

Dep.	Informática	Unidade Curricular	Introdução à Programação			
Curso	CTeSP em Redes e Sistemas Informáticos		Ano	1º	Sem.	1º
Projeto	Estatística no Andebol					
2024/2025	Época normal Entrega: 23h00 – 28 de dezembro de 2025					

1. Introdução

O registo de dados e a estatística desempenham um papel importante no desporto moderno. O andebol não é exceção a esta tendência, sendo habitual registar diversos parâmetros do desempenho dos atletas.

Suponha que a Federação Portuguesa de Andebol pretende desenvolver uma aplicação que permita registar alguns dados estatísticos relativamente às equipas que disputam um campeonato. Para este efeito, cada equipa é caracterizada pela sua designação e pelo grupo de atletas que a compõem (no máximo 15 atletas).

2. Descrição da aplicação e requisitos

A apresentação ou a formatação na consola dos menus fica ao seu critério.

2.1 Atletas

Cada atleta é caracterizado pelo seu nome, número de identificação (numInd: **número único e positivo de 7 dígitos**), ano de nascimento, posição que joga (ponta, lateral, central, pivô e guarda-redes), média de pontos por jogo (mPontos), média de remates por jogo (mRemates), média de perdas de bola por jogo (mPerdas), média de assistências por jogo (mAssist), média de fintas por jogo (mFintas) e total de minutos em jogo (tMinutos).

Com exceção dos guarda-redes, a valia de cada atleta é função destes parâmetros, mas o seu cálculo é diferente, consoante a posição em que o atleta joga, de acordo com as fórmulas seguintes:

Ponta: $valia = 5 * mPontos + 4 * mRemates + 2 * mAssist - 3 * mPerdas$

Lateral: $valia = 3 * mPontos + 2 * mRemates + 4 * mAssist - 3 * mPerdas$

Central: $valia = 2 * mPontos + 5 * mAssist + 4 * mFintas - 2 * mPerdas$

Pivô: $valia = 2 * mPontos + 1 * mRemates + 5 * mFintas - 2 * mPerdas$

3. Objetivos

A finalidade deste trabalho é desenvolver uma aplicação que permita registar os dados estatísticos de equipas de andebol. No arranque da aplicação devem ser carregados de um ficheiro, de forma automática, os dados retratados no ponto 4.

Testes (exemplos).

A aplicação deve permitir:

1. Registar uma equipa e respetivos atletas (os dados devem ser pedidos ao utilizador);
Mostrar a informação de determinado atleta com base no seu número de identificação;
2. Adicionar atletas de equipas;
3. Atualizar informação de um atleta de modo eficaz;

4. Listar os atletas de cada equipa, ordenados:
 - 4.1. Por ordem crescente, e decrescente, da respetiva valia;
 - 4.2. Alfabeticamente pelo 1º nome do atleta;
5. Pesquisar atletas com base em critérios definidos pelo utilizador:
 - 5.1. Valia superior/inferior a determinado valor (inserido pelo utilizador);
 - 5.2. Data de nascimento anterior/posterior a determinado ano (inserido pelo utilizador);
6. Calcular a valia de uma equipa (soma das valias dos seus atletas calculadas como descrito anteriormente);
7. Indicar qual a equipa mais valiosa e qual a menos valiosa;
8. Indicar qual o atleta mais valioso em cada posição. Para esse efeito só devem ser considerados atletas com mais de 100 minutos de jogo.
9. Permitir pesquisar quais os atletas nascidos em determinado ano (inserido pelo utilizador).
10. Remover atletas de equipas:
 - 10.1. Remover apenas determinado atleta;
 - 10.2. Remover todos os atletas;
11. Gravar a informação num ficheiro a pedido do utilizador;
12. Gravar automaticamente as alterações de dados realizadas em tempo de execução (as alterações aos dados devem ser atualizadas no ficheiro de suporte à aplicação);
13. Gravar de forma automática os dados aquando da saída da aplicação;

Pode optar por implementar a gestão dinâmica da memória para o armazenamento das equipas e atletas, permitindo criar e remover equipas e atletas sem limites pré-definidos. Neste caso, as estruturas deverão ser alocadas dinamicamente, utilizando funções como `malloc`, `calloc`, `realloc` e `free`, permitindo a criação e remoção de equipas e atletas sem limites pré-definidos. Deverá ainda ser assegurada a correta libertação da memória no encerramento da aplicação, evitando fugas de memória e acessos inválidos.

4. Testes (exemplos)

Para efeitos de testes deve criar um ficheiro (que será carregado no arranque da aplicação) com, pelo menos, 2 equipas. Cada equipa deve ter jogadores de todas as posições de jogo, e os dados de cada jogador devem estar registados convenientemente.

5. Observações

- Deverá criar as variáveis e estruturas de dados necessárias;
- A validação de dados será valorizada;
- O trabalho deve ser implementado em linguagem C e deve apenas usar os conhecimentos lecionados nas aulas de Introdução à Programação;
- A funcionalidade do programa é, nesta unidade curricular, muito mais importante que os aspetos estéticos. Será valorizada a clareza e simplicidade do código. Deverá assim estruturar o programa de modo a torná-lo simples, bem estruturado, e sem repetições desnecessárias de código. Será ainda valorizada a conveniente indentação do código e a inclusão de comentários, sempre que tal for interessante do ponto de vista da interpretabilidade;
- Possíveis melhorias incluídas no programa, devidamente fundamentadas e enquadradas com o programa desta unidade curricular, serão consideradas e valorizadas;
- Deve ser elaborado um relatório que descreva o trabalho produzido e as funcionalidades implementadas;
- Não serão admitidos plágios, mesmo que parciais - trabalhos copiados terão nota ZERO (tanto quem copiou como quem deixou copiar);
- Caso se revele necessário, poderão ser feitas atualizações ou alterações a este enunciado, pelo que os alunos deverão estar atentos a esta eventualidade. Qualquer atualização ou alteração será devidamente anunciada no Moodle desta unidade curricular.

6. Constituição dos grupos

Os grupos de trabalho deverão ser constituídos por um mínimo de três e um máximo de 4 estudantes (sendo recomendados grupos de 3 alunos).

7. Material a submeter no Moodle

Um dos alunos do grupo deverá submeter **via moodle até à data limite indicada no cabeçalho do enunciado**, os seguintes elementos:

Um ficheiro com o relatório de redação livre, com identificação dos alunos do grupo: nomes e números de aluno (*.pdf);

Arquivo compactado com todos os ficheiros com o código fonte da aplicação (*.zip, *.rar);

Ficheiro com dados dos atletas criados e carregados no arranque na aplicação (*.txt);

8. Apresentação, defesa e avaliação

A defesa do projeto é obrigatória, sendo o mesmo defendido individualmente, com a presença de todos os elementos do grupo de trabalho, em dia, horário e sala a serem definidos, oportunamente, na página do moodle. Deverão apresentar no vosso portátil.

A avaliação, individual, tem as seguintes componentes:

1. Funcionamento geral da aplicação (app a correr), i.e., utilização fácil e intuitiva por parte do utilizador, aplicação com capacidade de ultrapassar erros de entrada de dados ou sem estourar, apresentação clara dos resultados, etc.;
2. Estrutura do código da app, i.e., as opções tomadas para resolver o problema proposto, o código comentado, indentado, escrito segundo a convenção, a arrumação do código em funções, utilização de tipos de dados e estruturas adequadas, etc.;
3. Não se aceitam apps que não sejam de programação em C e que apresentem erros de compilação no dia da defesa;