

# Relatório de Desempenho: Polynomial Hashing vs. Multiplier Hashing

## 1. Introdução

Este relatório compara o desempenho de duas funções de hash (Hashing Polinomial e Hashing Multiplicador). Os critérios analisados incluem:

- Número de colisões
- Distribuição de chaves (clusterização)
- Tempos médios de inserção e busca

## 2. Funções de Hash e Tratamento de Colisões

### 2.1 Hashing Polinomial

Calcula o hash usando um polinômio baseado nos caracteres da chave.

### 2.2 Hashing Multiplicador

Usa multiplicação por uma constante e extrai a parte fracionária.

### 2.3 Tratamento de Colisões: Sondagem Linear

Método:

- Em caso de colisão, busca a próxima posição vazia sequencialmente.

Impacto na clusterização:

- Pode formar clusters longos (sequências de posições ocupadas), aumentando o tempo de busca.

## 3. Análise dos Resultados

### 3.1 Número de Colisões

Métrica	Polinomial	Multiplicador
Colisões Totais	9	18

Observação:

- Função Polinomial teve metade das colisões do Multiplier Hashing, indicando melhor distribuição.

### 3.2 Clusterização (Distribuição de Chaves)

Métrica	Polinomial	Multiplicador
Posições Ocupadas	44/67 (65.7%)	44/67 (65.7%)
Número de Clusters	15	13
Tamanho do Maior Cluster	9	13

Conclusões:

- Função Multiplicadora gerou um cluster maior (13 posições), indicando pior distribuição.
- Função Polinomial distribuiu melhor, com cluster máximo menor (9 posições).

3.3 Tempos de Operação (nanossegundos)

Operação	Polinomial	Multiplicador
Inserção	4645,455 ns	30040,909 ns
Busca	2150,000 ns	2100,000 ns

Observações:

- Inserção: Função Polinomial foi ~6,5x mais rápida.
- Busca: Tempos próximos, mas Multiplier Hashing teve leve vantagem (provavelmente devido à localidade de cache em clusters grandes).

4. Conclusão

4.1 Função Polinomial

Vantagens:

- Menos colisões (9 vs. 18).
- Distribuição mais uniforme (menor cluster máximo).
- Inserção muito mais rápida.

Desvantagens:

- Busca ligeiramente mais lenta (mas diferença mínima).

4.2 Função Multiplicadora

Vantagens:

- Busca marginalmente mais rápida (devido a clusters contíguos).

Desvantagens:

- Muito mais colisões.
- Clusterização problemática (maior agrupamento).
- Inserção extremamente lenta.

4.3 Recomendação Final

- Função Polinomial é a melhor escolha para inserção eficiente e menor taxa de colisões.

