

Algoritmos e Estruturas de Dados

2018/2019

**Tema 2 – Aluguer de Campos de Ténis**

**(Parte 2)**

**Turma 5 Grupo 2:**

Mariana Catarina Pereira Soares – [up201605775@fe.up.pt](mailto:up201605775@fe.up.pt)

João Miguel dos Santos Lima Monteiro – [up201705580@fe.up.pt](mailto:up201705580@fe.up.pt)

Miguel Rodrigues Pires – [up201406989@fe.up.pt](mailto:up201406989@fe.up.pt)

Índice

[**Descrição do Tema do trabalho** 3](#_Toc534731702)

[**Solução Implementada** 3](#_Toc534731703)

[**Casos de Utilização** 4](#_Toc534731704)

[**Principais Dificuldades** 6](#_Toc534731705)

[**Distribuição de trabalho pelo grupo** 6](#_Toc534731706)

# **Descrição do Tema do trabalho**

Nesta segunda parte do projeto foi necessário implementar três novas estruturas de dados, conforme é pedido no enunciado, e integrá-las na anterior implementação:

* **Binary Search Tree;**
* **Fila de prioridade (priority\_queue);**
* **Tabela de dispersão (unordered\_set).**

# **Solução Implementada**

De forma a resolver os problemas propostos, nós construímos três estruturas de dados não lineares:

1. ***Binary Search Tree***

A Binary Search Tree contém elementos do tipo Utente e ordena-os segundo o número de vezes que usufruíram os campos e em caso de empate por ordem alfabética dos nomes. Para pesquisar certos elementos da árvore foi usado um iterador em ordem.

1. ***Priority Queue***

A Priority Queue contém…

1. ***Hash Table***

A Hash Table contém elementos do tipo professores. Nesta tabela estão presentes todos os professores que já trabalharam ou que ainda trabalham no campo de ténis estando ordenados segundo o número de aulas dadas pelo professor.

# **Casos de Utilização**

O programa começa logo por ler as informações dos ficheiros necessários (Aulas.txt, Livres.txt, Professores.txt e Utentes.txt). De seguida, caso no ficheiro Aulas.txt/Livres.txt exista uma aula/livre que não seja possível guardar a informação dessa aula/livre por conter dados inválidos, aparece na consola quais aulas/livres não foram adicionadas(os) e qual o motivo.

Após a listagem do que não foi adicionado por conter dados inválidos, é apresentado um menu com três opções, sendo que a primeira opção (1- Consultar Informação) permite a consulta de informações, a segunda (2-Alterar dados) permite manipular (inserir ou remover) dados e a terceira (3-Sair) serve para sair do programa.

Após selecionar a primeira opção, um segundo menu será apresentado, dando a oportunidade ao utilizador de selecionar qual a informação que pretende ver.

Neste mesmo menu, caso escolha a opção 1, será indicado o número de campos que a empresa possui e uma matriz que representa a disponibilidade dos campos de ténis por *slot* e por mês. Esta matriz indica-nos a lotação total menos o número de aulas nessa dada hora e dia, ou seja, indica o número de livres que poderão ser adicionados nessa dada hora/dia.

Caso escolha a segunda, o utilizador poderá selecionar duas opções: a opção 1, que permite ver os horários das aulas, e a 2, que após o utilizador inserir o nome de um professor existente, poderá ver as aulas que este professor leciona.

Ao selecionar a terceira opção, surgirá um novo menu que permite que o utilizador selecione quais as informações, relativas a um utente, que pretende ver.

Assim, neste menu, o utilizador pode ver qual a frequência de um dado utente nos campos de ténis, bem como a conta de um utente (nome, idade, se tem cartão *gold* ou não) e pode ainda gerar um relatório num ficheiro .txt com todas as aulas e livres que o utente frequentou e o valor que terá de pagar nesse mês. Juntamente com este relatório é também criado um ficheiro com o progresso do utente.

A quarta opção é utilizada para voltar ao menu anterior e a quinta para sair do programa.

No menu que aparece mal se inicia o programa, ao selecionar a segunda opção, um novo menu será apresentado com quatro novas opções em que a primeira e segunda opção adicionam um utente ou um professor, respetivamente, e a terceira e quarta removem um utente ou professor.

Ao selecionar uma das duas primeiras será necessário introduzir a informação a adicionar, como o nome, a idade, o cartão (no caso de utente) e a sigla (no caso de professor).

Ao selecionar a terceira ou a quarta apenas será pedido para inserir o nome da pessoa a retirar. No caso de o nome inserido não existir nada acontecerá.

Para a Parte 2 do projeto, a única mudança na interface foi a adição de um menu chamado “Serviços Técnicos”, que é a terceira opção da primeira camada de opções. Escolhendo esta opção abrirá um novo menu onde podemos escolher entre:

* Consultar a informação de todos os técnicos – esta opção irá mostrar no ecrã o nome, a disponibilidade e o número de reparações de cada técnico;
* Reparação de campo de ténis – esta opção serve para selecionar, dependendo da disponibilidade dada, um técnico para reparar um campo de ténis;
* Adicionar técnico – esta opção faz o que o nome indica, serve para criar um novo técnico, introduzindo um novo nome, disponibilidade e quantas reparações já fez;
* Remover técnico – esta opção serve para eliminar um técnico do conjunto de técnicos existentes, identificado pelo nome dado.

Mais uma vez, a quinta opção será usada para voltar ao menu anterior e a sexta para sair do programa.

# **Principais Dificuldades**

Na segunda parte deste trabalho não encontramos tantas dificuldades como na primeira parte, pois muitas das nossas dificuldades anteriores passavam por uma interpretação clara do enunciado, o que neste novo enunciado foi-nos mais fácil de interpretar.

Houve apenas uma dúvida no enunciado que não esclarecemos, que era durante a implementação da *Hash Table* se a condição da atribuição equitativa de aulas para todos os professores se mantinha como na parte 1, o que iria implicar que todos os professores, além dos professores antigos, que têm zero aulas, iriam estar na mesma posição da tabela de dispersão. No final optamos por manter essa condição do enunciado da parte 1 do projeto.

Para concluir achamos que a única grande dificuldade que tivemos foi não conhecermos na totalidade as estruturas que tivemos que implementar, o que nos obrigou a alterar minimamente algumas das nossas classes de forma a cumprir os objetivos necessários e a várias situações de tentativa e erro.

# **Distribuição de trabalho pelo grupo**

Todos nós participamos ao longo do desenvolvimento do trabalho, tendo havido uma distribuição equitativa do esforço por todos os elementos do grupo.