



Universidade Federal do Ceará
Sistemas de Informação
Sistema de Presença e Planos de Aula

Plano de Ensino

Código: QXD0010 **Turma:** 01A-SI **Disciplina:** Estruturas de Dados

Período: 2023.1 **Créditos:** 4.0 **Créditos Práticos:** 2.0

Professor(a): Atilio Gomes Luiz

Horários: SEGUNDA 8h-10h; TERÇA 8h-10h;

Justificativa:

Ao chegar à disciplina de Estrutura de Dados, os alunos já aprenderam os conceitos básicos da arte de programar na disciplina de Fundamentos de Programação. Contudo, é na disciplina de Estrutura de Dados que os alunos irão começar a utilizar estruturas dinâmicas que permitem o desenvolvimento de aplicações mais eficientes, confiáveis e estruturadas.

Ementa:

- Introdução à linguagem C++ e revisão de Fundamentos de Programação: Arrays, Structs, Ponteiros e Alocação dinâmica de memória
- Recursão
- Noções de análise de complexidade de algoritmos
- Tipos Abstratos de Dados (TAD) e Introdução aos conceitos de Classe e Objeto
- Listas Sequenciais
- Listas Encadeadas
- Pilhas, Filas e aplicações
- Algoritmos de busca em vetor, Pesquisa Binária
- Árvores Binárias: representação, percurso e busca
- Árvores Binárias de Busca
- Algoritmos de ordenação básicos: Bubble Sort, Selection Sort e Insertion Sort
- Algoritmos de ordenação rápidos: Mergesort, Heapsort e Quicksort

Objetivos Gerais e Específicos:

A disciplina Estrutura de Dados visa dar ao aluno a capacidade de dominar as principais técnicas utilizadas na implementação de estruturas de dados básicas, de algoritmos de ordenação em memória principal e algoritmos de pesquisa. Ao final desta disciplina os alunos deverão ser capazes de:

- Efetuar análises simples da complexidade de algoritmos.
- Compreender a importância de algoritmos eficientes na construção de estruturas de dados.
- Analisar algoritmos simples e decidir sobre qual a melhor escolha a ser feita durante a implementação de uma dada estrutura de dados.
- Implementar estruturas de dados essenciais como listas dinâmicas, pilhas, filas e árvores binárias tendo em vista sempre a eficiência e a reusabilidade de código.

Aula	Data	Plano de Aula
1	13/03/2023	Apresentação do Plano de Ensino Semestral
2	14/03/2023	Introdução ao C++: Compilação, Primeiros Programas, Entrada e saída de Dados, Formatação de dados
3	20/03/2023	Revisão de FUP: arrays, estruturas de seleção e repetição, funções, arrays, matrizes, structs
4	21/03/2023	Ponteiros e passagem de parâmetro por referências
5	27/03/2023	Alocação dinâmica de memória, ponteiro para ponteiro, alocação dinâmica de matriz
6	28/03/2023	Referências (References) em C++
7	03/04/2023	Recursão
8	04/04/2023	Recursão
9	10/04/2023	Backtracking
10	11/04/2023	Noções de complexidade de Algoritmos
11	17/04/2023	Noções de complexidade de Algoritmos
12	18/04/2023	Tipos Abstratos de Dados

13	24/04/2023	Tipos Abstratos de Dados e os conceitos de Classe e Objeto
14	25/04/2023	Sobrecarga de Operadores
15	02/05/2023	Listas Sequenciais
16	08/05/2023	Listas Simplesmente Encadeadas
17	09/05/2023	Listas Circulares
18	15/05/2023	Pilhas
19	16/05/2023	Filas
20	22/05/2023	Algoritmos de Busca em Arrays
21	23/05/2023	Algoritmos de Busca em Arrays
22	29/05/2023	Aplicações de Pilhas e Filas
23	30/05/2023	Árvores Binárias
24	05/06/2023	Percurso em Árvores Binárias
25	06/06/2023	Árvore Binária de Busca
26	12/06/2023	Percursos em Árvore Binária de Busca
27	13/06/2023	Algoritmos de Ordenação Básicos
28	19/06/2023	Algoritmos de Ordenação Básicos
29	20/06/2023	Mergesort
30	26/06/2023	QuickSort
31	27/06/2023	HeapSort
32	03/07/2023	Guardado para possíveis atrasos no curso

Data da Prova Final:

17/07/2023

Metodologia de Ensino:

- Aulas expositivas com quadro branco e pincel.
- Aulas práticas em laboratório.
- Exercícios semanais de implementação de algoritmos via plataforma Moodle. Esses exercícios comporão a primeira parte da nota do aluno.
- Haverão dois projetos de implementação de estruturas de dados. Esses projetos comporão a segunda e terceira parte da nota do aluno.
- A linguagem de programação utilizada na disciplina será C++.

Atividades Discentes:

- Durante o semestre, os discentes serão encorajados à responder uma lista de exercício para cada conteúdo da disciplina.
- Haverão laboratórios virtuais de programação semanais disponibilizados na plataforma Moodle, que os estudantes deverão responder como parte da avaliação da disciplina. Além disso, ao longo do semestre, os discentes desenvolverão trabalhos de implementação que visam consolidar o entendimento das estruturas de dados vistas em aula.

 *** Código de Ética

Qualquer tentativa de fraude nas atividades implicará em nota igual a zero para todos os envolvidos. Alguns exemplos de fraudes são:

- enviar ou mostrar trechos de código para um amigo no intuito de ajudá-lo.
- copiar trabalho de outros.
- compartilhar trechos de código, utilizar trechos de códigos da internet ou de outras fontes.

Avaliação:

Durante o semestre, haverá atividades semanais explorando os temas vistos durante as aulas. À todas estas atividades semanais serão atribuídas notas de 0 a 10 e, ao final, todas estas notas comporão uma nota de Avaliação Parcial, a AP1. Haverão dois trabalhos práticos (AP2 e AP3) a serem entregues no decorrer do semestre.

- A média M do aluno na disciplina será calculada como a média ponderada das APs, do seguinte modo:

$$M = (2 \cdot AP1 + AP2 + AP3) / 4$$

- Se o aluno obtiver média $M \geq 7$, ele será considerado aprovado por média se também tiver pelo menos 75% de presença no curso; caso contrário, estará reprovado por falta.

- Caso o aluno obtenha média $4 \leq M < 7$, ele poderá fazer uma avaliação final. Seja AF a nota da avaliação final. A média final, F, para os alunos que precisam de avaliação final será calculada como: $F = (M + AF) / 2$. Para que o aluno que fez a AF seja considerado aprovado, ele deve obter média final $F \geq 5$ e deve ter pelo menos 75% de presença no curso.

Bibliografia Básica:

FEOFILOFF, Paulo. . Algoritmos em linguagem C. Rio de Janeiro: Elsevier, 2009. 208p. ISBN 9788535232493 (broch.).

DROZDEK, Adam. Estrutura de dados e algoritmos em C++. São Paulo: Thomson, 2002. 579p ISBN 852210295 (broch.).

ZIVIANI, Nívio; BOTELHO, Fabiano Cupertino. Projeto de algoritmos: com implementações em java e C++. São Paulo, SP: Thomson Learning, 2007. 620 p. ISBN 8522105251 (broch.).

Bibliografia Complementar:

CORMEN, Thomas H. Algoritmos: teoria e prática. Rio de Janeiro: Elsevier, 2002. 916 p. ISBN 8535209263 (broch.).

SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. Estruturas de dados e seus algoritmos. 2. ed. rev. Rio de Janeiro: Livros Técnicos e Científicos, c1994. 320 p. ISBN 8521610149 (broch.).

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e java. 2. ed. São Paulo, SP: Prentice Hall, 2007. viii, 434 p. ISBN 8576051480 (broch.).

AGUILAR, Luis. Fundamentos de programação: algoritmos, estrutura de dados e objetos. São Paulo, SP: McGraw-Hill, 2008. 690 p. ISBN 9788586804960 (broch.).

CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. Introdução a estrutura de dados: com técnicas de programação em C. Rio de Janeiro: Elsevier, 2004. 294 p. (Campus, Sociedade Brasileira de Computação) ISBN 8535212280 (broch.).

Recursos Didáticos:

- Plataforma Moodle e Software Visual Studio Code para realização de atividades semanais.
- Grupo no Telegram para comunicação rápida e efetiva.
- Livros e tutoriais de linguagem C++, disponíveis na internet.
- Quadro branco, pincel, slides, Datashow.