

# RA03 - Relatório de Análise de Comparações em Operações de Inserção e Busca

Thiago Grasso

November 4, 2024

## 1 Introdução

Este relatório visa analisar a eficiência das operações de inserção e busca em uma estrutura de dados. Para esta análise, utilizamos um arquivo contendo dados sobre operações realizadas, registrando o número de comparações necessárias para cada inserção e busca. O objetivo é compreender como o número de comparações varia entre essas operações e o que isso pode nos dizer sobre o desempenho do sistema.

## 2 Escolhas de Implementação

Para a análise das operações de inserção e busca em tabelas hash, foi necessário definir alguns parâmetros importantes: o tamanho do vetor, a função hash utilizada e o conjunto de dados. Abaixo estão as escolhas feitas e suas respectivas justificativas.

### 2.1 Tamanhos do Vetor

Escolhi tamanhos de vetor de  $N = 100$ ,  $N = 500$  e  $N = 1000$  para verificar como a quantidade de posições na tabela influencia o número de comparações. Valores maiores oferecem uma perspectiva de comportamento em tabelas mais densas, enquanto tamanhos menores permitem uma análise com menor número de colisões.

### 2.2 Funções Hash

Escolhi as variações de função hash Divisão, Multiplicação e Dobramento; Estas funções são simples e representam bem o trabalho de funções hash em um programa, além de fornecerem resultados diretos e compreensíveis.

## 2.3 Conjunto de Dados

Para as chaves, usamos números inteiros sequenciais de 1 a 100, simulando uma carga típica de operações com valores que aumentam gradualmente. Esse conjunto de dados permite observar o comportamento em relação ao preenchimento gradual da tabela e o impacto nas comparações.

Essas escolhas foram feitas para facilitar a compreensão dos resultados e garantir que as operações refletissem cenários práticos. A escolha da função hash e dos tamanhos de vetor buscou minimizar o efeito de colisões ao mesmo tempo em que evidencia o comportamento das comparações em tabelas de diferentes tamanhos.

## 3 Descrição dos Dados

O arquivo de dados utilizado, denominado `dados_convertidos.csv`, contém as seguintes colunas:

- **Operação:** Indica o tipo de operação realizada, sendo "Inserir" ou "Buscar".
- **Chave:** Identificador único para cada item inserido ou buscado.
- **Comparações:** Número de comparações realizadas para completar a operação.

Os dados foram extraídos e organizados para facilitar a visualização do desempenho das operações em uma tabela hash, permitindo uma comparação direta entre o número de comparações em inserções e buscas.

## 4 Objetivo da Análise

O objetivo desta análise é avaliar a quantidade de comparações necessárias para realizar operações de inserção e busca. Isso nos ajuda a entender a eficiência dessas operações em termos de tempo de execução, uma vez que o número de comparações pode impactar diretamente o desempenho da estrutura de dados.

## 5 Gráfico de Comparações

Abaixo, apresentamos um gráfico que mostra o número de comparações realizadas em cada operação de inserção e busca ao longo de diversas chaves.

### 5.1 Interpretação do Gráfico

Observando o gráfico, percebemos que:

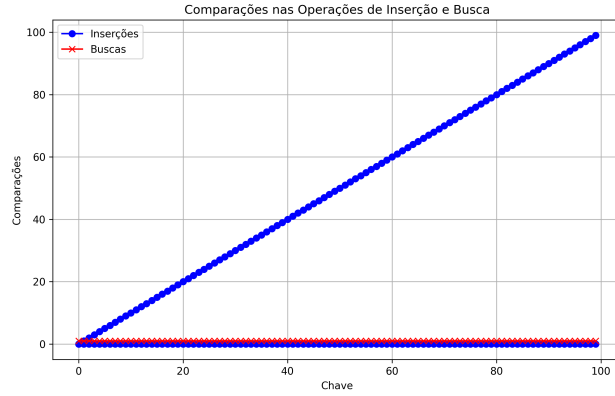


Figure 1: Comparações nas Operações de Inserção e Busca

- O número de comparações para **inserções** aumenta linearmente conforme o número de operações cresce, indicando uma maior complexidade conforme a tabela hash se preenche. Esse comportamento sugere que o número de comparações na inserção depende do estado atual da estrutura de dados (número de elementos já inseridos).
- Para **buscas**, o número de comparações permanece praticamente constante, o que sugere uma eficiência maior para esta operação, possivelmente devido ao uso de uma estratégia de busca que mantém o número de comparações baixo, mesmo com o aumento de chaves.

## 6 Conclusão

A análise demonstrou que as operações de inserção exigem um número crescente de comparações, enquanto as buscas mantêm um número de comparações estável. Esse comportamento pode indicar que, embora a estrutura de dados em análise seja eficiente para buscas, o custo de inserção aumenta à medida que mais elementos são adicionados.

Esses achados sugerem que, em aplicações onde o volume de inserções é alto, o desempenho pode ser impactado negativamente. Já para cenários onde buscas são mais frequentes, a estrutura parece adequada. Futuras análises podem incluir otimizações nas inserções ou investigar outros tipos de estrutura de dados para comparar o desempenho.