

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



O turismo inteligente: validação e análise da deslocação de turistas

Bruno Henrique Batista Cruz da Silva

Mestrado em Engenharia Informática
Especialização em Engenharia de Software

Versão Pública

Trabalho de Projeto orientado por:
Prof.^a Doutora Andreia Filipa Torcato Mordido

2019

Resumo

No âmbito da tese de Mestrado em Engenharia Informática, foi escrito este relatório final para descrever os principais objetivos do projeto, as tarefas que foram realizadas, o plano de trabalhos que foi cumprido, assim como, uma análise às tarefas que ficaram por cumprir, às limitações encontradas e ao trabalho futuro.

Com o crescimento contínuo do turismo, a indústria do turismo procura formas para que o movimento dos turistas numa cidade não fique dependente de atrasos nos transportes, isto para ajudar os turistas a manterem-se em movimento e aumentarem a sua satisfação, sendo esta a principal motivação deste trabalho.

O movimento de turistas numa cidade segue padrões determinados pelos pontos de interesse a visitar, ao contrário dos habitantes cujas deslocações são essencialmente determinadas pelo local de trabalho e de residência. Por esta razão, o movimento dos turistas é mais variável e é importante perceber estas alterações em tempo útil.

Para cumprir o que se propõe, neste trabalho pretende-se desenvolver um conjunto de otimizações e novas funcionalidades, a integrar numa aplicação móvel responsável pela venda de reservas para transportes turísticos. As reservas passarão a ser validadas com auxílio da tecnologia QR Code, que oferecerá uma maior eficiência e controlo na validação das mesmas. Para além disto será desenvolvida uma aplicação destinada aos turistas, que tira partido da tecnologia GPS e BLE e torna, assim, a utilização dos transportes turísticos mais satisfatória.

Com base em *inputs* operacionais, nomeadamente a localização das vendas das reservas e a localização dos turistas na proximidade das paragens, iremos oferecer uma camada que tenha uma visão global da presença/procura de turistas numa cidade, recorrendo a *business intelligence* (BI) e *multi-sensing*.

Palavras-chave: turistas, mobilidade, informação de localização, QR Code, GPS, BLE, *business intelligence*, *multi-sensing*

Abstract

In the scope of the Master thesis in Informatics Engineering, this final report was written to describe the main objectives of the project, the tasks that were carried out, the work plan that was accomplished, as well as an analysis of the tasks that remained unfulfilled, the limitations encountered and future work.

As tourism continues to grow, the tourism industry is looking for ways in which the movement of tourists in a city is not affected by transport delays. The main motivation of this work is to help tourists keep moving and to increase their satisfaction.

The movement of tourists in a city follows patterns determined by the points of interest to visit, unlike the inhabitants whose journeys are essentially determined by the place of work and residence. For that reason, the movement of tourists is more variable, and it is important to realize these changes in a timely manner.

To fulfil what is proposed, this work intends to develop a set of optimizations and new features, to be integrated in a mobile application responsible for the sale of reservations for tourist transport. Reservations are now validated using the QR Code technology, which offers greater efficiency and control in their validation. In addition, an application for tourists takes advantage of GPS and BLE to ensure movement of tourists is done smoothly and, thus, making tourist transport more satisfying.

Based on operational inputs, specifically the location of reservation sales and the presence of tourists near the stops, we offer a layer that has an overall view of tourist presence/demand in a city, using business intelligence and multi-sensing.

Keywords: tourists, mobility, information of location, QR Code, GPS, BLE, business intelligence, multi-sensing

Conteúdo

Lista de Figuras	viii
Lista de siglas e acrónimos.....	x
Capítulo 1 Introdução	2
Principal motivação	3
Objetivos	3
Instituição de acolhimento	3
Organização do documento	4
Capítulo 2 Trabalho relacionado	6
2.1 Desenvolvimento de aplicações móveis e aplicações web	6
2.2 Protocolos de integração, comunicação e segurança.....	9
2.3 Análise e armazenamento de dados	10
2.4 Tecnologias.....	13
2.5 Casos de estudo.....	19
Capítulo 3 O turismo inteligente	24
3.1 Sistema existente	24
3.2 Alterações ao sistema e novas contribuições	27
Capítulo 4 Arquitetura	28
Capítulo 5 Desenvolvimento e Implementação	30
Capítulo 6 <i>Data Profiling</i>	32
Capítulo 7 Considerações finais	34
7.1 Síntese e objetivos alcançados.....	34
7.2 Limitações e trabalho futuro.....	35
Bibliografia	36

Lista de Figuras

Figura 2-1: Diagrama UML da implementação do MVC em Android. Retirado de [26]	8
Figura 2-2: Especificação iBeacon. Retirado de [23]	17
Figura 2-3: Especificação AltBeacon. Retirado de [2]	18
Figura 2-4: Interface baseada em REST entre a aplicação Android (habilitada para BLE) e Microcontrolador Arduino. Retirado de [18]	21
Figura 2-5: Aplicação baseada em BLE e AltBeacon. Retirado de [18]	21
Figura 3-1: Menu da aplicação 4POS	24
Figura 3-2: Informação das reservas	25
Figura 3-3: Resumo de todas as reservas	26
Figura 3-4: Arquitetura do sistema	26

Lista de siglas e acrónimos

ACID: Atomicity, Consistency, Isolation, Durability.

BASE: Basically Available, Soft state, Eventual consistency.

BI: Business Intelligence.

BLE: Bluetooth Low Energy.

GPS: Global Positioning System.

HTTP: HyperText Transfer Protocol.

IIS: Internet Information Services.

IoT: Internet of Things.

NFC: Near Field Communication.

QR Code: Quick Responde Code.

REST: Representational State Transfer.

RFID. Radio Frequency IDentification

SOAP: Simple Object Access Protocol.

SQL: Structured Query Language.

TPA: Terminal de Pagamento Automático.

WSDL: Web Service Description Language.

Capítulo 1

Introdução

O turismo em Portugal tem aumentado gradualmente de ano para ano, sendo a cidade de Lisboa a segunda cidade europeia com maior crescimento no turismo¹. A capital portuguesa registou um aumento de 10,6% entre 2009 e 2017.

Em 2018, nos prémios atribuídos pelos *World Travel Awards*, Portugal foi considerado o “Melhor Destino do Mundo” e Lisboa a “Melhor Cidade Destino”², posto isto é expectável que o turismo em Portugal, nomeadamente em Lisboa, no mínimo mantenha os valores registados nos últimos anos.

Não é apenas o turismo que está em constante crescimento, a utilização de *smartphones* tornou-se indispensável para a vida das pessoas. Tendo em conta o estudo levado a cabo por Wee-Kheng Tan em [42], o *smartphone* foi rapidamente adotado como uma ferramenta de viagem turística, devido às suas capacidades avançadas de computação e omnipresença. Com um número crescente de utilizadores e uma ampla variedade de aplicações emergentes, o *smartphone* está a alterar substancialmente a forma de utilização e o entendimento das redes de transporte e das viagens turísticas.

O que tornou os *smartphones* em ferramentas importantes numa viagem turística é o facto de estes, tipicamente, incluírem mais de oito sensores que oferecem oportunidades de *context awareness* [15]. Além do microfone e da câmara, que se tornaram a base dos primeiros telemóveis, neste momento atributos como acelerómetros, giroscópios, GPS, bússolas digitais e sensores de proximidade tornaram-se características padrão em maior parte dos *smartphones*. Instalado em cada telemóvel como tecnologia discreta, o *software* é capaz de aceder a um único ou vários sensores a qualquer momento para suportar funções para aplicações específicas. Esses sensores permitem o desenvolvimento de sistemas adaptativos que operam efetivamente como um assistente pessoal móvel.

Assim, considerando a tecnologia já existente, dois exemplos que podem melhorar a experiência turística são:

- a utilização do *smartphone* com uma aplicação móvel permite que, tirando partido da câmara do *smartphone* e da tecnologia QR Code, se faça a validação de bilhetes, evitando assim que se tenha de andar com bilhetes em papel que cada vez mais se tornam incomodativos para os utilizadores;
- a identificação através do GPS e do Bluetooth em tempo real dos focos de presença de turistas que, ao contrário dos habitantes (cujas deslocações são essencialmente determinadas pelo local de trabalho e de residência), seguem padrões variados, como os pontos de interesse a visitar, com o objetivo de melhorar a rede de transportes também em tempo real.

Esta tese tem como principal objetivo a análise e a implementação de metodologias que incorporam as funcionalidades acima descritas em tecnologias já existentes. Por um lado, temos uma aplicação móvel a ser usada com o intuito de vender reservas para os transportes turísticos, a acrescentar a isto teremos a funcionalidade de validar essas reservas, para isso cada bilhete é identificado por um QR Code que serve como identificador e permitirá a sua validação de forma rápida e correta. Por outro lado, teremos uma aplicação móvel destinada aos turistas que permitirá aos turistas terem acesso às suas reservas de forma digital, mas também possibilitará que encontrem rapidamente as paragens para os transportes

¹ <https://www.jornaldenegocios.pt/empresas/turismo---lazer/detalhe/lisboa-e-a-segunda-cidade-europeia-com-maior-crescimento-no-turismo>

² <https://observador.pt/2018/12/01/oscares-do-turismo-portugal-e-o-melhor-destino-do-mundo-e-lisboa-a-melhor-cidade/>

turísticos. Paralelamente, será feita a análise à localização das vendas e ao número de turistas presentes nas paragens para que se adeque os serviços de transporte às necessidades atuais dos turistas.

Principal motivação

A indústria do turismo precisa de encontrar novas maneiras de adquirir clientes, vinculá-los e aumentar a satisfação desses clientes. Há uma elevada pressão competitiva e concorrência de preços – para os turistas os destinos de férias parecem ser facilmente permutáveis.

A tecnologia da informação transforma a posição estratégica das organizações, alterando a sua eficiência, diferenciação, custo operacional e tempo de resposta. Em particular, as novas tecnologias estimularam mudanças radicais na operação e distribuição da indústria do turismo [8]. Talvez o exemplo mais evidente no turismo é a reengenharia do processo de reservas, que gradualmente se torna racionalizado e permite que tanto os consumidores quanto a indústria economizem um tempo considerável na identificação, agregação, reserva e compra de produtos turísticos.

Como resultado, as empresas de turismo precisam de entender, incorporar e utilizar novas tecnologias, a fim de servir os seus mercados-alvo, de forma a melhorar a sua eficiência, maximizar a rentabilidade, melhorar os serviços e manter a prosperidade a longo prazo para si e para os destinos turísticos. O sucesso futuro das organizações de turismo e dos destinos serão determinados por uma combinação de gestão inovadora e *marketing*, intelecto e visão, bem como o uso estratégico de tecnologia avançada [8]. Posto isto, considerando o movimento de turistas em Lisboa já descrito anteriormente e a importância que o mesmo tem para um país como Portugal, este projeto torna-se bastante aliciante, uma vez que caso seja cumprido com sucesso pode fazer realmente a diferença na experiência que os turistas têm ao visitar Lisboa.

Objetivos

Um dos objetivos desta tese é introduzir no sistema turístico alguma tecnologia já existente de maneira a melhorar a forma como os turistas se deslocam, mais especificamente, pretende-se que a validação dos bilhetes turísticos passe a ser feita com a utilização da tecnologia QR Code, esta tecnologia permite que a aplicação, com o auxílio da câmara do *smartphone*, faça a leitura do QR Code correspondente a um bilhete, tornando a identificação dos bilhetes mais eficiente.

Além das diferentes integrações e melhorias propostas nesta tese, analisa-se também a possibilidade de utilizar o Power BI para oferecer uma camada que tenha uma visão global da presença/procura de turistas numa cidade, visão que pode permitir às empresas de turismo adaptarem os seus serviços de transporte de forma a satisfazer as necessidades dos turistas. Para tal, é tido como base *inputs* operacionais dos diversos sistemas de cada operador de serviços de mobilidade e da informação de localização, nomeadamente serão tidos em conta os *inputs* obtidos através da localização das vendas das reservas e o número de turistas presentes nas diversas paragens.

Instituição de acolhimento

Esta tese foi desenvolvida na Card4B³. Empresa com mais de 10 anos que nasceu após a identificação da necessidade de soluções para uma "nova cultura de mobilidade" nos centros urbanos, e está focada no fornecimento de componentes de *software* e serviços especializados de soluções integradas de

³ <https://www.card4b.pt/pt/about.html>

mobilidade e *city-services*, como transportes públicos, portagens, táxis, entre outros. A Card4B desenvolve e opera soluções integradas de mobilidade através de bilhética interoperável sem contato, informação ao passageiro, sistemas embebidos e *smartphones*, integração de sistemas e *business intelligence*. A abordagem é baseada no conceito de *Ticketing Kernel* para o diálogo entre todos os tipos de terminais e os mídias (por exemplo, cartões).

Organização do documento

Este documento encontra-se organizado em mais seis capítulos:

- o Capítulo 2 abrange principalmente o trabalho relacionado, com uma pequena descrição das tecnologias usadas e algumas aplicações que vão de encontro ao que este projeto pretende oferecer;
- no Capítulo 3 é descrito, de forma breve, o sistema existente, assim como a sua arquitetura. São ainda descritas as tarefas propostas a que nos propomos neste trabalho;
- o Capítulo 4 é composto pela informação relativa ao desenho do sistema, ou seja, é aqui que é exposto o desenho conceitual - “o que o sistema faz” - direcionado para o utilizador e o desenho técnico - “como o sistema faz” - direcionado para os programadores;
- o Capítulo 5 contém toda a informação importante para o desenvolvimento das tarefas propostas, nomeadamente é descrita a implementação das aplicações desenvolvidas;
- no Capítulo 6 é abordado o processo de analisar os dados disponíveis numa fonte de informação existente (por exemplo, uma base de dados) e gerar estatísticas ou resumos informativos sobre esses dados;
- no Capítulo 7 analisamos o trabalho realizado, descrevemos o trabalho que ficou por cumprir e discutimos o trabalho futuro.

Capítulo 2

Trabalho relacionado

Neste trabalho foram desenvolvidas duas aplicações móveis, para além disso existe uma aplicação servidora indispensável ao funcionamento do sistema. Neste capítulo iremos fazer uma breve descrição das ferramentas utilizadas para o desenvolvimento dessas aplicações, assim como algumas tecnologias utilizadas, por fim serão mencionados alguns trabalhos relacionados.

2.1 Desenvolvimento de aplicações móveis e aplicações web

Iremos fazer uma descrição das *frameworks* e das arquiteturas utilizadas para o desenvolvimento das aplicações móveis e aplicações web desta tese.

2.1.1 Abordagens de desenvolvimento de aplicações móveis

No desenvolvimento de aplicações móveis existem duas abordagens que se destacam, o desenvolvimento nativo e o desenvolvimento multiplataforma [36].

As aplicações desenvolvidas de forma nativa destinam-se a serem utilizadas em dispositivos específicos que suportam sistemas operativos, como *Android*, *Windows* ou *iOS*, e que pretendem tirar o máximo partido das suas características. Cada um destes sistemas operativos funciona de uma forma diferente e tem influência no desenvolvimento de aplicações, uma vez que a própria linguagem de programação pode ser diferente: *Java* para *Android*, *Objective C* para *iPhone* e *C++* para *Windows Mobile*. Assim, a mesma aplicação para *iOS* e *Android* traduz-se, na prática, em aplicações distintas, pois o código por trás é feito sobre as exigências da respetiva plataforma. Em teoria, as aplicações nativas oferecem um melhor desempenho, são mais rápidas, confiáveis e poderosas [36].

A necessidade de atender utilizadores de diferentes sistemas operativos fez com que fossem criadas ferramentas de desenvolvimento (por exemplo, *Titanium* [14] e *PhoneGap* [14]) onde um programador pode criar um modelo único compatível com múltiplas plataformas [36]. As interfaces destas ferramentas possibilitam o desenvolvimento com base nas funcionalidades e no *design*, e quando se seleciona quais as plataformas nas quais a aplicação deve funcionar, o *software* gera um pedaço de código diferente para cada sistema operativo. Tendo como grande vantagem o facto de que não será preciso o dobro do trabalho para gerar duas versões diferentes do mesmo produto.

Tendo em conta o contexto subjacente a esta tese e o *software* já existente, as aplicações móveis desenvolvidas neste trabalho seguirão a abordagem nativa, recorrendo à linguagem *Java*, visto que só será feito desenvolvimento para *Android*.

2.1.2 *Framework* para desenvolvimento de aplicações móveis

Quando é considerada a abordagem de desenvolvimento nativo é utilizada a *framework Android SDK* (*Software Development Kit*), recorrendo ao *Android Studio* [3], visto ser o ambiente de desenvolvimento integrado (*IDE*) oficial para o desenvolvimento de aplicações móveis *Android*. O *Android Studio* é baseado no *IntelliJ IDEA* [24], além do editor de código e das ferramentas de desenvolvimento avançados do *IntelliJ*, o *Android Studio* oferece ainda mais recursos para aumentar a produtividade na produção de aplicações *Android*, como: um sistema de compilação flexível baseado no *Gradle*, um ambiente unificado para se poder desenvolver para todos os dispositivos *Android*, ferramentas de

verificação de código para detetar problemas de desempenho, usabilidade e compatibilidade de versões. É possível também utilizar no *Android Studio* o *Android NDK (Native Development Kit)*, que se trata de um conjunto de ferramentas disponibilizadas pela *Google* que permite que códigos desenvolvidos em linguagem de programação *C/C++* sejam reaproveitados no desenvolvimento de projetos para dispositivos *Android*.

2.1.3 Arquitetura para desenvolvimento de aplicações móveis

Os autores do artigo [38] investigaram a influência que a escolha do tipo de arquitetura tem no desenvolvimento de aplicações móveis, principalmente em *Android*. Atualmente, o desenvolvimento à medida de aplicações móveis é muito alto[38]. Para ser competitiva, uma aplicação móvel deve ser económica e de boa qualidade. A arquitetura é importante para garantir a qualidade da aplicação ao longo do tempo e para reduzir o tempo de desenvolvimento.

O desenvolvimento em *iOS* é baseado no padrão *Model-View-Controller (MVC)* e é bem estruturado. O sistema *Android* não requer nenhum modelo específico. Soluções em que o padrão de desenho não é bem definido retardam o programador, enquanto a utilização de um padrão de desenho conhecido pode não apenas melhorar o tempo de desenvolvimento, mas também melhorar a capacidade de manutenção, a extensibilidade e o desempenho da aplicação.

Atualmente, a arquitetura oficial da aplicação nativa do *Android* é o padrão MVC [26]. Este é um padrão clássico que foi implementado com sucesso no desenvolvimento web. No entanto, no desenvolvimento em *Android*, a utilização deste padrão revelou alguns problemas. O principal problema é o facto de, no *Android*, a *View* e o *Controller* serem fortemente acoplados, a *Activity* torna-se em simultâneo a *View* e o *Controller*, o que dificulta a manutenção e o desenvolvimento.

Outros dois exemplos de arquiteturas são o *Model-View-Presenter* e o *Model-View-ViewModel* que são reivindicadas como sendo melhores do que a arquitetura MVC [26], no entanto existe pouca documentação de como implementar esta arquitetura no desenvolvimento de aplicações móveis e existem poucos artigos que fundamentem este argumento.

Como é ilustrado na Figura 2-1:

- o *Model* é tipicamente uma classe de objetos *Java* simples. É responsável pela lógica de negócio, por exemplo, obter dados de imagens, vídeo e/ou texto do servidor remoto.
- *View* contém tudo o que é visível no ecrã e oferece interação ao utilizador, são os ficheiros de *layout* em formato XML
- *Controller* é a atividade ou fragmento. Captura uma série de eventos da *View* e envia resposta de volta ou envia solicitações para o componente *Model* para obter e atualizar dados.

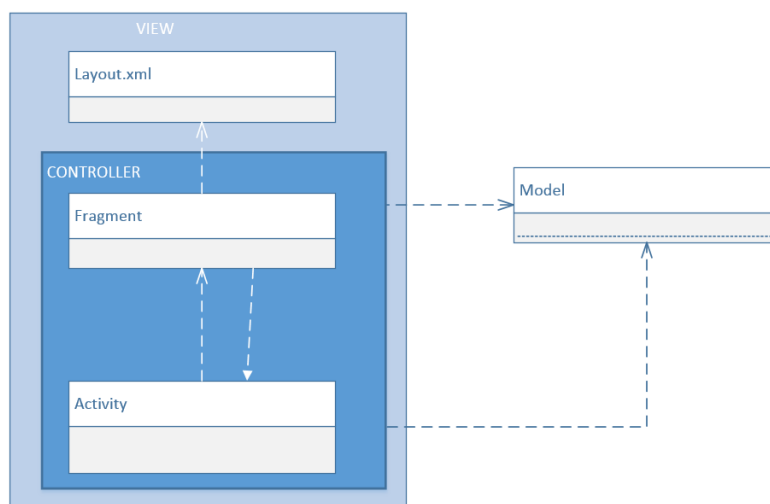


Figura 2-1: Diagrama UML da implementação do MVC em Android. Retirado de [26]

2.1.4 Framework para desenvolvimento de aplicações web

Server-side web frameworks são estruturas de software que facilitam a escrita, a manutenção e a escalabilidade de aplicações *web* e *web services*. Oferecem ferramentas e bibliotecas que simplificam as tarefas comuns de desenvolvimento da *web*, incluindo roteamento de URLs para *handlers* apropriados, interação com bases de dados, suporte a sessões e autorização de utilizadores, formatação de saída (por exemplo, HTML, JSON, XML) e aprimoramento da segurança contra ataques na *web*. Neste projeto está a ser utilizada a *framework* .NET [37], esta plataforma permite a execução, construção e desenvolvimento de *web services* de forma integrada e unificada. C# é a linguagem de programação utilizada, visto ser uma linguagem projetada para a plataforma .NET. O desenvolvimento é feito através do ambiente de desenvolvimento integrado (IDE) *Microsoft Visual Studio*.

2.1.5 Arquitetura para desenvolvimento de aplicações web

Uma arquitetura *multi-tier* [34] é uma arquitetura de software na qual diferentes componentes de software são organizados em camadas, em que fornecem funcionalidades dedicadas. A ocorrência mais comum de uma arquitetura *multi-tier* é um sistema de três camadas que consiste em uma camada de gestão de dados (na maior parte abrangendo um ou vários servidores de base de dados), uma camada de aplicação (lógica de negócios) e uma camada de cliente (funcionalidade de interface).

Implementações novas vêm com camadas adicionais. Os sistemas de informação da *web*, por exemplo, abrangem uma camada dedicada entre o cliente e a camada aplicacional. Conceitualmente, uma arquitetura *multi-tier* resulta de uma aplicação repetida do paradigma cliente/servidor. Um componente de uma das camadas intermediárias é cliente para a próxima camada inferior e, ao mesmo tempo, atua como servidor para a próxima camada superior.

2.1.6 Servidor web

Um servidor *web* [45] é um sistema para alojar aplicações *web*. É um sistema que fornece conteúdo ou serviços para utilizadores finais pela Internet. O objetivo básico do servidor *web* é armazenar, processar e entregar pedidos *web* aos utilizadores. Essa intercomunicação é feita usando HTTP (*Hypertext Transfer Protocol*). Além do protocolo HTTP, um servidor *web* pode suportar outros protocolos, como o SMTP (*Simple Mail Transfer Protocol*) para envio de correio eletrónico e o FTP (*File Transfer*

Protocol) para armazenamento. Ou seja, um servidor web pode suportar unicamente um destes protocolos ou mais do que um protocolo em simultâneo [45].

Dois exemplos destes servidores são o servidor *Apache* (*software* livre) e o *Internet Information Services* (IIS) fornecido pela *Microsoft*. IIS é um servidor web que é executado na plataforma *Microsoft .NET* no sistema operativo *Windows*. Mais comumente, o IIS é usado para alojar aplicações web ASP.NET [21] e sites estáticos. Este é formado basicamente por dois tipos de aplicações: páginas *web*, tradicionalmente acedidas por utilizadores, contém a extensão ASPX; *web services*, funções disponibilizadas pela rede, chamados por aplicações ASMX. O servidor *Apache* faz parte de um esforço colaborativo para desenvolver e manter um servidor HTTP de código aberto para sistemas operacionais modernos, incluindo UNIX e Windows. O objetivo deste projeto é fornecer um servidor seguro, eficiente e extensível que forneça serviços HTTP em sincronia com os padrões HTTP atuais [28].

2.2 Protocolos de integração, comunicação e segurança

Aqui iremos descrever os protocolos utilizados nesta tese para garantir a integração, comunicação e segurança entre as aplicações móveis e a aplicação web.

2.2.1 Protocolo de integração

Web Services [1] são soluções utilizadas na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

Um *Web Service* [1] é um conjunto de métodos acedidos e invocados por outros programas utilizando tecnologias Web, é utilizado para transferir dados através de protocolos de comunicação para diferentes plataformas, independentemente das linguagens de programação utilizadas nessas plataformas.

Os *Web Services* funcionam com qualquer sistema operativo, plataforma de hardware ou linguagem de programação de suporte *Web*. Transmitem apenas informação, ou seja, não são aplicações *Web* que suportam páginas que podem ser acedidas por utilizadores através de navegadores *Web*.

Os *Web Services* permitem reutilizar sistemas já existentes numa organização e acrescentar-lhes novas funcionalidades sem que seja necessário criar um sistema a partir do zero. Assim, é possível melhorar os sistemas já existentes, integrando mais informação e novas funcionalidades de forma simples e rápida.

2.2.2 Protocolo de comunicação

É necessária uma linguagem intermédia que garanta a comunicação entre a linguagem do *Web Service* e o sistema que faz o pedido ao *Web Service*. Para tal, existem protocolos de comunicação como o SOAP e o REST [29].

O protocolo SOAP utiliza XML para enviar mensagens e, geralmente, serve-se do protocolo HTTP para transportar os dados. Associado ao protocolo SOAP está o documento WSDL que descreve a localização do *Web Service* e as operações que dispõe. Além disso, fornece a informação necessária para que a comunicação entre sistemas seja possível.

O REST é um protocolo de comunicação mais recente que surgiu com o objetivo de simplificar o acesso aos *Web Services*. Este baseia-se no protocolo HTTP e permite utilizar vários formatos para representação de dados, como JSON (um dos mais utilizados), XML, RSS, entre outros.

Assim, uma das grandes vantagens do REST é a sua flexibilidade, já que não limita os formatos de representação de dados. O protocolo REST é também utilizado quando a performance é importante, uma vez que é um protocolo ágil e com a capacidade de transmitir dados diretamente via protocolo HTTP.

2.2.3 Protocolo de segurança para a comunicação

A falta de segurança nos sistemas cliente/servidor é uma preocupação evidente. Para isso existem mecanismos que garantem a segurança na comunicação, como conexões SSL/TLS para transmitir as solicitações dos clientes e as respostas dos servidores.

SSL/TLS [12] é um tipo de protocolo seguro que funciona entre a camada de transporte e a camada de aplicação. Ele tem como base um algoritmo criptográfico simétrico, um algoritmo criptográfico assimétrico para partilha de chaves, e um algoritmo de *hash*. Fornece confidencialidade, integridade e autenticação para a transmissão de dados de ponta a ponta. SSL/TLS é independente da camada de transporte, usa o certificado digital X.509 baseado no algoritmo de chave pública para negociar a chave simétrica de cifra/decifra, usa o algoritmo de *hash* para verificar a integridade e outros recursos seguros para transmissão de dados após a conexão TCP e antes da conexão segura estar estabelecida. Ele usa certificados para realizar autenticação de identidade obrigatória ao servidor e opcional ao cliente. Após a conexão segura SSL/TLS ser estabelecida, os clientes e servidores podem transmitir dados com segurança.

O SSL/TLS foi inicialmente projetado para os navegadores *web*, depois tornou-se o padrão seguro da Internet e o padrão prático do domínio da *web*.

2.3 Análise e armazenamento de dados

Iremos fazer uma descrição do sistema utilizado para o armazenamento dos dados, nomeadamente a as tecnologias utilizadas para armazenar os dados no servidor e para armazenar os dados nas aplicações móveis. Além disso, são descritos processos que podem ser usados para analisar os dados armazenados, tendo sido escolhido um deles no contexto deste trabalho.

2.3.1 Base de dados

As bases de dados são uma peça fundamental e indispensável na maior parte dos sistemas computacionais e uma das maiores decisões é escolher entre uma estrutura de dados relacional e não relacional [9]. Apesar de ambas serem opções viáveis, existem certas diferenças deveremos ter em conta. As bases de dados relacionais utilizam SQL como linguagem padrão para definição e manipulação dos dados, exigem que se usem esquemas pré-definidos para determinar a estrutura dos seus dados antes de trabalhar com eles, utilizam o sistema ACID para transações.

As bases de dados não relacionais não têm uma linguagem padrão (NoSQL), têm um esquema dinâmico para dados não estruturados, não garantem as propriedades ACID, mas garantem as propriedades BASE. Neste trabalho está a ser usado uma base de dados relacional, nomeadamente está a ser usado um sistema de gestão de base de dados (SGBD) da *Microsoft*, o *Microsoft SQL Server* [40].

Para aplicações móveis que utilizam persistência de dados, como é o caso deste trabalho, também é necessário definir um padrão. *Data Access Object* (DAO) [25] é um padrão que separa as regras de negócio das regras de acesso aos dados, implementado com linguagens de programação orientada a objetos (como a linguagem *Java*) e arquitetura *Model View Controller*. A vantagem de usar DAO é a separação simples e rigorosa entre duas partes importantes de uma aplicação que não devem e não podem conhecer quase nada uma da outra, e que podem evoluir independentemente. Alterar a lógica de

negócio motiva apenas a implementação de uma interface, enquanto que modificações na lógica de persistência não altera a lógica de negócio, desde que a interface entre elas não seja modificada.

O *Object Relational Mapping Lite* (ORM Lite) [30] fornece algumas funcionalidades simples e leves para persistir objetos *Java* em bases de dados SQL, evitando a complexidade e a sobrecarga de pacotes ORM mais padronizados. Esta *framework* suporta várias bases de dados SQL usando JDBC e também suporta o SQLite com chamadas nativas para APIs de bases de dados do sistema operativo *Android*. A documentação sobre como configurar o ORMLite para *Android* especificamente está disponível no manual [31].

2.3.2 Microsoft Azure

Microsoft Azure [27] consiste num conjunto de serviços na *cloud* que permite a *developers* e a profissionais de tecnologia da informação criar, implantar e gerir aplicações através de uma rede global de centros de processamento de dados. Disponibiliza vários modelos de computação, cada um com a sua função específica, que podem ser usados separadamente ou em conjunto dependendo da solução que se pretende. Iremos fazer uma breve descrição dos serviços relevantes para este projeto.

O *Azure* [10] oferece funções web, destinadas principalmente a facilitar a criação de aplicações baseadas na web. Cada instância de função da web tem o *Internet Information Services* pré-configurado dentro dela, portanto, criar aplicações usando o ASP.NET, o *Windows Communication Foundation* (WCF) ou outras tecnologias web é simples. Os *developers* também podem criar aplicações em código nativo. Isso significa que eles podem instalar e executar tecnologias não-*Microsoft*, incluindo PHP e Java.

O SQL Azure fornece um grande subconjunto das funcionalidades do SQL Server, incluindo relatórios, como um serviço de nuvem gerenciado. As aplicações podem criar bases de dados, executar *queries* SQL e muito mais, mas não é necessário administrar o sistema de base de dados ou a *hardware* em que ela é executada - a *Microsoft* cuida disso. Uma base de dados do SQL Azure pode ser acedida usando o protocolo TDS (*Tabular Data Stream*), assim como na versão local do SQL Server. Isso permite que uma aplicação do *Windows Azure* acesse a dados relacionais usando mecanismos conhecidos, como o *Entity Framework* e o ADO.NET.

2.3.3 Business Intelligence e Analytics

Durante muitos anos, as organizações geraram/guardaram uma enorme quantidade de dados utilizados, isto porque vinham de diversas fontes de informação e porque não era fácil manipular todos estes dados. Com o passar do tempo, as organizações perceberam que se essas informações não aproveitadas fossem agrupadas, seriam de enorme importância para elas e, com isso surgiu à necessidade de *softwares* que agrupassem essas informações e as exibissem de forma a que as pessoas pudessem usá-las para tomar decisões, daí surgiram os tão famosos *softwares* de *business intelligence*.

Business Intelligence (BI) [11] pode ser definido como a capacidade de extrair dados relevantes a um determinado negócio e transformar as mesmas em informações ou dados qualitativos para que, com isso, as pessoas possam tomar decisões mais fundamentadas.

Sempre foi um grande desafio para as empresas fazer com que os projetos de BI tivessem retorno rápido, devido à sua complexidade. Sempre exigiram profissionais muito bem qualificados com conhecimentos não só na área de BI, mas também nas inúmeras bases de dados para que pudesse ser feita a extração e unificado num *Dataware House*. Sem contar que as próprias ferramentas de BI sempre tiveram um custo muito alto.

Atualmente existem inúmeras ferramentas de BI no mercado. Passamos agora a rever brevemente as ferramentas QlikView e Power BI.

QlikView

O QlikView é uma plataforma de *Business Intelligence* para transformar dados em conhecimento [33]. Ela permite que os utilizadores obtenham percepções inesperadas sobre os negócios, compreendendo como os dados estão associados e quais os dados que não estão relacionados.

O que torna o QlikView único é o modelo de dados associativo "*in-memory*" [5]. Esta tecnologia faz com que as análises agreguem os dados e se construam hierarquias de dimensões em tempo de execução, sem pré-processamento. As associações entre tabelas são tratadas como se fossem um *full-outer join* entre tabelas no modelo relacional com SQL, permitindo que seja possível avaliar o que está associado e o que não está associado ao contexto de filtros aplicados pelo utilizador enquanto ele analisa os dados. Esta tecnologia associativa permite criar uma interface única simplificando radicalmente a implementação, o uso e a manutenção das consultas e análises. Ao invés de agregar dados em cubos inflexíveis, o QlikView usa uma tecnologia inovadora, o sistema de criar um grande arquivo com todas as associações, conhecido como *Data Cloud*. Desta forma, todos os dados relevantes em todas as dimensões ficam disponíveis em RAM.

AQL (*Associative Query Logic*) [35] é a tecnologia patenteada usada no software QlikView para associar dados, enquanto as *queries* são feitas na estrutura de dados QlikView. A tecnologia AQL promove análises não hierárquicas de dados, ao contrário das tecnologias OLAP (*Online Analytical Processing*) predominantes na maioria dos softwares de BI disponíveis no mercado. A tecnologia não hierárquica do AQL garante análises onde as *queries* não foram construídas numa ordem específica (como ano/trimestre/mês). Usando AQL, qualquer valor da estrutura de dados pode ser o ponto de partida para a análise.

Uma das grandes diferenças entre o QlikView e as demais ferramentas de BI está principalmente na facilidade da geração de relatórios gráficos. Traz um nível totalmente novo de análise, percepção e valor para dados existentes armazenados, com interfaces de utilizador que são simples, diretas e claras.

Power BI

O Power BI é um serviço de *business analytics* fornecido pela *Microsoft* [32]. Ele fornece visualizações interativas com recursos de *business intelligence* de *self-service*, em que os utilizadores finais podem criar relatórios e *dashboards* sozinhos, sem precisarem de depender da equipa de tecnologia da informação ou dos administradores das bases de dados.

O Power BI é uma coleção de serviços de software, aplicações e conectores que funcionam em conjunto para transformar as origens de dados não relacionadas em informações coerentes, visualmente envolventes e interativas [14]. Quer os dados sejam uma folha de cálculo do Excel simples ou uma coleção de dados híbridos baseados na *cloud*, o Power BI permite ligar facilmente às suas origens de dados, visualizar (ou detetar) o que é importante.

É composto por uma aplicação de ambiente de trabalho do *Windows* chamada Power BI Desktop, um serviço SaaS online (*software* como um serviço) denominado o serviço Power BI, e aplicações móveis do Power BI disponíveis em telemóveis e tablets *Windows*, bem como para dispositivos *iOS* e *Android*.

Um fluxo de trabalho comum no Power BI começa pela ligação a origens de dados e pela criação de um relatório no Power BI Desktop. Em seguida, esse relatório é publicado do *Desktop* para o serviço Power BI e partilhado para que os utilizadores no serviço possam ver e interagir com o relatório.

2.4 Tecnologias

Aqui iremos fazer uma descrição das tecnologias QR Code e Beacon utilizadas no âmbito desta tese, mais especificamente a tecnologia QR Code é utilizada para a validação dos bilhetes de transporte e a tecnologia do Beacon é usada como forma de obter o número de turistas nas paragens.

2.4.1 Tecnologia QR Code

QR Code [39] é um símbolo bidimensional. Foi inventado em 1994 pela Denso, uma das principais empresas do grupo Toyota, e aprovado como padrão internacional ISO (ISO/IEC18004) em junho de 2000. Este símbolo bidimensional foi inicialmente planeado para uso no controlo de produção de peças automóveis, mas difundiu-se noutros campos, como exemplo a validação de bilhetes. Agora, o QR Code é visto e usado todos os dias em todo o mundo pelas seguintes razões:

- Várias características melhores do que em códigos de barras lineares: e.g. densidade de dados muito maior, suporte a caracteres chineses
- Pode ser usado gratuitamente por qualquer pessoa, uma vez que a Denso liberou a patente para o domínio público
- Padrão de estrutura de dados não é pré-requisito para o uso corrente
- Os *smartphones* vêm equipados com câmaras que permitem a leitura de QR Codes para aceder a endereços da Internet automaticamente, simplesmente lendo um URL codificado no QR Code

Funcionamento dos QR Codes

Os QR Codes codificam um conjunto de caracteres num símbolo 2D. Este símbolo pode ser lido pelas câmaras dos dispositivos e decodificado para os caracteres originais.

À medida que a informatização se desenvolveu, nos últimos anos, cresceu a necessidade de armazenar mais informações e representar outras línguas além do inglês. Para tornar isso possível, um símbolo com densidade ainda maior do que os símbolos multiestados era necessário. Como resultado, o QR Code, que pode conter no máximo 7.000 dígitos de caracteres, incluindo caracteres chineses, foi desenvolvido em 1994.

Usos e benefícios

A utilização de QR Codes traz vários benefícios, tais como: é uma opção diferente de divulgação; pode ser exposto em qualquer lugar; é uma codificação rápida e segura; pode ser utilizado em vários formatos e qualquer tamanho; pode armazenar diferentes tipos de informações; se for utilizado para algo em contexto de confidencialidade, podem ser utilizados algoritmos criptográficos para proteger a informação.

Esta tecnologia também tem riscos, o QR Code pode ser utilizado para transmitir conteúdo ilegal, vírus, *malware* e outros códigos maliciosos. Ainda há várias pessoas que desconhecem a tecnologia e, como a utilização da mesma depende de uma ação do utilizador (ler o código). Os utilizadores precisam de ser instruídos.

Biblioteca ZXing

ZXing [7] (“*zebra crossing*”) é uma biblioteca de processamento de imagens de código de barras implementada em *Java*, compatível para outras linguagens. Possui suporte para códigos de barras de produtos 1D, industriais 1D e 2D.

O ZXing é usado pela pesquisa na *web* para tornar milhões de códigos de barras na *web* indexáveis. Também forma a base da aplicação *Barcode Scanner* do *Android* e é integrado ao *Google Product e Book Search*.

Recorreu-se a esta biblioteca, de código aberto, para a geração e leitura dos QR Codes.

2.4.2 Tecnologia *Beacon*

Os *beacons* têm o potencial de transformar a forma como os *smartphones* são utilizados pelos consumidores e pelos negócios. A vantagem inédita está no, quase omnipresente, suporte a dispositivos já existente no mercado, o maior desafio a ser superado é o debate sobre privacidade com o consumidor. Os *beacons* são pequenos transmissores *wireless* que usam a tecnologia *Bluetooth* de baixo consumo (BLE) para emitir sinais para dispositivos inteligentes (como *smartphones*) nas proximidades. São um dos mais recentes desenvolvimentos em tecnologia de localização e *marketing* de proximidade. Simplificando, eles ligam-se e transmitem informações para dispositivos inteligentes, tornando a pesquisa e a interação baseadas em localização mais fácil e precisa.

Nesta secção vamos analisar esta tecnologia em maior detalhe.

Funcionamento dos *beacons*

Cada dispositivo contém um CPU, rádio e baterias, e funciona transmitindo repetidamente um identificador. Esse identificador é captado por dispositivos que possuam a tecnologia BLE, geralmente um telemóvel, dentro de um raio próximo. O identificador é um número de identificação único que o *smartphone* reconhece como exclusivo para o *beacon*. As aplicações podem ser desenvolvidas para fazer com que eventos sejam executados caso o dispositivo esteja dentro do intervalo detetável do *beacon*. Além disso, o dispositivo é capaz de calcular o quão perto ou longe está do emissor, o que significa que diferentes eventos podem ser acionados dependendo se um dispositivo está dentro do raio de 5, 25 ou 70 metros de um emissor BLE. Um dispositivo pode identificar vários *beacons* simultaneamente e, calcular a sua distância relativa a cada um dos *beacons*, assim, o dispositivo pode utilizar os *beacons* como elementos de reconhecimento de localização.

Importância da utilização de *beacons*

Os *beacons* são importantes porque abordam vários desafios que se tentam resolver há muitos anos:

- Comunicação segura baseada em proximidade - Oferecer a dois dispositivos a capacidade de comunicarem com segurança um com o outro, quando estão próximos, é o desafio comercial que a tecnologia NFC tem tentado resolver. Os *beacons* também podem resolver esse desafio, mas com o benefício adicional de que o dispositivo não precisa ser colocado em contacto físico com um sensor.
- Geolocalização - A tecnologia GPS é excelente para utilização no exterior, mas os sinais de satélite são, por exemplo, significativamente menos eficazes dentro de um edifício. Os *beacons* oferecem uma solução económica para os serviços de localização, com a vantagem adicional de reduzir significativamente o consumo de energia na bateria de um *smartphone* do que a tecnologia GPS.
- Distribuição abrangente - A grande maioria dos *smartphones* produzidos nos últimos anos suporta a tecnologia BLE, o que significa que a massa crítica de utilizadores necessária para o sucesso de uma nova tecnologia já existe.

Usos e benefícios

Ao instalar os *beacons* em vários locais e ao combiná-los com aplicações criadas para os utilizadores, é possível identificar exatamente onde os utilizadores estão localizados e fornecer informações contextuais relevantes.

No entanto, é importante considerar que, nesta fase, os consumidores não estão completamente conscientes das capacidades do BLE. Para obter benefícios reais desta tecnologia, os clientes devem ter instalada uma aplicação e ter o *Bluetooth* ligado. Fazer com que os utilizadores abram a aplicação no local e liguem o *Bluetooth* pode ser um desafio.

Depois de superar esses problemas, a utilização dos *beacons*, geralmente, pode-se enquadrar numa das quatro categorias seguintes:

- Ponto de interesse estático: este recurso refere-se à ativação de uma ação ou evento quando um utilizador chega a uma determinada distância de um local físico específico. A localização do ponto de interesse é estática, daí o nome ponto de interesse estático.
- Mapeamento interior: empresas de todos os setores tentam resolver o desafio de geolocalização interna há algum tempo, e a BLE pode ser a tecnologia que fornece uma solução. Desenvolvendo aplicações e instalando BLE's em diversos locais, os clientes podem circular dentro dos grandes edifícios com mais facilidade, economizando tempo e garantindo que chegam ao sítio que queriam.
- Proximidade bidirecional: esta funcionalidade introduz a ideia do dispositivo do utilizador se tornar um *beacon*. Uma ação ou evento pode ser ativa num dispositivo que transmite o ID do utilizador e o local para a aplicação de outro dispositivo, abrindo a capacidade de localizar amigos, colegas, funcionários e clientes.

Algumas empresas podem achar mais económico criar a aplicação para se comportar como um *beacon*, em vez de instalar os *beacons* por toda a parte, para que se possam localizar os utilizadores. A proximidade bidirecional pode melhorar as interações, oferecer uma experiência mais personalizada e melhor serviço.

- Análise: a oportunidade para as empresas, de vários setores, recolherem dados valiosos das aplicações em conjunto com os *beacons* é enorme: saber coisas como quem entrou, em qual área, a que horas e por quanto tempo, ou quem não entrou, mas estava por perto, pode abrir todo um novo mundo. É importante que os profissionais usem esses dados para personalizar continuamente as ofertas, a fim de introduzir confiança e oferecer utilidade valiosa.

Privacidade dos utilizadores

Os *beacons* não recolhem dados dos utilizadores, mas as aplicações ligadas aos mesmos recolhem. A aplicação dos utilizadores deteta os *beacons* e localiza quando e onde o utilizador esteve. No entanto, a quantidade e o tipo de dados pessoais recolhidos dependem do tipo de aplicação. Se o cliente tiver fornecido muitas informações pessoais e já tiver dado permissão para usar esses dados, os dados reunidos poderão ser analisados num nível mais profundo. Se a aplicação não solicitar nenhuma informação pessoal a ser inserida, os dados serão anónimos e a análise dos dados será mais limitada.

Como em todos os dados do consumidor, a perceção real dependerá da qualidade da análise. Cada vez mais a privacidade é uma preocupação para os utilizadores. Convém garantir que as suas informações não sejam recolhidas ou, no mínimo, obrigar a uma política que, sempre que dados pessoais forem guardados, garanta que os utilizadores forneçam permissão explícita [20]. Do ponto de vista do consumidor, se quiserem ter os seus dados recolhidos por uma organização, a troca de valor para o

utilizador ao fazê-lo deve ser maior em termos de conteúdo e ofertas que receberão de volta. É importante para a segurança garantir que os dados sejam cifrados na origem.

Tipos de *beacons*

Existem várias bibliotecas disponíveis de *beacons* BLE (iremos abordar apenas duas das mais usadas: *AltBeacon* e *iBeacon*), cada uma com os seus próprios padrões e vantagens. Algumas são de código aberto e gratuitas, algumas não disponibilizam o código fonte e custam dinheiro. Irão ser abordados os dois tipos principais de *beacons* disponíveis, vantagens, desvantagens e alguns detalhes de implementação de baixo nível sobre como os *beacons* funcionam.

O BLE tem a capacidade de trocar dados em um de dois estados: modo conectado e modo de *advertising*. O modo conectado usa a camada GATT (*Generic Attribute*) para transferir dados numa conexão um-para-um. O modo de *advertising* usa a camada GAP (*Generic Access Profile*) para transmitir dados para quem está à escuta. O modo de *advertising* é uma transferência de um para muitos e não tem garantias sobre a coerência dos dados.

Os BLE *beacons* aproveitam o modo de *advertising* GAP para transmitir dados em pacotes de *advertising* periódicos, especialmente formatados. Cada tipo de *beacon* usa uma especificação personalizada para particionar os dados de *advertising*, dando-lhe significado.

iBeacon

O *iBeacon* [23] da *Apple* foi a primeira tecnologia BLE *Beacon* a ser lançada, portanto a maioria dos *beacons* inspiram-se no formato de dados *iBeacon*. Os *iBeacons* são ativados em vários *Software Development Kits* da *Apple* e podem ser lidos e transmitidos de qualquer *iDevice* habilitado para BLE. O *iBeacon* é um padrão proprietário e fechado. Existe um grande ecossistema em torno dos *iBeacons* e um grande conjunto de recursos para desenvolvedores, mas é necessário fazer parte da comunidade de desenvolvedores da *Apple*.

Especificação

Os *iBeacons* transmitem quatro informações:

1. Um UUID que identifica o *Beacon*.
2. Um número principal que identifica um subconjunto de *Beacons* dentro de um grupo grande.
3. Um número menor que identifica um *Beacon* específico.
4. Um nível de potência transmitida (*TX Power*) no cumprimento de 2, indicando a força do sinal a um metro do dispositivo. Esse número deve ser calibrado para cada dispositivo pelo utilizador ou fabricante.

Uma aplicação através do *Bluetooth* lê o UUID, o número principal e o número menor e faz pesquisa a uma base de dados para obter informações sobre o *beacon*; o *beacon* em si não contém informações descritivas - ele requer que essa base de dados externa seja útil. O campo de potência de transmissão é usado com a força do sinal medido para determinar a distância que o sinal está do *smartphone*.

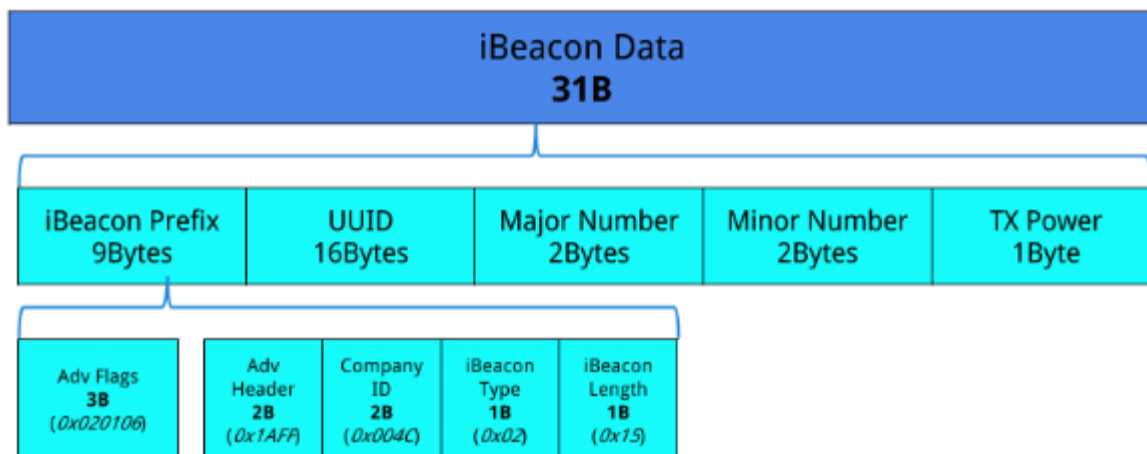


Figura 2-2: Especificação iBeacon. Retirado de [23]

O prefixo *iBeacon* contém os dados hexadecimais: 0x0201061AFF004C0215. Isso divide-se da seguinte maneira:

- 0x020106 define o pacote *advertising* como BLE *General Discoverable* e BR / EDR de alta velocidade incompatível. Efetivamente, diz que é apenas transmissão, não conexão.
- 0x1AFF diz que os seguintes dados têm 26 *bytes* e são dados específicos do fabricante.
- 0x004C é o *Bluetooth Sig ID* da *Apple* e é a parte desta especificação que faz com que seja dependente da *Apple*.
- 0x02 é um ID secundário que denota um *beacon* de proximidade, que é usado por todos os *iBeacons*.
- 0x15 define o comprimento restante como 21 *bytes* (16 + 2 + 2 + 1).

A proximidade UUID é um BLE UUID padrão de 16 *bytes* / 128 *bits* e é tipicamente exclusivo para uma empresa. Os números maior e menor são usados para denotar ativos dentro desse UUID; usos comuns são grandes números que são guardados (assim 65.536 valores possíveis) com números menores sendo *tags* individuais dentro das lojas (novamente 65.536 *tags* possíveis por guardar).

Exemplos

1. Um café X tem *iBeacons* numa prateleira de café. Quando um cliente entra e chega perto do café, uma aplicação móvel “vê” o *iBeacon*, pesquisa na base de dados *iBeacon* do café X, reconhece o *iBeacon* como pertencente ao café X, depois vê que há um cupão válido para café moído e notifica o utilizador através da aplicação.
2. *iDevices* podem transmitir um *iBeacon*. Isso pode ser usado para automatizar *check-ins* em eventos e rastrear movimentos em todos os locais.

Reflexões

Os *iBeacons* são amplamente suportados e, como são um produto da *Apple*, todos trabalham com eles e o ecossistema é o mais robusto possível. Tem a limitação de ser necessário uma base de dados para dar significado aos dados do *iBeacon*. Sem uma base de dados, os UUIDs não têm significado.

AltBeacon

AltBeacons [2] é um projeto de *beacons* com especificação aberta e desenho livre fornecido pela *Radius Networks*. A especificação *AltBeacon* parece ser uma resposta direta à especificação de *iBeacon* de código fechado. Ele cobre a mesma funcionalidade que um *iBeacon* tem, mas não é específico da empresa. Contudo, ainda não é tão amplamente suportado.

Uma diferença importante é que, enquanto os *iBeacons* têm 20 de 27 bytes disponíveis para os dados do utilizador (UUID + Major + Minor), os *AltBeacons* têm 25 de 28 bytes disponíveis (ID MFG, BeaconCode, BeaconID, MFG RSVD). Isso significa que pode haver mais dados entregues por mensagem.

Especificação

A especificação do *AltBeacon* [17] tem 28 bytes (26 bytes são modificáveis pelo utilizador). Os dois primeiros bytes do *AltBeacon* não são modificáveis pelo utilizador, mas são definidos pela pilha do BLE. ADV Length é 0x1B e ADV Type é 0xFF; estes especificarão o tamanho do pacote de dados de *advertising* e o tipo dos dados de fabricação, respetivamente. Depois disso, tudo depende do utilizador e pode ser inserido num campo de dados do fabricante de *advertising*.

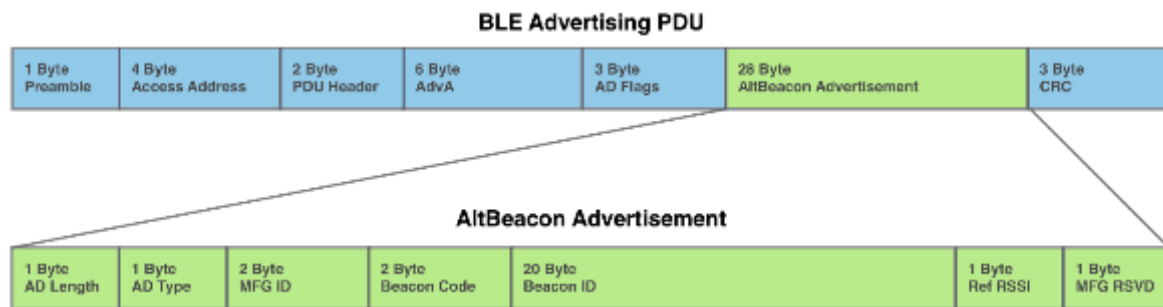


Figura 2-3: Especificação AltBeacon. Retirado de [2]

Exemplos

Os mesmos exemplos usados para os *iBeacons* aplicam-se aos *AltBeacons*, com as seguintes modificações:

1. A capacidade de ter IDs de fabricante diferentes.
2. A capacidade de ter diferentes códigos de *beacon*.
3. O byte de dados reservados no final que contém informações específicas do fabricante.

Com o *AltBeacon*, torna-se possível ter UUIDs específicos da aplicação, em vez de específicos da empresa, dando a capacidade de alterar o ID da empresa.

Reflexões

Os *AltBeacons* fazem bom uso do espaço que têm, podem ser retro compatíveis com os *iBeacons* e oferecem uma API. Sendo uma especificação muito nova, ainda não está muito difundida e enfrenta uma batalha frente à tecnologia já estabelecida no mercado dos *iBeacons*. Com o tempo, é provável que mais e mais pessoas comecem a usar *AltBeacons*, simplesmente porque podem carregar mais dados e ter um caso de uso mais amplo.

O *iBeacon* e o *AltBeacon* são muito semelhantes. A principal diferença está na disponibilidade da sua API. Ambos transmitem um UUID e alguns bytes de dados específicos para o dispositivo. Os *iBeacons*

têm um espaço ligeiramente menor do que os *AltBeacons* e são mais regulamentados, mas, em última análise, ambos são métodos de comunicação unidirecionais, e ambos exigem que uma base de dados externa e uma aplicação móvel com ligação *Bluetooth* para que sejam utilizados.

Neste trabalho é usada a implementação disponibilizada pela *Radius Networks*, a especificação *AltBeacon*, a principal razão é o facto de a biblioteca ser *opensource*, que não envolve custos adicionais.

2.5 Casos de estudo

Iremos fazer uma descrição de alguns casos de estudo com recurso a tecnologias utilizadas neste trabalho e que vão de encontro aos objetivos que pretendemos alcançar nesta tese.

2.5.1 Tecnologias inteligentes no turismo

As tecnologias inteligentes alcançam quase todos os aspetos da vida hoje em dia, embora muitas vezes ainda não sejam notadas pelos utilizadores. O projeto de investigação apresentado em [4] está concentrado na tecnologia *iBeacon*. Para determinar o real aprimoramento da experiência do consumidor, o evento SAIL Amsterdam 2015 é usado como estudo de caso. Os *developers* da tecnologia, os gerentes de eventos da SAIL Amsterdam 2015, os visitantes deste evento e os utilizadores da nova tecnologia, bem como especialistas, foram entrevistados. Os resultados mostram que as aplicações inteligentes de turismo melhoram as experiências dos visitantes. Um grande problema, porém, é a pouca informação que os visitantes têm das possibilidades que estas aplicações oferecem.

O artigo [4] preparado para a Conferência Internacional de Estudantes de Turismo do ano de 2016 sobre Tecnologias Inteligentes no Turismo é baseado num caso de estudo de 232 *iBeacons* que foram usados na SAIL Amsterdam, formando cinco redes de *beacon* públicas e privadas. O artigo debate o que é o Turismo Inteligente, como medir a experiência do cliente e as várias fases pelas quais um turista passa (antecipatório, planeamento, experiencial, reflexão).

Aproximadamente 3% de todos os visitantes, cerca de 69000 pessoas, do SAIL, usaram a aplicação e desses 47% tiveram o *Bluetooth* ligado. Isso permitiu que os organizadores acompanhassem o comportamento dos visitantes e gerissem os fluxos de visitantes.

Barten [4] discute a perceção da perda de privacidade dos visitantes ao instalar a aplicação e, como mitigação para este problema, propõe que se garanta que existe um entendimento dos utilizadores relativamente ao fornecimento dos seus dados de localização como um acordo justo para receber informações. Outra questão discutida são os diferentes interesses dos *developers*, patrocinadores e organizadores do SAIL ao criar a aplicação.

Após as entrevistas aos utilizadores e as revisões da aplicação SAIL, pôde-se concluir que a tecnologia *iBeacon* aprimora a experiência do turista, especialmente no contexto da gestão de multidões e do uso de aplicações.

No entanto, sendo uma nova tecnologia, enfrenta o desafio da falta de ampla informação e conhecimento entre os utilizadores e empresas. Os utilizadores geralmente não sabem como usar a aplicação corretamente e, conseqüentemente, não ativam a função *Bluetooth* no *smartphone*. Portanto, é aconselhável fornecer informações abrangentes sobre a tecnologia e o uso adequado dela, a fim de lucrar com o máximo de serviços oferecidos pelas aplicações. Além disso, as empresas estão dispostas a implementar a tecnologia *iBeacon*, mas muitas vezes ignoram os serviços e funções reais que a tecnologia oferece.

Posto isto, o sucesso no uso de *beacons* e aplicações está ligado à quantidade de informações que os

visitantes têm sobre as possibilidades e os benefícios ao instalar e usar a aplicação.

Yamaguchi, Akihiro et al. mostram num estudo [44] mais recente um sistema que fornece informações turísticas e obtém as tendências dos visitantes usando beacons e serviços. Um dispositivo BLE é usado para transmitir um identificador universalmente único. Além disso, os beacons são colocados em pontos turísticos e instalações turísticas. O sistema proposto compreende dois programas; um é um programa aplicativo do lado do cliente que fornece informações turísticas específicas da área correspondentes ao beacon detectado; outro é uma aplicação do lado do servidor para registrar informações de hora e local dos beacons detectados. Neste artigo [44], os autores mostram que através de beacons conseguem recolher informações de localização que podem ser usadas para identificar os caminhos feitos pelos turistas.

Um dos objetivos desta tese é tirar partido dos potenciais que a tecnologia dos beacons oferece. A identificação dos turistas na proximidade das paragens através dos beacons pode ser decisivo para a gestão de multidões, sendo que Barten e Yamaguchi, Akihiro et al. demonstram nas suas experiências que essa gestão pode mesmo acontecer com esta tecnologia.

2.5.2 Exemplo de uso da tecnologia BLE

Gowrishankar et al. demonstram com sucesso o papel do BLE na automatização da proximidade baseada em soluções para o IoT, considerando um caso de uso num cenário do mundo real [22].

Foi desenvolvida uma aplicação para mostrar como várias tecnologias diferentes são integradas para formar o paradigma IoT. Foi usado um nó fixo *AltBeacon* que permite que uma aplicação, em execução num *smartphone*, determine a sua proximidade em relação ao nó *AltBeacon*. Quando a aplicação se encontra na proximidade do nó *AltBeacon*, é possível controlar vários dispositivos habilitados para IP por meio do serviço RESTful.

O caso de uso considerado trata-se de uma luz que é ligada através de um microcontrolador quando o *smartphone* com o BLE ligado verifica que está na proximidade do nó do *AltBeacon*. A luz desliga-se no caso contrário.

A principal contribuição do trabalho em [22] é a proposta de uma estrutura que faz uso do BLE em IoT usando um serviço RESTful.

Para a implementação do caso de uso, os autores fizeram uso de vários recursos de hardware e software, como um *Android Smartphone* habilitado de BLE 4.0, um nó *AltBeacon*, microcontrolador *Arduino* com biblioteca aREST (biblioteca REST disponível para *Arduino* e outras plataformas como o *Raspberry Pi*). Esta implementação está representada na Figura 2-4. O microcontrolador *Arduino* conecta-se à Internet, suporta os protocolos TCP e UDP, fazendo uso da biblioteca de Ethernet. A aplicação *Android* habilitada para BLE faz um pedido HTTP usando a API REST. A biblioteca aREST é usada para projetar APIs REST, que permite que a aplicação *Android* comunique com o *Arduino*. Este pedido é enviado para o microcontrolador *Arduino* que atua como *gateway*. Um pequeno servidor executado no microcontrolador *Arduino* processa este pedido invocando as APIs REST. Nesta fase, os dados JSON são gerados para indicar que a solicitação HTTP foi processada e os dados JSON são enviados de volta à aplicação *Android* para processamento adicional. Além disso, a aplicação *Android* usa as APIs REST para controlar a funcionalidade do microcontrolador *Arduino*.

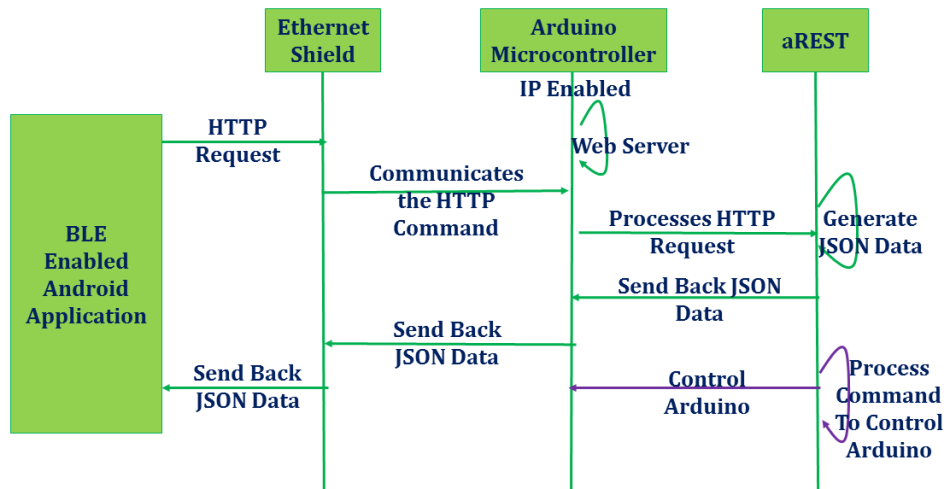


Figura 2-4: Interface baseada em REST entre a aplicação Android (habilitada para BLE) e Microcontrolador Arduino. Retirado de [18]

Assim que o dispositivo *Android* com a aplicação BLE deteta o *AltBeacon* é calculada a distância, através da fórmula “a razão da força do sinal *AltBeacon* para o poder calibrado do transmissor” [18]. Os autores decidiram que caso o dispositivo tivesse dentro do raio de 1.5m do *AltBeacon* a luz acenderia, caso contrário apagava.

Na Figura 2-5 é possível observar que a aplicação deteta o *beacon* em dois momentos distintos. No primeiro momento (lado esquerdo) a aplicação deteta o *beacon*, calcula a distância a que está o *beacon* e verifica que a distância é sensivelmente 3.5m, logo como é superior a 1.5m a luz do microcontrolador não acende. No segundo momento (lado direito) o processo é igual, mas a distância calculada é sensivelmente 1m, logo como é inferior a 1.5m a luz do microcontrolador acende.

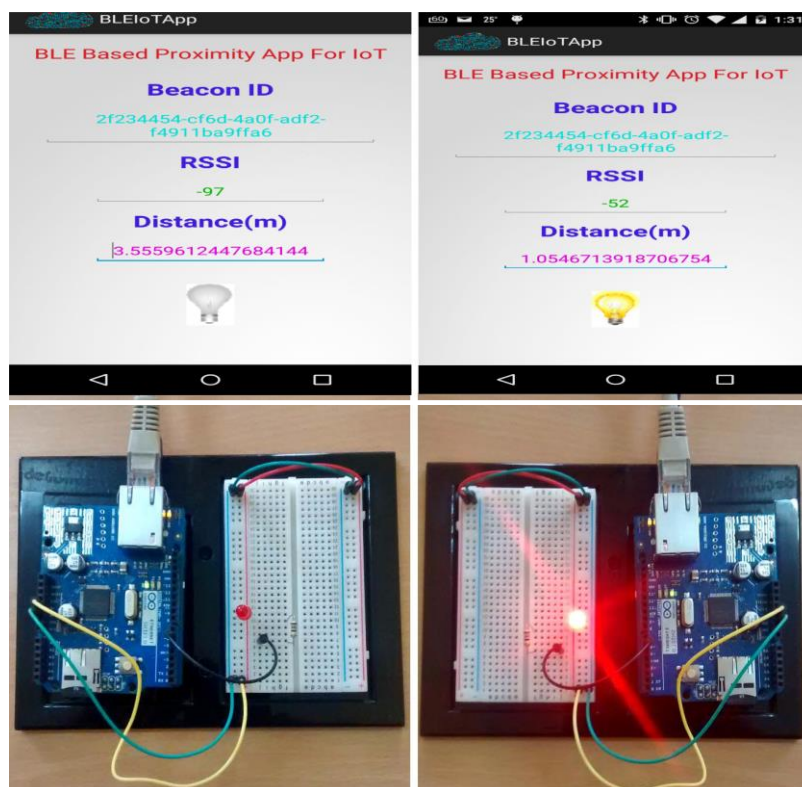


Figura 2-5: Aplicação baseada em BLE e AltBeacon. Retirado de [18]

As soluções de proximidade permitem gerar e entregar informações valiosas e relevantes para o utilizador final. A solução apresentada vai de encontro a um dos pontos deste trabalho, quando se pretende detetar o número de turistas presentes numa paragem, sendo por isso, um bom caso de uso do mundo real que foi considerado e implementado na plataforma *Android* para gerir eficazmente o desencadeamento de eventos com base na proximidade.

2.5.3 Exemplo de uso da tecnologia NFC e QR Code

A evolução das tecnologias de comunicação e o uso generalizado de *smartphones* possibilita o seu uso comum em inúmeras aplicações diárias. Em [18], Finžgar e Trebar descrevem a implementação de um sistema que permite o uso de telemóveis para adquirir bilhete eletrónico de transporte público. QR Codes e *tags* RFID são usadas para registar passageiros no início e no fim das viagens. Um passageiro pode ver os valores cobrados por um bilhete e outras informações, como partidas. Além disso, um condutor pode usar a aplicação *Android* para verificar a validade dos bilhetes. Além da possibilidade de obter um bilhete eletrónico, que é armazenado no telefone. Um site adicional foi projetado para que os administradores possam ver os dados sobre as viagens dos passageiros e para analisar os resultados estatísticos.

A identificação ou a marcação de um local de *check-in* ou *check-out* depende do *smartphone* usado no sistema eletrónico de bilhetes. Para telemóveis com tecnologia NFC, as *tags* RFID são geralmente a melhor escolha, caso contrário e para qualquer outro telemóvel com câmara, o QR Code pode ser usado. Descrição da viagem do passageiro é definida com quatro métodos de especificação do sistema:

Início da viagem:

- 1- Passageiro coloca o seu *smartphone* perto da *tag* RFID ou lê o QR Code no ponto de check-in numa estação.
- 2- Aplicação no *smartphone* recebe o nome da estação.
- 3- Aplicação envia solicitação para emissão de um ticket para o servidor.
- 4- Aplicação no servidor autoriza o utilizador e envia bilhete de volta na forma de QR Code ou confirmação de validade bilhete pré-pago.

Fim da viagem:

- 5- Nome da estação de check-out é lido a partir da *tag* RFID ou QR Code e é enviado para o servidor.
- 6- Servidor finaliza a viagem.

Verificação de bilhetes:

- 7- O condutor usa uma aplicação no seu telemóvel *Android* para ler o bilhete a partir do ecrã do telefone do passageiro e verifica isso.

Calculando a tarifa:

- 8- Baseado na conhecida estação de check-in e check-out, a aplicação no servidor calcula a tarifa.
- 9- Utilizadores recebem recibos das suas viagens anteriores no e-mail.

Os administradores podem aceder ao site e verificar, entre outros, os seguintes dados:

- 1- Número de passageiros em determinadas rotas (com gráfico);
- 2- Número de passageiros em determinadas rotas em determinados horários;
- 3- Rácio de passageiros que pagam taxa fixa aos que pagam por viagem;
- 4- Número de novos utilizadores registados para o último período definido.

A solução proposta de um sistema de bilhetes eletrónicos mostra uma ideia de eliminar uma infraestrutura complexa de muitos dispositivos eletrónicos no transporte público. Não há necessidade de ter cartões inteligentes, o que beneficia o passageiro e também as empresas de transportes que não

precisam de instalar leitores de cartões inteligente. Desta maneira, também não existe a necessidade de ter funcionários de suporte com sistema de bilhetes móvel. Ultimamente, novas tecnologias oferecem um preço muito apropriado para a identificação de estações e também permitindo o armazenamento do bilhete eletrônico, em forma de QR Code, num telefone. Todas as operações, incluindo a identificação de cada ponto de *check-in* e de *check-out*, ou até a verificação de bilhetes, pode ser feita por qualquer *smartphone*.

Depois de visto um exemplo da utilização dos QR Codes para a validação de bilhetes em transportes públicos, podemos observar que em 2017 Zilu Tian et al. fizeram uma análise empírica para perceber as atitudes dos turistas em relação aos serviços com QR Code [46]. O estudo utiliza 247 campos de resposta e confirma que o uso da tecnologia de QR Code no turismo influencia a satisfação das deslocções de um indivíduo e a satisfação da viagem, sugerindo que esta tecnologia pode ser usada para evoluir a indústria do turismo.

A utilização da tecnologia QR Code elimina os custos relacionados aos terminais do ponto de venda, economiza tempo e aumenta a eficiência dos funcionários e a satisfação dos clientes, permitindo a rápida conclusão das transações [46].

Capítulo 3

O turismo inteligente

Nesta tese iremos tirar partido de um sistema já existente, que se encontra a ser utilizado por vários clientes. Este sistema permite a venda de reservas de viagens turísticas numa determinada cidade. Existe uma aplicação móvel implementada que oferece mobilidade, uma vez que com um dispositivo móvel *Android* e uma impressora móvel os funcionários podem andar livremente pela cidade a vender reservas, não precisam de ficar num local específico, e permite aos motoristas dos transportes que também tenham acesso ao sistema. A motivação prende-se no facto de, juntando novas tecnologias, melhorar o serviço já existente, nomeadamente, melhorar a validação dos bilhetes, para além disso pretende-se oferecer uma aplicação móvel aos turistas que satisfaça as suas necessidades. Em resultado do uso das duas aplicações pretendemos fazer uma análise ao movimento dos turistas de forma a otimizar o serviço de transportes disponibilizado aos turistas.

Neste capítulo vamos especificar as principais tarefas inerentes a este trabalho. Começamos por apresentar, de forma breve, o sistema existente e as funcionalidades relevantes para este trabalho e, por fim, especificamos as alterações feitas ao sistema.

3.1 Sistema existente

Iremos fazer uma breve descrição das componentes e da arquitetura do sistema já existente na Card4B, indicando as componentes mais relevantes do mesmo para o âmbito deste trabalho.

4POS

O 4POS é a aplicação móvel utilizada pelos agentes que se encontram em diferentes locais das cidades ou pelos próprios motoristas dos autocarros. Após o início de sessão na aplicação é possível observar o ecrã inicial, apresentado na Figura 3-1, onde temos acesso às seguintes funcionalidades:

- Trocar voucher: permite trocar um voucher por uma reserva;
- Nova reserva: permite fazer uma reserva de um ou mais produtos;
- Resumo de vendas: permite ao funcionário visualizar as vendas realizadas por si naquele dia;
- Configurações: permite configurar a impressora, limpar a base de dados e definir o número do equipamento.

A aplicação 4POS encontra-se conectada a uma impressora, que pode ser um terminal de pagamento automático (TPA) ou uma impressora móvel Zebra, que permite a impressão dos bilhetes correspondentes às reservas. No caso do TPA permite ainda fazer o pagamento das reservas através de cartões com acesso à rede Multibanco. A aplicação 4POS é utilizada por diferentes clientes, sendo que para cada cliente é definido o tipo de impressora a utilizar. Isto significa

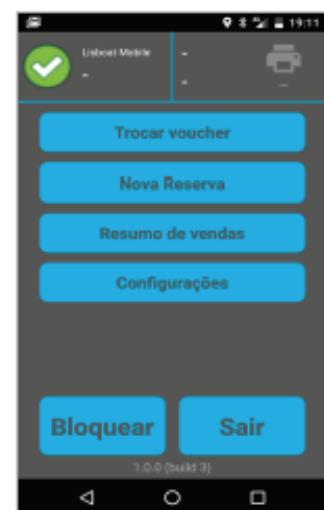


Figura 3-1: Menu da aplicação 4POS

que, apesar da existência das duas implementações, a aplicação só permite utilizar impressoras do tipo TPA ou do tipo Zebra e não os dois tipos em simultâneo.

4BOOKING

O 4BOOKING representa o *BackOffice* do sistema, coordena e gere os pedidos feitos pela aplicação 4POS ao servidor (por exemplo, pedido da aplicação 4POS sobre a disponibilidade de um determinado produto).

Para além dos serviços fornecidos à aplicação disponibiliza uma solução *Web* para gestão de reservas de produtos turísticos, com o seguinte suporte: configuração do produto; gestão de reservas; gestão de viagens; painéis e notificações; perfis de utilizadores (por exemplo, agências de viagens, hotéis)⁴.

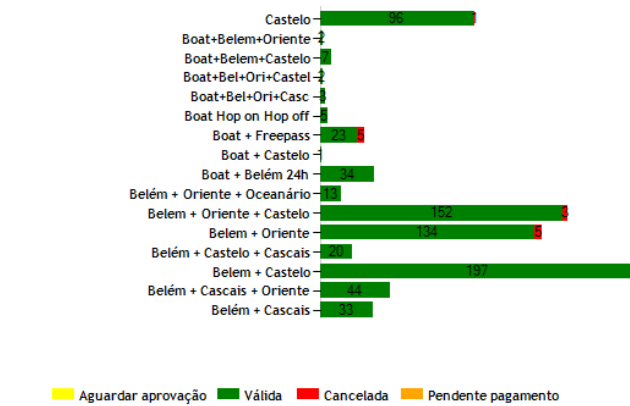
Na Figura 3-2 é possível observar parte do ecrã inicial do *BackOffice* do 4BOOKING, onde temos um resumo atualizado das reservas, conseguimos observar as reservas criadas no 4POS e os detalhes das mesmas, sendo relevante o estado de impressão dos bilhetes da reserva: “Impressão parcial” significa que não foram impressos todos os bilhetes da reserva, enquanto que “Impressão OK” significa que foram impressos todos os bilhetes da reserva.

Entidade	Canal	Data
705080 - Catia Cordeiro (Cityrama PDA)	Mobile	2019-07-30 14:31:42
[Impressão OK - Pago] Belem + Oriente + Castelo		2019-07-30 2pax
Cliente Final		
705079 - Natariel Almeida (Cityrama PDA)	Mobile	2019-07-30 14:31:34
[Impressão parcial - Pago] Belem + Castelo		2019-07-30 2pax
Cliente Final		
705078 - Antonio Monteiro (Cityrama PDA)	Mobile	2019-07-30 14:30:16
[Impressão OK - Pago] Belem + Oriente + Castelo		2019-07-30 2pax
Cliente Final		
705077 - Rui Pereira (Cityrama PDA)	Mobile	2019-07-30 14:29:12
[Impressão OK - Pago] Belem + Oriente		2019-07-30 2pax
Cliente Final		

Figura 3-2: Informação das reservas

⁴ <https://www.card4b.pt/projects/4booking.1.html>

A Figura 3-3 também faz parte do ecrã inicial do *BackOffice*. Aqui é possível observarmos o total de reservas realizadas por cada produto nesse dia. Permitindo perceber quais os produtos mais vendidos.



Categoria de produto	Aguardar aprovação	Pendente pagamento	Válida	Cancelada	Integrado
Circuitos Sightseeing Algarve	2	-	2	3	-
Circuitos Sightseeing Lisboa	-	2	207	13	-
Circuitos Sightseeing Porto	-	-	24	4	-
Hop On Hop Off Lisboa	-	-	843	14	-
Hop on Hop off Porto	-	-	14	-	-
Transferes	-	-	9	2	-
Walkinn Tours	-	-	2	-	-

Figura 3-3: Resumo de todas as reservas

Arquitetura

Aqui será dada uma visão global da arquitetura, posteriormente iremos descrever a arquitetura do sistema de um ponto de vista mais funcional e estrutural. O sistema divide-se em três camadas: camada de apresentação, camada lógica e camada de acesso aos dados.

- A camada de apresentação é a camada com que o utilizador interage, relativa à aplicação móvel 4POS e à interface *Web* do 4BOOKING.
- A camada lógica corresponde à camada de serviços 4BOOKING, serve de intermediária entre a aplicação móvel/interface *Web* e a base de dados através de *queries* feitas à base de dados, de modo à aplicação móvel/interface *Web* conseguir apresentar ao utilizador os dados pretendidos, assim como alterar dados existentes na base de dados de acordo com as operações executadas pelo utilizador, para além da execução de lógica de negócio que não faça sentido ser executada na aplicação móvel.
- A camada de acesso aos dados refere-se à base de dados, onde são armazenados os registos das tabelas.

Na Figura 3-4 apresenta-se o diagrama da arquitetura.

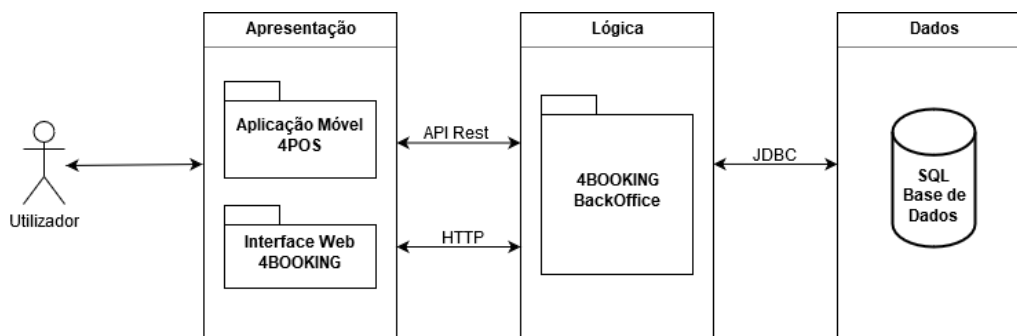


Figura 3-4: Arquitetura do sistema

3.2 Alterações ao sistema e novas contribuições

Aqui iremos apresentar, de forma objetiva, todas tarefas realizadas no âmbito desta tese para que seja possível identificar facilmente as alterações feitas no sistema existente e as novas contribuições.

a – Deslocação nos sistemas de turismo:

- Identificação de níveis de presença de turistas na cidade – recorrendo às metodologias de triangulação por GSM/Wi-Fi e à API da *Google Location Services*. Alimentar e mostrar em mapas no *BackOffice 4BOOKING*, com a localização das coordenadas GPS nas transações de venda e validação.
- Reconhecimento de turistas na proximidade de paragens – deteção através de *Beacons*, dispositivos que enviam sinais via *Bluetooth Low Energy* que podem ser captados por *smartphones* – integração e interpretação dos *Beacons*.

b – Integração com informação de bilhética:

- Complemento de venda a bordo e *check-in/check-out* – inclusão na aplicação, já existente 4POS de uma funcionalidade de validação de bilhetes (*QR Code* gerado quando é feita a compra de um bilhete que será depois validado através da aplicação 4POS).

c – Integração com sistema de localização e gestão operacional:

- Iniciar um serviço na aplicação 4POS para a função de validação saber que está na validade espacial e temporal correta e a relação entre a localização atual e a paragem.

d – Integração com sistema analítico de acessos:

- *Profiling* de clientes, produção de relatórios em Power BI. Relatórios em Power BI que agregam várias instâncias de 4BOOKING, e mostrando dados de localização de cada venda e validação em mapas.

Nos próximos capítulos detalhamos os desafios e detalhes de implementação de cada um destes tópicos, que consistem na principal contribuição deste trabalho.

Capítulo 4

Arquitetura

Este capítulo encontra-se omissa por motivos de confidencialidade.

Capítulo 5

Desenvolvimento e Implementação

Este capítulo encontra-se omissa por motivos de confidencialidade.

Capítulo 6

Data Profiling

Este capítulo encontra-se omissa por motivos de confidencialidade.

Capítulo 7

Considerações finais

Neste capítulo são apresentadas as conclusões desta dissertação. Na secção 7.1 é efetuada uma descrição sucinta do trabalho desenvolvido, realçando os objetivos atingidos. As limitações da solução desenvolvida e as ideias de trabalho futuro são descritas na secção 7.2.

7.1 Síntese e objetivos alcançados

Este trabalho surgiu essencialmente com o objetivo de melhorar a experiência dos turistas durante a sua deslocação pelos vários locais turísticos. Tendo em conta todas as tarefas que nos propusemos a realizar conseguimos observar que houve progresso em todas elas, mesmo tendo noção que com todos os imprevistos seriam objetivos difíceis de alcançar, por isso, acabamos por ficar satisfeitos com o resultado final.

Na aplicação 4POS a ser utilizada pelos funcionários das empresas de turismo para a venda de bilhetes de produtos turísticos tínhamos já uma implementação existente a ser utilizada à qual propusemos adicionar os seguintes pontos:

- Adição de um número único a cada bilhete gerado para venda;
- Geração de um QR Code a partir do número único de bilhete e impressão desse mesmo QR Code no bilhete;
- Validação de bilhetes através do QR Code gerado;
- Possibilidade de venda offline, que anteriormente não existia.

Todos estes pontos foram cumpridos com sucesso e estão a ser utilizados em contexto real.

No que diz respeito à aplicação 4POS a ser utilizada pelos turistas propusemo-nos a criar uma aplicação que cumprisse os seguintes requisitos:

- Acesso aos bilhetes das reservas em formato digital;
- Obtenção do tempo de espera para o transporte.

Ambos os pontos foram também realizados com sucesso, embora, até ao momento, a aplicação só tenha sido usada internamente pela Card4B, ou seja, ainda não temos a sua utilização em contexto real.

Quanto à parte da utilização da *Business Intelligence* para oferecer uma camada que tenha uma visão global da presença/procura de turistas numa cidade e oferecer mecanismos de recomendação aos sistemas operacionais individualmente ficamos parcialmente satisfeitos. Isto porque conseguimos obter uma visão global da presença de turistas, uma vez que conseguimos analisar os *inputs* produzidos pela aplicação 4POS a ser utilizada pelos funcionários das empresas de turismo para a venda de bilhetes. Contudo, quanto à parte de oferecer mecanismos de recomendação aos sistemas operacionais só será possível fazê-lo em termos práticos quando a aplicação 4POS a ser utilizada pelos turistas for utilizada em contexto real, o que acabou por não acontecer. Por este motivo, acabámos por não desenvolver nenhum mecanismo de recomendação que permitisse às empresas de turismo, tendo em conta essas recomendações, alterar os seus serviços.

7.2 Limitações e trabalho futuro

Embora tenham sido cumpridos os objetivos para a aplicação 4POS a ser utilizada pelos funcionários das empresas de turismo para a venda de bilhetes, foram encontrados alguns obstáculos durante o processo. Nomeadamente, devido a ser uma aplicação já com algum tempo deparámo-nos com alguns conflitos em termos da atualização de algumas partes da implementação do código. Para além disso, visto ser uma aplicação a ser utilizada ao mesmo tempo que estavam a ser feitos os desenvolvimentos tivemos que resolver diversos *bugs* que foram aparecendo, mesmo em partes da implementação que já estava desenvolvida. Assim, para trabalho futuro temos como objetivos a correção de eventuais erros que apareçam. Visto que apenas existe a implementação da aplicação para o sistema operativo *Android*, poderia ser feita a implementação para sistema operativo *IOS*, mas não é uma prioridade. Isto porque é uma questão do cliente adequar os equipamentos aos requisitos da aplicação, o que até agora não foi impeditivo para os clientes.

Relativamente à aplicação 4POS a ser utilizada pelos turistas uma das grandes questões em relação a esta aplicação é se esta vai ser utilizada pelos turistas, isto porque é uma aplicação que se for utilizada pelos turistas vai ser utilizada apenas durante o tempo em que estes se encontram em viagem, visto que depois deixa de ser útil, outra desconfiança por parte dos turistas pode ser o facto de pensarem que a aplicação vai utilizar os dados que guarda para de alguma forma extrair informação relativamente a eles para uso de terceiros. Este receio em relação à segurança da informação pessoal faz com que cada vez mais as pessoas se sintam apreensivas quanto à decisão de instalar uma aplicação. Posto isto é bastante importante garantir que a aplicação que estamos a oferecer vai trazer valor suficiente às pessoas para garantir que estas vão instalá-las no seu telemóvel e usá-las.

Outra limitação, como acontece anteriormente para trabalho futuro a implementação da aplicação para o sistema operativo *IOS*, sendo que neste caso esta preocupação é mais relevante, visto que muitos clientes utilizam o sistema operativo *IOS*.

Como já foi explicado anteriormente, o movimento dos turistas numa cidade não segue um padrão tão definido e constante como acontece com os habitantes numa cidade. De forma a perceber o padrão do movimento dos turistas num determinado momento foi desenvolvida a aplicação 4POS para turistas. Pretendemos futuramente adicionar ao sistema 4BOOKING um mecanismo que tendo em conta os dados recolhidos em tempo real pela aplicação, forneça recomendações para onde devem ser deslocados os transportes turísticos de forma a satisfazer as necessidades dos turistas ao momento. Para atingir este objetivo pretendemos criar mais relatórios no Power BI, desta vez tirando partido em grande parte da informação relativa aos turistas presentes nas paragens. Para além dos relatórios também se pretende criar *dashboards* que têm a grande vantagem de permitir ver rapidamente todas as informações necessárias para tomar decisões.

Bibliografia

- [1] Alonso, Gustavo, Fabio Casati, Harumi Kuno, and Vijay Machiraju. "Web services." *Web Services*. Springer, Berlin, Heidelberg, 2004, 123-149.
- [2] AltBeacon. <https://github.com/AltBeacon/spec>
- [3] Android Studio. <https://developer.android.com/studio/intro/>
- [4] Barten, Christa. "Smart Technologies in Tourism." Smart Technologies in Tourism Student Conference, 2016.
- [5] Behrisch, Michael, Dirk Streeb, Florian Stoffel, Daniel Seebacher, Brian Matejek, Stefan Hagen Weber, Sebastian Mittelstaedt, Hanspeter Pfister, and Daniel Keim. "Commercial visual analytics systems-advances in the big data analytics field." *IEEE Transactions on Visualization and Computer Graphics*, 2018.
- [6] Biblioteca Volley. https://github.com/codepath/android_guides/wiki/Networking-with-the-Volley-Library
- [7] Biblioteca ZXing. <https://opensource.google.com/projects/zxing>
- [8] Buhalis, Dimitrios. "Strategic use of information technologies in the tourism industry." *Tourism management*, 1998, 19.5: 409-421.
- [9] Cattell, Rick. "Scalable SQL and NoSQL data stores." *Acm Sigmod Record*, 2011, 39.4: 12-27.
- [10] Chappell, David. "Introducing windows azure." Microsoft, Dec 2009.
- [11] Chen, Hsinchun, Roger HL Chiang, and Veda C. Storey. "Business intelligence and analytics: from big data to big impact." *MIS quarterly*, 2012, 1165-1188.
- [12] Chen, Jianhua, Fang Miao, and Quanhai Wang. "SSL/TLS-based Secure Tunnel Gateway System Design and Implementation." 2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID). IEEE, 2007.
- [13] Christensen, Erik, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. "Web services description language (WSDL) 1.1.", 2001.
- [14] Dalmaso, Isabelle, Soumya Kanti Datta, Christian Bonnet, and Navid Nikaein. "Survey, comparison and evaluation of cross platform mobile application development tools." *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2013.
- [15] Dickinson, Janet E., Karen Ghali, Thomas Cherrett, Chris Speed, Nigel Davies, and Sarah Norgate. "Tourism and the smartphone app: Capabilities, emerging practice and scope in the travel domain." *Current Issues in Tourism*, 2014, 17.1: 84-101.
- [16] Distance Matrix API. <https://developers.google.com/maps/documentation/distance-matrix/intro>
- [17] Especificação AltBeacon. <https://github.com/AltBeacon/android-beacon-library>
- [18] Finžgar, Luka, and Mira Trebar. "Use of NFC and QR code identification in an electronic ticket system for public transport." *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*. IEEE, 2011.

- [19] Fused Location API. <https://developers.google.com/location-context/fused-location-provider/>
- [20] GDPR: Condições de consentimento. <https://gdpr-info.eu/art-7-gdpr/>
- [21] Guthrie, Scott D., and Dmitry Robsman. "ASP. NET HTTP runtime." U.S. Patent No. 7,162,723. 9 Jan. 2007.
- [22] Gowrishankar, S., N. Madhu, and T. G. Basavaraju. "Role of BLE in proximity based automation of IoT: A practical approach." *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. IEEE, 2015.
- [23] iBeacon. <https://developer.apple.com/ibeacon/>
- [24] IntelliJ IDEA. <https://www.jetbrains.com/idea/>
- [25] Kegalj, Hrvoje, and Dino Butorac. "Data access architecture in object-oriented applications using design partners." *Journal of information and organizational sciences*, 2003, 27.2: 81-91.
- [26] Lou, Tian. "A Comparison of Android Native App Architecture—MVC, MVP and MVVM.", 2016.
- [27] Microsoft Azure. <https://azure.microsoft.com/en-us/overview/what-is-azure/>
- [28] Mockus, Audris, Roy T. Fielding, and James Herbsleb. "A case study of open source software development: the Apache server." *Proceedings of the 22nd international conference on Software engineering*. Acm, 2000.
- [29] Mumbaikar, Snehal, and Puja Padiya. "Web services based on soap and rest principles." *International Journal of Scientific and Research Publications*, 2013, 3.5: 1-4.
- [30] ORM Lite. <http://ormlite.com/>
- [31] ORM Lite em Android. http://ormlite.com/sqlite_java_android_orm.shtml
- [32] Power BI. <https://powerbi.microsoft.com/en-us/>
- [33] QlikView. <https://www.qlik.com/us/products/qlikview>
- [34] Schuldt, Heiko. "Multi-tier architecture." *Encyclopedia of database systems*, 2009: 1862-1865.
- [35] Sendín-Raña, Pablo, E. Rodríguez-Fernández, Francisco J. González-Castaño, Enrique Costa-Montenegro, Pedro S. Rodríguez-Hernández, J. M. Pousada-Carballo, and Juan C. Burguillo-Rial. "Web-oriented business intelligence solution based on Associative Query Logic." *Software: Practice and Experience*, 2010, 40.9: 779-796.
- [36] Smutný, Pavel. "Mobile development tools and cross-platform solutions." *Carpathian Control Conference (ICCC), 2012 13th International*. IEEE, 2012.
- [37] Sobre o .NET. <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>
- [38] Sokolova, Karina, Marc Lemercier, and Ludovic Garcia. "Android passive MVC: a novel architecture model for the android application development." *International Conference on Pervasive Patterns and Applications*. 2013.
- [39] Soon, Tan Jin. "QR code." *Synthesis Journal* 2008, 2008: 59-78.
- [40] SQL Server Management Studio. <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>
- [41] Stinson, Douglas R. *Cryptography: theory and practice*. Chapman and Hall/CRC, 2005.

- [42] Tan, Wee-Kheng. "The relationship between smartphone usage, tourist experience and trip satisfaction in the context of a nature-based destination." *Telematics and Informatics*, 2017, 34.2: 614-627.
- [43] Volley API Client. <https://github.com/google/volley>
- [44] Yamaguchi, Akihiro, et al. "Beacon-based tourist information system to identify visiting trends of tourists." *Journal of Robotics, Networking and Artificial Life* 4.3 2017: 209-212.
- [45] Yeager, Nancy J., and Robert E. McGrath. *Web server technology*. Morgan Kaufmann, 1996.
- [46] Zilu Tian, Lou, Liguo and Joon Koh. "Tourist satisfaction enhancement using mobile QR code payment: An empirical investigation." *Sustainability* 9.7, 2017: 1186.