



Escola de Engenharia  
**Universidade do Minho**

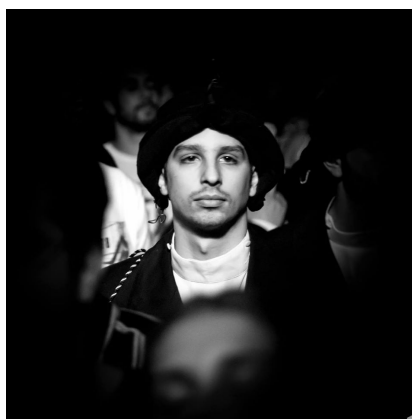
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA  
**Mestrado Integrado em Engenharia Informática**

## **Trabalho Prático**

### Processamento e Representação de Conhecimento



Tiago Baptista - A75328



Manuel Moreno - nr aluno

Braga, 8 de Julho de 2020

# Conteúdo

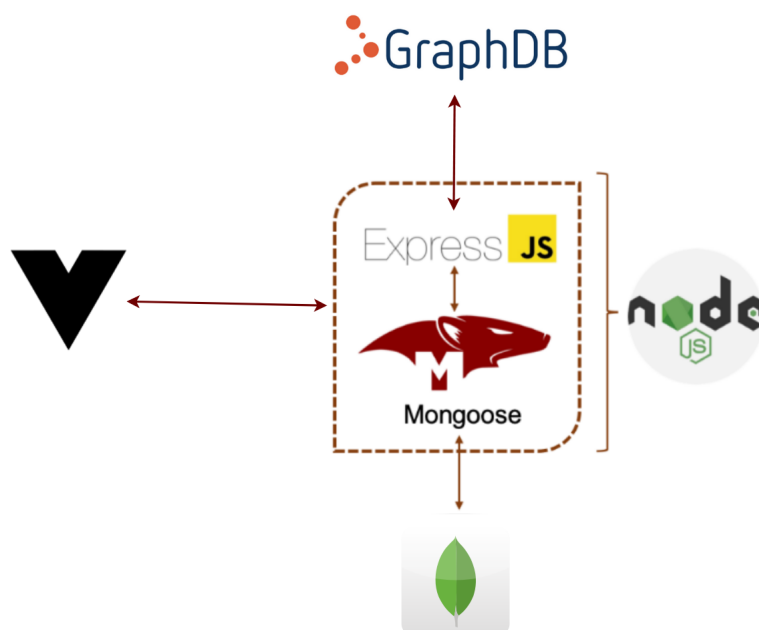
<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Opções de Desenvolvimento</b>	<b>3</b>
<b>3</b>	<b>Ontologia</b>	<b>4</b>
<b>4</b>	<b>Base de Dados</b>	<b>8</b>
4.1	MongoDb . . . . .	8
4.2	GraphDB . . . . .	8
<b>5</b>	<b>Autenticação</b>	<b>9</b>
<b>6</b>	<b>Conclusão</b>	<b>10</b>

# 1. Introdução

O trabalho descrito no presente documento foi realizado no âmbito da unidade curricular de Processamento e Representação de Conhecimento teve como principal objetivo a criação de uma aplicação utilizando a *framework* *Vue* e o *GraphDB* apoiado por uma ontologia de dimensão considerável.

Escolheu-se o desenvolvimento de uma aplicação com o intuito de funcionar como uma *fantasy league* simples utilizando jogadores do conhecido jogo *FIFA*.

No caso do desenvolvimento apresentado houve necessidade de utilizar o *mongodb* como auxílio para a gestão de utilizadores e outras operações relativas ao funcionamento da *fantasy league* em si.



**Figura 1.1:** Arquitetura da Aplicação Desenvolvida.

## 2. Opções de Desenvolvimento

Aquando o início do projeto existiram algumas decisão importantes, com necessidade de serem tomadas, dentro do grupo de trabalho. Assim sendo, o grupo optou por desenvolver o *frontend* utilizando o *vueify*. As restantes decisões tomadas foram relativas ao *backend*, desenvolvido em Node Js com auxílio da *framework* Express Js, e à base de dados, optando-se pela solução do MongoDB como complemento ao GraphDB.

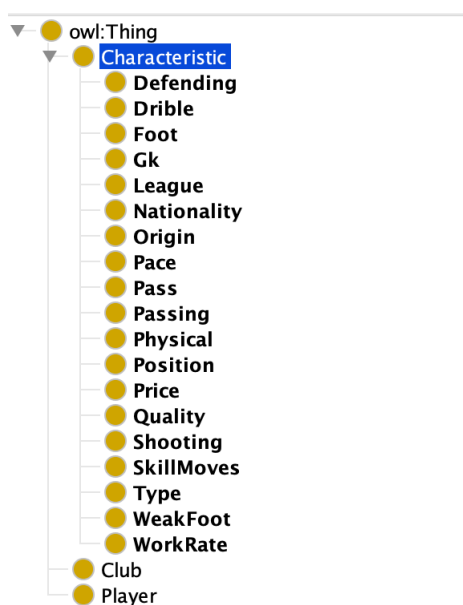
### 3. Ontologia

Após pesquisa por *datasets* relacionados com o *fifa* encontrou-se um dataset em *csv* com todos os jogadores e equipas presentes até Dezembro no *FIFA 20*.

De maneira a tornar estes dados utilizáveis foi necessário criar um *parser* de *csv* para *ttl* guardando a informação relevante para a aplicação na ontologia.

Para que o script fizesse sentido foi necessário definir a base da ontologia, para isso utilizou-se o *Protege*.

Como Classe principais criou-se a *Characteristic*, que inclui os valores de defesa, nacionalidade, entre outros da figura 3.1. Existem também a classe *Club* e *Player*, que se auto explicam pelo seu nome.



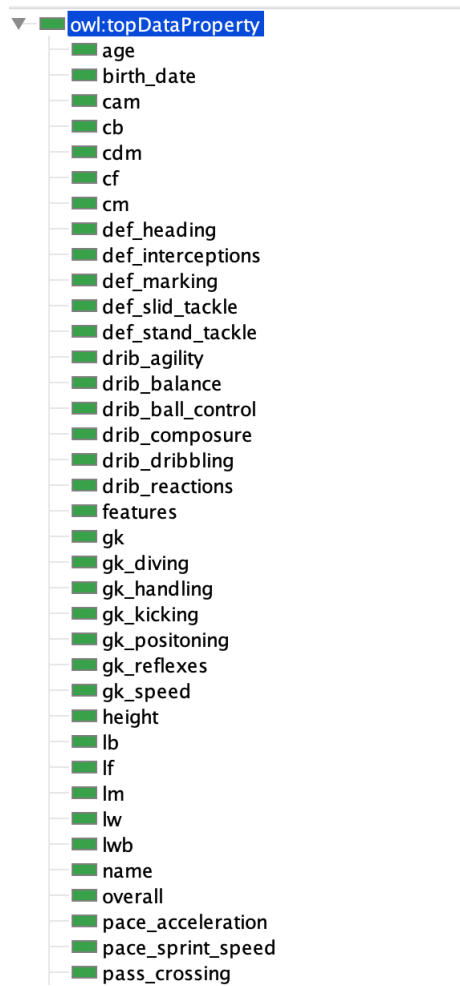
**Figura 3.1:** Classes

As object properties, que definem relações entre os elementos, são utilizadas para relacionar as classes referidas anteriormente, de destacar a *hasClub*, *isClubOf*, entre outras presentes na figura 3.2.



**Figura 3.2:** *Object Properties*

As *data properties* foram usadas para caracterizar os *Clubs*, *Characteristics* e *Players*. Como é possível verificar nas figuras 3.3 e 3.4.



**Figura 3.3:** *Object Properties*

Após a aplicação do *script* em *python* ao *dataset* em *csv*, obteve-se uma ontologia com 2 306 255 triplos.

gk_speed
height
lb
lf
lm
lw
lwb
name
overall
pace_acceleration
pace_sprint_speed
pass_crossing
pass_curve
pass_free_kick
pass_long
pass_short
pass_vision
pc_price_last
pc_price_max
pc_price_min
phys_aggression
phys_jumping
phys_stamina
phys_strength
ps_price_last
ps_price_max
ps_price_min
shoot_finishing
shoot_long_shots
shoot_penalties
shoot_positioning
shoot_shot_power
shoot_volleys
value
weight
xbox_price_last
xbox_price_max
xbox_price_min

**Figura 3.4:** *Object Properties*

Type:	Standard
Access:	Read/write
Total statements:	2,306,255
Explicit:	2,161,630
Inferred:	144,625
Expansion ratio (total/explicit):	1.07

**Figura 3.5:** Triplos



## 4. Base de Dados

### 4.1 MongoDB

Tal como já referido, a aplicação relatada no presente documento teve necessidade de ter uma base de dados em *MongoDB*. Vários modelos foram definidos e guardados em coleções distintas. Resumindo, existe uma coleção por cada tipo de dados. Para garantir a integridade dos dados, cada coleção possui um controlador único.

### 4.2 GraphDB

Como base para o carregamento da ontologia gerada utilizou-se o *GraphDB*. Para isso procedemos à instalação numa máquina remota (figura 4.1), assim como do servidor em *Node JS* e do resultado da *build* do *Vue*, para o caso não se passou para a versão *build* das duas tecnologias pois preferiu-se investir tempo no desenvolvimento do trabalho em vez de no *deploy* completo.

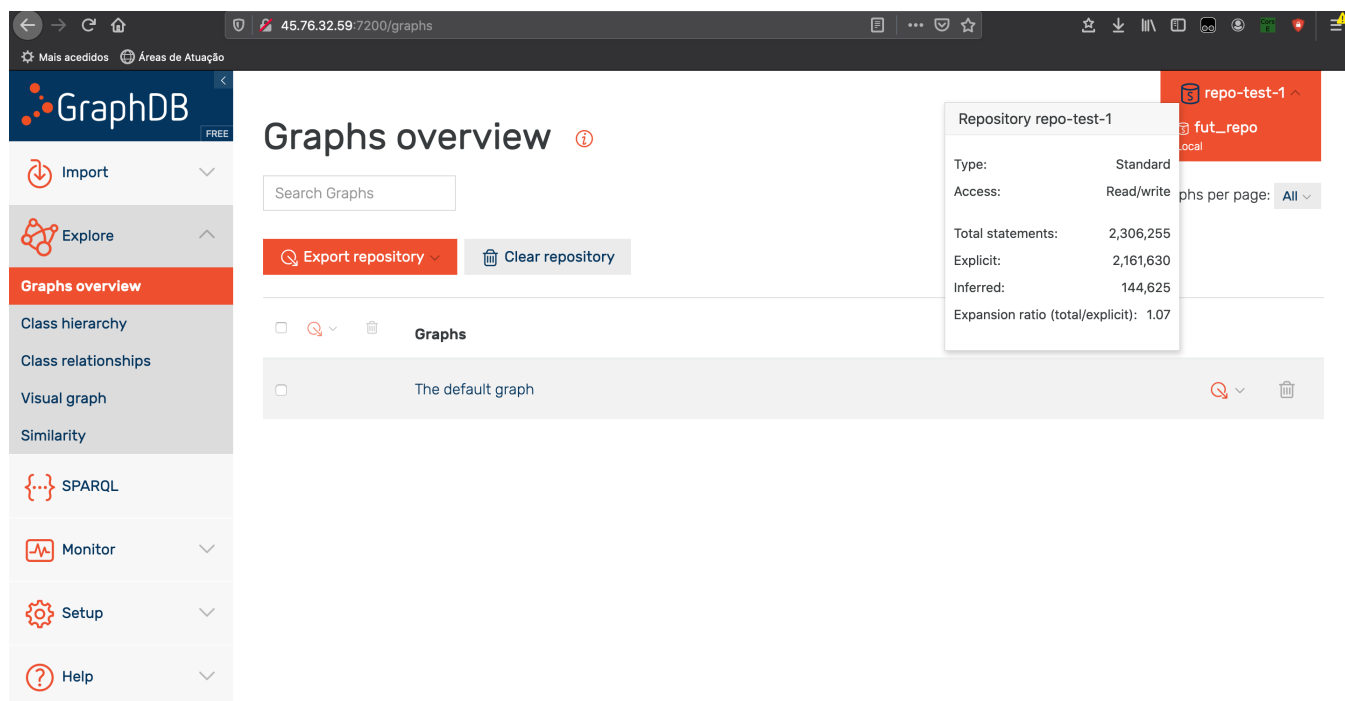


Figura 4.1: *Graphdb*

## 5. Autenticação

Para a autenticação utilizou-se a estratégia de sessões por tokens, usando jwt, aprofundado conhecimentos para além daquilo que foi leccionado na unidade curricular.

A quando do inicio da sessão é definida a duração da sessão e guardado no *token* o tipo de utilizador, permitindo desta forma controlar o acesso a rotas por tipo de utilizador. Necessitando apenas de acrescentar o metodo (verifier) como *middleware* nas rotas.

## 6. Conclusão

Este trabalho permitiu o desenvolvimento de capacidades a nível de desenvolvimento *web* nas duas vertentes de *front end* e *back end*.

A nível de trabalho desenvolvido ficou para trás um objetivo a que nos tínhamos proposto, o *deploy* completo da aplicação para uma versão de produção e o melhoramento do design da aplicação. Não se chegou a essa meta pois decidiu-se dar prioridade a funcionalidades do *core* da aplicação, como a criação de ligas, equipas, perfil do utilizador, assim como todo o *workflow* de simulação das jornadas e ligas .