

Programação com Objetos

Projeto de Grupo 2024/2025

Pretende-se desenvolver um programa para a gestão de uma liga de desportos eletrónicos (eSports). A liga organiza torneios entre equipas, permitindo a criação e acompanhamento de partidas, equipas e jogadores. As equipas são formadas por jogadores, que participam em uma ou mais competições (torneios). Os jogadores podem ser de três tipos: Jogador de FPS (First Person Shooter); Jogador de MOBA (Multiplayer Online Battle Arena); Jogador de eFootball.

Cada partida realizada no torneio envolve duas equipas e deve ser registado o resultado (vitória, derrota ou empate). Além disso, haverá uma tabela de classificação, para cada torneio, atualizada automaticamente com base nas vitórias e derrotas de cada equipa.

Entidades principais e dados a armazenar:

1. Jogador:

Cada jogador pertence a uma equipa e participa nos torneios. Os dados a armazenar incluem:

- **ID do jogador** (gerado automaticamente).
- **Nome completo.**
- **Nickname.**
- **Estatísticas básicas** (número de partidas jogadas, vitórias, derrotas).

JogadorFPS: Para jogadores que competem em jogos do género **FPS** (First-Person Shooter). Dados adicionais incluem:

- **Precisão** (percentagem de tiros acertados).
- **Headshots** (número de headshots feitos).

JogadorMOBA: Para jogadores de jogos do género **MOBA** (Multiplayer Online Battle Arena). Dados adicionais incluem:

- **Personagem principal** (herói ou campeão mais jogado).
- **Número de abates, mortes e assistências** (KDA - Kill/Death/Assist ratio).

JogadorEFootball: Para jogadores de jogos do género **eFootball**. Dados adicionais incluem:

- **Posição principal** (guarda-redes, defesa, médio, avançado).
- **Número de golos marcados/defendidos (no caso dos guarda-redes) e assistências.**

2. Equipa:

Cada equipa é composta por jogadores e participa nos torneios. Os dados a armazenar são:

- **ID da equipa** (gerado automaticamente).
- **Nome da equipa.**
- **Lista de jogadores.**
- **Número de vitórias e derrotas** (acumulado nas competições).

3. Treinador (Equipa):

O treinador é responsável pela gestão da equipa. Os dados a armazenar são:

- **ID do treinador** (gerado automaticamente).
- **Nome completo.**
- **Email.**
- **Equipa gerida** (relacionamento com a entidade "Equipa").

4. Administrador:

O administrador gere as competições e as equipas. Os dados incluem:

- **ID do administrador** (gerado automaticamente).
 - **Nome completo.**
 - **Email.**
-

4. Torneio:

Cada torneio é uma competição onde as equipas se enfrentam. Os dados a armazenar incluem:

- **ID do torneio** (gerado automaticamente).
 - **Nome do torneio.**
 - **Jogo** (CS2, eFootball, LoL)
 - **Equipas participantes.**
 - **Lista de partidas.**
 - **Tabela de classificação.**
-

5. Partida:

Cada partida ocorre entre duas equipas num torneio e deve registar o resultado. Os dados incluem:

- **ID da partida** (gerado automaticamente).
 - **Equipa A.**
 - **Equipa B.**
 - **Resultado final** (Pontos equipa A; Pontos equipa B).
 - **Data da partida.**
-

Funcionalidades principais:

- **Jogador:**
 1. Autenticar-se no sistema.
 2. Ver/editar os seus dados pessoais (nome, nickname).
 3. Ver os torneios em que participa.
 4. Ver as suas estatísticas pessoais.
- **Treinador (Equipa):**
 1. Autenticar-se como treinador.
 2. Criar e gerir a equipa (adicionar jogadores, remover, editar).

3. Inscrever a equipa em torneios disponíveis.
 4. Acompanhar as partidas, resultados e a posição da equipa nos torneios.
- **Administrador (Organização da Liga):**
 1. Autenticar-se como administrador.
 2. Adicionar/apagar utilizadores do tipo Jogador e do tipo Treinador
 3. Criar e gerir torneios.
 4. Agendar partidas entre as equipas inscritas.
 5. Registar os resultados das partidas e atualizar as classificações.
 6. Acompanhar as estatísticas e resultados dos torneios.
 - O programa deverá ser composto por duas partes: a parte de back-end que suporta o funcionamento do sistema, e a parte de front-end que suporta a interface que permite a interação com o utilizador. O back-end e o front-end deverão estar implementados em pacotes distintos, ou seja, todas as classes que digam respeito à interface deverão estar arrumadas num pacote e as classes respeitantes ao negócio, noutra. A entrada e saída de dados deve ser realizada recorrendo a uma interface em modo de texto (do género da que foi usada em AED1 e AED2).
 - O programa deve permitir persistência de dados, gravando o estado do sistema em ficheiros, permitindo a recuperação dos mesmos em futuras sessões.
 - O programa deve ser robusto, capaz de lidar com erros de entrada e manter a integridade dos dados (por exemplo: evitar a duplicação de equipas, jogadores, etc.).

Valorização

A aplicação dos conceitos e dos princípios fundamentais subjacentes ao paradigma de programação orientada aos objetos será amplamente valorizada.

A classificação dos trabalhos terá ainda em consideração aspetos como:

- Qualidade da execução do programa (com ou sem erros, validações, testes, ...)
- Qualidade da satisfação dos requisitos
- Facilidade de utilização do sistema
- Adequação da conceção e organização das classes
- Qualidade do código fonte apresentado
 - Correção (avaliação se o código está coerente com a especificação do enunciado e se não tem erros)
 - Design (avalia a forma como o código está escrito (eficiente, elegante, reutilizável))
 - Estilo (avalia a legibilidade do código (comentado, indentado, nomes de variáveis e funções adequadas,...))
- Desempenho do grupo no decorrer da apresentação

Datas e Formalidades de Entrega

O trabalho deve ser realizado em **grupos de 3 estudantes inscritos no mesmo turno PL** (só em casos muito excepcionais e devidamente justificados serão aceites grupos com menos de 3 estudantes). O projeto, em formato zip, deve ser submetido no BlackBoard até **15 de dezembro de 2024**. Juntamente com o projeto deve ainda ser entregue um relatório básico (em formato pdf) incluindo uma tabela contendo todas as funcionalidades implementadas e não implementadas.