

# Relatório Exodia

Lucas Villela Canôas

## 1 Objetivos

O objetivo é implementar um programa que faça a inserção dos livros de uma biblioteca em diversos tipos de estruturas de dados, como fila, pilha, árvore binária e árvore AVL. Também é considerado um objetivo a melhor modularização do código, fazendo bibliotecas e repartindo o código entre vários arquivos, o que facilita a manutenção e reuso do código, portanto seria impraticável a escrita em um único arquivo.

## 2 Ferramentas utilizadas

Foi usado o gcc versão 6.3.0 20170516 (Debian 6.3.0-18+deb9u1) como compilador, por vezes foi utilizado também o GNU Debugger para identificação de ponteiros com problemas e para a escrita do código, o programa de edição de textos para escrever o programa usado foi o Vim e para escrever o relatório o Gummi.

## 3 Estruturas Lineares

A pilha é uma estrutura linear que as operações são realizadas apenas no topo relativo da pilha, ou seja, as operações de inserção e remoção são feitas apenas de um lado arbitrário, então foram criadas as funções inserir início, remover início. Como era necessário implementar também a estrutura linear fila, foram criadas as funções remover início e remover final, pois alterando o código pode-se fazer a pilha crescer para baixo ou para cima ou fazer a fila crescer para a direita ou esquerda.

## 4 Estruturas não lineares

A árvore binária e a árvore AVL são estruturas não lineares, ou seja, existe mais de um caminho possível na estrutura. A árvore binária tem a carac-

terística de ter sempre dois filhos, já a árvore AVL é uma árvore binária balanceada pela altura. Para construir uma árvore AVL proposta, precisamos de funções de inserção, exibições em ordem, pré-ordem e pós-ordem.

## 5 Problemas encontrados

A resolução da pilha e da fila se deram de forma bastante trivial, porém a compreensão da implementação da árvore binária e AVL foram mais complexas, possivelmente devido ao uso exacerbado de recursividade, o que aumenta substancialmente a abstração do código. Algo que ainda apresenta uma dificuldade, é lidar com os erros do compilador, pois nem sempre são triviais a ponto de saber exatamente o que está acontecendo, por conta disso foi perdido tempo significativo na implementação da árvore binária, que até o presente momento contém ponteiros que perdem o direcionamento. Não foi obtido sucesso na implementação da árvore AVL, muitos erros de ponteiro e o código foi se tornando confuso para realizar a manutenção, mesmo realizando a tentativa de organizar melhor o código, devido ao problema do tempo a extensão da árvore AVL não foi devidamente implementada até a versão 0.8 do projeto. Na versão 0.9 todas as funções da árvore avl foram implementadas com sucesso.

## 6 Conclusão

Foi possível analisar com sucesso as estruturas lineares pilhas e filas, a estrutura da árvore binária, algo que pode-se considerar uma tarefa interessante em breve será transformar estruturas lineares em árvores, realizando uma função de inserção que faça ambas trocarem informações. A árvore AVL apresenta uma complexidade maior na implementação, mas compensa na velocidade de busca, pois não corre o risco de se tornar uma árvore desbalanceada e até mesmo degenerada.