



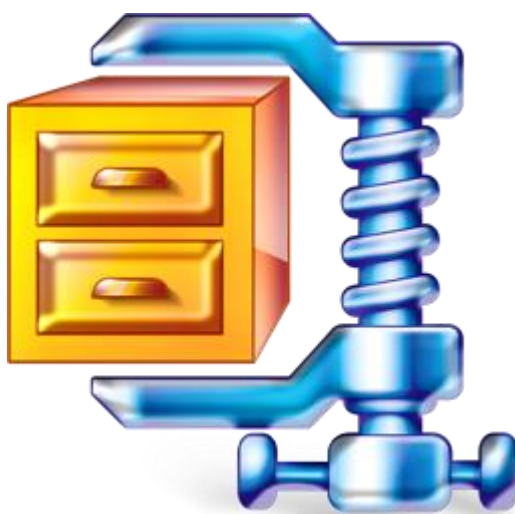
ESTRUTURA DE DADOS (T390/T923)	
Nome do Projeto: Compactador de Arquivos	
TERMO DE ABERTURA DO PROJETO	
Professor: Gilson Pereira do Carmo Filho	Semestre: 2021.1
Avaliação: AV3	Pontos: 7

## 1. Justificativa do projeto

Compactadores de arquivos são aplicativos que geram uma representação mais eficiente de um ou vários arquivos dentro de outro único arquivo, de modo que ocupem menos espaço na mídia de armazenamento ou o tempo de transferência deles em uma rede seja reduzido.

Os compactadores foram muito utilizados no passado quando as mídias de armazenamento tinham preços elevados e era necessário economizar espaço para armazenamento. Atualmente o uso deles é mais voltado a transferência de arquivos pela Internet, a fim de reduzir a massa de dados a ser transferida pela rede.

Neste projeto, você deverá desenvolver um **compactador de arquivos**, ou seja, um programa capaz de compactar e descompactar dados armazenados em um arquivo.



## 2. Descrição do produto do projeto

A **compressão de dados** é o ato de reduzir o espaço ocupado por dados num determinado dispositivo. Essa operação é realizada através de diversos algoritmos de compressão, reduzindo a quantidade de bytes para representar um dado, sendo este uma imagem, um texto, ou um arquivo qualquer.

Existem dois tipos principais de compressão de dados: com perdas e sem perdas. A compressão de dados **com perdas** é aquela na qual os detalhes da informação são perdidos durante a descompactação. Geralmente é usada para compactar áudios e vídeos na Internet, pois, nestes casos, a perda de precisão é benéfica e não afeta o entendimento da informação. Um exemplo disso, é a perda de qualidade de um vídeo disponibilizado online para evitar que ele demore a carregar caso a conexão com a Internet não esteja boa. Alguns exemplos de algoritmos que se baseiam na compressão de imagens com perdas são o JPEG, MP3 e MP4. Já para o uso de animações, alguns dos algoritmos existentes são Flash, VC-1 e H.261.

A compressão de dados **sem perdas** é aquela na qual o dado permanece o mesmo depois de descompactado. Geralmente é usada na compactação de documentos, trabalhos acadêmicos e diversos outros tipos de informações que devem permanecer inalteradas após a descompactação. Para se obter a compressão sem perdas, geralmente são usadas duas estratégias: a codificação de redundância mínima ou o método do dicionário.

A **redundância mínima** consiste em representar os símbolos que aparecem com mais frequência utilizando menos bits. Um exemplo de algoritmo que utiliza essa técnica é o Código de Huffman. Já o **método do dicionário** utiliza um arquivo como se fosse um dicionário de expressões com os seus valores correspondentes em bits. Essa estratégia é utilizada pelo algoritmo Lempel-Ziv-Welch (LZW).

Os **compactadores de arquivo** utilizam algoritmos de compressão de dados sem perdas para gerar a representação mais eficiente, combinando diversas técnicas conhecidas para um melhor desempenho.

Cada esquema de compressão gera um formato próprio de arquivo compactado que só pode ser descompactado pelo mesmo compactador que o gerou, ou então por outro compactador que também seja capaz de compreender o mesmo esquema.

O compactador de arquivos a ser desenvolvido neste projeto deverá utilizar a técnica de redundância mínima conhecida como **Código de Huffman**.

### 3. Principais entregas do projeto

As principais entregas do projeto são:

ENTREGA	REQUISITOS
Frequência	<ul style="list-style-type: none"><li>Receber do usuário, como entrada de dados, o nome de um arquivo do tipo texto (TXT) e a operação a ser realizada com o arquivo (compactar ou descompactar);</li><li>Realizar a contagem do número de ocorrências (frequência) de cada caractere do arquivo de texto informado.</li></ul>
Fila de prioridade	<ul style="list-style-type: none"><li>Inserir os caracteres e suas respectivas frequências em uma <b>fila de prioridade</b>, usando a frequência como prioridade.</li></ul>
Código de Huffman	<ul style="list-style-type: none"><li>A partir da fila de prioridade, gerar a árvore de Huffman, ou seja, uma árvore binária contendo todos os caracteres do arquivo e suas respectivas frequências;</li><li>Gerar uma tabela de códigos a partir da árvore de Huffman;</li><li>Codificar/decodificar o texto usando a tabela de Huffman;</li><li>Gerar, como saída de dados, outro arquivo do tipo texto, contendo o resultado da codificação/decodificação.</li></ul>

### 4. Estratégia de condução do projeto

- Dúvidas acerca dos requisitos do projeto deverão ser esclarecidas com o professor;
- O código fonte do projeto deverá ser enviado para o AVA (Moodle) até o prazo estipulado para a entrega. Não serão aceitas entregas após o prazo;
- Na data de apresentação dos projetos, o código fonte deverá estar disponível no AVA para o professor, que então executará e testará o programa na presença dos alunos. Estes, por sua vez, serão **arguidos pelo professor** sobre o código fonte;
- O código fonte do projeto será submetido a uma ferramenta de verificação de plágio. Qualquer tentativa de cópia do projeto de outro aluno ou da Internet, ou qualquer outra

tentativa de fraudar o projeto, incluindo cópia de trechos do código fonte, resultará em aplicação de nota **ZERO**.

## 5. Critério de avaliação do projeto

Na avaliação do projeto serão consideradas a execução correta das funcionalidades do programa e a conformidade do código fonte ao conteúdo abordado na disciplina.

A tabela a seguir mostra a pontuação das entregas do projeto.

ENTREGA	PONTOS
Frequência	2,0
Fila de prioridade	1,0
Código de Huffman	4,0
<b>TOTAL</b>	<b>7,0</b>

## 6. Premissas e restrições para o projeto

PREMISSAS	RESTRIÇÕES
<ul style="list-style-type: none"><li>O aluno deverá ter realizado as atividades práticas da disciplina e implementado as suas próprias classes das estruturas de dados.</li></ul>	<ul style="list-style-type: none"><li>O projeto tem um prazo total de 4 (quatro) semanas;</li><li>O programa deverá ser desenvolvido na linguagem <b>Java</b>;</li><li>O projeto deverá ser realizado em dupla ou individualmente.</li></ul>