



**Universidade do Minho**

Escola de Engenharia

## **Unidade Curricular de Laboratórios de Informática IV**

Ano Lectivo de 2013/2014

### **A volta ao Mundo em 80 Dias**

**Grupo Nº 15**

**Filipa Inês Andrade Portugal Rocha Nº 62091**

**João António Fernandes Rodrigues Nº 64356**

**Pedro Emanuel Silva Ferreira Nº 66066**

**Carlos Manuel Fernandes Vieira Nº 66075**

5 de Maio de 2014

Data de Recepção	
Responsável	
Avaliação	
Observações	

## **A volta ao Mundo em 80 Dias**

**Grupo Nº 15**

**Filipa Inês Andrade Portugal Rocha Nº 62091**

**João António Fernandes Rodrigues Nº 64356**

**Pedro Emanuel Silva Ferreira Nº 66066**

**Carlos Manuel Fernandes Vieira Nº 66075**

5 de Maio de 2014

## Resumo

O presente documento serve de base às fases de fundamentação e especificação do projeto proposto no âmbito da unidade curricular *Laboratórios de Informática IV*. Tal projeto consiste na conceção de um jogo intitulado “Volta ao mundo em 80 questões”.

O jogo, que deverá ter como plano de fundo um planisfério, passará por responder a um conjunto de questões, cada uma destas associada a um determinado local do mundo. Escolhido um ponto de partida, o jogador dispõe de 80 dias para percorrer o maior número de locais possíveis, voltando obrigatoriamente ao ponto de partida. Cada ponto só será ultrapassado se o jogador responder acertadamente à uma pergunta.

É objetivo da unidade curricular “analisar e especificar de forma completa todos os requisitos operacionais e funcionais de um sistema de *software*” (*Laboratórios de Informática IV-Apresentação da Disciplina 2013/2014*). Pretende-se assim uma abordagem organizada e estruturada ao processo desenvolvimento de *software*. Havendo liberdade tanto na escolha do tema como na definição de normas e funcionamento do jogo, cabe ao grupo de trabalho detalhar as especificações do *software* a desenvolver.

Assim, o documento que constitui a base da documentação necessária ao produto final, apresenta agora, em forma textual e modelada, a análise de requisitos do sistema, detalhando ainda o esquema de base de dados.

**Área de Aplicação:** Engenharia de Software

**Palavras-Chave:** Engenharia, Software, etapa, fundamentação, especificação, planear, requisitos, dados de entrada, especificações, cronograma, agenda, jogo, volta, mundo, tempo, dinheiro, viagem, pontos de paragem, transporte, ajuda

# Índice

Resumo	i
Índice	ii
Índice de Figuras	v
Índice de Tabelas	vi
1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	2
1.3. Motivação e Objetivos	2
1.4. Estrutura do Relatório	4
2. Qualidade e Planificação	5
2.1. Equipa de projeto	6
2.2. Objetivos	6
2.3. Dados de Entrada/Especificações	7
2.4. Plano de atividades e Controlos de execução	7
3. Fundamentação	8
3.1. Objetivo de Jogo	8
3.2. Tema do Jogo	9
3.3. Regras e Funcionamento de jogo	9
3.3.1 Glossário:	9
3.3.2 Início do jogo	9
3.3.3 Pontos de paragem	10
3.3.4 Perguntas	11
3.3.5 Tempo	13
3.3.6 Viagens e meios de transporte	13
3.3.7 Dinheiro e trabalhos	14
3.3.8 Ajudas	14
3.3.9 O que fazer nos Pontos de paragem	15
3.3.10 Voltar atrás	15
3.3.11 Ranking	16
3.4. Fluxo de Jogo	16
3.5. Público-alvo	17
4. Requisitos	20

4.1. Requisitos Funcionais	20
4.2. Requisitos não funcionais	23
5. Questões relativas ao funcionamento do sistema	25
5.1.1 Grafo de pontos de paragem	25
5.1.2 Passagem do Tempo, dinheiro e meios de transporte	25
5.1.3 Mecanismos de aprendizagem do sistema	26
5.1.4 Mecanismo probabilístico de lançamento de perguntas	27
6. Modelação	28
6.1. Modelo de Domínio	29
6.2. Modelo de <i>Use Cases</i>	31
6.3. Identificação dos subsistemas e Diagramas de Classe	34
6.4 Diagramas de sequência	38
7. Modelo de Base de dados	41
7.1. Modelo Conceptual	43
7.2. Modelo Lógico	44
8. Conclusões e Trabalho Futuro	45
Bibliografia	47
Referências WWW	48
Lista de Siglas e Acrónimos	49
Anexos	50
I. Dados de Entrada/Especificações	51
II. Plano de Atividades	52
III. Modelo de Use Cases	53
IV. Diagramas de Sequencia	69
1. Start Game	69
2. Comprar Ajuda	70
3. Vender Ajuda	71
4. Executar Trabalho	71
5. Viajar	72
6. Entrar na cidade	73
6.1 Carregar Nova Pergunta	74
7. Passagem do Tempo	74
V. Dicionário de Dados	75
1. Tabela Schedule	76
2. Tabela LifeLine	77
3. Tabela GameLifeLine	78
4. Tabela CityLifeLine	79
5. Tabela WorldRegion	80
6. Tabela City	81
7. Tabela Route	82

8.	Tabela ModeOfTravel	83
9.	Tabela Labour	84
10.	Tabela CityLabour	85
11.	Tabela GameLabour	86
12.	Tabela Trip	87
13.	Tabela Game	88
14.	Tabela Question	89
15.	Tabela QuestionCategory	90
16.	Tabela AnswerChoice	91
17.	Tabela GameAnswer	92
18.	Tabela User	93

# Índice de Figuras

<b>Figura 1</b> - Equipa do projeto.	6
<b>Figura 2</b> - Ciclo PDCA	6
<b>Figura 3</b> – Mapa <i>mundi</i> com os pontos de paragem.	10
<b>Figura 4</b> - Pergunta exemplo.	11
<b>Figura 5</b> - Área de abrangência dos pontos de paragem	12
<b>Figura 6</b> - Dependência associada géneros de jogos	18
<b>Figura 7</b> - Tipos de jogos por sexo e idade	18
<b>Figura 8</b> - Gráfico representativo da distribuição normal $X \sim N(6,1)$ .	27
<b>Figura 9</b> - Modelo de domínio.	30
<b>Figura 10</b> - Use Case principal da aplicação.	31
<b>Figura 11</b> - <i>Use Cases</i> relativos ao jogo.	31
<b>Figura 12</b> - Descrição tabular do <i>Use Case Lançar Pergunta</i> .	32
<b>Figura 13</b> - Descrição textual do <i>Use Case Atualizar Pergunta</i> .	32
<b>Figura 14</b> - Descrição textual do <i>Use Case Entrar na Cidade</i> .	33
<b>Figura 15</b> - Subsistema Jogos.	35
<b>Figura 16</b> - Subsistema Perguntas.	36
<b>Figura 17</b> - Subsistema Viagens.	37
<b>Figura 18</b> - Diagrama de Sequência da Passagem de tempo.	38
<b>Figura 19</b> - Diagrama de Sequência de Entrar na Cidade.	39
<b>Figura 20</b> - Modelo conceptual da base de dados.	43
<b>Figura 21</b> - Modelo lógico da base de dados.	44

## Índice de Tabelas

<b>Tabela 1</b> - Esquema de viagens disponível em Paris.	13
<b>Tabela 2</b> - Esquema resumo das funcionalidades obrigatórias.	22
<b>Tabela 3</b> - Tabela com as tecnologias utilizadas.	24



# 1. Introdução

## 1.1. Contextualização

O trabalho que o presente relatório documenta surge no último semestre da Licenciatura em Engenharia Informática, consistindo na conceção de um jogo intitulado “Volta ao mundo em 80 questões”.

Aprendidas diferentes linguagens de programação, noções algorítmica, de estruturas e armazenamento de dados e os primeiros conceitos de desenvolvimento de *software*, na última unidade curricular de laboratórios de informática é pedido que se dê o salto de programador para engenheiro de *software*.

Neste contexto, o trabalho não se resume à elaboração de um jogo. Pretende-se um primeiro e último ensaio daquilo que será o desenvolvimento de um *software*. Assumindo o grupo o papel de equipa de projeto e os docentes o papel de cliente, todo um processo de planeamento está inerente ao desenvolvimento do trabalho enquanto um produto.

O trabalho tem início numa fase de fundamentação, seguida de uma avaliação por parte do cliente. Note-se que a fundamentação do sistema pede um estudo prévio do problema, a definição dos objetivos, dos dados entradas e das especificações, pede a elaboração de um cronograma onde estarão agendadas cada uma das especificações (um plano de atividades), em suma, um carece de planeamento. Concluído este processo, e cumprindo o plano de atividades deverá sim passar-se para a fundamentação propriamente dita, definindo textualmente e tão detalhado quanto possível o funcionamento do *software*.

Sendo a fundamentação aprovada pelo cliente, passar-se-á à fase de especificação, onde os requisitos do sistema serão traduzidos em UML.

Uma nova avaliação/aprovação do cliente antecederá a fase final de implementação.

## 1.2. Apresentação do Caso de Estudo

O projeto proposto consiste na elaboração de um jogo intitulado “Volta ao mundo em 80 questões”. Havendo um conjunto de requisitos impostos no enunciado, a definição de um modelo de jogo que cumpra esses requisitos é tarefa do grupo de trabalho.

A presente subsecção apresenta uma breve introdução ao jogo a desenvolver.

O jogo decorre sobre o mapa *mundi*. É este o nosso tabuleiro. Em cada jogo deverá ser escolhido um ponto de partida e um nível de dificuldade. O jogador dispõe de 80 dias para, percorrendo o maior número de pontos possível, dar a volta ao mundo.

Por “dar a volta ao mundo” entenda-se, percorrer um conjunto de pontos obrigatórios e voltar ao ponto de partida. Em cada ponto, seja ele obrigatório ou não, será lançada uma pergunta. A resposta certa a essa pergunta dá crédito ao jogador e permite-lhe entrar na cidade correspondente. A resposta errada impede o jogador de prosseguir. Com dinheiro disponível um jogador poderá comprar viagens e assim prosseguir para o próximo ponto. O preço e a duração da viagem dependem do meio de transporte escolhido pelo jogador. Vendo-se sem dinheiro, o jogador poderá despende de tempo e trabalhar. Postos de compra e venda de ajudas em determinados pontos estarão também à disposição do jogador.

Dada a natureza do jogo, considera-se mais apropriado denominar o jogo efetivamente como “Volta ao mundo em 80 dias”.

## 1.3. Motivação e Objetivos

A definição do modelo de jogo foi a tarefa principal da etapa de fundamentação.

Está definida no enunciado do projeto uma linha comum que todos os grupos devem seguir.

O jogo deveria consistir num conjunto de perguntas associadas a pontos (georreferenciados) de um *mapa-mundi*, que seria o tabuleiro de jogo. O jogador escolheria um nível de dificuldade e um ponto de partida e a partir desse ponto poderia iniciar a sua volta ao mundo, respondendo a uma série de perguntas. A volta terminava quando o jogador regressasse ao ponto de partida. Os restantes pré-requisitos expõem-se na secção 3.6.

Tomando como base a linha apresentada, pretendia-se que o jogo a desenvolver fosse mais do que um simples *quiz*. Assim, ao acrescentar à problemática das rotas a gestão simultânea de tempo e dinheiro, procurou-se um modelo de jogo atrativo que para além da componente cultural inicialmente exigida conferisse ao jogo a componente de estratégia.

Neste sentido, adotaram-se os objetivos:

- Desenvolver um jogo atrativo para o jogador.
- Desenvolver um jogo que permita ao jogador testar e melhorar a sua cultura geral proporcionando-lhe simultaneamente o divertimento de um jogo de estratégia.

Este projeto permite simular uma situação real cliente/fornecedor virtualizando um contexto complexo em que pode partir para o desenvolvimento com total transparência e sintonia com as especificações e necessidades do cliente. Pretende-se adquirir a capacidade de conceber uma aplicação informática com ênfase na clareza da fundamentação e das especificações permitindo realizar um trabalho produtivo. Surge assim o objetivo:

- Cumprir todas as especificações do cliente.

A aplicação de normas de gestão de projetos ao processo de desenvolvimento de *software* é uma das finalidades do trabalho proposto.

O desenvolvimento de um *software* carece de uma série de etapas que antecedem a sua implementação propriamente dita. É necessário projetar e planejar, verificar e validar todo e qualquer passo da sua conceção, procurando garantir que estão a ser cumpridos os objetivos definidos.

A cada passo do projeto considera-se necessário verificar se estas competências estão a ser adquiridas. Só assim se conseguirá atingir o resultado pretendido. Neste sentido, a par dos objetivos anteriormente definidos, a equipa de projeto adotou os objetivos presentes no enunciado do trabalho prático:

- Dar rotina no domínio da Engenharia de Software com a aprendizagem dos conceitos e terminologia base da área.
- Apreender os métodos de trabalho mais relevantes no processo de desenvolvimento de *software*.
- Reforçar o “espírito” de trabalho de grupo e apresentar alguns conceitos sobre gestão e acompanhamento de projetos.
- Completar a formação em processos de pesquisa e escrita de relatórios e apresentação pública de trabalhos.
- Complementar a formação em ambientes e processos de desenvolvimento de *software*.
- Relevar a importância de um bom acompanhamento e gestão de um projeto de desenvolvimento de *software*.

(Belo, O., LEI, 3ºAno, Disciplina de Laboratórios de Informática IV)

## **1.4. Estrutura do Relatório**

Introduzido o trabalho, apresenta-se uma breve descrição do que se irá encontrar neste documento.

O corpo do relatório começa com a secção “Qualidade e planificação”, que justifica a opção estruturar um planeamento e apresenta a metodologia adotada para a planificação do projeto.

Na secção “Fundamentação”, expõe-se a filosofia de jogo e as particularidades idealizadas, contendo o esboço quase final daquilo que será o manual de utilização

As restantes secções são o resultado da fase de especificação. Definido o modelo de jogo, procedeu-se à conclusão do processo de levantamento de requisitos que se apresentam na em linguagem textual e traduzidos para uma linguagem gráfica de modelação (UML). Apresenta-se ainda a forma como se pretende responder a requisitos impostos no enunciado.

Às secções de Requisitos e Modelação segue-se a descrição do modelo de bases de dados onde, após uma explicação e justificação textual do que se pretende implementar, se expõem os esquemas conceptual e lógico da base de dados.

Explica-se ainda com algum detalhe algumas das funcionalidades que se irão implementar.

O relatório termina com a conclusão, onde se procede a uma reflexão sobre o trabalho realizado e se antecipa as tarefas que ainda estão por vir.

## 2. Qualidade e Planificação

Numa qualquer conceção de natureza minimamente complexa, quanto mais tempo investirmos na planificação menos gastaremos na realização das tarefas, ao mesmo tempo que melhoramos a qualidade do produto final, ou seja melhoraremos a produtividade. Estamos perante uma máxima da engenharia: o máximo de proveito com o mínimo de custo.

É como uma casa que deve ser projetada e planeada evitando começar de imediato a empilhar tijolos. Só assim se pode evitar as paragens por indecisão, o constante fazer e desfazer.

Com o intuito de melhor perceber o papel do planeamento e da qualidade em projetos consultou-se a norma NP ISO 10006. Importa referir que a metodologia proposta na norma foi apenas orientadora, não tendo sido integralmente seguida pela equipa do projeto. Da referida norma destaca-se a seguinte definição de projeto:

“Processo único que consiste num conjunto de atividades (3.1) coordenadas e controladas, com datas de início e de fim, realizadas para atingir um objetivo em conformidade com requisitos específicos, incluindo limitações de tempo, custos e recursos.”

Assim, um projeto é uma função que transforma dados de entrada em saídas procurando de forma estruturada atingir um objetivo. O fator qualidade que refere a norma consiste na sistemática verificação e validação da conformidade dos resultados com as especificações e objetivos.

Para além do estudo da norma, analisou-se ainda o projeto de Lançamento de um *software* de uma empresa com sistema de gestão da qualidade certificado pela NP EN ISO 9001:2006. Da estrutura do projeto constavam as seguintes secções:

- A. Equipa do projeto.
- B. Especificações: Documentos relativos aos objetivos do projeto, aos dados de entrada e respetivas especificações.
- C. Planeamento: Plano de atividades, Plano de controlo e Controlos de execução.

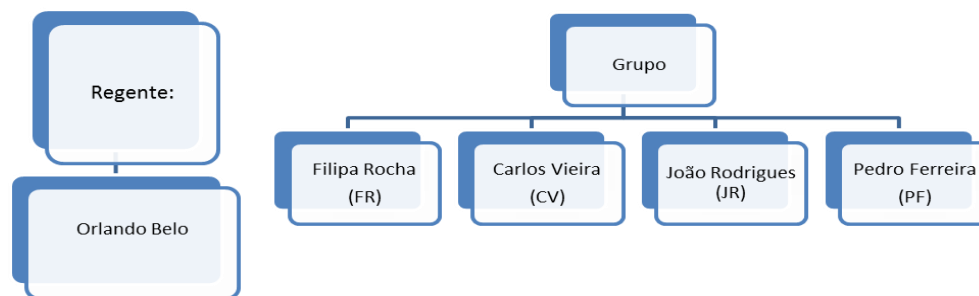
Considerando necessário um grau de experiência maior em elaboração e gestão de projetos, optou-se por não elaborar plano de controlo. Os restantes componentes da planificação do projeto foram adotados e apresentam-se nas subsecções seguintes.

## 2.1. Equipa de projeto

O projeto proposto permite a simulação de um ambiente comercial. O grupo de trabalho assume o papel da equipa responsável pelo desenvolvimento do projeto desde o seu planeamento e conceção até à implementação propriamente dita.

Por seu turno, a entidade cliente é representada pelos professores. Existem contudo uma série de requisitos técnicos que não são em geral impostos por um cliente. Mais, existe por parte dos professores a disponibilidade de acompanhar o projeto.

Nesse sentido, considera-se o professor não só como cliente mas também como um regente que impõe limites e monitoriza o estado projeto. Daí ser incluído na equipa do projeto.



**Figura 1** - Equipa do projeto.

## 2.2. Objetivos

Seguindo o *template* imposto, os objetivos encontram-se definidos na secção 1.3.

Importa só realçar que foram adotados os objetivos propostos no enunciado do trabalho prático. Isto porque a qualidade assenta no ciclo PDCA, *Plan, do, check, actue*. Às tarefas de planear e fazer, que por si só não garantem o cumprimento do planeamento, adiciona-se as grandezas de verificar e atuar, numa constante procura de corresponder aos objetivos do projeto. Assim, para que os objetivos referentes à qualidade e funcionamento do jogo sejam cumpridos considera-se que se deve cuidar que a obtenção as competências patentes nos objetivos definidos no enunciado estão a ser adquiridas.



**Figura 2** - Ciclo PDCA

## **2.3. Dados de Entrada/Especificações**

Conhecida a equipa e definidos os objetivos (ver 1.3) importa definir as especificações em que o projeto se traduz e as respetivas entradas de dados.

No Anexo 1 apresenta-se o documento que relaciona as entradas de dados com as especificações definidas.

## **2.4. Plano de atividades e Controlos de execução**

A representação em forma de cronograma das atividades definidas como especificações do projeto, apresenta-se no Anexo 2.

No plano de atividades encontram-se agendados controlos de execução, momentos em que se percorre o plano e se verifica se as atividades até á data agendadas foram ou não cumpridas. Nestes momentos procedem-se a algumas ponderações e caso as atividades planeadas se apresentem desadequadas ou se verifiquem atrasos, deve-se analisar as causas, definir ações e por fim atualizar o plano.

### **3. Fundamentação**

Planeado o projeto foi hora de idealizar e conceber o modelo de jogo, objetivo desta fase de fundamentação. Neste sentido definiram-se as especificações:

- Caracterização geral da aplicação a desenvolver, definindo de forma concisa os objetivos do jogo.
- Definição das funcionalidades a implementar, que deverão ser listadas de forma detalhada, indo de encontro aos requisitos impostos no enunciado.
- Definição do modelo de funcionamento do jogo/Elaboração da análise textual dos requisitos.

Depois de definidas as funcionalidades que se pretende implementar, pretende-se detalhar as regras de jogo, a forma como o jogador poderá interagir com a aplicação. Deverá ter como output o manual de instruções do jogo.

- Definição do tema. Prevendo-se elaborado o desenho da aplicação para que esta responda de forma independente do tema, deverá ainda na fase de fundamentação ser escolhido o tema.

As subsecções seguintes são output das especificações apresentadas.

#### **3.1. Objetivo de Jogo**

O objetivo do jogo é dar a volta ao mundo em 80 dias, percorrendo o maior número possível de pontos de paragem. O jogador completa o jogo se, passando por um conjunto de pontos obrigatórios, regressa ao ponto de partida. Se passados 80 dias a viagem não tiver terminado, o jogador perde o jogo.



## 3.2. Tema do Jogo

Avançar no jogo só é possível se o jogador responder acertadamente a uma pergunta. Essa pergunta será sobre **História e Geografia**.

## 3.3. Regras e Funcionamento de jogo

A elaboração do modelo de funcionamento do jogo exigiu um processo de reflexão e ponderação para cada uma das funcionalidades que se pretende implementar e para todas as outras ideias que surgiram e não se adotaram. Foi necessário definir qual deveria ser o objetivo e a filosofia de jogo. Aqui, a escolha final não foi a ideia original.

Definido aquele que seria o fio condutor do jogo, foram surgindo ideias para possíveis funcionalidades que o jogo incorporaria. Cada ideia foi analisada. Muitas foram rejeitadas por não se considerarem viáveis dado o tempo disponível, outras encontram-se descritas num documento e só serão implementadas se numa fase mais avançada se considerarem viáveis. Restam aquelas que se selecionaram e que se apresentam na presente secção.

Todo este capítulo constitui o esboço do que será o manual de instruções do jogo.

### 3.3.1 Glossário:

**Pontos de paragem:** Cidade por onde o jogador pode passar, sendo-lhe lançada pelo menos uma pergunta.

**Chave:** Resposta certa a uma pergunta lançada num ponto de paragem, permitindo a entrada do jogador.

**Vizinhos:** Seja P um ponto de paragem. Os vizinhos de P são os pontos para onde o jogador pode, a partir de P, transitar diretamente.

**Viagem:** Transição entre dois pontos de paragem vizinhos.

**Ponto Visitado:** Ponto de paragem no qual já se respondeu acertadamente a uma pergunta. Permite aceder a qualquer ponto vizinho.

**Ponto não visitado:** Ponto de paragem no qual ainda não se respondeu acertadamente a qualquer pergunta. Permite aceder apenas pontos vizinhos já visitados.

### 3.3.2 Início do jogo

O primeiro passo é o registo/autenticação do jogador. Caso o jogador já tenha conta de utilizador deve autenticar-se no sistema indicando nome de utilizador e *password*. Caso contrário deverá criar uma conta de utilizador e posteriormente proceder à autenticação.

Um jogador autenticado começa por indicar o nível de dificuldade em que pretende jogar (fácil, médio ou difícil).

Escolhido o nível de dificuldade, é apresentado um mapa *mundi* que será o tabuleiro de jogo. Os pontos de paragem aparecerão destacados e o jogador deve escolher aquele que será o ponto de partida.

Para iniciar o jogo só falta indicar qual a hora e o dia da semana em que pretende partir. Feito isto, o jogador entra na primeira cidade.

### 3.3.3 Pontos de paragem

Para dar a volta ao mundo o jogador tem obrigatoriamente que passar por um conjunto de cidades (pontos de paragem) e voltar ao ponto de partida. Os pontos obrigatórios são aqueles que Mr. Phileas Fogg percorreu no original *A Volta ao Mundo em 80 dias*<sup>1</sup>. São eles: São Francisco, Nova Iorque, Londres, Suez, Mumbai, Calcutá, Hong Kong e Yokohama.

Percorrendo estes pontos e voltando ao ponto de partida, o jogador dá a volta ao mundo. Mas lembra-se que o objetivo do jogo é percorrer o máximo de locais possíveis nos 80 dias disponíveis!

Além dos 8 pontos obrigatórios o jogador pode parar em Lisboa, Viena, Sicília, Estocolmo e Reykjavik, Kiev, Rabat, Mogadishu, São Paulo, Cidade do Cabo, entre outros.



**Figura 3** – Mapa *mundi* com os pontos de paragem. A vermelho os pontos obrigatórios, a púrpura os pontos opcionais<sup>2</sup>.

<sup>1</sup> Mais informações em: [http://en.wikipedia.org/wiki/Around\\_the\\_World\\_in\\_Eighty\\_Days](http://en.wikipedia.org/wiki/Around_the_World_in_Eighty_Days)

<sup>2</sup> Fonte do mapa: [http://upload.wikimedia.org/wikipedia/commons/a/a3/Map\\_of\\_the\\_world\\_1998.jpg](http://upload.wikimedia.org/wikipedia/commons/a/a3/Map_of_the_world_1998.jpg)

Nota: Os pontos de paragem, perguntas, ajudas e serviços disponíveis dependem exclusivamente da informação que será colocada na base de dados dado que estrutural e funcionalmente a aplicação não estará limitada a estes itens.

### 3.3.4 Perguntas

Ao tentar entrar numa cidade o jogador é sujeito a um interrogatório (uma pergunta). Respondendo acertadamente, o jogador demonstra ao responsável pela segurança da cidade que tem o conhecimento suficiente para aceder à cidade. Caso dê a resposta errada, o jogador é retido por um tempo determinado, após o qual se pode submeter a uma nova sessão de interrogatório ou pedir que seja libertado.

Cada pergunta é lançada com quatro hipóteses de resposta, sendo que apenas uma está correta. A resposta certa à pergunta passa, obviamente, pela escolha da opção correta.

As perguntas dizem respeito a uma determinada área não a um local em particular.

A figura 4 apresenta uma pergunta exemplo, que poderia ser lançada em qualquer um dos pontos de paragem localizados na América do Sul. A figura 5 apresenta a área de abrangência (lado esquerdo) de cada um dos pontos de paragem (lado direito).

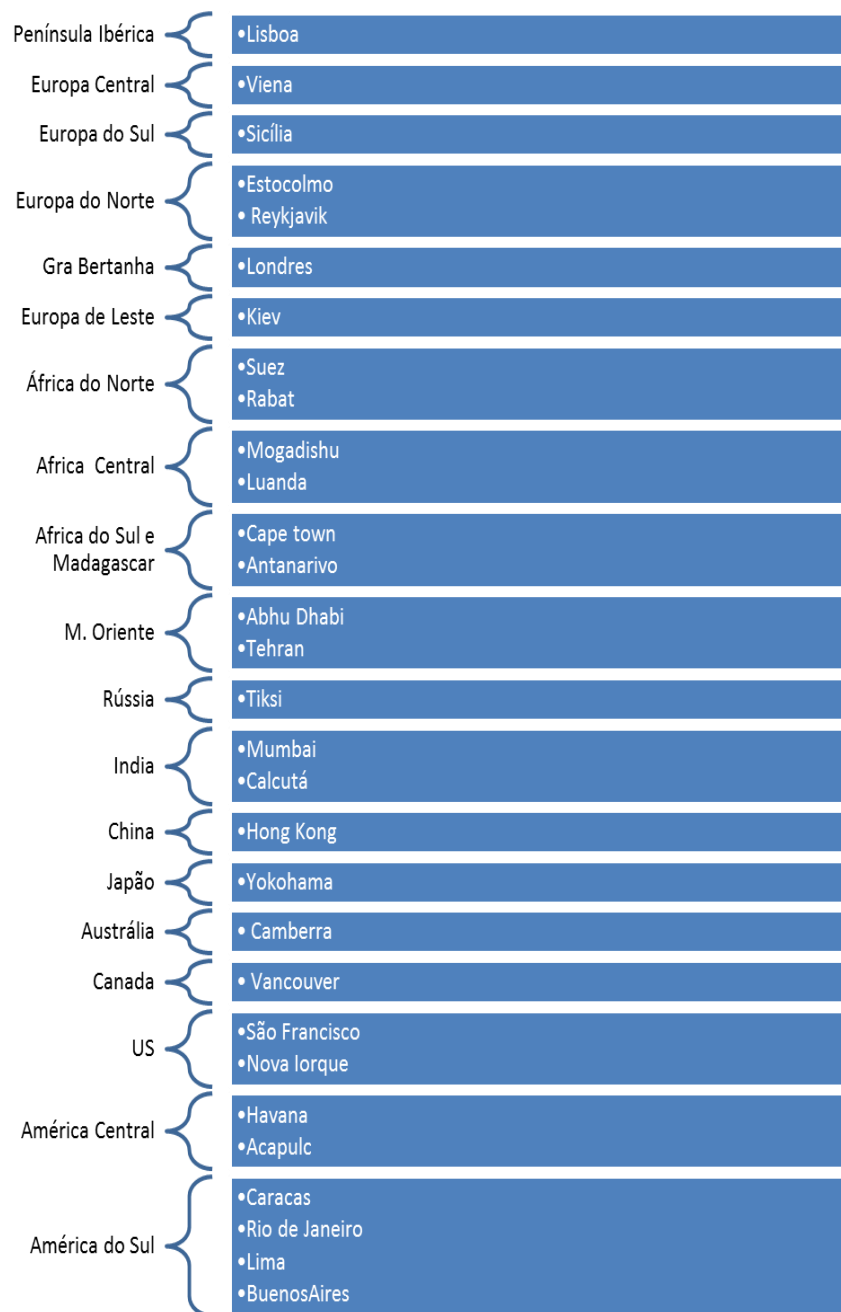
Assim, um jogador que pare em Lisboa pode responder a uma pergunta sobre qualquer ponto da Península Ibérica enquanto um jogador em São Francisco ou em Nova Iorque, responderá a uma pergunta sobre os Estados Unidos.

Respondendo acertadamente a uma pergunta, o ponto de paragem é marcado como visitado.

Qual o país administrador das ilhas da Páscoa?		[Tempo]
A: Equador	B: Chile	
C: Colômbia	D: Venezuela	

**Figura 4** - Pergunta exemplo com o tempo disponível listado no canto superior direito. Neste caso a resposta correta é a alínea B (Chile) desde 1888<sup>3</sup>.

<sup>3</sup> Pode-se ver aqui: <http://www.wmf.org/project/easter-island-rapa-nui-moai-conservation-project>



**Figura 5** - Área de abrangência dos pontos de paragem

### 3.3.5 Tempo

O tempo é uma grandeza de importância fundamental neste jogo. O jogador dispõe de 80 dias para dar a volta ao mundo.

No canto superior direito do ecrã um relógio indica a hora local e o tempo que o jogador ainda tem disponível.

Toda e qualquer ação despende tempo. Quando o jogador está parado a escolher qual o próximo ponto e qual o meio de transporte que vai utilizar, o tempo passa em tempo real. Enquanto o jogador responde, o tempo está sujeito a uma aceleração.

Também durante uma viagem o tempo passa mais depressa, sendo neste caso a duração dependente do meio de transporte.

### 3.3.6 Viagens e meios de transporte

Para se deslocar entre dois quaisquer pontos de passagens, um jogador tem que viajar.

Uma viagem consome tempo e dinheiro, consoante o meio de transporte que o jogador seleciona.

Os meios de transporte disponíveis dependendo do local de origem e do local de destino. Uma viagem pode ser feita de comboio, de autocarro, de avião, de Ferry, de carro ou de bicicleta.

Uma viagem de transporte público implica a compra de um bilhete e está sujeita a um horário. Assim, um jogador pode-se ver forçado a despende de tempo para esperar por um transporte público.

Em cada ponto de paragem, o jogador poderá ver os meios de transporte disponíveis para determinado destino e respetivos preços e horários.

Dias da Semana	Hora	Valor	Duração	Origem	Destino	Meio Transporte	
Seg-sex	11:00	120€	2:30m	Paris	Londres	TGV	
Seg-Sab	18:00	210€	1:00m	Paris	Londres	Air France	
Seg-Seg	9:00	50€	5h	Paris	Londres	BUS	
Seg-Seg	23:00	30€	8h	Paris	Londres	Barco e Autocarro	
Seg-Seg	9:00 – 18:00	600€	4h	Paris	Londres	Carro	
Seg	9:00	-20€ (ganho)				Paquete Queen Mary	Ajuda nos serviços de quarto.

**Tabela 1** - Esquema de viagens disponível em Paris.

Os destinos acessíveis a partir de um determinado ponto dependem do facto de este já ter sido ou não visitado.

Nalguns locais estão também disponíveis serviços de aluguer de viaturas, incluindo barcos, motos, carros, etc. Um determinado serviço de aluguer tem um custo associado à duração e serve apenas para um determinado trajeto, não sendo reutilizável.

### 3.3.7 Dinheiro e trabalhos

Para avançar no jogo, o jogador tem que viajar e para o fazer precisa de dinheiro.

A maneira imediata de ganhar dinheiro é acertar às perguntas. Num valor que depende do grau de dificuldade da pergunta, uma resposta correta dá dinheiro ao jogador.

Ainda assim, um jogador que não tem dinheiro para viajar pode optar por trabalhar. Mas lembre-se, tempo é dinheiro! Optando por trabalhar o jogador gasta tempo. Cada trabalho tem associado um valor euros/hora que define o dinheiro que o jogador poderá ganhar.

Tal como os meios de transporte, também os trabalhos dependem do ponto de paragem.

**Trabalhos:** Lavagem de carros (*rentings*), descarregar mercadoria (comboios, barcos e ferries), carregar malas, vender jornais, pedinchar.

### 3.3.8 Ajudas

O jogador tem a possibilidade de recorrer a ajudas para responder às perguntas, mas para isso terá que as comprar.

Nalguns pontos predefinidos existem ajudas que serão oferecidas ao “entrar na cidade.

Determinados pontos de paragem funcionam também como pontos de compra e venda de ajudas.

As ajudas disponíveis:

1. Eliminação de alternativas de resposta (1 ou 2). O número de alternativas eliminadas é o que determina o preço da ajuda.
2. Troca de pergunta: Custo baixo, permite responder logo a outra pergunta sem perder mais tempo.
3. Ajuda dos Jogadores: Em função das respostas dos outros jogadores à mesma pergunta, é apresentado um gráfico de tendências.
4. Salvo-conduto: Permite ao jogador entrar numa cidade sem responder a qualquer pergunta.

### **3.3.9 O que fazer nos Pontos de paragem**

Entrando num ponto de paragem um jogador pode:

1. Viajar (3.3.5);
2. Trabalhar (3.3.6);
3. Dependendo do ponto, comprar e vender ajudas. (3.3.7).

Clicando sobre qualquer um dos pontos de paragem, o jogador pode ver quais os trabalhos disponíveis, se é ponto de compra e venda e ainda os meios de transporte disponíveis com o respetivo horário.

Os serviços de Compra de ajudas, Compra de bilhetes e acesso à realização de serviços estão disponíveis num local fora do perímetro da cidade. Exemplos:

1. Um jogador pode aceder a uma loja de compra de ajudas disponível em Lisboa sem entrar na cidade (isto permite que o jogador compre uma ajuda no local para responder a uma pergunta no local).
2. Pode aceder aos serviços de transporte sem entrar no local – permite voltar para trás sem entrar na cidade.
3. Pode aceder a realização de serviços – permite amealhar dinheiro para comprar uma ajuda para entrar na cidade ou para comprar um bilhete para voltar para trás.

### **3.3.10 Voltar atrás**

Um jogador ao viajar para determinado ponto, não tem acesso imediato a esse ponto. Relembra-se que um local só é efetivamente visitado quando o jogador responde acertadamente a uma pergunta.

Se um jogador não consegue entrar na cidade pretendida não pode seguir para nenhum ponto por visitar. Enquanto o jogador está “à porta da cidade”, o tempo está a passar.

É contudo permitido ao jogador regressar aos pontos vizinhos já visitados. Desses pontos o jogador já tem a chave e portanto consegue aceder a qualquer ponto vizinho não visitado.

É de notar que voltar atrás não deixa de ser uma viagem e como tal consome tempo e dinheiro.

### 3.3.11 Ranking

Apesar do jogo ser *singleplayer*, existe um ranking de jogadores para cada nível de dificuldade. A posição do jogador no ranking depende primeiramente do número de pontos paragem visitados. Só em caso de empate se aplicam, ordenadamente, os seguintes critérios:

1. Duração da viagem.
2. Crédito restante.

O objetivo é conhecer o maior número de locais possível. Não adianta ser o mais rápido se não for o mais viajado.

## 3.4. Fluxo de Jogo

Recorde-se os estados possíveis dos pontos de paragem:

- Visitado: Se já se respondeu acertadamente a uma pergunta. Permite aceder a qualquer ponto vizinho
- Não visitado: Se ainda não se respondeu acertadamente a uma pergunta. Permite aceder apenas pontos vizinhos já visitados.

1. Início de Jogo
2. Escolhe Nível→Escolhe Tema
3. Apresenta Mapa
4. Escolhe Local 1º de Inicio [e Dia da Semana e Hora]
5. Arranca do local atual (ponto de inicio)
6. São apresentadas as Alternativas
  - 6.1. Compra e Venda de Ajudas
  - 6.2. Execução de Trabalhos
  - 6.3. Viajar
    - 6.3.1. Apresentados destinos possíveis (inclusive já visitados [mas sinalizados]).
      - 6.3.1.1. Os destinos possíveis de um local são:
        - 6.3.1.1.1. Todos os pontos com ligação ao local, caso o local já tenha sido “visitado” **ou o seja o ponto de partida.**
        - 6.3.1.1.2. Apenas os que já tenham sido visitados caso o local atual não esteja ainda visitado.
      - 6.3.2. É selecionado o Destino
      - 6.3.3. São apresentadas as Alternativas/Horários dos serviços de transporte
        - 6.3.3.1. Pode voltar atrás para selecionar outro destino (6.3) ou mesmo selecionar outras alternativas (6)
      - 6.3.4. Escolhido o serviço de transporte
        - 6.3.4.1. Compra bilhete
        - 6.3.4.2. Aguardar pelo próximo serviço de transporte



- 6.3.4.2.1. Pode fazer *skip* (passar o tempo rápido)
- 6.3.5. Em viagem
  - 6.3.5.1. Aguardar o, decorrer da viagem
    - 6.3.5.1.1. Pode fazer *skip*
- 6.3.6. Chega ao destino
- 6.3.7. Se é local já visitado
  - 6.3.7.1. Volta a seleccionar outras alternativas (6)
- 6.3.8. Se local não visitado
  - 6.3.8.1. Vai para “Acesso à cidade” (6.4)
- 6.4. Aceder à cidade
  - 6.4.1. Apresentada a pergunta de “Acesso à cidade”
  - 6.4.2. Possibilidade de uso ajuda
  - 6.4.3. Se responder errado
    - 6.4.3.1. Jogador é retido por pergunta errada durante um período de tempo e depois é libertado e deixado nas portas da cidade.
    - 6.4.3.2. Volta a alternativas (6).
  - 6.4.4. Se responder correto.
    - 6.4.4.1. Local NÃO é a de partida → Marca como “visitado” e volta a alternativas (6)
    - 6.4.4.2. Local É o de partida → Para o tempo! Fim de Jogo!

### 3.5. Público-alvo

A aplicação que se reporta, desenvolve-se com o intuito de chegar a um vasto grupo de jogadores.

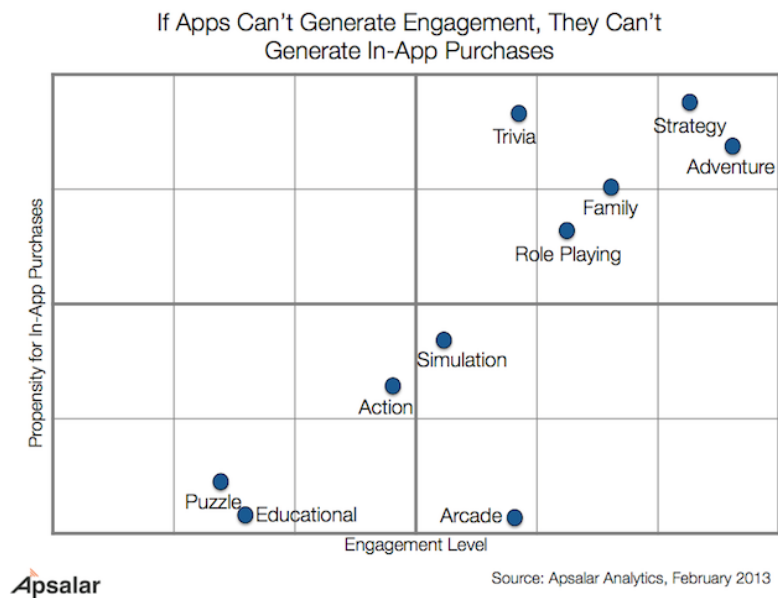
Apesar do jogo ser acessível a jogadores de todas as idades, gostos e sexos, existe efetivamente um conjunto de pessoas que se idealiza como potencial público-alvo: Pessoas interessadas em jogos de perguntas e respostas que pretendam mais do que um simples *quiz*.

O jogador tem o objetivo de, respondendo a questões, percorrer um conjunto de pontos em 80 dias. A passagem entre pontos implica viagens cuja rapidez depende do preço.

Pretende-se assim um produto que obrigue um jogador a uma gestão dual de tempo e dinheiro, capaz de aliciar quer pela sua vertente de *quiz*, quer pela sua vertente de estratégia.

Por outro lado, a existência de estatísticas de jogo e de um ranking de jogadores, com critérios de pontuação e desempate prevê-se que seja factor aliciante para pessoas com espírito competitivo.

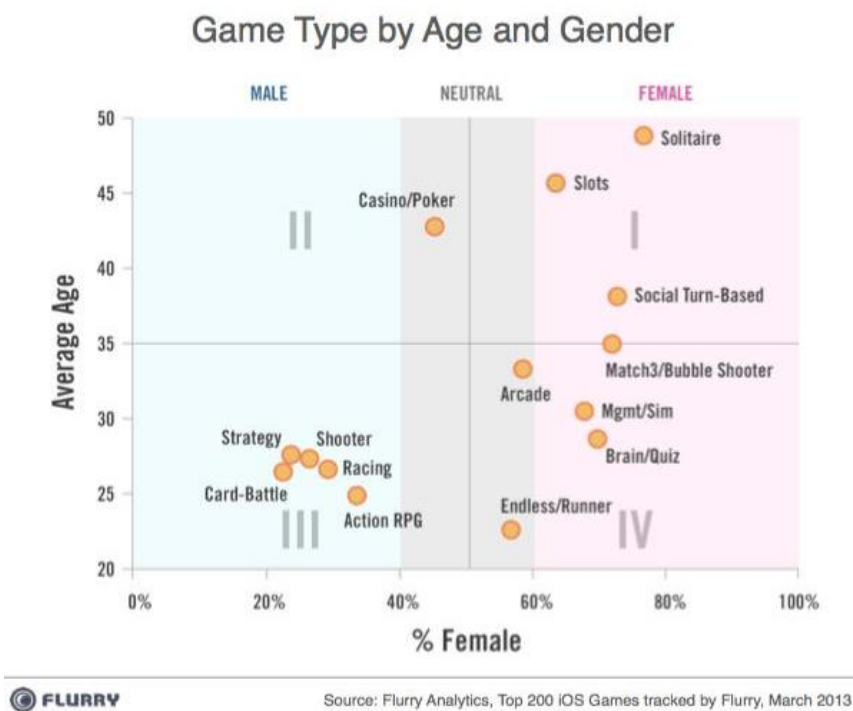
Analise-se a figura 6.



**Figura 6** - Dependência associada géneros de jogos

De facto, *quizes* e jogos de estratégia são jogos que, se por um lado são populares e propensos a serem comprados, por outro, se apelativos, despertam nos jogadores compromisso e dependência.

A figura 7 ajuda a esclarecer que tipo de jogadores que estarão interessados na aplicação.



**Figura 7** - Tipos de jogos por sexo e idade

Se os jogos de estratégia são jogados maioritariamente por homens, perto de 70% dos jogadores de *quiz* são mulheres, curiosamente, ambos entre os 25 e os 30 anos.

Assim, sendo o público-alvo da aplicação, pessoas interessadas em jogos de perguntas e respostas que pretendam mais do que um simples *quiz*, prevêem-se como maiores interessados homens e mulheres entre os 20 e os 30 anos.

## 4. Requisitos

O levantamento de requisitos obriga a um processo de análise e sistematização do que se pretende implementar.

Caracterizado o modelo de jogo importa organizar as funcionalidades oferecidas pelo jogo tendo em conta as restrições e decisões operacionais e tecnológicas, procurando uma visão consolidada que seja capaz de uniformizar interesses. Assim, é necessário tomar consciência dos requisitos que a equipa pretende implementar e procurar a aprovação por parte do cliente. Por outro lado, é necessário conhecer as funcionalidades e restrições que o cliente idealizou.

Com todos os intervenientes em sintonia pode-se ir mais além, procurando encontrar os requisitos ainda desconhecidos, quer pela parte interessada, quer pela equipa de desenvolvimento.

Identificados, filtrados e negociados os requisitos, é hora de os documentar, passa-los de forma estruturada para linguagem textual e posteriormente traduzi-los para uma linguagem gráfica.

Nas próximas subsecções expõe-se o resultado do processo de levantamento de requisitos.

### 4.1. Requisitos Funcionais

Os requisitos funcionais devem expor o comportamento e as funcionalidades que o sistema oferece. Assim, e já estando idealizado o modelo de jogo, pretende-se expor de uma forma clara e estruturada o comportamento e as interações com o sistema enquanto jogo.

Neste contexto listam-se as funcionalidades que se pretendem implementar:

- O jogo deve permitir o registo e autenticação de jogadores.
- Um jogador poderá iniciar um novo jogo.
- O jogador deverá ter a possibilidade de escolher o nível de dificuldade em que pretende jogar e qual o ponto de partida em que pretende começar.
- O sistema deverá oferecer ao jogador a possibilidade de ganhar a chave para determinada cidade. Para tal, deverá em cada ponto ser lançada uma pergunta, com grau de dificuldade que é função do nível escolhido pelo jogador.

- O sistema deverá lançar perguntas com 4 alternativas de resposta;
- O sistema deverá lançar perguntas que tenham vídeo, imagem ou som associado.
- Em jogo, o sistema deve permitir ao jogador transitar entre diferentes pontos de paragem. Em cada ponto, o sistema deve destacar os pontos de paragem alcançáveis e oferecer a possibilidade de consultar os diferentes meios de transporte disponíveis com respetivos preços e horários.
- Em viagem o sistema deverá reproduzir o som e a imagem associado ao meio de transporte.
- O dinheiro amealhado por um jogador deverá estar-lhe visível.
- O sistema deverá apresentar os trabalhos disponíveis com o respetivo valor euro/hora.
- Dever ser oferecido ao jogador a possibilidade de comprar e vender ajudas. Esta funcionalidade não estará associada a todos os pontos de paragem.
- O tempo de viagem restante deverá estar sempre visível ao jogador. Começando com 80 dias, o tempo apresentado no relógio deverá decrescer.
- O sistema terá mecanismos que ajustem a velocidade do tempo durante o período de resposta a uma pergunta. Deverá ainda oferecer a possibilidade de fazer *skip* enquanto decorre um trabalho ou uma viagem.
- O sistema deverá ser capaz de discriminar os pontos de paragem e em cada ponto disponibilizar ao jogador a informação relativa a viagens, trabalhos e ajudas correspondentes a cada ponto.
- Durante um jogo, o sistema deverá armazenar os pontos de paragem visitados permitindo ao jogador voltar atrás.
- O sistema deverá armazenar as respostas dadas por cada jogador por forma a:
  - Adaptar o grau de dificuldade médio das perguntas a lançar ao jogador.
  - Adaptar o grau de dificuldade de cada pergunta.
  - Disponibilizar ao jogador a *Ajuda do Jogadores*.
- O jogador deverá em qualquer poder consultar os dados estatísticos relativos ao jogo e ao seu perfil.
- Findado o jogo, o sistema deverá informar o jogador se é ou não vencedor mostrando os dados estáticos relativos ao jogo.
- O sistema deve permitir ao jogador consultar o ranking de jogadores.
- O sistema deve oferecer uma interface que permita a um administrador autenticado inserir/remover/alterar respostas e hipóteses de resposta.

Parte destes requisitos funcionais foram imposições do cliente. A tabela sintetiza as funcionalidades impostas no enunciado, e a forma como se pretende dar-lhes resposta.

Apresentar o jogo em questão, explicando o seu modo e princípios de funcionamento.	Desenvolver opção de consulta do manual
Permitir aos seus utilizadores definir uma configuração inicial para o jogo que pretendem jogar, deixando-os escolher o tema do jogo, um ponto de início de jogo e o nível de dificuldade pretendido – o ponto de início do jogo definirá também o ponto de chegada, assumindo que foi dada uma “volta” completa ao globo terrestre de acordo como o tema do jogo em questão.	Descrito em 3.3.1
Ao longo do jogo, o sistema deve acompanhar os seus diversos utilizadores, rastreando-lhes as suas atividades e calculando o seu perfil de jogo.	<p>Não estando completamente definido o perfil de utilizador pretende-se que este englobe os seguintes campos:</p> <ul style="list-style-type: none"> <li>• Percentagem de respostas certas.</li> <li>• Locais visitados.</li> <li>• Dinheiro ganho.</li> <li>• Dinheiro gasto.</li> <li>• Ajudas compradas.</li> <li>• Ajudas preferidas.</li> <li>• Meio de transporte preferido.</li> </ul> <p>Ao longo do jogo, cada ação do jogador deverá atualizar o respetivo campo do perfil.</p>
Ao longo do tempo, o sistema deve adaptar a dificuldade do jogo à perícia e conhecimento do seu utilizador, com base no seu perfil que foi gerado e caracterizado até ao momento.	Descrito em 6.1.3
A qualquer momento, o sistema deve saber justificar as questões apresentadas e apresentar (...) a resposta de acordo com algum tipo de regra definida.	A informação relativa a uma pergunta será armazenada com o grau de dificuldade, as hipóteses de resposta, a resposta correta e a fonte de onde foi retirada a pergunta/resposta
As questões deverão integrar diferentes tipos de dados – numéricos, alfanuméricos, georreferenciados, som, voz, imagem, vídeo, etc.	As questões lançadas serão sobre uma imagem, vídeo ou música. A recolha destes dados já está a ser elaborada.
Ao longo do jogo o utilizador deverá ter acesso (se desejar) a um <i>dashboard</i> (painel de controlo) que lhe fornecerá um conjunto de estatísticas sobre o jogo em curso ou sobre o seu perfil de jogo.	Descrito em 3.3.11. Os dados referentes ao perfil do jogador estarão disponíveis para consulta em qualquer momento do jogo.
A forma como o jogo se desenvolverá será da responsabilidade de cada grupo de trabalho, mas o seu “tabuleiro” deverá ser o globo terrestre.	O jogo desenvolver-se-á sobre pontos do planisfério, sendo este o tabuleiro de jogo.

**Tabela 2** - Esquema resumo das funcionalidades obrigatórias.

## 4.2. Requisitos não funcionais

A presente subsecção apresenta as restrições quer a nível de funcionalidades, quer a nível de tecnologias a adotar.

- O sistema não deve permitir jogo ou consultas de jogadores não autenticados.
- O sistema deve ajustar as possibilidades de compra de ajudas ou viagens ao dinheiro que o jogador tem disponível.
- O sistema deve impedir que um jogador a partir de um ponto de paragem ainda sem chave (sem resposta certa a uma pergunta associada ao ponto) aceda aos pontos não visitados.
- O sistema deve cuidar de que se lance perguntas só sobre a área que abrange a cidade.
- O sistema deve reagir a uma resposta errada do jogador impedindo que este prossiga por um período de tempo que é função do número de respostas erradas.
- O sistema deve impedir a inclusão de mais do que uma hipótese de resposta certa por pergunta.
- O sistema deve impedir que o jogador guarde o jogo no intervalo de tempo entre o lançamento da pergunta e a sua resposta.
- A parte interactiva da aplicação deverá funcionar num browser desktop ou móvel com um alto nível de implementação do standard HTML5.
- O sistema deve ser construído de forma a minimizar o impacto temporal da duração das suas ações internas de forma a não interferirem com a passagem do tempo de jogo.
- O sistema deve oferecer ao Utilizador a possibilidade de usufruir de suas ajudas principais de maneira rápida e fácil.
- O sistema deve impedir que o jogador guarde o jogo no intervalo de tempo entre o lançamento da pergunta e a sua resposta.

No enunciado do projeto, eram impostas restrições no que às decisões tecnológicas dizia respeito. A tabela 3 apresenta a forma como se pretende responder aos requisitos impostos.

Resposta Tecnológica ► Requisito ▼	MS SQL Server	MS SQL Server Integration Services	MS Entity Framework	VS LightSwitch	ASP.NET Web Pages 3.0 (Razor)	MS TypeScript 1.0	AngularJs	jQuery	Less	Twitter Bootstrap
Modelo Relacional	✓									
Tecnologias Microsoft	✓	✓	✓	✓	✓	✓				
Incorporação de dados externos		✓								
Manutenção dos dados de suporte ao jogo				✓						
Clientes universais (Web browsers)				✓	✓	✓	✓	✓	✓	✓
Dispositivos Móveis					✓	✓	✓	✓	✓	✓

**Tabela 3** - Tabela com as tecnologias utilizadas para os requisitos funcionais obrigatórios.

Agora descrevem-se as tecnologias adotadas:

- **Microsoft SQL Server:** Sistema Gestor de Base de Dados Relacional desenvolvido pela Microsoft e baseado no Sybase.
- **Microsoft SQL Server Integration Services:** Componente de ETL do SQL Server que permite extração transformação e importação ou exportação de dados entre fontes heterogêneas.
- **Microsoft Entity Framework:** Framework ORM criada pela Microsoft que permite criar facilmente uma camada de acesso a dados mapeando uma estrutura de classes numa estrutura relacional.
- **Visual Studio LightSwitch:** É uma extensão (ou uma ferramenta RAD) que permite criar aplicações LOB muito rapidamente.
- **ASP.NET Web Pages 3.0 (Razor):** Forma de combinar código do servidor com HTML de forma a disponibilizar conteúdo dinâmico aos clientes (browsers) utilizando os mais standards mais recentes de desenvolvimento web.
- **TypeScript:** É uma linguagem de programação de código aberto que permite escrever *Javascript* de forma rigorosa e tipada. Foi criado para permitir ajudar a escrever aplicações de dimensão considerável em *Javascript* sem se perder o controlo do desenvolvimento. Esta foi desenvolvida na Microsoft por Anders Hejlsberg, criador do Turbo Pascal e do Delphi e arquiteto principal do Visual C#.
- **AngularJs:** Framework de código aberto de desenvolvimento de aplicações web em *Javascript* de forma estruturada segundo o paradigma MVW (*Model-View-Whatever*) sendo incorporados mecanismos como *databinding*, *routing*, *controllers*, *dependency injection*, etc.
- **jQuery:** Biblioteca *Javascript* que tem como principal vantagem a simplificação da interação do *Javascript* com o HTML.
- **Twitter Bootstrap:** Framework de apresentação para HTML5.
- **Less:** Linguagem para criação de folhas de estilos (CSS) dinâmicas.



## **5. Questões relativas ao funcionamento do sistema**

Considera-se importante descrever como serão implementadas determinadas funcionalidades do sistema que obrigaram a uma reflexão mais profunda para garantir que o se está a propor é efetivamente realizável. Assim, na presente secção, apresenta-se o resultado dessa reflexão, uma perspectiva um pouco mais interna, de algumas funcionalidades que se consideraram relevantes detalhar.

### **5.1.1 Grafo de pontos de paragem**

Os pontos de paragem são nodos de um grafo, cujos arcos representam os caminhos possíveis entre os pontos.

A distância entre os pontos será o peso dos arcos.

### **5.1.2 Passagem do Tempo, dinheiro e meios de transporte**

A passagem do tempo é uma das funcionalidades que se considera mais crítica para o desenvolvimento do jogo, quer pela importância que tem face à filosofia de jogo, quer pela complexidade que se prevê que venha a ter a sua implementação.

Pretende-se que o jogo não pare e que em determinadas situações o tempo esteja sujeito a acelerações. Por exemplo, a resposta a uma pergunta não correrá em tempo real, cada segundo corresponderá a um intervalo de tempo que se antecipa superior a um minuto.

Da mesma forma, de cada vez que o jogador erra a pergunta, pretende-se que o tempo de jogo correspondente ao lançamento cresça exponencialmente com o número de respostas erradas.

A inclusão de mecanismos que geram dinheiro e de diferentes meios de transporte não é tão vital para o modelo de jogo como a problemática do tempo. Contudo, consideram-se fatores que tornarão o jogo bastante mais atrativo.

Ter um orçamento o tempo para gerir, definir rotas e meios de transporte em função desse orçamento e desse tempo fará com que o jogo não seja um simples *quiz*, acrescentando-lhe a dimensão de estratégia.

A velocidade dos meios de transporte define a duração de uma viagem. O meio de transporte escolhido depende do dinheiro que o jogador tem disponível. A escolha de um meio de transporte barato faz com que o jogador perca mais tempo.

As entradas de dinheiro têm que ser suficientemente altas para que um jogador médio consiga terminar o jogo mas não ao ponto de fazer perder a problemática da gestão dual de tempo e dinheiro.

A definição destas variáveis é vital para a coerência do jogo, sendo muito difícil de os definir sem que a aplicação esteja a funcionar. É uma problemática que carece de trabalho e que se prevê minucioso e iterativo. Assim, está previsto um período de tempo para testar e ajustar estes valores para que seja não só possível como também desafiante, dar a volta ao mundo em 80 dias.

### 5.1.3 Mecanismos de aprendizagem do sistema

O jogador escolhe um de 3 níveis (fácil, médio, difícil). Interno ao sistema, a cada pergunta estará associado um grau de dificuldade (por exemplo numa escala de 1 a 10). Cada um dos 3 níveis visíveis ao jogador terá como subconjunto graus de dificuldade. Por exemplo, ao médio estariam associados os graus de dificuldade 5,6 e 7.

Um dos requisitos impostos no enunciado do projeto é a capacidade do sistema de se adaptar ao jogo de um dado jogador.

O nível de dificuldade escolhido pelo jogador pode ser facilmente convertido para uma escala de 1 a 10. Represente-se por  $n$  o nível selecionado, já convertido.

Seja  $r$  a percentagem de respostas certas do jogador:

$$r = \frac{n^{\circ} \text{ respostas certas}}{n^{\circ} \text{ respostas}}$$

O grau de dificuldade médio ( $g$ ) das perguntas a atribuir ao jogador é:

$$g = \frac{r * 10 + n}{2}$$

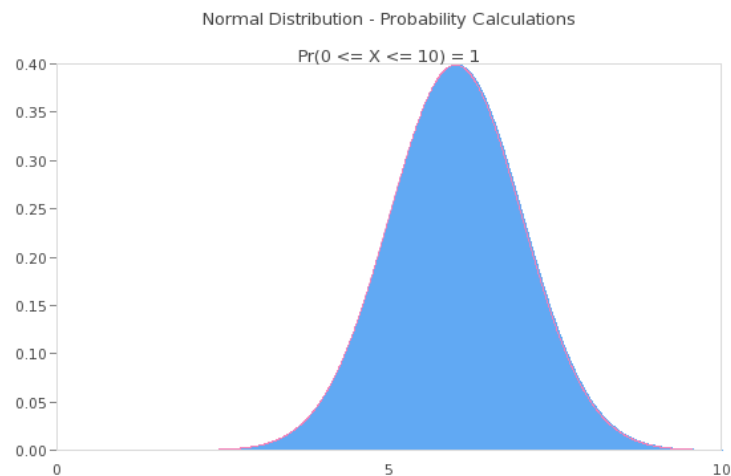
Pretende-se ainda adotar um mecanismo semelhante para atualizar periodicamente o grau de dificuldade de uma pergunta. Neste contexto,  $r$  é percentagem de respostas corretas à pergunta (contabilizando as respostas de todos os jogadores) e  $n$  é o grau de dificuldade pré-definido para a pergunta.

### 5.1.4 Mecanismo probabilístico de lançamento de perguntas

A opção por um nível de dificuldade não exclui a possibilidade de serem lançadas perguntas de um nível diferente. O mecanismo de seleção de perguntas será probabilístico, seguindo uma distribuição normal.

Considere-se que é 6 o grau de dificuldade médio das perguntas a atribuir a um jogador (ver 3.5.3).

Seja variável aleatória  $X$ : "Número de vezes em que sai uma pergunta de grau de dificuldade  $n$ "  $X \sim N(6, 1)$  com gráfico:



**Figura 8** - Gráfico representativo da distribuição normal  $X \sim N(6, 1)$ <sup>4</sup>.

---

<sup>4</sup> Gráfico gerado em: [http://www.mathcracker.com/normal\\_probability.php](http://www.mathcracker.com/normal_probability.php)

## 6. Modelação

Com a definição textual dos requisitos, o passo seguinte foi o de retratar o sistema a partir de uma linguagem gráfica capaz de demonstrar o seu comportamento.

O processo de modelação terá influência a dois níveis. Pretende-se, por um lado, que constitua parte fundamental da documentação do projeto. Por outro lado, prevê-se necessário que represente o esboço da aplicação, a partir do qual vai ser possível a sua implementação. Note-se que grande parte do tempo disponibilizado para a conceção do projeto estava predestinado à conceção do modelo de jogo, ao levantamento e análise de requisitos e a elaboração do modelo de base de dados, destacando-se a importância da adoção de uma metodologia de desenvolvimento de *software*. Assim quão mais cuidado e detalhado for o resultado desta fase de modelação mais produtiva será a implementação propriamente dita do sistema.

Os conhecimentos previamente adquiridos foram fundamentais para o cumprimento desta tarefa. Ao longo de todo o processo de modelação, procurou-se seguir a metodologia apresentada na unidade curricular *Desenvolvimento de Sistemas de Software*. Assim, o primeiro passo foi o de procurar reunir todos os termos que representarão classes e relações elaborando um modelo de domínio. Feito isto e recorrendo ao levantamento textual dos requisitos do sistema elaborou-se um modelo de *Use Case*. O passo seguinte foi o de identificar os subsistemas da aplicação criando os respetivos diagramas de classe. Com os subsistemas determinados foi momento de procurar elevar a especificação textual dos use cases a um nível mais próximo da implementação, traduzindo-os para diagramas de sequencia.

O resultado de cada um destes modelos apresenta-se nas subsecções seguintes.

## 6.1. Modelo de Domínio

Pretende-se que o modelo de domínio represente de forma relacionada as entidades relevantes para o sistema a desenvolver. Neste contexto, o primeiro passo foi o de procurar as palavras-chave candidatas a classes no modelo. Algumas surgiram ainda antes de conceber o modelo de jogo, aquando da apresentação do trabalho prático e da leitura do enunciado:

- Jogo.
- Jogador.
- Questão.
- Resposta.
- Nível de dificuldade.
- Ranking.
- Mapa.

Outras foram surgindo a quando da idealização do modelo de jogo:

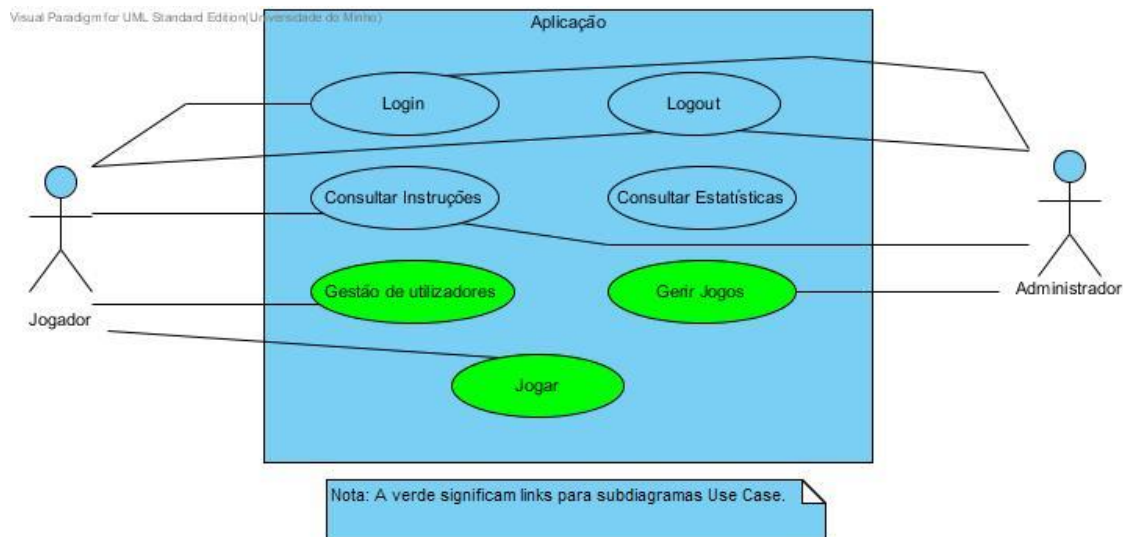
- Tempo.
- Trabalho.
- Viagem.
- Cidades.
- Meio de Transporte.
- Horário.
- Saldo.
- Custo.
- Chave.
- ...

De facto, a elaboração deste modelo, mais não foi do que organizar e passar para o papel o que já estava definido havia algum tempo.



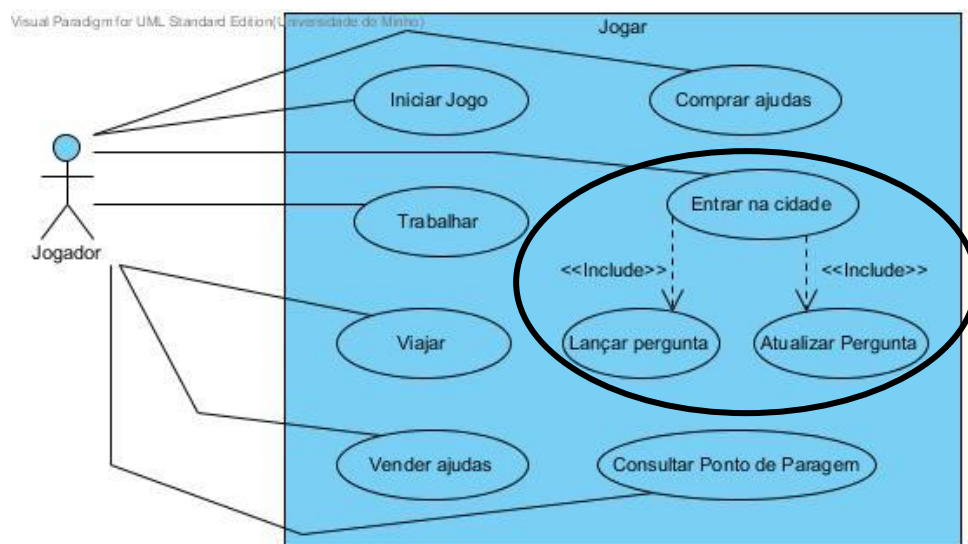
## 6.2. Modelo de Use Cases

Tendo definidas as funcionalidades que se pretendiam implementar, pretendia-se um modelo capaz de representar as interações dos utilizadores, sejam eles jogadores ou administradores, com o sistema. Definiu-se assim o modelo de *Use Case*, apresentando-se na figura 10 o diagrama principal.



**Figura 10** - Use Case principal da aplicação.

Mais do que listar as ações que disponíveis, importa detalhar a forma como o sistema responde a uma interação do utilizador. A especificação textual de cada *Use Case* é assim parte fundamental do diagrama. Das possíveis interações que se prevêem possíveis, destaca-se *Entrar na Cidade*.



**Figura 11** - Use Cases relativos ao jogo.

Para que um jogador possa entrar na cidade, marcando-se o ponto como visitado, é necessário que responda acertadamente a uma pergunta sobre a região que abrange a cidade em questão. Relembre-se o mecanismo de aprendizagem do sistema e o mecanismo de lançamento de perguntas apresentados nas secções 5.1.3 e 5.1.4, respetivamente.

O processo de lançamento de uma pergunta é um processo detalhado no *Use Case Lançar Pergunta*. A especificação textual:

<b>Brief Description</b>		
<b>Preconditions</b>		
Jogo a decorrer, com nível n de dificuldade (de 1 a 10)		
<b>Post-conditions</b>		
Pergunta		
<b>Flow of Events</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Determina a percentagem de respostas certas do jogador, dividindo o número de respostas certas pelo número de respostas dadas. Seja r esse valor.
	2	Determina o grau de dificuldade médio g das perguntas a atribuir ao jogador através da seguinte fórmula: $g_i = (r \cdot 10 + n) / 2$
	3	Seguindo uma distribuição normal centrada em $g_i$ , determina o grau de dificuldade g da pergunta a lançar.
	4	Selecciona uma pergunta válida de grau g.
	5	Selecciona 4 das hipóteses de resposta associadas à pergunta seleccionada: a hipótese correcta (tipo 0), duas hipóteses susceptíveis de causar dúvida ao jogador (tipo 1) uma hipótese completamente errada (de tipo 2).
	6	Lança pergunta e respectivas hipóteses de resposta

**Figura 12** - Descrição tabular do *Use Case Lançar Pergunta*.

Respondida a uma pergunta é hora de atualizar não só o número de respostas certas/erradas do jogador como também a propria pergunta para que, por um lado, se atualize o grau de dificuldade da pergunta, por outro, seja possível lançar a *ajuda dos jogadores* (ver 3.3.8). A especificação textual do *Use Case Atualizar Pergunta*:

<b>Brief Description</b>		
<b>Preconditions</b>		
Hipótese de resposta seleccionada		
<b>Post-conditions</b>		
Pergunta actualizada		
<b>Flow of Events</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Incrementa o número de respostas dadas à hipótese seleccionada
	2	Calcula r, a percentagem de respostas certas dadas à pergunta, dividindo o número de respostas certas pelo número total de respostas dadas.
	3	Sendo n o grau de dificuldade pré definido para a pergunta, calcula $g_p$ , o grau de dificuldade médio da pergunta: $g_p = (r \cdot 10 + n) / 2$

**Figura 13** - Descrição textual do *Use Case Atualizar Pergunta*.



Apresentados as acções incluídas para *Entrar na Cidade*, pode-se assim detalhar o processo de resposta a uma pergunta.

<b>Brief Description</b>		
<b>Preconditions</b>		
Jogo iniciado		
<b>Post-conditions</b>		
Ponto visitado		
<b>Flow of Events</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Jogador selecciona entrar cidade
	2	<<Include>> Lançar pergunta
	3	Acelera passagem do tempo
	4	Selecciona opção de resposta
	5	Valida resposta
	6	Marca ponto como visitado
	7	Actualiza estatísticas do jogador, incrementando o número de respostas certas
	8	<<Include>> Actualizar pergunta
	9	Adiciona à conta do jogador o dinheiro da pergunta
	10	Informa que o jogador tem a chave da cidade
<b>Cenário Alternativo 1 [Utiliza Ajuda](4)</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Clica em Utilizar Ajuda
	2	Apresenta ajudas adquiridas
	3	Selecciona Ajuda
	4	Apresenta resultado da ajuda
<b>Cenário Alternativo 2 [Jogador não tem ajudas](4.2)</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Informa que o jogador não tem ajudas
	2	Regressa a 6
<b>Cenário Excepção 1 [Resposta errada](6)</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Informa que a resposta dada está incorrecta
	2	Incrementa o número de respostas erradas na cidade
	3	<<Include>> Actualizar pergunta
	4	Calcula o tempo de retenção do jogador, exponencial ao número de respostas erradas
	5	Informa o tempo de retenção do jogador
	6	Liberta o jogador

**Figura 14** - Descrição textual do *Use Case Entrar na Cidade*.

Considera-se que não há lugar para as restantes especificações textuais no corpo do relatório, remetendo-as para anexo.

## 6.3. Identificação dos subsistemas e Diagramas de Classe

Com a definição e especificação das interações do utilizador com o sistema, o passo seguinte foi o de tentar “partir” o sistema num conjunto de subsistemas, cada um responsável por um conjunto de classes e de operações. Assim, consideraram-se os subsistemas:

- Jogos
- Viagens
- Perguntas

As entidades operações que representam e manipulam, utilizadores, níveis de dificuldade, ajudas disponíveis, trabalhos realizados e toda informação relativa ao jogo de um jogador foram incluídas no subsistema *Jogos*.

A figura 15 apresenta o diagrama de classes correspondente.

O subsistema viagens engloba as entidades que dizem respeito às cidades/pontos de passagem existentes, quais as rotas possíveis, que ajudas, trabalhos e meios de transporte se encontram disponíveis em cada cidade a visitar. Em suma, operações e classes que dizem respeito as viagens possíveis e ao que fazer em cada destino, fazem parte do subsistema *Viagens*.

A figura 16 apresenta o diagrama de classes correspondente.

Por fim, as entidades que representam e operam sobre perguntas, hipótese de resposta, grau de dificuldade, tema de jogo, tipos de ajuda disponíveis, são parte do subsistema *Perguntas*.

A figura 17 apresenta o diagrama de classes correspondente.

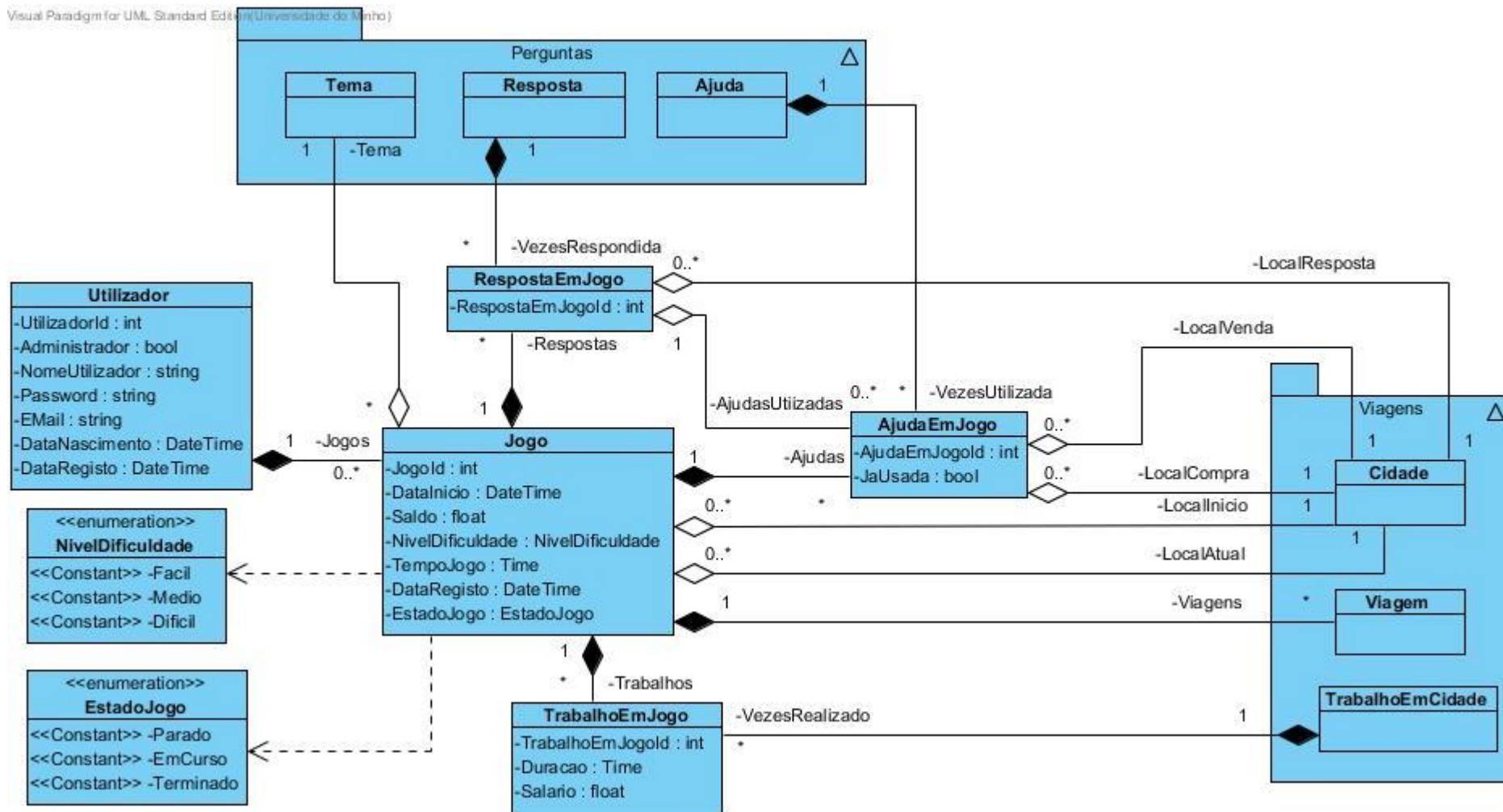


Figura 15 - Subsistema Jogos.

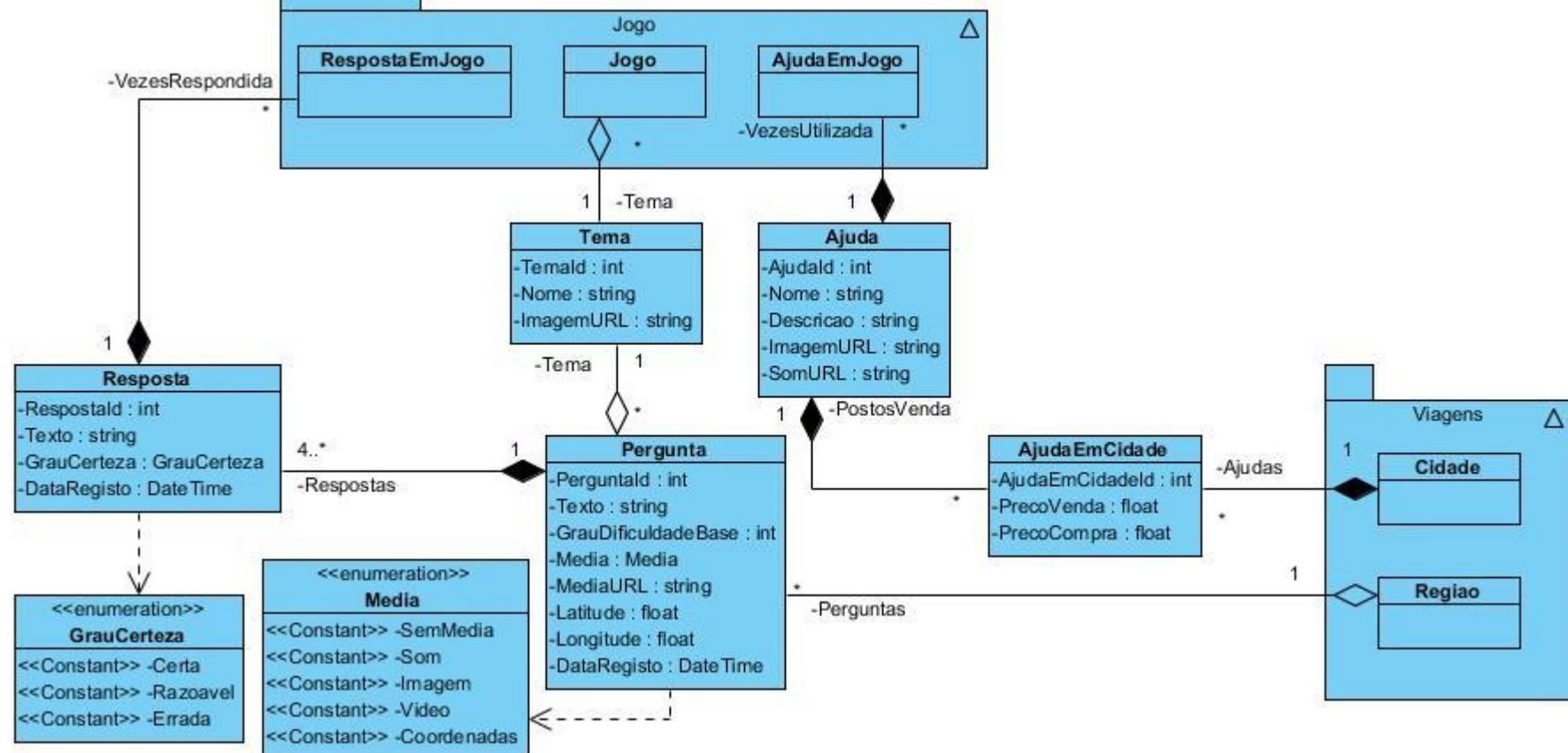


Figura 16 - Subsistema Perguntas.

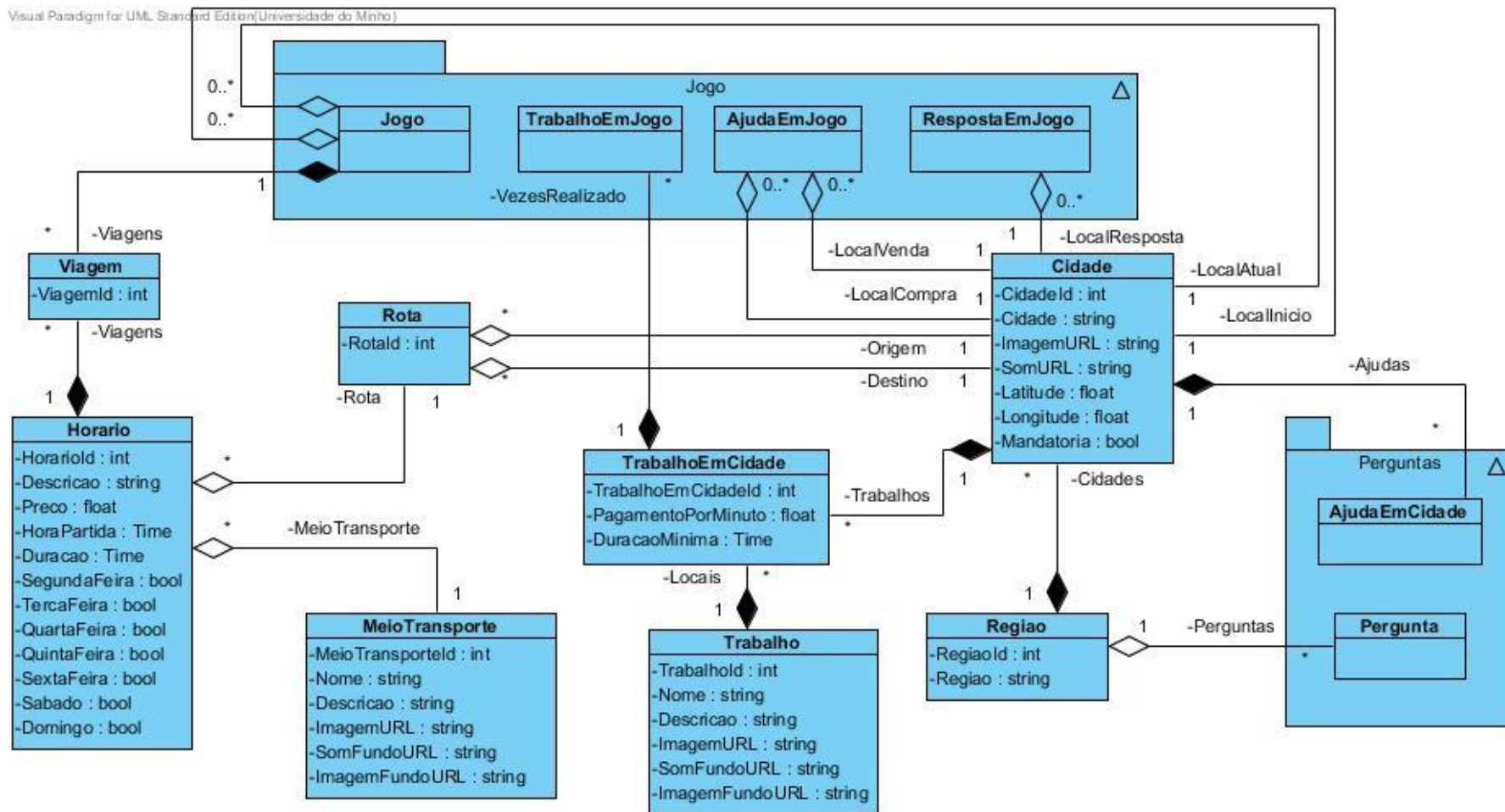


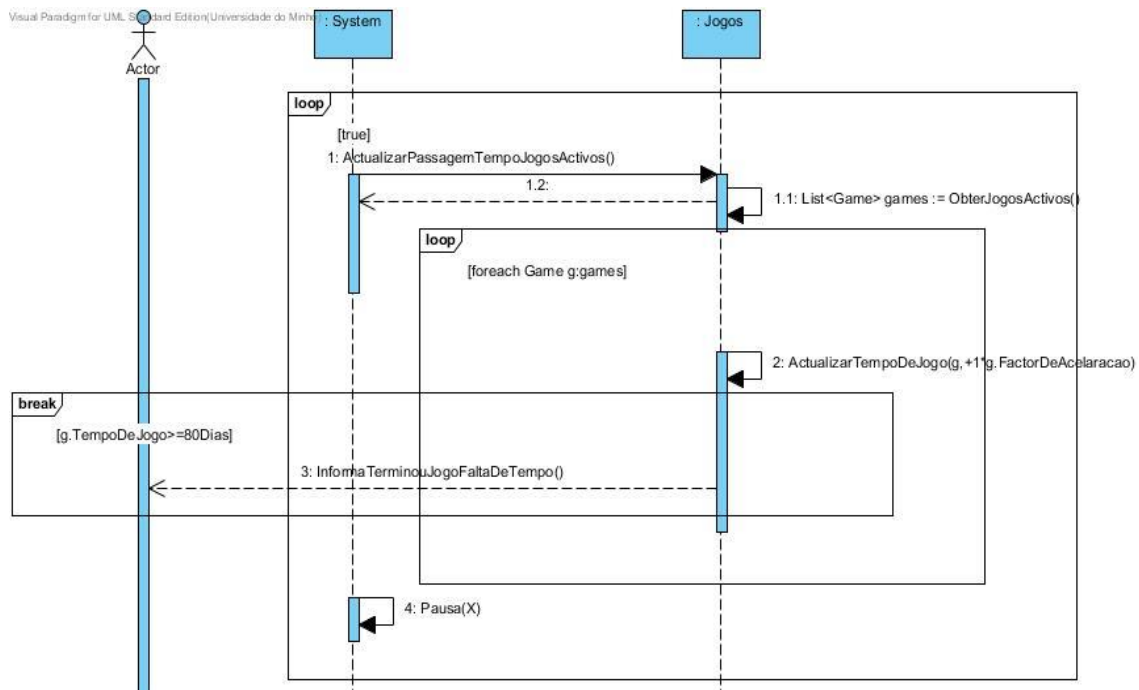
Figura 17 - Subsistema Viagens.

## 6.4 Diagramas de sequência

Definidos os subsistemas é agora possível elevar a especificação textual dos Use Case a um nível mais próximo do código, criando diagramas de sequência de subsistema.

Sendo o objetivo do jogo “dar a volta ao mundo em 80 dias”, o mecanismo de passagem do tempo é sem dúvida parte fulcral do funcionamento do jogo, estando associado a qualquer interação de um utilizador em jogo.

A figura 18 apresenta o diagrama de sequência correspondente.



**Figura 18** - Diagrama de Sequência da Passagem de tempo.

Dos diagramas elaborados, destaca-se ainda *EntrarNaCidade*, (figura 19) por se prever englobar um conjunto de mecanismos que deverão particularmente ser ponderados e estudos antes de partir para implementação.



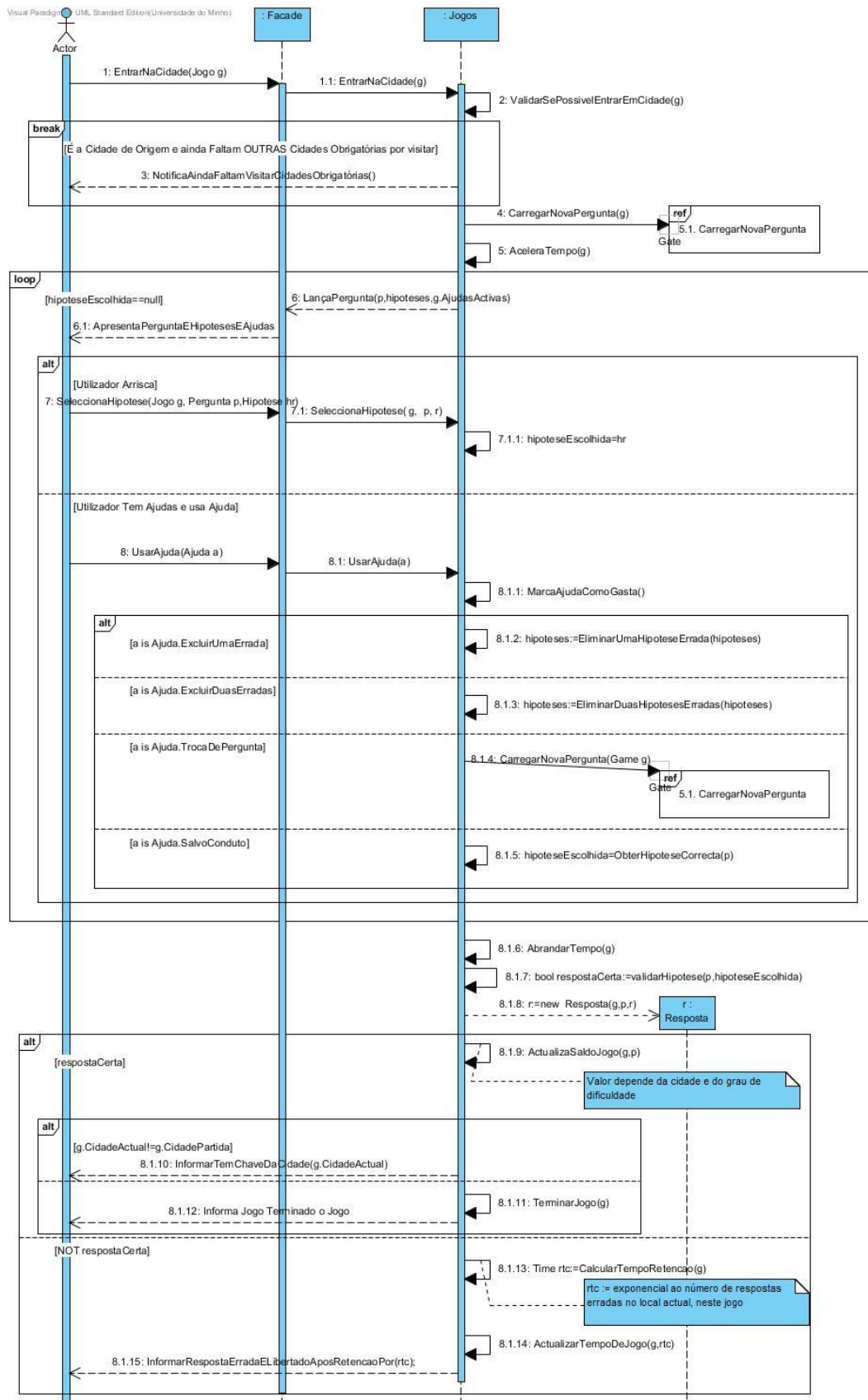


Figura 19 - Diagrama de Sequência de Entrar na Cidade.

Note-se que caso o jogador não tenha ainda a chave para entrar na cidade será necessário responder a, pelo menos, uma pergunta responder a uma pergunta. Daí a referencia para um outro diagrama de sequencia, *CarregarNovaPergunta*, que pode ser consultado em anexo, juntamente com os restantes diagramas de sequência.



## 7. Modelo de Base de dados

Terminado o processo de modelação dos requisitos levantados, importa definir o modelo de dados a usar. Pretende-se assim elaborar um plano de registo/armazenamento de toda a informação necessária ao suporte da aplicação.

Ora, um **jogador (Gamer)** regista-se no sistema indicando um nome de utilizador, um correio eletrónico e uma palavra-passe.

Cada **jogo (Game)** será jogado por um jogador, que define o ponto de partida, o dia e a hora inicial. A partir daí, serão percorridas um conjunto de pontos de paragem, cidades, (algumas obrigatórias) até que se volte ao ponto inicial.

Cada **cidade (City)**, terá associado um nome, uma imagem e um som de fundo e as coordenadas georreferências (latitude e longitude). Deverá ainda fazer-se a distinção dos pontos obrigatórios

Uma **pergunta (Question)** terá um tema e um grau de dificuldade estando associada a uma zona, não a uma cidade. Poderá ainda contemplar um anexo (imagem, vídeo ou som) sobre o qual recairá a pergunta.

Sobre o **tema (QuestionCategory)**, pretende-se apenas guardar o nome/descrição.

Uma **zona (WorldRegion)** consiste num aglomerado de cidades que contêm o mesmo conjunto de questões. Assim, na base de dados, cada cidade terá ainda algo que identifique a zona de que faz parte.

Para cada pergunta existe um conjunto de respostas, sendo que só uma delas será a correta

Uma **hipótese de resposta (AnswerChoice)** pode ser classificada em função do seu grau de correção em que 0 significa que está certa, 1 significa que está errada mas que é aceitável que coloque dúvidas ao jogador e 2 significa que se supõe que é declaradamente errada.

O sistema deve garantir que exista apenas uma resposta de grau 1 associada a cada pergunta. O lançamento de uma pergunta implica a seleção de 4 hipóteses de resposta, a de grau 0, duas de grau 1 e uma de grau 2.

Relembra-se que uma cidade visitada é uma cidade na qual o jogador já respondeu acertadamente a uma pergunta. Assim será necessário associar, em cada jogo, a resposta dada, não à zona, mas à cidade em que foi lançada. Aqui percebe-se a necessidade da existência de uma tabela de ligação (**Answer**) que faça corresponder a cada resposta num jogo a em que foi dada.

Uma transição entre uma cidade origem e uma cidade destino só é possível se entre elas existir uma ligação. Uma **ligação (Route)**, mais não é do que a identificação da origem e do destino.

Os **meios de transporte (ModeOfTravel)** podem ser vários: carros, comboios, aviões, autocarro, etc. Desta tabela, além da descrição do meio de transporte, pretende-se guardar uma imagem e um som associado, para fazer passar durante a viagem.

Note-se que os meios de transporte são comuns a várias ligações, da mesma forma que as ligações podem ser efetuadas a partir de vários meios de transporte. Contudo, para cada par meio de transporte/ligação existe um horário fixo.

Um **horário (Schedule)** tem associada uma duração, um preço, uma hora de partida e os dias da semana em que está disponível. Fica a faltar a ligação entre estas tabelas e um jogo de um jogador. Assim, uma **viagem (Trip)** é executada durante um jogo num determinado horário.

Será necessário guardar os diferentes tipos de **ajuda** disponíveis. Esta informação constará da tabela **LifeLine**. Sendo que determinadas cidades funcionarão também como ponto de compra e venda de ajudas, é necessário uma tabela entre as cidades e as ajudas. Neste contexto, em **CityLifeLine** ter-se-á, para cada par cidade/ajuda, o preço de compra e o preço de venda da ajuda.

Uma vez adquirida, a ajuda pode ser usada em qualquer cidade ou ainda vendida, se se tratar de um ponto de compra de ajudas. A tabela **GameLifeLine** permite rastrear a utilização de uma ajuda num dado jogo. Assim, prevê-se necessário guardar além de uma ligação ao jogo em questão, duas ligações a **cityLifeLine** (ponto de compra e ponto de venda) e uma para **Answer** (local onde a ajuda foi utilizada).

Relembre-se que se pretende dar ao jogador a possibilidade de, despendendo tempo, ganhar dinheiro. Os **trabalhos (Labour)** terão associada uma descrição, uma imagem e um som que os identifique. Sendo que a disponibilidade e remuneração de um dado trabalho depende da cidade, é necessário uma tabela (**CityLabour**) intermédia que faça a ligação. Da tabela deverá constar uma ligação para **Labour**, uma outra para **City** e o valor que o jogador ganhará por cada hora despendida.

Sendo um trabalho executado numa dada cidade durante um determinado tempo no decorrer de um jogo, cada entrada da tabela **GameLabour** terá uma ligação para **Game**, uma ligação para **CityLabour** e ainda o tempo despendido na execução da tarefa.

As subsecções seguintes apresentam o modelo conceptual e lógico da Base de Dados aqui descrita.

## 7.1. Modelo Conceptual

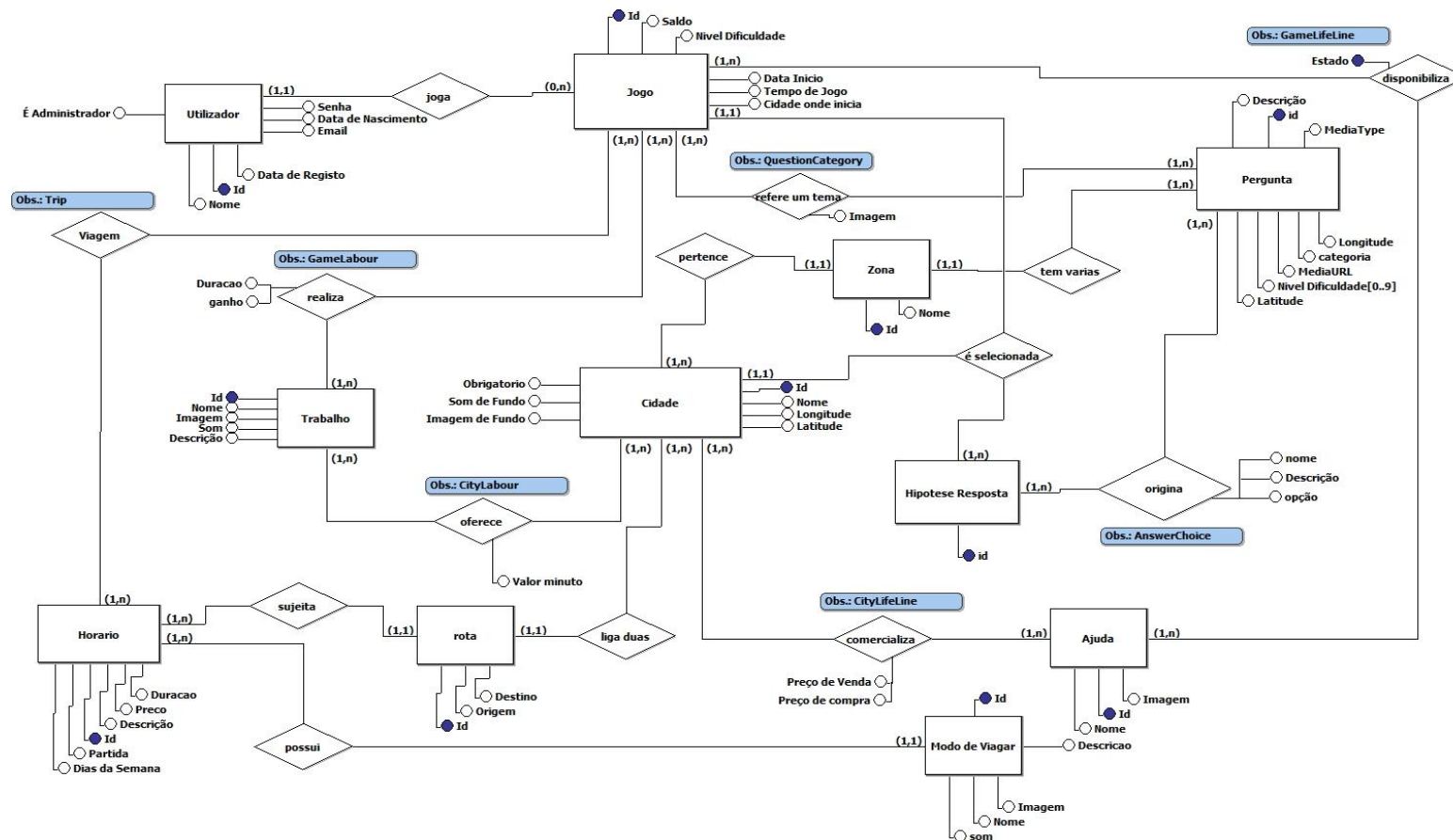


Figura 20 - Modelo conceptual da base de dados.

## 7.2. Modelo Lógico

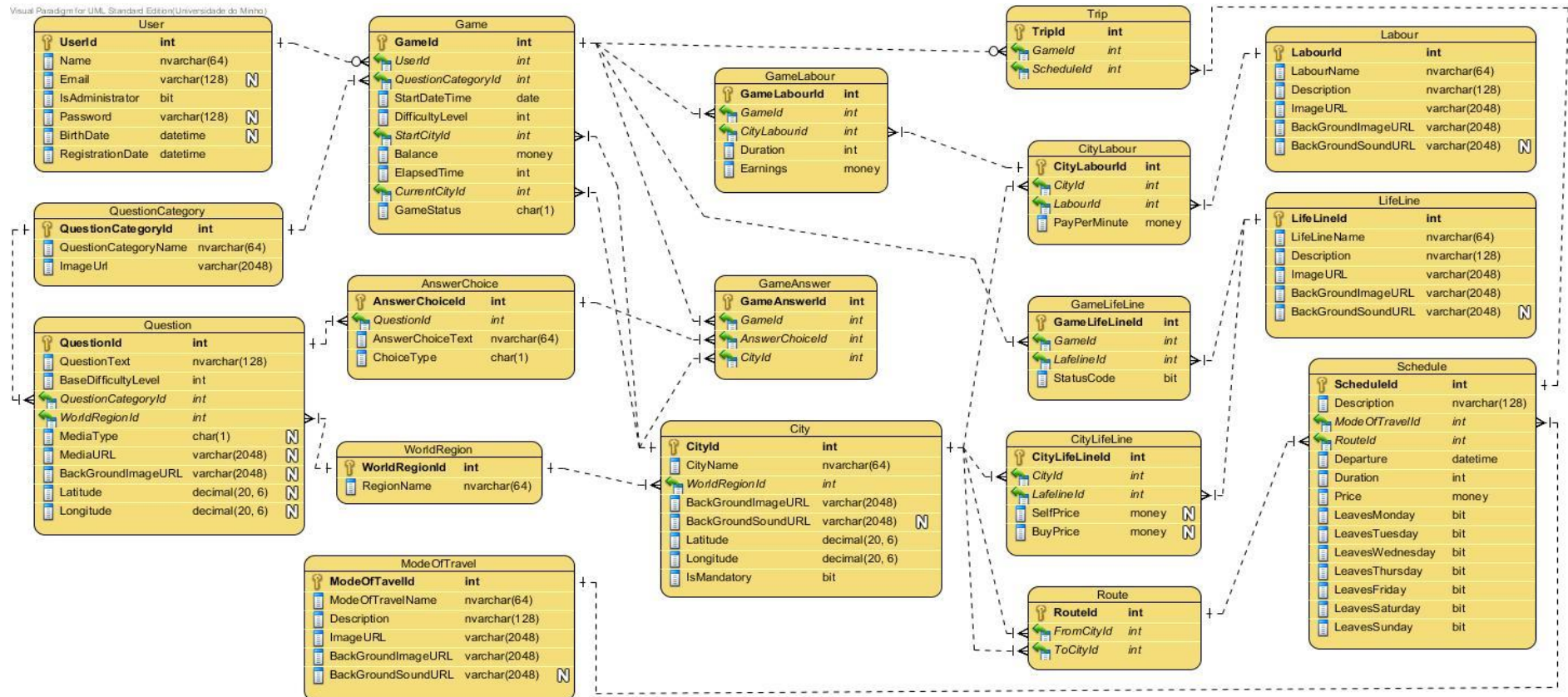


Figura 21 - Modelo lógico da base de dados.

## 8. Conclusões e Trabalho Futuro

Numa perspetiva diferente daquela até então usada no desenvolvimento dos projetos realizados durante a licenciatura, foi proposto um projeto que deveria ser executado em três fases: Fundamentação, Especificação e Construção.

Neste sentido, percorrem-se uma série de passos desde a leitura do enunciado até ao momento em que efetivamente se irá começar a implementar a aplicação. E é no final da fase de fundamentação que o nosso trabalho se encontra. Estamos no momento em que se começa a implementar.

Numa primeira fase de especificação as principais questões sobre o funcionamento do jogo foram levantadas. E muitas questões surgiram, muitas dúvidas se levantaram, muitas alterações se fez ao primeiro esboço mental do que seria a aplicação. De facto, e ainda não tendo começado a codificação, já se percebe a vantagem destas dúvidas e consequentes alterações no modelo terem surgido nesse momento e não já depois de termos começado a implementar.

Passou-se assim por um processo de definição de objetivos, de debate e escolha de ideias, com a certeza de quanto mais analisada e ponderada fosse cada decisão tomada, melhor será o resultado final e maior será a produtividade da equipa. De um quadro cheio de ideias, que foram sendo gradualmente filtradas, chegou-se assim ao conjunto de funcionalidades que levou à definição do modelo de jogo.

Nesta fase de especificação, foi necessário elevar estas especificações a um nível mais próximo do desenvolvimento. Já estando identificados os requisitos principais, em termos de modelo de jogo, identificaram-se os restantes (funcionais e não funcionais) e procedeu-se à definição textual dos mesmos. O passo seguinte foi o desenvolver os requisitos levantados, transformando o texto, de forma gradual, em diagramas de Use Case, de Sequência e de Classe.

Ainda nesta fase, desenhou-se o que será o alicerce da aplicação, a base de dados, concebendo os modelos conceptual, lógico e físico.

Desta forma estão concluídas as fases de fundamentação e especificação e o produto estará, espera-se, devidamente documentado.

Concluída esta fase considera-se que o modelo apresentado pode perder pela dimensão e pela complexidade que terá a sua implementação. Desenvolver todas as funcionalidades pretendidas, ajustar as variáveis por forma a dar jogabilidade e atratividade ao jogo será um processo que se prevê exigente e longo. Mas ao mesmo tempo esperamos que o modelo encontrado permita alcançar um jogo diferente, desafiante e divertido.

Ficará a faltar a construção do *software*, por em prática as ideias apuradas e que nesse momento já deverão estar suficientemente maduras para serem facilmente passadas para o código.

Tendo em conta o tempo que restará para esta fase de construção, face ao tempo global disponibilizado para a conceção do software, é fundamental que se parta para esta fase com a ideia clara do que se vai implementar ou seja, que o resultado destas das duas primeiras fases seja o mais preciso, pondera e detalhado possível.

## **Bibliografia**

Monopoly, Instruções.

Trivial Pursuit, Instruções.

Norma NP ISO 10006 2006

## Referências WWW

[01] **<http://www.mathcracker.com/>**

Ferramenta online para cálculos matemáticos.

[02] **[pt.wikipedia.org/wiki/A\\_Volta\\_ao\\_Mundo\\_em\\_80\\_Dias](http://pt.wikipedia.org/wiki/A_Volta_ao_Mundo_em_80_Dias)**

Página da wikipédia que contém o essencial sobre o livro *A Volta ao Mundo em 80 Dias*.

[03] **[www.nationalgeographic.com](http://www.nationalgeographic.com)**

Página principal da National Geographic. Aqui encontram-se disponíveis jogos, informação, vídeos e imagens sobre diversos temas, como fontes de energia, catástrofes naturais, animais, viagens, turismo, aventura etc.

[04] **<http://www.wmf.org/>**

Site da *World Monuments Fund*, organização não governamental internacional voltada à catalogação e preservação do património cultural mundial. Serviu de fonte da resposta à pergunta exemplo ilustrada na figura 5.

[05] **<http://www.flurry.com/about-flurry>**

Página de uma empresa de análise e estatísticas à cerca de aplicações móveis.



## Lista de Siglas e Acrónimos

ASP: *Active Server Pages*

CSS: *Cascading Style Sheets*

EF: *Entity Framework*

ETL: *Extract, Transform, Load*

HTML: *HyperText Markup Language*

ISO: *International Organization for Standardization*

LOB: *Locator Object*

MVW: *Model-View-Whatever*

NP: *Norma Portuguesa*

ORM: *Object-Relational Mapping*

RAD: *Rapid Application Development*

SQL: *Structured Query Language*

UML: *Unified Modeling Language*

**SQL: *Structured Query Language***

## **Anexos**

# I. Dados de Entrada/Especificações

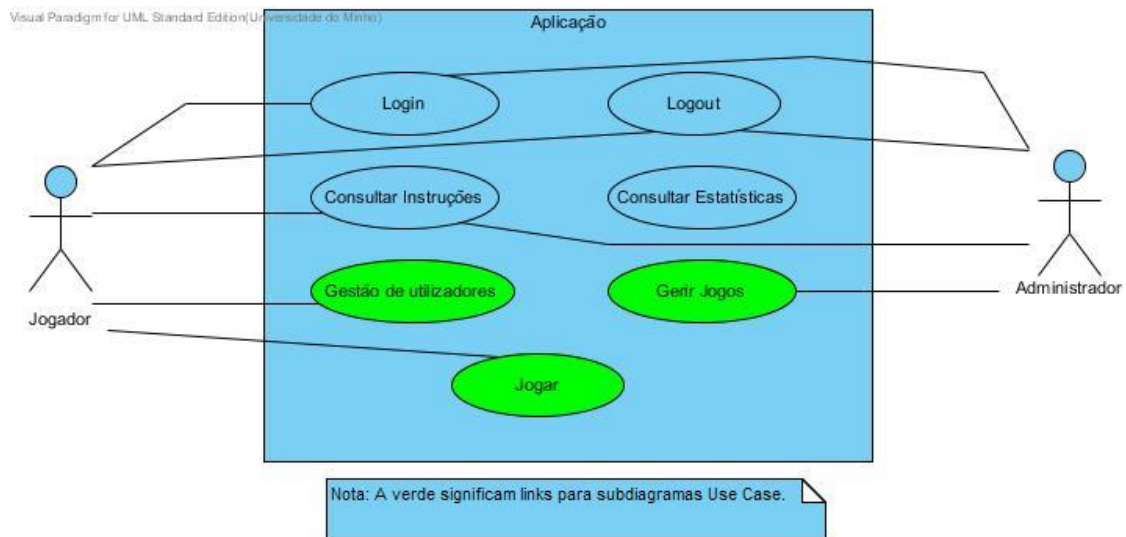
Dados de entrada	Especificações
<p>Enunciado Unidade Curricular.            Síntese das exigências do enunciado:                Manual de utilizador.                Seleção de tema e nível.                Volta ao Globo.                Perfil de Utilizador dinâmico.                Variar grau de dificuldade.                Questões áudio, vídeo e imagem.                Apresentar estatísticas.</p> <p>Trivial Pursuit- Instruções            Monopoly-Instruções  <a href="http://www.nationalgeographic.com">www.nationalgeographic.com</a>  <a href="http://pt.wikipedia.org/wiki/A_Volta_ao_Mundo_em_80_Dias">pt.wikipedia.org/wiki/A_Volta_ao_Mundo_em_80_Dias</a></p>	<p>Fundamentação:            Concepção do modelo de jogo:</p> <ul style="list-style-type: none"> <li>• Caracterização geral da aplicação a desenvolver, definindo de forma concisa os objectivos do jogo.</li> <li>• Definição das funcionalidades a implementar, que deverão ser listadas de forma detalhada, indo de encontro aos requisitos impostos no enunciado.</li> <li>• Definição do modelo de funcionamento do jogo/Elaboração da análise textual dos requisitos.              Depois de definidas as funcionalidades que se pretende implementar, pretende-se detalhar as regras de jogo. Deverá ter como output o manual de instruções do jogo.</li> <li>• Definição do tema. Prevendo-se elaborado o desenho da aplicação para que esta responda de forma independente do tema, deverá ainda na fase de fundamentação ser escolhido o tema.</li> </ul>
<p>Exercícios resolvidos slides da unidade curricular            Desenvolvimento de Sistemas de Software- 2013/2014</p>	<p>Especificação            Especificação detalhada da aplicação, definindo a análise de requisitos, agora em UML:</p> <ul style="list-style-type: none"> <li>• Modelo de domínio</li> <li>• Elaboração do modelo de Use Case              Diagrama de Use Case              Especificação textual dos Use Cases              Verificação e validação do modelo de Use Cases</li> <li>• Elaboração dos diagramas de sequência</li> <li>• Elaboração do Diagrama de classes</li> </ul> <p>Verificação e validação da modelação.</p>
<p>Análise de requisitos em UML  <a href="http://www.nationalgeographic.com">www.nationalgeographic.com</a></p>	<ul style="list-style-type: none"> <li>• Estudo da linguagem e ferramentas</li> <li>• Idealização da arquitetura do sistema</li> <li>• Idealizar plano de testes</li> <li>• Criar base de dados              Estudo aprofundado do tema              Elaboração das questões              Esquema conceptual da base de dados              Implementação e alimentação da base de dados</li> <li>• Implementação da camada de negócios</li> <li>• Implementação da interface gráfica</li> <li>• Ajuste dos valores das variáveis relativas tempo e dinheiro</li> <li>• Revisão das funcionalidades da aplicação</li> <li>• Verificação e validação da aplicação</li> </ul>

## II. Plano de Atividades

		3-9 Mar	10-16 Mar	17-23 Mar	24-30 Mar	31-6 Abr	7-13 Abr	14-20 Abr	21-27 Abr	28-4 Mai	5-11 Mai	12-18 Mai	19-25 Mai	26-mai
<b>Outputs</b>														
<b>Planeamento do Projeto</b>														
Especificações do projeto:														
Definição dos objetivos		•	•						•					
Definição das entradas de dados		•												
Definição das especificações														
Validação das especificações do projeto		•												
Controlo do Projeto														
Elaboração do Plano de trabalhos			•											
Controle de execução				•		•	•		•		•		•	
<b>Fase Fundamentação</b>														
Caracterização geral da aplicação a desenvolver		•	•											
Definição das funcionalidades a implementar		•	•	•										
Definição do modelo de funcionamento do jogo		•	•	•										
Elaboração de um mapa de navegação														
Elaboração da análise textual de requisitos														
Definição do tema			•											
Verificação e validação da fundamentação														
<b>Fase Especificação</b>														
Especificação detalhada da aplicação					•	•								
Modelo de domínio					•									
Elaboração do modelo de Use Case						•	•							
Diagrama de Use Case						•								
Especificação textual dos Use Cases							•							
Verificação e validação do modelo de Use Cases							•							
Elaboração dos diagramas de sequencia							•	•						
Elaboração do Diagrama de classes								•						
Verificação e validação da modelação				•				•						
<b>Implementação</b>														
Estudo da linguagem e ferramentas								•						
Idealização da arquitetura do sistema								•	•					
Idealizar plano de testes									•					
Criar base de dados														
Estudo aprofundado do tema								•	•					
Elaboração das questões								•	•					
Esquema conceptual da base de dados										•				
Implementação e alimentação da base de dados														
Implementação da camada de negócios										•	•	•		
Implementação da interface gráfica										•	•	•		
Ajuste dos valores das variáveis relativas tempo e dinheiro														
Revisão das funcionalidades da aplicação														•
Verificação e validação da aplicação														•
<b>Apresentações</b>														
Elaboração da Apresentação da fase de fundamentação				•										
Verificação e Validação Apresentação da fase de fundamentação					•									
Apresentação da fase de fundamentação					•									
Elaboração da Apresentação da fase de especificação									•				•	
Verificação e Validação da Apresentação da fase de especificação									•	•				
Apresentação da fase de especificação										•				
Elaboração da apresentação final													•	
Verificação e Validação da apresentação													•	
Apresentação final														
<b>Relatórios</b>														
Elaboração da Relatório da fase de fundamentação		•	•	•										
Verificação e Validação do Relatório da fase de fundamentação					•									
Elaboração da Apresentação da fase de especificação		•	•	•		•	•	•	•					
Verificação e Validação do Relatório da fase de especificação										•				
Elaboração do Relatório final		•	•	•		•	•	•	•		•	•	•	
Verificação e Validação do Relatório													•	

### III. Modelo de Use Cases

#### 1. Aplicação



#### 1.1 Login

Super Use Case			
Author	Filipa		
Date	1/Abr/2014 11:51:46		
Brief Description			
Preconditions			
Post-conditions	Utilizador autenticado		
Flow of Events		Actor Input	System Response
	1	Clica em Login	
	2		Pede para indicar nome de utilizador e password
	3	Indica nome de utilizador e password	
	4		Valida dados
	5		Informa que o utilizador está autenticado
Cenário de exceção [Dados inválidos](3)		Actor Input	System Response
	1		Informa que não há utilizador com credenciais correspondentes às inseridas
	2		Aborta processo

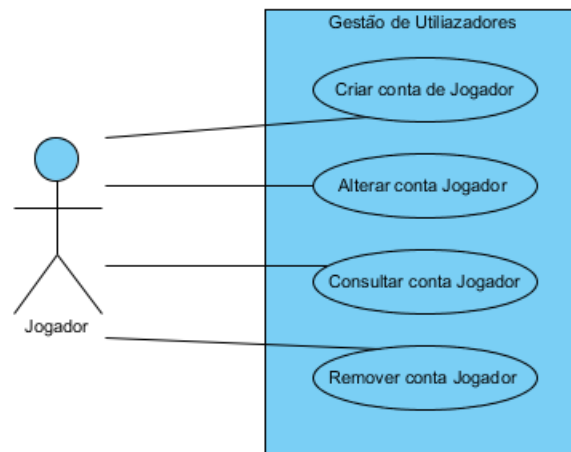
## 1.2 Logout

<b>Super Use Case</b>		
<b>Author</b>	Filipa	
<b>Date</b>	1/Abr/2014 11:56:01	
<b>Brief Description</b>		
<b>Preconditions</b>	Utilizador autenticado	
<b>Post-conditions</b>	Utilizador termina sessão	
<b>Flow of Events</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Clica em Sair
	2	Procede internamente ao termino da sessão

## 1.3 Consultar Instruções

<b>Super Use Case</b>		
<b>Author</b>	Filipa	
<b>Date</b>	1/Abr/2014 11:57:28	
<b>Brief Description</b>		
<b>Preconditions</b>	Utilizador autenticado	
<b>Post-conditions</b>		
<b>Flow of Events</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	Clica em Instruções
	2	Apresenta instruções

## 2. Gestão de Utilizadores



### 2.1 Criar Conta de Jogador

<b>Brief Description</b>			
<b>Preconditions</b>	True		
<b>Post-conditions</b>	Conta de Jogador Criada		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em Criar conta de Utilizador	
	2		Apresenta ficha de Jogador com: nome de utilizador; email; password; idade?; sexo?;
	3	Preenche ficha de Jogador	
	4	Submete ficha de Jogador	
	5		
	6		Valida ficha de Jogador
	7		Processa internamente ficha de Jogador
	8		Informa sucesso
<b>Cenário de exceção 1 [Dados inválido](5)</b>		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que os dados inseridos não são válidos.
	2		Aborta processo

## 2.2 Alterar Conta de Jogador

<b>Super Use Case</b>		
<b>Author</b>	Filipa	
<b>Date</b>	30/Mar/2014 13:48:45	
<b>Brief Description</b>		
<b>Preconditions</b>	Jogador autenticado	
<b>Post-conditions</b>	Conta de jogador alterada	
<b>Flow of Events</b>		<b>Actor Input</b>
	1	Clica em alterar conta
	2	
	3	Procede a alterações
	4	Submete ficha
	5	
	6	Confirma alterações
	7	
	8	
	9	
<b>Cenário de Excepção 1 [Jogador não confirma] (6)</b>		<b>Actor Input</b>
	1	Jogador não confirma
	2	
<b>Cenário de Excepção 2 [Dados Inválidos]</b>		<b>Actor Input</b>
	1	
	2	

## 2.3 Consultar Conta de Jogador

<b>Date</b>	30/Mar/2014 13:38:42	
<b>Brief Description</b>		
<b>Preconditions</b>	Jogador autenticado	
<b>Post-conditions</b>		
<b>Flow of Events</b>		<b>Actor Input</b>
	1	Clica em consultar conta de Jogador
	2	

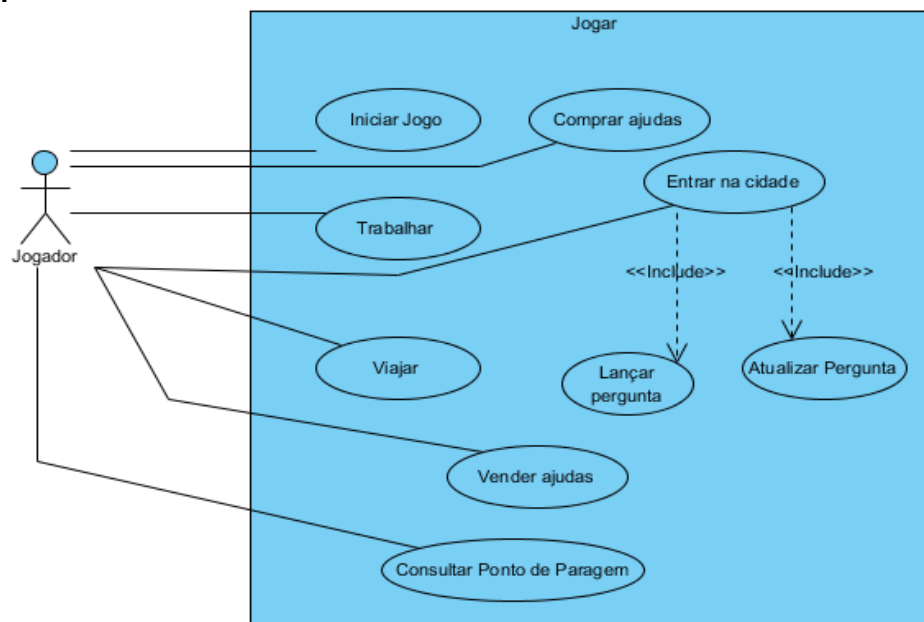


## 2.4 Remover Conta de Jogador

Super Use Case			
Author	Filipa		
Date	30/Mar/2014 13:40:05		
Brief Description			
Preconditions	Jogador autenticado		
Post-conditions	Jogador Removido		
Flow of Events		Actor Input	System Response
	1	Clica em Remover Conta	
	2		Pede confirmação de remoção
	3	Confirma Remoção	
	4		Processa internamente a remoção do jogador
	5		Notifica Sucesso
Cenário de Excepção 1 [Jogador não confirma](3)		Actor Input	System Response
	1	Não confirma remoção	
	2		Aborta processo

## 3. Jogar

4.



### 3.1 Iniciar Jogo

<b>Brief description</b>			
<b>Preconditions</b>	Jogador autenticado		
<b>Post-conditions</b>	Jogo iniciado		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em Novo Jogo	
	2		Pede para seleccionar tema
	3	Selecciona tema	
	4		Pede para seleccionar nível de dificuldade
	5		Lista níveis de dificuldade: fácil; médio; difícil;
	6	Selecciona nível de dificuldade	
	7		Apresenta mapa
	8		Pede para seleccionar ponto de partida
	9	Selecciona ponto de partida	
	10		Pede para indicar dia da semana e hora de partida
	11	Indica dia da semana e hora	
	12		Cria um novo jogo com os dados inseridos, convertendo o nível de dificuldade para uma escala de 1 a 10.
	13		Informa que o jogo começou e que o jogador está na porta da cidade

### 3.2 Trabalhar

<b>Preconditions</b>	Jogo iniciado		
<b>Post-conditions</b>	Tempo despendido e dinheiro amealhado		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em trabalhar	
	2		Apresenta empregos disponíveis e respetivo salário/hora
	3	Selecciona emprego	
	4		Pede para indicar o número de horas que pretende trabalhar
	5	Indica o número de horas que pretende trabalhar	
	6		Processa internamente a execução do trabalho (consome tempo, aumenta dinheiro, atualiza perfil)
	7	Aguarda a execução do trabalho	
	8		Informa do fim do serviço

### 3.3 Viajar

<b>Preconditions</b>	Jogo iniciado		
<b>Post-conditions</b>	Alcançado novo ponto de paragem		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em viajar	
	2		Destaca destinos possíveis
	3	Selecciona destino	
	4		Apresenta tabela, com meios de transporte, respetivo horário e preço
	5	Selecciona transporte	
	6		Pede confirmação da compra do bilhete
	7	Confirma compra do bilhete	
	8		Processa internamente compra do bilhete
	9	Aguarda pelo transporte (possivel ponto de extensão para trabalhar )	
	10		Informa da hora de embarque
	11		Processa internamente a viagem
	12		Informa da chegada ao destino
<b>Cenário de excepção 1</b> [Jogador não tem dinheiro](8)		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que o jogador não tem dinheiro para o bilhete
	2		Termina o processo de viagem
<b>Cenário alternativo 1</b> [Aluga meio de transporte](6)		<b>Actor Input</b>	<b>System Response</b>
	1	Jogador selecciona aluguer de ...	
	2		Regressa a 11

### 3.4 Vender Ajuda

Super Use Case			
Author	Filipa		
Date	1/Abr/2014 11:38:53		
Brief Description			
Preconditions	Jogo iniciado e é ponto de compra de ajudas		
Post-conditions	Ajuda vendida		
Flow of Events		Actor Input	System Response
	1	Clica em vender ajudas	
	2		Apresenta lista de ajudas do jogador e respetivo preço de venda
	3	Selecciona ajuda	
	4		Processa internamente a venda da ajuda
	5		Informa o jogador que a venda foi efectuada com sucesso
Cenário de Excepção [Jogador não tem ajudas](2)		Actor Input	System Response
	1		Informa que o jogador não tem ajudas para vender

### 3.5 Comprar Ajuda

Super Use Case			
Author	Filipa		
Date	1/Abr/2014 11:33:32		
Brief Description			
Preconditions	Jogo iniciado e é ponto de compra de ajudas		
Post-conditions	Ajuda comprada		
Flow of Events		Actor Input	System Response
	1	Clica em comprar ajudas	
	2		Lista ajudas disponiveis e respetivo preço
	3	Selecciona ajuda	
	4		Procede internamente à compra da ajuda
	5		Informa ao jogador que a ajuda foi adquirida
Cenário de Excepção [Jogador não tem dinheiro](4)		Actor Input	System Response
	1		Informa que o jogador não tem dinheiro para a ajuda pretendida
	2		Aborta o processo

### 3.6 Consultar Ponto de Paragem

<b>Super Use Case</b>			
<b>Author</b>	Filipa		
<b>Date</b>	23/Abr/2014 15:28:36		
<b>Brief Description</b>			
<b>Preconditions</b>	Jogo iniciado		
<b>Post-conditions</b>			
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Selecciona ponto de paragem e clica em consultar	
	2		Apresenta trabalhos disponíveis e respectiva remuneração
	3		Apresenta ajudas, se disponíveis, e respectivo preço
	4		Apresenta horários das ligações possíveis (com dia e hora de partida, duração e preço)

### 3.7 Entrar na cidade

Super Use Case			
Author	Filipa		
Date	30/Mar/2014 14:41:29		
Brief Description			
Preconditions	Jogo iniciado		
Post-conditions	Ponto visitado		
Flow of Events		Actor Input	System Response
	1	Jogador selecciona entrar cidade	
	2		<<Include>> Lançar pergunta
	3		Acelera passagem do tempo
	4	Selecciona opção de resposta	
	5		Valida resposta
	6		Marca ponto como visitado
	7		Atualiza estatísticas do jogador, incrementando o número de respostas certas
	8		<<Include>> Atualizar pergunta
	9		Adiciona à conta do jogador o dinheiro da pergunta
	10		Informa que o jogador tem a chave da cidade

<b>Cenário Alternativo 1 [Utiliza Ajuda](4)</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em Utilizar Ajuda	
	2		Apresenta ajudas adquiridas
	3	Selecciona Ajuda	
	4		Apresenta resultado da ajuda
	5		Processa uso da ajuda
<b>Cenário Alternativo 2 [Jogador não tem ajudas](4.2)</b>	6		Regressa a 4
		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que o jogador não tem ajudas
<b>Cenário Excepção 1 [Resposta errada](6)</b>	2		Regressa a 6
		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que a resposta dada está incorreta
	2		Incrementa o número de respostas erradas na cidade
	3		<<Include>> Atualizar pergunta
	4		Calcula o tempo de retenção do jogador, exponencial ao número de respostas erradas
	5		Informa o tempo de retenção do jogador
	6		Liberta o jogador

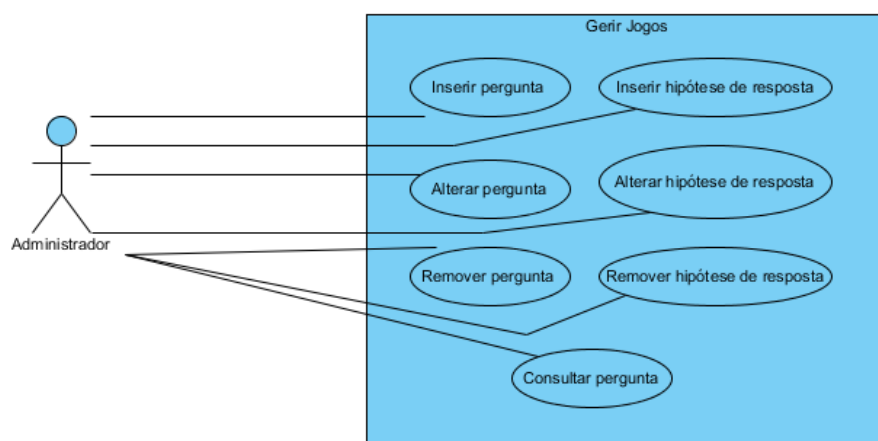
### 3.8 Lançar Pergunta

Super Use Case		
Author	Filipa	
Date	23/Abr/2014 14:54:48	
Brief Description		
Preconditions	Jogo a decorrer, com nível n de dificuldade (de 1 a 10)	
Post-conditions	Pergunta	
Flow of Events		

### 3.9 Atualizar Pergunta

Super Use Case		
Author	Filipa	
Date	23/Abr/2014 15:15:23	
Brief Description		
Preconditions	Hipótese de resposta seleccionada	
Post-conditions	Pergunta atualizada	
Flow of Events		

## 4. Gerir Jogo



### 4.1 Inserir Pergunta

Author	Filipa		
Date	17/Abr/2014 19:19:29		
Brief Description			
Preconditions	Utilizador tem permissão		
Post-conditions	Pergunta inserida		
Flow of Events		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em inserir pergunta	
	2		Lista temas
	3		Pede para seleccionar o tema
	4	Selecciona tema	
	5		Lista graus de dificuldade
	6		Pede para seleccionar grau de dificuldade
	7	Selecciona grau de dificuldade	
	8		Pede para inserir descrição
	9	Insera descrição	
	10		Lista Zonas
	11		Pede para seleccionar zona
	12	Selecciona Zona	
	13		Pergunta se quer inserir vídeo, imagem ou som
	14	Diz que sim	
	15		Pede para inserir vídeo, imagem ou som
	16	Insera anexo	
	17		Valida pergunta
	18		Processa internamente a criação da pergunta
19		Informa que a pergunta foi criada com sucesso	
Cenário alternativo 1 [Utilizador não selecciona anexo](14)		<b>Actor Input</b>	<b>System Response</b>
	1	Diz que não	
	2		Regressa a 17
Cenário de excepção 2 [Pergunta inválida](17)		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que determinado campo não é válido pergunta
	2		Aborta processo

## 4.2 Alterar Pergunta

<b>Date</b>	17/Abr/2014 19:54:26	
<b>Brief Description</b>		
<b>Preconditions</b>	Utilizador tem permissão	
<b>Post-conditions</b>	Pergunta alterada	
<b>Flow of Events</b>		<b>Actor Input</b>
	1	Clica em alterar pergunta
	2	
	3	Selecciona opção de filtragem
	4	
	5	Selecciona pergunta
	6	
	7	Procede a alterações
	8	Submete alterações
	9	
	10	Confirma alterações
	11	
	12	
	13	
<b>Cenário de excepção 1 [Utilizador não confirma](9)</b>		<b>System Response</b>
	1	
<b>Cenário de excepção 2 [alterações inválidas](11)</b>	2	Aborta processo
		<b>System Response</b>
	1	Informa que as alterações efectuadas não são válidas
	2	Aborta processo



### 4.3 Remover Pergunta

<b>Date</b>	17/Abr/2014 20:00:02	
<b>Brief Description</b>		
<b>Preconditions</b>	Utilizador tem permissão	
<b>Post-conditions</b>	Pergunta removida	
<b>Flow of Events</b>		<b>Actor Input</b>
	1	Clica em remover pergunta
	2	
	3	Selecciona opção de filtragem
	4	
	5	Selecciona pergunta
	6	
	7	Confirma remoção
	8	
	9	
		<b>System Response</b>
		Pede para seleccionar opção de filtragem (zona, código...)
		Lista perguntas
		Pede confirmação de remoção
		Procede internamente à remoção da pergunta
		Informa sucesso
<b>Cenário de excepção 1 [Utilizador não confirma](7)</b>		<b>Actor Input</b>
	1	Não confirma remoção
	2	
		<b>System Response</b>
		Aborta o processo
<b>Cenário de excepção 2 [Não é possível remover a pergunta](8)</b>		<b>Actor Input</b>
	1	
	2	
		<b>System Response</b>
		Informa que, por haver dependências, não é possível remover a pergunta
		Aborta o processo

### 4.4 Consultar Pergunta

<b>Author</b>	Filipa	
<b>Date</b>	17/Abr/2014 20:03:30	
<b>Brief Description</b>		
<b>Preconditions</b>	Utilizador tem permissão	
<b>Post-conditions</b>		
<b>Flow of Events</b>		<b>Actor Input</b>
	1	Clica em consultar perguntas
	2	
	3	Selecciona opção de filtragem
	4	
	5	Selecciona pergunta
	6	
		<b>System Response</b>
		Pede para seleccionar opção de filtragem (zona, código...)
		Lista perguntas com respectivo código, zona e grau de dificuldade
		Apresenta pergunta e respectivas hipóteses de resposta

## 4.5 Inserir Hipótese de Resposta

<b>Preconditions</b>	Utilizador tem permissao		
<b>Post-conditions</b>	Hipótese de resposta inserida		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	Clica em inserir hipótese de resposta	
	2		Pede para seleccionar opção de filtragem (zona, código...)
	3		Lista perguntas
	4	Selecciona pergunta	
	5		Pede para seleccionar grau (0,1,2)
	6	Selecciona grau	
	7		Pede para inserir descrição
	8	Inserir descrição	
	9		Pergunta se quer inserir imagem
	10	Diz que sim	
	11		Pede para inserir imagem
	12	Selecciona imagem	
	13		Valida hipótese de resposta
	14		Procede internamente à inserção da nova hipótese de resposta
	15		Informa sucesso
<b>Cenário de excepção 1[Dado inválido]</b>		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que determinado campo da hipótese de resposta não é valido
	2		Aborta processo

## 4.6 Alterar Hipótese de Resposta

<u>Preconditions</u>	Utilizador tem permissão		
<u>Post-conditions</u>	Hipótese alterada		
<u>Flow of Events</u>		<b>Actor Input</b>	<b>System Response</b>
	1	<u>Clica</u> em alterar hipótese de resposta	
	2		Pede para seleccionar opção de filtragem (zona, código...)
	3		Pede para seleccionar pergunta
	4	Selecciona pergunta	
	5		Lista hipóteses
	6		Pede para seleccionar hipótese
	7	Selecciona hipótese	
	8		Apresenta <u>hi</u> ótese
	9	Procede a alterações	
	10	Submete alterações	
	11		Pede confirmação
	12	Confirma alterações	
	13		Valida alterações
	14		Procede internamente à alteração da hipótese de resposta

Cenário de excepção 1 [Utilizador não confirma](12)		<b>Actor Input</b>	<b>System Response</b>
	1	Não confirma alterações	
	2		Aborta processo

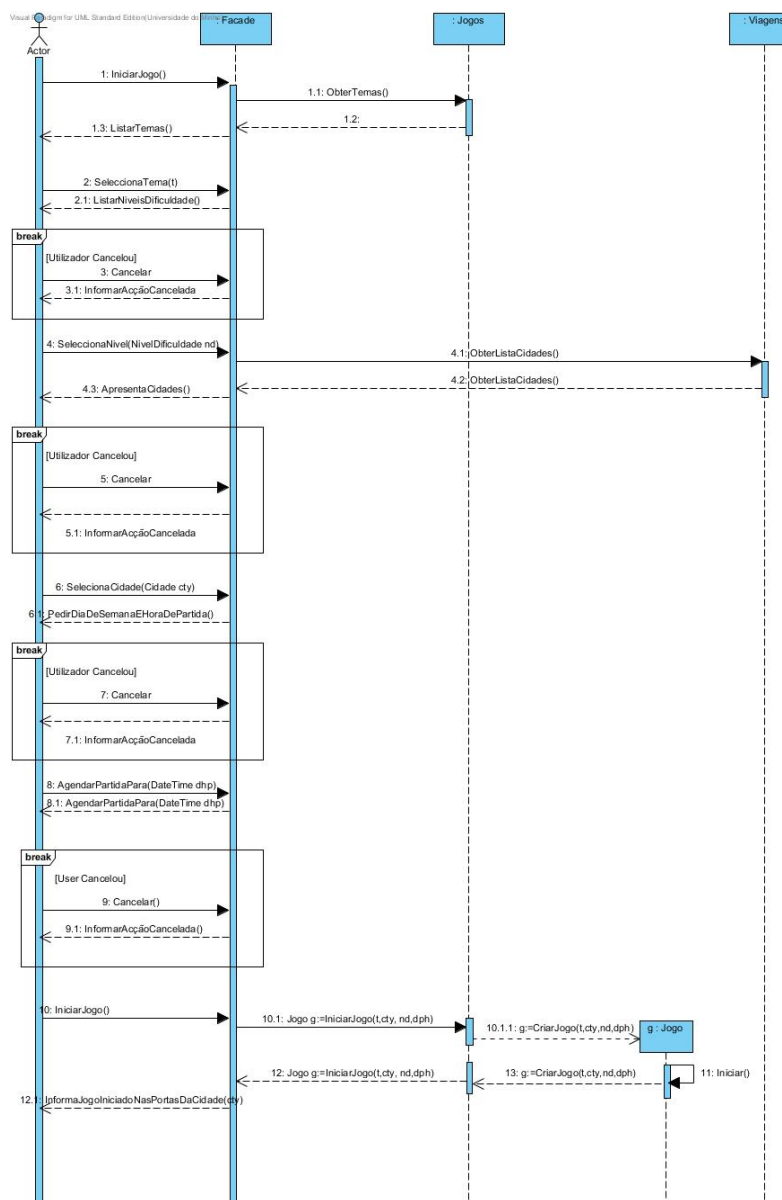
<u>Cenário de excepção 2</u> [Alterações inválidas](13)		<b>Actor Input</b>	<b>System Response</b>
	1		Informa que as alterações submetidas não são válidas
	2		Aborta processo

## 4.7 Remove Hipótese de Resposta

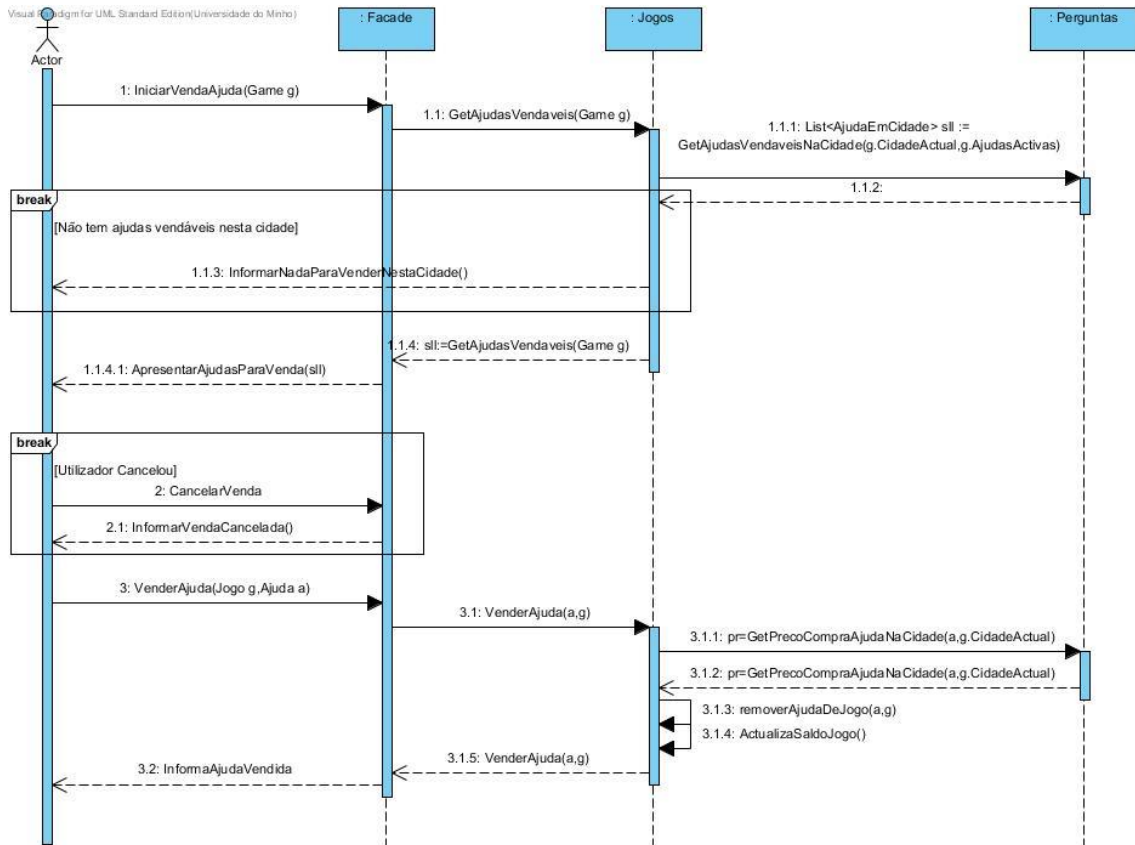
Date	17/Abr/2014 20:26:12	
Brief Description		
Preconditions	Utilizador tem permissão	
Post-conditions	Hipótese de resposta removida	
Flow of Events		<b>Actor Input</b>
	1	Clica em remover hipótese de resposta
	2	
	3	Selecciona opção de filtragem
	4	
	5	Selecciona pergunta
	6	
	7	Selecciona hipótese de resposta
	8	
	9	Confirma remoção
	10	
	11	
Cenário de excepção 1 [Utilizador não confirma](9)		<b>System Response</b>
	1	Utilizador não confirma
	2	
		Aborta processo

# IV. Diagramas de Sequencia

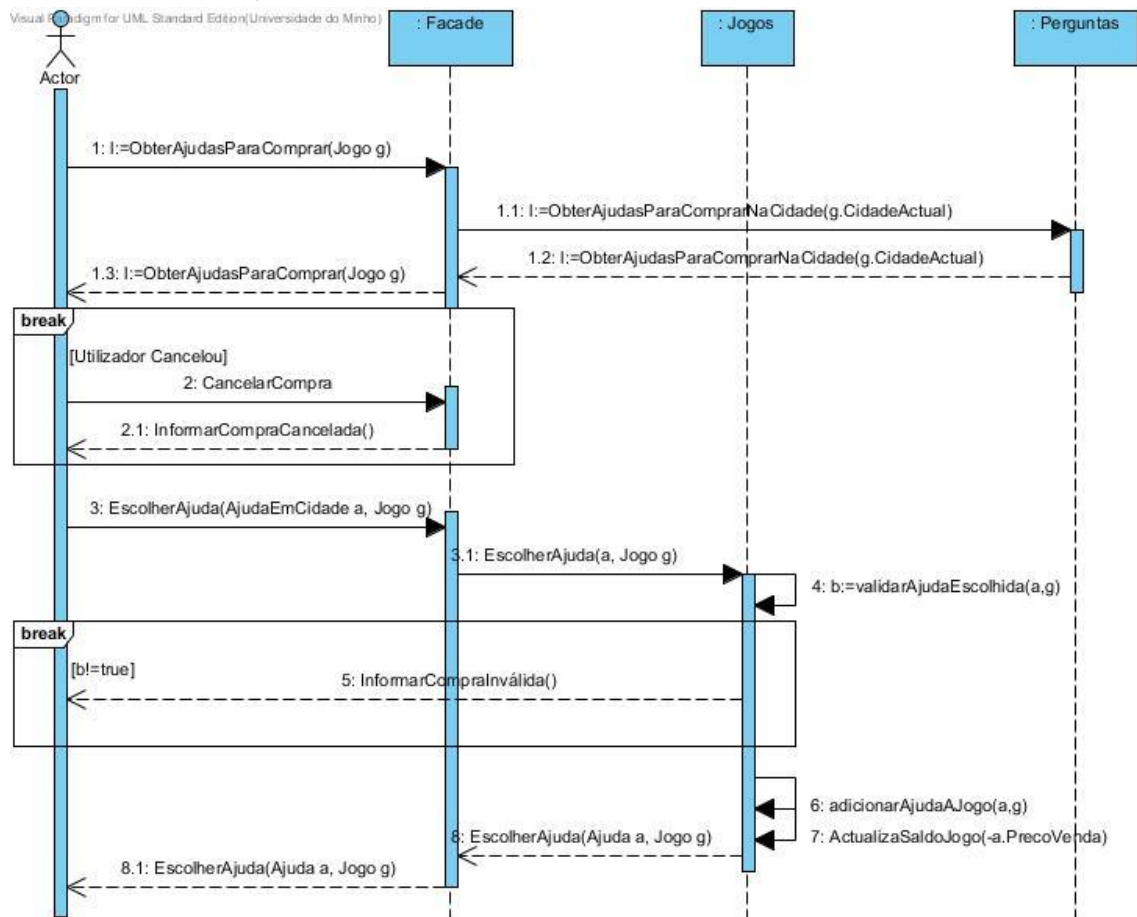
## 1. Start Game



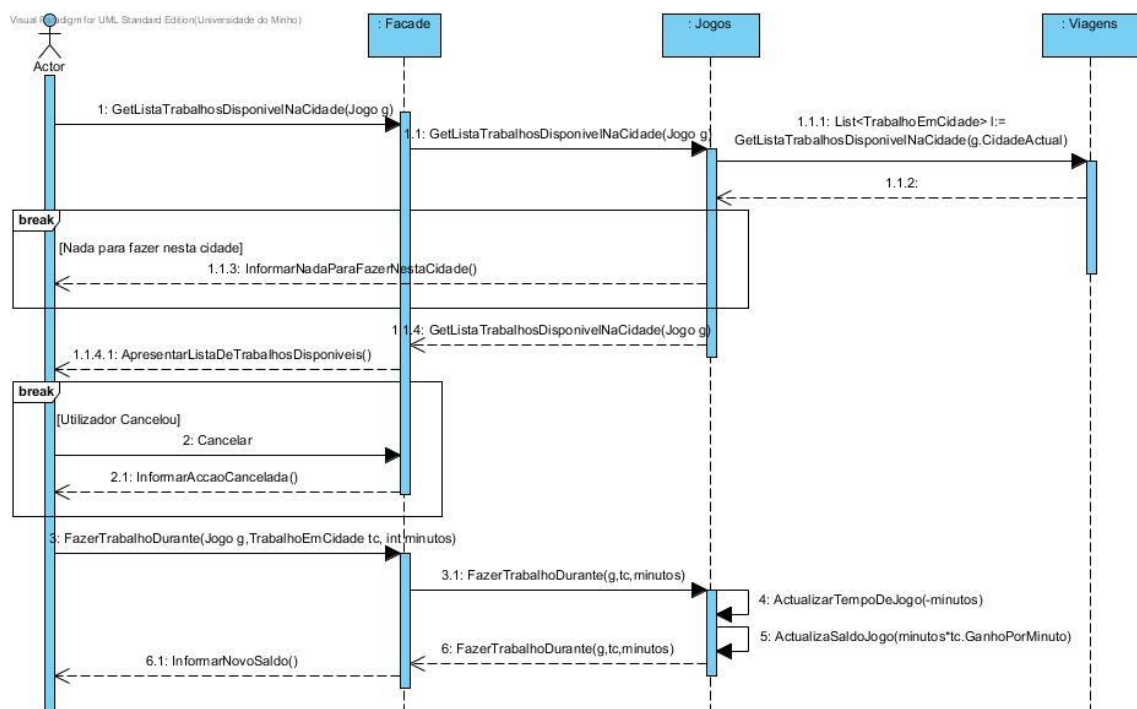
## 2. Comprar Ajuda



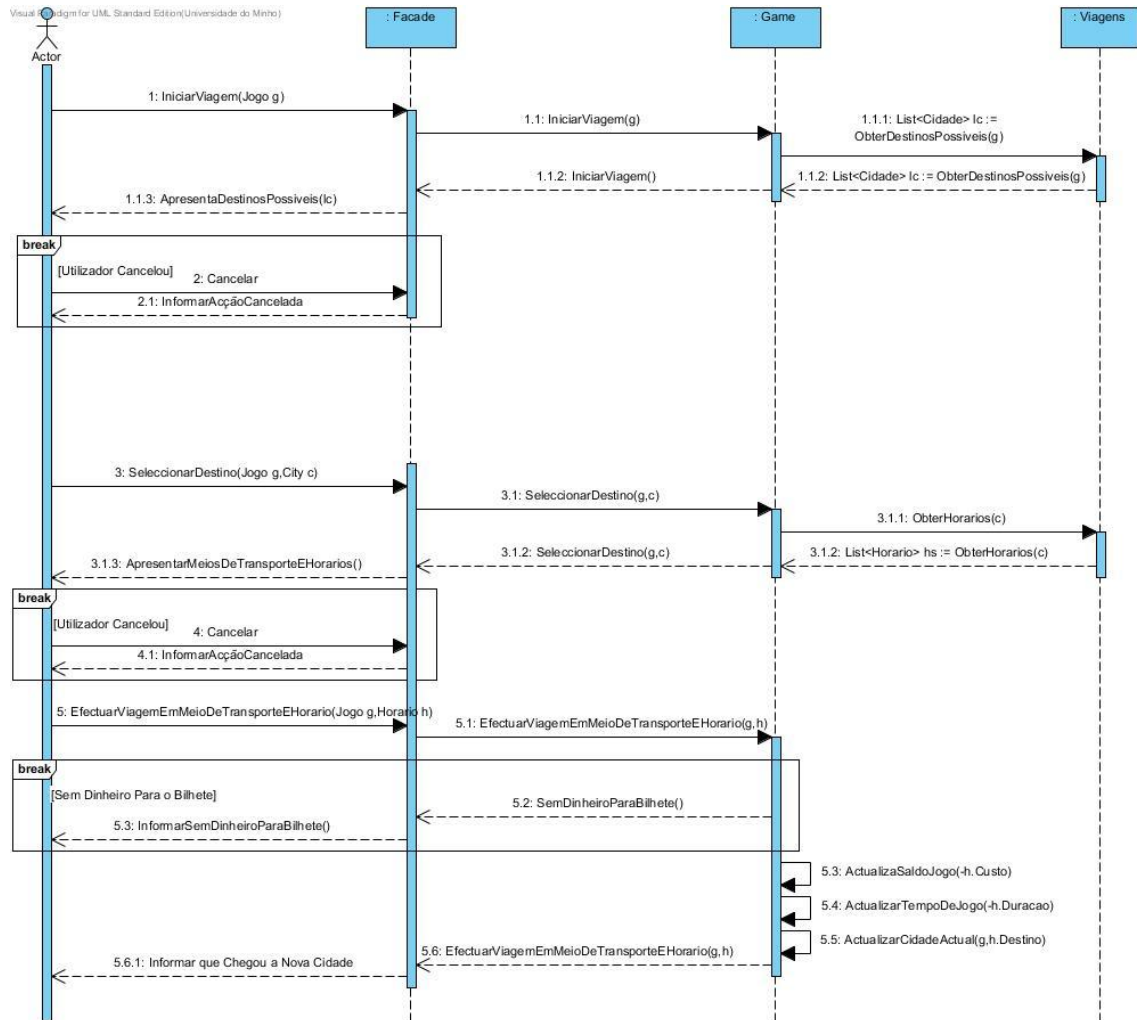
### 3. Vender Ajuda



### 4. Executar Trabalho

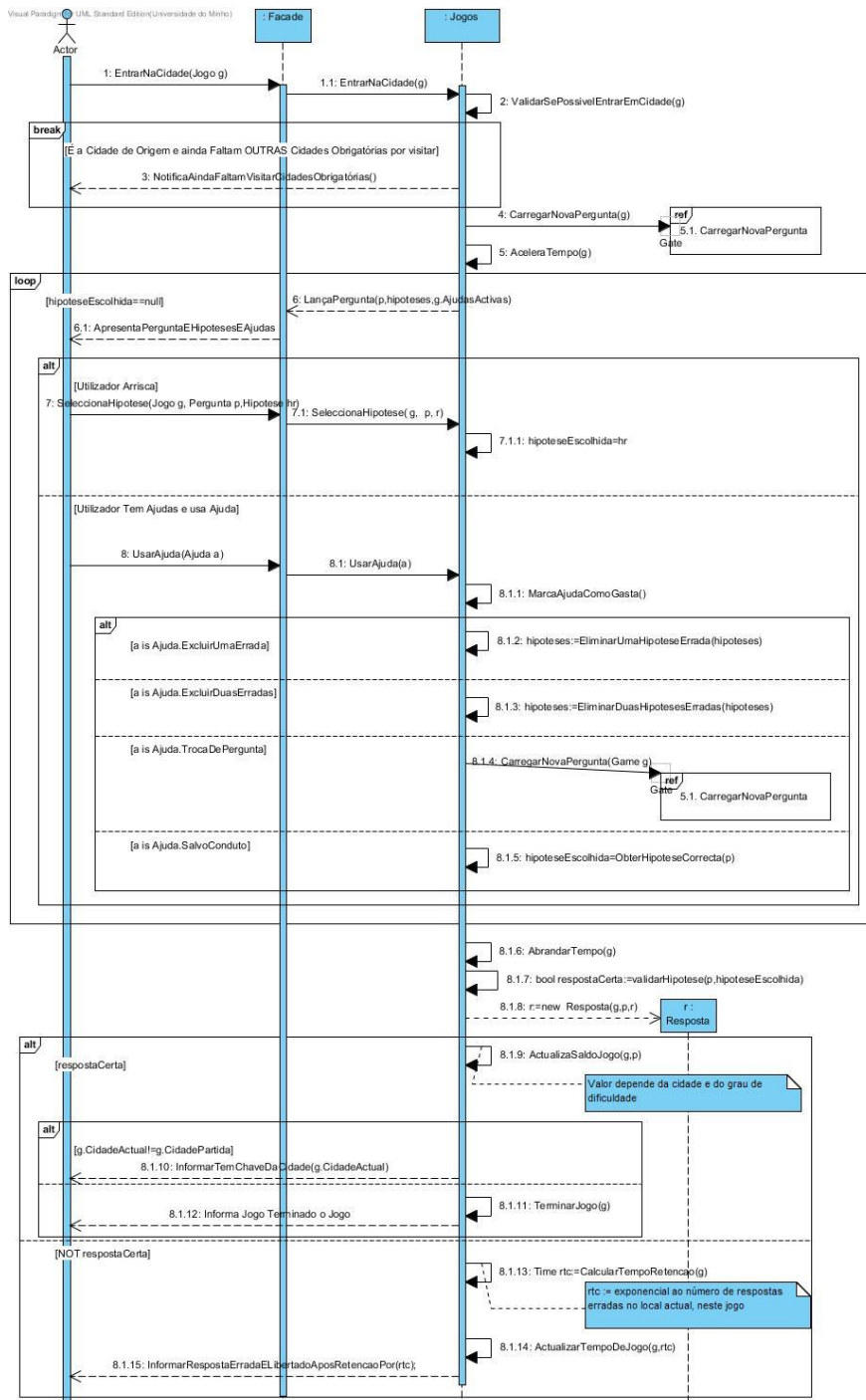


## 5. Viajar

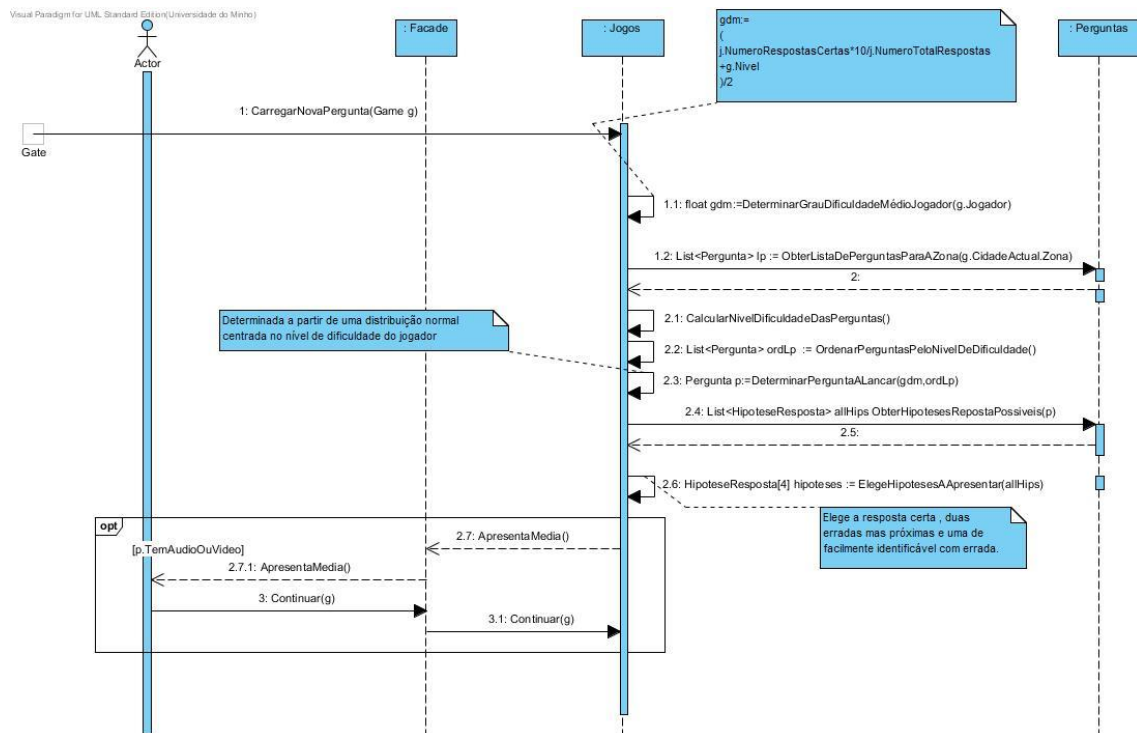




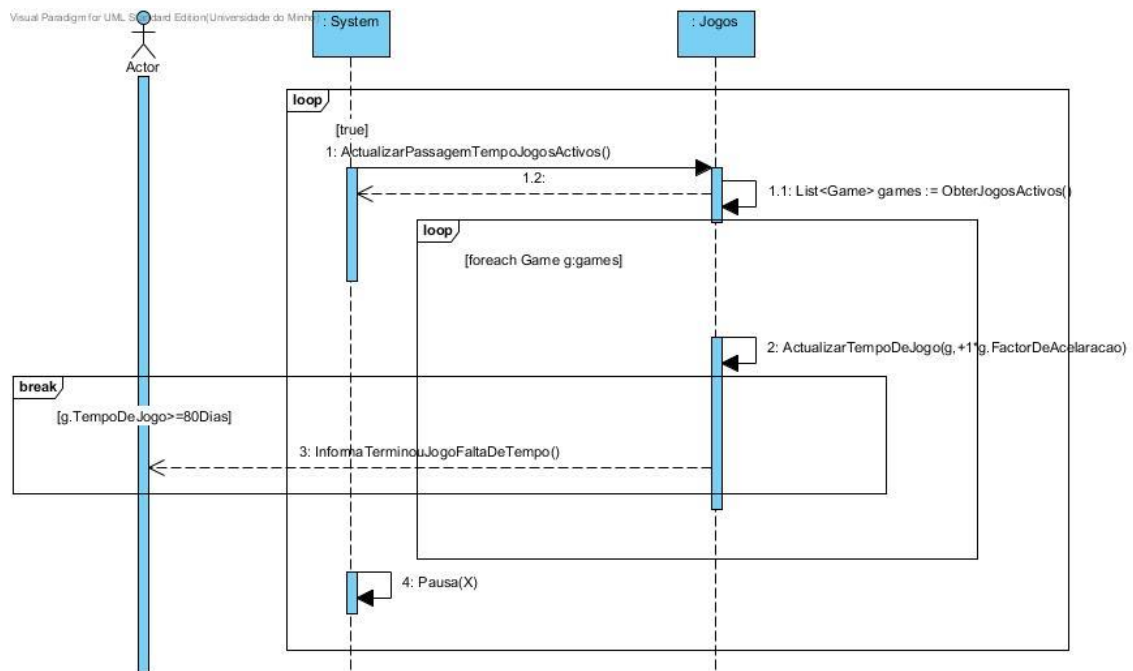
## 6. Entrar na cidade



## 6.1 Carregar Nova Pergunta



## 7. Passagem do Tempo



## V. Dicionário de Dados

TipoDados	TipoDados Base	Descrição
<b>Identifier</b>	Int	Inteiro com auto incremento de base 1
<b>Name</b>	nvarchar(64)	Nome
<b>Description</b>	nvarchar(1024)	Descrição
<b>Time</b>	DateTime	Hora
<b>TimeSpan</b>	long	Duração
<b>Url</b>	varchar(2048)	Url
<b>Flag</b>	char(1)	<i>Flag</i>
<b>Coordinate</b>	Decimal(20,6)	Coordenada Geográfica
<b>DifficultyLevel</b>	<b>Flag</b>	Nível de dificuldade: E-Fácil; M-Médio H-Difícil
<b>Code</b>	varchar(128)	Campo do tipo Código

# 1. Tabela Schedule

Tabela: Schedule				
Descrição: Tabela de Horários				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↪ScheduleId	Identifier	<input type="checkbox"/>	Primary Key	
Description	Description	<input type="checkbox"/>	Descrição do horário	
↪ModeOfTavelId	Identifier	<input type="checkbox"/>	Ligação a modo de transporte	ModeOfTavel
↪RouteId	Identifier	<input type="checkbox"/>	Ligação a Route.	Route
Departure	Time	<input type="checkbox"/>	Hora de Partida	
Duration	TimeSpan	<input type="checkbox"/>	Duração da viagem	
Price	Money	<input type="checkbox"/>	Custo da Viagem	
LeavesMonday	Bit	<input type="checkbox"/>	Sai à Segunda	
LeavesTuesday	Bit	<input type="checkbox"/>	Sai à Terça	
LeavesWednesday	Bit	<input type="checkbox"/>	Sai à Quarta	
LeavesThursday	Bit	<input type="checkbox"/>	Sai à Quinta	
LeavesFriday	Bit	<input type="checkbox"/>	Sai à Sexta	
LeavesSaturday	Bit	<input type="checkbox"/>	Sai à Sábado	
LeavesSunday	Bit	<input type="checkbox"/>	Sai ao Domingo	

## 2. Tabela LifeLine

Tabela: Lifeline				
Descrição: Tabela de Ajudas				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
<b>LifeLineId</b>	<b>Identifier</b>		<b>Primary Key</b>	
Name	Name	<input type="checkbox"/>	Nome da ajuda	
Description	Description	<input type="checkbox"/>	Descrição da Ajuda	
ImageUrl	Url	<input type="checkbox"/>	Imagem/Ícone Associado a Ajuda	
BackgroundImageUrl	Url	<input type="checkbox"/>	Imagem de fundo associada a ajuda	
BackgroundSoundUrl	Url	<input checked="" type="checkbox"/>	Som de fundo associado a ajuda	

### 3. Tabela GameLifeLine

Tabela: GameLifeLine				
Descrição: Tabela de ajudas associadas a um Jogo				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ CityLifeLineId	Identifier	<input type="checkbox"/>	Primary Key	
→ GameId	Identifier	<input type="checkbox"/>	Ligação a Jogo	Game
→ LifelineId	Identifier	<input type="checkbox"/>	Ligação a Ajuda	Lifeline
StatusCode	Flag	<input type="checkbox"/>	Estado da ajuda	Ativa ou Inativa(Gasta/Vendida)

## 4. Tabela CityLifeLine

Tabela: CityLifeLine				
Descrição: Tabela de ajudas disponíveis numa cidade				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ CityLifeLineId	Identifier	<input type="checkbox"/>	Primary Key	
→ CityId	Identifier	<input type="checkbox"/>	Ligação a Cidade	City
→ LifelineId	Identifier	<input type="checkbox"/>	Ligação a Ajuda	Lifeline
SellPrice	Money	<input checked="" type="checkbox"/>	Preço de Venda	Se nulo, as lojas desta cidade não vendem esta ajuda
BuyPrice	Money	<input checked="" type="checkbox"/>	Preço de Compra	Se nulo, as lojas desta cidade não compram esta ajuda

## 5. Tabela WorldRegion

Tabela: WorldRegion				
Descrição: Tabela de Zonas do Mundo				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ <b>WorldRegionId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
Name	Name	<input type="checkbox"/>	Nome da Zona	



## 6. Tabela City

Tabela: City				
Descrição: Tabela de Cidades				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
<b>↔ CityId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
Name	Name	<input type="checkbox"/>	Nome da Cidade	
<b>→ WorldRegionId</b>	<i>Identifier</i>	<input type="checkbox"/>	<i>Ligação a Zona do Mundo</i>	<i>WorldRegion</i>
BackgroundImageUrl	Url	<input type="checkbox"/>	Imagem de fundo associada a cidade	
BackgroundSoundUrl	Url	<input checked="" type="checkbox"/>	Som de fundo associado a cidade	
Latitude	Coordinate	<input type="checkbox"/>	Latitude da Cidade	
Longitude	Coordinate	<input type="checkbox"/>	Longitude da Cidade	
IsMandatory	Bit	<input type="checkbox"/>	Cidade é Ponto de passagem obrigatório	

## 7. Tabela Route

Tabela: Route				
Descrição: Tabela de Rotas entre cidades				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ <b>RouteId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
→ <i>FromCityId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Ligação a Cidade de Origem</i>	<i>City</i>
→ <i>ToCityId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Ligação a Cidade de Destino</i>	<i>City</i>

## 8. Tabela ModeOfTravel

Tabela: ModeOfTravel				
Descrição: Tabela de Modos de Transporte				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ ModeOfTravelId	Identifier	<input type="checkbox"/>	Primary Key	
Name	Name	<input type="checkbox"/>	Nome do modo de transporte	
Description	Description	<input type="checkbox"/>	Descrição do modo de transporte	
ImageUrl	Url	<input type="checkbox"/>	Imagem/Ícone Associado a Ajuda	
BackgroundImageUrl	Url	<input type="checkbox"/>	Imagem de fundo associada a ajuda	
BackgroundSoundUrl	Url	<input checked="" type="checkbox"/>	Som de fundo associado a ajuda	

## 9. Tabela Labour

Tabela: Labour				
Descrição: Tabela de trabalhos				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
<b>↔ LabourId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
Name	Name	<input type="checkbox"/>	Nome do Trabalho	
Description	Description	<input type="checkbox"/>	Descrição do Trabalho	
ImageUrl	Url	<input type="checkbox"/>	Imagem/Ícone Associado a Trabalho	
BackgroundImageUrl	Url	<input type="checkbox"/>	Imagem de fundo associada a trabalho	
BackgroundSoundUrl	Url	<input checked="" type="checkbox"/>	Som de fundo associado a trabalho	

## 10. Tabela CityLabour

Tabela: CityLabour				
Descrição: Tabela de trabalhos disponíveis numa cidade				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ CityLabourId	Identifier	<input type="checkbox"/>	Primary Key	
→ CityId	Identifier	<input type="checkbox"/>	Ligação a Cidade	City
→ LabourId	Identifier	<input type="checkbox"/>	Ligação a Trabalho	Labour
PayPerMinute	Money	<input type="checkbox"/>	Valor pago por minuto	
MinimumDuration	Duration	<input type="checkbox"/>	Tempo mínimo de execução do trabalho	

## 11. Tabela GameLabour

Tabela: GameLabour				
Descrição: Tabela de trabalhos realizados num jogo				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ GameLabourId	Identifier	<input type="checkbox"/>	Primary Key	
→ GameId	Identifier	<input type="checkbox"/>	Ligação a Jogo	Game
→ LabourId	Identifier	<input type="checkbox"/>	Ligação a Trabalho	Labour
Duration	TimeSpan	<input type="checkbox"/>	Tempo de trabalho	
Earnings	Money	<input type="checkbox"/>	Valor Ganho	

## 12. Tabela Trip

Tabela: Trip				
Descrição: Tabela de viagens realizadas				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ TripId	Identifier	<input type="checkbox"/>	Primary Key	
→ GameId	Identifier	<input type="checkbox"/>	Ligação a Jogo	Game
→ ScheduleId	Identifier	<input type="checkbox"/>	Ligação a Horário	Schedule

## 13. Tabela Game


Tabela: Game				
Descrição: Tabela de Jogos				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
<b>↔ Gameld</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
↗Userld	Identifier	<input type="checkbox"/>	Jogador	User
↗QuestionCategoryld	Identifier	<input type="checkbox"/>	Tema do jogo	QuestionCategory
StartAt	DateTime	<input type="checkbox"/>	Data/Hora de Início	
DifficultyLevel	DifficultyLevel	<input type="checkbox"/>	Nível de dificuldade	
↗StartCityld	Identifier	<input type="checkbox"/>	Cidade de origem	City
Balance	Money	<input type="checkbox"/>	Saldo do jogo	
ElapsedTime	TimeSpan	<input type="checkbox"/>	Tempo decorrido	
RegistrationDate	DateTime	<input type="checkbox"/>	Data de registo do jogo	
↗CurrentCityld	Identifier	<input type="checkbox"/>	Cidade actual	City
GameStatusCode	Flag	<input type="checkbox"/>	Estado do Jogo	S-Por Iniciar/Parado R-Em Curso F-Em execução acelerada T-Terminado



## 14. Tabela Question

Tabela: Question				
Descrição: Tabela de Perguntas				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ <b>QuestionId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
QuestionText	Description	<input type="checkbox"/>	Descrição da pergunta	
BaseDifficultyLevel	DifficultyLevel	<input type="checkbox"/>	Nível de dificuldade predefinido	
↔ <i>QuestionCategoryId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Tema do jogo</i>	<i>QuestionCategory</i>
↔ <i>WorldRegionId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Zona onde a pergunta se aplica</i>	<i>WorldRegion</i>
MediaType	Flag	<input checked="" type="checkbox"/>	Tipo de Media a apresentar	A-Audio V-Video
MediaUrl	Url	<input checked="" type="checkbox"/>	Media associado a Pergunta	
BackgroundImageUrl	Url	<input checked="" type="checkbox"/>	Imagem Associado a Pergunta	
Latitude	Coordinate	<input checked="" type="checkbox"/>	Latitude da Pergunta	
Longitude	Coordinate	<input checked="" type="checkbox"/>	Longitude da Pergunta	
RegistrationDate	DateTime	<input type="checkbox"/>	Data de registo da Pergunta	

## 15. Tabela QuestionCategory

Tabela: QuestionCategory				
Descrição: Tabela de Temas de Perguntas				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
 QuestionCategoryId	Identifier	<input type="checkbox"/>	Primary Key	
Name	Name	<input type="checkbox"/>	Nome do Tema	
ImageUrl	Url	<input type="checkbox"/>	Imagem associada ao tema	

## 16. Tabela AnswerChoice

Tabela: AnswerChoice				
Descrição: Tabela de Hipóteses de Resposta				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ AnswerChoiceId	Identifier	<input type="checkbox"/>	Primary Key	
→ QuestionId	Identifier	<input type="checkbox"/>	Pergunta	Question
AnswerChoiceText	Name	<input type="checkbox"/>	Hipótese de Resposta	
ChoiceTypeCode	Flag	<input type="checkbox"/>	Tipo de Hipótese	0-Correcta 1-Aproximada 2-Distante
RegistrationDate	DateTime	<input type="checkbox"/>	Data de registo da resposta	

## 17. Tabela GameAnswer

Tabela: GameAnswer				
Descrição: Tabela de Respostas dadas num jogo				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
↔ <b>GameAnswerId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
→ <i>GameId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Jogo associado</i>	<i>Game</i>
→ <i>AnswerChoiceId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Hipótese Escolhida</i>	<i>AnswerChoice</i>
→ <i>CityId</i>	<i>Identifier</i>	<input type="checkbox"/>	<i>Cidade onde foi lançada a pergunta</i>	<i>City</i>

## 18. Tabela User

Tabela: User				
Descrição: Tabela de Utilizadores/Jogadores				
Campo	TipoDados	Nulos	Descrição	Liga a/Restrições
<b>↔ UserId</b>	<b>Identifier</b>	<input type="checkbox"/>	<b>Primary Key</b>	
Name	Name	<input type="checkbox"/>	Nome do utilizador	
Email	Code	<input type="checkbox"/>	Email/Login	
IsAdministrator	Bit	<input type="checkbox"/>	Administrador	
Password	Code	<input type="checkbox"/>	Nome do utilizador	
BirthDate	DateTime	<input type="checkbox"/>	Data de nascimento	
RegistrationDate	DateTime	<input type="checkbox"/>	Data de registo	