



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

Bases de Dados

# Plataforma de Competições de Programação Online

Grupo 708

23 de maio 2021

Filipe Campos,  
Gabriel Rocco,  
Vasco Alves,

up201905609  
up201800172  
up201808031

# Índice

<b>Descrição</b>	<b>3</b>
<b>Modelo UML</b>	<b>4</b>
<b>Modelo UML Revisto</b>	<b>5</b>
<b>Esquema Relacional</b>	<b>6</b>
<b>Análise de dependências funcionais e formas normais</b>	<b>8</b>
<b>Restrições</b>	<b>10</b>
<b>Interrogações</b>	<b>14</b>
<b>Gatilhos</b>	<b>15</b>

# Descrição

Este projeto representa uma forma de armazenar informações relativas a uma plataforma de competições online.

O foco da plataforma são os seus **Utilizadores**, sobre os quais interessa saber o seu id, username, nome, email e data do último login. Para tornar a comunidade mais dinâmica, é possível o envio de **Mensagens** entre dois jogadores, sendo guardado o texto e a data de envio.

Os utilizadores subdividem-se em dois grupos, os **Jogadores** e as **Empresas**.

Os **Jogadores** têm associado o número de competições em que participaram e um valor de rating que representa as suas capacidades, a única forma de alterar o seu rating é através da **Participação** numa **Competição**, sendo guardada a sua data de inscrição, posição no ranking da competição e a alteração de rating.

Uma **Competição** é organizada por pelo menos um Organizador. Um organizador é um jogador que é autorizado a contribuir pela comunidade podendo receber pontos por esta contribuição. Cada competição é caracterizada por um identificador único (id), título e descrição, sendo também registada a data de início e fim, número de participantes, dificuldade média, e possivelmente um prémio. Cada competição é composta por pelo menos um **Problema**, associado a um título, descrição e nível de dificuldade representada numa escala de 0 a 10. De modo a manter a competição justa, um organizador não se pode inscrever como jogador numa competição organizada por si próprio.

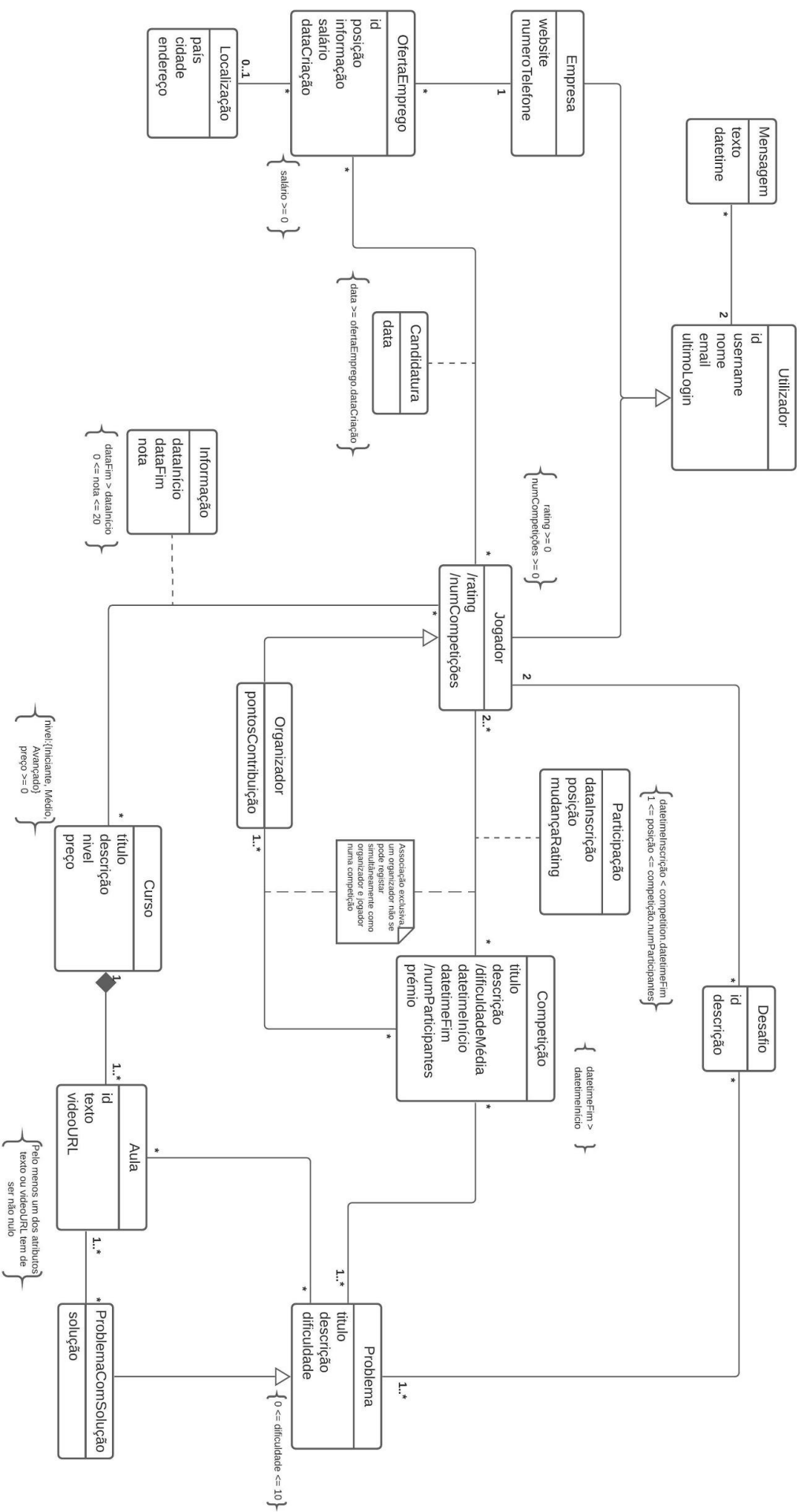
Cada jogador pode também desafiar um outro jogador para um **Desafio** que contém um id único, uma descrição e é composto por pelo menos um problema.

O objetivo desta plataforma é permitir que os seus utilizadores desenvolvam e melhorem as suas capacidades, por isso é fornecido um conjunto de **Cursos** caracterizados por um título, descrição, nível de dificuldade (Iniciante, Médio, Avançado) e preço, sendo composto por algumas **Aulas** (Uma aula pertence apenas a um único curso) por vídeo e/ou por texto, para desenvolver as capacidades práticas cada aula pode estar associada a **Problemas com Solução** e a problemas regulares. Qualquer jogador pode-se inscrever num curso sendo registada a data de início/fim e a sua nota (de 0 a 20).

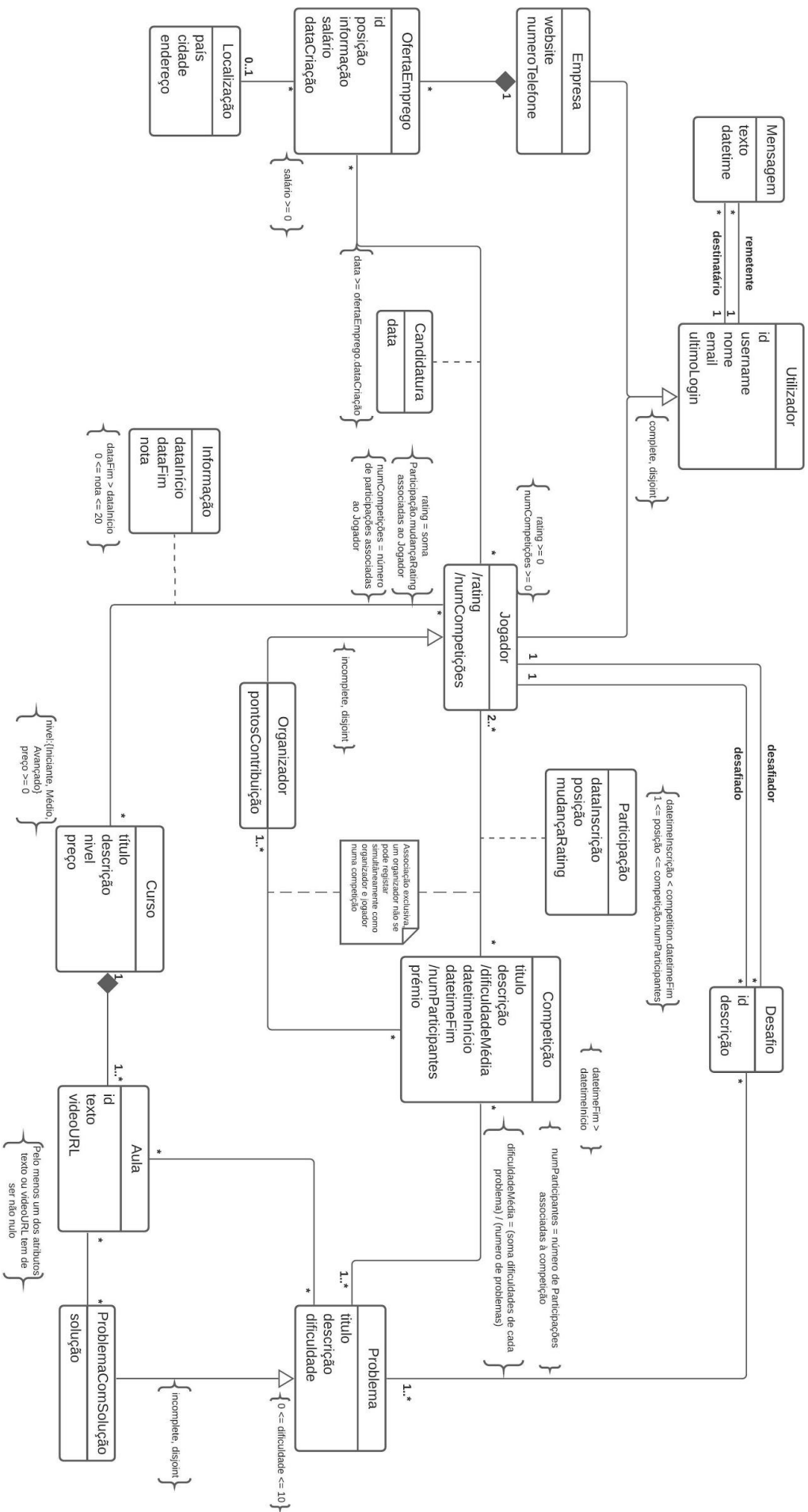
Na plataforma é possível também que empresas voltadas para o nicho de tecnologia se cadastrem para oferecer empregos, podendo ver os ratings dos jogadores.

Cada **Empresa** pode disponibilizar um website e número telefónico no seu perfil para facilitar a comunicação com os restantes utilizadores, e a sua utilidade principal é a criação de **Ofertas de Emprego**, sendo guardada o seu identificador único, a posição (i.e Backend Developer), a data de criação da oferta, o salário base e um pequeno texto com informações adicionais. Cada oferta pode também estar associada a uma **Localização** (país, cidade, endereço), no caso de trabalho remoto não é necessário especificar uma localização. Qualquer jogador pode-se candidatar a uma oferta de emprego, sendo guardada a data da candidatura.

# Modelo UML



# Modelo UML Revisto



# Esquema Relacional

**Utilizador**(id, username, nome, email, últimoLogin)

**Jogador**(idUtilizador → Utilizador, rating, numCompetições)

**Empresa**(idUtilizador → Utilizador, website, numeroTelefone)

**Organizador**(idJogador → Jogador, pontosContribuição)

**Mensagem**(id, idUtilizadorRemetente → Utilizador, idUtilizadorDestinatario → Utilizador, texto, datetime)

**Localização**(país, cidade, endereço)

**OfertaEmprego**(id, posição, informação, salário, dataCriação, [localPaís, localCidade, localEndereco] → Localização, idEmpresa → Empresa)

**Candidatura**(idOferta → OfertaEmprego, idJogador → Jogador, data)

**Competição**(id, título, descrição, dificuldadeMédia, datetimeInício, datetimeFim, numParticipantes)

**Participação**(jogadorId → Jogador, competiçãoId → Competição, dataInscrição, posição, mudançaRating)

**Contribuição**(organizadorId → Organizador, competiçãoId → Competição)

**Informação**(idCurso → Curso, idJogador → Jogador, dataInicio, dataFim, nota)

**Curso**(id, título, descrição, nível, preço)

**Aula**(id, texto, vídeoURL, idCurso → Curso)

**Problema**(id, título, descrição, dificuldade)

**ProblemaCompetição**(idProblema → Problema, idCompetição → Competição)

**ProblemaAula**(idAula → Aula, idProblema → Problema)

**ProblemaDesafio**(idProblema → Problema, idDesafio → Desafio)

**ProblemaComSolução**(idProblema → Problema, solução)

**Desafio**(id, idDesafiador → Jogador, idDesafiado → Jogador, descrição)

# Análise de dependências funcionais e formas normais

**Utilizador**(id, username, nome, email, últimoLogin)

{id} → {username, nome, email, últimoLogin}

{username} → {id, nome, email, últimoLogin}

{email} → {id, username, nome, últimoLogin}

**Jogador**(idUtilizador → Utilizador, rating, numCompetições)

{idUtilizador} → {rating, numCompetições}

**Empresa**(idUtilizador → Utilizador, website, numeroTelefone)

{idUtilizador} → {website, numeroTelefone}

**Organizador**(idJogador → Jogador, pontosContribuição)

{idJogador} → {pontosContribuição}

**Mensagem**(id, idUtilizadorRemetente → Utilizador, idUtilizadorDestinatario → Utilizador, texto, datetime)

{id} → {idUtilizadorRemetente, idUtilizadorDestinatario, texto, datetime}

**Localização**(país, cidade, endereço)

**OfertaEmprego**(id, posição, informação, salário, dataCriação, [localPaís, localCidade, localEndereco] → Localização, idEmpresa → Empresa)

{id} → {posição, informação, salário, localPaís, localCidade, localEndereco, idEmpresa}

**Candidatura**(idOferta → OfertaEmprego, idJogador → Jogador, data)

{idOferta, idJogador} → {data}

**Competição**(id, título, descrição, dificuldadeMédia, datetimeInício, datetimeFim, numParticipantes)

{id} → {título, descrição, dificuldadeMédia, datetimeInício, datetimeFim, numParticipantes}

**Participação**(jogadorId → Jogador, competiçãoId → Competição, dataInscrição, posição, mudançaRating)

{jogadorId, competiçãoId} → {dataInscrição, posição, mudançaRating}



**Contribuição**(organizadorId→Organizador, competiçãoId→Competição)

**Informação**(idCurso → Curso, idJogador → Jogador, dataInicio, dataFim, nota)  
{idCurso, idJogador} → {dataInicio, dataFim, nota}

**Curso**(id, título, descrição, nível, preço)  
{id} → {título, descrição, nível, preço}

**Aula**(id, texto, vídeoURL, idCurso → Curso)  
{id} → {texto, vídeoURL, idCurso → Curso}

**Problema**(id, título, descrição, dificuldade)  
{id} → {título, descrição, dificuldade}

**ProblemaCompetição**(idProblema → Problema, idCompetição → Competição)

**ProblemaAula**(idAula → Aula, idProblema → Problema)

**ProblemaDesafio**(idProblema → Problema, idDesafio → Desafio)

**ProblemaComSolução**(idProblema → Problema, solução)  
{idProblema} → {solução}

**Desafio**(id, idDesafiador → Jogador, idDesafiado → Jogador, descrição)  
{id} → {idDesafiador → Jogador, idDesafiado → Jogador, descrição}

Para cada relação, o lado esquerdo das suas dependências funcionais é uma chave dessa relação, portanto podemos concluir que o modelo relacional encontra-se na forma normal Boyce-Codd (**BCNF**), não existindo violações.

A partir desta informação também sabemos que o modelo encontra-se na terceira forma normal (**3NF**) pois esta é um subset da **BCNF**.

# Restrições

## Utilizador

- id é **PRIMARY KEY**
- username e email deverão ser **UNIQUE**
- O email deverá ter um formato válido (**CHECK** email LIKE “%@%.%”)
- últimoLogin deve ser não nulo (**NOT NULL**) e menor ou igual à data atual (**CHECK** últimoLogin <= now)

## Jogador

- idUtilizador é **PRIMARY KEY** de Jogador e **FOREIGN KEY** que referencia Utilizador
- Tanto rating como numCompetições devem ser maior ou igual a zero
  - **CHECK** rating >= 0
  - **CHECK** numCompeticoes >= 0

## Empresa

- idUtilizador é **PRIMARY KEY** e **FOREIGN KEY** que referencia Utilizador
- numeroTelefone é **NOT NULL** pois sempre deverá existir
- website precisa de ter um formato válido (**CHECK** website LIKE %.% , i.e google.com)
- [Nota] Tanto o número de telefone como o website de cada empresa não é único para acomodar sub-empresas que registam o número/telefone da empresa parente

## Organizador

- idJogador é **PRIMARY KEY** e **FOREIGN KEY** que referencia Jogador
- pontosContribuição é um valor não nulo (**NOT NULL**), com valor 0 por default (**DEFAULT**) e é sempre maior ou igual a 0 (**CHECK** pontosContribuição >= 0)

## Mensagem

- id é **PRIMARY KEY**
- idUtilizadorRemetente é **FOREIGN KEY** não nula (**NOT NULL**) que referencia Utilizador
- idUtilizadorDestinatario é **FOREIGN KEY** não nula (**NOT NULL**) que referencia Utilizador
- texto é **NOT NULL**
- datetime é **NOT NULL** e necessita de ser menor ou igual à data atual (**CHECK** datetime <= dataAtual)

### Localização

- [país, cidade, endereço] é a chave primária (**PRIMARY KEY**)

### OfertaEmprego

- id é a chave primária (**PRIMARY KEY**)
- posição é não nula (**NOT NULL**)
- O salário, se especificado, tem um valor positivo (**CHECK salario >= 0**)
- dataCriação não pode ser nula (**NOT NULL**) e necessita de ser menor ou igual à data atual (**CHECK dataCriação <= dataAtual**)
- idEmpresa é uma chave estrangeira (**FOREIGN KEY**) não nula (**NOT NULL**) que referencia Empresa
- [localPaís, localCidade, localEndereco] é uma chave estrangeira (**FOREIGN KEY**) que referencia Localização, a chave pode ser nula ou não nula (**CHECK**)

### Candidatura

- [idOferta, idJogador] é a chave primária (**PRIMARY KEY**)
- idOferta é uma chave estrangeira (**FOREIGN KEY**) que referencia OfertaEmprego
- idJogador é uma chave estrangeira (**FOREIGN KEY**) que referencia Jogador
- dataCandidatura é não nula (**NOT NULL**), precisa de ser menor ou igual à data atual (**CHECK dataCandidatura <= dataAtual**) e tem de ser maior ou igual a OfertaEmprego.dataCriação (Será implementada na 3ª entrega)

### Competição

- id é a chave primária (**PRIMARY KEY**)
- título não pode ser nulo (**NOT NULL**)
- dificuldadeMédia está entre os valores 0 e 10 (**CHECK dificuldadeMédia >= 0 AND dificuldadeMédia <= 10**)
- datetimeInício e datetimeFim não podem ser nulas (**NOT NULL**) e a data de fim necessita ser maior que a data de início (**CHECK datetimeFim > datetimeInicio**)

### Participação

- [idJogador, idCompetição] é a chave primária (**PRIMARY KEY**)
- idJogador é uma chave estrangeira (**FOREIGN KEY**) que referencia Jogador
- idCompetição é uma chave estrangeira (**FOREIGN KEY**) que referencia Competição
- dataInscrição não pode ser nula (**NOT NULL**) e precisa de ser menor que a data de início da competição (Será implementada na 3ª entrega)
- posição só pode ser nula se a data de fim da Competição for posterior à data atual (Será implementada na 3ª entrega) e o seu valor deve estar entre 1 (**CHECK posicao >= 1**) e Competição.numParticipantes (Será implementada na 3ª entrega)

### Contribuição

- [idOrganizador, idCompetição] é a chave primária (**PRIMARY KEY**)
- idCompetição é uma chave estrangeira (**FOREIGN KEY**) que referencia Competição
- idOrganizador é uma chave estrangeira (**FOREIGN KEY**) que referencia Organizador

- As relações Participação e Contribuição são exclusivas, logo um Organizador não pode participar numa competição organizada por si mesmo.

#### Curso

- id é a chave primária (**PRIMARY KEY**)
- título não pode ser nulo (**NOT NULL**)
- nível tem que ser iniciante, médio ou avançado (**CHECK**)
- preço tem que ser maior ou igual a 0 (**CHECK** preço  $\geq$  0)

#### Problema

- id é a chave primária (**PRIMARY KEY**)
- título não pode ser nulo (**NOT NULL**)
- descrição não pode ser nula (**NOT NULL**)
- dificuldade é um valor não nulo (**NOT NULL**) que está entre os valores 0 e 10 (**CHECK** dificuldade  $\geq$  0 AND dificuldade  $\leq$  10)

#### Informação

- [idCurso, idJogador] é a chave primária (**PRIMARY KEY**)
- idCurso é uma chave estrangeira (**FOREIGN KEY**) que referencia Curso
- idJogador é uma chave estrangeira (**FOREIGN KEY**) que referencia Jogador
- dataInicio e dataFim não podem ser nulas (**NOT NULL**) e a data de fim necessita ser maior que a data de início (**CHECK** dataFim > dataInicio)
- nota está entre os valores 0 e 20 (**CHECK** nota  $\geq$  0 AND nota  $\leq$  20)

#### Aula

- id é a chave primária (**PRIMARY KEY**)
- pelo menos um dos atributos videoURL ou texto tem de ser não nulo (**NOT NULL**)
- idCurso é uma chave estrangeira (**FOREIGN KEY**) que referencia Curso
- videoURL deve ter um formato válido (**CHECK** videoURL LIKE “%.%”)

#### ProblemaComSolução

- idProblema é chave primária (**PRIMARY KEY**) e uma chave estrangeira (**FOREIGN KEY**) que referencia Problema
- solução não pode ser nula (**NOT NULL**)

#### Desafio

- id é a chave primária (**PRIMARY KEY**)
- idDesafiador é uma chave estrangeira (**FOREIGN KEY**) que referencia jogador e é não nula (**NOT NULL**)
- idDesafiado é uma chave estrangeira (**FOREIGN KEY**) que referencia jogador e é não nula (**NOT NULL**)
- descrição não pode ser nula

#### ProblemaCompetição

- [idProblema, idCompetição] é a chave primária (**PRIMARY KEY**)
- idProblema é uma chave estrangeira (**FOREIGN KEY**) que referencia Problema
- idCompetição é uma chave estrangeira (**FOREIGN KEY**) que referencia Competição

### **ProblemaAula**

- [idProblema, idAula] é a chave primária (**PRIMARY KEY**)
- idProblema é uma chave estrangeira (**FOREIGN KEY**) que referencia Problema
- idAula é uma chave estrangeira (**FOREIGN KEY**) que referencia Aula

### **ProblemaComSolucaoAula**

- [idProblemaComSolucao, idAula] é a chave primária (**PRIMARY KEY**)
- idProblemaComSolucao é uma chave estrangeira (**FOREIGN KEY**) que referencia ProblemaComSolucao
- idAula é uma chave estrangeira (**FOREIGN KEY**) que referencia Aula

### **ProblemaDesafio**

- [idProblema, idDesafio] é a chave primária (**PRIMARY KEY**)
- idProblema é uma chave estrangeira (**FOREIGN KEY**) que referencia Problema
- idDesafio é uma chave estrangeira (**FOREIGN KEY**) que referencia Desafio

# Interrogações

1. Mostrar os três melhores Jogadores, baseado no seu rating, que participaram em pelo menos cinco competições.
2. Mostrar o número de problemas resolvidos por cada jogador via competição, desafio ou aula. Problemas com solução de uma aula não são contabilizados.
3. Mostrar o número de ofertas de emprego para cada (país, empresa).
4. Obter todos os jogadores da FEUP que não são contribuidores, que não apresentaram nenhuma candidatura e que nunca participaram numa competição cuja última data de login seja mais recente que 1 de janeiro 2020.
5. Mostrar as mensagens num formato fácil de ler, mostrando o username em vez do id dos utilizadores e a data formatada em vez do valor unixepoch.
6. Para cada curso mostrar o número de aulas, número de problemas/problemas com solução, o número de alunos e a nota média.
7. Mostrar o username e a nota média dos jogadores que concluíram todos os cursos, mostrando também o curso mais recentemente terminado por esses mesmos jogadores.
8. Para cada utilizador mostrar o seu id, username, nome, email, tipo de utilizador e última data de login. O tipo de utilizador é uma string de ["Jogador", "Organizador", "Empresa"]. Como todos os organizadores são também jogadores, a string "Organizador" tem prioridade.
9. Mostrar as empresas ordenadas pela sua procura (número de candidaturas recebidas) e também o rating médio dos jogadores que se candidataram.
10. Uma das formas de saber os usuários mais ativos e que mais contribuem com a comunidade é filtrar os mesmos pelos seus pontos de contribuição e saber a quantidade de contribuições de cada um. Mostra o username, os pontos de contribuição e o número de contribuições, ordenado pelos pontos de contribuição.

# Gatilhos

1. Gatilho que garante que, quando ocorre uma inserção em Contribuição, é impossível um jogador contribuir para uma competição onde irá participar e no caso de ser uma contribuição válida, atualiza os pontos de contribuidor. (Nota, também foi adicionada um gatilho simétrico que impede que um jogador participe numa competição para a qual contribuiu, no entanto, por ser muito semelhante não deve ser contabilizada para avaliação.)
2. Gatilho que atualiza a dificuldade média de uma competição quando é adicionado um novo problema.
3. Gatilho que verifica se uma data de candidatura é válida, isto é, não pode ser anterior à data de criação da oferta.

Para utilizar os gatilhos deve-se correr os ficheiros sql na seguinte ordem:

`criar.sql`, `gatilhoN_adiciona.sql`, `povoar.sql`,  
`gatilhoN_verifica.sql`

Os ficheiros `gatilhoN_verifica.sql` inserem e selecionam dados dentro de uma transação que é revertida no final da sua execução, de modo a não inserir dados de teste na base de dados.