

Título do Documento

Seu Nome

December 31, 2024

Teste

# 1. INTRODUÇÃO À INDEXAÇÃO

## 1.1 O que é indexação?

Um índice é uma estrutura de dados que melhora a velocidade das operações de query de dados em uma tabela, ao custo de escritas adicionais e espaço de armazenamento para manter a estrutura do índice. Índices permitem localizar dados rapidamente sem precisar buscar sequencialmente em cada linha de uma tabela.

A maioria dos softwares de banco de dados inclui tecnologia de indexação que permite buscas em tempo sub-linear para melhorar o desempenho, já que a busca linear é ineficiente para grandes bancos de dados.

[\[Wik24c\]](#)

## 1.2 Implementações de índices

### 1.2.1 B Tree

Uma **B-tree** é uma estrutura de dados em árvore auto-balanceada que mantém dados ordenados e permite buscas, acessos sequenciais, inserções e deleções em tempo logarítmico. A B-tree generaliza a árvore binária de busca, permitindo nós com mais de dois filhos.

É amplamente utilizada em sistemas de arquivos e bancos de dados. É uma estrutura que se beneficia da leitura e escrita em bloco, levando vantagem em um aspecto historicamente relevante, uma vez que o número de operações de I/O (em discos magnéticos) era igualmente relevante para o desempenho quanto o número de operações de comparação.

Foi inventada por Rudolf Bayer e Edward M. McCreight em 1972 (o B não foi explicado por eles).

*What Rudy (Bayer) likes to say is, the more you think about what the B in B-Tree means, the better you understand B-Trees!*

Os principais algoritmos associados a B-trees são: busca (algorithm 1) e inserção (algorithm 2) (existem variações para a operação de deleção).

São necessárias duas funções auxiliares para a inserção: SPLITCHILD, que divide um nó cheio em dois, e INSERTNONFULL, que insere uma chave em um nó não cheio.

---

**Algorithm 1** Algoritmo de busca na B Tree, assumindo que a chave  $k$  é o valor a ser buscado e  $x$  é o nó onde a busca começa.

---

```

1: procedure BTREESEARCH( $x, k$ )
2:    $i \leftarrow 0$ 
3:   while  $i < x.n$  and  $k > x.key[i]$  do
4:      $i \leftarrow i + 1$ 
5:   end while
6:   if  $i < x.n$  and  $k = x.key[i]$  then
7:     return  $x$ 
8:   end if
9:   if  $x.leaf$  then
10:    return None
11:  end if
12:  return BTREESEARCH( $x.child[i], k$ )
13: end procedure

```

---



---

**Algorithm 2** Algoritmo de inserção na B Tree, assumindo que a chave  $k$  é o valor a ser inserido.

---

```

1: procedure BTREEINSERT( $T, k$ )
2:    $r \leftarrow T.root$ 
3:   if  $r.n = 2(T.d) - 1$  then
4:      $s \leftarrow \text{new Node}$ 
5:      $T.root \leftarrow s$ 
6:      $s.child[1] \leftarrow r$ 
7:     SPLITCHILD( $s, 1$ )
8:     INSERTNONFULL( $s, k$ )
9:   else
10:    INSERTNONFULL( $r, k$ )
11:  end if
12: end procedure

```

---

[Wik24a]

### 1.2.2 B+ Tree

[Wik24b]

## BIBLIOGRAPHY

- [Wik24a] Wikipedia. *B tree*. 2024. URL: <https://en.wikipedia.org/wiki/B-tree>.
- [Wik24b] Wikipedia. *B+ tree*. 2024. URL: [https://en.wikipedia.org/wiki/B%2B\\_tree](https://en.wikipedia.org/wiki/B%2B_tree).
- [Wik24c] Wikipedia. *Database index*. 2024. URL: [https://en.wikipedia.org/wiki/Database\\_index](https://en.wikipedia.org/wiki/Database_index).