

# **Gestão de Um Clube Desportivo**

José Moreira, 79671

Bruno Aguiar, 80177

Base De Dados, 2018

# Introdução

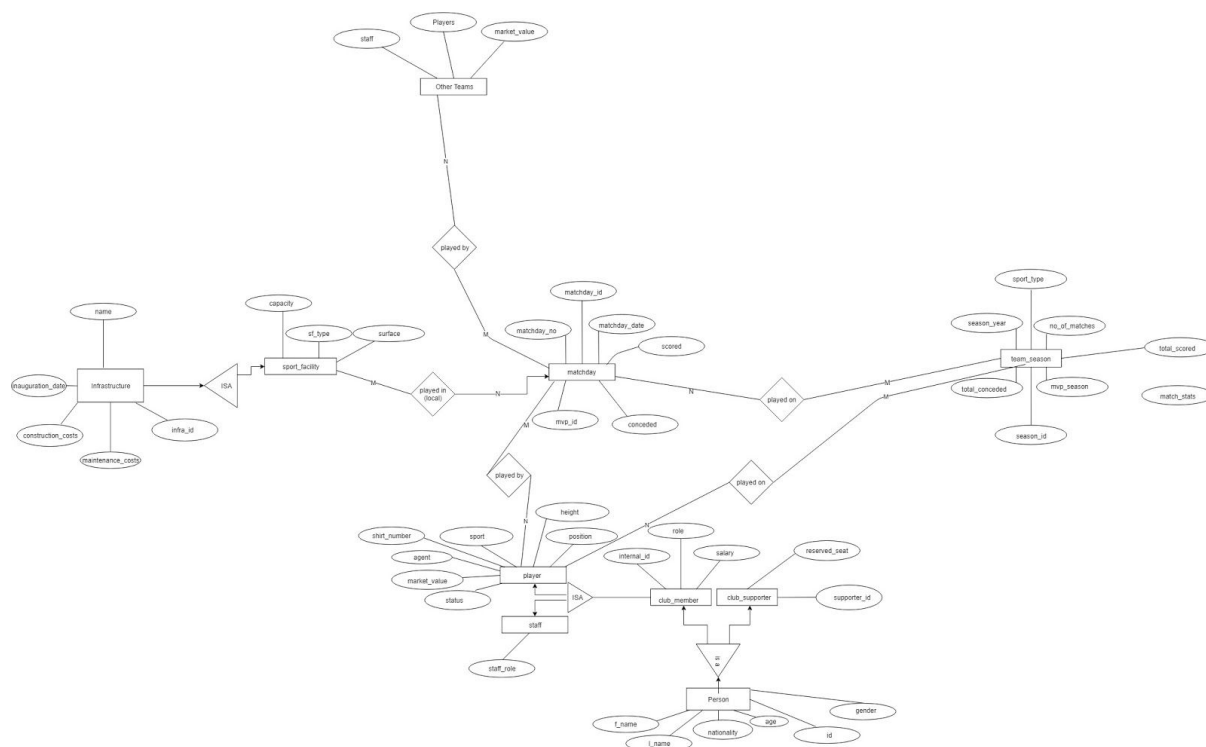
O projecto Gestão de um clube Desportivo tem como objectivo principal permitir ao cliente uma fácil, intuitiva e eficaz consulta e manipulação de informações do seu clube, como informações relativas a jogadores, staff, equipas adversárias, os seus sócios, partidas jogadas e temporadas mediante as modalidades, *et cetera*.

## Análise de Requisitos

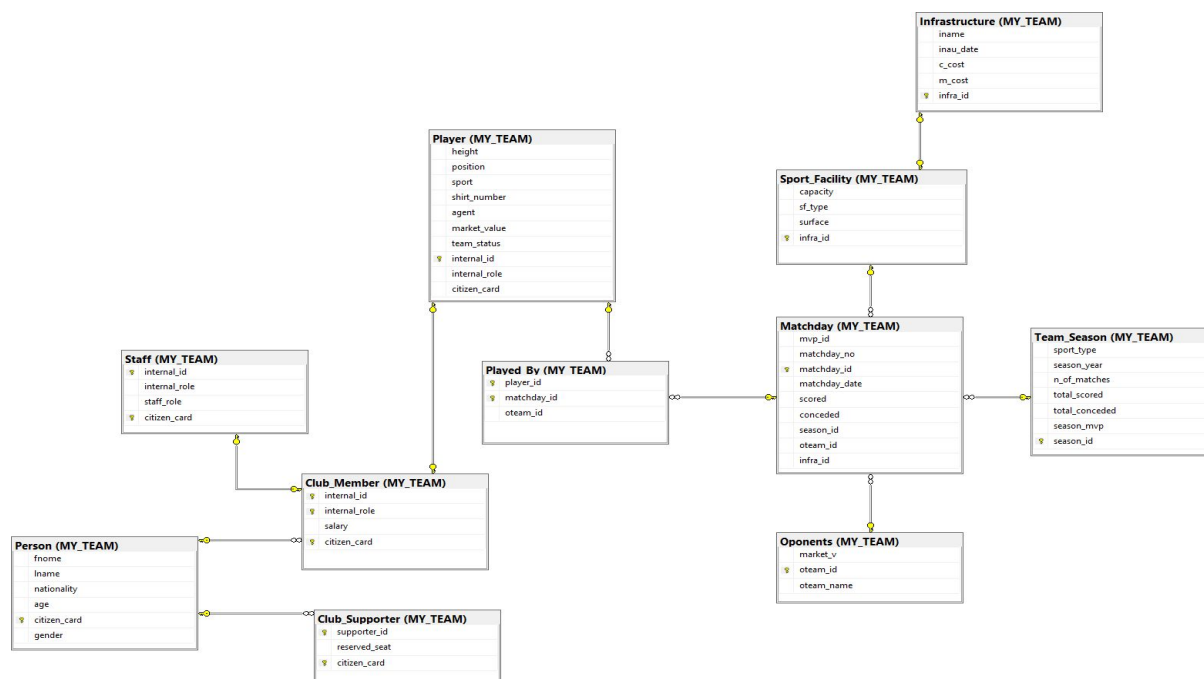
Entidades: Person, Club\_Member, Club\_Supporter, Staff, Player, Played\_By, Oponents, MatchDay, Sport\_Facility, Infrastructure, Team\_Season.

Relações: Played\_By, Played\_In, Played\_on.

## Esquema Relacional da BD



# Diagrama



Stored Procedure utilizados:

Foram utilizados 14 stored procedures, cuja utilidade é listar, por uma determinada ordem, as informações desejadas, e também adicionar ou remover informações de todas as entidades, promovendo assim uma comunicação interface/base de dados rápida, e cômoda.

Exemplo de dois stored procedures utilizados:

```

CREATE PROCEDURE MY_TEAM.Player_StoredProcedure
    @operation varchar(20), @delValue int = NULL, @fname varchar(20) = NULL, @lname varchar(20) = NULL,
    @nationality varchar(20) = NULL, @age tinyint = NULL, @citizen_card int = NULL, @gender varchar(20) = NULL,
    @internal_id int = NULL, @internal_role varchar(20) = NULL, @salary varchar(20) = NULL,
    @height int = NULL, @position varchar(20) = NULL, @sport varchar(20) = NULL,
    @shirt_number varchar(20) = NULL,
    @agent varchar(20) = NULL,
    @market_value int = NULL,
    @team_status varchar(20) = NULL
AS
BEGIN
    IF (@operation = 'insert') BEGIN
        INSERT INTO MY_TEAM.Person VALUES (@fname, @lname, @nationality, @age, @citizen_card, @gender);
        INSERT INTO MY_TEAM.Club_Member VALUES (@internal_id, @internal_role, @salary, @citizen_card);
        INSERT INTO MY_TEAM.Player VALUES (@height, @position, @sport, @shirt_number, @agent, @market_value, @team_status, @internal_id, @internal_role, @citizen_card);
    END
    IF (@operation = 'delete') BEGIN
        DECLARE @citizen_card1 int;
        SELECT @citizen_card1 = MY_TEAM.Club_Member.citizen_card FROM MY_TEAM.Club_Member WHERE MY_TEAM.Club_Member.internal_id = @delValue;
        DELETE FROM MY_TEAM.Player WHERE MY_TEAM.Player.internal_id = @delValue;
        DELETE FROM MY_TEAM.Club_Member WHERE MY_TEAM.Club_Member.internal_id = @delValue;
        DELETE FROM MY_TEAM.Person WHERE MY_TEAM.Person.citizen_card = @citizen_card1;
    END
END
GO
    
```

Funcionalidade: Inserir e remover jogadores. Necessárias alterações nas entidades Player, Club\_Member e Person. Procedimento muito semelhante para introduzir ou remover dados nas entidades TeamSeason, Supporter, Staff, SportFacility, Oponent e Matchday.

```
ListPlayersByX_Query.SV6.Cristiano (56) *
CREATE PROCEDURE MY_TEAM.ListPlayersByX_Query
    @arg varchar(20)
AS
BEGIN
    IF (@arg = 'salary')
    SELECT MY_TEAM.Person.fname, MY_TEAM.Person.lname, MY_TEAM.Person.age, MY_TEAM.Person.nationality, MY_TEAM.Player.sport, MY_TEAM.Player.position, MY_TEAM.Player.shirt_number,
        MY_TEAM.Player.team_status, MY_TEAM.Club_Member.salary
    FROM MY_TEAM.Player, MY_TEAM.Club_Member, MY_TEAM.Person
    WHERE MY_TEAM.Player.internal_id = MY_TEAM.Club_Member.internal_id AND MY_TEAM.Club_Member.citizen_card = MY_TEAM.Person.citizen_card
    ORDER BY len(MY_TEAM.Club_Member.salary) DESC, MY_TEAM.Club_Member.salary DESC;

    IF (@arg = 'name')
    SELECT MY_TEAM.Person.fname, MY_TEAM.Person.lname, MY_TEAM.Person.age, MY_TEAM.Person.nationality, MY_TEAM.Player.sport, MY_TEAM.Player.position, MY_TEAM.Player.shirt_number,
        MY_TEAM.Player.team_status, MY_TEAM.Club_Member.salary
    FROM MY_TEAM.Player, MY_TEAM.Club_Member, MY_TEAM.Person
    WHERE MY_TEAM.Player.internal_id = MY_TEAM.Club_Member.internal_id AND MY_TEAM.Club_Member.citizen_card = MY_TEAM.Person.citizen_card
    ORDER BY MY_TEAM.Person.fname ASC, MY_TEAM.PERSON.lname ASC;

    IF (@arg = 'age')
    SELECT MY_TEAM.Person.fname, MY_TEAM.Person.lname, MY_TEAM.Person.age, MY_TEAM.Person.nationality, MY_TEAM.Player.sport, MY_TEAM.Player.position, MY_TEAM.Player.shirt_number,
        MY_TEAM.Player.team_status, MY_TEAM.Club_Member.salary
    FROM MY_TEAM.Player, MY_TEAM.Club_Member, MY_TEAM.Person
    WHERE MY_TEAM.Player.internal_id = MY_TEAM.Club_Member.internal_id AND MY_TEAM.Club_Member.citizen_card = MY_TEAM.Person.citizen_card
    ORDER BY len(MY_TEAM.Person.age) ASC, MY_TEAM.PERSON.age ASC;
```

Funcionalidade: Listar informação dos jogadores, ordenada por qualquer atributo selecionado. Procedimento muito semelhante para listar todas as outras entidades.

## Proteção contra SQL Injection

Para proteção contra SQL Injection, todos os argumentos que são utilizados em Stored Procedures e que são inseridos pelo utilizador, têm predefinidos um SQL Parameter que não permitem qualquer “String” ou tipo de dados sejam inseridos, defendendo assim a aplicação de perda de dados em todas as entidades que têm algum tipo de “interação” com o utilizador.

## Utilização de Triggers

Foi utilizado um trigger para verificar o *internal\_id* inseridos na tabela “MY\_TEAM.Players”, no entanto este não está a funcionar a 100% conseguindo dar o erro quando o ID é menor que 1000 e maior que 9999, no então permite inserir na tabela.

## Conclusão

Após a realização deste trabalho, é possível concluir que os objetivos iniciais não foram totalmente alcançados. Embora a aplicação tenha algumas funcionalidades, não foi possível a inserção de triggers, ou índices, que certamente auxiliavam o seu bom funcionamento. No entanto, permite executar as tarefas mais comuns e importantes, que foram definidas no início, pois permite, de facto, uma gestão aceitável de jogadores, sócios, e bastantes outras informações relacionadas com um clube desportivo.