

# Relatório dos Trabalhos Práticos

Felipe Augusto Moraes Silva<sup>1</sup>

<sup>1</sup>Pontifícia Universidade Católica de Minas Gerais

famsilva@sga.pucminas.br

**Abstract.** *The following report was written with the objective of describing the assignments done during the class "Data Structures and Algorithms III", taught at Pontifícia Universidade Católica de Minas Gerais by Professor Hayala Curto.*

**Resumo.** *Este relatório tem como objetivo descrever as atividades práticas desenvolvidas durante a disciplina Algoritmos e estruturas de dados III, ministrada pelo Professor Hayala Curto, na Pontifícia Universidade Católica de Minas Gerais*

## 1. Introdução

Algoritmos executados em memória primária possuem um limite no que diz respeito a quantidade de armazenamento. Sabendo disso, faz-se necessário o uso de memória secundária para aumentar o máximo de memória que o algoritmo é capaz de utilizar.

## 2. Desenvolvimento

### 2.1. Trabalho Prático 01

No primeiro trabalho (tp01) foram trabalhadas técnicas com foco em pesquisa e armazenamento de arquivos, e a partir disso, foram desenvolvidos os seguintes algoritmos:

1. Arvore B+ em memória secundária
2. Lista invertida em memória secundária
3. Hash estendido em memória secundária

### 2.2. Trabalho Prático 02

No segundo trabalho (tp02) foram utilizados algoritmos de compressão por cima do tp02, sendo esses:

1. Huffman tree
2. LZW

### 2.3. Trabalho Prático 03

Por fim, foi trabalhada a idéia de criptografia e *String Matching* por meio dos algoritmos escolhidos pelo aluno:

1. KMP
2. Edit distance
3. Cifra de César

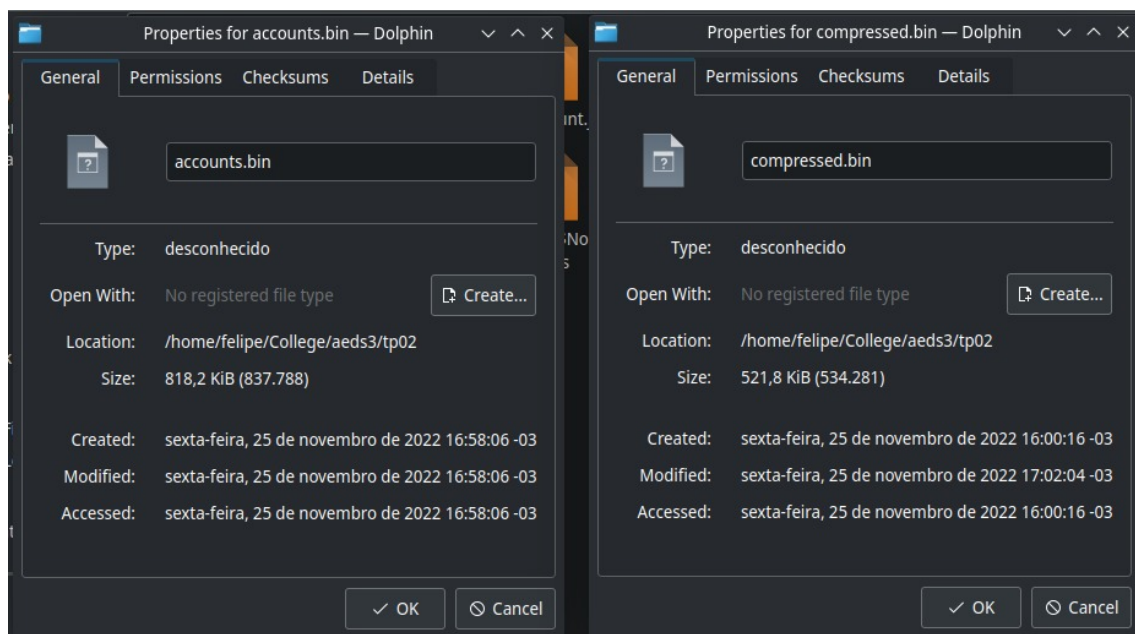
foram escolhidos os algoritmos que o aluno considerou mais fáceis de serem implementados

### 3. Testes e resultados

Foram realizados diversos testes com até 10.000 registros (com cada registro ocupando em torno de 80 bytes) em cada um dos tps. Além disso, também foi feita a análise de complexidade de todos os algoritmos implementados.

#### 3.1. Trabalho Prático 02

No trabalho prático 02 em especial, percebeu-se a não eficiência quanto ao uso de strings para a representação de cada *bit* de um *byte*, visto que apesar da implementação possuir uma complexidade de tempo  $O(n \log n)$ , sendo N o tamanho de registros a serem operados, o uso de String para representar cada *bit* expande consideravelmente o tamanho do espaço N a ser operado, visto que, para um *byte* X e um *bit* Y, a relação " $X = 8 * Y$ " é sempre verdadeira.



**Figura 1 - Tamanho do arquivo antes e depois da compressão**

### 4. Conclusão

Face ao exposto, conclui-se que a existência de algoritmos que atuam em memória secundária é de extrema importância, visto que nem todos os problemas podem ser resolvidos apenas com o uso da memória primária. Vale também ressaltar que é necessário um olhar mais atento à complexidade do espaço, visto que a falta de cuidado nesse aspecto, pode ser tão desastrosa quanto a falta de cuidado quanto a análise de tempo.