



# Introdução a Estrutura de Dados e Algoritmos – EDA

Por:

- ❖ **Dr. Alfredo Covele**
- ❖ **Eng. Cristiliano Maculuve**

Agosto 2023

# Plano Temático de EDA

No.	Temas
1	Introdução a EDA
2	Arrays, Interação e Invariantes
3	Listas, Recursividade, Pilhas e Filas
4	Pesquisas (lineares e binarias)
5	Arvores
6	Arvores de Pesquisa Binaria
7	Filas Prioritárias e Arvores <i>Heaps</i>
8	Ordenação
9	Tabelas Hash

# Estrutura de Dados e Algoritmos

Introdução

# Introdução

- **Algoritmo** – Pode ser definido como sendo “*uma sequência finita de instruções, das quais cada uma tem um significado claro e pode ser realizada com uma quantidade finita de esforço em um determinado tempo*”.
- Um algoritmo deve ser preciso o suficiente para ser compreendido por seres humanos;
- No entanto, para ser executado por um computador, geralmente precisamos de um programa que é escrito em uma linguagem formal rigorosa.

# Introdução

- ◉ **Porque estudar Algoritmos?**

- ◉ São importantes para inúmeras áreas da computação, como roteamento de redes, criptografia, computação gráfica, base de dados, biologia computacional, inteligência artificial, otimização combinatória, etc.
- ◉ Relevantes para inovação tecnológica pois, para resolver um problema computacional normalmente existem diversas soluções viáveis, por vezes com características e desempenho muito diferentes.

# Introdução

- Eles são interessantes, divertidos e desafiadores, pois o desenvolvimento de algoritmos mistura conhecimento técnico com criatividade.
- Por isso, na EDA irão tornar-se melhores programadores, melhorar habilidades analíticas e aprender a pensar algorítmicamente, i.e. ser capaz de entender as regras que regem diferentes processos.

# Introdução

- **Questões fundamentais sobre algoritmos:**
  - O que é suposto o algoritmo fazer?
  - Será que realmente faz o que é suposto fazer?
  - Com que eficiência faz?
- Os termos técnicos normalmente usados para as 3 questões:
  - Especificação;
  - Verificação;
  - Análise de performance.

# Introdução

- A **especificação** deve formalizar os detalhes cruciais do problema que o algoritmo destina-se a resolver.
- Tipicamente, terá que especificar como os *inputs* e *outputs* do algoritmo estão relacionadas;
- precisamos nos esforçar para **verificar** se o algoritmo está realmente correto;
- Em geral, o teste em alguns *inputs* particulares pode ser suficiente para mostrar que o algoritmo é incorreto.



# Introdução

- Finalmente, a **eficiência ou desempenho** de um algoritmo se relaciona aos recursos necessários por ele, como a velocidade de execução ou a capacidade da memória do computador que usará;
- Isso vai geralmente depender do tamanho da instância do problema, a escolha da representação de dados e os detalhes do algoritmo.

# Introdução

- ◉ **Estrutura de dados – visão geral**
- ◉ Usadas para organizar dados permitindo acesso rápido aos mesmos;
- ◉ Não existe estrutura perfeita, cada uma é eficiente para algumas operações e ineficiente para outras;
- ◉ Escolha a estrutura de dados mais simples que suporta todas as operações requisitadas pela sua aplicação.

# Introdução

- **Estrutura de dados - objectivos**
- Conhecer uma variedade de estruturas de dados;
- Entender os pontos fortes e fracos de cada uma, permitindo escolher onde utilizá-las;
- Saber como implementar e modificar as estruturas de dados para atender a necessidades específicas que surjam em suas aplicações.

# Atividade

- **Estudar os seguintes conceitos:**
  - (1) Tipo de dados abstratos
  - (2) Padrão de desenho.