizar software de controle de versão para desenvolvimento colaborativo
2. Trabalhar em equipe gerenciada usando Kanban

### Design Centrado no Usuário:

1. Aplicar ferramentas de design gráfico e tecnologias de front-end web em prototipação digital.
2. Avaliar protótipos digitais a partir de personas e cenários.
3. Aplicar testes de usabilidade.

### Projeto de Software:

1. Utilizar as linguagens, frameworks e arquiteturas apresentadas na disciplina a fim de implementar sistemas web;
2. Implementar banco de dados que atenda aos requisitos de persistência de dados de uma aplicação web e trabalhar com requisições HTTP enviando requests e tratando responses;
3. Utilizar as linguagens, frameworks e arquiteturas para resolver problemas reais dos usuários por meio da web;
4. Relacionar assuntos da fronteira da computação como soluções para um problema real;

## Conteúdo programático

### Programação:

1. Introdução a linguagens de programação e como o computador executa programas.
2. Introdução à linguagem Python, entrada e saída de dados.
3. Variáveis e tipos de dados.
4. Operadores condicionais.
5. Operadores de repetição.
6. Cadeias de caracteres.
7. Listas.
8. Funções.
9. Matrizes. 10 Estruturas de dados básicas, pilhas e filas.
10. Introdução à orientação a objetos.
11. Programação interativa e baseada em eventos;

### Times Ágeis:

1. Introdução a Git: repositórios, commits e releases
2. Método Kanban para organizar desenvolvimento de software
3. Versionamento de software

### Design Centrado no Usuário:

1. Princípios básicos de design gráfico.
2. Princípios básicos de desenvolvimento front-end;
3. Tecnologias para front-end para Web: CSS, HTML
4. Usabilidade em aplicações Web / Interação Humano-Computador
5. Prototipação digital.
6. Teste de usabilidade.

### Projeto de Software:

1. Arquitetura de sistemas baseados em web
2. Uso de bancos de dados relacionais via Object Relational Mappers
3. Gerenciamento básico de dados – CRUD (create, read, update and delete)
4. Modelo Model-View-Controller usando framework Django
5. Camada de comunicação entre back-end servidor e banco de dados
6. Geração de páginas web usando Templates
7. Tópicos introdutórios de assuntos avançados como IA, Machine Learning, Processamento de sinais e imagens.

## Bibliografia básica

1. DOWNEY, A. B., Pense em Python, Editora Novatec, 2016
2. SCHWABER, K. , Agile Project Management with Scrum, Microsoft Press, 2004
3. BARNUM, Carol., Usability Testing Essentials, Morgan Kaufmann, 2010

### Programação

Livros:

1. DOWNEY, A. B., Pense em Python, Editora Novatec, 2016
2. BROOKSHEAR, J. G., Ciência da Computação: Uma Visão Abrangente, Bookman., 2005
3. MENEZES, N. N. C., Introdução à Programação Com Python - Algoritmos e Lógica de Programação Para Iniciantes, 1ª ed., Novatec, 2010

### Times Ágeis

Livros:

1. SCHWABER, K. , Agile Project Management with Scrum, Microsoft Press, 2004

Artigo:

BLOM, M.. Is Scrum and XP suitable for CSE Development?. Procedia Computer Science. , v. 1 , n. 1 , p. 1511-1517 , 2010. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050910001699>. Acesso em: 27 maio 2019.

### Design

Livros:

1. BARNUM, Carol., Usability Testing Essentials, Morgan Kaufmann, 2010
2. KNAPP, J. et al., Sprint: O método usado no google para testar e aplicar novas ideias em apenas cinco dias., Intrínseca, 2017
3. RIES, Eric; SZLAK, Carlos (Trad.). , A startup enxuta, Leya, 2012

### Projeto de Software

Livros:

1. DATE, C. J., Introdução a sistemas de banco de dados, Campus, 2004
2. DUCKETT, J. HTML & CSS, Projete e construa websites, Alta Books, 2016

## Bibliografia complementar

1. BROOKSHEAR, J. G., Ciência da Computação: Uma Visão Abrangente, Bookman., 2005
2. MENEZES, N. N. C., Introdução à Programação Com Python - Algoritmos e Lógica de Programação Para Iniciantes, 1ª ed., Novatec, 2010
3. KNAPP, J. et al., Sprint: O método usado no google para testar e aplicar novas ideias em apenas cinco dias., Intrínseca, 2017
4. RIES, Eric; SZLAK, Carlos (Trad.). , A startup enxuta, Leya, 2012
5. DATE, C. J., Introdução a sistemas de banco de dados, Campus, 2004

### Programação

1. PIVA JR, D.; Engelbrecht, A. M.; Nakamiti, G. S.; Bianchi, F, Algoritmos e Programação de Computadores, Elsevier-Campus , 2012
2. SOUZA, Marco A.F.; GOMES, Marcelo M.; SOARES; Marcio V.; CONCILIO, Ricardo., Algoritmos e Lógica de Programação, 2ª ed., CENGAGE Learning, 2011
3. BARRY, Paul., Use a Cabeça! - Python, Alta Books, 2013
4. FEIJÓ, B.; Clua, E.; Silva, F. S. C., Introdução à Ciência da Computação com Jogos: Aprendendo a programar com entretenimento, Campus, 2009

### Times ágeis:

1. BECK, K.; ANDRES, C. , Extreme programming explained: Embrace change, Addison-Wesley,, 2005

### Design

1. WEINSCHENK, Susan., 100 Things Every Designer Needs to Know About People., New Riders, 2011
2. NORMAN, Donald., The Design of Everyday Things, Basic Books, 2002
3. TULLIS, Thomas; ALBERT, William., Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics, 1ª ed., Morgan Kaufmann, 2008
4. GARRETT, Jesse, The Elements of User Experience, 2ª ed., Ed. New Riders, 2010
5. KUMAR, Vijay. , 101 Design Methods: A Structured Approach for Driving Innovation in Your Organization. , Wiley & Sons, 2013

### Projeto de Software:

1. GAMMA, E; HELM, R; JOHNSON, R; VLISSIDES, J., Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos, Bookman, 2000
2. PRESSMAN, R., Engenharia de Software: Uma abordagem profissional., 8ª ed., AMGH, 2016
3. GEORGE, N; Build a Website with Django 3, GNW Independent Publishing, 2019

\newpage

# Ética, Computação e Sociedade

Carga Horária: 72 *Semestre 1º período*

## Ementa

Neutralidade da produção científica e tecnológica. Determinismo tecnológico. Construção social da ciência e da tecnologia. Metodologia científica. Ciência, tecnologia e sociedade. Ética, ciência e tecnologia: direitos humanos e acesso à tecnologia. Tecnologia e sociedade no Brasil em suas dimensões étnicas e raciais: inclusão ou exclusão? Relação entre avanço científico-tecnológico e desenvolvimento econômico.

## Objetivos

Ao final do curso o aluno deverá ser capaz de:

1. Entender e explicar as relações interdisciplinares entre ciência, tecnologia e sociedade;
2. Comparar e prever os efeitos de diferentes escolhas tecnológicas em distintos contextos sociais e econômicos;
   1. Analisar e avaliar os usos sociais da tecnologia à luz de temas e debates éticos contemporâneos.

## Conteúdo Programático

1. O que são: ciência, método científico e tecnologia;
2. Tecnologia como construção social;
3. Ética, ciência e tecnologia;
4. Educação em direitos humanos;
5. Relações étnico-raciais e elementos de história e cultura afro-brasileira, africana e indígena;
6. Inovação tecnológica e seus dilemas éticos;
7. Escolhas tecnológicas e seus usos sociais.

## Bibliografia Básica

Livros:

1. ARAUJO, Marcelo de. Novas Tecnologias e Dilemas Morais. Independently Published, 2019.
2. MOROZOV, Evgeny. Big Tech: A ascensão dos dados e a morte da política. São Paulo: Ubu Editora, 2018.
3. O’Neil Cathy. Algoritmos de destruição em massa: Como o Big Data aumenta a desigualdade e ameaça a democracia. 1ª Edição – Santo André, SP. Editora Rua do Sabão, 2020.

Artigos:

1. COSTA, Alda Cristina. A comunidade indígena e o mundo tecnológico: reflexões sobre os impactos das mídias sociais na vida dos Aikewára. 3º Simpósio Hipertexto e Tecnologias da Educação. , 2010. ; Disponível em: [<http://www.nehte.com.br/simposio/anais/Anais-Hipertexto-2010/Alda-Cristina-Costa.pdf>](http://www.nehte.com.br/simposio/anais/Anais-Hipertexto-2010/Alda-Cristina-Costa.pdf). Acesso em: 22.11.21.
2. LA RUE, Frank. United Nations-General Assembly. Report of the Special Rapporteur on the promotion and protection of the right to freedom of opinion and expression. - Human Rights Council Seventeenth session - Agenda item 3: Promotion and protection of all human rights, civil, political, economic, social and cultural rights, including the right to development, 2011. Disponível em: [<https://digitallibrary.un.org/record/706200>](https://digitallibrary.un.org/record/706200) Acesso em: 22.11.21.
3. SILVA, Tarcízio. Visão Computacional e Racismo Algorítmico: Branquitude e Opacidade no Aprendizado de Máquina. Revista da ABPN. v.12, n.31, dez/2019-fev/2020, pp. 428-448. Disponível em: [<https://www.researchgate.net/publication/339514173_Visao_Computacional_e_Racismo_Algoritmico_Branquitude_e_Opacidade_no_Aprendizado_de_Maquina>](https://www.researchgate.net/publication/339514173_Visao_Computacional_e_Racismo_Algoritmico_Branquitude_e_Opacidade_no_Aprendizado_de_Maquina) Acesso em: 22/11/21

## Bibliografia Complementar

Livros:

1. ALVES, Rubem. Filosofia da Ciência. Introdução ao jogo e as suas regras. São Paulo: Loyola, 2007.
2. HARARI, Yuval N. 21 Lições para o século XXI.São Paulo: 1ªedição. Cia das Letras, 2018.
3. SILVEIRA, Sergio Amadeu da; SOUZA, Joyce; AVELINO, Rodolfo. A sociedade de controle: Manipulação e modulação nas redes. 1ª edição. São Paulo: Editora Hedra, 2019.
4. PRETTO, Nelson De Luca; SILVEIRA, Sergio Amadeu da. Além das redes de colaboração: internet, diversidade cultural e tecnologias do poder. Salvador: EDUFBA, 2008. Disponibilizado gratuitamente pelos autores e pela Universidade Federal da Bahia em Creative Commons: <https://repositorio.ufba.br/ri/bitstream/ufba/211/4/Alem%20das%20redes%20de%20colaboracao.pdf> Acesso em 28.09.21
5. ZUBOFF, Shoshana. A era do capitalismo de vigilância: A luta por um futuro humano na nova fonteira do poder. 1ª Edição: Rio de Janeiro. Editora Intrínseca, 2020.

Artigos:

1. ARBIX, Glauco. A transparência no centro da construção de uma IA ética. Novos estudos CEBRAP.São Paulo. v.39, n.02, mai–ago 2020, pp. 395-413. Disponível em: [<https://doi.org/10.25091/s01013300202000020008>](https://doi.org/10.25091/s01013300202000020008) Acesso 22.11.21
2. MAGNO, M. E. da S. P., & Bezerra, J. S. (2020). Vigilância negra: O dispositivo de reconhecimento facial e a disciplinaridade dos corpos. Novos Olhares, 9(2), 45-52. Disponível em [<https://doi.org/10.11606/issn.2238-7714.no.2020.165698>](https://doi.org/10.11606/issn.2238-7714.no.2020.165698) Acesso 22.11.21.
3. MARQUES, Fabrício. Geografia da inovação: Nova metodologia avalia a robustez dos sistemas de ciência e tecnologia dos estados brasileiros. Pesquisa FAPESP.São Paulo, Edição 288 fev. 2020. Disponível em: [<https://revistapesquisa.fapesp.br/geografia-da-inovacao/>](https://revistapesquisa.fapesp.br/geografia-da-inovacao/) Acesso 22.11.21.

\newpage

# Bits e Processadores

Carga Horária: 72 *Semestre 2º período*

## Ementa

Sistemas Digitais; Sistemas de Numeração e Códigos; Aritmética Binária; Porta Lógica; Análise e Projeto de Circuitos Combinacionais; Minimização por Mapa de Karnaugh; Somadores; Decodificadores; Codificadores; Multiplexadores; Demultiplexadores; Análise e Síntese de Circuitos Sequenciais; Flip-Flops; Registradores; Dispositivos Lógicos Programáveis; Memória; Portas lógicas; Álgebra Booleana; Circuitos Aritméticos e Aritmética Binária; Eletrônica digital; Lógica; Compiladores; Gerenciamento de Memória; Gerenciamento de Dispositivos de Entrada/Saída; Organização e Arquitetura básica de um sistema computacional; Conjunto de Instruções; Benchmarks; Speedup e Lei de Amdahl; Hierarquia de memória; Cache; MissRate; Pipeline; Paralelismo de instruções; Predição de saltos; Paralelismo de dados SIMD, GPU.

## Objetivos

Ao final da disciplina o estudante será capaz de:

1. Implementar um computador digital simples a partir de componentes eletrônicos.
2. Integrar as camadas de programação e execução de um computador simples.
3. Trabalhar de forma colaborativa no desenvolvimento de um sistema computacional.
4. Explicar a evolução da informática.
5. Descrever como dados e instruções são armazenados e tratados em computadores.
6. Entender questões relacionadas a desempenho, operação e manutenção de sistemas digitais

## Conteúdo Programático

1. Lógica Combinacional
2. Benchmarks, speedup
3. Unidade Lógica Aritmética
4. Lógica Sequencial
5. Hierarquia de memória - Cache
6. Lógica de Controle
7. Linguagem de Máquinas
8. Assembly
9. Pipeline, Instruction Level parallelism, jump prediction
10. Assembler
11. Máquina Virtual
12. Data level parallelism Vector, SIMD e GPU
13. Controle de programa

## Bibliografia Básica

Livros:

1. NISAN, Noam; SCHOCKEN, Shimon., The Elements of Computing Systems, Second Edition: Building a Modern Computer from First Principles MIT Press, 2021
2. TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L., Sistemas Digitais: Princípios e Aplicações, 11ª ed., Pearson, 2011
3. HENNESSY, John. Arquitetura de Computadores - Uma Abordagem Quantitativa. Rio de Janeiro: GEN LTC, 2019. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595150669>. Acesso em: 31 de May 2022.

Artigos:

KHAN, S. et. Al.. A high performance processor architecture for multimedia applications.. Computer & Electrical Engineering. , v. 66 , p. 14-29 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045790616305894>. Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. HENNESSY, John. Organização e Projeto de Computadores. Rio de Janeiro: GEN LTC, 2017. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595152908>. Acesso em: 31 de Mai 2022.
2. FLOYD, Thomas. Sistemas Digitais. Porto Alegre: Bookman, 2011. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577801077>. Acesso em: 31 de Mai 2022.
3. MOORE, Cristopher ; MERTENS, Stephan., The Nature of Computation, University Press, 2011
4. STALLINGS, W., Arquitetura e organização de computadores, 8ª ed., Prentice Hall Brasil, 2010
5. LAING, Gordon., Digital Retro: The Evolution and Design of the Personal Computer, Sybex, 2004 NIELSEN, L., Computing: A Business History, New Street Communications LLC, 2012

Artigos:

MISHRA, D.. A survey of memory management techniques in virtualized systems.. Computer Science Review.. , v. 29 , p. 56-73 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013716301186>. Acesso em: 27 maio 2019.

\newpage

# Ciência dos Dados para Computação

Carga Horária: 72 *Semestre 2º período*

## Ementa

Estatística descritiva uni e bidimensional, Tipos de variáveis, Medidas Resumo de Centralidade e Dispersão, Diagramas para visualização dos dados; Análise exploratória de dados; Abertura de bases de dados, pré-processamento, filtragem e cruzamento de bases de dados; Teoria da probabilidade; Modelos probabilísticos para variáveis aleatórias discretas e contínuas; Distribuições amostrais de média, proporção e variância; Intervalos de confiança para parâmetros de interesse; Inferência estatística paramétrica; Inferência não paramétrica baseada em reamostragem.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Carregar bancos de dados, selecionar informações, tratá-las e realizar a limpeza de dados, permitindo sua manipulação
2. Elaborar análises exploratórias de dados (univariadas e multivariadas), utilizando ferramentas estatísticas e computacionais adequadas, gerando visualizações e permitindo suporte a decisões.
3. Especificar as distribuições de probabilidades adequadas para as variáveis quantitativas discretas e contínuas.
4. Conduzir testes inferenciais adequados que possam dar base à tomada de decisão.

## Conteúdo Programático

1. Variáveis quantitativas e qualitativas.
2. Medidas resumo, centralidade e dispersão.
3. Diagramas e recursos gráficos.
4. Introdução ao ambiente de tratamento de dados.
5. Abertura de base de dados, tratamento de valores inválidos, filtragem e seleção.
6. Análise bidimensional.
7. Teoria da probabilidade. Machine
8. Introdução à classificação.
9. Variáveis e distribuições discretas e contínuas.
10. Inferência estatística e por reamostragem.

## Bibliografia Básica

Livros:

1. MONTGOMERY, D. , Estatística Aplicada e Probabilidade para Engenheiros , 6ª ed., LTC, 2016
2. WILKE, C. O., Fundamentals of Data Visualization, 1ª ed. O'Reilly, 2019. Disponível em <https://clauswilke.com/dataviz>. Acesso em jul 20222.

Bibliografia Complementar

1. MAGALHÃES, M. N.; DE LIMA, A. C. P. , Noções de Probabilidade e Estatística, 7ª ed., Edusp, 2013
2. DOWNEY, A.B., Think Stats, 1ª ed., O'Reilly Media, 20111 1 . DEKKING, F. M.; KRAAIKAMP, C. , A Modern Introduction to Probability and Statistics: Understanding Why and How., Springer, 2010
3. KAFLIC, C. N., Storytelling com Dados, 2ª. ed, Alta Books, 2019

\newpage

# Matemática do Contínuo

Carga Horária: 98 *Semestre 2º período*

## Ementa

Cálculo Diferencial e Integral com funções de uma variável: limites, derivadas e aplicações, integrais, aproximação de funções. Geometria do R3: curvas e superfícies parametrizadas. Cálculo Diferencial com funções de várias variáveis: derivadas parciais, derivadas direcionais e otimização.

## Objetivos

Ao final do curso, o aluno deve ser capaz de:

1. Quantificar, interpretar e expressar algébrica e graficamente a derivada de uma função de uma variável real (OA1).
2. Calcular a integral de uma função e relacioná-la ao problema do cálculo do valor acumulado de uma variável dependente (OA2).
3. Interpretar e representar figuras geométricas tridimensionais por meio do estudo de curvas e superfícies parametrizadas e gráficos de funções de várias variáveis (OA3).
4. Relacionar as diferentes ferramentas do Cálculo Diferencial e Integral a métodos de cálculo numérico (OA4).

## Conteúdo Programático:

1. Modelagem discreta e contínua e os conjuntos numéricos: inteiros, racionais, reais.
2. Limite e continuidade de funções.
3. Taxa de variação e a noção de derivada.
4. Derivada em um ponto e função derivada: interpretações algébrica e geométrica.
5. Funções diferenciáveis.
6. Cálculo da derivada de diferentes funções.
7. Aplicações de derivadas: máximos e mínimos, concavidade, gráficos.
8. Integral definida e indefinida.
9. Teorema Fundamental do Cálculo.
10. Diferenciais, aproximações lineares e polinômio de Taylor.
11. Curvas e superfícies parametrizadas no R3.
12. Funções de Rn em R: definição, domínio, gráfico, curvas de nível.
13. Derivadas parciais e derivadas direcionais.
14. Vetor gradiente e algoritmos de otimização.

## Bibliografia Básica

Livros:

1. STEWART, J. , Cálculo, Volumes I, 7ª ed., Cengage Learning, 2013
2. STEWART, J. , Cálculo, Volume II, 7ª ed., Cengage Learning, 2013
3. ANTON, H.; RORRES, C., Álgebra Linear com Aplicações, 10ª ed., Bookman, 2012

## Bibliografia Complementar

Livros:

1. BOYCE, W. E.; DIPRIMA, R. C., Equações Diferenciais Elementares e Problemas de Valores de Contorno, 9ª ed., LTC, 2013
2. APOSTOL, T. M., Cálculo 1: Cálculo com Funções de uma Variável, com uma Introdução à Álgebra Linear, Reverté , 1998
3. POOLE, D, Álgebra Linear, 1ª ed., Pioneira Thomson Learning, 2004 ZILL, D., CULLEN, M. , Equações Diferenciais, 3ª ed., Pearson, 2001
4. ROGAWSKI, J., Cálculo, Volume 1, 1ª ed., Bookman, 2009

\newpage

# Projeto Ágil

Carga Horária: 72 *Semestre 2º período*

## Ementa:

Introdução à Engenharia de Software e fases do processo de desenvolvimento de software; Identificação do problema real aplicando a metodologia de design thinking; Representação de estrutura e comportamento de um modelo em diagramas UML: Casos de uso, classe, sequência, colaboração e implantação; Conceitos de Programação Orientada a Objetos; Técnicas de Planejamento e Gerenciamento de Software com o framework ágil SCRUM; Documentação de projetos de Software; Estratégias para Construção de Software; Mapeamento de modelo de análise em modelo de objetos; Representação de estrutura e comportamento de um modelo em diagramas UML: casos de uso, classe, sequência, colaboração e implantação; Planejamento de implementação de um código; Requisitos funcionais e não funcionais de software; Critérios de qualidade de código;

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Atuar em uma equipe que gere um projeto por meio dos métodos ágeis SCRUM e Kanban, revezando papeis nesta equipe
2. Utilizar o framework ágil SCRUM para gerir o desenvolvimento de um projeto.
3. Aplicar metodologia do design thinking para identificar e validar problemas de usuários
4. Identificar requisitos de sistema e expressá-los em casos de uso e histórias de usuário
5. Estruturar um sistema orientado a objeto a partir de requisitos
6. Expressar arquitetura e comportamento de um sistema a partir de diagramas UML

## Conteúdo Programático

1. Design Thinking
2. Introdução à Engenharia de Software: fases do processo de desenvolvimento de software
3. Introdução ao framework ágil Scrum e Kanban
4. Introdução à programação orientada a objetos
5. Diagramas UML
6. Introdução ao UML
7. Mapeamento de requisitos em objetos de domínio

## Bibliografia Básica

1. VALENTE, M.T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Publicação Independente. 1.a edição. Belo Horizonte, 2020. Disponível em <https://engsoftmoderna.info/>, Acesso em 20 jun 2024.
2. SCHWABER, K. , Agile Project Management with Scrum, Microsoft Press, 2004
3. THOMAS, D. HUNT, A. The Pragmatic Programmer: Your Journey to Mastery. 2.a ed. Addison-Wesley Professional, 2019.

## Bibliografia Complementar

1. CAROLI, P. Lean Inception: Como Alinhar Pessoas e Construir o Produto Certo. 1.a ed. Caroli, 2018.
2. MARTIN, ROBERT C. Clean Code: A Handbook of Agile Software Craftsmanship 2008. 1.a ed. Pearson.
3. PRESSMAN, R., Engenharia de Software: Uma abordagem profissional., 8ª ed., AMGH, 2016

# Programação Eficaz

## Ementa:

Desenvolvimento em ambiente colaborativo e controle de versão: práticas e ferramentas como Git; Introdução ao desenvolvimento de aplicações web: componentes, arquitetura, e separação de responsabilidades; Exploração eficiente de bases de código e documentação: técnicas e importância; Web services REST e Maturidade de Richardson: princípios e aplicação prática; Deploy completo de aplicações web: estratégias e ferramentas para frontend, backend e banco de dados; Desenvolvimento Orientado por Testes (TDD): conceitos e implementação; Automação de testes e uso de clients HTTP: estratégias e ferramentas; Uso de bases de dados relacionais e modelagem relacional: PostgreSQL e tipos de relacionamentos; Utilização e consumo de APIs de terceiros: integração e melhores práticas; Depuração eficiente de programas e tratamento de erros e exceções: estratégias e melhores práticas; Monitoramento de software e logging: técnicas e sistemas de logging; Introdução ao NoSQL com MongoDB: conceitos e práticas; Autenticação e autorização: mecanismos e implementação.

## Objetivos:

Capacitar o aluno a projetar, implementar e manter aplicações web modernas, empregando práticas avançadas como TDD, REST, automação de testes, e programação colaborativa. Introduzir bases de dados relacionais e NoSQL, e ensinar o processo completo de deploy de aplicações web.

Ao final da disciplina, o aluno será capaz de:

1. Desenvolver aplicações web completas, integrando frontend e backend.
2. Aplicar TDD para melhorar a qualidade do software.
3. Criar e consumir web services RESTful, melhorando a interoperabilidade.
4. Automatizar testes, garantindo a robustez da aplicação.
5. Utilizar bases de dados relacionais e NoSQL para gestão de dados eficiente.
6. Executar deploy completo de aplicações, incluindo configurações de servidor e banco de dados.
7. Integrar APIs de terceiros, expandindo a funcionalidade das aplicações.
8. Implementar autenticação e autorização, assegurando a segurança da aplicação.
9. Utilizar ferramentas de monitoramento e logs para manutenção proativa.
10. Colaborar em equipes de desenvolvimento distribuídas, utilizando controle de versão e práticas de código limpo.

## Conteúdo Programático:

1. Desenvolvimento em Ambiente Colaborativo e Controle de Versão: Práticas de desenvolvimento colaborativo; Uso de sistemas de controle de versão (ex: Git).
2. Introdução ao Desenvolvimento de Aplicações Web: Visão geral sobre aplicações web modernas e seus componentes (frontend e backend); Arquitetura de aplicações web: separação de responsabilidades entre cliente e servidor.
3. Exploração Eficiente de Bases de Código e Documentação: Técnicas para navegação e compreensão de bases de código existentes; Importância e práticas de documentação de software.
4. Web Services REST e Maturidade de Richardson: Princípios fundamentais de REST e serviços web RESTful; Aplicando os níveis da Maturidade de Richardson em projetos práticos.
5. Deploy Completo de Aplicações Web: Estratégias e ferramentas para deploy de aplicações: frontend, backend (REST), e base de dados.
6. Desenvolvimento Orientado por Testes (TDD): Conceitos e práticas de TDD; Implementação de ciclos de TDD para o desenvolvimento sustentável de software.
7. Automação de Testes e Uso de Clients HTTP: Estratégias para automação de testes em aplicações web: Utilização de ferramentas e clients HTTP para testes e desenvolvimento.
8. Uso de Bases de Dados Relacionais e Modelagem Relacional: Introdução ao PostgreSQL: conceitos básicos e práticas; Modelagem relacional: relacionamentos 1-1, 1-N, e N-M.
9. Utilização e Consumo de APIs de Terceiros: Integração com serviços externos via APIs; Melhores práticas e padrões de consumo.
10. Depuração Eficiente de Programas e Tratamento de Erros e Exceções: Estratégias e ferramentas para depuração de código; Melhores práticas para tratamento de erros e exceções.
11. Monitoramento de Software e Logging: Técnicas para monitoramento de aplicações web; Implementação de sistemas de logging para rastreamento de erros e atividades.
12. Introdução ao NoSQL com MongoDB:Conceitos fundamentais do NoSQL; Práticas com MongoDB: modelagem de dados e operações básicas.
13. Autenticação e Autorização: Mecanismos de controle de acesso; Implementação de sistemas de autenticação e autorização em aplicações web.

# Bibliografia Básica

1. VALENTE, M.T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Publicação Independente. 1.a edição. Belo Horizonte, 2020. Disponível em <https://engsoftmoderna.info/>, Acesso em 20 jun 2024.
2. THOMAS, D. HUNT, A. The Pragmatic Programmer: Your Journey to Mastery. 2.a ed. Addison-Wesley Professional, 2019.

# Bibliografia Complementar

1. MARTIN, ROBERT C. Clean Code: A Handbook of Agile Software Craftsmanship 2008. 1.a ed. Pearson.
2. MCCONNEL, S. Code Complete: A Practical Handbook of Software Construction. 2.a ed. Microsoft Press, 2004.
3. K.BECK., Test Driven Development: By example, Addison-Wesley Professional, 2002

# Sprint Session 2

Carga horária: 44h

## Ementa:

Elaboração da arquitetura de um software de armazenamento ou processamento de dados; Desenvolvimento de solução; Prototipação de software; Comunicação oral e escrita.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Interpretar requisitos de um cliente relacionadas a um software centrado em dados
2. Propor uma solução
3. Esboçar a arquitetura de uma solução argumentando sobre restrições de projeto, demonstrando compreensão da computação em camadas e de forma integrada
4. Atuar em um time gerenciado por métodos ágeis na criação de um protótipo
5. Apresentar o projeto e criar documentação escrita

## Conteúdo programático

1. Coleta de requisitos de cliente
2. Arquitetura de software
3. Gestão de times ágeis
4. Comunicação escrita
5. Técnicas de apresentação

## Bibliografia Básica

1. VALENTE, M.T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Publicação Independente. 1.a edição. Belo Horizonte, 2020. Disponível em <https://engsoftmoderna.info/>, acesso em jul 2024.
2. BROWN, TIM. Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, 2019.

## Bibliografia Complementar

1. MCCONNEL, S. Code Complete: A Practical Handbook of Software Construction. 2.a ed. Microsoft Press, 2004.
2. SCHWABER, K. , Agile Project Management with Scrum, Microsoft Press, 2004
3. BECK, K.; ANDRES, C. , Extreme programming explained: Embrace change, Addison-Wesley,, 2005

\newpage

# Técnicas de Programação

Carga Horária: 72 *Semestre 3º período*

## Ementa

Estratégias de resolução de problemas computacionais: recursão, divisão e conquista, backtracking; Algoritmos de ordenação: quicksort e mergesort; Algoritmos de busca: binária, em largura e profundidade; Introdução à notação assintótica; algoritmos simples com cadeias de caracteres; Pilhas, filas, matrizes e tabelas de Hash.

## Objetivos

Ao final do curso, o aluno deve ser capaz de...

1. Implementar em Java um algoritmo descrito em pseudo-código
2. Empregar Backtracking para resolver problemas computacionais
3. Empregar a técnica Divisão e Conquista para resolver problemas computacionais
4. Empregar estruturas de dados lineares (listas, matrizes, pilhas, filas, mapeamentos, conjuntos) para resolver problemas computacionais de maneira eficiente
5. Identificar como estratégias computacionais clássicas (busca, ordenação, otimização) podem ser adaptadas para resolver novos problemas computacionais
6. Estimar a complexidade computacional de um algoritmo usando uma argumentação informal baseada na contagem de vezes que uma linha executa

## Conteúdo Programático

* Tipos Abstratos de Dados
* Estruturas lineares: arrays, listas, pilhas e filas
* Tabelas de hash
* Matrizes
* Recursão
* Estratégias de resolução de problemas: backtracking e divisão e conquista
* Busca
* Introdução a notação assintótica
* Operações com strings

## Bibliografia Básica

1. SEDGEWICK, R.; WAYNE, K., 2011. Algorithms. Addison-Wesley Professional; 4th edition, 2011
2. ZIVIANI, Nivio. Projeto de Algoritmos: com implementações em Java e C++. São Paulo: Cengage Learning, 2011
3. SEDGEWICK, R.; WAYNE, K., Introduction to programming in Java: an interdisciplinary approach. Addison-Wesley Professional; 2017

## Bibliografia Complementar

1. VAUGHAN, L. Impractical Python: Playful Programming Activities to Make You Smarter. Ed. No Starch Press. 1.a ed. 2018.
2. WIRTH, N., Algoritmos e Estruturas de Dados. Rio de Janeiro: Editora LTC, 1989.
3. ZIVIANI, Nívio. Projeto de Algoritmos: com implementação em Pascal e C. 3ª ed. revista e ampliada. São Paulo: Cengage Learning, 2011
4. WENGROW, J. A Common-Sense Guide to Data Structures and Algorithms. 2e. Ed. The Pragmatic Programmers. 2020.
5. GUTTAG, J. V. Introduction to Computation and Programming Using Python: With Applications to Understanding Data. Ed. MIT Press. 2. ed. 2016.

\newpage

# Inteligência Artificial e Robótica

Carga Horária: 72 *Semestre : 3º período*

## Ementa

Introdução à Inteligência Artificial; Definições de Agente Autônomo; Arquitetura computacional de agentes e o laço percepção – planejamento e ação; Caracterização de Ambientes; Resolução de problemas usando espaço de busca; Estratégias de busca; Algoritmos de busca cega e informados; Conceito de Heurística; Teoria de Jogos e Ambientes Competitivos; Aprendizagem por Reforço; Percepção, sensores e incerteza; Noções de visão computacional e reconhecimento de padrões; Aplicação comercial de robôs e usos emergentes, soluções de plataformas robóticas e de software para robôs (R.O.S, OpenCV).

## Objetivos

Ao final da disciplina o estudante será capaz de:

1. Descrever os conceitos, técnicas e métodos para o desenvolvimento de Agentes Autônomos.
2. Identificar quais tipos de problemas podem ser resolvidos através do uso de Agentes Autônomos.
3. Criar soluções para alguns problemas clássicos desta área.
4. Especificar, desenvolver e testar projetos que façam uso de Agentes Autônomos para resolver problemas complexos.
5. Planejar e executar um trabalho em equipe, fornecendo e assimilando devolutivas.

## Conteúdo Programático

1. Definições de Agente Autônomo e resolução de problemas.
2. Estratégias de busca: algoritmos de busca cega e algoritmos informados.
3. Heurísticas.
4. Implementação de agentes autônomos utilizando estratégias de busca.
5. Programação por restrições (CSP).
6. Ambientes competitivos e teoria de jogos.
7. Algoritmo Min-Max e função de utilidade.
8. Implementação de agentes autônomos para ambientes competitivos.
9. Aprendizagem por Reforço.
10. Implementação de agentes autônomos usando aprendizagem por reforço.
11. Algoritmo Q-Learning.
12. Implementações de agentes autônomos usando o projeto Gym.
13. Implementação de um agente robótico.

## Bibliografia Básica

1. NORVIG, P.; RUSSELL, S., Inteligência Artificial, 3ª ed., Campus Elsevier, 2013

## Bibliografia Complementar

1. O'KANE, J., A Gentle introduction to ROS, CreateSpace Publishing, 2013
2. SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D., Introduction to Autonomous Mobile Robots., 2ª ed., MIT Press, 2011
3. SILVER, D.; SINGH S.; PRECUP D.; SUTTON R. Reward is enough. Artificial Intelligence. Vol 299, 2021. Disponível em [<https://doi.org/10.1016/j.artint.2021.103535>](https://doi.org/10.1016/j.artint.2021.103535).
4. SILVER, D.; HUBERT T.; SCHRITTWIESER, J.; ANTONOGLOU, I.; LAI, M.; GUEZ, A. [A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play](https://doi.org/10.1126/science.aar6404). Science 362, 1140-1144 (2018).

\newpage

# Álgebra Linear e Teoria da Informação

Carga Horária: 98 *Semestre : 3º período*

## Ementa

Vetores e espaços vetoriais; geometria em espaços vetoriais; sistemas de equações lineares; transformações lineares; decomposição em valores singulares; Informação e incerteza, entropia de Shannon, informação mútua; codificação e compressão, códigos corretores de erro.

## Objetivos

Ao final do curso, o aluno deve ser capaz de

1. Aplicar conceitos de espaços vetoriais para a descrição e manipulação de objetos
2. Resolver problemas de geometria espacial envolvendo conceitos de espaços vetoriais
3. Avaliar e resolver sistemas de equações lineares
4. Interpretar e analisar transformações lineares
5. Aplicar a decomposição em valores singulares
6. Explicar as noções de informação e codificação
7. Aplicar métodos de compressão e de correção de erro

Conteúdo Programático:

1. Vetores e espaços vetoriais
2. Norma e produto interno, distância, geometria linear: ângulo, projeção, plano
3. Combinações lineares, bases e dimensão, ortogonalização de Gram-Schmidt
4. Sistemas de equações lineares, eliminação de Gauss
5. Transformações lineares e representação matricial
6. Mudança de base e similaridade de transformações
7. Determinantes
8. Autovalores e autovetores
9. Decomposição em valores singulares
10. Informação e incerteza
11. Entropia de Shannon, informação mútua
12. Codificação e compressão
13. Códigos corretores de erro

Bibliografia Básica

1. HEFFERON, J.; Linear Algebra. 4.a ed. <https://hefferon.net/linearalgebra>
2. KLEIN, P. N.; Coding the Matrix: Linear Algebra Through Applications to Computer Science. Newtonian Press. 2013. <https://codingthematrix.com/>

Bibliografia Complementar

1. STRANG. G.; Álgebra Linear e Suas Aplicações. Ed. Cengage Learning. São Paulo: 2019
2. MACKAY, D. J.; Information Theory, Inference and Learning Algorithms.
3. STRANG. G.; Linear Algebra and Learning from Data. Wellesley-Cambridge Press. Wellesley: 2019.
4. ANTON, H.; RORRES, C. Álgebra Linear com Aplicações. Ed. Bookman. São Paulo: 2012.

\newpage

# Lógica e Matemática Discreta

Carga Horária: 72 *Semestre : 3º período*

## Ementa

Lógica proposicional; lógica de predicados; técnicas de provas; provadores automatizados; teoria de conjuntos; conjuntos numéricos; técnicas de enumeração; técnicas de contagem; relações e funções discretas; funções computáveis.

## Objetivos

Ao final do curso, o aluno deve ser capaz de:

1. Produzir abstrações domínios de problemas e representá-las utilizando Lógica
2. Realizar inferências e provas em domínios
3. Enumerar e contar elementos de domínios
4. Identificar relações e funções discretas entre elementos de domínios

## Conteúdo Programático

1. Conjuntos discretos: representação, operações, propriedades
2. Conjuntos numéricos: naturais, inteiros, racionais, reais, complexos e hipercomplexos, propriedades e aplicações
3. Técnicas de provas matemáticas: direta, indireta, redução ao absurdo e indução finita
4. Cardinalidade de conjuntos numéricos, números finitos e transfinitos, aplicações
5. Técnicas de enumeração e contagem diretas, aplicações
6. Contagem indutiva, relações de recorrência e aplicações
7. Relações e funções discretas e aplicações
8. Funções discretas computáveis
9. Lógica Proposicional: representação, inferência e aplicações
10. Lógica de Predicados: representação, inferência e aplicações
11. Provadores automatizados para Lógica Proposicional e de Lógica de Predicados

## Bibliografia Básica

Livros:

1. GERSTING, Judith L. Fundamentos Matemáticos para a Ciência da Computação: Um Tratamento Moderno de Matemática Discreta, 7ª ed., Editora LTC, 2016.
2. SILVA, Flávio S.C.; Finger, M.; de Melo, Ana C.V. Lógica para Computação, 2a. ed. Ed. Cengage Learning, 2017.
3. HUTH, M.; RYAN, M. Logic in Computer Science: Modelling and Reasoning about Systems, 2a. ed. Cambrige University Press, 2004.

## Bibliografia Complementar

Livros:

1. GRAHAM, R.; KNUTH, D.; PATASHNIK, O. Concrete Mathematics: A Foundation for Computer Science. 2 ed. Ed. Addison-Wesley 1994.
2. MENDELSON, E. Introduction to Mathematical Logic (Discrete Mathematics and Its Applications). 6 ed. Ed. CRC Press, 2015.
3. VELLEMAN, D. How to Prove It: A Structured Approach. 3 ed. Ed. Cambridge University Press, 2019.
4. BISPO, C. A. F.; CASTANHEIRA, L. B.; MELO, S. F., O. Introdução à Lógica Matemática. Ed. Cengage Learning, 2012.
5. STEIN, C., DRYSDALE, R. L., BOGART, K. Matemática Discreta para Ciência da Computação. Editora Pearson, 2013.

  \newpage

# Arquitetura de Objetos e Times Ágeis

Carga Horária: 72 *Semestre : 3º período*

## Ementa

Redes de computadores; Internet; Roteamento e interconexão de redes; Introdução ao TCP/IP; Protocolo HTTP; Arquiteturas de serviços web; Serviços REST e serviços SOAP; Representação de dados estruturados para tráfego e armazenamento de informações: JSON; Tecnologias de front-end Web; Planejamento e prototipação de front-ends web; Interfaces adaptáveis a múltiplos dispositivos clientes; Introdução a bancos de dados relacionais e SQL; Desenvolvimento de sistemas back-end (servidor); Usabilidade na web (Heurísticas de Usabilidade?! <https://www.nngroup.com/articles/ten-usability-heuristics/>).

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Interpretar e expres¬sar as necessidades dos usuários em plataformas web programadas seguindo as melhores práticas do mercado;
2. Implementar banco de dados que atenda aos requisitos de persistência de dados de uma aplicação web e trabalhar com requisições HTTP enviando requests e tratando responses;
3. Utilizar as linguagens, frameworks e arquiteturas apresentadas na disciplina a fim de implementar sistemas web;
4. Utilizar as linguagens, frameworks e arquiteturas para resolver problemas reais dos usuários por meio da web;
5. Planejar e acompanhar o projeto conforme princípios e valores ágeis, usando um framework diferente dos utilizados em outras disciplinas (Kanban, Scrum, squad, etc)
6. Consultar a documentação de bibliotecas/frameworks web e usá-las no desenvolvimento de um sistema web

## Conteúdo Programático:

Arquitetura da Web: IP, TCP/IP e roteamento

1. Arquitetura da Web: HTTP e WWW
2. Tecnologias para front-end: CSS, HTML, Javascript e Ajax/JSON
3. Usabilidade em aplicações Web / Interação Humano-Computador
4. Programação de servidores em Java: Servlets
5. SQL: Introdução e criação de CRUD
6. Camada de comunicação entre back-end servidor e banco de dados
7. Projeto de aplicações seguindo arquitetura Model-View-Controller
8. Hospedagem de aplicações web e publicação do trabalho na web
9. Projeto de aplicações seguindo arquitetura REST
10. Bases de dados não relacionais (NoSQL)
11. Front-ends com frameworks single-page (Angular)
12. Backends com Node JS

## Bibliografia Básica

Livros:

1. DATE, C. J., Introdução a sistemas de banco de dados, Campus, 2004
2. COMER, D. E., Redes de Computadores e Internet, 6ª ed., Bookman, 2016
3. LUCKOW, D. H.; DE MELO, A. A., Programação Java para a Web, 2ª ed., Novatec, 2015

Artigos:

GUTIERREZ, R. T.. Understanding the role of digital commons in the web: the making of HTML 5.. Telematics and Informatics. , v. 35 , n. 5 , p. 1438-1449 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S0736585317306342> . Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. SILVEIRA, P.et al., Introdução à Arquitetura e Design de Software: Uma Visão Sobre a Plataforma Java., 1ª ed., Campus Elsevier, 2012
2. DUCKETT, J.; RUPPERT, G.; MOORE, J., JavaScript & JQuery: desenvolvimento de interfaces web interativas, Alta Books, 2016
3. DUCKETT, J. HTML & CSS, Projete e construa websites, Alta Books, 2016
4. CANTELON, M. , Node. js in Action, 2ª ed., Manning, 2017
5. MCGOVERN, J., Java web services architecture, Morgan Kaufmann, 2003

Artigos:

GARRIGA, M. et al.. RESTful composition at a glance: A survey. Journal of Network and Computer Applications. , v. 60 , n. 32 , p. 32-53 , 2016. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804515002933>. Acesso em: 27 maio 2019.

\newpage

# Sprint Session 03 - Inovação Social

## Ementa

Computação como elemento de transformação e inovação na sociedade; Vieses inconscientes; Conhecimento do contexto sócio-econonômico dos usuários; Design Thinking para inovação social; Análise de requisitos; Modelagem de sistemas; Desenvolvimento, testes e deploy de solução de software; Exercício de práticas ágeis; Técnicas de feedback dentro de times; Reflexões sobre qualidade de trabalho em equipe.

## Objetivos de Aprendizagem

Ao final da disciplina o aluno será capaz de:

1. Projetar e prototipar software que atende às necessidades dos usuários por meio do uso de conhecimentos prévios de Ciência da Computação, Engenharia de Software e Ciência de Dados para resolver um problema do mundo real.
2. Empregar métodos ágeis para gestão das tarefas do grupo e feedback sobre andamento das atividades e sua gestão e organização.
3. Criar soluções a problemas reais que respondam às necessidades dos usuários seguindo boas práticas de design e experiência do usuário.
4. Utilizar técnicas de feedback, refletir sobre propósito, processos, papéis do grupo, comprometimento e aprendizados.

## Conteúdo Programático:

1. Inovação Social
2. Vieses Inconscientes
3. Processo de Design Thinking
4. Entendimento de dores dos usuários a partir de imersões e entrevistas
5. Análise de requisitos e priorização
6. Desenvolvimento de solução considerando requisitos do contexto
7. Planejamento da arquitetura do software
8. Gerenciamento do desenvolvimento usando métodos ágeis
9. Deploy da Solução

## Bibliografia básica

1. SMITH, A. et al. IDDS Design Workbook. Massachusetts Institute Of Technology, 2014. Disponível em: <https://www.idin.org/sites/default/files/resources/IDDS%20Design%20Workbook.pdf>. Acesso em: 01 jul. 2024.
2. VALENTE, M.T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Publicação Independente. 1.a edição. Belo Horizonte, 2020.
3. GAMA, KIEV et al. Hackathons as Inclusive Spaces for Prototyping Software in Open Social Innovation with NGOs. 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), Melbourne, Australia, 2023, pp. 58-70, doi: 10.1109/ICSE-SEIS58686.2023.00012.

## Bibliografia complementar

1. GABRYSIAK, G. et al. 2013. Cooperating with a non-governmental organization to teach gathering and implementation of requirements. In 2013 26th International Conference on Software Engineering Education and Training (CSEE&T). IEEE, 11–20. Disponível em <https://hpi.de/fileadmin/user_upload/fachgebiete/giese/pdf/cseet13main-id96.pdf>, acesso em jul 2024.
2. IRANY, LILLY. Hackathons and the Making of Entrepreneurial Citizenship. Science, Technology, & Human Values, Vol. 40, No. 5 (September 2015), pp. 799-824. Disponível em <https://www.jstor.org/stable/43671232>, acesso em jul 2024
3. INSTITUTO LOCOMOTIVA. 2021. Educação, cultura, periferia e racismo: um levantamento 778 do Instituto Locomotiva para a Central Única das Favelas. Technical Report. Disponível em <https://ilocomotiva.com.br/estudos/educacao-cultura-periferia-e-racismo/>

\newpage

# Algoritmos e Estruturas de Dados

Carga Horária: 72 *Semestre 4º período*

## Ementa

Recursão; comportamento assintótico e estratégias para projeto de algoritmos; memoização, divisão e conquista, backtracking; algoritmos de pesquisa; algoritmos de ordenação clássicos: bubble sort, selection sort, insertion sort, merge sort, heap sort, quick sort; algoritmos de ordenação lineares: counting sort e radix sort; heaps e filas de prioridade; estruturas de dados; inserção e remoção; listas encadeadas; implementação de pilhas e filas; tabelas hash; tipos abstratos de dados; conjuntos; mapas; caracterização de árvores: árvores binárias, árvores de busca binária, árvores de busca binária balanceadas; grafos; busca em largura e profundidade; algoritmo de Bellman-Ford-Moore; algoritmo de Dijkstra.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Inferir algoritmos procedurais iterativos a partir de ideias centrais.
2. Inferir algoritmos procedurais recursivos a partir de especificações de problemas.
3. Simular algoritmos procedurais recursivos.
4. Analisar a correção de algoritmos procedurais.
5. Analisar a eficiência de algoritmos procedurais.
6. Avaliar algoritmos procedurais.
7. Avaliar estruturas de dados.
8. Implementar estruturas de dados com gerenciamento manual de memória.
9. Implementar estruturas de dados com programação orientada a objetos.
10. Criar dinâmicas para aprendizado ativo.

## Conteúdo Programático

1. Introdução à Linguagem C.
2. Inferência de algoritmos iterativos a partir de problemas.
3. Simulação de algoritmos iterativos.
4. Inferência de algoritmos iterativos a partir de ideias.
5. Inferência de algoritmos recursivos a partir de problemas.
6. Simulação de algoritmos recursivos.
7. Introdução à análise de algoritmos.
8. Divisão e conquista.
9. Programação dinâmica.
10. Estruturas de dados lineares.
11. Algoritmos em grafos.
12. Filas de prioridade.
13. Estruturas de dados para dicionários.
14. Impacto da constante.

## Bibliografia Básica

Livros:

1. FEOFILOFF, P, Algoritmos em Linguagem C., Elsevier, 2008
2. CORMEN, T. H.; LEISERSON, C.E.; RIVEST, R. L.; STEIN, C., Algoritmos: Teoria e Prática, 3ª ed., Elsevier, 2012
3. SEDGEWICK, R. , Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, 3ª ed., Addison-Wesley Professional, 1997

Artigos:

RUBIO, M. A. et al.. Closing the gender gap in an introductory programming course.. Computers & Education. , v. 82 , p. 409-420 , 2015. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S0360131514002802>. Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. SKIENA, S. S., The Algorithm Design Manual, 2ª ed., Springer, 2008 CORMEN, T. H., Algorithms Unlocked, MIT Press,, 2013
2. BENTLEY, J., Programming Pearls, Addison-Wesley, 1999
3. AHO, A. V.; ULLMAN, J. D. , Foundations of Computer Science: C Edition. W. H. , Freeman, 1994
4. R.SEDGEWICK., Algorithms in C, Part 5, 3ª ed., Addison-Wesley, 2001

Artigos:

CODISH, M. et al.. Sorting nine inputs requires twenty-five comparisons.. Journal of Computer and System Sciences. , v. 82 , n. 3 , p. 551-563 , 2016. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022000015001397>. Acesso em: 27 maio 2019.

\newpage

# Machine Learning Para Computação

Carga Horária: 72 *Semestre : 4.o período*

## Ementa

Esta disciplina apresenta os fundamentos de aprendizado de máquina (machine learning), e suas aplicações em Engenharia. Os alunos irão aprender sobre as principais áreas do aprendizado de máquina: métodos supervisionados e não supervisionados, para classificação e regressão. Além disso, os alunos aprenderão sobre tópicos avançados na área, como reinforcement learning e deep learning. Nesta disciplina os alunos construirão seu conhecimento através de experimentos práticos com Python e bibliotecas para aprendizado de máquina, como scikit-learn e TensorFlow.

## Objetivos

O aluno deve ser capaz de:

1. Conceituar um problema envolvendo aprendizado de máquina: identificação de características, escolha de modelo e de estratégia de aprendizado
2. Projetar e implementar uma solução de aprendizado de máquina para problemas de regressão multivariada, e avaliar a qualidade do modelo obtido.
3. Projetar e implementar uma solução de aprendizado de máquina para problemas de classificação supervisionada, e avaliar sua sensibilidade e especificidade.
4. Projetar e implementar uma solução de aprendizado de máquina para problemas simples de classificação não-supervisionada, e avaliar a viabilidade do modelo obtido.
5. Descrever e experimentar com técnicas de reinforcement learning
6. Descrever e experimentar com técnicas de deep learning

## Conteúdo Programático

1. O aprendizado de máquina: análise exploratória, arquitetura de solução, seleção de características, métodos de avaliação (1,3)
2. Regressão: Regressão linear, regularização, Support Vector Machines, redes neurais (3)
3. Classificação: Regressão logística, árvores de decisão, boosting e métodos de ensemble (3)
4. Clustering e métodos não-supervisionados, análise de componentes principais. (3)
5. Deep learning. (3)
6. Reinforcement learning (2,3)

## Bibliografia Básica

Livros:

1. FACELI, K.; LORENA, A. C., GAMA, J., CARVALHO, A. C. P. L. F, Inteligência artificial: Uma abordagem de aprendizado de máquina, LTC, 2011
2. GÉRON, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2022, ISBN: 9781098125974
3. GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A., Deep Learning, MIT Press, 2016, <http://www.deeplearningbook.org>

Artigos:

LE CUN, Y.; BENGIO, Y.; HINTON, G.. Deep Learning.. Nature. , v. 521 , p. 436-444 , 0. ; Disponível em: <http://dx.doi.org/10.1038/nature14539>. Acesso em: 27 maio 2019.

Bibliografia Complementar Livros:

1. HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., , The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2ª ed., Springer, 2009
2. DUDA, R. O., HART, P. E., STORK, D. G, Pattern Classification, 2ª ed., Wiley, 2001
3. MURPHY, K. P., , Machine Learning: A Probabilistic Perspective, MIT Press, 2012
4. BURKOV, A., The Hundred-Page Machine Learning Book, 2019, ISBN-10 1777005477 Artigos: SILVER, D.et al.. Mastering the game of Go without human knowledge.. Nature. , v. 550 , p. 354-359 , 2017. ; Disponível em: <http://dx.doi.org/10.1038/nature24270>. Acesso em: 27 maio 2019.

\newpage

# Linguagens e Paradigmas

Carga Horária: 72 *Semestre : 4º período*

## Ementa

Fundamentos de linguagens formais. Linguagens regulares. Linguagens livres de contexto. Linguagens sensíveis ao contexto. Linguagens recursivamente enumeráveis. Paradigma imperativo. Paradigma estruturado. Paradigma funcional. Compiladores e interpretadores. Computabilidade. Intratabilidade.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Especificar linguagens
2. Estruturar programas nos paradigmas imperativo, estruturado e funcional
3. Implementar compiladores e interpretadores de linguagens
4. Elencar os limites da computabilidade

## Conteúdo Programático

1. Fundamentos de linguagens formais: alfabetos, palavras, linguagens e cadeias
2. Hierarquia de Chomsky
3. Linguagens regulares: propriedades, gramáticas, expressões regulares e autômatos finitos, aplicações
4. Linguagens livres de contexto: propriedades, gramáticas, autômatos à pilha, aplicações
5. Linguagens sensíveis ao contexto: propriedades, gramáticas, máquina de Turing de Fita Finita, aplicações
6. Linguagens recursivamente enumeráveis: propriedades, gramáticas, Máquina de Turing, aplicações
7. Computabilidade
8. Intratabilidade
9. Paradigmas de linguagens: imperativo, estruturado e funcional
10. Estrutura e implementação de compiladores e interpretadores: análise léxica, análise sintática, análise semântica e geração de código

## Bibliografia Básica

Livros:

1. RAMOS, M.V.M.; JOSÉ NETO, J.; VEGA, Í.S. Linguagens Formais: Teorias e Conceitos. 1. ed. Blucher, 2023. 608 p.
2. AHO, A.V.; LAM, M.S.; SETHI, R.; ULLMAN, J.D. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo: Pearson, 2008. 634 p.
3. SIPSER, M., Introdução à Teoria da Computação, 2ª ed., Thomson Pioneira, 2007. [Biblioteca Digital]

## Bibliografia Complementar

Livros:

1. SEBESTA, R.W. Concepts of Programming Languages. 11ª ed., Pearson, 2016. [Biblioteca Digital]
2. HOPCROFT, J.E.; ULLMAN, J.D.; MOTWANI, R. Introdução à teoria de autômatos, linguagens e computação. 1. ed. Rio de Janeiro: Elsevier, 2002. 560 p.
3. MICHAELSON, G. An Introduction to Functional Programming Through Lambda Calculus. Dover Publications, 2011. 336p.
4. VIEIRA DE MELO, A.C.; CORRÊA DA SILVA, F.S. Princípios de Linguagens de Programação. 1. ed. Blucher, 2003. 304 p.
5. CORMEN, T.H.; LEISERSON, C.E.; RIVEST, R.L.; STEIN, C. Algoritmos: teoria e prática. 3. ed. Rio de Janeiro: Elsevier, 2012. 926 p.

\newpage

# Sistemas Hardware-Software

Carga Horária: 72 *Semestre : 4º período*

## Ementa

Representação de informação; Representação de programas em código de máquina; Arquitetura de processadores e otimização de programas; Hierarquia de memória; Exceções e controle de fluxo em sistemas operacionais; Memória virtual; Gerenciamento de E/S; Programação concorrente.

## Objetivos

Ao final da disciplina o estudante será capaz de:

1. Explicar o ciclo de vida de um objeto executável, desde a sua geração a partir do código fonte, até a execução do código gerenciada pelo sistema operacional;
2. Explicar o papel do sistema operacional na gerência e mediação de recursos, incluindo gerência de processos, memória, e interface com o hardware;
3. Investigar, analisar e formular hipóteses sobre o desempenho de um programa de computador vis-a-vis a interação deste com os elementos fundamentais do hardware
4. Projetar e aperfeiçoar programas de computador para que usem as características do hardware visando atingir um alto desempenho;
5. Desenvolver sistemas de computador que exibem concorrência de execução e de uso de recursos; e
6. Contextualizar tecnologias atuais em relação à evolução de sistemas de computação, e discutir possibilidades futuras de evolução tecnológica nesta área.

## Conteúdo Programático

1. Introdução a programação em C
2. Arquitetura de computadores: interação entre CPU, memória e periféricos
3. Representação de dados e aritmética na CPU: inteiros de tamanho fixo, ponto flutuante, ponteiros
4. Engenharia reversa de programas em Assembly: organização de programas na memória, operações aritméticas e ponteiros, estruturas de controle de fluxo e chamada de funções
5. Gerenciamento de memória em sistemas operacionais: memória virtual, cache, alocação de memória, otimização de utilização de cache em programas;
6. Gerenciamento de processos em sistemas operacionais: exceções de hardware, sinais, agendamento, arquivos, Entrada/Saída;
7. Introdução a programação concorrente: threads vs. processos, primitivas de sincronização e controle de deadlocks, concorrência vs. paralelismo.

## Bibliografia Básica

Livros:

1. BRYANT, R. E.; O’HALLARON, D. R. , Computer systems: a programmer's perspective, 3ª ed., Pearson, 2015
2. D. A. PATTERSON; J. L. HENNESSY, Organização e Projetos de Computadores: a Interface Hardware/Software., 4ª ed., Elsevier, 2014
3. KOCHAN, S. G., Programming in C, 4ª ed., Addison-Wesley, 2014

Artigos:

KIVITY, A.; COSTA, D.; ENBERG, P.. Optimizing the Operating System for Virtual Machines.. Proceedings of USENIX ATC’14: 2014 USENIX Annual Technical Conference. , p. 61 , 2014.

## Bibliografia Complementar

Livros:

1. KERNIGHAN, B. W.; RITCHIE, D. M., The C programming language, 2ª ed., Prentice-Hall, 1988
2. J. H. SALTZER, M. F. KAASHOEK., Principles of Computer System Design: An Introduction, Morgan Kaufmann, 2009
3. STALLINGS, W., Arquitetura e organização de computadores, 8ª ed., Pearson, 2013
4. SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G., Operating System Concepts, 9ª ed., Wiley, 2013
5. TANENBAUM, A. S., Sistemas Operacionais Modernos, 4ª ed., Pearson, 2016

Artigos:

GERBER, F.; MÖSINGER, K.; FURRER, R.. DotCall64: An R package providing an efficient interface to compiled C, C++, and Fortran code supporting long vectors. SoftwareX. , n. 7 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352711018300785>. Acesso em: 27 maio 2019.

\newpage

# Projeto de Software e Gestão Ágil

Carga Horária: 72 *Semestre : 4º período*

## Ementa

Modelagem da Arquitetura de Software; Requisitos não funcionais; Arquitetura Limpa; Padrões de Projeto; Formas de documentação do projeto e boas práticas de codificação; Engenharia de Software; Padrões de Projeto; SOLID; Estratégias de testes automatizados de software: testes de interface, testes de integração, testes funcionais e testes de aceitação; Métricas estáticas de controle de qualidade. Dívida técnica. Decisão de projeto baseada em dados; Produtividade de times de software. Alinhamento dos projetos de software com as estratégias de negócios; Estimativas de custos de projetos: de mão-de-obra e arquitetura. Práticas Ágeis aplicadas ao desenvolvimento de software (Pair Programming, Code review); Papel e responsabilidade de um líder de projeto; Habilidades e características importantes para liderar equipes ágeis em projetos complexos.

## Objetivos:

Ao final da disciplina o aluno será capaz de:

1. Identificar um problema do mundo real e projetar a arquitetura da solução.
2. Criar a documentação do projeto.
3. Projetar a arquitetura do software seguindo as boas práticas de arquitetura limpa.
4. Considerar os requisitos não funcionais no desenvolvimento de software.
5. Desenvolver o código seguindo as boas práticas de codificação.
6. Entender e identificar padrões de projeto
7. Definir quais práticas ágeis são adequadas para cada necessidade de projeto
8. Aplicar as práticas ágeis em seus projetos
9. Medir e favorecer a qualidade do código
10. Identificar e gerir Dívida Técnica
11. Sistemas de CI/CD - integração e implantação contínuos
12. Definir estratégias e métricas que suportem a tomada de decisão baseada em dados
13. Utilizar data driven como forma de tomada de decisão do projeto e da equipe
14. Alinhar as decisões do time aos milestones do projeto e a estratégia de negócio
15. Liderar um time ágil em projetos complexos.

## Conteúdo Programático:

1. Padrões de Arquitetura de software
2. Boas práticas de uma arquitetura limpa
3. Boas práticas de codificação
4. Tipos de documentação de projeto
5. Práticas de Engenharia de Software
6. Práticas Ágeis
7. Padrões de Projeto
8. Dívida Técnica
9. Métricas de controle de qualidade de software
10. Características e habilidades de um líder ágil
11. Características de um projeto complexo
12. Refatoração
13. Indicadores de produtividade de um time ágil
14. Estimativas de custo de um projeto
15. Formas de alinhamento do projeto com as estratégias do negócio

## Bibliografia Básica

Livros:

HOHPE, GREGOR. The Software Architect’s Elevator.

MARTIN, C. ROBERT. Clean Architecture: A Craftsman’s Guide to Software Structure and Design. 1.a ed. 2017.

NEWMAN, SAM. Criando Microsserviços: Projetando Sistemas com Componentes Menores e Mais Especializados. 1.a ed. São Paulo: Novatec Editora, 2022.

Artigos:

S. Silva, A. Tuyishime, T. Santilli, P. Pelliccione and L. Iovino, "Quality Metrics in Software Architecture," 2023 IEEE 20th International Conference on Software Architecture (ICSA), L'Aquila, Italy, 2023, pp. 58-69

## Bibliografia Complementar

RICHARDS, MARK; FORD, NEAL. Fundamentals of Software Architecture: An Engineering Approach. 1.a ed. 2020.

VALENTE, M. T. Engenharia de Software Moderna: Principios e Práticas para Desenvolvimento de Software com Produtividade. 1.a ed. Belo Horizonte: Publicação Independente, 2020.

FOWLER, MARTIN. Refactoring: Improving the Design of Existing Code. 2.a ed. 2018.

KRUCHTEN, PHILIPPE; IPEK OZKAYA. Managing Technical Debt: Reducing Friction in Software Development. 1.a ed. 2019.

RICHARDSON, CHRIS. Microservices Patterns: With Examples in Java. 1.a ed. 2018.

Artigos:

L. Leite, C. Rocha, F. Kon., D. Milojicic & P. Meirelles, "A survey of DevOps concepts and challenges". ACM Computing Surveys (CSUR), 52(6), 1-35. 2019

\newpage

# Análise de Algoritmos e Entrevistas Técnicas

Carga Horária: 72 *Semestre 4º período*

## Ementa

Descrição formal de algoritmos. Notação assintótica. Análise de algoritmos determinísticos. Análise de algoritmos probabilísticos. Análise de programação dinâmica. Análise de algoritmos gulosos. Análise de complexidades de problemas. Classes de complexidade. Algoritmos aproximativos. Algoritmos baseados em oráculos. Maratonas de programação. Entrevistas técnicas.

## Objetivos

Ao final do curso, o aluno deve ser capaz de:

1. Identificar e demonstrar a complexidade de espaço e tempo de um algoritmos utilizando notação assintótica;
2. Projetar e implementar algoritmos determinísticos, probabilísticos, dinâmicos e gulosos;
3. Descrever as intuições e funcionamento de algoritmos no contexto de entrevistas técnicas de programação.

## Conteúdo Programático

1. Notação assintótica: crescimento de funções, ordens de complexidade de algoritmos, pior caso, melhor caso e caso médio
2. Demonstrações de propriedades de notação assintótica
3. Análise de algoritmos determinísticos: algoritmos iterativos e recursivos, análise de algoritmos iterativos e recursivos clássicos para ordenação e busca
4. Análise de programação dinâmica: programação dinâmica, algoritmos dinâmicos clássicos, análise de algoritmos dinâmicos
5. Análise de algoritmos gulosos
6. Análise de algoritmos probabilísticos: algoritmos probabilísticos, distribuições para entradas, análise probabilística de algoritmos determinísticos, análise probabilística de algoritmos probabilísticos
7. Entrevistas técnicas: estrutura, entrevistas nacionais e internacionais, prática em entrevistas técnicas

## Bibliografia Básica

1. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L; STEIN, C. Algoritmos: Teoria e Prática, 3a ed. Elsevier, 2012.
2. MCDOWELL, G. L., Cracking the coding interview, 6ª ed., CareerCup, 2015.
3. HALIM, S.; HALIM, F., Competitive programming: the new lower bound of programming contests, 3ª ed., 2013.

## Bibliografia Complementar

1. BENTLEY, J. Programming Pearls, 2nd ed. Addison-Wesley, 1999.
2. SKIENA, S. S. The Algorithm Design Manual, 2nd ed. Springer, 2008.
3. SEDGEWICK, R.; WAYNE, K. Algorithms, 4th ed. Addison-Wesley, 2011.
4. ERICKSON, J. Algorithms, 1st ed. 2019.
5. AZIZ, A. et al., Elements of programming interviews in Python: the insiders' guide, Createspace, 2016.

\newpage

# Sprint Session 4

Carga horária: 44h. Semestre período

## Ementa:

Elaboração de soluções que fazem uso de técnicas de aprendizagem de máquina supervisionada ou não supervisionada; Desenvolvimento de soluções de aprendizagem de máquina para problemas de classificação, regressão, clusterização, ou recomendação; Prototipação de software; Comunicação oral e escrita.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Interpretar requisitos de um cliente relacionadas a um software centrado em dados.
2. Propor uma solução que faz uso de modelos preditivos ou descritivos.
3. Executar o ciclo completo do CRISP-DM para desenvolvimento de modelos preditivos: aquisição de dados; pré-processamento de dados; modelagem; avaliação; entrega, e; acompanhamento do modelo e, eventual, atualização.
4. Executar projeto em uma ambiente distribuído de desenvolvimento de software.
5. Apresentar o projeto e criar documentação escrita.

## Conteúdo programático

1. Coleta de requisitos de cliente.
2. Criação de modelos preditivos ou descritivos usando CRISP-DM.
3. Desenvolvimento da solução e entrega do modelo como um serviço.
4. Comunicação escrita.
5. Técnicas de apresentação.

## Bibliografia básica

1. S. Amershi et al., "Software Engineering for Machine Learning: A Case Study," 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 2019, pp. 291-300, doi: 10.1109/ICSE-SEIP.2019.00042.
2. FACELI, K.; LORENA, A. C., GAMA, J., CARVALHO, A. C. P. L. F, Inteligência artificial: Uma abordagem de aprendizado de máquina, LTC, 2011
3. GÉRON, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2022, ISBN: 9781098125974

## Bibliografia complementar

1. Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2022. Challenges in Deploying Machine Learning: A Survey of Case Studies. ACM Comput. Surv. 55, 6, Article 114 (June 2023), 29 pages. <https://doi.org/10.1145/3533378>.
2. Pedro Domingos. 2012. A few useful things to know about machine learning. Commun. ACM 55, 10 (October 2012), 78–87. <https://doi.org/10.1145/2347736.2347755>

\newpage

# Megadados

Carga Horária: 80 *Semestre 5º período*

## Ementa

Banco de Dados; Modelagem e projeto de banco de dados; Modelo e diagrama Entidade-Relacionamento; Modelo e álgebra relacional; Transformação de modelos lógicos em tabelas; Dependências Funcionais. Normalização; Banco de dados relacional e orientado a objetos: Mapeamento objeto-relacional; Linguagens de consulta e manipulação de dados; Sistemas de Gerência de Banco de Dados: arquitetura, gerenciamento de transações, controle de concorrência, recuperação; Processamento e otimização de consultas; Bancos de Dados Distribuídos; Domínios de aplicação de megadados (big data) e desafios de sua aplicação; Integração, tratamento e consolidação de dados; Bancos de dados não relacionais; Armazenamento e processamento de dados distribuído - algoritmos e plataformas; Recuperação de informações.

## Objetivos

Ao final da disciplina o estudante será capaz de:

1. Entender o que são megadados e quais os desafios inerentes a dados com esta escala, complexidade, e requisitos de performance
2. Dado um problema, estabelecer uma estratégia de trabalho com megadados (integração, armazenamento, processamento, tomada de decisões)
3. Projetar software analítico capaz de utilizar estratégias de computação distribuída para tratar de forma eficaz grandes volumes de dados
4. Aplicar técnicas de recuperação de informação e mineração de dados.
5. Descobrir e avaliar criticamente, de forma autônoma, tecnologias emergentes em big data.

## Conteúdo Programático

1. Modelagem relacional, diagrama M-E-R (Modelo Entidade-Relacionamento)
2. Bancos de dados relacionais
3. SQL (Structured Query Language): DDL, DML, DQL, DCL
4. Schemas, domínios, views, tabelas temporárias
5. Segurança, transações, locks, ACID (Atomicidade, Consistência, Isolamento e Durabilidade)
6. Stored procedures, constraints e triggers, cursors
7. Tópicos em teoria de banco de dados: normalização, álgebra relacional
8. Arquitetura de soluções
9. ETL (Extract, Transform and Load), expressões regulares e processamento de texto.
10. Introdução a sistemas de processamento de dados em larga escala.
11. Introdução à recuperação de informação e máquinas de busca.
12. Bancos de dados não-relacionais: orientados a objeto, NoSQL
13. MapReduce e Spark
14. Indexação e busca

## Bibliografia Básica

Livros:

1. SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S, Sistema de Banco de Dados, 6ª ed., Campus Elsevier, 2012
2. OLIVEIRA, C. H. P, SQL: Curso Prático, Novatec, 2002
3. RIBEIRO-NETO, B.; BAEZA-YATES, R. , Recuperação de Informação - Conceitos e Tecnologia Das Máquinas de Busca, 2ª ed., Bookman, 2013

Artigos:

ASSUNÇÃO, M. D. et al.. Big data computing and clouds: trends and future directions.. J. Parallel and Distributed Computing. , p. 3-15 , 2015.

## Bibliografia Complementar

Livros:

1. MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. , Introduction to Information Retrieval, 1ª ed., Cambridge University Press, 2008
2. GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. , Database Systems: The Complete Book, 2ª ed., Pearson, 2008
3. WAMPLER, D.; PAYNE, A., Programming Scala: Scalability = Functional Programming + Objects, 2ª ed., O'Reilly Media, 2014
4. McCALLUM, ETHAN, Bad Data Handbook: Cleaning Up The Data So You Can Get Back To Work, 1ª ed., O'Reilly Media, 2012
5. RYZA, S.; LASERSON, U.; OWEN, S.; WILLS, J. , Advanced Analytics with Spark, O'Reilly Media, 2015

Artigos:

ZAHARIA, M. et al.. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.. Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. , 2012.

\newpage

# AI Startup

Carga Horária: 80 *Semestre 5º período*

## Ementa

O que é empreendedorismo; Impactos atuais e futuros da Inteligência Artificial; Desenvolvimento de startups na visão dos investidores; Desenvolvimento de Lean canvas; Apresentação do *pitch Problem-Solution-Opportunity*, e; Desenvolvimento de um MVP (Produto Viável Mínimo) utilizando boas práticas de desenvolvimento (i.e., desenvolvimento de código de forma colaborativa, testes automatizados, padronização de código e documentação).

## Objetivos

Ao final da disciplina o estudante será capaz de:

1. compreender competências empreendedoras e técnicas essenciais para a criação e gerenciamento bem-sucedido de startups;
2. compreender as nuances do empreendedorismo (como criar um negócio, defender uma ideia, captar recursos);
3. entender como criar e manter uma equipe;
4. implementar uma solução usando conceitos de Inteligência Artificial, e;
5. desenvolver e manter uma solução de software usando boas práticas de desenvolvimento de software.

## Conteúdo Programático

1. O que é empreendedorismo.
2. Impactos atuais e futuros da Inteligência Artificial (IA).
3. Exemplos de soluções que fazem uso de IA.
4. Desenvolvimento de startups na visão dos investidores.
5. Contato com investidores e com corporações com desemandas de IA.
6. Desenvolvimento de Lean canvas.
7. Apresentação do *pitch Problem-Solution-Opportunity*.
8. Desenvolvimento de um MVP (Produto Viável Mínimo) utilizando boas práticas de desenvolvimento.
9. Definição de infra-estrutura e processos para desenvolvimento de código de forma colaborativa.
10. Padronização de código e documentação.
11. Desenvolvimento de um MVP (Produto Viável Mínimo) utilizando boas práticas de desenvolvimento.
12. Definição de infra-estrutura e processos para desenvolvimento de código de forma colaborativa.
13. Implementação e utilização de testes automatizados.
14. Padronização de código e documentação.
15. Desenvolvimento da solução em sprints.
16. Apresentação da solução.

## Bibliografia Básica

1. Cooper, B. and Vlaskovits, P., 2010. The entrepreneur's guide to customer development: a" cheat sheet" to The Four Steps to the Epiphany (pp. 90-110). Newport Beach, CA: Cooper-Vlaskovits.
2. Blank, S. and Dorf, B., 2020. The startup owner's manual: The step-by-step guide for building a great company. John Wiley & Sons.

## Bibliografia Complementar

1. Horowitz, B. and Kenerly, K., 2014. The hard thing about hard things (p. 174). New York: HarperCollins.
2. Charlie Giattino, Edouard Mathieu, Veronika Samborska and Max Roser (2023) - "Artificial Intelligence". Published online at OurWorldInData.org. Disponível em [<https://ourworldindata.org/artificial-intelligence>](https://ourworldindata.org/artificial-intelligence). Último acesso em junho de 2024.
3. RIES, E., A Startup Enxuta: Como os Empreendedores Atuais Utilizam a Inovação Contínua para Criar Empresas Extremamente Bem-sucedidas., Leya, 2012
4. Andrew Ng. Opportunities in AI - 2023. Palestra ministrada em Stanford. Disponível em [<https://www.youtube.com/watch?v=5p248yoa3oE>](https://www.youtube.com/watch?v=5p248yoa3oE). Último acesso em junho de 2024.
5. Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2022. Challenges in Deploying Machine Learning: A Survey of Case Studies. ACM Comput. Surv. 55, 6, Article 114 (June 2023), 29 pages. <https://doi.org/10.1145/3533378>.

\newpage

# Jogos e Interação

Carga Horária: 80 *Semestre 5º período*

## Ementa

Tipologia de jogos; Processos de produção de jogos digitais; Game Design; Level Design; Técnicas ágeis de design de jogos; Design de interação para jogos; Tecnologias para desenvolvimento de jogos (GPU, bibliotecas e frameworks gráficos, engines); Produção de artefatos para jogos (cenários, modelos, animações, áudio, vídeo, scripts); Técnicas ágeis para prototipação de jogos; Integração de artefatos de jogos; Game Testing; Realismo físico em jogos; Processo de publishing e marketing de jogos; Negócios e empreendedorismo em jogos.

## Objetivos

1. Desenvolver em uma linguagem e ferramentas apropriadas projeto e implementação de um jogo profissional completo.
2. Projetar e gerenciar uma pipeline integrada de conteúdo para a produção de um jogo que integre programas e código a artefatos diversos (áudio, imagens, modelos 3D).
3. Identificar num projeto de jogo o papel de profissionais encarregados da produção de conteúdo artístico e comunicar-se efetivamente com eles.
4. Desenvolver sistemas de jogos de computador, usando rotinas de Inteligência Artificial e Computação Gráfica que realizam animações realistas em tempo real e executadas em um ambiente de rede.
5. Participar efetivamente de uma equipe para desenvolver software gráfico de jogo.
6. Construir um portfólio, com um currículo, carta de apresentação e elevator pitch.

## Conteúdo Programático

1. Histórico e evolução dos jogos e plataformas para jogos.
2. Frameworks para estudo e avaliação de jogos por tipologia.
3. Processos de projeto clássicos e ágeis para jogos digitais, com ênfase em produção de documentos de Game Design (GDD) e Level Design (LDD).
4. Tecnologias para produção e integração de artefatos para jogos.
5. Produção de artefatos para jogos (cenários, modelos, animações, áudio, vídeo, scripts).
6. Projeto de interação e análise de usabilidade em jogos.
7. Integração de artefatos e técnicas ágeis para produção de protótipos.
8. Planos de testes para jogos, processos de Game Testing e produção de relatórios de testes; instrumentação de jogos para testes.
9. Realismo gráfico e físico em jogos via GPU (shaders e coprocessadores de Física)
10. Inteligência Artificial aplicada a jogos: busca de caminhos mínimos, comportamentos inteligentes de bots, grupos e aprendizado.
11. Suporte de rede a jogos: protocolos de comunicação, detecção e compensação de lag, jogos multiplayer, online e multiplayer massivos online (MMO).
12. Compilação e implantação de jogos em consoles.
13. Aspectos de negócios em jogos: portfólios, processo de publishing, marketing e empresas indie de jogos.

## Bibliografia Básica

Livros:

1. GIBSON, J., Introduction to Game Design, Prototyping and Development, Addison-Wesley, 2015
2. Rogers, S.; Level Up! The Guide to Great Video Game Design, 2nd Edition, Wiley, 2014

## Bibliografia Complementar

Livros:

1. SILVA, G. F.; SILVA, T. S. C.; FILHO, M. P. et al. Game Design, SAGAH, 2021. E-book. ISBN 9786556901299.

\newpage

# Plataformas, Microsserviços e APIs

Carga Horária: 80

*Semestre: 5º período*

## Ementa

Conceitos de Aplicações em Monólíto e Microserviços; Conteinerização; Padrões de Construção de Softwares (Design Patterns); Monitoramento e Rastreabilidade de Aplicações; Segurança (autenticação e autorização); Mensageria; Dados Distribuídos; Orquestração de computação em nuvem, sistemas de gerenciamento, monitoramento e configuração de recursos virtualizados; Integração entre Desenvolvimento e Operação; Utilização de Plataformas em Nuvem para Produção (Cloud Computing); Aspectos de automação de gestão de sistema em cloud – DevOps. Serverless computing – FaaS - function as a service; Utilização da Plataforma como Produto para Negócios: Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS). Gestão de níveis de serviço (SLA - Service Level Agreement). Custos de projeto e de operação de sistemas em cloud.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Tomar decisões a respeito da escolha de estratégias de arquiteturas para o emprego de problemas computacionais;
2. Implementar e interconectar aplicações computacionais para a construção de plataformas de alto desempenho: escalabilidade por meio do uso de técnicas de computação em nuvem;
3. Administrar um sistema de gerenciamento de nuvem, provisionando a infraestrutura necessária como um serviço;
4. Construir, com o auxílio de frameworks, soluções de plataformas completas e integradas de forma profissional;
5. Arquitetar e implementar linhas de produções de softwares robustos;
6. Analisar, projetar e especificar uma solução de computação em nuvem mista baseada em hardware, software e redes para atender aos requisitos de determinado pacto de nível de serviço (SLA);
7. Planejar e analisar o uso de plataformas empresariais como subsídio para criação de novos negócios.

## Conteúdo Programático

1. Conceitos de Arquitetura e Microsserviços;
2. Microsserviços com Interface API - RESTful;
3. Introdução a Contêineres;
4. Introdução e Implementação de Design Patterns;
5. Apresentação de Design Patterns mais Complexos: Segurança, Mensageria, Cache, etc;
6. Fundamentos de Computação em Nuvem.
7. Orquestração, Implementação e Monitoramento de Ambientes Virtualizados e Distribuídos;
8. Infraestrutura como um Serviço.
9. Redes Definidas por Software;
10. Software como um Serviço;
11. Gestão de Níveis de Serviço.

## Bibliografia Básica

Livros:

1. FREEMAN, Emily. [DevOps Para Leigos. Rio de Janeiro: Editora Alta Books, 2021](https://www.amazon.com.br/DevOps-para-leigos-primeiros-sucesso/dp/855081573X/). E-book. ISBN 9788550816661. Disponível em: [<https://integrada.minhabiblioteca.com.br/#/books/9788550816661>](https://integrada.minhabiblioteca.com.br/#/books/9788550816661). Acesso em: 30 de maio de 2023.
2. VERONEZ, Fabricio. [Canal Kubernetes e DevOps](https://www.youtube.com/@fabricioveronez). Disponível em: [<https://www.youtube.com/@fabricioveronez>](https://www.youtube.com/@fabricioveronez). Acesso em: 24 de junho de 2024.

## Bibliografia Complementar

Livros:

1. XU, A., [System Design Interview - An insider's guide](https://www.amazon.com.br/System-Design-Interview-Insiders-English-ebook/dp/B08B3FWYBX/), 1ª ed., Independently Published, 2020.
2. MARTIN, R. C., [Arquitetura Limpa: o guia do artesão para estrutura e design de software](https://www.amazon.com.br/Arquitetura-Limpa-artes%C3%A3o-estrutura-software-ebook/dp/B085PP6Y8P/), 1ª ed., Alta Books, 2018.
3. PARKER, G. G.; VAN ALSTYNE, M. W.; CHOUDARY, S. P., [Plataforma: a revolução da estratégia](https://www.amazon.com.br/Plataforma-Revolu%C3%A7%C3%A3o-Estrat%C3%A9gia-Geoffrey-Parker/dp/8550806137/), 1ª ed., Alta Books, 2018.
4. SEHGAL, N. K.; BHATT, P. C. P.; ACKEN J. M., [Cloud Computing with Security and Scalability.: Concepts and Practices](https://www.amazon.com.br/Cloud-Computing-Security-Scalability-Practices/dp/3031072448/), 3ª ed., Springer, 2023.
5. KRIEF, M., [Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins](https://www.amazon.com/Learning-DevOps-comprehensive-accelerating-Kubernetes/dp/1801818967/), 2ª ed., Packt Publishing, 2022.

\newpage

# Cybersegurança para Ciência da Computação

Carga Horária: 80 *Semestre : 6º período*

## Ementa:

Princípios fundamentais da cibersegurança; Ameaças cibernéticas atuais; Técnicas de proteção de dados; Políticas de segurança; Conformidade regulatória; Disciplina de segurança contínua; Análise de riscos cibernéticos; Implicações da revolução digital na área de segurança para organizações e sociedade.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Implementar as melhores práticas de segurança em suas organizações, garantindo a integridade, confidencialidade e disponibilidade de informações críticas.
2. identificar, avaliar e mitigar os riscos cibernéticos, utilizando métodos e ferramentas apropriados.
3. Identificar possíveis ameaças, vulnerabilidades e riscos associados a ativos de informação
4. Identificar potenciais conflitos entre aplicações de informática e considerações legais ou éticas, e exercitar o julgamento profissional para resolver esses tipos de conflitos

## Conteúdo Programático:

1. Introdução: Princípios básicos da Segurança da Informação, Vulnerabilidades, Ameaças, Riscos, Ataques e Segurança.
2. Análise de tráfego de rede: Protocolos TCP/IP, ferramentas de análise e protocolos de segurança.
3. Compreensão dos conceitos de criptografia computacional.
4. Discussão sobre responsabilidade, privacidade e boas práticas éticas na área da Tecnologia da Informação e Comunicação.
5. Exploração dos aspectos éticos relacionados à segurança da informação e à cibersegurança.
6. Estudo dos serviços de segurança, como firewalls, Zero Trust, IDS/IPS, VPNs, e sua implementação em redes.
7. Compreensão dos conceitos de engenharia social e sua relevância na segurança da informação.
8. Coleta de informações e reconhecimento do ambiente tecnológico.
9. Ameaças cibernéticas e exploração de aplicações web.

## Bibliografia básica

1. Basta, Alfred, et al. Segurança de Computadores e teste de invasão - Tradução da 2ª edição norte-americana. Disponível em: Minha Biblioteca, Cengage Learning Brasil, 2014.
2. Stallings, William. Cryptography and network security: principles and practice / William Stallings. 7th ed. New York, NY : Pearson, c2017.
3. Abdullayeva, F. Cyber resilience and cyber security issues of intelligent cloud computing systems. Results Control Optim. 2023, 12, 100268.

## Bibliografia complementar

1. Pinheiro, Patricia P. Segurança Digital. Proteção de Dados nas Empresas. Disponível em: Minha Biblioteca, Grupo GEN, 2020.
2. Comer, Douglas E. Redes de computadores e internet. Disponível em: Minha Biblioteca, (6th edição). Grupo A, 2016.
3. Agra, Andressa, D. e Fabrício F. M. Barboza. Segurança de sistemas da informação. Disponível em: Minha Biblioteca, Grupo A, 2019.
4. Branquinho, Thiago, e Marcelo Branquinho. Segurança Cibernética Industrial. Disponível em: Minha Biblioteca, Editora Alta Books, 2021.
5. Moraes, Alexandre Fernandes D. Cibersegurança e a nova geração de Firewalls. Disponível em: Minha Biblioteca, Editora Saraiva, 2021.

# Projeto Final de Computação

Carga Horária: 160

*Semestre 6º período*

## Ementa

Condução de projetos em Computação. Desenvolvimento de planejamento, prototipação, validação, testes e documentação de projtos. Análise de metodologias de trabalho para projetos em Computação. Comunicação efetiva e assertiva. Identificação e viabilização de papeis e responsabilidades em equipes. Identificação de necessidades e expectativas de partes interessadas em projetos de engenharia. Detecção e mitigação de riscos. Negociação em projetos de engenharia. Análise de viabilidade técnica e econômica.

## Objetivos

1. Execução Técnica: Ser capaz de projetar, prototipar, desenvolver, validar, testar e documentar uma solução real de computação.
2. Organização: Escolher, seguir, adaptar e julgar uma metodologia de trabalho adequada ao projeto.
3. Comunicação: Comunicar efetivamente e assertivamente com as partes interessadas, mantendo informações e expectativas atualizadas em relação aos objetivos e andamento do projeto.
4. Trabalho em Equipe: Identificar e viabilizar os papeis e responsabilidades de todos os membros da equipe, garantindo o engajamento dos colegas de projeto.
5. Design/Empreendedorismo: Identificar as necessidades e expectativas das partes interessadas, tratando potenciais riscos e negociações necessárias, analisando sua viabilidade técnica e econômica.

## Conteúdo Programático

1. Metodologias de projetos
2. Cultura organizacional
3. Gestão de equipes
4. Co-design de proposta de projeto com empresas clientes
5. Planejamento e prototipação
6. Elaboração de relatório técnico-científico
7. Apresentação de resultados da etapa a stakeholders e avaliadores

## Bibliografia Básica

Livros:

1. BASTOS, C. L. Aprendendo a Aprender: Introdução à Metodologia Cientifica. 24. ed. Petrópolis: Vozes 2012
2. ULRICH, K. T., EPPINGER, S. D. Product Design and Development, McGraw-Hill Education, 5 edição, ISBN: 78-0073404776, 2011
3. RIES, M., SUMMERS, D. Agile Project Management: A Complete Beginner's Guide to Agile Project Management, CreateSpace Independent Publishing Platform, ISBN-13: 978-1539877301, 2016

Artigo:

HUNDHAUSEN, C. Special Issue on Capstone Projects. ACM Transactions on Computing Education. v. 18, n. 2, 2018. Disponível em < <https://dl.acm.org/citation.cfm?id=3239167> >. Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. KERZNER, H., Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Wiley, 12a edição, ISBN: 978-1119165354, 2017
2. MCCAHAN, S., ANDERSON, P., KORTSCHOT, M., WEISS, P. WOODHOUSE, K. Projetos de Engenharia. Uma Introdução. LTC, 1ªEdição, ISBN: 978-8521634454, 2017.
3. COOK, C., The Entrepreneurial Project Manager. Auerbach Publications, ISBN: 978-1498782357, 2017.
4. SROUR, R. H., Ética Empresarial, Elsevier, ISBN:9788535264487, 2013.
5. SILVA FILHO, C. F.; BENEDICTO, G. C.; CALIL, J. F., Ética, Responsabilidade Social e Governança Corporativa. 3. ed. Campinas: Alínea, 2014.

Artigo:

FRANCHETTI, M., ARISS, S. S., The Implementation of Senior Design Capstone Projects Combining Engineering and Business Students. Journal of STEM Education, v. 17, n. 4, 2016. Disponível em < <https://www.jstem.org/index.php/JSTEM/article/view/2163> >. Acesso em: 27 maio 2019.