

RELATÓRIO TÉCNICO



Universidade do Minho
Escola de Engenharia



PEI2020 BettingSpree

30 de Dezembro de 2020



Universidade do Minho
Escola de Engenharia



RELATÓRIO TÉCNICO

Projecto **Betting Spree**

ID Documento	RT-20201230-PEI2020
Versão	1.0
Acesso	Restrito
Data de emissão	30 de Dezembro de 2020
Autores	Adriana Meireles a82582 Davide Matos a80970 Guilherme Andrade a80426 João Nunes a82300 Luís Braga a82088 Luís Cunha a83099 Luís Martins a82298 Mateus Silva a81952 Shahzod Yusupov a82617
Destinatário	PEI2020

Conteúdo

1	Equipe de trabalho	3
1.1	Informação de Contacto	3
2	Sumário executivo	4
3	Introdução	5
3.1	Oportunidades	5
3.2	A Ideia	6
3.3	Requisitos	7
3.4	Estrutura do relatório	10
4	Estrutura aplicacional	12
4.1	Arquitetura	13
4.2	Tecnologias	14
4.2.1	Vue.js	14
4.2.2	Python	14
4.2.3	Node.js	14
4.2.4	MySQL	15
4.2.5	MongoDB	15
4.3	Backend	15

4.3.1	AuthApi	15
4.3.2	BetsApi	16
4.3.3	ChatApi	16
4.3.4	DataApi	16
4.4	Frontend	16
5	Modelo	17
6	Interface	20
7	Lógica de Controlo	24
7.1	Manutenção dos dados de jogo	24
7.2	Fecho de jogos e suas consequências	24
8	Instalação e manutenção	26
9	Conclusão	27

1

Equipe de trabalho

Caraterização dos elementos da equipe.

1.1 Informação de Contacto

Endereço Web	http://www.di.uminho.pt
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 descrição dos requisitos a implementar e a estrutura do relatório. Posteriormente, é apresentada e explicada a Estrutura Aplicacional do projeto. De seguida, são apresentadas as principais páginas *web* realizadas. Depois são descritos os 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 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.
Origem: Introspeção.
Conflitos: Nenhum

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

Requisito #4
Descrição: O utilizador pode seguir outro utilizador.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #5
Descrição: O utilizador pode comunicar com outros utilizadores.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #6
Descrição: O utilizador pode visualizar um ranking de utilizadores.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #7
Descrição: O utilizador pode visualizar uma lista de jogos disponíveis para apostar.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #8
Descrição: O utilizador pode fazer publicações.
Origem: Introspeção.
Conflitos: Nenhum

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

Requisito #10
Descrição: O utilizador pode comentar uma publicação.
Origem: Entrevista.
Conflitos: Nenhum

Requisito #11
Descrição: O utilizador pode apagar um comentário feito numa publicação.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #12
Descrição: O utilizador pode visualizar a lista de apostas.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #13
Descrição: O utilizador pode visualizar estatísticas sobre o encontro.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #14
Descrição: O utilizador pode criar rascunhos de apostas.
Origem: Entrevista.
Conflitos: Nenhum

Requisito #15
Descrição: O utilizador pode visualizar o perfil de um utilizador.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #16
Descrição: O utilizador pode depositar fundos.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #17
Descrição: O utilizador pode levantar fundos.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #18
Descrição: O utilizador pode visualizar publicações de outro utilizadores.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #19
Descrição: O utilizador pode editar o seu perfil.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #20
Descrição: O utilizador pode criar uma conta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #21
Descrição: O utilizador pode autenticar-se na sua conta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #22
Descrição: O utilizador pode desautenticar-se na sua conta.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #23
Descrição: O utilizador pode visualizar o histórico das suas apostas.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #24
Descrição: O utilizador pode pesquisar por outros utilizadores.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #25
Descrição: O utilizador pode entrar num grupo
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #26
Descrição: O utilizador pode sair de num grupo
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #27
Descrição: O utilizador tem acesso a um feed de noticias global da aplicação.
Origem: Introspeção.
Conflitos: Nenhum

Requisito #28
Descrição: O utilizador pode criar grupos.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #29
Descrição: O utilizador pode pesquisar por grupos.
Origem: Brainstorm.
Conflitos: Nenhum

Requisito #30
Descrição: O utilizador pode dar upvote a uma publicação.
Origem: Brainstorm.
Conflitos: Nenhum

3.4 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

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.

4.1 Arquitetura

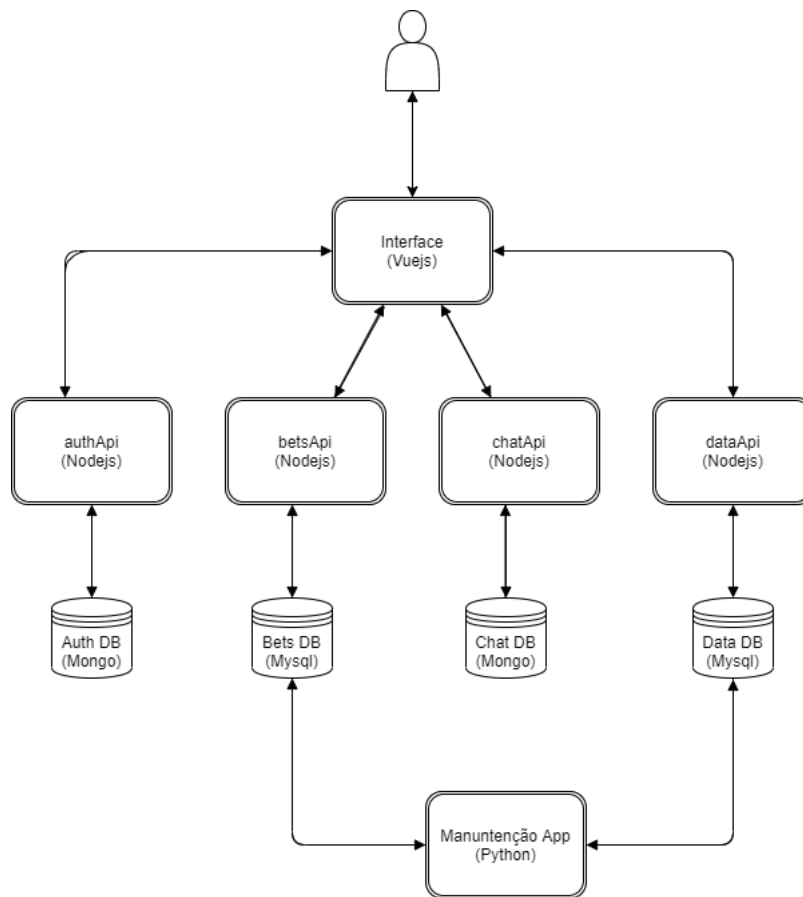


Figura 4.1: Arquitetura.

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 quatro serviços essenciais da aplicação. Primeiramente, o *Auth Api* fornece os serviços necessários para a autenticação, mais especificamente o *refresh* da sessão do utilizador de uma forma discreta. Os dados foram guardados numa base de dados em *MongoDB*.

Posteriormente, 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 obteve-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*.

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.

4.2 Tecnologias

Como é possível observar na figura 4.1, foram utilizadas diversas tecnologias em cada módulo da aplicação. As razões que levaram à escolha destas ferramentas é apresentada de seguida.

4.2.1 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.

4.2.2 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.

4.2.3 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.

4.2.4 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.

4.2.5 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.

4.3 Backend

O *backend* da plataforma encontra-se dividido por 4 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*.

4.3.1 AuthApi

Começando pelo servidor responsável pela autenticação no sistema, nomeadamente o servidor *authApi*, utilizou-se *JsonWebToken* para garantir a autenticação do utilizador e proteger as rotas de todas as outras *api*'s. Este microserviço, aquando de um pedido de autenticação por parte de um utilizador, é gerado um *JsonWebToken* com duração curta, de maneira a evitar que, caso seja roubado, este expire rápido. Para manter o utilizador autenticado ao longo do tempo sem precisar de voltar a introduzir as suas credenciais, foi gerado um *refreshToken* com a flag *HTTP_ONLY* para que não possa ser acedido/roubado com *JavaScript* do lado do cliente e também para que se possa renovar o *JsonWebToken* quando ele expirar e manter o utilizador autenticado. Os utilizadores, respetivas sessões e *refreshToken* são guardados numa *Base de Dados* em *Mongo*.

4.3.2 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*.

4.3.3 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*.

4.3.4 DataApi

Por último, encontra-se o servidor 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*.

4.4 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.

5

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*.

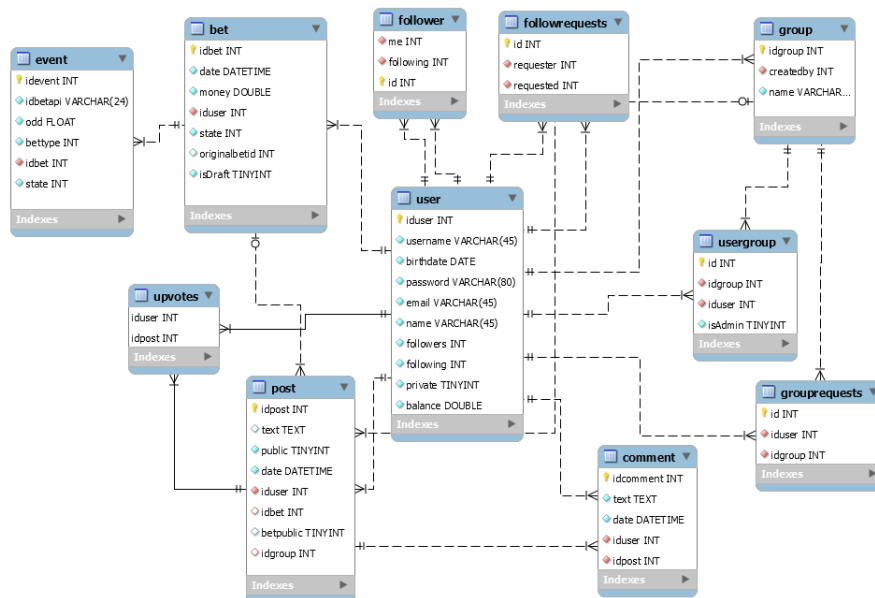


Figura 5.1: Base de dados da componente social (Mysql).

Em título exemplificativo, um utilizador (*user*) pode realizar apostas (*bet*) que contém a data que

em a aposta foi realizada, o dinheiro investido, o estado de aposta (se já terminou/ganhou/perdeu) e, caso seja uma aposta copiada possui o identificador da aposta copiada. Numa outra tabela, irão estar presentes o conjunto de jogos da aposta realizada (*event*). Para além disso, são guardados também dados relativos aos *posts* que o utilizador efetua, e, por exemplo, os grupos a que o utilizador pertence.

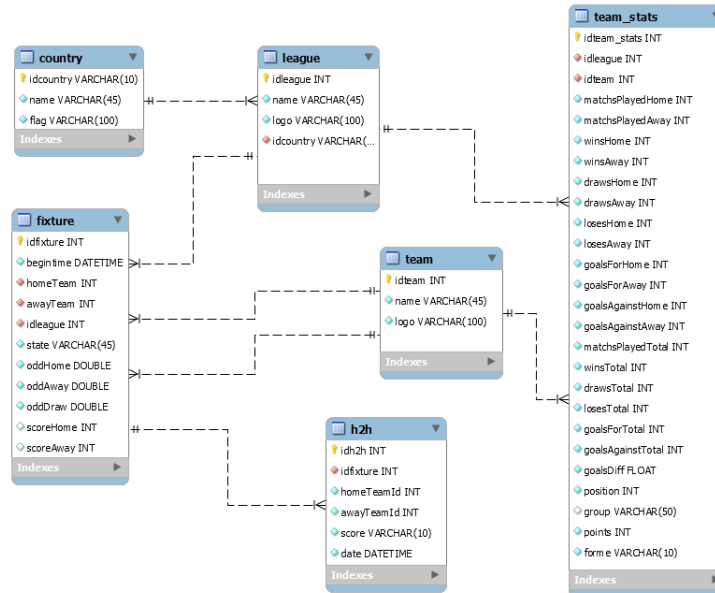


Figura 5.2: Base de dados da componente da aposta (Mysql).

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*.

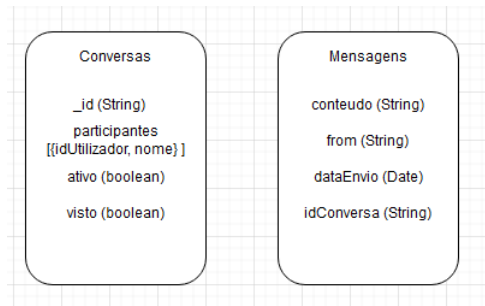


Figura 5.3: Base de dados da componente das conversas (MongoDB).

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.

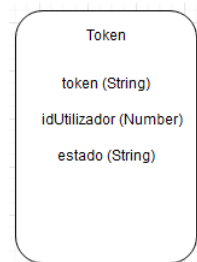


Figura 5.4: Base de dados da componente de autenticação (MongoDB).

Com o propósito de construir um sistema com *refreshToken*, estes foram guardados num documento muito simples que possui o identificador do utilizador e o atributo *token* indica o próprio *refreshToken*. Por fim, este documento contém também o atributo *estado*, que indica se o token encontra-se ativo ou não.

6

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.

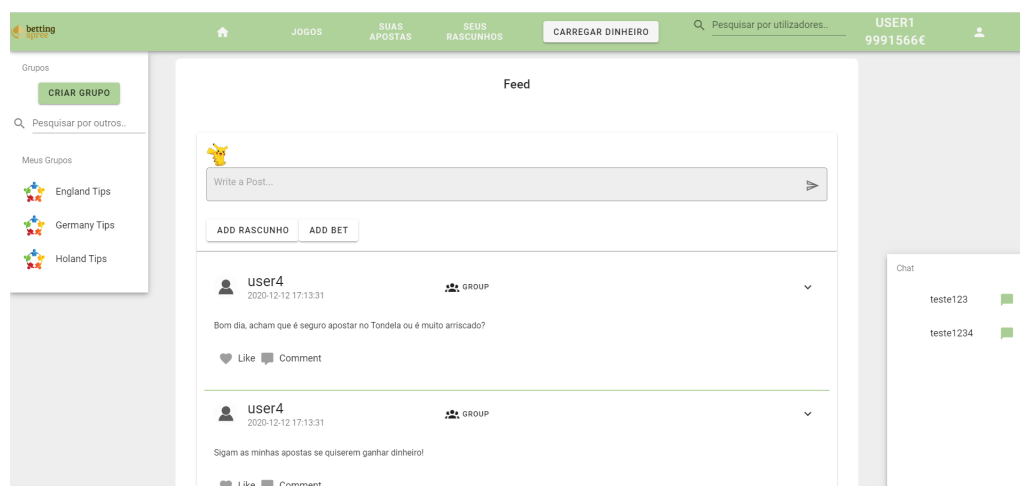


Figura 6.1: Página home do utilizador.

Relativamente à página das apostas realizadas o utilizador poderá consultar a lista de apostas que efectuou.

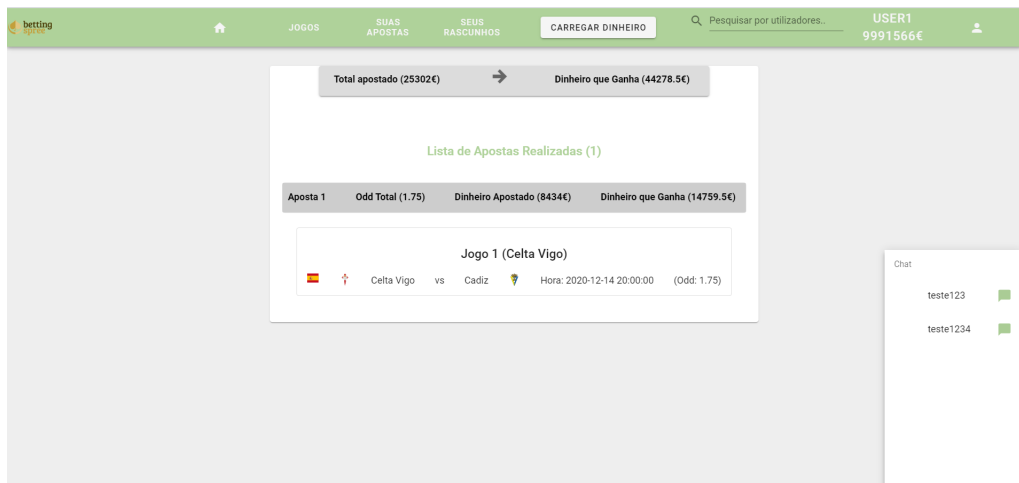


Figura 6.2: Página lista de apostas realizadas.

No que toca à página do utilizador, o *user* poderá examinar a lista das publicações que efectuou.

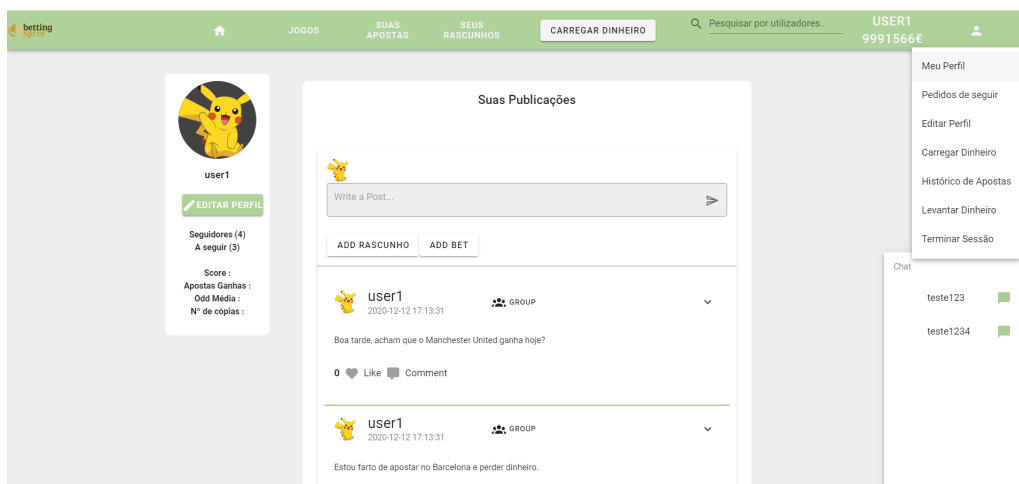


Figura 6.3: Página perfil do utilizador.

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. Nesta página também encontra-se o boletim de apostas, neste boletim é possível verificar a aposta atual do utilizador, pelo que este de seguida poderá escolher guardar o rascunho ou proceder com a aposta propriamente dita.

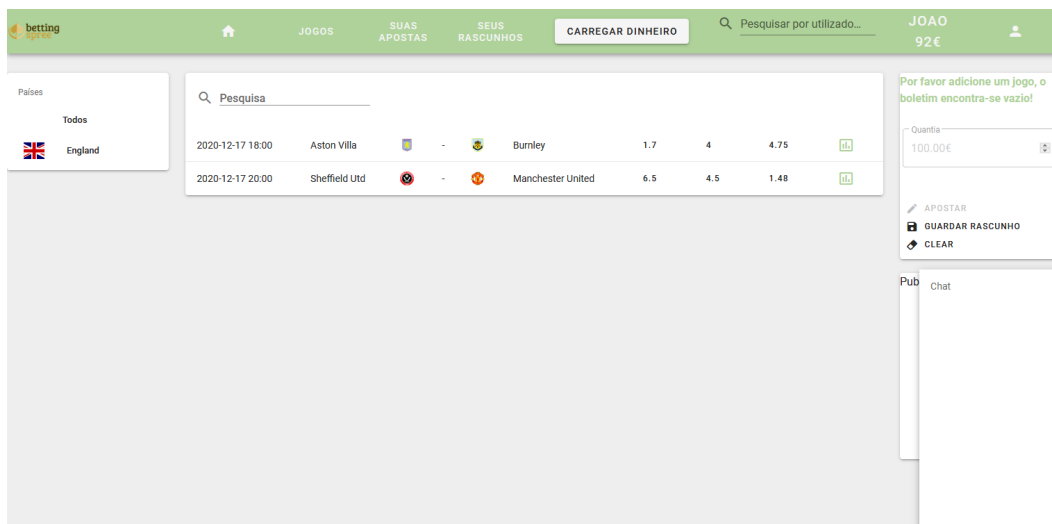


Figura 6.4: Página apostas desportivas.

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 Burnley

	Equipa	Posição	Pontos	Jogos Jogados	Vitórias	Empates	Derrotas	Golos Marcados
	Tottenham	1	25	12	7	4	1	24
	Liverpool	2	25	12	7	4	1	27
	Leicester	3	24	12	8	0	4	24
	Southampton	4	23	12	7	2	3	24
	Chelsea	5	22	12	6	4	2	25
	West Ham	6	20	12	6	2	4	20
	Everton	7	20	12	6	2	4	21
	Manchester	8	20	11	6	2	3	19

5 Jogos jogados em casa	5 Jogos jogados em casa
5 Jogos jogados fora de casa	6 Jogos jogados fora de casa
2 Vitórias em casa	1 Vitórias em casa
4 Vitórias fora	1 Vitórias fora
3 Derrotas casa	3 Derrotas casa
1 Derrotas fora	3 Derrotas fora
0 Empates fora	2 Empates fora
0 Empates casa	1 Empates casa

Figura 6.5: Dialog estatísticas da partida.

12 Golos marcados em casa	2 Golos marcados em casa
9 Golos marcados fora de casa	4 Golos marcados fora de casa
11 Golos sofridos em casa	6 Golos sofridos em casa
2 Golos sofridos fora	12 Golos sofridos fora
2.4 Média de golos marcados em casa	0.4 Média de golos marcados em casa
1.8 Média de golos marcados fora de casa	0.8 Média de golos marcados fora de casa
2.2 Média de golos sofridos em casa	1.2 Média de golos sofridos em casa
0.4 Média de golos sofridos fora de casa	2 Média de golos sofridos fora de casa
2.1 Média de golos marcados	0.55 Média de golos marcados
1.3 Média de golos sofridos total	1.64 Média de golos sofridos total

Data	Equipa Casa	Resultado	Equipa Fora
2020-01-01	Burnley	1-2	Aston Villa
2019-09-28	Aston Villa	2-2	Burnley
2015-05-24	Aston Villa	0-1	Burnley
2014-11-29	Burnley	1-1	Aston Villa

Figura 6.6: Continuação do dialog estatísticas da partida.

7

Lógica de Controlo

7.1 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.

7.2 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 seleccionam 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

Instalação e manutenção

Neste momento, a instalação ou *deploy* da plataforma não é uma prioridade, pois existem outras componentes, que o grupo de trabalho julga serem de maior importância, por atender. Contudo, se esta instalação for levada avante irá ser realizada em *containers docker* que serão posteriormente executados no servidor do Departamento de Informática da Universidade do Minho ou, alternativamente, na *cloud* pública.

Na prática irá ser 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.

9

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.