Relatório Referente ao trabalho II da disciplina de Estrutura de Dados II
Autor: Mauricio Benjamin da Rocha
Dupla: Lazaro Claubert Souza Rodrigues Oliveira

Resumo do Projeto
O presente trabalho tem como objetivo a implementação dos sistemas solicitados para a disciplina de Estrutura de dados II utilizando as estruturas de dados conhecidas por "Árvores" em suas diferentes variações. Em complemento ao desenvolvimento realizou-se a validação através do comparativo de desempenho entre as estruturas visando avaliar quais se sobressaem em diferentes cenários.

1. Introdução

Estruturas de dados lineares, embora sejam fundamentais, podem enfrentar desafios significativos quando lidam com grandes volumes de dados. O principal problema surge da sua natureza sequencial, onde a eficiência de operações como busca, inserção e exclusão tende a diminuir à medida que o volume de dados aumenta. Por exemplo, em listas ou arrays, a busca linear exige a verificação de cada elemento, o que se torna cada vez mais custoso conforme a quantidade de dados cresce. Além disso, operações de inserção ou remoção podem exigir deslocamento de uma quantidade considerável de elementos, impactando negativamente o desempenho, especialmente em cenários com grandes conjuntos de dados.

As árvores são estruturas de dados fundamentais na computação devido à sua capacidade de representar hierarquias complexas de forma eficiente. Ao contrário das estruturas lineares como listas ou pilhas, as árvores permitem a organização e busca de dados de maneira hierárquica, o que é crucial em muitos contextos. Por exemplo, em bancos de dados, as árvores são utilizadas para representar índices, acelerando significativamente a busca e recuperação de informações. Além disso, em algoritmos de busca, como o algoritmo de busca em árvore binária, a eficiência no tempo de execução é notavelmente superior quando comparada a estruturas lineares, pois as árvores permitem uma redução significativa no número de comparações necessárias para encontrar um elemento específico.

O presente Projeto foi desenvolvido utilizando linguagem de programação C e estruturado em três pastas separadamente visando garantir uma maior organização do código fonte de forma a facilitar seu acesso, manutenção e atualização. A Figura 1 Apresenta visualmente como estão organizadas as pastas, onde a pasta Q1 guarda todos os arquivos da questão 01 do projeto, Q2 e Q3 guardam respectivamente as questões 02 e 03 do projeto.

Figura 1: Organização dos arquivos do Projeto.

O presente trabalho está organizado em seções de forma visando facilitar o acesso a informações específicas. As seções são seções específicas, resultado da execução dos programas e conclusão do projeto.

2. Seções Específicas

Esta seção visa apresentar de forma mais precisa como cada questão foi trabalhada, desde as "Árvores" usadas até a interação com o usuário. A experimentação realizada visando analisar o desempenho foi realizada em uma máquina com a configuração descrita na Tabela 1.

Tabela 01: Hardware usada para os experimentos

Processador	**Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz**
Memória RAM	**16,0 GB DDR4 2400 MHZ**
Sistema Operacional	**Linux Ubuntu 22.04.3 LTS**

2.1. Q1 - Biblioteca de Músicas com Árvore Rubro Negro

A problemática abordada em Q1 envolve o desenvolvimento de um sistema para uma biblioteca de músicas onde devemos guardar informações de artistas, álbuns e músicas proporcionando para o usuário opções como cadastro, consultas e a remoção de informação. As informações referentes a artistas e álbuns foram foram estruturadas em formato de Arvore usando a Arvore Rubro Negro, já as musicas seguiram o formato de uma lista duplamente encadeada ordenada de forma a seguir com todas as normas solicitadas.

A Figura 2 apresenta os diagramas usados para a modelagem das estruturas e suas respectivas conexões nos permitindo visualizar relações como dependências existenciais, conexões de zero para muitos e etc.

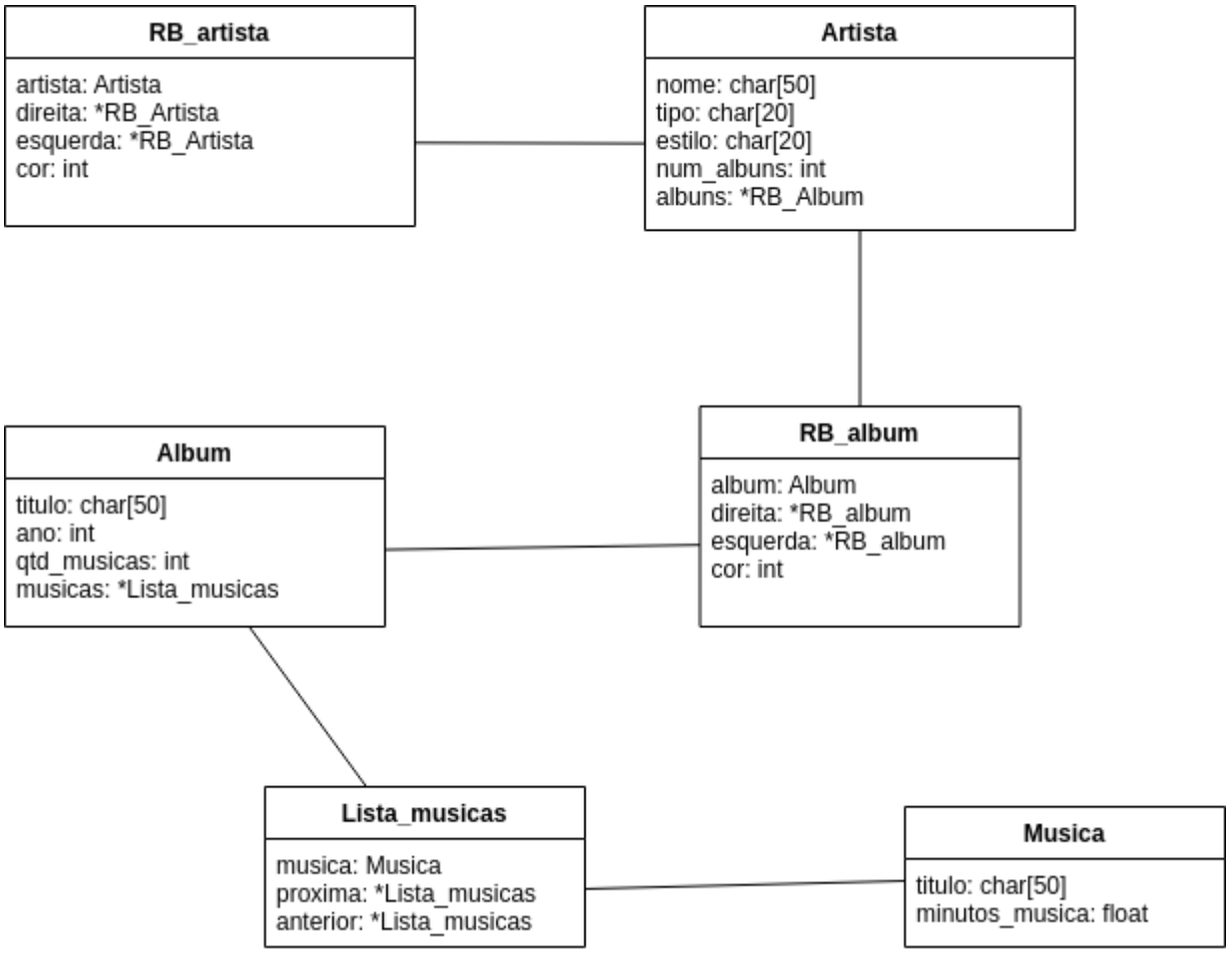

Figura 2: Modelagem das estruturas usadas pelo sistema.

Vale ressaltar que algumas regras de negócio não estão inclusas nas funcionalidades das estruturas mas devido a sua necessidade e impacto se tornam necessárias na ferramenta, sendos elas:

- Uma música só pode ser inserida em um álbum já cadastrado;
- Um álbum só pode ser inserido em um artista já cadastrado;
- Os dados devem ser organizados de forma alfabética;
- Para a remoção de um determinado álbum, lembrar ao usuário que todas as músicas daquele álbum serão removidas;
- Para remoção de um determinado artista, lembrar ao usuário que todos os álbuns e consequentemente todas as músicas daquele artista serão removidas.

A arvore rubro negro é uma estrutura de dados em forma de árvore binária que possuem propriedades especiais para manter o balanceamento e garantir operações eficientes. Cada nó de uma árvore rubro-negra é atribuído com uma cor, vermelho ou preto, e deve seguir algumas regras para preservar o balanceamento conforme ilustrado no exemplo da Figura 3. As principais propriedades incluem:

- A raiz sempre deverá ser preta;
- Se a direita do nó for preta, a esquerda deve ser obrigatoriamente preta também;
- Se a esquerda do nó atual é vermelha, a esquerda da esquerda não pode ser vermelha
- Se a esquerda e à direita de um nó são vermelhas então deve-se trocar a cor daquela região

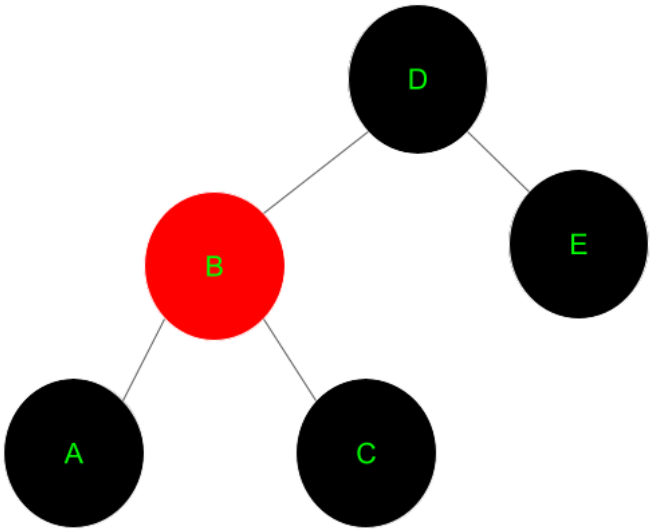

Figura 3: Exemplo de árvore rubro negro

Após a implementação da problemática abordada foram gerados oito arquivos contendo todo o código fonte da aplicação conforme a Figura 4, de forma a dividir as estruturas e funcionalidades da aplicação de forma eficiente para ser usada e atualizada na medida que novas necessidades surgirem.

1. **artista**

   Responsável pela estrutura Artista e Árvore Rubro Negro de artistas juntamente com as funções necessárias para operá-la.
2. **album**

   Responsável pela estrutura Album e Árvore Rubro Negro de albuns, jutamente com as funções necessárias para operá-la.

3. **musica**
   Responsável pela estrutura Musica e pela Lista Ordenada de músicas juntamente com as funções necessárias para operá-la.
4. **cmp**
   Responsável pela função que permite a comparação alfabética entre textos permitindo a ordenação dos dados com solicitado na regra de negócio do projeto.
5. **menu**
   Responsável pelos menus da interface que o usuário final irá usar para manipular o sistema.
6. **sistema**
   Responsável por garantir a execução das regras de negócio solicitadas para o projeto.
7. **main**
   Responsável pela interação com o usuário acessando os arquivos necessários para garantir todas as funcionalidades que o usuário solicitou.
8. **testes**
   Responsável pelos testes e avaliação de desempenho do sistema em cenários diversos.

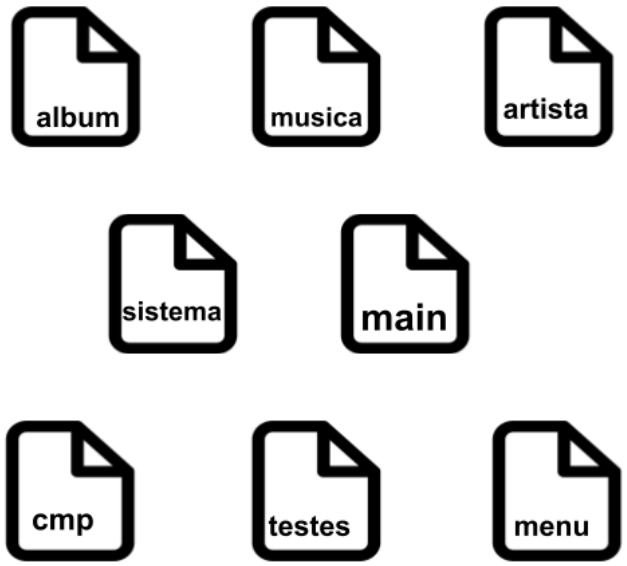

Figura 4: Divisão do código fonte da aplicação

2.2. Q2 - Biblioteca de Músicas com Árvore 2-3

A problemática abordada em Q2 gira em torno de solucionar a mesma problemática de Q1, entretanto dessa vez as informações referentes a artistas e álbuns foram foram estruturadas em formato de Árvore usando a Árvore 2-3, já as músicas seguirão o formato de uma lista duplamente encadeada ordenada novamente.

A Figura 5 apresenta os diagramas usados para a modelagem das estruturas e suas respectivas conexões nos permitindo visualizar relações como dependências existenciais, conexões de zero para muitos e etc.

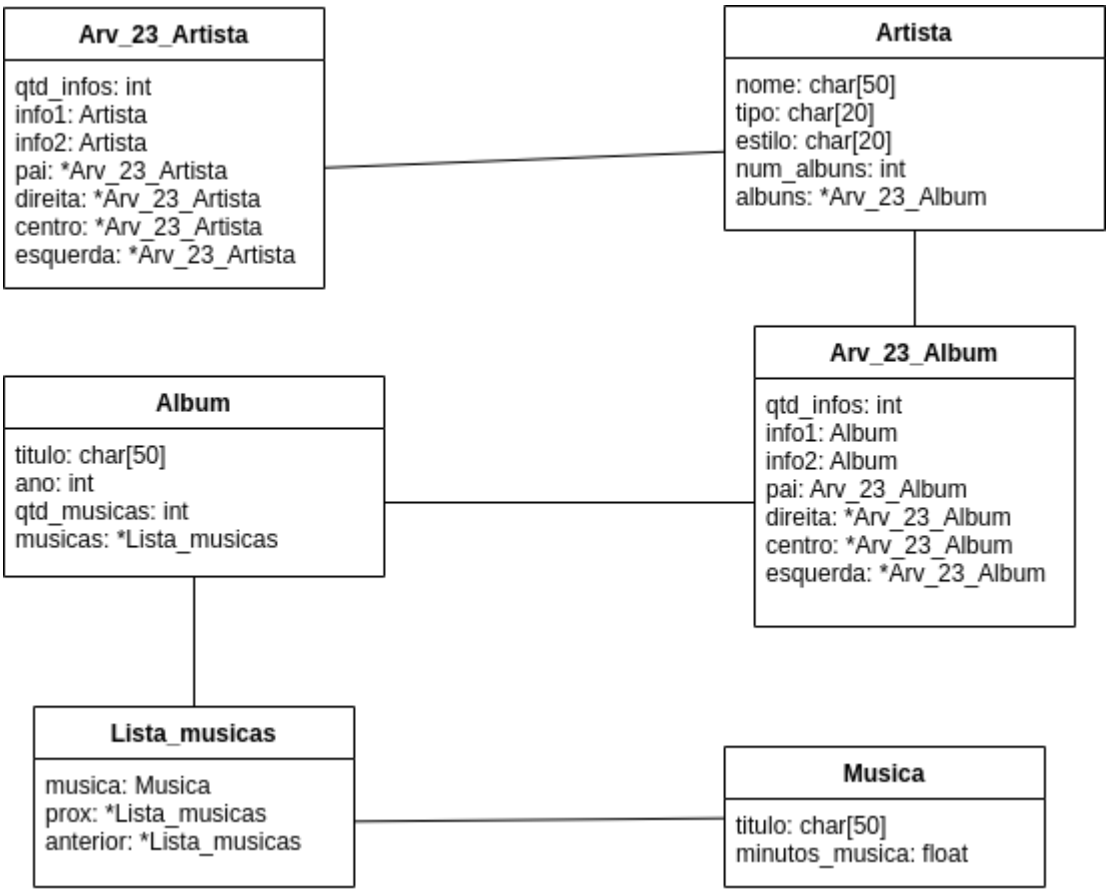

Figura 5: Modelagem das estruturas usadas pelo sistema.

Uma árvore 2-3 é uma estrutura de dados onde cada nó pode conter zero, uma ou duas informações e possui até três filhos. Essa árvore é balanceada e garante que todas as folhas estejam no mesmo nível. Elas são usadas para armazenar e gerenciar dados ordenados de forma eficiente, permitindo inserções, remoções e buscas com um tempo de execução proporcional ao logaritmo do número de chaves, mantendo o equilíbrio da árvore mesmo durante operações de modificação conforme pode ser visto na Figura 6. As principais propriedades incluem:

- A inserção de dados só acontece quando a árvore é nula ou em uma de suas folhas;
- Folhas com duas informações devem ter obrigatoriamente zero ou três informações;
- A remoção de um nó folha com apenas uma informação acarreta em impactos diferentes dependendo em qual parte da árvore o mesmo se encontra;
- Quando as informações se tornam escassas acontece o processo de juntar dados.

Tanto as regras de negócio quanto a estruturação dos arquivos se mantiveram após a implementação usando a Árvore 2-3, entretanto nosso desempenho nos testes teve um ganho de desempenho significativo.

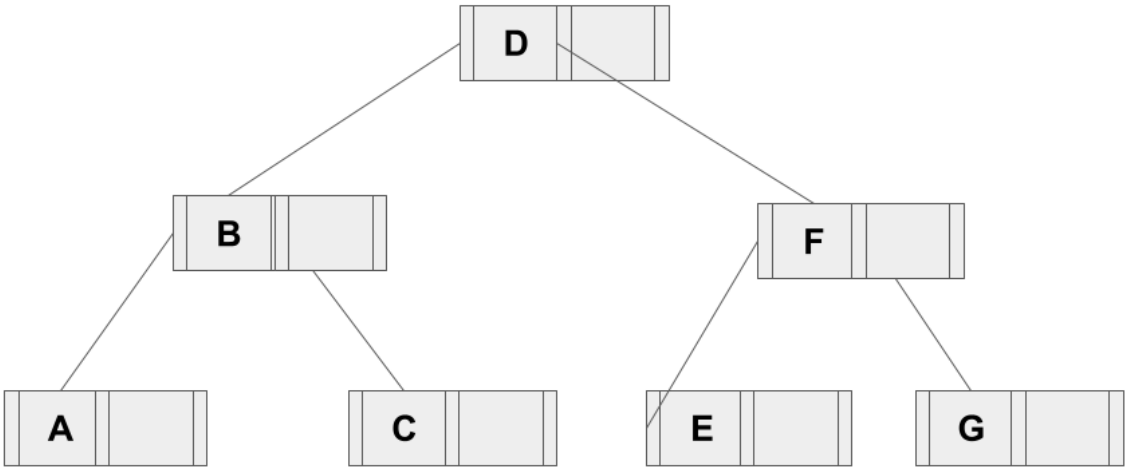

Figura 6: Exemplo de árvore 2-3

2.2. Q3 - Sistema Operacional com Árvore 4-5

Na questão Q3, o cenário simulado reproduz o funcionamento de uma memória, onde ocorre a divisão de blocos lógicos. O usuário possui acesso ao gerenciador de memória, permitindo especificar o início e o fim de um bloco na memória. O tamanho total da memória é determinado pelo usuário para alocar seus blocos. O primeiro bloco é definido manualmente pelo usuário, indicando seu início, fim e se está livre ou ocupado. Nos espaços subsequentes da memória, o usuário define apenas o final do espaço no bloco, enquanto o status de livre ou ocupado é automaticamente atribuído, continuando assim até que o bloco de memória esteja totalmente preenchido. As estruturas usadas para esta problemática estão apresentadas na Figura 7.

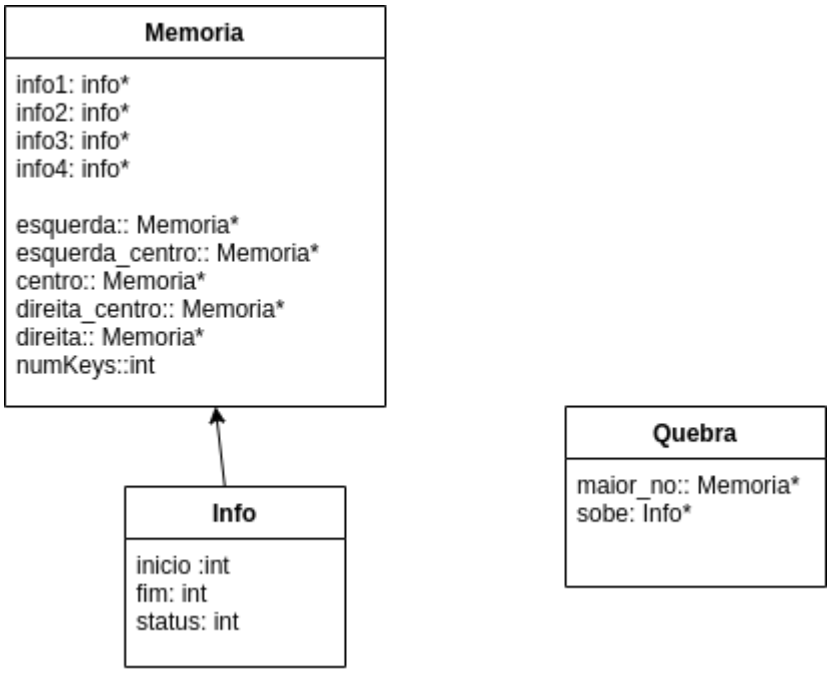

Figura 7: Estruturas usadas para a questão 3

A árvore 4-5 segue os passos de sua “parente distante” a árvore 2-3 onde cada nó pode conter zero, uma, duas, três ou quatro informações e possui até cinco filhos. Essa árvore também é balanceada e garante que todas as folhas estejam no mesmo nível. Elas são usadas para armazenar e gerenciar dados ordenados de forma eficiente, permitindo inserções, remoções e buscas com um tempo de execução proporcional ao logaritmo do número de chaves, mantendo o equilíbrio da árvore mesmo durante operações de modificação conforme pode ser visto na Figura 7. As principais propriedades incluem:

- A inserção de dados só acontece quando a árvore é nula ou em uma de suas folhas;
- Folhas com quatro informações devem ter obrigatoriamente zero ou cinco informações;
- A remoção de um nó folha com apenas uma informação acarreta em impactos diferentes dependendo em qual parte da árvore o mesmo se encontra;
- Quando as informações se tornam escassas acontece o processo de juntar dados.

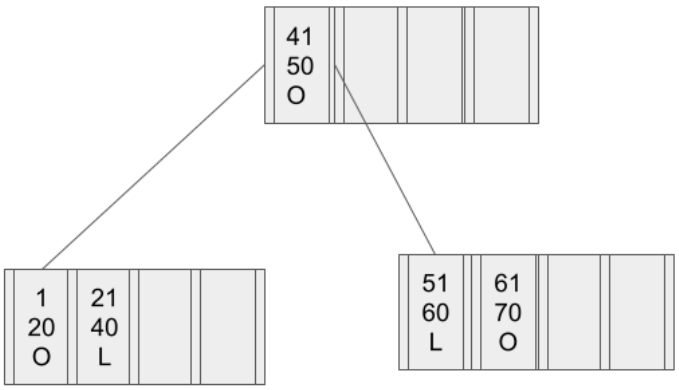

Figura 7: Exemplo de uma pequena árvore 4-5

3. Resultados da Execução do Programa

Nesta seção iremos visualizar e comparar os resultados obtidos nos experimentos realizados durante a utilização dos algoritmos gerados em Q1 e Q2 visando comparar os resultados obtidos entre ambos em algumas situações quando realizamos buscas em todos os itens de suas respectivas árvores.

Os dados obtidos nos experimentos estão registrados nas Tabelas 2 e 3. A análise revela que, apesar da variação nos volumes de dados inseridos, a árvore 2-3 demonstra consistentemente um desempenho superior em relação à árvore rubro-negra. À medida que o conjunto de dados aumenta, a árvore 2-3 mantém uma performance destacada. Embora a implementação da árvore 2-3 possa ser mais desafiadora em comparação com a árvore rubro-negra, seus resultados eficientes e consistentes a tornam uma escolha valiosa e recompensadora.

Tabela 2: Resultados de desempenho usando árvore rubro negro

Quantidade de Dados	**Tempo médio de busca (Milisegundos)**
30	**0.000467**
300	**0.000500**
3000	**0.000624**

Tabela 3: Resultados de desempenho usando árvore 2-3

Quantidade de Dados	Tempo médio de busca (Milisegundos)
30	**0.000367**
300	**0.000377**
3000	**0.000480**

4. Conclusão

Este projeto culmina com a implementação de três questões, cada uma requerendo uma árvore distinta para sua resolução. Ao longo do desenvolvimento, enfrentamos diversos desafios lógicos e técnicos, demandando um tempo significativo para compreender completamente os requisitos e iniciar a implementação. Habilidades de desenho foram essenciais para visualizar as estruturas das árvores e apoiar a codificação. Infelizmente, as questões Q2 e Q3 não puderam ser construídas de acordo com todos os critérios de qualidade estabelecidos para este problema, apesar dos esforços dedicados.

5. Apêndice

Todo o material usado para o desenvolvimento e experimentação segue em conjunto com este relatório de forma a ficar melhor a visualização e acesso com o mesmo.