



Algoritmos e Estrutura de Dados

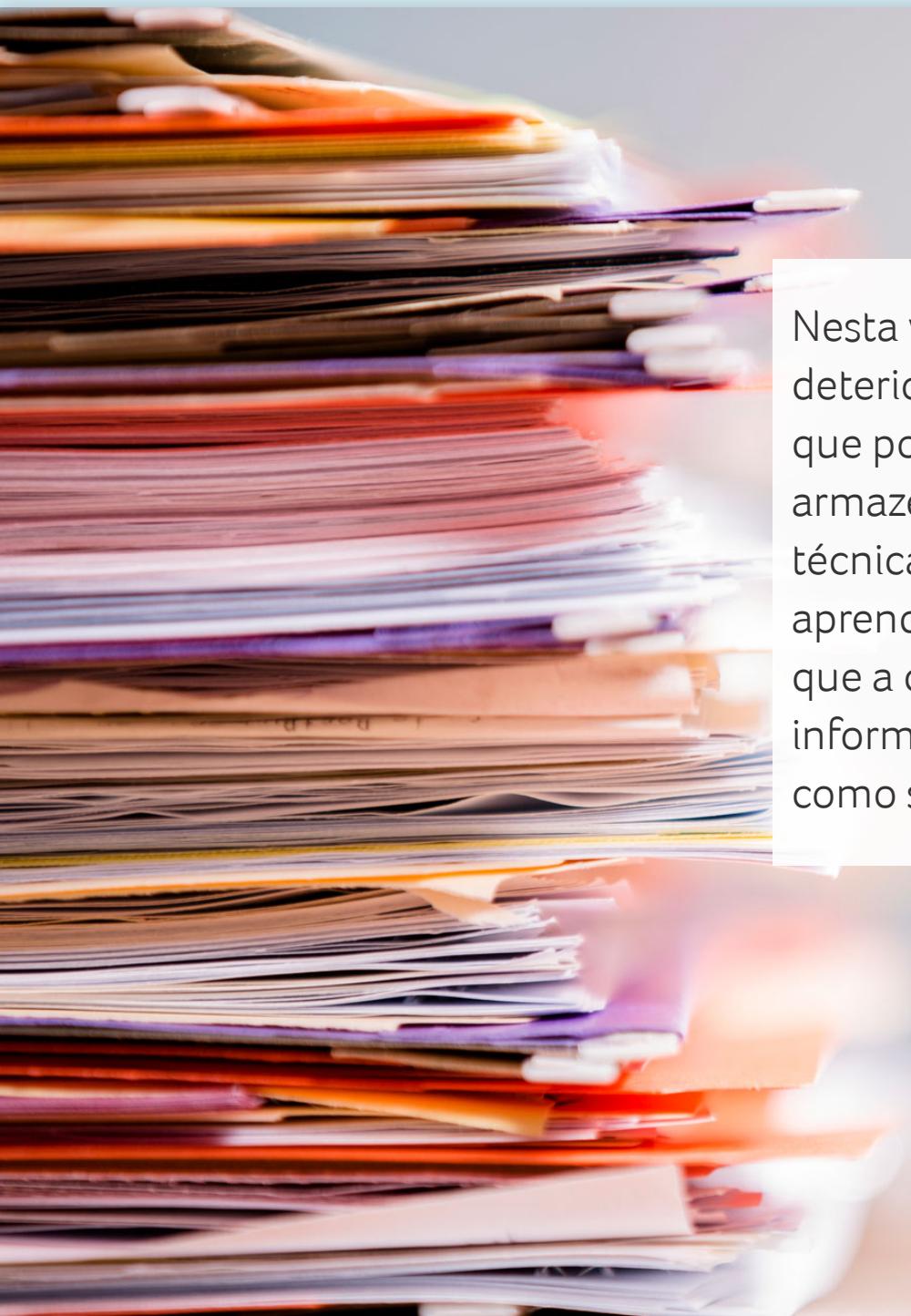
Unidade 3

Seção 3



Webaula 3

Otimização de tabelas de Espalhamento.



Nesta webaula, você poderá identificar por que ocorre a deterioração de desempenho em tabelas de espalhamento, que pode comprometer o desempenho na busca e armazenamento de informações. Irá conhecer como utilizar técnicas na otimização de desempenho em uma tabela, aprender como realizar a redução de colisões, em casos em que a colisão pode ocorrer em armazenamento de informações, e melhorar o espalhamento em tabelas, bem como suas aplicações na solução de problemas.



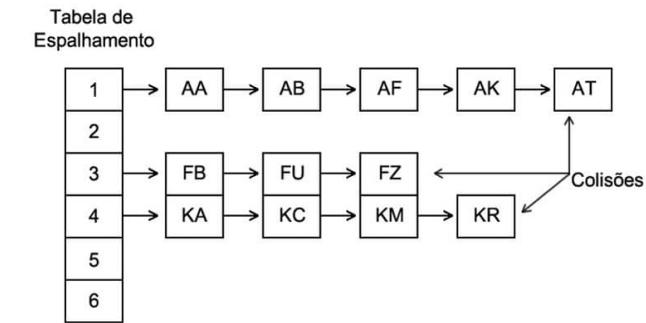
Deterioração de desempenho em tabelas de espalhamento

As Tabelas de Espalhamento podem sofrer com a deterioração do seu desempenho, causado principalmente pelo acúmulo de elementos com a mesma chave calculada pela Função de Espalhamento, o que chamamos de colisão.

Segundo Celes, Cerqueira, Rangel (2004), quanto mais colisões ocorrerem em uma Tabela de Espalhamento, pior será o desempenho de busca ou armazenamento de informações pela Função de Espalhamento, apesar ainda de ser aceitável para a função.



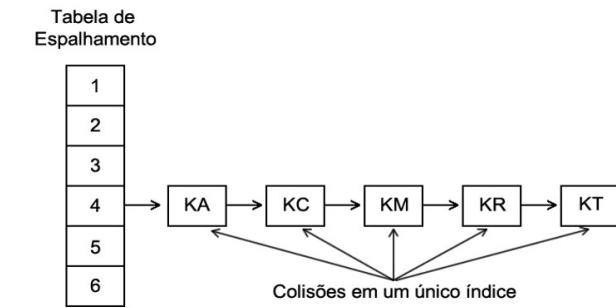
Na imagem, podemos observar que os índices 1, 2 e 3 da Tabela de Espalhamento possuem mais de um elemento armazenado em sua estrutura, necessitando, assim, de uma estrutura do tipo Lista Ligada para armazenar os elementos, enquanto os índices 2, 5 e 6 não possuem nenhum elemento armazenado.



Fonte: elaborado pelo autor.

Apesar de ser aceitável essa função, quanto mais elementos possuir um índice, maior será o esforço de busca para encontrar o elemento desejado.

Existe ainda a situação denominada de pior caso, onde todas as chaves da Função de Espalhamento são direcionadas para um único índice da tabela.



Fonte: elaborado pelo autor.



Técnicas de otimização em tabelas de espalhamento

Como você já aprendeu que, para evitar que colisões ocorram no cálculo de endereço de uma chave, pela Função de Espalhamento, a única forma é saber, de início, quais as chaves possíveis para inserção, criando uma Função de Espalhamento perfeita.

Conforme Drozdek (2016), existem diversas técnicas de otimização para funções, no entanto as três formas mais utilizadas são:

- Endereçamento Fechado;
- Endereçamento Aberto;
- Encadeamento Separado.



Segundo Drozdek (2016), no endereçamento fechado, a posição de inserção não muda. Todos devem ser inseridos na mesma posição, por meio de uma Lista Ligada em cada índice da tabela, função que vimos até o momento.

Para Drozdek (2016), no endereçamento aberto, elementos são inseridos diretamente nas posições calculadas pela Função de Espalhamento. Quando uma colisão ocorre, o registro que colide será inserido na próxima posição livre da tabela de espalhamento.



Segundo Celes, Cerqueira, Rangel (2004), o endereçamento aberto pode ser definido em duas formas diferentes, em:

- Função de Espalhamento Linear, que consiste em procurar a próxima posição vazia depois do endereço-base da chave;
- Função de Espalhamento Duplo, ao invés de incrementar a posição em 1 (um), uma Função de Espalhamento auxiliar é utilizada para calcular o incremento.

Já no Encadeamento Separado, ao invés de indexar um registro diretamente para o índice calculado, com base em sua chave pela Função de Espalhamento, o registro é indexado em uma estrutura de dados suplementar, tal como uma lista encadeada. Quando uma colisão ocorre, o registro que colide é inserido na lista encadeada.





Função de Espalhamento Linear

Segundo Silva (2007), a função linear, também conhecida como *Overflow Progressivo*, consiste em procurar a próxima posição vazia depois do índice base da chave. É uma função que possui como vantagem a simplicidade em definir o índice da Tabela de Espalhamento. No entanto, sua desvantagem está no caso de ocorrerem muitas colisões. Pode ocorrer um agrupamento (*clustering*) de chaves em um determinado índice.



Função de Espalhamento Duplo

Conhecido também como *re-hash*, a Função de Espalhamento Duplo, ao invés de incrementar a posição do elemento até a próxima posição vazia, utiliza uma função de espalhamento auxiliar para calcular qual o incremento que será dado à posição, levando em consideração o valor da chave.



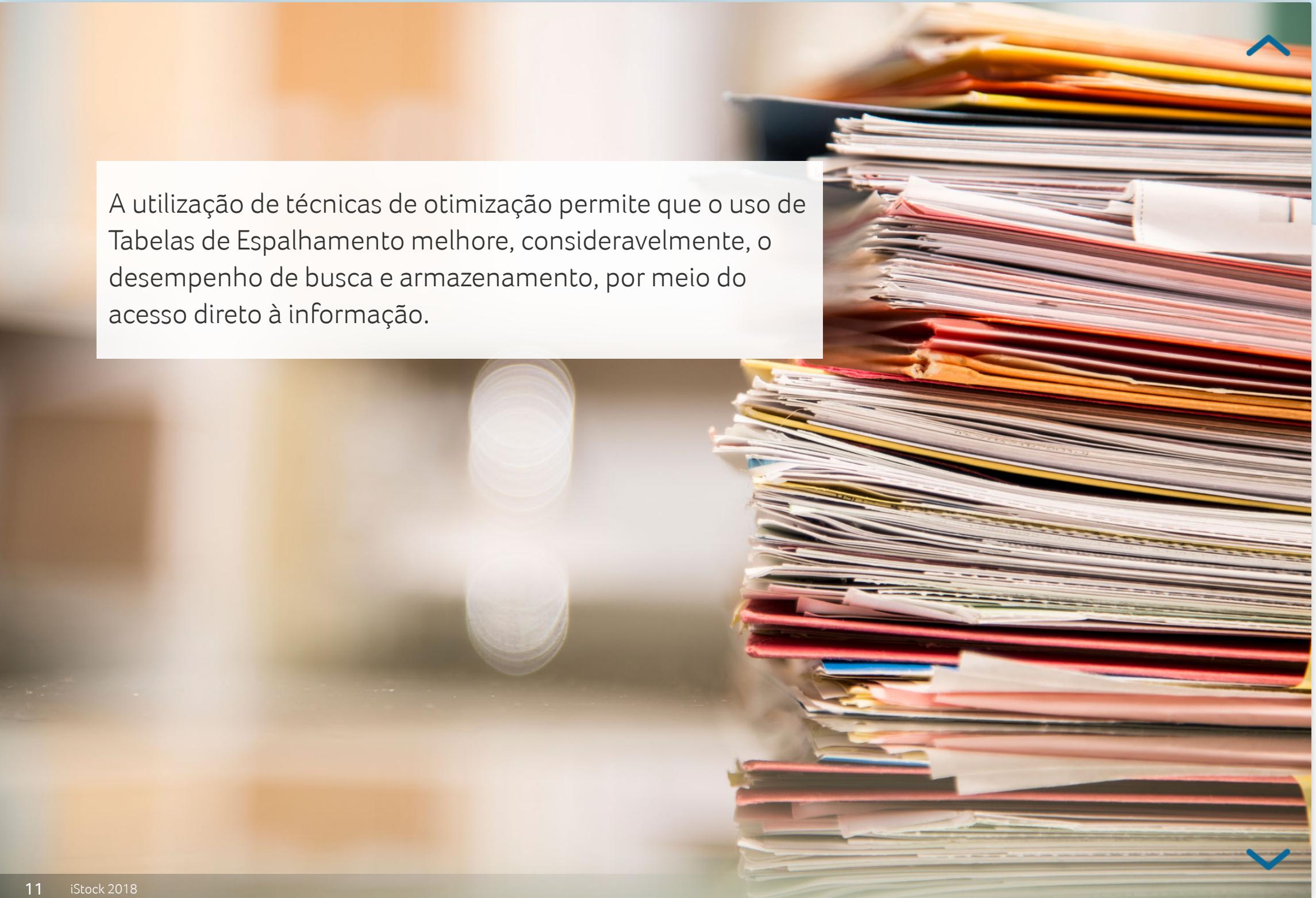
Melhorando o espalhamento

O desempenho da tabela diminui devido à falta de posições disponíveis, ocasionando o problema lentidão e uma solução seria aumentar a tabela e reinserir todos os elementos novamente, operação ineficiente para ser executada.

Conforme Drozdek (2016), para melhorar o espalhamento, uma solução seria a utilização de tabelas dinâmicas na estrutura, ao invés de utilizar uma Lista Ligada para armazenar os valores, podendo ser utilizada tanto com a Função de Espalhamento Linear, como com a Função de Espalhamento Duplo.

Assim, quando utilizamos este método, no momento em que a tabela passa de $N/2$ elementos, dobramos o seu tamanho, fazendo com que a tabela sempre tenha menos da metade dos elementos ocupados.





A utilização de técnicas de otimização permite que o uso de Tabelas de Espalhamento melhore, consideravelmente, o desempenho de busca e armazenamento, por meio do acesso direto à informação.



Vídeo de encerramento





Você já conhece o Saber?

Aqui você tem na palma da sua mão a **biblioteca digital** para sua **formação profissional**.

Estude no celular, tablet ou PC em qualquer hora e lugar sem pagar mais nada por isso.

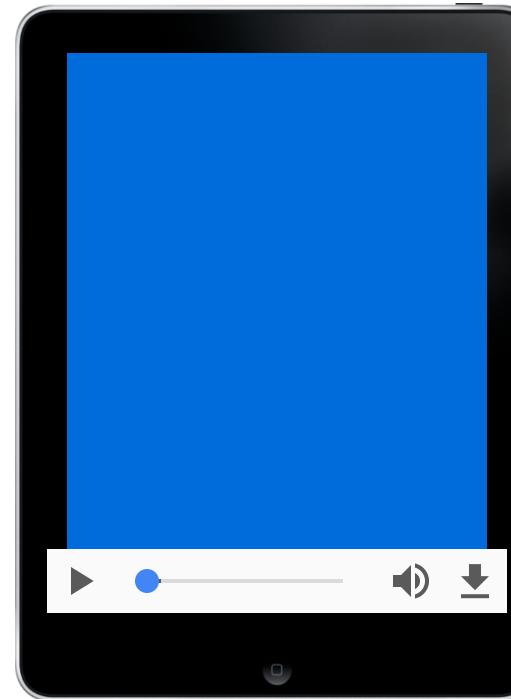
Mais de 450 livros com interatividade, vídeos, animações e jogos para você.



Android:
<https://goo.gl/yAL2Mv>



iPhone e iPad - IOS:
<https://goo.gl/OFWqcq>





Bons estudos!

