

Relatório Trabalhos Práticos AEDS 3

Gabriel Azevedo Fernandes¹, João Victor de Moura Teramatsu²

¹Pontifícia Universidade Católica de Minas Gerais (PUC-MG)
Belo Horizonte – MG – Brazil

²Ciência da Computação – Pontifícia Universidade Católica de Minas Gerais (PUC-MG)

Resumo. *O trabalho prático (TP) de Algoritmos e Estrutura de Dados, envolve a implementação de técnicas que desenvolvemos conforme o andamento do semestre, como por exemplo a compreensão de dados e casamento de padrões. O trabalho foi feito com base no tema Pokemon e foi realizado em dupla, utilizando a linguagem Java. Os TP's foram divididos em três etapas, cada uma dessas etapas com propostas novas de implementação/técnicas mais um vídeo explicativo de 5 minutos abordando o que foi feito nos códigos.*

1. Introdução

Comentando de forma geral, os trabalhos práticos possui como objetivo, pôr em prática o nosso conhecimento nas aplicações e implementações de algumas técnicas de programação. Técnicas essas que foram ensinadas em sala de aula e proposta pelo professor, com o intuito de nós alunos, trabalharmos de acordo com as especificações dadas a partir do tema que foi escolhido por cada dupla, sendo assim avaliado a clareza do código e sua robustez. Nosso TP utiliza como base de dados CSV o tema Pokemon, além disso, procuramos seguir a risca os critérios de avaliação para seguir uma conduta melhor nos códigos, ou seja, deixa-lo comentado e de maneira clara tanto no próprio código quanto nas explicações.

2. Trabalho Prático 1 (TP1)

O trabalho prático 1 consistiu na escolha de um tema, que nesse caso foi utilizando a base de dados CSV do tema pokemon, a fonte dessa base foi disponibilizada através da plataforma Kaggle. Após a escolha da base de dados, implementamos o sistema que é responsável para fazer as operações de CRUD (create, read, update e delete) em um arquivo sequencial e o arquivo Main para se integrar com os códigos dos três trabalhos práticos, dessa forma podemos fornecer ao usuário uma interface para interagir com o algoritmo e visualizar todos os recursos disponíveis. Em nosso código implementamos técnicas de manipulação de dados, como Ordenação Externa para tratar grandes volumes de dados que não cabem na memória principal, Indexação usando Árvore B+, que é eficaz em operações de busca e atualização em bancos de dados, Hash e Lista Invertida para rápida recuperação de dados e busca por palavras em conjuntos de texto. Esses códigos foram e são fundamentais para criar e gerenciar uma base de dados eficiente, permitindo a representação de entidades com campos variados, como strings de tamanho fixo ou variável, datas, listas de valores e números inteiros ou flutuantes. Essas técnicas são importantes para o processamento e análise eficazes de grandes volumes de dados.

```
Arquivo descomprimido com sucesso!
Escolha uma operação:
1- Create
2- Read
3- Update
4- Delete
5- Compress
6- Decompress
7- Pesquisar Padrão
8- Sair
7
Digite o nome do Pokemon que deseja encontrar: Charmander
Padrão encontrado no índice 432
Use a heurística do mau caractere ou do sufixo bom? (1 - Sufixo bom; 2 - Mau caractere):
```

2.1. Testes e Resultados TP1

3. Trabalho Prático 2 (TP2)

Esse TP, Trabalho Prático 2 consistiu na implementação e integração de algoritmos mais avançados de compressão de dados e casamento de padrões, aplicados na nossa base de dados. O objetivo principal nessa etapa trabalho é demonstrar a aplicação prática desses algoritmos em um contexto real, fazendo com que tenhamos uma compreensão mais profunda dos conceitos de algoritmos e estruturas de dados. Nesse trabalho, implementamos:

Compressão de Dados: Huffman e LZW. O algoritmo de Huffman foi usado para compressão de dados baseada em frequência de caracteres, otimizando o espaço de armazenamento ao codificar caracteres mais frequentes com bits mais curtos. E também usamos o LZW, baseado em dicionário, ele foi eficaz em compressão de dados de diversos tipos por se adaptar dinamicamente ao conteúdo sendo comprimido. Os dois métodos são implementados com capacidades de compressão e descompressão para garantir a recuperação completa dos dados originais.

Casamento de Padrões: Boyer-Moore e Knuth-Morris-Pratt (KMP). O Boyer-Moore tem uma abordagem eficiente que pula seções do texto que não correspondem ao padrão procurado, já o KMP usa um pré-processamento do padrão para evitar reexaminação do texto. Esses algoritmos foram aplicados na base de dados para localizar padrões específicos de forma mais eficiente.

3.1. Testes e Resultados TP2

```
String data = "";
System.out.println(x:"Digite o ID: ");
int readID = fetch.nextInt();
String lixo = fetch.nextLine();
if (crud.search(readID)) {
    // Personalizar para meu CSV e campos de Pokemon
    System.out.println(x:"Digite o nome do Pokemon: ");
    String pokemonName = fetch.nextLine();
    System.out.println(x:"Digite o número da pokedex: ");
    int pokedexID = fetch.nextInt();
    lixo = fetch.nextLine();
    System.out.println(x:"Escreva o número da geração: ");
    gen = fetch.nextLine();

    System.out.println(x:"Escreva o nome da especie de Pokemon: ");
    String specie = fetch.nextLine();

    System.out.println(x:"Escreva o nome da habilidade escondida: ");
    hiddenAbility = fetch.nextLine();
    // data
    System.out.println(x:"Digite a data de lançamento do Pokemon: ");
    System.out.println(x:"Ex.: yyyy-MM-dd");
    data = fetch.nextLine();
}
```

```
Arquivo da sequência compactada gerado: baselzwCompressao0.txt
Tempo de compressão LZW: 47ms
Compressão LZW foi 43,37% mais eficiente

Arquivo comprimido com sucesso!
Escolha uma operação:
1- Create
2- Read
3- Update
4- Delete
5- Compress
6- Decompress
7- Pesquisar Padrão
8- Sair
```

4. Trabalho Prático 3 (TP3)


Nesta etapa do trabalho, implementamos 2 tipos de criptografia, e dentro do contexto do nosso tema, decidimos implementar Cifra de César e a Cifra de Substituição. Ambos foram escolhidos devido à sua simplicidade e eficácia em cenários de criptografia básica.

Cifra de César: envolve deslocamento de cada letra do texto original por um número fixo de posições. No nosso sistema, implementamos uma variante da Cifra de César onde o deslocamento é definido por uma chave numérica. A implementação foi feita de forma que cada caractere do texto é deslocado conforme o valor da chave, resultando no texto cifrado.

Cifra de Substituição: Diferente da Cifra de César, a Cifra de Substituição substitui cada letra do texto original por outra letra. Neste projeto, cada caractere é mapeado para um caractere diferente conforme uma tabela de substituição predefinida. Essa abordagem oferece um nível maior de complexidade e segurança em relação à Cifra de César.

A integração dos algoritmos de criptografia foi realizada com foco no armazenamento seguro dos dados dos Pokémons. Decidimos aplicar a criptografia nos campos sensíveis dos registros de Pokémon, como nome, espécie e habilidades. Isso assegura que, embora os dados sejam armazenados de forma cifrada, eles podem ser descriptografados e apresentados em sua forma original quando acessados pelo sistema.

4.1. Testes e Resultados TP3



```
PROBLEMS 33 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Tempo de descompressão Huffman: 1158ms
Tempo de descompressão LZW: 29ms
Descompressão LZW foi 97,50% mais eficiente

Arquivo descomprimido com sucesso!
Escolha uma operação:
1- create
2- Read
3- Update
4- Delete
5- compress
6- Decompress
7- Pesquisar Padrão
8- Sair
```

5. Conclusão

Este relatório apresentou os Trabalhos Práticos (TPs) durante o semestre, focados no tema Pokémon. Ao longo dos três TPs, exploramos e implementamos diversas técnicas e algoritmos fundamentais, sempre com o objetivo de aprofundar nosso entendimento teórico e habilidades práticas.

No TP1, estabelecemos a base do nosso projeto com a criação de um sistema CRUD utilizando uma base de dados CSV sobre Pokémon. Implementamos técnicas como Ordenação Externa, Indexação com Árvore B+ e Hash, e Lista Invertida, que foram cruciais para o gerenciamento eficiente de grandes volumes de dados.

No TP2, avançamos para a integração de algoritmos mais complexos de compressão de dados e casamento de padrões. Os métodos de Huffman e LZW mostraram-se eficazes na compressão de dados, enquanto os algoritmos de Boyer-Moore e Knuth-Morris-Pratt (KMP) foram aplicados com sucesso para o casamento de padrões. Essas técnicas facilitaram e permitiram a manipulação de dados mais eficiente e uma busca mais rápida em nossa base de dados.

Finalmente, no TP3, focamos na implementação e integração de dois algoritmos de criptografia, a Cifra de César e a Cifra de Substituição, para garantir a segurança dos dados sensíveis dos registros de Pokémon. Essa etapa destacou a importância da criptografia em sistemas de informação e reforçou nossa compreensão dos princípios de segurança de dados.

Através destes trabalhos práticos, aprimoramos nossas habilidades em programação e estruturas de dados, e também desenvolvemos uma compreensão mais profunda desses conceitos e como podem ser aplicados em situações reais. Além disso, as técnicas e algoritmos implementados forneceram insights importantes sobre o processamento e análise eficazes de grandes volumes de dados, que são habilidades essenciais na nossa área.

Em resumo, este projeto foi combinou teoria e prática, proporcionando uma base sólida para futuros desenvolvimentos.