

Relatório – Sprint 1-2

EXPOSIÇÃO E EXPLICAÇÃO

Diogo Sá Dias – 1161605

Duarte Dias - 1190539

ESINF | 2021-22

Índice

Introdução	2
Sprint 1	3
Diagrama de Classes	3
Análise de Complexidade	3
Sprint 2	4
Diagrama de Classes	4
Análise de Complexidade	4
[US201] As a Port manager, I which to import ports from a text file and create a 2D-tree with port locations.	4
[US202] As a Traffic manager, I which to find the closest port of a ship given its CallSign, on a certain DateTime.	5

Introdução

Breve explicação do Diagrama de Classes:

Inicialmente, foi pensado fazer vários controllers e cada um dos mesmos iria implementar uma US. Posteriormente optamos por fazer um só controller que designa-mos de MainController, no qual esse fará a função de todos os outros controllers anteriormente mencionados.

Basicamente, é importado através da classe Import, um ficheiro csv, no qual possui todas as informações dos navios. Esses navios que são guardados numa árvore ShipTree que possui os navios. Cada navio tem uma árvore de movimentos que guarda todos os atributo dinâmicos do navio, ou seja, os relacionados com a posição do navio.

Complexidade das US's: varia desde $\log n$ até n ao quadrado

Sprint 1

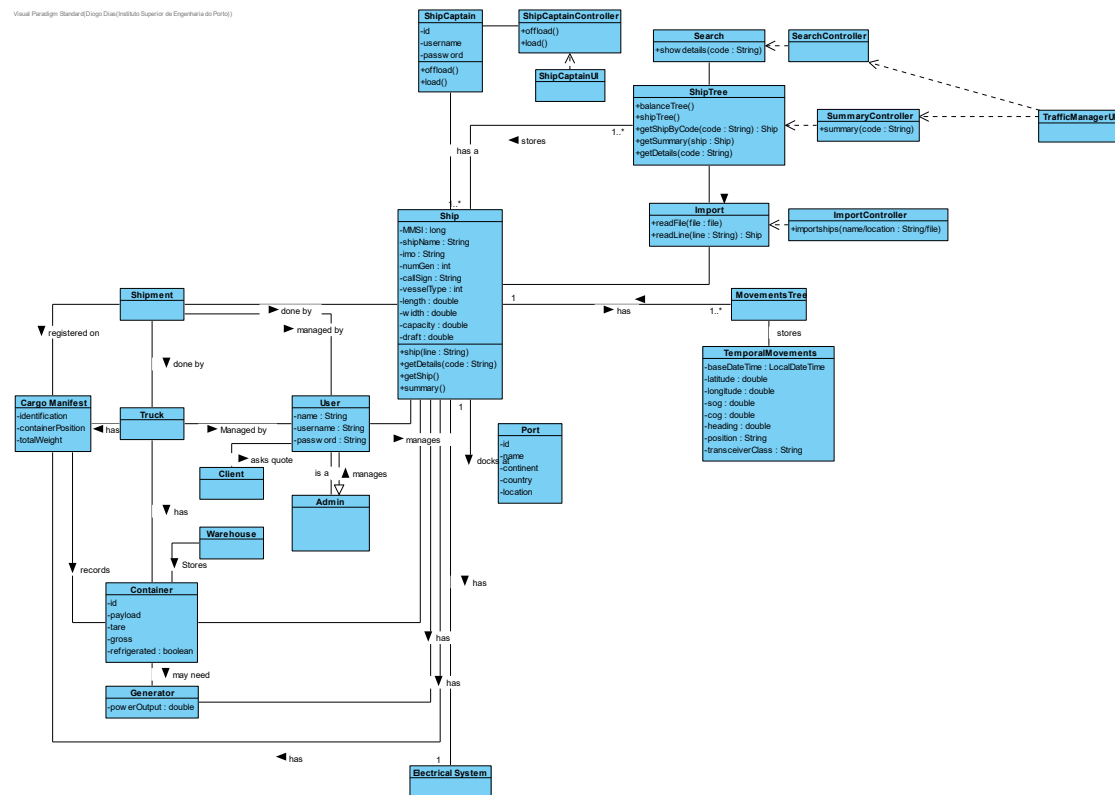
DIAGRAMA DE CLASSES

ANÁLISE DE COMPLEXIDADE

Sprint 2

DIAGRAMA DE CLASSES

Visual Paradigm Standard (Diogo Dias/Instituto Superior de Engenharia do Porto)



ANÁLISE DE COMPLEXIDADE

[US201] As a Port manager, I wish to import ports from a text file and create a 2D-tree with port locations.

De acordo com o pedido no enunciado, o que se pretende nesta funcionalidade é receber informação de um ficheiro e inserir a mesma numa estrutura adequada. Para isso teremos um processo que irá percorrer todas as linhas do documento indicado. Depois iremos a cada uma das linhas construir o objeto Porto e de seguida inserimos na árvore 2D-Tree.

Deste modo, ler a linha e construir o Porto tem complexidade 1 e inserir na 2d-Tree tem complexidade $\log n$. Como tal a criação desta árvore irá ter no seu pior cenário $n * \log n$ de complexidade temporal.

Diogo Sá Dias - 1161605

[US202] As a Traffic manager, I wish to find the closest port of a ship given its CallSign, on a certain DateTime.

De acordo com o enunciado pretende-se encontrar o porto mais próximo usando a 2D-Tree anteriormente criada. Para isso implementei um método da 2D-Tree designado de findNearestNeighbour. Esse método vai calcular a distância entre o ponto dado e um nó da 2d-Tree, começando pela raiz e continuando, recursivamente por ambas os ramos da 2d-Tree. No final guarda a informação do nó mais próximo.

Assim esta funcionalidade vai ter, no melhor caso complexidade $\log n$ e, caso a dimensão da árvore for alta ou a forma como os nós da árvore for especialmente organizada, a complexidade será n .

Duarte Dias - 1190539