# Trabalho Prático III: Um Resgate Inesperado

Lucas S. Teles

# 1. Introdução

Esta documentação descreve a modelagem e implementação da Árvore de Transliteração do terceiro e último Trabalho Prático da disciplina de Estruturas de Dados, semestre 2020/2 da UFMG.

Foi criado um módulo chamado CentroComunicacao para conter a Árvore de Transliteração e saber construí-la, decodificar e codificar mensagens com o uso dela.

Na seção 2, a implementação do módulo será descrita com mais detalhes; na seção 3, será feita a análise de complexidade de tempo e espaço do sistema completo; e na seção 4, estará a conclusão da documentação com comentários sobre o percurso durante a solução do problema.

# 2. Implementação

A Árvore implementada, foi baseada naquela ensinada em aula e é uma árvore de busca como pede a especificação do trabalho.

O módulo do centro de comunicação foi implementado visando utilizar as funcionalidades da árvore sem que a função de codificar e decodificar as mensagens ficasse por conta dela. Entretanto, o sistema ainda possui alto acoplamento, o que é uma desvantagem. Não foi encontrada uma maneira mais elegante de resolver esse problema, porém o sistema funciona perfeitamente.

Existem alguns métodos utilitários que são chamados pelas chamadas "funcionalidades" de modo a garantir ao menos um encapsulamento básico.

Foram utilizadas streams para facilitar a leitura do arquivo por linhas e permitir um código mais claro e enxuto.

Os arquivos seguem algum tipo de padronização na ordem dos #include (primeiro os arquivos próprios, depois as bibliotecas da linguagem), na declaração de métodos e variáveis, e implementação de métodos. As declarações e a implementação dos métodos é divida em seções que são similares de alguma forma, além de serem indentadas a fim de facilitar a legibilidade do código. Na declaração os métodos são separados entre os privados e públicos, já na implementação eles estão, de certa forma, agrupados de maneira a agilizar a leitura de algum outro método chamado pelo método que o leitor está acompanhando. Existem alguns comentários que foram considerados relevantes de alguma forma.

Somente um tratamento de erro é feito para o caso de quantidade incorreta de atributos na utilização do sistema, porém todos os casos de linhas foram devidamente tratados.

# 3. Análise de Complexidade

Todos os métodos e classes implementados possuem complexidade de espaço O(n) ou O(1) para aqueles que não utilizam a árvore.

A complexidade de tempo dos métodos que caminham pela árvore (com exceção da criação dela) são  $O(\log n)$ , que é a sua profundidade. A criação da árvore é O(n).

#### 4. Conclusão

Houve apenas uma grande dificuldade no método de decodificar a mensagem, que começou bastante complexo, com alguns erros ao ler o fim da linha sem nenhum tratamento. Porém após várias revisões e refatorações o resultado foi um método bem simples e que funciona sem erros. Foi desenvolvida, mais uma vez, a capacidade de depuração.

O autor não considera a modelagem a mais ideal porém o sistema serve para o propósito que foi concebido.

# **Bibliografia**

Chaimowicz, L. e Prates, R. (2020). Slides virtuais da disciplina de estruturas de dados. Disponibilizado via moodle. Departamento de Ciência da Computação. Universidade Federal de Minas Gerais. Belo Horizonte.

http://www.cplusplus.com/reference/ios/noskipws/

https://www.cplusplus.com/reference/cstdlib/rand/