



UNIVERSIDADE DO MINHO

MESTRADO EM ENGENHARIA INFORMÁTICA

PROCESSAMENTO DE LINGUAGENS E CONHECIMENTO

Processamento e Representação de Conhecimento



JOANA PEREIRA (A78275)

13 de Junho de 2019

Conteúdo

1	Introdução	1
2	Descrição do Problema	2
3	Descrição da Solução	2
3.1	Fonte de Dados	2
3.1.1	Ontologia	2
3.1.2	GraphDB	4
3.1.3	DBPedia	6
3.2	Especificação de Requisitos	7
4	Implementação	8
4.1	Express	8
4.2	Vue	10
5	Conclusões	17

Resumo

Este documento diz respeito ao trabalho prático proposto na unidade curricular de Processamento e Representação de Conhecimento da Universidade do Minho.

O objetivo deste projeto consiste no desenvolvimento e implementação de uma aplicação *web* para exposição de conteúdo recorrendo a *linked data*.

A aplicação desenvolvida segue como tema o *futebol*, e recorre a dados presentes no *GraphDB*, obtidos através do desenvolvimento de uma ontologia e povoamento com *datasets* existentes, e informação presente da *DBPedia*.

1 Introdução

Este trabalho prático enquadra-se na unidade curricular de Processamento e Representação de Conhecimento do Mestrado de Processamento de Linguagens e Conhecimento.

Linked Data é um dos focos principais da **Web Semântica**. O objetivo desta é a ligação entre dados de modo a que estes sejam compreensíveis a humanos e máquinas. *Linked Data* é um conjunto de princípios para partilha de informação ligada na *Web* e, por isso, fornece a melhor prática para ligar os dados.

Desta forma, é possível recorrer a *Open Linked Data* e ontologias, de forma a ser capaz de desenvolver uma aplicação *Web* que represente toda a informação ligada.

Neste contexto de desenvolvimento de uma aplicação *web* de apresentação de informação, foi tomada a decisão do desenvolvimento de uma ontologia e povoamento com *datasets* existentes, e com ligação a informação presente na *DBPedia*.

Estrutura do Relatório

No presente relatório, inicialmente, é feita uma pequena descrição do problema.

Posteriormente apresenta-se a descrição da solução utilizada, começando pela descrição da informação que vai ser utilizada e seguindo-se o levantamento e especificação de requisitos.

Depois é apresentado o método de implementação da aplicação, explicando as funções de cada um dos servidores utilizados.

Por último, é abordada uma conclusão sobre o projeto.

2 Descrição do Problema

O objetivo deste trabalho prático é desenvolver uma aplicação *web* para exposição de *linked data*¹ presente em *datasets*, ontologias, bases de dados ou recorrendo a *SparQL endpoints*.

3 Descrição da Solução

3.1 Fonte de Dados

O tema escolhido para o desenvolvimento deste projeto foi futebol. Os *datasets* utilizados foram os disponibilizados pela plataforma **FiveThirtyEight**².

Os ficheiros obtidos encontravam-se no formato *csv*, por isso foi necessário analisá-los de forma a criar a ontologia e desenvolver *scripts* para criar os indivíduos tendo por base o conteúdo destes ficheiros.

3.1.1 Ontologia

Após análise dos ficheiros *csv*, foram definidas as classes, *object properties* e *data properties* da ontologia. Aqui, foi utilizada a ferramenta **Protégé**³ para criar a ontologia.

Temos como resultado a ontologia composta pelas seguintes classes:

- **Team:** representa uma equipa de futebol.
 - **LeagueTeam:** subclasse de **Team**, representa uma equipa que joga numa liga de futebol.
 - **CountryTeam:** subclasse de **Team**, representa uma seleção de futebol.
- **League:** representa uma liga de futebol.
- **Confederation:** representa uma federação de futebol.
- **Game:** representa um jogo entre duas equipas.

¹Tipo de informação estruturada que está ligada a outros dados, tornando-se mais relevante para consultas semânticas.

²<https://data.fivethirtyeight.com>

³<https://protege.stanford.edu>

Temos também as seguintes *object properties*:

Nome	<i>Domain</i>	<i>Range</i>
:hasConfederation	:CountryTeam	:Confederation
:hasCountryTeam	:Confederation	:CountryTeam
:hasCountry	:League	dbo:Country
:hasGame	:League	:Game
:hasVictory	:Team	:Game
:hasDefeat	:Team	:Game
:hasDraw	:Team	:Game
:hasWinner	:Game	:Team
:hasLoser	:Game	:Team
:hasTie	:Game	:Team
:hasHomeTeam	:Game	:Team
:hasLeague	:Game	:League
:hasLeagueTeam	:League	:LeagueTeam
:hasVisitorTeam	:Game	:Team
:isHomeTeam	:Team	:Game
:isVisitorTeam	:Team	:Game
:playsIn	:LeagueTeam	:League

Tabela 1: *Object Properties* definidas na ontologia.

Finalmente, temos as seguintes *data properties* definidas:

Nome	<i>Domain</i>	<i>Range</i>
:date	:Game	xsd:string
:logo	:Team	xsd:string
:defense	:Team	xsd:float
:importanceHome	:Game	xsd:float
:importanceVisitor	:Game	xsd:float
:name		xsd:string
:offense	:Team	xsd:float
:prev_rank	:Team	xsd:integer
:rank	:Team	xsd:integer
:scoreHome	:Game	xsd:integer
:scoreVisitor	:Game	xsd:integer
:spi	:Team	xsd:float

Tabela 2: *Data Properties* definidas na ontologia.

Após ter a ontologia definida, foram desenvolvidos *scripts* em Python para criar os indivíduos de acordo com as classes e propriedades existentes. Utilizando o comando rapper verificamos a validade da ontologia criada e a quantidade de triplos existentes.

```
$ rapper -c -i turtle soccer.ttl

rapper: Parsing URI
file:///Users/Joana/Desktop/prc/prc-2019/datasets/ontology/soccer.ttl with parser
turtle
rapper: Parsing returned 366666 triples
```

Listing 1: Validação da ontologia criada.

3.1.2 GraphDB

A ontologia definida foi depois carregada para um repositório no **GraphDB**⁴.

Na figura seguinte, é possível ver o número de triplos carregados no repositório, e o número de triplos inferidos.

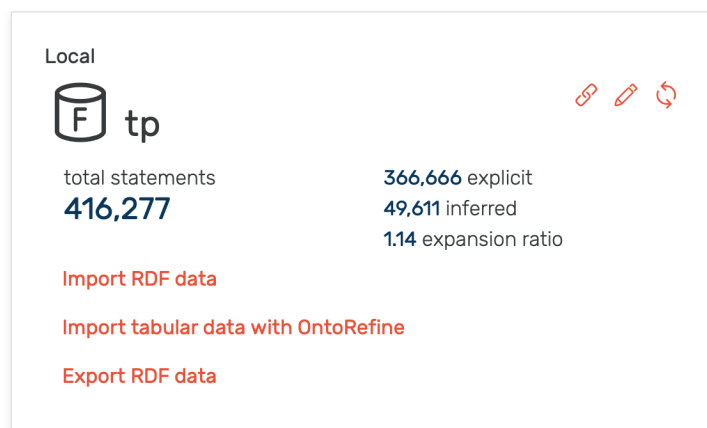


Figura 1: Informação sobre o repositório.

⁴<http://graphdb.ontotext.com>

É possível consultar também a hierarquia das classes presentes, tendo em conta o número de instâncias.

Class hierarchy ?

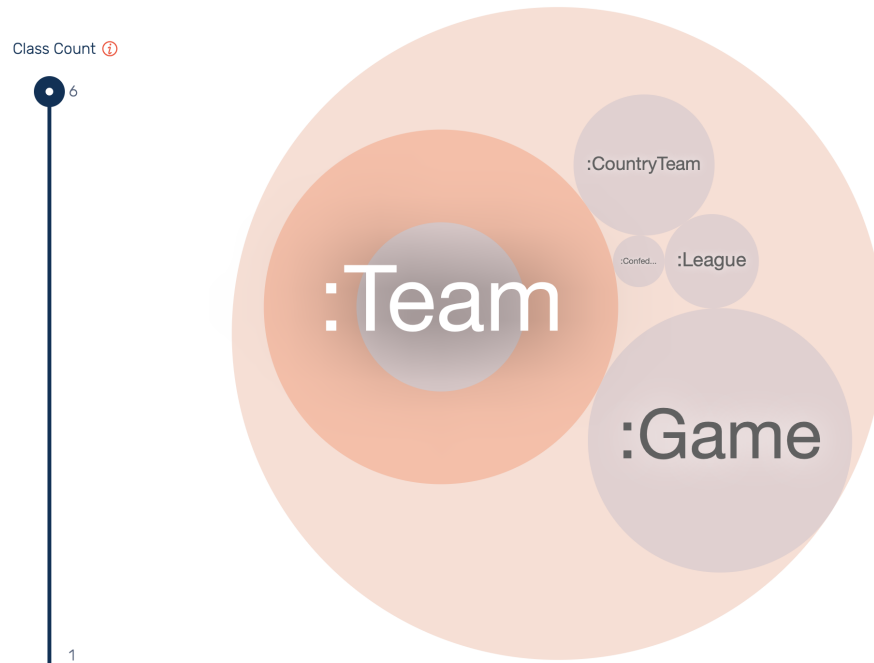


Figura 2: Hierarquia das classes.

É também possível ver no **GraphDB** o grafo gerado. Na imagem seguinte é possível ver o grafo gerado para o exemplo da *Liga Portuguesa*.

Visual graph ?

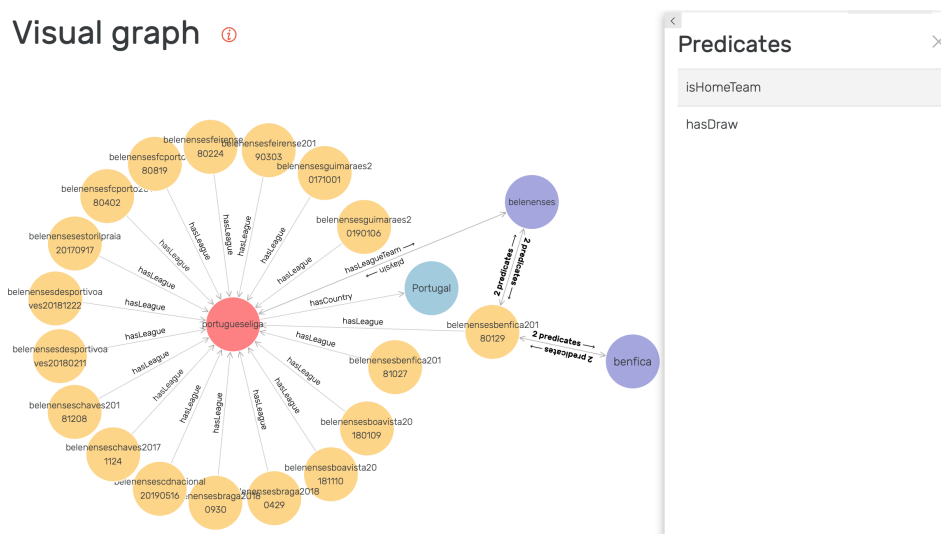


Figura 3: Grafo gerado para a *Liga Portuguesa*.

3.1.3 DBPedia

Para além da informação armazenada no repositório do **GraphDB**, vai ser também utilizada a informação sobre futebol da **DBPedia**⁵. Esta vai ser relacionada com a informação presente no repositório e serão efetuadas *queries* à **DBPedia** para obter os resultados.

As classes da **DBPedia** que vão ser utilizadas são:

- **<http://dbpedia.org/ontology/Country>**: representa o país a que pertence uma liga.
- **<http://dbpedia.org/ontology/SoccerClub>**: representa uma equipa de futebol.
- **<http://dbpedia.org/ontology/SoccerLeague>**: representa uma liga de futebol.
- **<http://dbpedia.org/ontology/SoccerManager>**: representa o treinador de uma equipa de futebol.

Tendo em conta estas classes, vão ser utilizadas as seguintes *data properties*:

Nome	Domain	Range
rdfs:comment	dbo:SoccerClub	xsd:string
dbo:abstract	dbo:SoccerClub	xsd:string
	dbo:SoccerLeague	
	dbo:SoccerManager	
rdfs:label	dbo:SoccerManager	xsd:string
	dbo:Country	
foaf:homepage	dbo:SoccerClub	xsd:string
	dbo:SoccerLeague	

Tabela 3: *Data Properties* utilizadas da **DBPedia**.

Relativamente a *object properties*, vai ser apenas considerada uma:

Nome	Domain	Range
dbo:manager	dbo:SoccerClub	dbo:SoccerManager

Tabela 4: *Object Properties* utilizadas da **DBPedia**.

⁵<https://wiki.dbpedia.org>

3.2 Especificação de Requisitos

Tendo em conta a ontologia desenvolvida e os dados disponíveis, foram levantados os seguintes requisitos:

- Consulta das 10 melhores equipas de futebol
- Consulta dos primeiros 5 classificados das 4 principais ligas
- Listagem de todas as equipas
- Listagem de todas as ligas
- Listagem de todas as seleções
- Consulta da informação de uma equipa
- Consulta dos jogos de uma equipa
- Consulta da informação de uma liga de futebol
- Consulta da classificação das equipas de uma liga
- Consulta dos jogos de uma liga
- Consulta das estatísticas de uma equipa

4 Implementação

Tendo em conta os requisitos levantados, foram utilizados dois servidores para o desenvolvimento da aplicação *web* final. O primeiro, que implementa a **API** que faz os pedidos necessários e o tratamento desses pedidos, retornando-os em **JSON**. E o segundo, que faz pedidos à **API** e apresenta os dados. O primeiro foi desenvolvido em *Express*⁶ e o segundo em *Vue*.⁷

4.1 Express

Implementação da **API** que faz *queries* ao *GraphDB* e à *DBpedia*, retornando o resultado em **JSON**.

```
const prefix = 'prefix :  
<http://www.semanticweb.org/joana/ontologies/2019/soccer#>\n'  
const lhost = 'http://localhost:7200/repositories/tp'  
  
async function execQuery(query) {  
  try {  
    var encoded = encodeURIComponent(query)  
    response = await axios.get(lhost + "?query=" + encoded)  
    return (response.data)  
  }  
  catch (erro) {  
    return ('ERRO na \'execQuery\' de \'' + query + '\': ' + erro)  
  }  
}  
  
Soccer.teamVictories = async (id) => {  
  const query = prefix + `select (count(?gh) as ?victoriesHome) (count(?ga) as  
    ?victoriesAway) where {  
    {:` + id + ` :hasVictory ?gh .  
    ?gh :hasHomeTeam :` + id + ` .}  
    union  
    {:` + id + ` :hasVictory ?ga .  
    ?ga :hasVisitorTeam :` + id + ` .}  
  }`  
  var res = await execQuery(query)  
  return res  
}
```

Listing 2: Exemplo de uma *querie* ao *GraphDB* (obter as vitórias de uma equipa).

⁶<https://expressjs.com>

⁷<https://vuejs.org>

Para fazer as *queries* ao *GraphDB* é necessário definir a *query SPARQL* e depois fazer um pedido *GET* ao servidor do *GraphDB* (neste exemplo encontra-se a correr na própria máquina na porta 7200). Recorrendo a *query string*, é possível também enviar a *query SPARQL* ao *GraphDB*.

```
var convert = require('xml-js')
var dps = require('dbpedia-sparql-client').default

function xml2json(xml) {
  var json = convert.xml2json(xml, { compact: true, spaces: 4 })
  return json
}

Soccer.getHomepage = async (url, _callback) => {
  var q = `select ?page where {
    <` + url + `> foaf:homepage ?page .
  }`
  dps.client()
    .query(q)
    .timeout(10000)
    .asXml()
    .then(function (r) {
      var res = JSON.parse(xml2json(r)).sparql.results.result
      _callback(res)
    })
    .catch(function (e) {
      console.log('ERRO em \'getHomepage\' de \'' + JSON.stringify(url) +
        '\': ' + e)
    })
}
```

Listing 3: Exemplo de uma *query* ao *GraphDB* (obter a *homepage* de uma liga ou equipa).

Para efetuar *queries* à *DBPedia* é necessário definir primeiro a *query SPARQL* pretendida. Depois é necessário fazer essa *query* à *DBPedia*. Para isso, foi utilizado o *dbpedia-sparql-client*⁸. Como este devolveu o resultado em **XML**, foi necessário converter este resultado para **JSON**.

⁸<https://www.npmjs.com/package/dbpedia-sparql-client>

Tendo em conta que o servidor corre na porta 8090, a **API** definida permite fazer os seguintes pedidos:

- **GET /top10** - devolve as 10 melhores equipas de futebol
- **GET /englandtop5** - devolve as 5 melhores equipas da primeira liga inglesa
- **GET /spaintop5** - devolve as 5 melhores equipas da primeira liga espanhola
- **GET /portugaltop5** - devolve as 5 melhores equipas da primeira liga portuguesa
- **GET /italytop5** - devolve as 5 melhores equipas da primeira liga italiana
- **GET /leagues** - devolve todas as ligas de futebol
- **GET /leagueTeams/:id** - devolve todas as equipas de uma determinada liga
- **GET /leagueGames/:id** - devolve todos os jogos de uma determinada liga
- **GET /teams** - devolve todas as equipas de futebol
- **GET /gamesTeam/:id** - devolve todos os jogos de uma equipa de futebol
- **GET /internationalTeams** - devolve todas as seleções de futebol
- **GET /teamComment/:id** - devolve uma breve descrição sobre uma equipa de futebol
- **GET /name/:id** - devolve o nome de uma equipa ou de uma liga de futebol
- **GET /teamLogo/:id** - devolve o logótipo de uma equipa de futebol
- **GET /abstract/:id** - devolve a descrição de uma equipa ou de uma liga de futebol
- **GET /teamManagerName/:id** - devolve o nome do treinador de uma equipa
- **GET /teamManagerAbstract/:id** - devolve a descrição do treinador de uma equipa
- **GET /homepage/:id** - devolve a *homepage* de uma equipa ou de uma liga de futebol
- **GET /teamNumberGames/:id** - devolve o número de jogos em casa e fora de uma equipa
- **GET /teamScoredGoals/:id** - devolve o número de golos marcados em casa e fora de uma equipa
- **GET /teamSufferedGoals/:id** - devolve o número de golos sofridos em casa e fora de uma equipa
- **GET /teamVictories/:id** - devolve o número de vitórias em casa e fora de uma equipa
- **GET /teamDefeats/:id** - devolve o número de derrotas em casa e fora de uma equipa
- **GET /teamTies/:id** - devolve o número de empates em casa e fora de uma equipa

4.2 Vue

Utilizando a **API** desenvolvida anteriormente e, recorrendo à *framework* *Vue.js* e à *framework* *Vuetify*⁹, foi desenvolvida a interface da aplicação.

A aplicação foi estruturada da seguinte forma:


- Página inicial da aplicação que apresenta as 10 melhores equipas de futebol e as 5 melhores equipas das ligas portuguesa, inglesa, espanhola e italiana.

path: /

view: Index.vue

components: SideBar.vue, Top10.vue, TopLeagues.vue

⁹<https://vuetifyjs.com/en/getting-started/why-vuetify>



TEAMS

LEAGUES

INTERNATIONAL TEAMS

TOP 10 TEAMS

Rank	Team	League	Country	SPI	Offense	Defense
1	Manchester City	Barclays Premier League	United Kingdom	94.39	3.04	0.2
2	Bayern Munich	German Bundesliga	Germany	93.5	3.26	0.36
3	Liverpool	Barclays Premier League	United Kingdom	92.93	2.92	0.26
4	Barcelona	Spanish Primera Division	Spain	91.74	3.01	0.38
5	Chelsea	Barclays Premier League	United Kingdom	85.44	2.4	0.45
6	Paris Saint-Germain	French Ligue 1	France	84.98	2.81	0.7
7	Real Madrid	Spanish Primera Division	Spain	84.54	2.57	0.59
8	Atletico Madrid	Spanish Primera Division	Spain	84.19	2.14	0.37
9	Olympiacos	Greek Super League	Greece	83.35	2.62	0.68
10	Ajax	Dutch Eredivisie	Netherlands	82.85	3.18	1.06

TOP LEAGUES

Portuguese Liga

Points	Team	Scored	Suffered
87	Benfica	103	31
85	FC Porto	74	20
74	Sporting CP	72	33
67	Braga	56	37
52	Guimaraes	46	34

Barclays Premier League

Points	Team	Scored	Suffered
98	Manchester City	95	23
97	Liverpool	89	22
72	Chelsea	63	39
71	Tottenham Hotspur	67	39
70	Arsenal	73	51

Spanish Primera Division

Points	Team	Scored	Suffered
87	Barcelona	90	36
76	Atletico Madrid	55	29
68	Real Madrid	63	46
61	Valencia	51	35
59	Getafe	48	35

Italy Serie A

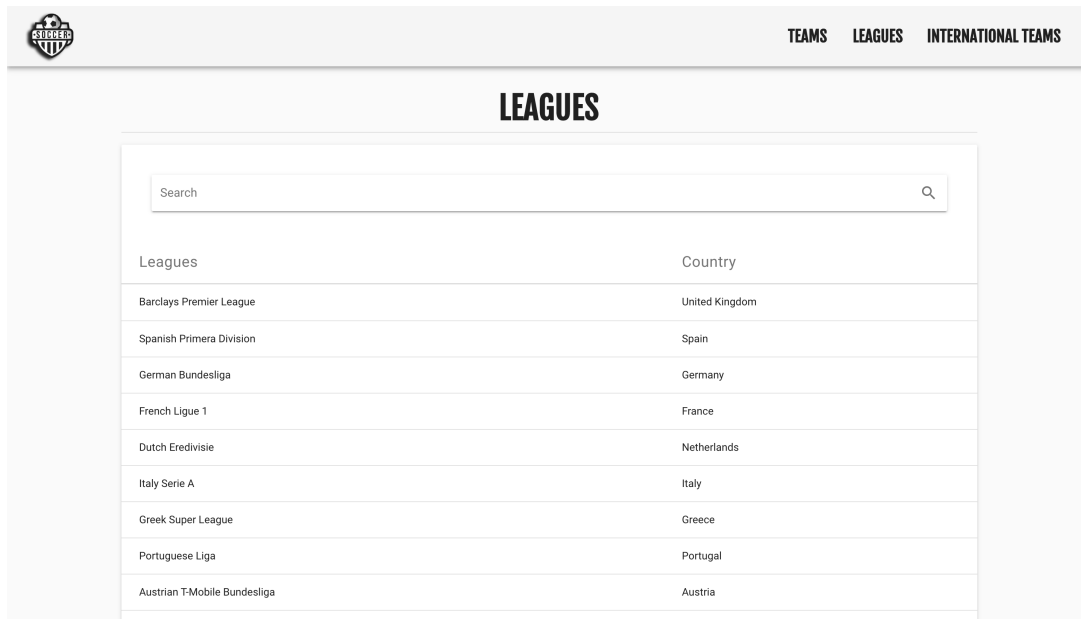
Points	Team	Scored	Suffered
90	Juventus	70	30
79	Napoli	74	36
69	Atalanta	77	46
69	Internazionale	57	33
68	AC Milan	55	36

- Página de apresentação de todas as ligas de futebol.

path: /leagues

view: Leagues.vue

components: SideBar.vue, Leagues.vue

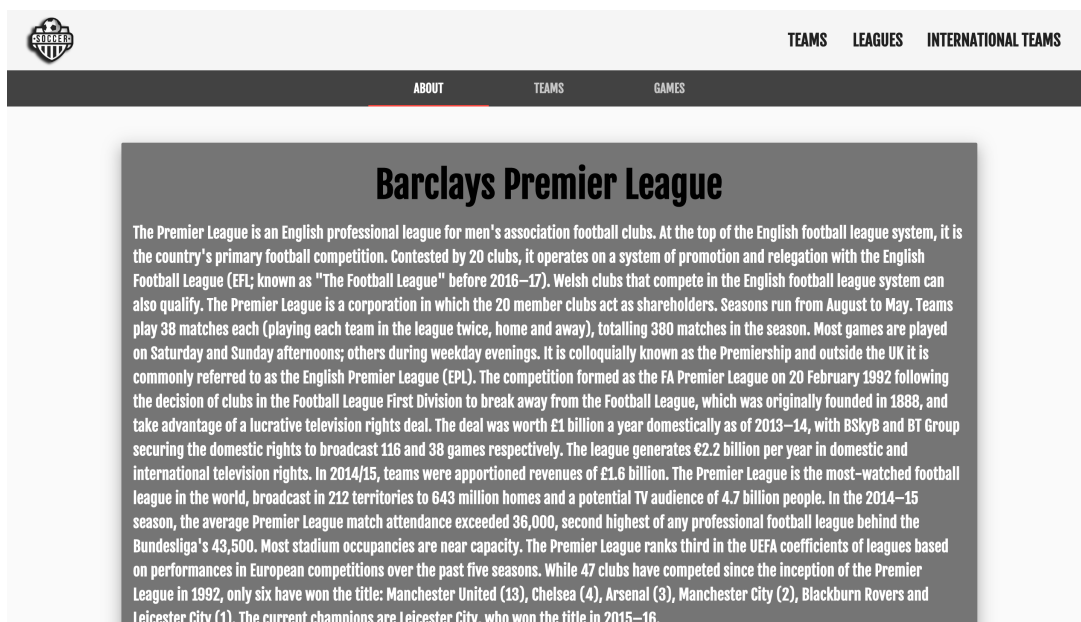



- Página de apresentação de uma liga de futebol, com informação sobre ela, classificação das suas equipas e jogos.


path: /leagues/:id

view: League.vue

components: SideBar.vue, About.vue, LeagueTeams.vue, Games.vue



<div>  <div>TEAMSLEAGUESINTERNATIONAL TEAMS</div> </div>									
<div> <div>ABOUT</div> <div>TEAMS</div> <div>GAMES</div> </div>									
Rank	Team	P	G	V	T	D	SG	CG	GD
1	Manchester City	98	38	32	2	4	95	23	72
2	Liverpool	97	38	30	7	1	89	22	67
3	Chelsea	72	38	21	9	8	63	39	24
4	Tottenham Hotspur	71	38	23	2	13	67	39	28
5	Arsenal	70	38	21	7	10	73	51	22
6	Manchester United	66	38	19	9	10	65	54	11
7	Wolverhampton	57	38	16	9	13	47	46	1
8	Everton	54	38	15	9	14	54	46	8
9	Leicester City	52	38	15	7	16	51	48	3
10	West Ham United	52	38	15	7	16	52	55	-3
11	Watford	50	38	14	8	16	52	59	-7
12	Crystal Palace	49	38	14	7	17	51	53	-2
13	Newcastle	45	38	12	9	17	42	48	-6


<div>  <div>TEAMSLEAGUESINTERNATIONAL TEAMS</div> </div>									
<div> <div>ABOUT</div> <div>TEAMS</div> <div>GAMES</div> </div>									
0.0 Brighton and Hove Albion 1 - 4 Manchester City 100.0		2019-05-12							
		2019-05-12		0.0 Burnley 1 - 3 Arsenal 0.5					
0.0 Crystal Palace 5 - 3 AFC Bournemouth 0.0		2019-05-12							
		2019-05-12		0.0 Fulham 0 - 4 Newcastle 0.0					
0.0 Leicester City 0 - 0 Chelsea 0.0		2019-05-12							
		2019-05-12		58.9 Liverpool 2 - 0 Wolverhampton 0.0					
0.0 Manchester United 0 - 2 Cardiff City 0.0		2019-05-12							

- Página de apresentação de todas as equipas de futebol.

path: /teams

view: Teams.vue

components: SideBar.vue, Teams.vue


<div>  <div>TEAMSLEAGUESINTERNATIONAL TEAMS</div> </div>		
TEAMS		
<div> <div>Search</div> <div></div> </div>		
Team	League	PSI
Manchester City	Barclays Premier League	94.39
Bayern Munich	German Bundesliga	93.5
Liverpool	Barclays Premier League	92.93
Barcelona	Spanish Primera Division	91.74
Chelsea	Barclays Premier League	85.44
Paris Saint-Germain	French Ligue 1	84.98
Real Madrid	Spanish Primera Division	84.54
Atletico Madrid	Spanish Primera Division	84.19
Olympiacos	Greek Super League	83.35

- Página de apresentação de uma equipa de futebol, com informação sobre ela, jogos e estatísticas.

path: /teams/:id


view: Team.vue

components: SideBar.vue, AboutTeam.vue, GamesTeam.vue, StatsTeam.vue



TEAMSLEAGUESINTERNATIONAL TEAMS

ABOUTGAMESSTATS



Liverpool

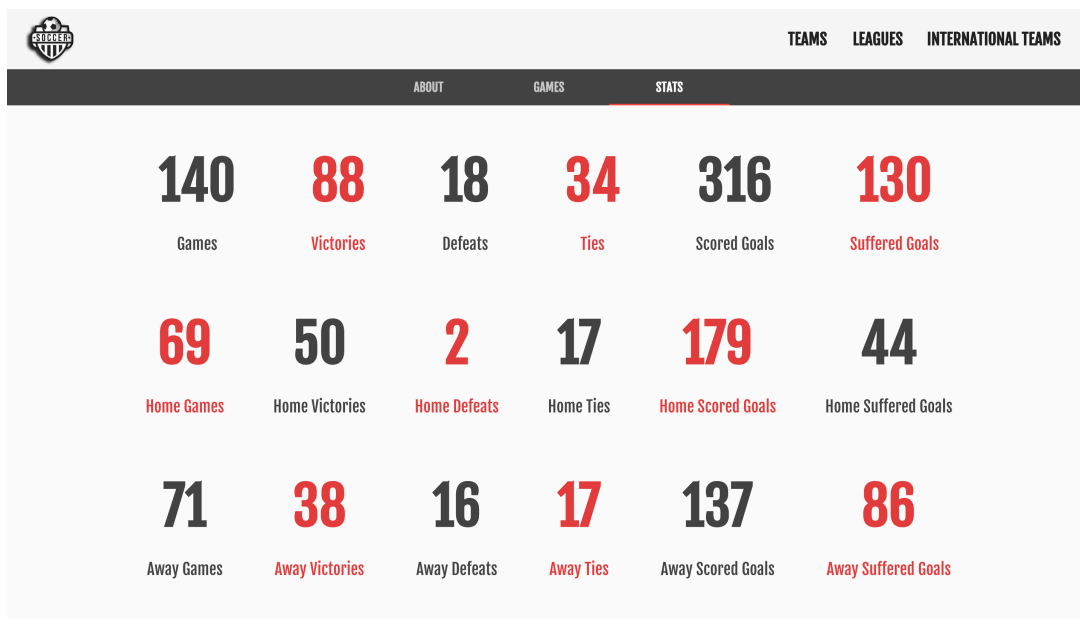
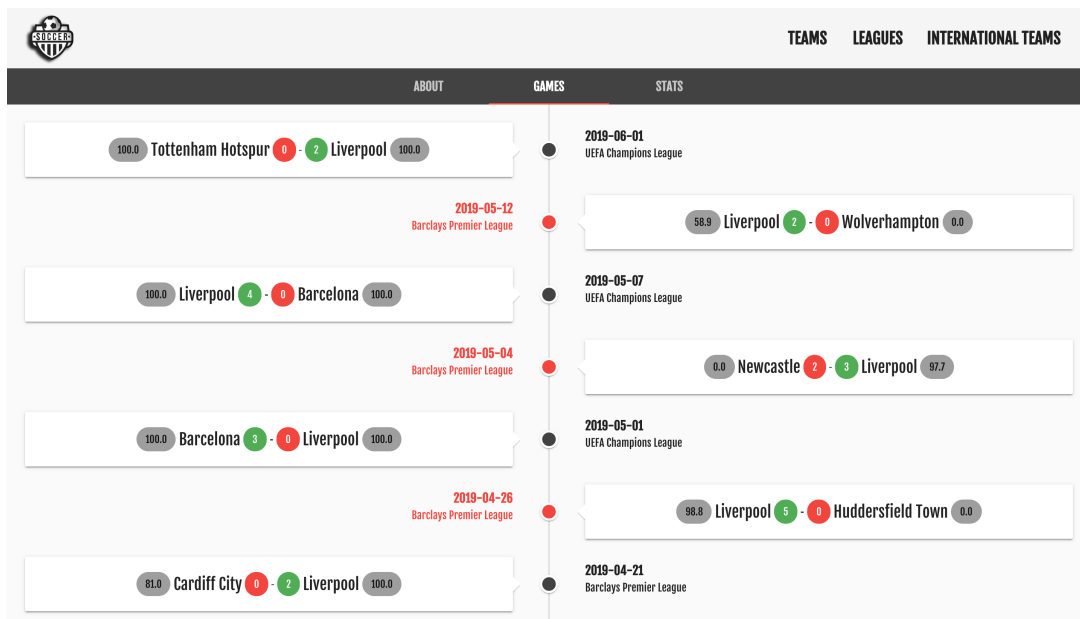
Liverpool Football Club (ⁱ/ˈlɪvərpuːl/) is a Premier League association football club based in Liverpool, Merseyside, England. The club has won five European Cups, three UEFA Cups, three UEFA Super Cups, 18 League titles, seven FA Cups, a record eight League Cups, and 15 FA Community Shields.

Abstract

Liverpool Football Club (ⁱ/ˈlɪvərpuːl/) is a Premier League association football club based in Liverpool, Merseyside, England. The club has won five European Cups, three UEFA Cups, three UEFA Super Cups, 18 League titles, seven FA Cups, a record eight League Cups, and 15 FA Community Shields. The club was founded in 1892 and joined the Football League the following year. The club has played at Anfield since its formation. Liverpool established itself as a major force in both English and European football during the 1970s and 1980s when Bill Shankly and Bob Paisley led the club to 11 League titles and seven European trophies. Under the management of Rafa Benítez and captained by Steven Gerrard Liverpool became European champion for the fifth time, winning the 2005 UEFA Champions League Final against Milan in spite of being 3–0 down at half time. Liverpool was the ninth highest-earning football club in the world for 2014–15, with an annual revenue of €391 million, and the world's eighth most valuable football club in 2016, valued at \$1.55 billion. The club holds many long-standing rivalries, most notably the North West Derby against Manchester United and the Merseyside derby with Everton. The club's supporters have been involved in two major tragedies. The first was the Heysel Stadium disaster in 1985, where escaping fans were pressed against a collapsing wall in the Heysel Stadium, with 39 people—mostly Italians and Juventus fans—dying, after which English clubs were given a five-year ban from European competition. The second was the Hillsborough disaster in 1989, where 96 Liverpool supporters died in a crush against perimeter fencing. The team changed from red shirts and white shorts to an all-red home strip in 1964 which has been used ever since. The club's anthem is "You'll Never Walk Alone".

Manager: Jürgen Klopp

Jürgen Norbert Klopp (German pronunciation: [ˈjʊʁɡən ˈklɔp] ; born 16 June 1967) is a German football manager and former professional player who is the current manager of Premier League club Liverpool. Klopp spent most of his 15-year playing career at Mainz 05, before going on to become their longest-serving manager from 2001 to 2008, during which time they achieved promotion to the Bundesliga. In 2008, Klopp joined Borussia Dortmund, leading them to back-to-back Bundesliga wins in 2011 and 2012, as well as the DFB-Pokal in 2012, the DFL-Supercup in 2013 and 2014, and their second appearance in a UEFA Champions League final in 2013. Klopp won the German Football Manager of the Year in 2011 and 2012, before leaving Dortmund in 2015 having also become their longest-serving manager. He became manager of Liverpool in October 2015.




- Página de apresentação de todas as seleções de futebol.

path: /internationalTeams

view: InternationalTeams.vue

components: SideBar.vue, InternationalTeams.vue



TEAMS

LEAGUES

INTERNATIONAL TEAMS

INTERNATIONAL TEAMS

Search

Q

Rank	Team	Confederation	PSI	Offense	Defense
1	Brazil	CONMEBOL	92.14	2.98	0.3
2	Spain	UEFA	92.05	3.48	0.53
3	Germany	UEFA	87.92	2.99	0.58
4	Belgium	UEFA	87.82	2.98	0.58
5	France	UEFA	87.0	2.74	0.5
6	England	UEFA	85.79	2.58	0.48
7	Argentina	CONMEBOL	84.68	2.47	0.48
8	Netherlands	UEFA	84.05	2.67	0.63
9	Switzerland	UEFA	82.08	2.63	0.71

5 Conclusões

Ao longo deste relatório está exposto todo o trabalho realizado para este projeto, assim como algumas das decisões tomadas na sua implementação.

Tendo em conta a proposta de trabalho apresentada, considero que o *frontend* e o *backend* da aplicação estão bastante positivos, pois cumprem com todos os requisitos levantados inicialmente.

Como trabalho futuro poderia ser automatizada a ligação entre os dados presentes no *GraphDB* e na *DBPedia*. Poderia também ser automatizada a forma de obtenção do logótipo de cada equipa, visto que essa informação não estava presente nos *datasets* utilizados, e é uma forma de melhorar a apresentação da aplicação.