



Pontifícia Universidade Católica de Minas Gerais

Instituto de Ciências Exatas e Informática

Ciência da Computação

PresenteFácil 4.0: Sistema de Gestão de Listas e Visualizador de Estruturas de Dados

Relatório Técnico Integrado e Atividade Extensionista (TP1 -
TP4)

Autores:

Domynic Barros Lima

Felipe Rivetti Mizher

Matheus Felipe Cavalcanti Xavier

Orientador:

Marcos Andre Silveira Kutova

Belo Horizonte

2025“

Sumário

1	Introdução	2
2	Parte I: Backend e Persistência (TP1–TP3)	3
2.1	Visão Arquitetural e Design	3
2.2	Formato dos Registros no Arquivo	3
2.3	Implementações de Índices	3
2.3.1	Hash Extensível	3
2.3.2	Árvore B+	4
2.3.3	Índice Invertido e TF-IDF (TP3)	4
2.4	Telas representativas do sistema de console	5
3	Parte II: Visualizador Web (TP4)	6
3.1	Objetivo e Justificativa	6
3.2	Tecnologias e Arquitetura do Visualizador	6
3.3	Funcionalidades principais e fluxo de uso	6
3.4	Telas do Visualizador	7
4	Avaliação com usuários (detalhamento)	9
4.1	Roteiro de Teste	9
4.2	Instrumento de Avaliação	9
4.3	Resultados Consolidados	9
4.4	Observações Qualitativas	9
5	Conclusão	10
6	Links e Referências	10

1 Introdução

Este relatório integra os resultados dos Trabalhos Práticos TP1, TP2, TP3 e TP4 da disciplina Algoritmos e Estruturas de Dados III, ministrada pelo Prof. Marcos Andre Silveira Kutova. O projeto **PresenteFácil 4.0** consiste em: (i) um sistema de console em Java com persistência própria (TP1 - TP3) e (ii) um visualizador web (TP4) cujo objetivo é tornar observáveis, para fins pedagógicos, as estruturas de dados e os arquivos binários gerados pelo backend.

O documento apresenta arquitetura, decisões de projeto, detalhes de implementação das estruturas de dados (Hash Extensível, Árvore B+, Índice Invertido com TF-IDF), a ferramenta web extensionista e os resultados da avaliação com 10 alunos.

2 Parte I: Backend e Persistência (TP1–TP3)

2.1 Visão Arquitetural e Design

O sistema segue o padrão **MVC**, com módulos bem definidos:

- **Model:** entidades (`Usuario`, `Lista`, `Produto`, `ListaProduto`), classes de persistência (`Arquivo*.java`) e estruturas (`HashExtensivel`, `ArvoreBMais`).
- **View:** interface de console com menus e fluxos de interação.
- **Controller:** regras de negócio, validações, orquestração de operações.

A pasta `Dados/` é gerada automaticamente no diretório do TP em execução e contém os `.db` e arquivos de índice persistente.

2.2 Formato dos Registros no Arquivo

Os arquivos binários usam o formato:

1. **Lápide (1 byte):** 0 para excluído, 1 para ativo.
2. **Indicador de Tamanho (4 bytes, int):** tamanho do corpo em bytes.
3. **Corpo do Registro:** sequência de campos serializados (IDs em 4 bytes, `writeUTF` para Strings, etc.).

Esse formato permite leitura aleatória via `RandomAccessFile`, reaproveitamento de espaço e atualização de registros variáveis.

2.3 Implementações de Índices

2.3.1 Hash Extensível

Empregado para lookup por chaves únicas (e-mail de usuário, GTIN-13, código de lista). A estrutura contempla:

- diretório persistente com ponteiros para buckets;
- buckets com profundidade local;
- regras de split que atualizam localmente o bucket e, se necessário, duplicam o diretório.

2.3.2 Árvore B+

Utilizada para índices secundários e para os relacionamentos 1:N e N:N:

- índice 1:N (usuário → listas) implementado com `ArvoreBMais<ParUsuarioLista>`;
- relacionamento N:N implementado com duas árvores B+ sobre a entidade associação `ListaProduto`: uma indexando por `idLista` e outra por `idProduto`.

2.3.3 Índice Invertido e TF-IDF (TP3)

O índice invertido armazena, para cada termo normalizado, uma lista de pares $(idProduto, TF)$ persistida via B+. A busca utiliza:

1. pré-processamento dos termos (remoção de stop words, normalização sem acentos, lowercase);
2. recuperação das postagens para cada termo e multiplicação $TF \times IDF$ (IDF calculado dinamicamente como $\log(N/n_t) + 1$, onde N é o número total de produtos e n_t a ocorrência do termo);
3. soma dos scores por `idProduto` e ordenação decrescente para apresentação.

2.4 Telas representativas do sistema de console

As imagens abaixo demonstram aspectos relevantes do sistema offline (TP1 - TP3). As imagens foram importadas da pasta img/Parte 1 (TP1, TP2 e TP3)/.

```
Presente Fácil 3.0
-----
(1) Login
(2) Novo usuário
(S) Sair

Opção: 1
```

Figura 1: Menu inicial (Login / Novo usuário).

```
Presente Fácil 3.0
-----
(1) Login
(2) Novo usuário
(S) Sair

Opção: 2

Inclusão de usuário
Nome (min 4 chars, vazio p/ cancelar): Teste
Email (vazio p/ cancelar): teste@pucminas.com
Senha: 123456
Pergunta secreta (para recuperação): Qual a cor do céu?
Resposta secreta: Azul
Usuário cadastrado com sucesso.

Pressione ENTER para continuar...■
```

Figura 2: Fluxo de cadastro de usuário (validação e hash SHA-256).

```
PresenteFácil 3.0
-----
> Início > Minhas listas

LISTAS
(1) Compras para Casa - N/D
(2) Presentes de Anivers?rio - 17/04/2026
(3) Setup Gamer - N/D

(N) Nova lista
(R) Retornar ao menu anterior

Opção: ■
```

Figura 3: Visualização das listas do usuário, ordenadas alfabeticamente.

```
PresenteFácil 3.0
-----
> Início > Produtos
O
(1) Buscar produto
(2) Listar todos os produtos
(3) Cadastrar um novo produto

(R) Retornar ao menu anterior

Opção: 1

Digite o GTIN-13 ou as palavras-chave: 7891000000001■
```

Figura 4: Busca de produto por GTIN-13.

3 Parte II: Visualizador Web (TP4)

3.1 Objetivo e Justificativa

O TP4 tem caráter extensionista, a ferramenta foi concebida para facilitar a compreensão de conceitos abstratos da disciplina ao oferecer uma visualização direta das estruturas em disco (lápide, offsets, serialização), o que beneficia alunos que tenham dificuldades com a visão apenas textual ou teórica.

3.2 Tecnologias e Arquitetura do Visualizador

A aplicação é uma SPA desenvolvida em HTML, CSS e JavaScript (Vanilla). Principais escolhas técnicas:

- **LocalStorage:** simula persistência local, permitindo exportação/importação;
- **Conversor JSON ↔ Bytes:** implementa regras de serialização compatíveis com Java `writeUTF/writeInt`;
- **Importação/Exportação:** geração de `backup.json` e `dados.db` (binário didático);
- **UX:** modais, toasts, layout responsivo e painel lateral redimensionável para visualização do Hex Dump.

3.3 Funcionalidades principais e fluxo de uso

O visualizador suporta:

- CRUD de produtos;
- visualização do produto em JSON (lógico) e em Hex Dump (físico);
- geração de arquivo binário com mesma estrutura usada no backend Java;
- importação atômica com validação (tudo ou nada);
- ordenação e busca por GTIN ou por palavra-chave (simulação da busca TF-IDF aplicada localmente).

3.4 Telas do Visualizador

As imagens abaixo estão na pasta `img/Parte 2 (TP4)/` e ilustram a interface e funcionalidades.

The screenshot shows two main windows side-by-side. On the left is the 'Gestão de Produtos' (Product Management) interface, which is currently empty. It features a search bar, a sorting dropdown ('Ordenar: ID (Criação)'), and a table with columns: GTIN-ID, PRODUTO, DESCRIÇÃO, and GERENCIAR. A message at the bottom states 'Nenhum produto cadastrado.' (No products registered). On the right is the 'Simulador de Arquivo' (File Simulator) interface, which displays binary data in JSON and hex formats. It includes buttons for 'Baixar (.db)', 'Importar (.json)', and 'Limpar Tudo'.

Figura 5: Dashboard inicial vazio do Visualizador Web.

This screenshot shows a modal window titled 'Adicionar Produto' (Add Product) overlaid on the dashboard. The modal contains fields for 'GTIN-13' (set to '123456789101'), 'Nome do Produto' (set to 'Produto Teste A'), and 'Descrição' (set to 'Produto apenas para teste e demonstração.'). Below these fields are 'Cancelar' and 'Salvar Produto' buttons. In the background, the dashboard shows the same empty product list as in Figure 5.

Figura 6: Modal de cadastro de produto (validações e avatar automático).

This screenshot shows the 'Gestão de Produtos' interface with one item registered: '123456789101' (PRODUTO: 'Produto Teste A', DESCRIÇÃO: 'Produto apenas para teste e demonstração.'). To the right, the 'Simulador de Arquivo' interface shows the JSON representation of the registered product. The JSON code is:

```
[{"id": 1, "gtin": "123456789101", "nome": "Produto Teste A", "descricao": "Produto apenas para teste e demonstracao.", "imagem": "https://cdn-icons-png.flaticon.com"}]
```

Figura 7: Produto cadastrado — visualização lógica (JSON).

The screenshot shows a web-based application for managing products. On the left, there's a search bar and a table listing a single product entry:

GTIN-13	PRODUTO	DESCRIÇÃO	GERENCIAR
1234567891011	Produto Teste A	Produto apenas para teste e demonstração.	

On the right, there's a hex dump viewer titled "Simulador de Arquivo". It shows the binary representation of the product data. The interface includes tabs for "JSON (Lógico)" and "Binário (Físico)". Below the hex dump are buttons for "Baixar (.db)", "Importar (.json)", and "Limpar Tudo".

Figura 8: Produto cadastrado — visualização física (Binário / Hex Dump).



Figura 9: Arquivos exportados: backup.json e dados.db.

4 Avaliação com usuários (detalhamento)

4.1 Roteiro de Teste

O roteiro aplicado aos 10 avaliadores incluía:

1. Cadastrar um novo produto com GTIN-13 válido;
2. Visualizar o produto em JSON e em Hex Dump;
3. Editar o produto e observar atualização no Hex Dump;
4. Exportar backup (JSON e .db) e reimportar (validação atômica);
5. Realizar buscas por palavra-chave e por GTIN.

4.2 Instrumento de Avaliação

Foi utilizado um formulário com 6 perguntas em escala Likert (1–5) cobrindo facilidade de uso, clareza de feedback, estética, utilidade pedagógica, performance e satisfação geral. Dados brutos foram armazenados em .

4.3 Resultados Consolidados

As médias das respostas (1–5) estão apresentadas na Tabela abaixo.

Critério	Média
Facilidade de Uso	5,0
Clareza do Feedback	5,0
Estética e Organização	4,7
Utilidade Pedagógica	5,0
Performance	5,0
Satisfação Geral	5,0

Tabela 1: Médias de avaliação (Likert 1–5)

4.4 Observações Qualitativas

Comentários dos avaliadores enfatizaram a clareza da visualização binária e a utilidade didática para compreender o conteúdo de AEDS III. Pequenos apontamentos sobre estética indicam oportunidades de refinamento visual em futuras versões.

5 Conclusão

O projeto atingiu plenamente seus objetivos técnicos e pedagógicos. Tecnicamente, implementamos um sistema de persistência com índices avançados e um motor de busca baseado em TF-IDF. Pedagogicamente, entregamos uma ferramenta que facilita o aprendizado de conceitos complexos, validada por estudantes da própria graduação.

Para trabalhos futuros, sugerimos:

- Interface web com autenticação integrada ao backend (simulação remota);
- Refinamentos estéticos e responsividade avançada;
- Persistência em servidor para compartilhamento entre usuários.

6 Links e Referências

- Repositório do projeto: <https://github.com/mioj0kt/AEDsIII>
- Vídeos de demonstração:
 - TP1: <https://youtu.be/ZYMZ9jcaeK0>
 - TP2: <https://youtu.be/GzxbBbwchY0>
 - TP3: <https://youtu.be/TK6eKy3TGHQ>
 - TP4: <https://youtu.be/-0FDD6R3JvM>
- Material de apoio: Kutova, M. (Material da disciplina AEDS III).