



## **Aplicação de Mecânicas de Card Collecting em Jogos Sérios para o Turismo**

**VASCO REIS MOREIRA DE CASTRO TEIXEIRA**

Julho de 2019

# **Aplicação de Mecânicas de *Card Collecting* em Jogos Sérios para o Turismo**

**Vasco Reis Moreira de Castro Teixeira**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Sistemas Gráficos e Multimédia**

**Orientador: Carlos Vaz de Carvalho**

Porto, julho de 2019





## Agradecimentos

Deixo um agradecimento a todos os que me ajudaram no desenvolvimento do projeto: ao professor Carlos Vaz de Carvalho pela dedicação e disponibilidade prestada durante todo o projeto, à minha família, pela completo e incondicional apoio e aos meus colegas.

## Abstract

Given the sustained growth in international tourism, there's a consequent increase in the number of participants in cultural tourism activities. Traditionally, such activities don't present a solid well-defined structure, being mostly circumstance driven. Therefore, there's a need of a tool that can provide a better structure and planning of the tourist's activities, maximizing the effectiveness and efficiency of the cultural learning process. In order to solve this problem, the use of a Serious Game solution would represent an improvement comparing to the current solutions with an identical purpose available in the market.

The project's scope constitutes the development of a serious game that, by using a Card Collecting mechanic and image classification systems, can help and promote cultural tourism activities, accompanied by the study of several interactive game mechanics, focusing on its consequences and effects among individuals and an direct experience with their physical surroundings.

There's an analysis and description of the multiple development stages associated with the project, as well as the processes involved in quality testing and requirement fulfillment along with the corresponding results.

The serious game obtained at the end stage of the project revealed to possess potential as a solution to the identified problem, presenting a high calculated value for its Global Quality level, as assessed with the Quantitative Evaluation Framework method.

**Keywords:** Serious Games, Card Collecting, Tourism, Image Classification



## Resumo

Dado o constante crescimento da área do turismo a nível internacional, consequentemente se tem registado um aumento da participação em atividades de turismo cultural. Tais atividades por definição não apresentam uma estrutura sólida, dependendo maioritariamente de decisões de circunstância do próprio participante. Ou seja, existe a necessidade de uma ferramenta que permita proporcionar uma melhor estruturação das suas atividades turísticas, assegurando uma melhor eficiência e eficácia no processo didático de aprendizagem de informação cultural. De forma a resolver o problema em questão, reconhece-se que uma solução de um jogo sério dedicado a auxiliar atividades turísticas culturais constituiria uma melhoria face às atuais soluções disponíveis no mercado com o mesmo propósito.

O tema do projeto constitui o desenvolvimento de um jogo sério que, recorrendo à mecânica de *Card Collecting* e uso de sistemas de classificação automática de imagem, consiga auxiliar e promover atividades turísticas de carácter cultural, estando também associado um estudo das consequências e dinâmicas entre indivíduos e várias mecânicas de jogo interativas, com foco na experiência direta com o meio físico.

São descritas as várias fases de desenvolvimento associadas ao projeto, assim como os processos utilizados para averiguar e testar a qualidade e cumprimento dos requisitos, acompanhados dos respetivos resultados.

O jogo sério obtido na fase final do projeto revelou potencial na solução dos objetivos propostos, demonstrando um elevado valor de Qualidade Global calculado segundo o método de Avaliação Quantitativa (QEF).

**Palavras-chave:** Jogos Sérios, Card Collecting, Turismo, Classificação Automática de Imagem



## Índice

|  |    |
|--|----|
| 1. Introdução .....                                      | 1  |
| 1.1. Enquadramento .....                                 | 1  |
| 1.2. Problema.....                                       | 1  |
| 1.3. Objetivos e Resultados Esperados.....               | 2  |
| 2. Contexto .....  | 5  |
| 2.1. Jogos, Jogos Sérios e Ferramentas Interativas ..... | 6  |
| 2.1.1. Soluções com Mecânicas de Exploração.....         | 7  |
| 2.1.2. Soluções de Quiz de Perguntas e Respostas.....    | 8  |
| 2.1.3. Soluções de Imagem .....                          | 9  |
| 2.1.4. Soluções Com Mecânicas de Coleção .....           | 9  |
| 2.2. Tecnologias .....                                   | 13 |
| 2.2.1. Motores de Jogo.....                              | 13 |
| 2.2.2. Classificação Automática de Imagens .....         | 15 |
| 2.3. Análise de Valor .....                              | 19 |
| 2.3.1.1. Identificação da Oportunidade.....              | 20 |
| 2.3.1.2. Análise da Oportunidade.....                    | 20 |
| 2.3.1.3. Conceção da Ideia.....                          | 22 |
| 2.3.1.4. Seleção da Ideia .....                          | 22 |
| 2.3.1.5. Definição do Conceito .....                     | 23 |
| 2.3.2. Valor .....                                       | 23 |
| 2.3.2.1. Valor para o Cliente .....                      | 24 |
| 2.3.2.2. Valor Percebido .....                           | 24 |
| 2.3.3. Perspetiva Longitudinal de Valor .....            | 25 |
| 2.3.4. Proposta de Valor.....                            | 26 |
| 2.3.5. Business Model Canvas .....                       | 27 |
| 2.3.6. Rede de Valor .....                               | 30 |
| 2.3.7. Método de Análise Hierárquica (AHP) .....         | 32 |
| 3. Desenho e Implementação .....                         | 41 |
| 3.1. Objetivos, Regras e Mecânicas de Jogo.....          | 41 |
| 3.1.1. Mecânica de <i>Card Collecting</i> .....          | 42 |
| 3.1.2. Mecânica de Quiz Visual .....                     | 44 |

|          |  |     |
|----------|--|-----|
| 3.1.3.   | Mecânica de Cartas Especiais.....                      | 46  |
| 3.2.     | Análise.....   | 47  |
| 3.2.1.   | Requisitos .....                                       | 47  |
| 3.2.2.   | Mapa de Navegação.....                                 | 50  |
| 3.3.     | Arquitetura.....                                       | 52  |
| 3.3.1.   | Estrutura e Organização do Projeto em Unity .....      | 55  |
| 3.3.2.   | Persistência e Sistema de Dados.....                   | 57  |
| 3.3.2.1. | Sistema de Dados do Jogo .....                         | 58  |
| 3.3.2.2. | Sistema de Dados do Jogador.....                       | 60  |
| 3.3.3.   | Sistema de Coleção Virtual de Cartas .....             | 63  |
| 3.3.4.   | Quiz Visual .....                                      | 67  |
| 3.3.4.1. | Modelo de Classificação .....                          | 69  |
| 3.3.4.2. | Classificador de Imagens .....                         | 72  |
| 3.3.5.   | Upgrade de Cartas – Networking.....                    | 78  |
| 4.       | Avaliação .....  | 86  |
| 4.1.     | Ciclos de Desenvolvimento na Fase de Produção .....    | 86  |
| 4.2.     | <i>Quantitative Evaluation Framework (QEF)</i> .....   | 87  |
| 4.2.1.   | Fase Alfa .....  | 90  |
| 4.2.2.   | Fase Beta .....  | 91  |
| 5.       | Conclusões.....  | 106 |
| 5.1.     | Objetivos Realizados .....                             | 106 |
| 5.2.     | Limitações e Trabalho Futuro .....                     | 107 |
| 6.       | Bibliografia .....                                     | 108 |
|          | Anexo A – Ficheiro cardInfoData_Default.json .....     | 112 |
|          | Anexo B – Objeto GameJsonData.....                     | 114 |
|          | Anexo C – Exemplo do ficheiro PlayerData.json .....    | 116 |
|          | Anexo D – Objeto PlayerJsonData .....                  | 117 |
|          | Anexo E – Objeto Player.cs.....                        | 118 |
|          | Anexo F – Métricas de avaliação QEF.....               | 120 |
|          | Anexo G – Questionário de Qualidade .....              | 129 |
|          | Anexo G – Conceitos Teóricos de Análise de Valor ..... | 132 |

## Índice de Figuras

|   |    |
|---|----|
| Figura 1 - Number of international tourist arrivals worldwide from 2005 to 2017, by region (in millions)[5] .....   | 5  |
| Figura 2 - Exemplo de um minijogo da aplicação Nexto Guide.....   | 11 |
| Figura 3 - Exemplo de um dos minijogos de captura de criaturas no jogo Pokémon GO .....   | 12 |
| Figura 4 - Representação esquemática do processo de classificação numa arquitetura de CNN[35] .....   | 18 |
| Figura 5 - Exemplo de três classes de classificação e respetivos datasets utilizados no processo de treino do modelo de classificação.....  | 19 |
| Figura 6 - Esquema representativo do processo de classificação de imagem com base num modelo de treino.....   | 19 |
| Figura 7 - Métricas associadas ao valor percebido[38] .....   | 25 |
| Figura 8 - Representação em grafo da Rede de Valor aplicada à solução pretendida ..   | 31 |
| Figura 9 - Árvore hierárquica de decisão.....   | 33 |
| Figura 10 - Matriz de comparação dos critérios.....   | 36 |
| Figura 11 - Tabela de comparação dos critérios com o vetor de prioridades relativas .   | 36 |
| Figura 12 - Matriz de comparação das hipóteses relativamente ao critério "Compatibilidade com Unity" .....  | 37 |
| Figura 13 - Tabela de comparação das hipóteses relativamente ao critério "Compatibilidade com Unity" com o vetor de prioridades relativas .....   | 37 |
| Figura 14 - Matriz de comparação das hipóteses relativamente ao critério "Experiência Prévia" .....   | 38 |
| Figura 15 - Tabela de comparação das hipóteses relativamente ao critério "Experiência Prévia" com o vetor de prioridades relativas.....   | 38 |
| Figura 16 - Matriz de comparação das hipóteses relativamente ao critério "Custo" ....   | 38 |
| Figura 17 - Tabela de comparação das hipóteses relativamente ao critério "Custo" com o vetor de prioridades relativas .....   | 39 |
| Figura 18 - Tabela da Matriz Prioridade e vetor das prioridades compostas .....   | 39 |
| Figura 19 - Esquema representativo de uma carta no contexto do projeto.....   | 43 |
| Figura 20 - Modelo de Domínio relativo ao sistema de coleção de cartas .....  | 43 |
| Figura 21 - Esquema gráfico representativo do sistema de coleção virtual projetado, utilizando o sistema de separadores gráficos para dividir vários percursos (Ex.: Porto, Gaia, Lisboa) ..... | 44 |
| Figura 22 - Ilustração representativa da mecânica de Quiz Visual .....  | 45 |
| Figura 23 - Esquema representativo da mecânica de Cartas Especiais .....  | 46 |
| Figura 24 - Diagrama de Casos de Uso relativos ao jogador .....   | 48 |
| Figura 25 - Mapa de Navegação relativo ao jogo.....   | 51 |
| Figura 26 - Diagrama de componentes relativo à primeira arquitetura projetada para o jogo .....   | 53 |
| Figura 27 - Diagrama de componentes relativo à arquitetura final do jogo.....   | 54 |

|  |    |
|--|----|
| Figura 28 - Raiz da árvore hierárquica do projeto de Unity, com os vários objetos<br>canvas representados..... | 55 |
| Figura 29 - Conjunto dos principais scripts Controller .....   | 55 |
| Figura 30 - Conjunto dos scripts Model.....  | 56 |
| Figura 31 - Conjunto dos scripts Utils.....  | 56 |
| Figura 32 - Objeto GameController do projeto de Unity do jogo .....  | 57 |
| Figura 33 - Método managePlayerData presente no script GameController.cs.....                                  | 58 |
| Figura 34 - Modelo de domínio relativo à estrutura de dados do jogo no ficheiro JSON<br>.....                  | 59 |
| Figura 35 - Interface de criação de conta de jogador .....   | 61 |
| Figura 36 - Menu Principal do jogo com informação relativa ao jogador.....                                     | 62 |
| Figura 37 - Método fillPlayerInfo do script GameController.cs .....  | 62 |
| Figura 38 - Menu Principal do jogo com o botão de prémio desbloqueado .....                                    | 63 |
| Figura 39 - Interface relativa ao Prémio das cartas especiais.....   | 63 |
| Figura 40 - Método presentCards do script CollectionController.cs.....   | 64 |
| Figura 41 - Método selectRoute do script CollectionController.cs .....   | 64 |
| Figura 42 - Menu de coleção de cartas do jogo .....  | 65 |
| Figura 43 - Interface do jogo que disponibiliza ao jogador a informação relativa a um<br>percurso.....         | 66 |
| Figura 44 - Interface do jogo com informação de uma carta (parte frontal da carta)...                          | 66 |
| Figura 45 - Interface do jogo com informação de uma carta (parte traseira da carta) .                          | 67 |
| Figura 46 - Método playGame presente no script CollectionController.cs.....                                    | 67 |
| Figura 47 - Interface do jogo a informar o jogador da sua coleção completa .....                               | 68 |
| Figura 48 - Método setupQuestion presente no script QuestionController.cs .....                                | 68 |
| Figura 49 - Interface de jogo relativa à apresentação da questão do quiz visual.....                           | 69 |
| Figura 50 - Conteúdo do diretório "Monuments" .....  | 71 |
| Figura 51 - Execução em terminal do script retrain.py .....  | 71 |
| Figura 52 - Output final da execução do script retrain.py.....   | 71 |
| Figura 53 - Visualização em Inspector do Unity do objeto ClassificationCanvas .....                            | 73 |
| Figura 54 - Método Start presente no script PhoneCamera.cs.....  | 74 |
| Figura 55 - Método LoadClassifier presente no script PhoneCamera.cs .....                                      | 74 |
| Figura 56 - Método Update presente no script PhoneCamera.cs.....   | 75 |
| Figura 57 - Método SnapPhoto presente no script PhoneCamera.cs.....  | 75 |
| Figura 58 - Método TFClassify presente no script PhoneCamera.cs.....   | 76 |
| Figura 59 - Método checkAnswer presente no script QuestionController.cs.....                                   | 76 |
| Figura 60 - Interface do jogo dedicada a informar o jogador da sua resposta de quiz<br>visual correta .....    | 77 |
| Figura 61 - Interface do jogo dedicada a informar o jogador da sua resposta de quiz<br>visual incorreta .....  | 78 |
| Figura 62 - Método saveCard presente no script GameController.cs.....  | 78 |
| Figura 63 - Esquema representativo do primeiro passo do sistema de salas projetado                             | 79 |

|  |     |
|--|-----|
| Figura 64 - Esquema representativo do segundo passo do sistema de salas projetado  | 80  |
| Figura 65 - Esquema representativo do terceiro passo do sistema de salas projetado   | 80  |
| Figura 66 - Interface do jogo relativa ao Upgrade de Carta~ .....  | 81  |
| Figura 67 - Método setCardData presente no script UpgradeCardController.cs.....  | 82  |
| Figura 68 - Método init presente no script UpgradeCardController.....  | 82  |
| Figura 69 - Objeto NetworkItem.cs .....  | 82  |
| Figura 70 - Interface do jogo relativa à leitura do código QR .....  | 83  |
| Figura 71 - Método QRCodeRead presente no script QRCodeController.cs.....  | 83  |
| Figura 72 - Métodos sendNetworkItem e SendItem presentes no script<br>PhotonController.....  | 84  |
| Figura 73 - Método compareNetworkItems presente no script<br>CardUpgradeController.cs .....  | 84  |
| Figura 74 - Interface do jogo responsável por informar o jogador do sucesso da<br>operação de transformação da carta .....   | 85  |
| Figura 75 - Interface do jogo responsável por informar o jogador de uma operação de<br>transformação de carta falhada .....  | 85  |
| Figura 76 - Gráfico de barras relativo às respostas da questão "How easy was to install<br>the game?" .....  | 92  |
| Figura 77 - Gráfico de barras relativo às respostas da questão "Did you find the game's<br>interface intuitive?" .....   | 93  |
| Figura 78 - Gráfico circular relativo às respostas da questão "Did you find any technical<br>errors during your gameplay session?" .....   | 93  |
| Figura 79 - Gráfico circular relativo às respostas da questão "At any moment of your<br>gameplay session, did you feel you lost control over the game's menu actions?" .....               | 94  |
| Figura 80 - Gráfico de barras relativo às respostas da questão "How would you rate the<br>game's feedback regarding your actions?" .....   | 95  |
| Figura 81 - Gráfico de barras relativo às respostas da questão "How would you rate the<br>Card Collection interface?" .....  | 96  |
| Figura 82 - Gráfico de barras relativo às respostas da questão "Did you find the card<br>Collection system intuitive?" .....   | 97  |
| Figura 83 - Gráfico de barras relativo às respostas da questão "Would you classify the<br>card's information as clear and absent of errors?" .....   | 98  |
| Figura 84 - Gráfico de barras relativo às respostas da questão "When answering a game<br>question, after taking a photo, how fast was the result given?" .....                             | 99  |
| Figura 85 - Gráfico de barras relativo às respostas da questão "After answering a game<br>question with the phone camera, how accurate was the result?" .....                              | 100 |
| Figura 86 - Gráfico circular relativo às respostas da questão "Did you find any offensive<br>messages within the game?" .....  | 101 |
| Figura 87 - Gráfico circular relativo às respostas da questão "Do you think this<br>experience could lead people to visit monuments and cultural landmarks in a tourist<br>context?" ..... | 101 |

|  |     |
|--|-----|
| Figura 88 - Gráfico circular relativo às respostas da questão "Would you rather play the game alone or with other people?" .....   | 102 |
| Figura 89 - Gráfico circular relativo às respostas da questão "Do you believe the game helped or could help you learn more about monuments/landmarks or other cultural information?" ..... | 103 |
| Figura 90 - Respostas dos inquiridos à questão "Which aspects do you want to be improved on future releases (if any)?" .....   | 103 |
| Figura 91 - Esquema representativo do processo de inovação[36] .....   | 132 |
| Figura 92 - Esquema representativo do modelo NCD[36] .....   | 133 |

## Índice de Tabelas

|  |     |
|--|-----|
| Tabela 1 - Resumo da Análise SWOT .....  | 21  |
| Tabela 2 - Resumo do Business Model Canvas .....   | 29  |
| Tabela 3 - Escala Fundamental de Saaty[40] .....   | 34  |
| Tabela 4 - Comparação dos critérios de decisão segundo a Escala Fundamental .....                          | 34  |
| Tabela 5 - Comparação das várias hipóteses por critério segundo a Escala Fundamental .....                 | 34  |
| Tabela 6 - Objetivo, Regras e Mecânicas do jogo.....   | 41  |
| Tabela 7 - Lista de percentagens de cumprimento da solução face aos requisitos relativos à Fase Alfa ..... | 90  |
| Tabela 8 - Percentagens de cumprimento de requisitos relativos à Fase Beta .....                           | 104 |



# 1. Introdução

## 1.1. Enquadramento

Tem-se registado um constante crescimento do turismo à escala global e, conseqüentemente, das atividades de caráter turístico. Uma das formas de turismo mais comum é o turismo cultural, caracterizado pela realização de atividades culturais, movidas pelo desejo por parte dos seus participantes de experienciar as várias características do meio que visitam, variando desde contacto linguístico ou gastronómico até à visita de monumentos ou marcos geográficos de valor cultural. Associado a esta última atividade está um desejo de conhecer e aprender informação e factos relativos ao local de visita.

Considerando que toda a experiência relativa a atividades turísticas é de caráter empírico, dado que está associado um contacto direto presencial com o meio, a interatividade assume um papel fulcral no que toca à aprendizagem e assimilação de conhecimento, sendo que seria contraproducente a participação em atividades turísticas não interativas, visto que o contacto direto é intrínseco ao valor do turismo cultural.

## 1.2. Problema

Quando envolvidos num contexto de turismo cultural, as várias atividades realizadas por parte do turista na localização que visita, tradicionalmente não se apresentam estruturadas segundo um plano concreto, dependendo grande parte de decisões de circunstância por parte dos seus participantes, podendo em alguns casos resultar de uma pesquisa prévia ou até após a sua chegada à nova localização geográfica. Estando associado a atividades desta natureza o desejo de experienciar uma nova cultura, existe a necessidade de rentabilizar ao máximo a sua experiência cultural, recorrendo a uma ferramenta que permita ao turista conhecer de forma informada os principais marcos culturais que caracterizam o local que visita. Atualmente existem no mercado várias ferramentas tecnológicas que visam resolver o problema em questão, no entanto, no entanto raramente se focam em oferecer uma experiência lúdica associada à experiência didática envolvida no processo de aprendizagem e participação em atividades turísticas, acabando por indiretamente separar e abstrair o processo de aprendizagem do contacto direto com a cultura.

Uma ferramenta com potencial para interligar organicamente estes dois aspetos na realização de atividades turísticas é a utilização de jogos sérios. O uso de jogos sérios como veículo de divulgação e aprendizagem têm-se revelado viável e cada vez mais tem sido utilizado para apoiar várias áreas de carácter didático, com indivíduos de diversas faixas etárias, géneros, etnias e habilitações literárias[1].

Uma técnica que constantemente se revela útil no aumento da eficácia de atividades didáticas é o uso de técnicas de reforço positivo e associação de sensações positivas ao conhecimento adquirido[2]. Uma das suas aplicações está presente em mecânicas de coleção de itens em vários jogos. O objetivo destas mecânicas é o apelo ao desejo inerente ao jogador de colecionar[3], induzindo-o a completar os desafios que lhe são apresentados em troca de um item que pode ser adicionado a uma coleção virtual presente no próprio jogo.

Tendo em conta as várias mecânicas mencionadas, uma solução que as combine num jogo sério tem potencial para ser uma ferramenta útil no auxílio das atividades turísticas culturais.

### 1.3. Objetivos e Resultados Esperados

O atual clima de crescimento de atividades de turismo internacional, acompanhado pela adoção das tecnologias de informação na dinâmica quotidiana, demonstra que o mercado se encontra recetivo a uma nova ferramenta tecnológica para o apoio de atividades turísticas.

Em virtude da potencialidade estudada, o principal foco do projeto é o desenvolvimento de um novo instrumento de apoio ao turismo cultural, mais concretamente um jogo sério que, recorrendo à mecânica de *Card Collecting* e outras mecânicas interativas entre o jogador e o meio que ele se encontra, fomente e encoraje a visita e a experimentação direta com a cultura que se pretende descobrir.

A etapas do projeto são:

- Estudo da mecânica de *Card Collecting* como um meio de maximizar o interesse de utilização e de transmitir conhecimento;
- Estudo de diversas mecânicas de interatividade em jogos e da forma como influenciam o jogador;
- Estudo e desenvolvimento de um modelo de treino preparado para a classificação automática de imagens relativas a monumentos e marcos culturais;

- Desenvolvimento de um jogo sério de *Card Collecting* orientado ao turismo cultural com um sistema que recorra ao modelo de treino para um classificador automático de imagens;
- Estudo das atividades, processos e dinâmicas associadas ao uso da solução desenvolvida em atividades turísticas.

Tendo por base os objetivos especificados, pretende-se que no final do projeto associado, se obtenha uma solução viável e eficiente que consiga proporcionar uma experiência lúdica aos seus utilizadores, mantendo um certo nível de abstração do seu caráter didático.



## 2. Contexto

Segundo o estudo “UNWTO Tourism Hightlights 2018” realizado pela agência das Nações Unidas *World Tourism Organizaton*, entre 2005 e 2017, foi registada uma tendência constante anual para o aumento no número de turistas internacionais, sendo que é na Europa que se registam os maiores valores[4]:

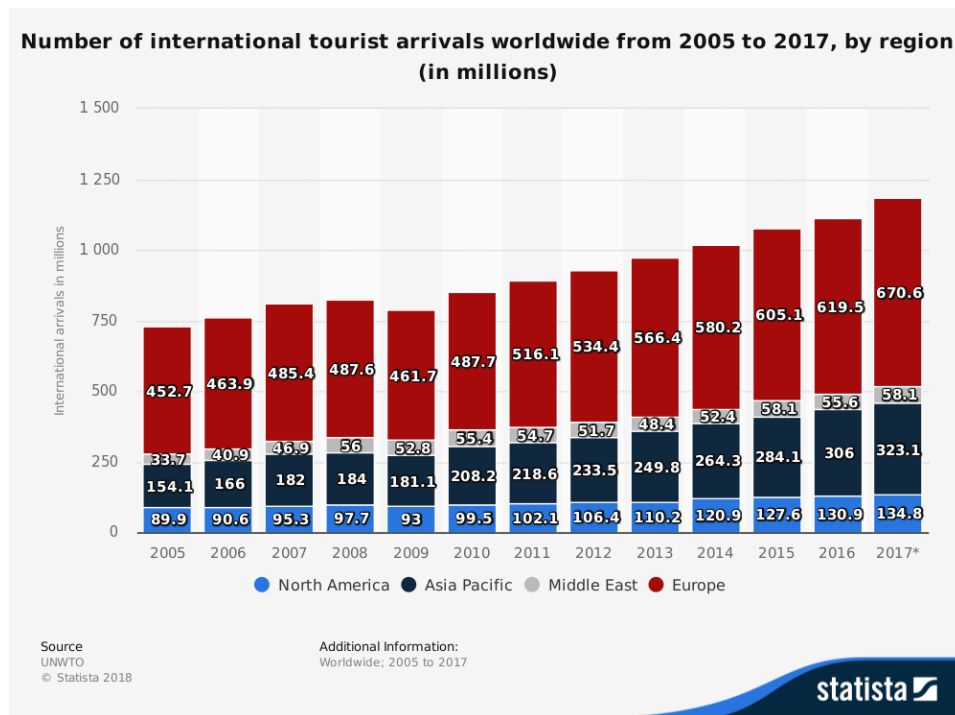


Figura 1 - Number of international tourist arrivals worldwide from 2005 to 2017, by region (in millions)[5]

Como se pode confirmar pelo estudo “*Tourism and Culture Synergies*”[6] realizado pela UNWTO, uma considerável parte da atividade turística registada representa turismo de carácter cultural (estimado um valor percentual entre 14 e 39% do volume de turismo global correspondente ao ano de 2014).

Associado a atividades de turismo cultural está o desejo por parte dos seus participantes de vivenciar presencialmente uma cultura diferente. Estando associadas a esse mesmo desejo, as várias atividades apresentam um carácter didático e costumam ser auxiliadas recorrendo a ferramentas externas tais como material literário, guias turísticos presentes no local, aplicações móveis e até pesquisa realizada pelos próprios participantes antes das suas viagens. Tal auxílio consiste na definição concreta de um itinerário e percurso turístico, sendo que sem qualquer ferramenta orientadora, a experiência cultural do turista não seria tão eficiente e estruturada.

## 2.1. Jogos, Jogos Sérios e Ferramentas Interativas

Desde o aparecimento dos primeiros jogos eletrônicos, existem cada vez mais organizações a recorrer a este meio com o objetivo de aproveitar as suas características de entretenimento como um veículo para cumprir os objetivos que lhes são específicos. Tal subgénero de jogos, constitui os **Jogos Sérios**.

Segundo Marc Prensky[7], a definição de Jogos Sérios é dada aos jogos eletrônicos cujo seu principal objetivo não constitui o entretenimento dos seus jogadores. Este tipo de jogos recorre às várias mecânicas características de jogos eletrônicos mais tradicionais a fim de despertar a motivação por parte dos seus jogadores. Os Jogos Sérios podem-se dividir em diversas categorias diferentes consoante os seus objetivos sérios, tais como jogos de persuasão, sendo exemplos os *Adver Games* (Jogos Sérios usados para promover e publicitar um produto, marca ou organização) e jogos que visam sensibilizar os seus jogadores relativamente a um determinado tema ou causa, jogos educativos, com o objetivo de transmitir informação didática (ex.: jogos em contexto de sala-de-aula, jogos de simulação, etc.), ou até jogos sérios com objetivo de promover ou informar de boas-práticas de saúde (que promovem um boa alimentação ou a prática saudável de exercício físico).

Apesar de não representar o seu propósito principal, a componente lúdica dos jogos sérios é um dos principais fatores na abstração dos seus processos didáticos. Por essa mesma razão tal método constitui um forte candidato na resolução do problema relativo à necessidade de uma ferramenta de apoio e auxílio nas atividades turísticas culturais, mais concretamente, numa ferramenta que disponibilize percursos de turismo cultural e respetiva informação de forma eficaz e orgânica para o turista.

Uma solução de jogo sério para o problema em questão implicaria um estudo e uso cuidadoso de técnicas e mecânicas de jogo que conseguissem aproveitar o meio físico que o turista visita, e que consecutivamente conseguissem assegurar um constante interesse por parte do jogador, mantendo-o motivado no processo didático.

Com o intuito de projetar uma solução de jogo sério, procedeu-se ao estudo de diversas soluções atualmente disponíveis no mercado. Tal estudo foi realizado segundo uma divisão em quatro tópicos diferentes, todos eles representantes de várias mecânicas de interação diferentes: **Soluções com mecânicas de exploração**,

## **Soluções de análise de imagem, Soluções de Quiz e Soluções com mecânicas de coleção.**

### 2.1.1. Soluções com Mecânicas de Exploração

De modo a induzir os jogadores a explorarem presencialmente o mundo real, seja com o intuito de inculcar bons hábitos de atividade física ou para promover o turismo de uma certa zona geográfica, existem vários jogos para plataformas móveis que utilizam mecânicas de exploração. Quatro exemplos de soluções que a são:

**Nexto Guide**[8] - Desenvolvida pela *Nexto Inc.*, esta aplicação visa apoiar os utilizadores que praticam turismo, oferecendo experiências interativas nas suas visitas aos vários locais turísticos. São disponibilizados mapas interativos, minijogos educativos e experiências de realidade aumentada, assim como um guia áudio que é ativado automaticamente quando o utilizador visita o local respetivo, entre outras atividades interativas. A aplicação dispõe de um sistema de troféus que são colecionados pelo utilizador resultantes da realização de certas atividades.

**Pocket Ranger**[9] – Desenvolvida pela *ParksByNature Network* com o objetivo de auxiliar e promover a visita aos vários parques nacionais dos EUA, esta aplicação disponibiliza desafios de “caça ao tesouro”, gravando o progresso e localização dos utilizadores a partir de GPS. Neste caso os “tesouros” são marcos geográficos ou lugares relevantes dentro dos parques. As várias proezas são registadas podendo ser futuramente partilhadas entre utilizadores, havendo também a possibilidade de ganhar prémios. Na aplicação também são fornecidas funções mais comuns de modo a auxiliar os utilizadores nas suas atividades tais como mapas, informação de meteorologia ou alertas de segurança.

**Pokémon GO**[10] – Baseado na série de jogos *Pokémon*, e desenvolvido pela *Niantic, Inc.*, *Pokémon GO* é um jogo disponível para as plataformas *Android* e *iOS*, onde os jogadores, navegando no mundo real, colecionam várias criaturas e itens virtuais presentes em várias localizações geográficas aleatórias, sendo que para as colecionar, é necessário deslocar presencialmente até à localização assinalada no mapa de GPS. O jogo também oferece interação entre jogadores sobre a forma de jogos cooperativos e troca de objetos virtuais.

**Geocaching**[11] – Desenvolvido pela *Groundspeak Inc.*, *Geocaching* é uma aplicação que permite os seus utilizadores participarem numa das maiores redes de atividades de caça-ao-tesouro no mundo real utilizando GPS. Recorrendo a marcações no mapa da aplicação, os seus utilizadores têm como objetivo encontrar vários recipientes

físicos escondidos chamados de “*geocaches*” que contêm objetos aleatórios deixados anteriormente por outros participantes da atividade.

### 2.1.2. Soluções de Quiz de Perguntas e Respostas

Nas várias plataformas de distribuição de aplicações mobile, existem disponíveis diversas soluções de jogos de quiz, onde os jogadores são desafiados, através de diferentes mecânicas e dinâmicas, a responder a perguntas de diversos temas.

A mecânica de quiz constitui um método eficaz de transmissão de conhecimento e informação, sendo a sua utilização popular no atual mercado de jogos em plataformas móveis.

Na seguinte lista são apresentadas e analisadas várias soluções de jogos em plataformas mobile que apresentam um formato de trívia/quiz de perguntas e respostas:

**HQ Trivia**[12] - Jogo disponível na plataforma iOS e futuramente na plataforma Android. Disponibiliza diariamente um concurso mundial, online em direto onde os jogadores têm 10 segundos para responder a cada uma de 12 questões de diversos temas. O jogador que acertar nas 12 é presenteado com parte do prémio diário, transferido para as suas contas de Paypal.

**Jeopardy! World Tour**[13] – Baseado no programa televisivo de perguntas e respostas *Jeopardy!*, este jogo apresenta diversos desafios com vários formatos, onde o jogador deve responder a perguntas de cultura geral, com tabelas de pontuação online, estimulando a competição entre diversos jogadores. Encontra-se disponível nas plataformas Android e iOS.

**Trivia Crack**[14] – Desenvolvido pela *Etermax*, *Trivia Crack* é um jogo de quiz onde os jogadores respondem a perguntas de uma categoria escolhida aleatoriamente entre 6 diferentes, com mecânica de duelo entre dois jogadores através das redes sociais. O jogo é disponibilizado na plataforma Android, iOS e Facebook.

**QuizUp**[15] – Jogo online de quiz desenvolvido pela *Glu*, onde os jogadores podem-se desafiar em duelos de perguntas de várias categorias existentes ou de uma categoria criada pelos próprios. Os jogadores também têm a possibilidade de definir as características que querem no oponente escolhido, tais como género, idade, distância à sua localização atual e interesses.

### 2.1.3. Soluções de Imagem

Atualmente existem várias soluções que, utilizando tecnologias de reconhecimento de imagem, permitem ao utilizador classificar e identificar objetos presentes no mundo real recorrendo à câmara do seu dispositivo.

A solução relativa ao projeto tem como foco a área do turismo, sendo que uso de mecânicas de jogo que envolvam imagem é apropriado ao tema, uma vez que a fotografia e o ato da sua captura representam atividades classicamente associados ao turismo cultural, e também considerando que a este tipo de atividades está fortemente associada uma componente visual.

Na lista que se segue serão analisados vários exemplos de aplicações mobile que, a fim de resolver diferentes problemas, fornecem reconhecimento automático de imagem:

**Google Lens**[16] – Aplicação de reconhecimento de imagem desenvolvida pela *Google* que, a partir da câmara do smartphone, permite identificar plantas e animais, mostrar resultados relevantes na internet com base em texto ou um objeto capturado e identificar marcos e pontos de referência geográficos.

**PlantNet**[17] – Aplicação financiada pela *Agropolis Fondation*, especializada na identificação e classificação de plantas através de fotos capturadas pelo dispositivo móvel recorrendo a uma base de dados botânica.

**Aipoly Vision**[18] – Aplicação desenvolvida pela *Aipoly Inc*, que tem como objetivo auxiliar pessoas com deficiência visual no seu quotidiano, permitindo a identificação de plantas/animais, comida, reconhecimento de texto de diversos idiomas, moedas e notas em dólares, identificação de milhares de tons diferentes de cor, assim como objetos mais gerais presentes no meio.

**Screenshop**[19] – Desenvolvida pela *Craze, Inc.*, esta aplicação especializada em moda, permite ao utilizador identificar artigos de roupa e acessórios, remetendo-os para lojas online que disponibilizam o artigo fotografado pelo utilizador para venda.

### 2.1.4. Soluções Com Mecânicas de Coleção

Uma das mecânicas mais populares nos jogos disponíveis em plataformas móveis é a mecânica de coleção de itens. Esta mecânica constitui um sistema de jogo que disponibiliza ao jogador uma coleção virtual de itens (dependentes do contexto do jogo), sendo que este é encorajado a adquirir novos itens completando um desafio ou

proeza inerente ao jogo. Quatro exemplos de jogos que atualmente utilizam a mecânica de coleção de itens são:

**Dragon Ball Legends**[20] – Desenvolvido pela *Bandai Namco Entertainment Inc.*, *Dragon Ball Legends* constitui um jogo para as plataformas *iOS* e *Android* baseado na série *Dragon Ball*. O objetivo do jogo assenta-se na coleção de diversas personagens da série obtidas pelo jogador a partir de um sistema de sorteio aleatório que pode ser acedido através de um sistema de sorteio, pago com dinheiro real ou moedas de jogo. Recorrendo à sua coleção, o jogador pode fazer equipas com um máximo de 6 personagens e lutar num sistema de jogo de luta contra outros jogadores online ou contra o computador.

**Hearthstone**[21] – Constitui um jogo de cartas por turnos online desenvolvido pela empresa *Blizzard Entertainment, Inc.*, disponível para as plataformas *Windows*, *MacOS*, *Android* e *iOS*. O jogo assenta num sistema de duelos de cartas segundo um conjunto de regras específico a diferentes modos de jogo. Cada deque de cartas é construído pelo próprio jogador a partir das cartas que possui na sua coleção completa. De forma a adquirir novas cartas para a sua coleção, o jogador pode comprar pacotes de cartas com dinheiro real ou dinheiro do jogo, ou pode adquirir novas cartas através de certos desafios e proezas definidos. Cada carta de jogo retrata uma personagem de fantasia e apresenta um determinado conjunto de atributos numéricos e efeitos de jogo.

**Clawbert**[22] – Disponível para as plataformas móveis *Android* e *iOS* e desenvolvido pela *HyperBeard Games*, *Clawbert* representa um jogo casual de coleção de itens baseado na simulação de uma máquina de “garra” com brindes. Tal como o sistema que pretende simular, o jogador controla uma garra a partir de um painel de controlos virtual, composto por um *joystick* e vários botões, a fim de retirar os vários brindes da “máquina”. O jogo disponibiliza um sistema de coleção onde são adicionados os vários brindes que o jogador vai ganhando nas suas sessões de jogo. O conteúdo das “máquinas” de brindes é atualizado com base numa janela virtual definida, dependendo dos vários itens que o jogador possui.

**Yu-Gi-Oh! Duel Links**[23] – Baseado no jogo de cartas físico e série do mesmo nome, *Yu-Gi-Oh! Duel Links* constitui um jogo cartas estratégico desenvolvido pela empresa *Konami* para as plataformas *Android*, *iOS* e *Windows*. O jogo constitui uma adaptação ligeiramente simplificada da sua versão física do jogo de cartas, regendo-se segundo um sistema de duelos entre dois jogadores online, cada um com um deque de cartas construído a partir da sua coleção virtual pessoal. O jogo apresenta um modo de história, onde o jogador duela contra várias personagens fictícios da narrativa. No entanto também tem a possibilidade de duelar contra amigos ou outros jogadores aleatórios online por todo o mundo. O processo de aquisição de novas cartas é baseado num sistema de pacotes de várias cartas, comprados numa loja virtual de

jogo, usado dinheiro de jogo ou dinheiro real a partir de um sistema de *microtransactions*.

Tendo em conta as várias soluções analisadas, entende-se que a solução da aplicação *Nexto Guide* é a que mais se aproxima da solução pretendida, visto que ambas têm como objetivo estimular os seus utilizadores a participar em atividades turísticas com um forte foco em mecânicas de interação com o mundo real. Outra mecânica que as duas soluções partilham é a coleção de itens por parte do utilizador. No entanto é relevante considerar que a solução pretendida, apesar de partilhar objetivos idênticos, representa um jogo sério, ao contrário de *Nexto Guide*, que pretende servir como uma aplicação guia turística interativa. Compreende-se que *Nexto Guide* contém na sua oferta de funcionalidades vários minijogos, porém, estes representam meramente uma parte secundária da experiência do utilizador, constituindo uma das etapas das atividades turísticas do local que o utilizador visita. Um dos exemplos de minijogo está representado na Figura 2, onde o utilizador deve descobrir qual o sítio correto que deve clicar com base na informação que lhe é dada.

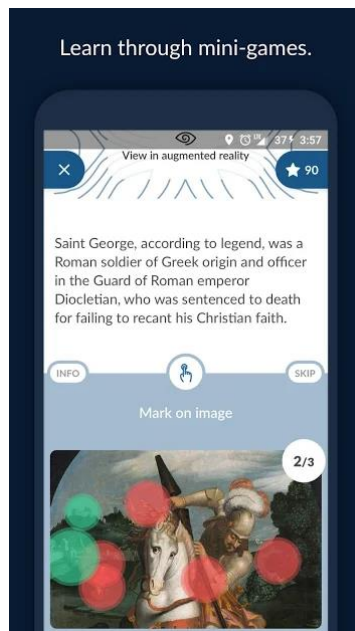


Figura 2 - Exemplo de um minijogo da aplicação *Nexto Guide*

Pretende-se que a solução projetada tenha como principal mecânica um jogo de trívia idêntico aos sistemas de perguntas e respostas presentes nas soluções de *quiz* apresentadas, sendo que o sistema de resposta depende de mecânicas de interação

com o mundo real (ao contrário dos vários jogos de *quiz* analisados). Esta característica constitui outra das diferenças na comparação com *Nexto Guide*, que, neste último caso, não está no principal foco da aplicação.

Relativamente à mecânica de *Card Collecting* que se projeta para a solução pretendida, de entre as soluções estudadas, **entende-se que a melhor se assemelha é a mecânica do *Pokémon GO***. O jogo em questão, apesar de ter sido analisado segundo uma perspetiva de mecânicas de exploração, também apresenta uma forte componente de coleção de itens, sendo que a atividade de coleção de criaturas e itens por parte do jogador constitui o principal foco do jogo (semelhante a outros jogos da mesma série). O jogador tem como objetivo colecionar os vários itens virtuais, sendo que para o conseguir, tem de vencer um desafio. No caso de *Pokémon GO*, o desafio constitui um minijogo de habilidade onde o jogador deve acertar com uma bola na criatura (Figura 3) ou um minijogo de duelo entre as criaturas previamente colecionadas e a criatura que o jogador pretende adicionar à sua coleção, enquanto que na solução pretendida, o desafio para adquirir o item (neste caso uma carta virtual) constitui uma pergunta de quiz que o jogador deve responder sobre o objeto representado na carta em questão.

No entanto, também se compreende que o jogo analisado ***Clawbert* apresenta um sistema de coleção promissor no contexto da solução desejada**, na medida em que o seu sistema de coleção baseia-se no conceito de “brinde”, em que cada item ganho é obtido com o simples propósito de ser colecionado e não para ser utilizado noutro modo de jogo, como é o caso dos vários jogos de duelos de cartas analisados (cada carta tinha um propósito específico na mecânica de duelos).



Figura 3 - Exemplo de um dos minijogos de captura de criaturas no jogo *Pokémon GO*

Em síntese, compreende-se que as várias mecânicas que se esperam na solução pretendida estão presentes em diversas soluções atuais, no entanto, a forma como se organizam entre si e o seu papel e foco na solução são diferentes das várias soluções estudadas.

## 2.2. Tecnologias

Ao nível das tecnologias associadas ao projeto e o respetivo desenvolvimento, dado que a solução pretendida representa um jogo sério, considera-se vantajoso o uso de um motor de jogo na construção da solução. No entanto, considerando que se pretendem implementar mecânicas que requerem a existência de um sistema de classificação automática de imagens, existe a necessidade de uma ferramenta que o permita, e que simultaneamente, seja compatível com o motor de jogo escolhido.

No presente capítulo serão analisadas várias hipóteses de tecnologias relativas a motores de jogo que permitam o desenvolvimento de jogos para plataformas móveis assim como várias ferramentas que possibilitam o desenvolvimento de um sistema de classificação automática de imagens.

### 2.2.1. Motores de Jogo

Um motor de jogo constitui uma aplicação que, a partir de certos níveis de abstração, permite aos seus utilizadores o desenho, desenvolvimento e construção de jogos de uma forma eficiente e ágil.

Atualmente existem diversas opções de motores de jogo que permitem o desenvolvimento de jogos para plataformas móveis, sendo que cada solução se especializa em diferentes atributos e características. De entre os diversas ofertas que o mercado oferece, destacam-se quatro motores de jogo:

- **Unity**[24] – Criado pela empresa estadunidense *Unity Technologies* e tendo a sua primeira versão lançada em 2005, o motor de jogo *Unity* permite simultaneamente o desenvolvimento de jogos em 2D e 3D para diversas plataformas, tais como *Windows, MacOS, iOS, Android, Playstation, Xbox, etc.*. O *Unity* domina o mercado dos motores de jogo utilizados em plataformas móveis, sendo que cerca de 50% dos jogos presentes em neste setor foram desenvolvidos recorrendo ao *Unity*[25]. Relativamente ao seu preço, o *Unity*

apresenta uma versão gratuita para uso pessoal, com a condição de a receita ou financiamento dos seus utilizadores não ultrapasse os 100000\$ ao ano, acompanhada de versões pagas mensalmente com funcionalidades adicionais, mediante a tarifa escolhida. O motor de jogo é apoiado por um sistema de *scripting* na linguagem C# e a sua versão mais atual corresponde ao *Unity 2018*.

- **Unreal Engine**[26] – Lançada a primeira versão em 1998 pela empresa de software *Epic MegaGames, Inc.*, atualmente conhecida como *Epic Games, Inc.*. O motor de jogo *Unreal Engine* visa apoiar primordialmente o desenvolvimento de jogos 3D sofisticados a nível mecânico e gráfico. Apesar de o seu principal alvo ser o desenvolvimento de jogos para computador e consolas (*Windows, MacOS, Linux, Playstation, Xbox*) o motor de jogo permite de igual forma o desenvolvimento para plataformas móveis, tais como *Android* e *iOS*. De forma semelhante a outros motores de jogo, o *Unreal Engine* apresenta um sistema de *scripting*, sendo neste caso específico, utilizada a linguagem C++. A sua utilização é gratuita, no entanto são exigidos 5% das receitas de venda caso sejam ultrapassados 3000\$ ganhos por jogo (desenvolvido a partir do *Unreal Engine*) por cada quadrimestre do ano. A versão mais recente do motor é o *Unreal Engine 4*.
- **GameMaker Studio**[27] – Lançada a primeira versão em 1999[28] por Mark Overmars e mais tarde adquirido pela empresa de distribuição de jogos *YoYo Games*, *GameMaker Studio* é um motor de jogo que permite o desenvolvimento de jogos 2D recorrendo a uma VPL (*Visual Programming Language* – Linguagem de programação Visual). No caso específico do *GameMaker Studio*, é utilizado o sistema *Drag and Drop (DnD)*, que a partir de uma interface de seleção e “arraste” de opções e elementos gráficos, remove a necessidade de codificar para o desenvolvedor. No entanto, o *GameMaker Studio* também permite a utilização de scripts em vez de *DnD*, que utiliza a linguagem proprietária *GML (GameMaker Language)*. O *GameMaker Studio* permite o desenvolvimento para as plataformas *Windows, MacOS, Ubuntu, Playstation 4, Xbox One, Android, iOS*, entre outras. Atualmente, o motor apresenta várias versões, uma versão de teste gratuita dedicada à aprendizagem da ferramenta por parte de novos desenvolvedores, e versões pagas de licença anual ou permanente de diferentes preços, que diferem nas plataformas que a versão possibilita de exportar o projeto desenvolvido.
- **Godot**[29] – Desenvolvido por Juan Linietsky e Ariel Manzur em 2007, *Godot* é um motor de jogo *open-source* para o desenvolvimento de jogos 2D e 3D. Dado o seu carácter *open-source*, não estão associados quaisquer custos de utilização

assim como nenhum pagamento de eventuais receitas obtidas através dos jogos desenvolvidos recorrendo à ferramenta, sendo possível contribuir diretamente para o desenvolvimento do próprio motor de jogo. O *Godot* disponibiliza várias interfaces que permitem o design e animação de conteúdo para os projetos, constituindo assim uma ferramenta para desenvolvedores e designers/artistas envolvidos. Identicamente a outros motores de jogo, o *Godot* apresenta um sistema de visual *scripting*, assim como um sistema de *scripting* clássico, baseado na linguagem proprietária *GDScript*, acompanhado de suporte para outras linguagens, tais como C++ e C#.

A escolha das ferramentas de desenvolvimento na fase de planeamento exige um estudo ponderado do foco do projeto, assim como uma análise dos vários requisitos associados e avaliação das características de cada potencial ferramenta.

Sendo que no caso específico do projeto, a primeira decisão de escolha de tecnologias corresponde à seleção do motor de jogo a utilizar no desenvolvimento da solução pretendida, o principal fator na escolha foi o nível de experiência prévia de utilização da ferramenta, acompanhado da preocupação de assegurar que o motor escolhido permitisse o desenvolvimento de soluções para plataformas móveis.

Dado que o *Unity* suporta o último requisito mencionado e considerando que das várias ferramentas analisadas, é ao *Unity* que está associado o maior nível de experiência prévia (proveniente da realização de projetos recorrendo à ferramenta em questão), considera-se que a sua escolha representa a opção mais viável para assegurar o desenvolvimento ágil e eficiente da solução pretendida.

### 2.2.2. Classificação Automática de Imagens

A necessidade de uma ferramenta que permita o desenvolvimento de um sistema de classificação automática de imagens está relacionada com a implementação da mecânica de jogo relativa à forma de resposta do jogador às questões do quiz, através da captura de uma foto por parte do jogador. A implementação desta mecânica exige que o jogo consiga corrigir as respostas, sendo que é necessário um sistema que consiga averiguar se a fotografia capturada pelo jogador corresponde efetivamente à resposta certa. Tal é possível a partir de um sistema que, através de uma imagem, consiga determinar se esta pertence a alguma das classes predeterminadas de classificação, e em caso afirmativo, se a classe a que pertence corresponde à resposta correta.

Com o intuito de desenvolver o sistema descrito procedeu-se a uma análise e estudo de ferramentas que possibilitem a classificação automática de imagens. Consideraram-se viáveis duas possíveis abordagens para a resolução do problema: a utilização de uma API REST de reconhecimento de imagens (neste contexto, a identificação de monumentos e marcos históricos) e a utilização de uma *framework* de *machine learning* que permita o desenvolvimento de um classificador de imagens.

Relativamente ao uso da API, seria disponibilizada a imagem que se pretende analisar e, a partir de uma chamada da API numa das classes do sistema de jogo, obter-se-iam os resultados da respetiva classificação por parte do serviço responsável pela API. No caso de uma solução que tivesse por base a utilização uma *framework* de *machine learning*, ao contrário da utilização de uma API externa, estaria associado o desenvolvimento e respetivo treino de um classificador preparado para identificar imagens com base num número pré-determinado de classes, sendo este sistema interno ao próprio jogo (independente de qualquer serviço ou servidor remoto).

Das soluções atualmente disponíveis, foram consideradas quatro *frameworks* de *machine learning* e uma API:

- **Tensorflow**[30] – Desenvolvida pelo *Google Brain* (equipa de investigação de inteligência artificial de *deep learning* interna da *Google*) em 2015, *Tensorflow* constitui uma biblioteca *open-source* de computação numérica orientada para a investigação e desenvolvimento de soluções de *machine learning*. Escrito maioritariamente nas linguagens *C/C++* e *Python*, permite o treino de redes neuronais que podem ser usadas em reconhecimento/classificação automática de imagens, conteúdo textual ou áudio, assim como a previsão de dados numéricos. O *Tensorflow* suporta primordialmente o desenvolvimento recorrendo a *Python*, existindo interfaces experimentais em *C++*, *Java* e *Go*, acompanhado de suporte para *C#*, *Julia*, *Ruby* e *Scala* criado pela comunidade *open-source*. O *Tensorflow* suporta o desenvolvimento para as plataformas *Windows*, *MacOS*, *Linux* e *Android*.
- **Keras**[31] – Da autoria do investigador de inteligência artificial François Chollet e lançada em 2015, *Keras* representa uma API *open-source* de alto nível destinada ao desenvolvimento de redes neuronais. Podendo ser executado sobre outras bibliotecas de *deep-learning* (*Tensorflow*, *CNTK* e *Theano*), *Keras* foi implementado com o objetivo de servir como interface em vez de uma *framework* independente. O seu principal foco é permitir o desenvolvimento e experimentação rápida com redes neuronais. A API permite a produção de modelos de *deep-learning* para plataformas móveis (*Android* e *iOS*), plataformas *Web* e *Java Virtual Machine*.
- **Torch**[32] – Lançada originalmente em 2002 por Ronan Collobert, Koray Kavukcuoglu e Clement Farabet, *Torch* constitui uma biblioteca *open-source* especializada em auxiliar o desenvolvimento de algoritmos de *machine learning*, *computer vision*, processamento paralelo, imagem, áudio e vídeo,

entre outros. Implementado na linguagem *C*, *Torch* utiliza a linguagem *LuaJIT*, baseada na linguagem *Lua*, e identicamente a outras ferramentas do mesmo paradigma, permite o desenvolvimento de soluções para plataformas móveis (*Android* e *iOS*).

- **Caffe**[33] – *Caffe (Convolutional Architecture for Fast Feature Embedding)* representa uma *framework open-source* de *deep learning* desenvolvida por Yangqing Jia. Desenvolvida na linguagem *C++*, e apresentando uma interface em *Python*, *Caffe* permite o desenvolvimento de soluções e algoritmos orientados para a classificação e segmentação automática de imagens. Proporcionando um sistema que possibilita uma mudança rápida entre computação de *CPU* e *GPU*, *Caffe* está disponível para os sistemas operativos *Windows*, *MacOS* e *Linux*.
- **Cloud Vision API** [34] – Em 2008, a *Google* disponibilizou no mercado o serviço *Google Cloud Platform*, composto por vários produtos que disponibilizam diversos serviços de computação remota em nuvem. Um dos serviços constitui a *Cloud Vision API*. A API REST em questão permite a análise de imagens recorrendo a *machine learning*. Ao nível de análise, a API permite a identificação de milhares de classes de objetos (flores, animais, localizações, etc.), extração de texto, logótipos, pesquisa no motor de busca da *Google* a partir de imagens, entre outras funcionalidades relativas à extração de informação de imagens. Ao contrário das várias soluções previamente analisadas, a *Cloud Vision API* constitui um serviço pago e não uma ferramenta de utilização gratuita, dado que toda a computação necessária é realizada remotamente em servidores da *Google*, abstraindo o desenvolvedor de todo o sistema responsável pela identificação automática de imagens (é estabelecida uma tarifa mensal por cada serviço, consoante vários intervalos de quantidades diferentes de imagens analisadas).

Resultante da análise das possíveis soluções para o desenvolvimento de um sistema de classificação automática de imagens, recorrendo ao Método de Análise Hierárquica (secção 2.3.7), considerou-se a utilização da *framework Tensorflow* como a solução mais viável face aos vários requisitos e características associados ao projeto.

Para o processo de classificação automática de imagens, o *Tensorflow* recorre a CNNs. CNN (Convolutional Neural Network) representa uma arquitetura de *deep learning* baseada no sistema nervoso de córtex visual[35]. À semelhança do sistema no qual se inspira, a CNN é composta por várias camadas que comunicam entre si, sendo que, a partir *input* de uma imagem, produzem um *output* final de resultados (conjunto de classificações com base na probabilidade da imagem pertencer a cada uma das classes definidas).

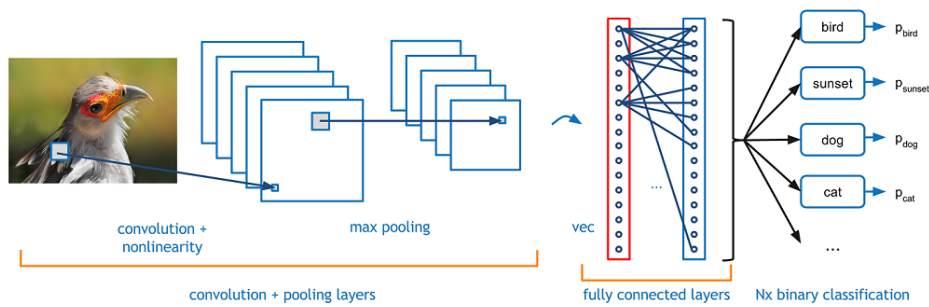


Figura 4 - Representação esquemática do processo de classificação numa arquitetura de CNN[35]

O processo de classificação automática de imagem é realizado a partir de um modelo de classificação pré-treinado, preparado para identificar um conjunto estabelecido de classes de imagens (Figura 6). A fim de se obter o modelo de treino em questão, é necessária a realização de um processo composto por duas fases:

- **Fase de treino** – É durante a fase de treino que o modelo é inicialmente formulado. Para tal, é necessária a definição das várias classes que se pretendem identificar durante o processo de classificação, assim como a construção de um *dataset* de dados representativo de cada classe (no caso específico de classificação de imagem, cada *dataset* é composto por diversas imagens representativas da classe em questão) (Figura 5). Com base nas várias classes e *datasets* definidos, o processo de treino ocorre, sendo que no final desta fase, é obtido um modelo de treino inicial.
- **Fase de teste** – Após a fase de treino, o modelo inicial resultante é submetido a um processo de testes, onde o modelo é testado com base nas classificações geradas a partir de um grupo de dados diferente dos dados utilizados no processo de treino, a fim de ser calculada a taxa de precisão do modelo de classificação, representando a percentagem de dados que foram corretamente classificados.



Figura 5 - Exemplo de três classes de classificação e respetivos datasets utilizados no processo de treino do modelo de classificação

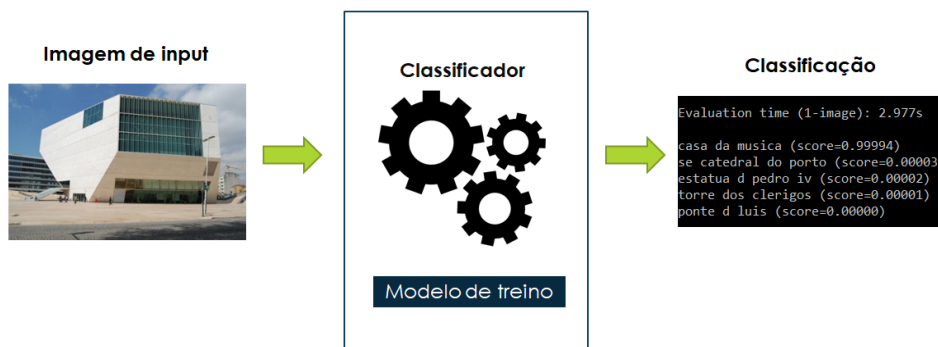


Figura 6 - Esquema representativo do processo de classificação de imagem com base num modelo de treino

### 2.3. Análise de Valor

De forma a rentabilizar os vários custos e esforços associados à criação de um novo produto ou abordagem, existe a necessidade de um estudo cuidado de análise de valor, para que seja possível atingir um equilíbrio saudável entre os recursos necessários e o valor gerado, ou até mesmo averiguar se tal é possível.

Com base no estudo teórico dos conceitos de Análise de Valor presente na Anexo G – Conceitos Teóricos de Análise de Valor, nas seguintes secções serão descritas as cinco atividades principais correspondentes à parte central do modelo NCD enquadradas no projeto em questão.

### 2.3.1.1. Identificação da Oportunidade

A oportunidade associada ao projeto onde a solução pretendida está inserida surge da constatação do potencial que vários fatores externos positivos providenciam: o aparente crescimento da área de negócio inerente à solução pretendida numa escala global, neste caso específico, a área do turismo cultural; o potencial da utilização de jogos sérios e várias mecânicas interativas no contexto das atividades de turismo cultural; o aumento da adoção das tecnologias de informação em tarefas e atividades do quotidiano por parte da sociedade geral.

A conjugação dos vários fatores descritos formula a oportunidade e justifica o desenvolvimento de uma solução preparada para auxiliar a área do turismo, dado que tanto o mercado da área de negócio como o público geral se encontram recetíveis a uma nova solução tecnológica com a finalidade de o auxiliar.

### 2.3.1.2. Análise da Oportunidade

Dada a complexidade de criação de uma nova solução ou produto associados a um negócio e as respetivas incertezas relativas à sua implementação, é necessária uma análise que tenham em consideração os vários fatores associados, de modo a aferir se é efetivamente benéfico avançar com o desenvolvimento do produto/solução.

De modo a auxiliar a análise em questão, procedeu-se à realização de uma análise SWOT (*Strengths, Weaknesses, Opportunities, Threats*), que perspetiva de um modo interno os vários pontos fortes e fracos associados ao desenvolvimento da solução pretendida, assim como os vários fatores do meio externo que a visam afetar, sendo as oportunidades os fatores positivos e as ameaças externas os fatores negativos:

**Forças** – O objetivo principal do jogo a desenvolver é o auxílio das atividades de turismo cultural, levando assim à promoção do turismo, sendo este benéfico para o crescimento do interesse pela cultura. O facto de a solução pretendida constituir um jogo sério representa um dos seus pontos fortes, visto que a aplicação de jogos em contextos sérios se tem demonstrado útil no cumprimento dos seus objetivos, mais concretamente, no aumento da eficiência em processos didáticos[1]. O uso da mecânica de *Card Collecting* constitui de igual forma um benefício na solução pretendida, considerando as suas potencialidades benéficas no ato de aprendizagem[2], [3]. Tendo em conta o papel da interatividade em atividades turísticas, a utilização de mecânicas de interatividade tais como a classificação automática de imagem representam uma mais-valia para a solução, estimulando os seus jogadores a visitar pessoalmente os vários locais turísticos. Dadas as mecânicas de

interação pretendidas, considera-se como força a promoção por parte do jogo da cooperação entre jogadores. De igual forma, pressupõe-se que a disponibilidade do jogo em plataformas móveis é benéfica, dados os benefícios associados ao caráter prático do uso destas plataformas em meios exteriores, sendo estes onde ocorrem a grande maioria das atividades turísticas.

**Fraquezas** – A solução pretendida lida com conceitos de negócio associados ao turismo, e dado que não existe um domínio significativo prévio por parte da equipa de desenvolvimento da área em questão, considera-se que tal constitui uma fraqueza.

**Oportunidades** – O atual panorama de crescimento aparente do setor turístico constitui uma oportunidade a soluções que lidem com o setor, sendo que esse mesmo fenómeno de crescimento aparente se verifica a um nível global, possibilitando assim a+ entrada da solução pretendida em diversos mercados internacionais. Tais oportunidades são ainda apoiadas pela crescente adoção por parte do público a tecnologias de informação e soluções tecnológicas que visam auxiliar as suas atividades quotidianas.

**Ameaças** – A existência de soluções com os mesmos objetivos de auxiliar o turismo cultural constitui uma possível concorrência à solução pretendida, sendo que estas se encontram disponíveis nas mesmas plataformas de utilização.

Na Tabela 1 encontram-se resumidos os vários fatores da análise SWOT relativa ao potencial negócio da solução pretendida.

Tabela 1 - Resumo da Análise SWOT

| Forças  | Fraquezas  |
|---|--|
| <ul style="list-style-type: none"> <li>• Promoção do turismo e interesse pela cultura;</li> <li>• Jogo Sérió;</li> <li>• Utilização da mecânica de <i>Card Collecting</i></li> <li>• Mecânicas de interatividade;</li> <li>• Promoção da cooperatividade;</li> <li>• Disponibilidade da solução em plataformas móveis.</li> </ul> | <ul style="list-style-type: none"> <li>• Inexistência de um domínio prévio de conceitos da área do turismo por parte da equipa de desenvolvimento</li> </ul> |
| Oportunidades   | Ameaças  |

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• Crescimento do setor turístico a nível global;</li> <li>• Possibilidade de entrada no mercado internacional;</li> <li>• Crescimento da adoção de tecnologias de informação.</li> </ul> | <ul style="list-style-type: none"> <li>• Existência de outras soluções no mercado com os mesmo objetivos</li> </ul> |
|---|---|

Tendo por base a análise realizada, entende-se que o desenvolvimento da solução pretendida seria vantajoso de um modo geral, tendo em conta o potencial contributo para o público-alvo, a implementação de mecânicas de interatividade inovadoras, tais como a classificação automática de imagens, o previsto crescimento do setor de negócio e do aumento da procura por soluções tecnológicas.

#### 2.3.1.3. Conceção da Ideia

Tendo por base as várias características identificadas na oportunidade, considera-se como projeto de solução o desenvolvimento de um jogo sério que, recorrendo a mecânicas interativas, auxilie os participantes de turismo cultural nas suas atividades, promovendo visitas físicas aos vários locais de importância cultural.

Relativamente às mecânicas de interação pretendidas, consideraram-se duas mecânicas:

- Um jogo sério que utilize um sistema de georreferenciação para estimular o jogador a visitar fisicamente os vários pontos de importância cultural.
- Um jogo sério que utilize a mecânica de *Card Collecting* e de classificação automática de imagens como forma de estímulo à visita de pontos turísticos.

#### 2.3.1.4. Seleção da Ideia

Face às várias ideias que surgiram durante a fase de seleção da ideia com base nas várias soluções estudadas no Estado da Arte e pesadas as consequências das duas hipóteses colocadas, considerou-se que a opção que traria maior valor global seria a opção de desenvolver o jogo sério utilizando a mecânica de *Card Collecting* associada à classificação automática de imagem, dado que esta traria um maior nível de interatividade às várias atividades turísticas que pretende auxiliar.

### 2.3.1.5. Definição do Conceito

Considerando a escolha realizada na fase de seleção de ideia das hipóteses formuladas, foi definido que a solução que haveria de dar resposta às oportunidades encontradas seria o desenvolvimento de um jogo sério que utilize a mecânica de *Card Collecting* num sistema de *quiz* de perguntas onde o processo de resposta envolve a captura por foto de pontos de interesse turístico, sendo que as respostas seria corrigidas através da classificação automática de imagens. O jogo em questão representa uma nova ferramenta interativa na área do turismo cultural internacional.

### 2.3.2. Valor

Ao conceito de “valor” são constantemente associadas diferentes definições, consoante a área que o estuda. No contexto empresarial/marketing, corresponde à percepção da relação dos atributos de um produto com os custos/recursos que lhe estão associados por parte de duas entidades diferentes, os consumidores e os produtores/distribuidores. À percepção de valor está associado um certo nível de subjetividade intrínseco a cada indivíduo, dado que pode ser dependente de características sociais, culturais, diferentes considerações económicas ou de utilidade[37].

Segundo o estudo de Woodall[37], é possível decompor o conceito de valor em quatro tipos diferentes:

- Valor de troca (*exchange value*) – valor diretamente dependente do objeto, das suas características inerentes e do mercado que o oferece.
- Valor intrínseco (*intrinsic value*) – valor exclusivamente dependente dos atributos do objeto (resultante do período anterior e durante o consumo).
- Valor de uso (*use value*) – valor de caráter circunstancial dependente do uso do objeto (resultante do período de consumo e posterior).
- Valor utilitário (*utilitarian value*) – valor também de caráter circunstancial, dependente de um balanço ponderado entre o valor intrínseco, valor de uso e os custos que o indivíduo suporta para poder usufruir dessas formas de valor.

No contexto particular do projeto em questão considera-se que à solução pretendida está associado maioritariamente o valor de uso, dado que constitui uma nova solução que só terá valor para o cliente se cumprir de forma eficiente os propósitos sérios ao qual se propõe.

### 2.3.2.1. Valor para o Cliente

Em 2003, Woodall defende no seu estudo que o valor para o cliente (*value for the customer* ou VC) constitui a ponderação resultante da comparação direta dos seus sacrifícios com os benefícios obtidos através destes, realizada de uma forma determinada e que pode ser expressa de forma racional ou intuitivamente[37].

Com o intuito de estabelecer uma divisão mais sólida relativamente aos vários tipos de valor para o cliente considerados anteriormente por diversos autores, Woodall identificou cinco formas diferentes do conceito:

- *Net VC* – Constitui o valor para o cliente resultante de um balanço direto entre os benefícios e sacrifícios relativos a um produto.
- *Marketing VC* – Representa o valor para o cliente baseado na sua perceção dos atributos do produto.
- *Sale VC* – Corresponde ao valor para o cliente atribuído considerando exclusivamente o preço do produto, representado pela redução do sacrifício por parte do cliente.
- *Rational VC* – Representa o valor para o cliente resultante da diferença relativa ao preço objetivo.
- *Derived VC* – Relativo ao valor para o cliente atribuído com base nas suas experiências e uso do produto.

Tendo por base os diferentes tipos de valor para o cliente identificados por Woodall, considera-se que, no caso específico da solução pretendida, está subjacente um valor para o cliente do tipo *Derived VC*, pressupondo de igual forma a classificação de valor de uso. A solução pretendida consiste numa ferramenta que auxilie atividades turísticas, consequentemente, só será ponderado o seu valor por parte dos clientes com base no seu uso e experiência nas suas atividades.

### 2.3.2.2. Valor Percebido

Na maioria dos casos, quando um indivíduo se depara com um produto, ocorre um processo de cálculo do valor, que tem por base uma comparação direta entre a sua perceção dos benefícios e respetivos custos associados à utilização do produto. Durante o cálculo, essa perceção é obtida com base em diferentes métricas que têm em conta todas as características do produto. Este processo é denominado de valor

percebido pelo cliente. Segundo Jozée Lapiere, na sua publicação “*Customer-perceived value in industrial contexts*”, é possível identificar 13 métricas, relativas a benefícios e sacrifícios associados ao produto, serviço e relação com a respetiva marca/organização que o indivíduo tem em conta ao formular a sua perceção[38]. As várias métricas em questão encontram-se representadas na Figura 7.

| Domain \ Scope | PRODUCT               | SERVICE  | RELATIONSHIP                   |
|----------------|-----------------------|--|--------------------------------|
| BENEFIT        | Alternative solutions |  |                                |
|                | Product quality       | Responsiveness                                     | Image                          |
|                | Product customization | Flexibility<br>Reliability<br>Technical competence | Trust<br>Solidarity            |
| SACRIFICE      |                       | Price  | Time/effort/energy<br>Conflict |

Figura 7 - Métricas associadas ao valor percebido[38]

Relativamente à solução pretendida, numa perspetiva de produto, entende-se que o cálculo de valor com base nos potenciais benefícios é maioritariamente dependente da perceção prévia da qualidade do jogo sério, tanto a nível de entretenimento, como da forma que as sessões de jogo iriam influenciar as atividades turísticas do indivíduo, sendo que a qualidade das interfaces e grafismo deste tipo de soluções também representa um fator crucial na formulação de opinião de valor por parte do cliente. No que diz respeito ao fator preço, assume-se que a existência do modelo de negócio baseada em aplicações “*freemium*” (conforme descrito na secção 2.3.5) faria com que o cliente se preocupasse exclusivamente com o preço da versão “*premium*” do jogo, resultante da sua experiência prévia com a versão gratuita e com base na comparação de preços no mercado das aplicações móveis. Considerando que a solução pretendida representa um projeto novo, não estando associada a qualquer projeto ou marca previamente estabelecida, não se espera qualquer perceção por parte do cliente relativamente ao fator “*relação*”, pressupondo que todas as características do produto compreendidas pelo cliente serão fruto de um primeiro contacto e experiência.

### 2.3.3. Perspetiva Longitudinal de Valor

Na publicação “*Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis*” [37], Woodall defende que é possível analisar o valor para o cliente seguindo um modelo de perspectiva longitudinal, onde os vários fatores são analisados seguindo uma linha temporal composta por 4 momentos críticos:

- **Pré-compra** – Constitui o período anterior à compra do produto/serviço. No caso da solução pretendida, considera-se presente um interesse prévio por parte do cliente de experimentar uma ferramenta em forma de jogo sério nas suas atividades turísticas, interesse este apoiado pelos hábitos do uso de aplicações móveis como apoio a atividades diárias. No entanto, nesta fase, pode estar associada uma incerteza por parte do cliente em utilizar um jogo sério nas suas atividades, pelo que surge a necessidade de recorrer a meios e técnicas de promoção de modo a eliminar possíveis estigmas e indecisões por parte do cliente, dado que o valor da solução está fortemente associado ao seu uso e que este poderia ser considerado pelo cliente um desperdício do seu tempo empregue em turismo. Considera-se que, associado à incerteza do cliente, também está a falta de conhecimento por parte do cliente da entidade que lhe fornece o produto, conforme descrito na secção 2.3.2.2..
- **Transação** – Corresponde ao momento durante o qual a transação ocorre. No contexto do projeto, considera-se benéfica a existência de uma versão gratuita da solução, facilitando assim o seu uso, dado que não estariam associados custos diretos. Dado o modelo de negócio baseado numa versão “*premium*” e “*freemium*”, sob um ponto de vista do cliente, não se consideram aspetos negativos, dado que este pode usufruir da ferramenta em questão sem qualquer custo direto.
- **Pós-compra** – Representa o momento de cálculo do valor para o cliente que ocorre após a aquisição do produto/serviço. Considera-se que nesta fase, o cliente classificará como positivo os objetivos não sérios do jogo em questão, mais concretamente o conteúdo de entretenimento da coleção de cartas e o uso das várias mecânicas de interatividade, tais como a captura de imagens e respetiva classificação automática, assim como as mecânicas de cooperação entre jogadores.
- **Pós-experiência** – Equivale à fase de cálculo do valor por parte do cliente que procede toda a experiência de utilização do produto. Dentro do contexto da solução pretendida, considera-se que o cliente, após a sessão de utilização de jogo, sinta associada à sua experiência de entretenimento e interatividade uma experiência didática de carácter cultural, validando assim o uso da ferramenta no seu contexto.

#### 2.3.4. Proposta de Valor

Com vista na aquisição e captura de clientes e a fim de promover um produto ou serviço, as organizações necessitam de possuir uma proposta de valor que comunique qual o seu produto, os seus segmentos de clientes, que valor oferecem aos seus clientes através do seu produto e as razões pelo qual o seu produto é único.

No caso específico do projeto, a solução pretendida constitui um jogo sério de *Card Collecting* orientado para auxiliar participantes em turismo cultural nas suas atividades. Através de mecânicas interativas com o meio e de um sistema de cooperação ente jogadores, é encorajada a visita e aprendizagem de marcos e monumentos de importância cultural, fomentando de igual forma a aprendizagem de informação cultural relevante.

### 2.3.5. Business Model Canvas

Para que seja assegurada uma estruturação sólida das várias características de negócio em que um produto assenta, é necessária a utilização de ferramentas com o objetivo de apoiar o planeamento estratégico de gestão. Uma das ferramentas com essa finalidade é o **Business Model Canvas**, que recorrendo a uma divisão consistente em 9 blocos, ajuda a planificar e esquematizar de uma forma prática os vários conceitos de negócio.

Conforme representado na Tabela 2, seguem-se definidos os vários blocos diretamente aplicados à solução pretendida:

- **Parceiros-Chave** – Considerando que a área de negócio da solução pretendida é o turismo, mais concretamente o turismo cultural, considera-se vantajosa a associação a parceiros envolvidos no setor, tais como agências turísticas, entidades diretamente responsáveis pelos vários pontos turísticos ou organizações de gestão de pontos de informação e apoio turístico. Estes podem servir como um veículo de promoção do jogo, como fontes de pesquisa para o seu conteúdo didático, podem também representar potenciais clientes.
- **Atividades-Chave** – Considera-se imperativo para o potencial arranque do negócio o desenvolvimento da solução pretendida, acompanhado da respetiva manutenção, e uma pesquisa de conteúdo didático, de modo a assegurar a sua qualidade e veracidade.
- **Recursos-Chave** – De forma a assegurar as atividades-chave para a concretização da proposta de valor aos segmentos de clientes, existe a necessidade de hardware e software essenciais ao desenvolvimento e manutenção da solução pretendida. Considerando que o jogo em questão necessita de conteúdo didático para as questões que vai apresentar aos seus utilizadores, existe a necessidade de uma base de conhecimento, neste caso,

de conteúdo didático dentro do tema do turismo cultural, resultante de uma pesquisa prévia devidamente creditada.

- **Proposta de Valor** – O principal objetivo da solução pretendida é disponibilizar aos participantes de turismo cultural uma ferramenta auxiliar às suas atividades, sob a forma de um jogo sério que, recorrendo a várias mecânicas interativas, encoraja a visita e aprendizagem de pontos turísticos e a cooperação entre jogadores.
- **Canais de Distribuição** – Dado que se pretende uma solução preparada para dispositivos móveis, entende-se que os principais canais de distribuição seriam plataformas de distribuição de aplicações móveis (ex.: *Google Play*). Indiretamente, considerando que os próprios parceiros-chave serviriam também como veículo promotor, assume-se que constituem também potenciais canais de distribuição.
- **Relações com Clientes** – Tendo por base que as plataformas de distribuição de aplicações móveis constituem canais de distribuição do produto, também podem servir como meios de comunicação a parte dos segmentos de clientes, mais concretamente, os participantes em turismo, dado que atualmente, estas plataformas de distribuição já apresentam recursos que possibilitam um contacto direto com os seus utilizadores, variando desde sistema de comentários, análise de aplicações e outros sistemas que permitem um *feedback* direto (contacto de e-mail). Pressupondo que os parceiros-chave funcionariam também como canal de distribuição, considera-se que também serviriam como intermediários na comunicação com o cliente.
- **Segmento de Clientes** – O principal segmento de clientes que se pretende capturar seriam os utilizadores diretos do jogo, neste caso, os participantes em atividades de turismo cultural. Consideram-se também como segmento de clientes os próprios parceiros-chave, dado que os seus negócios lidam diretamente com o primeiro segmento de clientes.
- **Estrutura de Custos** – Calcula-se que todos os custos necessários ao desempenho da proposta de valor correspondem ao custo necessário dos recursos-chave, que neste caso específico, são os custos de hardware e software necessários ao desenvolvimento e manutenção da solução pretendida, assim como os custos associados à pesquisa e validação do conteúdo didático do jogo.
- **Fontes de Receita** – Pressupõe-se um modelo de negócio baseado no mercado das aplicações para plataformas móveis, apoiado pelas receitas geradas pela publicação de publicidade nas versões “*freemium*” da aplicação do jogo (sendo esta uma versão gratuita) e nas receitas provenientes da venda direta da versão “*premium*” da aplicação do jogo (versão sem qualquer conteúdo publicitário).

Tabela 2 - Resumo do Business Model Canvas

|  |  |   |  |  |
|--|--|---|--|--|
| <p><b>Parceiros-Chave</b></p> <ul style="list-style-type: none"> <li>- Agências turísticas;</li> <li>- Entidades diretamente responsáveis por monumentos ou marcos turísticos relevantes;</li> <li>- Organizações com pontos de informação orientados a turistas.</li> </ul> | <p><b>Atividades-Chave</b></p> <ul style="list-style-type: none"> <li>- Desenvolvimento e respetiva manutenção do jogo;</li> <li>- Pesquisa de conteúdo didático.</li> </ul>   | <p><b>Proposta de Valor</b></p> <ul style="list-style-type: none"> <li>- Auxílio em atividades turísticas culturais;</li> </ul> | <p><b>Relações com Clientes</b></p> <ul style="list-style-type: none"> <li>- Realizadas pelas próprias plataformas de distribuição de aplicações móveis;</li> <li>- Parceiros-chave são potenciais intermediários;</li> <li>- Contacto de e-mail de apoio dedicado.</li> </ul> | <p><b>Segmentos de Clientes</b></p> <ul style="list-style-type: none"> <li>- Participantes em turismo cultural;</li> <li>- Parceiros-chave.</li> </ul> |
|  | <p><b>Recursos-Chave</b></p> <ul style="list-style-type: none"> <li>- Hardware e software necessários ao desenvolvimento e manutenção do jogo;</li> <li>- Base de conhecimento de conteúdo didático necessário ao jogo;</li> <li>- Servidor preparado para eventuais dados necessários ao jogo.</li> </ul> |   | <p><b>Canais de Distribuição</b></p> <ul style="list-style-type: none"> <li>- Plataformas de distribuição de aplicações móveis (Google Play)</li> <li>- Promoção do jogo por parte dos parceiros-chave.</li> </ul>   |  |

| Estrutura de Custos   | Fontes de Receita  |
|---|--|
| <ul style="list-style-type: none"> <li>- Custos relativos ao desenvolvimento e respetiva manutenção da solução;</li> <li>- Custos associados à pesquisa do conteúdo didático do próprio jogo</li> </ul> | <ul style="list-style-type: none"> <li>- Receitas de publicidade presente em versões “<i>freemium</i>” do jogo;</li> <li>- Receitas provenientes da venda da versão “<i>premium</i>” do jogo sem publicidade.</li> </ul> |

### 2.3.6. Rede de Valor

De forma a melhor perspetivar e analisar o valor global associado a um produto ou organização, existe a necessidade de decompor os vários processos inerentes do negócio em redes internas e externas de valor. Uma das metodologias que visa auxiliar tal estudo é a análise da **rede de valor**. Na rede de valor são esquematizados num grafo todos os intervenientes e respetivas interações tangíveis e intangíveis que contribuem para a entrega de valor por parte da organização. Segundo Verna Allee[39], para perceber de que forma o valor integral é criado, é necessário perceber globalmente o funcionamento e dinâmicas da rede, dado que o valor é uma propriedade emergente da rede.

Com base no estudo de Allee, para melhor se compreender o valor associado à solução pretendida, recorreu-se ao método descrito da rede de valor (Figura 8):

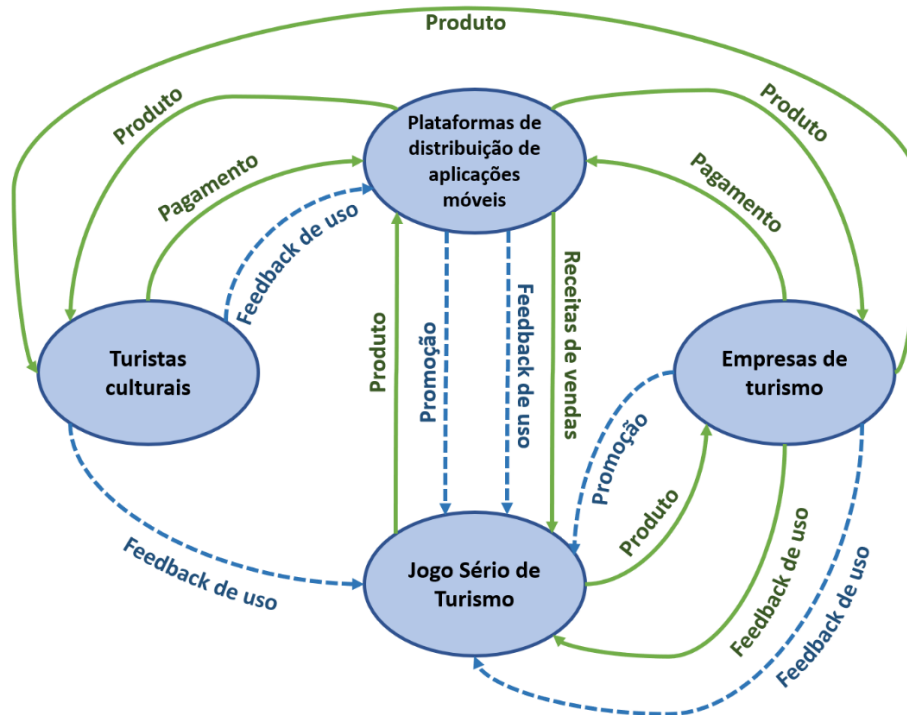


Figura 8 - Representação em grafo da Rede de Valor aplicada à solução pretendida

As várias entidades envolvidas no fluxo de valor são:

- **Jogo sérió de turismo** – Neste caso específico, esta entidade representa simultaneamente o produto e a organização. Relativamente às interações onde é participante, entende-se que a organização possui como principal canal de distribuição do seu produto as plataformas de distribuição de aplicações móveis, sendo que é a partir destas que obtém as receitas de venda provenientes do produto.
- **Plataformas de distribuição de aplicações móveis** – Dados que estas plataformas servem como canal de distribuição, são estas entidades que concedem o acesso ao produto aos segmentos de clientes, além de servirem simultaneamente como veículos de promoção ao produto e como plataforma de comunicação com os segmentos de clientes.
- **Participantes em turismo cultural** – Estes constituem o principal segmento de clientes do produto. Adquirem o produto através das plataformas de distribuição móvel e realizam o seu pagamento através das mesmas. No que toca ao contacto com a organização que disponibiliza o jogo sérió e respetivo *feedback* de uso, é realizado através da plataforma onde adquirem o produto assim como através de um contacto direto (ex.: *e-mail*) com a organização.
- **Empresas associadas a atividades de turismo** – A várias empresas associadas à área do turismo constituem uma possível forma de promoção do produto junto

dos seus associados, dado que também podem ser parceiros da organização responsável pelo produto, assim como podem também ser um potencial segmento de clientes, realizando a aquisição do produto através das plataformas de distribuição de aplicações móveis. No entanto, também está prevista a possibilidade de adquirirem o produto diretamente da organização, dependendo da natureza da parceria. Além de cederem o respetivo feedback de uso dos seus associados, também servem como fonte de pesquisa de conteúdo didático a adicionar ao jogo (ex.: informação sobre monumentos e marcos culturais).

### 2.3.7. Método de Análise Hierárquica (AHP)

Com o objetivo construir uma ferramenta orientada ao auxílio da tomada de decisão em processos de gestão, Thomas Saaty desenvolveu em 1980 o Método de Análise Hierárquica (AHP – *Analytic Hierarchy Process*)[40]. Este método permite esquematizar o problema e as várias hipóteses de resolução possíveis segundo uma perspetiva hierárquica, através de métricas de decisão qualitativas e quantitativas.

No caso específico do projeto associado à solução pretendida, um dos problemas de decisão de hipóteses presentes no processo de desenvolvimento é a escolha de uma ferramenta que permita o desenvolvimento de um sistema de classificação automática de imagens. Conforme exposto na secção 2.2, foram consideradas cinco soluções candidatas à resolução do problema em questão. Para auxiliar o processo de decisão de forma objetiva, procedeu-se ao uso do método AHP, dado que o problema em mãos apresentava um carácter de decisão entre um determinado número de hipóteses possíveis segundo um conjunto de fatores de decisão.

Numa primeira fase, conforme é ditado no método AHP, procedeu-se à esquematização hierárquica do problema, acompanhado dos vários fatores de decisão e das respetivas hipóteses consideradas (Figura 9):

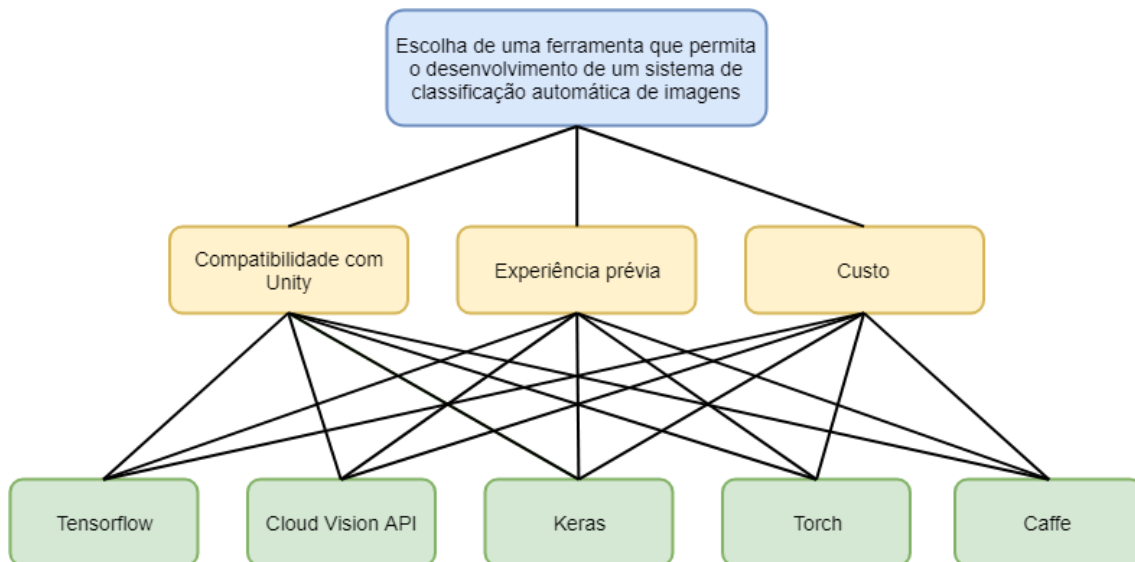


Figura 9 - Árvore hierárquica de decisão

Foram considerados três critérios de decisão:

- **Compatibilidade com Unity** – Dado que o motor de jogo escolhido para o desenvolvimento da solução pretendida é o Unity, o fator de compatibilidade e facilidade da ferramenta com o motor escolhido é tido em consideração.
- **Experiência prévia** – De modo a agilizar todo o processo de desenvolvimento, considera-se relevante o fator de experiência prévia no uso da ferramenta a escolher.
- **Custo** – Dado que existem ferramentas em que a sua utilização gratuita é limitada, estando associado um custo após um certo limite definido de utilização, classificou-se o custo como um fator importante no processo de decisão.

Após a hierarquização dos vários conceitos relativos ao problema de decisão, seguindo a metodologia AHP, estabeleceram-se as várias prioridades entre os vários critérios de decisão e entre as várias hipóteses segundo esses mesmos critérios. Para tal, procedeu-se ao cálculo das várias matrizes de comparação para cada nível hierárquico da árvore estabelecida. O seu cálculo é baseado nas classificações de importância segundo a Escala Fundamental[40] definida por Saaty. A Escala Fundamental é composta por valores numéricos, entre 1 e 9, que representam diferentes níveis de comparação (Tabela 3).

Tabela 3 - Escala Fundamental de Saaty[40]

| Nível de importância | Definição               | Explicação  |
|----------------------|-------------------------|---|
| 1                    | Igual importância       | As duas atividades contribuem igualmente para o objetivo                              |
| 3                    | Fraca importância       | A experiência e o julgamento favorecem levemente uma atividade em relação à outra     |
| 5                    | Forte importância       | A experiência e o julgamento favorecem fortemente uma atividade em relação à outra    |
| 7                    | Muito forte importância | Uma atividade é muito fortemente favorecida em relação a outra                        |
| 9                    | Importância absoluta    | A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza |
| 2, 4, 6, 8           | Valores intermediários  | Quando se procura uma condição de compromisso entre duas definições                   |

A fim de se obter a matriz de comparação de importância entre os vários critérios definidos, procedeu-se à classificação dos vários níveis aplicando os valores da Escala Fundamental (Tabela 4), assim como das classificações de comparação de importância das várias hipóteses dentro de cada um dos três critérios (Tabela 5).

Tabela 4 - Comparação dos critérios de decisão segundo a Escala Fundamental

|                           |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                    |
|---------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------------------|
| Compatibilidade com Unity | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Experiência Prévia |
| Compatibilidade com Unity | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Custo              |
| Experiência Prévia        | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Custo              |

Tabela 5 - Comparação das várias hipóteses por critério segundo a Escala Fundamental

| Compatibilidade com Unity |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
|---------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|
| Tensorflow                | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Cloud Vision API |
| Tensorflow                | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Keras            |
| Tensorflow                | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch            |
| Tensorflow                | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe            |
| Cloud Vision API          | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Keras            |
| Cloud Vision API          | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch            |

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Keras | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Keras | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Torch | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

**Experiência Prévia**

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Cloud Vision API |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Keras |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Keras |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Keras | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Keras | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Torch | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

**Custo**

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Cloud Vision API |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Keras |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Tensorflow | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Keras |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
| Cloud Vision API | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
| Keras            | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Torch |
| Keras            | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |
| Torch            | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Caffe |

Após as várias classificações de importância na Escala Fundamental, organizaram-se os resultados numa matriz de comparação de critérios relativa ao segundo nível, acompanhada da respetiva matriz normalizada (Figura 10). A partir da matriz normalizada procedeu-se ao cálculo do vetor de prioridades relativas (representado na última coluna da Figura 11) a fim de identificar a ordem de importância de cada critério definido.

| Critérios                 | Compatibilidade com Unity | Experiência Prévia | Custo |                  | Critérios                 | Compatibilidade com Unity | Experiência Prévia | Custo |
|---------------------------|---------------------------|--------------------|-------|------------------|---------------------------|---------------------------|--------------------|-------|
| Compatibilidade com Unity | 1                         | 3                  | 2     | Normalizada<br>→ | Compatibilidade com Unity | 6/11                      | 1/2                | 4/7   |
| Experiência Prévia        | 1/3                       | 1                  | 1/2   |                  | Experiência Prévia        | 2/11                      | 1/6                | 1/7   |
| Custo                     | 1/2                       | 2                  | 1     |                  | Custo                     | 3/11                      | 1/3                | 2/7   |

Figura 10 - Matriz de comparação dos critérios

| Critérios                 | Compatibilidade com Unity | Experiência Prévia | Custo | Prioridade Relativa |
|---------------------------|---------------------------|--------------------|-------|---------------------|
| Compatibilidade com Unity | 6/11                      | 1/2                | 4/7   | 0,5396              |
| Experiência Prévia        | 2/11                      | 1/6                | 1/7   | 0,1634              |
| Custo                     | 3/11                      | 1/3                | 2/7   | 0,297               |

Figura 11 - Tabela de comparação dos critérios com o vetor de prioridades relativas

Com base no vetor de prioridades relativas obtido, o critério com maior peso é “Compatibilidade com Unity”, seguindo-se do “Custo” e em último o critério “Experiência Prévia”.

Após o cálculo do vetor de prioridades relativas dos critérios, existe a necessidade de aferir a consistência dos vários julgamentos em relação a julgamentos aleatórios. A grandeza em questão é representada pela Razão de Consistência (RC). Seguindo a fórmula  $RC = \frac{IC}{IR}$ , em que *IC* representa o Índice de Consistência e *IR* representa o índice aleatório obtido através do cálculo em matrizes quadradas de ordem *n* pelo Laboratório Nacional de *Oak Ridge*, obteve-se o valor 0.0049 para o RC. Dado que este valor é inferior a 0.1, está provada a consistência dos valores das prioridades relativas.

Da mesma forma que foi realizado o cálculo da matriz de comparação de critérios e vetor de prioridades relativas, procedeu-se ao cálculo da matriz de comparação para cada hipótese aplicada a cada critério (Figura 12, Figura 14 e Figura 16) e o respetivo vetor de prioridades relativas (Figura 13, Figura 15, Figura 17).

| Compatibilidade com Unity | Tensorflow | Cloud Vision API | Keras | Torch | Caffe |                  |  | Compatibilidade com Unity | Tensorflow | Cloud Vision API | Keras | Torch | Caffe |
|---------------------------|------------|------------------|-------|-------|-------|------------------|--|---------------------------|------------|------------------|-------|-------|-------|
| Tensorflow                | 1          | 1/2              | 3     | 3     | 3     | Normalizada<br>→ |  | Tensorflow                | 1/4        | 2/9              | 3/10  | 3/10  | 3/10  |
| Cloud Vision API          | 2          | 1                | 4     | 4     | 4     |                  |  | Cloud Vision API          | 1/2        | 4/9              | 2/5   | 2/5   | 2/5   |
| Keras                     | 1/3        | 1/4              | 1     | 1     | 1     |                  |  | Keras                     | 1/12       | 1/9              | 1/10  | 1/10  | 1/10  |
| Torch                     | 1/3        | 1/4              | 1     | 1     | 1     |                  |  | Torch                     | 1/12       | 1/9              | 1/10  | 1/10  | 1/10  |
| Caffe                     | 1/3        | 1/4              | 1     | 1     | 1     |                  |  | Caffe                     | 1/12       | 1/9              | 1/10  | 1/10  | 1/10  |

Figura 12 - Matriz de comparação das hipóteses relativamente ao critério "Compatibilidade com Unity"

| Compatibilidade com Unity | Tensorflow | Cloud Vision API | Keras | Torch | Caffe | Prioridade Relativa |
|---------------------------|------------|------------------|-------|-------|-------|---------------------|
| Tensorflow                | 1/4        | 2/9              | 3/10  | 3/10  | 3/10  | 0,2741              |
| Cloud Vision API          | 1/2        | 4/9              | 2/5   | 2/5   | 2/5   | 0,4304              |
| Keras                     | 1/12       | 1/9              | 1/10  | 1/10  | 1/10  | 0,0985              |
| Torch                     | 1/12       | 1/9              | 1/10  | 1/10  | 1/10  | 0,0985              |
| Caffe                     | 1/12       | 1/9              | 1/10  | 1/10  | 1/10  | 0,0985              |

Figura 13 - Tabela de comparação das hipóteses relativamente ao critério "Compatibilidade com Unity" com o vetor de prioridades relativas

| Experiência Prévia | Tensorflow | Cloud Vision API | Keras | Torch | Caffe            |             |   | Experiência Prévia | Tensorflow | Cloud Vision API | Keras | Torch | Caffe |
|--------------------|------------|------------------|-------|-------|------------------|-------------|---|--------------------|------------|------------------|-------|-------|-------|
| Tensorflow         | 1          | 3                | 4     | 4     | 4                | Normalizada | → | Tensorflow         | 12/25      | 6/11             | 4/9   | 4/9   | 4/9   |
| Cloud Vision API   | 1/3        | 1                | 2     | 2     | Cloud Vision API |             |   | 4/25               | 2/11       | 2/9              | 2/9   | 2/9   |       |
| Keras              | 1/4        | 1/2              | 1     | 1     | Keras            |             |   | 3/25               | 1/11       | 1/9              | 1/9   | 1/9   |       |
| Torch              | 1/4        | 1/2              | 1     | 1     | Torch            |             |   | 3/25               | 1/11       | 1/9              | 1/9   | 1/9   |       |
| Caffe              | 1/4        | 1/2              | 1     | 1     | Caffe            |             |   | 3/25               | 1/11       | 1/9              | 1/9   | 1/9   |       |

Figura 14 - Matriz de comparação das hipóteses relativamente ao critério "Experiência Prévia"

| Experiência Prévia | Tensorflow | Cloud Vision API | Keras | Torch | Caffe | Prioridade Relativa |
|--------------------|------------|------------------|-------|-------|-------|---------------------|
| Tensorflow         | 12/25      | 6/11             | 4/9   | 4/9   | 4/9   | 0,4737              |
| Cloud Vision API   | 4/25       | 2/11             | 2/9   | 2/9   | 2/9   | 0,2011              |
| Keras              | 3/25       | 1/11             | 1/9   | 1/9   | 1/9   | 0,1084              |
| Torch              | 3/25       | 1/11             | 1/9   | 1/9   | 1/9   | 0,1084              |
| Caffe              | 3/25       | 1/11             | 1/9   | 1/9   | 1/9   | 0,1084              |

Figura 15 - Tabela de comparação das hipóteses relativamente ao critério "Experiência Prévia" com o vetor de prioridades relativas

| Custo            | Tensorflow | Cloud Vision API | Keras | Torch | Caffe            |             |   | Custo      | Tensorflow | Cloud Vision API | Keras | Torch | Caffe |
|------------------|------------|------------------|-------|-------|------------------|-------------|---|------------|------------|------------------|-------|-------|-------|
| Tensorflow       | 1          | 5                | 1     | 1     | 1                | Normalizada | → | Tensorflow | 5/21       | 5/21             | 5/21  | 5/21  | 5/21  |
| Cloud Vision API | 1/5        | 1                | 1/5   | 1/5   | Cloud Vision API |             |   | 1/21       | 1/21       | 1/21             | 1/21  | 1/21  |       |
| Keras            | 1          | 5                | 1     | 1     | Keras            |             |   | 5/21       | 5/21       | 5/21             | 5/21  | 5/21  |       |
| Torch            | 1          | 5                | 1     | 1     | Torch            |             |   | 5/21       | 5/21       | 5/21             | 5/21  | 5/21  |       |
| Caffe            | 1          | 5                | 1     | 1     | Caffe            |             |   | 5/21       | 5/21       | 5/21             | 5/21  | 5/21  |       |

Figura 16 - Matriz de comparação das hipóteses relativamente ao critério "Custo"

| Custo            | Tensorflow | Cloud Vision API | Keras | Torch | Caffe | Prioridade Relativa |
|------------------|------------|------------------|-------|-------|-------|---------------------|
| Tensorflow       | 5/21       | 5/21             | 5/21  | 5/21  | 5/21  | 0,2381              |
| Cloud Vision API | 1/21       | 1/21             | 1/21  | 1/21  | 1/21  | 0,0476              |
| Keras            | 5/21       | 5/21             | 5/21  | 5/21  | 5/21  | 0,2381              |
| Torch            | 5/21       | 5/21             | 5/21  | 5/21  | 5/21  | 0,2381              |
| Caffe            | 5/21       | 5/21             | 5/21  | 5/21  | 5/21  | 0,2381              |

Figura 17 - Tabela de comparação das hipóteses relativamente ao critério "Custo" com o vetor de prioridades relativas

Por último, procedeu-se à multiplicação da matriz composta pelas várias prioridades relativas a cada critério (Figura 13, Figura 15 e Figura 17) pelo vetor composto pelos pesos dos critérios calculados inicialmente (Figura 11). O vetor resultante da multiplicação matricial corresponde às prioridades compostas das várias hipóteses (Figura 18).

| Prioridades Relativas | Compatibilidade com Unity | Experiência Prévia | Custo  | Resultados   |
|-----------------------|---------------------------|--------------------|--------|--------------|
| Tensorflow            | 0,1479                    | 0,0774             | 0,0707 | <b>0,296</b> |
| Cloud Vision API      | 0,2322                    | 0,0329             | 0,0141 | 0,2792       |
| Keras                 | 0,0532                    | 0,0177             | 0,0707 | 0,1416       |
| Torch                 | 0,0532                    | 0,0177             | 0,0707 | 0,1416       |
| Caffe                 | 0,0532                    | 0,0177             | 0,0707 | 0,1416       |

Figura 18 - Tabela da Matriz Prioridade e vetor das prioridades compostas

Dado que a hipótese de resposta ao problema corresponde ao maior valor das prioridades compostas, conclui-se que o **método AHP classificou a hipótese do *Tensorflow* como a mais indicada para a resolução do problema** (com o valor de 0.296), neste caso particular, a escolha de uma ferramenta que permitisse o desenvolvimento de um sistema de classificação automática de imagens.



### 3. Desenho e Implementação

Na presente secção será analisada a solução que visa responder ao problema associado ao projeto em questão. Serão definidas as várias mecânicas inerentes ao jogo, acompanhadas da definição dos requisitos associados ao projeto, assim como a respetiva estrutura arquitetural global e relativa às principais funcionalidades envolvidas.

#### 3.1. Objetivos, Regras e Mecânicas de Jogo

Conforme definido na secção 1.3, pretende-se que seja explorada a mecânica de *Card Collecting* num jogo sério orientado para o auxílio de atividades de turismo cultural. Considerando o conceito idealizado, procedeu-se à definição dos objetivos e regras de jogo, de forma a melhor estruturar a ideia inicialmente projetada (Tabela 6).

Tabela 6 - Objetivo, Regras e Mecânicas do jogo

| Objetivo Principal   |
|--|
| O jogo disponibiliza ao jogador uma coleção virtual de cartas. Cada carta representa um monumento, marco ou objeto de importância cultural para um certo percurso turístico. O jogador deve progressivamente obter novas cartas para aumentar a sua coleção. |
| Regras e Mecânicas   |
| 1 - Para obter uma nova carta, o jogador deve responder a uma pergunta de quiz visual.   |
| 2 - Cada questão de quiz apresentada é relativa ao tema que a carta retrata.   |
| 3 - Para responder a uma questão, o jogador deve capturar uma fotografia com o seu telemóvel do objeto ao qual a pergunta se refere.   |
| 4 - Caso a resposta em formato de fotografia se encontra correta, é adicionada uma nova carta à coleção do jogador.  |
| 5 - Em qualquer momento do jogo, o jogador tem a possibilidade de transformar qualquer carta da sua coleção numa carta especial.   |
| 6 - Uma carta especial representa uma carta que foi remetida ao processo de “ <i>upgrade</i> ”. No jogo é representada com uma cor distinta das restantes cartas.  |
| 8 - Para transformar uma carta numa carta especial (“ <i>upgrade</i> ” da carta), o jogador deve interligar-se com outro jogador que possua a mesma carta na sua coleção   |

9 - Após o jogador conseguir completar a sua coleção e transformar 100% das suas cartas em cartas especiais, é premiado com um prémio.

Das várias mecânicas mencionadas na Tabela 6, destacam-se três, das quais constituem o núcleo do jogo projetado: mecânica de *Card Collecting*, mecânica de Quiz visual e mecânica das cartas especiais.

### 3.1.1. Mecânica de *Card Collecting*

Sendo um dos focos do projeto a utilização da mecânica de *Card Collecting* no jogo, foi esquematizada a sua utilização na sua forma mais pura, um sistema de jogo focado no ato de colecionar e aumentar progressivamente uma coleção virtual de cartas.

Como referido na secção 1.2, o objetivo da utilização da mecânica de *Card Collecting* é utilizar o desejo inerente ao jogador de colecionar como um fim para atingir os seus objetivos sérios. Ao introduzir um sistema de jogo que tenha como objetivo colecionar um número determinado de itens, o jogador sentir-se-á tentado a completar a sua coleção, estimulando e satisfazendo progressivamente o comportamento compulsivo de colecionar. Tal comportamento está constantemente presente desde os primórdios da evolução humana, diretamente associado com o processamento de sinais relacionados com o conceito de valor dado a objetos com base em experiências e memórias, realizado no lóbulo frontal do cérebro humano[3].

À semelhança dos vários jogos de cartas e cromos colecionáveis físicos no qual o sistema em questão se inspira, as várias coleções e cartas retratam um tema específico, sendo que cada carta representa a partir de uma imagem e informação textual um objeto relativo ao tema[41].

Dado que o tema sério do jogo associado ao projeto é o turismo cultural, as várias cartas do jogo representam vários monumentos, marcos e objetos de relevância cultural para uma certa localização geográfica, sendo que apresentam uma imagem ilustrativa e a respetiva descrição com informação relativa ao objeto (Figura 19).



Figura 19 - Esquema representativo de uma carta no contexto do projeto

Considerando que o sistema de coleção de cartas projetado tem potencial para albergar uma elevada quantidade de cartas relativas a diferentes localizações geográficas, considera-se conveniente um sistema de organização de cartas, onde estas são divididas com base nas suas localizações ou segundo um conjunto de percursos turísticos pré-definidos. A fim de colmatar a necessidade em questão, procedeu-se à projeção do sistema de coleção de cartas do jogador segundo o modelo de domínio representado na Figura 20.

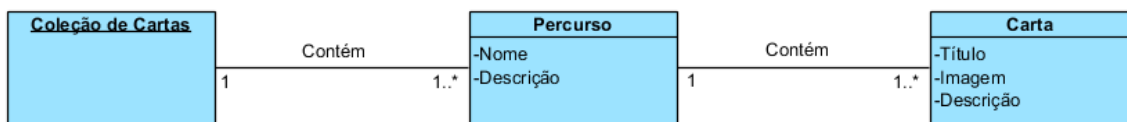


Figura 20 - Modelo de Domínio relativo ao sistema de coleção de cartas

Ao introduzir o conceito de “Percurso”, é possível estabelecer e organizar uma divisão clara entre as várias cartas. Segundo o modelo estabelecido, a coleção virtual de cartas do jogador é composta por vários percursos, que por sua vez contêm um conjunto de cartas que lhes são específicas. Além de simplificar a potencial interface de navegação da coleção virtual de cartas (ex.: divisão da totalidade das cartas da coleção por diversos separadores gráficos ou páginas, otimizando o processo de exploração da coleção, conforme o exemplo representativo na Figura 21) a introdução do conceito de percursos também permite a adição de informação que caracterize as várias localizações dos monumentos e marcos representados nas cartas, complementando assim a experiência didática que o jogador obtém através do jogo sério.

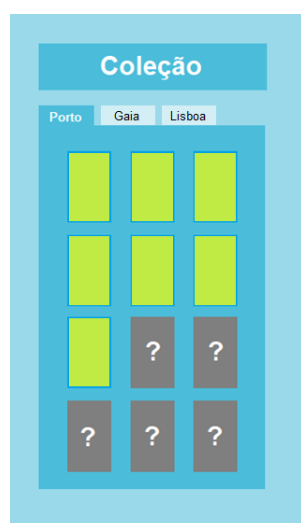


Figura 21 - Esquema gráfico representativo do sistema de coleção virtual projetado, utilizando o sistema de separadores gráficos para dividir vários percursos (Ex.: Porto, Gaia, Lisboa)

### 3.1.2. Mecânica de Quiz Visual

Dado que o objetivo sério do jogo projetado é o auxílio de atividades de turismo cultural, está pressuposto uma utilização do jogo numa localização exterior, diretamente nos locais de interesse cultural/turístico. Considerando este fator, conforme referido na secção 2, é importante que o jogo possa usufruir desta volatilidade do meio e, a partir de mecânicas de jogo que dependam da interação direta com o meio em que o jogador se encontra, consigam maximizar o proveito que o jogador tira da sua experiência de jogo, tanto a nível lúdico como didático.

Com base na necessidade mencionada, foi projetada a mecânica de **Quiz Visual**.

Identicamente ao sistema de quiz tradicional, a mecânica de quiz visual é composta por um sistema de jogo de questões e escolha de repostas por parte do jogador sobre determinado tema (sendo as repostas da versão tradicional do tipo resposta-aberta ou escolha múltipla). No entanto, no caso da versão visual, a resposta não é dada através de qualquer *input* textual, mas sim a partir da captura de uma fotografia por parte do jogador (Figura 22). Neste caso, uma resposta correta constitui uma fotografia que é classificada pelo sistema como sendo uma representação do objeto do qual a questão de quiz se refere.

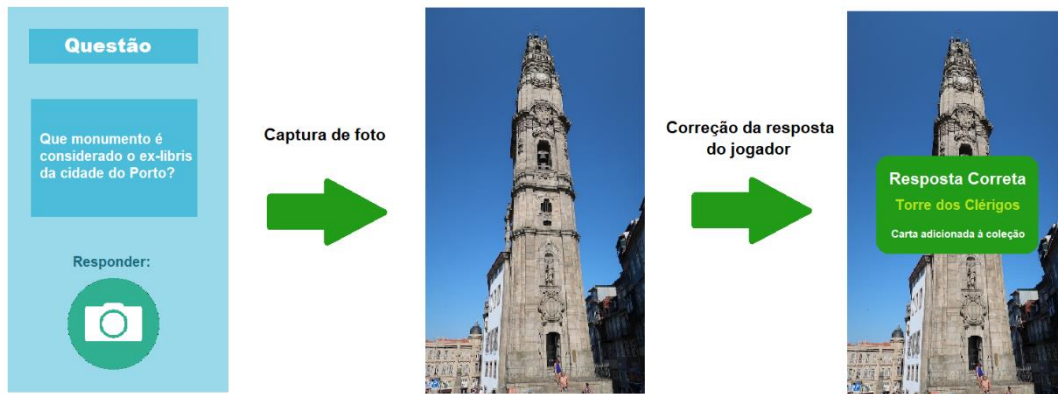


Figura 22 - Ilustração representativa da mecânica de Quiz Visual

O objetivo do sistema de quiz em questão é introduzir ao jogador uma forma de adquirir novas cartas para adicionar à sua coleção, servindo como complemento para a mecânica de *Card Collecting*. Por cada questão que o jogador responda corretamente durante o quiz visual, é adicionada uma nova carta à coleção, sendo a carta uma representação direta do objeto ao qual a questão e respetiva resposta retratam (secção 3.1.1).

Dado que para responder a uma questão é necessária uma fotografia capturada diretamente no local do monumento pelo próprio jogador, este necessita obrigatoriamente de se deslocar entre os diversos locais de interesse cultural da cidade em que pratica turismo.

O ato de incrementar a coleção com base em questões corretamente respondidas, tem por base o sistema de reforço positivo associado à atividade didática. Por reforço positivo, entende-se o ato de introduzir um estímulo positivo no indivíduo após um comportamento que se pretende que seja repetido no futuro[2]. O objetivo da técnica é fortalecer e assegurar o comportamento em questão. No caso específico do jogo, pretende-se assegurar que o jogador continue progressivamente a responder questões de quiz visual, introduzindo como estímulo positivo a adição de novas cartas na sua coleção virtual.

Relativamente à carta obtida através do processo de quiz visual, é utilizada a imagem capturada pelo jogador durante a sua resposta na ilustração da carta (Figura 19). A implementação da funcionalidade tem como objetivo oferecer ao jogador uma experiência de coleção personalizada, sendo que cada carta possui uma ilustração original, da autoria do próprio jogador.

### 3.1.3. Mecânica de Cartas Especiais

Ao comportamento e ato de colecionar está fortemente associada uma componente social, relacionada com o desejo de conviver e partilhar a coleção com outros indivíduos que apresentem o mesmo interesse[3]. Tal comportamento é observado na subida do nível da hormona oxitocina, no centro do sistema nervoso humano, sendo um fator importante na manifestação de empatia entre indivíduos[3], [42].

O fator descrito é continuamente observado nos comportamentos que ocorrem no caso das coleções físicas de cartas, onde os indivíduos sentem interesse em partilhar e exibir as suas coleções ou participar em diversas trocas de itens com outros indivíduos.

No caso específico da mecânica de *Card Collecting* implementada no jogo, considera-se que permitir ou encorajar a troca de cartas virtuais entre dois jogadores retiraria todo o propósito associado ao objetivo sério de auxiliar atividades turísticas, dado que cada carta é obtida através de uma experiência direta com o objeto que representa (fotografar o objeto no local de interesse cultural). Ao obtê-la pelo processo de troca, o jogador nunca teria experienciado fisicamente o monumento/marco cultural associado à carta recebida.

Com o intuito de introduzir uma mecânica que promova uma experiência social e fomente a cooperação entre jogadores, sem isolar o jogador de qualquer experiência cultural direta, foi projetada a **mecânica de cartas especiais**.

A mecânica em questão consiste num sistema que permite o jogador transformar qualquer uma das cartas que possui na sua coleção numa carta “especial”, ou seja, uma versão esteticamente diferente da versão normal da carta. O fator social da mecânica está presente no processo de transformação da carta: a fim de obter uma carta especial, o jogador deve interligar-se com outro jogador que possua a mesma carta na sua própria coleção. No final do processo, os dois jogadores participantes obtêm a versão especial da carta.

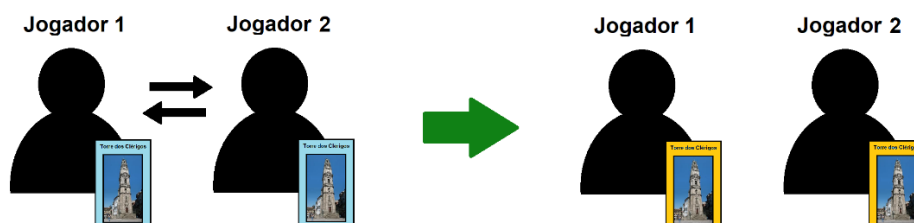


Figura 23 - Esquema representativo da mecânica de Cartas Especiais

Ao adicionar a mecânica de cartas especiais no jogo, é naturalmente introduzido um objetivo secundário para o jogador: completar a sua coleção virtual de cartas apresentado 100% das suas cartas como especiais.

De forma a solidificar o interesse do jogador neste novo objetivo, estipulou-se um sistema que visa premiar o jogador por uma coleção de cartas especiais 100% completa. O prémio em questão representa a oferta de um código promocional que pode ser usado pelo jogador para obter um desconto ou oferta num serviço turístico (entrada paga num monumento/museu cultural, bilhete de uma viagem de autocarro turístico, etc.). Além de apresentar ao jogador um objetivo concreto que se transmite num prémio que pode ser usado no meio real, é complementada a mecânica social e promovida de igual forma a participação e envolvimento em atividades de valor cultural.

### 3.2. Análise

De acordo com os objetivos associados ao projeto (secção 1.3), pretende-se o desenvolvimento de um jogo sério orientado ao turismo cultural recorrendo à mecânica de *Card Collecting* e classificação automática de imagens.

O desenvolvimento de uma solução tecnológica exige uma definição explícita do conjunto de requisitos associados, assim como, numa fase posterior, uma planificação do sistema de navegação. Nas seguintes secções serão explicitados e analisados os vários requisitos funcionais e não-funcionais associados à solução projetada, assim como a sua estrutura de navegação.

#### 3.2.1. Requisitos

Dado que a solução projetada representa um jogo, os vários requisitos funcionais estão fortemente interligados com as várias ações possíveis por parte do jogador, assim como as várias mecânicas de interação presentes nos objetivos do projeto. De forma a melhor esquematizar as várias ações do jogador, procedeu-se à realização do diagrama dos vários casos de uso (Figura 24).

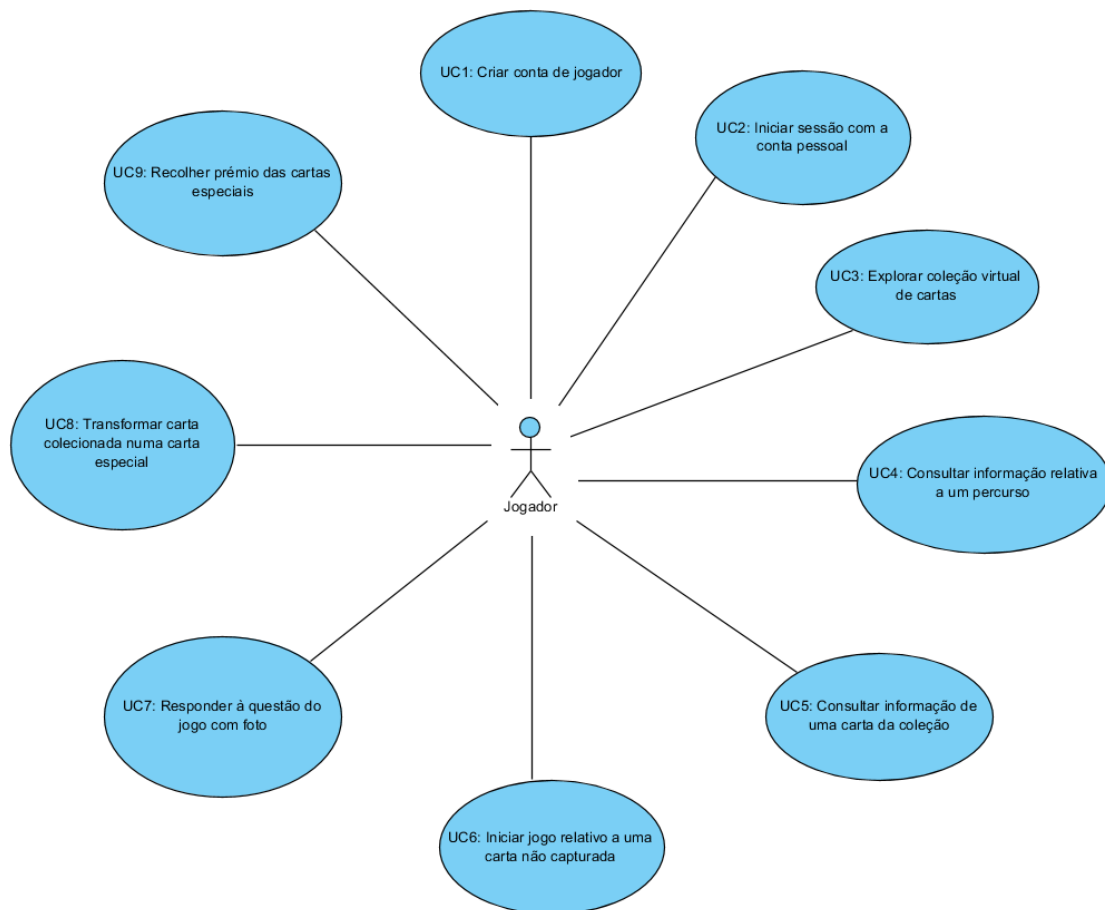


Figura 24 - Diagrama de Casos de Uso relativos ao jogador

Os casos de uso projetados para o utilizador são:

- **UC1: Criar conta de jogador** – O jogador, ao iniciar o jogo pela primeira vez, cria a sua conta de jogador ao qual a sua coleção virtual de cartas estará associada.
- **UC2: Iniciar sessão com a conta pessoal** – Após a primeira utilização de jogo, o jogador iniciará sessão na sua conta de jogador (processo é realizado automaticamente)
- **UC3: Explorar coleção virtual de cartas** – É apresentada ao jogador uma coleção virtual de cartas, composta por vários percursos. O jogador seleciona o percurso desejado e o sistema apresenta as cartas relativas ao percurso em questão (são apresentadas as várias cartas que o jogador possui, assim como as cartas bloqueadas que ainda lhe faltam colecionar).
- **UC4: Consultar informação relativa a um percurso** – O jogador seleciona a opção de apresentar informação relativa ao percurso atualmente selecionado e o jogo apresenta a respetiva informação numa nova janela gráfica.

- **UC5: Consultar informação de uma carta da coleção** – O jogador seleciona uma das cartas da sua coleção e o sistema mostra na sua interface uma versão graficamente aumentada da carta com o título do objeto que representa, a fotografia capturada pelo jogador e, na parte de trás da carta, a respetiva descrição em formato de texto.
- **UC6: Iniciar jogo relativo a uma carta não capturada** – Após a seleção de um percurso, o jogador seleciona a opção de iniciar jogo, o sistema inicia o quiz visual apresentando a questão relativa à primeira carta bloqueada do percurso escolhido, disponibilizando a opção do jogador responder à questão.
- **UC7: Responder à questão do jogo com fotografia** – Após o sistema apresentar a questão de quiz, quando o jogador seleciona a opção de responder, o jogo inicia a câmara do telemóvel permitindo ao jogador capturar uma fotografia do objeto que considera responder à questão apresentada. O sistema procede à correção da resposta em formato de fotografia do jogador, dando de seguida o respetivo feedback (informando da resposta errada ou informando da resposta correta e da respetiva adição da carta à coleção do jogador).
- **UC8: Transformar carta colecionada numa carta especial** – Após selecionar uma das cartas da sua coleção, o jogador seleciona a opção de transformar a carta numa carta especial. De seguida partilha um código QR com outro jogador que possua a mesma carta ou realiza a leitura do código QR do outro jogador em questão. Após o sistema validar o código, apresenta ao jogador a sua carta especial, atualizando de igual forma a sua coleção.
- **UC9: Recolher prémio das cartas especiais** – Quando o jogador consegue obter 100% das cartas especiais do jogo, é-lhe apresentado no menu principal uma nova opção de obter um prémio. Quando selecionada, a opção premeia-o com uma mensagem que contém um código promocional relativo a um serviço turístico.

Relativamente aos **requisitos não-funcionais**, destacam-se os seguintes aspetos:

- **Implementação** – Dado que solução pretendida constitui um jogo para o apoio de atividades de turismo, é esperado que seja executado numa plataforma móvel, sendo que no caso específico do projeto, é esperado que o jogo esteja disponível para a plataforma Android.
- **Desempenho** – Considera-se fulcral que o jogo apresente um bom desempenho em diferentes dispositivos móveis, de modo a manter um constante interesse sobre o jogo por parte do jogador. Por bom desempenho entende-se a ausência de longos períodos de espera entre as várias ações do jogo e os respetivos resultados, segundo a ótica do jogador.
- **Design** – Relativamente ao design da solução, pretende-se que o jogo apresente uma interface apelativa ao seu público-alvo, sendo de igual forma importante que apresente um design e uma estrutura de opções de jogo intuitiva, evitando ao máximo interromper o fluxo natural de jogo.

### 3.2.2. Mapa de Navegação

Durante o processo de desenvolvimento de uma solução tecnológica que envolva uma interface de contacto com um utilizador, mais concretamente na fase de planeamento, existe a necessidade da definição de uma estrutura de interfaces e possíveis ecrãs, associados às várias ações possíveis e respetivas consequências. Com o intuito de assegurar essa mesma solidez estrutural ao nível da interface e fluxo de utilização do jogo, procedeu-se à realização do mapa de navegação relativo ao jogo (Figura 25).

O mapa de navegação representado na Figura 25 encontra-se dividido em quatro segmentos diferentes:

- **Interfaces relativas ao próprio sistema de jogo** – Constitui o conjunto dos vários ecrãs e elementos da interface relativos a ações que não estão diretamente relacionadas com o tema sério ou mecânicas do jogo (ecrã inicial com o título do jogo, criação da conta de jogador, menu de opções principal e ecrã de informações gerais do jogo);
- **Interfaces relativas à coleção de cartas** – Conjunto das várias interfaces relativas à navegação na coleção virtual de cartas e os ecrãs que contêm informações relativas à própria coleção e cartas individuais;
- **Segmento do quiz visual** – Segmento composto pelos vários ecrãs relativos a todo o processo do quiz: o ecrã que apresenta a questão de quiz, a interface da câmara fotográfica que permite o jogador capturar a resposta à questão e o ecrã que apresenta o resultado e correção da resposta do jogador;
- **Segmento relativo à mecânica das cartas especiais** – Composto pelas várias interfaces relativas ao processo do jogador transformar as cartas da sua coleção em cartas especiais (ecrã principal do processo, leitura de códigos QR e ecrã que apresenta o resultado final do processo);

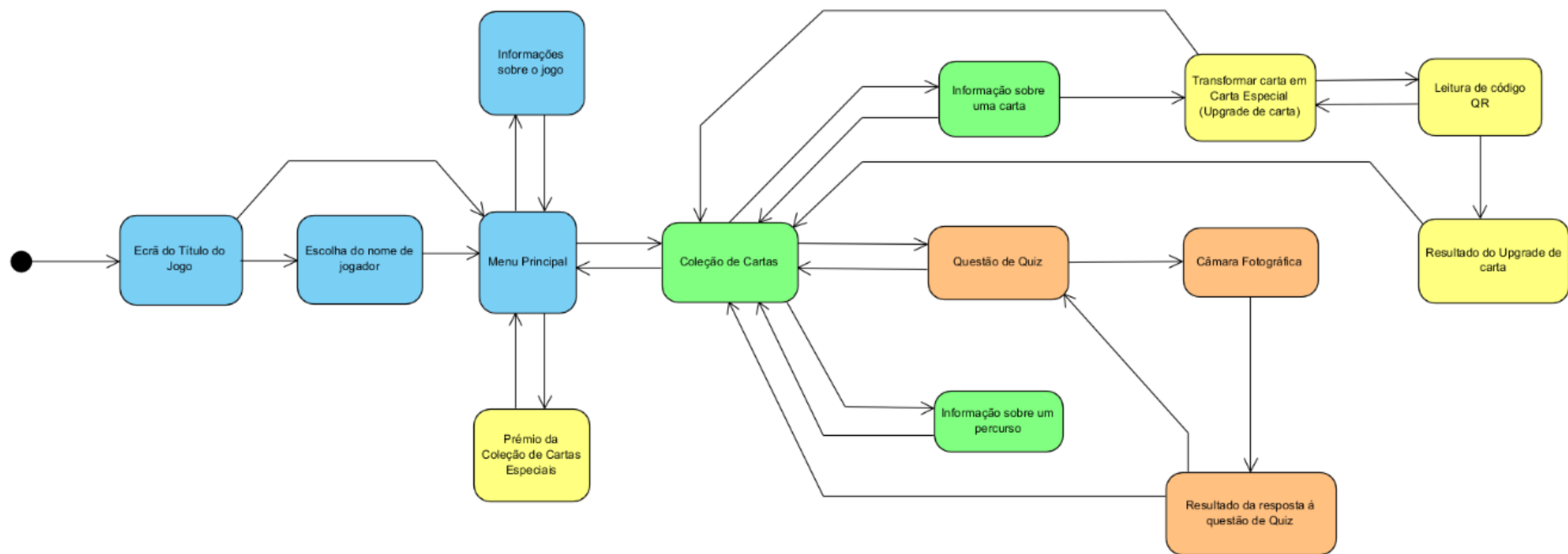


Figura 25 - Mapa de Navegação relativo ao jogo

Como se pode verificar no mapa de navegação, uma das preocupações durante o processo de desenho da estrutura e posição das interfaces no fluxo do jogo foi assegurar que o jogador possuísse, sempre que possível, a possibilidade de retornar a uma interface anterior, maximizando assim a eficiência das suas ações. Esse mesmo desejo de eficiência foi de igual forma o principal fator na decisão de agrupar interfaces em diferentes segmentos no fluxo de utilização de jogo, onde interfaces relativas a um mesmo propósito se encontram juntas no fluxo.

### 3.3. Arquitetura

De modo a assegurar a sua qualidade, qualquer solução tecnológica necessita de um planeamento cuidadoso da sua arquitetura, tanto ao nível da sua estrutura como ao nível dos recursos que lhe estão disponíveis, satisfazendo os vários requisitos estabelecidos de forma eficaz e eficiente.

Considerando a premissa referida, procedeu-se à definição arquitetura global do jogo com base nas relações e interações dos vários componentes envolvidos no sistema.

O processo de estruturação das relações entre componentes, no caso específico do planeamento do desenvolvimento do jogo associado ao projeto, representou um processo evolutivo, sendo a estrutura final o resultado de diversos aprimoramentos da estrutura inicialmente concebida, com base na introdução de novos requisitos, decididos como uma solução que melhor satisfaz e se adequa ao cenário de utilização do jogo projetado.

Inicialmente foi projetada a arquitetura representada no diagrama de componentes da Figura 26.

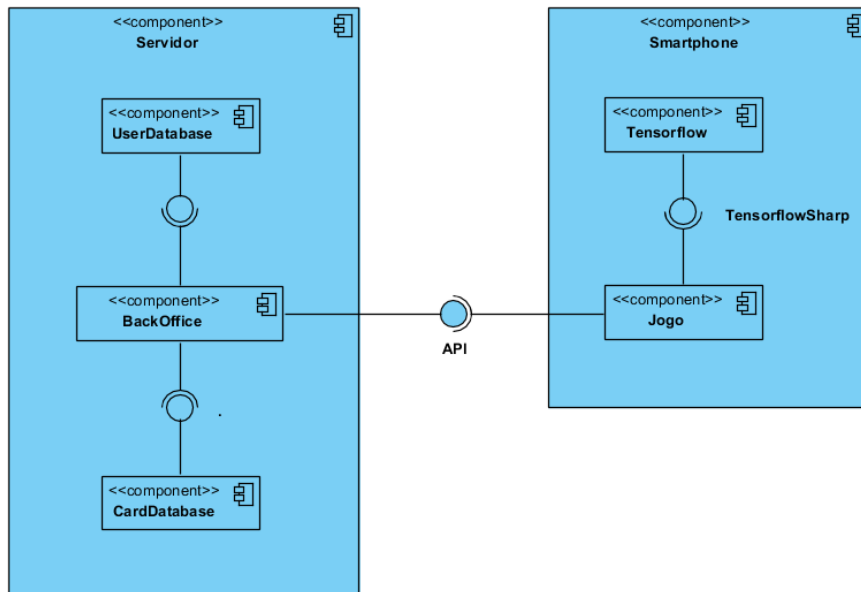


Figura 26 - Diagrama de componentes relativo à primeira arquitetura projetada para o jogo

Como se pode verificar no diagrama de componentes representado, o sistema geral do jogo encontrava-se dividido em dois componentes principais, o componente **smartphone** e o **servidor**.

Como especificado na listagem de requisitos não-funcionais (secção 3.2.1), projeta-se que o jogo seja executado em sistemas *Android* (*smartphone*), sendo este o componente onde se realiza todo o processamento diretamente relacionado com o jogo. No caso da arquitetura inicialmente concebida, o componente *smartphone* é constituído por dois componentes internos: componente que representa o jogo e o componente que representa a biblioteca *Tensorflow* (secção 2.2.2). O jogo acede à biblioteca referida a recorrendo ao *plugin TensorflowSharp*[43]. O *plugin* de *Unity* em questão permite a utilização de *Tensorflow* em projetos realizados em *Unity* (linguagem *C#*).

Relativamente ao componente servidor presente na primeira projeção da arquitetura, este constituiria um servidor remoto com a função de gerir e providenciar toda a informação relativa aos jogadores e conteúdo didático do jogo (informação das várias cartas) através de uma *API REST*. Tal funcionalidade seria apoiada através de um serviço de *backoffice* usado para a gestão de dados (tanto os dados do jogador como os dados relativos às várias cartas do jogo estariam armazenados numa base de dados dedicada).

A utilização desta primeira arquitetura implica o surgimento de uma nova condicionante: dado que toda a gestão dos dados do jogador e das várias cartas do jogo seria feita remotamente, sendo acedida através de uma chamada à API do

servidor remoto, existiria a necessidade de uma ligação constante à internet por parte do *smartphone* onde o jogo é executado.

Visto que é esperado que o jogador esteja em constante movimento durante as suas sessões de jogo, **considera-se mais viável uma solução arquitetural que minimize a necessidade de uma constante ligação a uma rede de internet.**

Com vista nesta nova necessidade, procedeu-se ao desenho de uma nova arquitetura de componentes (Figura 27).

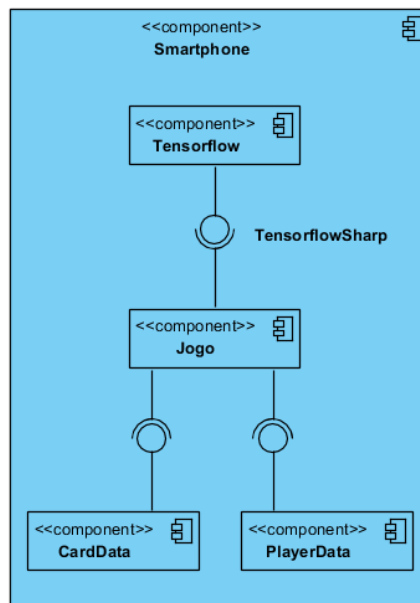


Figura 27 - Diagrama de componentes relativo à arquitetura final do jogo

De acordo com o novo desenho arquitetural representado pelo diagrama de componentes da Figura 27, todo o sistema encontra-se contido no *smartphone*, não sendo necessário qualquer comunicação com servidores externos para assegurar o funcionamento da maioria das funcionalidades de jogo. A única exceção deste princípio seria a funcionalidade das cartas especiais, conforme descrito na secção 3.3.5..

A ausência de qualquer comunicação com servidores remotos implica que a hospedagem e tratamento dos dados de jogador e cartas ocorra internamente na aplicação de jogo. Tal implica que o jogo possua um sistema de persistência e gestão de todos os dados.

### 3.3.1. Estrutura e Organização do Projeto em Unity

O processo de desenvolvimento de projetos realizados em *Unity* é caracterizado pela utilização de um **sistema de *scripts***. O sistema de *scripts* em *Unity* rege-se segundo a linguagem C#, sendo que cada *script* representa um conjunto de funcionalidades relativas a um objeto de jogo. No caso específico do projeto, todos os *scripts* correspondentes às várias funcionalidades encontram-se atribuídos aos objetos de canvas (interfaces de utilização) correspondentes (Figura 28).

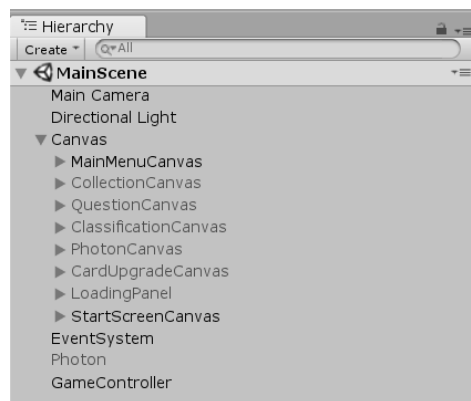


Figura 28 - Raiz da árvore hierárquica do projeto de Unity, com os vários objetos canvas representados

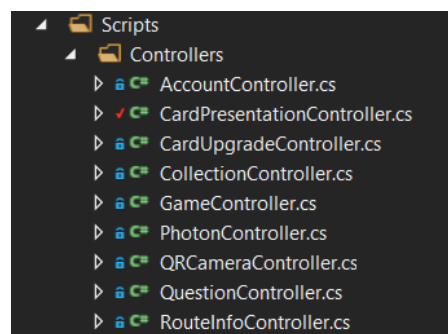


Figura 29 - Conjunto dos principais scripts Controller

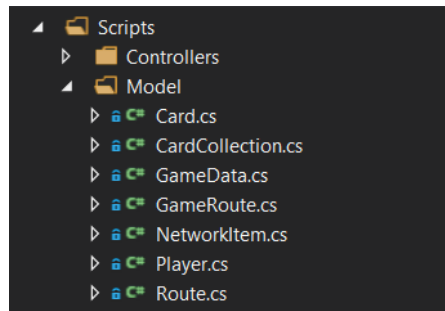


Figura 30 - Conjunto dos scripts Model

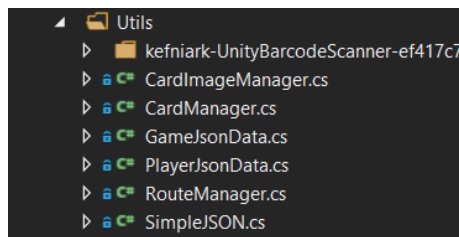


Figura 31 - Conjunto dos scripts Utils

Todas as principais funcionalidades e respetiva gestão das interfaces de jogo são realizadas a partir de um sistema de *scripts Controller* (Figura 29). Estando associados aos respetivos canvases, estes *scripts* são responsáveis por todo o funcionamento do sistema e controlo de informação de jogo, servindo de igual forma como ponto de ligação entre as várias classes *Model* (utilizados na serialização e instanciação de dados) (Figura 30) e as interfaces de utilização. Com o intuito de fornecer algoritmos responsáveis por controlar ações específicas de comportamento do sistema e interface do jogo, também foram desenvolvidos os *scripts Utils* (Figura 31).

Com o objetivo de controlar toda a informação do jogo, foi criado o objeto *GameController*, sendo-lhe atribuído o script de igual nome (Figura 32).

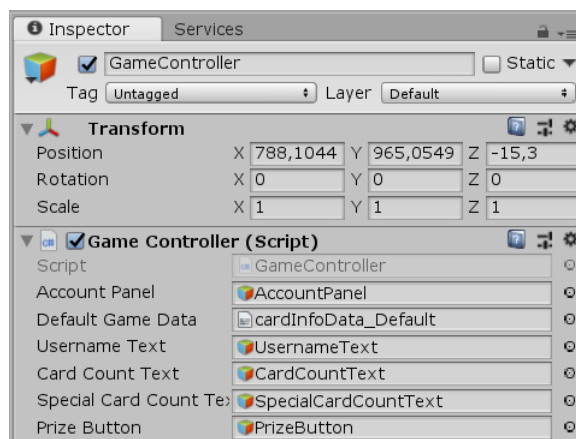


Figura 32 - Objeto GameController do projeto de Unity do jogo

Recorrendo ao script, o objeto *GameController* tem como função gerir toda a informação relativa ao jogo: gestão do dados e conteúdo didático, sistema de gestão do perfil do jogador e a respetiva persistência.

### 3.3.2. Persistência e Sistema de Dados

Conforme referido, com base no novo requisito de minimizar a necessidade do jogo de uma ligação constante à internet, procedeu-se ao desenvolvimento de um sistema de persistência dos vários dados de jogo.

Dado que a persistência ocorre localmente, é primordial um sistema dinâmico e prático de leitura e escrita de dados. De forma a colmatar o problema, foi projetado uma estrutura de dados com base em ficheiros de **formato JSON** (*JavaScript Object Notation*[44]). Este formato, além apresentar um elevado suporte e de se encontrar altamente padronizado, facilita o processo de desenvolvimento, dado que é legível por humanos.

Os dois sistemas de dados são:

- **Sistema de dados do jogo** – O sistema em questão visa gerir todos os dados de jogo (gestão dos vários percursos, informação de cartas, as questões de quiz relativas a cada carta).
- **Sistema de dados relativos ao jogador** – O sistema responsável pela persistência de toda a informação relativa ao próprio jogador (dados de identificação e registo da sua coleção virtual de cartas)

Em conformidade com o referido na secção 3.3, o principal script responsável por toda a gestão de dados do jogo constitui o script *GameController.cs*. O método do script diretamente incumbido da gestão é o *managePlayerData* (Figura 33)

```
37 public void managePlayerData()
38 {
39     //Checks if the system already contains a saved gamedata file
40     if (System.IO.File.Exists(cardInfoDataPath)) //if the system already contains a gamedata file, it proceeds to load it
41     {
42         this.gamedata = this.fileManager.loadGameData();
43     }
44     else //if not, the system loads a default gamedata file from resources, loads it and saves it
45     {
46         this.gamedata = this.fileManager.loadDefaultGameData(this.defaultGameData.text);
47         this.fileManager.saveGameData(this.gamedata);
48     }
49     //check is the system already contains a player savefile
50     if (System.IO.File.Exists(playerDataJsonPath))
51     {
52         Debug.Log("PLAYER DATA FOUND");
53         //Load Player data
54         this.player = fileManager.loadPlayerData();
55         Debug.Log("USERNAME: " + this.player.getUsername());
56         Debug.Log("ROUTE COUNT: " + this.player.getCollection().getRouteList().Count);
57     }
58     fillPlayerInfo();
59     this.accountPanel.SetActive(false);
60 }
61 else
62 {
63     Debug.Log("SYSTEM DIDN'T FIND ANY PLAYER DATA");
64     this.accountPanel.SetActive(true); //sets the player account creation screen visible
65 }
66 }
67 }
```

Figura 33 - Método *managePlayerData* presente no script *GameController.cs*

Como se pode verificar no método *menagePlayerData*, toda a leitura e escrita de ficheiros JSON do sistema é realizada recorrendo ao objeto *fileManager*. Este objeto constitui uma instanciação da classe *JsonFileManager*. A classe em questão foi criada com o propósito de abstrair o script *GameController.cs* de todas as operações de leitura e escrita de ficheiros do sistema, sendo que na sua inicialização, são-lhe passados por parâmetro os caminhos de leitura/gravação do ficheiro de dados de jogo e do ficheiro de dados do jogador (os vários caminhos estão definidos como o diretório raiz de instalação da aplicação de jogo).

### 3.3.2.1. Sistema de Dados do Jogo

Previamente ao desenvolvimento do sistema de dados do jogo, existe a necessidade de identificar os vários tipos de dados que são necessários persistir localmente, assim como a sua natureza, de modo a definir uma estrutura prática e eficiente.

De modo a albergar todos os dados necessários ao jogo numa mesma estrutura, foi definido o sistema de dados representado na Figura 34.

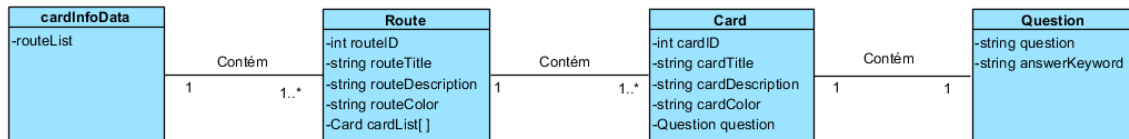


Figura 34 - Modelo de domínio relativo à estrutura de dados do jogo no ficheiro JSON

Segundo a estrutura definida, árvore de dados JSON tem como raiz o objeto **cardInfoData** e apresenta a seguinte estrutura:

- **routeList** – Constitui o conjunto da totalidade de percursos (*Routes*) presentes na coleção virtual do jogo. Cada percurso apresenta um conjunto de vários atributos que o classificam:
  - **routeID** – Valor de identificação numérica utilizado para representar globalmente o percurso;
  - **routeTitle** – Título que identifica textualmente o percurso;
  - **routeDescription** – Informação textual que descreve e apresenta informação do percurso;
  - **routeColor** – Representa uma cor em formato hexadecimal. O atributo em questão é utilizado pelo jogo para colorir o elemento da interface gráfica que representa o percurso (cor do botão gráfico de seleção).
  - **cardList** – Representa o conjunto das várias cartas de jogo (*Cards*) relativas ao percurso em questão. De igual forma, cada entidade *Card* apresenta um conjunto de atributos que a representam:
    - **cardID** – Valor de identificação numérica utilizado para representar globalmente a carta de jogo.
    - **cardTitle** – Título que identifica textualmente o título da carta (e consequentemente do objeto que esta representa);
    - **cardDescription** - Informação textual que descreve e apresenta informação da carta (constitui de igual forma a informação didática relativa ao objeto que a carta visa representar);
    - **cardColor** – À semelhança do atributo *routeColor*, o atributo *cardColor* representa a partir do formato hexadecimal a cor que a carta apresenta graficamente no jogo.
    - **question** – Dado que cada carta é obtida com base numa reposta de quiz correta, é necessária uma estrutura de dados que contenha toda a informação necessária para apresentar uma questão de quiz, assim como para a correção da respetiva

resposta. Cada entidade *question* encontra-se diretamente associada a uma carta única, contendo os seguintes atributos:

- **question** – Informação textual que representa diretamente a questão de quiz associada à carta
- **answerKeyword** – Informação textual que representa a resposta à questão. O atributo é utilizado no processo de correção da resposta visual do jogador.

Relativamente aos dados de jogo (informação relativa aos vários percursos e cartas de jogo), o sistema inicia o processo verificando se existe algum ficheiro *JSON* (*cardInfoData.json*) gravado no dispositivo que contenha a informação do jogo. No caso positivo, o sistema procede à leitura do ficheiro e respetiva instanciação dos dados. Em caso negativo (correspondendo este caso à primeira inicialização da aplicação de jogo por parte do jogador), o sistema instancia os dados a partir de uma versão pré-definida do ficheiro (***cardInfoData\_Default.json*** (Anexo A – Ficheiro *cardInfoData\_Default.json*)), gravando-o de seguida no dispositivo (o ficheiro é gravado com o nome “*cardInfoData.json*”). Todo este processo (representado na Figura 33 entre as linhas 40 e 48) é responsável pela instanciação dos dados diretamente relacionados com o funcionamento do jogo.

A instanciação dos dados de jogo é realizada a partir do objeto ***GameJsonData*** (Anexo B – Objeto *GameJsonData*) e segue a estrutura idealizada para os dados do jogo (Figura 34).

### 3.3.2.2. Sistema de Dados do Jogador

Identicamente ao sistema encarregue pela gestão de dados do jogo, existe uma igual necessidade de gerir e persistir os dados relativos ao jogador.

Relativamente à informação do jogador, o principal objetivo da sua persistência é salvaguardar a sua coleção virtual de cartas em todas as suas sessões de jogo.

No caso dos dados do jogador, a sua serialização em formato *JSON* (Anexo C – Exemplo do ficheiro *PlayerData.json*) é realizada a partir da serialização do objeto ***PlayerJsonData*** (Anexo D – Objeto *PlayerJsonData*), possuindo a seguinte estrutura:

- **username** – Nome identificativo do jogador (escolhido na primeira sessão de jogo);

- **collection** – Representa toda a informação relativa à coleção virtual do jogador. O atributo constitui uma lista de percursos do qual o jogador possui cartas. Cada percurso contém os atributos:
  - **routeID** – Identificador numérico do percurso;
  - **routeDescription** – Informação textual que descreve o percurso;
  - **cardList** – Conjunto das várias cartas previamente colecionadas, presentes na coleção virtual do jogador. Cada elemento da lista possui como atributo:
    - **cardID** – Identificador numérico da carta do jogador;
    - **isSpecial** – Variável booleana que representa o estado da carta relativo à sua classe na coleção. Se possuir o valor “true”, a carta representa uma carta especial presente na coleção do jogador. No caso do valor “false”, constitui uma carta normal.

De modo análogo ao processo de persistência e controlo dos dados de jogo, o processo de gestão dos dados do jogador ocorre também no método **managePlayerData** (representado entre as linhas 50 e 66 da Figura 33). O processo é iniciado pelo sistema a verificar se existe previamente gravado no dispositivo o ficheiro *JSON* relativo aos dados do jogador (**playerData.json**). Se for o caso, recorrendo ao método respetivo de *JsonFileManager*, procede-se à leitura do ficheiro e instanciação do objeto responsável por albergar os dados do jogador. No caso de o ficheiro não existir, o sistema inicia a interface de criação de conta do jogador (Figura 35) requisitando a escolha de um nome de jogador.

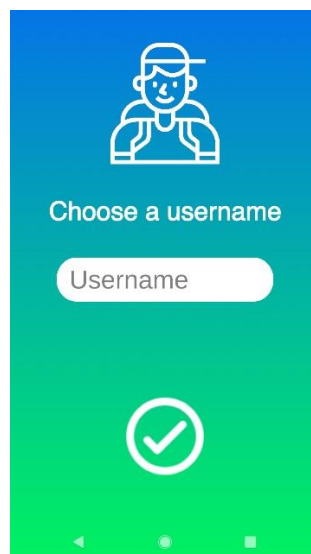


Figura 35 - Interface de criação de conta de jogador

Ao contrário do processo de gestão dos dados de jogo, em que é utilizado o mesmo objeto na instanciação e serialização (*GameJsonData*), os dados do jogador são instanciados num objeto dedicado (Anexo E – Objeto *Player.cs*) atributo do script *GameController.cs*, possibilitando o acesso e atualização dos dados do jogador por qualquer script do jogo durante a execução através de vários métodos.

Finalizada a instanciação do objeto com os dados relativos ao jogador, o sistema procede ao *display* dos vários dados através do menu principal do jogo (Figura 36). Tal é conseguido através da invocação do método *fillPlayerInfo* (Figura 37), que recorrendo aos dados presentes no objeto *Player*, preenche o elemento gráfico correspondente ao nome do jogador e calcula o número de cartas totais e especiais do jogador.

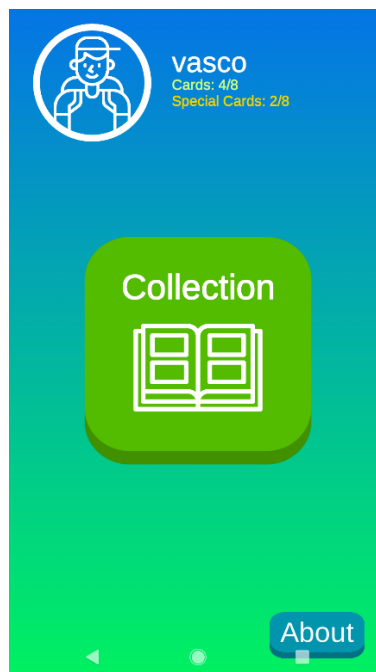


Figura 36 - Menu Principal do jogo com informação relativa ao jogador

```
92 //fills all the player info in the the game's main screen
93 void fillPlayerInfo()
94 {
95     int totalCards = this.gamedata.totalCards();
96     int totalPlayerCards = this.player.totalPlayerCards();
97     int totalPlayerSpecialCards = this.player.totalPlayerSpecialCards();
98
99     if(totalPlayerSpecialCards == totalCards)
100     {
101         this.prizeButton.SetActive(true);
102     }
103
104     this.usernameText.GetComponent<Text>().text = this.player.getUsername();
105     this.cardCountText.GetComponent<Text>().text = "Cards: "+totalPlayerCards + "/" + totalCards;
106     this.specialCardCountText.GetComponent<Text>().text = "Special Cards: " + totalPlayerSpecialCards + "/" + totalCards;
107 }
```

Figura 37 - Método *fillPlayerInfo* do script *GameController.cs*

O método *fillPlayerInfo* também é responsável pela gestão associada à atribuição de um prémio ao jogador com base no número de cartas especiais que possui (secção 3.1.3). Ao verificar se o número de cartas especiais do jogador é igual ao número total de cartas do jogo (linha 99 da Figura 37), em caso positivo, é ativado no menu principal o botão (Figura 38) que permite ao jogador aceder à interface do prémio (Figura 39).

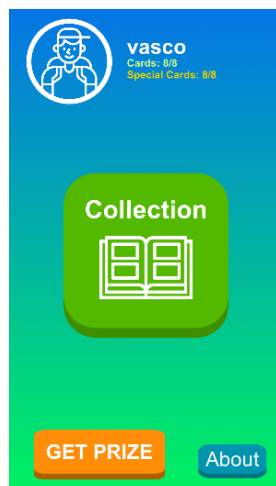


Figura 38 - Menu Principal do jogo com o botão de prémio desbloqueado

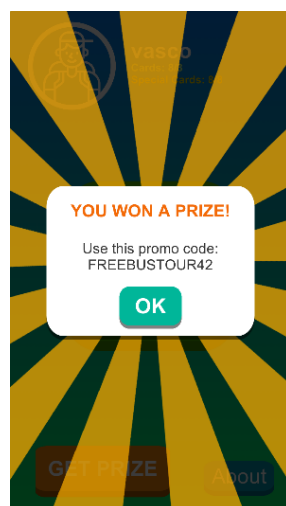


Figura 39 - Interface relativa ao Prémio das cartas especiais

### 3.3.3. Sistema de Coleção Virtual de Cartas

Dado que um dos principais focos do projeto é proporcionar ao jogador a experiência de possuir uma coleção virtual de cartas, é fulcral a existência de uma interface que permita ao jogador interagir de forma intuitiva com todas as suas cartas e aceder a toda a sua informação. Considerado o referido, o sistema encarregue da gestão da coleção do jogador necessita de ter acesso a todos os dados do jogo relativos à informação das cartas de jogo assim como os dados diretamente relacionados com o jogador (para que consiga reconstruir a coleção pessoal do jogador no início de todas as suas sessões de jogo).

Especificamente ao projeto, o objeto responsável pela apresentação e gestão da coleção virtual de cartas do jogador é o **CollectionCanvas** (Figura 28), acompanhado do respetivo script **CollectionController.cs** que lhe é atribuído. A ativação da interface em questão é resultante do evento do botão relativo à coleção presente no menu principal de jogo (botão “Collection” representado na Figura 36).

```

103 void presentCards(GameRouteData route)
104 {
105     foreach (GameCardData gc in route.cardList)
106     {
107         Card playerCard = this.gameController.GetComponent<GameController>().getPlayerCard(gc.cardID);
108
109         GameObject cardObj = Instantiate(cardButtonPrefab, cardContent.transform);
110         cardObj.transform.Find("Text").gameObject.GetComponent<Text>().text = gc.cardTitle;
111         cardObj.GetComponent<Image>().color = TextureTools.ToColor(gc.cardColor);
112         try
113         {
114             //Debug.Log("PLAYER CARD IS NOT NULL");
115             if (!playerCard.isSpecialCard()) //Player card is normal
116             {
117             }
118             else //Player card is special
119             {
120                 cardObj.GetComponent<CardManager>().upgradeCard();
121             }
122             loadCardImage(cardObj, gc);
123             cardObj.transform.Find("BlockedImage").gameObject.SetActive(false); //unlocks the card and turns it visible to the player
124             cardObj.GetComponent<Button>().interactable = true; //makes the card interactable
125             cardObj.GetComponent<CardManager>().setData(gc, this.mainCardPanel, playerCard.isSpecialCard());
126         }
127         catch (System.NullReferenceException e)
128         {
129             Debug.Log("BLOCKED CARD: " + gc.cardID);
130             this.blockedCardsList.Add(gc);
131         }
132     }
133 }
134

```

Figura 40 - Método presentCards do script CollectionController.cs

```

73 public void selectRoute()
74 {
75     this.blockedCardsList = new List<GameCardData>();
76     clearList(this.cardContent);
77     var currentEventSystem = EventSystem.current;
78     GameObject routeButton = currentEventSystem.currentSelectedGameObject;
79
80     //string routeTitle = routeButton.transform.Find("Text").gameObject.GetComponent<Text>().text;
81     GameRouteData routeData = routeButton.GetComponent<RouteManager>().getRouteData();
82     deselectOtherRouteButtons(routeData.routeID);
83     presentCards(routeData);
84     this.routeInfoPanel.GetComponent<RouteInfoController>().setupRouteInfo(routeData.routeTitle, routeData.routeDescription);
85 }

```

Figura 41 - Método selectRoute do script CollectionController.cs

Sendo o objetivo principal da interface de coleção disponibilizar de forma gráfica e explícita as várias cartas e percursos acompanhados da respetiva informação, os principais métodos responsáveis pelo display dos percursos e cartas constituem os métodos *presentCards* e *selectRoute* presentes no script *CollectionController.cs* (Figura 40 e Figura 41 respetivamente).

O método *selectRoute* tem como propósito responder ao evento do jogador de seleção do percurso, tanto para aceder à sua informação (Figura 43) como para o *display* das cartas que este contém. O processo de preenchimento da interface com a informação do percurso (linha 84 da Figura 41) é realizado através do método *setupRouteInfo* pertencente ao script *RouteInfoController.cs*, sendo-lhe passados por parâmetro os dados textuais relativos ao título e descrição do percurso. Relativamente ao processo de display das cartas pertencentes ao percurso, o método *selectRoute* recorre à invocação de *presentCards*.

O método *presentCards* é iniciado com uma iteração sobre todos os objetos que contêm a informação das cartas de jogo relativas ao percurso recebido por parâmetro, instanciando todas as cartas na interface com toda a informação associada (título, descrição, imagem da carta e respetiva verificação se a carta constitui uma carta especial), de modo a poder ser acedida quando o jogador seleciona uma carta (e apresentada com a cor correta no caso de se tratar de uma carta especial) (Figura 44 e Figura 45). De seguida, são desbloqueados graficamente no canvas de visualização todos os objetos de cartas pertencentes à coleção pessoal do jogador (Figura 42) (por definição os elementos gráficos correspondentes às cartas aparecem visualmente bloqueados).



Figura 42 - Menu de coleção de cartas do jogo

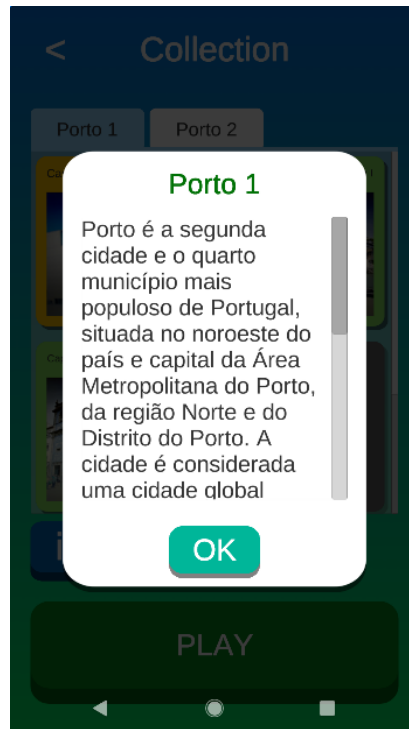


Figura 43 - Interface do jogo que disponibiliza ao jogador a informação relativa a um percurso

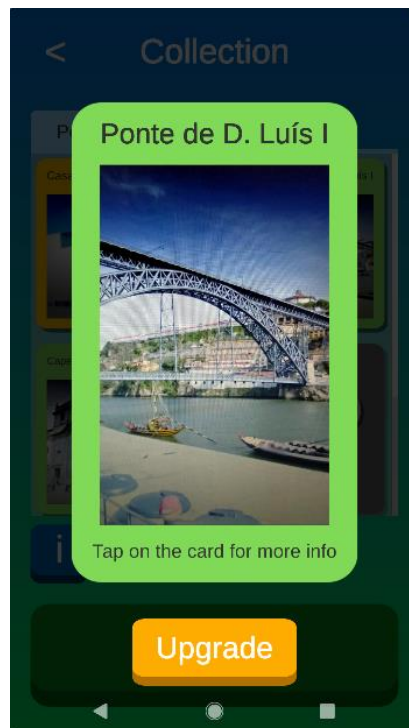


Figura 44 - Interface do jogo com informação de uma carta (parte frontal da carta)

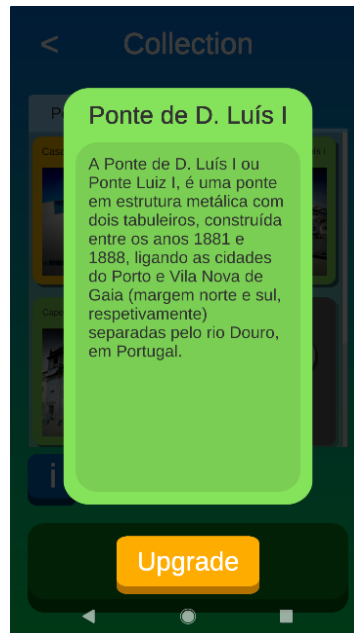


Figura 45 - Interface do jogo com informação de uma carta (parte traseira da carta)

### 3.3.4. Quiz Visual

No projeto associado ao jogo, o *script* que controla todo o processo de quiz visual constitui o **QuestionController.cs**. O *script* em questão tem como função controlar os principais processos relativos à interface e respetivo display gráfico da questão de quiz. A sua invocação e ativação da interface de utilização ao qual está associado é realizada através do método de resposta ao evento da seleção de início de jogo, o método **playGame** do *script* **CollectionController.cs** (Figura 46).

```

151 public void playGame()
152 {
153     if(this.blockedCardsList.Count > 0)
154     {
155         this.questionCanvas.SetActive(true);
156         this.questionCanvas.transform.GetComponent<QuestionController>().setupQuestion(this, chooseQuestion());
157     }
158     else //Player already completed the selected route collection
159     {
160         this.completedMessagePanel.SetActive(true);
161     }
162 }

```

Figura 46 - Método *playGame* presente no *script* **CollectionController.cs**

Ao verificar o número de cartas bloqueadas, o método *playGame* consegue determinar se o jogador já completou a coleção de cartas relativa ao percurso que escolheu. Caso o jogador já possua todas as cartas do percurso, não é iniciado o quiz e é ativada uma

interface a informá-lo (Figura 47). Caso contrário, a interface de quiz é inicializada e a partir do método **setupQuestion** presente no script *QuestionController.cs* (Figura 48), é enviado por parâmetro o objeto representativo da carta sobre o qual a questão vai incidir. O processo da escolha da carta da questão de quiz é realizado pelo método **chooseQuestion** (linha 156 da Figura 46), sendo que este devolve sempre a carta que se encontra em primeiro lugar na lista de cartas bloqueadas (cartas por colecionar).

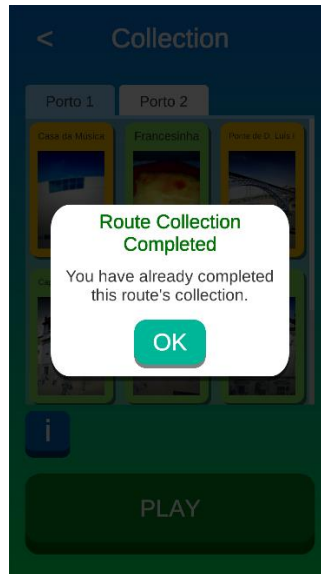


Figura 47 - Interface do jogo a informar o jogador da sua coleção completa

```
37 public void setupQuestion(CollectionController collectionController, GameCardData card)
38 {
39     this.collectionController = collectionController;
40     this.card = card;
41     this.questionText.GetComponent<Text>().text = card.cardQuestion.question;
42 }
```

Figura 48 - Método *setupQuestion* presente no script *QuestionController.cs*

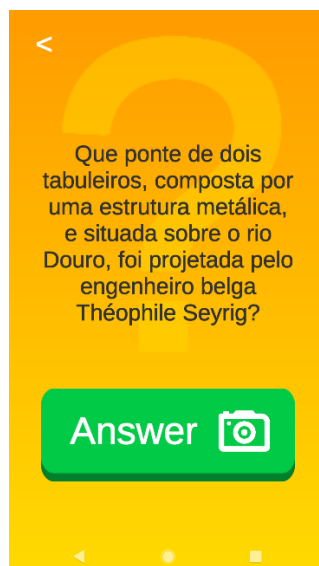


Figura 49 - Interface de jogo relativa à apresentação da questão do quiz visual

O método *setupQuestion* inicializa os dados relativos à carta e preenche o elemento gráfico da interface relativo à questão de quiz (Figura 49).

À mecânica de quiz visual está associado a utilização de um sistema de classificação automática de imagens de modo a permitir a correção das respostas visuais (fotografias) do jogador (secção 3.1.2).

Conforme analisado na secção 2.2.2, o processo de desenvolvimento de um sistema capaz de identificar uma imagem segundo um conjunto pré-definido de classes é composto por duas atividades:

- **Treino de um modelo de classificação de imagem;**
- **Desenvolvimento de um classificador que utilize o modelo de classificação;**

#### 3.3.4.1. Modelo de Classificação

Dado que todo o processo de classificação tem por base a atribuição de uma probabilidade de uma imagem de input pertencer a cada classe de um conjunto de classes de imagem, como primeiro passo, existe a necessidade de definir concretamente as várias classes que se pretendem identificar. No contexto do protótipo de jogo associado ao projeto, cada classe de imagem representa diretamente um monumento ou objeto de valor cultural, sendo que foram definidas **8 classes de identificação**:

- **Capela das Almas;**
- **Casa da Música;**
- **Estátua de D. Pedro IV**
- **Francesinha;**
- **Igreja dos Congregados;**
- **Ponte D. Luís;**
- **Sé Catedral do Porto;**
- **Torre dos Clérigos.**

Todas as classes definidas representam monumentos/objetos de valor cultural alusivos à cidade do Porto, sendo que se encontram num raio geográfico de fácil acesso entre os vários pontos.

Após a definição das várias classes, procedeu-se à construção dos vários *datasets* necessários ao processo de treino do modelo.

No caso específico do projeto, os *datasets* representam diretórios constituídos por várias centenas de imagens do objeto retratado pela classe em questão (exemplificado na Figura 5).

Cada classe contém um *dataset* com uma média de **146 imagens** representativas.

Durante o processo de construção dos *datasets*, uma das principais preocupações foi assegurar uma grande variedade de imagens para cada classe tanto ao nível de formatos de imagem, resoluções, iluminações e ângulo de visualização do objeto representado. Quanto maior for a variedade das características mencionadas, mais eficaz será o classificador a identificar as fotografias do jogador, dada a volatilidade das condições, tanto do meio em que o objeto é fotografado como do dispositivo que o jogador utiliza para jogar e conseqüentemente fotografar.

Posteriormente à definição e construção dos vários *datasets*, foi iniciado o processo de treino do modelo de classificação. Para tal, foi utilizado o script *retrain.py* [45].

Disponível no repositório oficial da biblioteca *Tensorflow*, o script *retrain.py* tem como propósito retreinar um modelo *MobileNet*[46]. *MobileNet* constitui um conjunto de vários modelos ImageNet[47] pré-treinados (modelos preparados para identificar cerca de 1000 classes diferentes), focado numa maior eficiência e orientado a sistemas móveis. O processo de retreinar o modelo *MobileNet* consiste na adição e treino de uma nova camada de topo (secção 2.2.2), de modo a permitir o reconhecimento das novas classes que se pretendem classificar (no contexto do projeto, correspondem às 8 classes definidas).

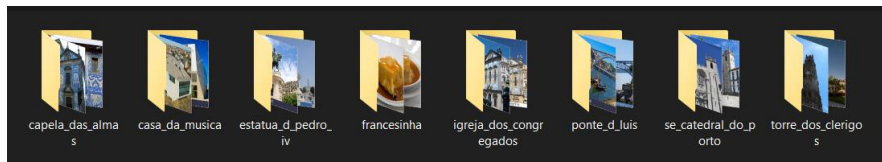


Figura 50 - Conteúdo do diretório "Monuments"

```

py .\scripts\retrain.py \
  --image_dir=Monuments \
  --bottleneck_dir=tf_files/bottlenecks \
  --model_dir=tf_files/models/ \
  --summaries_dir=tf_files\training_summaries \
  --output_graph=tf_files\retrained_graph.pb \
  --output_labels=tf_files\retrained_labels.txt \
  --how_many_training_steps=10000

```

Figura 51 - Execução em terminal do script retrain.py

Na execução em terminal do script *retrain.py* (Figura 51) destacam-se os seguintes argumentos:

- **image\_dir** – Indica o diretório que contém os datasets de imagens (neste caso, foi utilizado o diretório "Monuments" (Figura 50));
- **output\_graph** – Indica o diretório destino e nome do ficheiro do modelo de treino resultante do processo;
- **output\_labels** – Indica o diretório e nome do ficheiro de texto que contém os nomes das várias classes (o algoritmo presente no script automaticamente atribui os nomes das várias classes com base no nome dos subdiretórios de *image\_dir* e cria o ficheiro);
- **how\_many\_training\_steps** – Representa o número de iterações de treino durante o processo. Foram definidas 10000 iterações de treino.

```

INFO:tensorflow:2019-07-03 22:26:54.982615: Step 9970: Train accuracy = 100.0%
INFO:tensorflow:2019-07-03 22:26:54.982615: Step 9970: Cross entropy = 0.016868
INFO:tensorflow:2019-07-03 22:26:55.060723: Step 9970: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2019-07-03 22:26:55.872995: Step 9980: Train accuracy = 100.0%
INFO:tensorflow:2019-07-03 22:26:55.872995: Step 9980: Cross entropy = 0.007379
INFO:tensorflow:2019-07-03 22:26:55.951136: Step 9980: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2019-07-03 22:26:56.763442: Step 9990: Train accuracy = 100.0%
INFO:tensorflow:2019-07-03 22:26:56.763442: Step 9990: Cross entropy = 0.008458
INFO:tensorflow:2019-07-03 22:26:56.841549: Step 9990: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:2019-07-03 22:26:57.560137: Step 9999: Train accuracy = 100.0%
INFO:tensorflow:2019-07-03 22:26:57.560137: Step 9999: Cross entropy = 0.006652
INFO:tensorflow:2019-07-03 22:26:57.638236: Step 9999: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:Final test accuracy = 98.3% (N=234)

```

Figura 52 - Output final da execução do script retrain.py

No final do processo de treino, imediatamente antes de terminar a execução do *script*, é calculada a **taxa de precisão de teste final**, sendo obtido o valor **98.3%** (Figura 52), e é gravado no diretório indicado o ficheiro relativo ao modelo de classificação (*retrained\_graph.pb*) e o ficheiro de texto com os nomes das classes relativas ao treino (*retrained\_labels.txt*).

#### 3.3.4.2. Classificador de Imagens

Conforme referido anteriormente, o processo de correção das respostas de quiz visual por parte do jogo, implica um sistema capaz de identificar as fotografias capturadas pelo jogador, de modo a averiguar se esta se insere na classe relativa à questão. Recorrendo ao modelo de classificação treinado segundo as classes de cartas definidas, existe a necessidade de integrar no projeto de *Unity* um classificador automático de imagens (secção 2.2.2).

Iniciados através do evento relativo ao botão de resposta (Botão “Answer” representado na Figura 49), os objetos responsáveis pelo funcionamento da câmara e processo de classificação de imagem são respetivamente o ***ClassificationCanvas*** e o *script* ***PhoneCamera.cs*** que lhe está associado (Figura 53).

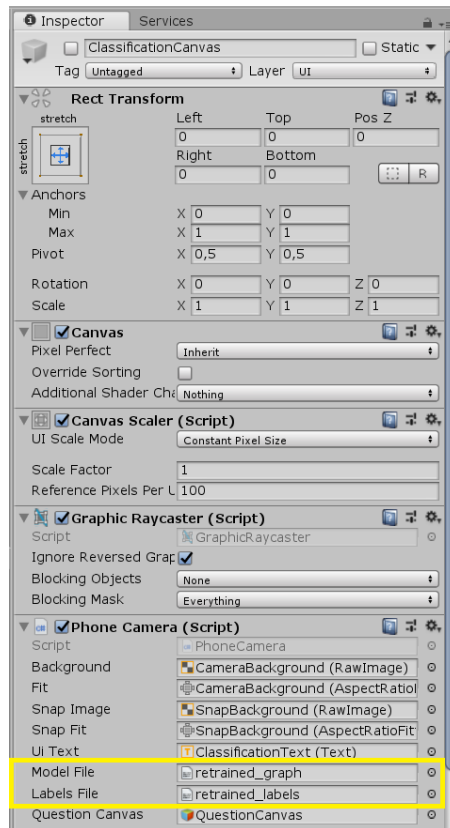


Figura 53 - Visualização em Inspector do Unity do objeto ClassificationCanvas

Dado que o processo de resposta do jogador implica a captura de fotografia, é necessário ao sistema o acesso à câmara do dispositivo onde o jogo é executado. O método **Start** (Figura 54) do script *PhoneCamera.cs* (executado automaticamente na ativação de *ClassificationCanvas*) visa a configuração e inicialização da câmara traseira do dispositivo. Para tal é utilizado o objeto **WebCamTexture**[48], constituinte do sistema *Unity* com o propósito de fornecer texturas com uma *feed* de vídeo em direto. Atribuindo a textura ao objeto de imagem presente na interface *ClassificationCanvas*, o jogador tem acesso à pré-visualização da imagem captada pela câmara do seu dispositivo.

Também é no método *Start* que, através da execução do método **LoadClassifier** (Figura 55), é inicializado o objeto **Classifier**. Criado por Andrey Ashikhmin[49], *Classifier* tem como propósito disponibilizar funcionalidades de classificação através do **TensorFlowSharp** (secção 3.3). Na sua inicialização, são-lhe passados por parâmetro os objetos relativos ao modelo de classificação e ficheiro de *labels*. Tais ficheiros representam o resultado do processo de treino (secção 3.3.4.1), sendo que foram adicionados aos *Resources* do projeto de *Unity* e enviados por parâmetro para o script *PhoneCamera.cs* a partir do sistema *Inspector* do *Unity* (assinalado a amarelo na Figura 53).

```
30 // Use this for initialization
31 public void Start () {
32     LoadClassifier();
33     defaultBackground = background.texture;
34     WebCamDevice[] devices = WebCamTexture.devices;
35     if(devices.Length == 0)
36     {
37         Debug.Log("No camera detected");
38         camAvailable = false;
39         return;
40     }
41     for(int i = 0; i < devices.Length; i++)
42     {
43         Debug.Log("New Camera");
44         if (!devices[i].isFrontFacing)
45         {
46             backCam = new WebCamTexture(devices[i].name, Screen.width, Screen.height);
47         }
48     }
49     if(backCam == null)
50     {
51         Debug.Log("Unable to find back camera");
52         return;
53     }
54     backCam.Play();
55     background.texture = backCam;
56     camAvailable = true;
57 }
58 }
```

Figura 54 - Método Start presente no script PhoneCamera.cs

```
136 private void LoadClassifier()
137 {
138     try
139     {
140         this.classifier = new Classifier(
141             this.modelFile.bytes,
142             Regex.Split(this.labelsFile.text, "\n|\r|\r\n")
143                 .Where(s => !String.IsNullOrEmpty(s)).ToArray(),
144             classifyImageSize);
145     }
146     catch (TFException ex)
147     {
148         if (ex.Message.EndsWith("is up to date with your GraphDef-generating binary.."))
149         {
150             this.uiText.text = "Error: TFException. Make sure you model trained with same version of TensorFlow as in Unity plugin.";
151         }
152     }
153     throw;
154 }
155 }
```

Figura 55 - Método LoadClassifier presente no script PhoneCamera.cs

A atualização no tempo da imagem da câmara projetada ao jogador é possível graças ao método **Update** (Figura 56). Sendo executado automaticamente pelo sistema por cada *frame* da câmara, o método *Update* aplica várias transformações à orientação, escala e rácio de resolução da textura de visualização presente na interface, assegurando constantemente uma correta pré-visualização, isenta de deformidades.

```

83 // Update is called once per frame
84 void Update () {
85     if (!camAvailable)
86     {
87         return;
88     }
89
90     if(backCam.width < 100)
91     {
92         return;
93     }
94
95     float ratio = (float)backCam.width / (float)backCam.height;
96     fit.aspectRatio = ratio;
97     snapFit.aspectRatio = ratio;
98
99     int orientation = -backCam.videoRotationAngle;
100    background.rectTransform.localEulerAngles = new Vector3(0, 0, orientation);
101    snapImage.rectTransform.localEulerAngles = new Vector3(0, 0, orientation);
102
103    float scaleY = backCam.videoVerticallyMirrored ? -1f : 1f;
104
105    background.rectTransform.localScale = new Vector3(1f*(1/ratio), scaleY*(1/ratio), 1f);
106    snapImage.rectTransform.localScale = new Vector3(1f * (1 / ratio), scaleY * (1 / ratio), 1f);
107 }

```

Figura 56 - Método Update presente no script PhoneCamera.cs

Tendo uma pré-visualização da câmara, o processo de resposta do jogador ao quiz é finalizado com a captura fotográfica do objeto ao qual este considera responder corretamente à questão. Ao ser selecionada a opção de captura fotográfica, através do evento associado ao botão da interface, é executado o método **SnapPhoto** (Figura 57).

```

109 public void SnapPhoto()
110 {
111     this.backCam.Pause();
112
113     float ratio = (float)backCam.width / (float)backCam.height;
114     snapFit.aspectRatio = ratio;
115
116     int orientation = -backCam.videoRotationAngle;
117     snapImage.rectTransform.localEulerAngles = new Vector3(0, 0, orientation);
118
119     float scaleY = backCam.videoVerticallyMirrored ? -1f : 1f;
120     snapImage.rectTransform.localScale = new Vector3(1f * (1 / ratio), scaleY * (1 / ratio), 1f);
121
122     Texture2D snappedPhoto = new Texture2D(backCam.width, backCam.height);
123     snappedPhoto.SetPixels((backCam as WebCamTexture).GetPixels());
124     snappedPhoto.Apply();
125
126     snapImage.texture = snappedPhoto;
127
128     TFClassify(snappedPhoto, ratio, orientation, scaleY);
129 }

```

Figura 57 - Método SnapPhoto presente no script PhoneCamera.cs

Na execução do método **SnapPhoto** é gravada a textura relativa ao *frame* do instante em que o jogador premiu o botão de captura e iniciado o processo de classificação da imagem. O processo tem por base a execução do método **TFClassify**.

```

157 private async void TFClassify(Texture2D snap, float ratio, float orientation, float scaleY)
158 {
159
160     var scaled = Scale(snap, classifyImageSize);
161     var rotated = await RotateAsync(scaled.GetPixels32(), scaled.width, scaled.height);
162
163     try
164     {
165         var probabilities = await this.classifier.ClassifyAsync(rotated);
166         this.questionCanvas.GetComponent<QuestionController>().checkAnswer(probabilities[0], snap, ratio, orientation, scaleY);
167     }
168     catch (NullReferenceException)
169     {
170         this.uiText.text = "Error: NullReferenceException. Make sure you set correct INPUT_NAME and OUTPUT_NAME";
171     }
172     finally
173     {
174         Destroy(snap);
175         Destroy(scaled);
176     }
177 }

```

Figura 58 - Método TFClassify presente no script PhoneCamera.cs

Após aplicadas várias transformações à imagem recebida por parâmetro, recorrendo ao objeto *Classifier* previamente mencionado, ocorre o processo de classificação, sendo devolvido o vetor **probabilities**, que contém todos os valores de probabilidade da imagem analisada pertencer a cada uma das 8 classes definidas.

Considerando que o vetor se encontra organizado segundo uma **ordem decrescente de probabilidades**, o primeiro elemento do vetor corresponde à classe à qual a imagem tem a maior probabilidade de pertencer.

Obtida a classificação da imagem, já estão satisfeitas todas as condições necessárias à correção da resposta do jogador. O processo de verificação da resposta é realizado através da execução do método **checkAnswer** (Figura 59) pertencente ao script *QuestionController.cs*, invocado imediatamente após a obtenção das probabilidades.

```

44 public void checkAnswer(KeyValuePair<string, float> result, Texture2D photo, float photoRatio, float photoOrientation, float photoScaleY)
45 {
46     bool isCorrect = false;
47     if (result.Key.Equals(card.cardQuestion.answerkeyword))
48     {
49         if (result.Value >= MINIMUM_CLASSIFICATION_VALUE){
50             isCorrect = true;
51         }
52     }
53     if (isCorrect)
54     {
55         this.classificationCanvas.SetActive(false);
56         Texture2D correctedPhoto = preparePhoto(photo, photoRatio, photoOrientation, photoScaleY);
57         this.cardTitleText.GetComponent<Text>().text = card.cardTitle;
58         this.resultPanel.SetActive(true);
59         this.gameController.GetComponent<GameController>().saveCard(this.card, correctedPhoto);
60         this.collectionController.Start();
61     }
62     else
63     {
64         Debug.Log("WRONG ANSWER!");
65         this.classificationCanvas.SetActive(false);
66         this.wrongPanel.SetActive(true);
67     }
68 }

```

Figura 59 - Método checkAnswer presente no script QuestionController.cs

Segundo o algoritmo contido em *checkAnswer*, para que a resposta do jogador seja considerada correta, é necessário obedecer a duas condições:

- A classe resultante da classificação da imagem ser igual ao termo definido como resposta correta à questão de quiz;
- A probabilidade associada à classe identificada na imagem ser igual ou maior que 70% (valor definido na variável **MINIMUM\_CLASSIFICATION\_VALUE**).

Caso as condições sejam confirmadas na sua totalidade, o método *checkAnswer* prepara e ativa a interface destinada a informar o jogador do sucesso da sua resposta (Figura 60), e procede à atualização dos dados do jogador (atualizando a sua coleção com a nova carta obtida).

Em caso negativo, é ativada a interface a informar da resposta incorreta (Figura 61).

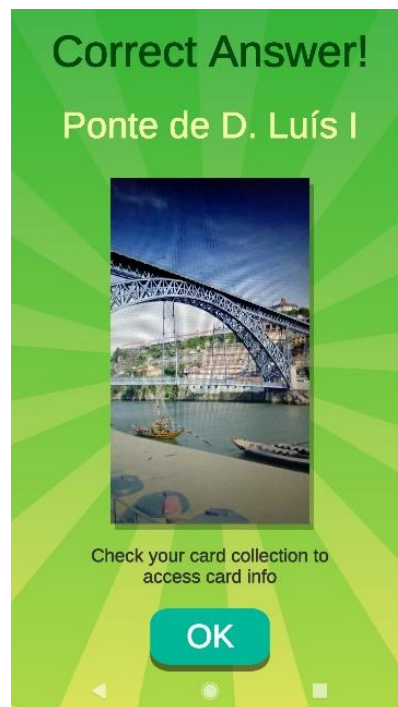


Figura 60 - Interface do jogo dedicada a informar o jogador da sua resposta de quiz visual correta

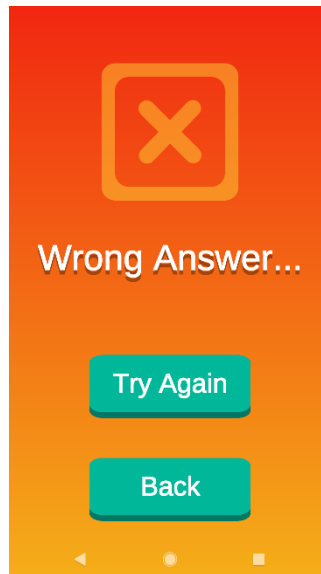


Figura 61 - Interface do jogo dedicada a informar o jogador da sua resposta de quiz visual incorreta

O processo de gravação e atualização dos dados do jogador ocorre no script *GameController.cs*, através do método **saveCard** (Figura 62). O método atualiza a coleção do jogador no objeto de dados *Player*, grava a imagem capturada (de forma a poder ser lida futuramente no processo de construção da interface de coleção de cartas) e através do método *managePlayerData* anteriormente mencionado, atualiza o ficheiro de persistência dos dados do jogador com os dados mais recentes.

```
160 public void saveCard(GameCardData card, Texture2D photo)
161 {
162     int routeID = this.gamedata.getRouteIDfromCard(card.cardID);
163     this.player.addCard(card, routeID);
164
165     saveCardPhoto(photo, card.cardID);
166
167     this.fileManager.savePlayerData(this.player);
168     managePlayerData();
169 }
```

Figura 62 - Método *saveCard* presente no script *GameController.cs*

### 3.3.5. Upgrade de Cartas – Networking

Conforme a definição e planeamento da mecânica de upgrade de cartas especificada na secção 3.1.3, é esperado um sistema que permita a comunicação entre dois

jogadores, de forma a averiguar se possuem a mesma carta nas suas respetivas coleções.

Dado que se pretende implementar uma funcionalidade que permita a comunicação entre dois dispositivos, surge a necessidade de desenvolver um sistema de jogo *multiplayer* em *Unity*.

De forma a colmatar o problema, procedeu-se à conceção de uma solução utilizando o **Photon Unity Network (PUN)** [50]. O PUN constitui um package disponível para *Unity* com o propósito de disponibilizar funcionalidades de *multiplayer* para soluções de jogos, de forma rápida e flexível. Toda a comunicação envolvida é assegurada através de servidores *Photon* dedicados, retirando a necessidade de hospedagem de comunicação das partes envolvidos no desenvolvimento de jogos.

A comunicação entre jogos através do PUN, é realizada com base num sistema de salas online: os vários jogadores têm a possibilidade de criar uma sala online ou de se juntarem a uma sala criada por outro jogador (a partir do ID da sala), sendo que dois jogadores presentes na mesma sala conseguem comunicar entre si.

No contexto do jogo associado ao projeto, foi projetado um sistema em que, após a seleção da carta que pretendem transformar, o dispositivo de cada jogador cria por definição a sua própria sala online (Figura 63), existindo a possibilidade de qualquer um dos dois se juntar à sala do outro jogador.

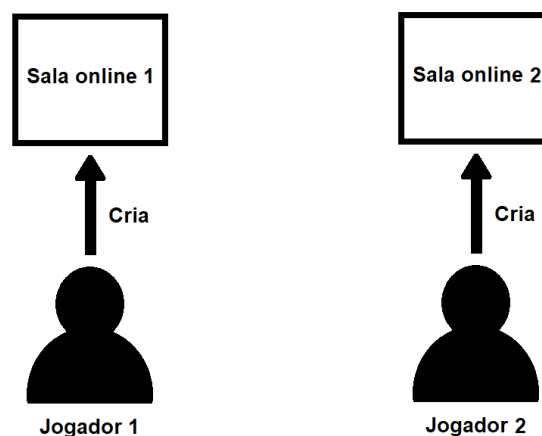


Figura 63 - Esquema representativo do primeiro passo do sistema de salas projetado

De seguida, o jogador que se pretenda juntar à sala do outro jogador, dado que precisa de conhecer o ID da sala a que se pretende juntar, procede à leitura de um código QR (cedido pelo outro dispositivo) que contém codificado o ID (Figura 64).

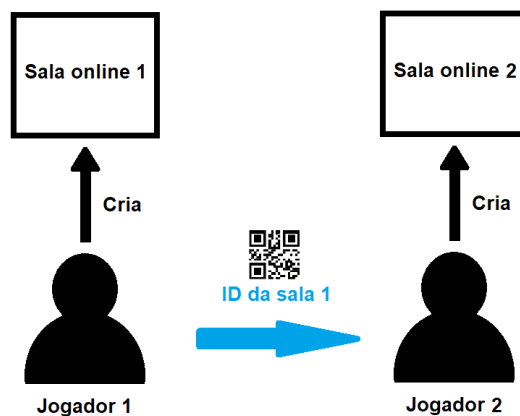


Figura 64 - Esquema representativo do segundo passo do sistema de salas projetado

Após obtido o ID da sala-destino, o jogador abandona a sala criada pelo seu próprio dispositivo, e procede a juntar-se à sala do outro jogador (Figura 65).

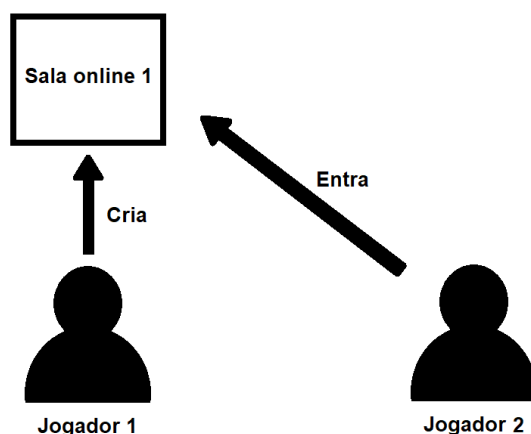


Figura 65 - Esquema representativo do terceiro passo do sistema de salas projetado

Estando os dispositivos de ambos os jogadores ligados à mesma sala, já é possível ocorrer o processo de comunicação e partilha de informação da carta de coleção escolhida.

Tendo em conta o sistema de gestão de salas *PUN* projetado, procedeu-se à sua aplicação no projeto de *Unity*.

Os objetos responsáveis pela gestão dos processos envolvidos na mecânica de *upgrade* de cartas são o **CardUpgradeCanvas** (Figura 28) e o respetivo script controlador **CardUpgradeController.cs** (Figura 29), apoiados pelo objeto **Photon** (Figura 28) (objeto criado com o propósito de isolar o controlo e funcionamento de todos os processos relativos ao PUN).



Figura 66 - Interface do jogo relativa ao Upgrade de Carta~

O processo é iniciado através da inicialização dos dados necessários à interface responsável (dados da carta a transformar - Figura 67) e futura comunicação entre dispositivos, acompanhado da inicialização e ativação do objeto *Photon* (Figura 68) e codificação em código QR do ID da sala criada pelo próprio dispositivo (Figura 66). Para ID da sala, foi definido que o sistema atribuiria automaticamente o valor do **deviceUniquelIdentifier**. Este valor, constituindo um identificador de dispositivo único atribuído pelo sistema do *Unity*, impede a existência de duas salas com o mesmo identificador.

A inicialização do objeto *Photon* através do script **PhotonController.cs** (linha 42 da Figura 68) desencadeia a criação automática da sala online no instante em que é assegurada uma ligação aos servidores PUN (realizada a partir do script **PhotonConnect.cs**)

```

97     public void setCardData(GameCardData cd)
98     {
99         this.cardData = cd;
100        this.card.GetComponent<CardPresentationController>().setCardData(cd,false);
101    }

```

Figura 67 - Método setCardData presente no script UpgradeCardController.cs

```

33     // Use this for initialization
34     public void init()
35     {
36         //Start();
37         createNetworkItem();
38         this.roomName = SystemInfo.deviceUniqueIdentifier; //Online room name is the device's unique identifier
39         fillQRCode();
40         //this.photonObject.GetComponent<PhotonController>().createRoom(this.roomName); //By default, a online room is created
41         this.photonObject.GetComponent<PhotonController>().setNetItem(this.netItem);
42         this.photonObject.GetComponent<PhotonController>().init();
43         this.photonObject.SetActive(true);
44     }
45 }

```

Figura 68 - Método init presente no script UpgradeCardController

Como se pode verificar, é também criado e inicializado o objeto **NetworkItem** (Figura 69), que tem a função de albergar a informação partilhada entre dispositivos, contendo o ID da carta a transformar.

```

1     using System;
2     using System.Collections;
3     using System.Collections.Generic;
4     using UnityEngine;
5
6     [Serializable]
7     public class NetworkItem : MonoBehaviour {
8
9         private int sentCardID;
10
11     public NetworkItem(int cardID)
12     {
13         this.sentCardID = cardID;
14     }
15
16     public int getSentCardID()
17     {
18         return this.sentCardID;
19     }
20
21 }

```

Figura 69 - Objeto NetworkItem.cs

O seguinte processo é dependente da ação que o jogador toma na interface de transformação da carta (Figura 66).

Caso o jogador opte pela leitura do código QR presente no dispositivo do outro jogador, através do evento associado ao botão correspondente, é ativada a interface de leitura através da câmara (Figura 70), estando-lhe associado o controlador **QRCameraController.cs**. Identicamente ao processo de captura de imagem na resposta do quiz visual, este controlador ativa a câmara traseira do dispositivo e realiza a sua pré-visualização na interface, sendo que neste caso, na atualização de cada *frame*, é executada automaticamente a procura de códigos QR na imagem, recorrendo à classe **Scanner.cs** [51]. Esta classe, criada por Kevin Destrem, permite a análise e procura automática de códigos QR utilizando o sistema de câmara nativo do *Unity*.

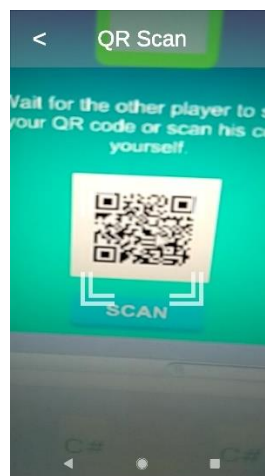


Figura 70 - Interface do jogo relativa à leitura do código QR

Na eventualidade de ser detetado um código QR, é invocado o método **QRCodeRead** (Figura 71) presente no script **QRCodeController.cs** a partir de **QRCameraController.cs**. O método, recebendo por parâmetro o código encontrado em formato textual, procede à invocação dos métodos de **PhotonController.cs** para que o dispositivo abandone a sala online criada pelo próprio e se junte à sala do outro jogador, guiado pelo ID desta.

```
115 public void QRCodeRead(string readQR)
116 {
117     this.loadingPanel.SetActive(true); //activates the screen with the loading animation
118     this.photonObject.GetComponent<PhotonController>().leaveRoom();
119     this.photonObject.GetComponent<PhotonController>().joinRoom(readQR);
120 }
```

Figura 71 - Método QRCodeRead presente no script QRCodeController.cs

Ao entrar na sala, através da *callback* interna do *PUN*, **OnJoinedRoom**, é executado o processo de envio do objeto **NetworkItem** previamente referido através do método **sendNetworkItem**, que através de um RPC (*Remote Procedure Call*[52]) configurado para ser executado em todos os restantes dispositivos da sala (neste caso específico, o outro jogador, dado que a sala se encontra configurada para suportar apenas dois jogadores), executa o método **SentItem** (Figura 72). Depois de verificar se o dispositivo da sala onde o método está a ser executado se trata do criador original da sala (de modo a assegurar que ambos os dispositivos envolvidos no processo enviam a informação da carta pelo menos uma vez), é invocado o método **compareNetworkItems** (Figura 73).

```
110 public void sendNetworkItem()
111 {
112     Debug.Log("SEND NETWORK ITEM");
113     this.photonView.RPC("SentItem", PhotonTargets.Others, this.netItem.getSentCardID());
114 }
115
116 [PunRPC]
117 public void SentItem(int sentCardID)
118 {
119     Debug.Log(SystemInfo.deviceUniqueIdentifier);
120     Debug.Log("SENTCARDID: "+sentCardID);
121     buildOtherPlayerNetItem(sentCardID);
122     if (this.createdRoom)
123     {
124         sendNetworkItem();
125     }
126     this.cardUpgradeCanvas.GetComponent<CardUpgradeController>().compareNetworkItems(this.otherPlayerNetworkItem);
127 }
```

Figura 72 - Métodos *sendNetworkItem* e *SentItem* presentes no script *PhotonController*

```
122 public void compareNetworkItems(NetworkItem otherPlayerNetItem)
123 {
124     if(netItem.getSentCardID() == otherPlayerNetItem.getSentCardID()) //Upgrade Successful
125     {
126         this.gameController.GetComponent<GameController>().upgradeCard(this.cardData.cardID); //saves the upgraded card data
127         this.collectionCanvas.GetComponent<CollectionController>().Start(); //updates the card collection screen
128         this.upgradedCard.GetComponent<CardPresentationController>().setCardData(this.cardData,false);
129         this.upgradedCard.GetComponent<CardPresentationController>().turnSpecialColor();
130         this.upgradeSuccessfulPanel.SetActive(true);
131     }
132     else //Upgrade Failed
133     {
134         this.resultFailedPanel.SetActive(true);
135     }
136     Debug.Log("My card ID: "+netItem.getSentCardID());
137     Debug.Log("Ohter player card ID: " + otherPlayerNetItem.getSentCardID());
138
139     deactivateLoadingScreen();
140 }
```

Figura 73 - Método *compareNetworkItems* presente no script *CardUpgradeController.cs*

Através do ID da carta contido no objeto recebido por parâmetro, o método **compareNetworkItems** compara se o ID da carta selecionada pelo próprio jogador corresponde ao ID da carta escolhido pelo outro jogador. Em caso positivo, são executados os métodos do *GameController* responsáveis pela atualização e persistência dos dados relativos à coleção do jogador, é atualizada a cor da carta

(sendo-lhe a atribuída a cor dourada) nas várias interfaces, e é ativada a interface que informa o jogador do sucesso da sua operação (Figura 74). Caso contrário, é ativada a respetiva interface informativa do insucesso (Figura 75).

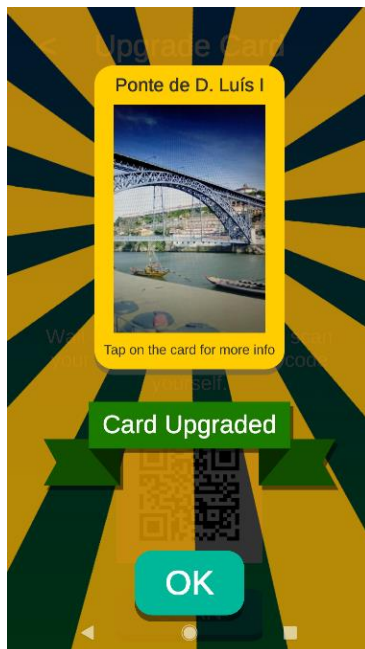


Figura 74 - Interface do jogo responsável por informar o jogador do sucesso da operação de transformação da carta

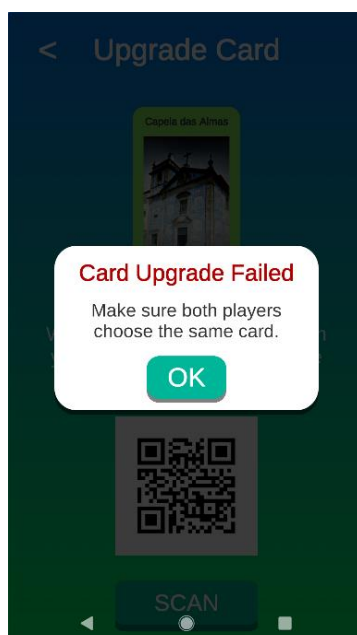


Figura 75 - Interface do jogo responsável por informar o jogador de uma operação de transformação de carta falhada

## 4. Avaliação

Dada a extensão e complexidade inerente ao processo de desenvolvimento de soluções de tecnologias de informação, existe a necessidade de assegurar um controlo rigoroso de qualidade e avaliação das funcionalidades, permitindo assim verificar o grau de cumprimento dos objetivos aos quais as soluções se propõem.

Sendo que existe a necessidade de esquematizar um plano sólido de avaliação e controlo de qualidade, tanto no cumprimento de objetivos, como nas características da própria solução, decidiu-se recorrer a um sistema de ciclos de desenvolvimento na fase de produção, acompanhado de um controlo apoiado pelo modelo de *Quantitative Evaluation Framework* (QEF).

### 4.1. Ciclos de Desenvolvimento na Fase de Produção

O sistema de ciclos de desenvolvimento na fase de produção visa organizar de forma temporal os diversos ciclos de atividades de produção, estabelecendo uma ordem sequencial com potencial concorrência. Este sistema é composto por três ciclos:

- **Ciclo Alfa** – Nesta primeira fase ocorrem as primeiras instâncias de desenvolvimento e implementação da solução pretendida. É durante a fase do ciclo alfa, que ocorrem os **Testes Alfa**. Estes testes aferem a qualidade do primeiro protótipo da solução e são realizados antes do produto ser disponibilizado para qualquer utilizador final, sendo que normalmente os seus participantes são elementos da própria equipa de desenvolvimento.
- **Ciclo Beta** – Durante a fase do ciclo da Beta continua o processo de desenvolvimento, com base nos resultados obtidos na fase anterior. Neste ciclo ocorrem os **Testes Beta**, que ao contrário dos Testes Alfa, participam na experimentação da solução indivíduos externos à equipa de desenvolvimento, maioritariamente do segmento de público-alvo. Os Testes Beta são compostos por uma sessão de testes diretos com a solução desenvolvida e pelo preenchimento de um questionário por parte dos seus participantes, a fim de se obter um feedback direto que permita quantificar e avaliar o cumprimento dos vários requisitos definidos.
- **Ciclo Gama** – Realizados os testes Beta, e com base no relatório das sessões de teste, ocorre a fase do Ciclo Gama, onde são realizados os aprimoramentos finais e se finaliza globalmente o processo de desenvolvimento.

#### 4.2. Quantitative Evaluation Framework (QEF)

O método *Quantitative Evaluation Framework* (QEF), tem como objetivo servir como ferramenta de apoio na avaliação quantitativa da qualidade do software de caráter educativo[53], permitindo obter uma análise objetiva sobre as várias características e cumprimento dos vários requisitos funcionais e não-funcionais em qualquer fase do desenvolvimento da solução pretendida.

A avaliação quantitativa rege-se segundo uma divisão por dimensões, em que cada dimensão se divide por vários fatores, sendo um fator constituído por um conjunto de requisitos.

Segundo o modelo QEF, a importância de cada fator ( $D_i$ ) para cada dimensão é calculada segundo a seguinte fórmula onde  $n$  representa o nº de fatores existentes na dimensão:

$$D_i = \sum_n (p_n \times factor_n) \quad \sum_n (p_n) = 1 \quad p_n \in [0,1]$$

Sendo que cada fator é avaliado através da fórmula ( $m$  representa o número de critérios definidos relativos ao fator em questão,  $pr_m$  representa o peso predefinido de  $m$  e  $pc_m$  equivale à percentagem de cumprimento de  $m$ ):

$$F_n = \frac{1}{\sum_m pr_m} \times \sum_m (pr_m \times pc_m) \quad p_n \in [0,1]$$

Pressupondo que o desvio global da solução face ao ideal ( $D$ ) é dado pela fórmula:

$$D = \sqrt{\sum_j \left(1 - \frac{Dim_j}{100}\right)^2}$$

a **Qualidade global da solução (Q)** traduz a diferença percentual entre o estado atual da solução desenvolvida e a solução ideal, sendo traduzida pela fórmula:

$$Q = 1 - \frac{D}{\sqrt{n}} \times 100 \quad Q \in [0,100]$$

Para que os vários cálculos sejam possíveis, é necessário atribuir a cada requisito o respetivo peso dentro do fator onde se inserem. Esta grandeza varia e pode assumir o valor no intervalo de valores {2,4,6,8,10}, sendo que quanto maior o valor, maior a importância do requisito junto do fator correspondente.

No caso específico da solução pretendida, os vários fatores e respetivos requisitos estipulados do QEF adaptado ao projeto encontram-se divididos segundo três dimensões: **Funcionalidade, Usabilidade e Adaptabilidade**:

- **Funcionalidade**
  - **Perfil**
    - FP1 – Criar conta de utilizador (Peso: 10) (Fase Alfa)
    - FP2 – Iniciar sessão com conta de utilizador (Peso: 10) (Fase Alfa)
    - FP3 - Persistência dos dados do utilizador (Peso: 10) (Fase Alfa)
  - **Coleção de Cartas**
    - FC1 – Consultar e navegar pela coleção virtual de cartas (Peso: 10) (Fase Alfa)
    - FC2 – Consultar informação relativa a uma carta (Peso: 10) (Fase Alfa)
    - FC3 – Transformar uma carta colecionada numa carta especial (Peso: 10) (Fase Alfa)
  - **Jogo**
    - FJ1 - Iniciar quiz relativo a um percurso (Peso: 10) (Fase Alfa)
    - FJ2 - Responder à questão apresentada (Peso: 10) (Fase Alfa)
    - FJ3 - Correção da resposta à questão apresentada (Peso: 10) (Fase Beta)
    - FJ4 - Adquirir a carta relativa à questão (Peso: 10) (Fase Alfa)
- **Usabilidade**
  - **Navegação de menus**
    - UN1 - Os menus encontram-se bem estruturados e permitem os utilizadores aceder ao seu conteúdo de uma forma intuitiva (Peso: 8) (Fase Beta)
    - UN2 - Não são apresentados quaisquer erros durante a execução (Peso: 8) (Fase Beta)
    - UN3 - O sistema de menus e as respetivas opções colocam o utilizador com uma sensação de controlo (Peso: 8) (Fase Beta)
    - UN4 - As ações importantes do utilizador apresentam feedback adequado (Peso: 8) (Fase Beta)
    - UN5 - Não é apresentada nenhuma informação de carácter ofensivo ao utilizador (Peso: 8) (Fase Beta)
  - **Coleção**
    - UC1 - O sistema de coleção apresenta uma interface apelativa para o jogador (Peso: 8) (Fase Beta)
    - UC2 - As várias cartas da coleção encontram-se organizadas de uma forma intuitiva e de rápido acesso (Peso: 8) (Fase Beta)
    - UC3 - A informação das cartas é apresentada de forma clara e sem erros (Peso: 8) (Fase Beta)

- **Jogo**
  - UJ1 - Todas as ações de caráter importante que ocorram durante a sessão de jogo apresentam feedback adequado (Peso: 8) (Fase Beta)
  - UJ2 - O processo de análise da foto capturada pelo jogador e respetiva correção da resposta é realizado de forma rápida (Peso: 8) (Fase Beta)
- **Suporte**
  - US1 - O jogo é de fácil instalação para o público-alvo (Peso: 8) (Fase Beta)
- **Adaptabilidade**
  - **Perfil do Jogador**
    - AP1 - Não são armazenados dados sensíveis relativos ao utilizador (Peso: 10) (Fase Alfa)
  - **Conteúdo sério**
    - AC1 - O jogo estimula os jogadores a visitar marcos e monumentos de importância cultural (Peso: 10) (Fase Beta)
    - AC2 - O jogo permite a aprendizagem de informação cultural relevante (Peso: 10) (Fase Beta)
    - AC3 - O jogo estimula a cooperação entre jogadores (Peso: 8) (Fase Beta)
  - **Versatilidade**
    - AV1 - As interfaces dos menus adaptam-se à resolução do ecrã dos dispositivos (Peso: 8) (Fase Beta)
    - AV2 - O jogo consegue manter um bom desempenho quando executado em diferentes sistemas (Peso: 10) (Fase Beta)

Visto que o cálculo da qualidade da solução se encontra matematicamente relacionado com os valores do grau de cumprimento de cada requisito individual, é necessária a existência de um sistema métrico que permite atribuir os vários valores percentuais de classificação a cada requisito durante os vários ciclos de desenvolvimento.

Relativamente à avaliação da solução associada ao projeto em questão, e visto que será implementado um sistema de ciclos de desenvolvimento na fase de produção (Secção 4.1), **será nos Testes Alfa e Beta que se realizará a atribuição da classificação de cumprimento de cada requisito**. Uma vez que nos Testes Alfa só participam elementos pertencentes à equipa de desenvolvimento, entende-se que será no final do Ciclo Alfa que se iniciará o processo de classificação dos requisitos relativos maioritariamente aos fatores da dimensão “Funcionalidade”. No entanto, relativamente aos requisitos pertencentes aos fatores de “Usabilidade” e “Adaptabilidade”, uma vez que estão relacionados com a forma como os elementos pertencentes ao grupo do público-alvo reagem e interagem com a solução pretendida,

entende-se que as classificações relativas e estas dimensões só podem ser atribuídas através de métricas que dependam do relatório das sessões de Testes Beta, com base no relatório de observação das próprias sessões, assim como no estudo de respostas a questões dos questionários de carácter de avaliação quantitativa por parte dos indivíduos participantes nos testes.

Tendo por base a natureza dos vários requisitos, procedeu-se à definição das métricas de avaliação relativas aos requisitos das várias dimensões e fatores (Anexo F – Métricas de avaliação QEF).

#### 4.2.1. Fase Alfa

Conforme mencionado anteriormente, a primeira fase onde ocorrem processos de verificação do grau de cumprimento de requisitos relativos à solução projetada constitui a Fase Alfa.

Dado que os testes alfa correspondem, na sua maioria, à verificação de funcionalidades objetivas do jogo, procedeu-se à listagem dos requisitos relativos, acompanhados das respetivas percentagens de cumprimento da solução (Tabela 7) na janela temporal em questão (atribuídos segundo as métricas presentes no Anexo F – Métricas de avaliação QEF).

*Tabela 7 - Lista de percentagens de cumprimento da solução face aos requisitos relativos à Fase Alfa*

| <b>Requisito</b> | <b>Cumprimento (%)</b> |
|------------------|------------------------|
| FP1              | 100                    |
| FP2              | 100                    |
| FP3              | 100                    |
| FC1              | 100                    |
| FC2              | 100                    |
| FC3              | 100                    |
| FJ1              | 100                    |

|     |     |
|-----|-----|
| FJ2 | 100 |
| FJ4 | 100 |
| AP1 | 100 |

#### 4.2.2. Fase Beta

Após finalizados e aprimorados todos os aspetos relativos às funcionalidade e interfaces de utilização, ocorre a Fase Beta.

De acordo com o que foi referido anteriormente, esta fase caracteriza-se pela realização dos testes beta, através do registo atento da experimentação da solução diretamente com um grupo de indivíduos pertencentes ao público-alvo.

De forma a averiguar todos os requisitos do QEF relativos a aspetos a serem avaliados na fase alfa, foi decidido que o principal critério de avaliação seriam **respostas a questionários de qualidade por parte dos indivíduos presentes nas sessões de teste beta**. Para tal, foi desenvolvido um questionário (Anexo G – Questionário de Qualidade) que através de questões de resposta binária e quantitativa, permitisse uma avaliação prática e objetiva dos requisitos diretamente relacionados com aspetos da utilização do público-alvo.

Estando preparado o jogo para testagem e desenvolvido o questionário de controlo de qualidade, procedeu-se à condução das sessões de testes beta.

Cada sessão de teste junto do candidato inquirido, foi composta pela testagem do jogo seguindo um guião de teste e o preenchimento do questionário de qualidade. O guião de teste é composto pelos seguintes passos:

1. Instalação do jogo;
2. Criação de conta de jogador;
3. Exploração livre de menus;
4. Aceder à interface de informação geral do jogo;
5. Aceder à coleção virtual de cartas;
6. Consultar informação sobre um percurso;
7. Iniciar processo de Quiz Visual;
8. Adquirir uma carta através de uma resposta correta;
9. Consultar informação relativa à carta ganha;
10. Repetir passos 7,8 e 9 um número de vezes à sua escolha;
11. Iniciar processo de transformação de carta em carta especial;
12. Leitura de código QR de outro dispositivo;
13. Transformar com sucesso uma carta na versão especial;

#### 14. Consultar informação de uma carta especial.

Terminada a fase beta, procedeu-se à organização e análise dos dados obtidos através das respostas aos inquéritos de qualidade.

Na totalidade, foram obtidas **18 respostas**.

Dado que o principal propósito da fase beta e respetiva análise dos dados resultantes é averiguar a percentagem de cumprimento do jogo de cada requisito relativo a aspetos diretamente ligados ao jogador, foi analisada individualmente a totalidade de respostas de cada pergunta, acompanhado do respetivo cálculo de cumprimento dos requisitos ao qual a questão se refere:

#### How easy was to install the game? (Rate in a scale of 1-5)

18 responses

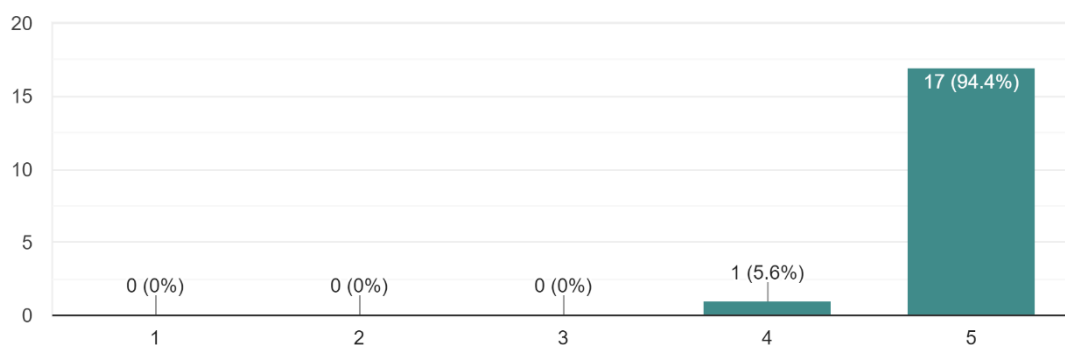


Figura 76 - Gráfico de barras relativo às respostas da questão "How easy was to install the game?"

Relativamente à questão "How easy was to install the game?" representada no gráfico de barras da Figura 76, pode-se verificar que 17 dos inquiridos atribuíram a classificação máxima de 5 pontos, acompanhado por um inquirido a classificar o nível de dificuldade de instalação com 4 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **US1**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

### Did you find the game's interface intuitive? (Rate in a scale of 1-5)

18 responses

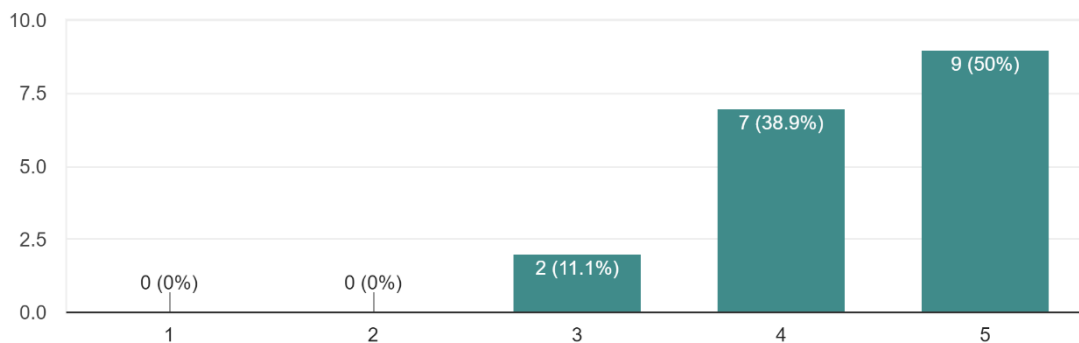


Figura 77 - Gráfico de barras relativo às respostas da questão "Did you find the game's interface intuitive?"

Relativamente à questão "Did you find the game's interface intuitive?" representada no gráfico de barras da Figura 77, pode-se verificar que 9 dos inquiridos atribuíram a classificação máxima de 5 pontos, 7 inquiridos atribuíram 4 pontos e 2 inquiridos atribuíram 3 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **UN1**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

### Did you find any technical errors during your gameplay session?

18 responses

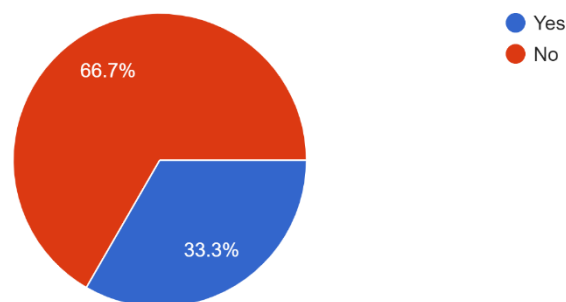


Figura 78 - Gráfico circular relativo às respostas da questão "Did you find any technical errors during your gameplay session?"

Relativamente à questão *"Did you find any technical errors during your gameplay session?"* representada no gráfico circular da Figura 78, pode-se verificar que 66.7% dos inquiridos respondeu "No", sendo que 33.3% respondeu "Yes".

Dado que nesta questão específica, é considerado como positiva a resposta "No", com base nas métricas de avaliação definidas para os requisitos **UN2**, **AV1** e **AV2**, é-lhes atribuído o valor de 75% de cumprimento (considerando que o valor se encontra no intervalo 60%-80%).

At any moment of your gameplay session, did you feel you lost control over the game's menu actions?

18 responses

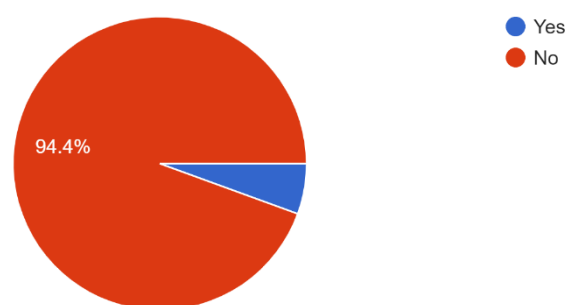


Figura 79 - Gráfico circular relativo às respostas da questão "At any moment of your gameplay session, did you feel you lost control over the game's menu actions?"

Relativamente à questão *"At any moment of your gameplay session, did you feel you lost control over the game's menu actions?"* representada no gráfico circular da Figura 79, pode-se verificar que 99.4% dos inquiridos respondeu "No", sendo que 5.6% respondeu "Yes".

Dado que nesta questão específica, é considerado como positiva a resposta "No", com base nas métricas de avaliação definidas para o requisito **UN3**, é-lhe atribuído o valor de 100% de cumprimento (considerando que o valor se encontra no intervalo 80%-100%).

How would you rate the game's feedback regarding your actions? (Rate in a scale of 1-5)

18 responses

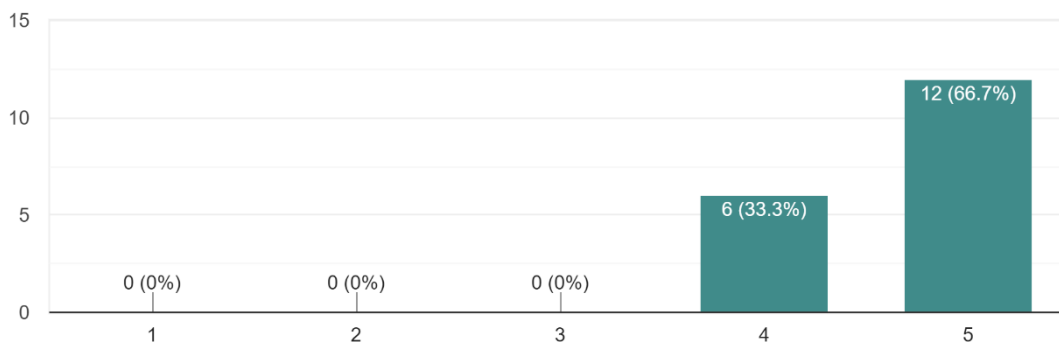


Figura 80 - Gráfico de barras relativo às respostas da questão "How would you rate the game's feedback regarding your actions?"

Relativamente à questão "How would you rate the game's feedback regarding your actions?" representada no gráfico de barras da Figura 80, pode-se verificar que 12 dos inquiridos atribuíram a classificação máxima de 5 pontos e 6 inquiridos atribuíram 4 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para os requisitos **UN4** e **UJ1**, é-lhes atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

## How would you rate the Card Collection interface? (Rate in a scale of 1-5)

18 responses

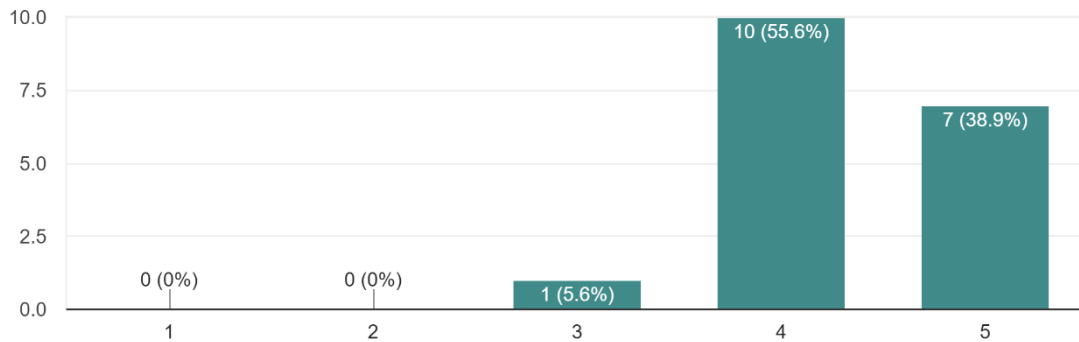


Figura 81 - Gráfico de barras relativo às respostas da questão "How would you rate the Card Collection interface?"

Relativamente à questão "How would you rate the Card Collection interface?" representada no gráfico de barras da Figura 81, pode-se verificar que 7 dos inquiridos atribuíram a classificação máxima de 5 pontos, 10 inquiridos atribuíram 4 pontos e 1 inquirido atribuiu 3 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **UC1**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

### Did you find the Card Collection system intuitive? (Rate in a scale of 1-5)

18 responses

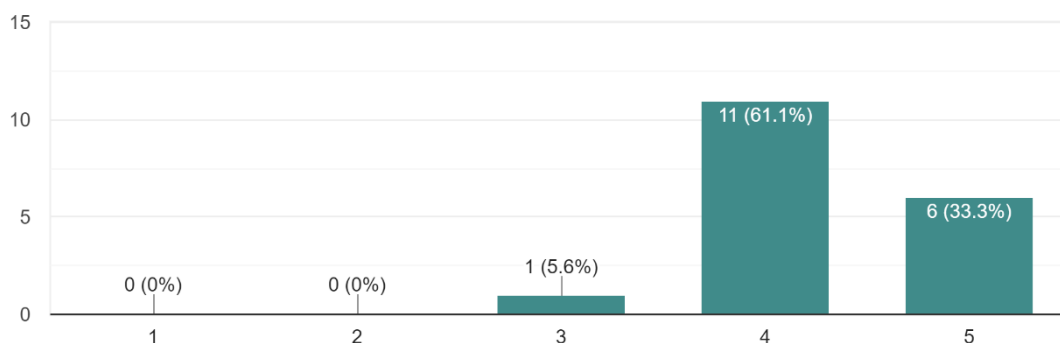


Figura 82 - Gráfico de barras relativo às respostas da questão "Did you find the card Collection system intuitive?"

Relativamente à questão "Did you find the card Collection system intuitive?" representada no gráfico de barras da Figura 82, pode-se verificar que 6 dos inquiridos atribuíram a classificação máxima de 5 pontos, 11 inquiridos atribuíram 4 pontos e 1 inquirido atribuiu 3 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **UC2**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

Would you classify the card's information as clear and absent of any errors?  
(Rate in a scale of 1-5)

18 responses

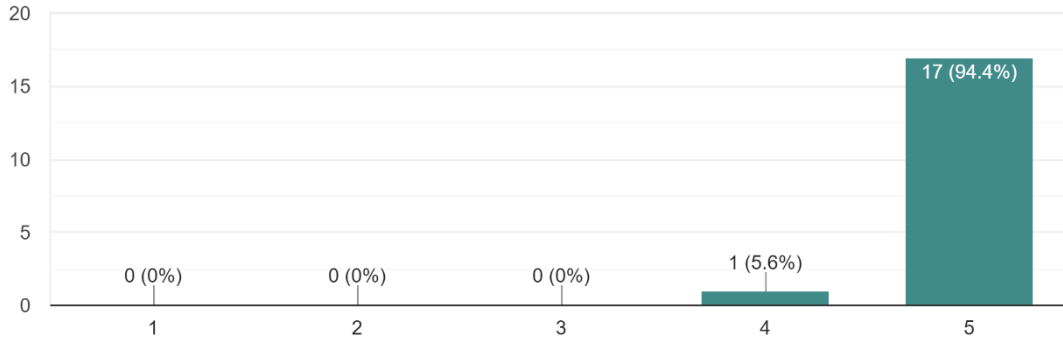


Figura 83 - Gráfico de barras relativo às respostas da questão "Would you classify the card's information as clear and absent of errors?"

Relativamente à questão "Would you classify the card's information as clear and absent of errors?" representada no gráfico de barras da Figura 83, pode-se verificar que 17 dos inquiridos atribuíram a classificação máxima de 5 pontos e 1 inquirido atribuiu 4 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **UC3**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

When answering a game question, after taking a photo, how fast was the result given? (Rate in a scale of 1-5)

18 responses

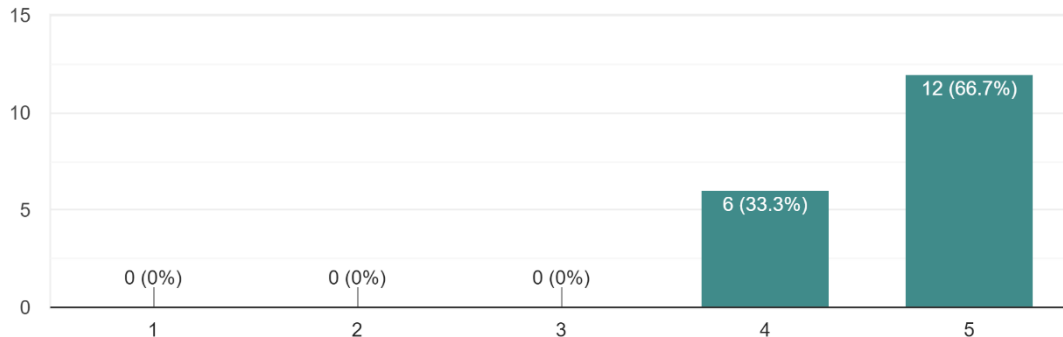


Figura 84 - Gráfico de barras relativo às respostas da questão "When answering a game question, after taking a photo, how fast was the result given?"

Relativamente à questão "When answering a game question, after taking a photo, how fast was the result given?" representada no gráfico de barras da Figura 84, pode-se verificar que 12 dos inquiridos atribuíram a classificação máxima de 5 pontos e 6 inquiridos atribuíram 4 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **UJ2**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

After answering a game question with the phone camera, how accurate was the result? (Rate in a scale of 1-5)

18 responses

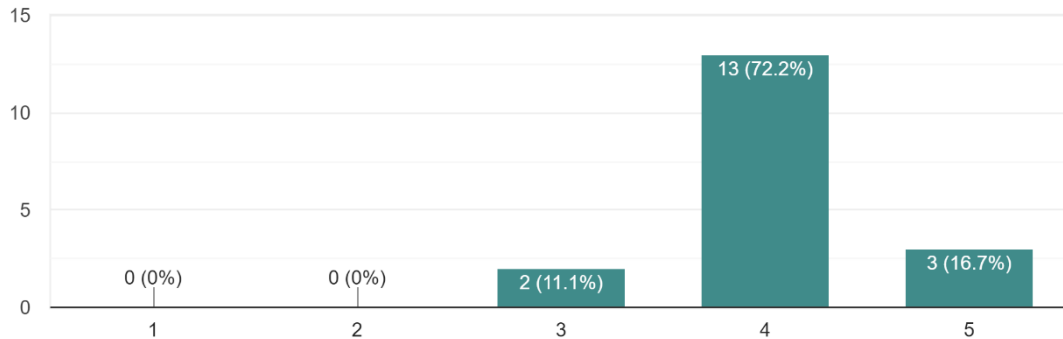


Figura 85 - Gráfico de barras relativo às respostas da questão "After answering a game question with the phone camera, how accurate was the result?"

Relativamente à questão "After answering a game question with the phone camera, how accurate was the result?" representada no gráfico de barras da Figura 85, pode-se verificar que 3 dos inquiridos atribuíram a classificação máxima de 5 pontos, 13 inquiridos atribuíram 4 pontos e 2 inquiridos atribuíram 3 pontos.

Dado que se considera uma classificação como positiva quando maior que 2 pontos, com base nas métricas de avaliação definidas para o requisito **FJ3**, é-lhe atribuído o valor de 100% de cumprimento (sendo que todas as classificações dadas como resposta são positivas).

Did you find any offensive messages within the game? (racist, xenophobic, etc.)

18 responses

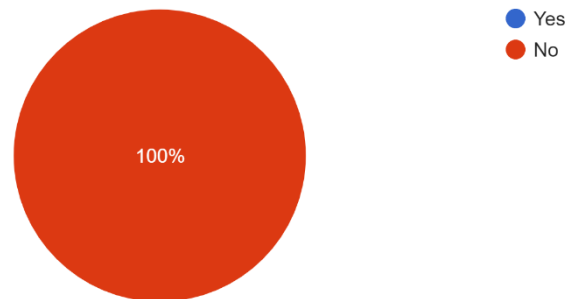


Figura 86 - Gráfico circular relativo às respostas da questão "Did you find any offensive messages within the game?"

Relativamente à questão "Did you find any offensive messages within the game?" representada no gráfico circular da Figura 86, pode-se verificar que 100% dos inquiridos respondeu "No".

Dado que nesta questão específica, é considerado como positiva a resposta "No", com base nas métricas de avaliação definidas para o requisito **UN5**, é-lhe atribuído o valor de 100% de cumprimento (considerando que o valor se encontra no intervalo 80%-100%).

Do you think this experience could lead people to visit monuments and cultural landmarks in a tourist context?

18 responses

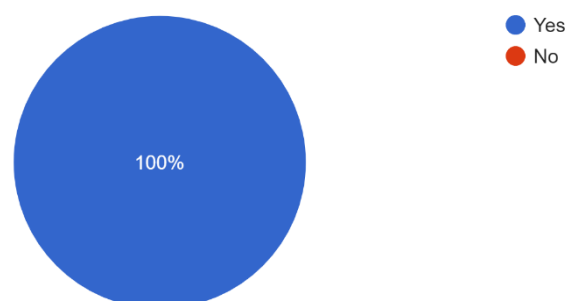


Figura 87 - Gráfico circular relativo às respostas da questão "Do you think this experience could lead people to visit monuments and cultural landmarks in a tourist context?"

Relativamente à questão *"Do you think this experience could lead people to visit monuments and cultural landmarks in a tourist context?"* representada no gráfico circular da Figura 87, pode-se verificar que 100% dos inquiridos respondeu "Yes".

Dado que nesta questão específica, é considerado como positiva a resposta "Yes", com base nas métricas de avaliação definidas para o requisito **AC1**, é-lhe atribuído o valor de 100% de cumprimento (considerando que o valor se encontra no intervalo 80%-100%).

Would you rather play the game alone or with other people?

18 responses

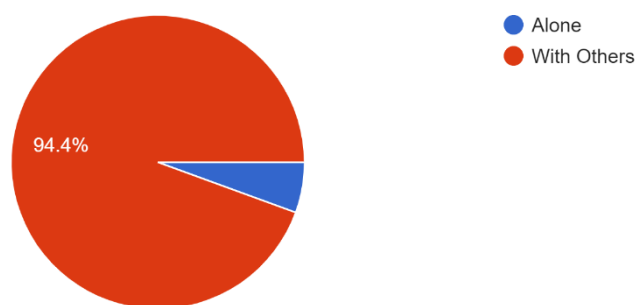


Figura 88 - Gráfico circular relativo às respostas da questão *"Would you rather play the game alone or with other people?"*

Relativamente à questão *"Would you rather play the game alone or with other people?"* representada no gráfico circular da Figura 88, pode-se verificar que 99.4% dos inquiridos respondeu "With Others", sendo que 5.6% respondeu "Alone".

Dado que nesta questão específica, é considerado como positiva a resposta "With Others", com base nas métricas de avaliação definidas para o requisito **AC3**, é-lhe atribuído o valor de 100% de cumprimento (considerando que o valor se encontra no intervalo 80%-100%).

### Do you believe the game helped or could help you learn more about monuments/landmarks or other cultural information?

18 responses

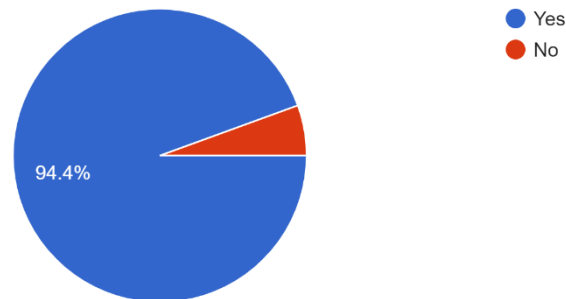


Figura 89 - Gráfico circular relativo às respostas da questão "Do you believe the game helped or could help you learn more about monuments/landmarks or other cultural information?"

Relativamente à questão "Do you believe the game helped or could help you learn more about monuments/landmarks or other cultural information?" representada no gráfico circular da Figura 89, pode-se verificar que 99.4% dos inquiridos respondeu "Yes", sendo que 5.6% respondeu "No".

Dado que nesta questão específica, é considerado como positiva a resposta "Yes", com base nas métricas de avaliação definidas para o requisito **AC2**, é-lhe atribuído o valor de 100% de cumprimento (considerando que o valor se encontra no intervalo 80%-100%).

### Which aspects do you want to be improved on future releases (if any)?

3 responses

- I would suggest adding GPS integration, so that tourists can use it to get directions or hints of where they may find the answer to the question.
- Melhorar reconhecimento da foto
- Content was too Hard

Figura 90 - Respostas dos inquiridos à questão "Which aspects do you want to be improved on future releases (if any)?"

Como se pode verificar nas respostas representada na Figura 90, apenas 3 inquiridos procederam à sugestão de melhorias e críticas construtivas à solução desenvolvida, confirmando em parte os resultados das restantes questões: a necessidade de melhorias ao nível da interface e melhorias na eficácia do classificador de imagens. No entanto, relativamente a este último aspeto, reconhece-se que a forma como foi testado o quiz visual pode ter influenciado a perceção da sua eficácia por parte dos inquiridos, dado que as fotografias das sessões de testes, para efeitos práticos, foram capturadas através de ecrãs de computador e telemóvel com os vários monumentos e marcos representados, em vez de serem obtidas diretamente num local físico onde se encontram os objetos.

Como se verificou na análise das várias respostas ao questionário, foi possível o cálculo dos restantes requisitos do QEF relativos à fase beta (Tabela 8).

*Tabela 8 - Percentagens de cumprimento de requisitos relativos à Fase Beta*

| <b>Requisito</b> | <b>Cumprimento (%)</b> |
|------------------|------------------------|
| FJ3              | 100                    |
| AC1              | 100                    |
| AC2              | 100                    |
| AC3              | 100                    |
| AV1              | 75                     |
| AV2              | 75                     |
| UN1              | 100                    |
| UN2              | 75                     |
| UN3              | 100                    |
| UN4              | 100                    |
| UN5              | 100                    |
| UC1              | 100                    |
| UC2              | 100                    |

|     |     |
|-----|-----|
| UC3 | 100 |
| UJ1 | 100 |
| UJ2 | 100 |
| US1 | 100 |

Estando calculada a totalidade das percentagens de cumprimento dos requisitos do QEF, através dos métodos referidos na secção 4.2, já é possível o cálculo da Qualidade Global da Solução (Q). **O valor Q calculado do projeto é 97%.**

Sendo este um valor bastante próximo de 100%, considera-se que a solução desenvolvida apresenta elevada qualidade face ao cumprimento quantitativo dos requisitos definidos.

Relativamente ao ciclo Gama, no contexto específico do projeto, dado que os resultados relativos aos vários testes Beta se revelaram bastante positivos e extremamente próximos da solução idealizada para o projeto, não ocorreram aprimoramentos relevantes tanto ao nível das interfaces como das funcionalidade da solução, considerando-se assim terminado o projeto e suprimindo-se a fase gama.

## 5. Conclusões

No presente capítulo serão expostas as várias conclusões associadas ao projeto, assim como uma reflexão sobre as várias limitações e possível trabalho futuro.

Tem ocorrido um constante crescimento no volume da participação em atividades de turismo cultural a um nível internacional. Tradicionalmente, tais atividades não apresentam uma estrutura sólida, pelo que um dos maiores problemas dos seus participantes, face ao desejo de vivenciar a cultura que visitam, é a falta de um percurso eficaz cuidadosamente construído para maximizar as suas experiências culturais.

De forma a resolver o problema em questão, reconhece-se que uma solução de um jogo sério dedicado a auxiliar atividades turísticas culturais constituiria uma melhoria face às atuais soluções disponíveis no mercado com o mesmo propósito, calculando-se que a utilização de Jogos Sérios para tal fim representaria a resposta ideal ao problema em questão, sendo que conseguiria proporcionar uma ligação orgânica entre os processos didáticos e lúdicos envolvidos no turismo cultural.

O tema do projeto constitui o desenvolvimento de um jogo sério que, recorrendo à mecânica de *Card Collecting* e uso de sistemas de classificação automática de imagem, consiga, a partir da definição de percursos compostos por diversos marcos e monumentos culturais, auxiliar e promover atividades turísticas assim como a sua visita direta.

Para tal foi realizado um estudo ponderado de várias soluções de jogos que possuam mecânicas de interação com potencial a ajudar na resolução do problema, quando aplicadas ao jogo sério projetado.

### 5.1. Objetivos Realizados

Relativamente aos vários objetivos associados ao projeto, considera-se que globalmente, foram atingidos na sua plenitude.

Foi realizado um estudo da mecânica de *Card Collecting*, ao nível dos vários processos que desencadeia no jogador e sistemas de jogo, assim como do potencial da sua utilização em contextos sérios e didáticos, mais concretamente na área do Turismo.

Foram estudadas e exploradas várias mecânicas de interatividade com o meio físico, com o objetivo de atribuir uma dimensão palpável e real a jogos virtuais, classicamente separado deste meio de comunicação.

Sendo este o objetivo primordial, foi desenvolvido um jogo sério, que utilizando a mecânicas de *Card Collecting* e de interação, tem como propósito o auxílio e promoção da cultura, através de atividades de turismo cultural, apresentando um elevado nível de Qualidade global calculado (97%) com base nos requisitos de projeto definidos.

## 5.2. Limitações e Trabalho Futuro

No que se refere à solução do jogo sério desenvolvido, conforme se pode verificar nas várias sessões de testes beta, apesar de globalmente constituir uma solução com qualidade e potencial nos seus contexto sérios, reconhece-se que possui espaço para futuras melhorias e aprimoramentos tanto ao nível da interface de jogo como do sistema de classificação automática de imagens. Considera-se que este sistema poderia ser mais sólido e mais eficaz através da introdução de maiores e melhores *datasets* de imagens (proprietários e capturados pela própria equipa de desenvolvimento) e através do aumento da sua qualidade, quantidade e diversidade (tanto ao nível dos ângulos de visualização do objeto representado nas imagens, como do dispositivo de captura).

Como se verificou nas várias sugestões dadas na fase beta por parte dos inquiridos, reconhece-se que a introdução de novas mecânicas interativas, tais como por exemplo, um sistema de localização GPS, seriam favoráveis no alcance dos objetivos sérios do jogo.

## 6. Bibliografia

- [1] M. Griffiths, «The educational benefits of videogames», vol. 20, p. 5, 2002.
- [2] S. Gans e MD, «Positive Reinforcement Can Help Favorable Behaviors», *Verywell Mind*. [Em linha]. Disponível em: <https://www.verywellmind.com/what-is-positive-reinforcement-2795412>. [Acedido: 09-Fev-2019].
- [3] TEDx Talks, *Our Brains are Wired to Collect Things | Daniel Krawczyk | TEDxSMU*. .
- [4] World Tourism Organization (UNWTO), Ed., *UNWTO Tourism Highlights: 2018 Edition*. World Tourism Organization (UNWTO), 2018.
- [5] «International tourist arrivals worldwide by region 2017 | Statista», *Statista*. [Em linha]. Disponível em: <https://www.statista.com/statistics/186743/international-tourist-arrivals-worldwide-by-region-since-2005/>. [Acedido: 07-Fev-2019].
- [6] World Tourism Organization (UNWTO), *Tourism and Culture Synergies*. World Tourism Organization (UNWTO), 2018.
- [7] M. Prensky, *Digital Game-Based Learning*, vol. 1. New York, 2001.
- [8] «Nexto Guide - Apps on Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=io.nexto&hl=en\\_US](https://play.google.com/store/apps/details?id=io.nexto&hl=en_US). [Acedido: 24-Fev-2019].
- [9] «Parks | Fish and Wildlife | Mobile Apps | GeoChallenge | Pocket Ranger». [Em linha]. Disponível em: <http://www.pocketranger.com/?fbclid=IwAR3SCy3Zrw3qddiLRKLztEh2CnIQ4zGib09PXsAUrTnFBGAvBHO-C6YqAsk>. [Acedido: 24-Fev-2019].
- [10] «Pokémon GO – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=pt_PT). [Acedido: 24-Fev-2019].
- [11] «Geocaching® – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.groundspeak.geocaching.intro&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.groundspeak.geocaching.intro&hl=pt_PT). [Acedido: 24-Fev-2019].
- [12] «HQ - Trivia & Words», *App Store*. [Em linha]. Disponível em: <https://itunes.apple.com/us/app/hq-trivia-words/id1232278996?mt=8>. [Acedido: 24-Fev-2019].
- [13] «Jeopardy! World Tour – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.sonypicturestelevision.jeopardy2&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.sonypicturestelevision.jeopardy2&hl=pt_PT). [Acedido: 24-Fev-2019].
- [14] «Trivia Crack - Apps on Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.etermax.preguntados.lite&hl=en\\_US](https://play.google.com/store/apps/details?id=com.etermax.preguntados.lite&hl=en_US). [Acedido: 24-Fev-2019].
- [15] «QuizUp – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.quizup.core&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.quizup.core&hl=pt_PT). [Acedido: 24-Fev-2019].
- [16] «Google Lens – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.google.ar.lens&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.google.ar.lens&hl=pt_PT). [Acedido: 24-Fev-2019].
- [17] «PlantNet Identificação Planta – Aplicações no Google Play». [Em linha]. Disponível em:

- [https://play.google.com/store/apps/details?id=org.plantnet&hl=pt\\_PT](https://play.google.com/store/apps/details?id=org.plantnet&hl=pt_PT). [Acedido: 24-Fev-2019].
- [18] «Aipoly Vision – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.aipoly.vision&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.aipoly.vision&hl=pt_PT). [Acedido: 24-Fev-2019].
- [19] «Screenshop by Craze», *App Store*. [Em linha]. Disponível em: <https://itunes.apple.com/us/app/screenshop-by-craze/id1254964391?mt=8>. [Acedido: 24-Fev-2019].
- [20] «DRAGON BALL LEGENDS – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.bandainamcoent.dblegends\\_ww&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.bandainamcoent.dblegends_ww&hl=pt_PT). [Acedido: 10-Jul-2019].
- [21] «Hearthstone – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.blizzard.wtcg.hearthstone&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.blizzard.wtcg.hearthstone&hl=pt_PT). [Acedido: 10-Jul-2019].
- [22] «Clawbert – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=com.Bow3.TheClaw&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.Bow3.TheClaw&hl=pt_PT). [Acedido: 10-Jul-2019].
- [23] «Yu-Gi-Oh! Duel Links – Aplicações no Google Play». [Em linha]. Disponível em: [https://play.google.com/store/apps/details?id=jp.konami.duellinks&hl=pt\\_PT](https://play.google.com/store/apps/details?id=jp.konami.duellinks&hl=pt_PT). [Acedido: 10-Jul-2019].
- [24] «Unity», *Unity*. [Em linha]. Disponível em: <https://unity3d.com>. [Acedido: 07-Mar-2019].
- [25] «Unity Public Relations Fact Page», *Unity*. [Em linha]. Disponível em: <https://unity3d.com/public-relations>. [Acedido: 07-Mar-2019].
- [26] «What is Unreal Engine 4». [Em linha]. Disponível em: <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>. [Acedido: 07-Mar-2019].
- [27] «GameMaker», *Yoyo Games*. [Em linha]. Disponível em: <http://www.yoyogames.com/gamemaker>. [Acedido: 11-Mar-2019].
- [28] «GameMaker Versions - GameMaker Wiki». [Em linha]. Disponível em: <http://game-maker.wikidot.com/game-maker-versions>. [Acedido: 11-Mar-2019].
- [29] «Godot Engine - Free and open source 2D and 3D game engine». [Em linha]. Disponível em: <https://godotengine.org/>. [Acedido: 07-Mar-2019].
- [30] «TensorFlow | TensorFlow». [Em linha]. Disponível em: <https://www.tensorflow.org/overview>. [Acedido: 14-Mar-2019].
- [31] «Home - Keras Documentation». [Em linha]. Disponível em: <https://keras.io/>. [Acedido: 14-Mar-2019].
- [32] «Torch | Scientific computing for LuaJIT.» [Em linha]. Disponível em: <http://torch.ch/>. [Acedido: 14-Mar-2019].
- [33] «Caffe | Deep Learning Framework». [Em linha]. Disponível em: <http://caffe.berkeleyvision.org/>. [Acedido: 14-Mar-2019].
- [34] «API Vision: análise do conteúdo de imagens | API Cloud Vision | Google Cloud». [Em linha]. Disponível em: <https://cloud.google.com/vision/>. [Acedido: 14-Mar-2019].
- [35] A. Deshpande, «A Beginner’s Guide To Understanding Convolutional Neural Networks». [Em linha]. Disponível em:

- <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>. [Acedido: 17-Jun-2019].
- [36] P. A. Koen *et al.*, *The PDMA toolbook 1 for new product development*. John Wiley & Sons, 2004.
- [37] T. Woodall, «Conceptualising “Value for the Customer”: An Attributional, Structural and Dispositional Analysis», *Acad. Mark. Sci. Rev.*, 2003.
- [38] J. Lapierre, «Customer-perceived value in industrial contexts», *J. Bus. Ind. Mark.*, vol. 15, 2000.
- [39] V. Allee, «Value Network Analysis and Value Conversion of Tangible and Intangible Assets», *J. Intellect. Cap.*, vol. 9, 2008.
- [40] T. Saaty, *The Analytic Hierarchy Process*. McGraw-Hill, 1980.
- [41] «A history of cigarette cards, trade cards and trading cards», *Cardworld*, 26-Out-2014. [Em linha]. Disponível em: <https://www.card-world.co.uk/collect/>. [Acedido: 25-Jun-2019].
- [42] H.-J. Lee, A. H. Macbeth, J. H. Pagani, e W. Scott Young, «Oxytocin: The great facilitator of life», *Prog. Neurobiol.*, vol. 88, n. 2, pp. 127–151, Jun. 2009.
- [43] M. de Icaza, *TensorFlow API for .NET languages. Contribute to migueldeicaza/TensorFlowSharp development by creating an account on GitHub*. 2019.
- [44] «JSON». [Em linha]. Disponível em: <https://www.json.org/>. [Acedido: 01-Jul-2019].
- [45] *A library for transfer learning by reusing parts of TensorFlow models.: tensorflow/hub*. tensorflow, 2019.
- [46] «MobileNets: Open-Source Models for Efficient On-Device Vision», *Google AI Blog*. .
- [47] «ImageNet». [Em linha]. Disponível em: <http://image-net.org/>. [Acedido: 03-Jul-2019].
- [48] U. Technologies, «Unity - Scripting API: WebCamTexture». [Em linha]. Disponível em: <https://docs.unity3d.com/ScriptReference/WebCamTexture.html>. [Acedido: 04-Jul-2019].
- [49] A. Ashikhmin, *Classifier script from the project TFClassify-Unity - An example of using Tensorflow with Unity for image classification and object detection. (使用 Tensorflow 和 Unity 进行图像分类和对象检测的示例): Syn-McJ/TFClassify-Unity*. 2019.
- [50] «Introduction | Photon Engine». [Em linha]. Disponível em: <https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro>. [Acedido: 05-Jul-2019].
- [51] K. Destrem, *Simple Unity Barcode Scanner. Contribute to kefnark/UnityBarcodeScanner development by creating an account on GitHub*. 2019.
- [52] «RPCs and RaiseEvent | Photon Engine». [Em linha]. Disponível em: <https://doc.photonengine.com/en-us/pun/v2/gameplay/rpcsandraiseevent>. [Acedido: 05-Jul-2019].
- [53] P. Escudeiro e J. Bidarra, «Quantitative Evaluation Framework (QEF)», *RISTI – Rev. Ibérica STI*, n. 1.



## Anexo A – Ficheiro cardInfoData\_Default.json

```
1. {
2.   "routeList": [
3.     {
4.       "routeID": 1,
5.       "routeTitle": "Porto 1",
6.       "routeDescription": "Porto é a segunda cidade e o quarto município
mais populoso de Portugal, situada no noroeste do país e capital da Área
Metropolitana do Porto, da região Norte e do Distrito do Porto. A cidade é
considerada uma cidade global gama. O município, com 41,42 km² de área, tem uma
população de 237 591 habitantes (2011) dentro dos seus limites administrativos,
sendo subdividido em sete freguesias.",
7.       "cardList": [
8.         {
9.           "cardID": 1,
10.          "cardTitle": "Casa da Música",
11.          "cardDescription": "Casa da Música é a principal sala de
concertos localizada na Avenida da Boavista, no Porto, em Portugal.",
12.          "cardQuestion": {
13.            "question": "Que sala de concertos situada na cidade do
Porto foi projetada pelo arquiteto Rem Koolhaas?",
14.            "answerkeyword": "casa da musica"
15.          },
16.          "cardColor": "#85E25B"
17.        },
18.        {
19.          "cardID": 2,
20.          "cardTitle": "Francesinha",
21.          "cardDescription": "A francesinha tem a forma de um sanduiche
e é constituída por linguiça, salsicha fresca, fiambre, carnes frias e bife de
carne de vaca, coberta com queijo posteriormente derretido. É guarnecida com um
molho à base de tomate, cerveja e piri-piri.",
22.          "cardQuestion": {
23.            "question": "Que sanduiche constitui um dos pratos
típicos da cidade do Porto?",
24.            "answerkeyword": "francesinha"
25.          },
26.          "cardColor": "#85E25B"
27.        },
28.        {
29.          "cardID": 3,
30.          "cardTitle": "Ponte de D. Luís I",
31.          "cardDescription": "A Ponte de D. Luís I ou Ponte Luiz I, é
uma ponte em estrutura metálica com dois tabuleiros, construída entre os anos
1881 e 1888, ligando as cidades do Porto e Vila Nova de Gaia (margem norte e sul,
respetivamente) separadas pelo rio Douro, em Portugal.",
32.          "cardQuestion": {
33.            "question": "Que ponte de dois tabuleiros, composta por
uma estrutura metálica, e situada sobre o rio Douro, foi projetada pelo
engenheiro belga Théophile Seyrig?",
34.            "answerkeyword": "ponte d luis"
35.          },
36.          "cardColor": "#85E25B"
37.        },
38.        {
39.          "cardID": 4,
40.          "cardTitle": "Capela das Almas",
41.          "cardDescription": "A Capela da Almas ou Capela de Santa
Catarina é uma capela situada na freguesia de Santo Ildefonso, na cidade do
Porto, em Portugal.",
42.          "cardQuestion": {
43.            "question": "Que capela teve origem numa capela em
madeira erguida em louvor de Santa Catarina?",
44.            "answerkeyword": "capela das almas"
45.          },
46.          "cardColor": "#85E25B"
47.        },
48.        {
49.          "cardID": 5,
50.          "cardTitle": "Monumento a D. Pedro IV",
51.          "cardDescription": "É uma estátua equestre da autoria do
escultor Célestin Anatole Calmels. A primeira pedra foi posta em 9 de julho de
```

```

1862. Foi inaugurado em 19 de outubro de 1866. Tem 10 metros de altura e cinco
toneladas de bronze.",
52.         "cardQuestion": {
53.             "question": "Que monumento inaugurado em 1866 constitui
uma estátua equestre onde são ilustrados vários episódios da vida de D. Pedro
IV?",
54.             "answerkeyword": "estatuad pedro iv"
55.         },
56.         "cardColor": "#85E25B"
57.     },
58.     {
59.         "cardID": 6,
60.         "cardTitle": "Igreja dos Congregados",
61.         "cardDescription": "Igreja de planta em cruz latina, composta
de nave única e capela-mor retangular. Volumes articuladas por coberturas
diferenciadas em telhado de duas águas.",
62.         "cardQuestion": {
63.             "question": "Que igreja construída no século XIX
apresenta na sua fachada azuleijos com autoria de Jorge Colaço?",
64.             "answerkeyword": "igreja dos congregados"
65.         },
66.         "cardColor": "#85E25B"
67.     },
68.     {
69.         "cardID": 7,
70.         "cardTitle": "Sé Catedral do Porto",
71.         "cardDescription": "A Sé do Porto é um edifício de estrutura
romano-gótica, dos séc. XII e XIII, tendo sofrido grandes remodelações no período
barroco (séc. XVII-XVIII). No interior conserva ainda o aspecto de uma igreja-
fortaleza, com ameias.",
72.         "cardQuestion": {
73.             "question": "Que catedral situada na cidade do Porto
representa um dos mais antigos monumentos de Portugal?",
74.             "answerkeyword": "se catedral do porto"
75.         },
76.         "cardColor": "#85E25B"
77.     }
78. ],
79.     "routeColor": "#FFFFFF"
80. },
81. {
82.     "routeID": 2,
83.     "routeTitle": "Porto 2",
84.     "routeDescription": "Porto é a segunda cidade e o quarto município
mais populoso de Portugal, situada no noroeste do país e capital da Área
Metropolitana do Porto, da região Norte e do Distrito do Porto. A cidade é
considerada uma cidade global gama. O município, com 41,42 km² de área, tem uma
população de 237 591 habitantes (2011) dentro dos seus limites administrativos,
sendo subdividido em sete freguesias.",
85.     "cardList": [
86.         {
87.             "cardID": 8,
88.             "cardTitle": "Torre dos Clérigos",
89.             "cardDescription": "A Igreja e a Torre dos Clérigos integram
uma edificação do século XVIII, de inspiração barroca, que marcou a configuração
urbana da cidade do Porto, localizada numa rua desnivelada, mas genialmente
aproveitada por Nicolau Nasoni, que conseguiu criar um edifício de referência.",
90.             "cardQuestion": {
91.                 "question": "Que monumento é considerado o ex-libris da
cidade do Porto?",
92.                 "answerkeyword": "torre dos clerigos"
93.             },
94.             "cardColor": "#85E25B"
95.         }
96.     ],
97.     "routeColor": "#FFFFFF"
98. }
99. ]
100. }

```

## Anexo B – Objeto GameJsonData

```
1. using System;
2. using System.Collections;
3. using System.Collections.Generic;
4. using UnityEngine;
5.
6. [Serializable]
7. public class GameJsonData{
8.
9.     public List<GameRouteData> routeList;
10.
11.     public GameJsonData()
12.     {
13.         this.routeList = new List<GameRouteData>();
14.     }
15.
16.     public int getRouteIDfromCard(int cardID)
17.     {
18.         int routeID = 0;
19.         foreach (GameRouteData grd in this.routeList)
20.         {
21.             if (grd.hasCard(cardID))
22.             {
23.                 routeID = grd.routeID;
24.             }
25.         }
26.         return routeID;
27.     }
28.
29.     public int totalCards()
30.     {
31.         int count = 0;
32.
33.         foreach (GameRouteData rd in this.routeList)
34.         {
35.             foreach (GameCardData cd in rd.cardList)
36.             {
37.                 count++;
38.             }
39.         }
40.         return count;
41.     }
42. }
43.
44.
45.
46.
47. [Serializable]
48. public class GameRouteData
49. {
50.     public int routeID;
51.     public string routeTitle;
52.     public string routeDescription;
53.     public List<GameCardData> cardList;
54.     public string routeColor;
55.
56.     public GameRouteData(int id, string title, string description, string color)
57.     {
58.         this.routeID = id;
59.         this.routeTitle = title;
60.         this.routeDescription = description;
61.         this.routeColor = color;
62.         this.cardList = new List<GameCardData>();
63.     }
64.
65.     public bool hasCard(int cardID)
66.     {
67.         bool result = false;
68.         foreach (GameCardData cd in this.cardList)
69.         {
70.             if (cd.cardID == cardID)
71.             {
72.                 result = true;
```

```
73.         }
74.     }
75.     return result;
76. }
77. }
78.
79. [Serializable]
80. public class GameCardData
81. {
82.     public int cardID;
83.     public string cardTitle;
84.     public string cardDescription;
85.     public QuestionData cardQuestion;
86.     public string cardColor;
87.
88.     public GameCardData(int id, string title, string description, QuestionData
question, string color)
89.     {
90.         this.cardID = id;
91.         this.cardTitle = title;
92.         this.cardDescription = description;
93.         this.cardQuestion = question;
94.         this.cardColor = color;
95.     }
96. }
97.
98. [Serializable]
99. public class QuestionData
100. {
101.     public string question;
102.     public string answerkeyword;
103.
104.     public QuestionData(string question, string answer)
105.     {
106.         this.question = question;
107.         this.answerkeyword = answer;
108.     }
109.
110. }
```

## Anexo C – Exemplo do ficheiro PlayerData.json

```
1. {
2.   "username": "VascoTeixeira",
3.   "collection": [
4.     {
5.       "routeID": 1,
6.       "routeDescription": "",
7.       "cardList": [
8.         {
9.           "cardID": 1,
10.          "isSpecial": true
11.        },
12.        {
13.          "cardID": 2,
14.          "isSpecial": false
15.        },
16.        {
17.          "cardID": 3,
18.          "isSpecial": false
19.        },
20.        {
21.          "cardID": 4,
22.          "isSpecial": false
23.        }
24.      ]
25.    },
26.    {
27.      "routeID": 2,
28.      "routeDescription": "",
29.      "cardList": [
30.        {
31.          "cardID": 8,
32.          "isSpecial": false
33.        }
34.      ]
35.    }
36.  ]
37. }
```

## Anexo D – Objeto PlayerJsonData

```

1. using System;
2. using System.Collections;
3. using System.Collections.Generic;
4. using UnityEngine;
5.
6. [Serializable]
7. public class PlayerJsonData
8. {
9.     public string username;
10.    public List<RouteData> collection;
11.
12.    public PlayerJsonData(string usern)
13.    {
14.        this.username = usern;
15.        this.collection = new List<RouteData>();
16.    }
17.
18.    public PlayerJsonData(Player player)
19.    {
20.        this.username = player.getUsername();
21.        this.collection = new List<RouteData>();
22.        if (player.getCollection().getRouteList().Count != 0)
23.        {
24.            foreach (Route route in player.getCollection().getRouteList())
25.            {
26.                RouteData rd = new RouteData(route.getRouteID());
27.                rd.routeDescription = route.getDescription();
28.                foreach (Card card in route.getCardList())
29.                {
30.                    rd.cardList.Add(new CardData(card.getCardID(),
31.                    card.isSpecialCard()));
32.                }
33.                this.collection.Add(rd);
34.            }
35.        }
36.    }
37. }
38.
39. [Serializable]
40. public class RouteData
41. {
42.     public int routeID;
43.     public string routeDescription;
44.     public List<CardData> cardList;
45.
46.     public RouteData(int id)
47.     {
48.         this.routeID = id;
49.         this.cardList = new List<CardData>();
50.     }
51. }
52.
53. [Serializable]
54. public class CardData
55. {
56.     public int cardID;
57.     public bool isSpecial;
58.
59.     public CardData(int id)
60.     {
61.         this.cardID = id;
62.         this.isSpecial = false;
63.     }
64.
65.     public CardData(int id, bool special)
66.     {
67.         this.cardID = id;
68.         this.isSpecial = special;
69.     }
70. }

```

## Anexo E – Objeto Player.cs

```
1. using System;
2. using System.Collections;
3. using System.Collections.Generic;
4. using UnityEngine;
5.
6. public class Player : MonoBehaviour {
7.
8.     private string username;
9.     private CardCollection collection;
10.
11.     public Player(string usern)
12.     {
13.         this.username = usern;
14.         this.collection = new CardCollection();
15.     }
16.
17.     public Player(PlayerJsonData jsonData)
18.     {
19.         this.username = jsonData.username;
20.         this.collection = new CardCollection();
21.         if (jsonData.collection != null)
22.         {
23.             foreach (RouteData rd in jsonData.collection)
24.             {
25.                 Route route = new Route(rd.routeID);
26.                 foreach (CardData cd in rd.cardList)
27.                 {
28.                     route.addCard(new Card(cd));
29.                 }
30.                 this.collection.getRouteList().Add(route);
31.             }
32.         }
33.     }
34.
35.     public Player() { }
36.
37.     public string getUsername()
38.     {
39.         return this.username;
40.     }
41.
42.     public CardCollection getCollection()
43.     {
44.         return this.collection;
45.     }
46.
47.     public Card GetCard(int cardID)
48.     {
49.         foreach (Route route in this.collection.getRouteList())
50.         {
51.             foreach (Card card in route.getCardList())
52.             {
53.
54.                 if (card.getCardID() == cardID)
55.                 {
56.                     return card;
57.                 }
58.             }
59.         }
60.         return null;
61.     }
62.
63.     public void addCard(GameCardData cardData, int routeID)
64.     {
65.         foreach(Route route in this.collection.getRouteList())
66.         {
67.             Debug.Log("R1: " + route.getRouteID() + " R2: " + routeID);
68.             if (route.getRouteID() == routeID)
69.             {
70.                 route.addCard(new Card(cardData.cardID));
71.             }
72.         }
```

```
73.     }
74.
75.     public void upgradeCard(int cardID)
76.     {
77.         foreach (Route route in this.collection.getRouteList())
78.         {
79.             foreach (Card card in route.getCardList())
80.             {
81.                 if (card.getCardID() == cardID)
82.                 {
83.                     card.upgrade();
84.                 }
85.             }
86.         }
87.     }
88.
89.     public int totalPlayerCards()
90.     {
91.         int count = 0;
92.         foreach (Route route in this.collection.getRouteList())
93.         {
94.             foreach (Card card in route.getCardList())
95.             {
96.                 count++;
97.             }
98.         }
99.         return count;
100.    }
101.
102.    public int totalPlayerSpecialCards()
103.    {
104.        int count = 0;
105.        foreach (Route route in this.collection.getRouteList())
106.        {
107.            foreach (Card card in route.getCardList())
108.            {
109.                if (card.isSpecialCard())
110.                {
111.                    count++;
112.                }
113.            }
114.        }
115.        return count;
116.    }
117. }
```

## Anexo F – Métricas de avaliação QEF

|                 |                   |
|-----------------|-------------------|
| <b>Dimensão</b> | Funcionalidade    |
| <b>Fator</b>    | Perfil do Jogador |

| Requisito                                    | Avaliação Métrica  | Porcentagem de Cumprimento (%) |    |  |    |   |
|--|--|--------------------------------|----|--|----|---|
|  |  | 0                              | 25 | 50   | 75 | 100                                       |
| FP1 - Criar conta de utilizador              | Na sua primeira sessão de jogo, o jogador criar o seu perfil de jogo escolhendo o seu nome de jogador.                                 | Não implementado               | -  | -  | -  | Implementado                              |
| FP2 - Iniciar sessão com conta de utilizador | Quando o jogador executa o jogo pela segunda vez, o jogo inicia a sessão de utilizador através da conta criada na primeira utilização. | Não implementado               | -  | Implementado, mas o processo não é automático, sendo necessário que o jogador volte a inserir o seu nome | -  | Implementado, sendo o processo automático |
| FP3 - Persistência dos dados do utilizador   | Todos os dados do jogo relativos ao jogador são persistidos localmente e carregados pelo sistema de forma automática.                  | Não implementado               | -  | -  | -  | Implementado                              |

|              |                   |
|--------------|-------------------|
| <b>Fator</b> | Coleção de cartas |
|--------------|-------------------|

| Requisito   | Avaliação Métrica   | Percentagem de Cumprimento (%) |  |  |    |                            |
|---|---|--------------------------------|--|--|----|----------------------------|
|   |   | 0                              | 25   | 50   | 75 | 100                        |
| FC1 - Consultar e navegar pela coleção virtual de cartas    | O jogador pode navegar e explorar a sua coleção de cartas, assim como consultar informação sobre os vários percursos.   | Não implementado               | -  | É apresentada a coleção virtual de cartas ao jogador, no entanto não é apresentada informação adicional sobre os percursos | -  | Implementado na totalidade |
| FC2 - Consultar informação relativa a uma carta             | Após selecionar uma carta, o jogador pode consultar informação relativa à mesma.  | Não implementado               | -  | -  | -  | Implementado               |
| FC3 - Transformar uma carta colecionada numa carta especial | Após selecionar uma carta, o jogador pode selecionar a opção de a transformar numa carta especial e, através da leitura ou partilha de um código QR com um amigo que possua a mesma carta na sua coleção, obtém uma versão especial da sua carta. | Não implementado               | Implementado sem a mecânica de comunicação com um amigo do jogador | -  | -  | Implementado na totalidade |

|               |      |
|---------------|------|
| <b>Factor</b> | Jogo |
|---------------|------|

| Requisito  | Avaliação Métrica  | Percentagem de Cumprimento (%)   |   |   |   |  |
|--|--|--|---|---|---|--|
|  |  | 0  | 25  | 50  | 75  | 100  |
| FJ1 - Iniciar quiz relativo a um percurso        | Após selecionar o percurso pretendido e a opção "jogar", é iniciado o quiz através da apresentação da questão relativa à primeira carta bloqueada do percurso.           | Não implementado   | -   | -   | -   | Implementado   |
| FJ2 - Responder à questão apresentada            | Depois de apresentada uma questão do quiz, quando o jogador seleciona a opção de responder, é ativada a câmara, permitindo que seja tirada uma fotografia como resposta. | Não implementado   | -   | A jogo não utiliza a câmara nativamente, sendo que o jogador tem de carregar manualmente uma fotografia capturada previamente | -   | Implementado na totalidade, com um sistema nativo de captura de fotos                      |
| FJ3 - Correção da resposta à questão apresentada | O sistema analisa a resposta do jogador e apresenta o respetivo resultado da correção.   | 0-20% de classificações positivas nos questionários quanto à exatidão das classificações | 20-40% de classificações positivas nos questionários quanto à exatidão das classificações | 40-60% de classificações positivas nos questionários quanto à exatidão das classificações                                     | 60-80% de classificações positivas nos questionários quanto à exatidão das classificações | 80-100% de classificações positivas nos questionários quanto à exatidão das classificações |
| FJ4 - Adquirir a carta relativa à questão        | Após a questão ser avaliada como correta, é adicionada uma nova carta à coleção virtual do jogador   | Não implementado   | -   | Carta é adicionada à coleção, mas não apresenta a imagem capturada pelo jogador   | -   | Carta é adicionada à coleção e apresenta a imagem capturada pelo jogador                   |

|                 |                   |
|-----------------|-------------------|
| <b>Dimensão</b> | Adaptabilidade    |
| <b>Fator</b>    | Perfil do Jogador |

| <b>Requisito</b>  | <b>Avaliação Métrica</b>  | <b>Percentagem de Cumprimento (%)</b> |           |           |           |            |
|---|---|---------------------------------------|-----------|-----------|-----------|------------|
|   |   | <b>0</b>                              | <b>25</b> | <b>50</b> | <b>75</b> | <b>100</b> |
| AP1 - Não são armazenados dados sensíveis relativos ao utilizador | Nenhuma informação de carácter sensível relativa ao jogador é armazenada, tanto localmente como remotamente | Não                                   | -         | -         | -         | Sim        |

|              |                |
|--------------|----------------|
| <b>Fator</b> | Conteúdo sério |
|--------------|----------------|

| Requisito  | Avaliação Métrica  | Percentagem de Cumprimento (%)                      |  |  |  |   |
|--|--|---|--|--|--|---|
|  |  | 0   | 25   | 50   | 75   | 100   |
| AC1 - O jogo estimula os jogadores a visitar marcos e monumentos de importância cultural | Durante o processo de resposta às questões apresentadas pelo jogo, os jogadores são estimulados a experienciar e visitar marcos e monumentos de importância cultural | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| AC2 - O jogo permite a aprendizagem de informação cultural relevante                     | O jogo permite a aprendizagem de informação cultural relevante   | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| AC3 - O jogo estimula a cooperação entre jogadores                                       | A mecânica de transformar as cartas da coleção em cartas especiais estimula a cooperação entre jogadores   | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |

|              |               |
|--------------|---------------|
| <b>Fator</b> | Versatilidade |
|--------------|---------------|

| <b>Requisito</b>   | <b>Avaliação Métrica</b>  | <b>Percentagem de Cumprimento (%)</b>               |  |  |  |   |
|--|---|---|--|--|--|---|
|  |   | <b>0</b>  | <b>25</b>  | <b>50</b>  | <b>75</b>  | <b>100</b>  |
| AV1 - As interfaces dos menus adaptam-se à resolução do ecrã dos dispositivos          | As interfaces dos menus adaptam-se à resolução do ecrã dos dispositivos, não apresentando quaisquer erros de escala ou aspeto gráfico | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| AV2 - O jogo consegue manter um bom desempenho quando executado em diferentes sistemas | O jogo consegue manter um bom desempenho quando executado em diferentes sistemas  | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |

|                 |                    |
|-----------------|--------------------|
| <b>Dimensão</b> | Usability          |
| <b>Fator</b>    | Navegação de menus |

| Requisito   | Avaliação Métrica  | Percentagem de Cumprimento (%)                      |  |  |  |   |
|---|--|---|--|--|--|---|
|   |  | 0   | 25   | 50   | 75   | 100   |
| UN1 - Os menus encontram-se bem estruturados a permitem os utilizadores aceder ao seu conteúdo de uma forma intuitiva | Os menus encontram-se bem estruturados, com uma estrutura similar, permitindo aos utilizadores aceder ao seu conteúdo de forma intuitiva | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| UN2 - Não são apresentados quaisquer erros durante a execução   | Durante o jogo, não ocorrem nem são apresentados quaisquer erros de execução ao jogador  | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| UN3 - O sistema de menus e as respetivas opções colocam o utilizador com uma sensação de controlo                     | O sistema de menus e as respetivas opções de jogo colocam o jogador com uma sensação de controlo   | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| UN4 - As ações importantes do utilizador apresentam feedback adequado   | Em quaisquer ações importantes que ocorram no jogo, é apresentado o feedback adequado ao jogador   | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |

|   |   |   |  |  |  |   |
|---|---|---|--|--|--|---|
| UN5 - Não é apresentada nenhuma informação de carácter ofensivo ao utilizador | Não é apresentada nenhuma informação ou mensagem de carácter ofensivo ao utilizador | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
|---|---|---|--|--|--|---|

|              |                   |
|--------------|-------------------|
| <b>Fator</b> | Coleção de cartas |
|--------------|-------------------|

| Requisito  | Avaliação Métrica   | Percentagem de Cumprimento (%)                      |  |  |  |   |
|--|---|---|--|--|--|---|
|  |   | 0   | 25   | 50   | 75   | 100   |
| UC1 - O sistema de coleção apresenta uma interface apelativa para o jogador                          | O sistema apresenta uma interface apelativa ao jogador                              | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| UC2 - As várias cartas da coleção encontram-se organizadas de uma forma intuitiva e de rápido acesso | As várias cartas encontram-se organizadas de uma forma intuitiva e de rápido acesso | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| UC3 - A informação das cartas é apresentada de forma clara e sem erros                               | A informação das cartas é apresentada de forma clara e sem erros                    | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |

|              |      |
|--------------|------|
| <b>Fator</b> | Jogo |
|--------------|------|

|  |   | Percentagem de Cumprimento (%)                      |  |  |  |   |
|--|---|---|--|--|--|---|
| Requisito  | Avaliação Métrica   | 0   | 25   | 50   | 75   | 100   |
| UJ1 - Todas as ações de caráter importante que ocorram durante a sessão de jogo apresentam feedback adequado             | Todas as ações de caráter importante que ocorram durante a sessão de jogo (quiz) apresentam feedback adequado         | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |
| UJ2 - O processo de análise da foto capturada pelo jogador e respectiva correção da resposta é realizado de forma rápida | O processo de análise da foto capturada pelo jogador e respectiva correção é realizado de forma rápida para o jogador | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |

|              |         |
|--------------|---------|
| <b>Fator</b> | Suporte |
|--------------|---------|

|  |  | Percentagem de Cumprimento (%)                      |  |  |  |   |
|--|--|---|--|--|--|---|
| Requisito  | Avaliação Métrica                                | 0   | 25   | 50   | 75   | 100   |
| US1 - O jogo é de fácil instalação para o público-alvo | O jogo é de fácil instalação para o público-alvo | 0-20% de classificações positivas nos questionários | 20-40% de classificações positivas nos questionários | 40-60% de classificações positivas nos questionários | 60-80% de classificações positivas nos questionários | 80-100% de classificações positivas nos questionários |

# Anexo G – Questionário de Qualidade

05/07/2019

City Lens Quality Form

## City Lens Quality Form

This form serves the purpose of assuring the quality of the game.

\* Required

### Usability Quality Evaluation

Game interface evaluation

1. How easy was to install the game? (Rate in a scale of 1-5) \*

Mark only one oval.

1      2      3      4      5

---

Very Difficult                  Quite Easy

2. Did you find the game's interface intuitive? (Rate in a scale of 1-5) \*

Mark only one oval.

1      2      3      4      5

---

Not intuitive                  Very intuitive

3. Did you find any technical errors during your gameplay session? \*

Mark only one oval.

Yes  
 No

4. At any moment of your gameplay session, did you feel you lost control over the game's menu actions? \*

Mark only one oval.

Yes  
 No

5. How would you rate the game's feedback regarding your actions? (Rate in a scale of 1-5) \*

Mark only one oval.

1      2      3      4      5

---

Awful                  Very Good

6. How would you rate the Card Collection interface? (Rate in a scale of 1-5) \*

Mark only one oval.

1      2      3      4      5

---

Awful                  Very Good

7. Did you find the Card Collection system intuitive? (Rate in a scale of 1-5) \*  
Mark only one oval.

|               |                       |                       |                       |                       |                       |                |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|               | 1                     | 2                     | 3                     | 4                     | 5                     |                |
| Not Intuitive | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Intuitive |

8. Would you classify the card's information as clear and absent of any errors? (Rate in a scale of 1-5) \*  
Mark only one oval.

|       |                       |                       |                       |                       |                       |                            |
|-------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------|
|       | 1                     | 2                     | 3                     | 4                     | 5                     |                            |
| Awful | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Quite clear/without errors |

9. When answering a game question, after taking a photo, how fast was the result given? (Rate in a scale of 1-5) \*  
Mark only one oval.

|           |                       |                       |                       |                       |                       |            |
|-----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------|
|           | 1                     | 2                     | 3                     | 4                     | 5                     |            |
| Very Slow | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Quite Fast |

10. After answering a game question with the phone camera, how accurate was the result? (Rate in a scale of 1-5) \*  
Mark only one oval.

|              |                       |                       |                       |                       |                       |               |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|
|              | 1                     | 2                     | 3                     | 4                     | 5                     |               |
| Not Accurate | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Accurate |

11. Did you find any offensive messages within the game? (racist, xenophobic, etc.) \*  
Mark only one oval.

Yes  
 No

12. Which aspects do you want to be improved on future releases (if any)?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Game content**

13. Do you think this experience could lead people to visit monuments and cultural landmarks in a tourist context?  
Mark only one oval.

Yes  
 No

05/07/2019

City Lens Quality Form

**14. Would you rather play the game alone or with other people?**

*Mark only one oval.*

Alone

With Others

**15. Do you believe the game helped or could help you learn more about monuments/landmarks or other cultural information?**

*Mark only one oval.*

Yes

No

---

Powered by  
 Google Forms

## Anexo G – Conceitos Teóricos de Análise de Valor

### New Concept Development

De um ponto de vista empresarial, o processo de inovação não é exclusivamente dependente da criação de um produto ou ideia novo, sendo que depende da sinergia do valor que fornece para os seus potenciais clientes e dos benefícios que introduz na própria organização responsável.

Geralmente, o processo de inovação é composto por três partes: FFE (*Fuzzy Front End*)[36], que constitui a etapa inicial e globalmente considerada a mais importante do processo, onde ocorre a análise e peso de oportunidades e a própria génese dos conceitos inerentes ao futuro produto de inovação, de carácter mais informal e não padronizado; NPD (*New Product Development*)[36], fase intermédia onde ocorre a definição esquematizada e padronizada dos vários conceitos, as várias atividades de desenho, desenvolvimento e controlo do produto, de modo a prepara-lo para a comercialização; fase final de comercialização e distribuição do produto.

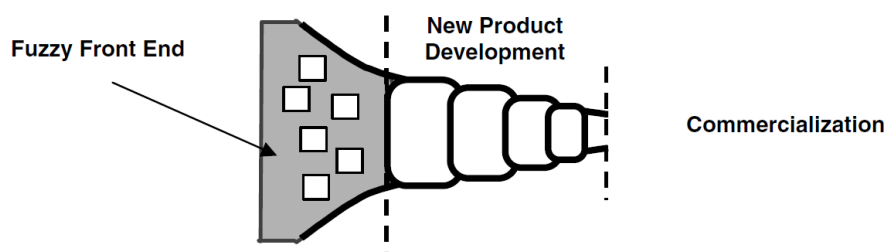


Figura 91 - Esquema representativo do processo de inovação[36]

Tendo em consideração o processo tripartido de inovação baseado em FFE, Peter Koen criou o modelo NCD (*New Concept Development*) com o intuito de estruturar e solidificar o processo de FFE, que anteriormente carecia de uma definição concreta de princípios e conceitos inerentes globalmente aceites[36].

O modelo NCD é composto por três partes[36] (representado na Figura 92):

- A parte central, denominada de Motor (*Engine*), que representa as características de liderança, cultura organizacional e estratégia de negócio que

controlam e influenciam diretamente as cinco atividades principais do processo de FFE.

- A parte interior caracterizada pelas cinco atividades principais que estruturam o processo de FFE (identificação da oportunidade, análise da oportunidade, conceção da ideia, seleção da ideia e definição do conceito).
- O conjunto dos vários fatores alheios ao controlo da própria organização que podem influenciar todo o processo de inovação, tais como os recursos da organização, fatores externos (canais de distribuição, fatores de carácter político, socioculturais, económicos, potenciais competidores, segmentos de clientes).

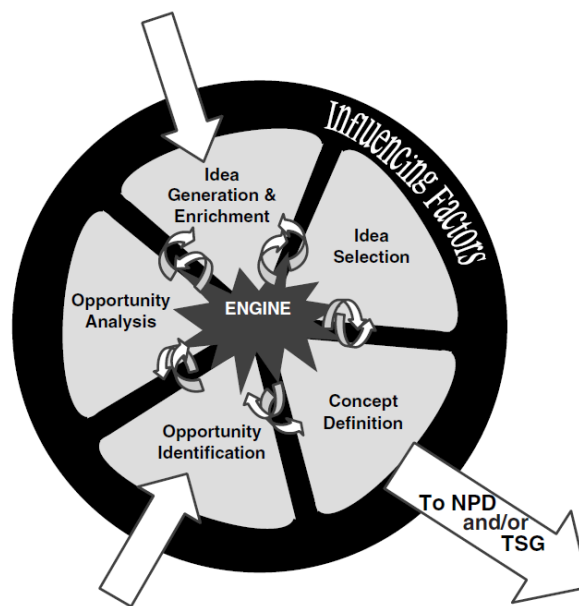


Figura 92 - Esquema representativo do modelo NCD[36]

O modelo apresenta uma forma circular, dando a entender que as várias atividades referenciadas fazem parte de um processo iterativo e não exclusivamente sequencial.