

# **APLICAÇÕES NA WEB**

Relatório do Projeto

Etapa 3



Mestrado de Informática

sexta-feira, 3 de junho de 2016

Grupo 001 – TP21:

João Vicente (Aluno n.º 44294)

Ana Salvado (Aluna n.º 44299)

Carlos Duarte (Aluno n.º 44313)

Inês de Matos (Aluna n.º 43538)

Eunice Branco (Aluna n.º 48478)

# ÍNDICE

<b>1. PLANEAMENTO</b>	<b>1</b>
<i>Fase 0: Modelagem</i>	3
<i>Fase 1: Recolha de dados e serviços web</i>	3
<i>Fase 2: Análise de dados</i>	3
<i>Fase 3: Visualização de informação</i>	3
<b>2. ARQUITECTURA</b>	<b>4</b>
<i>Definição de arquitectura</i>	4
<i>Requisitos estruturais e funcionais do sistema</i>	6
<i>Protocolos de comunicação entre componentes do sistema</i>	6
<b>3. BACK-END</b>	<b>7</b>
<i>Implementação de Back-End</i>	7
3.1. TECNOLOGIAS USADAS	7
<i>Recolha de dados de fontes externas</i>	7
<i>Tratamento e filtragem de dados</i>	7
3.2. WEB SERVICES USADOS	7
3.3. ESTRUTURA DE DADOS	7
<i>Armazenamento de dados numa base de dados SQL</i>	7
<i>Linked data</i>	8
3.4. COMPONENTES IMPLEMENTADOS	8
<i>Recolha de informação</i>	8
<b>4. WEB SERVICES</b>	<b>10</b>
4.1. TECNOLOGIAS USADAS	10
4.2. LISTA DE MÉTODOS	10
<i>Tabela de Web Services</i>	10
<i>Implementação de Web Services</i>	14
4.3. EXEMPLOS	15
<b>5. FRONT-END (ADMIN E USER)</b>	<b>17</b>
5.1. TECNOLOGIAS USADAS	17
<i>Implementação do Front-End</i>	17
<i>Front-End dinâmico com AJAX</i>	17
5.2. WEB SERVICES USADOS	17
<i>APIs de Conta:</i>	17
5.3. SCREENSHOTS	18

<i>Sketches de Interface</i>	18
<i>Visualização interactiva com D3.js</i>	20
<i>Mecanismos de feedback de utilizador</i>	20
<b>6. DISCUSSÃO</b>	<b>21</b>
6.1. <i>PROBLEMAS ENCONTRADOS</i>	21

## 1. PLANEAMENTO

Tarefa	Duração	Início	Conclusão	Elementos do grupo
<b>CP 0</b>		<b>Dom</b> <b>06/03/16</b>	<b>Dom</b> <b>06/03/16</b>	
Escolha de tema	1 dia	Sex 26/02/16	Sex 26/02/16	Ana, Eunice, Inês, João, Carlos
Escolha de tecnologias/webservices	2 dias	Sex 26/02/16	Seg 29/02/16	Inês, João
Instalação das tecnologias	5 dias	Ter 01/03/16	Dom 06/03/16	Ana, Carlos, Eunice, Inês, João
Design de Interfaces	3 dias	Qua 02/03/16	Sex 04/03/16	Ana, Eunice, Inês, João
Layouts	5 dias	Ter 01/03/16	Dom 06/03/16	Inês
<b>Redacção do Relatório</b>	<b>4 dias</b>	<b>Qua</b> <b>30/03/16</b>	<b>Sáb</b> <b>02/04/16</b>	<b>Ana, Eunice, Inês</b>
<b>CP 1</b>		<b>Dom</b> <b>03/04/16</b>	<b>Dom</b> <b>03/04/16</b>	
Construção BD ( <i>dummy</i> )	1 dia	Seg 07/03/16	Seg 07/03/16	Ana, Inês
Implementação do <i>Back-End</i> (PHP)	11 dias	Seg 07/03/16	Dom 20/03/16	João, Carlos
<b>Web service: Redes Sociais</b>	<b>11 dias</b>	<b>Seg</b> <b>07/03/16</b>	<b>Dom</b> <b>20/03/16</b>	<b>Ana, Eunice, Inês</b>
Recolha de informação	6 dias	Seg 07/03/16	Dom 13/03/16	Ana, Eunice, Inês
Tratamento e Filtragem	7 dias	Dom 13/03/16	Dom 20/03/16	Ana, Eunice, Inês
<b>Web service: Campeonato</b>	<b>11 dias</b>	<b>Seg</b> <b>07/03/16</b>	<b>Dom</b> <b>20/03/16</b>	<b>Carlos, João</b>
Implementação de Web service	11 dias	Seg 21/03/16	Dom 03/04/16	Inês, João, Carlos
<b>Actualização do Relatório</b>	<b>4 dias</b>	<b>Qua</b> <b>30/03/16</b>	<b>Sáb</b> <b>02/04/16</b>	<b>Ana, Eunice, Inês</b>
<b>CP 2</b>		<b>Dom</b> <b>01/05/16</b>	<b>Dom</b> <b>01/05/16</b>	
Análise e inferência	16 dias	Seg 04/04/16	Dom 24/04/16	Ana, Eunice, Carlos, Inês, João
<i>Triple-store</i>	16 dias	Seg 04/04/16	Dom 24/04/16	Inês, João, Ana, Eunice, Carlos
<i>Tagging</i> e inferência de dados	16 dias	Seg 04/04/16	Dom 24/04/16	Ana, Inês, Eunice, Carlos, João
Recurso a <i>Linked Data</i>	16 dias	Seg 04/04/16	Dom 24/04/16	Carlos, João, Ana, Eunice, Inês
Documentação do Web service online	11 dias	Seg 18/04/16	Dom 01/05/16	Ana, Eunice, Inês, Carlos, João
<i>Front-End</i> com AJAX	11 dias	Seg 18/04/16	Dom 01/05/16	Ana, Carlos, Eunice, Inês, João
<b>Actualização do Relatório</b>	<b>4 dias</b>	<b>Qua</b> <b>27/04/16</b>	<b>Sáb</b> <b>30/04/16</b>	<b>Ana, Eunice, Inês</b>

CP 3		Sex 27/05/16	Sex 27/05/16	
Revisão do <i>Layout</i>	17 dias	Seg 02/05/16	Ter 24/05/16	Inês, Ana, João
D3.js	17 dias	Seg 02/05/16	Ter 24/05/16	Eunice, Carlos
Mecanismos de feedback	17 dias	Seg 02/05/16	Ter 24/05/16	Inês, João
Classificação/Recuperação de informação	17 dias	Seg 02/05/16	Ter 24/05/16	Ana, Eunice
Vídeo	2 dias	Qua 25/05/16	Qui 26/05/16	Ana, Eunice, Inês
<b>Relatório Final</b>	<b>3 dias</b>	<b>Qua 25/05/16</b>	<b>Sex 27/05/16</b>	<b>Ana, Carlos, Eunice, Inês, João</b>

Há que notar que o planeamento apresentado está sujeito a futuras mudanças, sendo a tabela apresentada meramente indicativa do que o grupo entende ser o tempo necessário a despender na distribuição de tarefas entre os seus membros.

O presente relatório pretende ainda enquadrar o escopo do projecto a desenvolver, no âmbito da disciplina de Aplicações na *Web*, extensivamente à documentação do planeamento de tarefas e a sua distribuição pelos membros do grupo.

Cumpra, antes de mais, identificar ‘UEFA Euro 2016’ como o tema escolhido.

O projecto prender-se-á com o desenvolvimento de uma aplicação em que se pretende determinar se existem relações pertinentes entre jogadores de equipas, bem como outros protagonistas do Euro 2016. Com isto entenda-se treinadores e presidentes, entre outros intervenientes.

Estas relações serão criadas através da identificação das personalidades, em artigos e *posts* da *web*, com base no histórico futebolístico, bem como em aspectos pessoais, como sejam os familiares dos jogadores.

Pretendemos, também, que a aplicação apresente uma percentagem indicativa do nível da relação entre personalidades, por forma a evidenciar se as mesmas estão muito ou pouco relacionadas. Desta forma, pretendemos enfatizar as classificações das relações mais mediáticas.

Os utilizadores da nossa aplicação irão também ter um papel importante nestas relações, através de comentários e da partilha de informação sobre personalidades do mundo futebolístico. Iremos ainda apresentar um *top* pessoal. Isto é, conforme o número de pesquisas e comentários que o utilizador faça a propósito de determinadas relações, temos que estas surgirão em primeiro lugar na página inicial.

### *Fase 0: Modelagem*

Numa fase inicial iremos identificar os requisitos estruturais e funcionais do nosso sistema. Concentrar-nos-emos ainda na definição de arquitectura, bem como na comunicação entre componentes do sistema.

Concomitantemente, iremos também criar uma tabela de web services e apresentar sketches do aspecto da interface e elementos com os quais o utilizador poderá vir a interagir.

### *Fase 1: Recolha de dados e serviços web*

Nesta fase o grupo vira a sua atenção para os aspectos de implementação do Back-End. Uma vez explorados os elementos relevantes neste contexto, passamos a elencar as fontes externas de onde recolhemos os dados para armazenar informação numa base de dados SQL.

Como não podia deixar de ser, os processos supra mencionados importam o tratamento e filtragem dos dados recolhidos.

Terminamos esta etapa com uma descrição da implementação *de web services*.

### *Fase 2: Análise de dados*

A análise de dados e as inferências dela extraídas são parte integrante desta fase. Ademais recorreremos a *linked data*, bem como a AJAX, sendo que neste último caso o propósito final se prende com a criação de um *Front-End* dinâmico.

Por fim temos que, nesta fase, se pode consultar a documentação de *web services* online a que recorreremos.

### *Fase 3: Visualização de informação*

Esta última fase prende-se com a implementação do Front-End através da criação de uma experiência visual interactiva com D3.js.

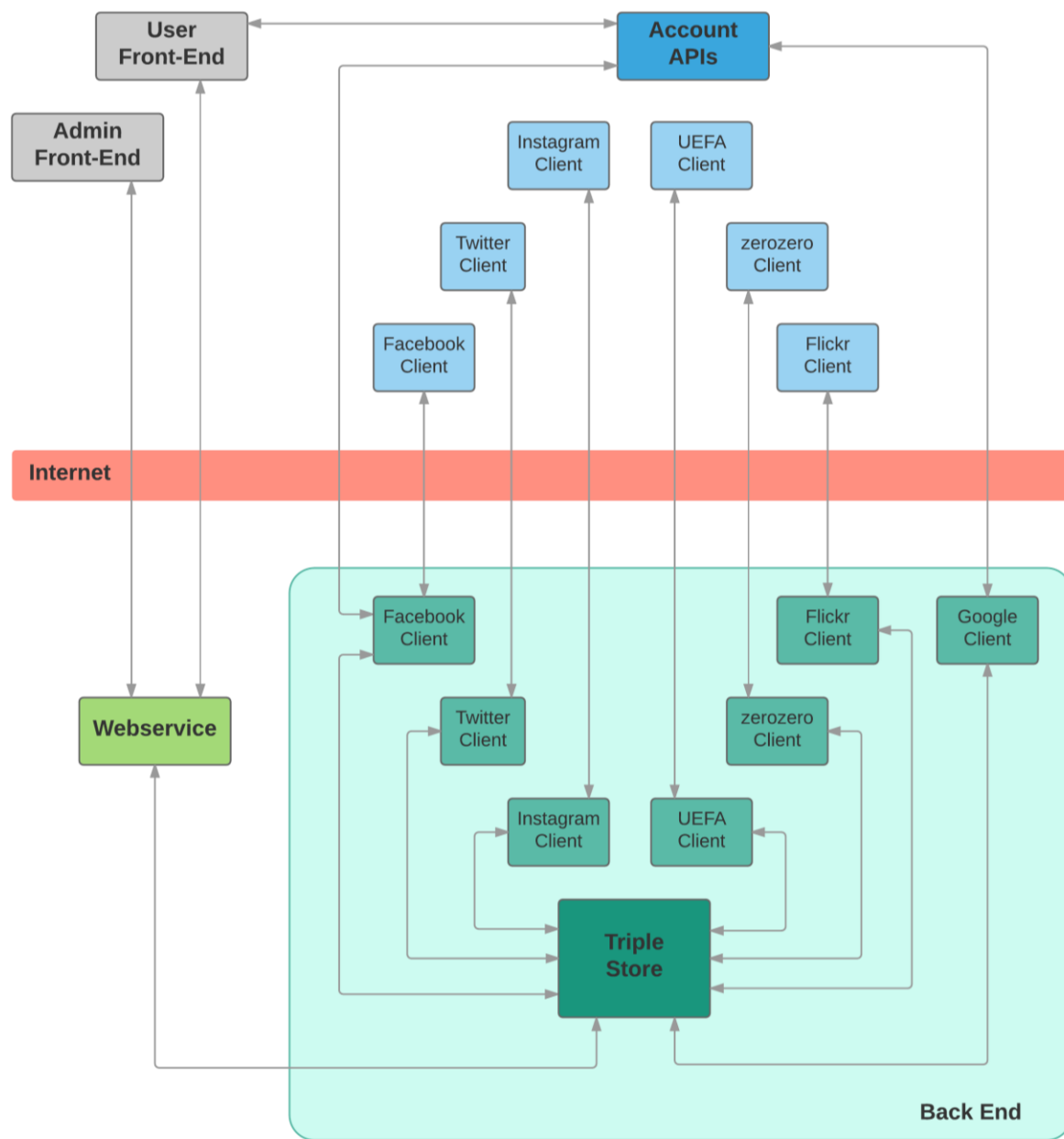
É também nesta fase que implementamos os mecanismos de *feedback* do utilizador, explorados no corpo do relatório.

Por fim cumpre mencionar a criação de um trailer que demonstra o funcionamento da aplicação no seu estado final.

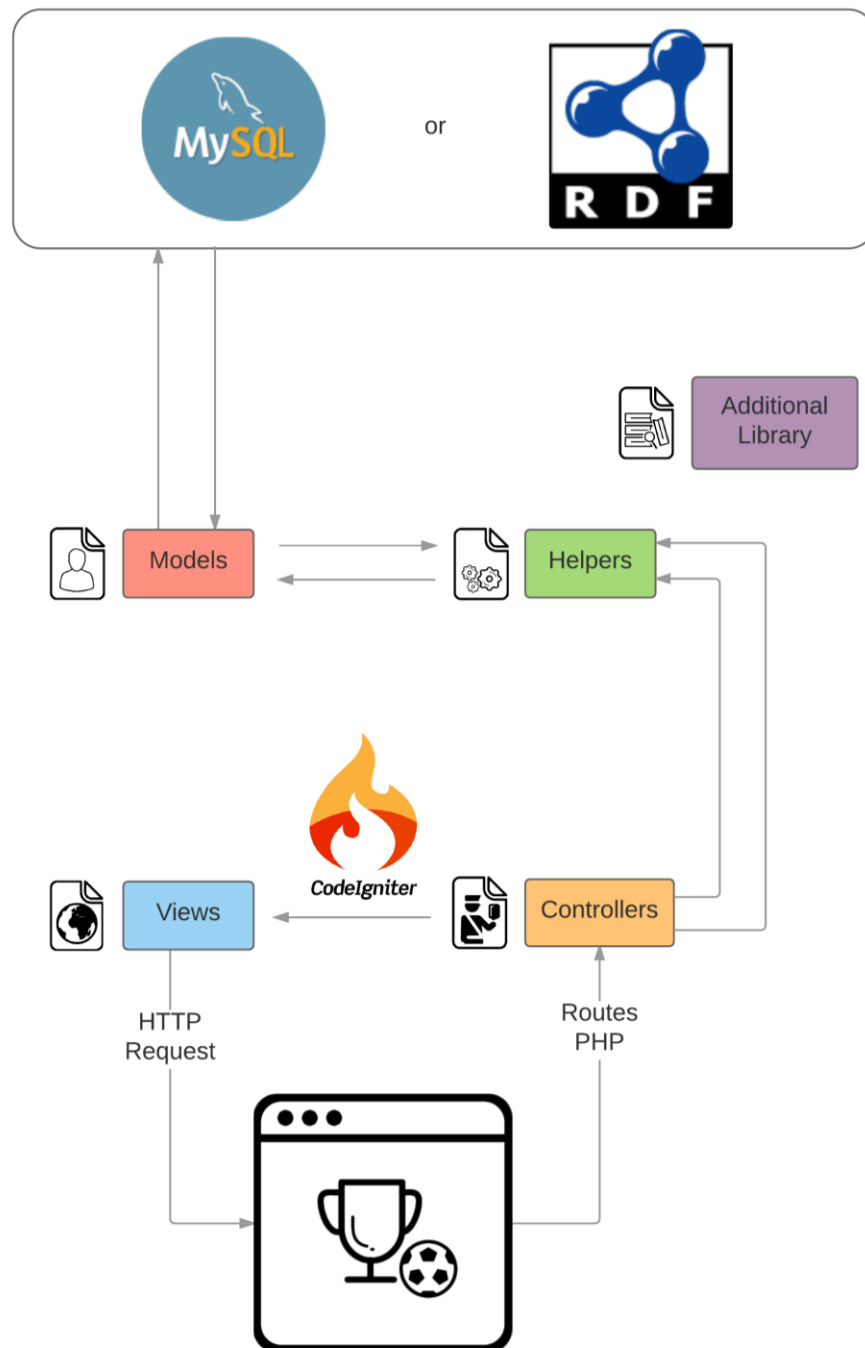
## 2. ARQUITECTURA

### Definição de arquitectura

A propósito da arquitectura apresentamos o modelo na imagem abaixo. No nosso modelo podemos observar as relações entre os vários componentes do sistema (*back-end*, *front-end* e *webservice*).



Adicionalmente reproduzimos o modelo de relações relativamente à arquitectura de uma outra tecnologia usada. Como é possível observar na imagem abaixo, o CodeIgniter tem um papel importante no nosso modelo.





### *Requisitos estruturais e funcionais do sistema*

Todo o projeto é baseado no Codeigniter, como já foi referenciado acima, no qual apresenta um modelo MVC. Sendo assim, a nossa aplicação ira-se dividir em três componentes, a apresentação,

### *Protocolos de comunicação entre componentes do sistema*

Para a comunicação dos Web Services é usado o HTTP.

### 3. BACK-END

#### *Implementação de Back-End*

A propósito das tecnologias a empregar no âmbito do desenvolvimento do Back-End, temos que nos iremos servir da *framework* CodeIgniter (PHP) por forma a estruturar a nossa aplicação.

Relativamente à base de dados, cumpre ainda mencionar que pretendemos utilizar MySQL numa fase inicial, sendo que posteriormente nos tencionamos fazer uso de uma base de dados firmada em *triple-store*. Planeamos ainda fazer uso de uma ferramenta para a realização de *queries*, como seja SPARQL.

#### 3.1. TECNOLOGIAS USADAS

##### *Recolha de dados de fontes externas*

APIs de redes sociais:

- Facebook;
- Twitter;
- Flickr;
- Instagram.

(Webscrapping) Desporto:

- UEFA;
- zerozero.

Vídeo:

- YouTube.

##### *Tratamento e filtragem de dados*

A nível do webscrapping usámos o XPATH de forma a percorrer a árvore do DOM, afim de obter a informação desejada. Com isto recolhemos maioritariamente informação do tipo “string”, a qual foi devidamente formatada de forma a ser mais acessível quando fosse acedida pela BD.

A nível das APIs, a do Twitter fornece um webservice que possibilita especificar os campos desejados (tais como o autor das mensagens, as mensagens, a data, entre outros).

#### 3.2. WEB SERVICES USADOS

#### 3.3. ESTRUTURA DE DADOS

##### *Armazenamento de dados numa base de dados SQL*

Actualmente temos uma estrutura de dados pequena, apenas representativa de alguns dados obtidos, tais como sejam os países e as pessoas envolvidas.

Não temos ainda dados recolhidos a partir das redes sociais, uma vez que considerámos desnecessária a obtenção dos mesmos para esta fase, tendo em conta que a estrutura de dados irá ser alterada para uma *Triplestore*.

Numa próxima fase, as relações podem vir a sofrer alterações e consequentemente ser diferentes das ilustradas.

Como caso concreto disto temos as tabelas, nas quais iremos adicionar entidades relativas às informações extraídas das redes sociais.

Posto isto, a estrutura de dados irá sofrer alterações e ser objecto de melhoramentos nas próximas fases. Cumpre, por fim, mencionar que uma pessoa pode ser um jogador, um treinador, um dirigente ou um árbitro.

#### *Linked data*

Usámos o SPARQL para fazer queries de recolha de informação sobre os estádios do EURO 2016 e para os jogadores.

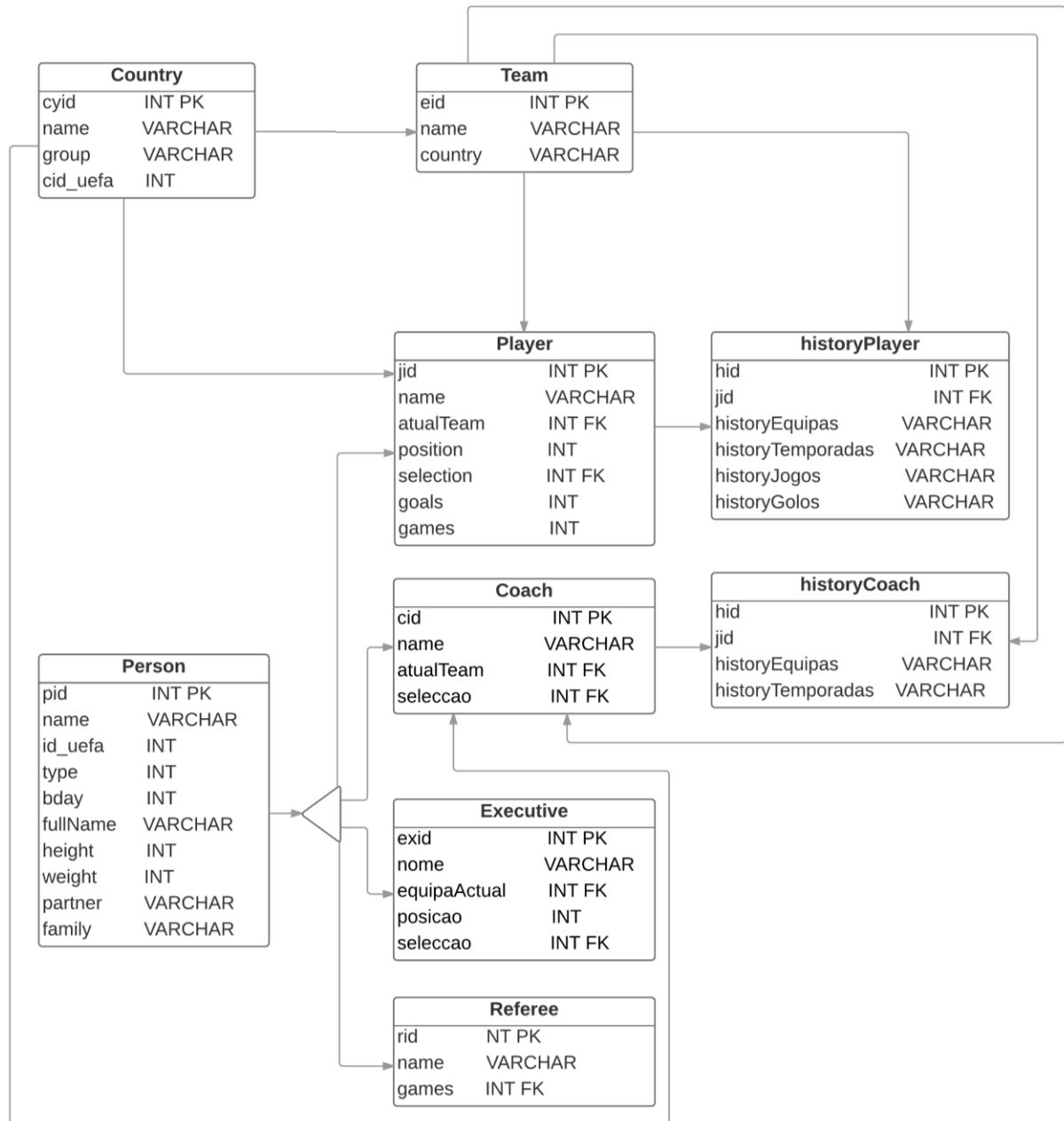
### **3.4. COMPONENTES IMPLEMENTADOS**

A propósito dos componentes implementados importa referir a existência de alguns *scripts* de *Web Scrapping* que permitem a obtenção de informação sobre jogadores (e os seus históricos), sobre treinadores, equipas da UEFA, bem como informação sobre os países envolvidos no EURO 2016

Consideramos também relevante mencionar o facto de termos, até à data, cerca de seis métodos fornecidos pelo nosso *Web Service*.

#### *Recolha de informação*

Na recolha de informação, pretendíamos usar APIs das redes sociais mais em voga como o Twitter, Facebook, Instragram, entre outros. Inicialmente tínhamos expectativas que outros sites pudessem fornecer o mesmo tipo serviço, o qual nos obrigou a fazer Web Scrapping sobre os mesmos (zerozero e uefa).



## 4. WEB SERVICES

### 4.1. TECNOLOGIAS USADAS

Em termos de tecnologias usadas, no âmbito do projecto, cumpre referir as seguintes:

- Filosofia RESTful;
- XML;
- JSON.

### 4.2. LISTA DE MÉTODOS

#### *Tabela de Web Services*

Quanto aos métodos empregues a propósito do *web service*, temos a enumerar os que se seguem.

#	Recurso	Método	Parâmetros	Resultado
1	/person/list	GET		Lista de todas as personalidades na competição.
2	/person/(:num)	GET	Id da personalidade	Informações pessoais da personalidade.
3	/person/(:num)/social	GET	Id da personalidade	Lista de <i>posts</i> de todas as redes da personalidades.
4	/person/(:num)/social/twitter	GET	Id da personalidade	Lista de <i>posts</i> do Twitter da personalidades.
5	/person/(:num)/social/twitter/image			
6	/person/(:num)/social/twitter/message			
7	/person/(:num)/social/facebook	GET	Id da personalidade	Lista de <i>posts</i> do Facebook da personalidades.

---

8 /person/(:num)/social/facebook/imagen

9 /person/(:num)/social/facebook/message

10 /person/(:num)/social/videos

11 /person/(:num)/image

12 /person/(:any)

13 /person/insertComment

14 /person/insertCommentTwo

15 /personalities/compare/(:any)/(:any)

16 /user/getuser

17 /search/saveQuestOne

18 /search/saveQuestTwo

19 /search/getQuestOne

20 /search/getQuestTwoe

21 /comments/(:num)

22 /comments/like\_dislike

---

23	/comments/(:num)/(:num)			
24	/relation/(:num)/(:num)/history	GET	Id da personalidade1; Id da personalidade2	Histórico da relação de duas personalidades.
25	/relation/(:num)/(:num)/percentile	GET	Id da personalidade1; Id da personalidade2	Percentagem da relação de duas personalidades.
26	/relation/(:num)/(:num)/social	GET	Id da personalidade1; Id da personalidade2	Todos os <i>posts</i> de todas as redes sociais da relação entre as personalidades
27	/relation/(:any)/(:any)/social			
28	/relation/(:any)/(:any)/social/twitter	GET	Id da personalidade1; Id da personalidade2	Todos os <i>posts</i> do Twitter da relação entre as personalidades.
29	/relation/(:any)/(:any)/social/twitter/image			
30	/relation/(:any)/(:any)/social/twitter/message			
31	/relation/(:any)/(:any)/social/facebook	GET	Id da personalidade1; Id da personalidade2	Todos os <i>posts</i> do Facebook da relação entre as personalidades.
32	/relation/(:any)/(:any)/social/facebook/image			
33	/relation/(:any)/(:any)/social/facebook			
34	/relation/(:num)/(:num)/social/news	GET	Id da personalidade1; Id da personalidade2	Todas as notícias que relacionam as duas personalidades.
35	/relation/(:num)/(:num)/social/videos	GET	Id da personalidade1; Id da personalidade2	Todos os vídeos que relacionam as duas personalidades.
36	/top/topduo	GET		Os três duos mais falados na <i>Web</i> .
37	/top/person/(:num)	GET	Id da personalidade	Os três indivíduos mais falados no geral.

38	/top/team/(:num)	GET	Id da personalidade	As três personalidades mais faladas de uma equipa.
39	/top/country/(:num)	GET	Id da personalidade	As três personalidades mais faladas de uma selecção.
40	/top/words/rel			
41	/country/list	GET		Lista de todos os países.
42	/country/list/euro16	GET		Lista de todos os países do euro16.
43	/country/(:any)/			
44	country/(:any)/bigflag			
45	country/(:any)/lilflag			
46	/country/list/(:any)	GET	Grupo	Lista de todos os países de um grupo.
47	/team/list	GET		Lista de todas as equipas.
48	/team/list/(:num)	GET	Id da equipa	Uma equipa.
49	/player/list	GET		Lista de todos os jogadores.
50	/player/(:num)	GET	Id do jogador	Um jogador.
51	/player/(:any)			
52	/player/(:num)/image			



53	/player/(:any)/history	GET	Id do jogador	Lista o histórico de um jogador (equipas, temporadas, jogos e golos).
54	/player/(:num) /(:num)/compareHistory			
55	/coach/list	GET		Lista de todos os treinadores.
56	/coach/(:num)	GET	Id do treinador	Um treinador.
57	/executive/list	GET		Lista de todos os executivos (directores desportivos, presidentes, etc).
58	/referee/list	GET		Lista de todos os árbitros.

### *Implementação de Web Services*

#### **Input**

Para cada pedido do *Web Service* é dado um URI, no qual cada segmento representa algo no nosso *Web Service*.

No primeiro segmento é dada a base do pedido: se esta é sobre países, equipas, jogadores, treinadores, dirigentes, árbitros, relações ou tops.

Os segmentos seguintes servem para especificar o que se pretende de cada um, indicando por vezes um parâmetro (ID de uma ou duas entidades ou a letra do grupo na competição).

#### **Output**

É enviado em JSON, uma lista com os resultados. O output é tratado pelos controladores, no qual obtém os resultados através de métodos, que tratam dos pedidos HTTP.

#### **Ligação ao Back-End**

Foi necessário integrar a implementação base de um *Web Service RESTful* no *CodeIgniter*. Assim, todas as funções que tratam de realizar pedidos HTTP são guardados como *helpers*.

Estas funções, por sua vez, fazem pedidos à base de dados através de chamadas aos *models*.

Nos *models* estão os métodos que ligam os dados à base de dados, através de operações CRUD.

Por fim, os controladores chamam as funções dos *helpers* de acordo com o URI recebido.

Existe um ficheiro fornecido pelo CI, `routes.php`, que facilita a configuração dos caminhos do URI para os métodos desejados. Evitando a configuração do `.htaccess`.

#### Exemplo:

```
$route['country/list/(:any)']='webservice/country/getteamsgroup'
```

#

- |   |                                      |   |
|---|--------------------------------------|---|
| 1 | <code>.../country/list/A</code>      | URI recebido, em que <code>A</code> corresponde ao grupo pretendido.  |
| 2 | <code>webservice/country</code>      | Caminho até ao controlador, neste caso <code>country</code> .   |
| 3 | <code>getteamsgroup</code>           | Método pretendido, que chama um <i>helper</i> para tratar do pedido HTTP, neste caso <code>country_rest_helper.php</code> . |
| 4 | <code>country_rest_helper.php</code> | que por sua vez chama funções do <code>country_model.php</code> .   |

Importa comentar que se distingue entre equipa e seleção/país, mesmo considerando que nesta competição (Euro 2016), as seleções são equipas.

A “equipa” é representativa dos clubes e a “seleção” representativa das nações em jogo.

### 4.3. EXEMPLOS

**Pedido:** `http://appserver.di.fc.ul.pt/~aw001/projecto/index.php/country/list`

**Resultado:**

```
[{"cyid":"1","name":"Albania\r\n","groups":"A","cyid_uefa":"2"}, {"cyid":"2", "name":"France\r\n","groups":"A","cyid_uefa":"43"}, {"cyid":"3","name":"R\nomania\r\n","groups":"A","cyid_uefa":"113"}, {"cyid":"4","name":"Switzerlan\nd\r\n","groups":"A","cyid_uefa":"128"}, {"cyid":"5","name":"England\r\n","g\nroups":"B","cyid_uefa":"39"}, {"cyid":"6","name":"Russia\r\n","groups":"B", "cyid_uefa":"57451"}, {"cyid":"7","name":"Slovakia\r\n","groups":"B","cyid_\nuefa":"58836"}, {"cyid":"8","name":"Wales\r\n","groups":"B","cyid_uefa":"14\n4"}, ...]
```

**Pedido:** <http://appserver.di.fc.ul.pt/~aw001/projecto/index.php/country/list/C>

**Resultado:**

```
[{"cyid":"9","name":"Germany\r\n","groups":"C","cyid_uefa":"47"},
{"cyid":"10","name":"Ireland\r\n","groups":"C","cyid_uefa":"63"},
{"cyid":"11","name":"Poland\r\n","groups":"C","cyid_uefa":"109"},
{"cyid":"12","name":"Ukraine\r\n","groups":"C","cyid_uefa":"57166"}]
```

**Pedido:** <http://appserver.di.fc.ul.pt/~aw001/projecto/index.php/person/list/35>

**Resultado:**

```
[{"pid":"35","name":"Hamdi Salihi","id_uefa":"66662","type":"2","bday":"19\01\1984","full Name":"Hamdi Salihi","height":"0","weight":"0","partner":"","family":""}]
```

**Pedido:** <http://appserver.di.fc.ul.pt/~aw001/projecto/index.php/player/list/37>

**Resultado:**

```
[{"jid":"37","name":"Heinz Lindner","atualTeam":"Frankfurt","position":"1","selection":"8","goals":"0","games":"0","pid_uefa":"0","pid_zz":"1122"}]
```

## 5. FRONT-END (ADMIN E USER)

### 5.1. TECNOLOGIAS USADAS

#### *Implementação do Front-End*

Quanto às tecnologias usadas para o desenvolvimento do *front-end* do website temos a referir as seguintes linguagens e bibliotecas a que recorreremos:

1. AJAX;
2. PHP;
3. jQuery;
4. D3.js;
5. Twitter Bootstrap.

#### *Front-End dinâmico com AJAX*

Existe uma secção de comentários referente a cada pessoa e relação entre pessoas, a qual vai sendo atualizada à medida que são recebidos novos posts das redes sociais e/ou comentários.

### 5.2. WEB SERVICES USADOS

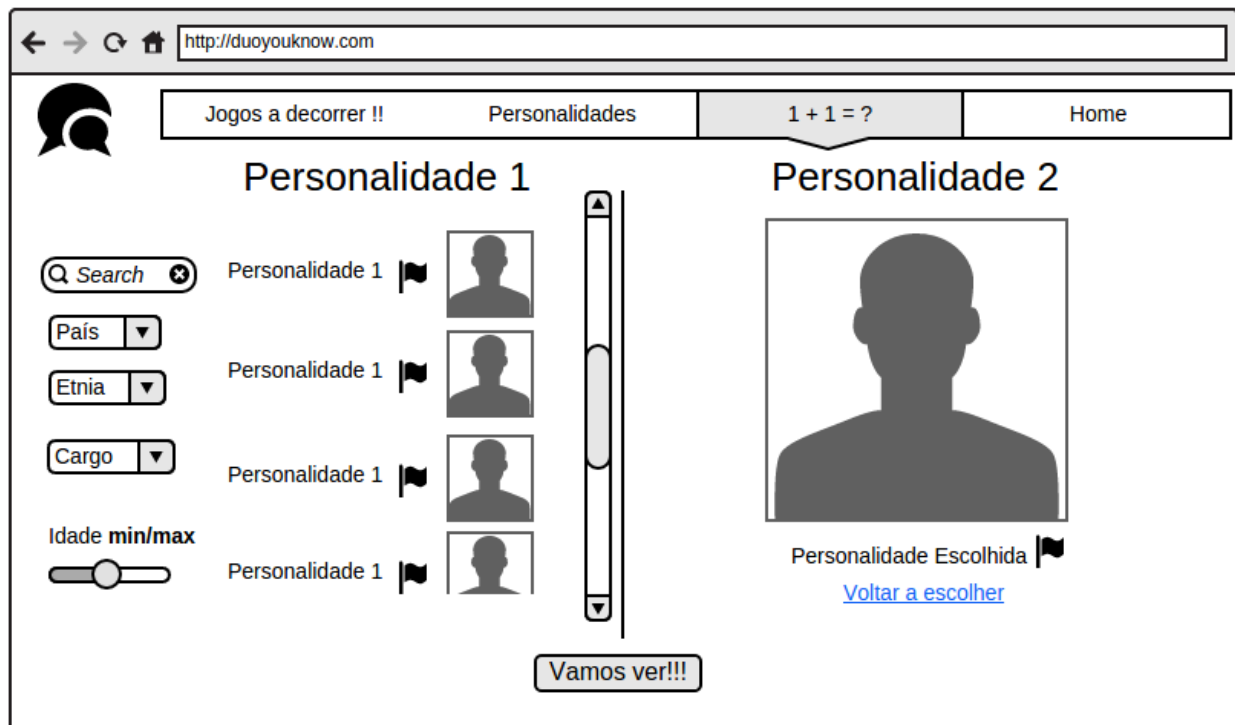
Irão ser usados dois *web services* para o registo/*login* dos utilizadores na aplicação.

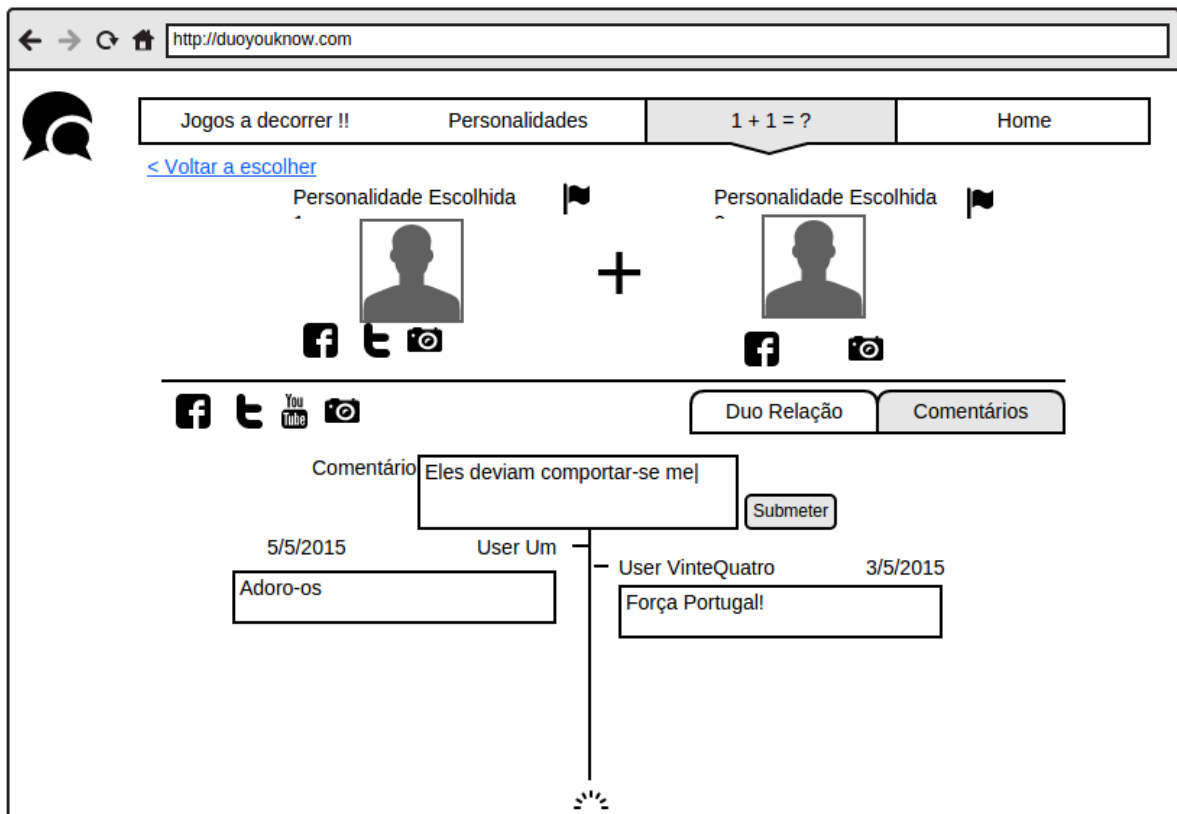
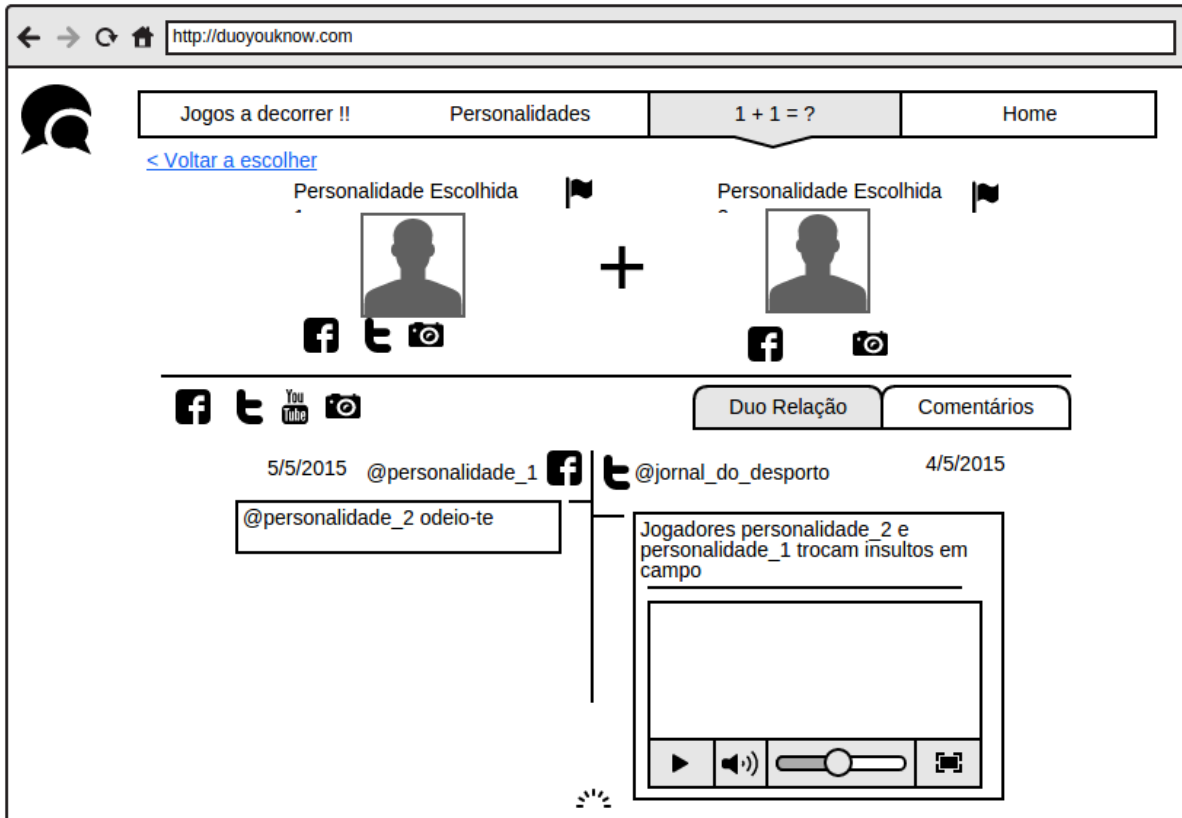
#### *APIs de Conta:*

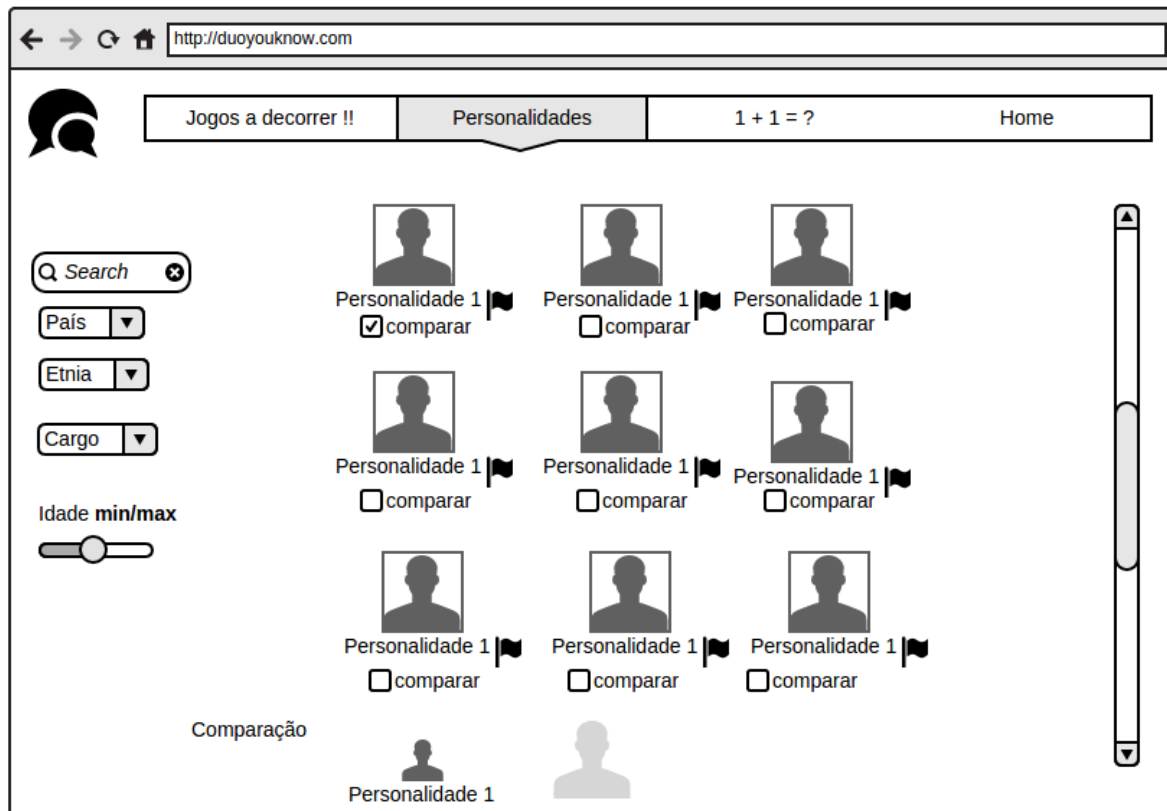
- Facebook;
- Google.

## 5.3. SCREENSHOTS

*Sketches de Interface*







### *Visualização interactiva com D3.js*

Relativamente ao D3, esta biblioteca é usada para disponibilizar informação relativa a pesquisas feitas pelos utilizadores, número de comentários e número de Tweets. Desta forma podemos avaliar a densidade das relações online.

### *Mecanismos de feedback de utilizador*

Fornecemos ao utilizador uma forma de interagir com outras pessoas na aplicação, através de comentários e o uso de “likes”.

## 6. DISCUSSÃO

### 6.1. PROBLEMAS ENCONTRADOS

A propósito dos desafios que encontrámos deparamo-nos com um contratempo que se materializa na forma de uma impossibilidade de extrair informação.

Queremos com isto dizer que a necessidade de realizarmos múltiplos pedidos num curto espaço de tempo, tem como consequência adversa o bloqueio ao acesso a websites de terceiros.

Como solução pretendemos obrigar a que haja um intervalo de tempo de espera entre cada pedido para evitar este problema.

A solução que propomos não é, porém, uma solução óptima, na medida em que o intervalo de tempo de espera entre pedidos se traduz numa extração morosa de dados que pode demorar várias horas.

Temos, pois, que se revela necessário visitar a possibilidade de reduzir a quantidade de informação a extrair. Temos, no entanto, consciência de que esta medida poderá ter um impacto negativo, podendo esta redução de informação desvalorizar a nossa aplicação, reduzindo o número de maneiras de testar o grau de relação entre personalidades.

Devido a fatores externos ao nosso controlo, foi-nos impossível fazer uso do Triple-Store, assim fomos forçados a usar MYSQL.

Quanto ao da API do Facebook, encontrámos contrariedades, tais como falta de documentação e clareza da mesma. Não conseguimos, ainda, recolher os dados que pretendíamos pelo que tivemos de arranjar uma solução análoga.