

Plano de Ensino

## **5ESAT**

### **Estrutura de Dados**

60.0 horas

#### **Ementa**

Nesta disciplina o aluno será capaz de implementar as principais estruturas de dados e praticar os conceitos fundamentais de programação. Desenvolver as estruturas de dados, com o intuito de aprimorar a utilização de técnicas de programação. O aluno utilizará as estruturas de dados juntamente com os conceitos de programação estruturada, desenvolvendo o raciocínio lógico, capacitando-se para utilizar corretamente as estruturas estáticas e dinâmicas com seus respectivos critérios (listas, pilhas, filas, árvores, etc.), de acordo com o problema a ser resolvido. O processo culmina em avaliar práticas e teorias, ao implementar projetos e trabalhos.

#### **Objetivos**

1. Implementar programas utilizando a programação estruturada de dados, para o efetivo desenvolvimento da lógica no uso das ferramentas de linguagens de programação.
2. Solucionar desafios de programação, utilizando conceitos de estrutura de dados de forma otimizada, para o aprimoramento da utilização das técnicas de programação.
3. Desenvolver pesquisas de softwares atuais, utilizando programação estruturada de dados, para o aperfeiçoamento dos critérios de programação de estrutura de dados.
4. Projetar softwares utilizando linguagens modernas para resolução de problemas clássicos da programação de computadores, através de atividades práticas.
5. Solucionar problemas e situações do mundo real, aplicando os conhecimentos aprendidos acerca de estruturas de dados computacionais.

#### **Conteúdo**

1. Revisão de Ponteiros: alocação e desalocação de memória, endereço de memória, referência de memória. Revisão de Lógica de Programação: estruturas sequenciais, condicionais, de controle, de repetição, operadores lógicos e aritméticos.
2. Estruturas Estáticas utilizando: Tipos Básicos, Operadores de incremento e Decremento, Array, Matrizes, Operador sizeof, Conversão de Tipos, Entrada e Saída com scanf e printf, funções, bloco, variáveis globais e locais, passagem por referência.
3. Estruturas Dinâmicas apresentando os conceitos de: Vetores e Matrizes dinâmicos, Tipos de estruturas lineares e não-lineares, Listas Encadeadas, Listas Simplesmente Encadeada, Lista Duplamente Encadeada, Lista Circular, Fila e Pilha.
4. Listas (conceito, introdução, abstração dos dados, declaração, inicialização, inserção, sobreposição, remoção, teste, exemplos, listar elementos), Lista Ordenada, inserção ordenada, procura, métodos de teste, Listas de Listas.
5. Listas Encadeadas: motivação, vetor, nós da lista, Estrutura com ponteiro, Implementações recursivas, Listas de tipos estruturados, Lista de inteiros, criação, inserção, remoção, impressão,

teste de vazio e busca.

6. Listas Circulares: definição, exemplo, função para imprimir uma lista circular. Listas Duplamente Encadeadas: definição, exemplo, função de inserção, função de busca, função para retirar um elemento da lista.

7. Fila em Estruturas Estática e Dinâmica: Estruturas de Dados Avançadas, conhecido mundialmente como FIFO (First In First Out) ou LIFO (Last In Last Out), Fila Estática Circular, Implementação Estática e Dinâmica Encadeada.

8. Pilha em Estruturas Estática e Dinâmica: conhecida internacionalmente como LIFO (Last In First Out = Último a Entrar é o Primeiro a Sair) ou FILO (First In Last Out = Primeiro a Entrar é o Último a Sair), implementação Estática e Dinâmica.

9. Lista com implementação Estática com Remanejamento e Lista com implementação Estática com Encadeamento: criação, inserção, remoção, busca, deslocamento, relocação, verificação de lista cheia e lista vazia.

10. Programação com Recursividade: conceitos, implementações, vantagens, desvantagens, aplicações na solução de problemas mundialmente conhecidos como, sequência de Fibonacci, Torres de Hanoi, uso de backtracking.

11. Programação com Árvores: conceitos envolvendo árvores, sub-árvores, nós, nó raiz, nó pai, nó filho (nós internos), folhas (nós externos), estrutura da árvore, representação gráfica, tipos de estruturas de árvores.

12. Programação com Árvores Binárias: definição e conceitos, aplicações, representação gráfica, árvore vazia, sub-árvore da direita (sad) e a sub-árvore da esquerda (sae), Ordens de percurso em árvores binárias. Árvores genéricas.

13. Programação com Árvores Binárias Balanceadas AVL -Adelson-Velskii e Landis: balanceamento, árvores completas, definição de uma árvore AVL, rotações em AVL simples e dupla, rotações RR (Right-Right), LL (Left-Left), LR (Left-Right), RL (Right-Left).

14. Programação com Árvores B: árvores múltiplas, definição de árvore B, ordem da árvore B, profundidade da árvore B, chave da árvore, inserção de elemento, busca por elemento, separação de nó (split), remoção de chaves. Árvore B\* e Árvore B+.

## **Bibliografia**

### **Básica**

CELES, W., CERQUEIRA, R., RANGEL, J. L. Introdução a estruturas de dados: com técnicas de programação em C. Rio de Janeiro: Elsevier, 2004.

TENENBAUM, Aaron M. et al. Estruturas de dados usando C. São Paulo: Makron Books, 1995.

ZIVIANI, Nivio. Projeto de Algoritmos com implementações em Pascal e C. São Paulo: Pioneira, 2002.

### **Complementar**

SILVA, O. Q. Estrutura de Dados e Algoritmos Usando C - Fundamentos e Aplicações. São Paulo: Ciência Moderna, 2007.

SZWARCFITER, Jayme L.; MARKENZON, Lilian. Estruturas de dados e seus algoritmos. Rio de Janeiro: LTC, 1994.

LORENZI, F., MATTOS, P. N., CARVALHO, T. P. Estruturas de Dados. São Paulo: Cengage Learning, 2007.

SCHILD, H. C completo e total. São Paulo: Makron / McGraw-Hill, 1997.

DROZDEK, Adam. Estrutura de dados e algoritmos em C++. São Paulo: Síntese, 2002.

em partes. Todos os direitos reservados.