

Modelo de artigo do Instituto de Ciências Exatas e de Informática*

Article template Institute of Mathematical Sciences and Informatics

Caio Ronan Horta¹

Resumo

Nosso objetivo é desenvolver um programa capaz de realizar as operações básicas de um CRUD, implementar uma lista invertida e possibilitar o armazenamento de dados em diferentes formatos. Além disso, o programa deve incluir funcionalidades como compressão, criptografia e casamento de padrões.

Palavras-chave: CRUD, criptografia, compressão, casamento de padrão

*Artigo apresentado ao Instituto de Ciências Exatas e de Informática da Pontifícia Universidade Católica de Minas Gerais como trabalho de AED3.

¹Aluno de Ciência da Computação, Brasil – caiorhorta@gmail.com.

Abstract

Our goal is to develop a program capable of performing the basic CRUD operations, implementing a reversed list, and enabling data storage in different formats. Additionally, the program should include features such as compression, encryption, and pattern matching.

Keywords: **CRUD, encryption, compression, pattern matching.**

1 INTRODUÇÃO

Este projeto foi proposto para ser desenvolvido ao longo do semestre, sendo dividido em várias etapas. Inicialmente, a tarefa consiste na implementação de um CRUD (Create, Read, Update, Delete) de arquivos, permitindo o tratamento da base de dados escolhida. A abordagem inclui a capacidade de armazenar os dados em formato binário para otimizar o uso de espaço, além de possibilitar a reorganização da tabela por meio de ordenação externa.

Além disso, o sistema deve suportar a manipulação de arquivos utilizando estruturas como árvores B, B+, e B* para otimização da busca e inserção. A implementação deve também contemplar o suporte a multilistas e listas invertidas, ampliando as opções de organização e acesso aos dados, e foi necessário incorporar funcionalidades de compressão, criptografia e busca de padrões na base de dados.

Dessa forma, a proposta engloba uma abordagem abrangente para o desenvolvimento de um sistema de gerenciamento de dados, abordando desde operações básicas de CRUD até estruturas mais avançadas, como árvores e técnicas de otimização, segurança e busca de padrões.

2 DESENVOLVIMENTO

O projeto teve início a partir das aulas ministradas, com a implementação dos conceitos abordados nos conteúdos estudados. Inicialmente, foi desenvolvido como um CRUD convencional, com a particularidade de manipular arquivos em bits, conforme ensinado em aula e com o suporte dos vídeos do Professor Cutova.

Posteriormente, foi comunicada a necessidade de incorporar a ordenação externa ao projeto, além de armazenar os dados em arquivos organizados, visando evitar uma leitura sequencial. Esse novo requisito seguiu o mesmo modelo previamente mencionado. A implementação foi realizada para a entrega da primeira parte do projeto, denominada "TP1", incluindo a criação da lista invertida.

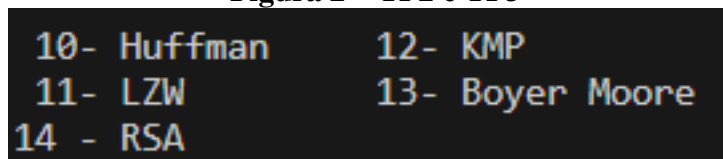
Figura 1 – CRUD

1- C reate	5- O rneção externa
2- R ead	6- H ashing estendido
3- U pdate	7- A vore B+
4- D elete	8- L ista Invertida

Iniciando o TP2, foi necessário incorporar a implementação de algoritmos de compressão, tais como Huffman e LZW, além de técnicas de casamento de padrões, a exemplo de LZW e Boyer-Moore. Essas implementações foram integradas ao sistema CRUD, visando evitar conflitos com os diversos arquivos gerados. Como parte do requisito, solicitou-se a avaliação da taxa de compressão e a medição do tempo de execução de cada algoritmo desenvolvido, com o objetivo de identificar possíveis áreas de otimização e realizar comparações entre eles.

Para concluir, no TP3, foi sugerida a implementação de um algoritmo de criptografia, sendo escolhido o RSA. Este algoritmo utiliza fórmulas baseadas em números primos, potenciação e operações de módulo para criptografar mensagens.

Figura 2 – TP2 e TP3

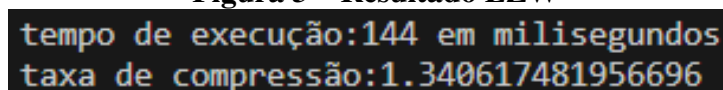


```
10- Huffman      12- KMP
11- LZW          13- Boyer Moore
14 - RSA
```

3 TESTES E RESULTADOS

O projeto demonstrou a capacidade de produzir resultados satisfatórios, especialmente em suas funções CRUD. Destaco que a estrutura fornecida permite um tratamento eficaz dos dados, possibilitando a comparação de diferentes métodos de compressão e casamento de padrões. Quanto à taxa de compressão, observou-se uma redução de 34% no tamanho do arquivo ao utilizar o algoritmo LZW, resultando em uma taxa de compressão de 1,34. Vale ressaltar que, embora tenha sido mencionada a implementação do algoritmo de Huffman, não foram fornecidos detalhes para comparações. Recomendaria incluir informações sobre o desempenho ou razões para a não implementação do algoritmo Huffman.

Figura 3 – Resultado LZW



```
tempo de execução:144 em milisegundos
taxa de compressão:1.340617481956696
```

Notavelmente, o algoritmo de Booyer Moore apresentou um desempenho superior ao algoritmo KMP, requerendo apenas 15.708 comparações para a palavra "SUV", enquanto o KMP necessitou de 43.183 comparações.

4 CONCLUSÃO

A implementação de algumas partes do projeto ainda é necessária, mas já tem o essencial para a utilização, apresentando pelo menos um artefato de cada tipo. Destaco que é possível realizar um tratamento eficaz dos dados com a estrutura entregue e permitir análises comparativas entre diferentes abordagens de compressão e casamento de padrões, com resultados específicos como do Booyer Moore e a taxa de compressão alcançada pelo LZW.