

# RELATÓRIO TÉCNICO



**Universidade do Minho**  
Escola de Engenharia



PEI2020 BettingSpree

26 Janeiro de 2021



**Universidade do Minho**  
Escola de Engenharia



# RELATÓRIO TÉCNICO

Projecto **Betting Spree**

ID Documento	RT-20201230-PEI2020
Versão	1.0
Acesso	Restrito
Data de emissão	26 Janeiro de 2021
Autores	Adriana Meireles a82582 Guilherme Andrade a80426 João Nunes a82300 Luís Braga a82088 Luís Cunha a83099 Luís Martins a82298 Mateus Silva a81952 Shahzod Yusupov a82617
Destinatário	PEI2020

# Conteúdo

<b>1</b>	<b>Equipe de trabalho</b>	<b>3</b>
1.1	Informação de Contacto . . . . .	3
<b>2</b>	<b>Sumário executivo</b>	<b>4</b>
<b>3</b>	<b>Introdução</b>	<b>5</b>
3.1	Oportunidades . . . . .	5
3.2	A Ideia . . . . .	6
3.3	Estrutura do relatório . . . . .	7
<b>4</b>	<b>Modelação do sistema</b>	<b>8</b>
4.1	Requisitos . . . . .	8
4.2	Diagramas de Use Cases . . . . .	13
4.2.1	Apostas . . . . .	14
4.2.2	Contas . . . . .	15
4.2.3	Social . . . . .	15
<b>5</b>	<b>Estrutura aplicacional</b>	<b>16</b>
5.1	Arquitetura . . . . .	17
5.2	Tecnologias . . . . .	18
5.2.1	Vue.js . . . . .	18

5.2.2	Python . . . . .	18
5.2.3	Node.js . . . . .	18
5.2.4	MySQL . . . . .	19
5.2.5	MongoDB . . . . .	19
5.3	Backend . . . . .	19
5.3.1	BetsApi . . . . .	19
5.3.2	DataApi . . . . .	19
5.3.3	ChatApi . . . . .	20
5.4	Frontend . . . . .	20
<b>6</b>	<b>Modelo</b>	<b>22</b>
<b>7</b>	<b>Interface</b>	<b>26</b>
<b>8</b>	<b>Lógica de Controle</b>	<b>33</b>
8.1	Manutenção dos dados de jogo . . . . .	33
8.2	Fecho de jogos e suas consequências . . . . .	33
8.3	Sistema de Ranking . . . . .	34
<b>9</b>	<b>Instalação e manutenção</b>	<b>35</b>
<b>10</b>	<b>Conclusão</b>	<b>37</b>

# 1

Equipe de trabalho

## 1.1 Informação de Contacto

Endereço Web	<a href="http://www.di.uminho.pt">http://www.di.uminho.pt</a>
Telefone	+351 253 510 170
Correio electrónico	alunos@.uminho.pt
Responsável do projeto	CloudCode
Morada	Departamento de Informática Universidade do Minho 4710-057 Gualtar, Braga

# 2

## Sumário executivo

O presente documento tem como principal objetivo a documentação da aplicação *BettingSpree*. Primeiramente, é feita uma introdução onde se focam os tópicos: Oportunidades, a ideia, a estrutura do relatório. Posteriormente, é apresentado o conjunto de requisitos levantados assim como os diagramas de *use cases*. De seguida, é apresentada e explicada a Estrutura Aplica- cional do projeto. São também apresentadas as principais páginas *web* realizadas seguido dos principais algoritmos implementados na manutenção dos dados de jogo e no fecho do mesmo. Por fim, foi feita uma apreciação crítica ao trabalho realizado.

### 3.1 Oportunidades

O mercado das apostas desportivas é um mercado grande e em crescimento. Só na Europa o mercado das apostas desportivas online é de cerca de 22 mil milhões de euros, e espera-se que este número cresça para 29 mil milhões já em 2022, verificando-se um crescimento de 10% ao ano.

Em Portugal, no primeiro trimestre de 2020 registou-se um valor de 34,5 milhões de euros em receita bruta das apostas desportivas à cota. Analogamente ao que aconteceu na Europa, também se verificou um aumento em cerca de 9,7 milhões de euros comparativamente ao mesmo período do ano passado.

Portanto o mercado encontra-se em crescimento, pelo que considera-se então fértil e adequado para o desenvolvimento de novas e inovadoras aplicações neste âmbito.

Como tal, e pela experiência dos próprios elementos do grupo, e terceiros, tipicamente o processo de apostar levanta vários problemas, o primeiro sendo a necessidade de utilizar várias plataformas externas de maneira a efetuar uma simples aposta. Para além do uso da própria casa de apostas para efetuar a dita aposta, utiliza-se tipicamente também um outro *website* para visualizar dados estatísticos relativamente à partida, e uma rede social de maneira a obter o *feedback* da aposta com outros utilizadores. Foi a partir desta situação retratada que surgiu o *BettingSpree* uma rede social de apostas desportivas.

## 3.2 A Ideia

A dimensão e potencial do mercado de apostas continua num crescimento acelerado. Cada vez mais plataformas com diferentes atributos e vantagens surgem, procurando principalmente a variedade de modalidades e tipo de apostas que se pode fazer. Esta tendência leva a que as soluções existentes se foquem no apostador como uma entidade individual. A diferenciação da *BettingSpree* está no foco dos apostadores como uma comunidade, promovendo o contacto e troca de informação entre diversas fontes.

O conjunto de possibilidades existentes, associado a um número enorme de combinações de múltiplos eventos desportivos num só boletim, pode apresentar-se como avassalador para novos apostadores. Os inúmeros fatores que contribuem para a tomada de decisão da aposta poderão não se mostrar explícitos de todo, desencorajando apostadores, quer experientes, quer principiantes, tornando o processo de escolha, só por si, limitador. De seguida, vem a questão do desfecho da aposta, um fator decisivo para a medida do sucesso da decisão tomada previamente e para a continuação da utilização de plataformas de apostas onde múltiplas apostas erradas poderão levar a uma desistência mais ou menos permanente. Por fim, chega a questão do após aposta, onde muitas vezes a decisão desdobra-se por apostar novamente ou levantar o dinheiro e finalizar a transação. Todo o processo está associado a pequenas decisões que, em grande escala, representam o futuro de uma plataforma do género.

A introdução de uma comunidade com um interesse comum representa uma oportunidade de assimilação por parte de todos os géneros de apostadores, onde o processo descrito anteriormente poderá ser auxiliado por outras entidades apostadoras, todas com o mesmo objetivo. Cada decisão tomada poderá ser alvo de discussão, à distância de uns poucos cliques, com várias comunidades definidas e criadas ao longo do espectro social. A vantagem do ponto de vista de negócio cai na grande variedade de perspetivas, em termos desportivos, partilhadas de forma pública, privada ou semiprivada, gerando montantes relevantes para várias abordagens, potencialmente oferecendo margens de lucro consideráveis aos *Stakeholders*.

A ideia de comunidade de apostadores não é um conceito novo. O problema é a utilização de plataformas externas para a sua criação, plataformas estas que não são desenhadas tendo em conta as necessidades do apostador comum.

A *BettingSpree* oferece especialização de serviços para o desenvolvimento saudável de infra-estruturas sociais entre jogadores, uma vertente com carácter único e inovador no mundo das apostas.



### 3.3 Estrutura do relatório

O presente relatório descreve o estado do projeto *Betting Spree* até ao momento que surgiu devido à necessidade que os utilizadores tem recorrer a plataformas externas para poderem partilhar, comunicar, simular e copiar apostas.

Começámos por descrever as oportunidades e a ideia do nosso projeto. Nestes tópicos, para além de se ter abordado a crescente expansão do mercado das apostas desportivas a nível nacional e na Europa também são expostas as principais limitações de plataformas já existentes.

Posteriormente, foi realizado o levantamento e análise dos requisitos bem como a sua especificação.

De seguida foi apresentada a estrutura aplicacional do projeto, onde foram descritas a arquitetura *Betting Spree* bem como as tecnologias utilizadas para a sua criação, destacando-se entre elas o **vue.js**, **python**, **node.js**, **MySQL** e **MongoDB**.

No capítulo intitulado *Modelo* falaremos sobre como é realizada a persistência de dados na plataforma.

Seguidamente, foram exibidas as principais páginas da interface que foram baseadas em *mocks* realizados anteriormente.

No capítulo *Lógica de Controlo* foram descritos os algoritmos mais relevantes. Foi principalmente focado na manutenção dos dados de jogo, referindo as principais dificuldades bem como as respetivas soluções. Também foi referido as consequências do término de jogos.

No capítulo *Instalação e Manutenção* abordar-se-á como será feita o *deploy* da plataforma.

No fim do trabalho é apresentada uma apreciação crítica do trabalho realizado.

# 4

## Modelação do sistema

Neste capítulo serão apresentados o conjunto de requisitos levantados, requisitos esses que guiam a posterior criação do sistema. De seguida, e de maneira a que o grupo conseguisse ter uma melhor noção, foram elaborados também diagramas de *use cases*.

### 4.1 Requisitos

Nesta secção são apresentados os requisitos funcionais levantados para o projecto. Para cada um é especificada a sua origem (método de levantamento), descrição e conflitos existentes entre requisitos.

Requisito #1
Descrição: O utilizador faz uma aposta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #2
Descrição: O utilizador pode partilhar uma aposta de forma privada ou pública.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #3
Descrição: O utilizador pode copiar uma aposta partilhada privada.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #4
Descrição: O utilizador pode colocar no seu boletim os jogos de uma aposta partilhada pública.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #5
Descrição: O utilizador pode seguir outro utilizador.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #6
Descrição: O utilizador pode comunicar com outros utilizadores.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #7
Descrição: O utilizador pode visualizar um ranking de utilizadores.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #8
Descrição: O utilizador pode visualizar uma lista de jogos disponíveis para apostar.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #9
Descrição: O utilizador pode fazer publicações.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #10
Descrição: O utilizador pode apagar publicações.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #11
Descrição: O utilizador pode comentar uma publicação.
Origem: Entrevista.
Conflitos: Nenhum

Requisito #12
Descrição: O utilizador pode apagar um comentário feito numa publicação.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #13
Descrição: O utilizador pode visualizar a lista de apostas.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #14
Descrição: O utilizador pode visualizar estatísticas sobre o encontro.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #15
Descrição: O utilizador pode criar rascunhos de apostas.
Origem: Entrevista.
Conflitos: Nenhum

Requisito #16
Descrição: O utilizador pode partilhar um rascunho.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #17
Descrição: O utilizador pode colocar os jogos de um rascunho partilhado no seu boletim.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #18
Descrição: O utilizador pode visualizar o perfil de um utilizador.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #19
Descrição: O utilizador pode depositar fundos.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #20
Descrição: O utilizador pode levantar fundos.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #21
Descrição: O utilizador pode visualizar publicações de outro utilizadores.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #22
Descrição: O utilizador pode editar o seu perfil.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #23
Descrição: O utilizador pode criar uma conta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #24
Descrição: O utilizador pode autenticar-se na sua conta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #25
Descrição: O utilizador pode desautenticar-se na sua conta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #26
Descrição: O utilizador pode visualizar o histórico das suas apostas.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #27
Descrição: O utilizador pode pesquisar por outros utilizadores.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #28
Descrição: O utilizador pode entrar num grupo
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #29
Descrição: O utilizador pode sair de um grupo
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #30
Descrição: O utilizador tem acesso a um feed de noticias global da aplicação.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #31
Descrição: O utilizador pode criar grupos.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #32
Descrição: O utilizador pode pesquisar por grupos.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #33
Descrição: O administrador pode aceitar pedidos de adesão aos grupos.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #34
Descrição: O administrado do grupo pode apagar conteúdo do grupo.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #35
Descrição: O administrador de um grupo pode tornar outro membro administrador.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #36
Descrição: O utilizador pode gostar de uma publicação.
Origem: Brainstorm.
Conflitos: Nenhum

## 4.2 Diagramas de Use Cases

A partir dos requisitos anteriores foram elaborados alguns diagramas de use cases em que foram identificados três subsistemas principais, o subsistema das apostas, o social e por fim o subsistema relativo às contas dos utilizadores.

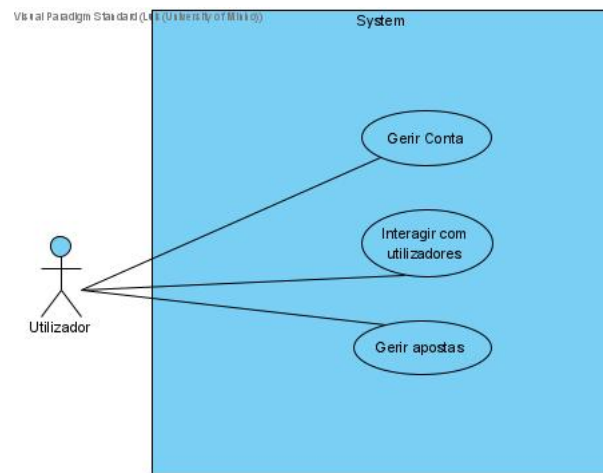


Figura 4.1: Diagrama de Use Cases geral.

## 4.2.1 Apostas

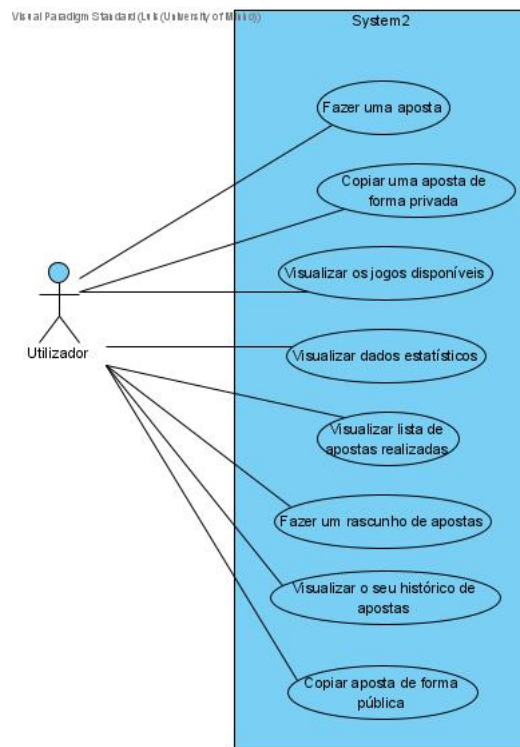


Figura 4.2: Diagrama de Use Cases das Apostas.

Na figura anterior, encontra-se os *use cases* pertencentes ao sub diagrama *Gerir Apostas*. E, como tal, encontram-se *use cases* relacionados com apostas desde fazer uma aposta até visualizar o seu histórico de apostas.



### 4.2.2 Contas

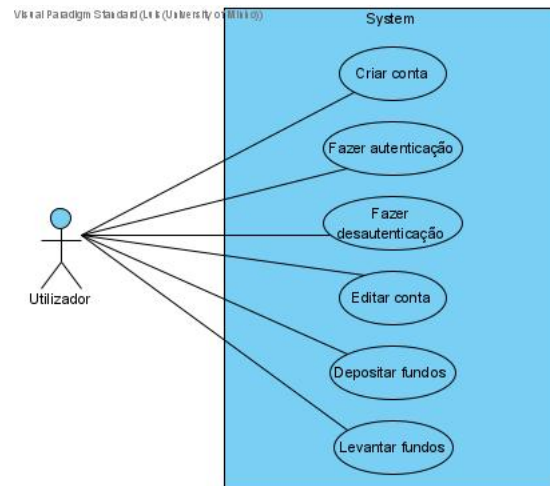


Figura 4.3: Diagrama de Use Cases das Contas.

No diagrama *use case* Contas estão representados todos os *use cases* associados à gestão de conta de um utilizador do sistema tais como, use cases de autenticação ou de gestão de perfil de conta.

### 4.2.3 Social

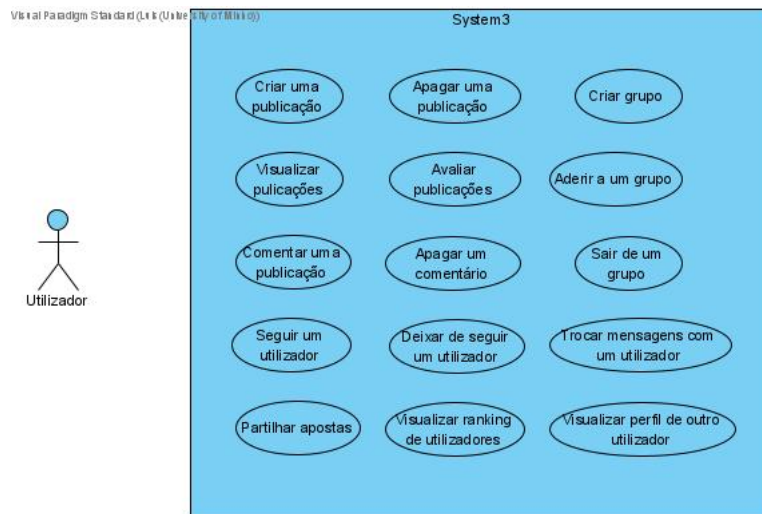


Figura 4.4: Diagrama de Use Cases da parte Social.

No diagrama supracitado encontram-se o conjunto de *use cases* relacionados com o sub-sistema social da plataforma. De tal forma que se encontram *use cases* relativos às operações de publicação, partilha e avaliação de *posts* por exemplo.

# 5

## Estrutura aplicacional

Nas secções seguintes irá ser apresentada a estrutura aplicacional do projeto, sendo apresentada a arquitectura do *Betting Spree* bem como as tecnologias utilizadas para a sua concepção.

## 5.1 Arquitetura

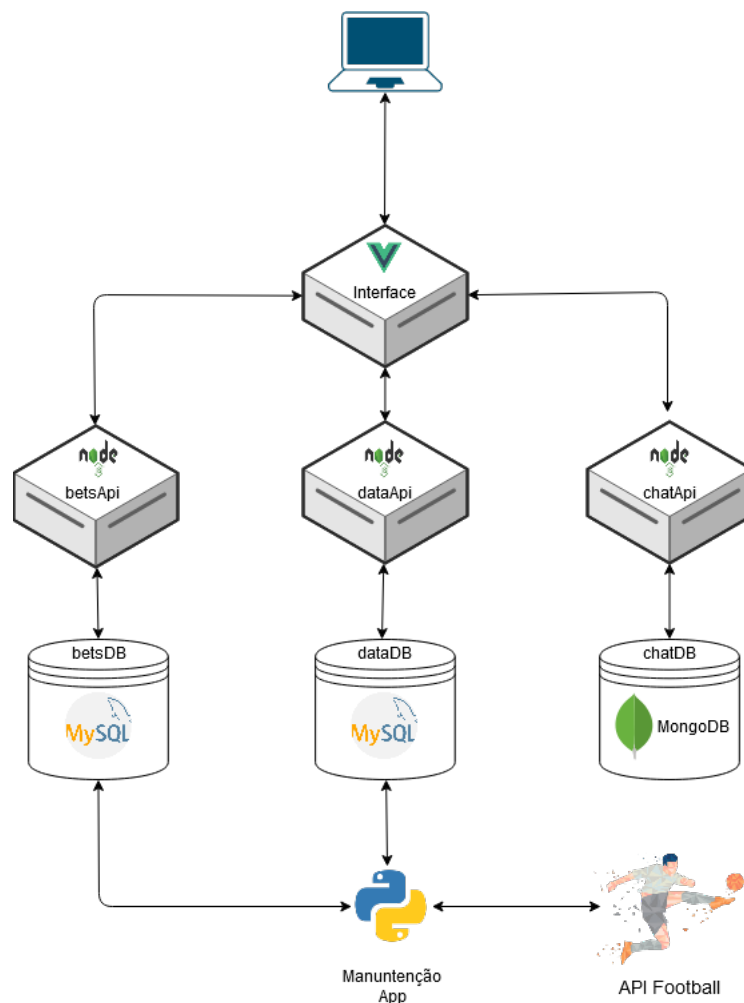


Figura 5.1: Arquitetura da plataforma.

Na figura acima está representada a arquitetura da aplicação.

Para o cliente visualizar as páginas *web* da interface, recorreu-se à ferramenta *Vue.js*.

No diagrama pode observar-se que existem três serviços essenciais de backend da aplicação. Primeiramente, temos o *Bets Api*. Este microserviço é responsável por servir os dados relativos aos jogos, isto é, países, ligas, equipas, jogos futuros, dados estatísticos sobre as equipas. Os dados foram guardados numa base de dados em *MySQL*.

De seguida, o microserviço *Chat Api* que facilita toda a troca de mensagens entre utilizadores na aplicação. Foi criado este microserviço único, em vez de o incluir no *Data Api*, uma vez que é uma funcionalidade bastante utilizada e que gera um grande volume de dados pelo que obtou-se por utilizar uma base de dados em *MongoDB*.

Por último, a *Data Api* trata da parte social da aplicação, isto é, publicações, informações de perfis, seguidores de um utilizador, comentários a publicações, grupos, entre outros dados. Os dados foram guardados numa base de dados em *MySQL*. Também fornece os serviços necessários para a autenticação, mais especificamente através da posse de um *jwt*.

Para a manutenção da aplicação, foram feitos alguns scripts, em *Python*, que permitem a atualização e inserção de novas apostas na base de dados e também para o próprio fecho das apostas.

## 5.2 Tecnologias

Como é possível observar na figura 9.1, foram utilizadas diversas tecnologias em cada módulo da aplicação. As razões que levaram à escolha destas ferramentas é apresentada de seguida.

### 5.2.1 Vue.js

Para construção das diversas páginas *web* recorreu-se à ferramenta *Vue.js*, que é um *framework JavaScript* de código-aberto, focado no desenvolvimento de interfaces. A escolha desta ferramenta teve em consideração a existência de bibliotecas já definidas que auxiliam o desenvolvimento, como o caso do *Vuetify*, que foi utilizado, e que possui um vasto conjunto de elementos já desenvolvidos.

### 5.2.2 Python

O *Python* é uma linguagem de programação popular que inclui múltiplos paradigmas, tanto o funcional como o orientado aos objectos. A escolha desta linguagem recaiu sobre a quantidade de módulos facilmente disponíveis através do *Python Package Index (PyPi)*. Esta linguagem foi utilizada de maneira a desenvolver os *scripts* que permitem a atualização e inserção de novas apostas na base de dados, e para além disso também é utilizado para o próprio fecho das apostas.

### 5.2.3 Node.js

O *Noje.js*, que pode ser definido como um ambiente de execução *Javascript server-side*, foi a ferramenta escolhida para desenvolver grande parte do *backend*, mais concretamente dos servidores das APIs.

### 5.2.4 MySQL

O *MySQL* é um popular sistema de gestão de bases de dados que utiliza a linguagem SQL como interface. Pelas suas características, mas fundamentalmente por ser relacional, foi a base de dados escolhida para armazenar grande parte dos dados da *Betting Spree*, nomeadamente os dados relacionados com o sistema de apostas e dados estatísticos, mas também os dados relacionados com a componente social da plataforma, como a informação dos utilizadores, das publicações, grupos e comentários.

### 5.2.5 MongoDB

A autenticação do sistema é efectuada através do uso de *JWT*. De modo a guardar registo de atividade dos utilizadores do sistema ou mesmo, forçar que um determinado utilizador se desconecte da plataforma é necessário guardar estes *tokens* numa base de dados. Para além disso, prevê-se que a utilização do *chat* da plataforma irá gerar um grande volume de dados pelo que este necessita de uma base de dados altamente escalável. Desta forma, utiliza-se uma base de dados MongoDB.

## 5.3 Backend

O *backend* da plataforma encontra-se dividido por 3 serviços, sendo assim uma arquitetura de micro serviços, cada micro serviço com a sua própria *Base de Dados* e uma funcionalidade própria. Todos os micro serviços estão desenvolvidos em *Nodejs*.

### 5.3.1 BetsApi

Este serviço está encarregue de lidar com os pedidos sobre países, ligas, jogos e respetivas odd's, que se encontram guardados na sua *Base de Dados* construída em *MySql*.

### 5.3.2 DataApi

Este servidor é responsável por grande parte da informação a nível de rede social, ou seja, gere a informação dos utilizadores, grupos, publicações, comentários, entre outros. Assim, para construir *Base de dados* foi utilizado *MySql*.

Além disso, este micro-serviço é responsável pela autenticação do sistema. Deste modo, grande

parte das rotas estão protegidas e para não ser devolvida uma resposta de não autorização, é necessário o pedido possuir um *jwt*. Para obter um *jwt*, é necessário realizar o pedido de login com as credenciais (email + password) e, caso estas estejam corretas, é devolvido um *jwt* com um determinado segredo definido e com um tempo de expiração associado. De referir que no payload do *jwt* são guardadas informações do utilizador que se autenticou (id do utilizador, seu email, etc).

### 5.3.3 ChatApi

Este componente tem a função de guardar e gerir todos os pedidos sobre mensagens entre utilizadores. Além disso, são usados *socket.io* que estão em comunicação com o Front End, de maneira a tornar possível a visualização e aparecimento de mensagens em tempo real. Dado que se prevê que a *Base de Dados* deste servidor cresça bastante, optou-se por desenvolver esta em *Mongo*.

De referir que de forma a que todas as rotas estivessem protegidas, os pedidos teriam de ser acompanhados de um *jwt* no body ou em *query string* e teriam que ser *jwt* válidos, ou seja, com um segredo utilizado pela aplicação e que não tenha expirado.

## 5.4 Frontend

O *frontend* da *Betting Spree* trata-se de um *frontend web*, e como tal, para o seu desenvolvimento, recorreu-se a uma ferramenta que permita o desenvolvimento de páginas *web*, o *Vue.js*. A escolha desta ferramenta deveu-se ao facto de grande parte dos elementos da equipa estarem familiarizados com a mesma. Para além disso, esta *framework JavaScript* tem um conjunto de vantagens, como a sua acessibilidade, versatilidade e excelente desempenho. A equipa decidiu também usar a biblioteca *Vuetify* que facilita a criação de diversos elementos.

Inicialmente, foram definidas as diversas rotas e criadas as *views* correspondentes. Foram criados também componentes do *Vue.js* que permitem a sua reutilização em diversas páginas.

A principal intenção no desenvolvimento do *frontend* foi que estas fossem intuitivas e simples por forma a facilitar a sua utilização.

Para além disso foi também criado um serviço adicional, uma página *web* adicional, separada da plataforma anterior, de maneira a auxiliar a tarefa de gestão por parte de administradores. Foi então criada uma *dashboard* que contém gráficos relativos ao dinheiro apostado e perdido na plataforma nos últimos cinco dias, o número de posts realizados nos últimos 5 dias e os posts realizados no dia anterior, e por fim o número de rascunhos criados nos últimos 5 dias e o

número de publicações privadas e públicas.

Para o utilizador ter acesso às funcionalidades do sistema, é necessário estar autenticado. Se as credenciais fornecidas pelo utilizador forem corretas, o Frontend, após realizar o pedido de login ao *dataApi*, o *jwt* recebido é guardado *Local Storage*, permitindo assim ao utilizador aceder às funcionalidades do sistema, enquanto o *jwt* for válido. Quando acabar o tempo de expiração do *jwt*, o sistema retira o *jwt* do *Local Storage* o que resulta numa desautenticação do utilizador.

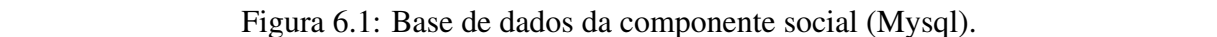
# 6

## Modelo

A partir dos requisitos chegaram-se aos seguintes modelos das bases de dados que são utilizadas de maneira a permitir a persistência dos dados dentro da plataforma.

O primeiro esquema montado foi relativamente à base de dados encarregue de armazenar os dados relativos à componente social da aplicação *web*. Estes dados são consumidos pelo micro serviço da *dataApi*.





Relativamente à entrega realizada anteriormente foram acrescentados mais campos na tabela bet e na tabela user. Na tabela bet acrescentou-se o campo *oddtotal* onde vai guardar a odd total de uma aposta. Na tabela user acrescentou-se o *copiaspriv* onde é guardado o número de cópias feitas, o *avgodd* que representa a odd média de todas as apostas feitas e o *rankscore* que guarda a pontuação do utilizador.

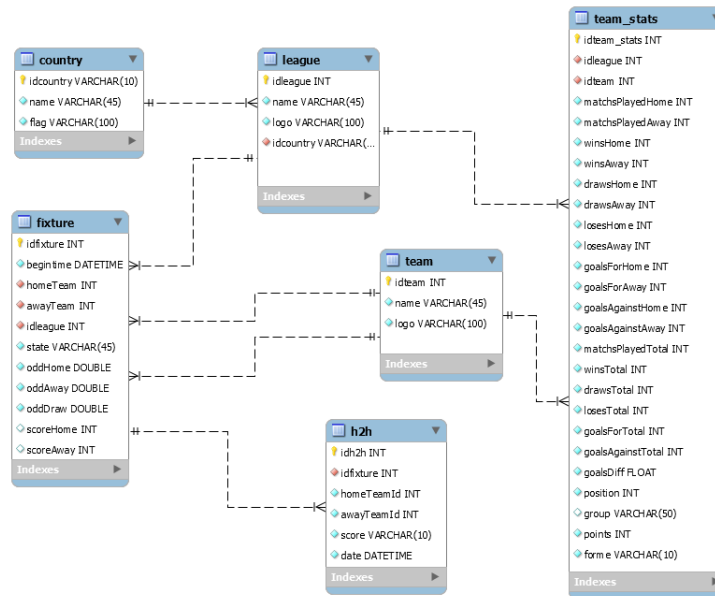


Figura 6.2: Base de dados da componente da aposta (Mysql).

O segundo esquema concebido foi relativamente aos eventos desportivos fornecidos na plataforma para efeitos de aposta. Para tal, após serem feitos os pedidos à *API* dos jogos, de seguida é necessário formatar os dados de maneira a depois ser possível o seu armazenamento e a posterior manipulação por parte do serviço da *betsApi*. Em título exemplificativo, uma liga terá a esta associada vários jogos (*fixtures*), na liga será guardada informação relativa ao seu logótipo (*logo*) e o nome da liga. Relativamente à *fixture*, é armazenada informação relativamente à hora de início do jogo, o estado do jogo (se já começou ou acabou), as *odds* das equipas que jogam em casa/fora e a *odd* de empate, bem como o *score* do jogo em questão. Existem também duas outras tabelas de maneira a armazenar dados estatísticos acerca das equipas (*team\_stats*) e outra para guardar a data e o resultado do confronto direto entre as duas equipas do evento desportivo (*fixture*) que é a *h2h*.

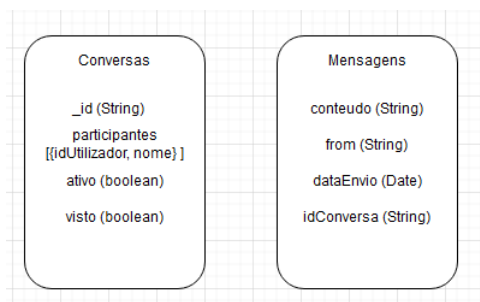


Figura 6.3: Base de dados da componente das conversas (MongoDB).

De maneira a implementar o *chatApi*, na figura anterior, encontra-se demonstrado como estão estruturados os documentos de maneira a guardar a informação relativa à comunicação, entre

utilizadores. No primeiro documento, encontra-se a entidade *Conversas* que guarda um identificador próprio gerado pelo *MongoDB*, um array de *participantes* de quem está na conversa, o atributo *ativo* indica se a conversa está ativa, ou seja, se não foi eliminada e, por último, o atributo *visto* que indica se a última mensagem foi visualizada pelo outro participante. No documento *Mensagens*, o *conteúdo* tem o papel de guardar o texto escrito e enviado por um dado utilizador (*from*). Além disso, é guardado a que conversa pertence a mensagem (*idConversa* e o atributo *dataEnvio* assinala a que horas e em que dias foi enviada a mensagem.

# 7

## Interface

No presente capítulo serão apresentadas as principais páginas realizadas com base nos *mockups* realizados anteriormente.

Na seguinte imagem apresenta-se a página *home* do utilizador, onde o utilizador poderá visualizar a lista de *posts* publicados por outros. Para além disso o utilizador nesta página poderá também navegar pelos grupos a que pertence.

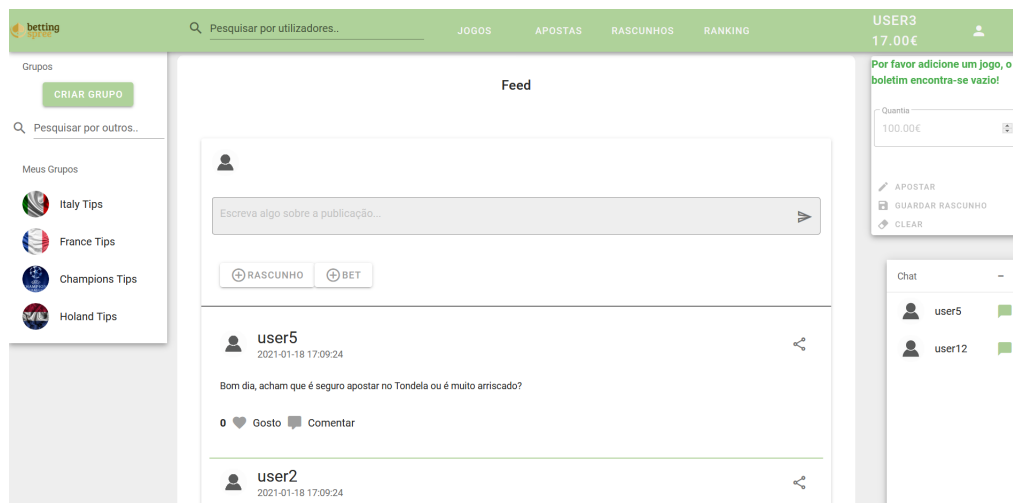


Figura 7.1: Página home do utilizador.

Ao clicar num dos grupos aos quais o utilizador pertence este será transportado para o *feed* de um grupo. Neste *feed* o utilizador terá acesso a publicações que são partilhadas para o tal grupo.

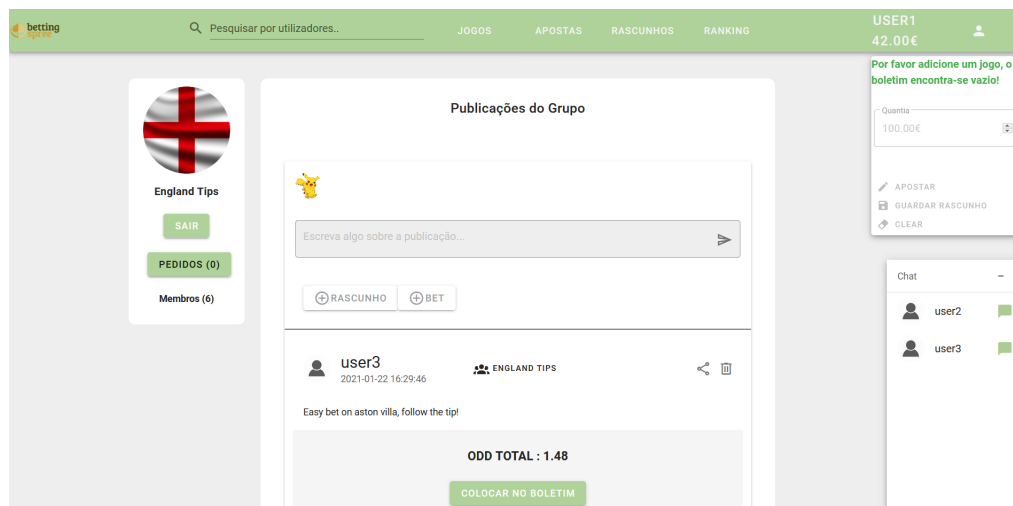


Figura 7.2: Feed de um grupo.

Relativamente à página das apostas realizadas o utilizador poderá consultar a lista de apostas que efectuou e que ainda não foram fechadas, isto é, o utilizador poderá consultar as apostas que ainda estão a decorrer.

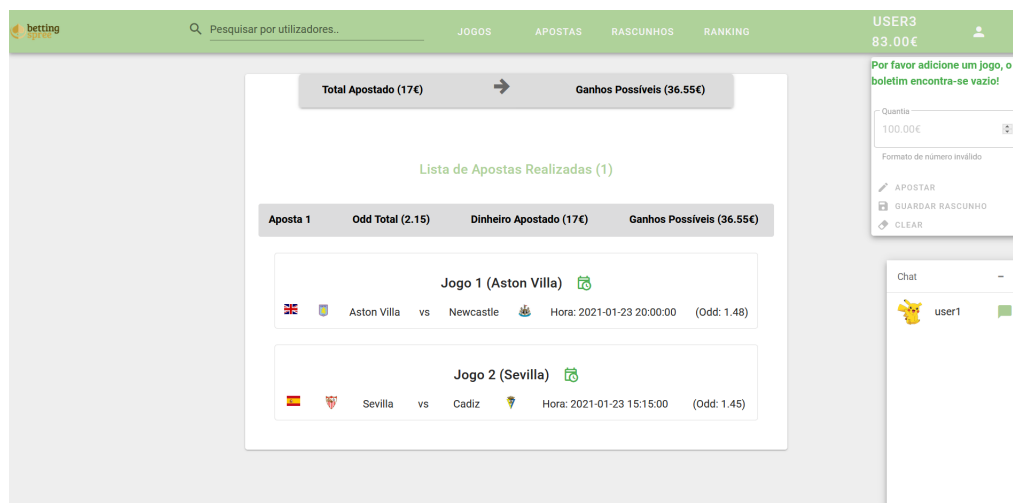


Figura 7.3: Página lista de apostas realizadas.

No que toca à página do utilizador, o *user* poderá examinar a lista das publicações que efectuou para o seu próprio *feed*, para além disso é nesta página que é possível consultar os seguidores e os utilizadores que este segue, para além de ser também possível aceder às definições da conta.

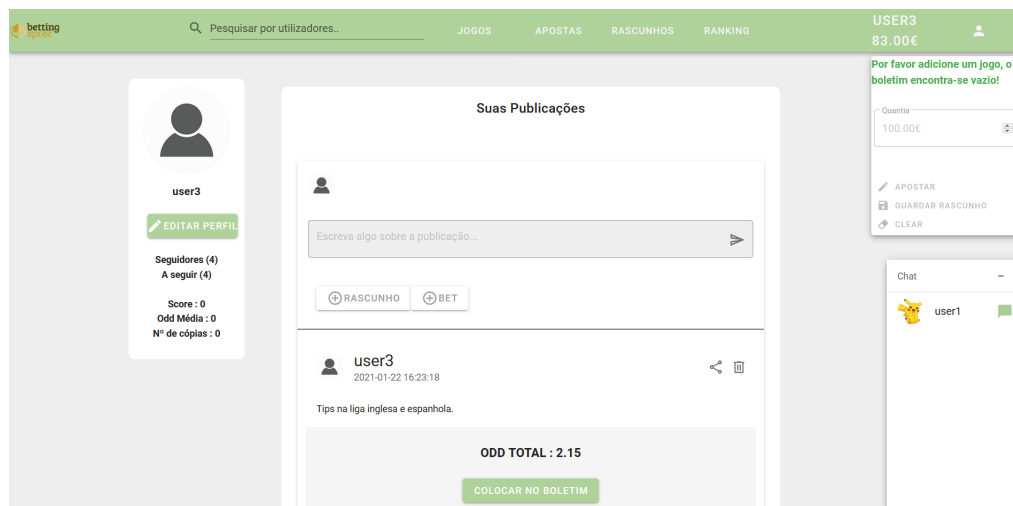


Figura 7.4: Página perfil do utilizador.

Os rankings dos utilizadores também se encontram disponíveis numa página própria, aqui é possível consultar os utilizadores com maior sucesso dentro da plataforma bem como os valores das métricas utilizadas para construir este ranking.

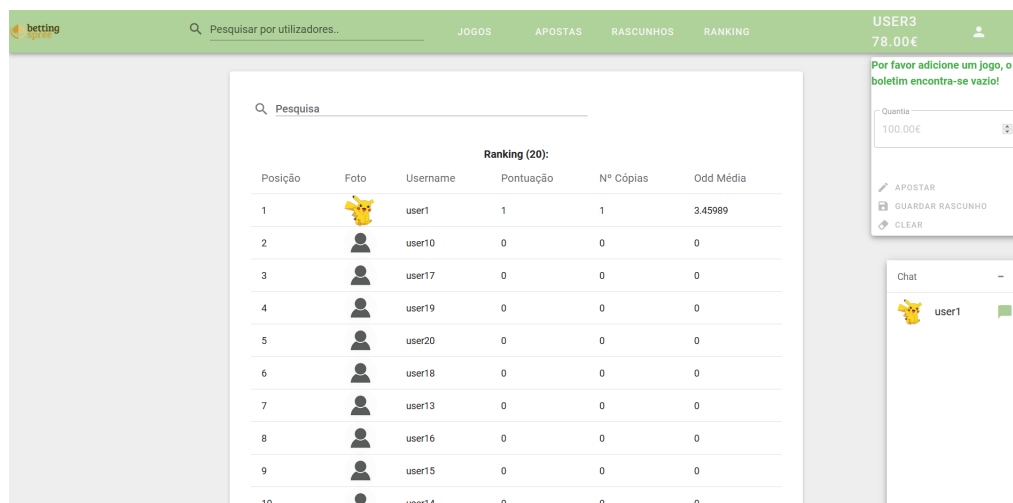


Figura 7.5: Página do ranking do utilizador.

Os utilizadores da plataforma também poderão consultar a lista das apostas que ganharam/perderam, nesta página o utilizador poderá navegar pelo conjunto de apostas que já foram fechadas.

Montante apostado: 65		Ganhos/Perdas: -58.00	
2021-01-20	Aposta 1	Odd Total (1.19)	Dinheiro Apostado (50€) Ganhos Possíveis (59.5€)
2021-01-20	Aposta 2	Odd Total (5.48)	Dinheiro Apostado (2€) Ganhos Possíveis (10.96€)
2021-01-20	Aposta 3	Odd Total (1.6)	Dinheiro Apostado (1€) Ganhos (1.6€)
2021-01-20	Aposta 4	Odd Total (3.8)	Dinheiro Apostado (1€) Ganhos Possíveis (3.8€)
2021-01-20	Aposta 5	Odd Total (5.5)	Dinheiro Apostado (1€) Ganhos Possíveis (5.5€)
2021-01-20	Aposta 6	Odd Total (3.42)	Dinheiro Apostado (2€) Ganhos Possíveis (6.84€)

Figura 7.6: Histórico das apostas realizadas.

Relativamente à vertente da casa de apostas, de seguida apresenta-se a página das apostas onde o utilizador poderá consultar a lista de jogos disponíveis para o próprio apostar. De maneira a auxiliar o utilizador, existe um conjunto de mecanismos de exploração de dados como o *filter* de apostas por país, ou a própria pesquisa por uma aposta específica.

Países	
Todos	
England	
Spain	

Pesquisa	
2021-01-23 20:00	Aston Villa - Newcastle 1.48 4.75 6
2021-01-23 15:15	Sevilla - Cadiz 1.45 4.1 8.25
2021-01-22 20:00	Levante - Valladolid 2.1 3.1 3.7
2021-01-23 20:00	Alaves - Real Madrid 7 4.6 1.45
2021-01-23 17:30	Real Sociedad - Real Betis 1.83 3.9 4.1
2021-01-23 13:00	Huesca - Villarreal 4.33 3.5 1.87

Figura 7.7: Página das apostas.

Para cada uma das partidas disponíveis é também possível consultar as estatísticas do jogo. De maneira a melhor informar o apostador, apresentam-se estatísticas como a classificação das equipas e um todo outro conjunto de estatísticas tais como golos marcados, golo sofridos, etc. Para além disso também se apresenta uma tabela com o confronto direto entre as duas equipas.

Premier League Estatísticas Aston Villa vs Newcastle

	Equipa	Posição	Pontos	Jogos Jogados	Vitórias	Empates	Derrotas	Golos Marcados	
	Aston Villa	11	26	16	8	2	6	29	
	Leeds	12	23	18	7	2	9	30	
	Crystal Palace	13	23	19	6	5	8	22	
	Wolves	14	22	19	6	4	9	21	
	Newcastle	15	19	18	5	4	9	18	
	Burnley	16	19	18	5	4	9	10	
	Brighton	17	17	19	3	8	8	22	

<b>Aston Villa:</b>	<b>Newcastle:</b>
7 Jogos jogados em casa	9 Jogos jogados em casa
9 Jogos jogados fora de casa	9 Jogos jogados fora de casa
3 Vitórias em casa	3 Vitórias em casa
5 Vitórias fora	2 Vitórias fora
3 Derrotas casa	4 Derrotas casa
3 Derrotas fora	5 Derrotas fora
1 Empates fora	2 Empates fora

Figura 7.8: Dialog estatísticas da partida.

11 Golos sofridos em casa	15 Golos sofridos em casa
7 Golos sofridos fora	15 Golos sofridos fora
2.14 Média de golos marcados em casa	1.11 Média de golos marcados em casa
2 Média de golos marcados fora de casa	0.89 Média de golos marcados fora de casa
1.57 Média de golos sofridos em casa	1.67 Média de golos sofridos em casa
0.78 Média de golos sofridos fora de casa	1.67 Média de golos sofridos fora de casa
1.81 Média de golos marcados	1 Média de golos marcados
1.13 Média de golos sofridos total	1.67 Média de golos sofridos total

Data	Equipa Casa	Resultado	Equipa Fora
2020-06-24	Newcastle	1-1	Aston Villa
2019-11-25	Aston Villa	2-0	Newcastle
2017-02-20	Newcastle	2-0	Aston Villa
2016-09-24	Aston Villa	1-1	Newcastle
2016-05-07	Aston Villa	0-0	Newcastle

Figura 7.9: Continuação do dialog estatísticas da partida.

Por fim, apresenta-se um chat que serve como forma de comunicação direta entre utilizadores. Através deste chat, o utilizador consegue enviar mensagens para todos os utilizadores que tenham a sua conta como pública. Caso se trate de uma conta privada, este deve seguir o utilizador para conseguir utilizar o chat com o mesmo.



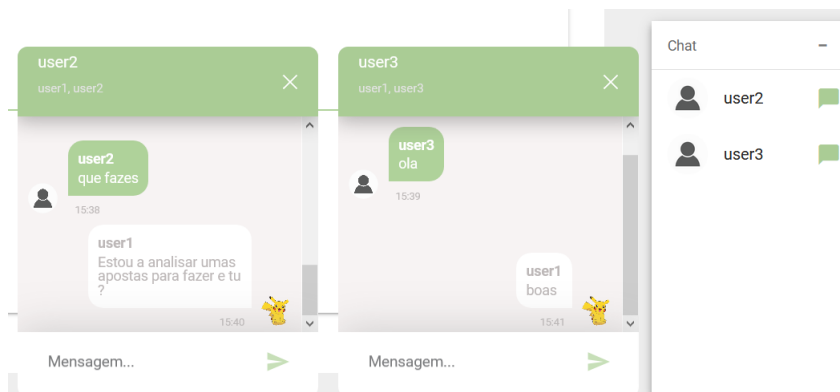


Figura 7.10: Bate papo entre utilizadores.

Para além disso, foi também criado um serviço distinto dos outros de maneira a providenciar informação em relação à utilização do sistema. A tarefa de gestão e análise por parte dos administradores do sistema pode constituir uma tarefa muito importante para um desenvolvimento saudável, complementando aquilo que pode ser visto como a manutenção de todas as componentes do projeto. A utilização de certas medidas pode providenciar *feedback* sobre o grau de sucesso de algumas das ferramentas implementadas na plataforma, o que permite avaliar o investimento de recursos assim como o potencial para melhorias futuras nestas mesmas. É importante acompanhar o impacto daquilo que são as inovações introduzidas pela plataforma, pois poderá validar toda a essência do projeto. Outras taxas de utilização mais comuns também são úteis para medir o sucesso do sistema como um todo. O mercado de apostas apresenta-se com grande volume mas bastante competitivo, daí ser bastante relevante obter estatísticas que permitam explicitar um crescimento da aplicação. Neste contexto, métricas de natureza prototipada foram desenvolvidas de forma a dar alguma noção. Para utilização real, recomenda-se maior flexibilidade na disposição dos dados retirados diretamente da base de dados, de forma a permitir a observação de padrões suficientemente apelativos para que decisões informadas sejam tomadas em relação à plataforma.



Figura 7.11: Dashboard do administrador.

# 8

## Lógica de Controlo

### 8.1 Manutenção dos dados de jogo

Um dos desafios deste projeto foi a obtenção de fontes de dados para alimentar o sistema de apostas. Neste momento, a plataforma apenas tem na sua base de dados jogos relativos a três ligas (*Premier League*, *LaLiga*, *Primeira Liga*) onde podem haver, por exemplo, até dez jogos por dia dependendo da época. Dito isto, com facilidade se prevê que ao adicionar mais ligas o número de jogos diários aumentará de uma forma que seria impossível de manter manualmente. De forma a mitigar este problema desenvolveu-se uma aplicação em *python* que automatiza a manutenção destes dados. Mais especificamente, periodicamente (de 24 em 24 horas) realiza pedidos à *API Football* para obter dados relativos aos jogos com cinco dias de antecedência. Uma atualização semelhante é feita duas vezes por dia para as estatísticas.

### 8.2 Fecho de jogos e suas consequências

O término de um jogo traz consigo várias implicações como, por exemplo, a alteração do estado do jogo na base de dados, o fecho de eventos em que o utilizador apostou e a possível decisão de ganhou ou perdeu nas apostas dos vários utilizadores. Para tal, adicionou-se ao serviço da secção anterior serviços que selecionam jogos a decorrer nas próximas duas horas e verificam quando o evento poderá ser fechado. Quando o *status* do jogo for *Match Finished* o jogo será fechado na base de dados e nos eventos da base de dados *Databettingspre*. De forma a completar este serviço elaborou-se um outro que verifica quais as apostas que ainda não se

encontram fechadas e averigua se, de facto, é possível fechá-las e atribuir ou não, prémio ao utilizador.

## 8.3 Sistema de Ranking

O sistema de *ranking*, uma das *features* da aplicação, é calculada da seguinte forma:

$$\text{score} = \text{odd média} * (\text{apostas ganhas} - \text{apostas perdidas}) + \text{nº de cópias}$$

As componentes da equação descrita acima podem ser descritos da seguinte forma:

- A *odd média*, como o nome indica diz respeito à *odd média* de todas as apostas realizadas pelo utilizador, sendo que apenas se tem em conta apostas realizadas pelo mesmo e não copiadas em privado de outrem;
- Nas apostas ganhas também somente contam apostas feitas pelo utilizador e não copiadas em privado;
- Nas apostas pedidas o caso é semelhante ao das apostas ganhas, isto é, apenas são utilizadas para cálculo apostas não copiadas em modo privado;
- O número de cópias diz respeito à quantidade de cópias realizadas a apostas privadas partilhadas pelo utilizador em questão.

Achou-se pertinente colocar o sistema de cálculo do *score* sob a forma de um *trigger* na base de dados da componente social, que é despoletado sempre que é realizada uma atualização na tabela *bet*.

# 9

## Instalação e manutenção

O deploy da aplicação foi realizado em docker na cloud pública, nomeadamente na Google Cloud Platform.

Foi elaborado um *dockerfile* para cada um dos microserviços, para a aplicação de *back office* que trata da manutenção dos dados, isto é, de obter dados relativos a jogos futuros, *odds* e estatísticas junto da *API* e, por fim, para as bases de dados.

Por fim, elaborou-se um ficheiro *docker-compose* onde são invocadas todas as imagens criadas e depositadas no *docker hub*.

Como seria de esperar houve alguns *stumbling blocks* neste processo. Houve algumas dificuldades no que toca à utilização do servidor de *proxy nginx* com a funcionalidade de *chat* onde se utiliza *sockets* e é necessário realizar a configuração correta.

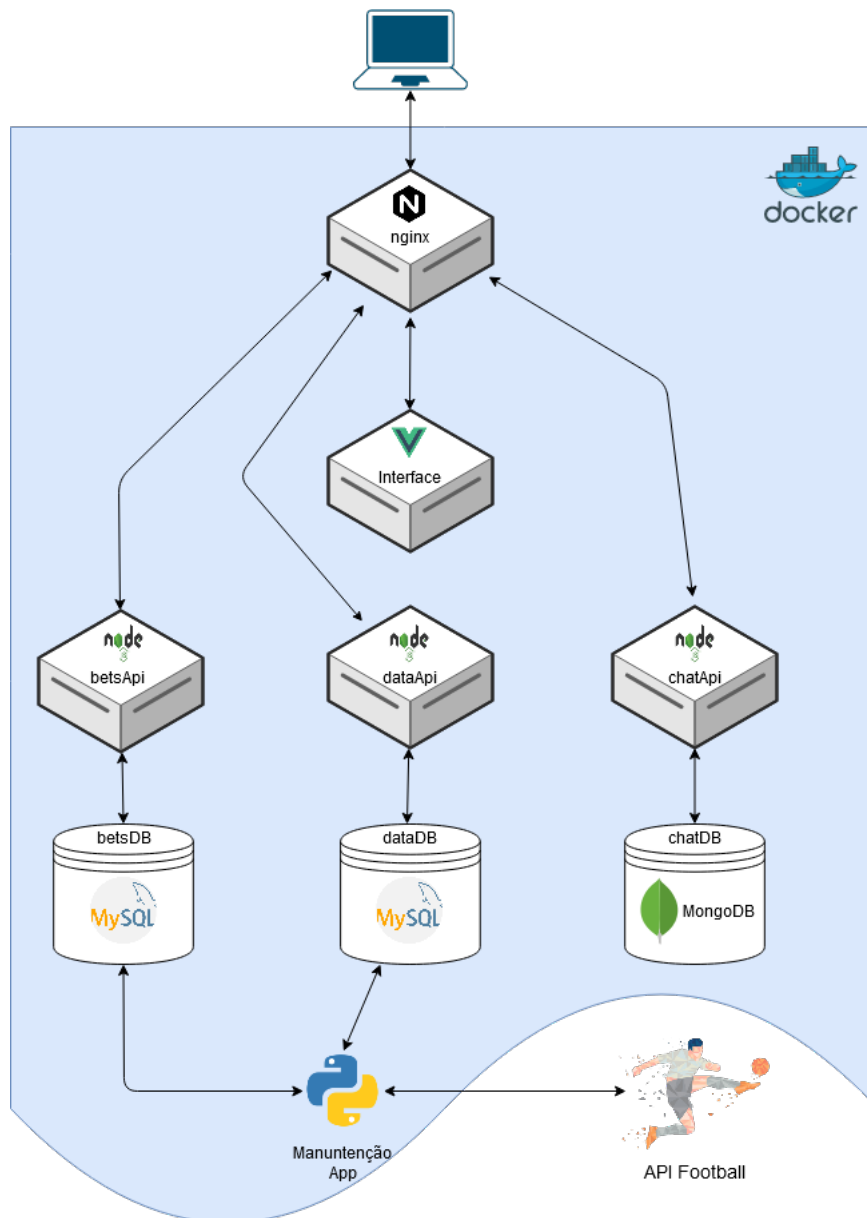


Figura 9.1: Arquitetura da plataforma em quando em *deploy*.

De uma forma geral a arquitetura mantém-se. Apenas é colocada em *containers docker* e acrescentada um servidor *proxy nginx*, que facilita a comunicação com o exterior.

# 10

## Conclusão

A ideia de comunidade está na essência humana. Este projeto procura juntar as mais diversas opiniões num assunto comum. É um aspecto fundamental e único que torna a execução deste viável, duma forma inovadora. Outros aspectos surgem que dificultam a sua execução, tal como assuntos legais, licenças, financiamento, investimento, manutenção, *design*, entre outros. Na hora de pesar os prós e os contras, resume-se todos os argumentos a uma questão de rentabilidade. Neste caso, existe uma grande margem de aproveitamento, tendo em conta o estado do mercado na área, assim como as necessidades exigidas, consciente ou inconscientemente, pelos apostadores. Deve ser mantido como objetivo principal, a promoção de uma sociedade entre jogadores, mas sempre mantendo presente o jogador como uma entidade individual, procurando uma longevidade do projeto direcionado às massas.