

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA
MESTRADO EM ENGENHARIA INFORMÁTICA

Requisitos e Arquiteturas de Software

Trabalho prático - RASBet
Turno PL6
2021/2022

2 de maio de 2022



Ariana Lousada
(PG47034)



Carlos Gomes
(PG47083)



Márcia Teixeira
(A80943)



Tiago Sousa
(PG47684)

Conteúdo

1	Introdução e Objetivos	4
1.1	Visão geral dos Requisitos	5
1.2	Metas de qualidade	5
1.3	<i>Stakeholders</i>	6
2	Restrições da arquitetura	8
3	Âmbito e contexto	10
3.1	Contexto empresarial	11
3.2	Contexto técnico	12
4	Estratégia da Solução	13
5	<i>Building block view</i>	14
5.1	Subsistema RASBet UI	15
5.1.1	Mockup do RASBet UI	16
5.2	Subsistema Login	16
5.3	Componente Application	17
5.4	<i>Whitebox Overall System</i>	17
5.4.1	IUtilizador	18
5.4.2	Jogo	19
5.4.3	Aposta	20
5.4.4	Boletim	20
5.4.5	HistoricoApostas	21
5.4.6	Desporto	21
5.4.7	Equipa_Indivíduo	21
6	<i>Runtime View</i>	23
6.1	Criar Aposta	23
6.2	Visualizar o histórico de apostas	24
6.3	Consultar estatísticas	25
6.4	Efetuar pagamento	26
6.5	Receber notificação de jogo	27
7	Crosscutting Concepts	29
7.1	Dependências	29
7.2	Domínio do Projeto	30

8	Decisões de Arquitetura	31
9	Qualidades	32
9.1	Árvore de Qualidades	32
9.2	Cenários de Qualidade	33
10	Riscos e Custos	34
10.1	Risco: A base de dados atinge a sua capacidade máxima	34
10.1.1	Minimização	34
10.2	Risco: API indisponível	35
10.3	Minimização	35
10.4	Risco: Serviço de pagamento indisponível	36
10.5	Minimização	36
11	Conclusões e Trabalho Futuro	37

Capítulo 1

Introdução e Objetivos

O principal objetivo estabelecido no documento de requisitos do projeto consiste no desenvolvimento de uma aplicação de apostas em desporto de amigável e intuitiva utilização aos apostadores.

Ao longo deste documento vão ser abordados diversos parâmetros também importantes para o desenvolvimento da aplicação, como a sua arquitetura, implementação e riscos possíveis e consequentes custos.

1.1 Visão geral dos Requisitos

Aquando do processo de desenvolvimento da primeira fase correspondente ao projeto em mãos, foi concebida de uma panóplia de requisitos funcionais bem como de não funcionais, que se encontram enunciados respetivamente nos capítulos 6 e 7 do documento de requisitos.

Após uma análise focada e detalhada de cada requisito não funcional formulado, tornou-se possível efetuar a escolha daqueles que melhor representarão o funcionamento interno da aplicação a desenvolver em fases seguintes.

A seguinte tabela compõe os três requisitos não funcionais selecionados, resultantes da ponderação, estudo e pesquisa por parte dos elementos integrantes da equipa de desenvolvimento.

Descrição dos Requisitos		
Número do requisito	Tipo de requisito	Descrição
26	Operacional	O sistema deverá ser capaz de aceder a fontes de informação acerca das equipas/indivíduos participantes em jogos disponíveis para aposta.
29	Requisito de manutenção	Deverá existir a tempo inteiro uma equipa de técnicos de informática para lidar com potenciais falhas no sistema.
18	Requisito de utilização	A aplicação deverá ter a capacidade de tratar input potencialmente incorreto proveniente do utilizador.

O fundamento que precede a seleção dos requisitos enunciados anteriormente encontra-se descrito na secção "Prefácio" do documento de requisitos.

1.2 Metas de qualidade

O produto final, que será proveniente do esforço contínuo e iterativo apresentado na documentação em mãos, tem como meta a sua inserção e posterior sucesso no mercado. Para que sejam garantidas estas premissas, torna-se fulcral o estabelecimento de metas que devem ser impostas aos *stakeholders* envolvidos no projeto. Estas metas serão dotadas de diferentes prioridades, relacionadas com a sua importância no contexto do projeto. Estas prioridades estão definidas de 1 a 5 em que 1 corresponde a "Pouco importante" e 5 corresponde a "Muito importante".

Metas de qualidade			
Qualidade	Prioridade	Cenário	<i>Stakeholders</i> associados
Segurança	5	A base de dados irá possuir credenciais e informações pessoais acerca dos utilizadores. Apenas o próprio utilizador deverá ter acesso a estes dados.	Utilizadores, Técnicos de informática, Instituição de proteção de dados
Adequação Funcional	4	É importante que a aplicação possua todas as funcionalidades necessárias e requisitadas por cada <i>stakeholder</i>	Instituições monetárias, Instituição de proteção de dados, Investidores
Operabilidade	3	Um utilizador deverá conseguir usar o produto final com relativa facilidade e rapidez.	Analistas de projeto, Gestores do projeto, Técnicos de informática, Utilizadores
Manutenção	2	A aplicação irá ser sujeita a uma manutenção constante, para garantir o seu correto funcionamento.	Técnicos de informática, Gestores do projeto
Desempenho	1	A aplicação, juntamente com uma manutenção constante, deverá ter um desempenho também aproximadamente constante e possibilitar a sua utilização a vários utilizadores simultaneamente.	Utilizadores, Técnicos de informática

1.3 *Stakeholders*

Tal como já se encontra enunciado no documento relativo aos requisitos, os *stakeholders* tomam parte do desenvolvimento do projeto como sendo de cariz fundamental e estruturante para o mesmo na medida em que se impõe como medida decisiva em termos de elaboração do produto. Isto deve-se ao facto de que se tratam das entidades que irão ser influenciadas pelo projeto.

É crucial que cada um dos *stakeholders* tenha completo conhecimento da arquitetura a desenvolver e da documentação desenvolvida para o mesmo propósito, uma vez que possuirão controlo

sobre as decisões a tomar sobre o futuro da mesma.

<i>Stakeholders</i>	
Papel	Expectativas
Instituição de proteção de dados	A aplicação não terá qualquer falha de segurança que comprometa os utilizadores assim como a sua informação.
Instituições monetárias	A aplicação deverá ter sucesso no mercado, de modo a ser mais utilizada e consequentemente aumentar o fluxo de valores monetários.
Utilizadores	O produto final deverá ser de fácil utilização e confiável(visto que dados pessoais estarão envolvidos).
Investidores	A aplicação deverá ter sucesso elevado no mercado, para o investimento efetuado no produto compensar.
Gestores do projeto	A aplicação irá ser exposta a uma gestão constante, a um nível apropriado à sua popularidade no mercado.
Analistas de projeto	O produto final irá influenciar o mercado no qual se encontrará inserido, assim como os seus utilizadores. Esta influência poderá ser estudada e analisada num futuro próximo.
Técnicos de informática	Será necessário uma manutenção constante da aplicação, para resolver potenciais falhas e garantir o seu funcionamento.

A secção 3.2.3. pertinente no documento de requisitos possui uma descrição e explicação mais minuciosa do papel associado a cada *stakeholder*.

Capítulo 2

Restrições da arquitetura

Tratando-se o presente projeto de um processo iterativo de desenvolvimento de todas as partes que constituem como um todo o caminho traçado desde a idealização até à conceção do produto, este terá associadas, para além de todas as funcionalidades, restrições adjacentes do seu planeamento e decisões tomadas em prol de um resultado agradável e com as capacidades perspectivadas. Na primeira fase do desenvolvimento em mãos, essas restrições, que se apresentam como sendo as principais da aplicação, encontram-se expostas em detalhe no Capítulo 5 do documento de requisitos.

Apesar da existência destas restrições que terão um cariz de maior importância no sistema, é imperativo definir o quanto antes restrições de outros tipos que virão a ser necessárias.

<i>Restrições</i>		
Restrição	Tipo de Restrição	Descrição
Base de dados	Técnica	Será necessária uma base de dados própria para a aplicação, para que seja possível a vigilância e salvaguarda dos dados dos utilizadores.
API	Técnica	A aplicação irá extrair dados sobre os jogos desportivos de uma dada API para auxiliar as suas funcionalidades.
Tipos de desporto	Técnica	A aplicação irá permitir que o utilizador aposte em jogos de futebol, golfe e basquetebol.
Métodos de pagamento	Técnica	A aplicação deverá disponibilizar vários tipos de pagamento ao utilizador.
Políticas	Política	A aplicação não deverá mostrar qualquer preferência a qualquer indivíduo participante nos jogos. Não deverá favorecer determinados utilizadores.
Estrutura	Técnica	A aplicação deverá apenas conter os módulos necessários para o seu correto funcionamento e auxiliares das suas funcionalidades.

As restrições que se encontram retratadas em primeiro e segundo lugar na tabela anterior, inserem um custo acrescido ao sistema. No entanto, este custo adicional, é compensado pelo benefício em que estas restrições se traduzem para o mesmo no futuro: a existência de uma base de dados própria permite uma gestão mais segura e rigorosa das componentes, dado que a estratégia de logística e proteção de dados será desenvolvida pela equipa de técnicos de informática e de

gestores do produto. Por sua vez, a utilização de uma API, economiza no espaço utilizado pela base de dados. Isto é garantido pelo facto de que apenas serão extraídos os dados estritamente necessários para garantir o correto funcionamento das funcionalidades da aplicação.

Capítulo 3

Âmbito e contexto

Tal como foi exposto no documento de requisitos, nomeadamente no capítulo 3, o produto em desenvolvimento tem como propósito final a disponibilização de um serviço de apostas *online*.

Tratando-se de um serviço disponibilizado via *online*, que consequentemente promete a capacidade de se encontrar disponível e prestável para vários utilizadores em simultâneo, torna-se necessário investir em atributos tais como a disponibilidade constante, de forma a evitar falhas no funcionamento resultantes da incapacidade do sistema em termos de escalabilidade.

Na primeira fase de desenvolvimento do projeto em mãos, foi constituído o diagrama apresentado abaixo que permite explicitar as interações da aplicação a desenvolver com as entidades externas. No caso prático em perspetiva, a aplicação terá apenas interação com a API, a Base de Dados e o serviço responsável pelo pagamento.

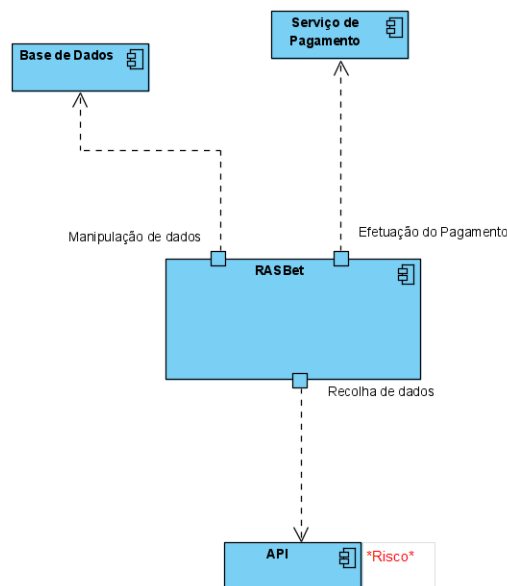


Figura 3.1: Diagrama de Contexto

Na tabela seguinte podem-se observar as funcionalidades de cada componente do diagrama de contexto.

Componente	Descrição
Base de Dados	Contém toda a informação relativa aos utilizadores da aplicação como credenciais e possivelmente dados de pagamento.
API	Aplicação a partir da qual se obtêm dados como estatísticas de equipas/individuos de um determinado desporto, assim como os resultados dos jogos.
Serviço de pagamento	Serviço responsável pela conclusão de pagamentos tanto das apostas como para os utilizadores.

Apesar de existirem possíveis riscos de segurança na interação com a base de dados, estes podem ser facilmente controlados através da formulação de uma boa estratégia de segurança com auxílio a encriptação. Contudo, existe sempre um risco de disponibilidade da API que foge do controlo da aplicação. Caso a API receba muitos pedidos simultaneamente, esta pode criar uma espécie de fila de espera, o que por sua vez poderá atrasar a recolha de dados por parte da aplicação.

3.1 Contexto empresarial

A aplicação em desenvolvimento poderá ser resumida numa interação entre utilizador e sistema. O sistema por sua vez interage com a sua base de dados, API e serviço de pagamento.

De modo a ser possível interagir com a API, será necessária a escolha de um protocolo do tipo HTTPS ou SSH.

Para a base de dados, possivelmente irá-se utilizar MySQL, construindo ligações necessárias para interação com o sistema.

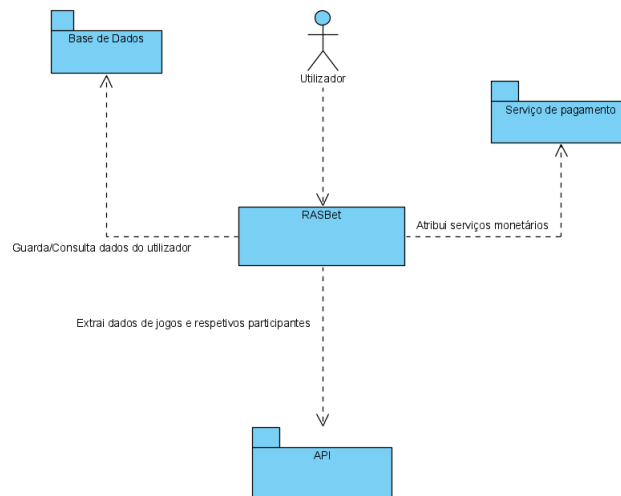


Figura 3.2: Interações entre as várias entidades externas ao sistema.

3.2 Contexto técnico

Após várias pesquisas e análise das tecnologias mais pertinentes e utilizadas hoje em dia, tendo também em conta a sua eficiência, foi decidido desenvolver a aplicação na linguagem de programação Elixir, visto que contém várias frameworks que poderão ser utilizadas no desenvolvimento do produto, como por exemplo Phoenix.

Em termos de protocolos, poderá ser utilizado HTTPoison para estabelecer a conexão para a aplicação, apesar de ainda estar em discussão entre os membros da equipa de trabalho.

A escolha da API será feita após uma pesquisa extensiva de modo a selecionar a mais eficiente possível.

Capítulo 4

Estratégia da Solução

As principais decisões durante o desenho da solução do projeto consistiram na escolha das tecnologias a serem utilizadas assim como a arquitetura da aplicação em si.

Tendo em conta as várias metas de qualidade que se pretendem atingir com o produto final foi necessário pensar no que a aplicação iria necessitar de modo a ser possível atingir todos os objetivos.

Com isto, na tabela seguinte foram apresentadas algumas propostas de soluções para os problemas expostos pela qualidade do projeto.

Assim, com referência às metas de qualidade expostas na secção 1.2:

Metas de qualidade - Soluções Propostas		
Qualidade	Cenário	Solução proposta
Segurança	Um utilizador deverá ter os seus dados guardados na aplicação de uma forma segura.	Desenvolver uma estratégia de segurança de dados, possivelmente recorrendo a alguma técnica de encriptação.
Adequação Funcional	Cada <i>stakeholder</i> envolvido no projeto poderá sugerir diferentes funcionalidades na aplicação, consoante o seu papel.	Marcação de reuniões regulares com os <i>stakeholders</i> , de modo a garantir que o produto final contém todas as funcionalidades desejadas.
Operabilidade	O utilizador irá usar a aplicação com relativa facilidade.	Investir na construção de uma interface amigável e de fácil utilização, através de pesquisas de vários designs utilizados atualmente.
Manutenção	A aplicação tem de ter a capacidade de armazenar sempre dados de utilizadores novos.	Desenvolver a aplicação de modo a facilitar a sua manutenção e alteração, principalmente na base de dados.
Desempenho	O produto irá estar constantemente disponível para vários utilizadores simultaneamente.	Utilização de protocolos recentes e eficientes, que possuam a capacidade de providenciar serviço a muitos clientes simultaneamente.

Capítulo 5

Building block view

Para uma visualização *top-down* da aplicação, desenvolveu-se o seguinte diagrama representante do nível 1 da *Building Block View* (que segue o contexto e âmbito, representado na figura 3.1).

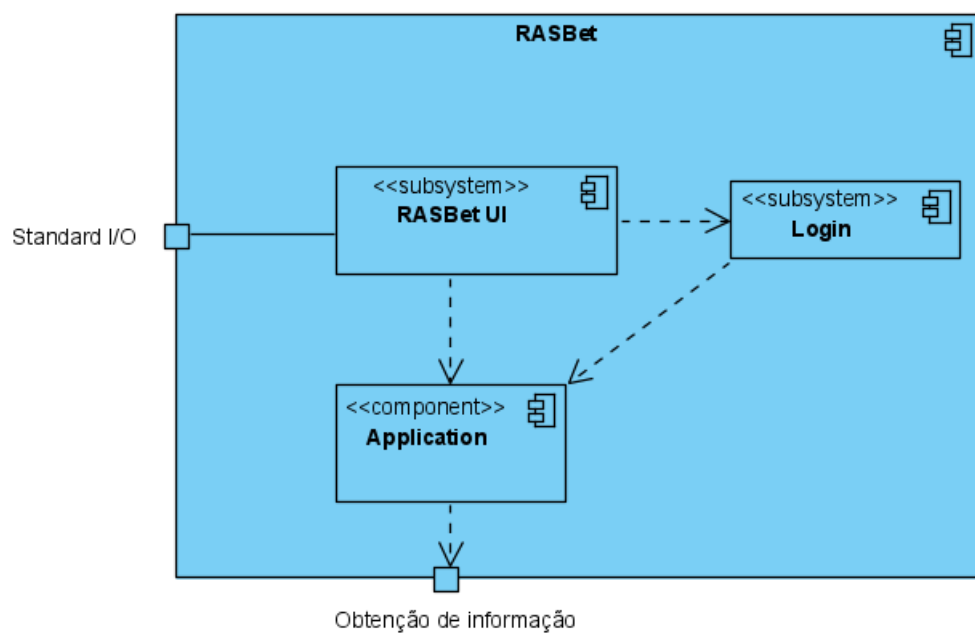


Figura 5.1: Building Block View - Nível 1

Componentes e Subsistemas	
Componente	Descrição
RASBet UI	Menu Inicial. Primeira interface da aplicação com a qual o utilizador tem o primeiro contacto com o produto.
Login	Secção na qual o utilizador poderá fazer login.
Application	Contém todos os subsistemas constituintes do produto. Pode ser acedida com login(utilizador registado) ou sem login(convidado).

5.1 Subsistema RASBet UI

Este subsistema é responsável por permitir a escolha de login ou seleção da aplicação como convidado.

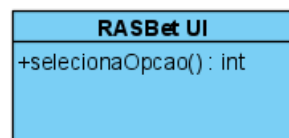


Figura 5.2: RASBet UI

Esta classe vai apenas receber uma das duas opções possíveis selecionada pelo utilizador(se pretende efetuar login ou não). A partir do resultado obtido, o utilizador será redirecionado para outro menu.

5.1.1 Mockup do RASBet UI

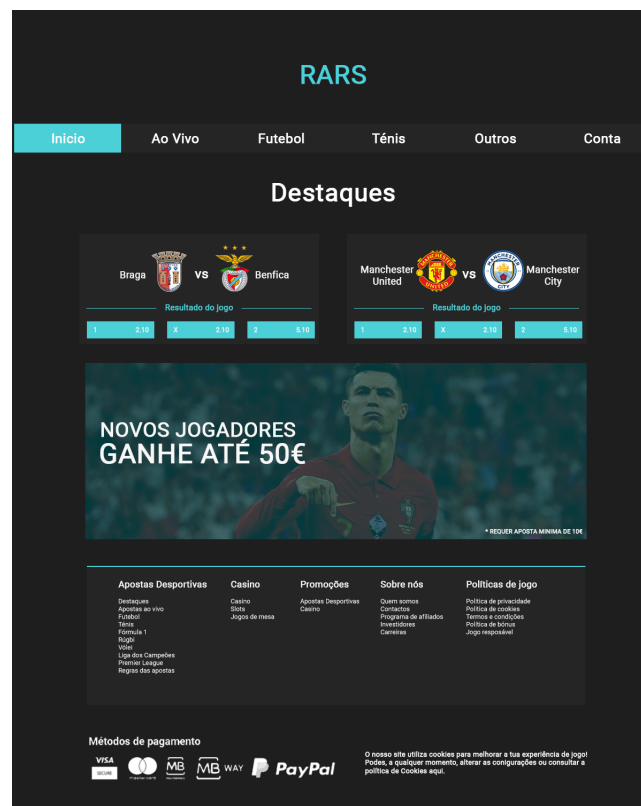


Figura 5.3: Página Inicial

5.2 Subsistema Login

Este subsistema é responsável pelo login do utilizador.¹

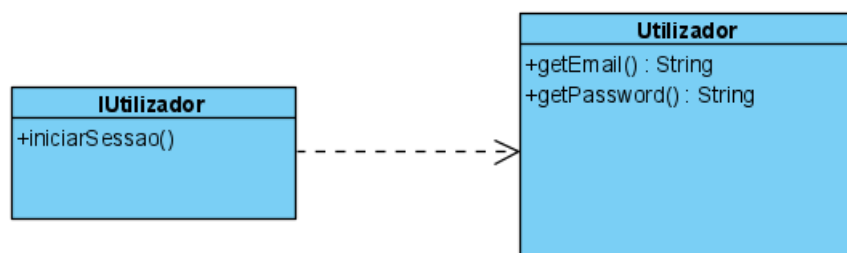


Figura 5.4: Subsistema Login

¹É de notar que cada uma das classes representadas contém mais métodos e variáveis. Não foram apresentados com o intuito de facilitar a observação dos métodos relevantes para esta funcionalidade.

O método `iniciarSessao` irá utilizar os métodos `getEmail` e `getPassword` para obter as credenciais necessárias.

5.3 Componente Application

Este componente é a base do produto. Contém todos os componentes necessários para as várias funcionalidades do sistema.

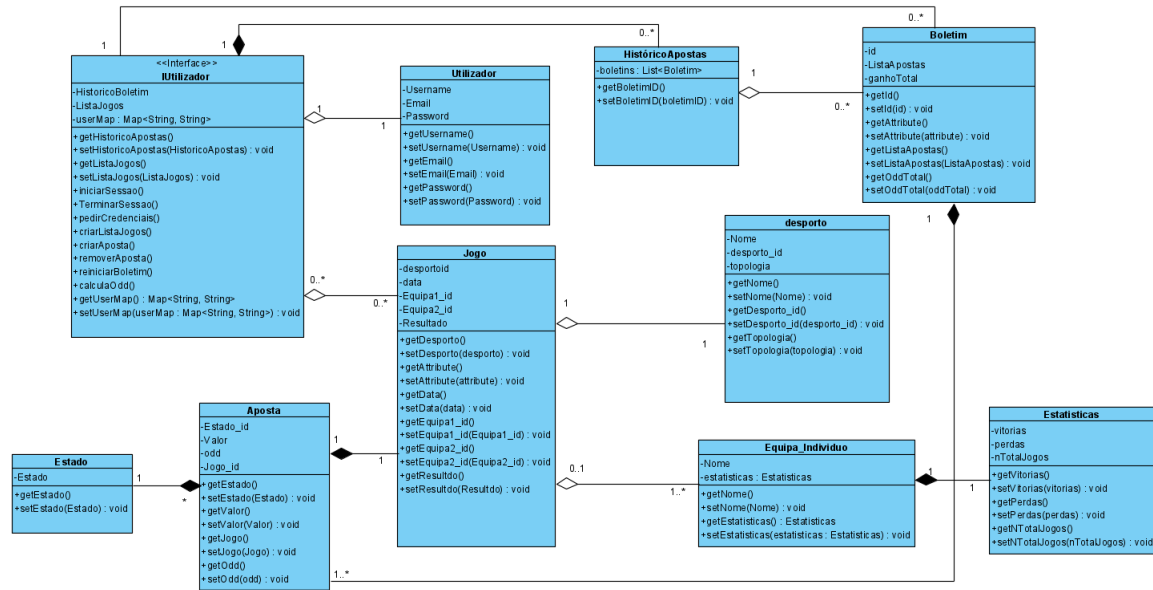


Figura 5.5: Application - Diagrama de Classes

Cada classe do diagrama refere uma entidade na aplicação, cujo papel será exposto na secção seguinte deste capítulo.

5.4 Whitebox Overall System

Visto que os subsistemas RASBet UI e Login são simples, nesta secção irão ser expostos os métodos e conteúdos do componente *Application* presentes na figura 5.5.

Tal como já foi referido anteriormente, a aplicação irá extrair dados de uma dada API e consultar/armazenar dados própria numa base de dados.

O componente responsável pela inicialização da conexão entre estes componentes externos ao sistema é precisamente a *Application*.

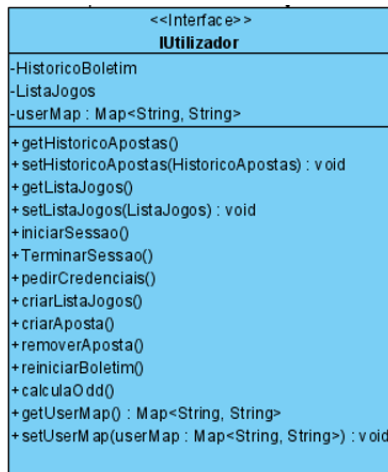


Figura 5.7: Interface do utilizador.

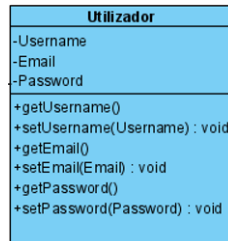


Figura 5.8: Utilizador e respetivos métodos

5.4.2 Jogo

Este componente contém todas as informações relacionadas com cada jogo, como data, equipas/indivíduos participantes e resultado.

Jogo
-desportoid
-data
-Equipa1_id
-Equipa2_id
-Resultado
+getDesporto()
+setDesporto(desporto) : void
+getAttribute()
+setAttribute(attribute) : void
+getData()
+setData(data) : void
+getEquipa1_id()
+setEquipa1_id(Equipa1_id) : void
+getEquipa2_id()
+setEquipa2_id(Equipa2_id) : void
+getResultdo()
+setResultado(Resultdo) : void

Figura 5.9: Jogo e respetivos métodos

5.4.3 Aposta

Este componente contém todas as informações necessárias para caracterizar uma aposta, como estado('Aberta' (disponível para aposta), 'Fechada' (terminada) ou 'Suspensa' (momentaneamente indisponível)), valor apostado, probabilidade de sucesso e identificador do jogo.

Aposta
-Estado_id
-Valor
-odd
-Jogo_id
+getEstado()
+setEstado(Estado) : void
+getValor()
+setValor(Valor) : void
+getJogo()
+setJogo(Jogo) : void
+getOdd()
+setOdd(odd) : void

Figura 5.10: Aposta e respetivos métodos

Estado
-Estado
+getEstado()
+setEstado(Estado) : void

Figura 5.11: Estado e respetivos métodos

5.4.4 Boletim

O boletim consiste numa lista de apostas 'Abertas' do utilizador e no total ganho que o utilizador pode potencialmente ganhar.

Boletim
-id -ListaApostas -ganhoTotal
+getId() +setId(id) : void +getAttribute() +setAttribute(attribute) : void +getListaApostas() +setListaApostas(ListaApostas) : void +getOddTotal() +setOddTotal(oddTotal) : void

Figura 5.12: Boletim e respectivos métodos

5.4.5 HistoricoApostas

O histórico de apostas possui uma lista de boletins. Contudo, as apostas incluídas em cada boletim são apostas de estado 'Fechada', isto é, apostas já terminadas.

HistóricoApostas
-boletins : List<Boletim>
+getBoletimID() +setBoletimID(boletimID) : void

Figura 5.13: Histórico de apostas e respectivos métodos

5.4.6 Desporto

Este componente contém todas as informações necessárias relativas a um desporto, como nome, identificador e topologia. A topologia vai determinar a estrutura da aposta para um dado desporto.

desporto
-Nome -desporto_id -topologia
+getNome() +setNome(Nome) : void +getDesporto_id() +setDesporto_id(desporto_id) : void +getTopologia() +setTopologia(topologia) : void

Figura 5.14: Desporto e respectivos métodos

5.4.7 Equipa_Indivíduo

Este componente é referente a uma dada equipa ou indivíduo. Uma vez que estas entidades podem ser identificadas com os mesmos dados, decidiu-se juntá-las numa só classe.

Cada equipa/indivíduo tem um nome associado e estatísticas. Estas estatísticas correspondem a dados extraídos da API como número total de perdas, vitórias e número total de jogos.

Equipa_Individuo
-Nome
-estatisticas : Estatisticas
+getNome()
+setNome(Nome) : void
+getEstatisticas() : Estatisticas
+setEstatisticas(estatisticas : Estatisticas) : void

Figura 5.15: Equipa/Individuo e respetivos métodos

Estatisticas
-vitorias
-perdas
-nTotalJogos
+getVitorias()
+setVitorias(vitorias) : void
+getPerdas()
+setPerdas(perdas) : void
+getNTotalJogos()
+setNTotalJogos(nTotalJogos) : void

Figura 5.16: Estatísticas e respetivos métodos

Capítulo 6

Runtime View

Os use cases que envolvem mais interações entre diferentes componentes do sistema e entidades externas correspondem a "Criar Aposta", "Visualizar o histórico de apostas", "Consultar estatísticas", "Efetuar Pagamento" e "Receber notificação de jogo".¹

Ao longo desta secção irão ser expostos diagramas temporais de como cada funcionalidade representada por estes use cases é executada.

6.1 Criar Aposta

Para ser possível criar uma aposta, as suas informações deverão ser armazenadas na base de dados da aplicação, associadas ao utilizador que a criou. No contexto da aplicação, não deverão existir apostas que não estejam associadas a um boletim. Assim, para este efeito, quando se pretende criar uma aposta um novo boletim é criado ou é seleccionado um já existente na base de dados.

Para observar a execução da funcionalidade ao longo do tempo:

¹Para uma descrição mais detalhada dos Use Cases consultar Apêndice A - Descrição de Use Cases do documento de requisitos.

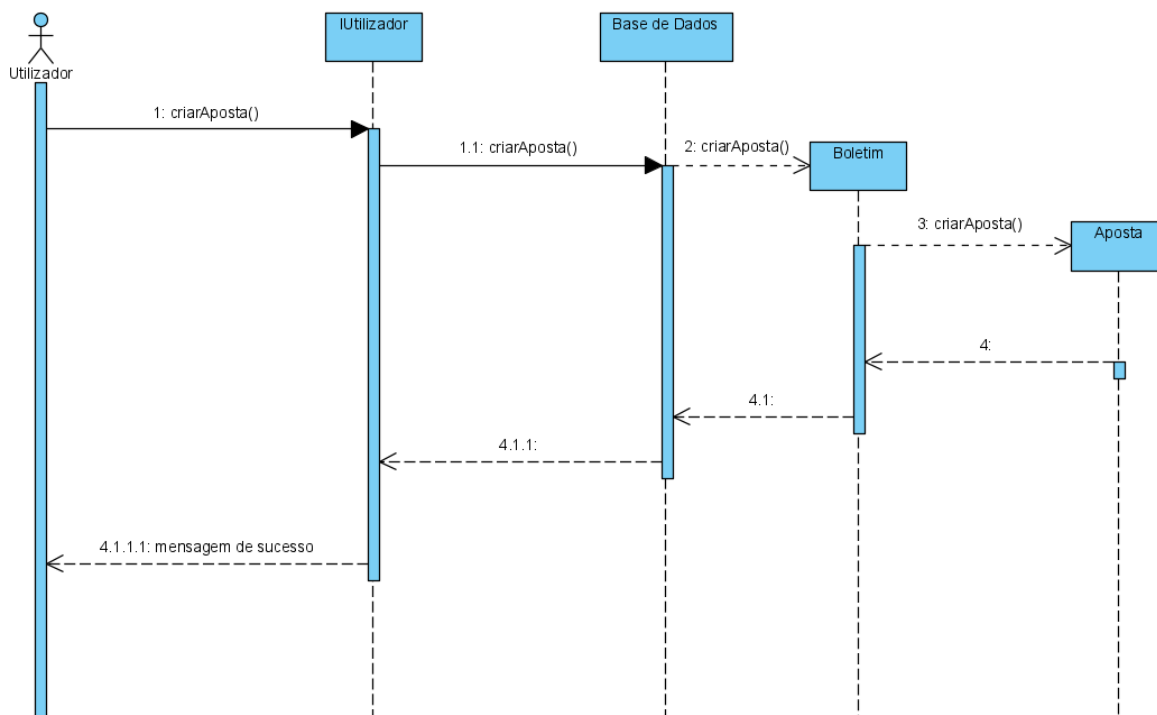


Figura 6.1: Diagrama de Sequência do use case **Criar Aposta**

6.2 Visualizar o histórico de apostas

De modo a ser possível consultar o histórico de apostas, este tem de ser anteriormente armazenado na base de dados associado ao respetivo utilizador. Com isto, para o consultar é necessário aceder à base de dados.

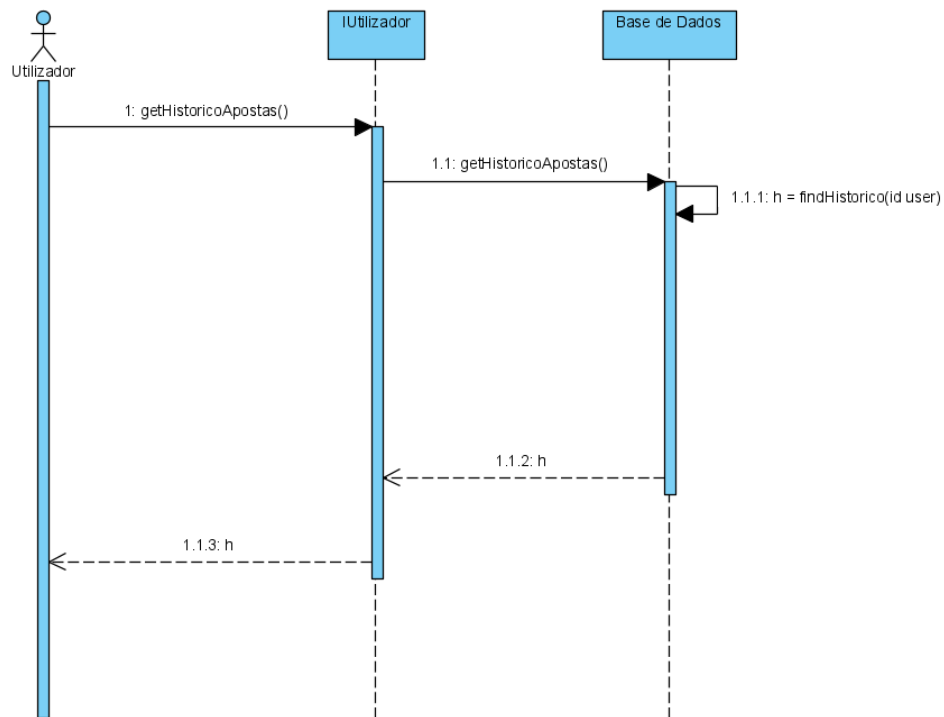


Figura 6.2: Diagrama de Sequência do use case **Visualizar histórico de apostas**

6.3 Consultar estatísticas

Para consultar as estatísticas de uma dada equipa ou indivíduo, é necessária uma conexão à API. A classe **Estatísticas** é responsável por retirar o número de vitórias, perdas e o número total de jogos.

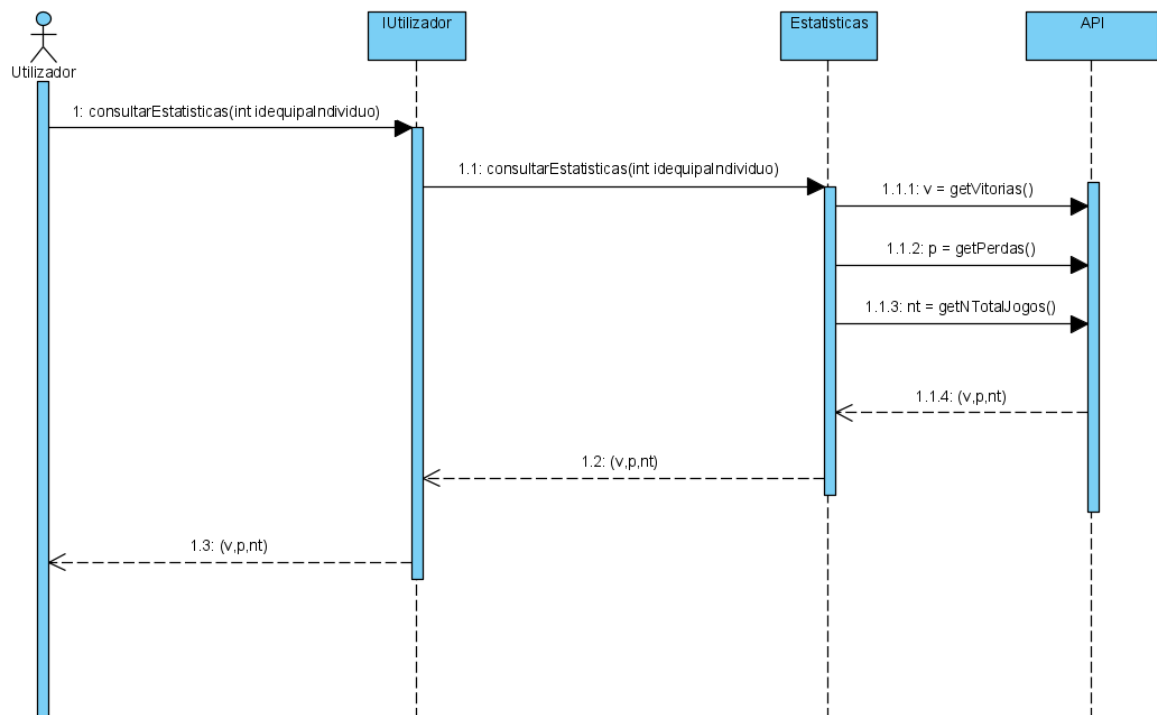


Figura 6.3: Diagrama de Sequência do use case **Consultar Estatísticas**

6.4 Efetuar pagamento

Quando o utilizador cria uma aposta, para esta ficar no estado 'Aberta', é necessário efetuar o pagamento. Este pagamento é feito através de um serviço de pagamento exterior ao sistema.

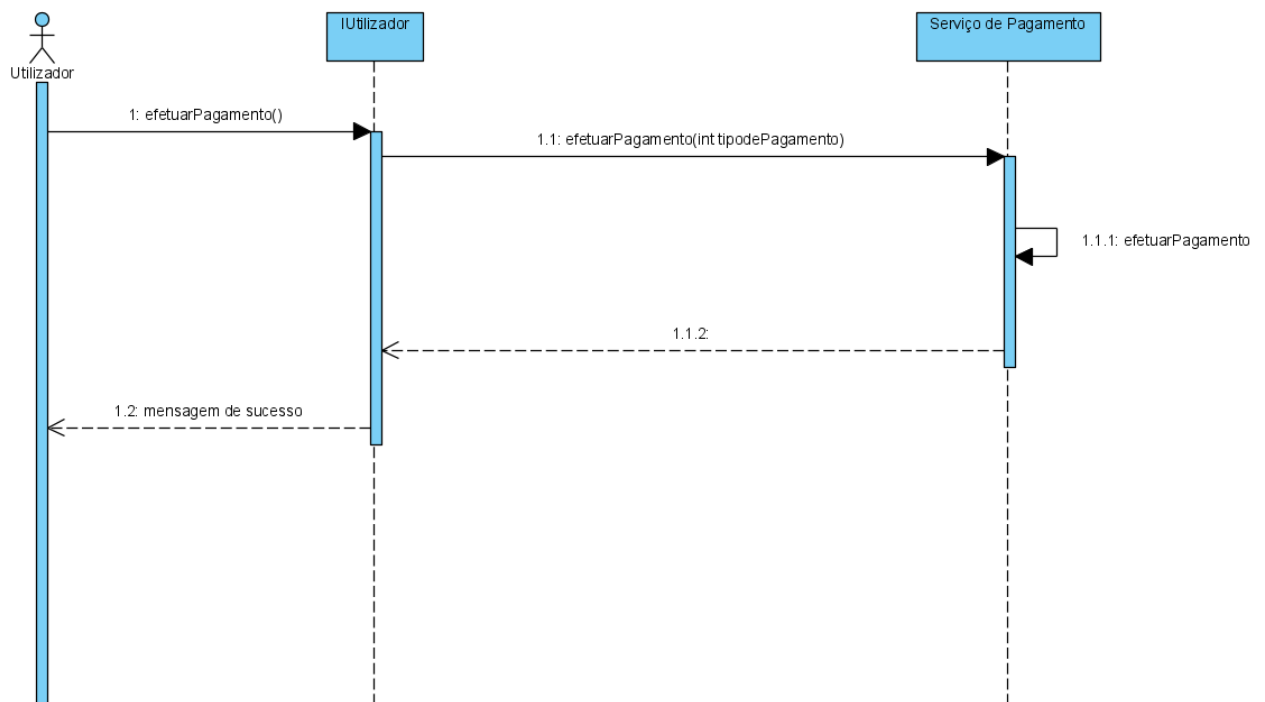


Figura 6.4: Diagrama de Sequência do use case **Efetuar Pagamento**

6.5 Receber notificação de jogo

Tendo em conta as apostas 'Abertas' do utilizador, a aplicação envia uma notificação sempre que um jogo está prestes a começar. Para este efeito, existirá um método que entre intervalos de tempo intercalados consulta a API para verificar se não existem nenhuns jogos incluídos em apostas prestes a começar.

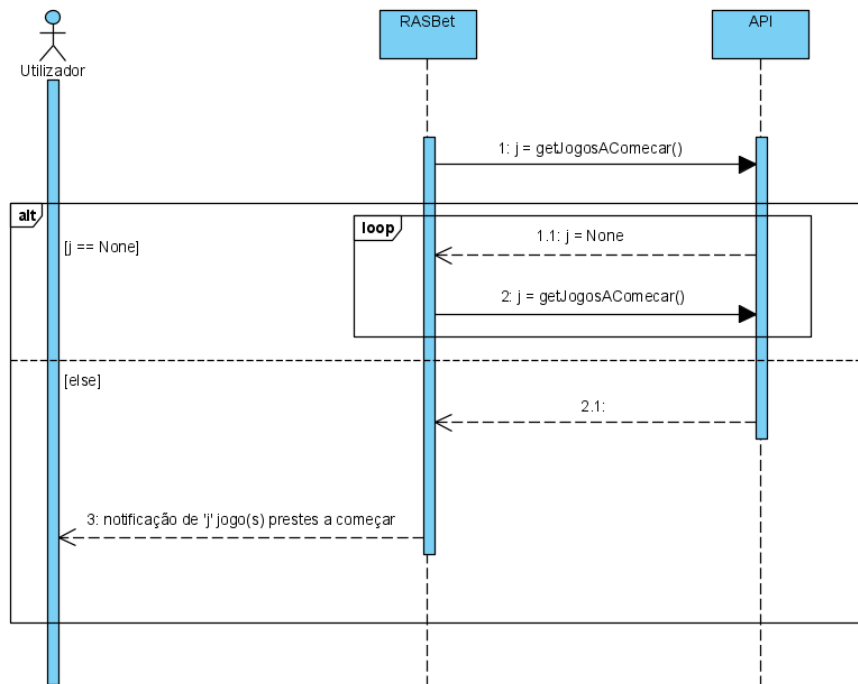


Figura 6.5: Diagrama de Sequência do use case **Receber notificação de jogo**

Capítulo 7

Crosscutting Concepts

7.1 Dependências

Tal como referido anteriormente no documento, a aplicação vai ser principalmente dependente da sua base de dados e da API.

Caso exista uma falha grave na base de dados a utilização da aplicação pode ser comprometida, uma vez que são os dados dos utilizadores que estão em risco. Isto também pode levar a um problema com registo de novos utilizadores, caso a base de dados, por exemplo, atinja a sua capacidade de armazenamento máxima.

Para além disto, também pode ocorrer indisponibilidades por parte da API. Caso este cenário ocorra, pode levar a atrasos na aplicação, nomeadamente na apresentação de resultados de jogos e de estatísticas dos seus participantes.

7.2 Domínio do Projeto

Através do seguinte diagrama podemos observar o domínio do produto em desenvolvimento.

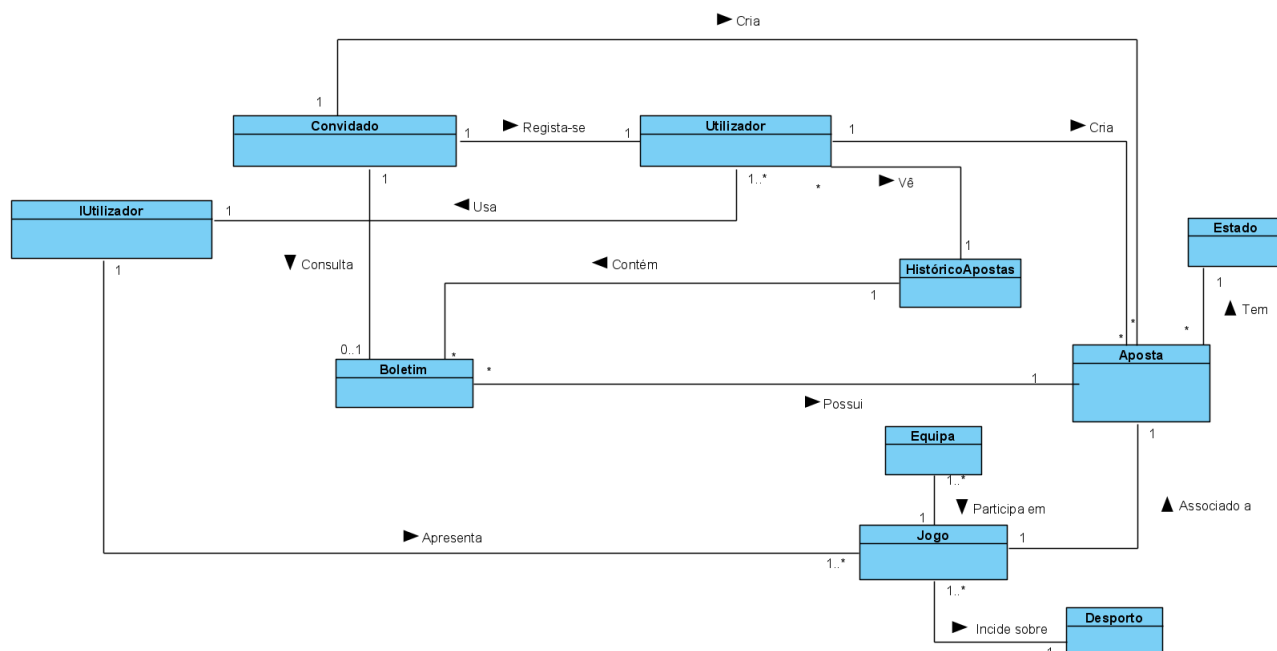


Figura 7.1: Modelo de Domínio

Capítulo 8

Decisões de Arquitetura

Com base nas restrições e funcionalidades pedidas pelos diversos *stakeholders* do projeto, foi desenvolvida a estrutura já exposta na primeira fase de desenvolvimento do projeto, apresentada no seguinte diagrama de componentes:

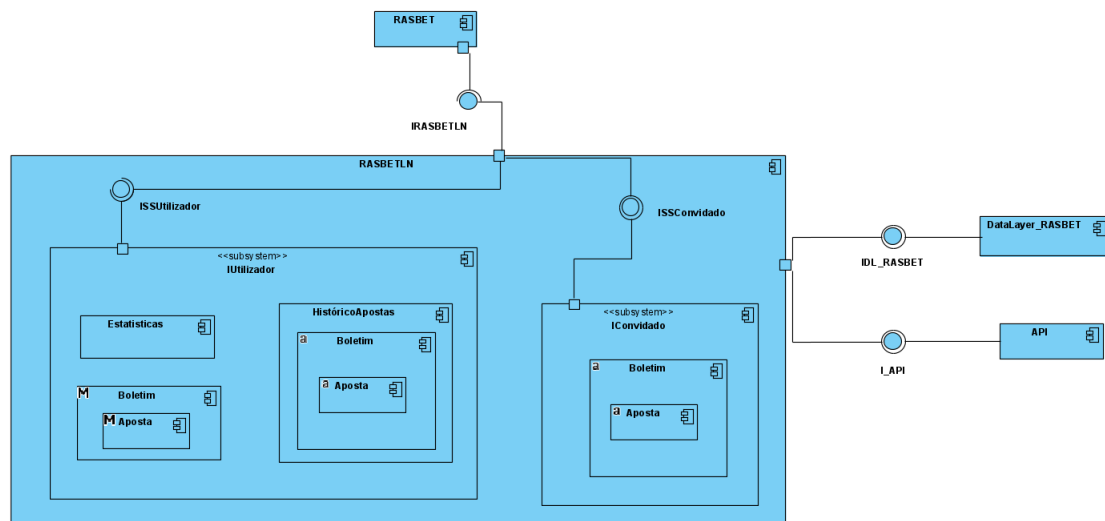


Figura 8.1: Diagrama de Componentes

Para atingir as metas de qualidade expostas na secção 1.2, decidiu-se construir a aplicação com uma base de dados a ela associada, com uma possível conexão a uma API.

Uma vez que a segurança é uma das metas de qualidade mais importantes do projeto, apesar do custo associado, investir numa base de dados própria é crucial. A sua gestão estrita permite uma maior segurança para os utilizadores da aplicação.

Também foi decidido construir a aplicação para permitir a sua portabilidade tanto para dispositivos móveis(por exemplo, smartphones) como para outro tipo de dispositivos(computadores).

Capítulo 9

Qualidades

9.1 Árvore de Qualidades

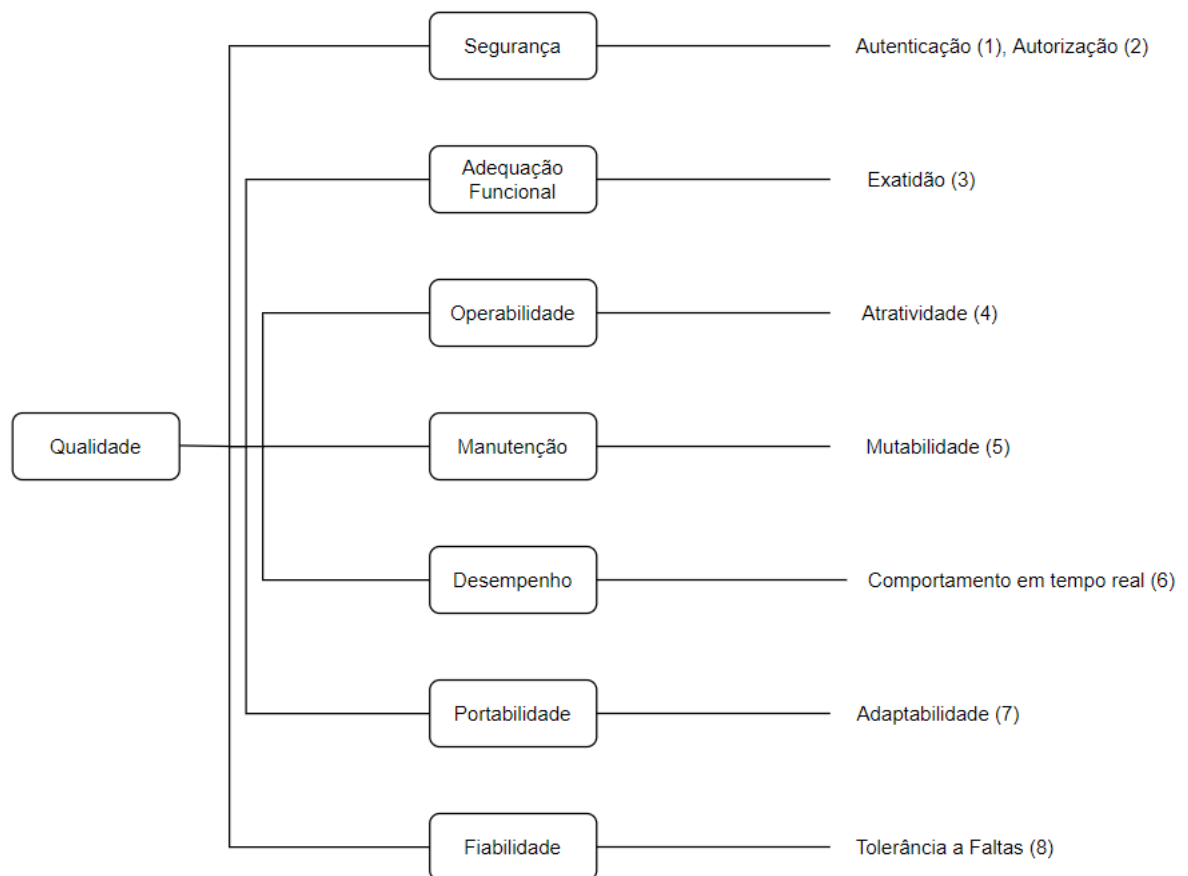


Figura 9.1: Árvore de Qualidades

9.2 Cenários de Qualidade

Cenários de Qualidade	
Número	Descrição
1	O utilizador, para conseguir aceder às suas informações pessoais (como credenciais, informações sobre pagamentos, apostas e histórico de apostas) tem de iniciar sessão na aplicação obrigatoriamente.
2	Só o utilizador pode aceder aos seus dados pessoais. Nem mesmo os técnicos de informática e gestores da aplicação podem aceder a informação pessoal de uma determinada conta pessoal.
3	Todas as funcionalidades requisitadas pelos vários <i>stakeholders</i> são implementadas e de possível uso ao utilizador.
4	A aplicação possuirá uma interface amigável e de fácil uso para o utilizador geral. Em cerca de 5 minutos um utilizador deverá ser capaz de executar cerca de 80% das funcionalidades da aplicação.
5	A base de dados será de fácil modificação, visto que vai necessitar de uma estratégia de limpeza de dados não necessários ou desatualizados entre determinados intervalos de tempo.
6	Depois da implementação de vários servidores e de vários testes os analistas do projeto poderão analisar o comportamento da aplicação em tempo real.
7	A aplicação será adaptável a vários tipos de dispositivos diferentes, principalmente computadores e smartphones.
8	A aplicação estará preparada para processar input incorreto inserido pelo utilizador. Este tipo de input não irá influenciar negativamente a aplicação.

Capítulo 10

Riscos e Custos

Uma vez que a aplicação tem certas dependências (tal como abordado na secção 7.1) existem potenciais riscos do sistema, com um determinado custo associado.

Ao longo desta secção irão ser expostos os vários riscos possíveis, assim como o seu respetivo custo, probabilidade de ocorrência e estratégias para diminuir a sua probabilidade e/ou custo.

10.1 Risco: A base de dados atinge a sua capacidade máxima

Se a aplicação tiver grande sucesso no mercado (que é o objetivo), esta vai ser muito utilizada com vários clientes simultaneamente. Isto irá sem dúvida gerar um grande armazenamento de dados dos diferentes utilizadores que se registam na aplicação e que fazem diversas apostas.

Este risco pode levar a um custo monetário bastante elevado pela parte da organização, uma vez que sem a base de dados a aplicação não irá funcionar (uma vez que não será capaz de armazenar as apostas criadas, levando por sua vez a uma paragem de fluxos monetários).

- Probabilidade de ocorrência: Alta;
- Custo: Alto;
- Gravidade: Alta.

10.1.1 Minimização

A probabilidade deste risco ocorrer pode ser diminuída significativamente com uma boa estratégia de gestão da base de dados: eliminação periódica de dados desatualizados e investimento num tamanho de armazenamento adequado à média de utilizadores que usam a aplicação por dia.

10.2 Risco: API indisponível

Caso a API utilizada bloqueie facilmente com pedidos em larga escala, a aplicação irá ser influenciada negativamente, nomeadamente na funcionalidade de obtenção de estatísticas e de consulta de resultados de jogos.

Este risco pode descer a qualidade da experiência do utilizador, uma vez que neste cenário irá deparar-se com uma redução repentina do desempenho da aplicação.

- Probabilidade de ocorrência: dependente da API a ser escolhida;
- Custo: Moderado;
- Gravidade: Moderada.

10.3 Minimização

Para evitar este risco será necessário escolher uma API eficiente que suporte muitos pedidos em simultâneo. A utilização de uma API rápida e de elevada capacidade irá diminuir significativamente a probabilidade de ocorrência deste risco.

10.4 Risco: Serviço de pagamento indisponível

Caso o serviço de pagamento utilizado falhe de alguma maneira vai provocar um bloqueio nos fluxos monetários, uma vez que vai impedir que as apostas sejam validadas e concluídas.

Este risco pode levar a uma perda significativa de lucro por parte da casa de apostas, apesar da sua probabilidade de ocorrência ser baixa¹.

- Probabilidade de ocorrência: Baixa/Moderada;
- Custo: Alto;
- Gravidade: Alta.

10.5 Minimização

Para evitar este risco será necessário escolher um serviço de pagamento que seja muito utilizado na atualidade, uma vez que os mais utilizados costumam ser os mais eficientes no mercado.

¹Isto porque os serviços de pagamento mais utilizados online na atualidade estão preparados para lidar com fluxos monetários em larga escala.

Capítulo 11

Conclusões e Trabalho Futuro

Nesta segunda parte do projeto tornou-se possível o planeamento mais aprofundado da arquitetura da solução a desenvolver, bem como a sua estrutura e objetivos.

Por fim, Na terceira e última fase, será elaborada a aplicação. Esta implementação seguirá o plano desenvolvido, que contém em detalhe, todo o processo que deverá ser seguido minuciosamente. Esta planificação permitirá um desenvolvimento pacífico e simplificado, uma vez que existe a garantia de que cada constituinte e processo se encontra descrito da melhor forma possível e sempre com o maior detalhe.

Não descartando a importância individual de cada parte constituinte do projeto, teve-se sempre em conta a concordância e consistência entre as mesmas, procurando constituir uma linha de montagem desde a primeira fase até à última.

É também de importância referir que tratando-se de um projeto iterativo, haverão sempre conceitos que poderão ser melhorados e modificados entre fases bem como os novos atributos que serão adicionados em cada passo. Todos estes atributos são tidos em conta, havendo documentação que apoie cada uma das decisões tomadas. Deste modo, é garantido que o processo que precede o desenvolvimento do produto, que é crucial em qualquer projeto, e a implementação da solução final, terão as melhores hipóteses de sucesso em termos de cumprimento de metas.