



Proximity based serious games for hospitality

ANA MARIA VIEIRA MENDES

Outubro de 2016

Proximity based serious games for hospitality

Ana Maria Vieira Mendes

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Computacionais**

Orientador: Ana Maria Madureira

Co-orientador: Marcelino Moreno

Porto, Outubro de 2016

Resumo

Cada vez mais as unidades hoteleiras procuram novas formas de aumentar as vendas e melhorar a experiência do hóspede quer no próprio estabelecimento, quer na cidade onde este se encontra. O uso de aplicações móveis neste setor é cada vez maior, mas limitam-se a um conjunto de funcionalidades similares que não exploram a experiência do hóspede na cidade. A visita pela cidade com acesso a questões, desafios e outras atividades, permitem um maior estímulo para a descoberta da cidade e o conhecimento sobre os locais visitados. O uso de pequenos jogos permite combinar a educação e o entretenimento para ajudar os visitantes a envolverem-se com a história e a cultura do seu destino, com o uso do dispositivo móvel.

Esta dissertação tem como objetivo a elaboração do módulo “proximity based serious games” e a sua integração na plataforma GuestU, produto da WeDeliverAwesomeApps. Pretende-se que o módulo permita a construção de jogos sérios para ajudar a conhecer a cidade onde os hóspedes se encontram, interagindo com o espaço envolvente.

Da análise dos resultados obtidos foi possível concluir quanto à apetência dos utilizadores para este tipo de abordagem que a implementação de Jogos Sérios no setor da hospitalidade e, em particular nesta plataforma, contribui para uma maior procura de conhecimento sobre a cidade por parte dos hóspedes, não descurando o facto dos mesmos considerarem esta abordagem apelativa e interessante.

Palavras-chave: Turismo, Hospitalidade, Jogos Sérios, Android, Tecnologias móveis

Abstract

More and more hotel units seek new ways to increase sales and improve the guest experience either in the establishment itself or in the city where it is located. The use of mobile applications in this sector is growing, but limited to a set of similar features that do not exploit the guest experience in the city. To visit the city and being challenged with questions, challenges and other activities, allow a greater stimulus to the discovery of the city and the knowledge of the places visited. The use of small games allows to combine education and entertainment to help visitors to engage with the history and culture of your destination, using the mobile device.

This thesis aims at preparing the module "proximity based serious games" and their integration into GuestU platform, product of WeDeliverAwesomeApps. It is intended that the module allows the construction of serious games to help meet the city where guests are interacting with the surroundings.

The analysis of the results it was possible to draw conclusions about the appetite of users for this type of approach to the implementation of Serious Games in the hospitality industry and on this platform, contributes to an increased demand for knowledge of the city by the guests, not forgetting the fact that they consider this appealing and interesting approach.

Keywords: Tourism, Hospitality, Serious Games, Android, Mobile Technologies

Agradecimentos

É impossível não deixar um agradecimento especial aos meus pais que me acompanharam ao longo destes anos nesta Instituição. Um obrigada ao meu namorado, André Ribeiro, que foi um pilar de incentivo para a realização desta dissertação, mesmo quando a vontade de o fazer não existia.

Não posso também de deixar um agradecimento especial à minha orientadora desta Instituição, Doutora Ana Maria Madureira, pelo apoio prestado em ambos os projetos de conclusão de Licenciatura e Mestrado. Por último, quero igualmente agradecer ao meu orientador da empresa, Marcelino Moreno, e à empresa WeDeliverAwesomeApps pela oportunidade de desenvolvimento deste projeto e pelo apoio prestado ao longo destes meses no seu desenvolvimento.

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema.....	2
1.3	Objetivos.....	2
1.4	Análise de valor.....	3
1.5	Abordagem preconizada.....	3
1.6	Resultados atingidos	3
1.7	Estrutura do documento	4
2	Contexto e Estado da Arte.....	5
2.1	Contextualização e problema	5
2.2	Análise de Valor	8
2.3	Jogos Sérios	12
2.3.1	Trabalhos relacionados.....	12
2.3.2	Fatores de sucesso para o desenvolvimento de Jogos Sérios	16
2.4	Tecnologias Móveis.....	16
2.4.1	Ecosistema de tecnologias móveis	17
2.4.2	Sistematização de tecnologias.....	18
2.4.3	Dispositivos <i>Beacons</i>	20
2.5	Sumário	22
3	Avaliação de abordagens.....	23
3.1	Jogos Sérios	23
3.2	Tecnologias Móveis.....	24
3.3	Sumário	28
4	Análise e Design da solução	29
4.1	Conceito do jogo	29
4.2	Requisitos.....	30

4.2.1	Requisitos funcionais.....	30
4.2.2	Requisitos não funcionais.....	31
4.3	Casos de uso.....	32
4.3.1	Backoffice	33
4.3.2	Aplicação móvel	36
4.1	Modelo de domínio	38
4.2	Arquitetura do sistema	39
4.3	Sumário.....	40
5	Implementação.....	43
5.1	Backoffice.....	43
5.1.1	Enquadramento	44
5.1.2	Diagrama de classes	45
5.1.3	Modelo de dados.....	46
5.1.4	Funcionalidades	48
5.1.5	Serviços para a aplicação móvel.....	55
5.2	Aplicação móvel	59
5.2.1	Considerações iniciais sobre a aplicação.....	59
5.2.2	Enquadramento	62
5.2.3	Diagrama de classes	63
5.2.4	Gestão das Conectividades	64
5.2.5	Funcionalidades	71
5.3	Sumário.....	76
6	Avaliação da solução.....	77
6.1	Grandezas e Hipóteses	77
6.2	Metodologias de avaliação	78
6.3	Resultados obtidos	79
6.3.1	Questões semelhantes aos questionários	79
6.3.2	Questões exclusivas ao módulo de jogos.....	83
6.3.3	Conclusões obtidas.....	84
6.4	Sumário.....	85
7	Conclusão e Trabalho Futuro.....	87

Lista de Figuras

Figura 1 – Ecrãs do jogo O'Munaciedd [26]	14
Figura 2 – Ecrã do jogo TravelPlot [28]	14
Figura 3 – Ecrã do jogo Time Mesh [31].....	15
Figura 4 – Ecrã do jogo Solis' Curse [32]	15
Figura 5 – Ecossistema de tecnologias móveis (Adaptado de [34]).....	17
Figura 6 – Características de <i>hardware</i> em <i>Beacons</i> (Adaptado de [37]).....	21
Figura 7 – Diagrama de casos de uso	32
Figura 8 – Modelo de domínio	38
Figura 9 – Deployment view do sistema	39
Figura 10 – Fluxograma para a configuração do módulo de jogos no Backoffice	44
Figura 11 – Diagrama de classes simplificado do Backoffice	45
Figura 12 – Modelo de dados do Backoffice	47
Figura 13 – Gestão de níveis de jogo no Backoffice.....	49
Figura 14 – Gestão de ações no Backoffice.....	50
Figura 15 – Excerto de código da página ManageAction / Base de dados para devolver línguas inseridas de uma ação.....	51
Figura 16 – Excerto de código da classe ActionController que devolve os locais disponíveis... 51	
Figura 17 – Excerto de código da classe ActionController para inserir informação de uma ação	52
Figura 18 – Códigos QR exemplo gerados no Backoffice.....	52
Figura 19 – Excerto de código da classe ManageAction que permite criar o código QR.....	53
Figura 20 – Classificação do jogo no Backoffice.....	53
Figura 21 – Gestão de campanhas no Backoffice	54
Figura 22 – Resultados das campanhas no Backoffice.....	55
Figura 23 – Excerto de código da classe JsonProtocolHandler que valida o serviço a ser invocado	57
Figura 24 – Excerto de código da classe ResponseHanlderController que devolve os dados do serviço gamelevels	57
Figura 25 – Exemplo de resposta do serviço gamelevels.....	58

Figura 26 – Excerto de código da classe JsonProtocolHandler que permite ler, converter e inserir informação sobre ações já realizadas	58
Figura 27 – Sincronização da informação e persistência na base de dados	59
Figura 28 – Pedido de ações ao Backoffice	60
Figura 29 – Excerto da classe Sync para a sincronização dos dados do jogo	60
Figura 30 - Fluxograma para realização de uma ação na aplicação móvel	62
Figura 31 – Diagrama de classes simplificado da aplicação móvel	63
Figura 32 - Diagrama de classes detalhado do serviço e conetividades	65
Figura 33 – Excerto de código que permite iniciar o serviço para detetar ações por Gps e Beacon	66
Figura 34 – Excerto de código da classe Beacon e Gps para lançar uma notificação	66
Figura 35 – Excerto de código da classe GPS para validação novas ações.....	67
Figura 36 – Excerto de código da classe Beacon para iniciar a deteção de ações por Beacons	69
Figura 37 – Excerto de código da classe GPS que permite identificar o término de uma ação .	70
Figura 38 – Excerto de código da classe GPS para despoletar um evento para uma ação “Quente e Frio”	70
Figura 39 – Ecrã para registo de utilizadores	71
Figura 40 – Excerto de código para aceder às ações já realizadas do utilizador, depois de registado.....	72
Figura 41 – Ecrã de listagem de locais e detalhes de um local, respetivamente	73
Figura 42 – Ecrã com descrição da ação.....	73
Figura 43 – Ecrã com ação do tipo “Pergunta-Resposta”	73
Figura 44 – Ecrã da ação tipo “Quente e Frio”	73
Figura 45 – Excerto de código da classe ActionActivity que guarda informação de uma ação realizada	74
Figura 46 – a) Ecrã das ações realizadas	75
Figura 47 – Excerto de código da classe GameResultsFragment para atualizar a classificação geral.....	76
Figura 48 – Respostas obtidas à questão “Quantos locais visitou na aplicação e leu efetivamente detalhes sobre o mesmo?”	81
Figura 49 – Respostas obtidas à questão “Quantos locais visitou fisicamente através da aplicação?”	81

Figura 50 – Respostas à questão “Se viajasse e o hotel onde ficasse tivesse uma aplicação deste género, usá-la-ia para visitar a cidade?”	82
Figura 51 – Resposta à questão “Considera que a aplicação contribui para o seu aumento de conhecimento sobre a cidade?”	82
Figura 52 – Resposta à questão “Esta aplicação contém um conceito de jogo incorporado, dando pequenas atividades aos utilizadores. Gosta deste tipo de abordagem?”	83
Figura 53 – Resposta à questão “Considera que o uso destas pequenas atividades o levam a procurar saber mais pela cidade onde se encontra?”	83
Figura 54 – Resposta à questão “Teve dificuldade em entender o funcionamento do jogo?” .	84
Figura 55 – Resposta à questão “Quantas atividades realizou?”	84
Figura 56 – Ecrã principal da aplicação	97
Figura 57 – Lista de rotas	97
Figura 58 – Ecrã para efetuar reservas	98
Figura 59 – Lista de pontos de interesse.....	98

Lista de Tabelas

Tabela 1 – Modelo de <i>canvas</i>	11
Tabela 2 - Sistematização de tecnologias (Adaptado de [36] [49] [50] [51]).....	25
Tabela 3 – Lista de fornecedores de <i>Beacons</i> e suas características	27
Tabela 4 – Descrição de alto nível dos casos de uso do Backoffice para gestão das campanhas	33
Tabela 5 - Descrição de alto nível dos casos de uso do Backoffice para gestão dos níveis do jogo	34
Tabela 6 - Descrição de alto nível dos casos de uso do Backoffice para gestão das ações	35
Tabela 7 – Descrição dos casos de uso da aplicação móvel para apresentação de informação	36
Tabela 8 - Descrição dos casos de uso da aplicação móvel para a dinâmica do jogo.....	37
Tabela 9 – Lista de serviços utilizados na aplicação móvel.....	56
Tabela 10 – Resultados obtidos às questões comuns a ambos os questionários.....	80

Acrónimos e Símbolos

Lista de Acrónimos

TIC	Tecnologias de Informação e Comunicação
OMT	Organização Mundial de Turismo
WDAA	WeDeliverAwesomeApps
ORM Lite	Object Relational Mapping Lite
SaaS	Software-as-a-Service
NFC	Near Field Communication
GPS	Global Positioning System
BLE	Bluetooth Low Energy
POI	Ponto de Interesse
QR	Quick Response
SDK	Software Development Kit
ARCON	Reference Model for Collaborative Network Organizations
CRUD	Create, Read, Update, Delete
UI	User Interface

1 Introdução

Neste capítulo será feita uma contextualização inicial do setor do Turismo e da empresa que suporta a realização desta dissertação. Além disso, será abordado o problema que se pretende resolver, os objetivos delineados, a respetiva análise de valor e a identificação dos clientes-alvo, e ainda a forma como se pretende resolvê-lo. Por fim, será realizada uma breve descrição da estrutura deste documento, assim como uma breve referência aos resultados obtidos.

1.1 Contexto

O setor do Turismo tem acompanhado o crescimento e evolução das Tecnologias de Informação e Comunicação (TIC) desde sempre, sendo a Internet um dos fatores mais influenciadores desse crescimento nas últimas décadas [1]. As tecnologias móveis são cada vez mais usadas em termos de atendimento ao cliente, reservas e comportamento do consumidor, sendo os dispositivos móveis parceiros cada vez mais usados pelos turistas durante a viagem [2].

É no setor do Turismo, mais propriamente no setor da Hospitalidade, que a WeDeliveAwesomeApps (WDAA) se enquadra. A WDAA tem como objetivo tornar-se numa empresa de referência no segmento de aplicações móveis para o setor do turismo e hotelaria, permitindo que as empresas possam disponibilizar tecnologia de ponta aos seus hóspedes e proporcionar uma melhor experiência enquanto os seus clientes se encontram no destino. A GuestU [3] [4] é uma das marcas através da qual a empresa atua no setor da hospitalidade. É uma plataforma que permite a criação de aplicações móveis para hotéis, *hostels* e apartamentos, fornecendo aos seus hóspedes informações sobre o hotel, roteiros, pontos de

interesse a visitar na cidade onde se encontram ou ainda eventos, serviços e/ou atividades disponíveis para serem realizadas.

1.2 Problema

Com uma variedade crescente de opções cada vez mais diversificada de hospitalidade, as unidades hoteleiras procuram novas formas de aliciar os turistas, aumentar as vendas e melhorar a experiência do hóspede quer no próprio estabelecimento, quer na cidade onde este se encontra. O uso de aplicações móveis ligadas ao conceito de hospitalidade e divulgação da cidade centram-se maioritariamente em oferecer aos seus clientes roteiros estáticos não interativos que se limitam a apresentar informação dos pontos de interesse para consulta do utilizador (fotos, descrições e, em alguns casos, informação áudio).

É assim pertinente proporcionar ao hóspede uma experiência mais enriquecedora e interativa, dando a possibilidade de se movimentar pela cidade e conhecer o que de melhor esta tem para oferecer. A visita pela cidade com acesso a pequenos jogos que atribuem pontos ao utilizador, e o coloque em competição com outros hóspedes e com o meio envolvente, permite um maior estímulo e motivação para a descoberta da cidade/localidade e o conhecimento sobre os locais visitados. Este tipo de abordagem é igualmente importante para criar um maior envolvimento dos utilizadores com os estabelecimentos hoteleiros em questão, permitindo aos mesmos a divulgação de serviços e atividades que possam ser usufruídos pelos hóspedes.

1.3 Objetivos

Esta dissertação tem como objetivo geral a elaboração de um módulo de “proximity based serious games” que permita a construção de jogos sérios entre os hóspedes alojados num mesmo local e que possibilite a execução dos mesmos quer em ambientes internos, quer em externos, interagindo com o espaço envolvente.

Tem como objetivo oferecer aos hotéis, *hostels* e apartamentos uma ferramenta que possibilite a configuração dos jogos e o modo como os seus hóspedes interagem entre eles e o meio envolvente. Por outro lado, o conceito dos jogos deve ser articulado com a divulgação das atividades, serviços e eventos realizados no estabelecimento hoteleiro, sendo também objetivo a possibilidade de configuração dos mesmos. Na perspetiva do hóspede, é objetivo contribuir

com uma aplicação que lhe permita dar a conhecer de forma mais lúdica a cidade/localidade onde se encontra.

1.4 Análise de valor

A plataforma GuestU, em conjunto com esta vertente de jogos para dar a conhecer o local onde a pessoa fica hospedada, tem como clientes-alvo hotéis, *hostels* e apartamentos. Em termos de proposta de valor para os clientes, este módulo pretende criar um maior envolvimento dos seus hóspedes com o estabelecimento em si, tornando-se assim um meio de comunicação, divulgação e promoção dos seus serviços e atividades (eventos/excursões que possam eventualmente ser organizados). Permite ainda proporcionar aos seus hóspedes uma forma lúdica e divertida para melhor conhecer a cidade/localidade onde se encontram.

Quanto aos custos para o cliente, este módulo traduzir-se-á numa quantia adicional ao atual valor praticado na aquisição da GuestU. Além disso, pode também ser necessário o investimento de mais tempo, ou até recursos adicionais para a configuração dos jogos pela cidade.

1.5 Abordagem preconizada

Para a realização deste módulo, será realizada uma pesquisa bibliográfica referente ao estado da arte sobre Jogos Sérios e as Tecnologias Móveis presentes em sistemas ligados ao Turismo. Com esta pesquisa, será feita uma análise sobre as possíveis abordagens que este tipo de sistemas poderá ter, no sentido de escolher a que mais se adequa a este problema. Por fim, perante a análise e os requisitos definidos com a empresa será desenvolvido um protótipo deste módulo integrado na plataforma GuestU, que será avaliado por um grupo de utilizadores selecionados.

1.6 Resultados atingidos

Destaca-se que o principal objetivo desta dissertação foi atingido, ou seja, o desenvolvimento do módulo de jogos sérios que permite a configuração de pequenas atividades para ajudar os hóspedes a conhecer melhor a cidade onde se encontram foi implementado com sucesso. No

entanto, uma pequena parte desse módulo não foi desenvolvido, tal como é possível observar a partir da seção de 4 - Análise e Design. Ainda assim, a avaliação da solução implementada é bem-recebida pelo grupo de pessoas inquiridas, o que permite evidenciar o sucesso da implementação.

1.7 Estrutura do documento

O documento encontra-se organizado em sete capítulos diferentes. O capítulo 2, Contexto e Estado da Arte, tem como objetivo abordar detalhes do contexto e descrever o problema desta dissertação, bem como destacar o valor acrescentado para os clientes-alvo. Neste capítulo, destaca-se ainda o Estado da Arte referente aos Jogos Sérios e Tecnologias Móveis que podem ser usadas neste contexto. O capítulo 3, Avaliação de Abordagens, pretende sistematizar as diferentes abordagens referidas no Estado da Arte, no sentido de selecionar a solução mais adequada ao problema proposto. No capítulo 4, Análise e Design da Solução, serão identificados os requisitos funcionais e não funcionais, a arquitetura do sistema proposto e o modelo de dados que irá suportar a plataforma desenvolvida. O capítulo 5, Desenvolvimento, tem como objetivo detalhar a implementação da solução proposta. O capítulo 6, Avaliação da Solução, pretende avaliar a solução implementada, através da identificação de grupos de teste e de questionários feitos a potenciais utilizadores, no sentido de perceber a sua viabilidade. O capítulo 7 de Conclusões e Trabalho Futuro apresenta as conclusões do estudo efetuado no âmbito desta dissertação, fazendo referência a trabalho futuro a ser realizado com base nos resultados obtidos. Nos Anexos, é possível observar informação adicional de suporte referenciada ao longo do documento, sempre que necessário.

2 Contexto e Estado da Arte

Neste capítulo, serão abordados detalhes sobre o contexto e problema que se pretende abordar nesta dissertação, bem como a análise de valor que este módulo apresenta para os clientes-alvo da WDAA. Por conseguinte, serão analisadas algumas soluções e/ou abordagens já existentes para a resolução do problema, assim como a análise de tecnologia relevante que pode ser utilizada neste trabalho de mestrado.

2.1 Contextualização e problema

Ao longo dos anos o setor do Turismo tem passado por um contínuo crescimento a nível mundial, sendo um motor essencial para o progresso socioeconómico internacional [5]. Dados da Organização Mundial de Turismo (OMT) de 2014 indicam que o número de turistas internacionais passou de 527 milhões em 1995 para 1133 milhões em 2014, prevendo-se um aumento para 1.8 biliões até 2030 [6].

Este setor tem acompanhado o crescimento e evolução das TIC desde sempre, sendo a Internet um dos fatores mais influenciadores desse crescimento nas últimas décadas [1]. Os avanços tecnológicos proporcionam melhorias na experiência do turista antes, durante e após a viagem. Assim sendo, atividades como o planeamento da viagem, a compra da viagem, a visita e acesso a informação de locais de interesse ou ainda a partilha constante da experiência vivida são algumas das atividades que podem ser efetuadas recorrendo às recentes tendências tecnológicas [7]. Mais especificamente, as tecnologias móveis são cada vez mais usadas em termos de atendimento ao cliente, reservas e recolha do comportamento do consumidor,

sendo os dispositivos móveis parceiros cada vez mais usados pelos turistas durante a viagem [2].

A GuestU é uma plataforma que permite a criação de aplicações móveis para hotéis, *hostels* e apartamentos. Dispõe de um Backoffice [8] onde são geridas as aplicações móveis dos clientes e toda a informação subjacente, sendo disponibilizadas aos utilizadores finais através de uma aplicação móvel (disponível no sistema operativo Android [9] e IOS [10]). Nestas aplicações, podem ser observadas informações sobre o hotel, pontos de interesse, roteiros e eventos na cidade onde se encontra, e ainda a possibilidade de efetuar reservas. No anexo 1 podem ser observadas algumas imagens de uma aplicação móvel exemplo.

Com um aumento de opções cada vez mais diversificado de hospitalidade, as unidades hoteleiras procuram novas formas de aumentar as vendas e melhorar a experiência do hóspede, quer no próprio estabelecimento, quer na cidade onde este se encontra. Neste sentido, existe uma maior aposta na inovação e, conseqüentemente, nas novas tecnologias para responder às necessidades de um cliente cada vez mais exigente.

O uso de aplicações móveis ligadas ao conceito de hospitalidade dispõe de um conjunto de funcionalidades que se centram, muitas das vezes, apenas na interação entre o utilizador e o estabelecimento onde se encontra hospedado, como por exemplo, informação do hotel, reservas de quartos e serviços, envio de *feedback* através de questionários, efetuar *check-in/check-out*, entre outras funcionalidades que podem ser observadas num estudo da eMarketer [11]. Apesar do elevado número de funcionalidades deste tipo de aplicações redirecionadas ao setor da hospitalidade, um dos principais fatores que levam as pessoas a viajar parece que é mais desvalorizado: a cultura, o património e a história de um determinado local. Atualmente, a cultura e o património são um importante fator de visita dos diferentes destinos turísticos por todo o mundo, sendo cada vez mais usados como estratégia de imagem [12]. Um estudo realizado pelo Turismo de Portugal, a turistas que visitaram Portugal entre 2013 e 2015, refere que o património histórico é a segunda maior motivação (51%) das viagens realizadas, independentemente do país escolhido [11]. Ou seja, proporcionar aos turistas formas de conhecer o local de visita além dos tradicionais guias, livros ou mapas é um importante ponto a desenvolver neste setor. No entanto, aplicações móveis para o setor da hospitalidade e divulgação das cidades centram-se maioritariamente em oferecer aos seus clientes roteiros estáticos não interativos que se limitam a apresentar informação dos pontos

de interesse para consulta do utilizador (fotos, descrições e, em alguns casos, informação áudio).

É assim pertinente proporcionar ao hóspede uma experiência mais enriquecedora e interativa, dando-lhe a possibilidade de se movimentar pela cidade e conhecer o que de melhor esta tem para lhe oferecer. A visita pela cidade com acesso a perguntas, desafios e outras atividades que atribuem pontos ao utilizador, e o coloque em competição com outros hóspedes, e com o meio envolvente permite um maior estímulo para a descoberta da cidade e o conhecimento sobre os locais visitados. O uso de pequenos jogos permite assim, combinar informação com conhecimento e entretenimento, para ajudar os visitantes a envolverem-se com a história e a cultura do seu destino, através do dispositivo móvel. Este tipo de abordagem será igualmente importante para criar um maior envolvimento dos utilizadores com as aplicações móveis onde sejam integradas, tornando-se um meio de fácil divulgação de atividades, eventos e serviços eventualmente realizados no estabelecimento hoteleiro. Consequentemente, essa divulgação pode levar a um maior uso e investimento por parte dos hóspedes.

Nesse sentido, é objetivo desta dissertação a elaboração de um módulo de “proximity based serious games” que permita a construção de jogos sérios para os hóspedes alojados num mesmo local e que possibilite a sua execução, quer em ambientes internos, quer externos, interagindo com o espaço envolvente. Este módulo será implementado na plataforma GuestU, permitindo assim a cada cliente poder configurar os seus jogos, e a forma como quer que os seus hóspedes interajam entre eles, e com a cidade onde se encontram. Além disso, este módulo terá como principal objetivo dar aos hóspedes, de um determinado estabelecimento hoteleiro, a possibilidade de conhecer de forma mais divertida o local onde se encontra.

Para o desenvolvimento deste módulo será necessário ter em atenção algumas restrições. Este será desenvolvido tendo por base a plataforma já existente da GuestU, ou seja, será dada continuidade ao Backoffice de gestão de conteúdos já existente. Assume-se que este módulo, a nível de Backoffice, será desenvolvido na *framework* ASP.Net [13], na linguagem de programação C# e a base de dados de informação será SQL Server 2012 [14]. Estando a aplicação móvel da GuestU desenvolvida para as plataformas Android e IOS, o módulo a desenvolver será igualmente numa destas plataformas. Para efeitos de demonstração, o módulo será integrado na aplicação móvel Android, com acesso a uma base de dados em SQL

Lite através do Object Relational Mapping Lite (ORM Lite) [15] que é usado para persistir a informação necessária na base de dados.

2.2 Análise de Valor

Uma proposta de valor bem definida pode ser a chave de sucesso para um dado produto ou serviço, ajudando na distinção perante a concorrência e no levantamento das necessidades e dos problemas que resolve. Além disso, na definição de valor, torna-se necessário ter em atenção a perceção do custo/benefício para o fornecedor e cliente, como sendo um fator que influencia a relação entre ambas as partes, permitindo um melhor conhecimento do cliente e do mercado [16].

Tal como já foi referido, este módulo será integrado na plataforma GuestU, comercializada como Software-as-a-Service (SaaS), e que tem como clientes-alvo hotéis, *hostels* e apartamentos. O módulo “proximity based serious games”, em conjunto com a plataforma onde se irá inserir, tem como principais atributos ser um serviço fiável, estável e dispor de suporte aos clientes. O módulo dispõe assim de uma funcionalidade adicional à plataforma existente, que é a configuração de jogos/atividades para que os hóspedes possam conhecer a cidade onde se encontram de forma lúdica e divertida. Além disso, este módulo usufrui da funcionalidade geral da plataforma de permitir conteúdos multilíngues. Para o cliente, esses atributos traduzem-se num maior envolvimento dos seus hóspedes com o estabelecimento hoteleiro, num meio de comunicação para divulgação de serviços, eventos e atividades e ainda para proporcionar aos seus hóspedes uma forma lúdica e divertida para melhor conhecer a cidade onde se encontram. Quanto aos custos para o cliente, este módulo traduzir-se-á num custo adicional ao atual valor praticado na aquisição da GuestU.

Numa perspetiva longitudinal de valor torna-se necessário ter em atenção os benefícios e custos para o cliente antes de efetuar a compra, no momento da compra ou experiência, depois da compra e após o uso e experiência. Atualmente, os estabelecimentos hoteleiros têm dificuldade em comunicar com os seus clientes de forma eficiente, principalmente quando estes estão fora do local de alojamento. A divulgação de eventos, serviços e atividades feitos no hotel são geralmente feitos com base em folhetos, *email* e alguns ainda através do envio de SMS. Ou seja, este é o atual cenário do cliente antes de adquirir este serviço e módulo. Na altura da compra e experiência, o cliente tem então acesso a uma plataforma que lhe permite melhorar

a forma como comunicam com os hóspedes e como divulgam serviços, atividades e eventos relevantes, através da realização de jogos que permita aos hóspedes conhecer melhor a cidade. O cliente tem acesso a um serviço fiável, estável, multilíngue e com disponibilidade para suporte de configuração da aplicação ou eventuais dúvidas de configuração que surjam, caso seja necessário. No entanto, esta plataforma carece de ser configurada e, portanto, poderá ser necessário investir algum tempo na configuração dos jogos e conteúdos relevantes, ou mesmo algum recurso adicional para tratar das atividades e jogos que o cliente possa querer realizar. Após a compra, o cliente poderá aceder ao *feedback* dos hóspedes, por exemplo através de questionários, ou ainda perceber se as reservas e vendas dos serviços, eventos e atividades aumentaram com a utilização deste serviço. Depois de usar e experimentar o serviço será necessário que o cliente avalie o retorno consoante o *feedback* obtido. Além disso, através da comparação do valor investido neste serviço e no número de reservas efetuadas poderá avaliar a viabilidade deste serviço.

Em relação ao negócio, considera-se que o atual modelo de negócios é uma negociação *win-win*, ou seja, ambas as partes colaboram entre si no sentido de conseguirem o máximo de benefícios de igual modo. Neste tipo de negociação é necessário definir inicialmente as expectativas e objetivos, e identificar os pontos que serão indiscutíveis durante a negociação. Por outro lado, torna-se necessário prever contraofertas que possam ser feitas ou recebidas e ainda estar preparado para o que o cliente espera [17].

Para melhor descrever este negócio foi concebido um modelo de negócio de *canvas* [17], que pode ser observado em detalhe na Tabela 1. O modelo de negócio descrito engloba a integração deste módulo na plataforma GuestU. Em termos de proposição de valor, o projeto, em conjunto com este módulo em particular pretende: ser um serviço estável, fiável e com bom desempenho; ser um meio de comunicação e promoção de serviços, eventos e atividades; obter um maior envolvimento dos seus hóspedes com o estabelecimento em si; oferecer conteúdos e traduções na aplicação móvel; ser uma forma lúdica e divertida para os hóspedes de conhecer a cidade onde estão hospedados. O relacionamento com o cliente será feito através de um *site* oficial, de *marketing* nas redes sociais, da presença em feiras e eventos de turismo ou ainda através do contacto presencial. A nível de estrutura de custos, foi feito um levantamento muito simples do que será necessário para este negócio. Custos com o escritório, material administrativo, equipa de vendas, desenvolvimento de conteúdos, gestão e desenvolvimento

da plataforma e ainda toda a infraestrutura de *hardware* e *software* necessários são alguns dos custos a ter em atenção.

Em termos de fluxo de receitas, este módulo integrará a plataforma GuestU, tal como já foi referido. Nesse sentido, o atual serviço é vendido como SaaS, através do pagamento de uma mensalidade. Este módulo refletiria um suplemento ao valor da plataforma GuestU, consoante o número de quartos de hotéis, isto é, consoante a ocupação que esse estabelecimento consiga ter. Outro ponto importante no fluxo de receitas, e que atualmente já é praticado na plataforma, é a comissão por reservas efetuadas através da aplicação móvel de eventos, atividades ou serviços do hotel.

A criação de valor pode ser analisada através do Modelo Conceptual para Decomposição do Valor para o Cliente, segundo quatro conceitos: formas de valor e posições de valor temporal, redes de valores, o conceito de ativos endógenos e exógenos da empresa, através do *Reference Model for Collaborative Network Organizations* (ARCON), e o conceito de benefícios e custos percebidos tanto pela empresa como pelo cliente. Numa primeira fase, é necessário construir a rede de valores da empresa, através da identificação das entregas tangíveis e intangíveis e os ativos (endógenos e exógenos) construídos e/ou utilizados no fornecimento do produto final. Torna-se assim possível perceber que o valor para o cliente pode ser dividido em componentes mais simples, integrando o valor percebido pelos membros da empresa para dada posição no tempo. Numa segunda fase, o objetivo é obter mais informações sobre a sua perceção de benefícios/custos junto da empresa cliente, num determinado período de tempo e posição. Através da seleção dos ativos mais relevantes para perceber que resultados serão utilizados para avaliar a forma como o cliente percebe a proposição de valor da empresa. A última fase requer uma análise da perspetiva da empresa e da perspetiva do cliente. Analisando os ativos/fornecimento de produtos e benefícios/custos percebidos na perspetiva da empresa em conjunto com a perceção do cliente em relação aos resultados relevantes dos benefícios/custos percebidos permite assim o ajuste da proposição de valor da empresa [18].

Tabela 1 – Modelo de *canvas*

Parceiros chave	Atividades chave	Proposições de valor	Relacionamento com cliente	Segmentos clientes
Empresas de conteúdos (TimeOut); Empresas de <i>software</i> especializadas; Empresas de comunicação.	Plataforma de construção das aplicações móveis; Suporte ao cliente;	Maior envolvimento dos seus hóspedes com o estabelecimento em si; Serviço estável, fiável com bom desempenho Meio de comunicação e promoção dos serviços, eventos e atividades; Forma lúdica e divertida de conhecer a cidade onde se encontra hospedado; Oferecer conteúdos e traduções à partida com a aplicação (serviço chave na mão).	Marketing em redes sociais; Site oficial.	Hotéis, <i>Hostels</i> e Apartamentos; Mercado Nacional e Internacional.
	Recursos chave		Canais	
	Equipa de desenvolvimento e produto; Equipa de gestão; Equipa de vendas/suporte e <i>marketing</i> ; Equipa de conteúdos.		Telefone e contacto presencial; Site oficial; Redes Sociais; Presença em eventos e feiras de Turismo.	
Estrutura de custos		Fluxo de receitas		
Escritório e custos associados Equipa Fornecedores Infraestrutura de <i>hardware</i> e <i>software</i>		<i>Software</i> as a service Mensalidade, variável por dimensão do cliente (número de quartos) Comissões de reservas de serviços e atividades		

2.3 Jogos Sérios

Um jogo é um contexto estruturado, onde um ou mais jogadores têm de ultrapassar um conjunto de desafios de modo a atingir um objetivo: ganhar. Um conjunto de regras devem ser respeitadas e, no caso de não serem cumpridas, constituem um erro, o que implica uma determinada punição [19]. O sucesso dos jogos de computador, impulsionado pelos avanços tecnológicos, deram lugar a novos tipos de jogos, tais como os jogos sérios [20].

Em 2006, Michael e Chen definem Jogos Sérios como *“um jogo em que a educação (...) é o principal objetivo, ao contrário do entretenimento”* [21]. Mais recentemente, Carvalho et. al definem jogos sérios como *“jogos (digitais) usados para outros fins que não o entretenimento”* [22]. Os autores salientam a finalidade educativa/pedagógica/de aprendizagem e não necessariamente lúdica de um jogo sério, sendo a motivação e o envolvimento/compromisso os seus efeitos mais marcantes [22].

2.3.1 Trabalhos relacionados

Os Jogos Sérios têm sido usados em diversas áreas de aplicação: saúde [23] [24], militar [23] [24], educação [23] [24], cultura/património/história [23] [24] [22], marketing [23] [24], entre outras. Sendo objetivo desta dissertação o desenvolvimento de um módulo de construção de jogos sérios que permita ajudar os turistas a conhecer melhor a cultura/património da cidade onde se encontram, será dada especial atenção aos desenvolvimentos já realizados na área da cultura, património e história.

Mortaraa et. al analisaram uma vasta gama de Jogos Sérios desenvolvidos na área do património cultural e apresentam um conjunto de géneros de jogos que podem ser usados para o desenvolvimento de Jogos Sérios nesta área. Assim sendo, os Jogos Sérios surgem no património cultural numa variedade de categorias: ação, estratégia, simulação, quiz/puzzle e aventura [25]:

- **Jogos de ação:** São a categoria menos representativa num contexto de património cultural. Estes exigem movimentos precisos e rápidos para executar tarefas ou evitar obstáculos que dificilmente são relacionados ao ganho cognitivo. Podem ser incluídos

num jogo sério como sub-tarefas ou mini-jogos, no entanto, o ganho cognitivo é relativamente baixo, acabando por prevalecer mais o envolvimento do jogador do que propriamente a aquisição de conhecimento;

- **Jogos de estratégia:** Implicam naturalmente uma reflexão e um planeamento cuidadoso para alcançar a vitória. No entanto, os autores consideram que esta categoria é mais adequada para melhorar o pensamento tático e não realmente ensinar;
- **Jogos de simulação:** Geralmente usados para simular os aspetos reais. O mecanismo de simulação é o mais adequado para fazer o jogador apreciar as consequências dos seus atos no jogo ou apreciar as suas tomadas de decisão. Podem ser também usados para aumentar a consciência sobre questões afetivas e morais, ou seja, o jogador acaba por "encarnar" outra personagem, vivendo as dificuldades e injustiças que lhe são transmitidas;
- **Quiz/Puzzles:** Os quiz são geralmente propostos sob a forma de perguntas e respostas onde é suposto o jogador aprender, ser cativado ou consciencializado para um determinado assunto a partir da informação adicional fornecida pelo jogo. Por outro lado, os jogos de puzzle funcionam através da resolução de enigmas de lógica, navegando por labirintos ou combinando peças. Os autores indicam ainda que os puzzles são bastante usuais em jogos sérios nesta área em particular;
- **Jogos de aventura:** São adequados para implementar a abordagem “aprender fazendo”, onde o jogador aprende através da construção de conhecimentos ao realizar uma determinada atividade. Os jogadores têm tendência a recordar-se mais do conhecimento adquirido através da conclusão de uma atividade por eles realizada, do que pela informação fornecida diretamente.

O objetivo dos Jogos Sérios relacionados com o património cultural e a história não é apenas fornecer alguma educação cultural. No fundo eles também conseguem promover e sensibilizar o público-alvo para a preservação do espaço ou da cultura envolvida no jogo [23]. Os trabalhos que se seguem pretendem mostrar a diversidade de jogos sérios que podem existir no setor do turismo, na área cultura mais especificamente. Será dada especial atenção a jogos

desenvolvidos para plataformas móveis, por este ser o principal âmbito desta dissertação. Além dos referidos, outros podem ser encontrados em [20] [24] [25].

2.3.1.1 O'Munaciedd

Jogo sério baseado em localização, projetado para crianças [26] que tem como objetivos: estimular o desenvolvimento cognitivo através de atividades com a finalidade de melhorar a sua concentração, memorização e habilidades de resolução de problemas; fornecer informações sobre o património artístico e cultural da cidade de Matera, em Itália (Figura 1) [27].



Figura 1 – Ecrãs do jogo O'Munaciedd [26]

Uma criança que o joga tem de visitar os locais históricos da cidade de Matera, seguindo as pistas e tesouros deixados por Munaciedd (a personagem principal) com o objetivo de encontrar o seu chapéu. Através de um dispositivo móvel e, graças à deteção em tempo real da localização, as crianças são guiadas ao longo de uma rota turística através da velha cidade à procura de pistas. Este jogo foi desenvolvido para dispositivos móveis IPod, iPhone e iPad [27].

2.3.1.2 TravelPlot [28]

Este jogo permite aos turistas visitar a cidade do Porto de forma interativa, ao longo de nove capítulos que correspondem a eventos históricos que ocorreram na cidade (Figura 2). Peter, um turista inglês, tem a missão de salvar o vinho do Porto. Para isso, terá de descobrir a localização da taça de Baco e o restante tesouro escondido pelos Cale, em várias localizações da cidade do Porto ao longo dos séculos. Só com a visita aos locais (42 no total, espalhados pelos nove capítulos) é que esta missão poderá ser bem-sucedida. Quando chega a um determinado local o utilizador tem de se posicionar em frente à fotografia do local e fazer *check-in*, para tentar encontrar o tesouro e receber informações adicionais sobre esse local [29].



Figura 2 – Ecrã do jogo TravelPlot [28]

Em termos técnicos, este jogo baseia-se no posicionamento fornecido pelo GPS, tendo ainda uma componente de redes sociais, permitindo partilhar os locais que visita através do

Facebook, Twitter ou *email*. Encontra-se desenvolvido para as plataformas Android (disponibilizada gratuitamente) e IOS, tendo também um *website* onde todo o conceito do jogo é explicado [28].

2.3.1.3 Time Mesh [30]

Jogo Sério de aventura para aprendizagem da história, da cultura e das relações sociais de regiões europeias, através de três cenários diferentes relacionados com as Descobertas Marítimas, a Revolução Industrial e a Segunda Guerra Mundial (Figura 3). Este jogo tem como objetivos de aprendizagem [31]: compreender as datas, o vocabulário e as convenções que descrevem períodos



Figura 3 – Ecrã do jogo Time Mesh [31]

históricos; construir um quadro cronológico dos períodos históricos representados no jogo; entender as ideias, crenças e atitudes das antigas sociedades e como estas influenciaram o mundo e, por fim, analisar e explicar os motivos e os resultados dos eventos históricos, situações e mudanças.

O jogo dispõe de uma plataforma *online*, multilíngue, onde é possível aos alunos inscreverem-se gratuitamente e formarem as suas equipas. Este jogo foi testado e usado em várias escolas de diferentes países (Portugal, Espanha, Inglaterra, Eslovénia, Estónia e Bélgica).

2.3.1.4 Solis' Curse [32]

Jogo Sério 3D, desenvolvido para público juvenil, com o objetivo de dar a conhecer o Panteão Nacional de forma lúdica (Figura 4). O jogo é constituído por cinco níveis,



Figura 4 – Ecrã do jogo Solis' Curse [32]

cada um com uma questão/enigma diferente e que abre portas para o nível seguinte.

O jogador não existe fisicamente no jogo, ou seja, não há uma interação direta com o cenário do jogo, apenas com o agente virtual através da configuração do reconhecimento de voz para uma interação pergunta-resposta. Sendo assim, é o agente que garante a passagem entre os

diferentes níveis. Importante ainda salientar que cerca de 1400 pessoas experimentaram o jogo entre Junho e Dezembro de 2011, confirmando assim a aplicabilidade prática deste [32].

2.3.2 Fatores de sucesso para o desenvolvimento de Jogos Sérios

Laamarti et al [23] apresentam um conjunto de fatores que podem ser tidos em atenção para a conceção de jogos sérios de sucesso. Os fatores que se seguem foram conseguidos através da análise destes autores:

- Incorporação da música no jogo, o que ajudará a motivar os jogadores;
- Orientação dos jogadores para dentro do jogo, ou seja, fornecer orientações/conhecimentos necessários, especialmente nas fases iniciais do jogo, não deixando os jogadores sentirem-se perdidos ou confusos;
- Evitar consequências negativas no jogo, como resultado do fraco desempenho do jogador;
- Manter o jogador informado do seu desempenho, por forma a ajudar na motivação das pessoas envolvidas;
- Jogos multijogador são mais motivantes e envolventes do que jogos de um único jogador;
- O desafio é o elemento do jogo que mais aumenta o envolvimento do utilizador. Um nível de desafio muito alto poderá resultar em desmotivação, assim como níveis muito baixos tornariam o jogo desinteressante.

2.4 Tecnologias Móveis

Os dispositivos móveis são considerados atualmente uma plataforma segura de investimento para desenvolvimento dos Jogos Sérios. Permitem aos utilizadores participarem de um processo de aprendizagem contínuo, estando disponíveis para serem usados onde quer que seja. O Global Positioning System (GPS), a câmara e os ecrãs de alta definição, a capacidade de reconhecimento de imagem e a ligação à Internet são pontos fortes associados a dispositivos que são considerados pela literatura como potenciais vantagens [27].

Carvalho et. al [22] considera que o futuro dos Jogos Sérios passa pelos fatores: uso de plataformas de comunicação emergentes (*smartphones, tablets*, computadores, consolas de

jogos) amplamente utilizados por todos; conteúdo mais flexível e personalizado de acordo com o perfil do utilizador, ou seja, preferências pessoais ou conhecimento prévio. George et. al [33] considera também que o uso de tecnologias móveis em jogos sérios deve ser capaz de extrair, interpretar e utilizar informações contextuais adaptando assim as funcionalidades ao atual contexto.

2.4.1 Ecosistema de tecnologias móveis

Benkendroff et. al [34] apresentam um ecossistema de tecnologias móveis no contexto do turismo, de modo a explicar melhor o seu impacto neste setor (Figura 5). Os principais componentes deste ecossistema são os dispositivos móveis, o contexto, os fornecedores de conteúdos e rede, os viajantes e os amigos/companheiros.

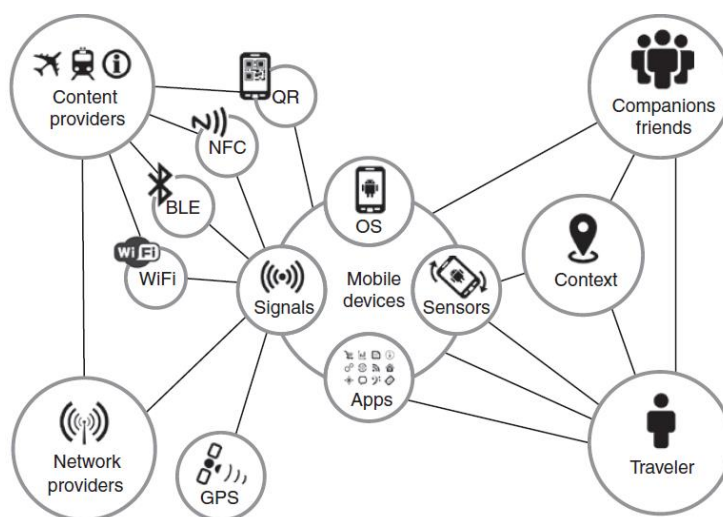


Figura 5 – Ecosistema de tecnologias móveis (Adaptado de [34])

Os **dispositivos móveis** recolhem e combinam informação de uma variedade de *inputs*, como as tecnologias de deteção (por exemplo, câmara, acelerómetro ou bússola) e as tecnologias de sinalização (por exemplo, Bluetooth ou WiFi) para usá-los em aplicações que executam funções específicas. O GPS, os *smartphones* e os *tablets* são os dispositivos mais usados pelos turistas. Além disso, as tecnologias de deteção e de sinalização trabalham em conjunto para recolher **informação contextual** (contexto) sobre o espaço, o tempo, o ambiente, o próprio utilizador e o seu contexto social de modo a apresentar informação mais relevante para o turista. Os **fornecedores de conteúdos** podem adquirir informações sobre viagens, sítios, mídias sociais ou dados de terceiros, como meteorologia e conversões monetárias. Os utilizadores devem inscrever-se num **fornecedor de rede** ou operadora de telecomunicações para ter acesso a

serviços de voz e Internet no dispositivo móvel. Com isso, poderá aceder a múltiplas aplicações móveis que dependem da Internet para mostrar informações ao utilizador. Finalmente, os atores deste ecossistema são o viajante e os turistas em geral. O **viajante** é aquele que interage com o dispositivo móvel no contexto de viagens. Os restantes turistas ou mais especificamente os **companheiros/amigos** usam os dispositivos móveis para interagir uns com os outros, sendo este o elemento que incorpora aspetos sociais da vida quotidiana na experiência da viagem.

2.4.2 Sistematização de tecnologias

Reunido todo o ecossistema de tecnologias móveis que podem ser usadas no contexto do turismo, apresenta-se de seguida uma breve explicação das principais tecnologias que poderão influenciar positivamente o desenvolvimento do módulo de Jogos Sérios. Os dispositivos móveis contam com um conjunto de tecnologias de sinalização e deteção que ajudam a fornecer *inputs* e dados contextuais. Os *smartphones* e *tablets* podem assim recorrer ao uso de tecnologias de sinalização [34] tais como o Near Field Communication (NFC), o Bluetooth, o Wifi, o GPS e as tecnologias de deteção [34] como a câmara (usada para ler códigos Quick Response (QR)), o acelerómetro ou a bússola. A informação captada deste tipo de tecnologias poderá ser usada para que haja uma maior interação e ligação entre os utilizadores, o dispositivo móvel e o ambiente que os rodeia.

Assim sendo, segue-se uma breve explicação sobre cada uma das tecnologias de sinalização acima referidas e sobre os códigos QR, considerados importantes também neste tipo de aplicações.

2.4.2.1 Near Field Communication (NFC)

O NFC é um tipo de comunicação padrão, que nos dispositivos móveis mais recentes permite uma comunicação bidirecional, podendo o dispositivo móvel atuar como recetor ou emissor. Este tipo de tecnologia permite aos dispositivos comunicar com etiquetas NFC em curtas distâncias, sendo a informação transmitida automaticamente [35] [34]. Os dispositivos NFC podem ser usados em sistemas de bilheteiras eletrónicas, exposições e atrações, partilha de informações, entre outros [34].

2.4.2.2 Bluetooth

O Bluetooth é uma tecnologia sem fios de curto alcance que permite a troca de informação. Semelhante ao NFC mas com um alcance muito maior. Os *beacons* são dispositivos de baixo consumo de energia que usam a última versão do Bluetooth, o Bluetooth Low Energy (BLE). Informações mais detalhadas sobre estes dispositivos serão dadas na seção 2.4.3. Num contexto de viagens, esta tecnologia pode ser usada para fornecer informações turísticas localizadas, interpretação e orientação, de modo a proporcionar uma experiência mais rica [34].

2.4.2.3 Wifi

Wifi é uma tecnologia que permite aos dispositivos móveis trocar informação ou ligação sem fios à Internet. Essa ligação estabelecida entre o dispositivo e o aparelho de emissão de sinal é variante na sua distância, dependendo se é um dispositivo de emissão de sinal de interior ou exterior. Este tipo de redes são usadas em ambientes domésticos, mas também por hotéis, aeroportos, transportes públicos, cafés e espaços públicos [34].

2.4.2.4 Global Positioning System (GPS)

O GPS é um sistema de satélite baseado no espaço que proporciona a localização e informação em qualquer lugar. Estes dispositivos podem ser usados para análise de congestionamento do trânsito, navegação, limites de velocidade, apresentação de pontos de interesse (Poi) ou também para atividades recreativas ao ar livre (montanhistas, BTT) [35] [34].

No setor do turismo, estes dispositivos são usados principalmente por motoristas, ciclistas e pedestres para encontrar atrações e outros pontos de interesse. No entanto, a utilização de dispositivos de navegação pode também influenciar negativamente a experiência de viagem, porque muitas vezes o caminho mais rápido não é necessariamente o mais seguro ou atrativo [34].

2.4.2.5 Quick Response (QR)

Os códigos QR são representações de informação sobre a forma de uma imagem. A vantagem deste tipo de representação de informação é que não requer nenhum componente especial. Uma simples câmara no telemóvel permite a sua leitura, assim como a sua representação pode ser feita em papel [36]. No setor do turismo estes códigos podem ser usados para fornecer

informações aos turistas, os hotéis podem criar guias ou podem ainda ser colocados pelo destino turístico um conjunto de QRs que forneçam informações sobre o contexto histórico e cultural da região [34].

2.4.3 Dispositivos *Beacons*

Os *Beacons* são pequenos dispositivos, usados essencialmente como uma tecnologia de localização interna, não sendo invalidado o seu uso em ambientes externos. A tecnologia de comunicação subjacente ao *Beacon* é o BLE, parte do novo *standard* Bluetooth 4.0. O BLE foi projetado para transferir pequenas quantidades de informação com uma baixa taxa de transferência, sendo o principal benefício o baixo consumo de energia [37]. Estes dispositivos têm surgido por toda parte e cada vez mais é visível o seu impacto, principalmente no setor do retalho. A ABI Research [38] prevê que o mercado de vendas ultrapasse os 400 milhões de unidades em 2020 [39].

2.4.3.1 Limitações do mercado

Uma das grandes limitações ao uso deste tipo de tecnologia passa pelo facto dos dispositivos móveis ainda não estarem todos equipados com a tecnologia BLE. Estudos indicam que em 2015, 24% dos *smartphones* estão capazes de comunicar com *Beacons*, esperando-se um aumento significativo para 67% em 2016 [37].

Outra limitação é o sistema operativo dos dispositivos móveis. Dados de 2014 indicam que praticamente todos os dispositivos da Apple estão aptos. No entanto, no sistema operativo Android apenas 12,1% têm a versão 4.3 ou superior, sendo este um requisito obrigatório. Por outro lado, nos dispositivos com o sistema operativo Windows, apenas a versão Windows 8 ou superior tem a possibilidade de suportar BLE, o que se verifica em 50% dos dispositivos [37].

2.4.3.2 Requisitos de Hardware e Especificações

Atualmente existe uma grande oferta em relação aos dispositivos *Beacons* [40], tornando assim a escolha mais complexa. Um estudo elaborado pela GSMA [41] em 2014 refere um conjunto de requisitos de mercado que considera ser importante ter em atenção na análise de mercado por parte das organizações. O *hardware*, a gestão de serviços, os dados de localização e a análise de dados são os pontos salientados. No contexto desta dissertação, consideram-se os

requisitos de *hardware* como sendo de maior relevância para a decisão dos dispositivos a usar [37]. Existe um conjunto de características e guias que devem ser tidas em atenção durante o processo de decisão, podendo ser observadas na Figura 6.

O tempo de **bateria** de um *Beacon* depende do respetivo fabricante, estando relacionado com o número de transações enviadas por segundo. Em alguns casos é possível aumentar ou reduzir a frequência de envio de transações, podendo assim reduzi-las para aumentar a bateria, mas por vezes trazendo para o utilizador final uma pior usabilidade. Outro fator que influencia o tempo de bateria é a potência da transmissão de sinal, que pode ser configurada de várias maneiras.

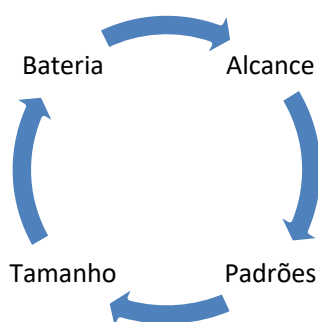


Figura 6 – Características de *hardware* em *Beacons* (Adaptado de [37])

Em relação ao **alcance**, testes efetuados concluíram que os *Beacons* deveriam ser configurados para emitir informação até 20 metros, cobrindo eficazmente uma área com um diâmetro de entre 30 e 40 metros [37]. No que diz respeito ao **tamanho** do dispositivo, os *Beacons* devem ser o mais discretos e pequenos possíveis para que passem despercebidos às pessoas, de modo que estes não corram riscos de serem roubados.

Os **padrões** dizem respeito às diferentes especificações existentes, bem como todo o conjunto de Software Development Kit (SDK) disponíveis tanto para a plataforma Android como IOS. Em relação aos SDKs existe uma variedade de possibilidades que podem ser usadas consoante a escolha do fabricante (Radius Networks [42], Estimote [43], StickNFind [44], entre outros). No que diz respeito às especificações, apesar do iBeacon funcionar inicialmente apenas com equipamentos da marca Apple, outras marcas começaram a desenvolver produtos semelhantes. Nesse sentido existem atualmente outras especificações além da iBeacon [45], nomeadamente a Eddystone [46], criada pela Google. A especificação **iBeacon** é composta por um identificador que está dividido em três seções: *UUID*, *Major* e *Minor*. Esta especificação

contém ainda um último elemento, o *TXPower*, que é usado para determinar a proximidade a partir do Beacon. A Apple explica esta divisão com um exemplo de uma cadeia de lojas: um UUID geral que representa a loja no mundo, o *Major* identifica cada país onde a loja está presente e o *Minor* identifica cada secção dessa loja nesse país [45]. Tal como o **IBeacon**, a **Eddystone** contém também um identificador (Eddystone-UID) dividido em duas secções: *Namespace* e *Instance* [47], que permite a mesma divisão estrutural do **IBeacon**. Este formato permite ainda outros pacotes de dados além do Eddystone-UID, nomeadamente o Eddystone-URL e o Eddystone-TLM. O Eddystone-URL contém um único campo que é o URL que pode ser de tamanho variável. Difere do UID por não ser necessário ao telemóvel receber um identificador e com ele dar alguma informação ao utilizador. Neste caso, o URL é recebido diretamente no telemóvel para poder ser acedido. Por outro lado, o Eddystone-TLM pode ser usado em conjunto com o UID ou com o URL para dar informações sobre a voltagem da bateria, temperatura do dispositivo ou as contagens de pacotes de transmissão [48].

2.5 Sumário

Ao longo deste capítulo foram abordados os principais conceitos subjacentes no âmbito desta dissertação. Contextualizou-se o setor no turismo, o problema proposto e a análise de valor para os clientes-alvo. De seguida foi feito um enquadramento dos Jogos Sérios a nível de trabalhos relacionados na área da cultura, história e património cultural e de algumas características deste tipo de sistemas. Por fim, foi analisado o impacto das tecnologias móveis no setor dos Jogos Sérios, assim como uma sistematização das principais tecnologias que podem ser usadas para o desenvolvimento do módulo de proximidade em jogos sérios.

3 Avaliação de abordagens

Neste capítulo, pretende-se fazer uma avaliação das abordagens especificadas no capítulo do Contexto e Estado da Arte em relação aos Jogos Sérios. Será feita também uma sistematização das tecnologias móveis analisadas, no sentido de identificar as que mais se adequam a este módulo.

3.1 Jogos Sérios

No que diz respeito à análise feita de Jogos Sérios, não só a nível de trabalhos relacionados, como também de características positivas que estes podem agregar para serem bem-sucedidas, os fatores usados para a avaliação são essencialmente os requisitos/objetivos do desenvolvimento deste módulo, definidos pela empresa. Foi analisado um conjunto de trabalhos com o objetivo de mostrar a diversidade de critérios e de aplicações diferentes que estes jogos podem ter: diferentes públicos-alvo, diferentes plataformas de desenvolvimento (dispositivos móveis, *web*), diversidade de tecnologias usadas (sistemas baseados em localização, reconhecimento de voz, modelação 3D) ou ainda diferentes objetivos (aprendizagem da cultura, de uma cidade ou de um museu/monumento em particular).

Comparando com os objetivos desta dissertação, verifica-se que não existe um jogo com as características exatas pretendidas, ou seja, um jogo que possa ser configurado pelos clientes para criar pequenas atividades/jogos que lhes permitam conhecer a cidade onde o estabelecimento se encontra localizado. Salienta-se que cada um dos jogos analisados está redirecionado ou para um edifício específico, uma cultura ou sobre um excerto da história em

particular. Os jogos O'Munaciedd [27] e TravelPlot [28] são talvez as que mais se aproximem do pretendido, por serem desenvolvidos para dispositivos móveis, recorrendo ao uso do GPS para deteção da localização e com a finalidade de dar a conhecer a história de uma cidade. No entanto, são aplicações que estão associadas a uma cidade específica, não podendo ser um sistema mais generalizado como aquilo que se pretende. Além disso, apesar do jogo O'Munaciedd ser redirecionado para crianças, tem em comum o facto de realizar pequenas atividades para dar a conhecer algo, ao contrário do TravelPoint, que apenas se limita a mostrar informação com base num *check-in* quando o utilizador chega a um local desejado. No entanto, é importante salientar a competição e a envolvimento que o TravelPlot pode criar nos seus utilizadores, pela criação de uma temática de caça ao tesouro para dar a conhecer a cidade do Porto.

A análise feita por Mortara et. al [25] sobre as possíveis categorias de jogos que podem ser integradas neste conceito (ação, estratégia, simulação, quiz/puzzle e aventura) será tida em atenção na construção dos jogos/atividades para este módulo. Ou seja, pequenas atividades de quiz ou puzzles poderão ser realizadas para dar a conhecer um determinado ponto da cidade. Pequenas atividades de aventura podem também ser usadas para despertar o interesse dos visitantes em descobrir determinados locais e a entrar na dinâmica do jogo. Por outro lado, o estudo realizado por Laamarti et al [23] sobre os fatores que devem ser tidos em atenção para a conceção de jogos sérios serão igualmente tidos em conta. Evitar consequências negativas no jogo, manter o jogador informado do seu desempenho e desafiá-lo à descoberta do conhecimento são apenas alguns dos aspetos que podem ser usados para a realização deste módulo. Aliás, tal como o autor [23] refere, o desafio é o fator que mais aumenta a imersão no jogo, o que contribuirá para um maior envolvimento do hóspede com a aplicação móvel e, consequentemente, com o estabelecimento hoteleiro em si.

3.2 Tecnologias Móveis

Tal como foi referido na seção 2.4, existem diversas tecnologias que podem ser utilizadas na área do turismo móvel e que são pertinentes no contexto dos jogos interativos, entre o hóspede e o meio envolvente. Para a avaliação das tecnologias a usar convém ter em atenção a alguns fatores: os jogos podem ser realizados em ambiente interno e externo; não deveriam ser escolhidas muitas tecnologias diferentes, isto porque a configuração dos jogos/atividades pode

ser feita pelos clientes e o excesso de diferentes tecnologias pode ser um entrave à sua utilização, pela dificuldade que pode causar na sua configuração; a compatibilidade dessas tecnologias com a maior diversidade de dispositivos móveis; e o preço em geral desse tipo de tecnologia.

A Tabela 2 apresenta um breve resumo das principais tecnologias descritas na seção anterior, o Wifi, o *Beacon*, o GPS, o NFC e os códigos QR. Desta tabela, fatores como o custo, o alcance, o consumo de bateria, o tipo de localização e a compatibilidade são os principais aspetos a destacar. Estas características, em termos de tecnologias móveis, deverão ser analisadas no sentido de seleccionar as tecnologias que melhor se adaptam às situações pretendidas. Ainda em relação a esta tabela não foi tido em conta o Bluetooth, assumindo que este é representado pelos *Beacons* através do seu formato mais recente, o BLE.

Tabela 2 - Sistematização de tecnologias (Adaptado de [36] [49] [50] [51])

	Wifi	Beacon	GPS	NFC	Código QR
Localização externa	Sim	Sim	Sim	Sim	Sim
Localização interna	Sim	Sim	Não	Sim	Sim
Precisão	Baixa (3 m)	Elevada Flexível (1 m)	Baixa	Elevada Abaixo de 1 m	Elevada Abaixo de 1 m
Alcance	Depende das perturbações do ambiente e potência do sinal	Até 70/100 m	Global	--	--
Consumo de bateria	Médio	Baixo (dispositivo dura entre 18 meses – 5 anos)	Alto	Baixo	Baixo
Custo	Depende do número de Access Points	5.00 a 35.00 €	Gratuito	De 0.10 a 0.60 €	Baixo
Compatibilidade	Posicionamento baseado no WiFi não é possível com dispositivos iOS	Previsão de 67% dos <i>smartphones</i> habilitados a comunicar com beacons (Android e IOS)	Sem restrições em IOS e Android	Quaisquer produtos da Apple antes do iPhone 6 não suporta	Sem restrições em IOS e Android

Começando pela compatibilidade a nível de *smartphones*, as etiquetas NFC perdem em relação a todas as tecnologias restantes, isto porque praticamente não podem ser lidas por dispositivos da Apple. Nesse sentido, a aplicação móvel para os clientes na plataforma IOS fica em desvantagem. No que diz respeito aos *Beacons*, estes também apresentam restrições a nível de compatibilidade, no entanto, as suas vantagens podem superar este facto. Estes dispositivos são uma tecnologia em ascensão e a perspectiva é que cada vez mais dispositivos consigam interagir com eles, abrangendo assim um maior número de pessoas. Além disso, têm a vantagem de poderem ser usados em ambientes internos e externos e de conseguirem tanto a transmissão de informação a longas distâncias, como a curtas distâncias, não esquecendo a elevada precisão de 1 metro. Os *Beacons* podem assim ser usados para realizar inúmeras atividades para o módulo de jogos que se pretende realizar. Em princípio mais nenhuma das restantes tecnologias tem como entrave a compatibilidade com os *smartphones*.

Outro ponto importante é o consumo de bateria do *smartphone*. Mais uma vez, os *Beacons* suportam esta barreira, através de novo formato BLE. Com baixo consumo de bateria temos também os códigos QR. Não tendo praticamente nenhum custo, e, podendo ser usados em ambientes internos e externos, os códigos QR podem ser também uma boa alternativa para a realização de atividades. Os hóspedes podem ser conduzidos a um local onde, para saber mais informação sobre ele tenham de ler o código.

Nesse sentido, falta apenas avaliar o GPS e o Wifi. Mesmo tendo um elevado consumo de bateria, o GPS é a melhor forma de guiar os hóspedes pela cidade, no caso de ser necessário. Por último, o Wifi não trás nenhuma vantagem de utilização para este módulo em específico. Tem uma baixa precisão e o seu alcance depende muito do local e da quantidade de *access points* que existam. Por outro lado, até poderia ser usado como sistema de localização em ambientes internos, no entanto esta funcionalidade não é possível com dispositivos iOS.

Concluiu-se assim o uso de três tecnologias pelas justificações acima referidas: os *Beacons*, os códigos QR e o GPS. Ou seja, estas três tecnologias serão usadas na construção do módulo.

Em relação aos *Beacons*, a escolha do fornecedor é também um fator importante a ter em atenção. Em termos de negócio, no caso de ser a empresa a construir os jogos, esta deve ter em atenção as diversas possibilidades que existem a nível de fornecedores. Mediante os critérios referidos na seção 2.4.3.2 foi realizado um levantamento de dispositivos *Beacons* de alguns fornecedores por forma a ter uma visão geral do que o mercado oferece. Na Tabela 3 é

possível observar a sistematização da pesquisa efetuada e que tem em consideração aspetos como a bateria, o alcance, as dimensões, as especificações disponíveis e o preço.

Tabela 3 – Lista de fornecedores de *Beacons* e suas características

Fornecedor	Estimote [52]	BlueCats [53]	Gimbal [54]	BlueSense Networks (BlueBar Beacon Waterproof) [55]
Preço	3 por 88 euros	1 por 26 euros	1 por 27 euros	1 por 52 euros
Dimensões		76mm x 27.80mm (H/W)	86 mm x 77 mm x 25 mm (L/W/H)	120mm x 66mm x 42mm (L/W/H)
Bateria	3 até 5 anos (sempre ligada)	3 anos (sempre ligada)	18 meses (sempre ligada)	Até 2 anos (sempre ligada)
Alcance	Até 70 m	De 30 a 60 m	Até 50 m	30 Até 100 m
SDK disponível	x	x	x	x
Especificações	IBeacon	x	x	x
	Eddystone	x	x	
Características suplementares	À prova de água Range configurável Acelerómetro e sensor de temperatura		Intervalo de transmissão e potência configuráveis Sensor de temperatura	À prova de água Sensor de temperatura Resistente aos raios UV

Dos fornecedores referidos, a Estimote [43] é talvez a mais completa. A bateria pode durar até 5 anos no máximo e pode ter um alcance até 70 m (numa situação ideal) configurável. Além de ser à prova de água, o que implica que pode ser usada em ambientes internos e externos, está disponível nas duas especificações possíveis, **Eddystone** e **IBeacon**. Em comparação com outros fornecedores, este dispõe de uma duração de bateria superior, um dos maiores alcances, não

esquecendo que está disponível nas duas especificações possíveis, ao contrário de alguns dos fornecedores que apenas se encontra disponível na especificação **iBeacon**. Em relação ao preço, os *Beacons* da Estimote são cerca de 7.99€ mais caros que os da Gimbal e BlueCats e bastante mais acessíveis que os da Radius Network. Considerando as vantagens acima referidas, a Estimote poderá compensar a diferença. No entanto, é importante ter em atenção que não foram realizados testes reais com cada um dos fornecedores referidos e, por exemplo, a duração da bateria pode não corresponder à realidade mediante a potência de transmissão de sinal que pode ser necessária em determinadas situações.

No que diz respeito às especificações disponíveis, torna-se importante ter em atenção não só as suas funcionalidades, como também o consumo de bateria que cada um dos protocolos consome no *Beacon*. Foram realizados testes com os *Beacons* da Estimote e, apesar do formato Eddystone ser mais completo (possibilidade de enviar URL's) a duração média da bateria indicada pela Estimote é de 36 meses para o formato **iBeacon** e 20 meses para o formato **Eddystone**. Nesse sentido, a escolha recai sobre o formato que torne o uso de *Beacons* da Estimote o mais económico possível, ou seja, o formato **iBeacon**.

3.3 Sumário

Neste capítulo, foi realizada uma avaliação da pesquisa do Estado da Arte realizada na seção 2. Ou seja, foram analisados os trabalhos relacionados na área do património, cultura e história, bem como as categorias de jogos mais adequadas a este tipo de jogos e ainda as principais características que estes devem possuir para atingirem o sucesso. De seguida foi realizada a mesma avaliação para as tecnologias móveis pesquisadas, concluindo que os *Beacons*, os códigos QR e o GPS poderão ser uma mais-valia para a resolução do problema proposto.

4 Análise e Design da solução

Neste capítulo, será realizada uma pequena introdução ao conceito do jogo, seguido dos requisitos funcionais e não funcionais definidos para este módulo. Além disso, será ainda apresentada a arquitetura na qual será desenvolvido e também o modelo de dados usado para persistir a informação necessária.

4.1 Conceito do jogo

Pretende-se a elaboração de um jogo aberto, ou seja, configurável em relação às ações que se podem realizar, em relação à escolha dos locais a conhecer que terão determinadas ações e ainda em relação à tabela de classificação e atribuição de prémios (através de campanhas). Nesse sentido, cada cliente ou a empresa pode configurar de forma bastante flexível, os jogos que pretende fazer, as classificações e, caso desejem, a atribuição de prémios.

As ações é que vão permitir criar a dinâmica do jogo. Cada ação terá sempre associado dois pontos, que representam dois determinados locais, numa determinada cidade. Um ponto será usado para ativar a ação e o outro ponto será efetivamente o local que se pretende dar a conhecer. Cada ação terá ainda que escolher a forma como pretende ser ativada: GPS, Beacon ou Código QR. Por exemplo, a ação A será ativada no Ponto A através de GPS, com o objetivo de descobrir o Ponto B, também com ajuda do GPS. Ou seja, a escolha da tecnologia (GPS, Código QR e Beacon) numa dada ação é usada tanto para a sua ativação, como para a sua descoberta. Importante ainda referir que cada ação será de um tipo. Os tipos que abaixo se seguem foram decididos tendo por base o Estado da Arte analisado para este problema:

- **Pergunta – Resposta:** Chegado a um determinado local, o hóspede é confrontado com uma determinada questão de escolha múltipla à qual tem de responder;
- **Descoberta:** O objetivo é dar instruções ao hóspede para que este consiga descobrir um determinado local;
- **Antes e depois:** É mostrado ao hóspede um conjunto de fotos antigas de um local e pistas para lá chegar. O hóspede tem assim de descobrir esse local, pelas semelhanças entre o antes e o agora, de um dado local;
- **Quente e Frio:** Permite ao hóspede descobrir um determinado local. Neste caso, o hóspede será informado à medida que caminha se se encontra perto ou longe do local a encontrar.

Todas as ações poderão ser acompanhadas de fotos ilustrativas, bem como de um ficheiro áudio, caso seja pertinente o seu uso no contexto de uma dada ação. Por forma a tornar o jogo mais aliciante todas as ações terão um tempo máximo para serem realizadas.

No que diz respeito às campanhas, estas poderão ser configuradas para estarem ativas por um dado período de tempo, e serão ganhas pelos hóspedes caso atinjam um determinado nível do jogo e/ou realizem um determinado número de ações, com sucesso.

4.2 Requisitos

O levantamento de requisitos abrange o processo de recolha de informação para a definição das funcionalidades que o módulo deve apresentar. Nesse sentido, apresentam-se de seguida os requisitos funcionais e não funcionais definidos.

4.2.1 Requisitos funcionais

A definição dos requisitos funcionais que se seguem foi estabelecida em conjunto com a empresa e com a sua perceção das necessidades para o desenvolvimento de um módulo que permita resolver o problema proposto.

- **Efetuar registo:** A aplicação móvel deve permitir aos utilizadores o registo dos hóspedes, através da indicação de alguns dados relativos sobre a pessoa. O sistema deve ainda verificar que, caso o utilizador já esteja registado naquele hotel, deve

receber toda a informação sobre o estado do seu “jogo” até ao momento. Importante salientar que este requisito já se encontra implementado no projeto atual;

- **Interação com a cidade:** A aplicação móvel deve ajudar a descobrir a cidade de uma forma lúdica e com apoio de pequenas atividades realizadas em determinados pontos de interesse da cidade;
- **Gestão dos jogos e atividades:** O Backoffice deve permitir a gestão dos jogos atribuídos a cada cliente. Isso implica que também é necessária a gestão das ações, dos níveis do jogo, das classificações e das campanhas (prémios);
- **Visualização de classificações:** O utilizador deve ter acesso, na aplicação móvel, à sua classificação e nível enquanto jogador, bem como à classificação geral dos utilizadores. Estes poderão assim competir por um lugar melhor dentro da cidade onde se encontram/encontraram hospedados ou em termos gerais de classificação;
- **Comunicação com os hóspedes:** A aplicação móvel deve permitir a comunicação com o hóspede, promovendo os serviços, os eventos e as atividades do estabelecimento hoteleiro, sendo possível articular esta promoção com os jogos/atividades realizados;
- **Funcionamento em modo *offline*:** Este módulo deve funcionar em modo *offline*, tal como a restante aplicação móvel se encontra atualmente. No entanto, o registo de utilizadores e mesmo a atribuição de prémios (através das campanhas) assume-se que funcionará apenas em modo *online*.

4.2.2 Requisitos não funcionais

Os requisitos não funcionais relacionam-se em geral com um conjunto de padrões para um sistema que se pretende, de qualidade. Apresentam-se de seguida os requisitos não funcionais definidos para este módulo em particular:

- **Desempenho:** A realização dos jogos deve ser processada de forma eficiente, pois os hóspedes pretendem utilizar uma ferramenta que responda rapidamente às suas necessidades e que não seja moroso na resposta. O mesmo acontece com a configuração dos jogos no Backoffice por parte dos clientes-alvo deste módulo;
- **Usabilidade:** O módulo de jogos implementado na aplicação móvel tem de ser de fácil perceção de utilização para os hóspedes, além de ter uma interface amigável com o utilizador.

- **Segurança:** O módulo deve seguir a mesma lógica de permissões de acesso já implementadas, não permitindo aos Clientes (hóteis, *hostels* e apartamentos) o livre acesso para gestão de toda a informação, tal como acontece ao Administrador.

4.3 Casos de uso

Um caso de uso é uma história de como o sistema se comporta para atingir um determinado objetivo, promovendo a compreensão e descrição de requisitos [56]. Na Figura 7, é possível observar o diagrama de casos de uso desenvolvido para o problema em questão.

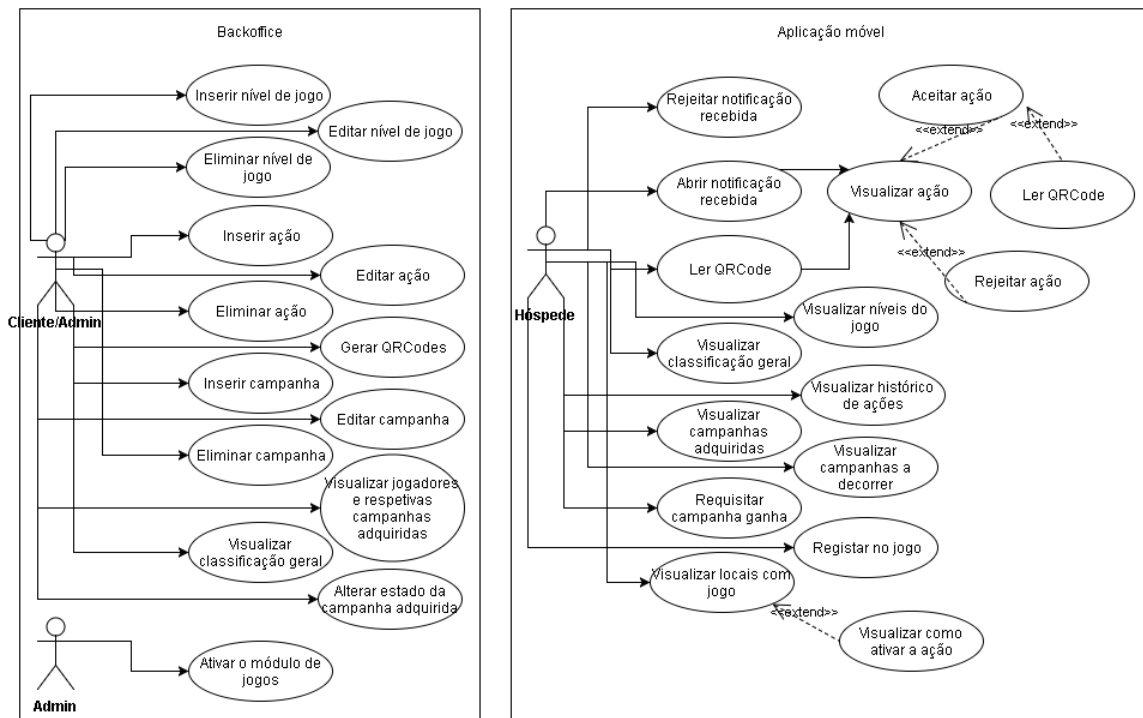


Figura 7 – Diagrama de casos de uso

É possível observar dois diagramas de casos de uso. O da esquerda refere-se aos requisitos do Backoffice, o da direita à aplicação móvel. A nível de atores do sistema, considera-se no Backoffice a existência de dois: o Cliente que é uma pessoa responsável no hotel, *hostel* ou apartamento por aceder ao Backoffice e configurar o jogo e visualizar o seu estado. Optou-se por colocar a par do Cliente o Administrador do Backoffice da empresa, isto porque este ator tem exatamente os mesmos casos de uso, com a diferença, de que pode fazer para qualquer Cliente, ao contrário do próprio Cliente que apenas o pode fazer para o seu estabelecimento hoteleiro. O ator da aplicação móvel será o hóspede de cada hotel, *hostel* ou apartamento.

4.3.1 Backoffice

Nas Tabela 4, Tabela 5 e Tabela 6 é possível observar as descrições de alto nível dos casos de uso do Backoffice, agrupados por conjuntos de funcionalidades.

Tabela 4 – Descrição de alto nível dos casos de uso do Backoffice para gestão das campanhas

Inserir Campanha	<p>O administrador inicia a especificação de uma nova campanha. O sistema solicita os dados do nome da campanha, o número de hóspedes que a podem adquirir, a data de início e a entidade (cliente) para a qual deverá ser criada. Outros dados, como a data de fim da campanha, a descrição, uma imagen representativa ou ainda o número de ações que devem ser concluídas com sucesso para adquirir a campanha podem ser preenchidos pelo utilizador. O sistema grava a campanha e atualiza a lista de campanhas.</p> <p>No caso do cliente o procedimento é igual, com excepção de que o sistema não requisita a indicação da entidade (cliente). Neste caso, assume o cliente que se encontra autenticado.</p>
Editar Campanha	<p>O administrador inicia a especificação para editar uma determinada campanha. O sistema disponibiliza toda a informação associada a essa campanha. O utilizador atualiza os dados que desejar (os dados necessários são referidos no caso de uso <i>Inserir Campanha</i>). O sistema grava as alterações da campanha e atualiza a lista de campanhas.</p> <p>No caso do cliente o procedimento é igual, com excepção de que o sistema não requisita a indicação da entidade (cliente). Neste caso, assume o cliente que se encontra autenticado.</p>
Eliminar Campanha	<p>O administrador inicia a especificação para eliminar uma dada campanha. O sistema questiona o utilizador se pretende efetuar esta operação. O utilizador confirma a operação. O sistema elimina a campanha e atualiza a lista de campanhas.</p> <p>No caso do cliente o procedimento é igual.</p>
Visualizar Jogadores e respectivas Campanhas adquiridas	<p>O administrador inicia a especificação para visualizar os jogadores e respetivas campanhas adquiridas. O sistema solicita a indicação da entidade (cliente) para a qual pretende visualizar a informação. O utilizador indica essa entidade. O sistema atualiza a lista de jogadores e campanhas, em relação à entidade escolhida.</p> <p>No caso do cliente o procedimento é igual, com excepção de que o sistema não requisita a escolha da entidade (cliente), atualizando desde logo a lista de jogadores e campanhas para o cliente em questão.</p>
Alterar estado da Campanha adquirida	<p>O administrador passa pelo mesmo processo referido no caso de uso <i>Visualizar Jogadores e respectivas Campanhas</i> adquiridas. Na lista de jogadores e campanhas adquiridas, o administrador inicia a especificação para alterar o estado da campanha. Para um dado jogador da lista, altera o estado da campanha que este adquiriu. O sistema atualiza o estado da campanha para o jogador em questão e atualiza a lista de jogadores e campanhas adquiridas.</p> <p>No caso do cliente o procedimento é igual.</p>

Nas tabelas é possível observar o nome dos casos de uso do lado esquerdo e as respectivas descrições do lado direito, separadas pelos atores participantes. Na Tabela 4, é possível observar a descrição para os casos de uso para gestão de campanhas no Backoffice. O conceito de campanha é usado para atribuir prêmios aos utilizadores da aplicação móvel, funcionando também como uma fonte de divulgação de serviços, eventos e atividades que o estabelecimento hoteleiro tenha. Por exemplo, com este conceito é possível ter uma campanha que vai atribuir a todos os jogadores que atingirem o nível três, uma hora de utilização no SPA. O Cliente e o Administrador têm acesso a todas as funcionalidades descritas, no entanto, com algumas restrições em relação ao Cliente.

Tabela 5 - Descrição de alto nível dos casos de uso do Backoffice para gestão dos níveis do jogo

Inserir Nível de Jogo	<p>O administrador inicia a especificação de um novo nível de jogo. O sistema solicita os dados do nome do nível do jogo, o número de pontos máximo que pode atingir esse nível e a entidade (cliente) para a qual deverá ser criado. Outros dados, como a indicação de um ícone representativo do nível do jogo, pode igualmente ser preenchido pelo utilizador. O sistema grava o nível de jogo e atualiza a lista de níveis de jogo.</p> <p>No caso do cliente o procedimento é igual, com excepção de que o sistema não requisita a indicação da entidade (cliente). Neste caso, assume o cliente que se encontra autenticado.</p>
Editar Nível de Jogo	<p>O administrador inicia a especificação para editar um determinado nível de jogo. O sistema disponibiliza toda a informação associada a esse nível de jogo. O utilizador atualiza os dados que desejar (os dados necessários são referidos no caso de uso <i>Inserir Nível de Jogo</i>). O sistema grava as alterações do nível de jogo e atualiza a lista de níveis de jogo.</p> <p>No caso do cliente o procedimento é igual, com excepção de que o sistema não requisita a indicação da entidade (cliente). Neste caso, assume o cliente que se encontra autenticado.</p>
Eliminar Nível de Jogo	<p>O administrador inicia a especificação para eliminar um dado nível de jogo. O sistema questiona o utilizador se pretende efetuar esta operação. O utilizador confirma a operação. O sistema elimina o nível de jogo e atualiza a lista de níveis de jogo.</p> <p>No caso do cliente o procedimento é igual.</p>

Na Tabela 5, é possível observar detalhes sobre os casos de uso relativos à gestão dos níveis do jogo. O conceito de nível do jogo permite associar os jogadores a um determinado nível, colocando-o em competição com os restantes jogadores à medida que vão jogando, isto é, concluindo ações com sucesso. O Cliente e o Administrador têm acesso a todas as funcionalidades descritas, no entanto, com algumas restrições em relação ao Cliente.

Na Tabela 6, são apresentados os casos de uso referentes à gestão das ações no Backoffice. O conceito de ação permite criar a dinâmica do jogo na aplicação móvel, representando a informação que permite dar a conhecer um dado local.

Tabela 6 - Descrição de alto nível dos casos de uso do Backoffice para gestão das ações

Inserir Ação	<p>O administrador inicia a especificação de uma nova ação. O sistema solicita os dados do nome da ação, o tipo da ação, o número de pontos ganhos em caso de sucesso, o local de ativação e descoberta, o tempo máximo para a concluir, a forma como a ação é ativada/descoberta (GPS, Beacon, Código QR) e a entidade (cliente) para a qual deverá ser criada. Outros dados, como a descrição, uma mensagem de sucesso e/ou insucesso, as possíveis respostas no caso de uma ação do tipo “Pergunta-Resposta” ou ainda imagens e um ficheiro áudio podem ser preenchidos pelo utilizador. O sistema grava a ação e atualiza a lista de ações.</p> <p>No caso do cliente o procedimento é igual, com exceção de que o sistema não requisita a indicação da entidade (cliente). Neste caso, assume o cliente que se encontra autenticado.</p>
Editar Ação	<p>O administrador inicia a especificação para editar uma determinada ação. O sistema disponibiliza toda a informação associada a essa ação. O utilizador atualiza os dados que desejar (os dados necessários são referidos no caso de uso <i>Inserir Ação</i>). O sistema grava as alterações da ação e atualiza a lista de ações.</p> <p>No caso do cliente o procedimento é igual, com exceção de que o sistema não requisita a indicação da entidade (cliente). Neste caso, assume o cliente que se encontra autenticado.</p>
Eliminar Ação	<p>O administrador inicia a especificação para eliminar uma dada ação. O sistema questiona o utilizador se pretende efetuar esta operação. O utilizador confirma a operação. O sistema elimina a ação e atualiza a lista de ações.</p> <p>No caso do cliente o procedimento é igual.</p>
Gerar Códigos QR	<p>O sistema disponibiliza todas as entidades existentes (clientes) para gerar os códigos QR. O administrador indica a entidade (cliente) para a qual os pretender gerar e inicia a especificação para gerar os códigos QR. O sistema apresenta os códigos QR gerados com a respetiva indicação do local associado. O utilizador imprime os códigos, para os poder colocar nos respetivos locais.</p> <p>No caso do cliente o procedimento é igual, com exceção de que o sistema não disponibiliza o conjunto de entidades (clientes), assumindo o que se encontra autenticado no sistema.</p>
Visualizar Classificação Geral	<p>O administrador inicia a especificação para visualizar a classificação geral dos utilizadores. O sistema solicita a indicação da entidade (cliente) para a qual pretende visualizar a classificação. O utilizador indica essa entidade. O sistema atualiza a lista de classificações, em relação à entidade escolhida.</p> <p>No caso do cliente o procedimento é igual, com exceção de que o sistema não requisita a escolha da entidade (cliente), atualizando desde logo a lista de classificações para o cliente em questão.</p>

Com este conceito é possível obter, por exemplo, uma ação que vai ser despoletada por GPS nos Clérigos, com o objetivo de descobrir a Avenida dos Aliados em 5 minutos, adquirindo 10 pontos caso seja concluída com sucesso. O Cliente e o Administrador têm acesso a todas as funcionalidades descritas, no entanto, com algumas restrições em relação ao Cliente.

Por fim, o caso de uso *Ativar o Módulo de Jogos* não se enquadra em nenhum dos três grupos de informação (ações, níveis de jogo e campanhas) já referidos. Esta funcionalidade é apenas da responsabilidade do Administrador e permite ativar o módulo de jogos para um determinado Cliente. Sendo ativado, os Clientes terão acesso aos menus de configuração de níveis de jogo, de ações e de campanhas, assim como terão acesso às configurações efetuadas na aplicação móvel, ou seja, poderão visualizar o jogo na aplicação.

4.3.2 Aplicação móvel

Nas Tabela 7 e Tabela 8 é possível observar as descrições de alto nível dos casos de uso da aplicação móvel.

Tabela 7 – Descrição dos casos de uso da aplicação móvel para apresentação de informação

Visualizar Níveis de jogo		Depois de registado no jogo, o hóspede inicia a especificação para visualizar os níveis de jogo. O sistema apresenta a lista com os níveis de jogo disponíveis.
Visualizar Classificação Geral		Depois de registado no jogo, o hóspede inicia a especificação para visualizar a classificação geral do jogo. O sistema apresenta a lista com classificação geral de todos os utilizadores daquele hotel, <i>hostel</i> ou apartamento, sendo destacado o seu lugar na classificação.
Visualizar Histórico de Ações		Depois de registado no jogo, o hóspede inicia a especificação para visualizar o histórico de ações. O sistema apresenta a lista com todas as ações já realizadas por ele.
Visualizar Campanhas adquiridas		Depois de registado no jogo, o hóspede inicia a especificação para visualizar as campanhas adquiridas. O sistema apresenta a lista com todas as campanhas (prémios) que conseguiu adquirir.
Visualizar Campanhas a decorrer		Depois de registado no jogo, o hóspede inicia a especificação para visualizar as campanhas a decorrer. O sistema apresenta a lista com todas as campanhas que ainda pode tentar adquirir.
Visualizar Locais com Jogos		Depois de registado no jogo, o hóspede inicia a especificação para visualizar as campanhas a decorrer. O sistema apresenta a lista de todos os locais disponíveis na aplicação, assinalando aqueles que tem ações que podem ser realizadas.
Visualizar como ativar uma Ação		Depois de registado no jogo, o hóspede seleciona um local com jogo disponível. O sistema mostra os detalhes desse local (nome, descrição, fotografias). O hóspede inicia a especificação para visualizar como ativar uma dada ação. O sistema apresenta a informação que indica como despoletar a ação para aquele local em específico.

Nas tabelas é possível observar o nome dos casos de uso do lado esquerdo e as respectivas descrições do lado direito. Neste caso, o único ator é o Hóspede do hotel. Na Tabela 7, são apresentados os casos de uso referentes à apresentação de informação ao Hóspede na aplicação móvel, tais como os resultados do jogo e campanhas, os níveis do jogo, os locais com jogos ativos, entre outros.

Na Tabela 8, são apresentados os casos de uso referentes à dinâmica do jogo, que vão desde o registo no jogo, a leitura de códigos QR e as notificações, até à aceitação e/ou rejeição de uma ação.

Tabela 8 - Descrição dos casos de uso da aplicação móvel para a dinâmica do jogo

Registar no jogo	O hóspede inicia a especificação para registar o jogo. O sistema solicita os dados do nome e número de telefone. O sistema grava o novo utilizador e redireciona-o para a lista de locais disponíveis.
Abrir notificação	Depois de registado no jogo, o sistema deteta uma ação despoletada por GPS ou Beacon e lança uma notificação. O hóspede clica sobre a notificação. O sistema mostra os detalhes da ação ativada (nome, descrição, fotos, áudio ...).
Rejeitar notificação	Depois de registado no jogo, o sistema deteta uma ação por GPS ou Beacon e lança uma notificação. O hóspede rejeita a notificação. O sistema marca esta ação como rejeitada, voltando a mostra-la 24 h depois, caso as suas condições de ativação estejam reunidas.
Ler Código QR	Depois de registado no jogo, o hóspede inicia a especificação para ler um código QR. O sistema verifica se o código que está a ler é de uma nova ação e, caso seja verdade, mostra os detalhes da ação ativada. Se o sistema verificar que o código QR lido corresponde ao término de ação, mostra então a mensagem de sucesso, indicativa de que a ação foi concluída com sucesso.
Aceitar ação	Depois de registado no jogo e do sistema apresentar os detalhes de uma determinada ação que foi despoletada, o hóspede inicia a especificação para aceitar uma ação (aceitar o desafio proposto). O sistema começa a decrescer o tempo que essa ação tem para ser concluída. Para a ação ser finalizada, ou o sistema deteta via GPS ou Beacon que o hóspede está no local pretendido, ou o hóspede lê o código QR indicativo de que encontrou o local.
Rejeitar ação	Depois de registado no jogo e do sistema apresentar os detalhes de uma determinada ação que foi despoletada, o hóspede inicia a especificação para rejeitar uma ação (rejeitar o desafio proposto). O sistema verifica que, caso esta ação tenha surgido de uma notificação recebida, esta voltará a estar disponível 24 horas após a rejeição, caso volte a passar no mesmo local.

4.1 Modelo de domínio

O modelo de domínio é uma representação visual das classes conceituais ou de objetos reais do domínio de interesse, permitindo a identificação dos principais conceitos, dos seus atributos e das associações entre eles [56]. Foi desenhado o modelo de domínio (Figura 8) de modo a representar a relação entre os novos conceitos impostos por este módulo e os restantes já usados no Backoffice, no conceito do produto da empresa.

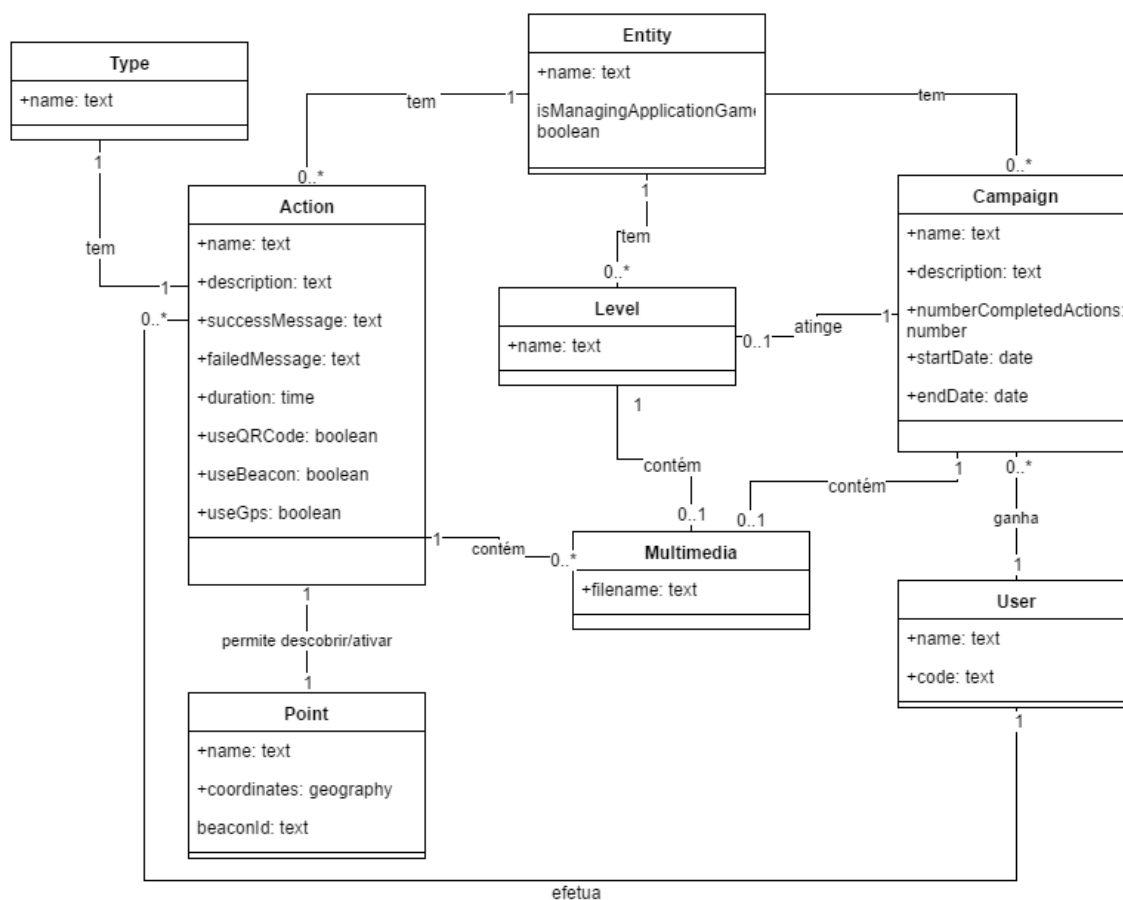


Figura 8 – Modelo de domínio

O conceito **Entity**, **Point**, **Multimedia** e **User** já faziam parte do sistema. Uma **Entity** representa o hotel, *hostel* ou apartamento, ou seja, são os clientes. Um **Point** representa um determinado local, com coordenadas geográficas e informação detalhada sobre o mesmo. A **Multimedia** diz respeito a todos os ficheiros multimédias que se encontram no sistema, sendo distinguidos por um tipo (exemplo: Ficheiro, Imagem, Audio). O conceito **User** já está criado e é usado atualmente para registo de utilizadores no serviço de **Concierge**, funcionalidade existente na aplicação. Este conceito será assim utilizado para o registo de utilizadores no jogo.

No que diz respeito aos novos conceitos, a **Action** é o principal. Representa a atividade de realizar algo (por exemplo: encontrar um determinado local), conseguindo pontos caso a termine com sucesso. As **Action** são configuradas por **Entity** e tem como informação associada não só número de pontos que vale, como também o **Point** onde será ativada e o **Point** que permitirá descobrir. Por outro lado, o **Level** representa os níveis que o jogo pode ter, configurados por **Entity**. A **Campaign** representa os prémios que os clientes poderão dar aos seus **Users** caso atinjam um determinado nível de jogo, ou realizem um determinado número de atividades com sucesso, sendo também configuradas por **Entity**. Por último, o **User** representa os utilizadores do jogo, podendo realizar ações e adquirir campanhas.

4.2 Arquitetura do sistema

Sendo objetivo deste módulo a sua integração na solução existente, torna-se importante referir a atual arquitetura do sistema em geral, de modo a tentar perceber onde este se irá integrar. Na Figura 9 é possível observar a *deployment view* do atual sistema.

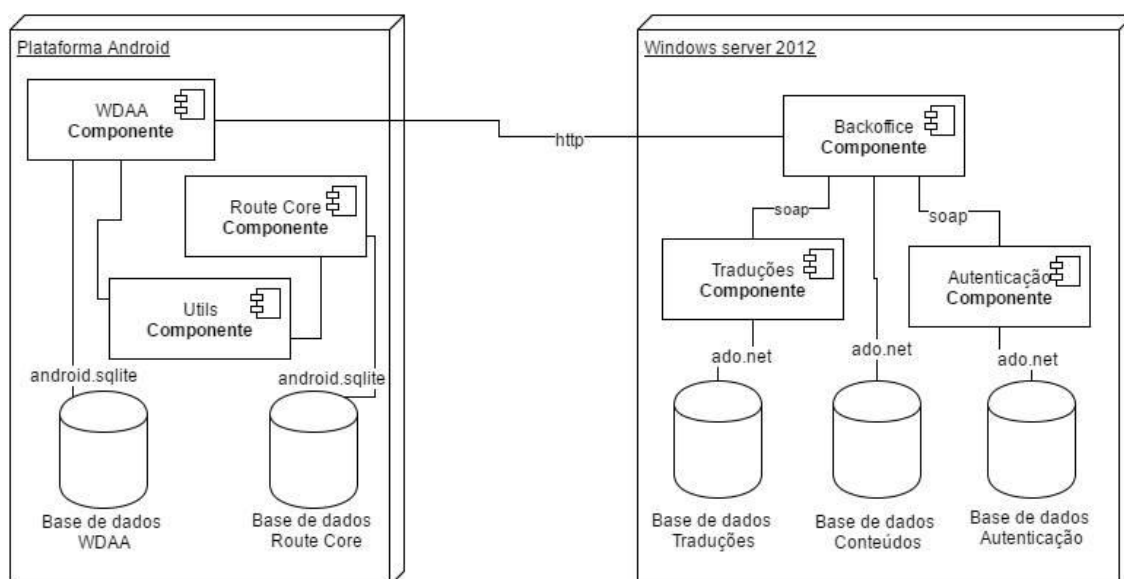


Figura 9 – Deployment view do sistema

O bloco da direita é composto por três componentes: Backoffice, Traduções e Autenticação. O Backoffice permite visualizar e gerir toda a informação sobre as aplicações móveis dos clientes e toda a informação que nela persiste, desde a informação do estabelecimento hoteleiro, eventos, pontos, roteiros, as respetivas traduções. Este componente comunica com os restantes (Autenticação e Traduções) via SOAP. Cada um destes módulos comunica com a sua

respetiva base de dados, via ADO.NET. Neste caso três bases de dados distintas para cada um dos módulos. As bases de dados onde são persistidos os dados deste sistema encontram-se em SQL Server 2012. Importante ainda salientar que o Backoffice encontra-se estruturado numa arquitetura em três camadas, usando o padrão Table Module, Table Data Gateway [57]. Para este módulo em específico serão feitos desenvolvimentos apenas no componente Backoffice e respetiva base de dados.

O bloco da esquerda representa as aplicações móveis desenvolvidas para a plataforma Android e que comunicam com o componente Backoffice, para obter toda a informação necessária para ser apresentada na aplicação. A comunicação é feita via *HTTP*, através do uso de um Generic Web Handler, classe que contém os serviços necessários à aplicação móvel. Em relação à plataforma Android, esta também se encontra dividida em três componentes. O componente RouteCore é o mais antigo do projeto, sendo usado para gerir informação relacionada com rotas e pontos de interesse. De notar que este componente comunica com a sua respetiva base. O componente WDAA é o mais recente, tendo sido criado para este projeto em específico, que é gerida toda a restante informação da aplicação e é onde o módulo de jogos será integrado. Tal como o RouteCore, o componente WDAA comunica com a sua base de dados. Em ambos os componentes, a informação é persistida numa base de dados SQLite [58], através do ORMLite [15]. Por último, o componente Utils é utilizado para aceder a informação mais geral como as traduções, a gestão dos pedidos feitos ao Backoffice, entre outros.

4.3 Sumário

Neste capítulo foi realizada uma breve análise do conceito de jogo e dos requisitos que o sistema deverá ter. Perante isso foi desenhado o diagrama de casos de uso e foi feito o levantamento de conceitos importantes para este problema no diagrama de domínio. Foi ainda escrita a atual arquitetura geral do sistema, onde o novo módulo será enquadrado. No que diz respeito ao módulo que se pretende implementar, este será incluído no módulo de gestão de conteúdos e, conseqüentemente, na camada de acesso a dados. Nesse sentido e, tal como já foi referido nas restrições para o desenvolvimento, o módulo será desenvolvido na *framework* ASP.NET, na linguagem C# e com acesso à base de dados SQL Server 2012. Em termos de aplicação móvel, este módulo será integrado na plataforma Android GuestU. De salientar que

a base de dados usada é o SQL Lite, através do ORM Lite que é usado para persistir a informação necessária na base de dados.

5 Implementação

Neste capítulo são apresentados os principais desenvolvimentos realizados a nível do Backoffice, dos serviços para a aplicação móvel e da própria aplicação móvel. Inicialmente são apresentados detalhes do Backoffice, dando destaque às principais funcionalidades implementadas e ao modelo de dados e de classes usado para tal. De seguida, apresentam-se os serviços que são suporte à aplicação e, por fim, são evidenciadas as principais funcionalidades desenvolvidas na aplicação móvel. Este último ponto inicia-se com a apresentação de algumas considerações iniciais sobre a forma como é desenvolvida.

5.1 Backoffice

Os desenvolvimentos feitos a nível do Backoffice serão descritos tendo em conta a arquitetura na qual ele se encontra desenvolvido, isto é, numa arquitetura de três camadas. São apresentadas ainda algumas referências ao Módulo de Traduções, com o qual é necessário interagir. Nesse sentido, são feitas algumas alusões ao diagrama de classes implementado tendo em conta a arquitetura em questão e, de seguida, é apresentado o modelo de dados desenvolvido para o problema proposto. Por forma a mostrar as diversas funcionalidades implementadas, é ainda apresentada uma breve explicação de cada uma delas, dando especial atenção à gestão das ações. Por último, são apresentados os diversos serviços desenvolvidos que dão suporte à aplicação móvel.

5.1.1 Enquadramento

Por forma a tornar a leitura desta seção mais perceptível, torna-se importante fazer um enquadramento entre a forma como esta é apresentada e o fluxo de construção/configuração do módulo de jogos na aplicação GuestU.

Para o módulo de jogos ficar visível na aplicação de uma determinada unidade hoteleira é necessário a sua configuração prévia no Backoffice. Inicialmente, o Administrador do Backoffice tem de ativar o módulo para o Cliente em questão. Na Figura 10, é possível observar um fluxograma simplificado para a configuração do módulo a nível do Backoffice, por parte de um Cliente.

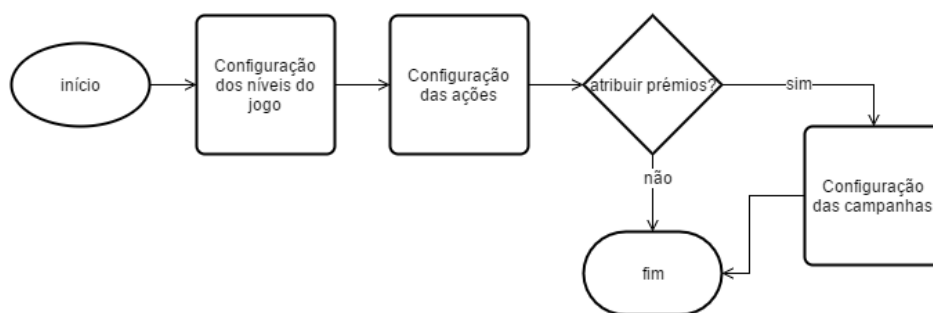


Figura 10 – Fluxograma para a configuração do módulo de jogos no Backoffice

Assumindo que o estabelecimento hoteleiro já tem o módulo de jogos ativo, este deverá começar pela configuração dos níveis de jogo. Seguidamente, deverá efetuar a configuração das ações necessárias, assumindo que já tem todos locais (pontos) previamente configurados no Backoffice. Por último e, caso pretenda que o jogo ofereça prémios aos seus jogadores, deverá configurar as campanhas que pretender. Nesta seção pode ser observado em maior detalhe a implementação deste processo. Esta informação é acedida pela aplicação móvel através de serviços, também da responsabilidade do Backoffice. Importante salientar que, além dos passos de configuração do módulo já referidos, o Cliente tem acesso a outro tipo de informação, nomeadamente o *ranking* de jogadores e o estado das suas campanhas. Informação igualmente detalhada nesta seção.

5.1.2 Diagrama de classes

Na Figura 11, é possível observar o diagrama de classes simplificado para o módulo a desenvolver, organizado pelas três camadas da arquitetura.

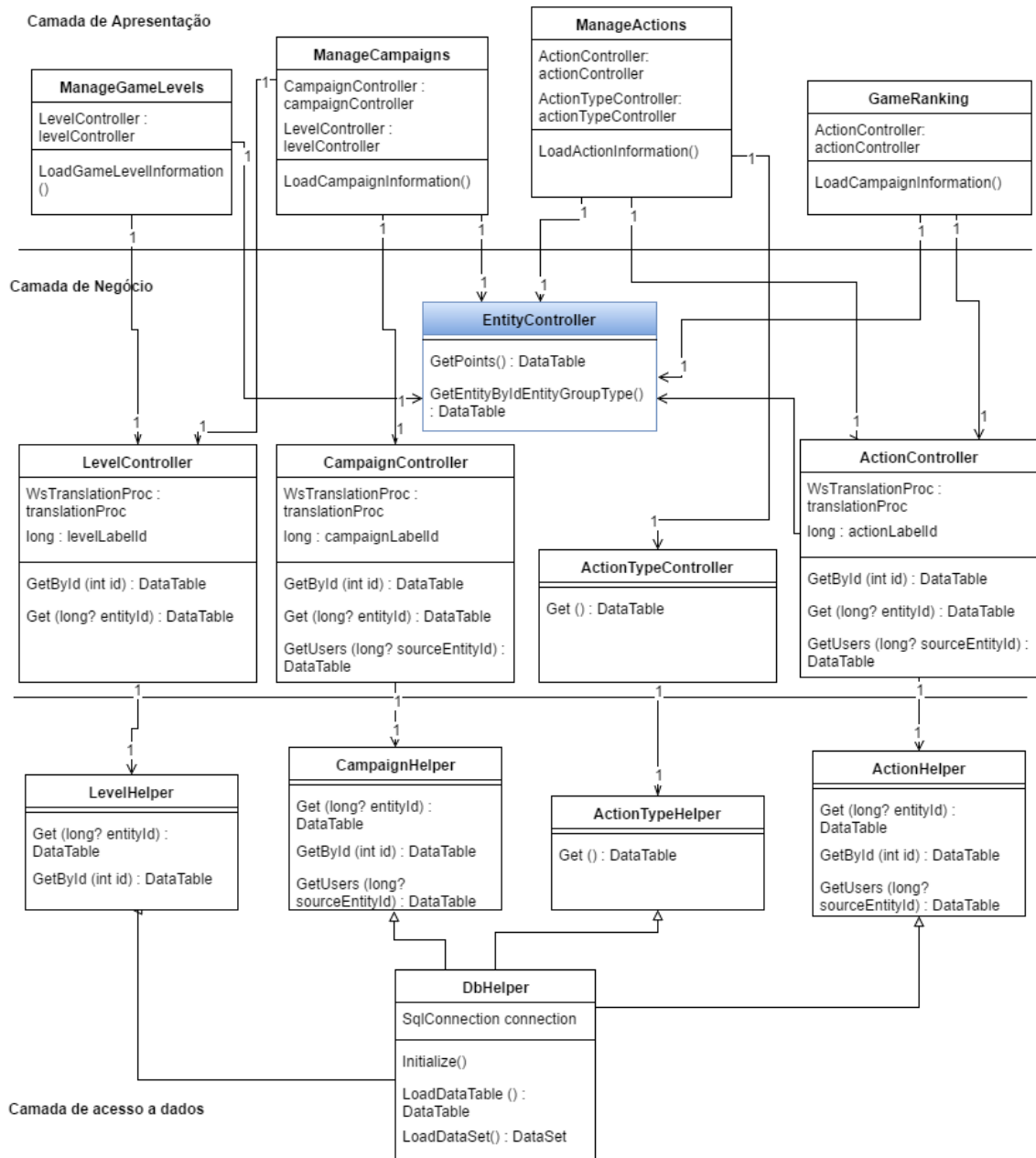


Figura 11 – Diagrama de classes simplificado do Backoffice

As classes da camada de acesso a dados são identificadas pelo sufixo *Helper* e as classes da camada de negócio são identificadas pelo sufixo *Controller*. A classe *EntityController* já fazia parte do sistema, pelo que se encontra assinalada e azul e não está representada a respetiva

classe *Helper*. Outras classes necessárias ao desenvolvimento deste módulo também não se encontram representadas, tal como o [ContentController](#) (responsável por gerir os ficheiros multimédia) e o [PointController](#) (responsável por gerir os locais). Tal como já foi referido no capítulo de Análise e Design (4.2) é usado o padrão Table Module (classes Controller) e Table Data Gateway (classes Helper), como na restante aplicação.

Na camada de acesso a dados a informação é persistida em *DataTables/DataSets*, não sendo usados objetos. As classes *Helper* comunicam assim com a base de dados SQL Server através de procedimentos, onde são efetuadas as operações *Create*, *Read*, *Update* e *Delete* (CRUD) necessárias. Todas as classes *Helper* estendem ainda a classe [DBHelper](#), responsável por ligar-se à base de dados e converter a informação que vem dos procedimentos em *DataTables* ou *DataSets*, caso necessário.

A camada de negócio efetua a ponte entre a camada de apresentação e a camada de acesso a dados. É também nesta camada que são disponibilizadas para tradução, nas mais diversas línguas, os conteúdos necessários. O módulo de traduções é representado pelo objeto [WSTRANSLATION_PROC](#). Ainda nesta camada, torna-se importante referir que as operações mais complexas são realizadas dentro de Transações. Por exemplo, quando se faz o *Update* a uma ação (objeto [Action](#)) torna-se necessário gerir os ficheiros multimédia que esta tem, usando para isso o objeto [ContentController](#), ou ainda verificar se foram disponibilizadas novas línguas de traduções, verificando assim se todos os conteúdos dessa ação se encontram nela. Todo este processo, que envolve um *Update*, é assim realizado dentro de uma Transação.

Na camada de apresentação foram criadas quatro páginas tendo em conta os requisitos necessários: *ManageGameLevels*, *ManageActions*, *ManageCampaigns* e *GameRanking*. Na seção 5.1.4 são explicadas, em maior detalhe, cada uma destas páginas.

5.1.3 Modelo de dados

A Figura 12 apresenta o modelo de dados do módulo que se pretende integrar, tendo em atenção o atual modelo seguido na gestão de Conteúdos. Todas as tabelas têm ainda um conjunto de atributos base que não foram aqui especificados (*TimeCreate*, *TimeChange*, *UserCreate*, *UserChange* e *IsActive*) e que ajudam a entender em que data e quem foi a pessoa que os criou. Já o atributo *IsActive* é usado nas operações de remoção de informação, sendo

colocado a 0 (falso), de modo a que seja acessível uma eventual recuperação dos dados, caso seja necessário.

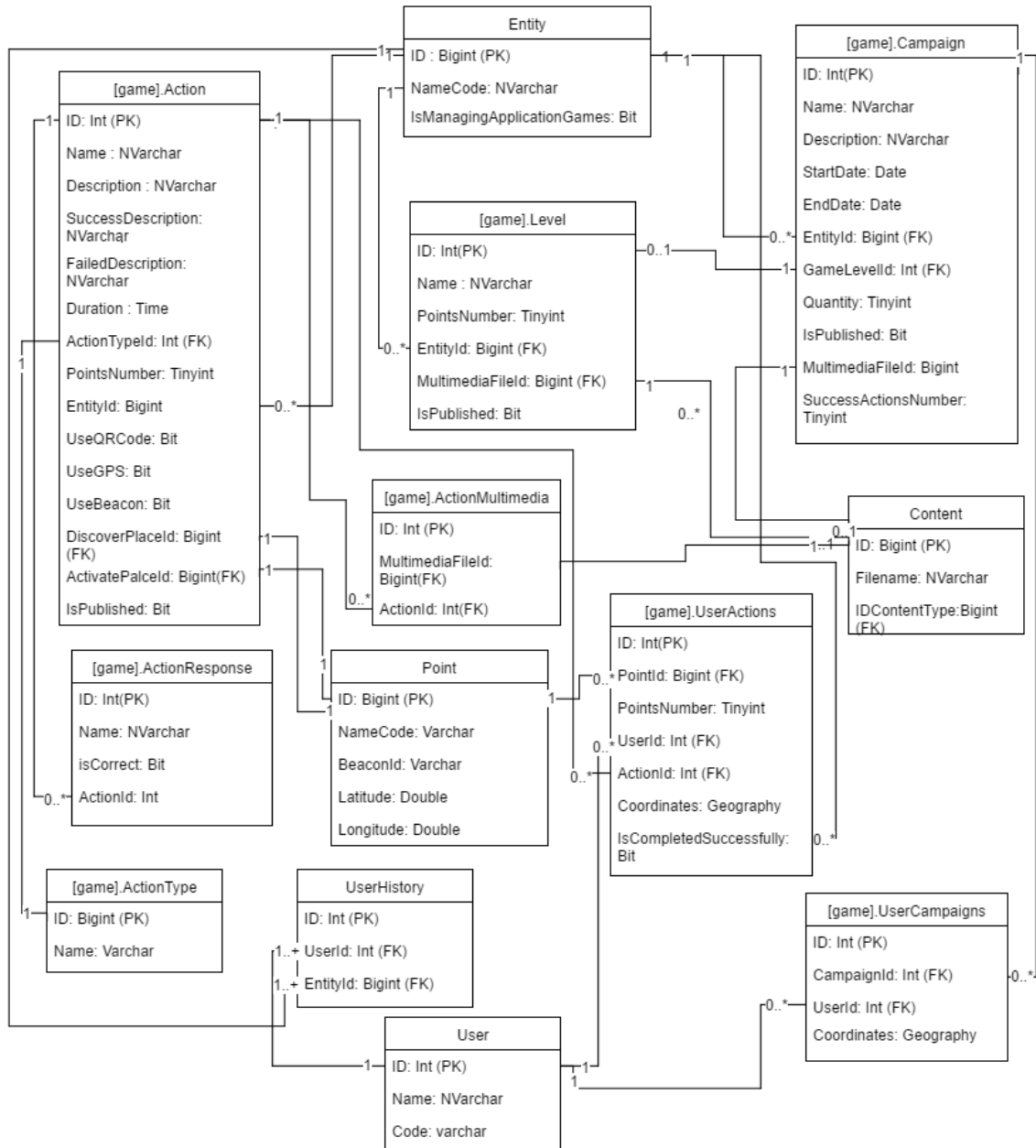


Figura 12 – Modelo de dados do Backoffice

Foi criado o *schema game* que agrega todas as tabelas que foram criadas para este módulo em específico. A tabela *User* e *UserHistory* já utilizadas para registo de utilizadores no serviço de Concierge (*chat*), serão igualmente utilizadas para o registo de utilizadores do jogo. Ainda em relação à tabela *UserHistory*, esta permitirá que o mesmo hóspede se consiga registar em várias aplicações diferentes, caso utilize o mesmo nome e número de telemóvel. No entanto,

para o hóspede, enquanto utilizador da aplicação móvel isso não será relevante, visto que apenas poderá aceder ao jogo e às ações já realizadas individualmente em cada uma das aplicações. À tabela [Point](#), que permite identificar os diversos locais que os hóspedes podem visitar, foi adicionado o atributo *BeaconId*. Este atributo corresponde ao UUID de um Beacon, ou seja, o seu identificador único, que será usado na aplicação móvel. Cada local (tabela [Point](#)) que seja ativado ou descoberto através de *beacons* deverá ter este atributo preenchido.

As principais tabelas que vão fornecer informação para a aplicação móvel ([Action](#), [Level](#) e [Campaign](#)) tem um atributo *IsPublished* que, só quando se encontra assinalado como verdadeiro é que é enviado no serviço para as aplicações móveis. Na tabela [Entity](#), o atributo *IsManagingApplicationGames* permitirá indicar se aquele cliente terá o módulo de jogos ativo, caso contrário, não terá acesso aos menus de configuração dos jogos no Backoffice nem ao jogo em si na aplicação móvel.

A tabela [Level](#) contém a informação necessária para os níveis do jogo, a tabela [Campaign](#) contém a informação para persistir a informação de uma campanha e as tabelas [Action](#), [ActionMultimedia](#) e [ActionResponse](#) tem os atributos necessário para toda a informação que uma ação deve ter. A [ActionMultimedia](#) permite persistir todas as imagens ou áudio que uma ação pode ter e a [ActionResponse](#) é usada para ações apenas do tipo “Pergunta-Resposta”. Ou seja, terá as possíveis respostas de uma determinada ação, sendo possível assinalar a resposta correta.

5.1.4 Funcionalidades

Nesta secção são apresentadas as principais funcionalidades desenvolvidas. A gestão de níveis dos jogos, das ações, das campanhas e a classificação geral estão distribuídas por diferentes páginas e apresentam uma estrutura semelhante. A gestão de ações encontra-se explicada com maior detalhe, sendo apresentados ainda alguns excertos de código mais relevantes e que explicam o modo como está implementado. As restantes funcionalidades encontram-se implementadas da mesma forma. Estas são essenciais para que a aplicação móvel receba toda a informação necessária para construir o jogo.

Destaca-se ainda a existência de pequenos desenvolvimentos na página de gestão de pontos (ManagePoint) para incluir o campo *BeaconId* (identificador único do *beacon*) que permite

associar um determinado Beacon a um dado local. Foram também feitos desenvolvimentos nas configurações da **Entidade** (página ManageEntityConfiguration) para incluir o atributo *IsManagingApplicationGames* que permite ativar o módulo de jogos na aplicação móvel e no Backoffice, permitindo aos Clientes aceder às páginas de configuração dos jogos (níveis do jogo, ações e campanhas).

5.1.4.1 Gestão de níveis de jogo

Esta funcionalidade encontra-se disponível no Backoffice no menu *games*, submenu *game levels* e envolve todo o CRUD dos níveis de jogo.

The screenshot displays two parts of the 'game levels' management interface. On the left, under 'game levels', there is a search bar and a table of existing levels. The table has columns: id, name, entity, published, points, and last update. On the right, under 'new game level', there is a form with fields for 'name *', 'points nr.', 'entity *', and an 'icon' upload area. The 'entity *' dropdown is currently set to '18328 - citizenM Paris Charles de Gaulle Airport'. The 'icon' area has a placeholder 'Add Image (.gif .jpeg .jpg .png .tiff)' and a note 'recommended image dimensions: 70(W)70(H)'. There are 'back' and 'save' buttons at the bottom right of the form.

id	name	entity	published	points	last update
7	New	GuestU	<input checked="" type="checkbox"/>	300	2016-07-31 09:36:58
6	Novo Level	GuestU	<input checked="" type="checkbox"/>	200	2016-08-02 10:43:15
3	Expert traveller	Clarice Demo	<input type="checkbox"/>	100	2016-04-28 04:47:14
5	Game Level 2	GuestU	<input checked="" type="checkbox"/>	100	2016-08-02 10:45:32
4	Game Level 1	GuestU	<input checked="" type="checkbox"/>	12	2016-08-02 10:42:58
2	Mini traveller	Clarice Demo	<input type="checkbox"/>	10	2016-04-28 04:46:40

Figura 13 – Gestão de níveis de jogo no Backoffice

A Figura 13 mostra a User Interface (UI) desenhada para este ecrã. Para um nível de jogo é necessário que seja indicado o seu nome, o número de pontos máximo que o nível deve atingir e uma imagem ilustrativa. A nível de traduções, de publicação, edição, inserção e remoção o funcionamento é semelhante ao explicado anteriormente.

5.1.4.2 Gestão de ações

A gestão de ações encontra-se acessível através do menu *game*, no submenu *actions*. Nesta página é possível visualizar, inserir, editar e apagar toda a informação relacionada com ações. Tal como é possível observar na Figura 14, o ecrã encontra-se dividido em duas partes. Do lado esquerdo é possível observar a lista de ações já inseridas, onde é possível editar e eliminar cada uma delas. Além disso é também possível publicar cada uma das ações, para que estas estejam disponíveis para serem enviadas para a aplicação móvel. É também possível disponibilizar cada ação para ser traduzida nas mais diversas línguas, clicando para isso na lista de bandeiras.

Por fim, é ainda possível gerar os Códigos QR para os locais cujas ações são ativadas por Código QR. Do lado direito, encontra-se toda a informação que é necessário preencher para se inserir ou editar uma ação. É ainda possível verificar a existência do campo *entity*, acompanhado de uma *combobox*. Este campo aparece apenas para utilizadores com permissões de Administrador (empresa), tendo como objetivo a possibilidade de inserir e pesquisar ações para um cliente em específico.

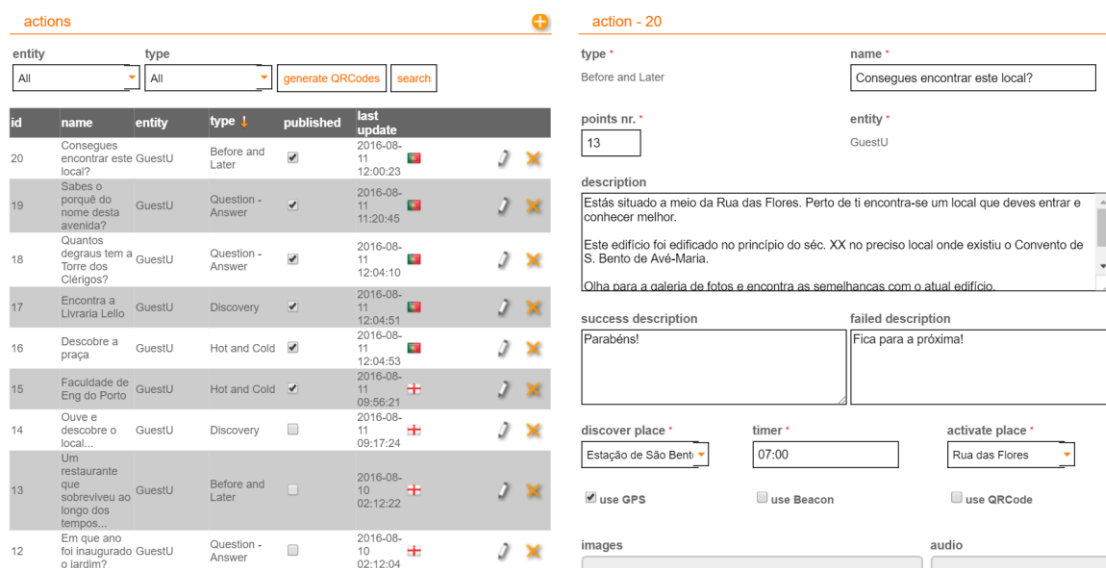


Figura 14 – Gestão de ações no Backoffice

Em termos de código, é necessário carregar inicialmente a informação da lista de ações, de entidades, de tipos de ações e dos locais disponíveis para associar as ações. Em relação à UI, é usado na grande maioria os elementos disponíveis da *framework* ASP.NET.

Para cada ação é necessário apresentar todas as línguas nas quais o conteúdo já foi traduzido, tal como é possível observar na Figura 14, na lista de ações. O método `GetAvailableTranslationLanguages(...)` (Figura 15) acede ao procedimento `[game].[ActionGetAvailableTranslationLanguages]`, que por sua vez comunica com a base de dados de traduções para aceder a todas as línguas nas quais uma determinada ação já foi inserida.

Para inserir/editar uma ação é necessário o preenchimento de diversos campos, sendo diferentes consoante o tipo de ação. Por exemplo, para o tipo de ação “Pergunta-Resposta”, uma secção para inserir as possíveis respostas fica visível. Além disso, consoante a entidade

escolhida os locais disponíveis também podem ser diferentes. Os locais disponíveis são todos aqueles que já estão associados à entidade, com exceção daqueles que já foram associados a ações dessa mesma entidade. A Figura 16 mostra o excerto de código que devolve a lista de locais disponíveis para associar a entidades.

```
//Parte do método LoadFlagTranslationsAction

Repeater rptflags = (Repeater)e.Item.FindControl("rptActionTranslations");

DataTable dtlanguages = actionController.GetAvailableTranslationLanguages(
ActionId, MobileApplicationId);

rptflags.DataSource = dtlanguages;
rptflags.DataBind();

// Parte do procedimento [game].[ActionGetAvailableTranslationLanguages]

SELECT L.ID, L.Icon AS ICON, L.Name AS LANGUAGE, CAST('1' as bit) as ADDED
FROM [game].[Action] A
INNER JOIN [4TRANSLATION].dbo.Translation T ON A.Name = T.RefCode
INNER JOIN [4TRANSLATION].dbo.CustomerContent CC ON CC.ContentCode = T.RefCode
INNER JOIN [4TRANSLATION].dbo.Content2Translator C2T ON C2T.IDCustomerContent
= CC.ID
INNER JOIN [4TRANSLATION].dbo.refLanguage L ON C2T.IDLanguage = L.ID
WHERE A.ID = @id
```

Figura 15 – Excerto de código da página ManageAction / Base de dados para devolver línguas inseridas de uma ação

Além do tipo de ação e entidade é necessário o preenchimento do nome da ação, do número de pontos a atribuir, caso a consiga concluir, o local onde a ação vai ser ativada, o local cujo objetivo é ser descoberto, o tempo máximo em que a ação pode ser concluída e o tipo de tecnologia que vai ser usada para ativar e descobrir os locais assinalados.

```
DataTable availablePlaces = null;

DataTable places = EntityController.GetPoints(entityId);

using (ActionHelper helper = new ActionHelper())
{
    DataTable actions = helper.Get(entityId, null);

    // Verifica a existência de locais já associados a ações, retirando-
    os das possibilidades.

}
```

Figura 16 – Excerto de código da classe ActionController que devolve os locais disponíveis

Outros campos podem ser preenchidos, como uma descrição da ação, imagens ilustrativas, um ficheiro áudio, ou ainda as respostas possíveis no caso de uma ação do tipo “Pergunta-Resposta”. Na Figura 17 é possível observar um excerto de código da classe `ActionController` que permite inserir a informação de uma ação. A operação é realizada dentro de uma transação para que se consiga inserir toda a informação da ação e os ficheiros multimédia de forma atómica. Importante ainda referir que, sempre que um conteúdo é inserido pela primeira vez, é adicionado na língua que se encontra em sessão naquele momento.

```
using (TransactionScope scope = new TransactionScope
(TransactionScopeOption.Required, option))
{
    using (ActionHelper helper = new ActionHelper())
    {
        actionId = helper.Insert(name, description, successDescription,
failedDescription, duration, pointsNumber, useGps, useQrCode, useBeacon,
activatePlaceId, discoverPlaceId, actionTypeId, entityId,
mobileApplicationId, username, languageId, customerId,
backofficeApplicationId);

        if (actionId > 0)
        {
            // Inserir os ficheiros multimédia (ActionMultimedia)
            // Inserir as respostas (ActionResponse) no caso de uma ação do
tipo “Pergunta-Resposta”
        }
    }
    scope.Complete();
}
```

Figura 17 – Excerto de código da classe `ActionController` para inserir informação de uma ação

Outro ponto importante a salientar ainda nesta página é a funcionalidade de gerar os Códigos QR. Tal como já foi referido, o objetivo é gerar os Códigos QR para todos os locais, de todas as ações, cuja ativação e descoberta seja feita por Código QR. É gerada uma página, tal com a lista dos códigos (Figura 18), para que possam ser recortados e colocados nos respetivos locais.



Figura 18 – Códigos QR exemplo gerados no Backoffice

Em termos de implementação, foi usada uma biblioteca chamada QR Code [59] para gerar os códigos QR. De entre as muitas disponíveis, foi tido em atenção o acesso à documentação disponível e o facto de estar disponível no *Nuget Packages* [60], um gestor de packages para plataformas de desenvolvimento Microsoft, usado no projeto do Backoffice para uma melhor gestão das bibliotecas usadas nele.

```
QRCodeEncoder encoder = new QRCodeEncoder
{
    QRCodeErrorCorrect = QRCodeEncoder.ERROR_CORRECTION.H,
    QRCodeEncodeMode = QRCodeEncoder.ENCODE_MODE.BYTE
};

Bitmap qrCodeImg = new Bitmap(encoder.Encode(data));
```

Figura 19 – Excerto de código da classe ManageAction que permite criar o código QR

Na Figura 19, é possível observar o excerto de código que permite gerar o código QR. A variável *data*, uma *string*, contém o valor que se pretende que o código possua. Nesse sentido, o formato definido é o seguinte:

```
"wdaa_p:" + Id do Local + "_a:" + Id da Ação
```

Por exemplo, considerando uma ação com o identificador único 1 e um local com o identificador único 33, o valor do código QR é *wdaa_p:33_a:1*. Este conteúdo será lido pela aplicação móvel, verificando se existe correspondência com o formato usado.

5.1.4.3 Classificação no jogo

No menu *games*, no submenu *ranking* é possível observar (Figura 20) a classificação dos diferentes utilizadores do jogo.

game ranking

entity

GuestU

user	level	points ↓
ana	Super viajante	288
maRta sa	Viajante sénior	143
sgsd	Viajante sénior	120
RicardoGuestU	Viajante intermédio	75
hh	Viajante intermédio	50
Moreno Teste	Viajante intermédio	40

Figura 20 – Classificação do jogo no Backoffice

A classificação é obtida pela soma dos pontos de todas as ações realizadas até ao momento. Mais uma vez, um utilizador com permissões de Administrador pode escolher a entidade para a qual pretende ver os resultados.

5.1.4.4 Gestão de campanhas

Na Figura 21, é possível observar a UI para a funcionalidade de gerir campanhas no Backoffice. Esta é acessível através do menu *games*, no submenu *campaigns*, no separador *campaign*. Nesta página é possível efetuar as operações CRUD para gerir as campanhas. Tal como na gestão das ações e de níveis de jogos, a UI usada é semelhante. Uma campanha é assim caracterizada por um nome, uma quantidade (número de pessoas que a podem adquirir), uma data de início e de fim (caso não seja permanente), uma descrição e uma imagem. Além disso, uma campanha pode ser adquirida caso atinja um determinado nível de jogo e/ou realize um determinado número de ações com sucesso, campos esses também disponíveis para serem preenchidos.

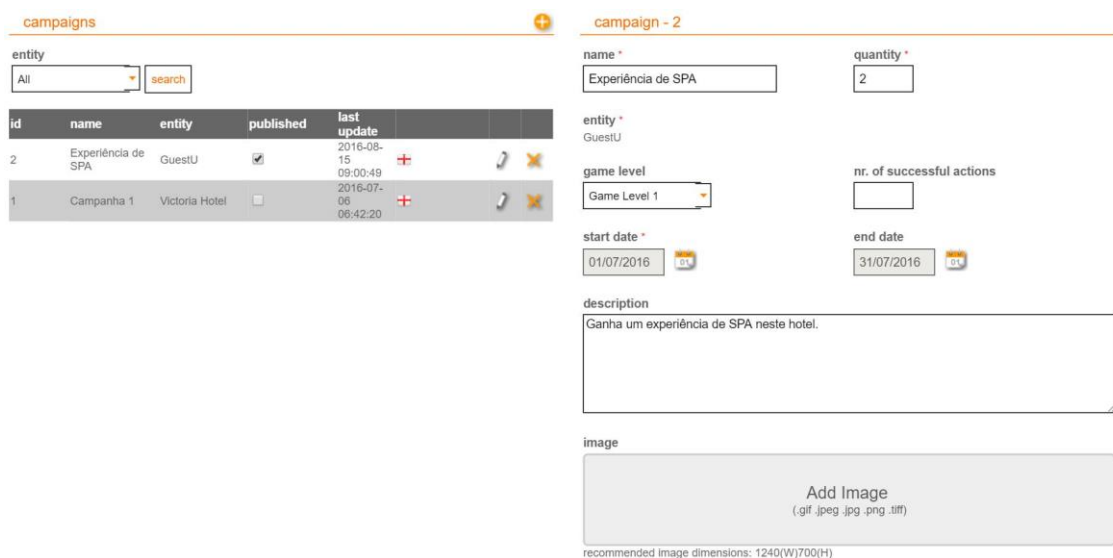


Figura 21 – Gestão de campanhas no Backoffice

5.1.4.5 Resultados das campanhas

A página de resultados das campanhas está acessível a partir do menu *games*, no submenu *campaigns*, no separador *results* e, tal como é possível observar na Figura 22, constitui uma lista com todos os utilizadores que ganharam uma determinada campanha.

Caso esteja autenticado como Cliente apenas verá os da sua entidade. Neste caso, o Administrador pode selecionar a entidade que pretender. Para que os Clientes tenham controlo das campanhas entregues e a quem, eles vão mudando o estado da campanha para aquele determinado hóspede, de modo a perceber se já recebeu o prémio ou não.

results

entity

16307 - GuestU

user	campaign	start date	end date ↓	state
ana	Experiência de SPA	01/07/2016 00:00:00	31/07/2016 00:00:00	Not delivered

Page 1 of 1

Figura 22 – Resultados das campanhas no Backoffice

5.1.5 Serviços para a aplicação móvel

Tal como já foi referido na seção 4.2, os serviços usados na aplicação móvel estão acessíveis através de um *Generic Web Handler* (.ashx), tal como todos os outros que a aplicação necessita. Os pedidos da aplicação móvel são feitos via HTTP e o formato de armazenamento e transmissão de informação é o JSON. Este *Handler* encontra-se inserido na arquitetura do Backoffice, ou seja, ele comunica com a camada de negócio que, por sua vez, comunica com a camada de acesso a dados. Os pedidos são identificados através do URL, no entanto, todos têm uma estrutura semelhante identificada por:

```
http://{{server}}/handlers/JsonProtocolHandler.ashx?Request=[{{serviceName}}=true;applicationId={{mobileAppId}};customerId={{customerId}}]
```

A classe *JsonProtocolHandler* representa o *Handler*; o campo *serviceName* indica o nome do serviço, ou seja, que ação ele deve desencadear; e o campo *applicationId* identifica o identificador único da entidade. Para este módulo foram necessários criar os serviços, descritos na Tabela 9.

Ainda em relação aos serviços, foi necessário acrescentar o atributo *IsManagingApplicationGames* ao serviço *entities=true*, que permite identificar se essa entidade tem acesso ao módulo de jogos. O mesmo acontece com o *BeaconId* (identificador único de um Beacon num dado local), tendo sido adicionado esse atributo aos serviços *points=true* e

routedescription=true. Em relação a estes dois últimos serviços foi adicionado o atributo *HasGame* que permite identificar se um dado local, para uma determinada entidade, tem um jogo associado.

Tabela 9 – Lista de serviços utilizados na aplicação móvel

Nome do serviço	Método HTTP	Descrição
actions = true	GET	Devolve todas as ações publicadas para uma dada entidade
gamelevels = true	GET	Devolve todos os níveis de jogo publicados para uma dada entidade
campaigns = true	GET	Devolve todas as campanhas publicadas para uma dada entidade
gameranking = true	GET	Devolve a pontuação de todos os jogadores para uma dada entidade
useractions = true	GET	Devolve todas as ações feitas por um determinado hóspede, numa determinada entidade
useractions = true	POST	Envia para o Backoffice a lista de ações de um determinado hóspede, numa determinada entidade, que ainda não tenham sido enviadas
usercampaigns = true	POST	Envia para o Backoffice a lista de ações de um determinado hóspede, numa determinada entidade, que ainda não tenham sido enviadas
messagingregisteruser = true	POST	Envia para o Backoffice os dados de um utilizador para que este se registre. Caso não exista é inserido. É devolvido o registo inserido ou já existente. Este serviço já se encontrava desenvolvido.

A informação entre o Backoffice e a aplicação móvel é transmitida em formato JSON, através da biblioteca *Json.Net* [61]. Por forma a mostrar o trabalho desenvolvido a nível de código apresenta-se de seguida o serviço *gamelevels = true* (método GET) e o serviço *useractions = true* (método POST). Os restantes serviços comportam-se de forma semelhante.

5.1.5.1 Serviço GameLevels (GET)

O serviço *gamelevels* é identificado ainda na classe *JsonProtocolHandler*, onde é feita a comparação entre o seu nome e o método HTTP requerido, tal como é possível observar na Figura 23. O método *GenerateGameLevelsResponse(context, requestParameters)* é responsável por comunicar com a classe na camada de negócio que gere os serviços (*ResponseHandlerController*) e retornar em formato JSON a lista de níveis de jogo.

Ainda na classe *ResponseHandlerController* é possível observar o excerto de código da Figura 24. Cada serviço tem por norma duas classes que lhe dão suporte: a classe *LevelDTO* (Data

Transfer Object), onde são mapeados os dados que vem da base de dados; a classe `GameLevelsResponse`, que contém a lista de objetos a retornar, neste caso, uma lista de `LevelDTO` e ainda um Dicionário de traduções (`Dictionary<string, Dictionary<string, string>>`).

```
if (parameters.ContainsKey("gamelevels"))
{
    if (!context.Request.HttpMethod.Equals("GET", StringComparison.
OrdinalIgnoreCase))
    {
        throw new ApplicationException("Invalid request Http method.");
    }

    GenerateGameLevelsResponse(context, requestParameters);
    return;
}
```

Figura 23 – Excerto de código da classe `JsonProtocolHandler` que valida o serviço a ser invocado

Para cada serviço é geralmente criado um procedimento novo que devolve um *DataSet* com toda a informação necessária. Neste caso, o *DataSet* devolve uma *DataTable* com a lista dos níveis de jogo e outra *DataTable* com todas as traduções associadas a esses níveis. O método `ConvertDataTableToTranslation(...)` é usado em todos os serviços que devolvam traduções.

```
LevelController levelController = new LevelController();
GameLevelsResponse gameLevelResponse = new GameLevelsResponse();

foreach (DataRow row in dtGameLevelsList.Tables[0].AsEnumerable())
{
    LevelDTO gameLevel = new LevelDTO()
    {
        Id = row.Field<int>("Id"),
        Name = row.Field<string>("Name"),
        PointsNumber = row.Field<int>("PointsNumber"),
        IconPath = row.Field<string>("Multimedia"),
        TimeChange = row.Field<DateTime>("TimeChange")
    };
    gameLevelResponse.GameLevels.Add(gameLevel);
}

// Translations
gameLevelResponse.Translations =
ConvertDataTableToTranslation(dtGameLevelsList.Tables[1]);
```

Figura 24 – Excerto de código da classe `ResponseHanlderController` que devolve os dados do serviço `gamelevels`

Por fim, é possível observar na Figura 25 um exemplo de resposta gerada por um pedido GET ao serviço *gamelevels*. É assim devolvida uma lista de *gamelevels* e um dicionário de traduções, convertidos do objeto *GameLevelResponse*.

```
{
  "gameLevels": [
    {
      "id": 4,
      "name": "GameLevel_name_4",
      "pointsNumber": 12,
      "iconPath": "http://wdaa-qa-01.cloudapp.net/Images/icons/GameLevel//backpack1_636057313606589614.png"
    },
    {
      "timeChange": "2016-08-16T18:30:28.870Z"
    }
  ],
  "translations": {
    "en": {
      "GameLevel_name_4": "Mini viajante"
    }
  }
}
```

Figura 25 – Exemplo de resposta do serviço *gamelevels*

5.1.5.2 Serviço *UserActions* (POST)

Neste serviço, e ainda na classe *JsonProtocolHandler* (Figura 26), é necessário ler e converter a informação enviada num objeto que contém as ações realizadas de um dado utilizador (objeto *UserActionDTO*).

```
List<UserActionDTO> userActions;

using (StreamReader requestReader = new
StreamReader(HttpContext.Request.InputStream))
{
    string requestContent = requestReader.ReadToEnd();
    userActions = JsonConvert.DeserializeObject
<List<UserActionDTO>>(requestContent);
}

ResponseHandlerController.UserActionsInsert(userActions);
```

Figura 26 – Excerto de código da classe *JsonProtocolHandler* que permite ler, converter e inserir informação sobre ações já realizadas

A lista de *UserActionDTO* é enviada para a camada de negócio e, de seguida, para a camada de acesso a dados que se vai encarregar de invocar o procedimento para inserir os dados na base de dados.

5.2 Aplicação móvel

Nesta seção, são apresentados os desenvolvimentos feitos na aplicação móvel. É importante referir que não foram implementadas as funcionalidades relativas às campanhas, tendo sido apenas realizados os desenvolvimentos no Backoffice. Nesse sentido são apresentadas inicialmente algumas considerações sobre a aplicação de modo a explicar alguns desenvolvimentos que já se encontram feitos e como se encontram implementados. De seguida, é apresentado um diagrama de classes simplificado e, por fim, detalhes da implementação e das funcionalidades desenvolvidas.

5.2.1 Considerações iniciais sobre a aplicação

Tal como já foi referido na seção 4.2 os principais desenvolvimentos são feitos no componente WDAA. Torna-se assim importante referir alguns aspetos iniciais de como a aplicação se encontra desenvolvida. Nesse sentido são feitas algumas referências à base de dados e à forma como é feita a sincronização de dados. São referidos aspetos de desenvolvimentos já pertencentes à aplicação antes deste módulo, sendo seguidas as mesmas regras para o desenvolvimento do jogo.



Figura 27 – Sincronização da informação e persistência na base de dados

O componente WDAA encontra-se estruturado por *packages* que agregam as diferentes funcionalidades da aplicação. Neste caso, é possível observar o package *services* na Figura 27, responsável pela gestão dos objetos persistidos na base de dados e pela sincronização da

informação. O *package games* contém todos os objetos criados para suportar a estrutura do jogo desenvolvido.

A classe `AppDatabaseHelper` gere a criação da base de dados e o acesso aos Data Access Object (DAO) de cada objeto (através da biblioteca ORM Lite). Por outro lado, a classe `API` contém os pedidos efetuados ao Backoffice. Por exemplo, na Figura 28 é possível observar o pedido para devolver as ações de uma dada entidade. Para efetuar os pedidos HTTP é usada a biblioteca Volley [62] e para converter o JSON retornado/ou enviar um JSON é usada a biblioteca GSON [63].

```
public static Task<Action.ActionData> getActions() {
    return MySightApi2.Get().get("actions=true", Action.ActionData.class,
    true);
}
```

Figura 28 – Pedido de ações ao Backoffice

Por sua vez, a classe `Sync` permite efetuar a sincronização dos dados com o Backoffice, sendo invocada sempre que a aplicação é aberta ou então forçando a atualização da informação, opção existente na aplicação.

```
public static Task<Void> fetchGameInformation() {

    Task<Void> task = Task.forResult(null);

    return task.continueWithTask(new Continuation<Void, Task<Void>>() {
        @Override
        public Task<Void> then(Task<Void> task) throws Exception {
            return Action.fetch().makeVoid();
        }
    }).continueWithTask(new Continuation<Void, Task<Void>>() {
        @Override
        public Task<Void> then(Task<Void> task) throws Exception {
            return UserRanking.fetch().makeVoid();
        }
    });

    // De notar que não estão presentes todos os pedidos/envios de informação
}
```

Figura 29 – Excerto da classe `Sync` para a sincronização dos dados do jogo

O método `fetchGameInformation()`, na Figura 29, permite que sejam efetuados os pedidos e envios de informação ao Backoffice e, conseqüentemente, a sua atualização dos dados na base de dados. Ainda em relação à Figura 29, observa-se que a sincronização de dados é feita com recurso a *Tasks*, através da biblioteca BoltsAndroid [64]. Operações que possam demorar mais

tempo como os pedidos HTTP ou mesmo acesso à base de dados são feitos com recurso a esta biblioteca. O método `fetch()` permite efetuar o pedido ao Backoffice (Figura 28), recebe os dados, converte-os no respetivo objeto (através do GSON) e insere ou atualiza na base de dados.

Ao terem sido feitas referências ao mecanismo de gestão da informação da aplicação (serviços e base de dados) torna-se importante explicar essa relação no que diz respeito aos objetos criados para o módulo a desenvolver. Na Figura 26, foram criados vários objetos que persistem a informação vinda dos serviços. A classe `Action` persiste a informação de cada ação, em conjunto com as classes `ActionResponse` e `ActionMultimedia` que dizem respeito às respostas (de uma ação do tipo “Pergunta-Resposta” e aos ficheiros multimédia, respetivamente. Esta é persistida quando é feito um pedido ao serviço `actions=true`. A classe `GameLevel` contém a informação dos níveis do jogo, sendo persistida ao pedido do serviço `gamelevels=true`. Por outro lado, a classe `UserAction`, persistida pelo pedido ao serviço `useractions=true`, contém informação de todas as ações realizadas pelo utilizador da aplicação. Por último, a classe `UserRanking` contém a classificação geral de todos os utilizadores que estejam inscritos, sendo persistida pelo pedido ao serviço `userranking=true`.

Tal como o Backoffice, a aplicação móvel encontra-se preparada para gerir as traduções de tudo que possa ser traduzível. A classe `Localization` trata do mapeamento entre as diversas línguas e é igualmente usada neste módulo. Além disso, palavras estáticas que apareçam no módulo dos jogos, como por exemplo títulos, nome de botões, nome de separadores são igualmente traduzidos no Backoffice e enviadas para a aplicação móvel. Por exemplo, o excerto de código `Localization.tf(AppTranslationKeys.GAME_NORESULTS_WINNERS);` indica que se está a aceder à tradução da atual língua da aplicação de uma *label* cujo texto é “Não existem resultados”.

A nível de *design*, o módulo de jogos usa as regras seguidas na aplicação, usando para isso os mesmos elementos de *design*. De notar que as cores da aplicação são também customizáveis no Backoffice. Outro aspeto importante a referir é o uso já existente na aplicação da API do *Google Maps* para mostrar num mapa onde se encontram determinados locais, tendo apenas sido necessário reconfigurar a chave de acesso.

5.2.2 Enquadramento

Por forma a tornar a leitura desta seção mais perceptível torna-se importante fazer um enquadramento entre a forma esta é apresentada e o fluxo de operações que acontece quando uma ação é realizada.

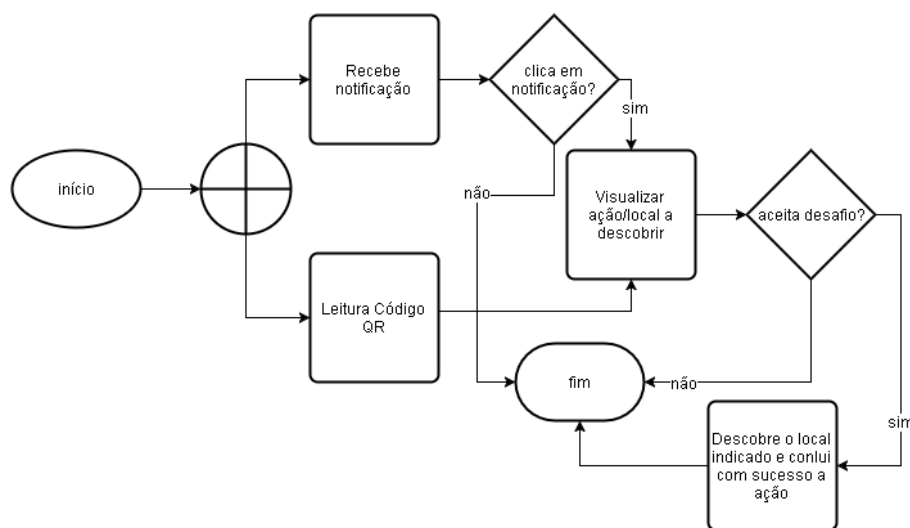


Figura 30 - Fluxograma para realização de uma ação na aplicação móvel

Na Figura 30, é possível observar o fluxograma que representa, ainda que de forma simplificada, a realização de uma qualquer ação na aplicação móvel, a partir do momento em que o utilizador se regista. Após o registo concluído, o utilizador está apto a jogar, isto é, a realizar ações. Para isso, ou recebe uma notificação no telemóvel (despoletada via GPS ou *Beacon*) ou lê um código QR num determinado local, ambas as situações representativas da existência de uma ação. Tanto ao ler o código QR, como ao clicar sobre a notificação, é mostrado ao jogador os detalhes da ação que foi ativada. São dadas dicas e informações relevantes de como chegar ao local pretendido e ainda é dada a possibilidade de aceitar ou não o desafio. Aceitando o desafio e encontrando o local pretendido, quer seja por leitura de um novo código QR ou despoletada por GPS ou *Beacon*, a ação é dada como concluída com sucesso e o registo de ações já realizados por parte daquele jogador em questão é atualizado.

Sendo a gestão do GPS, dos *Beacons* e dos códigos QR uma parte relevante desta seção, é dada especial atenção na subseção 5.2.4 (Gestão das Conectividades) onde é detalhada a forma como as ações interagem com as diferentes conectividades e, principalmente, com os diferentes tipos de ações que podem ser realizados. Importante salientar que, além do registo

e visualização de atividades, o utilizador tem acesso a outro tipo de informação, nomeadamente os níveis do jogo, a classificação geral e a lista de ações já realizadas por ele. Informação igualmente detalhada nesta seção.

5.2.3 Diagrama de classes

Na Figura 31 é possível observar parte do diagrama de classes implementado para o módulo de jogos a nível da aplicação móvel. As classes que representam os objetos persistidos na base de dados e o processo de sincronização de dados não se encontram representados, tendo sido feita uma breve referência a esse processo na seção anterior (5.2.1).

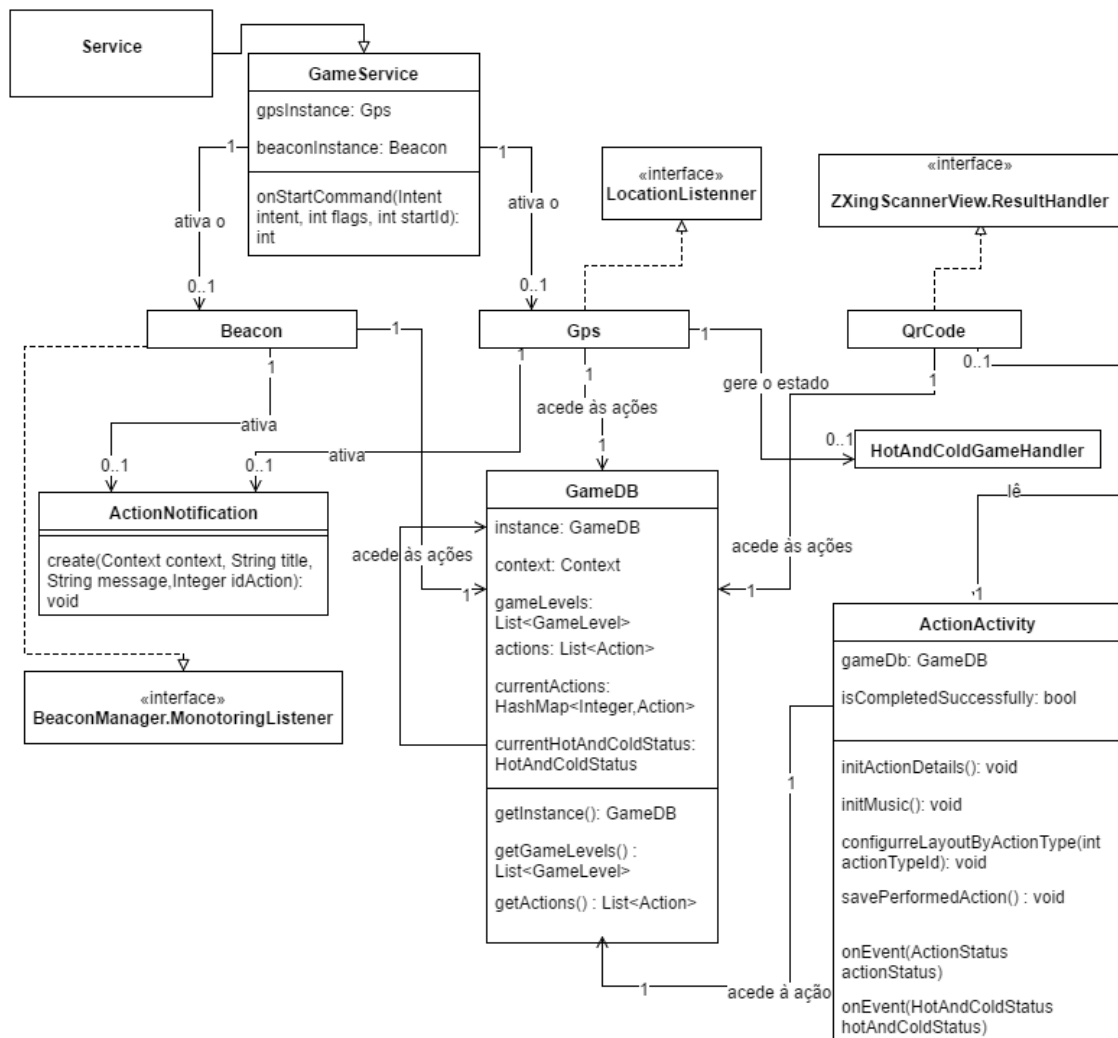


Figura 31 – Diagrama de classes simplificado da aplicação móvel

Não se encontram também representadas grande parte das classes da UI da aplicação móvel, sendo apenas apresentada a classe `ActionActivity`, responsável por apresentar a informação de uma dada ação a realizar. Nesta classe é assim gerida toda a informação relativa a uma ação, desde a sua ativação até ao seu término. Nesse sentido, ficaram por representar classes tais como `GameRegisterActivity`, responsável pelo registo dos utilizadores, ou ainda a classe `GameActivity`, responsável por apresentar a informação relativa às ações já efetuadas, à posição do jogador no jogo e os níveis do jogo existentes.

As classes `Beacon`, `GPS` e `QRCode` representam as três conectividades presentes neste jogo, sendo que a classe `GameService` é um serviço que permite a ativação das conectividades `Beacon` e `GPS`. Por outro lado, a classe `ActionNotification` permite o envio de notificações sempre que uma ação ativada por `GPS` ou `Beacon` é descoberta. Mais detalhes sobre a gestão deste serviço são apresentados na seção 5.2.4.

A classe `GameDB`, um *singleton*, é responsável por gerir toda a informação necessária ao jogo. Informação como a lista de ações ou a lista de níveis de jogo, que apenas podem sofrer alterações quando os dados da aplicação móvel são atualizados, são persistidos em objetos nessa classe. Esta classe é também responsável por gerir a lista de ações que estão atualmente a decorrer, ainda que o jogo tenha a limitação de apenas se poder efetuar uma ação de cada vez.

Torna-se importante referir que, nas próximas seções serão apresentados mais detalhes tendo por base o diagrama de classes da Figura 31.

5.2.4 Gestão das Conectividades

O jogo usa três tipos de conexões para interagir com o jogador, ou seja, as ações podem estar configuradas para serem ativadas/descobertas por código QR, Bluetooth ou GPS. O código QR permite um modo de jogo em que o utilizador necessita de ler um código QR para iniciar a ação e outro para a terminar. Apesar de ser manual, representa mesmo o modo mais económico para a bateria do telemóvel. Outro modo de jogo é através do Bluetooth 4.0+, caso este esteja disponível no telemóvel. Este modo permite ao jogador que, caso tenha o Bluetooth ligado, procurar por *Beacons* na sua proximidade que dão o início/fim à ação. Por fim, o GPS comporta-

se de forma semelhante. Quando ligado permite ao jogador iniciar/terminar ações sempre que é detetado a uma determinada distância do local de ativar/descobrir.

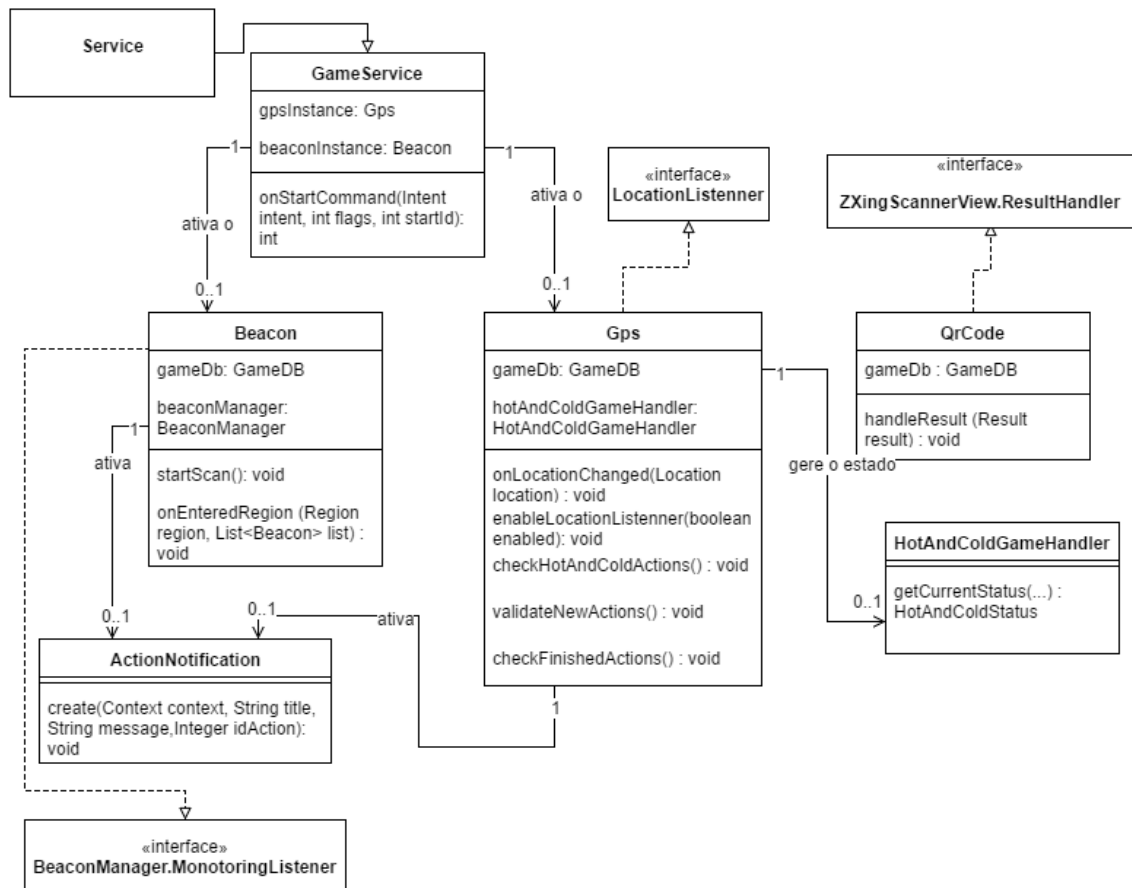


Figura 32 - Diagrama de classes detalhado do serviço e conectividades

A gestão do GPS e *Beacon* é feita através de um serviço, sendo sempre enviada uma notificação de cada vez que uma ação é despoletada. Assim, mesmo tendo a aplicação minimizada, o utilizador recebe as notificações, notificando-o de possíveis atividades que pode realizar. Na Figura 32 é possível observar o diagrama de classes do serviço e das respetivas conectividades que são ativadas automaticamente pela aplicação.

A classe `GameService` está responsável por monitorizar as conectividades, existindo métodos para inicializar e parar a procura de ações por *Beacons* e GPS. O serviço é iniciado após o *SplashScreen* da aplicação, verificando primeiro se o hóspede já se encontra registado. O código usado pode ser observado na Figura 33.

Na atual solução as conectividades estão sempre ativas (`GPS_ON` e `BEACON_ON`), no entanto, durante o desenvolvimento e para facilitar os testes existia uma secção na aplicação que permitia desligar as conectividades, daí a existência de código para iniciar/parar as respetivas.

```
startService(new Intent(this, GameService.class).putExtra("connections",  
GameService.BEACON_ON + GameService.GPS_ON));
```

Figura 33 – Excerto de código que permite iniciar o serviço para detetar ações por Gps e Beacon

Nas subsecções seguintes é explicado em maior detalhe o funcionamento das notificações e de cada uma das conectividades existentes. No caso de GPS e do Beacon são tidos em atenção os métodos presentes no diagrama de classes da Figura 32. Antes de serem referidos mais detalhes, é importante ter em atenção que o conceito do jogo só suporta a realização de uma ação de cada vez, ou seja, sempre que uma ação é inicializada e é aceite o desafio, não pode ser inicializada outra ação sem que esta esteja concluída.

5.2.4.1 Notificações

Quando uma ação que é descoberta por *Beacon* ou GPS é desencadeada, é enviada uma notificação ao utilizador, usando para isso a classe `ActionNotification`. Clicando na notificação tem acesso à ação em si, podendo ou não aceitar o desafio. As notificações são enviadas através do código representado na Figura 34.

```
ActionNotification.create(context,  
Localization.tf(AppTranslationKeys.GAME_NOTIFICATION_TITLE),Localization.tf(Ap  
pTranslationKeys.GAME_NOTIFICATION_MESSAGE),action.getId());
```

Figura 34 – Excerto de código da classe Beacon e Gps para lançar uma notificação

Se o utilizador descartar a notificação, a ação que está associada a esta é marcada como `IsAlreadyUsed = true`, parâmetro usado pela `Action` e que permite indicar que a ação em específico já foi realizada e que não pode voltar a ser realizada. No entanto, foi considerado que, mesmo descartando a ação, deveria ser possível ao utilizador voltar a receber essa notificação. Nesse sentido, sempre que o hóspede descarta uma notificação, ele volta a ser notificado 24 h depois, caso volte a passar no mesmo local. Isto foi conseguido através do uso das `SharedPreferences`, sendo gravado o identificador da ação e a data da última vez em que foi notificado.

Sempre que o utilizador clica na notificação esta é redirecionada para a classe `ActionActivity`, sendo enviado o identificador único da ação (`action.getId()`) para que a classe tenha acesso à ação correta e possa mostrar os detalhes da mesma.

5.2.4.2 GPS

O método `onLocationChange(Location location)`, implementado na interface nativa do Android `LocationListener`, é invocado sempre que é detetada uma nova localização emitida pelo GPS. Quando tal acontece é necessário verificar três situações: se existem novas ações que podem ser ativadas (`validateNewActions()`), se existem ações que necessitam de ser terminadas (`checkFinishedActions()`) e se existe uma ação do tipo Quente e Frio (`checkHotAndColdAction()`) a decorrer.

```
private void validateNewActions(){
    if(currentActions.isEmpty()){
        for(Action action : availableActions){
            if(action.getActivatePlace()!=null &&
currentActions.containsKey(action.getId())==false && action.getUseGPS()){
if(this.location.distanceTo(action.getActivatePlace().getLocation()) <
(action.getActivatePlace().getRadius()!=0 ?
action.getActivatePlace().getRadius() : DISTANCE_TO_ACTION)){
                if(!action.isAlreadyUsed() &&
gameDB.isReadyToNotify(action.getId())){
                    // mostra notificação e adiciona esta ação às SharedPreferences
                }
            }
        }
    }
}
```

Figura 35 – Excerto de código da classe GPS para validação novas ações

Na Figura 35, é possível observar o excerto de código do método `validateNewActions()`. A validação da existência de novas ações envolve alguns pormenores. Primeiro é necessário verificar se já não está uma a decorrer. De seguida, é importante verificar se na lista de ações disponíveis esta é uma ação que deva ser ativada por GPS e se a distância a que o jogador se encontra está contida na distância mínima definida para aquele local no Backoffice. Ou seja,

todos os locais (`Point`) tem um atributo `Radius` que indica o raio a que pode ser detetado. Não estando este atributo preenchido é aplicado por omissão o valor de 30 m (`DISTANCE_TO_ACTION`). Por fim, esta ação apenas deve ser notificada caso se verifiquem duas situações: se o utilizador nunca a jogou, ou caso já tenha sido notificada num período máximo de um dia (pormenor já referido na seção 5.2.4.1).

No método `checkFinishedActions()` é igualmente necessário verificar alguns pontos. Primeiro se alguma ação está a decorrer e, caso esteja, se a mesma é descoberta por GPS. Caso seja, e o local a descobrir estiver à distância definida mínima para ser detetado (através do atributo `Radius` do local ou do atributo `DISTANCE_TO_ACTION`) é despoletado um evento para a classe `ActionActivity` dando indicação que esta ação já terminou.

O método `checkHotAndColdActions()` tem algumas particularidades. Este método é usado para ações do tipo “Quente e Frio”, visto que a deteção da proximidade do utilizador é sempre feita por GPS. Ou seja, uma ação pode ser do tipo “Quente e Frio” mas ser ativada e descoberta pela leitura de um código QR. Em termos de implementação optou-se pela criação de cinco estados (`ICE_COLD`, `COLD`, `WARM`, `HOT`, `VERY_HOT`), que correspondem a cinco cores diferentes, sendo atribuído a cada um deles uma distância mínima e máxima que os identifica. Caso a ação que esteja a decorrer e seja do tipo “Quente e Frio”, este método despoleta um evento para a classe `ActionActivity`, onde envia o atual estado, previamente calculado. Na seção 5.2.4.5 são apresentados mais detalhes.

5.2.4.3 Beacon

Para a implementação da pesquisa por *Beacons* foi usado o SDK disponibilizado pela Estimote, o fornecedor escolhido, tendo sido adicionado ao projeto `'com.estimote:sdk:0.10.5@aar'`. A classe `Beacon` implementa assim a interface `BeaconManager.MonitoringListener` para que este receba os UUID dos *Beacons* pretendidos (identificadores únicos) e os consiga detetar.

O método `startScan()`, na Figura 36, permite inicializar a pesquisa de *Beacons*. Para tal, o método `startMonitoring` espera receber um conjunto de regiões que se pretendam monitorizar. Essas regiões são identificadas pelos `UUIDs` de cada Beacon. O `UUID` de cada Beacon está associado a cada local em específico, através do atributo `BeaconId`. De notar que, na Figura 36, apenas é apresentada uma parte do código em que associa os `UUID` dos locais a ativar, no entanto, é igualmente necessário adicionar os `UUID` dos locais a descobrir.

O método `onEnteredRegion(Region region, List<com.estimote.sdk.Beacon> list)` é implementado da interface `MonitoringListener` e é invocado de cada vez que é detetado um Beacon na região. Este método permite então validar a existência de novas ações ou a finalização de alguma ação que esteja eventualmente a decorrer. O funcionamento é semelhante ao do GPS, no entanto, torna-se necessário verificar se a ação é ativada/descoberta por *Beacon* e se o UUID encontrado (obtido através de `region.getProximityUUID()`) é o pretendido.

```
beaconManager.setMonitoringListener(this);

beaconManager.connect(new BeaconManager.ServiceReadyCallback() {
    @Override
    public void onServiceReady() {

        for (Map.Entry<UUID, Action> entry : activateActions.entrySet()) {

            if(entry.getValue().getUseBeacon()){
                beaconManager.startMonitoring(new Region("monitored region",
entry.getKey(), null, null));
            }
        }
    }
});
```

Figura 36 – Excerto de código da classe Beacon para iniciar a deteção de ações por Beacons

A nível de distância de deteção dos *Beacons*, os testes que foram realizados não permitem obter valores concretos. No entanto, a Estimote indica que, numa situação real a deteção seja efetuada entre os 40 a 50 metros [65].

5.2.4.4 Código QR

Para a leitura dos códigos QR foi usada a biblioteca BarcodeScanner [66]. Nesse sentido, a classe `QRCode` implementa a interface `ZXingScannerView.ResultHandler` para ter acesso ao método `handleResult(Result rawResult)` responsável pelo resultado de um código QR acabado de ler com sucesso. Este método é assim responsável por ativar e terminar uma ação.

A validação do resultado da leitura do código QR é feito através do padrão `Pattern.compile("^wdaa_p:([0-9]+)_a:([0-9]+)")`, que permite identificar desde logo como grupos, o identificador da ação e do ponto a ser lido. Não sendo verificada que a *string* lida se encontra no padrão esperado, é apresentada uma mensagem indicativa de que o código QR não pertence ao jogo.

Mais uma vez o processo para descobrir/terminar uma ação é semelhante aos anteriores. Neste caso, são feitas validações adicionais:

- Se o utilizador ativar uma ação e, de seguida, voltar a ler o mesmo código é apresentada uma mensagem indicativa de que a ação já foi iniciada;
- Se o utilizador tentar ativar uma ação lendo um código QR de um local que é para descobrir é apresentada uma mensagem indicativa de que o local não pode ser ativado, apenas descoberto;
- Se o utilizador realizar uma ação e, mais tarde, tentar realizá-la de novo, recebe uma mensagem indicativa de que já realizou a ação;
- Se o utilizador ler um código QR de um local que, por algum motivo, já não tem jogo associado, é apresentada uma mensagem de que a ação já não está ativa.

5.2.4.5 Publisher/Subscriber

Tal como foi referido em todas as conectividades, sempre que uma ação é terminada, é despoletado um evento para a classe `ActionActivity`.

```
action.getStatus().setStatus(ActionStatus.Status.END);
EventBus.getDefault().postSticky(action.getStatus());
```

Figura 37 – Excerto de código da classe GPS que permite identificar o término de uma ação

Na Figura 37, é possível observar o excerto de código que envia o evento. A classe `EventBus` pertence a uma biblioteca [67] já usada na aplicação que permite a gestão de eventos *publish/subscriber*. Através destes eventos é permitido às conectividades comunicarem com as ações na aplicação sem terem que gerir o estado em que as mesmas se encontram. Pelo `EventBus` estas enviam um evento, notificando a ação atual que esta tem que ser terminada.

```
HotAndColdGameHandler.HotAndColdStatus status =
HotAndColdGameHandler.getCurrentStatus((int)this.location.distanceTo(action.ge
tDiscoverPlace().getLocation()));

EventBus.getDefault().postSticky(status);
```

Figura 38 – Excerto de código da classe GPS para despoletar um evento para uma ação “Quente e Frio”
Por outro lado, quando uma ação do tipo “Quente e Frio” está a decorrer, é enviado um evento para a atividade atual do tipo `HotAndColdStatus` (ICE_COLD, COLD, WARM, HOT, VERY_HOT). Para este tipo de evento é usada a classe `HotAndColdGameHandler` que recebe a distância atual do utilizador para o local a descobrir. Através dessa distância, verifica em que intervalo de

distâncias se encontra, e atribui um dos cinco estados possíveis, enviando-o para a atividade (Figura 38).

5.2.5 Funcionalidades

Nesta secção são apresentadas as principais funcionalidades implementadas na aplicação móvel. Foram desenvolvidos diversos ecrãs para o registo dos hóspedes, da listagem das ações já realizadas, da classificação geral, dos níveis do jogo e da ação a realizar. Além disso, foram feitas alterações na listagem de locais e nos detalhes dos mesmos.

É importante referir que, durante todo o documento foi sempre referida a palavra **Ação** como o conceito principal do jogo, sendo este o termo usado não só a nível do Backoffice, como também da aplicação em móvel. Mas, em relação à UI da aplicação móvel, o termo **Ação** foi substituído por **Atividade**, por se considerar ser de mais fácil compreensão para o utilizador final. Não é usado o termo **Atividade** em todo o desenvolvimento porque no Backoffice este conceito já existe.

5.2.5.1 Registo dos utilizadores e Visualização dos locais com ações

O hóspede apenas pode jogar a partir do momento que se inscreve no jogo. Quando abre a aplicação, aparece um ícone identificativo da existência do jogo. Clicando nele aparece o ecrã esquerdo da Figura 39.

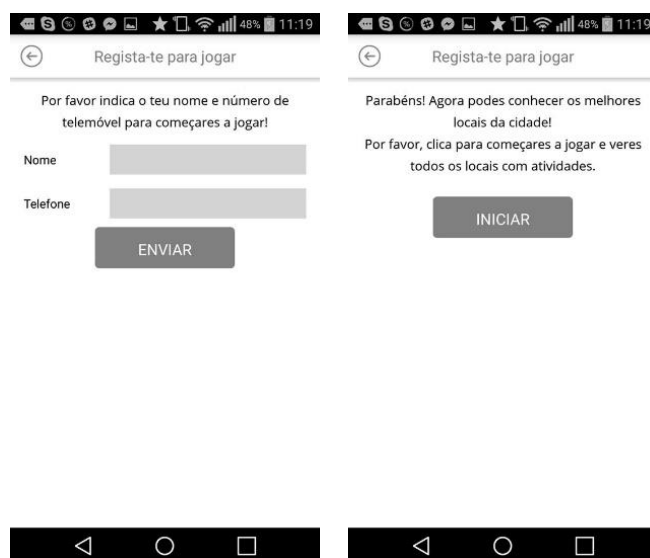


Figura 39 – Ecrã para registo de utilizadores

Para se registar tem então de indicar apenas o nome e o número de telemóvel. Caso o utilizador já se encontre registado, são devolvidos os seus dados e o identificador do utilizador já existente. Esta verificação é feita no Backoffice e corresponde à verificação do número de telefone. Depois de registado, o utilizador acede ao ecrã do lado direito, onde tem acesso a uma breve explicação e, ao clicar no botão “Iniciar” é redirecionado para o ecrã da lista de locais. Em termos de implementação, depois de efetuar o pedido ao Backoffice para o registo, os seus dados são gravados na classe `Registry` que se encarrega de gravar os mesmos nas `SharedPreferences` do telemóvel. Na mesma `Task` em que o utilizador é registado, é necessário verificar se este hóspede já tem ações realizadas.

O excerto de código da Figura 40 permite fazer um pedido ao Backoffice para retornar as ações já realizadas, sendo necessário a posterior atualização da lista de ações existentes na aplicação, através do método `UpdateWithActionsPerformed()`. É ainda inicializado o serviço que permite detetar ações por GPS e *Beacons*.

```
Task<Void> taskUserAction = Sync.fetchUserActions();
taskUserAction.onSuccess(new Continuation<Void, Void>() {

    @Override
    public Void then(Task<Void> task) throws Exception {
        GameDB.getInstance().UpdateWithActionsPerformed();
        return null;
    }
});
```

Figura 40 – Excerto de código para aceder às ações já realizadas do utilizador, depois de registado

A partir do momento em que o hóspede se regista, pode aceder à lista de locais e ver aqueles que têm ações para realizar, identificadas por um ícone ilustrativo do jogo (Figura 41, ecrã esquerdo). Importante ter em atenção que o ícone do jogo só está visível se a entidade tiver o módulo de jogos ativos (`isManagingApplicationGames()`), se o utilizador estiver registado (`isRegistered()`) e se o próprio local tiver um jogo associado (`getHasGame()`).

A pequena bola acima do ícone pode ser vermelha ou verde, consoante o utilizador já tenha realizado a atividade ou não (bola vermelha caso ainda não tenha realizado). Neste caso, na Figura 41 é possível observar o local Avenida dos Aliados, com um jogo disponível para se realizar (bola vermelha). Caso um local tenha um jogo, ao ver os detalhes desse local pode então aceder a informação que lhe indica o que ele tem de fazer para ser ativado (Figura 41, ecrã direito). Além disso, terá acesso ao mapa para perceber como chega ao local para iniciar a ação

(já desenvolvido anteriormente). As mensagens de como o jogo é ativado estão pré-definidas e mudam consoante a ação seja ativada por código QR, GPS ou Bluetooth.

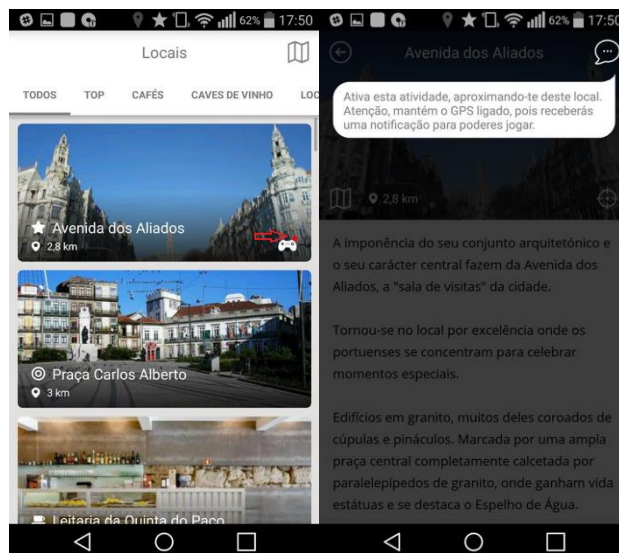


Figura 41 – Ecrã de listagem de locais e detalhes de um local, respetivamente

5.2.5.2 Visualizar e Realizar uma ação

Nas Figura 42, Figura 43 e Figura 44, é possível observar ecrãs de ações em diferentes momentos.



Figura 42 – Ecrã com descrição da ação

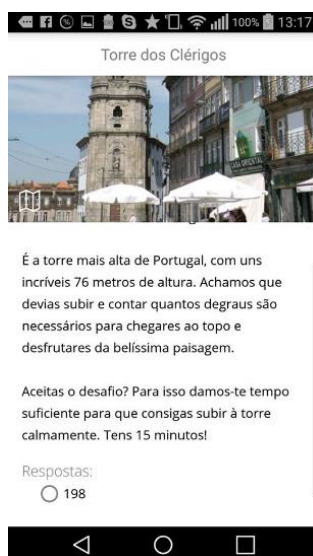


Figura 43 – Ecrã com ação do tipo “Pergunta-Resposta”



Figura 44 – Ecrã da ação tipo “Quente e Frio”

No primeiro ecrã (Figura 42), é possível observar toda a informação de uma ação: as imagens em que, caso não existam é usada uma por omissão; o mapa, onde é possível observar o local onde está naquele momento; o áudio que, caso não exista, o ícone é escondido; o nome e a descrição da ação e os botões em que é possível aceitar ou não a ação. Aceitando uma ação e, escolhendo que a quer efetivamente realizar, deixa de a poder cancelar, estando obrigado a que, caso não queira jogar, tenha de deixar o tempo terminar. O segundo e terceiro ecrã mostram diferentes tipos de ações quando depois de aceites. Numa ação do tipo “Pergunta-Resposta” (ecrã da Figura 43) é possível visualizar as possíveis respostas e o botão para responder.

Numa ação do tipo “Quente-Frio” (ecrã da Figura 44) é possível observar um esquema de cores (Cold, Warm, Very Hot) que vai pintando consoante a proximidade do local pretendido. Ambos os ecrãs têm em comum um cronómetro (ainda que não seja visível na Figura 43) em que o tempo decresce até aos 0 minutos, altura em que a ação termina sem sucesso. As ações que sejam ativadas e descobertas por código QR têm visível uma câmara no canto superior direito (Figura 44) que, neste caso, é usada para ler o código QR que permite terminar a ação com sucesso.

```
UpdateUserAction(userAction);
UpdateUserRanking();

if (NetworkingUtils.hasInternet()) {
    List<UserAction> userActions = new ArrayList<UserAction>();
    userActions.add(0, userAction);

    return API.sendUserActions(userActions).onSuccess(new
Continuation<UserAction, Boolean>() {
        @Override
        public Boolean then(Task<UserAction> task) throws Exception {
            boolean ok = task.isFaulted();
            if (!ok) {
                userAction.setConnected(true);
                UpdateUserAction(userAction);
            }
            return ok;
        }
    }).continueWith(TaskUtils.<Boolean>getPassThruLogErrorContinuation(TAG,
"Error fetching User Actions")).makeVoid();
}
```

Figura 45 – Excerto de código da classe ActionActivity que guarda informação de uma ação realizada

Na Figura 45 é possível observar parte do excerto do código que gere a informação sempre que uma ação é terminada. Neste caso, se houver ligação à Internet, são enviados diretamente para o Backoffice os dados relativos a essa ação, sendo igualmente gravados na base de dados. Caso não haja ligação, as ações realizadas são enviadas sempre que a aplicação é aberta ou sempre que é forçado um atualizar da aplicação. Os métodos `UpdateUserAction(userAction)` e `UpdateUserRanking()` permitem gravar a ação realizada na base de dados e atualizar a posição do jogador (número total de pontos e nível atual) na classificação atual. Estes pormenores são importantes para que o jogo consiga funcionar em modo *offline*.

No fim, é apresentada uma mensagem de sucesso ou insucesso de ação realizada, usando para isso a `SuccessDescription` e `FailedDescription` inseridos no Backoffice. Além disso, se uma ação é terminada com sucesso o topo do ecrã da ação fica com o nome do local descoberto, assim como o mapa, que passa a indicar o preciso local onde se encontram.

5.2.5.3 Lista de ações realizadas, de classificação geral dos níveis do jogo

Nas Figura 46 é possível observar o ecrã de informações gerais do jogo que contém três separadores: a lista de ações já realizadas, a classificação geral e a lista dos níveis do jogo.

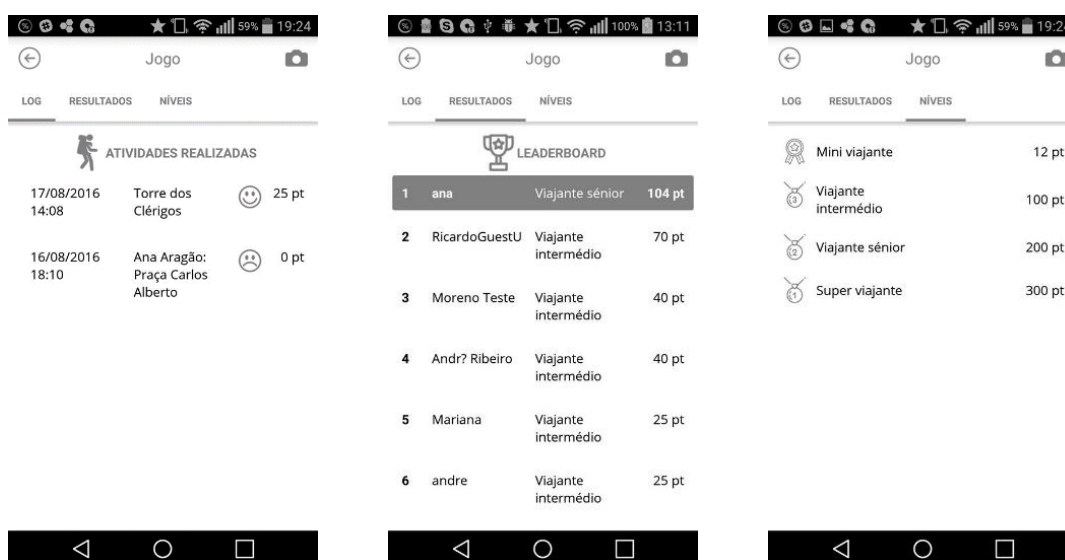


Figura 46 – a) Ecrã das ações realizadas

b) Ecrã da classificação geral

c) Ecrã dos níveis do jogo

Este ecrã é acessível ao clicar no ícone do jogo, presente na página inicial da aplicação. Tal como já foi referido, este ícone tem duas ações diferentes, ou seja, quando o utilizador não está

registado é aberto o ecrã de registo, caso contrário é sempre aberto o da Figura 46. Na lista de ações realizadas observa-se a data em que foi realizada, o local que foi descoberto, um ícone que indica se foi concluída com sucesso ou não e ainda os pontos adquiridos. Na lista de resultados, ordenados por ordem crescente, é possível verificar a posição, o nome do utilizador, o nível em que se encontra e o número de pontos adquiridos até ao momento. Além disso, o utilizador que usa a aplicação encontra-se sempre destacado, de modo a que consiga verificar melhor a sua posição no jogo.

De notar que, em ambas as listas e de cada vez que se clica em cada uma, se existir ligação à *Internet* é feito um pedido ao Backoffice para devolver os dados, caso contrário mostra os que estão atualmente na base de dados. Por exemplo, a Figura 47 mostra essa atualização de dados para a classificação geral.

```
if (NetworkingUtils.hasInternet()) {  
    Sync.fetchUserRanking();  
}
```

Figura 47 – Excerto de código da classe GameResultsFragment para atualizar a classificação geral

O ícone da camara, pertencente a este ecrã, é a única forma de ativar ações que sejam ativadas por código QR. Para o fazer, o utilizador terá sempre de aceder a este ecrã e clicar na câmara para ler o respetivo código QR.

5.3 Sumário

Nesta seção, foram descritos detalhes da implementação para a solução proposta. Foram apresentados detalhes dos desenvolvimentos no Backoffice, não esquecendo o suporte aos serviços necessários à aplicação móvel. No fim, foram apresentados os desenvolvimentos na aplicação móvel.

6 Avaliação da solução

Neste capítulo, é apresentada a avaliação realizada à solução proposta. Para tal, são apresentadas inicialmente as grandezas, as hipóteses e as metodologias usadas para a avaliação deste módulo. São disponibilizados os resultados obtidos mediante a análise estatística.

6.1 Grandezas e Hipóteses

Para avaliar este trabalho, considera-se que será necessário ter em atenção três grandezas. Primeiro, a satisfação dos utilizadores finais, ou seja, os hóspedes. Isto porque a implementação deste módulo só tem sucesso se os utilizadores realmente gostarem deste tipo de abordagem. Só assim voltarão a usar a aplicação e a recomendá-la. Segundo, e ainda em relação aos utilizadores, é também importante perceber se com o uso da aplicação os utilizadores procuraram obter um maior conhecimento da cidade onde se encontram e se entendem que este tipo de jogos pode realmente contribuir para tal. Por fim, a terceira grandeza que se pretende avaliar está direcionada para os clientes deste módulo, hotéis, *hostels* e apartamentos. Torna-se importante perceber se os utilizadores passaram a usar mais a aplicação com este tipo de abordagem, principalmente no que toca à aquisição de serviços, reservas, atividades e eventos fornecidos pelo estabelecimento em si.

Neste sentido, impõem-se três hipóteses em avaliação:

- *Os utilizadores gostam deste tipo de abordagem?;*
- *Usando jogos sérios, os utilizadores procuraram obter um maior conhecimento da cidade onde se encontram?;*
- *Os utilizadores passam a consumir mais os serviços do hotel (reservas de atividades, eventos, quartos,...)?.*

6.2 Metodologias de avaliação

Idealmente deveriam ser feitos testes com este módulo num estabelecimento hoteleiro e os respetivos hóspedes, durante um período alargado de tempo, por forma a ter acesso a resultados o mais realista possíveis. Mas, não sendo objetivo desta dissertação a integração deste módulo num contexto real de negócio é necessário recorrer a testes num contexto simulado e com grupos de utilizadores mais reduzidos. Sendo assim, para avaliar as grandezas acima referidas foi usado um grupo de trinta e duas pessoas que tiveram, em dada altura, na cidade do Porto. Todos tiveram acesso a uma aplicação móvel que representa um hotel fictício.

Os testes dividiram-se em duas fases: numa primeira fase o grupo testou uma aplicação sem o módulo de jogos implementado, ou seja, uma versão da aplicação GuestU tal como existe atualmente; Numa segunda fase, o mesmo grupo teve acesso à mesma aplicação, mas com o módulo de jogos integrado. É importante destacar que a aplicação fornecida com o módulo de jogos não tinha ações a serem ativadas por Beacon, porque não havia condições para os colocar nos respetivos locais. Foi igualmente fornecida uma imagem com os códigos QR que correspondiam a ações que deveriam ser ativadas/descobertas por eles. A aplicação foi assim configurada para conter cinco atividades espalhadas pela baixa da cidade do Porto.

Tanto na primeira fase de testes, como na segunda, o conjunto de pessoas respondeu a um questionário. Estes permitiram avaliar as duas primeiras grandezas, ou seja, a satisfação do utilizador e o conhecimento adquirido. De notar que, o conjunto de questões foi igual em ambas as fases, no entanto, o questionário da segunda fase teve mais quatro questões relativas ao módulo de jogos. Para cada questão semelhante dos questionários é apresentada a proporção amostral e o respetivo intervalo de confiança. De forma a permitir a comparação entre aplicações com e sem jogos optou-se por uma comparação de proporções entre as duas fases, traduzida através de intervalos de confiança para 95%. Para o grupo de questões exclusivas do módulo de jogos é apresentada a proporção amostral.

A terceira grandeza não foi alvo de avaliação. Só faria sentido avaliá-la se as campanhas tivessem sido implementadas. Tal como já foi referido, as campanhas funcionariam como ferramenta para os clientes (hotéis, *hostels* e apartamentos) comunicarem com os seus hóspedes, permitindo assim a divulgação dos seus serviços, atividades e eventos.

6.3 Resultados obtidos

Apresentam-se de seguida os resultados obtidos em relação às questões comuns a ambos os questionários e em relação às questões efetuadas apenas no questionário relativo à aplicação com o módulo de jogos incorporado. Em ambos os casos, são apresentados gráficos (Figura 48 à Figura 51) para uma facilitada observação dos resultados, onde é possível observar as proporções obtidas em percentagem e as respetivas barras de erro (intervalos de confiança), obtidas através do cálculo do intervalo de confiança para 95% (no caso das questões semelhantes).

6.3.1 Questões semelhantes aos questionários

Da Figura 48 à Figura 51 é possível observar as questões feitas em ambas as fases. Mais detalhes podem ser analisados na Tabela 10, onde é possível observar cada uma das questões efetuadas, e os respetivos valores obtidos para o grupo que usou a aplicação com jogos e o grupo que usou a aplicação sem jogos. De notar que, o **P** representa a proporção, o **IC** representa o intervalo de confiança e o **Pc – Ps** (Proporção com jogos – Proporção sem jogos) representa a diferença de proporções entre aplicações com jogos e sem jogos, para 95% de intervalo de confiança.

De uma forma geral, destaca-se de forma positiva a utilização da aplicação com o módulo de jogos integrado. As questões feitas tentam perceber a recetividade deste tipo de aplicações junto do público-alvo, tendo em consideração se as pessoas usam a aplicação para ler informação sobre os locais, se visitam esses locais fisicamente através da aplicação e se realmente consideram que esta aplicação contribui para o seu aumento de conhecimento sobre a cidade onde se encontram. No entanto, é importante salientar que, para resultados mais credíveis o número de inquiridos deveria ser maior, para obter valores conclusivos em todas as questões.

Na Figura 48 (*Quantos locais visitou na aplicação e leu efetivamente detalhes sobre o mesmo?*), observa-se que, com o módulo de jogos, as pessoas têm tendência a ler mais informação sobre os diversos locais: entre 6 a 8 locais com 50,00% a contrastar com os 21,9% que usaram a aplicação sem o módulo de jogos. Se observamos a diferença de proporções para esta questão obtemos valores inconclusivos para as duas primeiras opções, porque o 0 pertence a ambos os intervalos obtidos. No entanto, a proporção (%) de pessoas que visitaram entre 6 a 8 locais na aplicação com o módulo de jogos integrado é significativamente superior à mesma proporção (%) de pessoas na utilização da aplicação sem o módulo de jogos, ou seja, $P_c > P_s$. Destaca-se assim uma tendência para visitar e visualizar mais locais disponíveis na aplicação.

Tabela 10 – Resultados obtidos às questões comuns a ambos os questionários

1 - Quantos locais visitou na aplicação e leu efetivamente detalhes sobre o mesmo?						
	0 - 2		3 - 5		6 - 8	
	P	IC	P	IC	P	IC
Com jogos	21,9%	7,57% a 36,2%	28,1%	12,53% a 43,67%	0,50%	32,67% a 67,32%
Sem jogos	31,3%	15,2% a 47,4%	46,9%	29,61% a 64,19%	21,9%	7,57% a 36,23%
Pc - Ps	0% a 12,13%		0% a 42,01%		5% a 50,58%	
2 - Quantos locais visitou fisicamente por recomendação da aplicação?						
	0 - 2		3 - 5		6 - 8	
	P	IC	P	IC	P	IC
Com jogos	34,4%	11,12% a 57,67%	65,6%	42,32% a 88,88%	0%	0%
Sem jogos	87,5%	71,30% a 103,7%	12,5%	0% a 28,71%	0%	0%
Pc - Ps	-81,46% a -24,74%		24,74% a 81,46%		0%	
3 - Se viajasse e o hotel onde ficasse tivesse uma aplicação deste género, usá-la-ia para visitar a cidade?						
	Não		Provavelmente		Sim	
	P	IC	P	IC	P	IC
Com jogos	3,1%	0% a 9,11%	31,3%	15,23% a 47,37%	65,6%	49,14% a 82,06%
Sem jogos	15,6%	3,0% a 28,17%	65,6%	39,11% a 73,49%	28,1%	12,53% a 43,67%
Pc - Ps	0% a 1,43%		-48,53% a -1,43%		14,84% a 60,16%	
4- Considera que a aplicação contribui para o seu aumento de conhecimento sobre a cidade?						
	Pouco		Satisfatório		Bastante	
	P	IC	P	IC	P	IC
Com jogos	0%	0%	31,3%	15,23% a 47,37%	68,8%	52,74% a 84,85%
Sem jogos	28,1%	12,53% a 43,67%	0,50%	32,68% a 67,32%	21,9%	7,57% a 36,23%
Pc - Ps	-43,67% a -12,53%		0% a 4,93%		25,38% a 36,23%	

Na Figura 49 (*Quantos locais visitou fisicamente por recomendação da aplicação?*), destaca-se o aumento na visita física de mais locais em relação ao módulo sem jogos. É importante, porque no módulo sem jogos os valores indicam que 87,5% dos utilizadores visitaram entre 0 a 2 locais, ou seja, grande parte deles pode não ter visitado nenhum local. A diferença de proporções

permite evidenciar os resultados observados. Para a opção de 0 a 2 locais visitados, a diferença de proporções obtidas indica que, para a aplicação sem jogos, a proporção (%) sem jogos é significativamente maior (-81,46% a -24,74%), ou seja, $P_s > P_c$. Para a opção de 3 a 5 locais visitados fisicamente, os resultados da diferença de proporções invertem-se positivamente para o módulo com jogos. Ou seja, a proporção (%) de pessoas que visitaram 3 a 5 locais por recomendação da aplicação com jogos é significativamente superior à mesma proporção de pessoas para a aplicação sem jogos (24,74% a 81,46%, logo $P_c > P_s$).

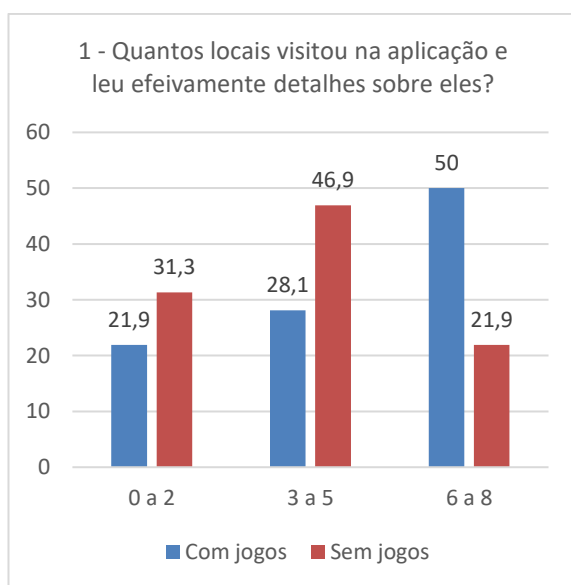


Figura 48 – Respostas obtidas à questão “Quantos locais visitou na aplicação e leu efetivamente detalhes sobre o mesmo?”

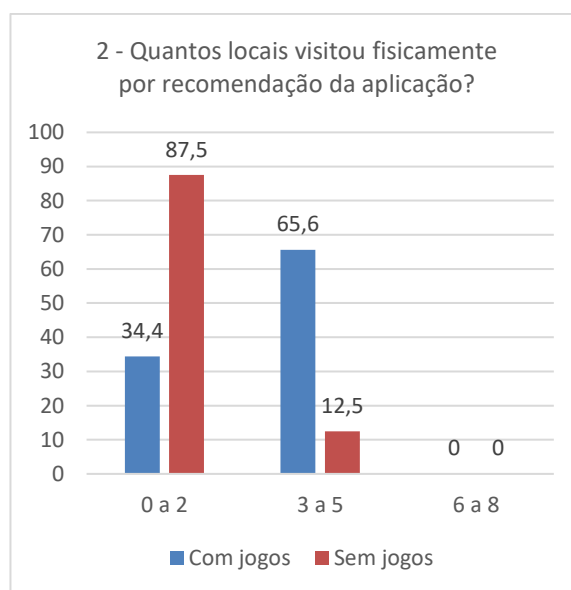


Figura 49 – Respostas obtidas à questão “Quantos locais visitou fisicamente através da aplicação?”

Na Figura 50 (*Se viajasse e o hotel onde ficasse tivesse uma aplicação deste género, usá-la-ia para visitar a cidade?*), em termos de diferença de proporções, os dados são inconclusivos para a opção primeira opção. Observando os resultados obtidos para as restantes opções, conclui-se que, na aplicação sem jogos, as pessoas consideravam como Provável a possibilidade de usar uma aplicação deste género para visitar a cidade (-48,53% a -1,43%, logo $P_s > P_c$). No entanto, para a opção Sim a diferença de proporções é significativamente superior (14,84% a 60,16%, logo $P_c > P_s$) para a aplicação com o módulo de jogos integrado, em relação à aplicação sem o mesmo módulo. Assim sendo, se na aplicação sem o módulo de jogos integrado as pessoas consideravam como provável o uso desta aplicação, na aplicação com o módulo de jogos as pessoas têm mais certeza do uso da mesma.

Por último, a questão da Figura 51 (*Considera que a aplicação contribui para o seu aumento de conhecimento sobre a cidade?*) confronta os utilizadores com uma pergunta chave para esta dissertação e os resultados obtidos são positivos.

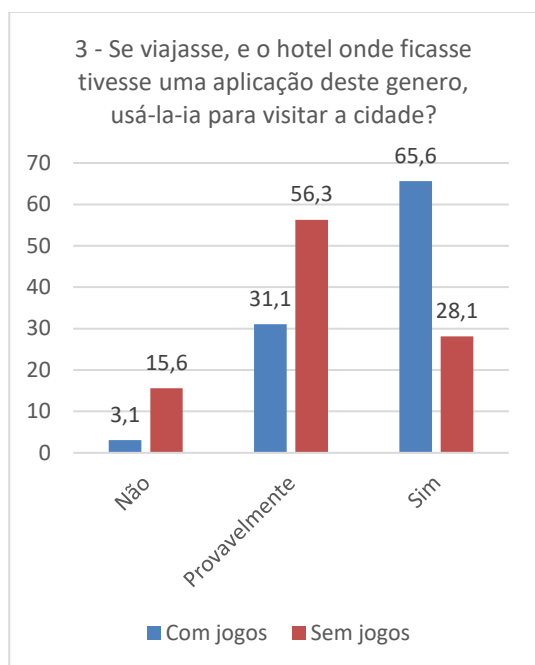


Figura 50 – Respostas à questão “Se viajasse e o hotel onde ficasse tivesse uma aplicação deste género, usá-la-ia para visitar a cidade?”

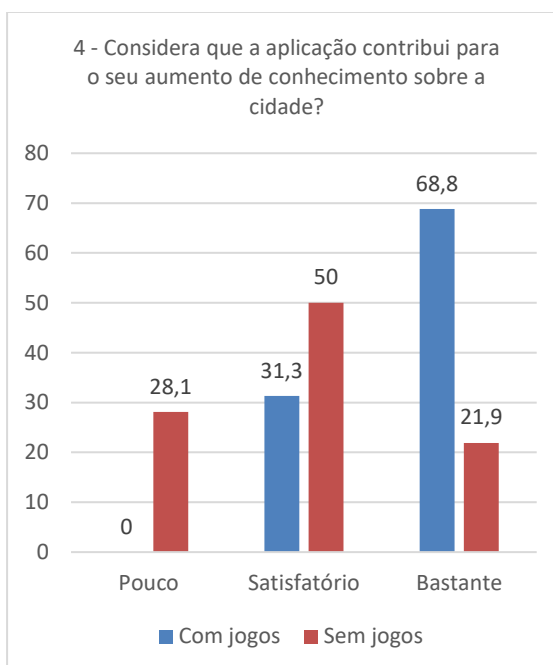


Figura 51 – Resposta à questão “Considera que a aplicação contribui para o seu aumento de conhecimento sobre a cidade?”

No módulo sem jogos, 28,1% dos utilizadores assinalaram como Pouco, considerando que esta aplicação pode não ser uma contribuição para o aumento de conhecimento. Por outro lado, no módulo com jogos, nenhum utilizador assinalou Pouco como resposta. Neste caso, a grande maioria dos inquiridos (68,8%) escolheram a resposta Bastante, em contraste com os 21,9% no módulo sem jogos. Analisando a diferença de proporções, para a opção Pouco os resultados evidenciam proporções (%) significativamente maiores em aplicações sem jogos e, para a opção Bastante, os resultados indicam proporções (%) significativamente maiores em aplicações com jogos. No entanto, a opção Satisfatório é dada como inconclusiva. Ou seja, para a opção intermédia não é possível obter diferenças significativas por nenhuma das aplicações.

6.3.2 Questões exclusivas ao módulo de jogos

Da Figura 52 à Figura 55 é possível observar as questões exclusivas ao módulo de jogos. Este conjunto de questões pretende perceber se os inquiridos gostam deste tipo de abordagem e se consideram que pode realmente contribuir para o conhecimento da cidade.

Das questões colocadas destaca-se o facto de 96,9% dos utilizadores gostarem deste tipo de atividades (Figura 52), no entanto, o mesmo conjunto de utilizadores não considera que o levem a procurar saber mais pela cidade (Figura 53). Ainda assim, 75,0% (Figura 53) dos inquiridos consideram que os pode ajudar a saber mais sobre determinados locais da cidade onde estão alojados.

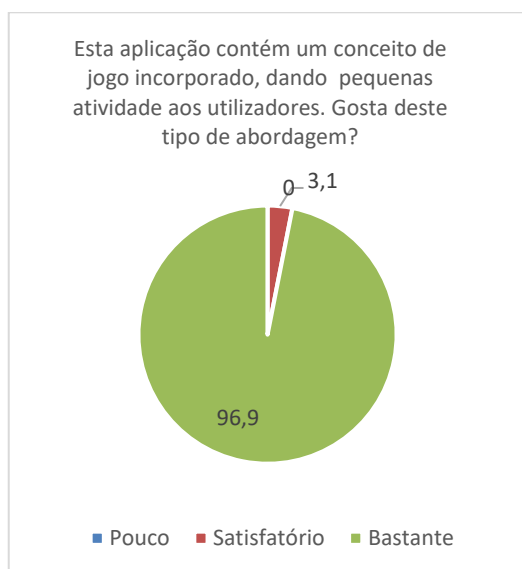


Figura 52 – Resposta à questão “Esta aplicação contém um conceito de jogo incorporado, dando pequenas atividades aos utilizadores. Gosta deste tipo de abordagem?”

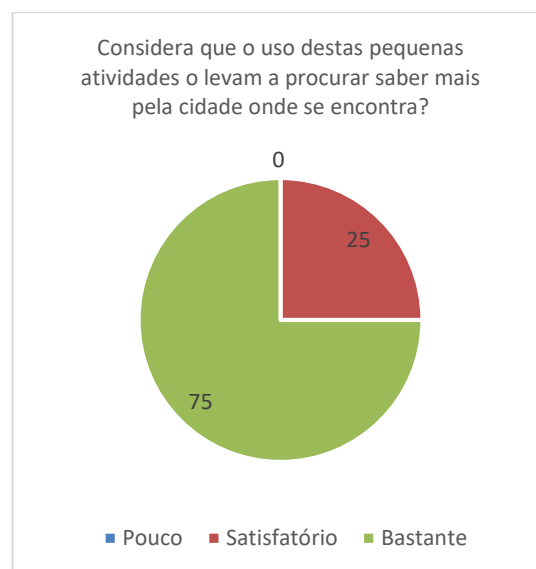


Figura 53 – Resposta à questão “Considera que o uso destas pequenas atividades o levam a procurar saber mais pela cidade onde se encontra?”

A pergunta da Figura 54, que questiona sobre a dificuldade de entender o jogo, pode ser o ponto mais fraco a destacar desta análise. Apesar de 40,6% dos inquiridos considerar que teve pouca dificuldade, outros 40,6% já assinalaram como Satisfatório o nível de dificuldade e 18,8% sentiram bastante dificuldade, ou seja, o modo como o jogo foi organizado pode não ser o mais intuitivo.

Por último, das cinco atividades presentes na aplicação com o módulo com jogos destaca-se que a maior parte do público realizou entre duas a três atividades, dados observáveis da Figura 55. Isto pode significar que houve algum interesse por parte do público em realmente experimentar o jogo e o seu modo de funcionamento.

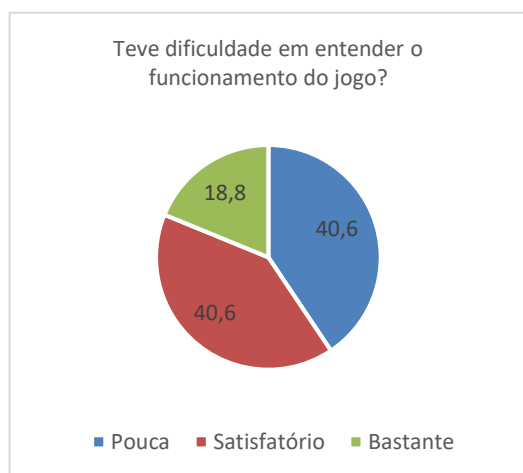


Figura 54 – Resposta à questão “Teve dificuldade em entender o funcionamento do jogo?”

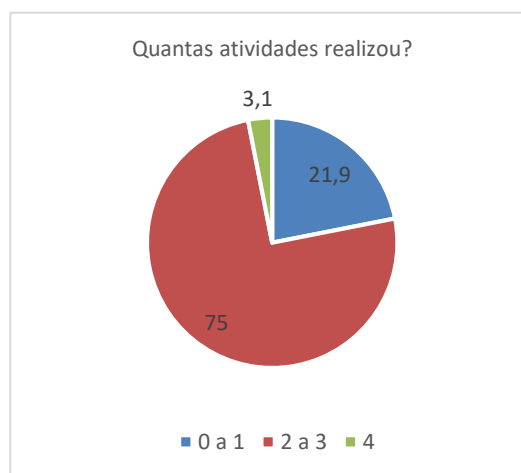


Figura 55 – Resposta à questão “Quantas atividades realizou?”

6.3.3 Conclusões obtidas

Após a análise dos dados recolhidos, torna-se importante relembrar as duas hipóteses que estão a ser avaliadas por forma a tirar uma conclusão. Para a hipótese *Os utilizadores gostam deste tipo de abordagem?*, os resultados obtidos permitem responder positivamente. Os valores obtidos à terceira questão do questionário comum (Se viajasse e o hotel onde ficasse tivesse uma aplicação deste género, usá-la-ia para visitar a cidade?), bem como os resultados obtidos na primeira questão do questionário apenas para o módulo com jogos (*Esta aplicação contém um conceito de jogo incorporado, oferecendo pequenas atividades aos utilizadores. Gosta deste tipo de abordagem?*) permitem indicar que os inquiridos são recetivos a este tipo de abordagem. Do mesmo modo, em relação à hipótese *Usando jogos sérios, os utilizadores procuraram obter um maior conhecimento da cidade onde se encontram?*, os valores obtidos permitem responder positivamente a esta questão. As restantes questões de ambos os questionários suportam a validade desta abordagem para ajudar os utilizadores a obter um maior conhecimento da cidade. No entanto, é importante referir que uma amostra de dados

maior seria importante para uma maior credibilidade dos dados, visto que em algumas opções das questões os resultados foram inconclusivos.

6.4 Sumário

Neste capítulo foi definida a abordagem metodológica seguida para a avaliação do módulo a desenvolver, tendo em atenção quais as grandezas e as hipóteses que é necessário avaliar para que este módulo resolva o problema a que se propõe. Foram também apresentados os resultados obtidos e as conclusões alcançadas, tendo em conta as hipóteses colocadas.

7 Conclusão e Trabalho Futuro

Esta dissertação tem como principal objetivo a elaboração de um módulo de “proximity based serious games” que permita a construção de jogos sérios entre os hóspedes alojados num mesmo local e que possibilite a sua execução quer em ambientes internos, quer em externos, interagindo com o espaço envolvente.

A solução final apresentada é suportada por um período de pesquisa e análise relevante. A elaboração da análise de valor permitiu situar este módulo num contexto real de negócio, tendo sido apresentado o seu potencial para os clientes alvo (hotéis, *hostels* e apartamentos) e respetivos hóspedes. Por outro lado, o levantamento do Estado da Arte sobre o tema Jogos Sérios, mais especificamente em relação a património, cultura e história, foi importante para perceber o que se encontra atualmente feito no que diz respeito a este tipo de jogos. Foi igualmente realizada uma pesquisa de possíveis Tecnologias Móveis que podiam ser usadas no contexto do turismo móvel, de modo a analisar quais as que melhor se adaptariam à resolução do problema proposto. Desta pesquisa bibliográfica percebeu-se que tipo categorias de jogos podem ser usados em Jogos Sérios desta área, bem como que características estes podem ter para se tornarem um jogo de sucesso. Concluiu-se também que o GPS, os *Beacons* e os códigos QR como as tecnologias móveis mais adequadas à resolução do problema proposto. Isto permitiu desenvolver o conceito do jogo e realizar o levantamento dos requisitos. Por fim, procedeu-se à implementação da solução proposta, o que levou a implementações no Backoffice, nos serviços para a aplicação móvel e na própria aplicação móvel.

O principal objetivo foi assim cumprido. O Backoffice suporta a criação de jogos (ações) que, através da aplicação móvel, permitem aos hóspedes interagir com a cidade onde se encontram quer em ambientes internos, quer em ambientes externos. O hóspede consegue ainda visualizar a sua posição em relação aos outros utilizadores e as ações que já realizou. Consegue ainda visualizar os locais com ações e a respetiva forma de as ativar. Por outro lado, o estabelecimento hoteleiro consegue observar o posicionamento dos seus hóspedes no jogo. No entanto, o requisito que indicava que a aplicação deveria permitir a comunicação com o hóspede promovendo os serviços, eventos e atividades do estabelecimento hoteleiro não foi implementado. Este seria desenvolvido através das campanhas, por forma a ser possível articular essa divulgação de informação do hotel com os jogos/atividades realizados. Destaca-se apenas o seu desenvolvimento a nível do Backoffice e parte dos serviços necessários.

Mediante a avaliação da solução desenvolvida, conclui-se que este tipo de abordagem pode ser bem aceite pela população alvo. Torna-se importante referir que uma amostra de resultados maior seria importante para uma evidência estatística mais sólida, no entanto, dentro da amostra conseguida os resultados são positivos. Os questionários realizados comprovam que o uso destas pequenas atividades leva as pessoas a procurar em saber mais, a ler mais sobre os locais e a visitar mais a cultura, património e história de uma dada cidade. Claro que o sucesso deste tipo de abordagem depende também da originalidade/criatividade das diversas atividades que possam ser desencadeadas. Ainda em relação aos resultados obtidos destaca-se que uma parte da amostra que usou a aplicação com o módulo de jogos considera que estes não são intuitivos. Nesse sentido, este será um dos pontos a melhorar no futuro.

No que diz respeito ao trabalho futuro, há muitos aspetos que podem ajudar a melhorar o conceito do jogo. Começando pelos desenvolvimentos que ficaram por caracterizar, ou seja, a implementação das Campanhas. Dando um espaço ao estabelecimento hoteleiro para divulgar os seus serviços, eventos ou atividades, através da atribuição de pequenos prémios, contribuirá não só para que as pessoas joguem mais, visitem mais locais e possam posteriormente investir mais no hotel onde se encontram. Em relação ao modo do jogo em si, há aspetos que podem ser melhorados. Por exemplo, o utilizador poderia desistir de uma ação enquanto ela está a decorrer, algo que hoje não pode fazer. Se tal acontecesse, eram atribuídos 0 pontos e essa ação era dada como realizada. Além disso, seria necessário melhorar a forma como as ações chegam ao utilizador. Este ponto pode ser conseguido se houver uma breve explicação das regras do jogo no telemóvel. Poderia também existir uma lista de “ações prontas para ativar”

que estavam ordenadas por proximidade e que não dependiam apenas da atual notificação. As ações apareciam/desapareciam da lista consoante fosse possível ou não ativá-las. Outro aspeto a melhorar seria a inserção de novos tipos de ações que pudessem cativar o hóspede a realizar mais atividades.

Todo o processo de análise, desde o tipo de tecnologias que poderiam ser usadas, até à idealização do conceito do jogo foi muito interessante. Tal como referido, há muitas melhorias a realizar, mas considera-se que o principal objetivo foi atingido e que foi criado um conceito interessante exequível num contexto real de negócio.

Referências

- [1] World Tourism Organization (UNWTO) and Business School, Technology in tourism, 2011.
- [2] Statista, “Leading mobile device uses among travelers worldwide as of October 2014,” [Online]. Available: <http://www.statista.com/statistics/368307/leading-mobile-device-uses-among-travelers-worldwide/>. [Acedido em Janeiro 2016].
- [3] WeDeliverAwesomeApps, “GuestU - Mobile app for your hospitality business,” [Online]. Available: <http://guestu.com/>.
- [4] WeDeliverAwesomeApps, “GuestU,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.xtourmaker.guestu>. [Acedido em Dezembro 2016].
- [5] Organização Mundial de Turismo, “Why Tourism?,” 2015. [Online]. Available: <http://www2.unwto.org/content/why-tourism>. [Acedido em Dezembro 2015].
- [6] Organização Mundial de Turismo, “Tourism Highlights,” 2015.
- [7] Think Digital & Bournemouth University, “Technonoloy enhanced tourism experience,” 2013.
- [8] WeDeliverAwesomeApps, “GuestU - Your mobile concierge,” [Online]. Available: <http://app.guestu.com/login.aspx>. [Acedido em Janeiro 2016].
- [9] Google, “Android,” [Online]. Available: <https://www.android.com/>. [Acedido em Janeiro 2016].
- [10] Apple, “The world’s most advanced mobile OS~,” [Online]. Available: <http://www.apple.com/ios/what-is/>. [Acedido em Janeiro 2016].
- [11] eMarketer, “For Hotels, Mobile Offers a Plethora of Options,” 2013. [Online]. Available: <http://www.emarketer.com/Article/Hotels-Mobile-Offers-Plethora-of-Options/1010181>. [Acedido em Dezembro 2015].
- [12] OECD, The impact of Culture on Tourism, 2009.
- [13] Microsoft, “Asp.Net,” [Online]. Available: <http://www.asp.net/>. [Acedido em Janeiro 2016].

- [14] Microsoft, "Microsoft® SQL Server® 2012 Express," [Online]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=29062>. [Acedido em Janeiro 2016].
- [15] G. Watson, "OrmLite - Lightweight Java ORM Supports Android and SQLite," [Online]. Available: http://ormlite.com/sqlite_java_android_orm.shtml. [Acedido em Janeiro 2016].
- [16] S. Nicola, "Value Analysis," [Online]. Available: https://moodle.isep.ipp.pt/pluginfile.php/91647/mod_resource/content/2/An%C3%A1lise_Valor_Aula1.pdf. [Acedido em Janeiro 2016].
- [17] S. Nicola, "Negociation," [Online]. Available: https://moodle.isep.ipp.pt/pluginfile.php/92054/mod_resource/content/3/An%C3%A1lise_Valor_Aula3.pdf. [Acedido em Janeiro 2016].
- [18] S. Nicola, E. P. Ferreira e J. J. P. Ferreira, "A Quantitative Model for Decomposing & Assessing the Value for the Customer," *Journal of Innovation Management*, vol. 2, pp. 104-138, 2014.
- [19] D. Gouveia, D. Lopes e C. de Carvalho, "Serious gaming for experiential learning," em *Frontiers in Education Conference (FIE)*, 2011.
- [20] E. F. Anderson, L. McLoughlin, F. Liarokapis, C. Peters, P. Petridis e S. de Freitas, "Serious Games in Cultural Heritage," em *The 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*, 2009.
- [21] D. Micheal e S. Chen, "Serious Games Defined," em *Serious Games: Games that Educate, Train and Inform*, 2006, p. 17.
- [22] C. V. de Carvalho, P. M. Latorre Andrés e F. J. Arbeloa, "Serious Games Network," *Virtual Archaeology Review*, vol. Vol 4, 2013.
- [23] F. Laamarti, M. Eid e A. El, "An Overview of Serious Games," em *International Journal of Computer Games Technology*, 2014.
- [24] D. Djaouti, J. Alvarez, J.-P. Jessel e O. Rampnoux, "Origins of serious games," em *Serious Games and Edutainment Applications*, Springer London, 2011, pp. 25-43.
- [25] M. Mortaraa, C. E. Catalan, F. Bellottib, G. Fiuccic, M. Houry-Panchettid e P. Petridise, "Learning cultural heritage by serious games," *Journal of Cultural Heritage*, vol. 15 (n° 3), pp. 318-325, 2014.

- [26] Istituto per le Tecnologie Didattiche - Palermo, "O'Munaciedd," [Online]. Available: <http://munaciedd.pa.itd.cnr.it/>. [Acedido em Dezembro 2015].
- [27] D. L. Guardia, M. Arrigo e O. Di Giusepp, "A Location-Based Serious Game to Learn About the Culture," em *International Conference: The Future of Education*.
- [28] INESC TEC, "Travel Plot - Can you find the treasure?," [Online]. Available: <http://www.travelplot.com/en/>. [Acedido em Janeiro 2016].
- [29] S. . A. M. Ferreira, "Location based transmedia storytelling: enhancing the tourism experience," 2015.
- [30] SELEAG, "TimeMesh," [Online]. Available: <http://www.timemesh.eu/?lang=pt>. [Acedido em Dezembro 2015].
- [31] D. Gouveia, D. Lopes, V. de Carvalho e R. Batista, "TimeMesh: An Educational Historical Game," Fourth IEEE International Conference On Digital Game And Intelligent Toy Enhanced Learning, 2012.
- [32] J. Neto, R. Silva, J. Neto, J. Pereira e J. Fernandes, "Solis'Curse - A Cultural Heritage Game Using Voice Interaction with a Virtual Agent," em *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2011 Third International Conference on* , 2011.
- [33] S. George e A. Serna , "Introducing Mobility in Serious Games: Enhancing Situated and Collaborative Learning," em *Proceedings of the 14th international conference on Human-computer interaction: users and applications* , 2011.
- [34] P. J. Benckendorff, P. J. Sheldon e D. R. Fesenmaier, "Mobilities and Information Technology," em *Tourism Information Technology, 2nd Edition*, Cabi Tourism Texts, 2014.
- [35] GSMA, *A Guide to Bluetooth Beacons*, 2014.
- [36] J. A. P. Montañés, A. Mendoza e I. S. Prieto, "Smart Indoor Positioning/Location and Navigation: A Lightweight Approach," *International Journal of Artificial Intelligence and Interactive Multimedia*, vol. 2, 2013.
- [37] GSMA, "A Guide to Bluetooth Beacons," 2014.
- [38] AbiResearch, "ABIResearch: Intellegence for Inovators," [Online]. Available: <https://www.abiresearch.com/>. [Acedido em Janeiro 2016].

- [39] AbiResearch, “BLE Beacon Shipments Break 400 Million in 2020,” Julho 2015. [Online]. Available: <https://www.abiresearch.com/press/ble-beacon-shipments-break-400-million-in-2020/>. [Acedido em Janeiro 2016].
- [40] Beekn, “Guide to iBeacon Hardware,” [Online]. Available: <http://beekn.net/guide-to-ibeacons/>. [Acedido em Dezembro 2015].
- [41] GSMA, “GSMArena.com - The ultimate resource for GSM handset information.,” [Online]. Available: <http://www.gsmaarena.com/>. [Acedido em Dezembro 2015].
- [42] RadiusNetworks, “We Make Proximity Work — Radius Networks,” [Online]. Available: <http://www.radiusnetworks.com/>. [Acedido em Dezembro 2015].
- [43] Estimote, “Estimote Real-world context for your apps Stick our tiny beacons in any location or to any object to create new, contextually rich mobile experiences.,” [Online]. Available: <http://estimote.com/>. [Acedido em Dezembro 2015].
- [44] StickNFind , “StickNFind - Home of the StickNFind,” [Online]. Available: <https://www.sticknfind.com/>. [Acedido em Dezembro 2015].
- [45] Apple, “iBeacon for Developers,” [Online]. Available: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>. [Acedido em Dezembro 2015].
- [46] Google, “Introducing Eddystone, The New Beacon Format From Google,” [Online]. Available: <http://developer.radiusnetworks.com/2015/07/14/introducing-eddystone.html>. [Acedido em Dezembro 2015].
- [47] Estimote, “What is Eddystone™?,” [Online]. Available: <http://developer.estimote.com/eddystone/>. [Acedido em Janeiro 2016].
- [48] Google , “Specification for Eddystone, an open beacon format from Google,” [Online]. Available: <https://github.com/google/eddystone>. [Acedido em Janeiro 2016].
- [49] NFC World, “NFC phones: The definitive list,” [Online]. Available: <http://www.nfcworld.com/nfc-phones-list/>. [Acedido em Janeiro 2016].
- [50] Lighthouse, “Indoor location technologies compared,” [Online]. Available: <http://lighthouse.io/indoor-location-technologies-compared/> . [Acedido em Janeiro 2016].
- [51] Infsoft, “Indoor navigation using WiFi as a positioning technology - See more at: <http://www.infsoft.com/blog/2015/indoor-navigation-using-wifi-as-a-positioning->

- technology#sthash.la61wZ5i.dpuf,” [Online]. Available: <http://www.infsoft.com/blog/2015/indoor-navigation-using-wifi-as-a-positioning-technology>. [Acedido em Janeiro 2016].
- [52] Estimote, “Dev Kit with 3 Bluetooth Beacons,” [Online]. Available: <http://estimote.com/#jump-to-products>. [Acedido em Janeiro 2016].
- [53] Blue Cats, “AA Beacon,” [Online]. Available: <http://store.bluecats.com/collections/bluecats-hardware/products/aa-beacon-3>. [Acedido em Janeiro 2016].
- [54] Gimbal, “Gimbal Proximity Beacon Series 21,” [Online]. Available: <https://store.gimbal.com/collections/beacons/products/s21>. [Acedido em Janeiro 2016].
- [55] BlueSense Networks, “BlueBar Beacon Weatherproof,” [Online]. Available: <http://bluesensenetworks.com/product/bluebar-beacon-weatherproof/>. [Acedido em Janeiro 2016].
- [56] ISEP, “REQUISITOS, ANÁLISE E DESIGN,” 2009-2010. [Online]. Available: https://moodleant.isep.ipp.pt/file.php/233028/Conteudos/EAPLI_2009-2010_Requisitos_Analise_e_Design_T_.pdf. [Acedido em Agosto 2016].
- [57] P. Sousa, “Padrões de Aplicações Empresariais,” [Online]. Available: https://moodleant.isep.ipp.pt/file.php/234890/Padrao_TM_TDGV2.pdf. [Acedido em Agosto 2016].
- [58] Hwaci, “SQLite,” [Online]. Available: <https://www.sqlite.org/>. [Acedido em Fevereiro 2016].
- [59] j. hellemons, “Qr Code library 1.3.0,” [Online]. Available: <https://www.nuget.org/packages/MessagingToolkit.QRCode/>. [Acedido em Junho 2016].
- [60] .NET Foundation, “Nuget,” [Online]. Available: <https://www.nuget.org/packages>. [Acedido em Agosto 2016].
- [61] Newtonsoft , “JSON.Net - Popular high-performance JSON framework for .NET,” [Online]. Available: <http://www.newtonsoft.com/json>. [Acedido em Agosto 2016].
- [62] Google, “Transmitting Network Data Using Volley,” [Online]. Available: <https://developer.android.com/training/volley/index.html>. [Acedido em Agosto 2016].

- [63] <https://github.com/inder123>, "GSON," [Online]. Available: <https://github.com/google/gson>. [Acedido em Agosto 2016].
- [64] S. Kawano, "Bolts," [Online]. Available: <https://github.com/BoltsFramework/Bolts-Android>. [Acedido em Agosto 2016].
- [65] Estimote, "Beacon Tech Overview," [Online]. Available: <http://developer.estimote.com/>. [Acedido em Agosto 2016].
- [66] D. Maguluru, "Barcode Scanner," [Online]. Available: <https://github.com/dm77/barcodescanner>. [Acedido em Agosto 2016].
- [67] greenrobot.org, "EventBus," [Online]. Available: <https://github.com/greenrobot/EventBus>. [Acedido em Agosto 2016].
- [68] Google, "Google Analytics," [Online]. Available: <https://www.google.com/analytics/>. [Acedido em Janeiro 2016].

Anexos

1. Plataforma GuestU

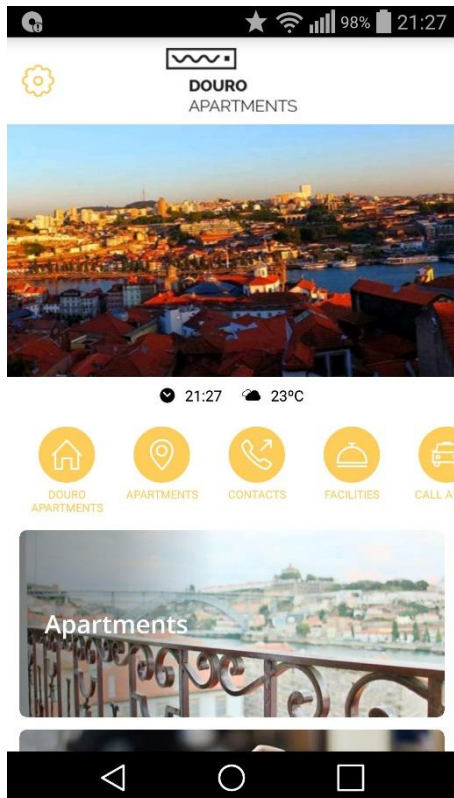


Figura 56 – Ecrã principal da aplicação

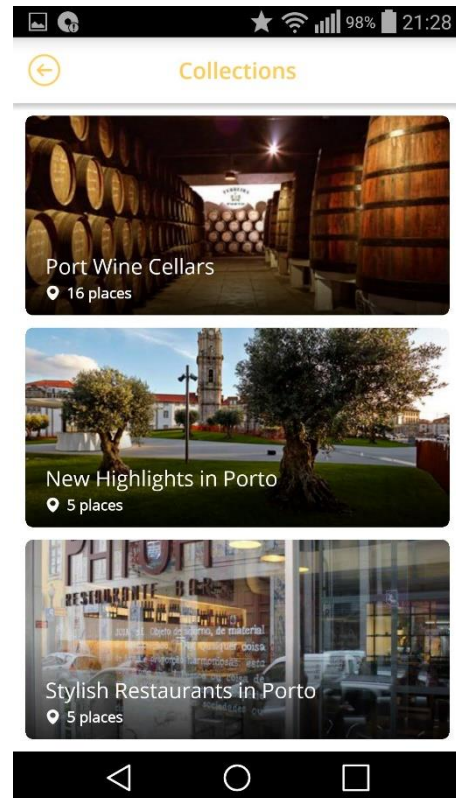


Figura 57 – Lista de rotas

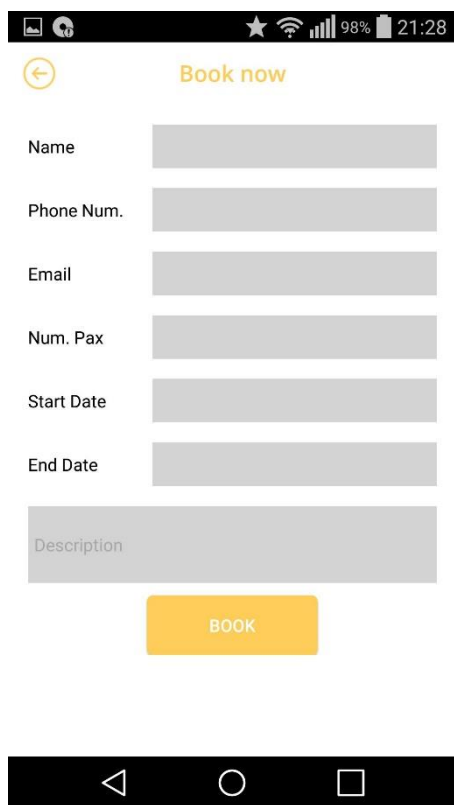


Figura 58 – Ecrã para efetuar reservas

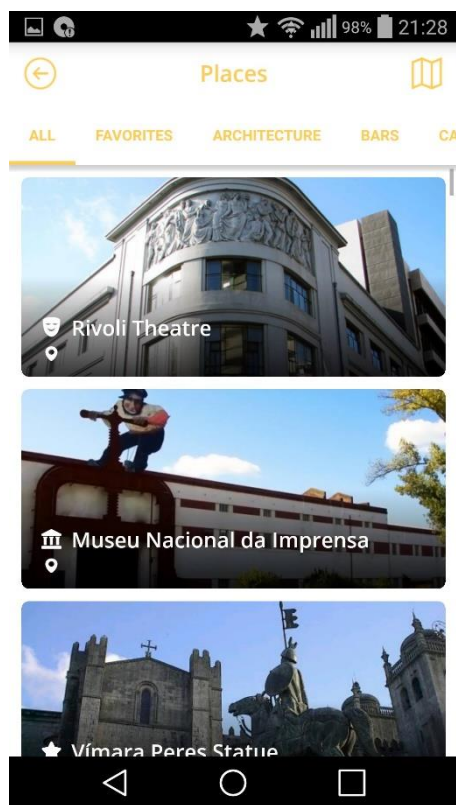


Figura 59 – Lista de pontos de interesse