



## ESTRUTURA DE DADOS

### Projeto Índice Remissivo

**Professor:** Túlio Ribeiro

**Avaliação AV3 - Pontuação Máxima: 7,0**

#### 1. Justificativa do Projeto

Muitas aplicações necessitam de um relatório de referências cruzadas. Por exemplo, a maioria dos livros apresenta um índice remissivo, que é uma lista alfabética de palavras-chave ou palavras relevantes do texto, com a indicação dos locais no texto onde cada palavra-chave ocorre. Para ilustrar, considere um arquivo contendo o seguinte texto:

*“Good programming is not learned from generalities, but by seeing how significant programs can be made clean, easy to read, easy to maintain and modify, human-engineered, efficient, and reliable, by the application of common sense and by the use of good programming practices.”*

Assumindo que o índice remissivo seja constituído pelas palavras-chave:

*programming, programs, easy, by, human-engineered, and, be, to*

O programa para criação do índice deve produzir a seguinte saída:

```
and 4 5 6
be 3
by 2 6 7
easy 3 4
human-engineered 5
programming 1 7
programs 3
to 3 4
```

Note que a lista de palavras-chave está em ordem alfabética. Adjacente a cada palavra, há uma lista de números representando as linhas em que a palavra ocorre no texto.

Uma estrutura de dados desse tipo é conhecida como arquivo invertido, que é um mecanismo amplamente utilizado em sistemas de busca textual, como os motores de busca na Web. Ele permite localizar rapidamente todas as ocorrências de uma palavra em um conjunto de documentos, sendo essencial para a eficiência e eficácia de buscas em grandes volumes de texto

## 2. Descrição do Produto do Projeto

O objetivo deste projeto é desenvolver um programa para produzir um índice remissivo a partir de um texto e das palavras-chave fornecidas. O programa deve seguir os seguintes passos:

1. Ler um texto de tamanho arbitrário;
2. Processar as palavras do texto para armazená-las, juntamente com suas ocorrências (linhas do texto), nas estruturas de dados apropriadas;
3. Após as estruturas de dados estarem devidamente preenchidas com as palavras e suas ocorrências no texto, o programa deve ler um número arbitrário de palavras-chave que formarão o índice remissivo;
4. Gerar o índice remissivo.

Implemente um método eficiente para verificar se uma palavra lida do texto pertence ou não ao índice. **Para resolver esse problema, devem ser utilizadas as seguintes estruturas de dados principais**

- **Árvore Binária de Busca (ABB);**
- **Tabela de Dispersão (Hash);**
- **Listas encadeadas.**

**Tabela 1 - Vantagens e Desvantagens das Estruturas de Dados**

	<b>ABB</b>	<b>Hash</b>
<b>Vantagem</b>	Facilita a impressão das palavras em ordem alfabética após a atualização do índice remissivo.	Proporciona uma busca rápida, tornando o processo de verificação de palavras mais eficiente.
<b>Desvantagem</b>	Pode ser menos eficiente em termos de tempo de busca e inserção comparado ao hash, especialmente em grandes volumes de dados.	A impressão das palavras em ordem alfabética é problemática, necessitando uma etapa adicional de ordenação após a atualização do índice remissivo.

### **3. Estratégias para o Desenvolvimento do Projeto**

- Todas as dúvidas acerca dos requisitos do projeto devem ser direcionadas ao professor. Utilizar as horas de atendimento do professor e os fóruns de discussão no AVA (Moodle) para obter esclarecimentos necessários;
- O código fonte do projeto deve ser enviado através do AVA (Moodle) até a data limite estipulada;
- É fundamental estar atento aos prazos de entrega, pois não serão aceitas submissões após o prazo final;
- Na data agendada para a apresentação, o código fonte deve estar disponível no AVA para que o professor possa acessá-lo;
- Durante a apresentação, o professor executará e testará o programa na presença dos alunos.
- Os alunos serão questionados pelo professor sobre o código fonte, sendo necessário entender e explicar todas as partes do código.
- O código fonte do projeto será submetido a uma ferramenta de verificação de plágio.
- Qualquer tentativa de copiar o projeto de outro aluno, da Internet ou de qualquer outra fonte, resultará na atribuição de nota ZERO.
- A originalidade e a autoria do código são fundamentais. Mesmo pequenos trechos de código copiados sem a devida atribuição serão considerados fraude.
- Essas estratégias visam garantir que o projeto seja desenvolvido de forma correta e ética, promovendo a aprendizagem e a integridade acadêmica.

### **4. Principais Entregas do Projeto**

<b>Atividade</b>	<b>Requisitos</b>
<i>Entrada de Dados</i>	<ol style="list-style-type: none"><li>1. Ler um arquivo do tipo TXT contendo um número arbitrário de palavras-chave que constituirão o índice remissivo;</li><li>2. Ler um arquivo do tipo TXT contendo o texto a ser esquadrinhado em busca das palavras que pertençam ao índice remissivo.</li></ol>
<i>Geração do Índice Remissivo</i>	<ol style="list-style-type: none"><li>1. Criar o índice remissivo, incluindo as palavras-chave dentro do índice;</li><li>2. Utilizar um método eficiente para verificar se uma palavra lida do texto está contida no índice, utilizando três estruturas de dados: <b>Hash, ABB e Listas Encadeadas</b>;</li><li>3. Realizar a varredura dos elementos de forma</li></ol>

	<p>ordenada.</p> <p><b>Atenção:</b> Estas atividades incluem a utilização de Funções de Hash e a inserção de elementos nas estruturas (Lista, ABB e Hash).</p>
<i>Saída de Dados</i>	<p>1. Gerar um arquivo do tipo TXT contendo o índice remissivo.</p>

## 5. Estrutura Esperada do Projeto

1. Criar uma classe para as palavras, que terá dois atributos: a palavra (string) e a lista das ocorrências dessa palavra no texto (lista encadeada);
2. Criar uma Tabela Hash, na qual a chave será a primeira letra da palavra; ou seja, cada compartimento da Hash armazenará palavras que começam com a mesma letra.
3. Em cada compartimento da Tabela Hash, deverá haver uma Árvore Binária de Busca (ABB), cujos elementos armazenados em seus nós serão da classe palavra.
4. O código deve percorrer todo o texto e armazenar todas as palavras na estrutura. Caso o nó correspondente à palavra não exista, ele será criado, e o índice (número da linha) será acrescentado à lista encadeada da classe palavra. Caso o nó já exista, a nova ocorrência da palavra será adicionada à sua lista;
5. Por fim, o código deve ler o arquivo .txt com as palavras-chave, fazer a busca na estrutura desenvolvida e retornar um arquivo .txt de saída com o índice remissivo.

## 6. Avaliação

<b>Atividade</b>	<b>Pontuação</b>
<i>Entrada de Dados</i>	0,50
<i>Geração do Índice Remissivo</i>	Hash
	ABB
	Lista Encadeada
<i>Saída de Dados</i>	1,00
<i>Tratamento de Exceções</i>	1,00
<b>Total</b>	<b>7,00</b>

- O projeto deve ser realizado utilizando as estruturas de dados (Hash, ABB e Lista Encadeada) implementadas por vocês. Isso significa criar um TAD (Tipo Abstrato de Dados) para cada uma das estruturas, juntamente com seus respectivos métodos;
- O texto contido no item 1 deste documento pode servir como teste para o programa, mas também será fornecido um texto maior e mais complexo para uma avaliação mais robusta do programa.

## 7. Restrições do Projeto

- O projeto deve ser apresentado e entregue até a data limite disponível no AVA;
- O projeto deve ser desenvolvido na linguagem **Java**;
- O projeto deve ser realizado em **dupla ou individualmente**;
- Todas as estruturas de dados devem ser implementadas pela equipe. A utilização de bibliotecas do próprio Java para as estruturas de dados descharacteriza a entrega do projeto;
- **Estruturas de Dados Obrigatórias:** Tabela de Dispersão (Hash), Árvore Binária de Busca (ABB) e Listas Encadeadas;
- Entregar apenas o código fonte **não** garante a pontuação do projeto;
- **As equipes devem participar da arguição do professor, onde serão questionadas sobre o código fonte e seu funcionamento;**
- Essas restrições garantem que o projeto seja realizado conforme os requisitos estabelecidos, incentivando a compreensão profunda e a implementação das estruturas de dados essenciais, além de assegurar a originalidade e a autoria do trabalho.