

# **Relatório final – MI Programação**

**Tutor: Thiago Mariano**

**Vagner José dos Santos**

Vagnerjose01@gmail.com

## **1. Introdução**

Como primeiro produto de sua empresa, Monster Júnior resolveu implementar um software compactador/descompactador de arquivos o qual deu o nome de WINMonster. Esse programa deverá ter capacidades semelhantes as do WINZIP e do WINRAR. Para garantir que o WINMonster seja criado com êxito Monster Júnior utilizou o padrão MVC, verificou as possibilidades de exceções, criou testes de unidades e telas para o sistema com objetivo de otimizar a experiência do usuário.

O processo de entrada de dados nesse aplicativo se dá através da leitura de arquivos compactados ou descompactados. Feito isso é aplicado o algoritmo de Huffman para compactar ou descompactar, resultando em arquivos de saída compactados ou descompactados.

Durante esse processo é preciso executar algumas medidas para analisar se o arquivo continua íntegro. Uma delas é a aplicação de uma função hash com a finalidade de verificar se houve alteração de conteúdo no ato da descompactação.

## **2. Fundamentação Teórica**

Compactar um arquivo é o ato de reescrevê-lo de uma maneira mais inteligente para ocupar menos espaço em disco. Geralmente, os programas compactadores de arquivos analisam o arquivo a ser compactado em busca de caracteres que se repetem muito e geram um código que é armazenado em um tipo de dicionário. Então a ocorrência redundante é substituída por uma consulta na referencia do código, diminuindo assim a ocorrência de repetições. Com isso o arquivo resultante se torna muito menor que o original.

O processo de descompactação o inverso da compactação, o software acessa o dicionário do arquivo compactado e um mapa (gerado no processo de compactação) que serve como um guia para gerar o arquivo original, decodificando os caracteres com base no dicionário á medida em que sua frequência de aparição é descoberta.

Após a descompactação é necessário verificar a integridade do arquivo através da aplicação de uma função hash. Uma função hash é um algoritmo que mapeia informações de comprimento variável para dados de comprimento fixo.

Um programa compactador/descompactador de arquivos tem como finalidade tornar o armazenamento de arquivos mais eficiente, uma vez que eles passam a ocupar menos espaço em memória sem perder informações. Ele é usado frequentemente quando é

necessário manipular arquivos grandes para o processo ao qual se deseja realizar, ou com objetivo de economizar memória em dispositivos de armazenamento.

Um algoritmo de compactação é uma sequência de passos que são necessários para que o programa consiga reduzir o tamanho de um arquivo através da eliminação da repetição exagerada de dados que é comum em vários tipos de arquivos mas que, do ponto de vista do armazenamento em disco, é desnecessária.

Todos os aplicativos de compressão de arquivos possuem um algoritmo de compactação. Um dos algoritmos de compactação mais famosos é o Algoritmo de Huffman, que foi desenvolvido em 1952, a fim de diminuir o tempo de transmissão de mensagens, por David A. Huffman que, na época, cursava doutorado em Engenharia Elétrica no MIT, e foi publicado no artigo "A Method for the Construction of Minimum-Redundancy Codes".

A codificação de Huffman é um método de compressão que consiste na obtenção de códigos de tamanhos variados para cada caractere baseados na frequência dos símbolos (quantidade de vezes que os símbolos aparecem) presentes no conjunto de dados a serem comprimidos.

A eficácia do algoritmo está na definição dos códigos binários de menor comprimento aos caracteres mais frequentes e dos códigos binários de maior comprimento para os símbolos menos frequentes. Para isso, uma árvore binária, conhecida como Árvore de Huffman, é construída. Nela, as folhas contêm os símbolos presentes no arquivo e suas respectivas frequências. Das folhas em direção a raiz os nós são compostos pela soma das frequências dos nós filhos. Logo o nó raiz armazena a soma da quantidade de ocorrência de todos os caracteres do arquivo a ser compactado.

Com a árvore construída, cada uma de suas arestas são associadas a um dígito binário (0/1) com base no sentido (direita/esquerda). Sendo assim, o código de cada caractere é estabelecido pelo binário formado pelo conjunto de representações binárias das arestas da raiz até a folha correspondente a esses caracteres.

O gerenciamento das frequências antes de serem inseridas na árvore é feito em uma estrutura de dados ordenada, que pode ser uma fila de prioridade, por exemplo.

Fila de Prioridade é uma estrutura de dados capaz de armazenar informações em compartimentos por ordem de prioridade. Normalmente a o conjunto de informações armazenado no compartimento (nó) que possui a menos chave é o mais prioritário, logo é o primeiro elemento da fila. O tamanho da fila de prioridade é adequada a necessidade de armazenamento. As operações básicas e fundamentais em uma fila de prioridade é a inserção ordenada de dados, a verificação de tamanho, a remoção de informações no início da fila (maior prioridade).

É a partir dos dados da fila de prioridade que a árvore de Huffman é gerada, promovendo assim o processo de compactação que ainda inclui a gravação desses dados em um arquivo de saída de modo que realmente haja uma redução de necessidade de espaço para armazenamento.

### 3. Metodologia

O padrão MVC foi definido como forma de organização de código com objetivo de promover a flexibilidade em possíveis alterações de código futuras. Para o desenvolvimento do software, foi utilizada a linguagem de programação Java. Devido a portabilidade promovida pela presença da JVM (Java Virtual Machine) na plataforma Java, houve a liberdade de implementações do código do programa em diferentes sistemas operacionais.

Para alcançar a resolução do problema proposto, foram realizadas reuniões nas quais os itens questões, fatos, ideias e metas acerca do desenvolvimento da solução em código do software. Com isso, em cada reunião pôde-se responder às dúvidas estabelecidas nas sessões anteriores, definir novas ideias, concluir cada vez mais fatos sobre o problema e sua solução e traçar novas metas a serem cumpridas.

Um dos métodos usados para evitar o mal funcionamento do programa foi a utilização de testes unitários. Foram testadas várias funcionalidades do sistema afim de evitar possíveis erros de operação. Isso garante que ao finalizar a implementação do algoritmo, a quantidade de falhas sejam mínimas.

Um outro artifício que foi utilizado para evitar falhas no sistema foi a verificação de possíveis exceções. Entre elas algumas foram evitadas com a utilização de algumas técnicas e ferramentas de programação. Por exemplo, durante uma das reuniões foi discutida a possibilidade do usuário tentar acessar um arquivo que não existe. Foi discutido ainda se haveria um diretório específico para o programa ler e gravar os arquivos compactados/descompactados. A solução surgiu com a utilização de um recurso de interface gráfica responsável por filtrar e selecionar arquivos, navegando nos diretórios.

Além das reuniões com dias fixos durante a semana, encontros extras foram marcados, devido ao pouco tempo compreendido no prazo estabelecido para o término do processo de desenvolvimento. Essas sessões extras permitiram a obtenção de conclusões importantes sobre como fazer o programa.

Uma função hash foi utilizada para garantir que o arquivo que será descompactado é o mesmo arquivo de origem. Esse procedimento é muito importante para o êxito do funcionamento do programa, pois a característica mais importante do programa compactador/descompactador de arquivos é a capacidade de trabalhar sem perda de dados.

Para promover uma experiência agradável para o usuário ao usar o WINMonster, foi desenvolvida uma interface gráfica que permite a utilização interativa e abstrata das funcionalidades do sistema.

Essa Interface foi desenvolvida com recursos disponíveis em bibliotecas do próprio Java.

#### **4. Resultados e Discussões**

O WINMonster foi implementado após o período aproximado de 1 mês possuindo suas principais funções. Alguns teste de execução foram realizados sobre ele. Inicialmente não havia vantagem em utilizá-lo para compactar arquivos, pois o arquivo que deveria ser compactado estava ocupando mais espaço de memória que o arquivo original. Porém, após alguma observações, chegou-se a conclusão que se o método de gravação em arquivo fosse alterado, a compactação poderia passar a ser eficaz. Foi decidido então, utilizar arquivo binário como saída. Com isso, foi verificado que o arquivo de saída tem sido bem menor que o original.

Devido ao pouco período para desenvolver, junto com o tempo demandado para aprender a manipular os conceitos necessários, no fim do prazo estipulado para o término do desenvolvimento da aplicação o código não está totalmente completo. A função de descompactação do programa ainda precisa ser implementada, pois a versão inicial do código de descompactação não deu certo. Mas esse problema pode ser facilmente resolvido com um pouco mais de tempo.

Como o programa ainda não está totalmente pronto, não é viável tentar comparar a taxa de compactação do mesmo com outros programas do gênero disponíveis. Mas por questão de informação, a taxa de compactação do WINRAR é um pouco mais eficaz que que a taxa de compactação do WINZIP.

#### **5. Conclusão**

Na utilização do algoritmo de Huffman, quanto menor a variação dos elementos distintos, maior a taxa de compactação. O WINMonster apenas opera com arquivos de texto. A utilização do mesmo algoritmo para comprimir arquivos multimídias exigiria mais tratamentos em nível de código, combinação com outros algoritmos, e ainda assim haveria perda de dados como ocorre nos outros compactadores disponíveis no mercado. Com um pouco mais de implementação, o WINMonster estará funcionando como deveria. Uma ideia a ser considerada é a versão móvel do programa, já que o esse tipop de aplicativo é escasso nessa plataforma.

#### **6. Bibliografia Consultada**

<http://www.tecmundo.com.br/tecmundo-explica/54730-tecmundo-explica-funciona-compactacao-arquivos-video.htm> – Acesso em 09/04/2016

<http://www.cknow.com/cms/articles/what-is-file-compression.html> -Acesso em 09/04/2016

[https://pt.wikipedia.org/wiki/Codifica%C3%A7%C3%A3o\\_de\\_Huffman](https://pt.wikipedia.org/wiki/Codifica%C3%A7%C3%A3o_de_Huffman) -Acesso em 09/04/2016

[https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o\\_hash](https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o_hash) -Acesso em 09/04/2016