

## Gerenciamento de Arquivos em Memória Secundária com Tabela Hash

Uma empresa de tecnologia está desenvolvendo um sistema de gerenciamento de arquivos armazenados em memória secundária (discos rígidos ou SSDs). Os arquivos possuem as seguintes informações:

- Nome do arquivo (ex.: "relatorio.pdf")
- Caminho completo no sistema de arquivos (ex.: "/documentos/relatorio.pdf")
- Tamanho do arquivo (em KB)

A empresa quer implementar uma **tabela hash** para organizar os arquivos com base no nome do arquivo como chave. O sistema deve permitir as seguintes operações:

1. Adicionar arquivos à tabela hash.
2. Buscar arquivos pelo nome.
3. Remover arquivos pelo nome.
4. Listar todos os arquivos armazenados.

Como requisito, o sistema precisa tratar colisões utilizando encadeamento separado (ou seja, cada posição da tabela armazena uma lista de arquivos no caso de colisão).

### Exemplo de Uso

Considere os seguintes arquivos armazenados:

1. Nome: "relatorio.pdf", Caminho: "/documentos/relatorio.pdf", Tamanho: 1024 KB
2. Nome: "foto.jpg", Caminho: "/imagens/foto.jpg", Tamanho: 2048 KB
3. Nome: "dados.csv", Caminho: "/planilhas/dados.csv", Tamanho: 512 KB
4. Nome: "backup.zip", Caminho: "/backup/backup.zip", Tamanho: 4096 KB

Com esses arquivos, o sistema deve ser capaz de realizar as seguintes operações:

1. Adicionar os arquivos à tabela hash.
2. Buscar pelo arquivo "dados.csv" e retornar suas informações.

3. Remover o arquivo "foto.jpg".
4. Listar todos os arquivos restantes armazenados na tabela hash.

**Regras:**

1. A tabela hash deve ser implementada com um tamanho fixo inicial (ex.: 10 posições).
2. A função hash deve ser baseada no nome do arquivo, convertendo-o para um índice dentro do intervalo da tabela.
3. Caso dois arquivos gerem o mesmo índice (colisão), o sistema deve tratá-los utilizando encadeamento separado.