Universidade do Minho

Mestrado em Engenharia Informática

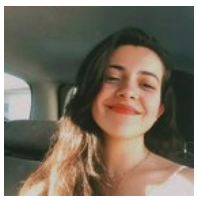
Requisitos e Arquiteturas de Software

Fase 2:

Solução Arquitetural

Grupo 1 – PL1


Ana Murta
pg50184


Ana Henriques
pg50196


Hugo Gomes
pg51242


João Lourenço
pg50464


Rui Armada
pg50737

dezembro, 2022

Conteúdo

Lista de Figuras

Capítulo 1

Introdução e Objetivos

No âmbito da Unidade Curricular de Requisitos e Arquiteturas de Software, foi solicitada a construção de uma aplicação de apostas em jogos de vários tipos de desportos.

Na primeira fase, foi atualizado um documento no qual foram descritos os vários requisitos funcionais e não funcionais, assim como o domínio do problema e o objetivo do sistema. Nesta segunda fase, o propósito é definir o domínio da solução, começando por revisar os requisitos funcionais propostos na fase anterior. Dentro das funcionalidades da plataforma, destacam-se as seguintes: registo e *login*; alterar perfil; consultar históricos de apostas, de transações e de notificações; entre outras.

Em relação aos requisitos não funcionais, foi tido em conta o desempenho do produto RASBET quando recorre à utlização da API externa, visando, como tal, desenvolver um sistema simples e rápido, capaz de suportar controlo de erros e a constante evolução da plataforma, através da implementação de uma base de dados. Para além disto, o sistema deve também poder ser acedido a partir de diferentes *browsers* e em diferentes sistemas operativos.

Outro detalhe ambicionado para esta fase foi a utilização de outras APIs, que suportasse outros tipos de desportos que a API fornecida pelos docentes não sustentassem.

Capítulo 2

Restrições

Para construir a plataforma de apostas proposta, o grupo escolheu recorrer à linguagem *Javascript*, e à sua *framework React*, para o desenvolvimento do *frontend* e do *backend*, bem como a linguagem *Sass* também para a componente *frontend* do *website*.

De modo a cumprir com um sistema intuitivo e de rápida aprendizagem de utilização, tal como prometido, o grupo tomou decisões de implementação que visassem a simplicidade de um modelo arquitetural, adaptável com a expansão da plataforma, quer a nível de utilizadores, quer a nível de desportos disponíveis. Qualquer tipo de modificação necessária também é exiquível sem grandes problemas devido, então, à simples estrutura da solução.

Devido a uma má organização do tempo, o grupo enfrentou algumas dificuldades em implementar todos os requisitos funcionais e não funcionais previamente definidos na primeira fase, nomeadamente a distinção entre as funcionalidades dos três tipos de utilizador que este sistema suposta. Como tal, a equipa optou por implementar algumas funcionalidades destinadas a cada tipo de utilizador, mas não estabeleceu a respetiva ligação com as restantes comuns a todos.

Capítulo 3

Contexto e Âmbito do Sistema

Assim como referido anteriormente, este sistema será desenvolvido projetando uma aplicação web, caracterizada por uma componente *backend* e uma componente *frontend*.

Na Figura 6.7, temos a arquitetura da componente *backend*, onde os ficheiros `*.controller.js` estão ligados aos `*.routes.js` que, por sua vez, estão ligados ao `index.js`. Os ficheiros `*.routes.js` são responsáveis por trabalhar as rotas para cada página da plataforma; enquanto que os ficheiros `*.controller.js` objetivam a construção da lógica de negócio do sistema. O *backend* também é composto por uma base de dados, desenvolvida a partir do *MySQL*.

Relativamente ao *frontend*, e como indicado no Capítulo 1, foram criadas as páginas responsáveis por apresentar o registo e login dos utilizadores, as informações de perfil, o histórico de transações e o histórico de apostas, os desportos disponíveis, a pesquisa de eventos desportivos, o boletim de apostas, o painel de notificações e, ainda, as promoções oferecidas. Outra funcionalidades destinadas a apenas um determinado tipo de utilizador, como a adição de um evento desportivo, a alteração de uma odd, entre outras, também foram implementadas; no entanto, não possuem ainda uma ligação com o resto do website visto não ter sido ainda programada a hierarquia de utilizadores nesta fase do projeto.

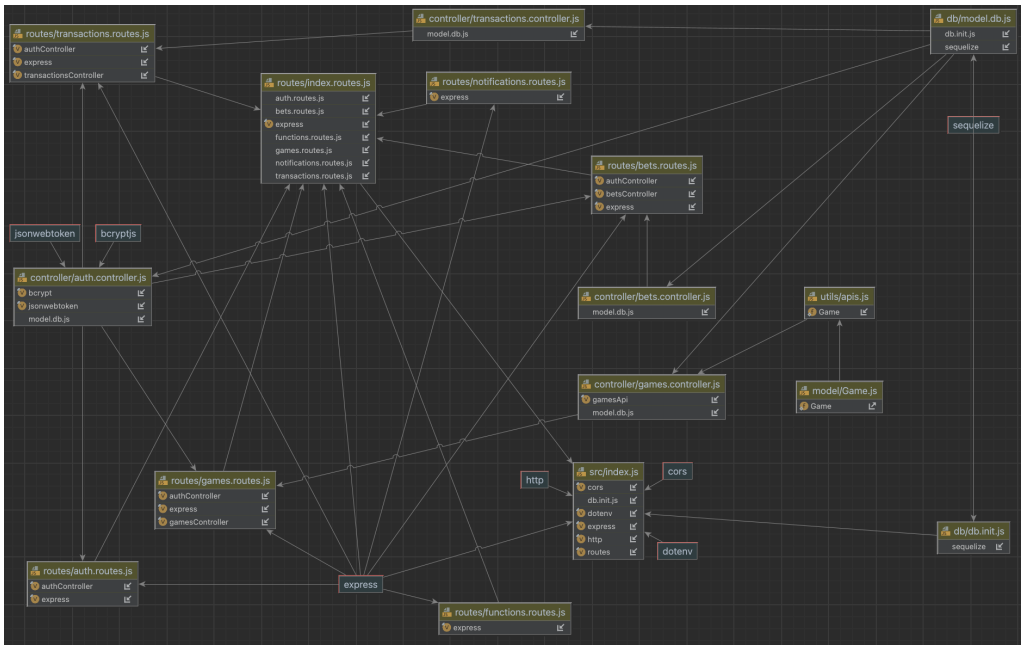


Figura 3.1: Diagrama de classes referente à componente *backend*

Capítulo 4

Estratégia da Solução

Assumindo como principais decisões no decorrer da conceção da aplicação, realçam-se a escolha das tecnologias a utilizar bem como a definição da arquitetura da aplicação.

Tendo em consideração todas as metas de qualidade ambicionadas para o produto final, foi necessário pensar no que é que a aplicação careceria a fim de atingir tais metas e proporcionar a melhor experiência ao utilizador.

Dito isto, as seguintes secções apresentam as propostas de soluções, encontradas durante o desenvolvimento da aplicação, para os problemas expostos pela qualidade da aplicação.

4.1 Segurança

Cenário: O utilizador deverá possuir os seus dados guardados de forma segura.

Solução proposta: Desenvolver uma estratégia de encriptação dos dados do utilizador.

4.2 Operabilidade

Cenário: O utilizador deverá utilizar a aplicação com relativa facilidade.

Solução proposta: Investir no desenvolvimento uma inferface amigável e de fácil utilização.

4.3 Desempenho

Cenário: O produto deve ser capaz de responder de forma rápida e eficaz.

Solução proposta: Desenvolvimento de estratégias capazes de prevenir algum *delay* provocado pelas *APIs* através do armazenamento dos jogos na base de dados.

4.4 Adequação funcional

Cenário: Cada *stakeholder* da aplicação poderá sugerir novas funcionalidades.

Solução proposta: Marcação de reuniões regulares entre a equipa de desenvolvimento e os *stakeholders* de forma a garantir a implementação de todas as funcionalidades desejadas.

4.5 Manutenção

Cenário: A aplicação deve ser capaz de armazenar dados novos de utilizadores e das *APIs*.

Solução proposta: Desenvolver a aplicação de modo a facilitar a sua manutenção e alteração, especialmente na base de dados.

Capítulo 5

Building Block View

Relativamente à arquitetura do sistema, a equipa de trabalho começou por identificar quais os potenciais subsistemas, agrupando adquadamente, em cada um, os ficheiros desenvolvidos. Isto permite conseguir uma melhor organização.

Na Figura 5.1, podemos ver o diagrama de componentes elaborado para nível 1 do sistema, que nos permite então analisar como é que o sistema está repartido em subsistemas e a dependência entre os mesmos.

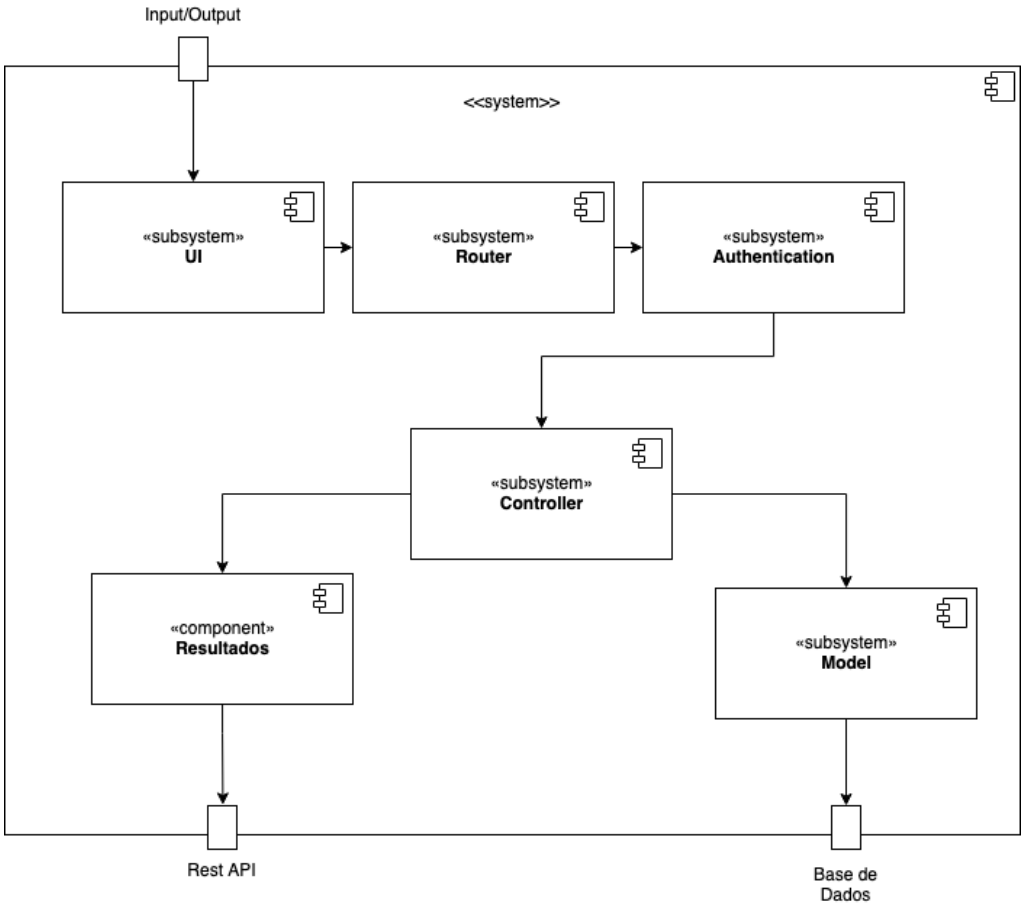


Figura 5.1: Diagrama de Componentes

5.1 UI

Este subsistema engloba frontend do cliente aplicação e como este comunica com o backend através de pedidos http.

5.2 Router

Este subsistema consiste no redirecionamento dos pedidos http para o controller correto.

5.3 Authentication

O subsistema de authentication age como middleware entre o Router e o Controller, verificando se o cliente que efetuou o pedido está ou não autenticado no sistema e se tem ou não permissões para fazer o pedido.

5.4 Controller

Este susbsistema é responsável por executar toda a lógica dos pedidos http, desde a comunicação com a base de dados, até do fornecimento de uma resposta ao cliente.

5.5 Model

O Model é o subsistema que permite toda a interação com a base de dados, desde a criação da mesma até à sua manipulação.

5.6 Resultados

Os resultados são os dados gerados pelo Controller e que vão ser enviados como resposta ao cliente.

Capítulo 6

Runtime View

6.1 Registar Utilizador

No Capítulo 2, mencionou-se que não foi cumprida a hierarquia de funcionalidades respetivas a cada tipo de utilizador tal como foi proposto na primeira fase. Por motivos de má gestão de tempo, o grupo desenvolveu o registo de um utilizador sem diferenciar ainda se ele é apostador, especialista ou administrador, dependendo dos dados impostos para validar cada um dos tipos.

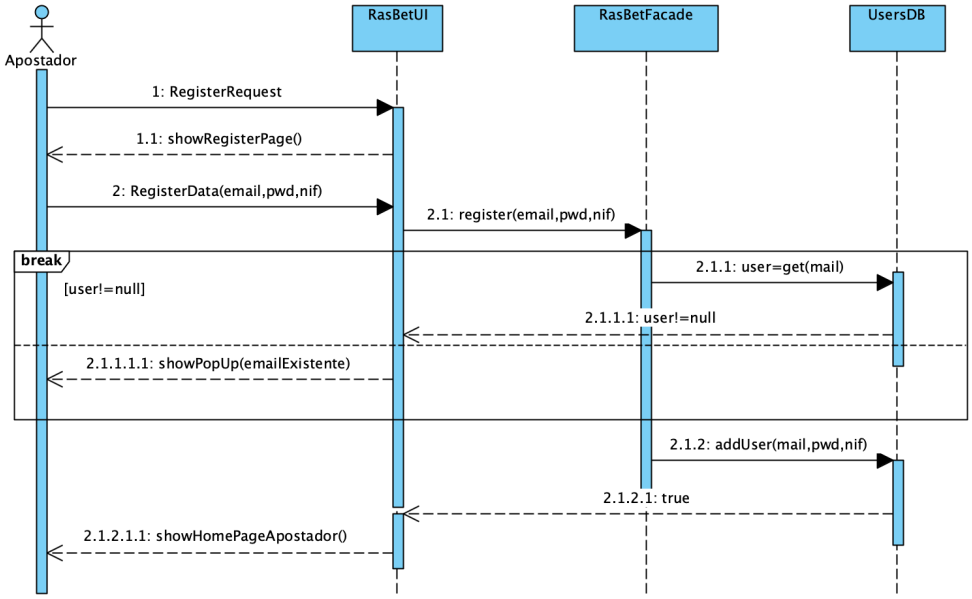


Figura 6.1: Diagrama de Sequência de Registar Utilizador

6.2 *Login* Utilizador

Considerando novamente o incumprimento da diferenciação do tipo de utilizador, esta funcionalidade foi concebida, fornecendo apenas os dados que o utilizador precisa para poder entrar na aplicação. No entanto, como as informações requeridas dizem respeito apenas às do apostador, ao aceder à plataforma, o utilizador só terá acesso às funcionalidades de um apostador.

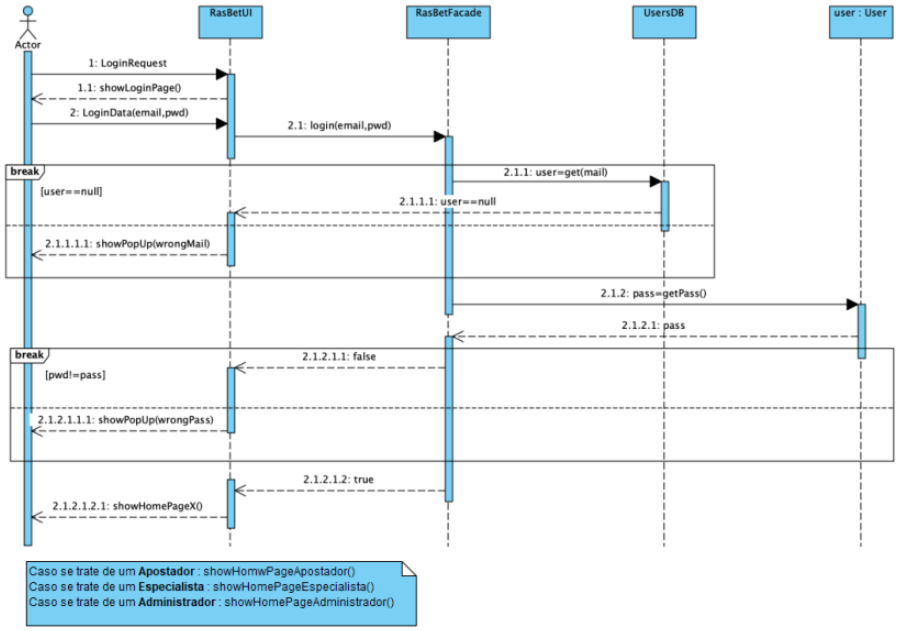


Figura 6.2: Diagrama de Sequência de *Login* Utilizador

6.3 Consultar Jogos Desportivos

O utilizador pode consultar os vários jogos disponíveis para futebol e basquetebol, selecionando primeiramente o tipo de desporto em questão e, posteriormente, pode até fazer uma pesquisa para chegar ao resultado pretendido mais rapidamente. Relativamente aos outros desportos, como ténis e motoGP, ainda não existem jogos disponíveis.

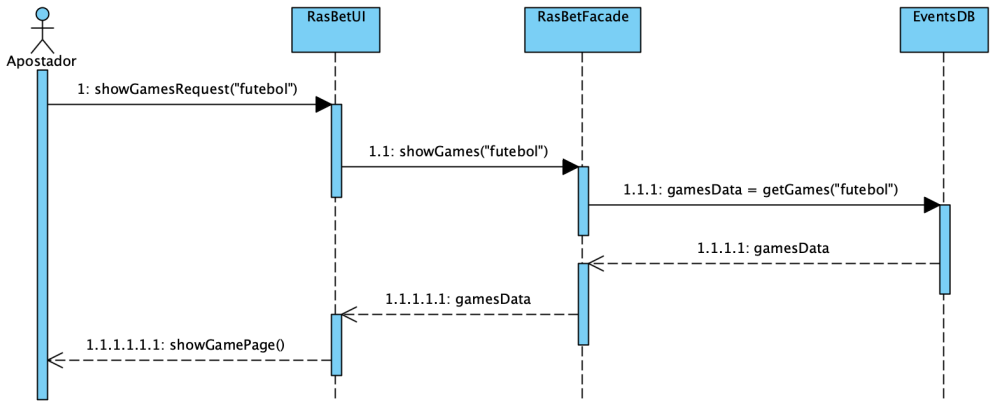


Figura 6.3: Diagrama de Sequência de Consultar Jogos

6.4 Fazer Aposta

O utilizador pode efetuar uma aposta simples, múltipla ou combinada, identificando em primeiro lugar o tipo da aposta. Seguidamente, é preciso escolher o(s) jogo(s) desportivo(s) no(s) qual/ais quer apostar e o valor da *odd*. Para terminar o processo, tem de pagar, optando por um meio de pagamento. A aposta será finalizada assim que o pagamento for validado.

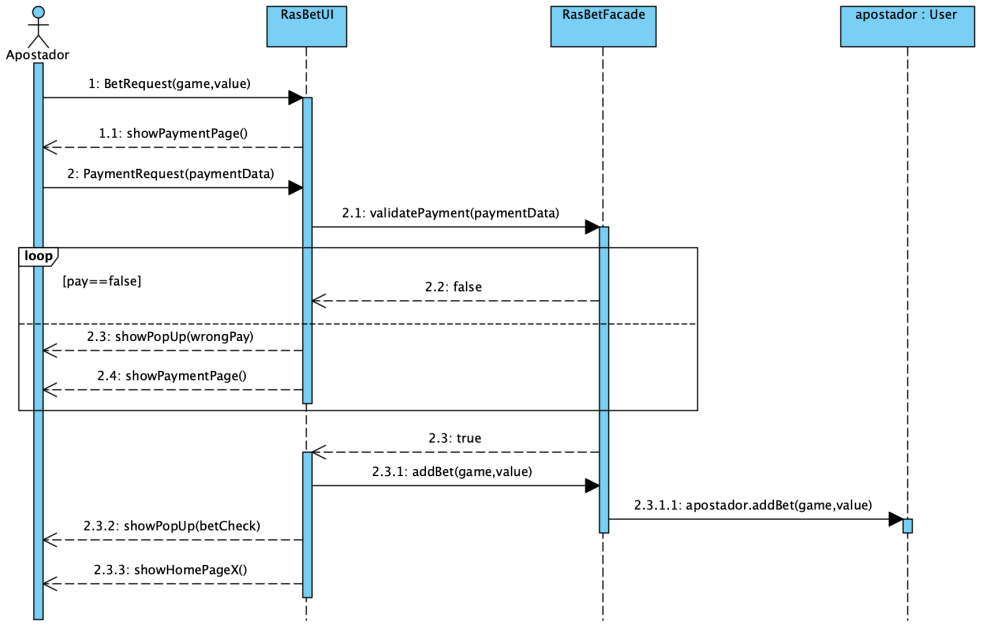


Figura 6.4: Diagrama de Sequência de Fazer Aposta

6.5 Inserir *Odd*

A funcionalidade de inserir uma odd (e também alterá-la) pertence a um especialista. Todavia, e como igualmente mencionado no Capítulo 2, embora não tenham sido estabelecida a ligação com o resto do *website*, existem algumas funcionalidades destinadas a especialistas e administradores que foram desenvolvidas soltamente, é o caso desta. Para inserir, então, uma *odd*, o utilizador precisa de selecionar o desporto, o jogo e o valor.

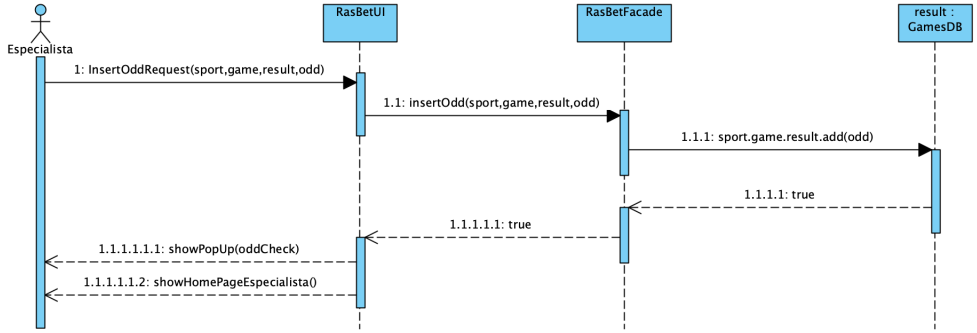


Figura 6.5: Diagrama de Sequência de Inserir *Odd*

6.6 Consultar Histórico de Apostas/Transações

Para consultar tanto o histórico de apostas, como o histórico de apostas, o utilizador precisa de apenas de aceder ao conjunto de funcionalidades no botão identificativo do seu perfil. O processo de consultar o painel de notificações, é exatamente igual.

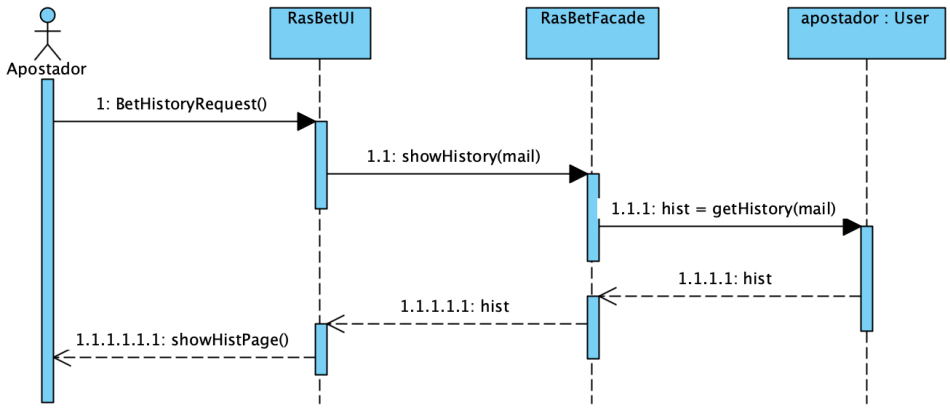


Figura 6.6: Diagrama de Sequência de Consultar Apostas

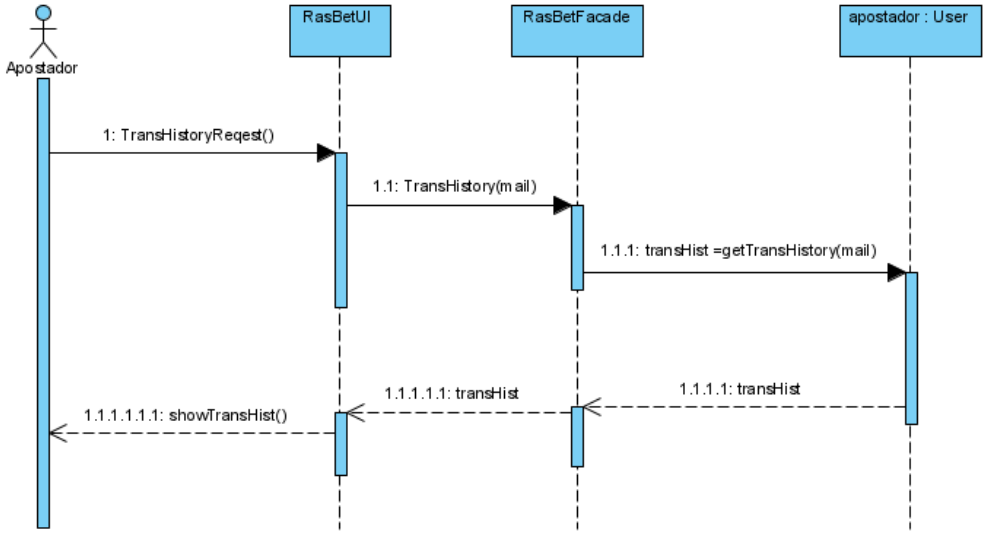


Figura 6.7: Diagrama de Sequência de Consultar Transações

Capítulo 7

Deployment View

Com o objetivo de ilustrar a implementação da melhor forma possível, foi desenvolvido o diagrama apresentado na Figura 7.1, que descreve a maneira como a aplicação foi implementada para satisfazer os requistos levantados.

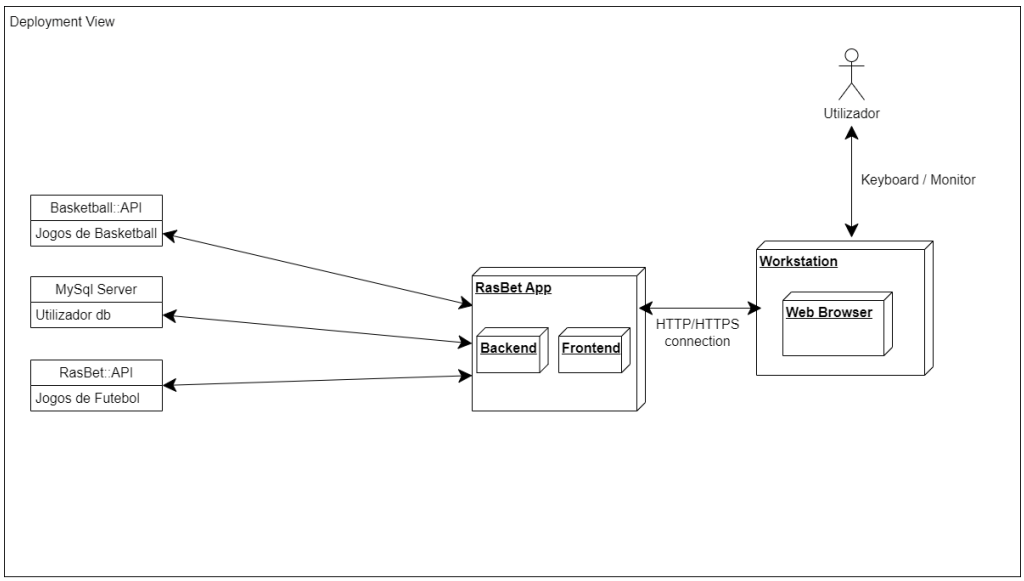


Figura 7.1: Diagrama ilustrativo do *Deployment View*

Capítulo 8

Conclusões e Trabalho Futuro

Nesta segunda fase do trabalho prático, foi possível elaborar o planeamento e implementação da arquitetura da aplicação a desenvolver, bem como a sua estruturação e objetivos. Adicionalmente, foi necessário definir estratégias importantes que tornassem a experiência do utilizador mais segura e sem muitos *delays*, melhorando, consequentemente, bastante a qualidade da aplicação.

Para além disto, ainda foi prestado especial cuidado no cumprimento dos requisitos levantados na fase anterior, assim como à concordância e consistência entre a primeira e a segunda fase, pretendendo-se manter o mesmo rigor para a fase final deste projeto.

Deste modo, foi desenvolvida uma solução satisfatória para os problemas e requisitos anteriormente propostos. Numa fase final, ambiciona-se concluir a implementação dos requisitos em falta, tal como aperfeiçoar algumas questões, se for necessário, construídas na presente fase do projeto.