

**Projeto Prático #2:** O projeto deve ser desenvolvido por grupos de *no máximo* três alunos. É estritamente proibida a partilha de código entre alunos de grupos diferentes.

**Plágio:** Além de inspeção manual, será também executado um software de deteção de plágio no código-fonte do projeto. *Todos* os alunos que submeterem código obtido através de plágio terão os seus projetos anulados.

**Uso de Inteligência Artificial:** É permitido utilizar ferramentas de IA como o ChatGPT, Copilot, DeepSeek e similares. No entanto, todos os membros do grupo devem ser capazes de compreender e *explicar* o projeto *na sua totalidade* aos docentes quando pedido. Se houver código no projeto que o grupo não consiga explicar, o projeto será considerado como *plágio*.

**Submissão:** A data de submissão do projeto é **03/06/2025** às 23:59. A descrição do processo de submissão pode ser encontrada no enunciado.

## SoccerNow: Gestão de Jogos de Futsal

### 1 Contextualização

O clube desportivo **Ciências Soccer Society (CSS<sup>1</sup>)** organiza, de forma recorrente, um conjunto de jogos e torneios de futsal. Surge, assim, a necessidade de um sistema que cubra todas as etapas deste processo.

Um utilizador pode interagir com o sistema de duas formas: como jogador ou como árbitro. Os jogadores podem integrar várias equipas e devem ser capazes de indicar a posição que preferem jogar. Por sua vez, o papel de árbitro pode ser desempenhado por um utilizador com ou sem certificado. Jogadores não podem ser árbitros e vice-versa.

Para a gestão de equipas, o sistema permite criar equipas com um nome, manter um histórico dos jogos em que participaram e das conquistas, isto é, posições de pódio em campeonatos. É também necessário registar os jogadores que compõem cada equipa. As equipas não têm limite de tamanho, e cada jogador pode integrar várias equipas.

No que diz respeito aos jogos de futsal, deverá ser possível criar jogos amigáveis, sem associação a qualquer campeonato, bem como jogos que façam parte de um campeonato. A cada jogo associa-se uma data, horário, local, equipas e estatísticas relevantes. Os organizadores devem poder atribuir um ou mais árbitros aos jogos. Se houver mais de um árbitro, um deles deverá ser considerado o principal. Cada jogo tem duas equipas de cinco (5) jogadores, dos quais um será designado como guarda-redes.

A criação e organização de campeonatos envolve a definição de diferentes modalidades. Todos os campeonatos serão disputados por pontos, mas pretende-se estender (no futuro) o sistema para que seja suportada também a funcionalidade de campeonatos no formato de *eliminatória*. Um campeonato é disputado por um mínimo de 8 equipas e é necessária a presença de, pelo menos, um árbitro certificado nas partidas de campeonato.

Por fim, o registo de resultados e estatísticas deve permitir que, no final de cada jogo, se registe o resultado final (placar), a equipa vitoriosa, bem como quaisquer cartões atribuídos a jogadores. Em competições por pontos, o sistema atualiza a pontuação e a classificação das equipas. No formato de eliminatória, define-se quem avança para a fase seguinte.

---

<sup>1</sup>Entidade fictícia

## 2 Casos de Uso

Apresenta-se a lista completa dos casos de uso que devem ser desenvolvidos durante a segunda fase.

- A. Login com autenticação (Nota: vamos fazer mock e qualquer palavra-passe será aceite).
- B. Registo de utilizadores.
- C. Verificar, remover e atualizar um utilizador.
- D. Buscar por jogadores.
- E. Criação de equipas.
- F. Verificar, remover e atualizar as equipas.
- G. Buscar por equipas
- H. Criação de jogos.
- I. Registrar o resultado de um jogo.
- J. Criação de campeonatos.
- K. Buscar por campeonatos.
- L. Verificar, remover e atualizar os campeonatos.
- M. Cancelamento de um jogo de campeonato.
- N. Filtrar a busca dos jogadores.
- O. Filtrar a busca dos árbitros.
- P. Filtrar a busca das equipas.
- Q. Filtrar a busca dos jogos.
- R. Filtrar a busca dos campeonatos.

## 3 Requisitos Não Funcionais

- A autenticação poderá ser feita via mock (futura integração com sistema real).
- Toda a informação deverá ser armazenada numa base de dados relacional.
- A plataforma deverá ser implementada em Spring Boot, usando Java. A camada de dados será implementada utilizando JPA/Spring Data.
- A aplicação deverá lidar com pedidos concorrentes, sem criar inconsistências.
- A camada de negócios deverá usar um Domain Model rico e anotações JPA para facilitar o mapeamento.

- A interface gráfica implementada de duas maneiras, cada uma delas deverá realizar uma parte dos casos de uso. Uma das interfaces será através do browser numa interface web e utilizando uma interface nativa com JavaFX.
- O repositório deve aceitar apenas código com o nível de qualidade aceite pela equipa (ex.: *pre-commit*, testes).
- O projecto correrá num ambiente Docker (será a forma de deploy no servidor de produção).
- O controlo da participação de cada membro do grupo será feito via atividade no repositório git, com base no número e qualidade dos *commits*.
- As decisões técnicas devem ser justificadas num relatório técnico.

## 4 Tarefas da Fase 2

As tarefas da fase 2 presumem que as tarefas da fase 1 foram realizadas com sucesso. Caso alguma das tarefas ainda esteja pendente, é necessária a sua conclusão antes de iniciar a fase 2.

O cliente notou que o sistema de buscas era muito simples e portanto pediu que fossem adicionados novos casos de uso para permitir buscas mais complexas através do uso de filtros (Casos de uso N, O, P, Q e R). Os filtros que devem ser implementados são os seguintes:

- Filtros do Jogador:
  - Filtro por nome.
  - Filtro por posição.
  - Filtro por número de golos marcados.
  - Filtro por número de cartões recebidos.
  - Filtro por quantidade de jogos realizados.
- Filtros do Árbitro:
  - Filtro por nome.
  - Filtro por número de jogos oficiados.
  - Filtro por número de cartões mostrados.
- Filtros da Equipa:
  - Filtro por nome.
  - Filtro por número de jogadores.
  - Filtro por número de vitórias, empates ou derrotas.
  - Filtro por conquistas.
  - Filtro por ausência de jogadores numa posição específica.
- Filtros do Jogo:
  - Filtro por jogos realizados.
  - Filtro por a serem realizados.

- Filtro por quantidade de golos.
- Filtro por localização.
- Filtro por turno do jogo (manhã, tarde ou noite).
- Filtros do Campeonato:
  - Filtro por nome.
  - Filtro por por equipa.
  - Filtro por número de jogos realizados.
  - Filtro por número de jogos a realizar.

Pretende-se implementar a interface gráfica que permita aos utilizadores realizarem os casos de uso de forma interativa. A interface gráfica será desenvolvida de duas formas diferentes: i) através do browser, numa interface web, com renderização server-side onde será possível realizar as consultas a informação do sistema (casos de uso marcados a **vermelho**) e através de uma aplicação nativa desenvolvida em JavaFX, onde será possível inserir dados ao sistema (casos de uso marcados a **azul**). Os casos de uso em **castanho** devem ser implementadas nas duas interfaces. De forma a demonstrar a funcionalidade do sistema, devem fazer um vídeo para cada interface gráfica a mostrar o funcionamento do sistema.

## 5 Como Entregar

Para entregar o trabalho, basta criar uma tag chamada *fase2* e enviá-la para o repositório. O repositório deverá conter o código-fonte do projecto, os ficheiros necessários para o correr num ambiente Docker, um relatório de decisões técnicas (em PDF) e os vídeos da utilização do sistema através da interface gráfica.

```
git tag fase2
git push origin fase2
```

Deve-se confirmar que o vosso projecto está acessível à conta `css000` na tag *fase2*. Caso contrário, terão 0 nesta entrega. É importante garantir que o ambiente de execução possa ser reproduzido pela equipa docente sem erros de compilação ou outros que impeçam a sua execução. Para tal, recomenda-se executar o projecto em cada um dos computadores dos membros da equipa. O projecto **obrigatoriamente** deverá estar num ambiente Docker. Falhas em **EXECUTAR** o projecto, num ambiente Docker, pela equipa docente, resultarão em **grave penalização nesta entrega**.

## 6 Critérios de Avaliação

Esta entrega corresponde a 60% da nota do projecto. Os restantes 40% foram avaliados durante a primeira fase. Para perceberem onde se devem focar, segue uma lista de critérios que serão avaliados:

### 6.1 Processo de Desenvolvimento

- Granularidade e frequência dos *commits*. Todos os membros do grupo devem contribuir de forma equitativa. *Disparidades significativas serão penalizadas*.

- Qualidade das mensagens de *commit*.
- Qualidade do código presente no repositório.
- Quão bem os vários padrões leccionados são aplicados no projecto

## 6.2 Requisitos

- A percentagem dos requisitos implementados (ponderados pela sua complexidade)
- Qualidade da implementação dos requisitos nas escolhas técnicas associadas (apresentadas no relatório)
- Usabilidade da interface desenhada para cumprir os objectivos.

## 6.3 Relatório

- Descrição da arquitectura do projecto.
- Escolha e justificação das decisões técnicas tomadas na arquitectura da aplicação, suportadas pelos padrões desenvolvidos na aula
- Escolha e justificação das decisões técnicas no desenho da interface Web.
- Escolha e justificação das decisões técnicas no desenho da interface JavaFX.

---

**Nota Final:** Em caso de dúvidas, utilizem o fórum de discussão no Moodle. Boa sorte com o desenvolvimento do *SoccerNow*!