



INSPER INSTITUTO DE ENSINO E PESQUISA

EMENTÁRIO

# CIÊNCIA DA COMPUTAÇÃO

Outubro de 2021

1	Developer Life				Ética, Computação e Sociedade
	Sessão Sprint 1				
2	NAND to Tetris: Bits e Processadores	Ciência dos Dados	Matemática da Variação	Projeto Ágil e Programação Eficaz	
	Sessão Sprint 2				
3	Técnicas de Programação	Inteligência Artificial e Robótica	Álgebra Linear e Teoria da Informação	Lógica e Matemática Discreta	Programação Web e Times Ágeis
	Sessão Sprint 3				
4	Algoritmos e Estruturas de Dados	Machine Learning	Teoria da Computação e Linguagens Formais	Sistemas Hardware- Software	Projeto de Software e Gestão de Projetos Ágeis
	Sessão Sprint 4				
5	Análise de Algoritmos e Entrevistas Técnicas	Megadados	AI Startup	Jogos e Interação	Plataformas, Microserviços e APIs
6	Análise de Redes Sociais	Cybersegurança	Projeto Final de Computação		
7	Eletiva	Eletiva	Eletiva	Estágio Supervisionado 1	
8	Eletiva	Eletiva	Eletiva	Estágio Supervisionado 2	

Figura 1 - Estrutura curricular ao longo dos semestres

- Experiência como Desenvolvedor - Developer Life
  - Ementa
  - Objetivos
  - Conteúdo programático
  - Bibliografia básica
  - Bibliografia complementar
- Ética, Computação e Sociedade
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Nand to Tetris: Elementos de Sistemas Digitais
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Ciência dos Dados
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
- Matemática da Variação
  - Ementa
  - Objetivos
  - Conteúdo Programático:
  - Bibliografia Básica
  - Bibliografia Complementar
- Projeto Ágil e Programação Eficaz
  - Ementa:
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Técnicas de Programação
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Inteligência Artificial e Robótica
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica

- Bibliografia Complementar
- Álgebra Linear e Teoria da Informação
  - Ementa
  - Objetivos
- Lógica e Matemática Discreta
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Programação Web e Times Ágeis
  - Ementa
  - Objetivos
  - Conteúdo Programático:
  - Bibliografia Básica
  - Bibliografia Complementar
- Algoritmos e Estruturas de Dados
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Machine Learning
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
- Teoria da Computação e Linguagens Formais
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Sistemas Hardware-Software
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Projeto de Software e Gestão de Projetos Ágeis
  - Ementa
  - Objetivos:
  - Conteúdo Programático:
  - Bibliografia Básica
  - Bibliografia Complementar
- Análise de Algoritmos e Entrevistas Técnicas

- Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Megadados
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- AI Startup
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Jogos e Interação
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar
- Plataformas, Microsserviços e APIs
  - Ementa:
  - Objetivos
  - Conteúdo Programático:
  - Bibliografia Básica
  - Bibliografia Complementar
- Projeto Final de Computação
  - Ementa
  - Objetivos
  - Conteúdo Programático
  - Bibliografia Básica
  - Bibliografia Complementar

# Experiência como Desenvolvedor - 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

Projeto de Software:

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 Sistemas Complexos:

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. Compreender como assuntos da fronteira da computação podem ser usados para resolução de um problema real;    ## Conteúdo programático

#### Projeto de Software:

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 Sistemas Complexos:

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

### Projeto de Software:

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. PIVA JR, D.; Engelbrecht, A. M.; Nakamiti, G. S.; Bianchi, F, Algoritmos e Programação de Computadores, Elsevier-Campus , 2012

### Artigo:

MIKAMI, K. et al.. Effectiveness of Game Jam-based iterative program for game production in Japan.. Computers & Graphics. , v. 61 , p. 1-10 , 2016. ; Disponível em: <http://www.sciencedirect.com/science/article/pii/S0097849316300863>. Acesso em: 27 maio 2019.

### Desenvolvimento colaborativo:

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

### Artigos:

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:

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

### Artigos:

BRADE, J. et al.. Being there again – presence in real and virtual environments and its relation to usability and user experience using a mobile navigation task.. International Journal of Human-Computer Studies. , v. 101 , p. 76-87 , 2017. ; Disponível em: <https://doi.org/10.1016/j.ijhcs.2017.01.004>. Acesso em: 27 maio 2019.

### Projeto de Sistemas Complexos:

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

### Projeto de Software:

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

### Desenvolvimento colaborativo:

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 Sistemas Complexos:

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

# Ética, Computação e Sociedade

Carga Horária: 80 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;
3. Analisar e avaliar os usos sociais da tecnologia à luz de temas contemporâneos.

## Conteúdo Programático

1. O que é ciência e método científico;
2. Sociologia e campo científico;
3. O que é tecnologia e sua construção social;
4. Ética, ciência e tecnologia;
5. Educação em direitos humanos
6. Relações étnico-raciais e elementos de história e cultura afro-brasileira, africana e indígena
7. Economia e inovação tecnológica;
8. Escolhas tecnológicas e seus usos sociais.

## Bibliografia Básica

Livros:

1. ALVES, Rubem, Filosofia da Ciência. Introdução ao jogo e as suas regras, Loyola, 2007
2. COLLINS, Harry and PINCH, Trevor, O Golem: Tudo que você queria deveria saber sobre Ciência, UNESP, 2003
3. ARAUJO, Marcelo de. Novas Tecnologias e Dilemas Morais. Independently Published, 2019.

4. MOROZOV, Evgeny. Big Tech: A ascensão dos dados e a morte da política. São Paulo: Ubu Editora, 2018.
5. 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/simpósio/anais/Anais-Hipertexto-2010/Alda-Cristina-Costa.pdf>. Acesso em: 27 maio 2019.
2. FERNANDES, João Carlos Lopes; SOUZA Mônica Maria Martins de; OLIVEIRA, Daniel de. A INCLUSÃO DIGITAL DO NEGRO NO BRASIL. Anais do IV Seminário Internacional de Integração Étnico-Racial e as Metas do Milênio in: Brasil para todos Revista Internacional sobre questões étnico-raciais.. , v. 3 , n. 1 , 2016. ; Disponível em: [https://ojs.eniac.com.br/index.php/Anais\\_Sem\\_Int\\_Etn\\_Racial/article/view/364/452](https://ojs.eniac.com.br/index.php/Anais_Sem_Int_Etn_Racial/article/view/364/452). Acesso em: 27 maio 2019.
3. GOULART, Guilherme Damásio. O impacto das novas tecnologias nos direitos humanos e fundamentais: o acesso à internet e a Liberdade de expressão. Revista Direitos Emergentes na Sociedade Global (RedesSG). , v. 1 , n. 1 , 2012. ; Disponível em: [https://periodicos.ufsm.br/REDESG/article/view/5955/pdf\\_1#.W3G7qtJKiUk](https://periodicos.ufsm.br/REDESG/article/view/5955/pdf_1#.W3G7qtJKiUk). Acesso em: 27 maio 2019.

### Bibliografia Complementar

#### Livros:

1. BRUNO, Fernanda; CARDOSO, Bruno; [et al] orgs. Tecnopolíticas da vigilância. 1ª edição – São Paulo: Boitempo, 2018. [Facilmente encontrável, mas foi disponibilizado on-line gratuitamente pelos autores em Creative Commons: <https://lavits.org/livro-tecnopoliticas-da-vigilancia-da-lavits-disponivel-para-download-livre/?lang=pt> ] Último acesso em 28.09.21
2. HARARI, Yuval N. 21 Lições para o século XXI. São Paulo: 1ª edição. Cia das Letras, 2018. [Facilmente encontrável; best seller]
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. [Facilmente encontrável]
4. Pretto, Nelson De Luca; SILVEIRA, Sergio Amadeu da. Além das redes de colaboração: internet, diversidade cultural e tecnologias do poder. Salvador: EDUFBA, 2008. Facilmente encontrável, mas foi disponibilizado on-line gratuitamente pelos au-

tores e pela Universidade Federal da Bahia em Creative Commons:  
<https://repositorio.ufba.br/ri/bitstream/ufba/211/4/Alem%20das%20redes%20de%20colaboracao.pdf>  
Último acesso em 28.09.21

5. ZUBOFF, Shoshana. A era do capitalismo de vigilância: A luta por um futuro humano na nova fronteira do poder. 1ª Edição: Rio de Janeiro. Editora Intrínseca, 2020. [Facilmente encontrável]
6. BOURDIEU, Pierre, Os Usos Sociais da Ciência, UNESP, 2004
7. LATOUR, Bruno, Vida de Laboratório. A Produção dos Fatos Científicos, Relume Dumará, 1997
8. SCHUMPETER, J. , Teoria do Desenvolvimento Econômico, Abril Cultural, 1983

Artigos:

HOGAN, M.; SHEPHERD, T. Information Ownership and Materiality in an Age of Big Data Surveillance. Journal of Information Policy. , v. 5 , p. 6 - 31 , 2015. ; Disponível em: <https://www.jstor.org/stable/10.5325/jinfopoli.5.2015.0006>. Acesso em: 27 maio 2019.

# Nand to Tetris: Elementos de Sistemas Digitais

Carga Horária: 80 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; Latches e Flip-Flops; Minimização de Estado; Registradores; Registradores de Deslocamento; Dispositivos Lógicos Programáveis; Memória; Portas lógicas; Circuitos de temporização e pulsos; Álgebra Booleana; Circuitos Aritméticos e Aritmética Binária; Eletrônica digital; Sistemas operacionais; Lógica; Compiladores; Analisadores Léxico, Sintático e Semântico; Gerenciamento de Memória; Gerenciamento de Dispositivos de Entrada/Saída; Organização e Arquitetura básica de um sistema computacional; Conjunto de Instruções; Mecanismos de Interrupção e Exceção.

## 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. Compreender 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. Unidade Lógica Aritmética
3. Lógica Sequencial
4. Lógica de Controle
5. Linguagem de Máquinas
6. Assembly
7. Assembler
8. Máquina Virtual
9. Controle de programa
10. Linguagem Orientada a Objetos

11. Análise Léxica e Sintática
12. Linguagens

## **Bibliografia Básica**

Livros:

1. NISAN, Noam; SCHOCKEN, Shimon., The Elements of Computing Systems, MIT Press, 2005
2. TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L., Sistemas Digitais: Princípios e Aplicações, 11<sup>a</sup> ed., Pearson, 2011
3. PATTERSON, D.; HENNESSY, J. L., Organização e Projeto de Computadores: A Interface Hardware/Software., 4<sup>a</sup> ed., Campus, 2014

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. FLOYD, T. L., Digital Fundamentals, 11<sup>a</sup> ed., Pearson, 2015
2. MOORE, Cristopher ; MERTENS, Stephan., The Nature of Computation, University Press, 2011 STALLINGS, W., Arquitetura e organização de computadores, 8<sup>a</sup> ed., Prentice Hall Brasil, 2010
3. 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.

# Ciência dos Dados

Carga Horária: 80 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; Introdução à classificação; Regressão linear simples e múltipla.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Elaborar análises exploratórias de dados (univariadas e multivariadas), utilizando ferramentas estatísticas e computacionais adequadas.
2. Selecionar informações de bancos de dados, tratá-los e prepará-los para processamento.
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.
5. Analisar relações entre as variáveis, utilizando ferramentas estatísticas inferenciais adequadas.

## 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.
8. Introdução à classificação.
9. Variáveis e distribuições discretas e contínuas.
10. Inferência estatística e por reamostragem.
11. Regressão linear simples e múltipla.

## Bibliografia Básica

Livros:

1. MAGALHÃES, M. N.; DE LIMA, A. C. P. , Noções de Probabilidade e Estatística, 7<sup>a</sup> ed., Edusp, 2013
2. MONTGOMERY, D. , Estatística Aplicada e Probabilidade para Engenheiros , 6<sup>a</sup> ed., LTC, 2016
3. DOWNEY, A.B., Think Stats, 1<sup>a</sup> ed., O'Reilly Media, 2011

Artigos:

BOLLEN, J. et al.. Twitter mood predicts the stock market.. Journal of Computational Science. , v. 2 , n. 1 , p. 1-8 , 2011. ; Disponível em: <http://www.sciencedirect.com/science/article/pii/S187775031100007X>. Acesso em: 27 maio 2019

## Bibliografia Complementar

Livros:

1. DEKKING, F. M.; KRAAIKAMP, C. , A Modern Introduction to Probability and Statistics: Understanding Why and How., Springer, 2010
2. SCHILLER, J.; SRINIVASAN, A.; SPIEGEL, M. , Probability and Statistics, McGraw-Hill, 2011
3. HAYTER, Anthony J, Probability and Statistics for Engineers and Scientists, 4<sup>a</sup> ed., Duxbury Press, 2012
4. MCKINNEY, W., Python for data analysis: data wrangling with Pandas Numpy and IPython., O'Reilly Media, 2012
5. GRUS, J., Data Science from Scratch, 1<sup>a</sup> ed., O'Reilly Media, 2015

Artigos:

NASSIRTOUSSI et al.. Text mining for market prediction: A systematic review.. Expert Systems with Applications. , v. 41 , n. 16 , p. 7653-7670 , 2014. ; Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417414003455>. Acesso em: 27 maio 2019.



# Matemática da Variação

Carga Horária: 110 Semestre 2º período

## Ementa

Cálculo Diferencial e Integral com funções de uma variável: introdução, taxa de variação, limite, derivadas, integrais. Resolução analítica de equações diferenciais ordinárias. Álgebra Linear e a resolução analítica de sistemas de equações diferenciais. Desenvolvimento da autonomia em relação ao aprendizado de conteúdos matemáticos.

## Objetivos

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

1. Quantificar, interpretar e expressar algébrica e graficamente as taxas de variação média e instantânea de uma grandeza em relação a outra (OA1).
2. Interpretar e calcular o valor acumulado de uma variável dependente com a alteração do valor da variável independente (OA2).
3. Utilizar os conceitos e ferramentas vistos no curso para criar modelos de situações da realidade, envolvendo principalmente equações diferenciais, com o objetivo de estabelecer previsões e tomar decisões (OA3).
4. Utilizar ferramentas da Álgebra Linear para resolver modelos matemáticos originados da caracterização de sistemas dinâmicos (OA4).
5. Aprender a aprender matemática, ou seja, deve desenvolver autonomia, em relação ao conhecimento matemático, para buscar fontes de estudo e selecionar métodos que tornem seu aprendizado mais eficiente (OA5).

## Conteúdo Programático:

1. Os problemas fundamentais do Cálculo: taxa de variação instantânea e cálculo da acumulação de uma variável.
2. Limite de sequências, limite e continuidade de funções.
3. Derivada em um ponto e função derivada: interpretações algébrica e geométrica.
4. Cálculo da derivada de diferentes funções.
5. Integral definida e indefinida.
6. Teorema Fundamental do Cálculo.
7. Cálculo da integral de diferentes funções.
8. Equações diferenciais ordinárias de 1ª ordem: métodos analíticos de resolução.
9. Equações diferenciais ordinárias de 2ª ordem: métodos analíticos de resolução.
10. Aplicações de derivadas: máximos e mínimos, concavidade, gráficos.
11. Diferenciais, aproximações lineares e polinômio de Taylor.

Funções de  $R^n$  em  $R$ : 1. Funções de duas variáveis reais e representação gráfica; funções de  $n$  variáveis reais. 2. Curvas de nível. 3. Limites e continuidade. 4. Derivadas parciais e interpretação geométrica; derivadas direcionais; derivadas de ordens superiores. 5. Vetor gradiente: interpretação e cálculo. 6. Plano tangente e reta normal; diferencia

## **Bibliografia Básica**

Livros:

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

## **Bibliografia Complementar**

Livros: 1. BOYCE, W. E.; DIPRIMA, R. C., Equações Diferenciais Elementares e Problemas de Valores de Contorno, 9<sup>a</sup> 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, 1<sup>a</sup> ed., Pioneira Thomson Learning, 2004 4. ZILL, D., CULLEN, M. , Equações Diferenciais, 3<sup>a</sup> ed., Pearson, 2001 5. ROGAWSKI, J., Cálculo, Volume 1, 1<sup>a</sup> ed., Bookman, 2009

# Projeto Ágil e Programação Eficaz

Carga Horária: 160 Semestre 1º período

## Ementa:

Identificação do problema real aplicando a metodologia design thinking; Representação de estrutura e comportamento de um modelo em diagramas UML: casos de uso, classe, sequência, colaboração e implantação; Programação Orientada a Objetos; Técnicas de Planejamento e Gerenciamento de Software; Documentação de projetos de Software; Estratégias para Construção de Software: TDD (desenvolvimento baseado em testes) e DDD (desenvolvimento baseado em domínio); 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; Estratégias de testes automatizados de software: testes de unidade; Técnicas eficientes de depuração de programas; Técnicas eficientes de explorar bases de código; Desenvolvimento em ambiente colaborativo e distribuído e controle de versão; Modelo de colaboração usando submissões, revisão de código, integração e entrega contínuos, testes automatizados e padrões de código; Sinalização e tratamento de erros e exceções; Monitoramento de software e arquivos de registro de histórico (log).

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Atuar em uma equipe gerenciada por métodos ágeis revezando papéis nesta equipe
2. Estruturar um sistema orientado a objeto a partir de requisitos
3. Expressar arquitetura e comportamento de um sistema a partir de diagramas UML
4. Utilizar ferramentas e soluções de desenvolvimento para que o time esteja apto a colaborar num projeto
5. Escrever testes de unidade que expressem comportamento esperado de um subsistema
6. Familiarizar-se com uma base de código pré-existente e entendê-la a ponto de fazer intervenções
7. Resolver problemas de implementação a partir de resultados encontrados na Web e avaliar criticamente as fontes encontradas
8. Dentre um conjunto de requisitos identificar quais são funcionais e quais são não funcionais

## Conteúdo Programático

1. Design Thinking

2. Introdução à programação orientada a objetos
3. Diagramas UML
4. Introdução ao UML
5. Mapeamento de requisitos em objetos de domínio
6. Como estruturar um projeto de software
7. Colaboração usando controle distribuído de versão
8. Desenvolvimento orientado a testes
9. Técnicas de depuração de código

## **Bibliografia Básica**

1. MARTIN, ROBERT C. Clean Code: A Handbook of Agile Software Craftsmanship 2008. 1.a ed. Pearson.
2. THOMAS, D. HUNT, A. The Pragmatic Programmer: Your Journey to Mastery. 2.a ed. Addison-Wesley Professional, 2019.
3. VALENTE, M.T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Publicação Independente. 1.a edição. Belo Horizonte, 2020.
4. BROWN, TIM. Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, 2019.

## **Bibliografia Complementar**

4. MCCONNEL, S. Code Complete: A Practical Handbook of Software Construction. 2.a ed. Microsoft Press, 2004.
5. SOMMERVILLE, I. , Engenharia de Software , 9ª ed., Pearson, 2011
6. SCHWABER, K. , Agile Project Management with Scrum, Microsoft Press, 2004
7. PRESSMAN, R., Engenharia de Software: Uma abordagem profissional., 8ª ed., AMGH, 2016
8. K.BECK., Test Driven Development: By example, Addison-Wesley Professional, 2002
9. BECK, K.; ANDRES, C. , Extreme programming explained: Embrace change, Addison-Wesley,, 2005

# **Técnicas de Programação**

Carga Horária: 80 Semestre 3º período

## **Ementa**

Estruturas ligadas simples, pilhas e filas; estratégia de resolução de problemas com recursão; operações com matrizes; imagens representadas como matrizes; algoritmos simples em imagens; introdução à notação assintótica; algoritmos simples com cadeias de caracteres; busca e backtracking; testes de software.

## **Objetivos**

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

1. Resolver problemas de divisão e conquista utilizando recursão
2. Empregar estruturas ligadas para implementar pilhas e filas
3. Aplicar estruturas lineares simples (pilhas e filas)
4. Problemas geométricos usando computação gráfica e processamento de imagem
5. Relacionar pixels a formas e contornos
6. Identificar casos de complexidade linear e quadrática e expressá-la usando notação assintótica
7. Resolver problemas computacionais que exijam representação com matrizes
8. Trabalhar com arquivos, diferentes tipos de representação e diferentes tipos de acesso
9. Expressar alocação de recursos e liberação após o uso em uma linguagem de programação
10. Escrever algoritmos simples de busca e comparação de cadeias de caracteres
11. Resolver problemas simples com busca e backtracking
12. Escrever testes unitários simples e asserções para testar software

## **Conteúdo Programático**

1. Operações em matrizes
2. Estruturas lineares - pilhas e filas
3. Recursão
4. Busca e backtracking
5. Matrizes como imagens
6. Introdução a notação assintótica
7. Operações com strings
8. Arquivos, alocação e liberação de recursos
9. Introdução a testes de software.

## **Bibliografia Básica**

1. ZIVIANI, Nívio. Projeto de Algoritmos: com implementação em Pascal e C. 3ª ed. revista e ampliada. São Paulo: Cengage Learning, 2011
2. SEDGEWICK,, Robert. Algorithms in C - Parts1-4: Fundamentals, Data Structures, Sorting, Searching. 3ª ed. Addison-Wesley, 1997
3. SEDGEWICK, R.; WAYNE, K.; DONDERO, R. Introduction to Programming in Python: An Interdisciplinary Approach. Ed. Addison Wesley. 2015.

## **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 implementações em Java e C++. 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.

# Inteligência Artificial e Robótica

Carga Horária: 80

Semestre : 3º período

## Ementa

Arquitetura computacional de agentes e o laço percepção – planejamento e ação; Percepção, sensores e incerteza; Representação de conhecimento incerto e redes de representação de conhecimento; Robótica probabilística e Filtragem Estatística; Localização e mapeamento; Lógica proposicional, lógica de predicados de 1ª ordem e planejadores; Arquiteturas deliberativas; Arquiteturas cognitivas reativas, robótica comportamental, modelo de subsumption; Arquiteturas cognitivas híbridas; Processamento de imagens, amostragem, realce, filtragem e segmentação; Noções de visão computacional e reconhecimento de padrões: obtenção de informações de alto nível cognitivo a partir de imagens, contornos e classificadores baseados em posições e em vetores de características; 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. Especificar sensores, atuadores e plataforma computacional para um robô que precisa resolver um determinado problema
2. Desenvolver programas de computador capazes de resolver problemas de percepção, planejamento e controle/atuação
3. Selecionar melhores técnicas de programação e estruturas de dados adequadas a programas que habilitem robôs a funcionar adequadamente
4. Aplicar tecnologias computacionais atuais voltadas a robótica para prover uma dada funcionalidade a um usuário
5. Planejar e executar um trabalho em equipe, fornecendo e assimilando desenvolvimentos
6. Caracterizar e depurar problemas de desempenho de sistemas compostos de computação, sensores e atuadores

## Conteúdo Programático

1. Introdução à disciplina, evolução e aplicações da robótica
2. Enfoques computacionais para robótica
3. Introdução a processamento de imagens: modelos de imagem, cores, percepção, luz com onda eletromagnética, metameria
4. Imagens em cores, modos de representação e histograma
5. Modelo pinhole de câmera e fotogrametria, calibração de câmeras

6. Filtros lineares, realce, suavização, detecção de bordas e transformada de Hough
7. Robot Operating System - arquitetura de eventos - listeners e subscribers
8. Programação em Python para ROS
9. Arquiteturas comportamentais reativas e deliberativas
10. Máquinas de estados e modelo subsumption
11. Probabilidade condicional e teorema de Bayes - revisão
12. Estimativa de valor de estado dadas as leituras de sensores - estimativas discretas de posição e filtro de partículas
13. Navegação e localização em robôs móveis
14. Lógica de predicados e programação lógica
15. Descrição lógica de espaço de estados e transformação de estados
16. Planejamento
17. Tópicos avançados: Localização e Mapeamento Simultâneos
18. Tópicos avançados: Classificação de padrões aplicada a robótica
19. Tópicos avançados: Visão computacional avançada

## Bibliografia Básica

Livros:

1. NORVIG, P.; RUSSELL, S., Inteligência Artificial, 3ª ed., Campus Elsevier, 2013
2. SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D., Introduction to Autonomous Mobile Robots., 2ª ed., MIT Press, 2011
3. SZELISKI, R.; GRIES, D.; SCHNEIDER, F. B. (Ed.), Computer Vision: Algorithms and Applications, Springer, 2011

Artigos:

INGRAND, F.; GHALLAB, M.. Deliberation for autonomous robots: a survey.. Artificial Intelligence. , v. 247 , p. 10-44 , 2017. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370214001350>. Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. KAEHLER, A.; BRADSKI, G., Learning OpenCV: Computer Vision in C++ with the OpenCV Library, 2ª ed., O'Reilly Media, 2015
2. O'KANE, J., A Gentle introduction to ROS, CreateSpace Publishing, 2013
3. SCHERZ, P.; MONK, S., Practical electronics for inventors, 3ª ed., McGraw Hill, 2013
4. ASTRÖM, K.; MURRAY, R., Feedback Systems: An Introduction for Scientists and Engineers, Princeton University Press, 2008
5. THRUN, S.; BURGARD, W; FOX, D., Probabilistic Robotics., MIT Press, 2006



Artigos:

TENORTH, M.; BEETZ, M.. Representations for robot knowledge in the KnowRob framework. Artificial Intelligence. , v. 247 , p. 151-169 , 2017. ; Disponível em: <http://www.scienced>

# Álgebra Linear e Teoria da Informação

Carga Horária: 80

Semestre : 3º período

## Ementa

Vetores, bases e projeções; espaços vetoriais em geometria; decomposição em valores singulares; sistemas lineares e método de eliminação de Gauss; aplicações de álgebra linear em processamento de sinais; aplicações em grafos; aplicações em processamento de imagens.

## Objetivos

Ao final do curso, o aluno deve ser capaz de

1. Resolver problemas aplicados de Geometria Espacial que envolvam decomposições e projeções de vetores em bases
2. Reconhecer problemas em que a linearidade permite expressar um elemento de um conjunto como soma de outros elementos
3. Identificar aplicações em computação de conceitos de Álgebra Linear
4. Aplicar métodos de resolução de problemas baseados em Álgebra Linear
5. Reconhecer como a decomposição de informação e a reconstrução por linearidade está presente nas várias aplicações da computação
6. Reconhecer como os conceitos de sinal, ruído, perda e informação se manifestam nas várias aplicações da computação
7. Entender as noções de informação de Shannon e de Turing e sua interação nas aplicações da computação

Conteúdo Programático: 1. Vetor 2. Base e Espaço Vetorial 3. Decomposição e Projeção 4. Álgebra Linear em Geometria 5. Outros espaços vetoriais: Aplicações em sinais 6. Aplicações em processamento de imagens 7. Aplicações em grafos 8. Eliminação de Gauss 9. Decomposição em valores singulares .

Bibliografia Básica 1. KLEIN, P. N. Coding the Matrix: Linear Algebra Through Applications to Computer Science. Newtonian Press. 2013. 2. AXLER, S. Linear Algebra Done Right. 3rd. ed. Springer. 2014. 3. STRANG, G. Álgebra Linear e Suas Aplicações. Ed. Cengage Learning. São Paulo: 2019 4. MacKay, D. J. Information Theory, Inference and Learning Algorithms.

Bibliografia Complementar

1. STRANG, G. Introduction to Linear Algebra. Wellesley-Cambridge Press. 2019.
2. CHAPRA, Steven C.; CANALE, Raymond P. Métodos numéricos para engenharia. 5a. edição. São Paulo: McGraw-Hill, 2008.
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.
5. STEINBRUCH, A.; WINTERLE, P. Álgebra Linear. Ed. Pearson.
6. HEFFERON, J. Linear Algebra. 4.a ed.
7. COHEN, X. M. Linear Algebra: Theory, Intuition and Code. Sixxpress, 2021.

# Lógica e Matemática Discreta

Carga Horária: 80

Semestre : 3º período

## Ementa

Lógica proposicional; lógica de predicados e quantificadores; relações lógicas e tabela verdade; regras de dedução; leis de De Morgan; indução e recursividade; relações sobre conjuntos e o produto cartesiano; regras de inferência direta; aspectos matemáticos de sistemas provadores de teoremas; provas matemáticas por contradição; recursões, recorrências e indução; análise probabilística discreta de estruturas de dados; algoritmos aleatorizados; NP completude.

## Objetivos

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

1. Aprender a representar conhecimento através de lógica proposicional
2. Aprender a derivar sentenças verdadeiras a partir de um conjunto de proposições
3. Aplicar lógica de predicados para representar relações mais complexas
4. Raciocinar de forma matemática sobre predicados lógicos
5. Entender limitações computacionais de sistemas provadores de teoremas
6. Aplicar lógica para representar conhecimento
7. Realizar provas por contradição sobre propriedades matemáticas e computacionais
8. Aplicar probabilidade para analisar estruturas de dados e algoritmos
9. Reconhecer problemas clássicos que são NP-Completo e aplicar heurísticas e estratégias para encontrar soluções aproximadas

## Conteúdo Programático

1. Lógica proposicional
2. Tabelas verdade
3. Regras de dedução e leis de DeMorgan
4. Regras de inferência direta
5. Aspectos matemáticos de sistemas provadores de teoremas
6. Provas matemáticas por contradição
7. Recursões, recorrências e indução
8. Análise probabilística de estrutura de dados
9. Algoritmos aleatorizados
10. NP completude

## **Bibliografia Básica**

1. STEIN, C., DRYSDALE, R. L., BOGART, K. Matemática Discreta para Ciência da Computação. Editora Pearson, 2013.
2. GERSTING, Judith L. Fundamentos Matemáticos para a Ciência da Computação: Um Tratamento Moderno de Matemática Discreta, 7<sup>a</sup> ed., Editora LTC, 2016.
3. SILVA, Flávio S.C.; Finger, M.; de Melo, Ana C.V. Lógica para Computação, 2a. ed. Ed. Cengage Learning, 2017.

## **Bibliografia Complementar**

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. ROSS, K. A. e WRIGHT, C. R. B. Discrete Mathematics. Prentice-Hall, 1992.

# Programação Web e Times Ágeis

Carga Horária: 80

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 expressar 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. Aprender a aprender a desenvolver sistemas para web focando em uma escolha madura de linguagem, framework e arquitetura dependendo de cada contexto de implementação.
6. Planejar e acompanhar o projeto conforme princípios e valores ágeis, usando um framework diferente dos utilizados em outras disciplinas (Kanban, Scrum, squad, etc)
7. Tomada de decisão sobre adoção de frameworks de desenvolvimento levando em conta interesses organizacionais

## Conteúdo Programático:

Arquitetura da Web: IP, TCP/IP e roteamento

2. Arquitetura da Web: HTTP e WWW
3. Tecnologias para front-end: CSS, HTML, Javascript e Ajax/JSON
4. Usabilidade em aplicações Web / Interação Humano-Computador
5. Programação de servidores em Java: Servlets

6. SQL: Introdução e criação de CRUD
7. Camada de comunicação entre back-end servidor e banco de dados
8. Projeto de aplicações seguindo arquitetura Model-View-Controller
9. Hospedagem de aplicações web e publicação do trabalho na web
10. Projeto de aplicações seguindo arquitetura REST
11. Bases de dados não relacionais (NoSQL)
12. Front-ends com frameworks single-page (Angular)
13. 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.

# Algoritmos e Estruturas de Dados

Carga Horária: 80 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, 3<sup>a</sup> ed., Elsevier, 2012
3. SEDGEWICK, R. , Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, 3<sup>a</sup> 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, 2<sup>a</sup> ed., Springer, 2008
2. CORMEN, T. H., Algorithms Unlocked, MIT Press,, 2013
3. BENTLEY, J., Programming Pearls, Addison-Wesley, 1999
4. AHO, A. V.; ULLMAN, J. D. , Foundations of Computer Science: C Edition. W. H. , Freeman, 1994
4. R.SEDGEWICK., Algorithms in C, Part 5, 3<sup>a</sup> 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.

# Machine Learning

Carga Horária: 80

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, análise de texto, 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 classificação supervisionada, e avaliar sua sensibilidade e especificidade.
3. Projetar e implementar uma solução de aprendizado de máquina para problemas de regressão multivariada, e avaliar a qualidade do modelo obtido.
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 análise de texto via aprendizado de máquina
6. Descrever e experimentar com técnicas de reinforcement learning
7. Descrever e experimentar com técnicas de deep learning

## Conteúdo Programático

1. O aprendizado de máquina: arquitetura de solução, seleção de características, métodos de avaliação
2. Regressão: Regressão linear, regularização, Support Vector Machines, redes neurais
3. Classificação: Regressão logística, k-NN, árvores de decisão, boosting e métodos de ensemble
4. Clustering e métodos não-supervisionados, análise de componentes principais.

5. Análise de texto: análise de sentimento e modelagem de tópicos
6. Reinforcement learning
7. Deep learning.

## 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 and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, 2017
3. RIBEIRO-NETO, B.; BAEZA-YATES, R. , Recuperação de Informação - Conceitos e Tecnologia Das Máquinas de Busca, 2ª ed., Bookman, 2013

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. MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. , Introduction to Information Retrieval, 1ª ed., Cambridge University Press, 2008 3. DUDA, R. O., HART, P. E., STORK, D. G, Pattern Classification, 2ª ed., Wiley, 2001 4. O'NEILL, C., SCHUTT, R., , Doing Data Science: Straight Talk from the Frontline, O'Reilly, 2014 5. MURPHY, K. P., , Machine Learning: A Probabilistic Perspective, MIT Press, 2012 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.

# Teoria da Computação e Linguagens Formais

Carga Horária: 80

Semestre : 4º período

## Ementa

Lógica proposicional e de predicados; Teoria da Computabilidade; Decidibilidade e Intratabilidade; Máquina de Turing; Hierarquia de Chomsky; Gramáticas e reconhecedores (autômatos) para linguagens regulares, livres de contexto, sensíveis a contexto e recursivamente enumeráveis; Forma normal de Backus-Naur; Analisadores léxicos, sintáticos e semântico; Compiladores.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Especificar uma gramática para reconhecer uma linguagem de interesse.
2. Saber programar e gerar o diagrama explicativo de um autômato que implementa uma gramática.
3. Especificar um analisador léxico por meio de expressões regulares.
4. Compreender os conceitos básicos sobre Lógica Matemática, Teoria da Computabilidade e Máquina de Turing.
5. Entender todos os passos necessários à compilação de programas.

## Conteúdo Programático

1. Linguagens regulares e autômatos finitos
2. Linguagens livres de contexto e autômatos de pilha
3. Linguagens sensíveis ao contexto
4. Linguagens recursivamente enumeráveis
5. Máquina de Turing
6. Decidibilidade e intratabilidade
7. Complexidade e as classes P e NP
8. Lógica proposicional e de predicados
9. Compiladores
10. Análise léxica, sintática e semântica
11. Geração de código

## Bibliografia Básica

Livros:

1. JOSÉ NETO, J., Introdução à Compilação., 1ª ed., Elsevier, 2016
2. BOOLOS, G. S.; BURGESS, J. P.; JEFFREY, R. C., Computabilidade e Lógica, 1ª ed., Unesp, 2012

3. AHO, A.V.; LAM, M. S.; SETHI, R.; ULLMAN, J., Compiladores: Princípios, Técnicas e Ferramentas, 2ª ed., Longman, 2007

Artigos:

GOOD, J.; HOWLAND, K.. Programming language, natural language? Supporting the diverse computational activities of novice programmers.. Journal of Visual Languages & Computing.. , v. 39 , p. 78-92 , 2017. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S1045926X16301963>. Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. RAMOS, M. V. M.; JOSÉ NETO, J.; VEJA, I. S., Linguagens Formais. Teoria, Modelagem e Implementação, 1ª ed., Bookman, 2009
2. CORMEN, Thomas H., LEISERSON; Charles E., RIVEST; Ronald L., STEIN, Clifford., Algoritmos: teórica e prática., 3ª ed., Elsevier-Campus., 2012
3. HOPCROFT, J. E.; ULLMAN, J. D; MOTWANU, R., Introdução à Teoria dos Autômatos, Linguagens e Computação, 1ª ed., CAMPUS, 2002
4. SIPSTER, M., Introdução à Teoria da Computação, 2ª ed., Thomson Pioneira, 2007
5. SILVA, F. C.; FINGER, M.; MELO, A. C. V., Lógica para Computação, 2ª ed., Cengage, 2017

Artigos:

KONDOH, H., FUTATSUGI, K.. To use or not to use the goto statement: Programming styles viewed from Hoare Logic.. Science of Computer Programming. , v. 60 , n. 1 , p. 82-116 , 2006. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167642305000924>. Acesso em: 27 maio 2019.

# Sistemas Hardware-Software

Carga Horária: 80

Semestre : 5º 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, 3<sup>a</sup> ed., Pearson, 2015
2. D. A. PATTERSON; J. L. HENNESSY, Organização e Projetos de Computadores: a Interface Hardware/Software., 4<sup>a</sup> ed., Elsevier, 2014
3. KOCHAN, S. G., Programming in C, 4<sup>a</sup> 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, 2<sup>a</sup> 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, 8<sup>a</sup> ed., Pearson, 2013
4. SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G., Operating System Concepts, 9<sup>a</sup> ed., Wiley, 2013
5. TANENBAUM, A. S., Sistemas Operacionais Modernos, 4<sup>a</sup> 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.

# Projeto de Software e Gestão de Projetos Ágeis

Carga Horária: 80

Semestre : 4º período

## Ementa

Modelagem da Arquitetura de Software; Padrões de Projeto; Formas de documentação do projeto e boas práticas de codificação; Engenharia de Software; Padrões de Projeto; Princípios da responsabilidade única, princípio do aberto/fechado, princípio da substituição de Liskov, da segregação de interface e da inversão de dependência. Refatoração de código. 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. Desenvolver o código seguindo as boas práticas de codificação.
5. Entender e identificar padrões de projeto
6. Definir quais práticas ágeis são adequadas para cada necessidade de projeto
7. Aplicar as práticas ágeis em seus projetos
8. Garantir e mensurar a qualidade do código
9. Identificar e gerir Dívida Técnica
10. Definir estratégias e métricas que suportem a tomada de decisão baseada em dados
11. Utilizar data driven como forma de tomada de decisão do projeto e da equipe
12. Alinhar as decisões do time aos milestones do projeto e a estratégia de negócio
13. 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**

APPELO JURGEN. Management 3.0: Leading Agile Developers, Developing Agile Leaders (Addison-Wesley Signature Series (Cohn)) (English Edition) 1ª

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

HOHPE, GREGOR. The Software Architect's Elevator.

KRUCHTEN, PHILIPPE., IPEK OZKAYA Managing Technical Debt: Reducing Friction in Software Development (SEI Series in Software Engineering) (English Edition), 2019.

MARTIN ROBERT. C, MICHAEL C. FEATHERS, TIMOTHY R. OTTINGER. Clean Code: A Handbook of Agile Software Craftsmanship, 2018.

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

SUTHERLAND, JEFF. SCRUM: a arte de fazer o dobro do trabalho na metade do tempo, 2019.

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

WINTERS, TITUS, TOM MANSHREK, HYRUM WRIGHT. Software Engineering at Google. O'Reilly's Media, Inc, 2020.

## **Bibliografia Complementar**

1. GAMMA, ERICH, RICHARD HELM e outros. Padrões de Projetos: Soluções Reutilizáveis de Software Orientados a Objetos, 2000.
2. GREEVER, TOM. Articulando Decisões de Design: Converse com os Stakeholders, Mantenha sua Sanidade e Crie a Melhor Experiência do Usuário, 2021.
3. RIGBY DARREL, SARAH ELK e outros. Ágil do Jeito Certo: Transformação sem caos. Edição Português, 2020.

# Análise de Algoritmos e Entrevistas Técnicas

Carga Horária: 80 Semestre 5º período

## Ementa

Invariantes; indução; análise probabilística; análise amortizada; limitantes inferiores e superiores; Teorema Mestre; algoritmos gulosos; simulação de recursão com pilhas; adição de memoização; treap; union-find; árvores geradoras mínimas; fluxo máximo e corte mínimo; polinomialidade; pseudopolinomialidade; certificados; redução; problemas NP-completos e NP-difíceis.

## Objetivos

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

1. Aplicar estratégias clássicas de projeto de algoritmos.
2. Aplicar estratégias clássicas de análise de algoritmos.
3. Analisar a correção de algoritmos procedurais.
4. Analisar a correção de algoritmos recursivos.
5. Avaliar a complexidade de algoritmos.
6. Avaliar a tratabilidade de problemas.
7. Identificar ambiguidades e pontos vagos na descrição de um problema;
8. Projetar e implementar uma solução técnica eficiente para problemas típicos de entrevistas técnicas de programação;
9. Descrever o processo de elaboração da solução e identificar a sua complexidade computacional.

## Conteúdo Programático

1. Revisão de algoritmos clássicos e notação assintótica
2. Análise de correção por invariantes.
3. Análise de correção por indução.
4. Estratégias para análise precisa de caso médio.
5. Estratégias para análise precisa de algoritmos recursivos.
6. Estratégias para otimização de algoritmos recursivos.
7. Estruturas de dados avançadas.
8. Algoritmos clássicos de otimização combinatória.
9. Avaliação de complexidade.
10. Avaliação de tratabilidade.
11. Conceitos e algoritmos: manipulação de bits, desafios de lógica e matemática, recursão, programação dinâmica, ordenação e busca;
12. Projeto de testes.

## Bibliografía Básica

5. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L; STEIN, C. Algoritmos: Teoría e Práctica, 3a ed. Elsevier, 2012.
6. SEDGEWICK, R. Algorithms in C, Part 5: Graph Algorithms, 3rd ed. Addison-Wesley, 2001.
7. MCDOWELL, G. L., Cracking the coding interview, 6<sup>a</sup> ed., CareerCup, 2015
8. HALIM, S.; HALIM, F., Competitive programming: the new lower bound of programming contests, 3<sup>a</sup> ed., , 2013
9. SKIENA, S. S.; REVILLA, M. A., Programming Challenges: The Programming Contest Training Manual, Springer-Verlag, 2003
10. SANCHEZ-ORO, J.; DUARTE, A. Iterated Greedy algorithm for performing community detection in social networks. Future Generation Computer Systems, v. 88, págs 785-791, 2018.

## Bibliografía Complementar

1. BENTLEY, J. Programming Pearls, 2nd ed. Addison-Wesley, 1999.
2. SKIENA, S. S.; REVILLA, M. A. Programming Challenges: The Programming Contest Training Manual. Springer, 2003.
3. SKIENA, S. S. The Algorithm Design Manual, 2nd ed. Springer, 2008.
4. SEDGEWICK, R.; WAYNE, K. Algorithms, 4th ed. Addison-Wesley, 2011.
5. AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.
6. POUNDSTONE, W., How would you move Mount Fuji?: Microsoft's cult of the puzzle, how the world's smartest companies select the most creative Thinkers, Little, Brown and Company, 2004
7. MCDOWELL, G.L., The Google Resumé, Wiley, 2011
8. POUNDSTONE, W., Are you smart enough to work at Google?: Trick questions, zen-like riddles, insanely difficult puzzles, and other devious interviewing techniques you ... know to get a job anywhere in the new economy, Little, Brown and Company, 2012
9. AZIZ, A. et al., Elements of Programming Interviews: the insiders' guide, Createspace, 2012
10. AZIZ, A. et al., Elements of programming interviews in Python: the insiders' guide, Createspace, 2016
11. GAI, K.; QIU, M.; LIU, M.; XIONG, Z. In-memory big data analytics under space constraints using dynamic programming. Future Generation Computer Systems, v. 83, págs 219-227, 2018.

# 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.

# AI Startup

## Ementa

Ciclo de adoção de técnicas de inteligência artificial e técnicas analíticas avançadas por empresas. Start ups versus organizações. Canvas do modelo de negócios. Geração de valor de um projeto. Análise de ambiente externo e interno. Análise de custos. Produto mínimo viável. Proposta de valor. Segmentação de mercado. Noções de direito para start ups. Relacionamento com o cliente. Dimensionamento de atividades e recursos.

## Objetivos

Ao final da disciplina o aluno será capaz de:

1. Situar fenômenos de inovação, e antecipar suas consequências como fonte de criação de valor.
2. Ser capaz de analisar uma empresa tecnológica e definir as dinâmicas envolvidas, bem como as competências necessárias às pessoas que nelas atuam.
3. Entender processos de pesquisa e inovação e medir fatores de eficácia como vetor de sucesso de uma empresa.
4. Enfatizar a importância da estratégia em termos de criação e captura de valor.
5. Identificar fenômenos de criação de valor em setores fortemente impactados pela economia do conhecimento (TI, e-commerce, redes sociais).
6. Caracterizar uma empresa em função do seu posicionamento face aos movimentos de inovação inerentes ao seu setor (redes de inovação).
7. Compreender os fatores de transformação de uma indústria e o papel dos mecanismos nacionais de inovação no apoio à criação de valor via inovação.
8. Entender a importância da propriedade intelectual e os mecanismos para protegê-la.
9. Explicar os modelos de negócios para projetos envolvendo inovação tecnológica, e o papel do capital de risco no seu financiamento.
10. A partir de casos reais, analisar os fatores que levaram empresas tecnológicas ao sucesso ou ao fracasso.
11. Conhecer e integrar os componentes básicos de um plano de projeto.

## Conteúdo Programático

1. Por que empreender com embasamento tecnológico.
2. O que sabemos hoje: startups não são versões menores de grandes corporações.
3. Canvas do Modelo de Negócios: uma ferramenta para projetar os empreendimentos de amanhã.
4. Pense como cientista, aja como empreendedor.
5. Canvas do Modelo de Projetos.

6. O entendimento da geração de valor do projeto para a organização.
7. Quais produtos/serviços/resultados o projeto está entregando? Quais são suas características diferenciadas que atendem às necessidades do cliente do projeto?
8. Quem, da organização, dedicar-se-á ao projeto e por quanto tempo? Com quais órgãos do ambiente externo ao projeto teremos que lidar?
9. Qual trabalho, com foco em resultado, cada um tem que produzir? Em quais condições esse trabalho pode e deve ser feito? 10 Em que prazo/custo podemos nos comprometer considerando os principais riscos?
10. MVP.
11. Proposta de Valor.
12. Direito para startups.
13. Segmento de clientes.
14. Relacionamento com clientes.
15. Canais de entrega da proposta de valor para os clientes.
16. Formas inovadoras de geração de receita.
17. Atividades principais em uma startup baseada em ETCM.
18. Conhecimento, aprendizado e concepção (design).
19. Recursos principais em uma startup baseada em ETCM.
20. Parceiros para desenvolvimento de produtos e do negócio.
21. Custos em uma startup baseada em ETCM.
22. Organização da nova empresa.
23. Desenvolvimento de clientes.
24. Como criar produtos que os clientes desejem.
25. Como se relacionar com os clientes focando em reduzir os riscos e aumentar os ganhos da empresa.
26. Como definir as atividades e os recursos da empresa visando redução do tempo até o mercado.
27. Criando parceiros que trazem credibilidade e minimizem os custos para a empresa.

## Bibliografia Básica

Livros:

1. BLANK, S.; DORF, B., The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company, 1ª ed., K&S Ranch, 2012
2. OSTERWALDER, Alexander; PIGNEUR, Yves, Business Model Generation: A Handbook for visionaries, game changers, and challengers. , John Wiley & Sons, 2010
3. FINOCCHIO JR, J., Project Model Canvas: Gerenciamento de Projetos sem Burocracia, Elsevier Brasil, 2014 Artigos: WOUTERS, M.; ANDERSON, J. C.; KIRCHBERGER, M.. New-Technology Startups Seeking Pilot Customers: Crafting a Pair of Value Propositions.. California Management Review. , v. 60 , n. 4 , p. 101-124 , 2018. ; Disponível em: <http://journals.sagepub.com/doi/abs/10.1177/0008125618778855>.



Acesso em: 27 maio 2019.

## **Bibliografia Complementar**

Livros:

1. BYERS, T.; DORF, R.; NELSON, A., Technology Ventures: From Idea to Enterprise, 4<sup>a</sup> ed., McGraw-Hill Education, 2014
2. SPINELLI, S.; ADAMS, R., New Venture Creation: Entrepreneurship for the 21st Century, McGraw-Hill, 2011
3. AULET, B., Disciplined Entrepreneurship: 24 steps to a successful startup., John Wiley & Sons, 2013
4. RIES, E., A Startup Enxuta: Como os Empreendedores Atuais Utilizam a Inovação Contínua para Criar Empresas Extremamente Bem-sucedidas., Leya, 2012
5. MANKIW, G., Introdução à Economia., 6<sup>a</sup> ed., Cengage Learning, 2013 Artigos: BOGERS, M., CHESBROUGH, H., MOEDAS, C.. Open innovation: research, practices, and policies.. California Management Review. , v. 60 , n. 2 , p. 5-16 , 2018. ; Disponível em: <http://journals.sagepub.com/doi/full/10.1177/0008125617745086>. Acesso em: 27 maio 2019.

# 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; Jogos inteligentes; Jogos multiplayer; Jogos para consoles; 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, como 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. MARTINHO, C.; SANTOS, P.; PRADA, R., Design e Desenvolvimento de Jogos, FCA, 2014
3. MILES, J., Unity 3D e PlayMaker Essentials: Game Development from Concept to Publishing, CRC Press,, 2016

Artigos:

KARPINSKYJ, S. et. al.. Video game personalisation techniques: A comprehensive survey.. Entertainment Computing.. , v. 5 , n. 4 , p. 211-218 , 2014. ; Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84907845372&doi=10.1016%2fj.entcom.2014.09.002&partnerID=40&md5=605f2683fac288315360c62C6049a2Cd>. Acesso em: 27 maio 2019.

## Bibliografia Complementar

Livros:

1. BARRERA, R.; KYAW, A.S.; PETERS, C.; SWE, N., Unity AI Game Programming, 2ª ed., Packt Publishing, 2015 RANI, K. A., Learning Unity Physics., 2014ª ed., Packt Publishing, 0
2. SAUNDERS, K.D.; NOVAK, J. , Game Interface Design, 2ª ed., Cengage Learning, 2012 STAGNER, A.R, Unity Multiplayer Games., Packt Publishing, 2013
3. VASCONCELOS, J.B.; RIBEIRO, N., Tecnologias de Programação de Jogos, FCA, 2013

Artigos:

WALLNER, G.; KRIGLSTEIN, S.. Visualization-based analysis of gameplay data - A review of literature.. Entertainment Computing. , v. 4 , n. 3 , p. 143-155 , 2013. ; Disponível em: <https://www.scopus.com/inward/record.uri?eid=2->

s2.0- 84875590999&doi=10.1016%2fj.entcom.2013.02.002&partnerID=40&md5=e7f5b9129a9fa8e7d2366  
266db57d87f. Acesso em: 27 maio 2019.

# Plataformas, Microsserviços e APIs

Carga Horária: 80

Semestre : 4º período

## Ementa:

Aspectos tecnológicos e de mercado do oferecimento de computação como um serviço segundo o paradigma de computação em nuvem (cloud computing); Principais modelos de consumo de computação em nuvem: Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS); Arquiteturas distribuídas fundamentais para viabilizar a computação em nuvem; Máquinas virtuais, contêineres, instâncias e provisionamento; Orquestração de computação em nuvem, sistemas de gerenciamento, monitoramento e configuração de recursos virtualizados; Desenvolvimento e implantação de software capaz fazer uso eficiente de computação em nuvem e apresentar escalabilidade; Monitoramento de performance para gerenciamento de aplicações; Questões de segurança e privacidade da informação associadas à computação em nuvem; Gestão de níveis de serviço (SLA - Service Level Agreement). Custos de projeto e de operação de sistemas em cloud. Aspectos de automação de gestão de sistema em cloud – DevOps. Serverless computing – FaaS - function as a service

## Objetivos

Ao final da disciplina o estudante será capaz de:

1. Identificar conceitos relevantes de sistemas distribuídos no contexto de computação em nuvem.
2. Administrar um sistema de gerenciamento de nuvem, provisionando a infraestrutura necessária como um serviço.
3. Desenvolver aplicações que apresentem o atributo da escalabilidade por meio do uso de técnicas de computação em nuvem.
4. 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).

## Conteúdo Programático:

1. Fundamentos de Computação em Nuvem.
2. Infraestrutura como um serviço.
3. Orquestradores de implantação.
4. Gerenciamento de ambientes virtualizados e distribuídos.
5. Introdução a contêineres.
6. Redes definidas por Software.
7. Micro serviços com interface API.
8. Software como um serviço.

9. Gestão de níveis de Serviço.

## **Bibliografia Básica**

Livros:

1. TANENBAUM, A. S.; VAN STEEN, M., Sistemas Distribuídos, 2<sup>a</sup> ed., Prentice Hall Brasil, 2007
2. KAVIS, M., Architecting the Cloud: Design Decisions for Cloud Computing Service Models , Wiley, 2014
3. TANENBAUM, A. S; Wetherall, D. J. , Redes de Computadores, 5<sup>a</sup> ed., Pearson Prentice Hall, 2011 Artigos: NOOR, T. H. et al.. Mobile cloud computing: Challenges and future research directions.. Journal of Network and Computer Applications. , v. 115 , p. 70-75 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804518301504>. Acesso em: 27 maio 2019.

## **Bibliografia Complementar**

Livros:

1. PATTERSON, D.; FOX, A. , Engineering Software as a Service: An Agile Approach Using Cloud Computing, 3<sup>a</sup> ed., LLC Press, 2013
2. ERL, T.; PUTTINI, R.; MAHMOOD, Z. , Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013
3. JACKSON, K.; BUNCH, C.; SIGLER, E. , Openstack Cloud Computing Cookbook., 3<sup>a</sup> ed., Packt Publishing, 2015
4. PORTNOY, M., Virtualization Essentials, 2<sup>a</sup> ed., John Wiley, 2016
5. GENG, H. , Datacenter Handbook, John Wiley, 2015

Artigos:

- AI, Y.; PENG, M.; ZHANG, K.. Edge computing technologies for Internet of Things: a primer.. Digital Communications and Networks. , v. 4 , n. 2 , p. 77-86 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352864817301335> . Acesso em: 27 maio 2019.

# 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 projetos. Análise de metodologias de trabalho para projetos em Computação. Comunicação efetiva e assertiva. Identificação e viabilização de papéis 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 papéis 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 Científica. 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.