

Faculdade de Engenharia da Universidade do Porto



Universidade do Porto

FEUP Faculdade de Engenharia

Proteção Civil

Algoritmos e Estruturas de Dados

Novembro 2017 - MIEIC

Equipa: Turma1_G7

Elementos:

Helena Montenegro: up201604184@fe.up.pt

Mariana Dias: up201606486@fe.up.pt

Índice

1. Introdução.....	3
2. Sistema.....	4
2.1. Árvore binária de pesquisa.....	4
2.2. Fila de prioridade.....	5
2.3. Tabela de dispersão.....	6
3. Diagrama de Classes.....	7
4. Principais dificuldades.....	8
5. Contribuição de cada elemento do grupo.....	8

1. Introdução

No âmbito da disciplina de Algoritmos e Estruturas de Dados, desenvolvemos um projeto com o tema “Proteção Civil”, que consiste na criação de um sistema que permita atribuir socorristas aos acidentes que ocorrem em Portugal, de uma forma eficiente. Nesta segunda parte, foram adicionadas ao sistema estruturas que nos permitem gerir a informação sobre as marcas de veículos, tal como os seus condutores, envolvidos num acidente de viação, e uma estrutura que permite distribuir da forma mais eficiente os carros que necessitem de ser consertados pelas diferentes oficinas disponíveis para os mesmos.

2. Sistema

O sistema desenvolvido, além de ser constituído pelos acidentes e pelos postos de socorro existentes para atender aos acidentes, contém uma árvore binária que permite gerir a informação sobre as marcas de veículos envolvidos num acidente de viação, uma tabela de dispersão na qual são cadastrados os condutores dos veículos, envolvidos nesses acidentes, e uma fila de prioridade, para possibilitar o acesso à oficina mais apropriada para um condutor consertar o seu veículo.

2.1. Árvore binária de pesquisa

Quando se inicia o programa, é guardada na árvore binária de pesquisa toda a informação sobre marcas de veículos, presente no ficheiro de texto: “veiculos.txt”. Esta informação inclui o nome da marca, a data do último acidente que ocorreu com um veículo dessa marca e o número de acidentes em que a marca esteve envolvida. As marcas estão então definidas na classe Veiculo. Na árvore, as marcas estão ordenadas por ordem decrescente em relação ao número de acidentes e à data do último acidente. Em caso de empate, estão ordenadas por ordem alfabética.

Pode-se adicionar um novo veículo à árvore, isto é, incrementar o número de acidentes da marca do veículo se esta estiver presente na árvore ou adicioná-la caso não esteja, ao adicionar-se um novo acidente de viação ao sistema, através da opção B do menu. Com a opção L do menu, é possível visualizar uma listagem das marcas existentes na árvore e a sua informação. As opções N e O são, do mesmo modo, funções de visualização, que permitem verificar quantos acidentes de viação envolveram uma marca especificada pelo utilizador e quais as marcas envolvidas num número de acidentes igual ou superior ao número especificado pelo utilizador. A opção M permite retirar da árvore marcas cujo último acidente tenha ocorrido à mais de um número de anos, sendo este número definido pelo utilizador. No final do programa, todas as alterações efetuadas na árvore são guardadas no ficheiro de texto correspondente às marcas: “veiculos.txt”.

A árvore binária de pesquisa foi implementada com a biblioteca “BST.h”, fornecida pelos docentes na página da disciplina no moodle.

2.2. Fila de prioridade

Na fila de prioridade está presente a informação sobre os objetos da classe Oficina, que tal como o nome indica, se refere às diferentes oficinas disponíveis para consertar os veículos que necessitem de reparos. Estes objetos contêm um nome, para facilitar a sua identificação na visualização de informação, a sua disponibilidade, isto é, o número de dias que necessitam até ficarem disponíveis, e as marcas que representam. No início do programa, a informação presente no ficheiro de texto: "oficinas.txt", relativa às oficinas, é colocada na fila de prioridade.

É possível adicionar uma nova oficina ao sistema, editar a informação sobre uma oficina presente no sistema e remover uma oficina do sistema através das opções Q, R e S do menu, respetivamente. Para colocar um veículo em reparação numa oficina existe a opção T do menu, que pede ao utilizador informação sobre a marca do veículo em questão e apresenta o nome da oficina que ficará disponível mais cedo e que representa essa marca. A disponibilidade desta oficina é incrementada. A opção U representa a passagem de um dia, sendo que em todas as oficinas será reparado um veículo, decrementando-se o valor da disponibilidade em cada uma delas. As opções V e W permitem a visualização da informação sobre as oficinas. No final do programa, a informação relativa às oficinas é guardada no ficheiro de texto: "oficinas.txt".

2.3. Tabela de dispersão

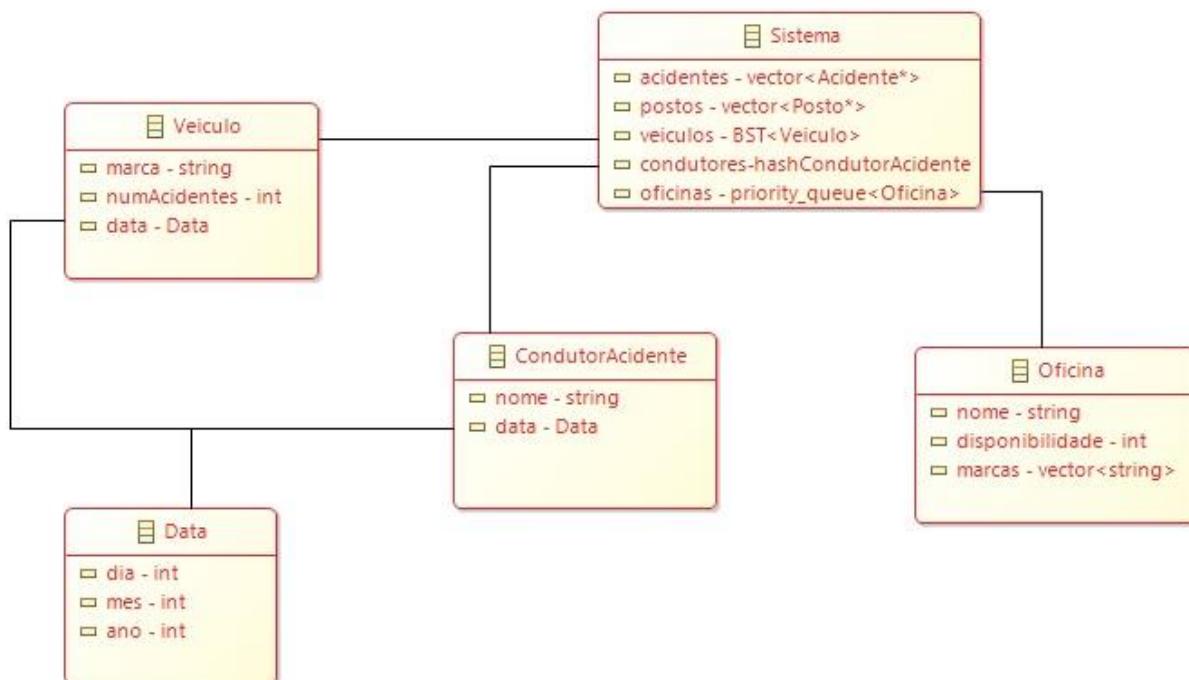
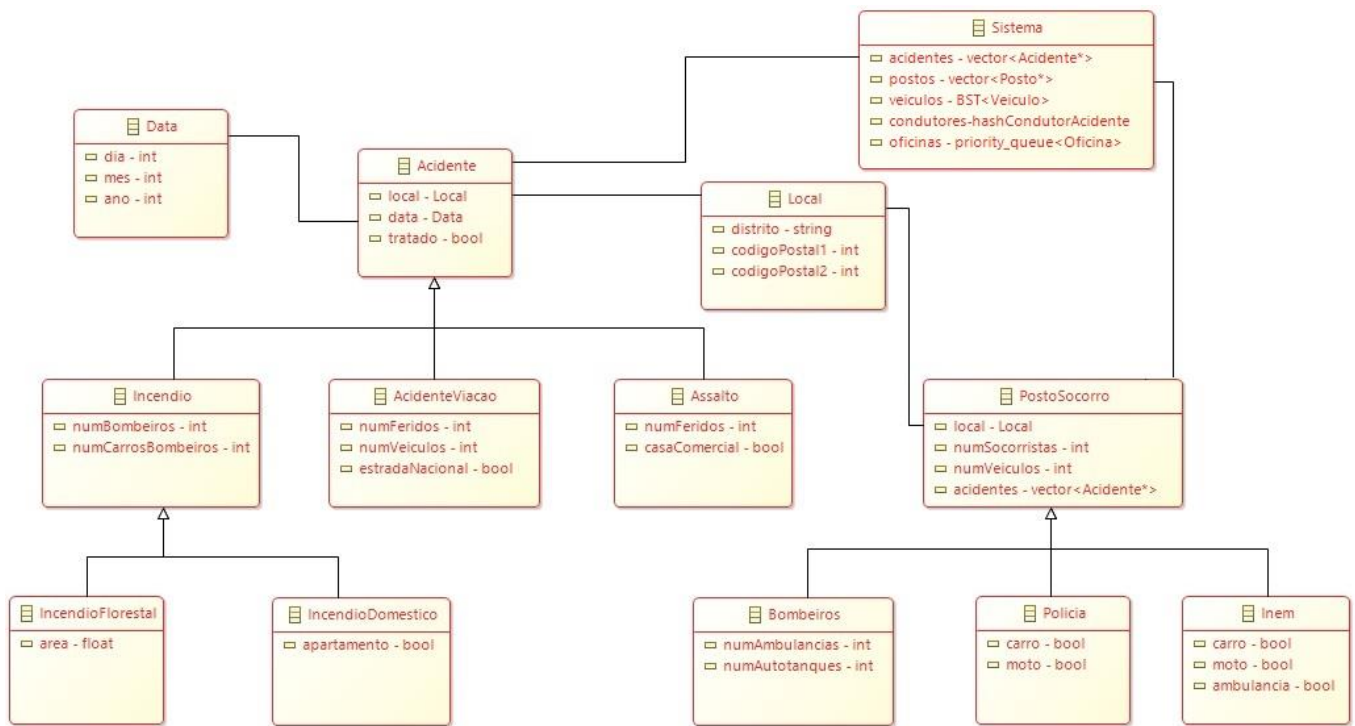
Os condutores envolvidos em acidentes de viação são adicionados ao sistema através da função *guardaInformacaoViacao()*, sempre que é adicionado um novo acidente de viação, após serem abertos os ficheiros que contêm a informação sobre acidentes, postos de socorro, veículos e condutores. São cadastrados o nome do condutor e a data do último acidente em que esteve envolvido (atributos da classe *CondutorAcidente*) na tabela de dispersão *condutores*.

Em função da data do novo acidente adicionado, compara-se essa data a todas as armazenadas no sistema. Se a data atual for 5 anos mais recente que qualquer uma das armazenadas, essa última é eliminada da tabela.

Relativamente à estrutura de dados em si, implementamos a tabela de dispersão através do *unordered set* do TR1, em que a função de espalhamento escolhida usa o método de Horner, isto é, para cada string *s* de tamanho *L*, a função é dada pelo seguinte polinómio - $hash = s[0] * 37^{L-1} + \dots + s[L-2] * 37^1 + s[L-1] * 37^0$. Em caso de colisão, a biblioteca utilizada tem funções como *max_load_factor()*, que por default é 1.0, e que age como um limite que força o redimensionamento da tabela para evitar que dois elementos se situem na mesma posição da tabela, sendo a tabela reconstruída tendo em conta o seu novo tamanho - *rehash()*.

A opção P do menu permite a visualização da informação dos condutores envolvidos em acidentes de viação existentes no sistema, sendo que, para cada condutor pertencente à tabela de dispersão do sistema, mostra-se a data do último acidente. No final do programa é guardado no ficheiro de texto: "condutores.txt", toda a informação presente na tabela de dispersão, sobre os condutores envolvidos em acidentes de viação.

3. Diagrama de Classes



4. Principais dificuldades

Não houve dificuldades significativas no desenvolvimento desta parte do projeto.

5. Contribuição de cada elemento do grupo

A aluna Helena Montenegro desenvolveu todas as funções e classes relacionadas com a árvore binária de pesquisa, e desenvolveu parte da fila de prioridade: a classe Oficina, à exceção da função de visualização, a função que guarda na fila da prioridade a informação disponível no ficheiro de texto respectivo, e as funções correspondente às opções Q, T e U do menu, isto é, funções que permitem adicionar uma nova oficina ao sistema, colocar um veículo em reparação e reparar um veículo em cada oficina, respetivamente.

A aluna Mariana Dias desenvolveu todas as funções e classes que dizem respeito à tabela de dispersão, e, na parte da fila de prioridade, ficou encarregue de: função de visualização da classe Oficina, função que permite guardar no ficheiro de texto a informação sobre as oficinas presente na estrutura e funções correspondentes às opções R, S, V e W, que permitem alterar e remover uma oficina do sistema, verificar quais as oficinas que representam uma determinada data e visualizar a informação sobre todas as oficinas do sistema.

A função *guardaInformacaoViacao()* que diz respeito a ambas as estruturas, árvore binária de pesquisa e tabela de dispersão, foi desenvolvida por ambas as alunas, sendo que a parte onde se pede o nome do condutor e a inserção de um novo condutor na tabela de dispersão foi definida pela Mariana, enquanto que a parte onde se pede a marca do veículo do condutor e se insere um novo veículo na árvore binária de pesquisa foi desenvolvida pela Helena.

Deste modo, conseguimos repartir o trabalho de forma equilibrada por ambas as alunas.