

**ESINF**

**Morse Code**

**Alunos**:

1161071, José Marques

1150503, Gonçalo Fonseca

**Docente**:

Nuno Filipe Malheiro

**Dezembro de 2018**

**Resumo**

O presente documento pretende de uma forma sucinta descrever o desenvolvimento de uma aplicação que permita codificar e descodificar código morse.

Após uma análise de requisitos foi proposta a melhor solução para responder aos problemas propostos. Utilizando o paradigma da programação orientada a objetos e uma estratégia de desenvolvimento dirigida por testes.

**Palavras-Chave:**

BST; Trees; Tree; Java; Nodes; Unit; Testing; Morse; MorseCode; Managing; Software; Maps; Code; Graphs; Nodes; Algorithm;

1. Introdução

No âmbito da unidade curricular de Estruturas de Informação, lecionada no 1º semestre do 2º ano de Licenciatura em Engenharia Informática, foi proposto o desenvolvimento de um serviço que sirva para codificar e descodificar mensagens em código morse.

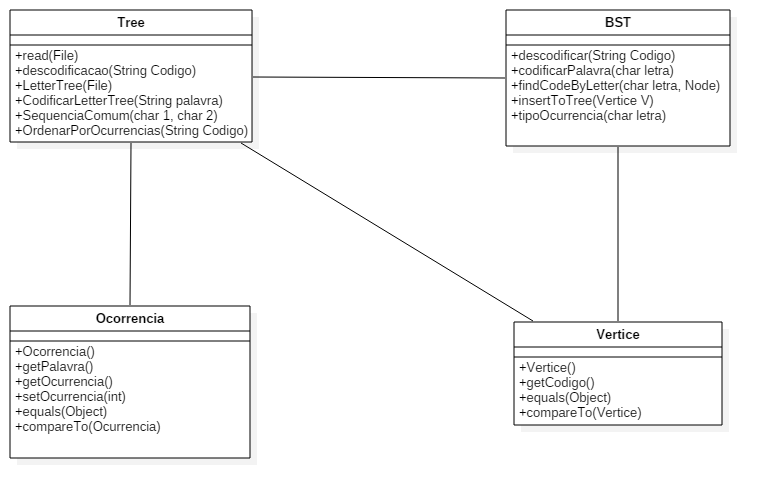
Neste trabalho implementamos as classes desenvolvida nas TP’s e PL’s de Trees, Graphs, Edges and Vertex.

Após começarmos a desenvolver o projeto, sentimos a necessidade de criar algumas outras classes para organizar e otimizar o nosso código, bem como para prevenir tarefas futuras.

O projeto é feito de forma incremental, em que na primeira parte é proposto o desenvolvimento de métodos começando com as classes mais simples até se chegar ao sistema final.

Para este trabalho, foi dada especial atenção ao desenvolvimento da estrutura de dados implementada para suportar esta aplicação visto ter um impacto determinante no funcionamento da aplicação.

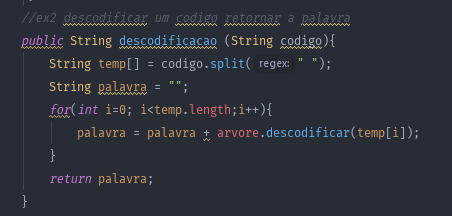
1. Diagrama de Classes



1. Análise de complexidade

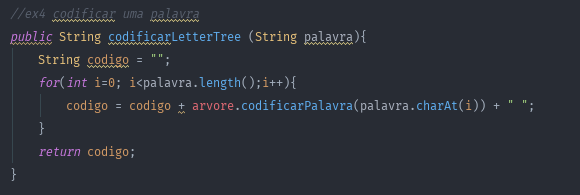


Exo1 – read é O(N) + insertTree O(V) determinístico porque no reader vai se sempre ler até ao fim do ficheiro. O(N+V)



Ex2 – O(N) pois vai sempre percorrer o array (String) na totalidade. Descodificar é O(N) logo no total é O(2N)



 Ex3 - O(N) + O(N) = O(2N)

Ex4 – O(N) Determinístico + O(1) + O(N) não determinístico. (Na melhor das hipóteses é O(LOG N)



Ex5 – O(N) + O(1) + O(N) não determinístico na pior das hipóteses e na melhor O(LOG N)



Ex6 –

1. Conclusão

Numa forma de conclusão, estamos mais satisfeitos com esta iteração do que com as outras duas. Houve menos descoordenação na realização da aplicação, o que provocou esta poder estar mais bem polida. Consideramos ter feito um trabalho positivo, tanto a nível académico como a nível pessoal.

Não foi possível, com o tempo disponível elaborar e aprofundar muito mais este documento, que irá requerer mais atenção e trabalho num futuro próximo.

A nível aplicacional, este foi maioritariamente testado com testes unitários.

Finalmente, de notar que o trabalho permitiu o aprofundamento de técnicas relevantes a nível de programação.