

# Jogos de tabuleiro

Relatório 2ª Fase



Universidade do Porto

Faculdade de Engenharia

**FEUP**

Mestrado Integrado em Engenharia Informática e  
Computação

Base de dados

**Grupo 601:**

Hugo Ari Rodrigues Drumond — 201102900

Ricardo Jorge Matos Figueiredo — 201100687

Gustavo Assis Freitas — 200602187

Faculdade de Engenharia da Universidade do Porto  
Rua Roberto Frias, 4200-65 Porto, Portugal

6 de Abril de 2014

## 1 Contexto

Esta base de dados destina-se a um Salão de jogos que organiza jogos de Tabuleiro e foi desenvolvida, inicialmente, pensando num só jogo, o xadrez. Posteriormente, então, foi feita uma generalização. Em suma, são guardados os dados dos jogadores e dos torneios de uma dada temporada. Isto, possibilitará aos utilizadores da base de dados: rigor ao planejar eventos, versatilidade, facilidade de registo, entre outros. Por exemplo, se eu quisesse organizar um torneio, em condições, iria ter saber à priori: os escalões, o jogo a que diz respeito, a temporada, as equipas inscritas e os patrocinadores. E registá-los em algum sítio para que depois possa associá-los, direta ou indiretamente, a uma partida e uma partida a equipas. A nossa base de dados tem como único propósito tornar esse {pré,pós}registo trivial.

## 2 Conceitos Principais

Num torneio de jogos de tabuleiro podem haver várias partidas entre duas ou mais equipas num dado escalão de um torneio. As partidas podem ocorrer em diversos sítios, têm de ser reguladas por um ou mais árbitros qualificados para o jogo em disputa e os resultados devem ser guardados. Cada equipa é formada por um ou mais jogadores.

## 3 Passagem ao modelo relacional e normalização

A transição do diagrama de classes para o schema da base de dados foi feita sem grandes problemas. Optámos por separar a generalização em duas tabelas segundo o estilo orientado a objetos, porque ao fazer isto ficamos entre o ponto de equilíbrio entre poupar nas ligações de tabelas e nos recursos de armazenamento. Uma relação, menos ligações embora haja desperdício de armazenamento; Três relações, mais ligações e mesmo uso de recursos de armazenamento. A única ternária no nosso uml foi decomposta em quatro tabelas: Equipa, Patrocinador, Torneio, EquipaPatrocinadorTorneio. A classe de associação sofreu uma transformação idêntica à ternária: Equipa, Partida, EquipaPartida.

Database Schema:

Jogador( idJogador, codigoPostal, dataNascimento, numeroAndar, rua, telefone, idPais->Pais, idCidade->Cidade, idExtensao->Extensao, email)

Equipa( nome, abreviatura)

JogadorEquipa( idEquipa->Equipa, idJogador->Jogador )

Árbitro( idÁrbitro, idPais->Pais, idCidade->Cidade, idExtensao->Extensao, observacoes )

LocalEncontro( idLocalEncontro, idCidade->Cidade, idExtensao->Extensao, codigoPostal, rua, telefone )

Cidade( idCidade, nome, idPais->Pais )

TipoJogo( idTipoJogo, nome )

ArbitroTipoJogo( idArbitro->Arbitro, idTipoJogo->TipoJogo )

Partida( idPartida, dataInicio, duracao, idEscalao->Escalao )

ArbitroPartida( idArbitro->Arbitro, idPartida->Partida

Escalao( idEscalao->Escalao, nome )

EquipaPartida( idEquipa->Equipa, idPartida->Partida, posicao, resultado )

Patrocinador( idPatrocinador->Patrocinador, nome )

EquipaPatrocinadorTorneio( idEquipa->Equipa, idPatrocinador->Patrocinador,  
idTorneio->Torneio )

## 4 Linguagem de Definição de dados e Restrições

Na nossa base de dados existem algumas restrições e gatilhos. Restrições de valor: NOT NULL, limites de atributo. Restrições de atributos: ainda não temos, temos de arranjar maneira de colocar isto em qualquer lado Asserção: arranjar uma maneira de colocar isto E obviamente que foram definidas chaves primárias e estrangeiras que são um tipo de restrição.

## 5 Linguagem de Manipulação de dados

## 6 Diagrama de classes UML melhorado