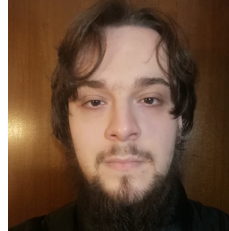
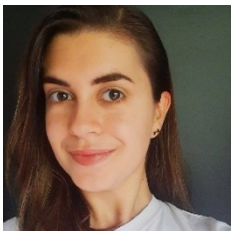


UMinho
Mestrado em Engenharia Informática
Requisitos e Arquiteturas de Software

Grupo ?, PL5 / Entrega 1

PG46983	Ana Luísa Lira Tomé Carneiro
PG46988	Ana Rita Abreu Peixoto
PG47194	Francisco José Vilão Matos Queiróz Peixoto
A89582	Henrique Manuel Ferreira da Silva Guimarães Ribeiro
PG47428	Luís Miguel Lopes Pinto



19 de Novembro de 2021
2021/2022

Conteúdo

Prefácio	4
Introdução	5
Triggers do Projeto	6
1. Propósito do Sistema	6
2. Clientes, consumidor, <i>stakeholders</i>	6
3. Utilizadores do sistema	7
Restrições do Projeto	8
4. Restrições Obrigatórias	8
Restrições de Tempo	9
Restrições de Ambiente de Implementação do Sistema	10
Restrições da Solução	11
5. Taxonomia e Definições	13
6. Factos e Pressupostos	14
Requisitos Funcionais	15
7. Âmbito do Trabalho	15
Modelo de Domínio do Sistema	15
Diagrama de Estados do Evento	17
Diagrama de sistemas	18
8. Âmbito do Sistema	19
Especificação de Use Cases	20
Especificação de Use Cases Macro	23
Diagrama de atividades	26
Mockups do Sistema	28
9. Requisitos Funcionais e de Dados	35
Requisito 1 - Autenticação	35
Requisito 2 - Registar Utilizador	36
Requisito 3 - <i>Log-out</i>	36
Requisito 4 - Consultar eventos	37
Requisito 5 - Aceder à área pessoal	37
Requisito 6 - Consultar histórico apostas	38
Requisito 7 - Consultar notificações	38

Requisito 8 - Alterar Configuração Conta	39
Requisito 9 - Levantar Saldo	39
Requisito 10 - Depositar Saldo	40
Requisito 11 - Consultar estatísticas	40
Requisito 12 - Atualizar Resultado	41
Requisito 13 - Caracterização da aposta	41
Requisito 14 - Registrar aposta	42
Requisito 15 - Alterar <i>odds</i> evento	42
Requisito 16 - Adicionar evento	43
Requisito 17 - Caracterização do evento	43
Requisito 18 - Atualizar estado aposta	44
Requisito 19 - Receber notificação	44
Requisito 20 - Receber quantia	45
Requisito 21 - Fechar apostas	45
Requisitos Não-Funcionais	46
10. Aparência	46
Requisito 22 - Atrativo a fãs de desporto	46
Requisito 23 - <i>Look</i> identificável	47
Requisito 24 - Representação do logótipo	47
11. Usabilidade	48
Requisito 25 - Suporte de várias moedas	48
Requisito 26 - Suporte a vários meios de pagamento	49
Requisito 27 - Suporte a várias línguas	49
Requisito 28 - Pessoas não familiarizadas com sites de apostas	50
Requisito 29 - <i>Time-zone</i> dos eventos	50
12. Performance	51
Requisito 30 - Nível de precisão	51
Requisito 31 - Atualização rápida de informações	52
Requisito 32 - <i>Uptime</i> Alto	52
Requisito 33 - Armazenamento dos dados	53
Requisito 34 - Crescimento no número de utilizadores	53
13. Operacional	54
Requisito 35 - Formato <i>WEB</i>	54
Requisito 36 - Formato aplicação	55
14. Manutenção e suporte	56
Requisito 37 - Suporte ao utilizador	56
Requisito 38 - Atualização do sistema	57
Requisito 39 - Resolução de Erros	58
15. Segurança	59
Requisito 40 - Informação Errada	59
Requisito 41 - Encriptação da informação	60

Requisito 42 - Políticas de privacidade	61
Requisito 43 - Gerir Informação	62
16. Cultural e político	63
Requisito 44 - Casa de apostas legal	63
17. Legal	64
Requisito 45 - Maiores de 18 anos	64
Requisito 46 - Leis de apostas	65
Resumo dos Requisitos	66
Conclusão	69

Prefácio

No decorrer da realização deste documento de requisitos foram encontradas algumas dificuldades, tais como o estudo do domínio do problema e consequente modelo de domínio. Neste tópico, a maior dificuldade foi representar de forma clara todos os conceitos inerentes a uma casa de apostas de forma a tornar o modelo completo, representativo do tema e não limitante. É importante referir que o trabalho realizado visa deixar abertura à implementação de novas funcionalidades no futuro, sem que seja necessário efetuar alterações bruscas aos modelos já implementados.

No que toca aos requisitos levantados (funcionais e não funcionais) consideramos relevante referir quais os métodos usados no seu levantamento, entre eles :

- **Questionários:** a equipa questionou algumas pessoas que utilizam outros *sites* de apostas quanto à usabilidade do sistema.
- **Observação:** alguns utilizadores foram observados a utilizar sistemas semelhantes ao RASBet.
- **Entrevistas:** os *stakeholders* foram entrevistados para fornecer a sua visão do sistema e funcionalidades importantes.
- **Brainstorming:** consideramos relevante discutir ideias entre os elementos da equipa de modo a gerar e consolidar novas ideias.
- **Análise do domínio:** ao analisar o domínio do problema, foi possível observar alguns requisitos importantes para o projeto.

Por último, consideramos relevante analisar o contributo de cada elemento do grupo para o desenvolvimento do trabalho. Desta forma, cada elemento é classificado com + para alunos que contribuíram significativamente acima da média, 0 para alunos cujo contributo foi próximo da média e - para alunos que contribuíram significativamente abaixo da média:

- | | |
|----------------------|---|
| • Ana Luísa Carneiro | 0 |
| • Ana Rita Peixoto | 0 |
| • Francisco Peixoto | 0 |
| • Henrique Ribeiro | 0 |
| • Luís Miguel Pinto | 0 |

Introdução

Nesta 1ª fase do trabalho foi elaborado um documento de requisitos para a implementação de uma aplicação que dá suporte a apostas desportivas, RASBet. Este documento, desenvolvido no âmbito da Unidade Curricular de Requisitos e Arquiteturas de Software (RAS), aborda diversas questões relevantes para uma implementação futura, tais como os *triggers* do sistema, restrições do projeto, requisitos funcionais e não funcionais, e ainda é possível observar qual o âmbito do trabalho e do sistema a implementar.

Esta aplicação tem como principal objetivo o registo de apostas desportivas por parte dos utilizadores (público alvo). O conceito é um apostador aposta uma quantia num resultado relativo a um evento e, de acordo com a *odd* e o resultado final do evento, pode receber uma quantia monetária. Além disso, esta aplicação também tem em consideração outra entidade, um gestor de apostas, bem como as funcionalidades a si associadas (gestão do *site* de apostas).

No seguinte documento é possível analisar com detalhe os tópicos referidos anteriormente e obter uma visão da aplicação que será implementada no futuro.

Triggers do Projeto

1. Propósito do Sistema

Este projeto foi criado com o propósito de dar suporte a casas de apostas desportivas através de um produto informático que permite que os utilizadores façam vários tipos de apostas desportivas antes do evento desportivo ou no decorrer do mesmo. Com este produto pretende-se que utilizadores consigam apostar no resultado final de vários tipos de desportos e que recebam notificações com o resultado final das apostas realizadas e respetivos ganhos. Além disso, o produto final deve permitir ao gestor de apostas gerir os eventos disponibilizados pelo sistema, através da adição de novos eventos e através da atualização ou conclusão das apostas associadas a esse evento desportivo. Finalmente, o produto final também deve permitir ao gestor atualizar as *odds* no decorrer das apostas quer estas sejam feitas antes ou durante o evento.

2. Clientes, consumidor, *stakeholders*

Neste projeto consideramos como *stakeholders*:

- Os *odds*makers, que serão os responsáveis por determinar as *odds* do sistema, e terão interesse em toda a informação pertinente às apostas nele registadas.
- Os peritos legais, que serão necessários para garantir o cumprimento de todas as leis que incidem sobre aplicações de apostas *online*.
- Os peritos de *marketing*, que serão consultados para o design dos aspetos estéticos da aplicação.
- Os peritos de usabilidade, que deverão indicar estratégias e filosofias de *design* para garantir o fácil uso e navegação do sistema.

Consideramos como *stakeholders* também os utilizadores do sistema, que estão detalhados na secção seguinte, bem como a casa de apostas que comissionou o desenvolvimento desta aplicação, tendo esta, como **único** cliente e consumidor do projeto, o direito de acesso a estatísticas relativas ao desempenho do sistema, como o número de registados ou o total apostado pelos utilizadores da aplicação.

3. Utilizadores do sistema

O sistema de apostas *RASBet* pode ser acedido por 3 tipos de entidades distintas: utilizador, apostador e gestor de apostas.

O **utilizador** abriga a possibilidade de uso da aplicação quando ainda não foi efetuado *log-in* no sistema, podendo visualizar todos os eventos em aberto. No entanto, este não está autorizado para efetuar apostas. Permitir que o *website* seja visível antes de efetuar o *log-in* é um passo importante na medida em que fornece uma visão do produto, das funcionalidades, dos desportos e dos eventos do sistema a um possível apostador. Esta apresentação do sistema permite incitar no utilizador o desejo de apostar. Além disso, este tipo de utilizador é chave para o sucesso do produto, dado que são potenciais apostadores e estes irão determinar o sucesso da aplicação. É importante referir que a idade do utilizador deve ser superior a 18 anos para se poder registar no sistema, dado o domínio do programa.

A entidade **apostador** trata-se de um utilizador autenticado no sistema, com acesso a todos os eventos e capacidade de fazer apostas. Esta entidade é consequência da entidade utilizador. Após um utilizador se registar e posteriormente autenticar no sistema, passa a possuir acesso a outras funcionalidades do sistema relativas à sua conta de utilizador e apostas efetuadas. Para além de poder efetuar o registo de apostas (funcionalidade núcleo do programa), o que o torna um utilizador de extrema importância no sistema, também pode verificar informações relativas a apostas anteriores e, efetivamente, lucrar com os seus palpites para o resultado dos eventos.

Por fim, o **gestor de apostas** é encarregue de atualizar, adicionar e finalizar eventos, assim como calcular as *odds* de cada resultado. Para isso, é necessário que seja um perito em apostas desportivas e, consequentemente, que seja alguém com grande conhecimento neste domínio. Esta entidade é responsável por disponibilizar todos os eventos e apostas ao apostador, tratando de gerir o sistema de apostas e, por isso, é chave para o funcionamento da aplicação. Este tipo de utilizador é importante e prioritário na medida em que as suas ações suportam a principal funcionalidade do programa: permitir a realização de apostas aos apostadores.

Restrições do Projeto

4. Restrições Obrigatórias

Esta secção descreve as restrições no eventual *design* do produto. As restrições obrigatórias do projeto devem ser tratadas como um tipo de requisitos, tal como os funcionais e não funcionais. Posto isto, torna-se evidente que as restrições serão expressas na mesma forma tabular que os requisitos.

Em particular, existem sete tipos de restrições obrigatórias passíveis de serem abordadas nesta secção, sendo elas: restrições de solução (Tipo A), restrições de ambiente de implementação do sistema atual (Tipo B), restrições de parcerias (Tipo C), restrições *Software Off-the-shelf* (Tipo D), restrições ambiente de trabalho antecipado (Tipo E), restrições de tempo (Tipo F) e restrições de orçamento (Tipo G).

Neste trabalho não foram abordadas restrições do tipo C, D, E, uma vez que a equipa não considera que estes temas se enquadram no âmbito do trabalho a desenvolver. Em adição, dado que o projeto está a ser desenvolvido em contexto académico, também não são consideradas restrições de *budget* (restrições do tipo G). No entanto, é importante referir que este seria um ponto a considerar caso fosse um projeto no mundo real.

Restrições de Tempo

- Restrição 01 - *Deadline*

Requisito #:	RO01	Tipo de Requisito:	Tipo F	Use Case #:	-
Descrição:	O sistema deve ser implementado em 3 fases diferentes cada uma com <i>deadlines</i> específicas.				
Raciocínio:	É necessário que o sistema seja desenvolvido por fases sobre restrições temporais para que haja um acompanhamento da realização do projeto e para q haja uma verificação e validação de todos os processos e tarefas realizadas em cada uma das fases.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	Devem existir 3 fases, uma que corresponde ao documento de requisitos para ser entregue em novembro de 2021, a segunda que alberga aspetos arquiteturais da solução a ser entregue a dezembro de 2021 e por fim a terceira que representa o produto final a ser entregue janeiro de 2022.				
Satisfação do Consumidor:	5	Insatisfação do Consumidor:	5		
Prioridade:	5	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

Restrições de Ambiente de Implementação do Sistema

- **Restrição 02 - Conexão com a *internet***

Requisito #: **RO02**

Tipo de Requisito: **Tipo B**

Use Case #: -

Descrição: **O sistema necessita obrigatoriamente de conexão com *internet*.**

Raciocínio: **Como o produto final será implementado em formato *WEB* ou *mobile* então a única restrição do sistema será de este necessitar conexão à *internet* para que os utilizadores da aplicação utilizem as funcionalidades.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deve mostrar uma mensagem de erro caso o sistema não detete conexão com *internet* .**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **5**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

Restrições da Solução

- Restrição 03 - Base de dados

Requisito #: **RO03**

Tipo de Requisito: **Tipo A**

Use Case #: -

Descrição: **O sistema deve ser implementado utilizando uma base de dados .**

Raciocínio: **O produto deverá ser implementado utilizando um sistema de base de dados que vai armazenar a informação dos utilizadores, apostas e eventos de forma a que este conteúdo persista na aplicação.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deverá ter um *front-end* que vai permitir a interação entre os utilizadores e o *back-end*, sendo que essa *back-end* vai ter interações com a base de dados implementada.**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **5**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

- **Restrição 04 - Modelo MVC**

Requisito #: **RO04**

Tipo de Requisito: **Tipo A**

Use Case #: -

Descrição: **O sistema deverá ser construído usando o modelo MVC.**

Raciocínio: **Além de um design simples e fácil de implementar, o modelo permite uma fácil integração das camadas visuais e de interação do utilizador com a lógica de negócios.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deve ter uma camada de *model*, que representa o conjunto de funcionalidades do sistema, o *view*, que representa a camada *front-end* da aplicação e o *controller*, que representa a camada que vai receber dados do utilizador e enviar esses dados para o sistema.**

Satisfação do Consumidor: **4**

Insatisfação do Consumidor: **4**

Prioridade: **4**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

5. Taxonomia e Definições

<i>RASBet</i>	Nome do sistema de apostas a implementar
RAS	Requisitos e Arquiteturas de Software
UML	<i>Unified Modeling Language</i>
1x2	Tipo de aposta relacionada com da vitória, o empate e da derrota de uma equipa.
<i>odd</i>	Medida de probabilidade de um dado resultado.
U.M.	Unidade Monetária
EGBA	<i>European Gaming and Bet-ting Association</i>
SRIJ	Serviço de Regulação Inspeção de Jogos

6. Factos e Pressupostos

Para a realização deste trabalho foi necessário considerar alguns factos e pressupostos relativos a aspetos que afetam o sistema. Estes pressupostos são transitórios, ou seja, irão originar um requisito ou restrição. Deste modo, os pressupostos considerados foram os seguintes:

- As apostas ficam suspensas quando o gestor pretender que os apostadores não façam mais apostas, momentaneamente.
- As apostas ficam abertas quando o gestor pretender que os apostadores façam mais apostas.
- As apostas ficam fechadas quando o gestor fechar o evento associado às apostas
- As apostas não podem ser canceladas pelo apostador depois de estarem registadas.
- Um evento pode ser cancelado pelo gestor, sendo que os apostadores recebem de volta o valor da sua aposta.
- Quando um apostadores perde uma aposta este ganha +0.
- Quando um apostadores ganha uma aposta este ganha o valor da aposta a multiplicar pela *odd* da aposta.
- As *odds* de um evento podem ser alteradas no decorrer do evento contudo não afeta as apostas já armazenadas no sistema.
- As *odds* de um evento são alteradas quando o gestor de apostas achar por bem atualizar.
- Os eventos podem ter um número arbitrário de equipas, mas pressupõem um número finito de resultados com um único vencedor.
- O tipo de apostas que vão ser implementadas são simples e do tipo 1x2 (vencedor 1, empate e vencedor 2) e vencedor (qual é o participante que ganha o evento).
- Os utilizadores podem apostar nos eventos de Futebol, Andebol, Ténis e Fórmula 1.

É de notar que apesar de ser implementado alguns tipo de desportos e tipos de apostas, o sistema foi criado de forma a dar a possibilidade de acrescentar novos desportos e tipos de apostas ao sistema a qualquer momento. Além disso, supõem-se que o domínio do problema em questão é relativo ao mundo das casas de apostas, estando por isso fora do nosso contexto o meio de pagamento que os utilizadores usam para fazerem as apostas, levantar e ou depositar saldo. Contudo, é de notar que se este fosse um projeto para ser implementado em contexto real e não em contexto académico este seria um domínio que o sistema deveria obrigatoriamente implementar.

Requisitos Funcionais

7. Âmbito do Trabalho

Modelo de Domínio do Sistema

De forma a melhor caraterizar o projeto *RASBet* e o domínio do problema (casa de apostas) foi implementado um modelo de domínio (figura 1), cujo o objetivo é mostrar as diferentes entidades do sistema e as relações entre elas. Deste modo, consideramos pertinente representar diferentes tipos de desportos, que podem ser classificados como: Americano, Europeu ou Ordem. Os desportos americanos não permitem empates ao contrário dos desportos europeus. No caso dos desportos de ordem, esta é relativa aos desportos em que a vitória depende da posição dos participantes, como é o caso das corridas ou da fórmula 1. É de notar que os diferentes desportos indicados no modelo de domínio também podem ser caraterizados como individual ou coletivo.

Além disso, também podemos observar no modelo de domínio a existência de duas entidades que irão utilizar o sistema: apostador e gestor de apostas. O apostador pode efetuar uma aposta, enquanto que o gestor de apostas pode gerir o estado das apostas de um evento e gerir os eventos, bem como calcular as *odds* de cada resultado possível em cada evento.

É necessário ter em consideração os estados possíveis para uma dada aposta: Aberto, Suspenso ou Fechado. Inicialmente, um evento é inserido no sistema no estado Aberto pelo gestor, que permite aos apostadores poderem registar as suas apostas. Quando o estado das apostas associados ao evento é alterado para Suspenso, os apostadores deixam de poder fazer apostas momentaneamente. Por fim, as apostas ficam no estado Fechado quando o evento termina, seja devido ao encerramento convencional do evento ou por fatores externos ao sistema (por exemplo: desastres naturais). No caso em que um evento termina num registo normal, os apostadores são notificados do seu resultado e, caso tenham vencido a aposta, é creditado o valor ganho no seu saldo. Quando os eventos são terminados devido a outros fatores, os apostadores recebem uma notificação e é devolvido o valor da aposta na íntegra. É importante ter em consideração que o estado das apostas de um dado evento é da responsabilidade do gestor e que este tem controlo sobre o seu fluxo.

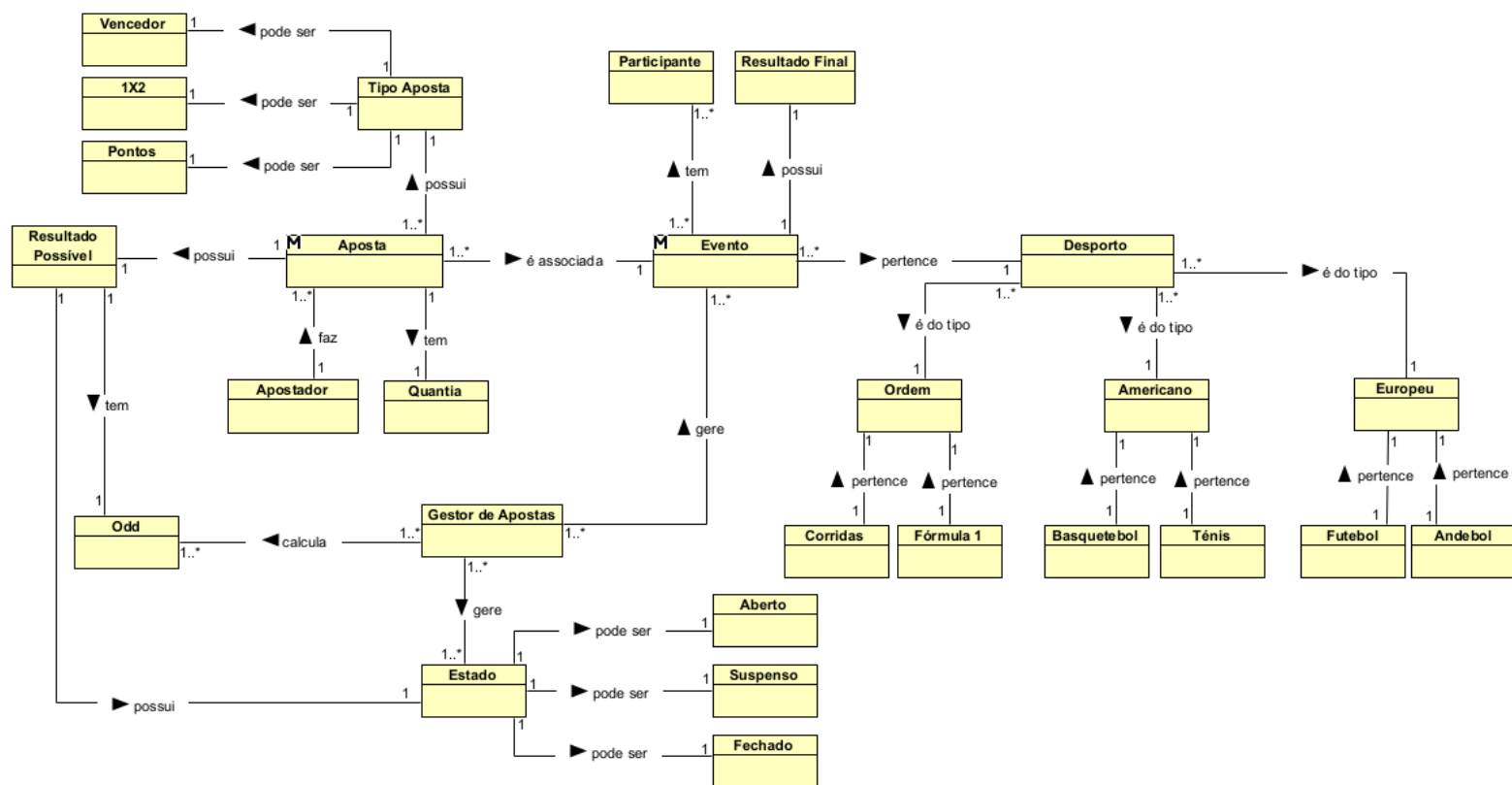


Figura 1: Modelo de domínio

No contexto do nosso sistema vamos limitar o domínio do problema aos desportos basquetebol, futebol, ténis, andebol e fórmula 1. Para o tipo de apostas vamos limitar a 1x2 (vencedor 1, empate e vencedor 2) para os eventos de futebol e andebol e ao Vencedor (qual é o participante que ganha o evento) para os eventos de formula 1 e basquetebol. No que toca às funcionalidades do sistema vão ser tanto implementadas as funcionalidade ligadas ao apostador como as ligadas ao gestor de apostas, tal como se encontra no modelo de domínio. Além disso, de forma a tornar o sistema mais completo e representativo de uma plataforma *WEB*, vamos ainda incluir funcionalidades relacionadas com notificações sobre o resultado das apostas, configurações de conta, credenciação, consulta de histórico de apostas, entre outras tal como se encontra na secção 8.

Diagrama de Estados do Evento

O diagrama de estados da figura 2 serve para demonstrar visualmente o motivo subjacente à existência de diversos estados das apostas associadas a um dado evento, bem como para possibilitar a enumeração dos possíveis fluxos de transição entre os estados e ilustrar as causas que podem despoletar essa mesma transição.

Tal como podemos ver na figura, existem apenas dois motivos que tornam um evento fechado, i.e, quando o jogo termina normalmente ou devido a um motivo externo que perturbe o normal funcionamento do mesmo. Só quando uma aposta se encontra "Fechada" é que são divulgados os resultados finais do evento em questão. Para além disso, quando um evento é criado todas as apostas a ele associadas tem estado "Aberto" e só quando o gestor altera o estado para "Suspenso" (devido a razões fora do controlo do sistema) é que não são permitidas novas apostas momentaneamente. Da mesma forma, quando uma aposta passa de "Suspenso" para "Aberto" o gestor está a abrir a possibilidade para que novas apostas surjam no evento.

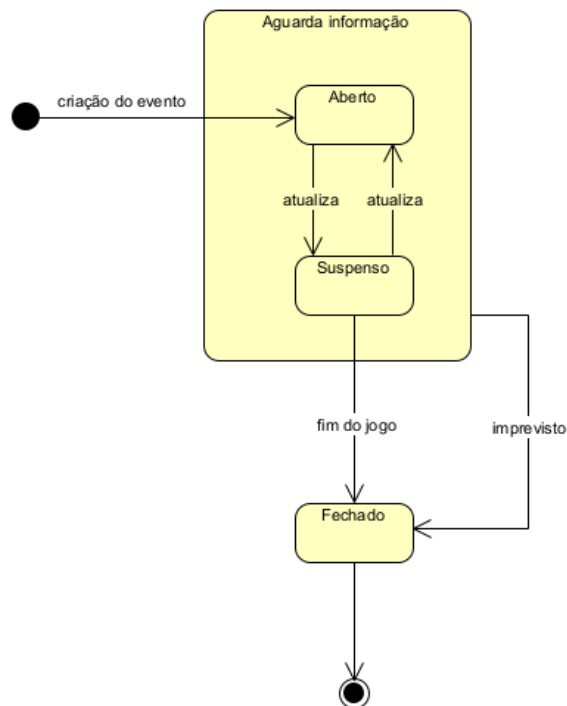


Figura 2: Diagrama de Estados de um Evento

Diagrama de sistemas

Na figura 3 podemos observar o diagrama de sistemas elaborado para o produto RASBet. Neste diagrama é possível observar os componentes do *back-end* e *front-end* da aplicação, assim como as principais funcionalidades associadas. No *back-end* estão presentes as camadas de negócio e de dados, onde estará a lógica do programa e os dados que o suportam, respetivamente. Na componente *front-end* aplicacional, está a vertente da interface que permitirá ao utilizador interagir com o sistema, através de versões *Web* e *mobile*.

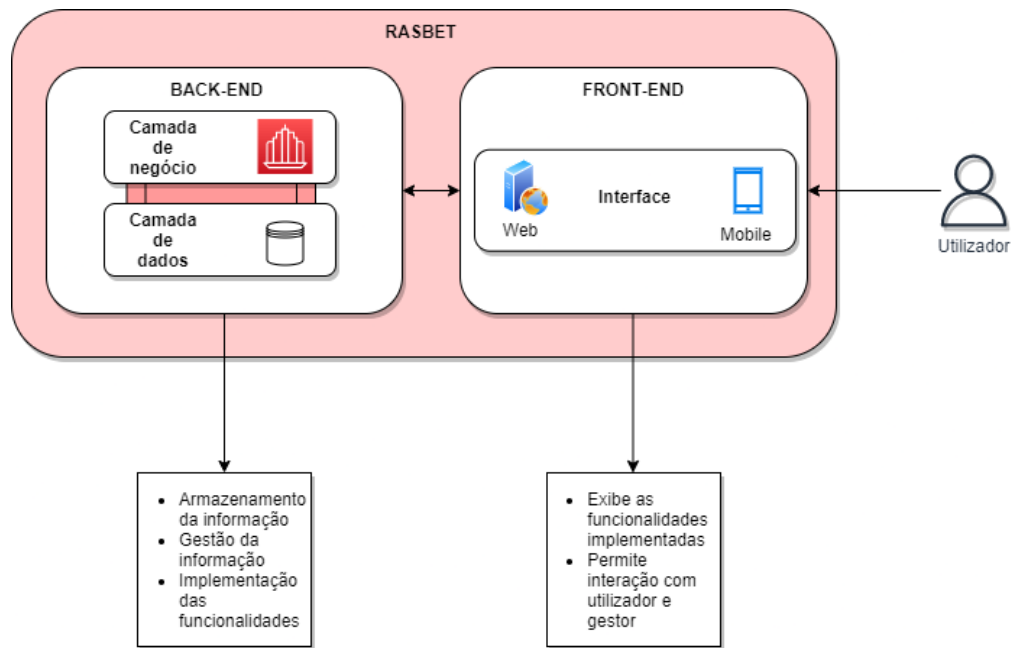


Figura 3: Diagrama de sistemas

8. Âmbito do Sistema

De forma a representar as funcionalidades do sistema, foi criado um diagrama de use cases com o intuito de modelar as interações entre os utilizadores do sistema e as funcionalidades de cada um.

Como podemos ver na imagem as funcionalidades macro do sistema, isto é, as funcionalidades principais do sistema foram especificadas a cor diferente no diagrama. Assim sendo, a funcionalidade principal do apostador é realizar apostas enquanto que o gestor de apostas pode atualizar as *odds*, adicionar novos eventos e atualizar e finalizar as apostas associadas a um dado evento.

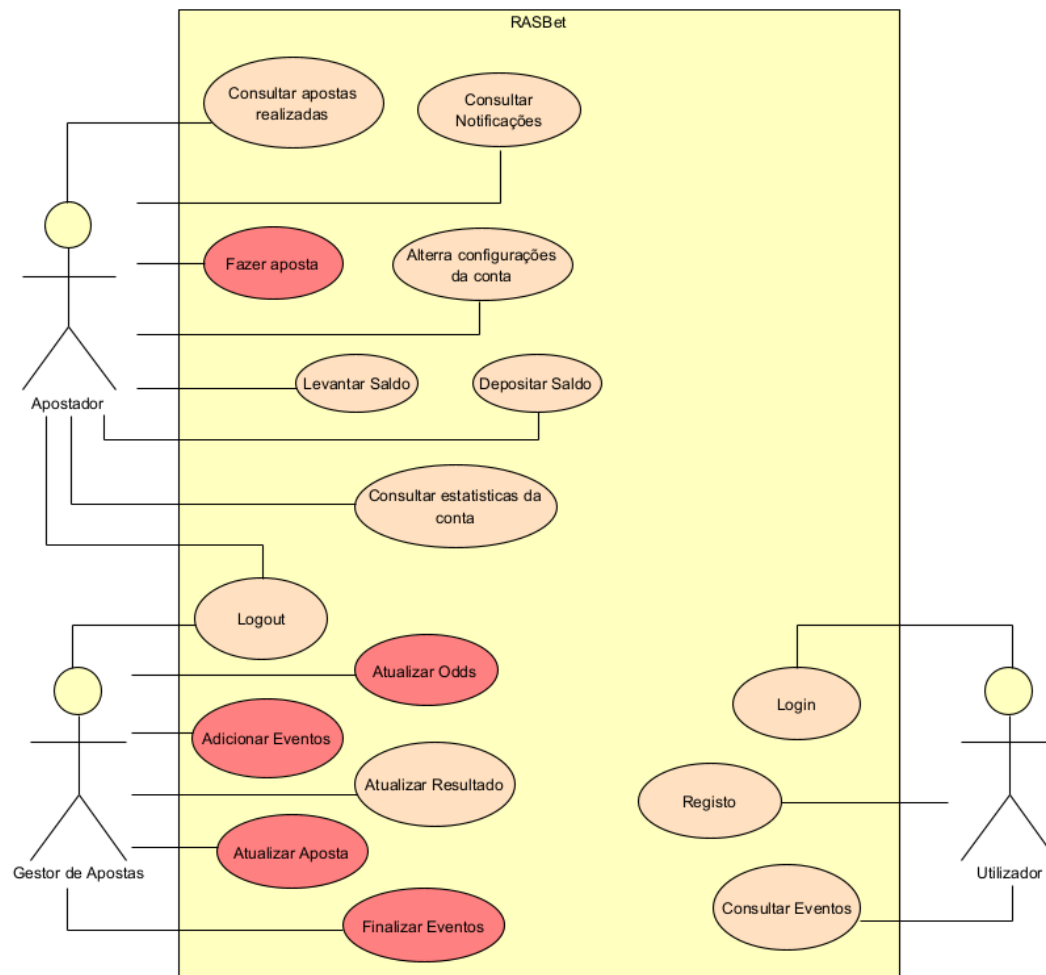


Figura 4: Diagrama de use cases

Especificação de Use Cases

- **UC01 - Login:**

Use Case: *Login* do utilizador.

Descrição: Utilizador autentica-se na aplicação.

Pré-Condição: Utilizador não está aplicação.

Pós-Condição: Utilizador fica com a sessão iniciada.

Cenário Normal:

1. O utilizador adiciona o email e password.
2. Sistema valida parâmetros.
3. Sistema inicia sessão do utilizador

Cenário Excecional 1[utilizador insere incorrectamente os parâmetros] (passo 2):

- 2.1. Sistema não valida parâmetros.
- 2.1. Sistema apresenta mensagem de erro.

- **UC02 - Registo:**

Use Case: Registar utilizador.

Descrição: O utilizador regista-se na aplicação.

Pré-Condição: O utilizador não está registado no sistema.

Pós-Condição: O utilizador fica registado na aplicação .

Cenário Normal:

1. O utilizador adiciona o nome de utilizador, email, password, data de nascimento e saldo.
2. Sistema valida parâmetros.
3. Sistema inicia sessão do utilizador

Cenário Excecional 1[utilizador insere um email já registado no sistema] (passo 2):

- 2.1. Sistema não valida parâmetros.
- 2.1. Sistema apresenta mensagem de erro.

- **UC03 - Logout:**

Use Case: *Logout* do utilizador.

Descrição: O gestor de apostas ou apostador sai da aplicação.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: O gestor de apostas ou apostador fica com a sessão terminada.

Cenário Normal:

1. O gestor de apostas ou apostador termina sessão na aplicação.
2. Sistema pergunta ao gestor de apostas ou apostador se pretende sair da aplicação.
3. Sistema desconecta gestor de apostas ou apostador da aplicação.

Cenário Excecional 1[O gestor de apostas ou apostador cancela fim de sessão] (passo 3):

3.1 Sistema cancela fim de sessão da aplicação.

- **UC04 - Consultar Eventos:**

Use Case: Consultar eventos.

Descrição: O utilizador consulta eventos disponíveis na aplicação.

Pré-Condição: -

Pós-Condição: O utilizador tem acesso às apostas disponíveis.

Cenário Normal:

1. Sistema consulta/disponibiliza os vários eventos disponíveis para apostar.

- **UC05 - Consultar apostas realizadas:**

Use Case: Consultar apostas realizadas.

Descrição: O apostador consulta histórico de apostas.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: O apostador tem acesso ao histórico.

Cenário Normal:

1. Sistema apresenta histórico de todas as apostas realizadas e pendentes assim como respetivos ganhos.

Cenário Excecional 1[apostador não tem apostas realizadas] (passo 1):

1.1 Sistema apresenta mensagem "Ainda não realizou apostas".

- **UC06 - Consultar notificações:**

Use Case: Consultar notificações.

Descrição: O apostador consulta notificações recebidas.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: O apostador tem acesso à lista de notificações.

Cenário Normal:

1. Sistema apresenta lista de notificações sobre a mudança de estado das apostas pendentes, e em caso disso, os ganhos da aposta.

- **UC07 - Alterar configurações da conta:**

Use Case: Altera configurações da conta.

Descrição: Apostador altera informações da conta.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: Apostador fica com as informações da conta alteradas.

Cenário Normal:

1. Sistema apresenta lista de informações que apostador pode a alterar.
2. Apostador escolhe informação a alterar.
3. Sistema atualizada informação alterada.

Cenário Excepcional 1[Apostador insere email inválido] (passo 2):

- 2.1 Apostador escolhe alteração do email.
- 2.2 Apostador insere email já registado no sistema.
- 2.3. Sistema apresenta mensagem de erro.

• **UC08 - Levantar Saldo:**

Use Case: Levantar Saldo.

Descrição: Apostador levanta saldo da conta.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: O apostador fica com um saldo menor do que anteriormente.

Cenário Normal:

1. Apostador indica quantia a levanta.
2. Sistema retira quantia do saldo disponível na conta.
4. Sistema apresenta mensagem de sucesso.

Cenário Excepcional 1[Apostador indica quantia superior ao saldo] (passo 2):

- 2.1 Sistema apresenta mensagem de erro.

• **UC09 - Depositar Saldo:**

Use Case: Depositar Saldo.

Descrição: Apostador deposita saldo da conta.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: O apostador fica com um saldo maior do que anteriormente.

Cenário Normal:

1. Apostador indica quantia a depositar.
2. Sistema adiciona quantia ao saldo disponível na conta.
4. Sistema apresenta mensagem de sucesso.

Cenário Excepcional 1[Apostador adiciona mais 0 unidades monetárias ao saldo] (passo 2):

- 2.1 Sistema apresenta mensagem de erro.

• **UC10 - Consultar estatísticas de conta:**

Use Case: Consultar estatísticas de conta.

Descrição: Apostador pode consultar estatísticas de conta.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: O apostador tem acesso às estatísticas do sistema.

Cenário Normal:

1. Sistema apresenta ganhos e as perdas com as apostas, os desportos, equipas e atletas mais apostados e a percentagem de apostas vencidas e perdida.

Cenário Excecional 1[apostador não tem apostas realizadas] (passo 1):

1.1 Sistema apresenta mensagem "Ainda não realizou apostas".

- **UC11 - Atualizar Resultado:**

Use Case: Atualizar Resultado.

Descrição: Gestor de apostas atualiza resultado do evento durante a aposta.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: Sistema regista atualização do resultado do evento.

Cenário Normal:

1. Gestor indica resultado do evento.
2. Sistema regista novo resultado.

- **UC11 - Atualizar Resultado:**

Use Case: Atualizar Resultado.

Descrição: Gestor de apostas atualiza resultado do evento durante a aposta.

Pré-Condição: O utilizador tem de estar autenticado no sistema.

Pós-Condição: Sistema regista atualização do resultado do evento.

Cenário Normal:

1. Gestor indica resultado do evento.
2. Sistema regista novo resultado.

Especificação de Use Cases Macro

- **UC12: Fazer Aposta**

Use Case: Fazer Aposta.

Descrição: Apostador regista aposta no sistema.

Pré-Condição: Apostador está registado na aplicação.

Pós-Condição: Aposta fica registada.

Fluxo Normal:

1. Apostador escolhe evento para fazer aposta.
2. Apostador escolhe possibilidade de resultado.
3. Apostador indica quantia a apostar.

4. Sistema avalia se quantia é menor ou igual ao saldo do apostador.
5. Sistema calcula possíveis ganhos.
6. Apostador aceita aposta.
7. Sistema regista nova aposta.

Cenário Alternativo[saldo menor que a quantia](passo 4):

- 4.1. Sistema avisa que a quantia apostada é superior ao saldo disponível.
- 4.2. Retorna ao passo 3.

Cenário Excepcional 1[apostador não encontra evento adequado] (passo 1):

- 1.1. Apostador cancela aposta.
- 1.2. Sistema não regista aposta.

Cenário Excepcional 2[apostador decide não colocar quantia] (passo 3):

- 3.1. Apostador cancela aposta.
- 3.2. Sistema não regista aposta.

Cenário Excepcional 3[apostador decide não aceitar aposta] (passo 6):

- 6.1. Apostador cancela aposta.
- 6.2. Sistema não regista aposta.

- **UC13: Atualizar Odds**

Use Case: Atualizar Odds.

Descrição: O gestor de apostas atualiza as *odds* de um evento.

Pré-Condição: Gestor autorizado na aplicação e os eventos a consultar no estado "Aberto"

Pós-Condição: As *odds* da aposta são alteradas.

Cenário Normal:

1. O gestor seleciona o evento para o qual vai atualizar as *odds*.
2. O gestor seleciona um ou mais resultados possíveis.
3. O gestor insere no sistema a *odd* respetiva a cada resultado.
4. Sistema atualiza as *odds* do evento.

- **UC14 - Adicionar Evento**

Use Case: Adicionar um Evento.

Descrição: Gestor adiciona um novo evento.

Pré-Condição: Gestor está autorizado na aplicação.

Pós-Condição: Existe um novo evento registado no sistema.

Cenário Normal:

1. Gestor fornece nome, equipas ou lista de participantes, data e hora do evento e *odds* para cada resultado.
2. Sistema valida parâmetros fornecidos.
3. Sistema guarda um novo evento, com as apostas em estado "Aberto", no sistema.
4. Sistema indica sucesso na criação do evento.

Cenário Alternativo[Data e Hora do evento inválidas](passo 2):

- 2.1. Sistema informa que a data e/ou hora do evento é inválida.
- 2.2. Retorna 1.

Cenário Excepcional[Gestor não coloca dados](passo 1):

- 2.1. Gestor cancela o evento.
- 2.2. Sistema não regista evento.

• UC15 - Atualizar Aposta

Use Case: Atualizar Aposta.

Descrição: Gestor atualiza o estado das apostas associadas a um evento.

Pré-Condição: Gestor está autorizado na aplicação e consulta eventos com o estado "Aberto" e "Suspendido".

Pós-Condição: Evento fica com um registo atualizado no sistema.

Cenário Normal:

1. Gestor escolhe evento que pretende atualizar.
2. Gestor escolhe estado para o qual pretende atualizar ("Aberto" ou "Suspendido")
3. Sistema regista a mudança no estado do evento.
4. Sistema atualiza estado das apostas relativas ao evento.

Cenário Excepcional 1[Gestor seleciona o estado a atualizar igual ao estado do evento](passo 3):

- 3.1. Sistema apresenta mensagem de erro.

• UC16 - Finalizar Evento

Use Case: Finalizar um Evento.

Descrição: Gestor finaliza um evento.

Pré-Condição: Gestor está autorizado na aplicação e o gestor só consulta eventos com o estado "Aberto" ou "Suspendido".

Pós-Condição: Evento fica com apostas fechadas.

Cenário Normal:

1. Gestor seleciona um evento.
2. Gestor indica que o evento terminou e o seu resultado.
3. Sistema regista resultado final do evento.
4. Sistema regista as apostas associadas ao evento com o estado "Fechado".
5. Sistema fecha as apostas relativas ao evento.
6. Sistema notifica apostadores que o evento terminou.
7. Sistema atribui as devidas quantias aos apostadores.

Cenário Alternativo[Evento termina antecipadamente](passo 2):

- 2.1. Gestor indica ao sistema qual a razão do evento terminar.
- 2.2. Sistema regista as apostas associadas ao evento com o estado "Fechado".
- 2.2. Retornar ao passo 4.

Diagrama de atividades

Os diagramas de atividades tem o propósito de demonstrar o fluxo entre as atividades e será utilizado para fazer a modelação de aspectos dinâmicos do sistema. No contexto do problema foram desenvolvidos diagramas de atividades que mostram a dinâmica e fluxo de cada use case macro ('Fazer Aposta', 'Adicionar Evento', 'Finalizar Evento', 'Atualizar Aposta' e 'Atualizar Odds'). Com estes dois diagramas conseguimos modelar e, conseqüentemente, melhor entender os passos e decisões fundamentais para a implementação dos use cases macro.

Na figura 5 conseguimos ver o fluxo de ações necessárias à futura implementação do use case de 'Fazer Aposta'. Com este diagrama conseguimos ver que o use case pode ser concluído em 2 estados diferentes: com uma nova aposta registada no sistema ou com a aposta cancelada e que são necessários 4 actividades fundamentais para o desenvolvimento da funcionalidade.

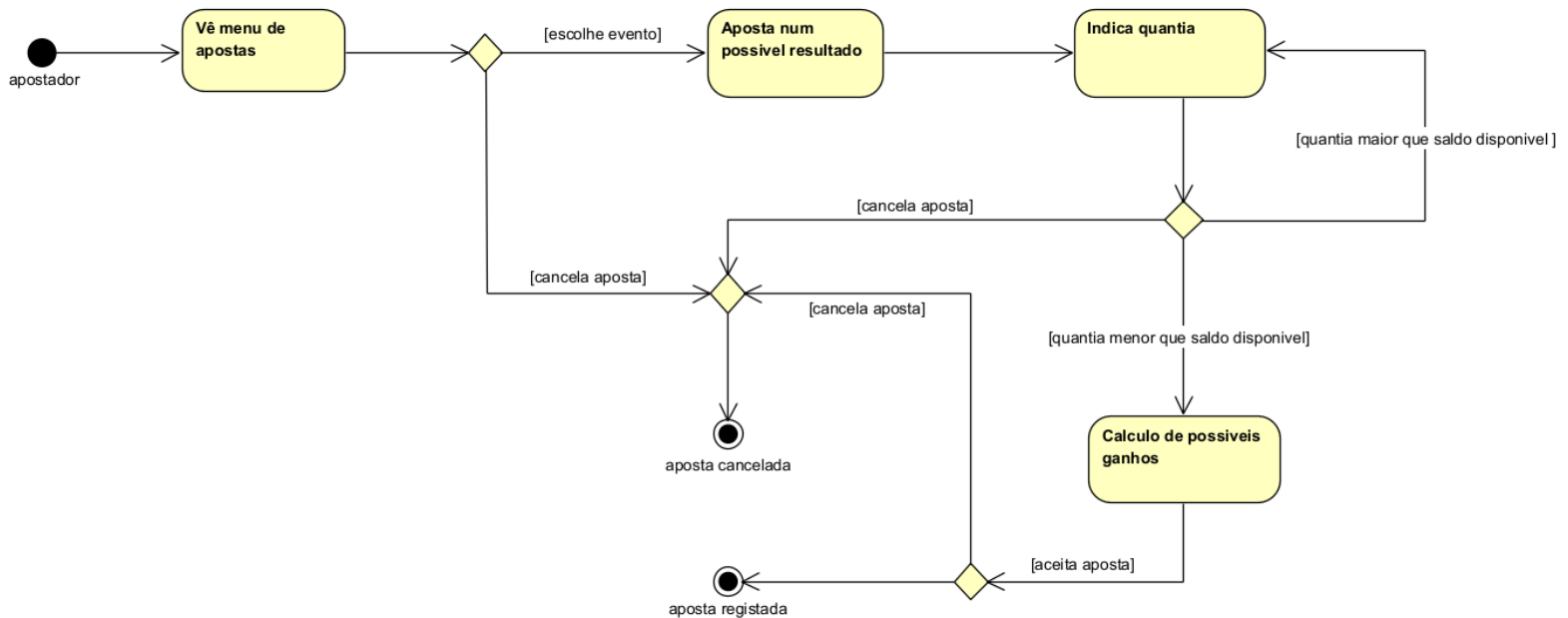


Figura 5: Diagrama de Atividade do use case 'Fazer Aposta'

No modelo de atividades da figura 6 conseguimos ver todas as atividades e decisões necessárias à implementação dos use cases de 'Adicionar Evento', 'Atualizar Apostas', 'Finalizar Evento' e 'Atualizar Odds', todos eles relacionados com o gestor de apostas. Com este diagrama podemos não só ver as funcionalidades base do gestor de apostas como também as diversas atividades, decisões e conseqüente resultado de cada use case que o sistema tem de disponibilizar ao gestor.

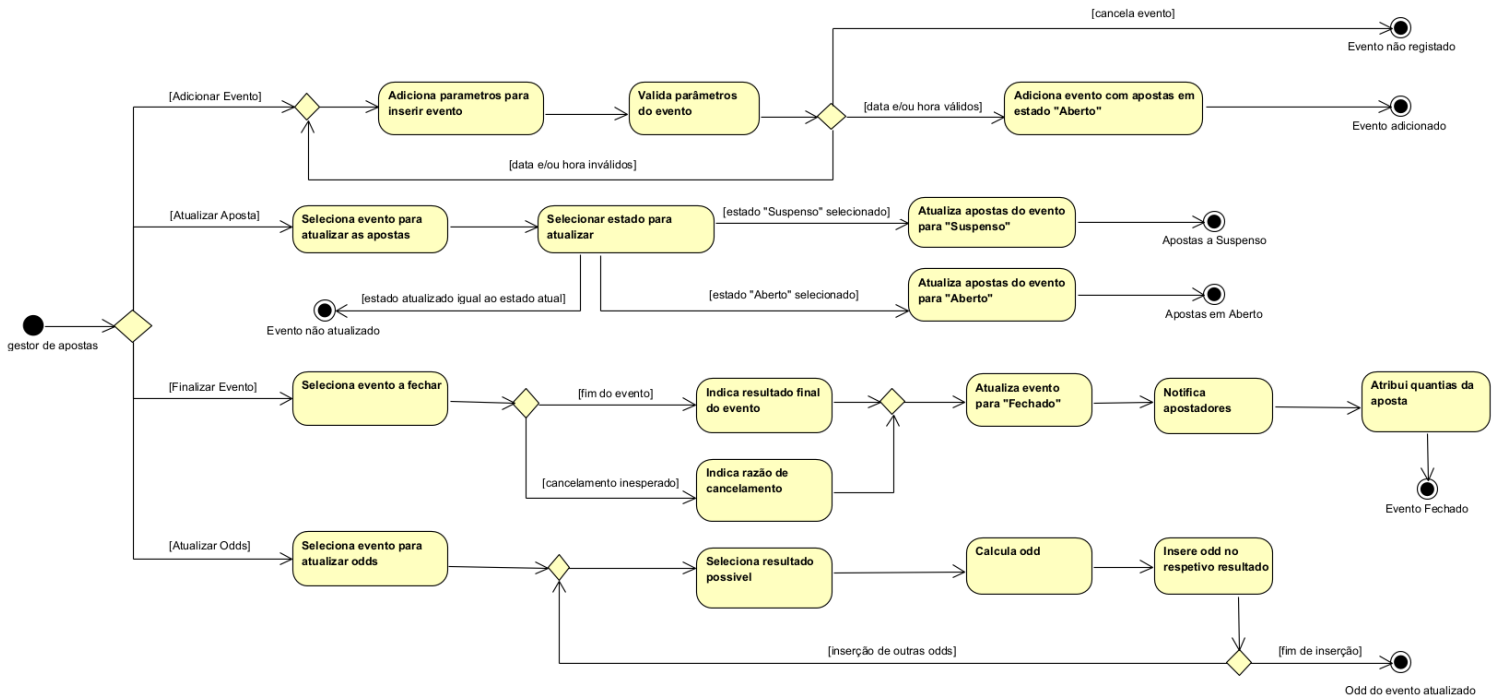


Figura 6: Diagrama de Atividade dos use cases do gestor de apostas

Mockups do Sistema

As *mockups* do sistema tem como objetivo representar os vários menus e páginas que os utilizadores do sistema vão ter acesso, modelando e idealizando desta forma a interface final do produto. Para isso foram desenvolvidos 5 *mockups* que representam os use cases macro.

Na figura 7 conseguimos ver a representação gráfica do menu que o apostador pode aceder quando pretender fazer uma aposta ou simplesmente consultar os diversos tipos de eventos disponíveis. Conseguimos também ver que o apostador tem de estar autenticado na aplicação, que os eventos podem ser filtrados por desporto e que o apostador vai ter acesso ao conjunto de apostas realizadas na aplicação na secção 'Apostas'. Na imagem podemos observar, a título de exemplo, que o apostador faz uma aposta no evento Irlanda-Portugal onde aposta em Portugal como vencedor. No boletim aparece a aposta sinalizada assim como a quantia apostada (20 euros) e os ganhos possíveis (25.80 euros), calculados de acordo com a *odd* 1,29. Com esta *mockup* conseguimos ver todas as atividades definidas no diagrama da figura 5, desde do menu de aposta até aos ganhos possíveis, passando por possível resultado (neste caso vitória 1) e quantia apostada (20,00 euros).

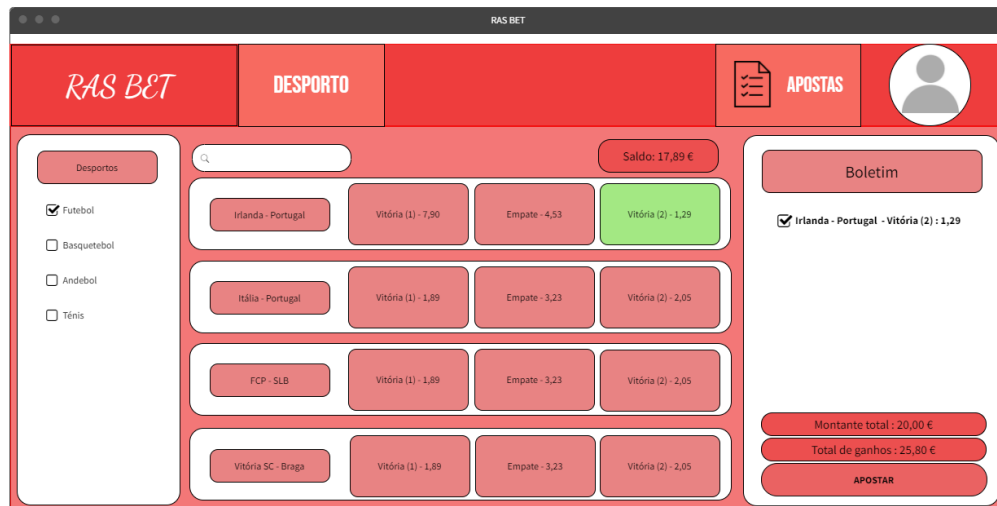


Figura 7: *Mockup* relativo ao Use case de 'Fazer aposta'.

Na figura 8 conseguimos ver a representação gráfica do menu que o gestor de apostas vai ter acesso quando necessitar gerir os eventos. Conseguimos ver que o gestor de apostas tem de estar autenticado na aplicação, que os eventos podem ser filtrados por desporto e que este pode adicionar novo evento, configurar e finalizar evento e, por fim, atualizar aposta. Com esta *mockup* conseguimos ver a atividade relacionada com a escolha e seleção de eventos presente no diagrama da figura 6.

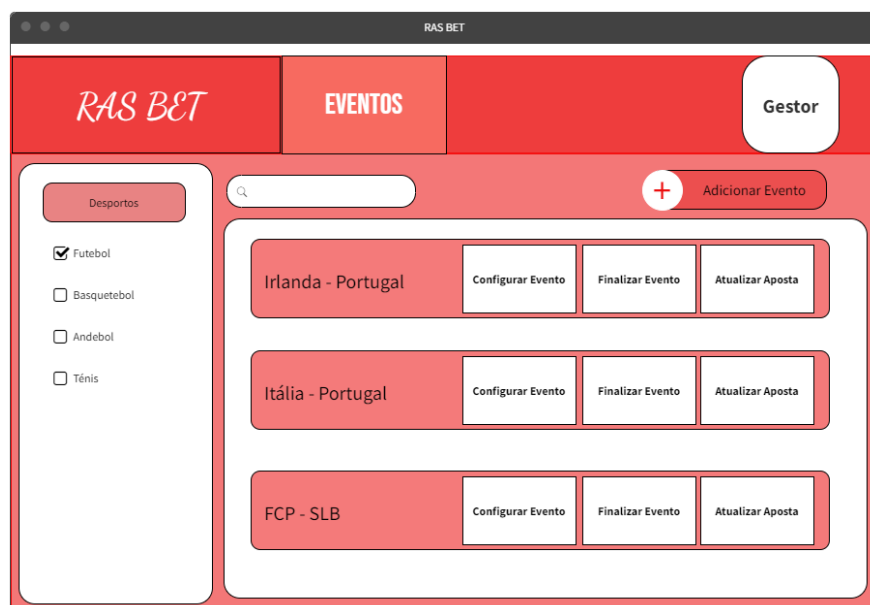


Figura 8: *Mockup* relativo ao menu do gestor de apostas.

Na figura 9 conseguimos ver a interface do menu que o gestor de apostas vai ter acesso quando este pretender atualizar os eventos, isto é, quando o gestor de apostas pretender atualizar *odds* do evento. Para alterar as *odds* o gestor de apostas tem escrever qual a *odd* a alterar (nome - 'vitória equipa 1', 'empate' e 'vitória equipa 2') e o seu novo valor (*odd* atualizada). Com esta *mockup* conseguimos ver as atividades relacionada com a escolha do resultado possível para alterar a *odd* e a inserção da nova *odd* presente no diagrama da figura 6. A atividade de calcular *odd* é realizada pelo gestor fora do contexto do sistema a implementar, daí não estar representada na *mockup*.

A mockup da interface do sistema RAS BET, intitulada 'RAS BET' no topo. O cabeçalho principal é dividido em três seções: 'RAS BET' (logotipo), 'EVENTOS' e 'Gestor'. Abaixo do cabeçalho, há uma barra lateral esquerda com o título 'Desportos' e uma lista de opções: 'Futebol' (selecionada com um ícone de caixa de seleção marcada), 'Basquetebol', 'Andebol' e 'Tênis'. No topo da área principal, há uma barra de pesquisa e um botão '+ Adicionar Evento'. O conteúdo principal da interface é focado no evento 'Irlanda - Portugal'. Dentro desta seção, há um formulário com dois campos de entrada: 'Nome:' e 'Odd Atualizada:'. Ao lado do campo 'Nome:', há um ícone de lupa. Ao lado do campo 'Odd Atualizada:', há um botão '+ Adicionar Resultado'. Abaixo do formulário, há um botão vermelho proeminente com o texto 'Atualizar Odd'.

Figura 9: *Mockup* relativo ao use case de 'Atualizar Odd'.

Na figura 10 conseguimos ver a interface gráfica do menu que o gestor de apostas vai ter acesso quando este pretender adicionar novos eventos. Para os adicionar é necessário que o gestor de apostas insira parâmetros como o nome e data do evento, hora de início, equipas e as diversas *odds*. Com esta *mockup* conseguimos ver a atividade relacionada com a inserção dos parâmetros para adicionar um novo evento tal como está presente no diagrama da figura 6. As restantes atividades que fazem parte do fluxo de adicionar evento são feitas internamente pelo sistema e por isso não são apresentadas na *mockup*.

A mockup apresenta a interface de um navegador com o título 'RAS BET'. O cabeçalho é dividido em três seções: à esquerda, o logótipo 'RAS BET' em uma barra vermelha; no centro, o menu 'EVENTOS' em uma barra vermelha; e à direita, um botão 'Gestor' em uma barra vermelha com um círculo branco. Abaixo do cabeçalho, há uma barra lateral esquerda com o título 'Desportos' e uma lista de opções: 'Futebol' (selecionada com um ícone de caixa de seleção marcada), 'Basquetebol', 'Andebol' e 'Ténis'. À direita da barra lateral, há uma barra de pesquisa e um botão '+ Adicionar Evento'. O formulário principal, que ocupa a maior parte da tela, contém campos para 'Nome do Evento', 'Data do Evento' e 'Hora de Início'. Abaixo desses campos, há uma seção para adicionar equipas, com um campo 'Equipa/Participante 1' e um botão '+ Adicionar Equipa'. Segue-se uma seção para adicionar acontecimentos, com campos para 'Nome 1' e 'Odd 1', e um botão '+ Adicionar Acontecimento'. No final do formulário, há um botão vermelho 'Criar Evento'.

Figura 10: *Mockup* relativo ao use case de 'Adicionar Evento'.

Na figura 11 conseguimos ver a interface do menu que o gestor de apostas vai ter acesso quando este pretender finalizar um evento. Para que isso aconteça é necessário que o gestor de apostas insira parâmetros como o nome de cada participante e respetivo resultado final. Caso o evento seja cancelado por algum motivo externo, o gestor de apostas deve preencher o campo 'motivo'. Com esta *mockup* conseguimos ver as atividade relacionada com a inserção do resultado final no fim do evento e com a inserção do motivo para o cancelamento do evento, tal como está presente no diagrama da figura 6.

A mockup da interface de 'Finalizar Evento' no RAS BET apresenta o seguinte layout:

- Barra Superior:** Contém o logo 'RAS BET', o menu 'EVENTOS' e um botão 'Gestor'.
- Lado Esquerdo:** Um menu 'Desportos' com opções: Futebol (selecionado), Basquetebol, Andebol e Ténis.
- Área Central:**
 - Um campo de busca.
 - Um botão '+ Adicionar Evento'.
 - Um formulário para o evento 'Irlanda - Portugal' com campos para 'Motivo:', 'Nome 1:', e 'Resultado 1:'.
 - Um botão '+ Adicionar Resultados' ao lado dos campos de nome e resultado.
 - Um botão 'Finalizar Evento' na base.

Figura 11: *Mockup* relativo ao use case de 'Finalizar Evento'.

Na figura 12 conseguimos ver a interface do histórico de apostas e notificações recebidas pelo utilizador. Ao terminar um evento, os utilizadores que apostaram nesse evento são notificados quanto ao resultado do evento e os ganhos. Na seguinte figura podemos observar as notificações não lidas, destacadas num tom acinzentado e as restantes denotam as notificações já lidas. Além disso, as apostas destacadas a verde são as ganhas, e as coloridas a vermelho são as perdas. Com esta *mockup* conseguimos ver a atividade de notificação dos apostadores após o fim ou cancelamento do evento, tal como está presente no diagrama da figura 6. As restantes atividades de atualizar para o estado "Fechado" e de atribuição de quantias são realizadas internamente pelo sistema e por isso não fazem parte das *mockups*.

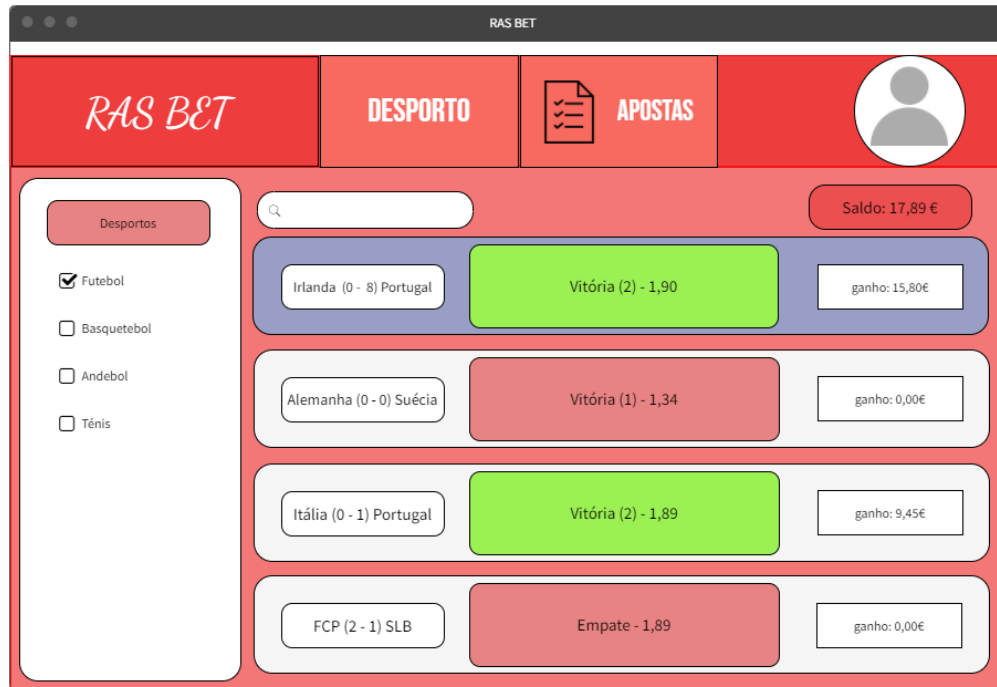


Figura 12: *Mockup* relativo ao use case de 'Finalizar Evento'.

Na figura 13 conseguimos ver a interface do menu que o gestor de apostas vai ter acesso quando este pretender atualizar uma aposta, quer seja para colocar o seu estado em "Aberto" ou em "Suspense". Para que isso aconteça é necessário que o gestor de apostas insira parâmetros como o nome da aposta e selecione o respetivo estado. Caso a aposta possua o estado "Aberto", o gestor pode alterar o seu estado para "Suspense". Caso possua o estado "Suspense", então este pode ser modificado para "Aberto". Com esta *mockup* conseguimos ver a atividade relacionada com a seleção do estado a atualizar, tal como está presente no diagrama da figura 6. As restantes atividades ligadas ao fluxo de atualização das apostas são implementadas internamente no sistema e por isso não fazem parte desta *mockup*.

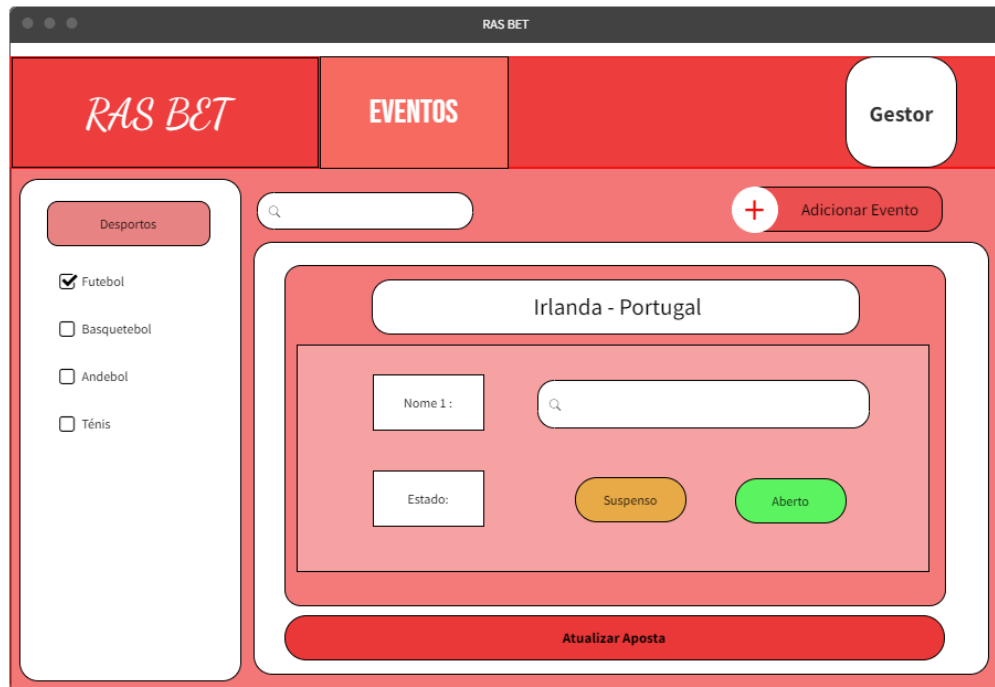


Figura 13: *Mockup* relativo ao use case de 'Atualizar Aposta'.

9. Requisitos Funcionais e de Dados

- **Requisito 1 - Autenticação**

Requisito #: **RF01**

Tipo de Requisito: **Funcional**

Use Case #: **UC01,UC05-UC11**

Descrição: **Utilizador deve poder efetuar *login* na App.**

Raciocínio: **É necessário ter um sistema de autenticação que permita ao utilizador aceder à aplicação para efetuar apostas.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Na autenticação é necessário indicar o *email* e *password* e deve estar de acordo com a informação contida na base de dados.**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **3**

Conflitos: **-**

Materiais de Apoio: **-**

Histórico: **Criado em Novembro 2021**

• Requisito 2 - Registar Utilizador

Requisito #: RF02	Tipo de Requisito: Funcional	Use Case #: UC02
Descrição: Utilizador deve conseguir registar-se na aplicação.		
Raciocínio: É necessário ter um sistema de registo que permita ao utilizador registar-se no sistema através de um <i>email</i>, <i>password</i>, nome, data de nascimento e saldo na U.M. devida para o utilizador poder efetuar apostas na aplicação.		
Originador: Engenheiro de Software		
Critério de Aptidão: Os dados inseridos devem ser válidos (<i>email</i>, data de nascimento e saldo).		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• Requisito 3 - Log-out

Requisito #: RF03	Tipo de Requisito: Funcional	Use Case #: UC03
Descrição: Utilizador deve conseguir fazer <i>log-out</i> da aplicação.		
Raciocínio: É necessário que o utilizador possa sair da aplicação por motivos de privacidade e segurança.		
Originador: Engenheiro de Software		
Critério de Aptidão: Caso o utilizador faça <i>log-out</i>, não poderá aceder às informações da sua conta.		
Satisfação do Consumidor: 4	Insatisfação do Consumidor: 4	
Prioridade: 4	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 4 - Consultar eventos**

Requisito #:	RF04	Tipo de Requisito:	Funcional	Use Case #:	UC04
Descrição:	Utilizador deve conseguir consultar os eventos disponibilizados.				
Raciocínio:	É necessário ter um sistema que permita ao utilizador navegar pela aplicação, mesmo não tendo conta.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	Os eventos disponibilizados devem possuir o estado "Aberto".				
Satisfação do Consumidor:	5	Insatisfação do Consumidor:	5		
Prioridade:	some more text	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

• **Requisito 5 - Aceder à área pessoal**

Requisito #:	RF05	Tipo de Requisito:	Funcional	Use Case #:	UC05-UC11
Descrição:	Apostador pode aceder à sua área pessoal.				
Raciocínio:	É necessário ter um sistema que permita ao utilizador aceder a várias informações e configurações relativas à sua conta.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	A informação relativa ao utilizador deve estar associada ao seu <i>email</i> e de acordo com a base de dados.				
Satisfação do Consumidor:	4	Insatisfação do Consumidor:	4		
Prioridade:	4	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro de 2021				

- **Requisito 6 - Consultar histórico apostas**

Requisito #: RF06	Tipo de Requisito: Funcional	Use Case #: UC05
Descrição: Apostador pode consultar o histórico de apostas realizadas.		
Raciocínio: É necessário ter um sistema que permita ao utilizador rever o histórico das suas apostas para que possa observar informações das mesmas.		
Originador: Engenheiro de Software		
Critério de Aptidão: As informações disponibilizadas devem ser relativas ao utilizador em questão.		
Satisfação do Consumidor: 4	Insatisfação do Consumidor: 4	
Prioridade: 4	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

- **Requisito 7 - Consultar notificações**

Requisito #: RF07	Tipo de Requisito: Funcional	Use Case #: UC06
Descrição: Apostador deve ter notificações recebidas.		
Raciocínio: É necessário que o sistema permita ao utilizador ver as notificações quando este as recebe		
Originador: Engenheiro de Software		
Critério de Aptidão: O utilizador sempre que recebe uma notificação deve receber uma mensagem e a partir dela listar a notificação		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 2	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

• Requisito 8 - Alterar Configuração Conta

Requisito #: RF08	Tipo de Requisito: Funcional	Use Case #: UC07
Descrição: Apostador pode alterar configurações de conta.		
Raciocínio: É necessário que o sistema permita ao utilizador alterar <i>email, password</i> e nome.		
Originador: Engenheiro de Software		
Critério de Aptidão: A conta após a configuração tem de estar atualizada.		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

• Requisito 9 - Levantar Saldo

Requisito #: RF09	Tipo de Requisito: Funcional	Use Case #: UC08
Descrição: Apostador deve conseguir levantar dinheiro da conta.		
Raciocínio: É necessário que o sistema permita ao utilizador inserir quantia a levantar da conta.		
Originador: Engenheiro de Software		
Critério de Aptidão: O saldo do utilizador deve ficar menor do anterior.		
Satisfação do Consumidor: 4	Insatisfação do Consumidor: 4	
Prioridade: 4	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

- **Requisito 10 - Depositar Saldo**

Requisito #: RF10	Tipo de Requisito: Funcional	Use Case #: UC09
Descrição: Apostador deve conseguir depositar dinheiro da conta.		
Raciocínio: É necessário que o sistema permita ao utilizador inserir quantia a depositar na conta.		
Originador: Engenheiro de Software		
Critério de Aptidão: O saldo do utilizador deve ficar maior do anterior.		
Satisfação do Consumidor: 4	Insatisfação do Consumidor: 4	
Prioridade: 4	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

- **Requisito 11 - Consultar estatísticas**

Requisito #: RF11	Tipo de Requisito: Funcional	Use Case #: UC10
Descrição: Apostador deve conseguir consultar as estatísticas da conta.		
Raciocínio: É necessário ter um sistema em que o apostador possa consultar a evolução do seu saldo, desportos mais apostados, percentagem de apostas vencidas e perdas e dinheiro gasto ou perdido.		
Originador: Engenheiro de Software		
Critério de Aptidão: Devem ser listadas informações sobre as apostas realizados pelo apostador e que já foram "fechadas".		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

• Requisito 12 - Atualizar Resultado

Requisito #: RF12	Tipo de Requisito: Funcional	Use Case #: UC11, UC16
Descrição: O gestor de apostas deve conseguir atualizar o resultado do evento desportivo.		
Raciocínio: É necessário ter um sistema em que o gestor de apostas possa alterar o resultado de em evento, no fim do jogo ou no decorrer deste.		
Originador: Engenheiro de Software		
Critério de Aptidão: Os utilizadores devem ter acesso ao resultado atualizado do jogo		
Satisfação do Consumidor: 4	Insatisfação do Consumidor: 3	
Prioridade: 4	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro de 2021		

• Requisito 13 - Caracterização da aposta

Requisito #: RF13	Tipo de Requisito: Funcional	Use Case #: UC12
Descrição: Uma aposta está caracterizada por evento, quantia, ganhos possíveis e resultado possível.		
Raciocínio: É necessário que um sistema onde o apostador possa indicar quantia apostada, resultado possível num certo evento e que possa ver quais os ganhos possíveis com a aposta.		
Originador: Engenheiro de Software		
Critério de Aptidão: A aposta deve estar armazenada no sistema com os parâmetros definidos		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 4	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

- **Requisito 14 - Registrar aposta**

Requisito #:	RF14	Tipo de Requisito:	Funcional	Use Case #:	UC12
Descrição:	Apostador deve conseguir registrar aposta no sistema.				
Raciocínio:	É necessário ter um sistema que permita ao apostador registrar nova apostas no sistema, uma vez que esse é o objetivo principal do sistema				
Originador:	Engenheiro de Software				
Critério de Aptidão:	A aposta deve ficar armazenada com estado "Aberta" no sistema.				
Satisfação do Consumidor:	5	Insatisfação do Consumidor:	5		
Prioridade:	5	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

- **Requisito 15 - Alterar odds evento**

Requisito #: RF15	Tipo de Requisito: Funcional	Use Case #: UC13
Descrição: Gestor de apostas pode alterar as <i>odds</i> do evento.		
Raciocínio: É necessário ter um sistema que permita ao gestor de apostas atualizar as <i>odds</i> de acordo com os desenrolar do evento e das apostas.		
Originador: Engenheiro de Software		
Critério de Aptidão: Quando um utilizador consulta os eventos, as <i>odds</i> devem estar sempre atualizadas.		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

- **Requisito 16 - Adicionar evento**

Requisito #: RF16	Tipo de Requisito: Funcional	Use Case #: UC14
Descrição: Gestor de apostas pode adicionar novo evento ao sistema.		
Raciocínio: É necessário ter um sistema que permita ao gestor introduzir novos eventos sob os quais se vão basear as apostas.		
Originador: Engenheiro de Software		
Critério de Aptidão: Ao ser adicionado um evento, as apostas associadas a este têm de ter estado "Aberto" e devem estar visível a todos os utilizadores do sistema.		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

- **Requisito 17 - Caracterização do evento**

Requisito #: RF17	Tipo de Requisito: Funcional	Use Case #: UC14
Descrição: Um evento deve ser caracterizado com nome, equipa ou lista de participantes, data e hora e odds.		
Raciocínio: É necessário ter um sistema que permita ao gestor indicar o nome, participantes, data e hora e as odds de um evento.		
Originador: Engenheiro de Software		
Critério de Aptidão: O evento deve ficar armazenado no sistema com os seguintes parâmetros.		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 18 - Atualizar estado aposta**

Requisito #:	RF18	Tipo de Requisito:	Funcional	Use Case #:	UC15
Descrição:	Gestor de apostas pode atualizar estados de uma aposta.				
Raciocínio:	É necessário ter um sistema que permita alterar o estado das apostas de um evento para permitir o fluxo do programa.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	Deve ser possível abrir ou suspender apostas de um evento.				
Satisfação do Consumidor:	5	Insatisfação do Consumidor:	5		
Prioridade:	5	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

• **Requisito 19 - Receber notificação**

Requisito #: RF19	Tipo de Requisito: Funcional	Use Case #: UC16
Descrição: Utilizador deve receber notificação sobre o término de uma aposta.		
Raciocínio: É necessário ter um sistema que informe o utilizador do término das suas apostas, seja pelo fim de um evento ou pelo cancelamento do mesmo.		
Originador: Engenheiro de Software		
Critério de Aptidão: Deve ser possível notificar um utilizador através de notificações.		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 20 - Receber quantia**

Requisito #: RF20	Tipo de Requisito: Funcional	Use Case #: UC16
Descrição: O utilizador deve receber a devida quantia associada ao resultado da sua aposta.		
Raciocínio: É necessário ter um sistema que envie aos utilizadores o valor ganho (+ odd * quantia apostada, caso ganhe ou +0, caso perca).		
Originador: Engenheiro de Software		
Critério de Aptidão: O saldo relativo ao utilizador deve ser atualizado no sistema.		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 21 - Fechar apostas**

Requisito #: RF21	Tipo de Requisito: Funcional	Use Case #: UC16
Descrição: Gestor de apostas deve conseguir fechar as apostas.		
Raciocínio: É necessário ter um sistema que permita ao gestor de apostas finalizar o evento (caso o evento tenha decorrido como previsto) ou cancelar evento (caso o evento tenha sido cancelado por um motivo externo).		
Originador: Engenheiro de Software		
Critério de Aptidão: O evento a que as apostas estão associadas deve ficar indisponível aos utilizadores e os apostadores que apostaram no evento devem receber a devida notificação (fim de jogo ou cancelamento de evento).		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

Requisitos Não-Funcionais

10. Aparência

Esta secção está relacionada com a identidade visual do sistema, como logótipos, temas e cores usados. Abaixo apresentamos os requisitos de aparência que foram considerados para o sistema.

- **Requisito 22 - Atrativo a fãs de desporto**

Requisito #: **RNF22**

Tipo de Requisito: **Aparência** Use Case #: -

Descrição: **O sistema deve ser atrativo a fãs de desporto.**

Raciocínio: **É necessário ter um sistema que seja visualmente atrativo para fãs de desporto, uma vez que eles serão o publico-alvo da aplicação.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **A aparência do sistema deverá estar avaliada acima de 3.5 estrelas pelos utilizadores do sistema.**

Satisfação do Consumidor: **4**

Insatisfação do Consumidor: **4**

Prioridade: **4**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

• **Requisito 23 - Look identificável**

Requisito #: RNF23	Tipo de Requisito: Aparência	Use Case #: -
Descrição: O sistema deve ter um <i>look</i> identificável e diferente de produtos semelhantes.		
Raciocínio: É necessário ter um sistema que marque pela diferença para ser capaz de se destacar dos demais sites de apostas existentes.		
Originador: Engenheiro de Software		
Critério de Aptidão: Os utilizadores devem avaliar o <i>look</i> do sistema em 4 estrelas numa escala de 1 a 5 sendo 1 pouco identificável e 5 muito identificável		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 24 - Representação do logótipo**

Requisito #: RNF24	Tipo de Requisito: Aparência	Use Case #: -
Descrição: O sistema deve ter representado o logótipo da empresa nas diversas plataformas.		
Raciocínio: É necessário ter um logótipo para representar a empresa e para identificar facilmente de que empresa é o sistema.		
Originador: Engenheiro de Software		
Critério de Aptidão: O logótipo deve estar presente em todos os menus.		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

11. Usabilidade

Esta secção inclui requisitos relacionados com a facilidade de uso, personalização, internacionalização, facilidade de aprendizagem, compreensão e acessibilidade. De seguida, apresentamos os requisitos de usabilidade que foram considerados para o sistema.

- **Requisito 25 - Suporte de várias moedas**

Requisito #: **RNF25**

Tipo de Requisito: **Usabilidade** Use Case #: -

Descrição: **O sistema deve suportar várias moedas (euro, dólar, etc.)**

Raciocínio: **É necessário ter um sistema que suporte várias moedas de modo a conseguir suportar utilizadores de diferentes regiões.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O utilizador deve conseguir consultar o site de acordo com a moeda que este escolheu nas definições do sistema.**

Satisfação do Consumidor: **2**

Insatisfação do Consumidor: **2**

Prioridade: **Engenheiro de Software**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

• **Requisito 26 - Suporte a vários meios de pagamento**

Requisito #:	RNF26	Tipo de Requisito:	Usabilidade	Use Case #:	-
Descrição:	O sistema deve suportar vários meios de pagamento.				
Raciocínio:	É necessário ter um sistema que suporta vários meios diferentes de pagamento como o <i>Paypal</i>, <i>MBWay</i>, <i>Multibanco</i>, entre outros.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	Os utilizadores devem ter acesso pelo menos 5 métodos de pagamento.				
Satisfação do Consumidor:	4	Insatisfação do Consumidor:	4		
Prioridade:	4	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

• **Requisito 27 - Suporte a várias línguas**

Requisito #: RNF27	Tipo de Requisito: Usabilidade	Use Case #: -
Descrição: O sistema deve suportar várias línguas.		
Raciocínio: É necessário ter um sistema que suporte várias línguas diferentes como português, inglês, espanhol, entre outros de modo a maximizar o número de utilizadores que podem usar a aplicação.		
Originador: Engenheiro de Software		
Critério de Aptidão: O utilizador deve conseguir consultar o site de acordo com a língua que este escolheu nas definições do sistema.		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 28 - Pessoas não familiarizadas com sites de apostas**

Requisito #:	RNF28	Tipo de Requisito:	Usabilidade	Use Case #:	-
Descrição:	O sistema deve ser intuitivo para pessoas que não estão familiarizadas com sites de apostas.				
Raciocínio:	É necessário um sistema de modo a garantir que potenciais apostadores não sejam impossibilitados de usar aplicação caso não estejam habituados a usar aplicações informáticas.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	Pelo menos 70% de pessoas pouco familiarizadas com apostas devem conseguir realizar mais do que uma aposta no sistema em menos de 5 minutos.				
Satisfação do Consumidor:	4	Insatisfação do Consumidor:	4		
Prioridade:	4	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

• **Requisito 29 - Time-zone dos eventos**

Requisito #: RNF29	Tipo de Requisito: Usabilidade	Use Case #: -
Descrição: O sistema deve converter as horas dos eventos para a <i>time-zone</i> do utilizador.		
Raciocínio: É necessário um sistema converta o horário do evento de modo a que o utilizador consiga utilizar facilmente o sistema em várias localizações do globo.		
Originador: Engenheiro de Software		
Critério de Aptidão: O utilizador deve consultar os eventos com a data e hora associada à <i>time-zone</i> onde este se encontra.		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

12. Performance

Nesta secção englobamos os requisitos relacionados com o tempo de processamento das tarefas, tempos de resposta, a precisão dos resultados, a precisão dos dados, a disponibilidade, a robustez, tolerância a faltas, capacidade de armazenamento e a escalabilidade. Abaixo apresentamos os requisitos de performance que foram considerados para o sistema.

- **Requisito 30 - Nível de precisão**

Requisito #: RNF30	Tipo de Requisito: Performance	Use Case #: -
Descrição: O sistema deve ter um alto nível de precisão no cálculo dos ganhos possíveis, nas transações bancárias e odds.		
Raciocínio: É necessário um sistema que garanta a integridade das transferências financeiras e do cálculo correto das odds e ganhos possíveis.		
Originador: Engenheiro de Software		
Critério de Aptidão: O sistema deve apresentar todos os cálculos e quantias na ordem dos centimos e as odds na ordem das centésimas.		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 31 - Atualização rápida de informações**

Requisito #: RNF31	Tipo de Requisito: Performance	Use Case #: -
Descrição: O sistema deve atualizar rapidamente as <i>odds</i> e resultado dos eventos.		
Raciocínio: É necessário um sistema onde o utilizador consulta as <i>odds</i> e resultados sempre atualizadas devido às possíveis alterações das <i>odds</i> e resultados no decorrer do evento.		
Originador: Engenheiro de Software		
Critério de Aptidão: Ao fim de uma mudança, as <i>odds</i> ou os resultados deverão ser atualizadas em menos de 1 segundo.		
Satisfação do Consumidor: 2	Insatisfação do Consumidor: 2	
Prioridade: 2	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 32 - Uptime Alto**

Requisito #: RNF32	Tipo de Requisito: Performance	Use Case #: -
Descrição: O sistema deverá ter um <i>uptime</i> alto.		
Raciocínio: É necessário que o sistema tenha um <i>uptime</i> alto visto que vão haver eventos ativos em quase todas as horas do dia.		
Originador: Engenheiro de Software		
Critério de Aptidão: O sistema deverá ter uma taxa de <i>uptime</i> de 98% por ano.		
Satisfação do Consumidor: 3	Insatisfação do Consumidor: 3	
Prioridade: 3	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

• **Requisito 33 - Armazenamento dos dados**

Requisito #:	RNF33	Tipo de Requisito:	Performance	Use Case #:	-
Descrição:	O sistema deverá guardar e processar os dados de grandes quantidades de utilizadores sem comprometer o funcionamento normal do sistema.				
Raciocínio:	É necessário que o sistema consiga acompanhar o crescimento da população de utilizadores.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	O sistema não deverá ter um aumento superior a 30% nos tempos de resposta quando existe 0 a 300 utilizadores ativos no sistema.				
Satisfação do Consumidor:	3	Insatisfação do Consumidor:	3		
Prioridade:	3	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

• **Requisito 34 - Crescimento no número de utilizadores**

Requisito #:	RNF34	Tipo de Requisito:	Performance	Use Case #:	-
Descrição:	A população do sistema deverá apresentar um crescimento contínuo durante os primeiros 3 anos.				
Raciocínio:	Este crescimento é necessário para justificar os custos de desenvolvimento do sistema.				
Originador:	Engenheiro de Software				
Critério de Aptidão:	O sistema deverá crescer até 300 utilizadores ao fim de 3 anos.				
Satisfação do Consumidor:	4	Insatisfação do Consumidor:	4		
Prioridade:	4	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

13. Operacional

Os requisitos operacionais estão relacionados com o contexto (ambiental ou tecnológico) no qual o sistema irá operar. Abaixo apresentamos os requisitos operacionais que foram considerados para o sistema.

- **Requisito 35 - Formato *WEB***

Requisito #:	RNF35	Tipo de Requisito:	Operacional	Use Case #:	-
Descrição:	O sistema deve estar disponível em formato <i>web</i>.				
Raciocínio:	É necessário ter um sistema que esteja disponível em vários navegadores <i>web</i> de forma a que o sistema abranja o maior número de utilizadores possível.				
Originador:	Engenheiro de <i>Software</i>				
Critério de Aptidão:	O sistema deve funcionar em pelo menos quatro dos cinco navegadores mais populares (por exemplo: <i>Chrome, Firefox, Brave, Safari, Microsoft Edge</i>).				
Satisfação do Consumidor:	5	Insatisfação do Consumidor:	5		
Prioridade:	5	Conflitos:	-		
Materiais de Apoio:	-				
Histórico:	Criado em Novembro 2021				

• **Requisito 36 - Formato aplicação**

Requisito #: **RNF36**

Tipo de Requisito: **Operacional** Use Case #: -

Descrição: **O sistema deverá estar disponível em lojas de aplicações eletrônicas.**

Raciocínio: **É necessário ter um sistema acessível aos utilizadores que usam aplicações *mobile*.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deverá estar disponível nas 2 maiores lojas de aplicações eletrônicas (*Play Store e App Store*).**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **5**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

14. Manutenção e suporte

Os requisitos de manutenção estão relacionados com a manutenção e reparação do sistema, isto é, requisitos que permitem manter o sistema atualizado e dar apoio aos utilizadores no uso da aplicação. Abaixo apresentamos os requisitos de manutenção e suporte que foram considerados para o sistema.

- **Requisito 37 - Suporte ao utilizador**

Requisito #: **RNF37**

Tipo de Requisito: **Manutenção e suporte** Use Case #: -

Descrição: **O sistema deve disponibilizar mecanismos de suporte ao utilizador**

Raciocínio: **É necessário um sistema que permita aos utilizadores terem acesso a informação sobre o funcionamento do sistema e como resolver problemas técnicos.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deve disponibilizar uma secção de "Perguntas mais Frequentes" e um contacto caso os utilizadores necessitam de suporte técnico.**

Satisfação do Consumidor: **4**

Insatisfação do Consumidor: **4**

Prioridade: **4**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

• **Requisito 38 - Atualização do sistema**

Requisito #: **RNF38**

Tipo de Requisito: **Manutenção** Use Case #: -
e suporte

Descrição: **O sistema deve permitir atualizações mensais do sistema em alturas com pouco tráfego no sistema.**

Raciocínio: **É necessário que o sistema seja atualizado e que essas manutenções aconteçam em alturas com pouco movimento para que não haja impacto no uso do sistema pelos utilizadores.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Deve-se avaliar ao longo de 1 ano quais as 8 horas mensais onde o sistema tem um tráfego inferior a 30% de utilização e nessas alturas atualizar o sistema.**

Satisfação do Consumidor: **3**

Insatisfação do Consumidor: **3**

Prioridade: **3**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

- **Requisito 39 - Resolução de Erros**

Requisito #: **RNF39**

Tipo de Requisito: **Manutenção** Use Case #: -
e suporte

Descrição: **O sistema deve permitir a resolução de erros de forma eficaz.**

Raciocínio: **É necessário que o sistema resolva os erros para que este não contenha *bugs* e que funcione de forma correta.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Deve-se disponibilizar 3h semanais para a resolução de erros relacionados com informação, funcionamento ou outro tipo de problemas.**

Satisfação do Consumidor: **3**

Insatisfação do Consumidor: **3**

Prioridade: **3**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

15. Segurança

Nesta secção abordamos os requisitos relacionados com a confidencialidade, integridade e privacidade dos dados, assim como a resistência a ataques de acesso à informação dos utilizadores. De seguida, apresentamos os requisitos de segurança considerados para o sistema.

• Requisito 40 - Informação Errada

Requisito #: **RNF40**

Tipo de Requisito: **Segurança** Use Case #: -

Descrição: **O sistema deve rejeitar informação incorreta de ser armazenada.**

Raciocínio: **É necessário que o sistema garanta a integridade dos dados no sistema.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Não deverá ser possível ao utilizador usar as funcionalidades do sistema sem ter verificado o email.**

Satisfação do Consumidor: **3**

Insatisfação do Consumidor: **3**

Prioridade: **3**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

- **Requisito 41 - Encriptação da informação**

Requisito #: **RNF41**

Tipo de Requisito: **Segurança** Use Case #: -

Descrição: **O sistema deverá assegurar a segurança das informações dos utilizadores.**

Raciocínio: **É necessário um sistema que garanta a segurança dos dados confidenciais de cada utilizador.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Os dados dos utilizadores do sistema deverão estar todos encriptados na base de dados.**

Satisfação do Consumidor: **4**

Insatisfação do Consumidor: **4**

Prioridade: **4**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

- **Requisito 42 - Políticas de privacidade**

Requisito #: **RNF42**

Tipo de Requisito: **Segurança** Use Case #: -

Descrição: **O sistema deverá notificar os utilizadores da mudança nas políticas de privacidade.**

Raciocínio: **É necessário um sistema que informe os utilizadores da mudança nas políticas de privacidade para que estes tomem conhecimento da maneira como os dados inseridos no site estão a ser usados e para que motivos.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deve lançar uma mensagem de aviso após o registo de um novo utilizador com informação sobre as políticas de privacidade para que este consinta com as políticas.**

Satisfação do Consumidor: **4**

Insatisfação do Consumidor: **4**

Prioridade: **4**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

• **Requisito 43 - Gerir Informação**

Requisito #: **RNF43**

Tipo de Requisito: **Segurança** Use Case #: -

Descrição: **O sistema deverá garantir que só os gestores de apostas é que gerem as odds e os eventos.**

Raciocínio: **É necessário um sistema que garanta a integridade do sistema de apostas e evitar manipulação da informação do sistema por terceiros.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Todas as informações dos eventos e das odds do sistema deverão estar de acordo com o log de atualizações propostas pelos gestores.**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **5**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

16. Cultural e político

Os requisitos culturais vão conforme características de um grupo particular definido por língua, religião, cozinha, hábitos sociais, arte, entre outros. Os requisitos políticos são especificados por fatores relacionados com estratégias e organizações que afetam o sistema. De seguida, apresentam-se os requisitos culturais e políticos que foram considerados para o sistema.

- **Requisito 44 - Casa de apostas legal**

Requisito #: **RNF44**

Tipo de Requisito: **Cultural e político** Use Case #: -

Descrição: **O sistema deve estar registado como uma casa de apostas legal na SRIJ**

Raciocínio: **É necessário que o sistema esteja registado como casa de apostas legal na SRIJ para que os utilizadores possam utilizar de forma legal o sistema.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **O sistema deve estar presente na lista de entidades licenciadas da SRIJ.**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **5**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

17. Legal

Esta secção trata dos requisitos que estejam em conformidade com as leis estabelecidas, neste caso, de acordo com leis relacionadas com o mundo das apostas. De seguida, apresentam-se os requisitos legais que foram considerados para o sistema.

- **Requisito 45 - Maiores de 18 anos**

Requisito #: **RNF45**

Tipo de Requisito: **Legal**

Use Case #: -

Descrição: **O sistema não deve ser utilizado por indivíduos menores do que 18 anos.**

Raciocínio: **É necessário que o sistema cumpra com as leis relativas às apostas *online*.**

Originador: **Engenheiro de Software**

Critério de Aptidão: **Não deverão haver contas registadas no sistema cuja data de nascimento implica uma idade menor que 18 anos.**

Satisfação do Consumidor: **5**

Insatisfação do Consumidor: **5**

Prioridade: **5**

Conflitos: -

Materiais de Apoio: -

Histórico: **Criado em Novembro 2021**

• **Requisito 46 - Leis de apostas**

Requisito #: RNF46	Tipo de Requisito: Legal	Use Case #: -
Descrição: O sistema deve estar de acordo com a leis promovidas pela <i>European Gaming and Betting Association</i>		
Raciocínio: É necessário que o sistema esteja de acordo com as leis da EGBA para que os ganhos dos utilizadores sejam legais e para que o sistema consiga estar disponível na <i>WEB</i> como uma casa de apostas e não cometa ilegalidades.		
Originador: Engenheiro de Software		
Critério de Aptidão: A atualização das <i>odds</i>, os ganhos de cada utilizador assim como todo o funcionamento do sistema deve promover o <i>fair gaming</i>, combater comportamentos fraudulentos, proteger utilizadores vulneráveis, entre outros.		
Satisfação do Consumidor: 5	Insatisfação do Consumidor: 5	
Prioridade: 5	Conflitos: -	
Materiais de Apoio: -		
Histórico: Criado em Novembro 2021		

Resumo dos Requisitos

Nesta secção, apresentamos em forma tabular uma associação entre os requisitos levantados, isto é, entre os requisitos funcionais e não funcionais. Como podemos ver na tabela seguinte, cada *Use Case* surge representado pelo seu identificador, **UCxx**, e a cada *Use Case* estão associados os seus respetivos requisitos funcionais, identificados por **RFxx**.

Também os requisitos não funcionais se encontram agrupados, neste caso de acordo com as categorias de classificação existentes : Aparência, Usabilidade, Performance, Operacional, Manutenção e Suporte, Cultural e Politico, Segurança e Legal. Analogamente aos requisitos funcionais, cada requisito não funcional será também representado pelo seu identificador, neste caso **RNFxx**.

Assim, através do uso de identificadores, conseguimos criar uma tabela de associações entre requisitos funcionais e não funcionais. Além disso, uma vez que todos os requisitos se encontram caracterizados pela tabela de *Volere* para consultar a sua descrição detalhada bastará procurar pelo seu identificador nas respetivas tabelas de *Volere*.

Use Cases	Requisitos Funcionais	Requisitos Não Funcionais							
		Aparência	Usabilidade	Performance	Operacional	Manutenção e Suporte	Segurança	Cultural e Politico	Legal
UC01	RF01	RNF24					RNF43		
UC02	RF02	RNF24					RNF40 RNF41	RNF45	
UC03	RF03								
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC04	RF04	Todos	RNF27 RNF28 RNF29	RNF30					
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC05	RF05	RNF24	RNF27						
	RF06		RNF28	RNF30					

	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC06	RF07		RNF28						
	RF05	RNF24	RNF27						
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC07	RF08						RNF40 RNF41		
	RF05	RNF24	RNF27						
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC08	RF09		RNF25 RNF26 RNF28	RNF30					
	RF05	RNF24	RNF27						
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC09	RF10		RNF25 RNF26 RNF28	RNF30					
	RF05	RNF24	RNF27						
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC10	RF11		RNF28	RNF30					
	RF05	RNF24	RNF27						
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC11	RF12			RNF31			RNF43		
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC12	RF13		RNF25	RNF30					
	RF14	Todos	RNF25 RNF28	RNF30					

	RF04	Todos	RNF27 RNF28 RNF29	RNF30					
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC13	RF15			RNF30 RNF31			RNF43		
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC14	RF16			RNF30			RNF43		
	RF17			RNF30					
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC15	RF18						RNF43		
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		
UC16	RF19		RNF28	RNF30			RNF43		
	RF20		RNF25	RNF30					
	RF21			RNF31			RNF43		
	RF12			RNF31			RNF43		
	RF02	RNF24					RNF40 RNF41	RNF45	
	RF01	RNF24					RNF43		

Conclusão

Dada por concluída a primeira etapa do projeto, fará sentido apresentar uma visão crítica, refletida e ponderada do trabalho realizado.

No espetro positivo, consideramos relevante destacar o facto do documento de requisitos ser representativo e completo do domínio do problema, indicando os seus requisitos funcionais e não funcionais, e é também um bom guia para a futura implementação do programa.

Por outro lado, em termos de melhorias ou *upgrades* no nosso projeto, consideramos que seria benéfico aumentar o domínio do nosso problema para abranger o domínio das transferências bancárias e implementar todos os requisitos não funcionais, caso o projeto fosse desenvolvido no mundo real invés do contexto académico.

Em suma, consideramos que o balanço do trabalho é positivo, as dificuldades sentidas foram superadas, os requisitos propostos foram cumpridos e os aspetos a melhor podem ser facilmente cumpridos na próxima fase. Desta forma, ainda que numa fase inicial, o projeto tem uma base sólida para prosseguir com o seu desenvolvimento.