



Universidade do Minho

Escola de Engenharia

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2013/2014

“Animais 360”

A61021 Mário Leite

A61031 Diana Lemos

A61041 Mariana Medeiros

A61049 Miguel Pinto

Junho 2014

Data de Receção	
Responsável	
Avaliação	
Observações	

“Animais 360”

A61021 Mário Leite

A61031 Diana Lemos

A61041 Mariana Medeiros

A61049 Miguel Pinto

Junho 2014

Resumo

O nosso projeto consiste num jogo de estratégia cujo principal objetivo é completar uma volta ao Mundo superando desafios. Assim sendo, decidimos dar-lhe o nome “Animais 360”, já que o tema principal das questões aborda assuntos relacionados com a vida animal e pretende-se que o jogador realize uma volta completa ao globo terrestre (360 graus).

Para conseguir concluir a sua árdua missão, o jogador terá de conquistar os vários continentes do planeta, mostrando que sabe tudo sobre as reservas naturais de cada uma dessas zonas e sobre os animais que nelas habitam.

Thomas Hobbes dizia: “Conhecimento é poder”. Será que o jogador estará à altura deste desafio?

Índice

1. Introdução	1
2. Contextualização e Público Alvo	2
3. “Animais 360” - Fundamentação	3
3.1. Área de jogo	3
3.2. Início do jogo	3
3.3. Dificuldade	3
3.4. Sistema de ajuda	4
3.5. Sistema de pontuação	5
3.6. Desafios	6
3.6.1 Pergunta	6
3.6.2 Sons	6
3.6.3 Completar frases	6
3.6.4 Sombras	6
3.7. Regras	6
3.7.1 Requisitos mínimos de progressão	6
3.7.2 Tempo de resposta	7
3.7.3 O Derradeiro Desafio!	7
3.7.4 Mudança de áreas dentro de um continente	7
3.8. O utilizador	8
3.9. Dashboard utilizador	8
3.10. Home Page	8
4. “Animais 360” – Modelação	9
4.1. Modelo de Domínio	10
4.2. Diagrama de Classes	13
4.3. Diagramas de Use Case (e especificação)	14
4.3.1 Use Case Registrar	15
4.3.2 Use Case Login	15
4.3.3 Use Case Editar Perfil	16
4.3.4 Use Case Deixar Feedback	16
4.3.5 Use Case Logout	17
4.3.6 Subsistema Gerir Áreas Protegidas	18
4.3.7 Subsistema Gerir Desafios	22
4.3.8 Subsistema Consultar Informação	26
4.3.9 Subsistema Consultar Estatística de Jogos	28
4.3.10 Subsistema Gerir Utilizadores	29
4.3.11 Subsistema Gerir Modo de Jogo	32

4.4. Diagrama de Atividades	35
4.4.1 Diagrama Modo de Jogo	35
4.5. Diagramas de Sequência	37
4.5.1 Diagrama Modo de Jogo	37
4.5.2 Diagrama Iniciar Jogo	38
4.5.3 Diagrama Resolver Desafio	38
4.5.4 Diagrama Utilizar Ajuda	40
4.5.5 Diagrama Selecionar Próximo Destino	40
4.5.6 Diagrama Desistir do Jogo	41
4.5.7 Diagrama Adicionar Desafio	42
4.5.8 Diagrama Editar Desafio	43
4.5.9 Diagrama Remover Desafio	44
4.5.10 Diagrama Adicionar Área Protegida	44
4.5.11 Diagrama Editar Área Protegida	46
4.5.12 Diagrama Remover Área Protegida	47
4.6 Base de Dados	48
4.6.1 Análise de requisitos	48
4.6.2 Modelação conceptual da base de dados	49
5 Desenvolvimento do Projeto	63
5.1 Modelo de gestão utilizado	63
5.2 Tomada de decisões e divisão de tarefas	65
5.3 Ferramentas utilizadas no Projeto	66
5.4 Sobre o código	67
6 Custo final do Projeto	68
7 Interface	69
7.1 Página Inicial	69
7.2 Página Login e Registo	70
7.3 Dashboard	70
7.4 Página de jogo	71
8 Conclusões Finais	73
Lista de Acrónimos	74
Referências WWW	75
Referências Bibliográficas	76
 Anexos	
I. Análise das pontuações em casos concretos	78
II Outras páginas da Interface	82

Índice de Figuras

Figura 1 - Diagrama de Modelo de Domínio.	12
Figura 2 - Diagrama de classes.	13
Figura 3 - Use Case Geral.	14
Figura 4 - Especificação do Use Case Registrar.	15
Figura 5 - Especificação do Use Case Login.	15
Figura 6 - Especificação do Use Case Editar Perfil.	16
Figura 7 - Especificação do Use Case Deixar feedback.	16
Figura 8 - Especificação do Use Case Logout.	17
Figura 9 - Subsistema Gerir áreas protegidas.	18
Figura 10 - Especificação do Use Case Adicionar nova área protegida.	19
Figura 11 - Especificação do Use Case Editar área protegida.	1920
Figura 12 - Especificação do Use Case Remover área protegida.	21
Figura 13 - Subsistema Gerir desafios.	22
Figura 14 - Especificação do Use Case adicionar novo desafio.	23
Figura 15 - Especificação do Use Case editar desafio.	24
Figura 16 - Especificação do Use Case remover um desafio.	25
Figura 17 - Use Case Consultar informação.	26
Figura 18 - Especificação do Use Case Consultar tutorial do jogo.	26
Figura 19 - Especificação do Use Case Consultar regras do jogo.	27
Figura 20 - Especificação do Use Case Consultar vídeo promocional.	27
Figura 21 - Especificação do Use Case Consultar ranking dos jogadores.	27
Figura 22 - Especificação do Use Case Consultar estatística geral do sistema.	27
Figura 23 - Subsistema Consultar estatística de jogos.	28
Figura 24 - Especificação do Use Case Consultar estatística do último jogo.	28
Figura 25 - Especificação do Use Case Consultar estatística de todos os jogos.	29
Figura 26 - Subsistema Gerir utilizadores.	29
Figura 27 - Especificação do Use Case Adicionar um novo administrador.	30
Figura 28 - Especificação do Use Case Bloquear um utilizador.	30
Figura 29 - Especificação do Use Case Remover um utilizador.	31
Figura 30 - Especificação do Use Case Consultar informação acerca de um utilizador.	31
Figura 31 - Use Case Modo de jogo.	32
Figura 32 - Especificação do Use Case Iniciar um novo jogo.	32

Figura 33 - Especificação do Use Case Resolver um desafio.	33
Figura 34 - Especificação do Use Case Utilizar uma ajuda.	34
Figura 35 - Especificação do Use Case Selecionar próximo destino.	34
Figura 36 - Especificação do Use Case Desistir do jogo.	34
Figura 37 - Diagrama de Atividades referente ao modo de jogo.	36
Figura 38 - Diagrama de sequência referente ao Modo de Jogo.	37
Figura 39 - Diagrama de sequência referente ao início de jogo.	38
Figura 40 - Diagrama de sequência referente a resolver desafio.	39
Figura 41 - Diagrama de sequência referente à utilização de ajuda.	40
Figura 42 - Diagrama de sequência referente a selecionar o próximo destino.	40
Figura 43 - Diagrama de sequência referente a desistir do jogo.	41
Figura 44 - Diagrama de sequência referente a adicionar desafio.	42
Figura 45 - Diagrama de sequência referente a editar um desafio.	43
Figura 46 - Diagrama de sequência referente a remover desafio.	44
Figura 47 - Diagrama de sequência referente a adicionar área protegida.	45
Figura 48 - Diagrama de sequência referente a editar área protegida.	46
Figura 49 - Diagrama de sequência referente a remover área protegida.	47
Figura 50 - Atributos pertencentes à entidade Utilizador.	51
Figura 51 - Atributos pertencentes à entidade Jogo.	52
Figura 52 - Atributos pertencentes à entidade Classificação.	53
Figura 53 - Atributos pertencentes à entidade Continente.	54
Figura 54 - Atributos pertencentes à entidade Pais.	54
Figura 55 - Atributos pertencentes à entidade Ajuda.	55
Figura 56 - Atributos pertencentes à entidade AreaProtegida.	56
Figura 57 - Atributos pertencentes à entidade Questao.	57
Figura 58 - Atributos pertencentes ao relacionamento Utilizador(N) – Responde em – AreaProtegida (M).	58
Figura 59 - Modelo físico da base de dados.	62
Figura 60 - Relação entre as várias fases de desenvolvimento de um modelo em cascata e as fases do projeto.	64
Figura 61 - Página Inicial do jogo.	69
Figura 62 - Página Login e criar conta.	70
Figura 63 - Página <i>Dashboard</i> .	70
Figura 64 - Página de jogo.	71
Figura 65 - Página de gerir utilizadores.	72

Índice de Tabelas

Tabela 1 - Ilustração dos três tipos de dificuldades.	4
Tabela 2 – Sistema de pontuação por cada nível principal.	5
Tabela 3 - Estimativa do tempo de trabalho.	68

1. Introdução

Foi proposto aos alunos inscritos da Unidade Curricular de Laboratórios de Informática IV no presente ano letivo 2013/2014 que desenvolvessem um *software*.

“*Animais 360*” é um jogo de estratégia que possibilita ao jogador vestir a pele de uma personagem aventureira que enfrenta verdadeiros desafios com o objetivo de obter o título de conquistador do Mundo. Para que tal aconteça, a personagem terá de percorrer todos os continentes do nosso planeta, demonstrando os seus conhecimentos sobre os animais e as reservas naturais que o jogo contempla.

O principal motivo pela escolha do tema “Animais” foi a certeza que se conseguiria suscitar interesse à maioria das pessoas. Exemplo disso é a grande quantidade de pessoas que seguem o canal e revista da *National Geographic* ou que realizam longas viagens apenas para observar de perto uma espécie animal em particular. Para além disso, o facto de ser um tema bastante abrangente e com elevada quantidade de informação de qualidade facilita a procura de material a ser usado no projeto.

Com a elaboração deste projeto, pretende-se proporcionar ao público-alvo um bom serão, acompanhado de conhecimentos sobre a vida animal: o que comem, onde vivem, como se reproduzem e até curiosidades sobre os mesmos. Muito importante também é alertar os jogadores para importância da preservação da diversidade animal no planeta, incutindo o sentido de responsabilidade perante a situação de decadência da vida animal.

Desenvolver uma aplicação desta amplitude e magnitude é um desafio muito motivador que esperamos cumprir com máximo rigor, dando sempre muita atenção aos pormenores, pois é neles que reside a diferença.

Ao longo deste relatório, serão discutidas as várias funcionalidades existentes no jogo e é explicado todo o processo de análise e modelação da aplicação pedida. É feito um estudo de requisitos suportado pelo modelo de domínio, diagrama de classes, use cases (e sua especificação), diagramas de atividades do sistema e, por fim, o modelo conceptual da base de dados, tudo isto de forma a enquadrar e descrever da maneira mais fiel o sistema desenvolvido.

Por fim, é apresentada uma proposta de interface e a sua possível relação com as funcionalidades do sistema.

2. Contextualização e Público Alvo

O público a quem pretendemos chegar com este nosso projeto é bastante abrangente da medida em que este pode ser jogado por crianças até aos mais idosos, desde que tenham as capacidades de leitura e análise necessários para a compreensão das questões colocadas. No entanto, este jogo é mais apropriado para crianças e adolescentes.

Toda a linguagem utilizada na nossa aplicação será de fácil compreensão, por forma a facilitar o acesso do maior número de jogadores.

“Animais 360”, independentemente da idade, vai incutir aos seus utilizadores o gosto pela diversidade animal, os seus habitats e mostrar localizações de reservas naturais, que muita gente desconhece.

Toda esta aventura, o ser posto à prova e a obtenção de pontos por cada desafio ultrapassado, vai suscitar no utilizador uma vontade de obter mais e mais pontos, de forma a poder bater a concorrência e ser o melhor aventureiro.

3. “Animais 360” - Fundamentação

3.1. Área de jogo

O tabuleiro de jogo é o planeta Terra que, por conveniência, será dividido em zonas, coincidentes com os 6 continentes: África, América, Antártida, Ásia, Europa e Oceânia.

Em cada um dos continentes, abordaremos não todos mas apenas aqueles países cujas zonas de reservas naturais protegidas são mais relevantes.

Para concluir uma volta ao Mundo, o jogador terá, obrigatoriamente, de resolver sete desafios por continente. Só depois de superar essas sete questões é que poderá avançar para o próximo continente.

3.2. Início do jogo

No início do jogo, o jogador tem a possibilidade de escolher a personagem que deseja ser, onde começar a sua aventura e o grau de dificuldade que pretende (fácil, médio ou difícil). Esta última decisão implicará diretamente com a complexidade das questões e dos desafios propostos.

3.3. Dificuldade

Para além da escolha inicial do jogador referente à dificuldade geral do jogo, existe também uma outra variedade (quantitativa) de dificuldade, que depende do percurso do jogador. Ou seja, à medida que o jogo se desenrola, o sistema irá ser capaz de adaptar a dificuldade dos desafios à perícia e conhecimento do utilizador. Isto é conseguido porque a complexidade das perguntas vai aumentando ou diminuindo conforme o desempenho do jogador.

Em modo de exemplo, se o jogador opta por uma dificuldade fácil para o seu jogo, então os obstáculos que ele irá enfrentar serão de uma dificuldade quantitativa compreendida entre 0 e 5. Para os restantes dois níveis, há um comportamento semelhante.

Esquematizando:

Dificuldade qualitativa (geral)	Dificuldade quantitativa (de desafio para desafio)
Fácil	0 a 5
Médio	4 a 8
Difícil	7 a 10

Tabela 1 - Ilustração dos três tipos de dificuldades.

3.4. Sistema de ajuda

Durante esta jornada, o jogador terá sempre ao seu dispor três tipos de ajuda. Cada uma delas tem um preço, sendo este definido à custa da sua potencialidade. Ou seja, a ajuda com menor custo pode, de facto, dar apenas uma pista para a resposta correta, enquanto que a ajuda com maior custo pode ser, só por si e em alguns casos, a solução para o desafio. O jogador pode usar qualquer uma das ajudas sempre que bem entender, desde que tenha pontos suficientes para as adquirir.

A título de exemplo, num qualquer desafio, a ajuda de menor custo (Ajuda 1) pode auxiliar o jogador dando uma pista para a resposta, a ajuda de custo intermédio (Ajuda 2) pode retirar metade das hipóteses de resposta e, por fim, a ajuda de maior custo (Ajuda 3) pode encaminhar o aventureiro para uma página da *National Geographic* que lhe dá a resposta ao problema.

Por último, convém também referir que o custo das ajudas depende da dificuldade qualitativa que o jogador está a enfrentar.

3.5. Sistema de pontuação

O jogador inicia a sua aventura com um certo número de pontos, que pode gastar ao longo do seu percurso para adquirir ajudas.

Por cada obstáculo ultrapassado, o jogador recebe os pontos referentes a essa vitória. Contrariamente, por cada fracasso, o jogador perde essa mesma quantidade de pontos. Como é óbvio, quando o jogador não consegue ultrapassar um certo desafio, um outro novo é-lhe lançado até que o consiga superar.

O uso de ajudas em nada influencia a atribuição dos pontos, ou seja, uma pergunta respondida acertadamente merece recompensa tanto tenha sido necessária ajuda ou não e do mesmo modo, uma pergunta respondida erradamente merece penalização independentemente do uso ou não de ajudas.

Todo o sistema de pontuação depende da dificuldade geral que o jogador escolheu:

	Fácil	Médio	Difícil
Pontuação Inicial	50	300	1000
Ganho por desafio superado	5	20	200
Perda por desafio não superado	-5	-20	-200
Custo das ajudas	6,8,10	40,50,60	300,500,600

Tabela 2 – Sistema de pontuação por cada nível principal.

Tal como já foi referido anteriormente, o objetivo do jogo é terminar a volta ao Mundo com maior número de pontos possível. O jogador precisa de acabar o seu percurso com um elevado número de pontos, pois só assim consegue ganhar a todos os outros conquistadores. Assim sendo, é crucial que os jogadores tracem a sua estratégia no que toca ao uso de pontos em ajudas.

Por fim, é de salientar que atingindo pontuação negativa o jogo acaba.

3.6. Desafios

O jogador tem de provar que é valente e, por isso, terá de estar preparado para superar vários tipos de desafios. Existirão quatro tipos possíveis.

3.6.1 Pergunta

É lançada uma pergunta com quatro possibilidades de resposta, em que o jogador tem de indicar a única correta.

3.6.2 Sons

O jogador terá de dar uso à sua capacidade auditiva: ao ouvir um som, terá de ser capaz de o associar a apenas uma das oito espécies de animais apresentadas.

3.6.3 Completar frases

É apresentada uma frase com espaços vazios. A tarefa do jogador é completar esses espaços, seleccionando quais das oito palavras sugeridas fazem mais sentido em cada um desses espaços.

3.6.4 Sombras

É apresentada uma informação factual relativa a um animal e uma imagem desfocada com sombras de vários animais. O jogador tem de, primeiramente, saber a qual animal essa informação corresponde e depois descobrir qual a sua sombra.

3.7. Regras

3.7.1 Requisitos mínimos de progressão

Em cada continente, o jogador terá de obrigatoriamente completar com sucesso sete desafios em áreas protegidas diferentes, para que possa progredir na sua aventura.

3.7.2 Tempo de resposta

Existe um limite de tempo para que o jogador possa responder ao problema. Este limite depende da dificuldade. Assim, para o grau de dificuldade fácil, o jogador disponibilizará de dois minutos para responder, no nível médio o limite desce para um minuto e, por fim, na dificuldade difícil, o jogador apenas terá trinta segundos.

O incumprimento desta regra leva a que se considere a resposta como errada e à consequente perda de pontos.

No caso do uso da Ajuda 1 e da Ajuda 2, o tempo de resposta reinicia-se no momento em que o jogador clica no botão da ajuda. Já na Ajuda 3, o tempo limite para uma resposta não se aplica.

3.7.3 O Derradeiro Desafio!

No momento em que o jogador está prestes a terminar a sua jornada num dado continente (após completar sete desafios em áreas protegidas distintas), é-lhe imposto um desafio especial: O Derradeiro Desafio!

Neste momento, é passado um vídeo de alguns segundos sobre um animal específico. No final do vídeo, são lançadas três questões sobre algo que foi referido durante o mesmo. O jogador terá de selecionar a única resposta correta de entre as quatro possibilidades apresentadas, para cada uma das perguntas. A grande dificuldade deste desafio é que o jogador tem de memorizar a maior quantidade de informação que conseguir pois o vídeo só é emitido uma vez e terá de responder acertadamente a todas as três questões, sem a possibilidade de usar as ajudas. Se for mal sucedido, o aventureiro terá de repetir o desafio.

Após completar com sucesso este grande obstáculo, o jogador recebe um crachá que conquista desse continente e avança para o próximo.

3.7.4 Mudança de áreas dentro de um continente

O jogador tem a possibilidade de escolher em qual dos seis continentes pretende iniciar a sua jornada. No entanto, a rota percorrida pelo explorador entre continentes é definida pelo sistema. Já dentro de um continente, o jogador terá total liberdade de “saltitar” entre áreas protegidas. O sistema apresentar-lhe-á uma lista de países e respetivas áreas protegidas para as quais ele pode avançar. De seguida, o jogador só terá de escolher para qual quer partir.

3.8. O utilizador

No jogo, a personagem principal é o jogador, que tenta conquistar todos os desafios propostos pelo sistema numa determinada área e assim atingir o objetivo de dar a volta ao mundo.

O utilizador deverá ter acesso através de um painel (*dashboard*) ao seu historial de jogo, assim pode observar a área onde falha mais desafios, sendo a área onde tem mais dificuldades e da mesma maneira pode observar o espaço onde domina o jogo.

3.9. Dashboard utilizador

A qualquer momento o jogador terá disponível um *menu* que lhe dará acesso à estatística do jogo que está a decorrer. A área natural que demorou mais tempo no desafio, o número de respostas certas e erradas, o número de graus e áreas que faltam para completar a volta ao mundo.

Adicionalmente no fim do jogo, será mostrado nesta *dashboard* o nível a que o jogador está a competir, isto contempla, o número de voltas que deu ao mundo, o número de jogos efetuados e pontuações, o país/área que errou e acertou mais vezes e o número de crachás conquistados.

3.10. Home Page

Na página inicial é possível encontrar um local para o registro de novos jogadores, um vídeo motivacional a descrever o jogo e uma parte bastante importante, as regras. Existe ainda um local onde mostrará os jogadores melhor classificados nesta volta ao mundo e algumas estatísticas.

Existem dois tipos de utilizadores no *software* a ser criado, a personagem aventureira que pretende ser o melhor a dar voltas ao mundo e o administrador, que tem como principal função controlar os desafios existentes no sistema, monitorizar os utilizadores e o estado do sistema.

4. “Animais 360” – Modelação

A modelagem de *software* consiste em construir modelos que detalhem o sistema de *software* a ser desenvolvido, nomeadamente as suas características e funcionalidades, comportamento e quais as etapas da sua construção. Sendo tudo isto feito com um levantamento prévio de requisitos.

A modelagem assume um papel crucial no desenvolvimento de *software* pois reduz a probabilidade de se cometerem erros, e acima de tudo ajuda a equipa de desenvolvimento a ter uma visão mais abrangente, abstrata e clara do funcionamento do futuro *software*. A abstração permite isolar aspetos que sejam importantes para algum propósito e suprimir os que não forem.

4.1. Modelo de Domínio

O Modelo de Domínio procura capturar as entidades do problema e a relação que se estabelece entre elas, de forma sucinta e legível.

No projeto em questão, as entidades são facilmente intuídas e relacionadas com conceitos como jogo, mapa, questão, utilizador, estatística, etc.

Assim sendo, procede-se à listagem e breve descrição das entidades escolhidas:

A entidade principal, que representa o sistema em geral, é denominada de **Animais360**. Esta entidade tem a ela associada um mapa de jogo, um conjunto de utilizador(es) e uma estatística global de todos os utilizadores registados no jogo.

Um **mapa** é a entidade física constituída por vários continentes. Estes, por sua vez, são constituídos por um conjunto de países que, por último, são constituídos por áreas protegidas. Tanto os **continentes**, **países** como as áreas protegidas têm a eles associado um **nome** e uma pequena **descrição**.

Uma **área protegida** é definida pelas suas coordenadas geográficas, ou seja, a sua latitude e a sua longitude.

A entidade **estatística** está, evidentemente, relacionada com a entidade utilizador.

A um **utilizador** está associado um **nickname**, um **e-mail**, uma **password** e um **avatar** (fotografia de perfil), utilizados para identificar o utilizador no âmbito do jogo que joga. Para além disso, um utilizador possui ainda as suas estatísticas pessoais, **data de registo** e o **tipo** de utilizador (administrador ou não).

As **estatísticas pessoais** são o conjunto de dados estatísticos que resultam dos jogos efetuados pelo utilizador. Têm informação relativa ao **nível** do jogo em que o utilizador se encontra, à sua **pontuação máxima**, ao seu histórico de jogo referente a cada país (**DadosPaís**), ao **número de jogos** realizados pelo utilizador e ao **número de voltas** que o utilizador efetuou. O número de jogos refere-se à quantidade de jogos que o jogador completou sem sucesso enquanto que o número de voltas refere-se à quantidade de jogos que o jogador realizou com sucesso, isto é, conseguiu completar a volta ao mundo.

À entidade **jogo** está associado um tipo de **dificuldade**, a dificuldade qualitativa. A **dificuldade qualitativa** (Fácil, Médio e Difícil) é escolhida pelo utilizador no início de um jogo. Tal como de costume, um jogo é constituído por um conjunto de desafios/questões, tem **respostas certas** e **respostas erradas** e a **pontuação** total que o jogador consegue obter no final dele. Para além disso, está também associado à entidade **personagem** que é o “boneco” escolhido pelo jogador no início do jogo e que o representa na sua jornada.

O utilizador responde a uma série de desafios num determinado **tempo**, com o objetivo de aumentar a sua **pontuação**. Estes podem ter uma imagem como base e são definidos por um conjunto de **hipóteses** de resposta mas apenas uma resposta correta - **Resposta**. A **questão** ou desafio tem um **tipo** que pode ser **PerguntasOpcoes**, **Sons**, **Sombras**, **CompletarFrases** ou o **DerradeiroDesafio**, ou seja, há 5 tipos de desafios diferentes que o

jogador poderá ter de enfrentar. Tal como a entidade jogo, também a entidade questão está associada a uma dificuldade: a dificuldade quantitativa, embora a dificuldade qualitativa também aqui surta efeito. A **dificuldade quantitativa** (de 0 a 10) é induzida pelo sistema, dependendo do percurso do jogador até ao momento.

O jogador tem a possibilidade de beneficiar de ajudas para ultrapassar os seus desafios/questões. Estas **ajudas** são pistas que podem ser de **graus** distintos já que existem três tipos de ajudas diferentes e têm a elas associado um **custo**, na medida em que o jogador só as pode utilizar se tiver pontos suficientes para as adquirir. Claro está que o custo que o jogador paga para beneficiar de ajudas implica diretamente com a **pontuação** com que ele fica no fim.

De seguida, é apresentado o esquema do modelo de domínio:

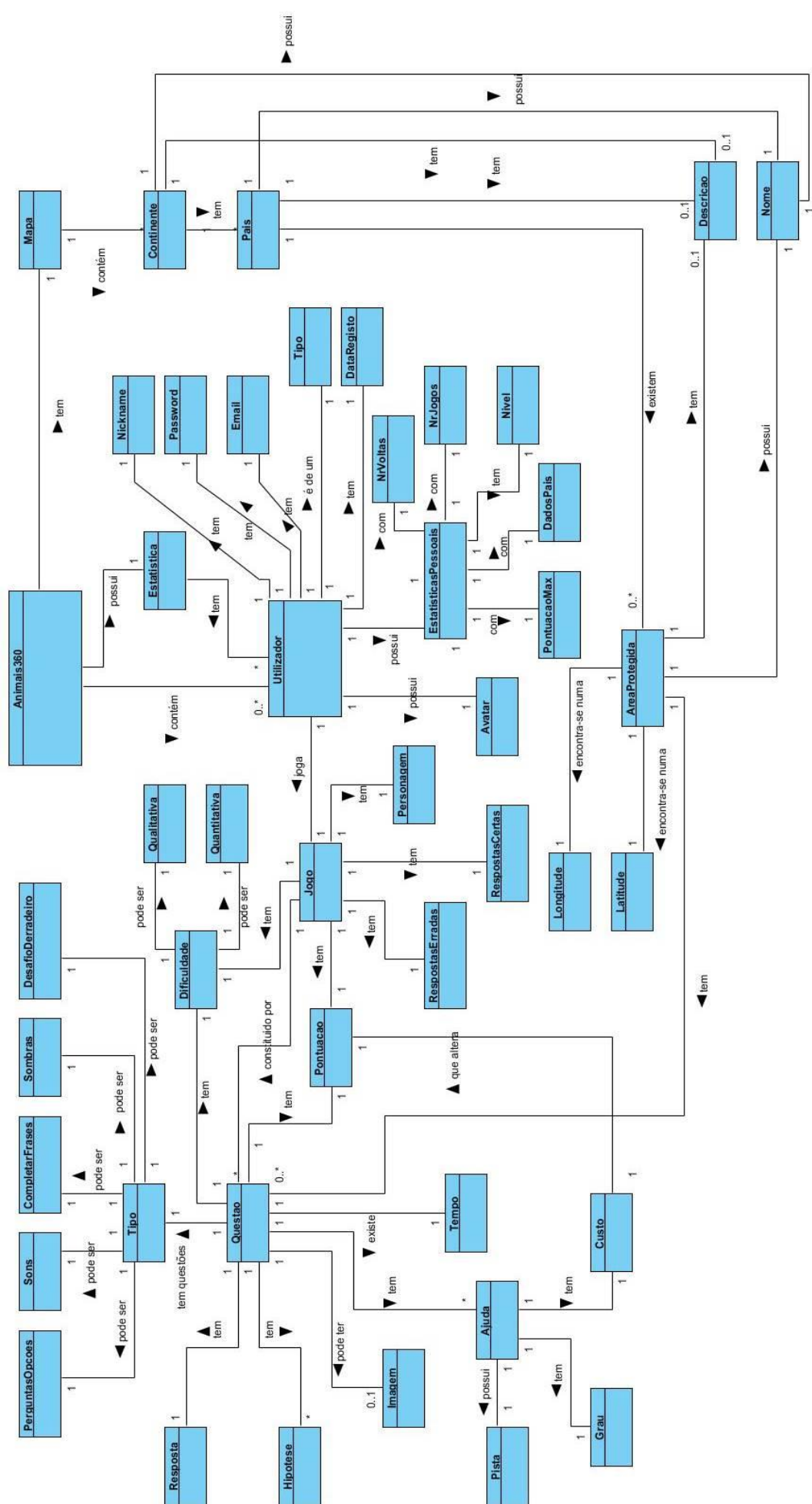


Figura 1 - Diagrama de Modelo de Domínio.

4.2. Diagrama de Classes

O diagrama de classes representa a estrutura do sistema, recorrendo ao conceito de classes e suas relações, onde são identificados os objetos relevantes do sistema. Os objetos do sistema possuem comportamento e identidade, que consistem, respetivamente, ao modo como ele age e reage a estímulos externos e aos atributos que o distingue de todos os outros.

A seguir é apresentado o diagrama de classes do sistema onde são representadas todas as classes do jogo.

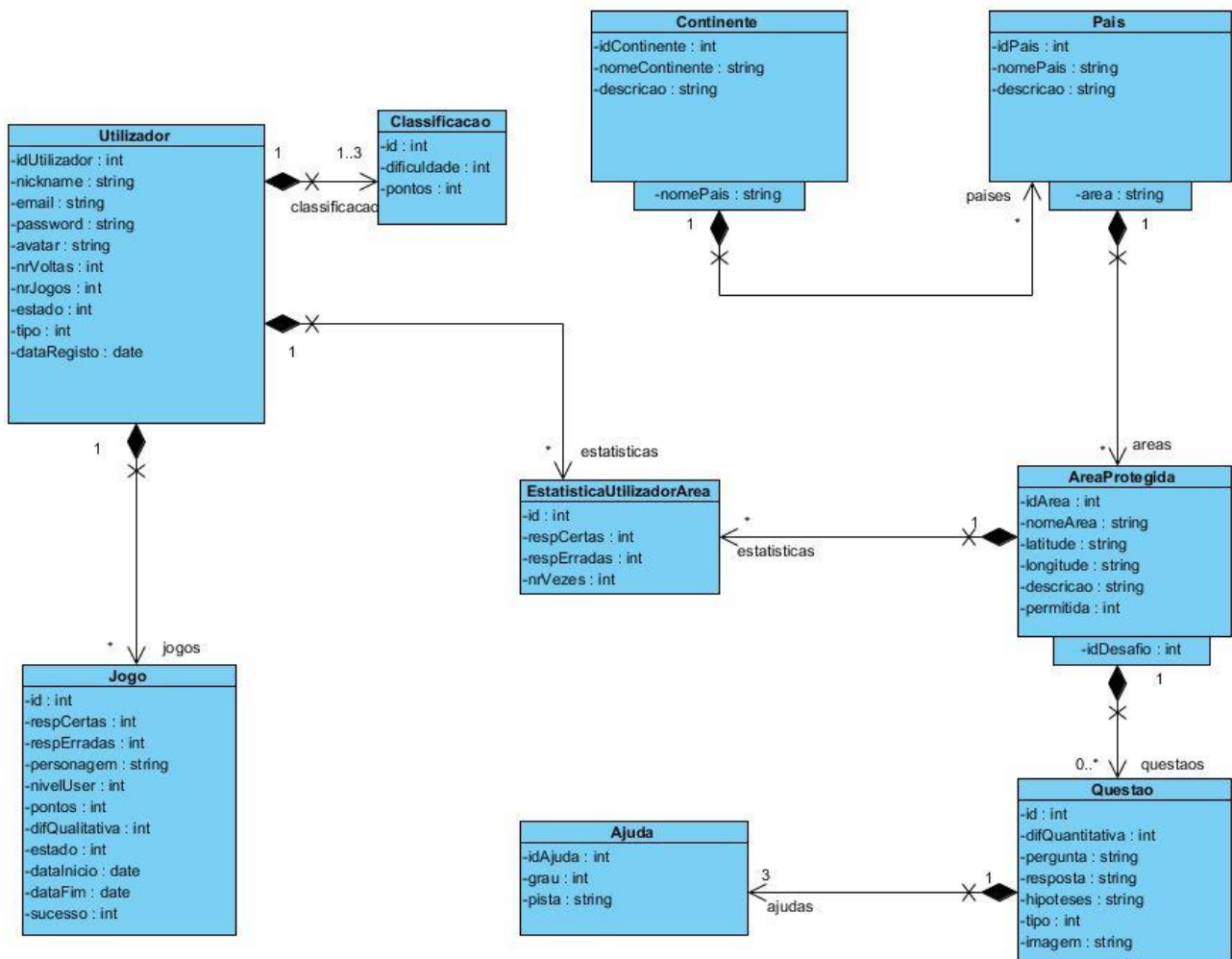


Figura 2 - Diagrama de classes.

4.3. Diagramas de Use Case (e especificação)

Animais 360 é uma aplicação com tipos de utilizadores distintos, todos eles com diferentes papéis.

Um utilizador do tipo Convidado é aquele que ainda não se registou no jogo e apenas pode fazer consultas sobre estatísticas gerais, saber mais sobre as regras do jogo, etc.

Um utilizador registado é aquele que já se encontra registado no sistema. Este tipo de utilizador pode tirar partido de toda a aplicação, com ênfase para a possibilidade de realizar jogos e consultar uma variedade de dados estatísticos próprios e globais.

Por último, o utilizador do tipo Administrador tem permissões para gerir toda a aplicação tendo, deste modo, permissões para fazer tudo o que os outros utilizadores podem fazer e mais ainda: pode bloquear e remover utilizadores indesejados, adicionar um outro administrador, adicionar novos desafios ao sistema, etc.

Num diagrama de use case, são representados todos os atores envolventes de um dado sistema, assim como todas as ações que estes podem executar.

O diagrama de use case referente ao nosso sistema é o seguinte:

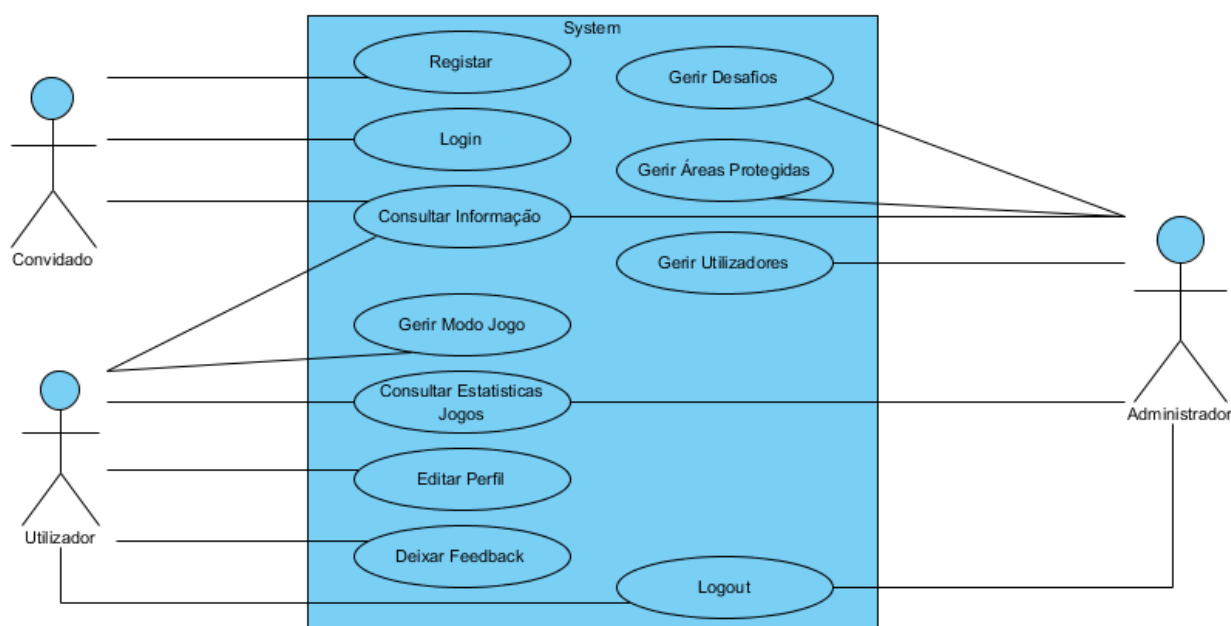


Figura 3 - Use Case Geral.

Neste diagrama estão representados todos os subsistemas da nossa aplicação, incluindo todos os atores e as suas ações. Para uma melhor compreensão, os subsistemas irão ser apresentados individualmente e, para cada um deles, o respetivo diagrama de use case e as suas especificações.

4.3.1 Use Case Registrar

Este use case representa o registo de um utilizador no sistema. Este passo é necessário para que o utilizador possa usufruir de todas as funcionalidades do jogo.

De seguida, encontra-se a especificação deste use case:

Brief Description	Efectuar registo.	
Preconditions		
Post-conditions	Registo efectuado	
Flow of Events		Actor Input
	1	Solicita formulário de registo.
	2	
	3	Preenche campos do formulário.
	4	
	5	
	6	
	7	
	8	
Alternativa 1 [Dados Inválidos] Passo 4		Actor Input
	1	
	2	
	3	Preenche formulário.
	4	

Figura 4 - Especificação do Use Case Registrar.

4.3.2 Use Case Login

Este use case representa o login de um utilizador no sistema. Este passo, em conjunto com o registo do utilizador, é necessário para que o mesmo possa usufruir de todas as funcionalidades do jogo.

De seguida, encontra-se a especificação deste use case:

Brief Description	Efectuar <u>login</u> .	
Preconditions	<u>Login</u> não efectuado	
Post-conditions	Utilizador autenticado	
Flow of Events		Actor Input
	1	Pede formulário para efectuar <u>login</u> .
	2	
	3	Preenche formulário.
	4	
	5	
Alternativa 1		Actor Input
	1	
	2	
	3	Preenche campos.
	4	

Figura 5 - Especificação do Use Case Login.

4.3.3 Use Case Editar Perfil

Este use case representa a edição do perfil de um utilizador registado no sistema, mais propriamente de um jogador.

De seguida, encontra-se a especificação deste use case:

Brief Description	Editar perfil de jogador.		
Preconditions	Utilizador registado e autenticado		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Pede formulário de edição de perfil.	
	2		Devolve formulário.
	3	Preenche os campos com as alterações.	
	4	Submete formulário.	
	5		Verifica validade dos dados alterados.
	6		Confirma validade das alterações.
	7		Actualiza alterações feitas no perfil.
	8		Notifica que alterações foram guardadas com sucesso.
	9		
Alternativa 1 [Dados Inválidos] Passo 5		Actor Input	System Response
	1		Notifica que dados alterados são inválidos.
	2		Devolve formulário.
	3	Preenche formulário com alterações.	
	4		Regressa ao passo 5.

Figura 6 - Especificação do Use Case Editar Perfil.

4.3.4 Use Case Deixar Feedback

Caso um utilizador registado no sistema pretenda deixar a sua opinião em relação ao jogo, pode fazê-lo enviando uma mensagem ao(s) administrador(es). Assim sendo, este use case representa a ação de deixar esse feedback.

De seguida, encontra-se a especificação deste use case:

Brief Description	Deixar feedback.		
Preconditions	Utilizador autenticado.		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Envia mensagem com feedback.	
	2		Recebe mensagem.

Figura 7 - Especificação do Use Case Deixar feedback.

4.3.5 Use Case Logout

Este use case representa a saída do utilizador do sistema. Este passo é necessário para que o utilizador feche sessão no jogo, caso o pretenda. A partir deste momento, o utilizador apenas pode realizar consultas, tal como se fosse um utilizador não registado.

De seguida, encontra-se a especificação deste use case:

Brief Description	Logout		
Preconditions	Estar logado		
Post-conditions	Fica com sessão encerrada		
Flow of Events		Actor Input	System Response
	1	Pede para fazer logout.	
	2		Verifica se operação pode ser realizada.
	3		Pede confirmação de operação.
	4	Confirma operação.	
	5		Efectua logout e notifica operação.
	6		Fecha sessão.
Excepção 1 [Não pode efectuar logout] Passo 2		Actor Input	System Response
	1		Notifica que operação não pode ser realizada.
	2		
Excepção 2 [Não confirma operação] Passo 5		Actor Input	System Response
	1		Cancela operação e notifica.
	2		

Figura 8 - Especificação do Use Case Logout.

4.3.6 Subsistema Gerir Áreas Protegidas

Este subsistema apenas envolve um ator, o Administrador. O diagrama use case deste subsistema é relativo à gestão - adição, remoção e edição - de áreas protegidas.

Adicionar uma área protegida equivale a associar um determinado nome a um ponto geográfico concreto, com uma certa latitude e longitude, do mapa de jogo. Nesse momento, essa área passa a pertencer ao conjunto de áreas abordadas no âmbito do jogo.

Remover uma área protegida implica retirar todo o significado a um certo ponto físico do mapa de jogo, na medida em que essa área protegida já não será mais contemplada pelo mesmo.

Por último, editar uma área protegida implica alterar os seus dados: nome, localização, permissão, etc.

Convém referir que algumas áreas protegidas são de carácter informativo, não pertencendo à lista de áreas às quais o jogador pode responder a desafios.

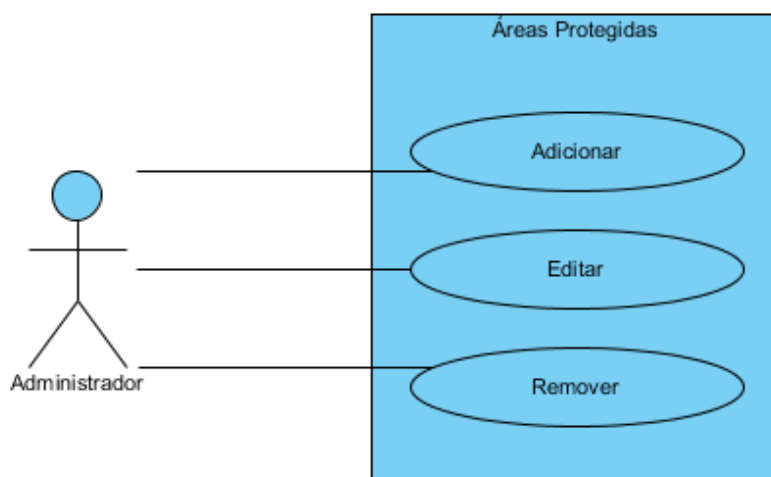


Figura 9 - Subsistema Gerir áreas protegidas.

De seguida, encontram-se as especificações dos use case do diagrama deste subsistema:

Brief Description	Adicionar nova área protegida num país.		
Preconditions	Administrador autenticado		
Post-conditions	Nova área protegida adicionada		
Flow of Events		Actor Input	System Response
	1	Pede ao sistema formulário para inserir nova área protegida.	
	2		Devolve o formulário.
	3	Selecciona o continente onde será inserida a área.	
	4	Selecciona o país onde será inserida a área.	
	5		Verifica se país existe no sistema.
	6		Verifica que país existe.
	7	Submete o formulário.	
	8		Verifica todos os campos do formulário.
	9		Verifica que campos são válidos.
	10		Adiciona nova área à lista de áreas protegidas no sistema.
	11		Notifica que área foi inserida com sucesso.
Alternativa 1 [País não existe no sistema] Passo 5		Actor Input	System Response
	1		Pergunta se pretende inserir o país no sistema.
	2	Responde afirmativamente.	
	3		Devolve formulário para inserir país.
	4	Preenche formulário.	
	5		Verifica dados preenchidos.
	6		Verifica que país é válido.
	7		Insere país na lista de países do sistema.
	8		Notifica sucesso da operação.
	9		Volta ao passo 6.
Excepção 1 [Responde negativamente] Passo 2 Alternativa 1		Actor Input	System Response
	1		Cancela operação.
	2		Notifica utilizador.
Alternativa 2 [Dados inválidos] Passo 5 Alternativa 1		Actor Input	System Response
	1		Notifica que dados do país são inválidos.
	2		Devolve formulário.
	3	Preenche formulário.	
	4		Volta ao passo 5 da alternativa 1.

Figura 10 - Especificação do Use Case Adicionar nova área protegida.

Brief Description	Editar informação sobre uma área protegida.		
Preconditions	Área protegida existe no sistema; Administrador autenticado		
Post-conditions	Área protegida editada com sucesso		
Flow of Events		Actor Input	System Response
	1	Pede ao sistema o formulário para editar informação relativa a uma área protegida.	
	2		Devolve formulário.
	3	Efectua alterações pretendidas.	
	4	Submete formulário.	
	5		Verifica validade dos campos alterados.
	6		Confirma validade das alterações.
	7		Actualiza alterações feitas na área protegida.
	8		Notifica que alterações foram guardadas com sucesso.
Alternativa 1 [Dados Inválidos] Passo 5		Actor Input	System Response
	1		Verifica se o formulário errado foi submetido menos de 3 vezes.
	2		Notifica que dados são inválidos e devolve formulário.
	3	Preenche formulário.	
	4	Submete formulário.	
	5		Regressa ao passo 5.
Excepção 1 [Dados Inválidos 3 vezes] Passo 1 Alternativa 1		Actor Input	System Response
	1		Notifica que dados são inválidos pela terceira vez.
	2		Cancela operação.

Figura 11 - Especificação do Use Case Editar área protegida.

Brief Description	Remover uma área protegida.		
Preconditions	Administrador autenticado		
Post-conditions	Área protegida removida com sucesso		
Flow of Events		Actor Input	System Response
	1	Selecciona o país de onde quer remover a área.	
	2	Selecciona a área protegida que pretende remover.	
	3		Verifica se área existe no sistema.
	4		Verifica se área protegida não contém desafios.
	5		Pede confirmação da operação.
	6	Confirma operação.	
	7		Elimina área da lista de área do país seleccionado.
	8		Actualiza lista de áreas.
	9		Notifica sucesso da operação.
Excepção 1 [Área não existe] Passo 3		Actor Input	System Response
	1		Notifica que área protegida não existe.
	2		Cancela operação.
Excepção 2 [Não confirma operação] Passo 6		Actor Input	System Response
	1		Cancela operação.
Excepção 1 [Área protegida contém desafios] Passo 4		Actor Input	System Response
	1		Notifica que área contém desafios.
	2		Cancela operação.

Figura 12 - Especificação do Use Case Remover área protegida.

4.3.7 Subsistema Gerir Desafios

À semelhança do subsistema anterior, também este subsistema envolve apenas o Administrador, na medida que apenas este tem permissões que lhe permite efetuar a gestão - adição, remoção e edição - dos desafios.

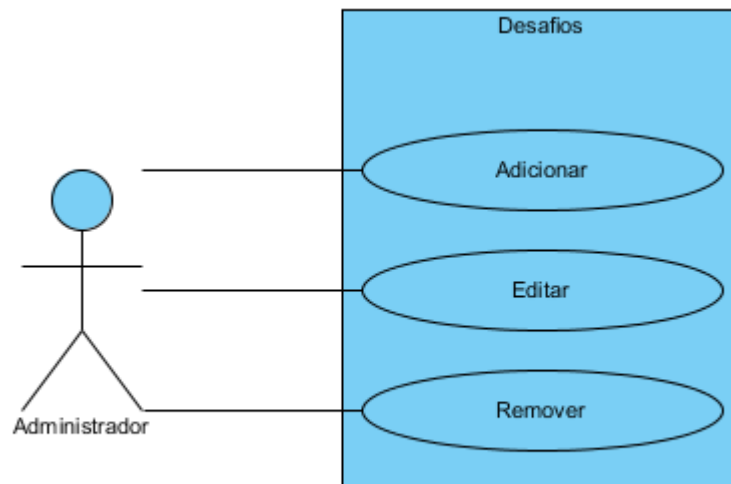


Figura 13 - Subsistema Gerir desafios.

De seguida, encontram-se as especificações dos use case do diagrama deste subsistema:

Brief Description	Adicionar novas perguntas a um desafio.		
Preconditions	Desafio existe no sistema; Administrador autenticado		
Post-conditions	Novo desafio adicionado		
Flow of Events		Actor Input	System Response
	1	Pede ao sistema formulário de criação de nova pergunta no desafio.	
	2		Devolve o formulário.
	3	Selecciona país onde será inserida a pergunta.	
	4	Selecciona a área onde será inserida a pergunta.	
	5	Selecciona dificuldade da pergunta.	
	6	Selecciona qual o tipo de pergunta.	
	7	Insere pergunta.	
	8	Submete o formulário.	
	9		Verifica todos os campos do formulário.
	10		Verifica que desafio é válido.
	11		Adiciona desafio à lista de desafios existentes no sistema.
	12		Notifica que desafio foi inserido com sucesso.
Alternativa 1 [Campos Inválidos] Passo 8		Actor Input	System Response
	1		Verifica se número de submissões erradas é menor que 3.
	2		Notifica que campos preenchidos são inválidos e devolve formulário.
	3	Preenche e submete formulário.	
Excepção 1 [Campos Inválidos 3 vezes] Passo 1 Alternativa 1	4		Volta ao passo 9.
		Actor Input	System Response
	1		Cancela operação.
	2		Notifica utilizador.

Figura 14 - Especificação do Use Case adicionar novo desafio.

Brief Description	Editar desafio.		
Preconditions	Desafio existe no sistema; Administrador autenticado		
Post-conditions	Desafio editado com sucesso		
Flow of Events		Actor Input	System Response
	1	Pede ao sistema o formulário para editar um desafio.	
	2		Devolve formulário.
	3	Efectua alterações pretendidas.	
	4	Submete formulário.	
	5		Verifica validade dos campos alterados.
	6		Confirma validade das alterações.
	7		Actualiza alterações feitas no desafio.
	8		Notifica que alterações foram guardadas com sucesso.
Alternativa 1 [Alterações Invalidas] Passo 5		Actor Input	System Response
	1		Verifica se submeteu o formulário errado menos de 3 vezes.
	2		Notifica que dados são inválidos e devolve formulário.
	3	Preenche formulário.	
	4	Submete formulário.	
	5		Regressa ao passo 5.
Excepção 1 [Dados Inválidos 3 vezes] Passo 1 Alternativa 1		Actor Input	System Response
	1		Notifica que dados são inválidos.
	2		Cancela operação.

Figura 15 - Especificação do Use Case editar desafio.

Brief Description	Remover um desafio do jogo.		
Preconditions	Administrador autenticado		
Post-conditions	Desafio removido com sucesso		
Flow of Events		Actor Input	System Response
	1	Selecciona o país e a área de onde pretende remover um desafio.	
	2	Selecciona o desafio que pretende remover.	
	3		Verifica se desafio existe no sistema.
	4		Pede confirmação da operação.
	5	Confirma operação.	
	6		Elimina desafio da lista de desafios.
	7		Actualiza lista de desafios da área seleccionada.
	8		Notifica sucesso da operação.
Excepção 1 [Desafio não existe] Passo 3		Actor Input	System Response
	1		Notifica que desafio não existe.
	2		Cancela operação.
Excepção 2 [Não confirma operação] Passo 5		Actor Input	System Response
	1		Notifica que desafio não foi removido.
	2		Cancela operação.

Figura 16 - Especificação do Use Case remover um desafio.

4.3.8 Subsistema Consultar Informação

Este subsistema envolve os três tipos de utilizadores (convidado, utilizador registado e administrador). O diagrama use case deste subsistema é relativo à consulta de informação acerca do jogo “Animais 360”. Consultar o tutorial do jogo, regras, ver vídeo, ranking dos utilizadores e algumas estatísticas gerais da aplicação.

Através da consulta do tutorial e das regras do jogo, o utilizador pode-se familiarizar com o mesmo. Também pode consultar algumas estatísticas dos jogos realizados pelos jogadores registados e quais deles são os melhores jogadores, através da consulta do ranking geral.

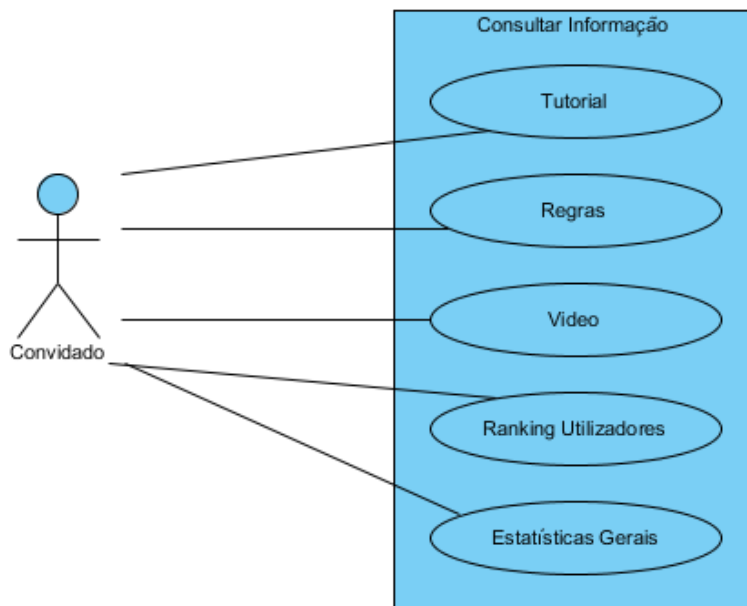


Figura 17 - Use Case Consultar informação.

De seguida, encontram-se as especificações dos use case do diagrama deste subsistema:

Brief Description		Consultar tutorial do jogo	
Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Solicita tutorial do jogo.	
	2		Devolve tutorial do jogo.
	3	Vê tutorial.	
	4		Fecha sessão.

Figura 18 - Especificação do Use Case Consultar tutorial do jogo.

Brief Description	Consultar regras do jogo	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Solicita a consulta das regras do jogo.
	2	
	3	Visualiza as regras.
	4	Fecha sessão

Figura 19 - Especificação do Use Case Consultar regras do jogo.

Brief Description	Consultar o video motivacional.	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Solicita a visualização do video.
	2	
	3	Visualiza o video.
	4	Fecha sessão.

Figura 20 - Especificação do Use Case Consultar vídeo promocional.

Brief Description	Consultar o ranking dos jogadores registados no jogo.	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Solicita o ranking dos jogadores.
	2	
	3	Visualiza o ranking
	4	Fecha sessão.

Figura 21 - Especificação do Use Case Consultar ranking dos jogadores.

Brief Description	Consultar a estatística geral de todo sistema	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Solicita ver a página que contém a estatística geral do jogo.
	2	
	3	Visualiza as estatísticas.
	4	Fecha sessão

Figura 22 - Especificação do Use Case Consultar estatística geral do sistema.

4.3.9 Subsistema Consultar Estatística de Jogos

Este subsistema envolve dois tipos de utilizadores, o utilizador registado e o administrador. O diagrama use case deste subsistema é relativo à consulta de informação acerca da estatística dos jogos realizados pelo jogador.

O jogador pode consultar a estatística relativa a todos os jogos realizados, e se assim preferir pode também consultar a estatística de apenas o último jogo.

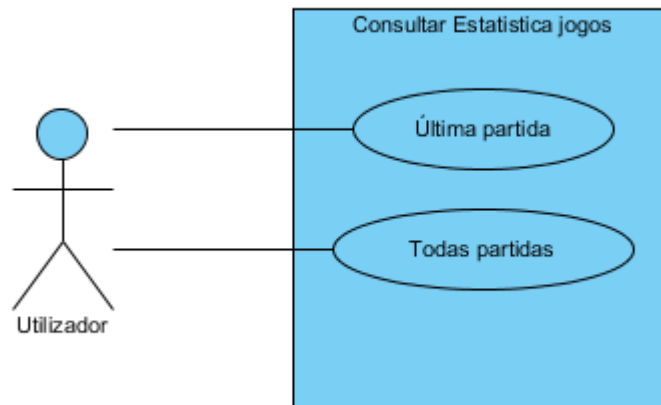


Figura 23 - Subsistema Consultar estatística de jogos.

De seguida, encontram-se as especificações dos use case do diagrama deste subsistema:

Brief Description	Consultar a estatística relativa à última partida realizada.		
Preconditions	Jogador autenticado		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Pede para consultar a estatística da última partida que jogou.	
	2		Verifica a última partida.
	3		Mostra informação relativa à estatística.
	4	Visualiza as estatísticas.	
	5		Fecha sessão.
Excepção 1 [Não existem partidas realizadas] Passo 2		Actor Input	System Response
	1		Notifica que é impossível visualizar a estatística.
	2		Fecha sessão.

Figura 24 - Especificação do Use Case Consultar estatística do último jogo.

Brief Description	Consulta estatísticas relativas a todos as partidas realizadas.		
Preconditions	Utilizador autenticado;		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Solicita visualizar página relativa à estatística de todos os jogos realizados.	
	2		Verifica jogos realizados pelo utilizador.
	3		Mostra estatísticas.
	4	Consulta estatísticas.	
	5		Fecha sessão.
Excepção 1 [Não existem jogos realizados] Passo 2		Actor Input	System Response
	1		Notifica que é impossível mostrar estatísticas.
	2		Fecha sessão.

Figura 25 - Especificação do Use Case Consultar estatística de todos os jogos.

4.3.10 Subsistema Gerir Utilizadores

Este subsistema apenas envolve um ator, o Administrador. O diagrama use case deste subsistema é relativo à gestão - adição, remoção e edição - de utilizadores e à possibilidade de nomeação de um outro administrador.

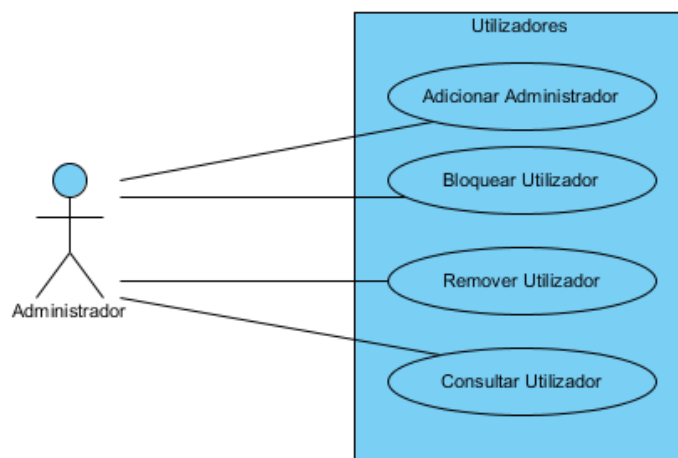


Figura 26 - Subsistema Gerir utilizadores.

De seguida, encontram-se as especificações dos use case do diagrama deste subsistema:

Brief Description	Adicionar um novo administrador.		
Preconditions	Administrador autenticado		
Post-conditions	Novo administrador adicionado com sucesso		
Flow of Events		Actor Input	System Response
	1	Pede ao sistema formulário de criação de um novo administrador.	
	2		Devolve o formulário.
	3	Preenche campos com os dados.	
	4		Verifica todos os campos do formulário.
	5		Adiciona administrador à lista de administradores.
	6		Notifica sucesso de operação.
	7		
	8		
Alternativa 1 [Campos Inválidos] Passo 4		Actor Input	System Response
	1		Notifica que campos são inválidos
	2		Devolver formulário.
	3	Preenche todos os campos do formulário.	
	4		Regressa ao passo 4.

Figura 27 - Especificação do Use Case Adicionar um novo administrador.

Brief Description	Bloquear um utilizador.		
Preconditions	Administrador autenticado		
Post-conditions	Utilizador bloqueado		
Flow of Events		Actor Input	System Response
	1	Selecciona utilizador que pretende bloquear.	
	2		Verifica dados do utilizador.
	3		Pede confirmação.
	4	Confirma operação.	
	5		Bloqueia utilizador.
	6		Actualiza lista dos utilizadores activos.
Excepção 1 [Não confirma operação] Passo 4		Actor Input	System Response
	1		Cancela operação.

Figura 28 - Especificação do Use Case Bloquear um utilizador.

Brief Description	Remover um utilizador.		
Preconditions	Administrador autenticado; Utilizador existe no sistema		
Post-conditions	Utilizador removido		
Flow of Events		Actor Input	System Response
	1	Selecciona utilizador que pretende bloquear.	
	2		Verifica se utilizador está bloqueado.
	3		Remove utilizador.
	4		Actualiza lista de utilizadores do sistema.
	5		Notifica sucesso da operação.
Excepção 1 [Utilizador não está bloqueado] Passo 2		Actor Input	System Response
	1		Notifica que utilizador não pode ser removido.
	2		Cancela operação.

Figura 29 - Especificação do Use Case Remover um utilizador.

Brief Description	Consultar informações acerca de um utilizador		
Preconditions	Administrador autenticado; Utilizador registado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Pede ao sistema informação sobre um utilizador.	
	2		Verifica informação do utilizador.
	3		Mostra informação solicitada.
	4	Consulta informação.	
	5		Fecha sessão.

Figura 30 - Especificação do Use Case Consultar informação acerca de um utilizador.

4.3.11 Subsistema Gerir Modo de Jogo

Este subsistema envolve um ator, o Jogador (utilizador registado). O diagrama use case deste subsistema é relativo ao modo de jogo, isto é, para que um jogo se realize, o jogador terá de realizar um conjunto de ações, sendo uma delas a *iniciar jogo*. Há medida que a partida se vai desenrolando o jogador resolve desafios, podendo utilizar ajudas para os resolver, também tem de escolher o próximo destino a ser explorado e, caso o pretenda, desistir do jogo.

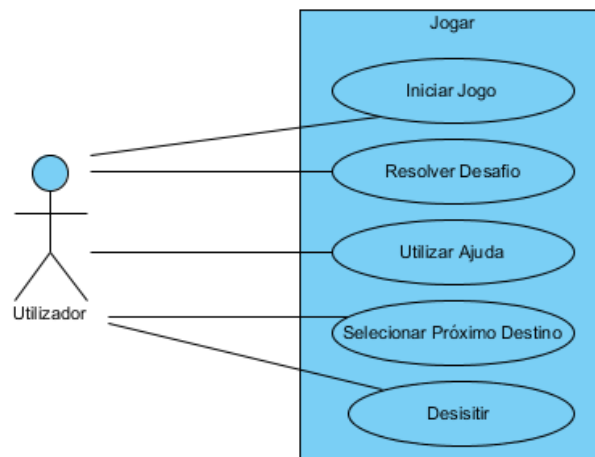


Figura 31 - Use Case Modo de jogo.

De seguida, encontram-se as especificações dos use case do diagrama deste subsistema:

Brief Description	Iniciar um novo jogo.		
Preconditions	Jogador autenticado		
Post-conditions	Jogo iniciado		
Flow of Events		Actor Input	System Response
	1	Selecciona a personagem com a qual quer fazer o jogo.	
	2	Selecciona a dificuldade de jogo.	
	3	Selecciona o continente onde começa o jogo.	
	4	Inicia jogo.	
	5		Verifica se todos os campos foram seleccionados.
	6		Confirma campos seleccionados.
	7		Inicia jogo.
Alternativa 1 [Campos Inválidos] Passo 5		Actor Input	System Response
	1		Notifica que campos são inválidos.
	2		Apresenta campos a seleccionar.
	3	Preenche todos os campos.	
	4		Volta ao passo 5.

Figura 32 - Especificação do Use Case Iniciar um novo jogo.

Brief Description	Resolver um desafio		
Preconditions	Jogador autenticado; Jogo iniciado		
Post-conditions	Desafio ultrapassado		
Flow of Events		Actor Input	System Response
	1	Selecciona o país onde pretende jogar.	
	2		Mostra desafio que tem de ultrapassar.
	3	Responde ao desafio.	
	4		Verifica resposta.
	5		Valida resposta.
	6		Actualiza número de pontos.
	7		Verifica número de pontos.
	8		Verifica se tem desafio seguinte.
	9		Notifica sucesso de operação.
Alternativa 1 [Não sabe resposta] Passo 3		Actor Input	System Response
	1	Selecciona uma das ajudas.	
	2		Verifica se tem moedas suficientes.
	3		Valida ajuda.
	4		Mostra ajuda.
	5		Actualiza número de moedas.
	6	Utiliza ajuda e responde ao desafio.	
	7		Regressa ao passo 4.
Excepção 1 [Número de pontos negativo] Passo 7		Actor Input	System Response
	1		Notifica que número de pontos é negativo.
	2		Actualiza e guarda estatísticas do jogo.
	3		Notifica fim de jogo.
	4		Finaliza jogo.
Alternativa 2 [Resposta errada] Passo 4		Actor Input	System Response
	1		Notifica que resposta é errada.
	2		Actualiza e verifica número de pontos.
	3		Mostra um novo desafio.
	4	Responde ao desafio.	
	5		Regressa ao passo 4.
Excepção 2 [Não tem mais desafios] passo 8		Actor Input	System Response
	1		Notifica que chegou ao fim do jogo.
	2		Fecha sessão.

Figura 33 - Especificação do Use Case Resolver um desafio.

Brief Description	Utilizar uma ajuda.	
Preconditions	Jogo iniciado; Jogador autenticado	
Post-conditions	Ajuda utilizada	
Flow of Events		Actor Input
	1	Selecciona ajuda que pretende utilizar.
	2	
	3	
	4	
	5	Utiliza ajuda e responde ao desafio.
Excepção 1 [Quantidade insuficiente] Passo 2		System Response
	1	Notifica que quantidade de moedas é insuficiente.
	2	Cancela operação.
	3	

Figura 34 - Especificação do Use Case Utilizar uma ajuda.

Brief Description	Selecciona próximo destino.	
Preconditions	Jogador autenticado; Jogo iniciado	
Post-conditions	Jogador encontra-se no destino pretendido	
Flow of Events		Actor Input
	1	Selecciona país para onde pretende seguir.
	2	
	3	
	4	
Flow of Events		System Response
	1	Verifica país seleccionado.
	2	Actualiza jogo.
	3	Notifica sucesso de operação.
	4	

Figura 35 - Especificação do Use Case Seleccionar próximo destino.

Brief Description	Desistir do jogo.	
Preconditions	Jogador autenticado; Jogo iniciado	
Post-conditions	Jogo cancelado	
Flow of Events		Actor Input
	1	Selecciona que pretende desistir do jogo.
	2	
	3	
	4	
	5	Confirma operação.
	6	
	7	
Excepção 1 [Não confirma operação] Passo 5		System Response
	1	Cancela operação de cancelar jogo.
Flow of Events		System Response
	1	Actualiza número de pontos e moedas.
	2	Actualiza local do mapa onde se encontra.
	3	Pede confirmação da operação.
	4	Cancela jogo.
	5	Fecha sessão.
	6	
	7	

Figura 36 - Especificação do Use Case Desistir do jogo.

4.4. Diagrama de Atividades

O diagrama de atividades representa o controlo de fluxo de uma atividade para outra. Trata-se de um caso especial de diagrama de estado, no qual a maioria dos estados é um estado de ação e as transições são ativadas pela conclusão de ações nos estados precedentes.

Estes diagramas são usados para se compreender o comportamento de um dado requisito e servem como auxílio aos Use Cases.

4.4.1 Diagrama Modo de Jogo

Neste trabalho de modelagem, foi feito o diagrama de atividades que diz respeito ao Modo de Jogo. Este diagrama torna-se útil para compreender todo o processo de jogo, desde a escolha da personagem que acompanhará o jogador, o tipo de dificuldade (qualitativa) e o continente onde terá início a partida. Após essas escolhas, o jogador terá de decidir se pretende iniciar o jogo ou não. Caso dê início à sua aventura, terá de escolher a área protegida, existente num país, onde resolverá o primeiro desafio.

Ao resolver qualquer desafio, o jogador terá sempre a opção de usar ou não ajuda, para tal basta ter o número suficiente de pontos. Se o jogador acertar a pergunta, o sistema verifica se é necessário trocar de continente. Caso isso aconteça, é lançado o Derradeiro Desafio que o jogador terá de ultrapassar. Após resolvido esse desafio, é verificado se há mais continentes a serem explorados, se a resposta for negativa o jogo terminará, caso contrário o jogador terá de escolher outra área protegida (já noutro continente) para visitar.

No caso de o jogador errar na resposta à pergunta, ser-lhe-á retirado o número de pontos correspondente e há uma verificação destes. Se o jogador atingir um número inferior a zero pontos terá perdido e chegado ao fim do jogo, caso contrário continua a sua aventura.

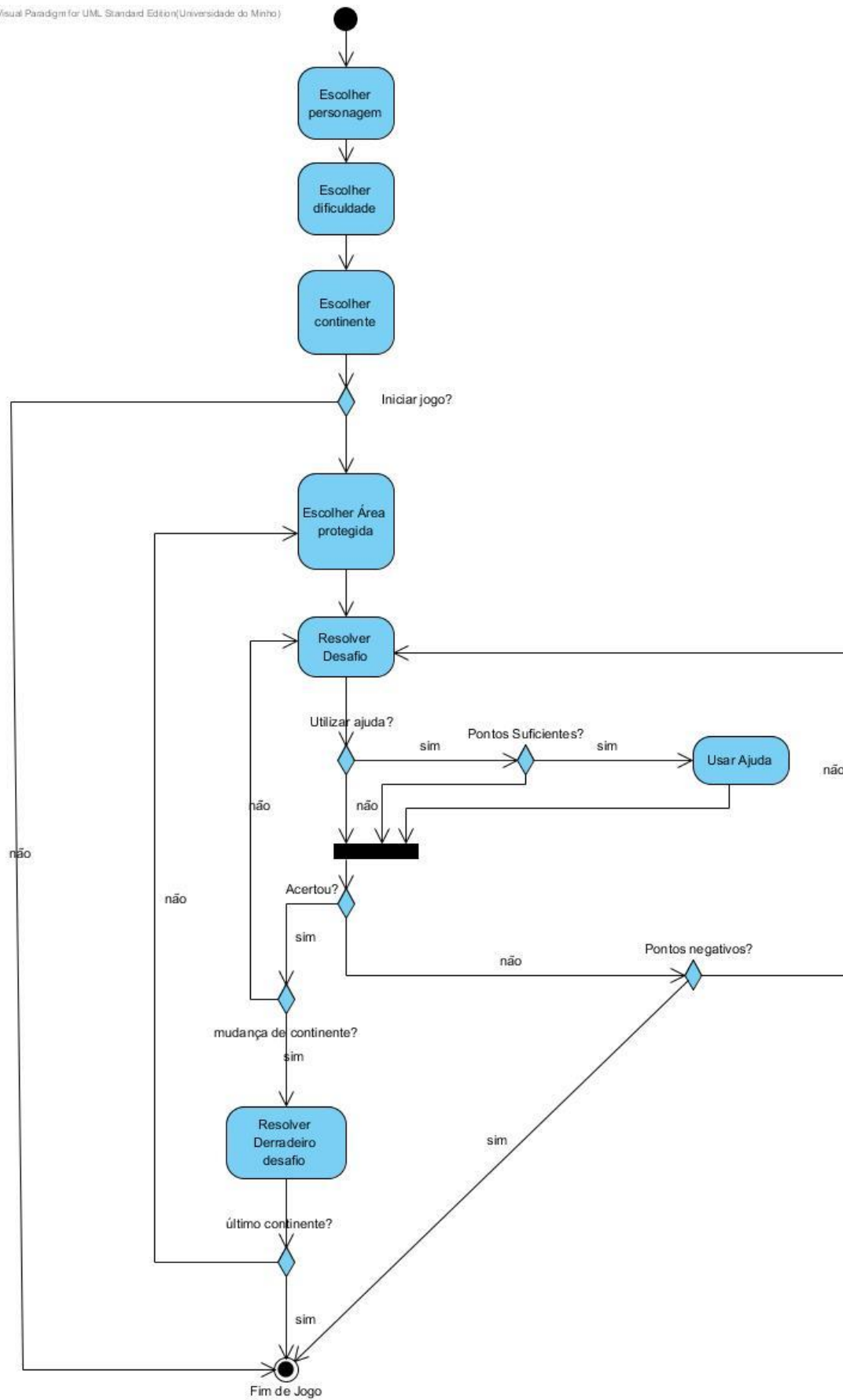


Figura 37 - Diagrama de Atividades referente ao modo de jogo.

4.5. Diagramas de Sequência

Com os diagramas de *use case* textuais podemos passar à especificação “dinâmica”, ou seja, ao comportamento do sistema.

Começamos então a fazer a especificação através dos diagramas textuais dos *use cases* descritos até agora, criando diagramas de sequência. Estes diagramas enfatizam a perspetiva temporal e permitem representar a sequência de interações, isto é, as mensagens passadas entre objetos num determinado sistema.

Fizemos apenas os diagramas de sequência das iterações que consideramos mais relevantes, nomeadamente do modo como decorre um jogo.

4.5.1 Diagrama Modo de Jogo

O diagrama *Modo Jogo* faz referência a outros diagramas, tais como *Iniciar Jogo*, *Resolver Desafio*, *Selecionar Próximo Destino* e *Desistir Jogo*.

Tal como já foi referido anteriormente, apenas um utilizador registado pode realizar jogos. O próximo diagrama retrata o possível comportamento desse jogador durante um jogo.

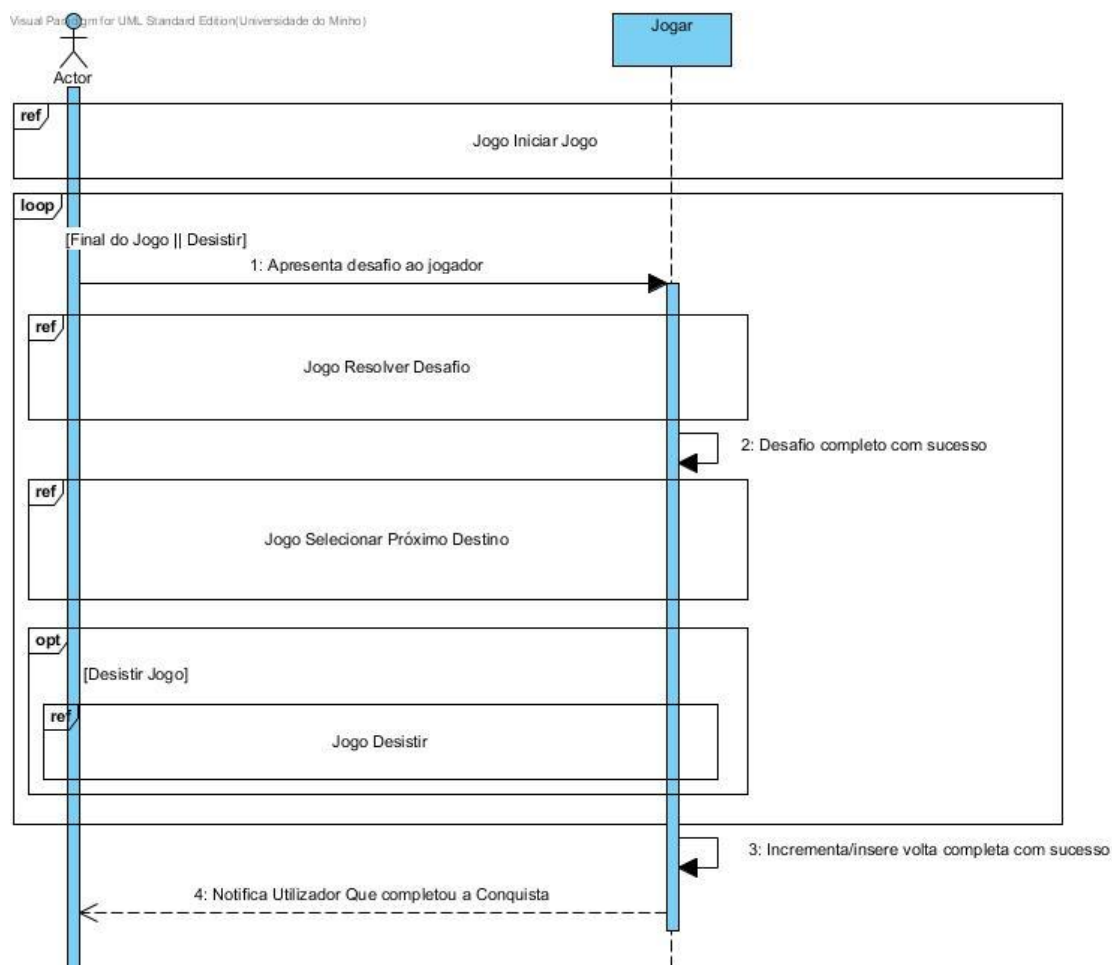


Figura 38 - Diagrama de sequência referente ao Modo de Jogo.

4.5.2 Diagrama Iniciar Jogo

Ao iniciar um novo jogo, o utilizador necessita de tomar imediatamente algumas decisões, nomeadamente, escolher com que personagem pretende jogar, o nível de dificuldade qualitativa que está disposto a enfrentar e o continente onde quer começar a sua jornada. Após esses passos, já se encontra em condições de partir à conquista do mundo.

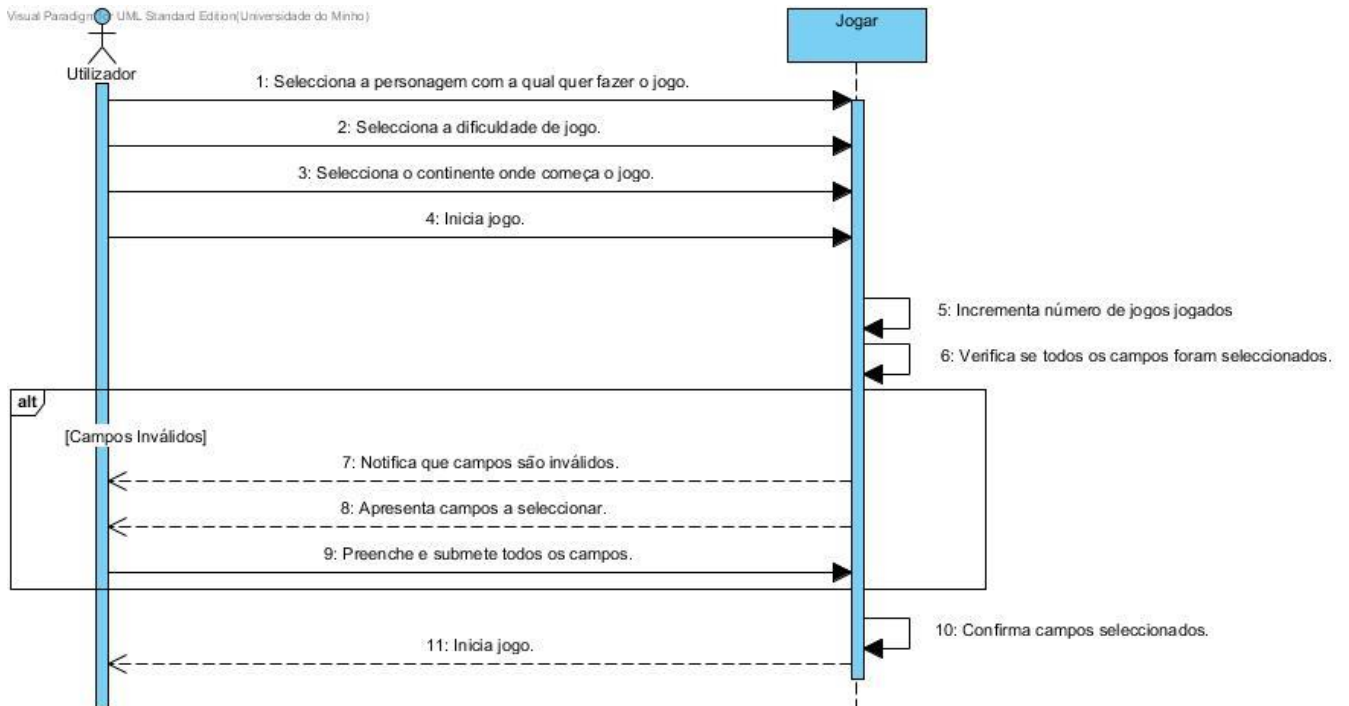


Figura 39 - Diagrama de sequência referente ao início de jogo.

4.5.3 Diagrama Resolver Desafio

Neste momento, o jogador já se encontra num continente, mas precisa de escolher o país/área protegida em que quer provar os seus conhecimentos. Após essa escolha, é-lhe apresentado o desafio, ao qual tenta responder.

Se responder erradamente e o seu número de pontos for subtraído para um valor abaixo de zero, o sistema informa-o da impossibilidade de realizar mais desafios e o jogo termina.

Se responder erradamente mas a consequente subtração dos pontos perdidos ao total de pontos do jogador resultar num valor acima de zero, o sistema apenas regista o insucesso de resposta na área protegida em questão e apresenta um novo desafio.

Se a resposta for acertada, o seu número de pontos é atualizado e é ainda feito o registo do sucesso obtido na área protegida em questão.

Por fim, a utilização de ajudas por parte do jogador está descrita no diagrama *Utilizar Ajuda*.

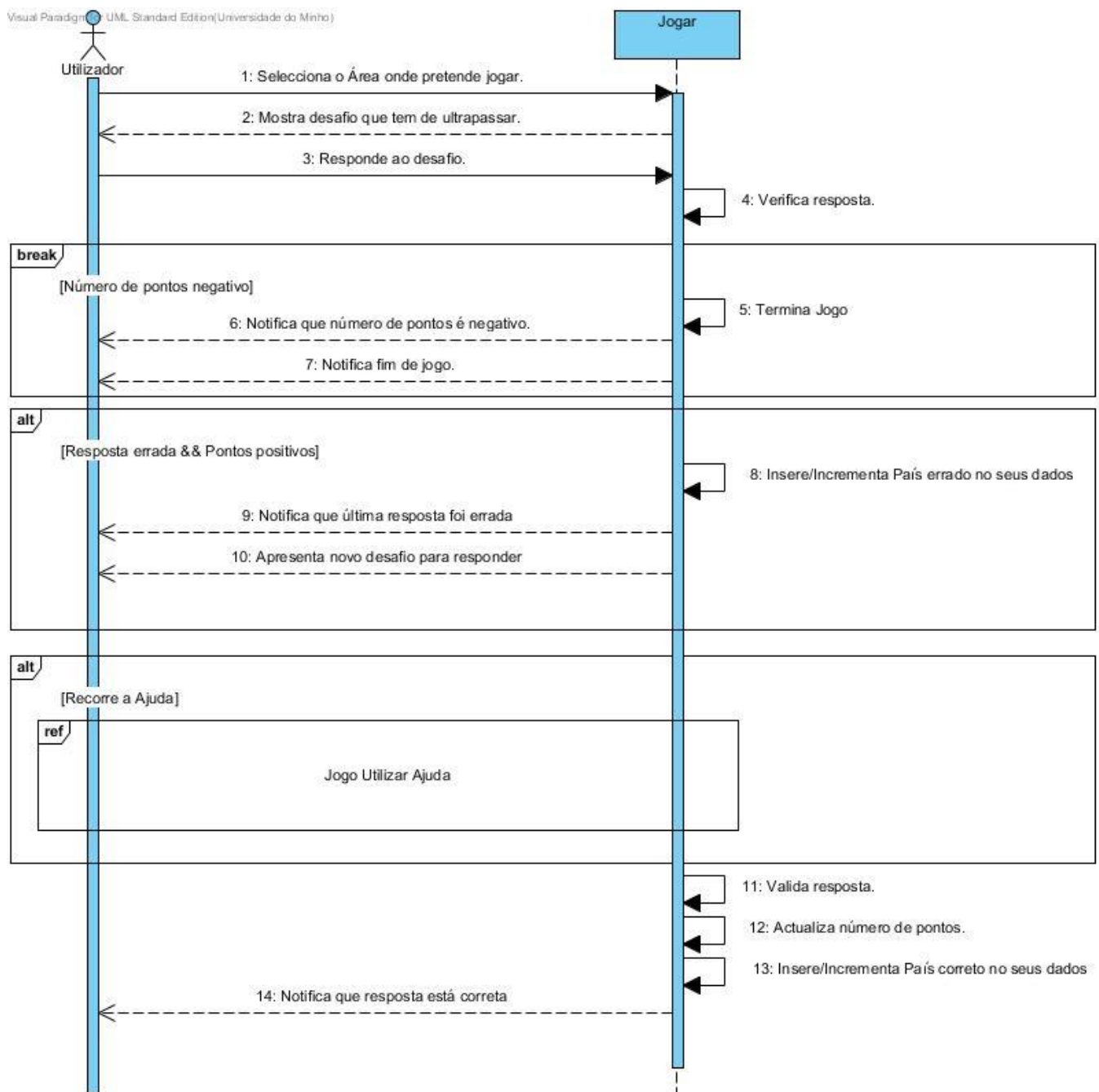


Figura 40 - Diagrama de sequência referente a resolver desafio.

4.5.4 Diagrama Utilizar Ajuda

Quando o jogador não sabe responder a um desafio, pode sempre optar por utilizar uma ajuda.

Nesse caso, primeiramente tem de seleccionar qual dos três graus de ajuda precisa de usar. O sistema verifica se o jogador tem pontos suficientes para utilização da ajuda escolhida. Caso tenha, valida a ajuda, o número de pontos são atualizados e o jogador recebe a ajuda. Por outro lado, caso não tenha pontos suficientes, o sistema cancela a operação e avisa o jogador da impossibilidade da sua escolha.

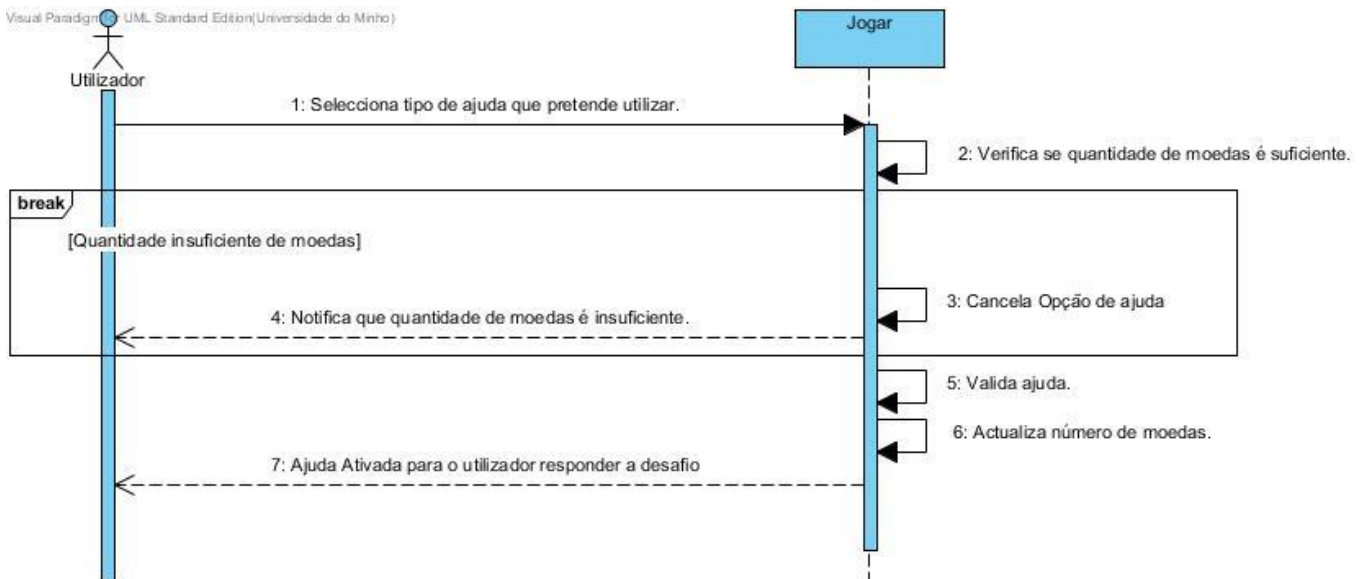


Figura 41 - Diagrama de sequência referente à utilização de ajuda.

4.5.5 Diagrama Selecionar Próximo Destino

Ao concluir o desafio de um dado país/área reservada, o jogador escolhe para qual dos países apresentados pretende ir a seguir. A única função do sistema é atualizar o jogo.

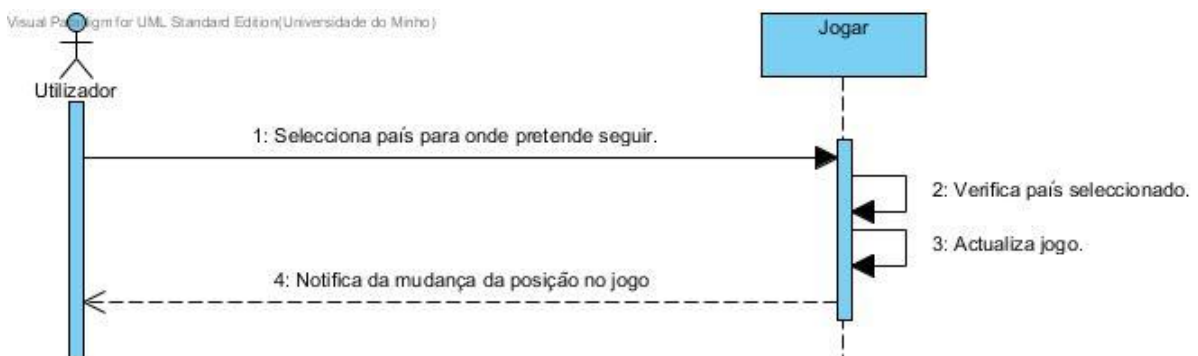


Figura 42 - Diagrama de sequência referente a seleccionar o próximo destino.

4.5.6 Diagrama Desistir do Jogo

A qualquer momento, o jogador pode decidir abandonar o jogo. Para isso, basta informar que assim o deseja. Caso o utilizador confirme, o jogo é cancelado e jogador volta à página inicial, à dashboard de jogo. Caso o utilizador opte por continuar, o sistema cancela a operação de cancelar o jogo e retoma o jogo.

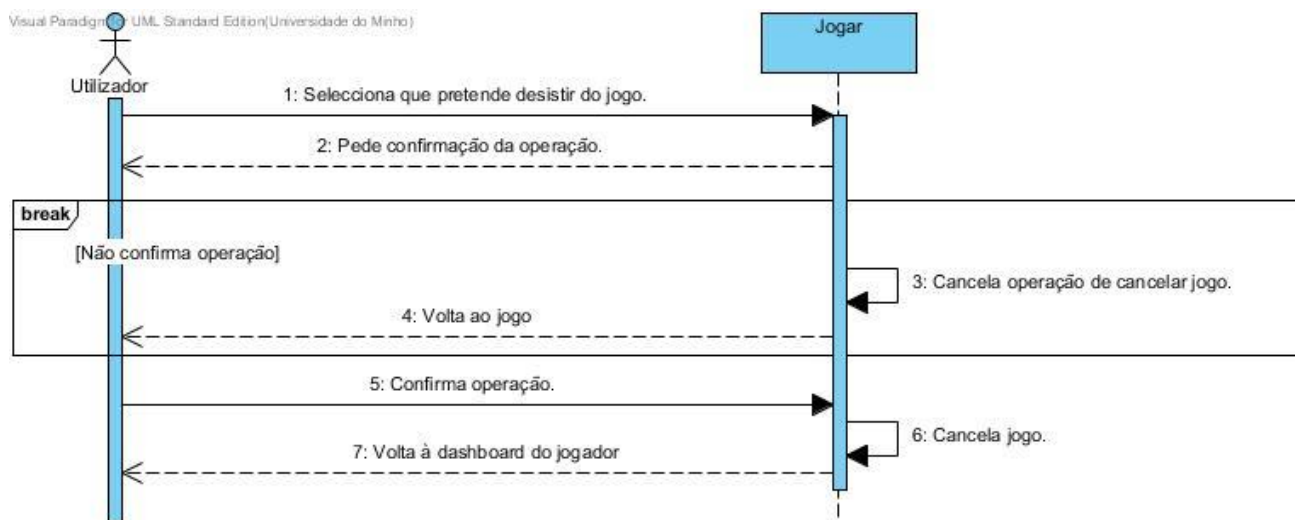


Figura 43 - Diagrama de sequência referente a desistir do jogo.

4.5.7 Diagrama Adicionar Desafio

Para cada país do mapa é possível adicionar um novo desafio (pergunta), ficando este associado a uma área protegida desse mesmo país. Sendo assim, ao adicionar uma nova pergunta, o administrador terá de seleccionar o país e posteriormente a área protegida. Também terá de identificar a dificuldade e o tipo da pergunta. Caso algum destes campos for preenchido de forma errada, o administrador será notificado e poderá corrigir o erro (máximo de 3 vezes).

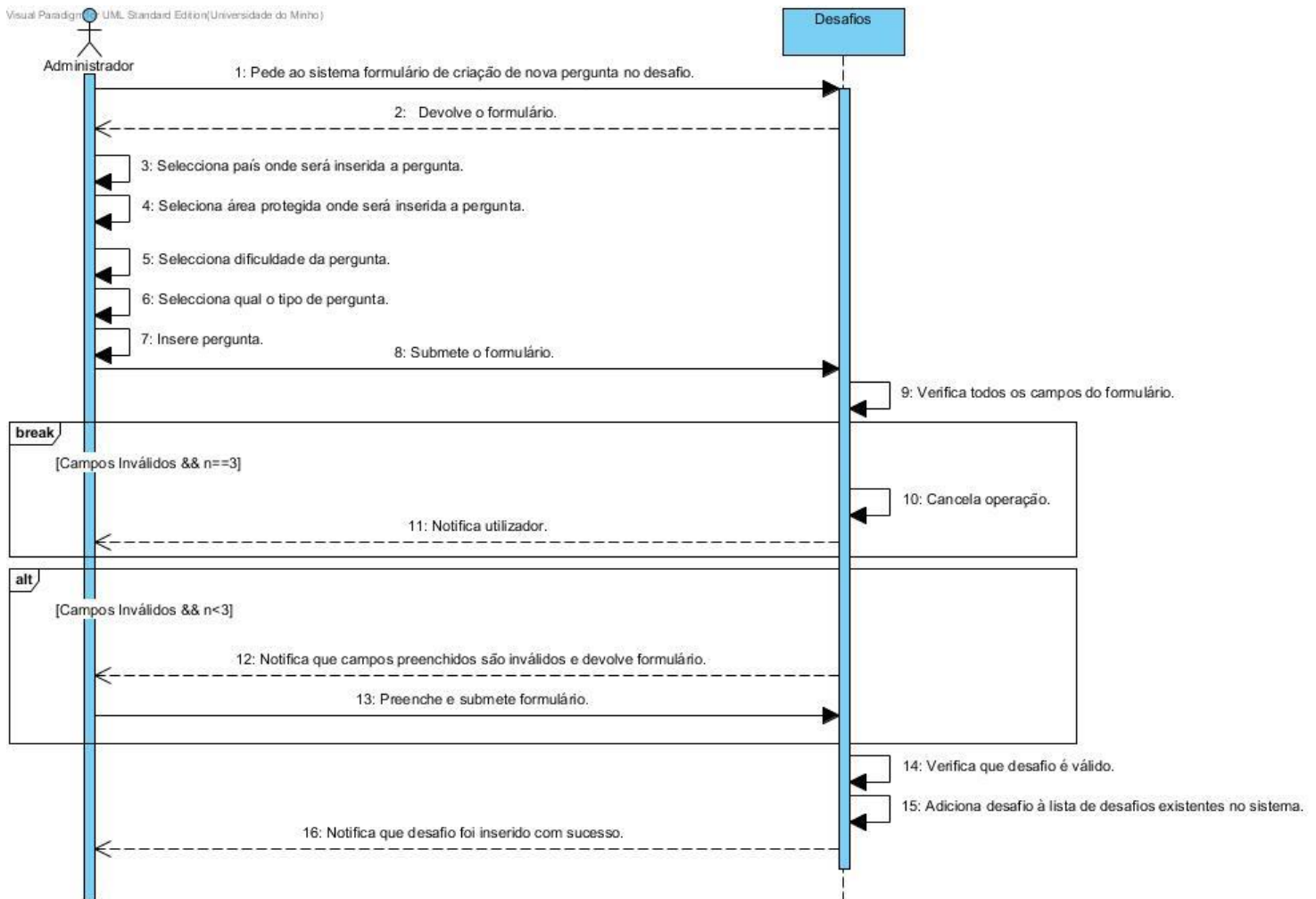


Figura 44 - Diagrama de sequência referente a adicionar desafio.

4.5.8 Diagrama Editar Desafio

Caso seja necessário modificar algum desafio existente no jogo, é possível editá-lo. Isto possibilita corrigir erros que não foram detetados aquando da adição do desafio.

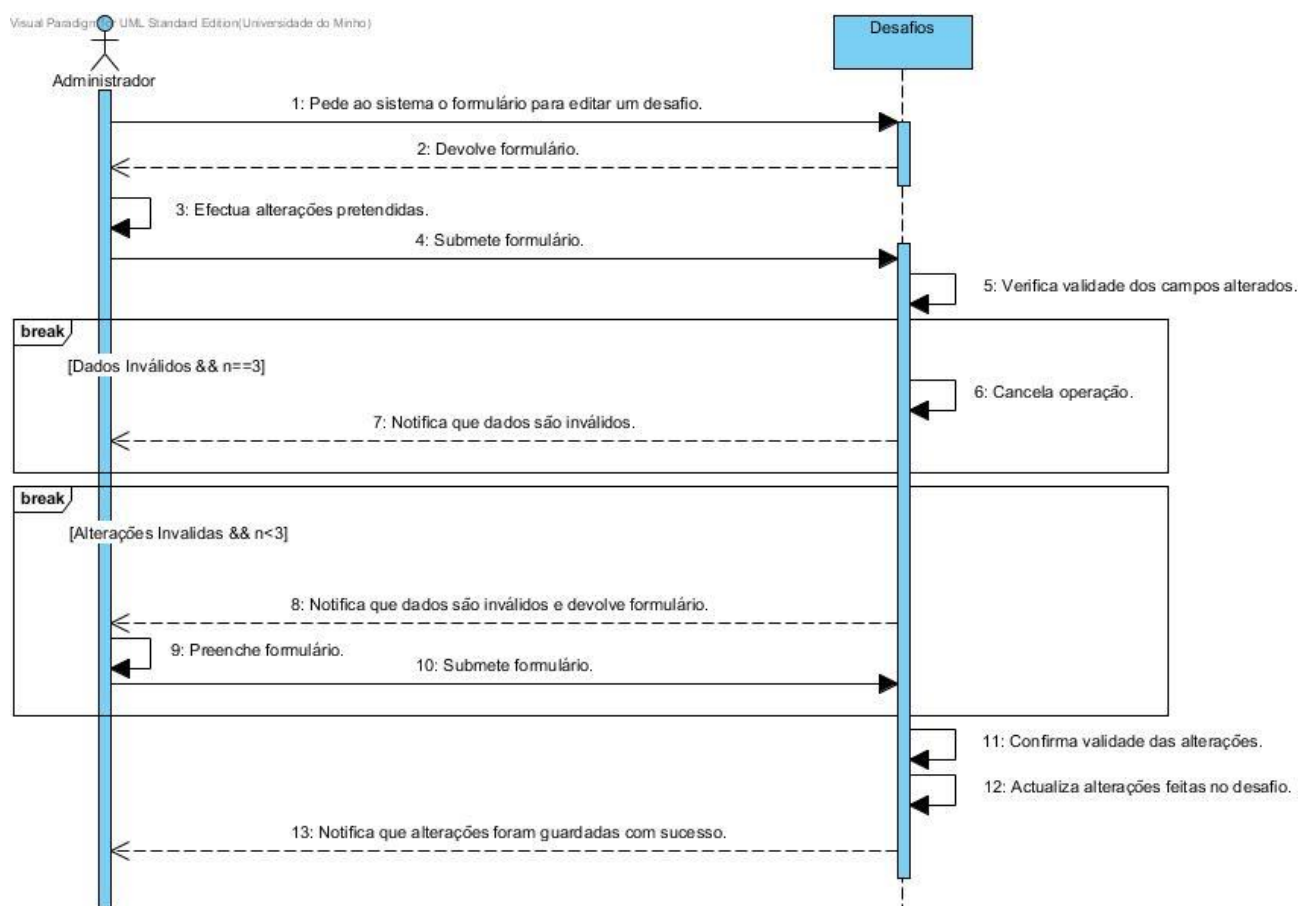


Figura 45 - Diagrama de sequência referente a editar um desafio.

4.5.9 Diagrama Remover Desafio

A qualquer momento, o administrador pode eliminar um determinado desafio. Para tal terá de seleccionar a área protegida em causa e daí escolher qual dos desafios remover. Esta opção possibilita a diversidade do jogo, já que os desafios podem ser removidos e substituídos por outros diferentes.

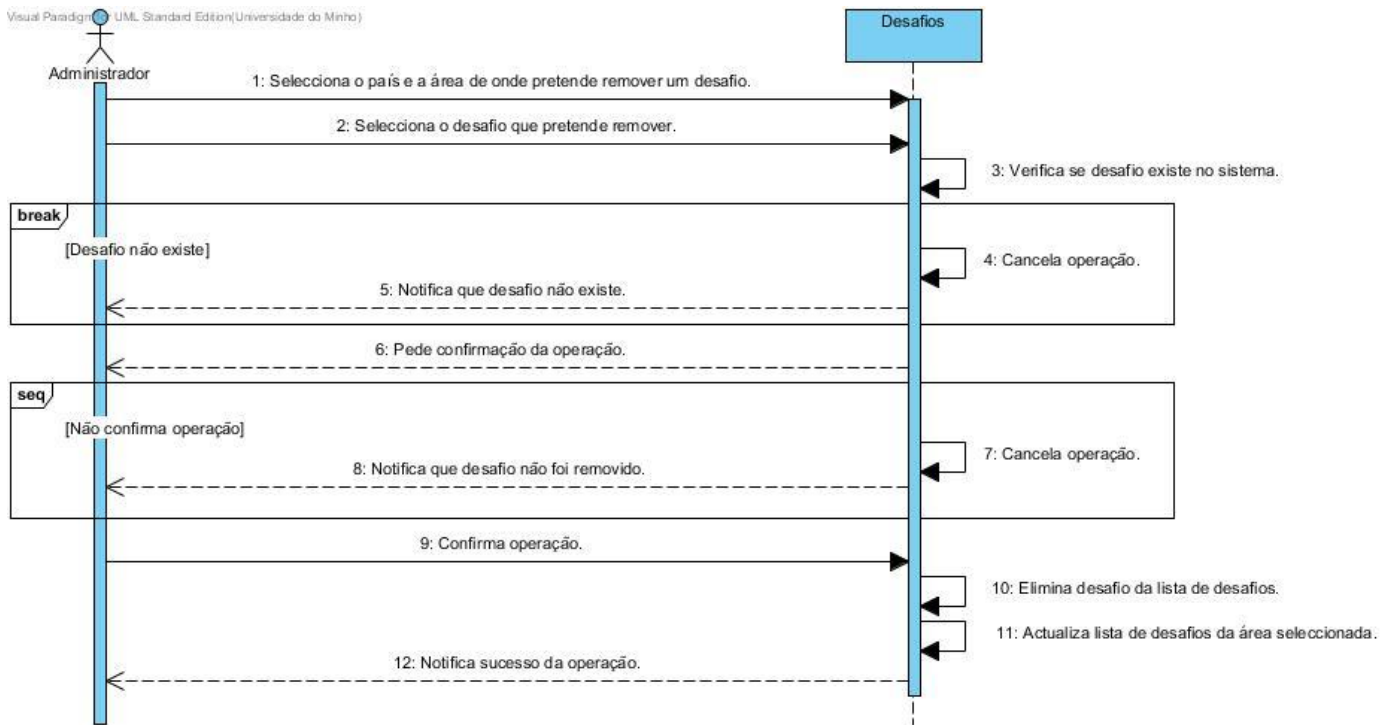


Figura 46 - Diagrama de sequência referente a remover desafio.

4.5.10 Diagrama Adicionar Área Protegida

Num país pode existir uma área protegida, várias ou até nenhuma (neste caso o país não faz parte do mapa de jogo). A qualquer momento o administrador pode adicionar uma nova área protegida, ficando este associada a um país.

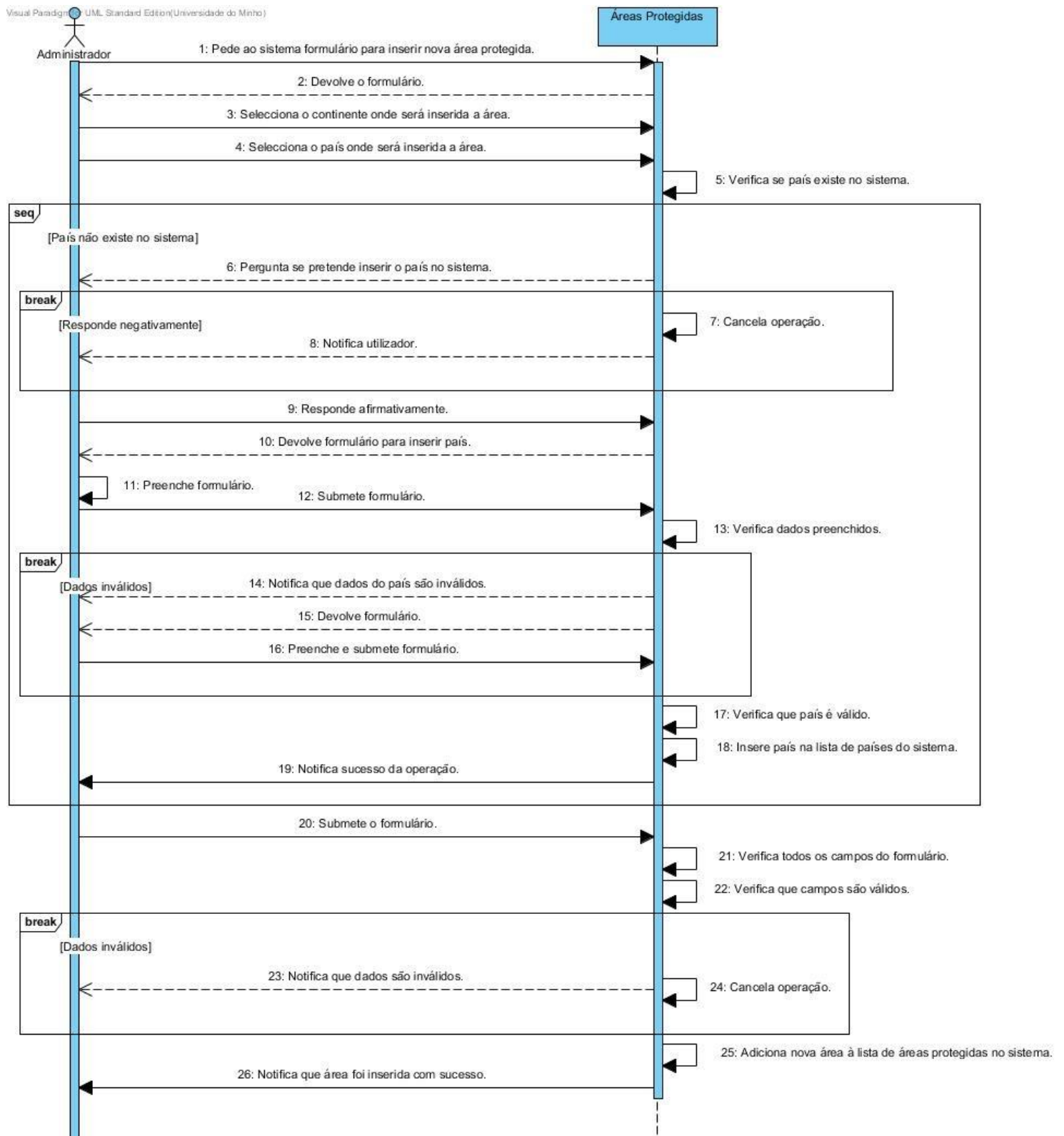


Figura 47 - Diagrama de sequência referente a adicionar área protegida.

4.5.11 Diagrama Editar Área Protegida

Por forma a permitir a correção de erros ou até mesmo por mera vontade do administrador, este pode editar os dados associados a uma determinada área protegida. Como modo de exemplo, o administrador pode alterar o nome de uma área.

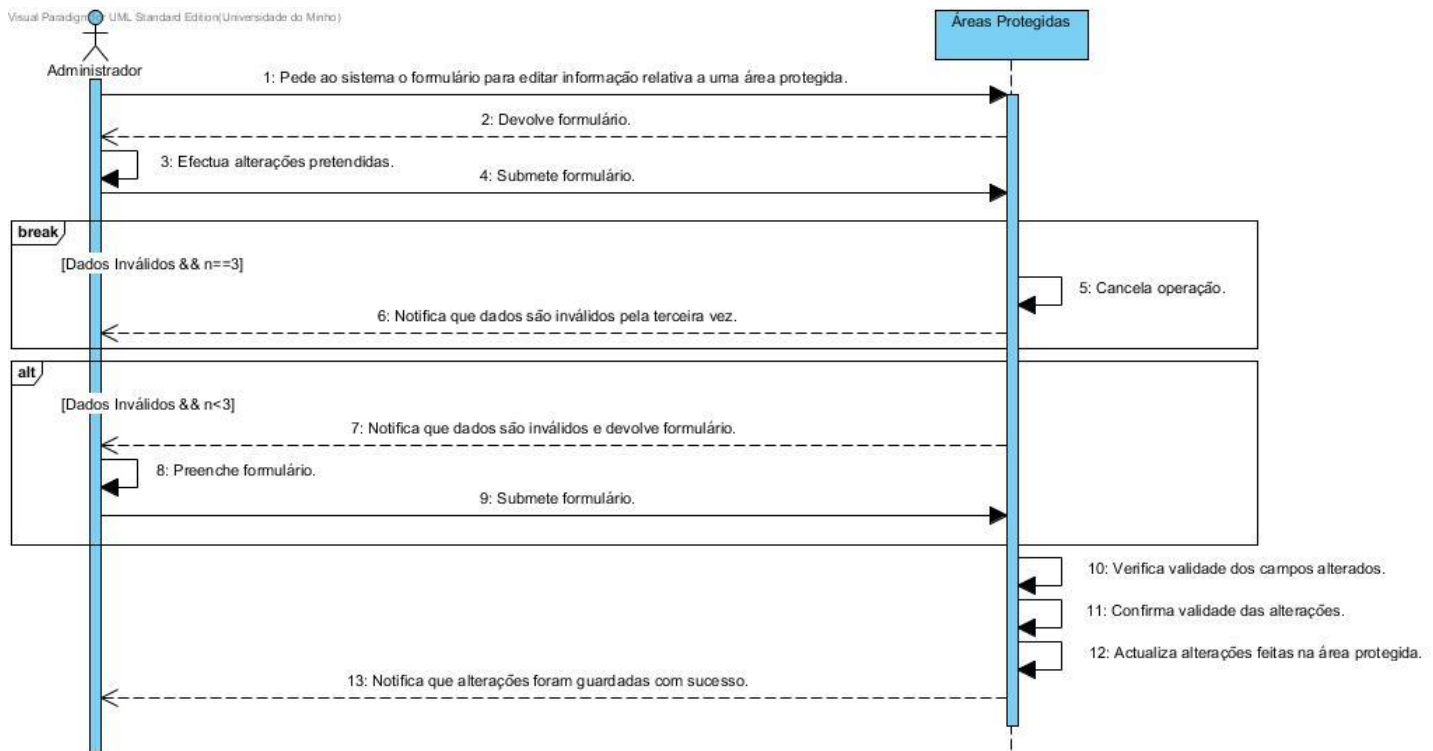


Figura 48 - Diagrama de sequência referente a editar área protegida.

4.5.12 Diagrama Remover Área Protegida

Da mesma maneira que os desafios podem ser removidos, também as áreas protegidas podem ser eliminadas. Neste caso se o país apenas possuir uma área, após a remoção desta, o país deixa de pertencer ao mapa de jogo.

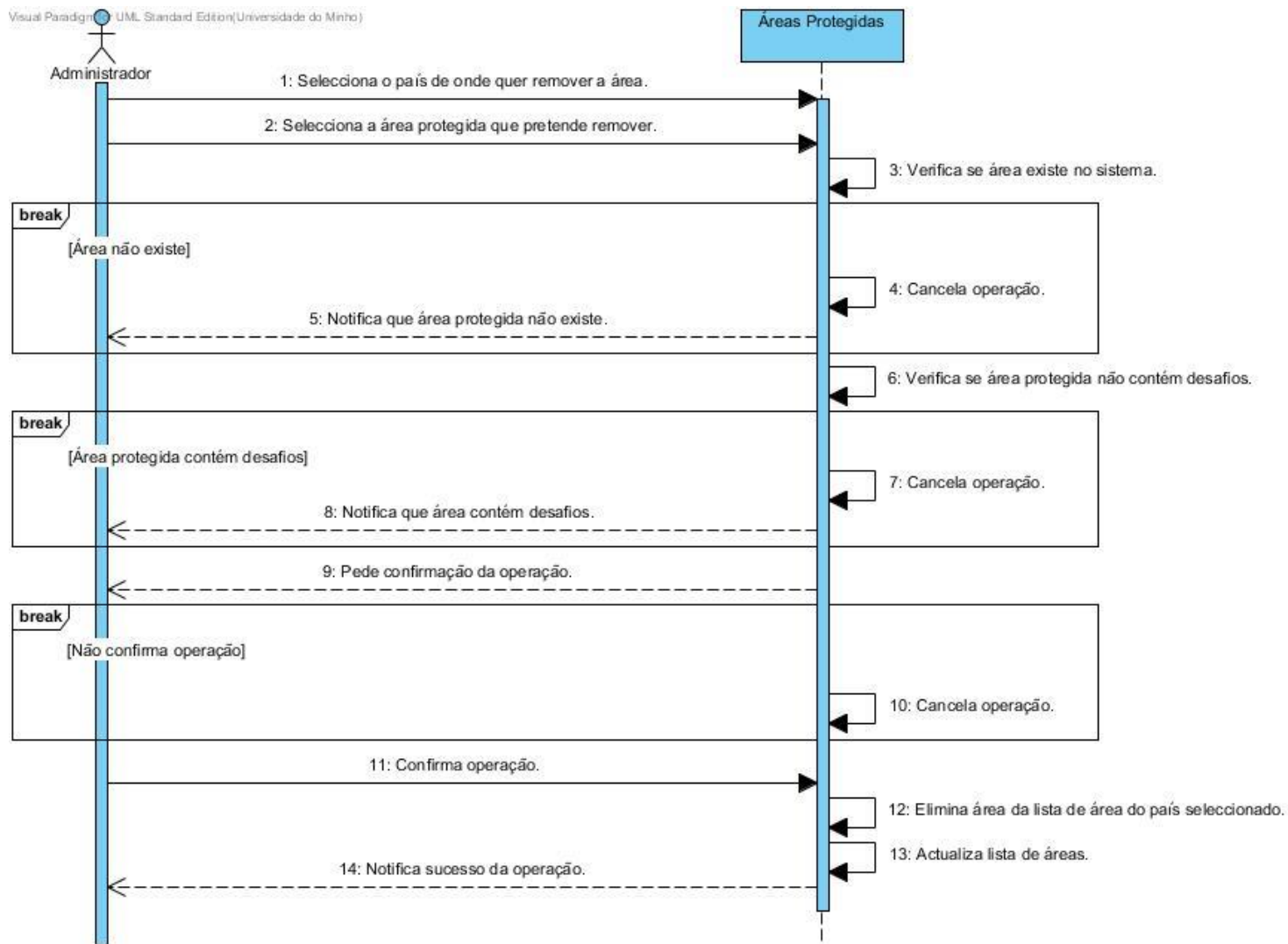


Figura 49 - Diagrama de sequência referente a remover área protegida.

4.6 Base de Dados

Nesta secção pretende-se documentar e descrever o processo de aplicação de uma metodologia de desenho e arquitetura da base de dados que dá resposta às necessidades do nosso jogo, “Animais 360”. A metodologia seguida consiste num conjunto de passos que guiam a tarefa de criar a Base de Dados segundo técnicas apropriadas a cada uma das fases do projeto.

Assim sendo, numa primeira fase torna-se vital fazer o levantamento dos requisitos, a partir do nosso caso de estudo, com o máximo rigor. “Animais 360”, é um jogo de estratégia cujo principal objetivo é completar uma volta ao Mundo respondendo a questões. Para conseguir concluir a sua árdua missão, o jogador terá de conquistar os vários continentes do planeta, mostrando que sabe tudo sobre as reservas naturais de cada uma dessas zonas e sobre os animais que nelas habitam.

4.6.1 Análise de requisitos

Neste capítulo será detalhado o processo de levantamento e análise dos requisitos relacionados com o desenvolvimento da Base de Dados que servirá de suporte ao Animais360.

De seguida, será exposto o funcionamento de todo o projeto para melhor se perceber as entidades, atributos e as suas relações.

Animais 360 será um jogo de estratégia que possibilitará ao jogador vestir a pele de uma personagem aventureira que enfrenta verdadeiros desafios com o objetivo de obter o título de conquistador do Mundo. Para que tal aconteça, a personagem terá de percorrer todos os continentes do nosso planeta, mostrando saber tudo sobre os animais e as reservas naturais que o jogo contempla.

O jogo poderá ser jogado na Web, depois de o **utilizador** se encontrar registado. A um **utilizador** está associado um nickname, um e-mail, uma password e um avatar (fotografia de perfil). Para além disso, um **utilizador** possui ainda as suas estatísticas pessoais, que vão desde o número de voltas ao mundo que já conseguiu completar ao número de jogos que este já iniciou.

Cada nova aventura iniciada pelo utilizador, cada **Jogo**, terá o total de respostas certas e erradas, pontuação, o nível (que será o indicativo da prestação média do utilizador ao longo do jogo), a personagem e a dificuldade pretendida para essa mesma aventura.

Para cada nível de dificuldade (fácil, médio e difícil) existe uma **classificação** com as melhores prestações de cada utilizador em cada nível.

Animais360 contará como tabuleiro de jogo um mapa que representa todo o globo terrestre. Contendo os vários **continentes**. Estes, por sua vez, são constituídos por um conjunto de **países** que, por último, são constituídos por **áreas protegidas**. Tanto os

continentes, países como as **áreas protegidas** têm a eles associado um nome e uma pequena descrição. Uma **área protegida** é definida pelas suas coordenadas geográficas, ou seja, a sua latitude e a sua longitude, e um conjunto de **questões**. Para além disso, tem também um estado a dar conta se essa área protegida é uma das permitidas ao jogador para responder. Já que podem existir áreas em que não existem questões, mas existem apenas com informações sobre essa reserva. Cada **questão** tem a pergunta a ser colocada ao jogador, o conjunto de hipóteses, a resposta certa, pode ou não ter uma imagem relacionada, é de um determinado tipo (perguntas com opções, sons, sombras, completar frases ou do tipo derradeiro desafio), terá também uma dificuldade quantitativa associada. Por último, uma **questão** tem várias **ajudas**, de diferentes graus, contendo pistas para o utilizador chegar à resposta certa.

De acordo com os resultados obtidos por cada **utilizador** em cada **área protegida**, existirá uma série de **dados estatísticos** contendo o número de respostas certas e erradas, e o número de vezes que respondeu a questões naquele local.

4.6.2 Modelação conceptual da base de dados

Neste capítulo iremos descrever com detalhe a nossa abordagem em relação à Modelação Conceptual no contexto deste projeto. Esta fase é totalmente guiada pela análise de requisitos realizada anteriormente.

4.6.2.1 Identificação das entidades

- **Utilizador**

Entidade que representa todos os tipos de utilizadores no sistema.

- **Continente**

Entidade que se refere ao conjunto de continentes existente no nosso planeta, que serão representados no sistema.

- **País**

A entidade país abstrai o conceito de um país.

- **Área Protegida**

Esta entidade representa as reservas naturais protegidas existentes no sistema.

- **Questão**

Entidade que agrega um conjunto de questões/desafios específicas de cada área protegida.

- **Ajuda**

Esta entidade inclui todas as ajudas possíveis para solucionar uma dada questão do sistema.

- **Jogo**

Por um jogo compreende-se que é uma entidade que acompanha o desenvolvimento do jogador numa determinada aventura do sistema.

- **Classificação**

Entidade que visa representar estatisticamente a performance dos jogadores.

4.6.2.2 Identificação dos relacionamentos entre entidades

Neste passo vão ser identificados os relacionamentos entre entidades, tomando como referência o texto produzido durante o levantamento e análise de requisitos.

- **Utilizador (1) - joga - Jogo (N)**

Este relacionamento traduz o facto de um utilizador poder jogar várias vezes em momentos diferentes. Um utilizador tem assim um conjunto de jogos jogados.

- **Utilizador (1) - tem - Classificacao (N)**

Um utilizador de acordo com a sua performance no jogo, vai ter a sua melhor prestação para os vários níveis de dificuldade guardados.

- **Utilizador (N) - responde em - AreaProtegida (M)**

Este relacionamento permite que um utilizador responda a questões em várias áreas protegidas, e ao mesmo tempo, uma área protegida possua vários utilizadores a responder às “suas” questões.

- **Continente (1) - tem - Pais (N)**

Relacionamento que dá conta que um continente pode ter uma série de países associado.

- **Pais (1) - tem - AreaProtegida (N)**

Tal como um continente tem vários países, um país pode ter várias áreas protegidas.

- **AreaProtegida (1) - tem - Questao (N)**

Uma área protegida tem a si associada um conjunto de questões.

- **Questao (1) - tem - Ajuda (N)**

Uma questão tem a si associada um conjunto de ajudas.

4.6.2.3 Identificação e descrição dos atributos das entidades e relacionamentos

Neste passo vai ser identificado o conjunto de atributos associados a cada uma das entidades.

- **Utilizador**

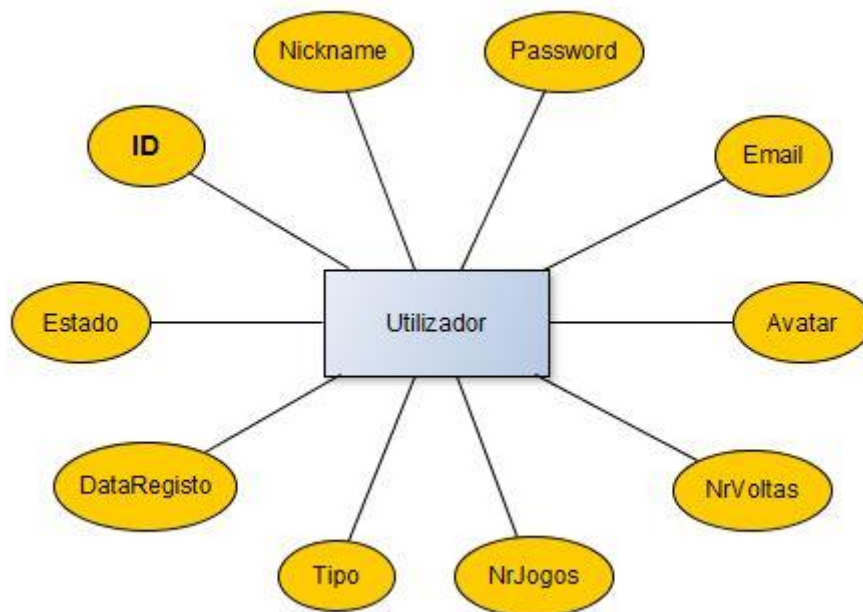


Figura 50 - Atributos pertencentes à entidade Utilizador.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ Nickname;
- ✓ Password;
- ✓ Email;
- ✓ Avatar;
- ✓ NrVoltas (número de voltas ao Mundo, total);
- ✓ NrJogos (número de jogos realizados);
- ✓ Estado (ativo ou não);
- ✓ Tipo (Admin ou não);
- ✓ DataRegisto.

- **Jogo**

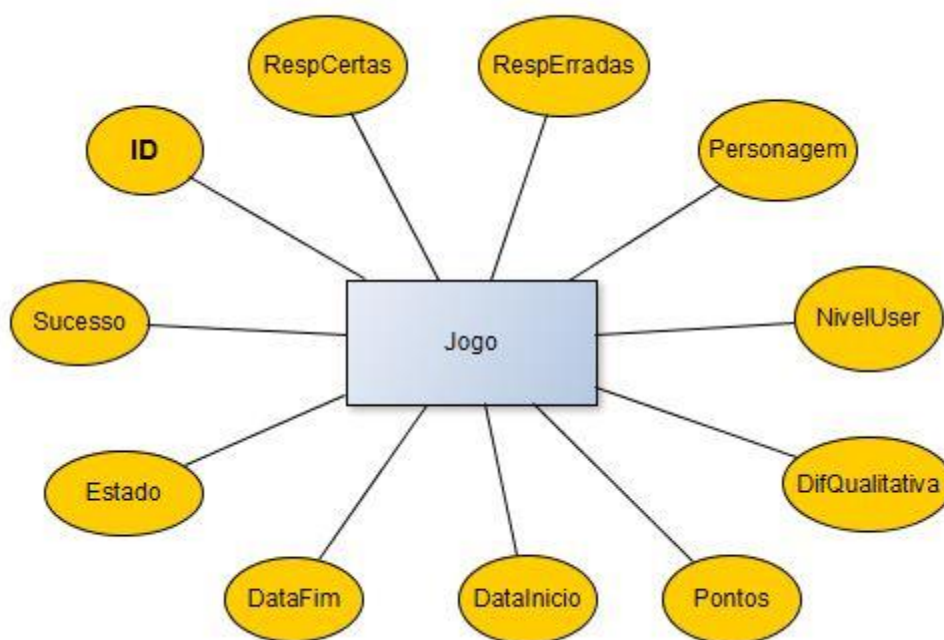


Figura 51 - Atributos pertencentes à entidade Jogo.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ RespCertas (respostas certas);
- ✓ RespErradas (respostas erradas);
- ✓ Personagem ("boneco" escolhido para a aventura);
- ✓ NivelUser (verificando as respostas dadas pelo o utilizador no jogo, existe uma avaliação do nível a que o utilizador se encontra para ser feita uma escolha de perguntas futuras);
- ✓ Pontos (pontuação);

- ✓ DifQualitativa (dificuldade escolhida);
- ✓ Estado (terminado ou não);
- ✓ DataInicio;
- ✓ DataFim;
- ✓ Sucesso (terminado com sucesso ou não).

- **Classificação**

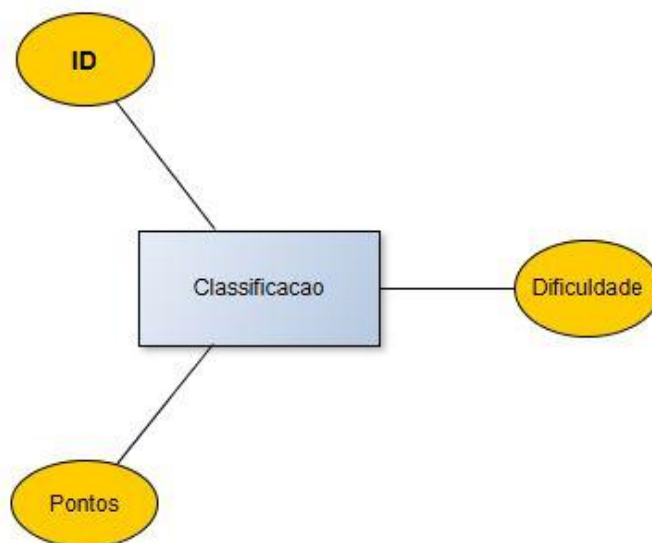


Figura 52 - Atributos pertencentes à entidade Classificação.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ Dificuldade;
- ✓ Pontos.

- **Continente**

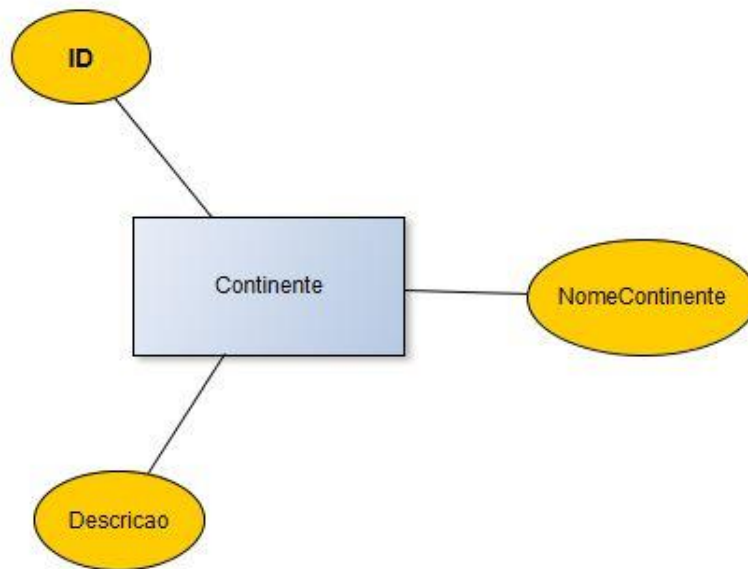


Figura 53 - Atributos pertencentes à entidade Continente.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ NomeContinente;
- ✓ Descricao.

- **País**

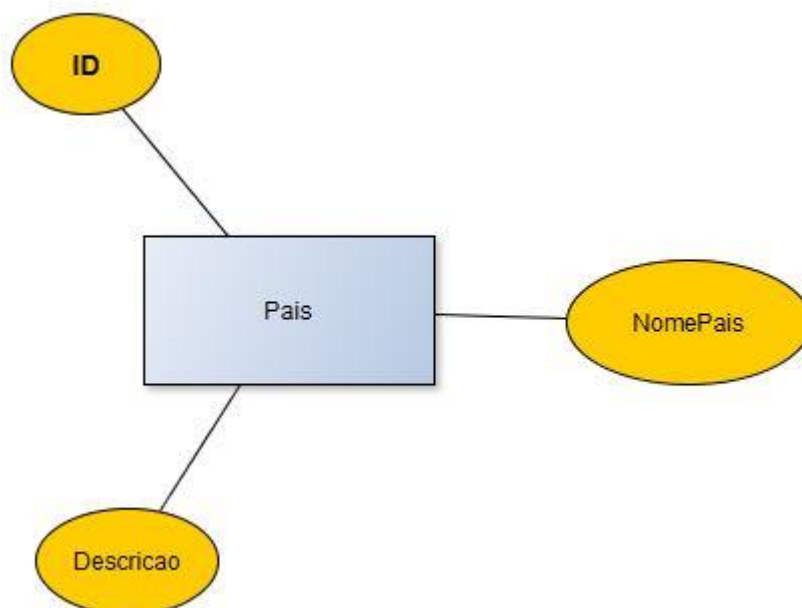


Figura 54 - Atributos pertencentes à entidade País.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ NomePais;
- ✓ Descricao.

- **Ajuda**

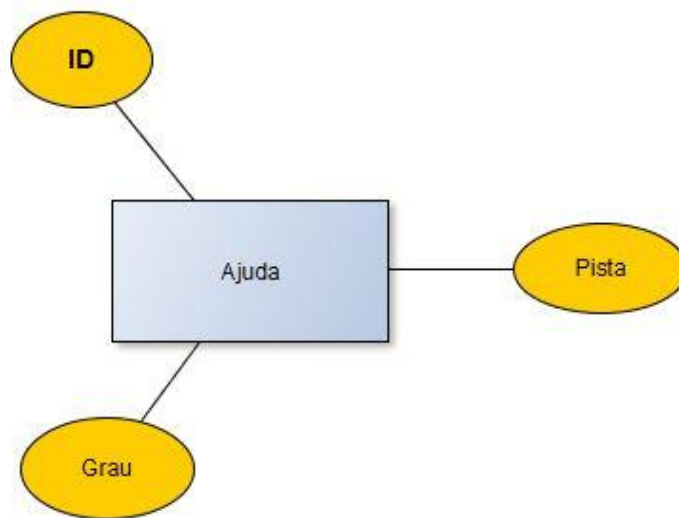


Figura 55 - Atributos pertencentes à entidade Ajuda.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ Grau (nível de ajuda);
- ✓ Pista.

- **AreaProtegida**

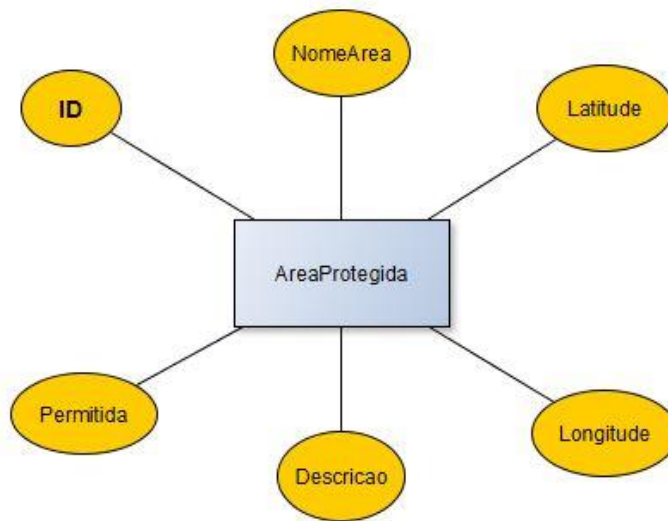


Figura 56 - Atributos pertencentes à entidade AreaProtegida.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ NomeArea;
- ✓ Latitude;
- ✓ Longitude;
- ✓ Descricao.
- ✓ Permitida (se é uma área em que existem questões, visto que iremos ter casos em a área existe e está referenciada, mas não terá questões).

- **Questao**

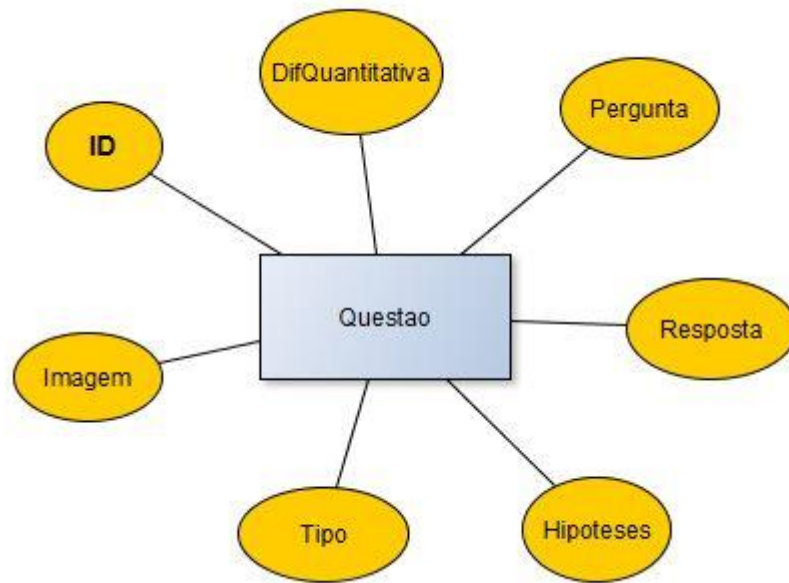


Figura 57 - Atributos pertencentes à entidade Questao.

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ difQuantitativa (dificuldade da questão);
- ✓ Pergunta (enunciado);
- ✓ Resposta;
- ✓ Hipoteses (hipóteses de resposta);
- ✓ Tipo;
- ✓ Imagem.

- **Utilizador (N) - Responde em - AreaProtegida (M)**

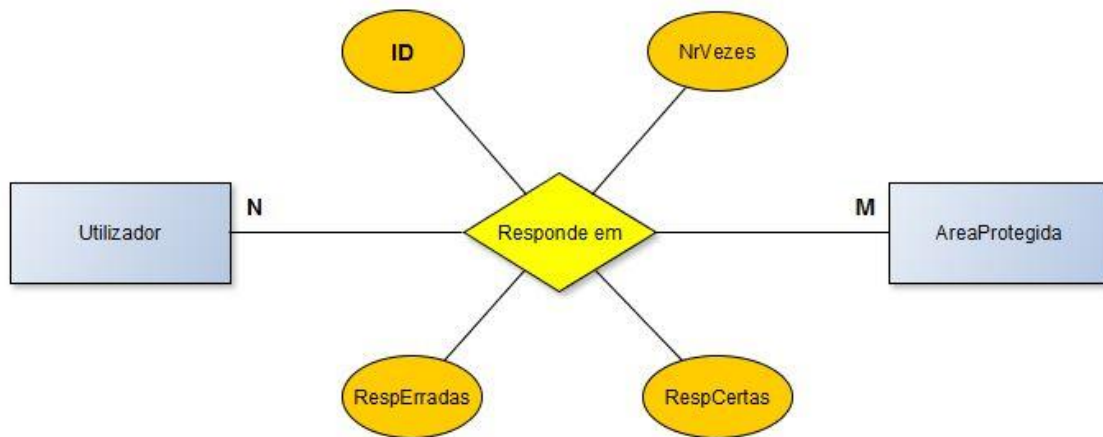


Figura 58 - Atributos pertencentes ao relacionamento Utilizador(N) – Responde em – AreaProtegida (M).

Esta entidade é definida com o seguinte conjunto de atributos:

- ✓ Id;
- ✓ RespCertas (respostas certas);
- ✓ RespErradas (respostas erradas);
- ✓ NrVezes (número de vezes que o utilizador respondeu nessa área).

4.6.3 Modelação lógica da base de dados

A representação das relações lógicas entre as entidades/tabelas segue um conjunto de normas estipuladas:

- A chave primária é identificada pelo sublinhado;
- as chaves estrangeiras são identificadas pelo itálico;
- as chaves estrangeiras que também são chaves primárias são identificadas pelo sublinhado e itálico.

4.6.3.1 Tabelas Originadas a partir de Entidades

Após análise do modelo conceptual é possível saber quais as tabelas que são originadas pelas entidades. Existem, pelo menos, 8 tabelas na base de dados:

Utilizador(Id, Nickname, Password, Email, Avatar, NrVoltas, NrJogos, Estado, Tipo, DataRegisto)

Continente(Id, NomeContinente, Descricao)

Pais(Id, NomePais, Descricao)

AreaProtegida(Id, NomeArea, Latitude, Longitude, Descricao, Permitida)

Questao(Id, DifQuantitativa, Pergunta, Resposta, Hipoteses, Tipo, Imagem)

Ajuda(Id, Grau, Pista)

Jogo(Id, RespCertas, RespErradas, Personagem, Pontos, DifQualitativa, Estado, Sucesso, NivelUser, DataInicio, DataFim)

Classificacao(Id, Dificuldade, Pontos)

4.6.3.2 Tabelas Originadas a partir de Relacionamentos

São tabelas que são geradas através de relações entre tabelas.

UtilizadorAreaProtegida(Id, RespCerta, RespErrada, nrVezez)

4.6.3.3 Relacionamentos

- **Relações “1 para 1”**

Este tipo de relacionamento não altera a estrutura da tabela. Na base de dados, referente a este projeto, não irão existir deste tipo de relações.

- **Relações “1 para N”**

Neste tipo de relacionamento entre as entidades, é criado na tabela que tem cardinalidade N um novo atributo, designado de chave estrangeira com ligação a tabela com cardinalidade 1.

Continente (1) - Pais (N)

Pais(Id, NomePais, Descricao, *IdContinente*)

Pais (1) - AreaProtegida (N)

AreaProtegida(Id, NomeArea, Latitude, Longitude, Descricao, Permitida, *IdPais*)

AreaProtegida (1) - Questao (N)

Questao(Id, DifQuantitativa, Pergunta, Resposta, Hipoteses, Tipo, Imagem, *IdArea*)

Questao (1) - Ajuda (N)

Ajuda(Id, Grau, Pista, *IdQuestao*)

Utilizador (1) - Jogo (N)

Jogo(Id, RespCertas, RespErradas, Personagem, Pontos, DifQualtitativa, Estado, Sucesso, NivelUser, DataInicio, DataFim, *IdUtilizador*)

Utilizador (1) - Classificao (N)

Classificacao(Id, Dificuldade, Pontos, *IdUtilizador*)

- **Relações “N para M”**

Estes relacionamentos dão origem a uma nova tabela, que é composta por duas chaves estrangeiras que se relacionam com cada uma das tabelas envolvidas.

Utilizador (N) - AreaProtegida (M)

UtilizadorAreaProtegida(Id, IdUser, IdAreaProtegida, RespCerta, RespErrada, nrVezes)

4.6.3.4 Validação das relações recorrendo às Regras de Normalização

Os principais objetivos da normalização de um esquema são:

- Certificar que na arquitetura da Base de Dados, o esquema tem o menor número possível de atributos necessários para suportar os requisitos de armazenamento de dados do caso de estudo.

- Garantir o mínimo de redundância nos dados de forma a maximizar a consistência. Alguma redundância, no entanto, é necessária para permitir a existência de relacionamentos entre as tabelas. O processo de normalização de um dado conjunto de relações obedece a um conjunto de Formas Normais, cada uma definida com o objetivo de reduzir um tipo de redundância.

Normalmente o processo de verificação de normalização de tabelas fica-se pela 3ª Forma Normal, pois uma Base de Dados nesta Forma não possuirá problemas com redundância de dados. O processo utilizado garante que, devido à utilização de regras bem definidas, a passagem do modelo conceptual para o lógico dê origem a esquemas Lógicos que se encontram na 3ª Forma Normal.

4.6.4 Modelo Físico

Após a conclusão da modelação do modelo lógico conseguiu-se fazer o modelo físico da base de dados. Apresenta-se assim, os elementos do modelo lógico e a sua interpretação no modelo físico por via de tabelas e as transações existentes.

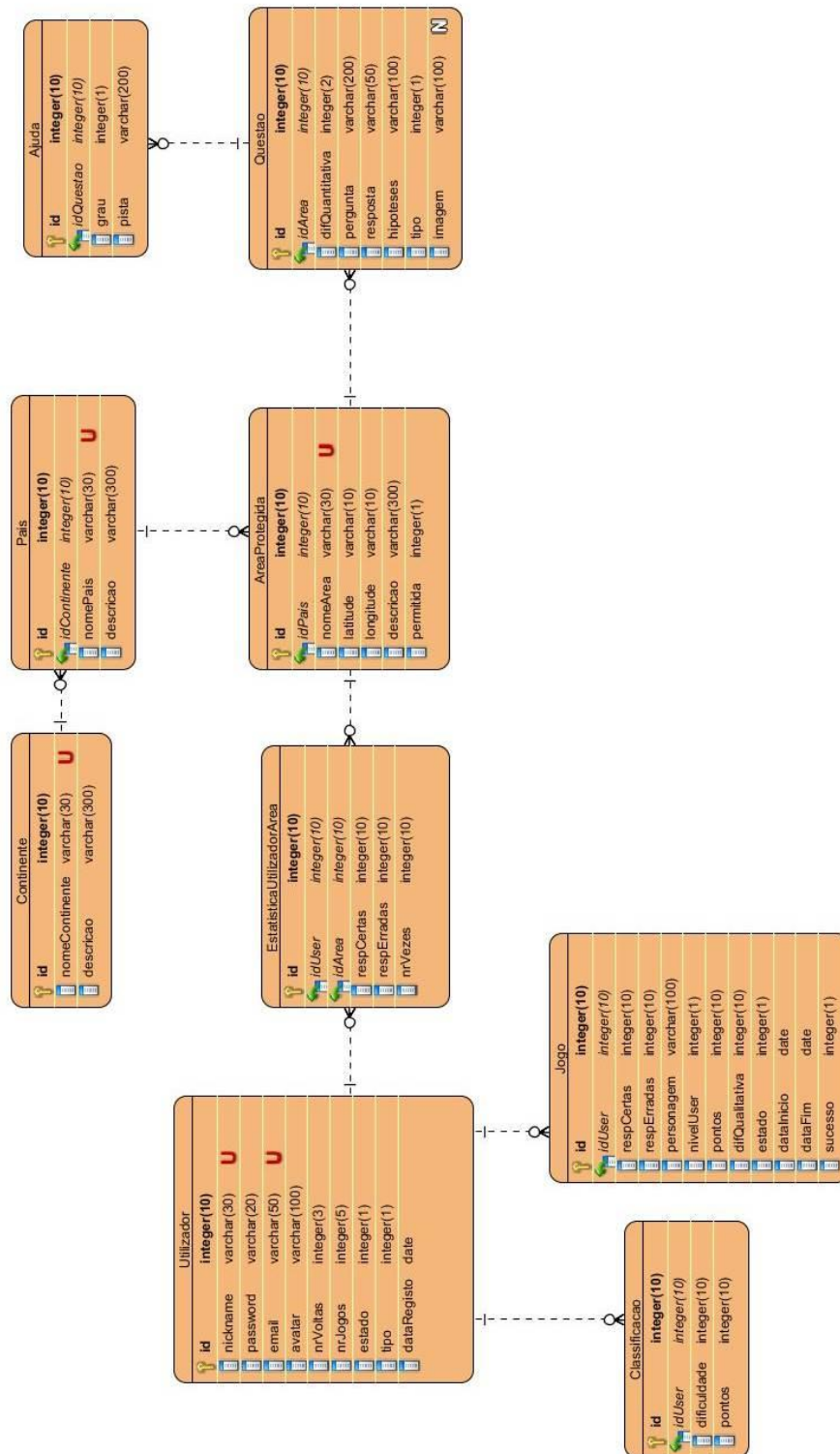


Figura 59 - Modelo físico da base de dados.

5 Desenvolvimento do Projeto

5.1 Modelo de gestão utilizado

Logo desde o arranque desta Unidade Curricular foi-nos ensinada a importância de adotar uma metodologia para a construção de *Software*. Em anos anteriores, conceitos como: planeamento, levantamento de requisitos, protótipos, etc. eram raramente postos em prática. A maior parte das vezes eramos impulsionados a “meter a mão na massa” antes mesmo de ter uma solução definida, o que só nos prejudicava em fases mais avançadas do projeto. Nesse sentido e porque a disciplina assim o obriga, desta vez tivemos de alterar a forma habitual como fazíamos as coisas. Resolvemos portanto adotar uma metodologia para o desenvolvimento do nosso projeto. O modelo utilizado foi então o modelo em cascata:

“O modelo em cascata é um modelo de construção de Software sequencial, no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das várias fases que o constituem.” - Wikipédia.

Uma das principais características deste modelo é que uma tarefa só pode ter início quando a tarefa anterior estiver terminada, validada e aceite pelo cliente.

As fases de desenvolvimento de uma aplicação, segundo o modelo em cascata, são:

- Levantamento e análise dos requisitos;
- Especificação e modelação das funcionalidades;
- Implementação do código;
- Teste/avaliação da solução gerada;
- Manutenção (melhorias e correções).

As principais características de um modelo deste tipo e as suas várias fases podem ser facilmente transportadas e relacionadas com todo o processo de desenvolvimento que o nosso projeto sofreu, tal como é demonstrado na seguinte figura. Convém ainda referir que fomos submetidos a avaliação em cada uma das fases do trabalho, respeitando um dos pressupostos desta metodologia de desenvolvimento de *Software*.

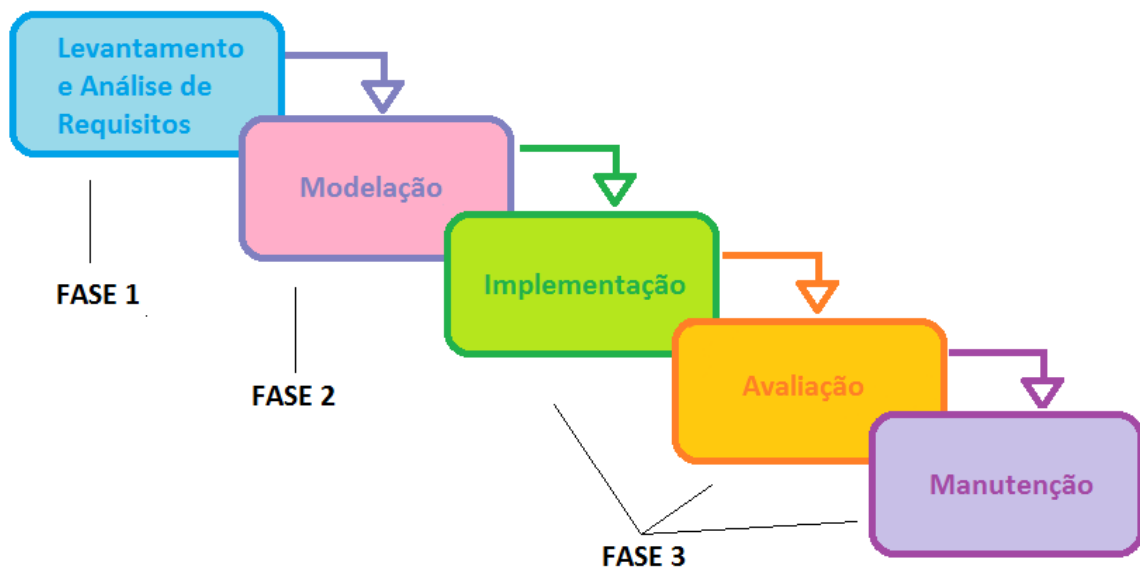


Figura 60 - Relação entre as várias fases de desenvolvimento de um modelo em cascata e as fases do projeto.

5.2 Tomada de decisões e divisão de tarefas

Nas duas primeiras fases do projeto o grupo trabalhou sempre em conjunto pois na tomada de decisões era importante a opinião e o consenso entre todos os elementos, até mesmo para as coisas mais pequenas. Contudo, nesta última fase, foi essencial haver uma divisão de tarefas, devido à carga de trabalho. Esta divisão teve em atenção o que era necessário ser feito mas mais importante ainda, a divisão de tarefas foi feita tendo em consideração qual a área em cada elemento do grupo se encontrava mais à vontade. Desta maneira, cada elemento do grupo desenvolveu sobre uma área em que se encontrava mais motivado, o que levou a um melhor desempenho de trabalho por parte de cada um e, acima de tudo, a um melhor resultado final.

Desde o início desta etapa foi essencial que o grupo de trabalho se mantivesse fiel à planificação do projeto. De maneira a garantir que tal fosse cumprido, para além da divisão das tarefas, foram também tomadas decisões acerca de como esta etapa do projeto seria desenvolvida. Mais propriamente, estabelecer prioridades e a ordem pela qual o trabalho iria ser executado. Esta divisão de trabalho possibilitou um maior rendimento do tempo de trabalho.

A primeira atitude tomada foi a escolha do tema HTML e partindo dessa decisão deu-se início ao seu desenvolvimento, tendo sempre em atenção todas as necessidades do jogo e ao que tinha sido planeado.

À medida que esta atividade se desenvolvia, eram executadas outras em paralelo. De uma forma bastante sucinta, primeiramente iniciou-se o desenvolvimento do código no *Visual Studio* com recurso a algumas *frameworks*. Um pouco depois foi implementada os modelos da Base de Dados e a sua administração. Em último lugar foi desenvolvido o método de jogar na plataforma.

5.3 Ferramentas utilizadas no projeto

Para o desenvolvimento do projeto o grupo sabia previamente que era necessário usar ferramentas desenvolvidas pela *Microsoft* na sua implementação. Atendendo a isto, foi criado o projeto em *Visual Studio* com recurso a duas *frameworks*: *ASP.NET MVC* e *Entity*.

O recurso a estas duas *frameworks* foi prioritário para esquematizar o projeto de uma forma diferente do habitual. A arquitetura do *MVC* separa a interação do utilizador com a aplicação. A parte do *Model* contém os dados da aplicação que são geridos pelo *SQL SERVER*, já na *View* é solicitado um modelo de informação, necessário para gerar uma representação de saída ao utilizador (com recurso a *HTML5*, *CSS3* e *JavaScript*). Os *Controllers* enviam comandos para alterarem/atualizarem um estado num modelo. Aqui encontra-se código em *C#* e funções que pertencem à *Entity Framework*.

Na modelagem de sistema, o uso do *Visual Paradigm* foi vital para se perceber como estes modelos iriam estar interligados.

Foi necessário, também, a utilização da API do *Google Maps* para a inclusão de um mapa no nosso jogo e definição visual de alguns elementos deste.

Num projeto desta dimensão, é importante ter associado algum controlo de versões, para isso foi essencial o uso do *GIT*¹. Para discussão e divisão de tarefas foi utilizado o *Trello*². Já na preparação de documentos e apresentações foi preferido o uso da plataforma colaborativa *Google Docs* por uma questão de facilitismo. Enquanto que em relação aos relatórios, estes foram elaborados na plataforma *Microsoft Office*.

¹ <https://github.com> – Plataforma de controlo de versões.

² <https://trello.com> – Plataforma de planificação de projetos.

5.4 Sobre o código

Para o desenvolvimento da aplicação e para corresponder a todos os critérios que tínhamos definido na análise de requisitos, foi necessário criar alguns algoritmos de verificação de dados durante a execução de um jogo.

O algoritmo responsável por alterar a dificuldade quantitativa durante uma partida acede, inicialmente, ao nível de dificuldade qualitativa que o utilizador escolheu e depois, dependendo do número de resposta certas e erradas do jogador naquele continente, o motor atribui um “grau” que será usado na busca de um novo desafio para o jogador.

Quando um jogador escolhe um continente para responder a desafios, o motor da aplicação vai verificar o número de áreas existentes neste e depois vai buscar aleatoriamente algumas áreas (não repetidas) que o jogador tem obrigatoriamente de passar.

Assim que um jogador dá início a um desafio numa determinada área, o motor acede ao número de perguntas que este detém para assim seleccionar aleatoriamente um desafio. Neste desafio vêm também incluídas as ajudas possíveis para responder à questão. Quando um jogador falha na resposta e caso este ainda tenha saldo positivo de pontos, o motor procura um novo desafio, caso contrário o jogo é terminado e guardado no sistema como finalizado sem sucesso. Caso a resposta seja válida então o número de pontos é incrementado, sendo que este número de pontos varia com os níveis de dificuldade qualitativa e com o eventual uso de ajudas. Se todos os continentes já tiverem sido concluídos satisfatoriamente é então guardado na plataforma para um determinado utilizador concluir com sucesso uma aventura.

6 Custo final do projeto

No início do projeto foram, desde logo, estipuladas três fases de entregas com três prazos de entrega diferentes.

Para o cumprimento desses prazos, foi necessário o grupo organizar-se tentando conciliar as disponibilidades de cada elemento. De salientar que o grupo não encontrou grandes dificuldades em fazer várias reuniões e em se encontrar para trabalhar.

Na fase da fundamentação do projeto, primeira fase, estavam previstas três semanas de trabalho. Sendo esta a etapa onde se iria fazer tomar as decisões mais importantes, era bastante conveniente reunir todos os elementos do grupo para fazer um levantamento do maior número de ideias que acentassem nas bases do trabalho em questão. Dos vários debates realizados começou a surgir a caracterização geral da aplicação a desenvolver, tendo por base os termos de modelo de negócio pretendidos.

Na segunda fase, de acordo com as decisões tomadas na primeira, era necessário fazer a modelação do projeto. Modelação essa que contribui e muito para o sucesso do *Software*. Utilizou-se uma linguagem de modelagem que fosse dotada de diagramas porque facilita o entendimento e padroniza a comunicação do problema. Com esta análise de estudo garantiu-se que não iam ocorrer erros ou possíveis imprevistos em fases avançadas do projeto. No final desta etapa de modelação, o grupo entendeu que um programador tinha os elementos suficientemente especificados para implementar o que se pretendia.

Na terceira e última etapa, tinha-se quatro semanas para construir tudo aquilo que já estava especificado nas fases anteriores. Com as tarefas efetuadas até aqui, o trabalho estava bastante facilitado. O grupo dividiu-se de forma a conseguir cumprir tudo aquilo a que se propôs fazer.

Feita uma estimativa do tempo despendido nas diferentes fases:

Fase	Período de trabalho	h/semanas	Total (horas)
Fundamentação	3 semanas	8h/semanais	24h
Especificação	5 semanas	12h/semanais	60h
Construção	4 semanas	16h/semanais	64h

Tabela 3 - Estimativa do tempo de trabalho.

7 Interface

Nesta seção serão apresentadas as páginas principais da *interface* do jogo que foi desenvolvido.

7.1 Página Inicial

Na página inicial são apresentados os *links* para as restantes páginas que compõem a página inicial. Página com o resumo do jogo, as regras, o vídeo promocional, um tutorial e a página para fazer login e registo.

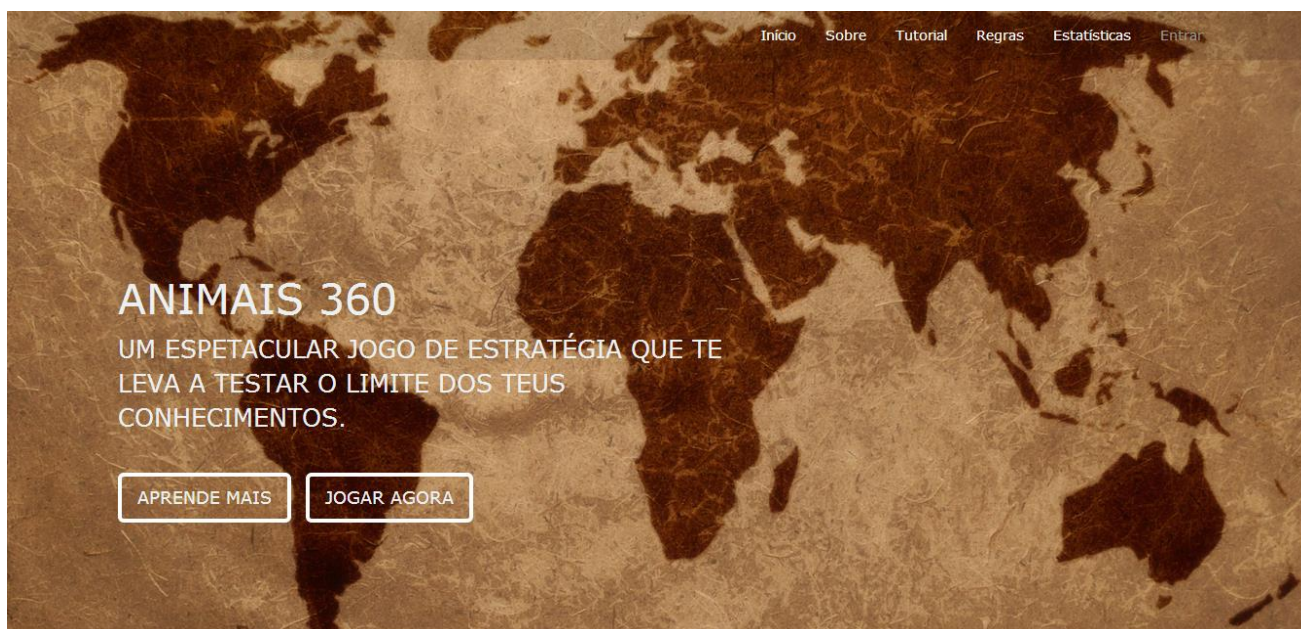


Figura 61 - Página Inicial do jogo.

7.2 Página Login e Registo

Nesta página é possível fazer o login dos jogadores e administradores no jogo. Também é possível fazer o registo de novos jogadores.

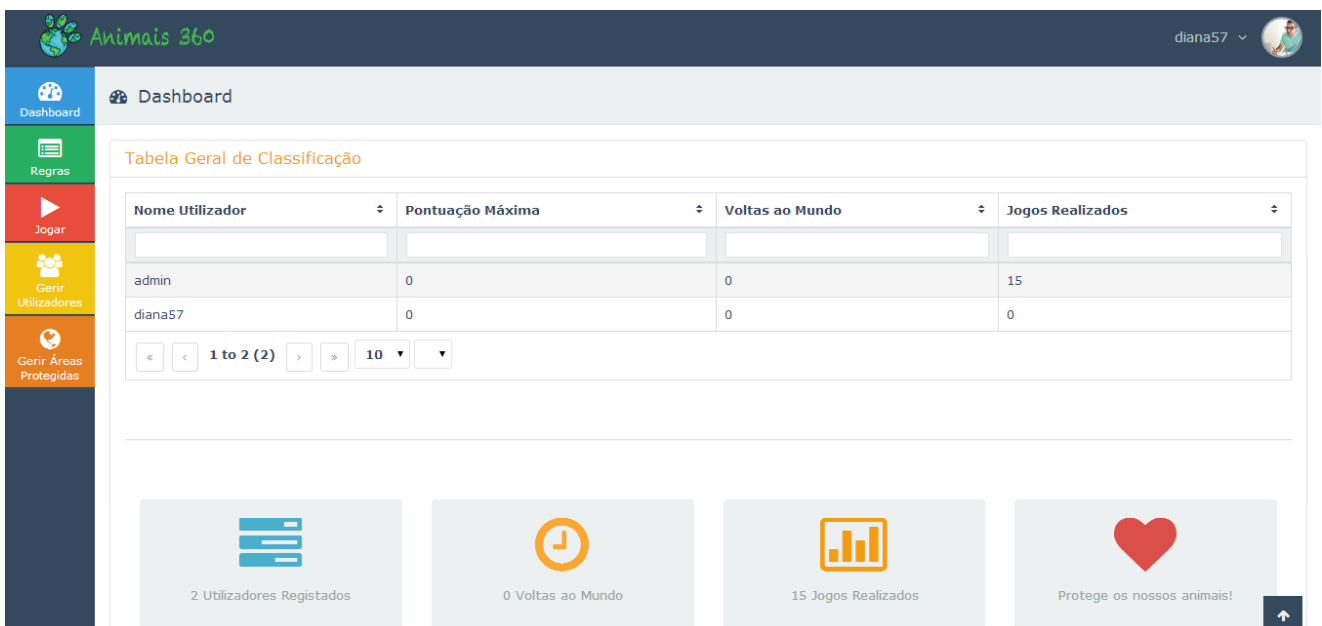


The screenshot shows the login and registration interface for 'Animais 360'. At the top, there is a dark blue header with the game's logo and name. Below this, the login form consists of two input fields for 'E-mail' and 'Password', followed by a checkbox labeled 'Manter sessão iniciada.' and a green 'Entrar' button. Below the button, there are two links: 'Esqueceste-te da password?' and 'Ainda não tens conta? [Cria uma conta.](#)'. At the bottom, a footer contains the text 'Animais 360©' and a link 'Regressar à Página Inicial'.

Figura 62 - Página Login e criar conta.

7.3 Dashboard

Após fazer o login, esta página é apresentada. Nesta, é apresentada a estatística geral do jogo, ou seja, a estatística referente a todos os jogadores.



The screenshot shows the dashboard interface for 'Animais 360'. The top header includes the game's logo, name, and a user profile section for 'diana57'. A left sidebar contains navigation links: 'Dashboard', 'Regras', 'Jogar', 'Gerir Utilizadores', and 'Gerir Áreas Protegidas'. The main content area features a 'Tabela Geral de Classificação' table with columns for 'Nome Utilizador', 'Pontuação Máxima', 'Voltas ao Mundo', and 'Jogos Realizados'. The table lists 'admin' and 'diana57' with their respective scores. Below the table is a pagination control showing '1 to 2 (2)' and a '10' dropdown. At the bottom, four summary cards display: '2 Utilizadores Registados', '0 Voltas ao Mundo', '15 Jogos Realizados', and 'Protege os nossos animais!'. A small 'up' arrow icon is in the bottom right corner.

Nome Utilizador	Pontuação Máxima	Voltas ao Mundo	Jogos Realizados
admin	0	0	15
diana57	0	0	0

Figura 63 - Página *Dashboard*.

7.4 Página de jogo

Quando o jogador pretende realizar uma partida, entra na página “Jogar”. Nesta página é apresentado o mapa do mundo, no qual o jogador escolhe o local onde começa a partida, a partir de uma lista inicial. Também aqui tem disponíveis as três ajudas, que pode utilizar sempre que tiver pontos suficientes. Para que o jogador se situe no mapa de jogo, cada ponto do mapa onde é possível responder a perguntas, está identificado.

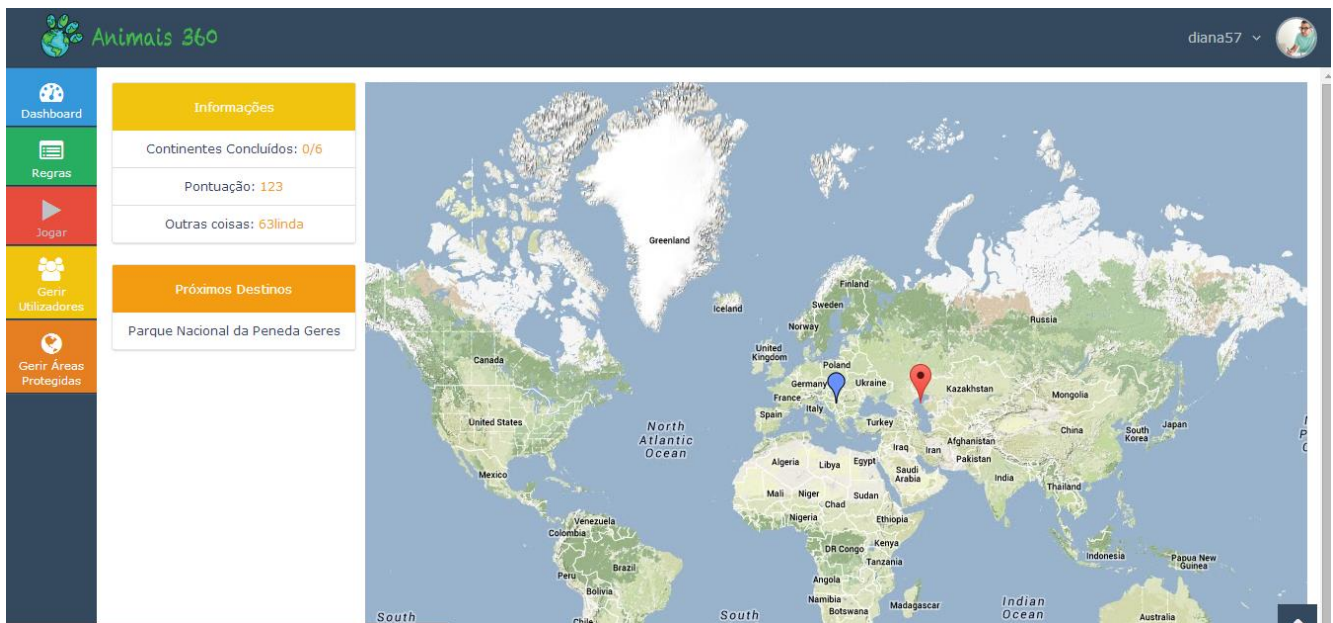


Figura 64 - Página de jogo.

7.5 Página de administração

O administrador gere toda a aplicação, dessa maneira existem certas páginas onde pode efetuar essas mudanças tal como, a página onde gere todos os utilizadores registados no jogo, uma página onde gere as áreas protegidas e uma outra onde pode gerir os desafios.

De seguida é apresentada a página de gerir utilizadores, as restantes estão em anexo.

The screenshot displays the 'Gerir Utilizadores' page. The top navigation bar includes the 'Animais 360' logo and the user 'diana57'. The left sidebar has buttons for 'Dashboard', 'Regras', 'Jogar', 'Gerir Utilizadores' (highlighted), and 'Gerir Áreas Protegidas'. The main content area is titled 'Gerir Utilizadores' and features a table with the following data:

Id	Nome de Utilizador	Tipo	
1	admin	1	Ver Perfil Editar Banir
2	diana57	0	Ver Perfil Editar Banir

Below the table, there are pagination controls showing '1 to 1 (2)' and a dropdown menu set to '10'.

Figura 65 - Página de gerir utilizadores.

8 Conclusões Finais

Este projeto distinguiu-se de todos os outros projetos que fizemos em semestres anteriores principalmente devido ao cuidado no cumprimento de prazos e no maior e mais regular supervisionamento por parte dos professores.

O que acontecia antes é que o projeto tinha apenas uma fase de avaliação que acontecia geralmente no final do semestre. A nossa falta de método e a sensação de que o prazo de entrega ainda se encontrava muito longe, era meio caminho andado para se deixar tudo para o final. Quando assim acontecia, os últimos dias eram penosos e parecia que nada corria como esperado, os erros multiplicavam-se e algumas das funcionalidades pedidas tinham de ser abandonadas pelo caminho. Com LI4 foi diferente: com avaliações mais regulares e a obrigatoriedade de planeamento, incutiu-nos a responsabilidade de sermos mais regrados e manteve-nos sempre a trabalhar num ritmo constante e saudável.

Chegada esta etapa final é necessário fazer uma reflexão de todo o processo de construção desta magnífica plataforma web. Aprendemos que quando é necessário fazer projetos desta envergadura é essencial planear todo o seu desenvolvimento para que no futuro não ocorram erros ocasionais por falta deste planeamento.

8.1 Secção do Gestor

Como gestor deste projeto é satisfatório chegar a este produto final apesar de alguns requisitos não terem sido totalmente concluídos. O nível de auto aprendizagem neste projeto foi elevado para se conseguir algumas das tarefas planeadas.

É também bom reconhecer todo o empenho e dedicação que os elementos do grupo tiveram para se conseguir chegar até este final. Sei que sem as mais valias individuais que cada um deles possui não seria possível obter este tipo de aplicação.

Lista de Acrónimos

ASP: *Active Server Pages*

EF: *Entity Framework*

CSS: *Cascading Style Sheets*

HTML: *HyperText Markup Language*

MVW: *Model-View-Whatever*

ORM: *Object-Relational Mapping*

SQL: *Structured Query Language*

UML: *Unified Modeling Language*

Referências WWW

[01] **www.nationalgeographic.com**

Página principal da sociedade *National Geographic*. Aqui podemos encontrar uma das maiores instituições científicas e educacionais sem fins lucrativos do mundo. Esta página engloba interesses como arqueologia, geologia e ciências naturais, bem como a promoção da conservação ambiental e histórica.

Referências Bibliográficas

- [01] CONNOLLY, Thomas; BEGG, Carolyn - **DataBase Systems, A Practical Approach to Design, Implementation and Management, 4/E.** Addison-Wesley, 2005. ISBN 9780321294012.

Este livro coloca uma forte ênfase nas boas práticas de projeto, permitindo aos alunos dominar a metodologia de *design* passo por passo. Trata-se de uma simples introdução a questões de implementação, sendo também um tratamento extenso de linguagens de base de dados.

Anexos

I. Análise das pontuações em casos concretos

FÁCIL

Inicial: 50

Ganho: 5

Ajudas:

1 - 6

2 - 8

3 - 10

MÉDIO

Inicial: 300

Ganho: 20

Ajudas:

1 - 40

2 - 50

3 - 60

DIFÍCIL

Inicial: 1000

Ganho: 200

Ajudas:

1 - 300

2 - 500

3 - 600

Exemplos a usar sempre a mesma ajuda, ou não, e GANHAR:

Legenda:

GS - ganhar sempre, sem ajudas;

G1 - ganhar, com a ajuda 1;

G2 - ganhar, com a ajuda 2;

G3 - ganhar, com a ajuda 3.

Questão	GS	G1	G2	G3
i	50	-	-	-
1	55	49	47	45
2	60	48	44	40
3	65	47	41	35
4	70	46	38	30
5	75	45	35	25
6	80	44	32	20
7	85	43	29	15
8	90	42	26	10
9	95	41	23	5
10	100	40	20	-----

(...)

Questão	GS	G1	G2	G3
i	300	-	-	-
1	320	280	270	260
2	340	260	240	220
3	360	240	210	180
4	380	220	180	140
5	400	200	150	100
6	420	180	120	60
7	440	160	90	20
8	460	140	60	-----

(...)

Questão	GS	G1	G2	G3
i	1000	-	-	-
1	1200	900	700	600
2	1400	800	400	200
3	1600	700	-----	-----
4	1800	600	-----	-----
5	2000	500	-----	-----

(...)

Exemplos a usar sempre a mesma ajuda, ou não, e PERDER (ou seja, menos o valor de ganho):

Legenda:

PS - perder sempre, sem ajudas;

P1 - perder, com a ajuda 1;

P2 - perder, com a ajuda 2;

P3 - perder, com a ajuda 3.

Questão	PS	P1	P2	P3
i	50	-	-	-
1	45	39	37	35
2	40	28	24	30
3	35	17	11	15
4	30	6	-2	0
5	25	-5	-----	-----
6	20	-----	-----	-----
7	15	-----	-----	-----
8	10	-----	-----	-----
9	5	-----	-----	-----
10	0	-----	-----	-----
11	-----	-----	-----	-----

(...)

Questão	PS	P1	P2	P3
i	300	-	-	-
1	280	240	230	220
2	260	180	160	140
3	240	120	90	60
4	220	60	20	-----
5	200	0	-----	-----
6	180	-----	-----	-----
7	160	-----	-----	-----
8	140	-----	-----	-----

(...)

Questão	PS	P1	P2	P3
i	1000	-	-	-
1	800	500	300	200
2	600	0	-----	-----
3	400	-----	-----	-----
4	200	-----	-----	-----
5	0	-----	-----	-----

(...)

Assumindo que haverá 30 questões:

Fácil:

- Melhor caso: 200 pontos

Médio:

- Melhor caso: 900 pontos

Difícil:

- Melhor caso: 7000 pontos

Para além de termos as estatísticas divididas por grau de dificuldade, é possível ter um tabela geral (com todos os utilizadores em todos os graus de dificuldade).

Neste sistema de pontuação, o melhor de grau fácil nunca estará à frente de alguém com uma prestação média no grau médio (já que este começaria nos 300). Porém, poderá estar à frente de alguém muito fraco do grau médio.

O mesmo se passa do médio para o difícil.

II. Outras Páginas da Interface

A página seguinte diz respeito a uma das páginas iniciais, mais propriamente a página que contém o resumo do jogo.



A página seguinte é a página que contém as regras do jogo.



A página apresentada de seguida diz respeito à página com todas as regras do jogo. Esta página é apresentada já quando os utilizadores ou administrados fizeram *login*.

The screenshot shows the 'Regras de Jogo' page in the 'Animais 360' application. The left sidebar contains navigation links: Dashboard, Regras, Jogar, Gerir Utilizadores, and Gerir Áreas Protegidas. The main content area is titled 'Regras de Jogo' and contains three sections:

- Requisitos Mínimos de Progressão**: Em cada continente, é obrigatório completar com sucesso sete desafios em áreas protegidas diferentes. Só depois poderás progredir na tua aventura.
- Tempo de Resposta**: Para responderes a um desafio, terás de o fazer dentro de um limite de tempo. O incumprimento desta regra leva a que a tua resposta seja considerada errada e como tal à consequente perda de pontos. No caso de utilizares ajuda, esta regra não se aplica.
- O Derradeiro Desafio**: No momento em que estás prestes a terminar a tua jornada num dado continente, é-te imposto um desafio especial: O Derradeiro Desafio! Neste momento, terás de responder corretamente a um conjunto de três questões relativas a um vídeo de alguns segundos sobre um animal específico, sem a possibilidade de usar qualquer tipo de ajuda. Se fores mal sucedido, terás de repetir o desafio. Após completares com sucesso este obstáculo, irás receber um crachá de conquista desse continente e estarás apto para avançar para o próximo continente.

A imagem seguinte refere-se à página de gestão das áreas protegidas.

The screenshot shows the 'Gerir Áreas Protegidas' page in the 'Animais 360' application. The left sidebar contains navigation links: Dashboard, Regras, Jogar, Gerir Utilizadores, and Gerir Áreas Protegidas. The main content area is titled 'Gerir Áreas Protegidas' and contains a table of protected areas.

At the top right of the table is a button: **+ Adicionar Área Protegida**.

The table has columns for Continentes (Europa, África, Ásia, América, Oceania, Antártida) and Desafios. The first section is for **Espanha** and the second for **Portugal**.

Continentes	Desafios	
Europa		
África		
Ásia		
América		
Oceania		
Antártida		
Espanha	Desafios	
Parque Nacional de Doñana	0	Ver Desafios Editar Bloquear
1 to 1 (1) 10		
Portugal	Desafios	
Parque Natural da Arrábida	0	Ver Desafios Editar Bloquear
Parque Natural da Ilha do Pico	0	Ver Desafios Editar Bloquear
1 to 1 (1) 10		

A seguinte imagem corresponde à página onde são geridos os desafios.

The screenshot displays the 'Animais 360' web application interface. At the top, a dark blue header bar contains the logo and name 'Animais 360' on the left, and the user profile 'diana57' with a dropdown arrow on the right. Below the header is a light gray sidebar with five navigation buttons: 'Dashboard' (blue), 'Regras' (green), 'Jogar' (red), 'Gerir Utilizadores' (yellow), and 'Gerir Áreas Protegidas' (orange). The main content area is titled 'Dashboard' and features a section for 'Gerir Desafios' (Manage Challenges). This section includes a '+ Adicionar Desafio' button and a table with the following columns: 'ID', 'Desafio', 'Dificuldade', and 'Tipo'. The table has a pagination bar at the bottom showing '10' items per page. A small dark blue button with an upward arrow is located in the bottom right corner of the page.