

## Plano de ensino

**Curso:** SIN-BAC - Bacharelado em Sistemas de Informação

**Turma:** BSIN182-4 - BSIN182-4

**Disciplina:** 4DAD203 - ESTRUTURA DE DADOS II

**Período letivo:** 2022/2

**Carga horária:** 72

**Professor:** 211021223 - ALEXANDRE MENDONCA FAVA

### Ementa

1. Meios de armazenamento, dispositivos de acesso sequencial, dispositivos de acesso aleatório, registros, listas invertidas, hashing, sort, backup. Métodos de ordenação e busca de dados. Noções sobre complexidade de algoritmos na avaliação de desempenho de programas.

### Objetivo geral

1. Possibilitar ao acadêmico a habilidade de analisar problemas e desenvolver soluções computacionais que envolvam estrutura de dados, ou proporções de aprimoramento de algoritmos.

### Objetivo específico

1. O discente deverá, ao final do semestre letivo, ser capaz de:
  - 1.1 Compreender as formas de construir e manipular estrutura de dados;
  - 1.2 Implementar estruturas de dados para a resolução de problemas;
  - 1.3 Desenvolver a noção de complexidade de tempo e espaço no desenvolvimento de softwares;
  - 1.4 Implementar os principais algoritmos de ordenação.

### Conteúdo programático

1. Aula Inaugural
  - 1.1 Apresentação do Professor
  - 1.2 Apresentação da Disciplina
  - 1.3 Metodologia de Ensino Utilizada
  - 1.4 Critérios de Avaliação Estudantil
  - 1.5 Bibliografia Básica e Complementar
2. Meios de armazenamento, dispositivos de acesso sequencial, dispositivos de acesso aleatório e conceito de registros:
  - 2.1 Conceitos básicos e classificação dos meios de armazenamento
  - 2.2 Estrutura dos discos, operações e tempos de acesso
  - 2.3 Características dos sistemas de arquivos, interface, vantagens e desvantagens
  - 2.4 Conceitos de registros, campos e chaves de acesso
  - 2.5 Manipulação de arquivos, registros de tamanho fixo e registros de tamanho variável
  - 2.6 Tipos de fluxo de dados, acesso direto, acesso sequencial, sequencial indexado e acesso aleatório
3. Listas invertidas, hashing, sort e backup:
  - 3.1 Composição, estrutura e endereçamento de listas invertidas
  - 3.2 Conceitos de hashing e tabelas de dispersão, função de espalhamento, colisões, aplicação e limitações
  - 3.3 Utilização de coleções, algoritmos destrutivos, operações de sort e backup
4. Avaliação (P1)
5. Métodos de Ordenação e busca de dados:
  - 5.1 Conceitos de ordenação, tipos, vantagens e desvantagens
  - 5.2 Prática com métodos bubbleSort, insertSort, selectSort, heapSort, mergeSort e quickSort
  - 5.3 Conceitos de busca de dados, tipos, vantagens e desvantagens
  - 5.4 Prática com métodos de busca linear e busca binária
6. Avaliação (P2)
7. Noções sobre complexidade de algoritmos na avaliação de desempenho de programas:
  - 7.1 Conceitos básicos de complexidade de algoritmos, complexidade espacial e temporal, notações assintóticas
  - 7.2 Conceitos básicos de desempenho, pior caso, melhor caso e caso médio
  - 7.3 Algoritmos ótimos
8. Avaliação (P3)

## Plano de ensino

### Metodologia

1. A disciplina será desenvolvida através de aulas expositivo-dialogadas, com exercícios práticos orientados em sala de aula e exercícios a serem passados para resolução extraclasse (conforme postagens no diretório da disciplina no Moodle e/ou Teams).

Os períodos disponibilizados para atendimento individualizado são: quartas-feiras, das 18:10 às 20:50 h, na sala do professor. Outros períodos para atendimentos devem ser previamente agendados pelo correio eletrônico: alexandre.fava@udesc.br

O material didático poderá ser disponibilizado na plataforma Moodle e/ou Teams.

### Sistema de avaliação

1. Do desempenho do aluno:  
A qualidade do desempenho do aluno será avaliada com base no desenvolvimento das seguintes atividades e com os seguintes critérios:

- a) Avaliação contínua englobando assiduidade e participação em aula (AC);
- b) Provas Individuais (P1, P2 e P3);
- c) Exercícios Avaliativos (EA).

A Média Semestral (MS) será calculada pela fórmula:

$$MS = (0.25 \times P1) + (0.25 \times P2) + (0.25 \times P3) + [0.25 \times (EA + AC)]$$

As notas serão expressas na escala de 0 (zero) a 10 (dez) e poderão ser fracionadas em apenas um dígito após a vírgula, adotando-se o arredondamento estatístico. A Média Semestral (MS) deve ser igual ou superior a 7,0 (sete vírgula zero).

Caso o aluno não obtenha aprovação direta pela Média Semestral (MS), deverá fazer um Exame Final (EF).

Após Exame Final (EF) o aluno deverá obter Média Final (MF) para aprovação igual ou superior a 5,0 (cinco vírgula zero).

A Média Final (MF) será calculada pela fórmula:

$$MF = [(MS \times 6) + (EF \times 4)]/10$$

Será considerado aprovado o acadêmico que obtiver Média Semestral (MS) igual ou superior a 7,0 (sete vírgula zero) e frequência não inferior a 75%. O acadêmico que não obtiver a média 7,0 (sete vírgula zero) estará, obrigatoriamente, em exame, cujo desempenho será composto por média semestral com peso 6 (seis) e o exame final com peso 4 (quatro), devendo atingir a média final de, no mínimo, 5,0 (cinco vírgula zero) e frequência não inferior a 75%.

Das regras para revisão das avaliações:

Depois da publicação das notas pelo professor, os alunos têm 07 dias corridos para solicitar a revisão com o professor. Esta revisão será feita na sala do professor, preferencialmente em horário de atendimento aos alunos, ou em um horário do qual o professor possa atendê-los.

Do desempenho da disciplina e do professor:

Os discentes terão, igualmente, a oportunidade de fazer uma avaliação mais completa do desempenho do professor e da disciplina através do sistema de avaliação eletrônico SIGA.

### Bibliografia básica

1. CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L. e STEIN Clifford. Algoritmos: Teoria e Prática. 6. ed. Rio de Janeiro: Elsevier, 2002.  
  
GOODRICH, Michael T; TAMASSIA, Roberto. Estruturas de dados e algoritmos em Java. 4. ed. Porto Alegre: Bookman, 2007.  
  
MENEZES, Nilo Ney Coutinho. Introdução à Programação com Python: Algoritmos e lógica de programação para iniciantes. 2 ed. 5 reimp. São Paulo: Novatec, 2017.  
  
ZIVIANI, Nivio. Projeto de Algoritmos: com implementações em Pascal e C. 2. ed. rev. e ampl. São Paulo: Pioneira Thomson Learning, 2005.

### Bibliografia complementar

1. ASCENCIO, Ana F. G.; ARAÚJO, Graziela S. de. Estruturas de dados: algoritmos, análise da complexidade e implementações em Java e C/C++. São Paulo: Pearson, 2010.

## **Plano de ensino**

DEITEL, Harvey M.; DEITEL, Paul J. Java: como programar. São Paulo: Prentice Hall, 2010.

EDELWEISS, Nina; GALANTE, Renata. Estruturas de dados. Porto Alegre: Bookman, 2009.

PINTO, Rafael A.; PRESTES, Lucas P.; SERPA, Matheus da S.; et al. Estrutura de dados. Grupo A, 2020. 9786581492953. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9786581492953/>.

RAMALHO, Luciano. Python Fluente: Programação clara, concisa e eficaz. São Paulo: Novatec, 2015.

VETORAZZO, Adriana de S.; SARAIVA, Márcio de O.; BARRETO, Jeanine dos S.; JR., Ramiro S C. Estrutura de dados. Grupo A, 2018. 9788595023932. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788595023932/>.