

# ALGORITMOS AVANÇADOS

## Apresentação do Plano de Ensino

Luiz Leão – [luizleao@gmail.com](mailto:luizleao@gmail.com)

<http://www.luizleao.com>



**Estácio**

## Introdução

- É a base sobre a qual muitos outros campos dessa área são construídos.
- O conhecimento dos Algoritmos Avançados é importante para estudantes que desejem trabalhar em **implementações, testes e manutenção** de projetos de qualquer sistema de software.

## Objetivos Gerais

- Desenvolver técnicas para representação de Algoritmos Avançados e as operações sobre os mesmos, de maneira que seja possível solucionar problemas, escolhendo os Algoritmos mais adequados para representação e manipulação dos dados em problemas específicos.

## Objetivos Específicos

- Identificar e conhecer a complexidade do elemento da análise assintótica: Notação  $O$ .
- Conhecer sobre recursividade e suas definições / implementações.
- Analisar a complexidade do algoritmo de ordenação por intercalação (mergesort).
- Analisar a complexidade do algoritmo de ordenação rápida (quicksort).
- Conhecer o Algoritmo em Grafos e as suas representações.
- Identificar e analisar as complexidades das estruturas de dados dos tipos Árvore Binária e Árvore AVL.

# UNIDADE I – Análise de Algoritmo - Notação O

- 1.1 - Algoritmo
- 1.2 - Estrutura de Dados
- 1.2.1 - Revisão de Programas em C envolvendo Vetores, Matrizes, Ponteiros, Registros (Struct) e Funções.
- 1.3 - O que é Análise de Algoritmos
- 1.4 - Sobre o Elemento da Análise Assintótica - Notação O
- 1.4.1 - Notação O
- 1.4.2 - Sobre a função
- 1.4.3 - Operações com a Notação O

## UNIDADE II – Recursividade

- 2.1 - Definições recursivas
- 2.2 - Como implementar recursividade
- 2.3 - Quando não usar recursividade
- 2.4 - Desenvolvendo algoritmos com recursividade

## UNIDADE III – Algoritmo de ordenação por intercalação (mergesort)

- 3.1 - Definição
- 3.2 - Dividir para conquistar
- 3.3 - Problema da intercalação
- 3.4 - O algoritmo de ordenação por intercalação mergesort.
- 3.5 - Análise da complexidade do algoritmo mergesort.



## **UNIDADE IV – Algoritmo de ordenação rápida (quicksort)**

- 4.1 - Definição
- 4.2 - Ordenação rápida
- 4.3 - O algoritmo de ordenação rápida quicksort
- 4.4 - Análise da complexidade do algoritmo quicksort



## UNIDADE V – Estruturas de dados dos tipos Árvore Binária e Árvore AVL

- 5.1 - Árvores, Árvores Binárias e Árvores Binárias de Busca
- 5.2 - Implementando as Árvores Binária
- 5.3 - Percorrendo uma Árvore Binária de Busca
- 5.4 - Percurso em Árvore
- 5.5 - Inserção
- 5.6 - Remoção
- 5.7 - Balanceando uma Árvore
  - 5.7.1 - O Algoritmo DSW
  - 5.7.2 - Árvores AVL

## UNIDADE VI – Algoritmos em Grafos

- 6.1 - Conceitos de grafos.
- 6.2 - Representação de grafo
- 6.3 - Algoritmos de busca
- 6.4 - Algoritmo do caminho mínimo
- 6.4.1 - Encontrar o melhor caminho do vértice origem ao vértice destino

# Linguagem

- C
- Java (a definir)

# Ferramentas

- Dev C++
- Netbeans

## Bibliografia

- ASCENCIO, Ana Fernanda Gomes; ARAÚJO, Graziela Santos de. **Estruturas de Dados: algoritmos, análise da complexidade e implementações em JAVA, C/C++**. São Paulo: Pearson Prentice Hall, 2010.