



# **Construção de Sistemas de Software**

-

## **SoccerNow**

## **Introdução**

O soccerNow é um sistema de gestão de jogos de futsal desenvolvido para o clube desportivo Ciências Soccer Society (CSS). O sistema permite a gestão completa de jogadores, árbitros, equipas, jogos e campeonatos, cobrindo todas as etapas do processo organizativo de eventos desportivos.

## **Contexto e motivação**

O mundo desportivo moderno exige sistemas de gestão eficientes que permitam organizar eventos, gerir participantes e manter registos precisos. O futsal, sendo um desporto com regras específicas e necessidades particulares de organização, beneficia de um sistema dedicado que compreenda as suas particularidades.

O SoccerNow foi concebido para responder às necessidades específicas do CSS, oferecendo uma solução completa que vai desde o registo de jogadores até à organização de campeonatos complexos, passando pela gestão de resultados e estatísticas.

## **Objetivos**

- Gestão de utilizadores: Sistema para jogadores e árbitros com perfis específicos
- Administração de equipa: Criação, edição e gestão de equipas com históricos completos
- Organização de jogos: Suporte para jogos amigáveis e de campeonato com validações rigorosas
- Criação e gestão de campeonatos: Sistema flexível preparado para diferentes modalidades
- Registo de estatísticas: Acompanhamento detalhado do desempenho individual dos jogadores
- Interface de consulta: Sistema de pesquisa e filtragem avançado para análise de dados

## Arquitetura do Sistema

O sistema foi desenvolvido seguindo uma arquitetura em camadas baseada no padrão MVC (Model-View-Controller), com separação clara de responsabilidades:

- Camada de apresentação: Duas interfaces distintas
  - JavaFX para inserção e gestão de dados (interface administrativa)
  - Thymeleaf para consultas e visualização (interface de consulta)
- Camada de controlo: Controllers REST e Web Controllers para gestão de pedidos
- Camada de Negócio: Services com lógica de negócio e validações
- Camada de dados: Repositórios com JPA/Spring Data para abstração da persistência
- Camada de Persistência: Base de dados PostgreSQL com esquema relacional normalizado

## Entidades Principais

- Utilizadores (Hierarquia de herança)
  - Jogadores
  - Árbitros
- Equipas
- Campeonatos
- Jogos
- Estatísticas

## Relações

- Jogador – Equipa: Many-to-Many (um jogador pode jogar em várias equipas)
- Jogo – Árbitro: Many-to-Many (um jogo pode ter vários árbitros)
- Jogo – Árbitro Principal: Many-to-One (um jogo tem um árbitro principal)
- Jogo – Campeonato: Many-to-One (jogos podem ser amigáveis – sem campeonato)
- Estatísticas – Jogador: Many-to-One (um jogador tem várias estatísticas)
- Estatísticas – Jogo: Many-to-One (um jogo tem estatísticas de vários jogadores)

## **Validações e Regras de Negócio**

- Número de camisa: únicos globalmente no sistema, para evitar complexidade no projeto
- Equipas de Jogo: Exatamente 5 titulares + 1 guarda-redes por equipa
- Árbitros Certificados: Pelo menos um árbitro certificado para jogos de campeonato
- Cartões: Máximo 2 amarelos e 1 vermelho por jogador/jogo
- Estados de Jogo: Transições controladas (AGENDADO -> EM\_ANDAMENTO -> FINALIZADO)

## **Constraints da Base de Dados**

- Chaves primárias: Auto-incremento para todas as entidades
- Unicidade: Username e email únicos para utilizadores
- Not Null: Campos obrigatórios identificados