



**Elaborado por**

**Grupo 01**

Isabel Maria dos Santos F. Dias Jesus (1200361)

Marlene Fernandes Lima (1200369)

Miguel Esteves Salgado (1200370)

Rute Daniela Tavares da Graça Lontro (1200375)

**Docentes**

André Miguel da Silva Mendes (MED)

Pedro da Silva Alves Magalhães Vasconcelos (VAS)

**Unidade Curricular**

Tópicos de Base de Dados

Ano Letivo 2020/2021

**Curso / Instituição**

Curso Técnico Superior em

Desenvolvimento Ágil de Software

Instituto Superior de Engenharia do Porto - IPP

Superliga Europeia de Futebol

Relatório do projeto final

22 de Junh0 de 2021

**ÍNDICE**

[Introdução 1](#_Toc75291099)

[Desenvolvimento 2](#_Toc75291100)

[1. Enquadramento do Projeto 2](#_Toc75291101)

[2. Modelo de Entidade-Relação 2](#_Toc75291102)

[a) Modelo de Entidade-Relação 2](#_Toc75291103)

[b) Desafios encontrados e proposta de soluções 3](#_Toc75291104)

[c) Pontos Fortes do Modelo 3](#_Toc75291105)

[Conclusão 5](#_Toc75291106)

[ANEXOS i](file:///C:\Users\ruteg\OneDrive\Ambiente%20de%20Trabalho\CTESP_1ANO\1%20ANO\2-semestre\DIAS\git-superliga\Documentos\RelatórioTBD_Grupo01_2206.docx#_Toc75291107)

[ANEXO A \_ Guia de Erros ii](#_Toc75291108)

**ÍNDICE DE FIGURAS**

[Figura 1 : Modelo de Entidade-Relação 2](file:///C:\Users\ruteg\OneDrive\Ambiente%20de%20Trabalho\CTESP_1ANO\1%20ANO\2-semestre\TDB\Trabalho%20Final\DocFinal%20TBD.docx#_Toc75271861)

**Introdução**

O presente relatório surge no âmbito da disciplina de Tópicos de Base de Dados que, em colaboração com a disciplina de Desenvolvimento Iterativo e Ágil de Software (DIAS), visa o desenvolvimento de um sistema de gestão da Superliga Europeia de Futebol (SLE).

Este relatório incide, essencialmente, no enquadramento e decisões tomadas pelo grupo, tendo em conta os pressupostos específicos desta disciplina, nomeadamente o planeamento, análise e estruturação do modelo de entidade-relação e o desenvolvimento da base de dados que integra o projeto.

**Desenvolvimento**

1. **Enquadramento do Projeto**

O projeto proposto visa a estruturação e desenvolvimento de um software de apoio à gestão da Superliga Europeia de Futebol (SLE), sendo este gerido apenas por um tipo de utilizador e ao qual deve ser permitido executar ações como:

* Consultar, criar, editar e eliminar todo o tipo de registos envolvidos neste sistema de gestão (nomeadamente clubes/equipas, estádios, épocas, jornadas, jogadores, jogos, eventos, entre outros);
* Consultar classificações gerais e por equipa;
* Consultar estatísticas provenientes dos dados armazenados (particularmente a consulta de melhores marcadores, cartões, equipas com mais vitórias e com mais derrotas, entre outros).

1. **Modelo de Entidade-Relação**

## Modelo de Entidade-Relação

Obs.:

Por motivos de simplificação, na tabela Jogo, optou-se por criar um **Id** como chave (primária) simples, em vez de constituir chave composta pela equipa\_casa, equipa\_fora, nr\_epoca, nr\_jornada.

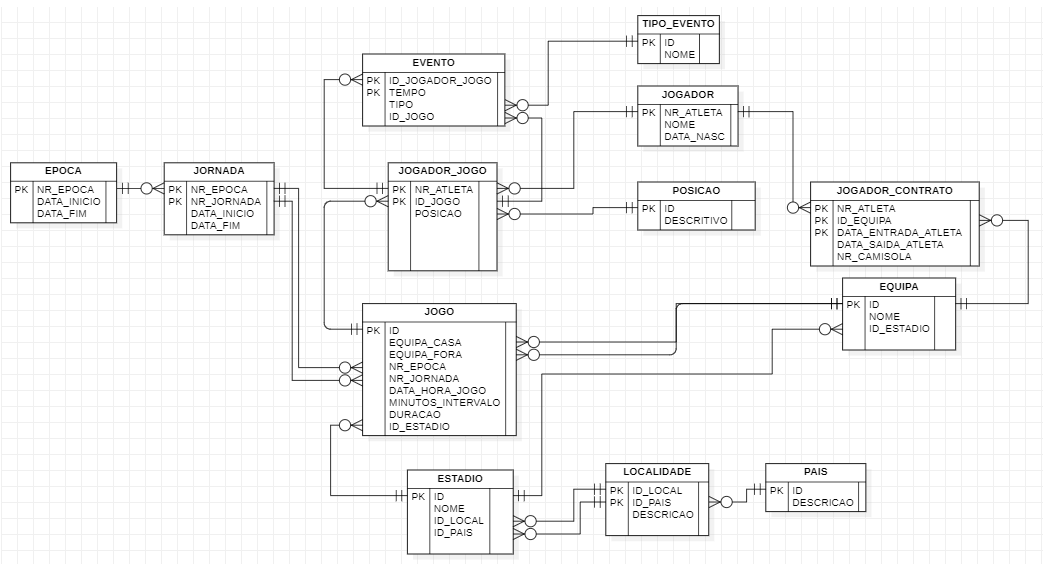


Figura 1 : Modelo de Entidade-Relação

## Desafios encontrados e proposta de soluções

Um dos maiores desafios com que o grupo se deparou foi a temática do projeto em si, uma liga de futebol, cujo programa é extenso e com uma certa complexidade, e ao qual nenhum dos elementos tinha um conhecimento mais consolidado.

Desta forma, e como se deve proceder no desenvolvimento de qualquer projeto, a maior preocupação da equipa foi a compreensão do tema, das dinâmicas e regras que terão de ser aplicadas neste software para garantir o seu bom funcionamento e ir de encontro ao que o cliente solicita.

O maior desafio e investimento de tempo da equipa foi, portanto, a análise do problema, clarificação de conceitos e consequentes soluções, nomeadamente como funciona uma liga de futebol, análise do fluxo de um jogo, aspetos relevantes e a considerar na atribuição de cartões, jogadores e os demais eventos que influenciam o tratamento de dados.

Outro desafio foi a análise e seleção das views, user stored procedures, functions e triggers que se teria de desenvolver e que teriam utilidade no projeto.

## Pontos Fortes do Modelo

O grupo considera que a análise elaborada de forma mais ponderada tornou o projeto final mais completo, integrando aspetos não solicitados no enunciado, mas que são situações reais e que conferem uma mais-valia e que complementam o projeto.

Desta forma, o modelo está preparado estruturalmente para incorporar mais do que uma época da superliga.

Além disso, analisando casos reais onde os jogos podem ocorrer em estádios que não pertencem nem à equipa local nem à equipa visitante, estruturou-se o presente modelo para incluir o fluxo de estádios e respetivas localizações/países, independentemente de pertencerem ou não às equipas envolvidas no jogo.

Houve sempre a preocupação de aplicar as boas práticas, tanto na padronização da nomenclatura e de programação, como na seleção de chaves compostas sempre que possível, usando apenas Id’s com auto incremento em casos pontuais, quando a alternativa era menos prática e/ou se justificava, em termos de simplificação.

Além disso, este projeto foi complementado com tratamento de erros, os quais podem ser consultados no [Anexo A](#_ANEXO_A__).

Neste projeto, desenvolveu-se os diferentes tipos de objetos:

|  |  |
| --- | --- |
| **Views:**   * Onze Inicial * Melhores marcadores gerais * Melhores equipas marcadoras * Cartões gerais por jogador * Cartões gerais por equipa * Equipa com mais vitórias/derrotas | **User Stored Procedures (USP):**   * usp\_registarJogo (estadio, equipa\_casa, equipa\_fora, nr\_epoca, nr\_jornada, data\_hora, minutos\_intervalo, duracao) * usp\_registarJogadorJogo (id\_jogo, equipa, jogador, id\_posicao) * usp\_registarEvento (id\_evento, id\_jogo, id\_jogador\_jogo, tempo) |
|  |  |
| **Functions:**   * fn\_PontosEquipa (id\_equipa) * fn\_Vitorias (id\_equipa) * fn\_Derrotas (id\_equipa) | **Triggers:**   * trigger\_DoisCartoesAmarelos (on create) * trigger\_JogadorContrato (on create & update) |
|  |  |
| **Exceptions:**  Estão implícitas nos Procedures | **Indexes:**   * idx\_tipo\_evento\_nome |

Breve explicação dos tipos de objetos criados:

* View: **view\_OnzeInicialJogo**: Informação do onze inicial de um determinado jogo.
* View: **view\_MelhoresMarcadoresGerais**: Informação dos melhores marcadores gerais.
* View: **view\_MelhoresEquipasMarcadores**: Informação das melhores equipas marcadoras.
* View: **view\_CartoesGeraisEquipa**: Informação dos cartões gerais por equipa.
* View: **view\_CartoesGeraisJogadores**: Informação dos cartões por jogador.
* View: **view\_EquipaVitoriasDerrotas**: Informação das esquipas com mais vitórias/derrotas.
* Procedure (USP): **usp\_RegistarJogo**: Este procedure tem como objetivo registar o jogo.
* Procedure (USP): **usp\_RegistarJogadorJogo**: Este procedure tem como objetivo registar o jogador do jogo.
* Procedure (USP): **usp\_RegistarEvento**: Este procedure tem como objetivo registar um evento que ocorra no jogo.
* Trigger: **trigger\_DoisCartoesAmarelos**: Este trigger tem como objetivo criar um cartão vermelho quando existirem dois amarelos.
* Trigger: **trigger\_JogadorContrato**: Este trigger tem como objetivo impedir que um atleta esteja no ativo em mais que um clube.
* Index**: idx\_tipo\_evento\_nome**: Índice para pesquisa pelo nome do tipo de evento.
* Função: **fn\_PontosEquipa**: Esta função devolve a quantidade de pontos da equipa numa época.

PARAMETROS ENTRADA:

PONTOS\_VITORIAS

PONTOS\_EMPATES

PARAMETROS DE SAÍDA

Quantidade de Pontos

* Função: **fn\_Vitorias** : Esta função devolve a quantidade de vitórias, para uma equipa numa época.

PARAMETROS ENTRADA:

EPOCA

EQUIPA

PARAMETROS DE SAÍDA

Quantidade de Vitórias

* Função: **fn\_Derrotas**: Esta função devolve a quantidade de derrotas, para uma equipa numa época.

PARAMETROS ENTRADA:

EPOCA

EQUIPA

PARAMETROS DE SAÍDA

Quantidade de Derrotas

**Conclusão**

De uma forma geral, consideramos que o produto desta entrega (modelo de entidade-relação e a base de dados) está estruturado de acordo com o solicitado e vai de encontro aos objetivos deste projeto e do cliente final.

Este projeto permitiu aplicar e consolidar conhecimentos adquiridos no decorrer do semestre, bem como atribuir dinamismo à implementação do projeto e da sua interface com o utilizador.

# ANEXOS

## ANEXO A \_ Guia de Erros

**Formato de Erro:**

{Variador} {Camada Dependência} {Número De Tabela} {Reservado A Erros} {Reservado A Erros}

**Variador**:

Valor igual ou superior a 5, incrementa para dar maior alcance de erros caso necessário.

**Camada Dependência**:

Nível de dependência relacional das tabelas.

**Número de Tabela**:

Valor da posição na coletânea de nomes de tabelas da camada de dependência associada, ordenado alfabeticamente

**Reservado a Erros**:

Dígito reservado à atribuição de erros.

|  |  |  |  |
| --- | --- | --- | --- |
| Erro | Camada Dependência | Número de Tabela | Nome de Tabela |
| 51000 | 1 | 0 | Epoca |
| 51100 | 1 | 1 | Jogador |
| 51200 | 1 | 2 | Pais |
| 51300 | 1 | 3 | Tipo\_Evento |
| 52000 | 2 | 0 | Localidade |
| 53000 | 3 | 0 | Estadio |
| 54000 | 4 | 0 | Equipa |
| 54100 | 4 | 1 | Jornada |
| 55000 | 5 | 0 | Jogador\_Contrato |
| 55100 | 5 | 1 | Jogo |
| 56000 | 6 | 0 | Jogador\_Jogo |
| 57000 | 7 | 0 | Evento |

Exemplo:

* ***Error 54102***
  + Nível de dependência 4
  + Tabela número 1 (Jornada)
  + Erro identificado como '02’