

# Capítulo 1 — Análise de Algoritmo (Análise da Complexidade)

- **O que é:** Estudo do tempo de execução e do espaço (memória) que um algoritmo consome com base no tamanho da entrada ( $n$ ) [Wikipédia](#).
- **Objetivo:** Não basta que o algoritmo funcione; ele precisa ser eficiente. Um bom algoritmo em máquina lenta pode superar um ruim, mesmo em máquina rápida [Wikipédia](#).
- **Complexidade de pior caso:** Mede o tempo máximo necessário para executar o algoritmo, dado qualquer entrada de tamanho  $n$ . É um limitante superior útil para comparar algoritmos [Wikipédia](#).
- **Notação O (“Big O”):** Utilizada para expressar ordens de crescimento. Por exemplo, identificar quando um algoritmo é  $O(n^2)$  ou  $O(n^3)$ , independentemente dos termos constantes [PasseidiretoWikipédia](#).

**Por que guardar isso?** São conceitos fundamentais para escolher algoritmos eficientes, otimizar código e compreender críticas sobre performance nas explicações teóricas e práticas.

---

# Capítulo 3 — Estrutura de Dados do Tipo Listas

- **Listas Ligadas (Encadeadas):**
    - São estruturas dinâmicas formadas por nós, onde cada nó contém um valor e um ponteiro para o próximo elemento [Wikipédia](#).
    - Não precisam estar contíguas na memória — são flexíveis quanto ao tamanho e posição [Wikipédia](#).
    - **Vantagens:**
      - Inserção/remoção rápida (especialmente na cabeça da lista) sem deslocamento de elementos.
      - Tamanho adaptável — sem necessidade de definir limite máximo antecipadamente [Wikipédia](#).
    - **Desvantagens:**
      - É necessário percorrer a lista para acessar posição  $n$  (sem acesso direto como em vetores).
      - Se os ponteiros forem gerenciados incorretamente, pode-se perder toda a estrutura [Wikipédia](#).
  - **Listas Duplamente Encadeadas:**
    - Cada nó possui ponteiros tanto para o próximo quanto para o anterior (next e prev) [Wikipédia](#).
    - Permitem percorrer em ambos os sentidos com mais facilidade e eficiência, embora usem mais memória e demandem mais atenção ao manipular ponteiros [Wikipédia](#).
- 

## Ponteiros & Estruturas de Dados

- **Definição em geral:** Um ponteiro é uma variável cujo valor é o endereço de memória de outra variável [Wikipédia](#).
  - **Uso em estruturas:** São fundamentais para estruturas ligadas (como listas), porque definem como os elementos se conectam na memória [Wikipédia](#).
  - **Vetor vs. Ponteiro:** Em vetores, os elementos estão em sequência contígua, o que torna eficientes acessos e paralelismo, mas menos flexível em tamanho [Wikipédia](#).
- 

## Resumo Visual para Revisão

Conceito	Importância Principal
<b>Análise de Algoritmo</b>	Permite medir e comparar eficiência de algoritmos.
<b>Notação Big O</b>	Classifica algoritmos por ordem de crescimento.
<b>Listas Ligadas (simples)</b>	Dinâmicas, eficientes em inserção/remoção; necessitam travessia.
<b>Listas Duplamente Encadeadas</b>	Permitem travessia bidirecional; mais memória e complexidade.
<b>Ponteiros</b>	Essenciais para conectar elementos em estruturas ligadas; apontam para endereços.