

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu



Mauro Loureiro Lima

Sistema Embebido de Baixo Custo para Fidelização
de Clientes

Dezembro de 2019

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Mauro Loureiro Lima

Sistema Embebido de Baixo Custo para Fidelização de Clientes

Tese de Mestrado

Sistemas e Tecnologias de Informação para as Organizações

Professor Doutor José Francisco Morgado

Professor Doutor Rui Pedro Duarte



Dezembro de 2019

A todos aqueles que de sempre acreditaram seria possível

A vocês Inês Bondoso, Pais, Orientadores e Amigos

Muito obrigado!

RESUMO

Nos dias de hoje, o marketing digital é, cada vez mais, uma realidade que surge da necessidade que as entidades têm em vender mais e melhor de modo a fazer face à concorrência. Surge, assim, a necessidade da evolução tecnológica neste âmbito, pois é de todo o interesse conseguir massificar a ideia de entrega de produtos ou serviços especializados consoante os interesses do utilizador. No entanto, existia um problema de identificação e localização de pessoas em locais fechados. É neste contexto que surge o *Beacon*. Em 2013 a reconhecida marca Apple lança o primeiro *beacon* do mercado, denominado de *iBeacon*. Rapidamente se percebeu que este era o caminho certo e bastante viável para a implementação de um marketing personalizado ou de proximidade. Nesse mesmo ano, foram espalhados *beacons* por vários estabelecimentos comerciais para oferecer informações personalizadas ao cliente. Outras aplicações foram desenvolvidas recorrendo à tecnologia *beacon*, desde localização de pessoas, navegação num determinado local e até mesmo sistema de ajuda de decisão.

Este projeto pretende divulgar que é possível criar um sistema de navegação *indoor* eficaz utilizando tecnologias atuais e com um baixo custo de implementação e manutenção. Neste documento foi também estudada uma forma de melhorar a fórmula de comunicação genérica entre o *beacon* e o dispositivo inteligente, pois em determinados ambientes verificou-se que nem sempre era tão precisa quanto esperado. Adicionalmente é importante também referir o método de navegação da aplicação, que tem como base um conceito bastante minimalista, desmaterializado a típica navegação num mapa, para um método baseado em navegação magnética, mas com o auxílio de *beacons* em vez do campo magnético terrestre.

Por último pretende-se também que a aplicação seja um exemplo de usabilidade e que permita o utilizador uma navegação simples, eficiente e permita ao utilizador atingir os seus objetivos: poupar dinheiro nas suas compras utilizando a aplicação *ShowMe*, desenvolvida neste trabalho.

No decorrer deste documento serão estudadas as várias aplicabilidades de um sistema multifacetado, que permitirá notificar os utilizadores sobre os seus interesses, permitir que estes efetuem trocas de promoções diretamente com a loja, com o objetivo de potenciar o seu interesse num produto específico, bem como garantir a navegação do cliente para a loja selecionada.

A presente investigação também irá contemplar um levantamento do estado da arte para conhecer outras soluções tecnológicas possíveis para o mesmo fim. Este estudo tornar-se-á uma mais-valia para comparações futuras e publicações na área. Será ainda efetuado um levantamento das várias soluções em que o *beacon* foi utilizado como base tecnológica, permitindo assim desenhar um sistema escalável, tendo em conta todos estes exemplos.

ABSTRACT

Nowadays and in order to cope with increasingly competitive markets, digital marketing has become an indispensable tool for any company to promote and sell high quality products. So for these to be shaped according to the user's interests and preferences, technology has been pushed to evolve every day trying to keep up with a global high demand.

However, there was a particular problem in identifying and locating people in indoors spaces. In 2013, Apple launches its iBeacon as the first beacon available in the market. Quickly this technology become an important step towards the implementation of a user's location tailored marketing. In that same year, multiple beacons were deployed into commercial buildings to provide customized information to its clients. Many applications were developed leveraging beacon's technology, from user's localization and indoor navigation to personalized assistants that would assist in the decision making process.

This project will disclose that it is possible to develop an efficient indoor navigation system with current technologies at low cost implementation and maintenance. This document also studies an improvement of the generic communication method, which, at the moment, and regarding certain conditions, proved lacking the precision that was expected. Additionally, it is important to note that the application unique navigation system map is based on a very minimalistic concept. It acquires knowledge from the classic magnetic navigation method but in this case it is supported by beacons instead of the actual earth's magnetic field. Finally, an application is developed (named *ShowMe*) as a proof of concept and is capable to provide a simple and efficient navigation system to the user with the benefit of reduced costs.

Throughout the document, several features are presented of what a versatile system can provide, from notifications about user's interests or promotion coupons of a store, making it easier and faster to sell a product or to navigate the user directly to a selected store.

This work will also survey the current state of the art technologies. Hopefully it will become a valuable resource for future research regarding this subject. To conclude, multiple solutions will be demonstrated where the beacon was used as base technology, allowing us to design a scalable system.

PALAVRAS CHAVE

Mestrado

Beacons

Localização interior

Aplicações móveis

Otimização

Emissores de sinal

Desenho centrado no utilizador

Navegação interior

KEY WORDS

Master
Beacons
Localization
Indoor
Mobile Applications
Optimization
Signal emitters
User's shaped design
Indoor navigation

AGRADECIMENTOS

Nesta última etapa do Mestrado foram encontradas várias adversidades e obstáculos durante a realização deste projeto, que apesar de superadas não o teria conseguido, ou pelo menos não tão bem, sem a ajuda de pessoas que direta ou indiretamente me acompanharam e apoiaram até ao fim deste desafio.

Por isso, quero começar por agradecer aos meus orientadores de Mestrado, professor José Morgado e professor Rui Pedro Duarte que, através de críticas construtivas e sugestões de modificações da aplicação fizeram com que chegasse a um produto final favorável.

Seguidamente e como não poderia deixar de ser, queria agradecer à minha namorada e companheira Inês Bondoso, que tanto me ajudou e me apoiou mesmo nos momentos mais difíceis, incentivando-me sempre dando uma opiniões de melhoria ao meu trabalho.

Quero ainda manifestar um carinho especial aos meus pais que sempre me ajudaram a manter o foco e me apoiaram em todos os sentidos para chegar ao fim de mais uma meta que tanto desejava.

Resta ainda agradecer a paciência de todos os meus amigos que sempre me apoiaram e me deram uma palavra certa no momento certo para conseguir finalizar este projeto com sucesso.

ÍNDICE GERAL

ÍNDICE GERAL	xiii
ÍNDICE DE FIGURAS	xvii
ÍNDICE DE QUADROS	xix
ABREVIATURAS E SIGLAS	xx
NOTAÇÃO.....	xxii
1. Introdução.....	1
1.1 Contextualização.....	1
1.2 Motivação	3
1.3 Contribuições principais	4
1.3.1 Comunicação da aplicação através da tecnologia BLE.....	4
1.3.2 Sistema de navegação	4
1.3.3 Satisfação do utilizador	5
1.4 Estrutura do documento	5
2. Estado da arte.....	6
2.1 Algoritmos de localização interior.....	6
2.1.1 Lateração	6
2.1.2 Fingerprinting	7
2.1.3 Cell Based.....	9
2.1.4 Angulação.....	10
2.1.5 Dead reckoning.....	10
2.1.6 K-Vizinhos mais próximos (k-NN)	11
2.2 Tecnologias inteligentes de localização interior	12
2.2.1 Wi-fi e Bluetooth.....	12
2.2.2 Near field communication (NFC).....	13
2.2.3 Rede de telecomunicações.....	14
2.2.4 Magnetómetro.....	14
2.2.5 Sensor de imagem.....	15

2.2.6	Giroscópio e acelerómetro	15
2.2.7	ZigBee	16
2.2.8	RFID.....	16
2.3	Sistemas relacionados.....	17
2.3.1	Indoo.rs	17
2.3.2	Estimote	18
2.3.3	Infsoft.....	19
2.4	Desenvolvimento de aplicações	20
2.4.1	Aplicações nativas.....	20
2.4.2	Aplicações Web	20
2.4.3	Aplicações híbridas	20
2.4.4	Aplicações Cross Platform.....	21
2.4.5	Comparativo de tecnologias de desenvolvimento.....	21
2.5	Enquadramento do trabalho.....	23
3.	Metodologia e planeamento	24
3.1	Metodologia.....	24
3.2	Planeamento	26
3.2.1	Primeiro semestre.....	26
3.2.2	Segundo semestre.....	26
4.	Requisitos.....	28
4.1	Stakeholders	28
4.2	Requisitos funcionais	29
4.3	Requisitos não funcionais.....	30
5.	Arquitetura do Sistema.....	33
5.1	Especificação geral do sistema.....	33
5.2	Arquitetura da aplicação.....	35
5.2.1	Comunicação entre utilizador e <i>beacon</i>	35
5.2.2	Navegação	36
6.	Implementação	38
6.1	Diagrama de desenvolvimento	38
6.2	Módulo de sistema de autenticação	39
6.3	Diagrama de classes	40

6.4	Base dados	40
6.5	Módulo de comunicação da aplicação	42
6.6	Módulo de navegação da aplicação	44
6.7	Módulo funcional da aplicação	46
7.	Conclusão e trabalho futuro.....	53
7.1	Conclusão.....	53
7.2	Trabalho futuro	54
	Referências	55
	Anexo 1	59
	Anexo 2	63
	Anexo 3	64

ÍNDICE DE FIGURAS

Figura 1: Comunicação <i>beacon/smartphone</i> (7)	2
Figura 2: Utilizador alertado com conteúdo do seu interesse (7)	3
Figura 3: Lateração (11)	7
Figura 4: Etapas do algoritmo Fingerprinting (13).....	8
Figura 5: Intersecção de comunicações	9
Figura 6: Posicionamento do utilizador com recurso ao algoritmo de Angulação 2D e 3D	10
Figura 7: Gráfico de representativo de um percurso efetuado pelo utilizador	11
Figura 8: Aplicação de navegação interior da indoo.rs	18
Figura 9: Módulo de proximidade e notificação da empresa Estimote	19
Figura 10: Ciclo de vida de cada requisito	25
Figura 11: Requisito com NOK e o respetivo descritivo.....	26
Figura 12: Arquitetura geral do sistema	34
Figura 13: Arquitetura método de comunicação da aplicação	35
Figura 14: Arquitetura método de navegação da aplicação.....	36
Figura 15: Funcionamento do método de rotação da seta da navegação.....	37
Figura 16: Diagrama de desenvolvimento.....	38
Figura 17: Diagrama de sistema de autenticação	39
Figura 18: Diagrama de classes	40
Figura 19: Modelo relacional da BD	41
Figura 20: Gráfico de curvas de medições	43
Figura 21: Mini mapa da aplicação ShowMe.....	45
Figura 22: Seleção de categorias de interesse	47
Figura 23: Notificação de promoções e listagem todas as lojas aderentes para geração de cupão.....	48
Figura 24: Geração de prémio e opções de uso disponibilizadas	48
Figura 25: Direcionar o utilizador para loja e possibilitar reclamar o seu prémio	49
Figura 26: Reclamar o prémio do cupão	50
Figura 27: Lista de prémios para troca e efetivação da troca	51
Figura 28: Lista “Meus descontos”	52

ÍNDICE DE QUADROS

Quadro 1: Stakeholders e as suas responsabilidades	28
Quadro 2: Amostras de medições de distância entre Beacon e o Smartphone.....	43

ABREVIATURAS E SIGLAS

IPV	Instituto Politécnico de Viseu
ESTGV	Escola Superior de Tecnologia e Gestão de Viseu
BD	Base de Dados
Wi-Fi	Wireless Fidelity
RF	Radiofrequência
BLE	Bluetooth Low Energy
RSSI	Received Signal Strength Indication
2D	Duas dimensões
3D	Três dimensões
LOS	Line of Sight (Linha de visão)
AP	Access Point (Ponto de acesso)
NFC	Near Field Communication
RFID	Radio-Frequency IDentification (Identificação por radiofrequência)
SDK	Software Development Kit
API	Application Programming Interface
NOK	Not OK
N/A	Não aplicável
TPower	Potência de transmissão
RGPD	Regulamento geral sobre a proteção de dados

NOTAÇÃO

a) Minúsculas latinas

r Raio

b) Minúsculas gregas

ϕ Teta

α Ângulo

c) Índices superiores gerais

tan Tangente

sin Seno

cos Cosseno

Coef Coeficiente

d Distancia

d) Símbolos

Σ Somatório

∞ Infinito

1. Introdução

Neste capítulo será efetuada um enquadramento da presente dissertação, contextualizando o problema, é identificada a motivação para este estudo, bem como as principais contribuições.

1.1 Contextualização

Com os avanços tecnológicos ao nível dos dispositivos móveis, a localização das pessoas, dispositivos e tudo o que nos rodeia passou a ser uma realidade. Assim, atualmente existem diversas formas de localização de pessoas ou dispositivos, sendo que a geolocalização (GPS) e o WI-FI são as mais comuns. No entanto, existem sempre limitações no uso de cada uma delas.

O GPS, totalmente operacional a partir de 1993 [1], é uma das principais tecnologias utilizadas neste âmbito, que carece de uma enorme limitação quando utilizado em espaços interiores. Esta tecnologia baseia-se em triangulação de satélites e emissão de ondas numa determinada frequência. Esta frequência, que para além de não se mover facilmente através de objetos sólidos, pode ser prejudicada por ruídos que existentes dentro dos edifícios [2].

A *Wireless Fidelity* vulgarmente designada de WI-FI é um método de comunicação sem necessidade de recorrer a cabos, é também uma tecnologia que pode ser utilizada como uma forma de localização indoor. Baseia-se na potência do sinal RF fornecido por terminais de rede que são captados por *routers* e que encaminham a informação para um servidor para estimar a localização do utilizador. Esta estimativa pode ser realizada tanto por algoritmos, como por mapeamentos de *routers* em relação ao espaço [3].

Esta tecnologia possui diversos constrangimentos a ter em conta, como por exemplo, velocidade contratada, quantidade de pessoas a utilizar a rede, qualidade do sinal, etc.

Partindo destes constrangimentos de alto nível identificados, tornou-se então premente o desenvolvimento de uma tecnologia que permitisse a criação de uma solução de baixo custo escalável para muitos utilizadores, e que funcionasse como “GPS indoor” [4].

Em 2013, a reconhecida marca Apple, apresenta o primeiro *beacon* do mercado. Esta tecnologia tem vindo a revelar-se, nos últimos anos, que é o futuro na comunicação indoor no que toca à localização de dispositivos ou pessoas [5]. Para além disso, tem um enorme potencial de utilização ao nível do marketing de proximidade. Um dispositivo *beacon* funciona como um transmissor de dados assente numa tecnologia já bastante consolidada: o Bluetooth. Cada *beacon* poderá ter uma série de conteúdos a si associados (localização, notificações, entre outros), e cada vez que existe contacto com outro dispositivo inteligente através de proximidade (como, por exemplo *Smartphone*), o *beacon* consegue disponibilizar esse mesmo conteúdo para o dispositivo móvel. Este conteúdo é sempre transmitido de forma indireta, ou seja, cada um dos *beacons* está mapeado numa Base de Dados (BD) com todo o tipo de informação que o administrador da aplicação desejar. Ainda em 2013, a Apple decide instalar cerca de 250 *beacons* nas suas lojas com o intuito de proporcionar aos clientes um sistema de notificações dos seus produtos, tanto ao nível de análise funcional e técnica do seu produto, bem como ofertas promocionais dos mesmos. Já no final de 2014, a Aruba Networks decide implementar, com sucesso, um sistema de navegação indoor, o "blue-dot", utilizando apenas a tecnologia *beacon*, oferecendo uma alternativa mais barata e viável do que a WI-FI, anteriormente utilizada [6].

Este projeto propõe-se centralizar os dois grandes tópicos abordados anteriormente numa única aplicação: o marketing de proximidade e a localização de dispositivos. Esta aplicação será capaz de efetuar um estudo dos clientes na proximidade com interesse em determinado produto e notificar o mesmo de promoções da sua loja, garantindo assim um marketing de proximidade através da troca de informação um para um (Figura 1).

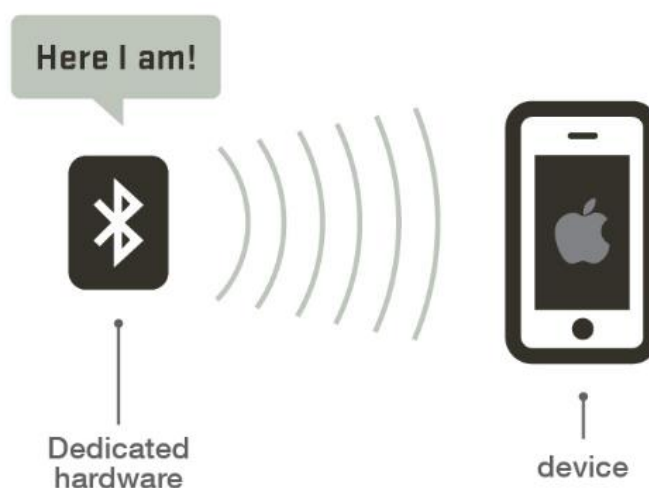


Figura 1: Comunicação *beacon/smartphone* [7]

Neste sentido, poderá ser afirmado que esta nova aplicação, não servirá apenas um cliente, mas sim uma série de entidades. Ao nível da entidade proprietária de um edifício, esta aplicação poderá ajudar a recolher dados estatísticos de quais os "pontos de interesses" mais visitados. Em relação à entidade intermédia do processo, por exemplo uma loja, poderá utilizar esses mesmos *beacons* para parametrizar notificações de campanhas de marketing, promoções de produtos, lançamento de novos produtos, bem como garantir o acesso fácil e rápido aos utilizadores de cupões de desconto na sua loja. Já para o consumidor final, esta aplicação irá identificar os interesses configurados por si mesmo, mantendo-o informado de tudo que se passa à sua volta com uma simples notificação (Figura 2). A funcionalidade de geração de cupões de desconto das lojas aderentes é também um outro incentivo ao uso da aplicação, garantindo que se não ficar satisfeito com o prémio gerado, poderá efetuar a troca por outro cupão compatível, disponibilizado pela loja.

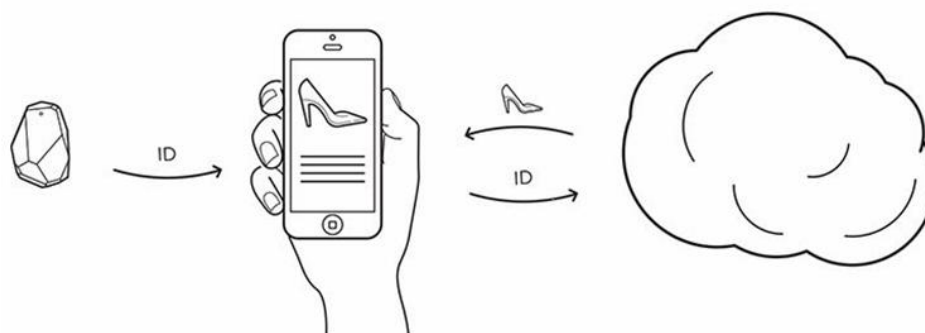


Figura 2: Utilizador alertado com conteúdo do seu interesse [7]

Esta aplicação será desenvolvida tendo sempre em consideração a aplicabilidade da mesma em diversos cenários, mantendo sempre a capacidade de notificar e recolher informação útil para todos os intervenientes: proprietário do espaço, proprietário de loja e cliente final.

1.2 Motivação

Com a concorrência a crescer de dia para dia, foram criados novos padrões de comércio mundialmente. A população está instruída para “Existe uma aplicação para isto”, ou seja, a sua necessidade pode ser colmatada por uma aplicação para um dispositivo móvel. Isto faz perceber que cada vez mais as empresas verificam que se já era essencial para uma marca ter um *website*, hoje em dia é tão ou mais importante ter uma aplicação para um dispositivo móvel, seja ela nativa ou *web responsive*.

Apesar desta grande procura, existem também imensas aplicações disponíveis, sendo assim é necessário garantir ao utilizador que a aplicação desenvolvida é inovadora e vai de encontro às suas necessidades, bem como, e cada vez mais importante, é segura e não expõe o seu conteúdo pessoal a qualquer entidade.

O facto da sociedade de hoje em dia, estar mais que habituada a usar o seu *smartphone* em qualquer lado, e que esteja sempre consigo, faz com que a aplicação desenvolvida neste projeto (denominado de *ShowMe*) tenha uma elevada probabilidade de aceitação por parte dos utilizadores, proporcionando-lhes uma ótima experiência comercial.

Assim, o objetivo do desenvolvimento da aplicação para além de notificar o utilizador sobre eventuais promoções, novos produtos, etc., pretende também auxiliar o utilizador a encontrar a loja que procura, funcionando como uma espécie de bússola interior. Com a implementação da aplicação *ShowMe* consegue-se unir as pessoas, e fazer com que estas possam ter um lado lúdico. Assim, esta aplicação irá conter um conceito de *gamification* que permitirá ao utilizador tentar a sua sorte na geração de cupões de desconto, bem como garantir que quando este não está totalmente satisfeito pode efetuar troca por outro cupão de desconto.

1.3 Contribuições principais

As principais contribuições deste trabalho podem dividir-se em três grandes contextos

- **Comunicação da aplicação através da tecnologia BLE**, cujo objetivo principal é garantir uma comunicação eficiente entre o dispositivo inteligente (telemóvel) e o dispositivo BLE (*Beacon*).
- **Sistema de navegação intuitivo**, permitindo de uma forma bastante simples fazer com que o cliente chegue rapidamente à loja pretendida.
- **Satisfação do utilizador**, considerando que esta forma de marketing é benéfica para o cliente, deverá garantir uma navegação intuitiva e potenciar o interesse na compra.

De seguida será efetuada uma descrição mais detalhada de cada uma das contribuições.

1.3.1 Comunicação da aplicação através da tecnologia BLE

Este é um dos três objetivos mais importantes desta dissertação, pois trata-se de demonstrar que a comunicação é feita de forma correta e eficiente pela aplicação desenvolvida. Como referido anteriormente, atualmente o mercado oferece diversas possibilidades de comunicação entre dispositivos, no entanto torna-se importante enquadrar o ambiente de utilização da aplicação, bem como demonstrar que é possível oferecer um serviço minimalista, mas eficiente, a todas as entidades envolvidas.

1.3.2 Sistema de navegação

Esta segunda contribuição já vai um pouco além da prova de conceito, pretende também que seja criado um sistema de navegação próprio. A aplicação *ShowMe* irá inicialmente mapear todos os *beacons* existentes no edifício posicionados num eixo (x, y) (configurado em BD) e garantir que dada a localização atual do dispositivo o utilizador tem de seguir para um determinado local através de um mapeamento do conceito de orientação, de forma a encontrar

a sua loja. Esta funcionalidade será semelhante às atuais bússolas recorrendo a *Beacons* e considerando a localização atual do dispositivo do cliente.

O método de navegação implementado deverá ser eficiente, pois é bastante importante para os espaços comerciais ou lojas, garantir que o cliente efetivamente é reencaminhado corretamente, assim como é importante proporcionar uma excelente e rápida experiência ao utilizador.

A aplicação deverá ajudar efetivamente de uma forma clara e direta a navegação do utilizador no espaço comercial, garantindo que não existe nenhuma segunda intenção de manipular a sua passagem por determinadas áreas.

1.3.3 Satisfação do utilizador

A terceira contribuição, passa por proporcionar uma agradável experiência de utilização da aplicação, garantindo que esta não seja vista como mais uma aplicação de *spam* de publicidade, mas sim como uma aplicação que conhece efetivamente o utilizador, garantido sempre a sua privacidade, e que vá colmatar necessidades de compras que o mesmo tenha.

Tal como referido anteriormente uma das grandes preocupações no mundo atual e a privacidade de dados, como tal a aplicação *ShowMe* garante que não é necessário efetuar qualquer registo para guardar as suas preferências, cupões, etc.

Resumidamente, a aplicação pretende proporcionar uma divertida ida às compras para qualquer cliente, de qualquer idade e tendo em conta os seus gostos.

1.4 Estrutura do documento

Este documento pretende refletir o trabalho realizado e encontra-se organizado da seguinte forma: No presente capítulo, Introdução é efetuada a contextualização da dissertação, bem como a motivação e principais contribuições. No capítulo dois, Estado da Arte, existe uma reflexão sobre sistemas semelhantes ao proposto e as tecnologias de localização e comunicação interiores. O terceiro capítulo, referente à Metodologia e Planeamento tem como principal objetivo descrever o método *Kanban* que auxilia todo o planeamento do projeto, bem apresentar a calendarização das tarefas. O capítulo quatro pretende relatar os requisitos funcionais e não funcionais do sistema. A arquitetura do sistema surge no quinto capítulo, onde é fundamentada e descrita detalhadamente a forma como o sistema está organizado e como funciona. No sexto capítulo, Implementação, é documentada a implementação da tecnologia BLE numa aplicação móvel, bem como detalhar a API e BD desenhadas nesta dissertação. No sétimo e último capítulo, a Conclusão, é realizada uma síntese e análise crítica ao presente documento e todo o trabalho nele descrito, bem como perspectivas de trabalho futuro.

2. Estado da arte

Este capítulo pretende descrever alguns algoritmos de localização existentes, bem como as tecnologias que permitam essa mesma localização de dispositivos inteligentes dentro de edifícios. Neste contexto, várias abordagens foram utilizadas ao nível dos algoritmos de localização *indoor* e várias tecnologias foram tidas em conta, que se encontram resumidas em [8].

2.1 Algoritmos de localização interior

Hoje em dia existem vários tipos de algoritmos de localização de dispositivos. Esta secção pretende descrever alguns deles, que são adequados para o desenvolvimento de aplicações.

2.1.1 Lateração

Lateração, ou *Lateration* em Inglês, é um algoritmo utilizado para determinar a localização de um dispositivo inteligente num espaço fechado [9] [10] [11]. Este algoritmo tem como base a distância entre um dispositivo, por exemplo *smartphone*, e três pontos denominados de pontos de âncora, que têm por base a posição de *beacons*.

Esta distância é usualmente fornecida pelo parâmetro *Received Signal Strength Indication (RSSI)* de cada *beacon* e será usada como raio de cada circunferência desenhada à volta da localização do *beacon* (Figura 3).

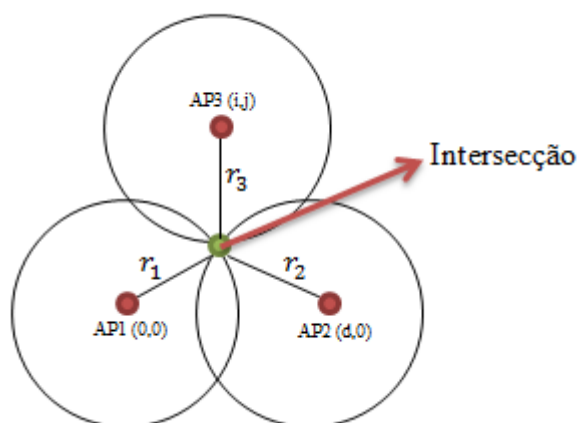


Figura 3: Lateração [11]

Pode-se então inferir que o algoritmo baseia-se na equação da circunferência:

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2 \quad (2-1)$$

Tendo em conta que o algoritmo necessita de pelo menos três *beacons* para fazer a triangulação, é aplicada a equação da circunferência em cada um deles, sendo que os valores x_i e y_i correspondem às posições de cada *beacon*. O valor r_i é o raio da circunferência dado pela distância fornecida pelo *RSSI*. O resultado deste sistema de equações permitirá conhecer o valor da intersecção dos círculos, que representa paralelamente a localização do *smartphone* no espaço *indoor*.

Este método foi aplicado com sucesso em diferentes contextos. Por exemplo, Yapeng et al. [9] aplicaram este algoritmo para analisar a distância entre dois dispositivos *Bluetooth*, tendo os autores obtido bom desempenho e com uma taxa de erro reduzida nos algoritmos aplicados.

Ainda com o base no algoritmo de lateração, Oguejiofor et al. [10] desenvolveram um novo algoritmo para determinar a posição de nós numa rede Wi-Fi, o resultado obtido foi bastante favorável e com uma margem mínima de erro de localização, apenas 0.74m.

Em [11], Nor Mahiddin et al., aplicaram o algoritmo de lateração como base para um sistema de localização de pessoas. Utilizando a força de sinal emitido por cada AP, o *smartphone* do utilizador através de uma aplicação, efetua o cálculo de conversão da força do sinal emitido pelo AP para uma distância real (em metros), importante citar que esta fórmula proposta no documento tem como base técnicas de lateração.

2.1.2 Fingerprinting

Este algoritmo é mais uma opção de localização de dispositivos *indoor* que tem por base a divisão do mapa de um determinado local em segmentos [12] [13]. É composto por duas fases: *online* e *offline*. O ponto de partida é a fase de treino (*online*) em que a cada segmento do mapa é atribuído um conjunto de propriedades únicas, tais como uma média de *RSSI*'s medidos nesta fase de treino, bem como a posição do segmento no mapa. Estes valores são,

tipicamente, armazenados numa Base de Dados (BD).

Depois desta primeira fase de treino estar definida, o algoritmo inicia a segunda fase, a de localização do dispositivo no determinado local. Esta fase, denominada de *offline*, recolhe exatamente os mesmos dados que na primeira fase, só que desta vez esses dados recolhidos são utilizados pelo sistema de localização para se efetuar uma comparação com aqueles armazenados na primeira fase. A melhor correspondência entre os dados da primeira fase e da segunda definirá a posição do utilizador num determinado *fingerprint* conforme apresentado na Figura 4.

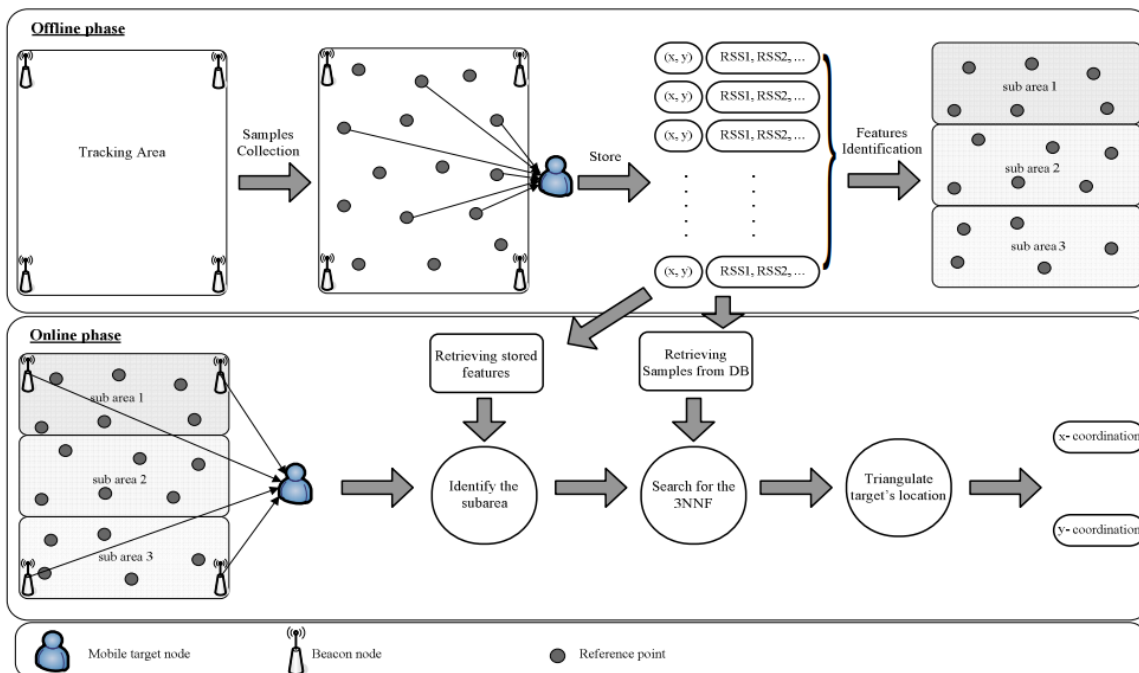


Figura 4: Etapas do algoritmo Fingerprinting [13]

Kaemarungsi [12] propõe um novo sistema modelar que pretende não só otimizar o tempo de implementação, bem como quantificar uma melhoria de precisão e rapidez com que o ajuste ao desempenho do algoritmo é feito. Estes ajustes são dos processos mais demorados durante a implementação do mesmo. O resultado deste estudo deixou os autores bastante satisfeitos, sendo que existem três vantagens das suas novas implementações: o tempo da fase de estudo foi reduzido significativamente pois não necessitam de armazenar tantos pontos de referência. A precisão do cálculo da localização é também um elemento bastante importante a reter, pois a margem de erro obtida no estudo e mínima, de 1 a 3 metros. A terceira contribuição passa por dividir as áreas de rastreio por subáreas o que pode também minimiza o erro do algoritmo. Em 2017 Xuxing et al [14], efetuaram um estudo com vista a melhorar a localização interior tornando-a híbrida mas ao mesmo tempo homogénea em qualquer ambiente. Este estudo usou o algoritmo *fingerprinting* mas com a particularidade da fase *online* do ponto de referência mais próximo ser encontrado dinamicamente. Uma vez mais concluiu-se que este algoritmo é efetivamente bastante válido para a localização interior, sendo que com o auxílio deste estudo

e adotando uma perspectiva, segundo os autores, híbrida do algoritmo a precisão da localização aumenta em cerca de 37% em relação ao método tradicional.

2.1.3 Cell Based

Este algoritmo tem como base a instalação de vários *beacons* num edifício com o objetivo de alcançar qualquer parte do edifício. Assim é possível garantir que em qualquer posição do utilizador no espaço comercial existe cobertura por um ou mais *beacons*. Depois de efetuada a instalação é iniciada uma fase de aprendizagem, que determinará que cada *beacon* garante cobertura de uma determinada área, sendo essa informação armazenada em base de dados.

Depois da fase de aprendizagem referida anteriormente, o sistema está preparado para a localização do utilizador. Esta localização é feita por base das interseções de sinal num determinado espaço [15], conforme apresentado na Figura 5. O utilizador (a vermelho), caso receba o sinal de ambos os *Beacon A* e *B*, é viável concluir que se encontra na zona de intersecção sinalizada a escuro.

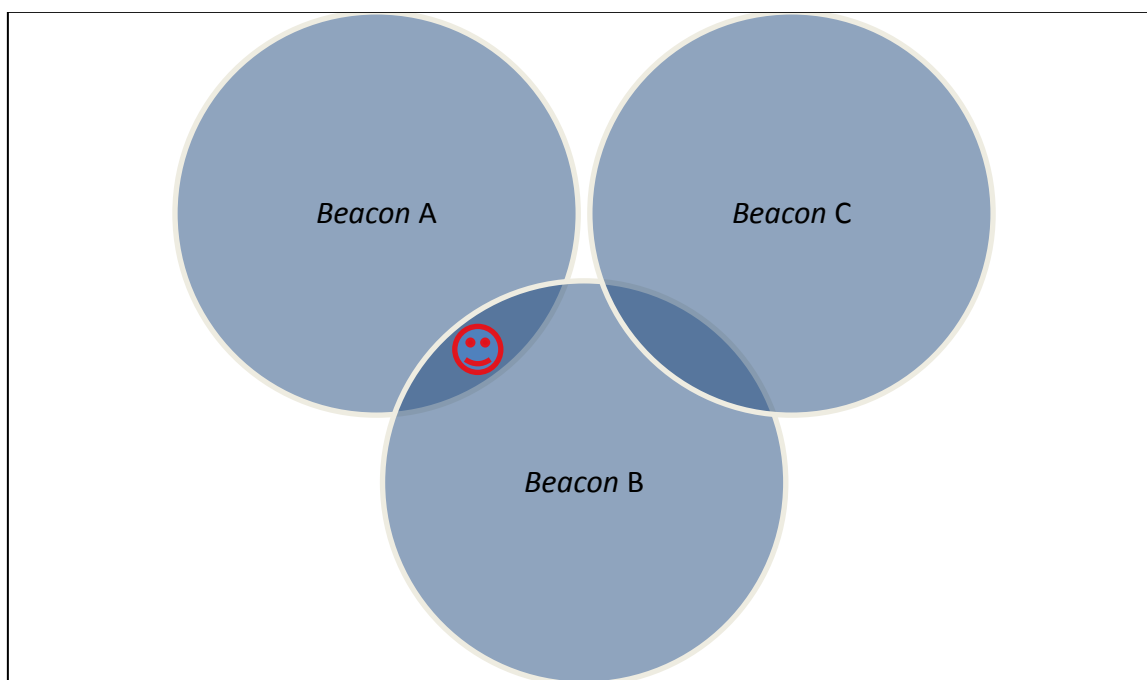


Figura 5: Intersecção de comunicações

Em 2008 foi publicado um artigo por Sudarshan Chawathe que propõe um sistema de localização de dispositivos móveis [15], utilizando um computador portátil ou telemóvel recorrendo a elipsóides. Como o próprio indica o método apresentado no artigo tem em conta disparidade da elipse que cada *Beacon* de forma a tornar o mesmo mais eficiente e preciso. O autor garante que é uma solução bastante simples mas funcional, no entanto indica que está em curso novos estudos de outros algoritmos com vista a melhorar a precisão do sistema.

2.1.4 Angulação

Este algoritmo tem como base o uso da geometria para desenvolver um método de localização [16], sendo um conceito bastante semelhante ao método de navegação desenvolvido nesta tese, que será descrito mais à frente.

Este modo de localização necessita de pelo menos dois pontos de comunicação com o *smartphone*, representados na Figura 6, por A e B. A localização do dispositivo inteligente (ponto P, representado na Figura 6) é feita tendo em conta a intersecção de um ou mais pares de linhas de direção angular, ou seja, é efetuado o cálculo do ângulo do Beacon A, por exemplo, em relação à posição do utilizador e é traçada uma linha reta. A mesma operação se repetirá para o Beacon B. A posição do utilizador será precisamente o ponto onde as duas retas se intercetam. É importante referir que estas variáveis são as mínimas para o funcionamento de um cálculo posicional numa visão 2D. Para um cálculo de posição num espaço 3D já são necessário pelo menos a comunicação com três *beacons* [17].

A grande vantagem deste algoritmo é o facto de estimar com alguma exatidão a posição do dispositivo utilizando poucos pontos de comunicação, no entanto é importante citar que a precisão vai-se degradando com o afastar do utilizador dos pontos de comunicação A e B.

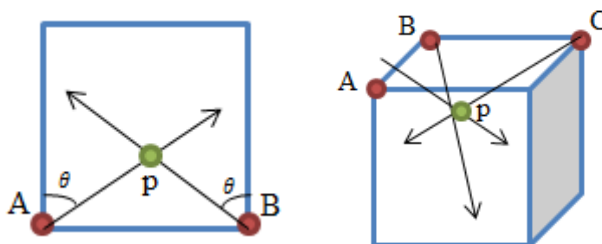


Figura 6: Posicionamento do utilizador com recurso ao algoritmo de Angulação 2D e 3D

Em [18] é proposta uma implementação de um método de identificação da posição do utilizador com base no algoritmo de angulação. Tal como os autores indicam existem vários fatores que influenciam a qualidade da precisão, tais como diferenças de temperatura, ruídos, etc.. O sistema desenvolvido conseguiu atenuar esses fatores e melhorar em cerca de 45% a precisão de distância, o que significa também uma melhoria considerável para o algoritmo.

Tal como indicado anteriormente este algoritmo foi o ponto de partida para o sistema de navegação desenvolvido nesta dissertação. De acordo com Hightower et al. [17], este algoritmo tem que, numa primeira instância, calcular o ângulo do *beacon* em relação ao utilizador. Sucintamente o cálculo do ângulo é efetuado do ponto de vista do utilizador para com o *beacon*, sendo essa a direção original para o mesmo se dirigir, sendo que o valor do ângulo vai sendo alterado consoante o movimento de deslocação do utilizador.

2.1.5 Dead reckoning

Este algoritmo pretende calcular a posição atual do utilizador com base em outras posições calculadas anteriormente. Resumidamente, este método tem três variáveis de entrada:

distância percorrida, ângulo de movimento e ponto de localização anterior do utilizador [19]. Estas três entradas permitem indicar a nova posição para qual o utilizador se dirigiu, ou seja, se este se encontra atualmente no P_2 e o ponto anterior que estava era o P_1 , percorreu uma distância de S com um ângulo de θ , conforme apresentado na Figura 7.

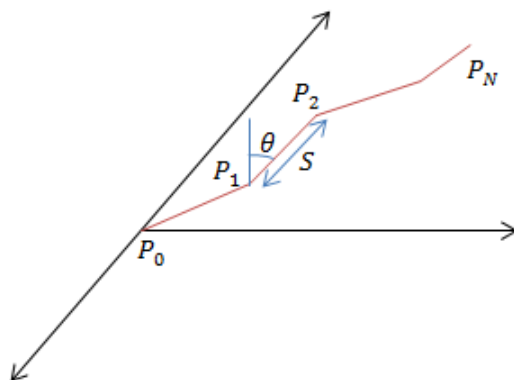


Figura 7: Gráfico de representativo de um percurso efetuado pelo utilizador

Neste contexto, G. Trein et al. [19] apresentam uma solução de partilha da trajetória do utilizador com o intuito de auxiliar os sistemas de mapas abertos a mapear, por exemplo, um determinado edifício. Este sistema de navegação e partilha de informação que tem como base o algoritmo *Dead reckoning*, que recebe toda a informação necessária para calcular a localização do utilizador e a distância percorrida através dos sensores de acelerômetro e do giroscópio do *smartphone*.

2.1.6 K-Vizinhos mais próximos (k-NN)

Este algoritmo é usado nas mais diversas áreas de conhecimento, sendo a classificação de galáxias, atribuição de crédito, navegação, avaliação de performance algumas das soluções onde o k-NN se encontra implementado [20] [21] [22].

Este algoritmo tem como método de aprendizagem a parença de dois objetos ou seres que pertencem a um mesmo conjunto de dados. A ideia é que se são próximos então têm características semelhantes. Existe um conceito humorístico sobre o funcionamento deste algoritmo designado como o teste do pato. “Se ele se parece com um pato, nada como um pato e grasna como um pato, então provavelmente é um pato” [23].

Tal como indica a citação anterior o objetivo deste algoritmo para ajudar a clarificar um novo objetivo passa por encontrar os seus vizinhos mais próximos e perceber a que grupo ou classe estes pertencem, ou seja, se por exemplo X tem k vizinhos próximos de si, onde a grande maioria é rotulada como Y, então X pertence a Y e quanto mais próximo forem mais certeza de pertencerem ao mesmo grupo se tem [23]. Esta distância é calculada tendo em conta a Equação 2-2:

$$D_{(x_i, x_j)} = \sqrt{\sum_{k=1}^n ((x_{ik} - x_{jk})) \times 2} \quad (2-2)$$

A desvantagem deste algoritmo é que apenas pode ser aplicado em casos em que todas as métricas são valores numéricos, e que efetivamente não é possível atribuir importância às variáveis da fórmula, ou seja, é dado o mesmo peso a cada uma das variáveis. Tal como referido anteriormente este algoritmo é bastante versátil e aplicado com sucesso em diversas realidades distintas, no entanto e tendo em conta o âmbito da dissertação, em Yasir e Ramanakopparapu [22] é comprovado que este pode também ser utilizado para sistema de localização de dispositivos *indoor*. Os autores propõem um método de localização de dispositivos com base no k-NN mas com uma pequena variante do algoritmo tradicional, em que consideram que o número ideal de vizinhos é avaliado por um método adaptativo próprio. Segundo os testes efetuados pelos autores a precisão de localização foi melhorada em comparação ao método original.

2.2 Tecnologias inteligentes de localização interior

Com o decorrer da rápida evolução tecnológica vivida nos últimos anos, surgem diversas tecnologias utilizadas na localização de dispositivos inteligentes dentro de edifícios. Os métodos apresentados na secção anterior podem ser usados recorrendo a diferentes tecnologias, que se encontram resumidas em [8].

2.2.1 Wi-fi e Bluetooth

Nos dias de hoje, todos os *smartphones* e *tablets* têm a funcionalidade de *Bluetooth* e *Wi-fi*, tornando-se apetecíveis de serem utilizadas em contexto de localização de dispositivos inteligentes dentro de edifícios [24].

O Bluetooth é um sistema de comunicação sem fios de curta distância, que foi criado com a necessidade de substituir a ligação por cabo entre dispositivos fixos e móveis. Esta tecnologia é bastante robusta, com bastante consumo de energia, tanto ao nível dos recetores como principalmente dos consumidores. A grande inovação do Bluetooth em relação a tecnologias anteriores é que esta não necessita de uma linha de visão (LOS – *Line of Sight*) para comunicar, contrariamente os infravermelhos que eram a tecnologia mais utilizada antes do aparecimento do Bluetooth [25].

O *Wireless fidelity* tipicamente designada por Wi-Fi, é uma norma padrão de comunicação baseada em rádio frequências na faixa dos 2.4 GHz e 2.5GHz [26]. Desde a sua criação standard existiu uma evolução significativa, tanto ao nível da velocidade de acesso, bem como ao nível das frequências radio de comunicação.

Este sistema consiste na instalação de vários pontos de acesso (Access Point – AP) estrategicamente instalados com o intuito de disponibilizar acesso a um maior número de pessoas e numa maior abrangência de espaço. Hoje em dia praticamente todos os dispositivos

móveis, como *smartphones*, telemóveis, computadores portáteis, etc. conseguem-se ligar a estes APs permitindo que comuniquem entre si e que tenham acesso à internet [27].

Tendo em conta essa realidade, foram desenvolvidas diversas soluções de localização de pessoas tendo por base estas duas tecnologias, como é possível verificar na secção 2.3. Assim, é criada uma teia de comunicação, espalhando emissores (dispositivos Wi-fi ou Bluetooth) num determinado edifício. Estes emissores emitem um sinal, o RSSI, que poderá determinar a posição de um utilizador num determinado local. A aplicação que trata os dados recebidos pelos emissores de sinal e define a posição do utilizador no local deve ter em conta o problema de variação de sinal [28]. Esta variação é bastante acentuada nestas duas tecnologias, e por isso deverá ser criada um método que atenua esta mesma variação.

Outro problema destas tecnologias, que ainda hoje está em estudo, é o ruído causado por outros objetos entre o emissor e o recetor [29]. Existem inúmeras causas no que toca às interferências de comunicações, desde a temperatura, humidade do ar, emissão de radio frequências de outros aparelhos. Todas estas variantes são importantes e influenciam a qualidade do sinal recebido. Assim, é importante que todos os sistemas tenham isto em conta de forma a filtrar o sinal antes de o mesmo ser processado. Nesta dissertação também se verificou algumas incongruências na precisão desse sinal, tendo sido desenvolvido um método de acerto que pode ser otimizado ao longo do tempo, ou seja, considerando mais amostras. Este método será descrito detalhadamente na secção 6.4.

No entanto, considerando todos estes pontos menos positivos, é imperativo dizer-se que são tecnologias bastante viáveis para a localização de dispositivos dentro de edifícios [30].

2.2.2 Near field communication (NFC)

O NFC é uma tecnologia sem fios de curto alcance e baixo custo. Tipicamente utilizada em pagamentos, partilha de informação e até mesmo identificação de objetos, através da troca de informação entre dispositivos com esta tecnologia e uma etiqueta *NFC*.

Em 2011 [31], foi apresentado um sistema de navegação baseado no *NFC*. Este sistema tem como paradigma a instalação de várias etiquetas *NFC* dentro do edifício em diversas localizações. De seguida, e quando o utilizador pretender consultar informações relativas, por exemplo, à melhor rota para uma determinada loja, basta que este aproxime o *smartphone* desta etiqueta e a aplicação irá processar a posição atual e indicar qual a rota que deve seguir. Este processo repete-se até que o utilizador chegue ao local pretendido.

Importante referir que a margem de erro desta tecnologia em relação a localização do utilizador é nula [32], no entanto nem todos os sistemas operativos móveis permitem, para já, tirar partido da mesma. Uma outra desvantagem deste sistema de navegação é o esforço do utilizador para usufruir em plenitude deste sistema, uma vez que é necessária uma proximidade bastante significativa para efetuar a leitura destas etiquetas.

2.2.3 Rede de telecomunicações

A tecnologia abordada neste tópico, que é já muito utilizada noutros contextos, como por exemplo realizar e receber chamadas funciona através de troca multidirecional de informação em ondas magnéticas entre o telemóvel e as antenas de captação/emissão de sinal [33].

Dado que cada antena tem um número limitado de emparelhamentos, para uma zona com bastante densidade populacional são instaladas várias antenas garantido a melhor comunicação possível.

Neste contexto, Otsason et al. [34] propuseram que esta tecnologia seja utilizada também para estimar a posição do utilizador, com base no algoritmo de *fingerprinting*. Os autores mapeiam 6 antenas a distância de 1,5 metros umas das outras, permitindo assim obter a posição de um utilizador num edifício fechado. Os escritores deste artigo, referiram que apesar de este método ser utilizado hoje em dia para a realização de chamadas telefónicas, não é possível aceder, através dos sistemas operativos dos *smartphones*, à força do sinal das redes de telecomunicação, tornando-se, portanto, uma não solução por fatores de incompatibilidade.

Ye Tian em [35] revelou ser afinal possível utilizar esta tecnologia para a localização interior, tal como os anteriores utilizaram esta tecnologia sincronizada com o algoritmo de localização *fingerprinting*, mas com a particularidade de utilizarem sensores do *smartphone* que permitam efetuar o acerto quando necessário. O autor indica que este sistema é capaz de se manter fiável com uma margem de assertividade superior a 80% durante meses sem qualquer intervenção, esta taxa de precisão, segundo Ye Tian, pode ser aumentada com técnicas de pós-processamento.

2.2.4 Magnetómetro

O magnetómetro é um dos vários sensores existentes nos *smartphones* da atualidade, tipicamente usado para indicar a direção para qual o dispositivo está orientado. No entanto, tal como o GPS, este sensor tem bastantes fragilidades na precisão das leituras efetuadas. Esta limitação acontece quando utilizado dentro de edifícios visto que grande parte destes são tipicamente construídos com bastante ferro e betão que prejudica e muito a sua precisão.

Tendo em conta estas limitações, foi proposto por Chung et al. [36] a utilização do algoritmo *fingerprint* para definir a localização do utilizador através desta tecnologia. As anomalias referidas anteriormente foram colmatadas por um mapeamento na fase *online* do algoritmo, descrito na secção anterior, permitindo estimar a posição do utilizador quando esta não é possível ser obtida pelo sensor.

Esta tecnologia também é bastante influenciada por todos os ruídos que se sentem num edifício, emissões de outros objetos eletrónicos, o que faz com que a localização do utilizador tenha ainda uma margem de erro médio bastante avultada 4,7 metros [37].

Já em [38] é apresentado o desenvolvimento de um sistema de localização interior que recorre ao uso de um microfone, sensor de luz e também do magnetómetro. Segundo os autores os resultados foram bastante positivos e precisos, indicando que a fórmula desse sucesso e a suas principais contribuições foram as de mostrar que o facto de conjugarem os valores dos três

sensores para a localização do utilizador, o que pode otimizar um sistema de localização, ou seja, a utilização de vários sensores em simultâneo poderá reduzir a margem de erro da localização exata de um dispositivo.

2.2.5 Sensor de imagem

Esta tecnologia tem como base a utilização da câmara do *smartphone* e é, atualmente, uma das formas de comunicação indoor mais permissória do mercado [39].

Fundamentalmente, esta tecnologia será bastante semelhante à comunicação *Beacon-Smarphone*, mas com frequências emitidas pelas lâmpadas que o sensor de imagem é capaz de captar. Assim, será necessário que a aplicação desenvolvida saiba interpretar esta frequência emitida pela lâmpada e a considere como um ponto âncora, bem como calcular o ângulo entre a câmara e a luz emissora. Com estas duas variáveis, depois de aplicado o sistema de cálculo de Angulação é sabido a posição do utilizador com uma margem média de erro de 1,5 metros [39].

Navarro et al. [38] fizeram uso deste sensor para o desenvolvimento de uma infraestrutura de localização *indoor*. Este sensor em particular tem como principal funcionalidade a monitorização da luz ambiente e, por isso, a proposta foi testada em várias estações do ano de forma a garantir que o sistema de localização e o sensor de imagem correspondiam em qualquer ambiente.

2.2.6 Giroscópio e acelerómetro

O giroscópio e o acelerómetro, tipicamente utilizados em conjunto, permitem, respetivamente, indicar qual a inclinação do *smartphone* no eixo (x, y, z) , bem como medir a aceleração não gravitacional do movimento do utilizador.

Neste âmbito, Trein, et al. [19] utilizaram os resultados destes dois sensores com o algoritmo de *dead reckoning* para obter a localização do utilizador. Tendo em conta que este algoritmo tem como variáveis de entrada a distancia percorrida e o angulo em que distancia foi realizada, os autores obtiveram esses valores da seguinte forma:

- O ângulo foi medido pelo sensor magnetómetro e pelo giroscópio e quando estes valores coincidem este seria considerado para a fórmula, caso contrário é ignorado e utilizado o ultimo angulo válido obtido da mesma forma.
- A distância seria inferida tendo em conta cada passo dado pelo utilizador, que passa sempre por uma fase de aprendizagem. O sistema tem capacidade de calcular que um passo da pessoa com X metros de altura equivale a Y metros de distância. Como consequência, sabe-se que N passos dado pelo utilizador multiplicado pelo valor de referência de um passo (Y) indicará a distância efetuada pelo utilizador. No entanto, os resultados foram bastante inconclusivos pois verificou-se que não foi possível calcular o caminho total percorrido pelo utilizador, fazendo mesmo com que estes sugerissem a utilização de outras tecnologias como por exemplo o *Bluethooth*.

Ladetto et al. [33] usaram um sistema semelhante, mas com objetivo diferente, de apenas de localização do utilizador que efetivamente resultou, mas com uma margem de erro bastante elevada, cerca de 10 metros [33].

2.2.7 ZigBee

O ZigBee contrariamente às tecnologias referidas até aqui não é uma tecnologia, mas sim um protocolo de comunicação. No entanto, e dada a sua vasta utilização em soluções de navegação e localização, foi considerado para referência neste documento.

Tal como dito anteriormente ZigBee é um nome de um conjunto de protocolos de comunicação feitas em rádio frequências de baixa potência, baseados no padrão IEEE 802.15.4 para redes pessoais [40].

Esta tecnologia de comunicação tem três componentes próprios para a comunicação, o coordenador, o *router* e o dispositivo final. O Coordenador é o responsável por criar uma rede própria, sendo ele o elemento raiz dessa rede e por isso o único dispositivo capaz de comunicar com outras redes. É também neste elemento que estão todas as configurações necessárias para a rede criada por si, desde chaves seguras de acesso, parâmetros que definem o número máximo de filhos e a profundidade da rede. Depois de toda a rede configurada os *routers* podem estimar o parâmetro Cskip, que indica qual o tamanho do conjunto de endereços filhos disponíveis na rede. O último interveniente deste sistema, é o dispositivo final do ZigBee, que tem apenas funções de troca de informação com o nó pai (coordenador ou *router*). Pode, por exemplo, ser uma lâmpada inteligente, um sensor de humidade, etc. [41]. Visto que em certas alturas este nó pode ficar adormecido, o seu tempo médio de vida é bastante elevado.

Em [42], Yao Zhao et al., propõem um algoritmo de localização eficaz com a utilização do ZigBee. Este algoritmo utiliza o indicador de qualidade da tecnologia, que está diretamente relacionado com a intensidade do sinal recebido, para adquirir distâncias entre os nós da rede e conseqüentemente dos utilizadores.

2.2.8 RFID

Esta tecnologia de identificação de elementos baseia-se em ondas eletromagnéticas e é vulgarmente utilizada para identificar produtos como, componentes, veículos, pessoas, máquinas, *assets*, serviços, caixas, paletes e contentores [43].

Dentro desta tecnologia existem dois tipos de *tags* (troca de informação entre o leitor e recetor): as passivas que são informações do objeto gravadas em etiquetas RFID e devidamente indexadas ao item, que tipicamente substituem o conhecido código de barras, e as *tags* ativas, que emitem ativamente o seu ID de identificação e todas as restantes informações adicionais. Apesar de um custo mais elevado em relação às *tags* passivas, as *tags* ativas têm a vantagem de terem um maior alcance de comunicação [44]. Assim, esta tecnologia permite a implementação de um sistema indoor de posicionamento e comunicação,

tipicamente baseado no algoritmo *fingerprinting* e com o auxílio de uma grelha de *tags* de referências. Estas *tags* são distribuídas por cada metro quadrado e têm a informação do local em que foram instaladas, permitindo intrinsecamente saber a posição do utilizador com uma margem de erro máximo de 2 metros [45].

Existe inúmeros sistemas de localização *indoor*. Lionel M. Ni et al. [45] apresentam um protótipo do sistema LANDMARC que utiliza a rádio frequência de identificação para localizar objetos dentro de edifícios. Esta aplicação, de acordo com os autores, tem a vantagem de melhorar o desempenho geral de localização de objetos utilizando o conceito de *tags* de referência, simultaneamente assinalam também que é uma solução viável e bastante económica.

Em [46] Ricardo O. Mitchell et al, apresentam um sistema de monitorização de peregrinos Muçulmanos à cidade de Meca. A densidade de pessoas, cerca de 3 milhões por ano, em movimento pode causar inúmeros problemas de segurança, então esta aplicação tem exatamente o propósito de efetuar a gestão de multidões, bem como localização de pessoas e serviços. Esta solução foi desenhada para ser utilizada no *smartphones* dos peregrinos e com o auxílio da tecnologia RFID.

2.3 Sistemas relacionados

Nesta secção serão apresentadas aplicações existentes atualmente no mercado no que toca à localização de dispositivos inteligentes indoor.

2.3.1 Indoo.rs

O indoo.rs é um sistema de navegação indoor (Figura 8) que tem como base um carregamento prévio da planta ou o desenho da mesma na aplicação. Podem ser carregados várias plantas de pisos diferentes no mesmo edifício. A aplicação que tem como base tecnológica o uso de *beacons* permite a pesquisa do caminho que o utilizador tem de percorrer para um determinado local desse edifício. Esta aplicação necessita de uma fase de aprendizagem, em que o gestor da aplicação define os pontos necessários para o percurso [47]. Esta aplicação usa um sistema de triangulação para o seu sistema de navegação, e segundo os próprios, o seu sistema de navegação é bastante preciso e pode ultrapassar chegar a uma precisão inferior a 5 metros.

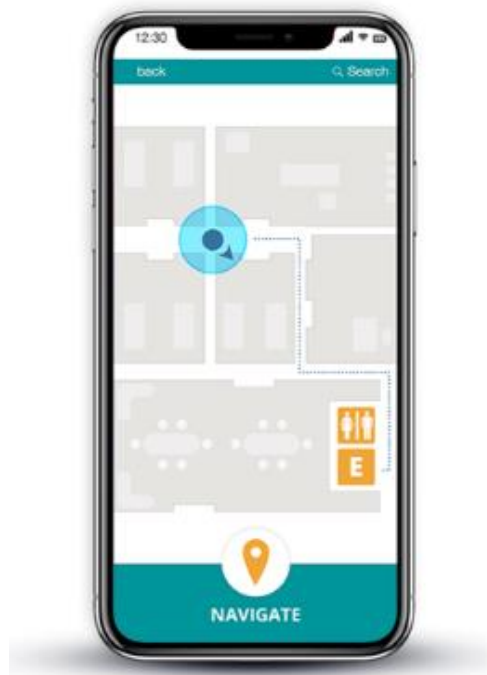


Figura 8: Aplicação de navegação interior da indoo.rs

Como será apresentado no capítulo 6.5, na aplicação desenvolvida no âmbito deste trabalho, este problema de carregamento prévio de mapas deixa de se colocar, uma vez que foi desenvolvido uma forma minimalista de navegação, sem mapas e apenas com uma seta que indica a direção que o utilizador deve tomar. Importante referir ainda que este método de navegação é automaticamente adaptado sempre que exista um novo *beacon* configurado em sistema, permitindo também uma escalabilidade grande desse ponto de vista.

2.3.2 Estimote

A *estimote* é uma empresa que oferece uma aplicação faccionada em módulos distintos: o módulo da proximidade, o de localização, o de rastreio de *assets* em tempo real e o de apresentar conteúdos de forma personalizado recorrendo à tecnologia *Beacon* [48].

Todos estes módulos são vendidos em separado, no sítio web da empresa. O módulo de proximidade, por exemplo, custa aproximadamente 100\$ e inclui a aplicação e quatro *beacons* (Figura 9).



Figura 9: Módulo de proximidade e notificação da empresa Estimote

Este módulo irá notificar o utilizador com uma mensagem em função da sua proximidade a um determinado *beacon*. No módulo de localização é representado, em tempo real, e numa determinada planta a localização do utilizador. O módulo de rastreio de *assets* é bastante semelhante a este último, mas funciona para *assets* em vez de pessoas e com a particularidade de haver hipótese de rastreio *indoor* e *outdoor*. O último módulo da *estimote* é o de representar conteúdos de forma personalizada num determinado ecrã, ou seja, consoante aquilo que o utilizador esteja interessado ou a visualizar no ecrã poderá aparecer informação adicional sobre esse produto.

De uma perspetiva mais técnica, tratando-se de um *software* proprietário, não foi possível perceber quais os algoritmos utilizados para a comunicação e navegação da aplicação.

2.3.3 Infsoft

A Infsoft, dentro dos vários produtos que possui, oferece um sistema de posicionamento, rastreio e navegação indoor com Wi-Fi. Este sistema que, como os próprios assumem, tem uma margem de erro de 5 a 15 metros, é maior que a margem da solução usando *beacons* [49].

Esta solução passa pela instalação de *hotspots*, routers ou um outro dispositivo capaz de comunicar por internet, num determinado edifício. Utilizando o RSSI e o endereço MAC emitido por cada um dos dispositivos anteriormente referidos a aplicação da Infsoft, utilizando o algoritmo *fingerprinting*, consegue calcular a posição do utilizador. Oferecem ainda um sistema de navegação, mas que por Wi-Fi está apenas disponível para *smartphones* com sistema operativo *Android*.

Esta empresa tem vários casos de uso em locais como aeroportos, hospitais, escritórios, etc., no entanto não foi possível verificar o preço da solução pois a mesma é orçamentada de caso para caso.

2.4 Desenvolvimento de aplicações

Nesta secção serão enumeradas as várias soluções com que se podem desenvolver aplicações móveis, bem como efetuado um comparativo entre as mesmas.

2.4.1 Aplicações nativas

As aplicações nativas são desenvolvidas especificamente para um sistema operativo móvel utilizando um SDK (*Software Development Kit*) e API (*Application programming Interface*) da própria plataforma, bem como as suas linguagens de programação: em *Android* a linguagem de programação utilizada é o Java e em iOS é o objective-C ou mais recentemente o Swift [50] [51].

Estas ferramentas e linguagens de programação permitem que o programador durante o desenvolvimento das suas aplicações sinta que pode tomar um melhor partido de performance e de funcionalidades do sistema operativo e do *smartphone*.

No entanto existe o reverso destas vantagens, pois se o objetivo do programador ou empresa é estender a aplicação a um maior número de pessoas, têm de ser desenvolvidas duas aplicações de raiz, o que faz com que os custos e o tempo de desenvolvimento sejam consideravelmente maiores.

2.4.2 Aplicações Web

As aplicações web são desenvolvidas utilizando tecnologias recentes como o HTML5, CSS e o Javascript, este último pode recorrer estar encapsulado em livrarias atualmente muito poderosas como o AngularJS ou ReactJS [52].

A maior vantagem da utilização desta tecnologia é a compatibilidade entre plataformas, sendo que as aplicações desenvolvidas usando estas tecnologias tanto podem ser utilizadas em *smartphones*, como em *tables*, como em computadores, pois são acedidas através de um *browser*. Assim sendo, a ligação à internet é obrigatória para o funcionamento da aplicação. A par disso verifica-se também que existem limitações no acesso a funcionalidades específicas do dispositivo, como por exemplo os contactos, tornando-se assim uma desvantagem bastante significativa em aplicações um pouco mais complexas.

2.4.3 Aplicações híbridas

Os vários *SDKs* nativos referidos anteriormente contêm uma *Web View* capaz de interpretar e mostrar conteúdo HTML. É através desta vista que as aplicações híbridas fornecem os seus

conteúdos aos utilizadores. Esta vista é uma aplicação web executada numa API nativa, que são desenvolvidas em HTML, CSS e *Javascript*, e em certos casos podem usar bibliotecas compatíveis apenas com os *smartphones* e *tablets*, como por exemplo *JQuery* mobile [53]. Estas aplicações têm a vantagem de ser compatíveis com todos os sistemas operativos móveis com a particularidade de ter acesso a algumas funcionalidades nativas do sistema operativo e *smartphone*. Este acesso é possível pois existe uma *framework* intermediária a fazer a interpretação das chamadas e respostas aos conteúdos nativos, podendo torna-se um problema ao nível de performance em aplicações mais complexas.

2.4.4 Aplicações Cross Platform

A necessidade de haver uma solução compatível com múltiplas plataformas, mas sem perder capacidade de performance fez com que recentemente surgissem várias aplicações designadas de *Cross-platform*. Estas aplicações possibilitam ao programador utilizar todas as funcionalidades nativas e ao mesmo tempo serem compatíveis com vários sistemas operativos sem comprometerem a performance e a viabilidade [54].

Tratando-se de uma tecnologia recente e com bastante mercado nos dias de hoje [55] foi a escolhida para o desenvolvimento da aplicação *ShowMe*, sendo desenvolvida totalmente em *Xamarin Forms* usando a linguagem de programação C#.

Existem inúmeras aplicações de sucesso desenvolvidas em *Xamarin*, como por exemplo a aplicação CA Mobile do banco Crédito Agrícola e a aplicação da transportadora internacional MRW [56].

2.4.5 Comparativo de tecnologias de desenvolvimento

Considerando o referido nas secções anteriores, torna-se importante estabelecer um comparativo entre as varias tecnologias e qual (ou quais) as melhores tecnologias relativamente a aspetos como o custo, portabilidade, acesso a funcionalidades nativas, consistência da interface, distribuição e performance (é possível também consultar esta análise em forma de tabela no Anexo 2).

Custo e portabilidade

Nativa: Elevado, quando se pretende desenvolver para mais que um sistema operativo.

Web: Médio, pois apesar de se tratar de uma tecnologia bastante abrangente no mercado é necessário um grau de performance da aplicação bastante elevado e por isso necessário programadores com bastante background.

Híbrida: Médio/Baixo, pois trata-se de uma tecnologia com bastante abrangência de mercado e compatível com vários sistemas operativos.

Cross Plataforma: Baixo, pois a para além da compatibilidade com os vários sistemas operativos é também esta compatível com diversas linguagens de programação (no caso específico do *Xamarin*).

Funcionalidades nativas

Nativa: Acesso a todas as funcionalidades do sistema operativo e *smartphone*.

Web: O acesso a estas funcionalidades é bastante restrito.

Híbrida: O acesso às funcionalidades nativas depende sempre da *framework* em utilização e por isso poderá haver limite de algumas funcionalidades em detrimento de outras.

Cross Plataforma: Do que foi possível experienciar todas as funcionalidades nativas são facilmente acessíveis. Paralelamente não foi encontrado nenhum relato de limitações.

Consistência da interface

Nativa: Nas aplicações existem componentes de interface uniformizados por sistema operativo.

Web: As *frameworks* web permitem alcançar uma interface bastante semelhante à nativa.

Híbrida: As *frameworks* web permitem alcançar uma interface bastante semelhante à nativa.

Cross Plataforma: As aplicações desenvolvidas nesta tecnologia permitem alcançar uma interface bastante semelhante à nativa.

Distribuição

Nativa: As lojas de aplicações móveis têm benefícios no que toca ao *marketing* customizado, no entanto existe uma série de restrições que têm de ser compridas para ter a aplicação disponível em loja.

Web: Não tem qualquer restrição no entanto também não tem qualquer benefício de *marketing*.

Híbrida: As lojas de aplicações móveis têm benefícios no que toca ao *marketing* customizado, no entanto existe uma série de restrições que têm de ser compridas para ter a aplicação disponível em loja.

Cross Plataforma: As lojas de aplicações móveis têm benefícios no que toca ao *marketing* customizado, no entanto existe uma série de restrições que têm de ser compridas para ter a aplicação disponível em loja.

Performance

Nativa: O código nativo acede diretamente a todas as funcionalidades do sistema operativo e do *smartphone* e por isso a sua performance é notória.

Web: Performance pode ser um problema quando existe uma má ligação à internet.

Híbrida: Quando se trata de uma aplicação complexa o facto de existir um sistema de multicamadas pode prejudicar a performance da aplicação.

Cross Plataforma: Apesar de não ser uma aplicação com acesso nativo a todas as funcionalidades a interpretação do acesso às mesmas não é sentida pelo utilizador. Existe por exemplo, a aplicação CA Mobile, implementado nesta plataforma tecnologia e com um feedback bastante positivo por parte dos utilizadores.

Assim é possível concluir que tanto as aplicações nativas como as aplicações *Cross Platform* são soluções bastante válidas e com benefícios semelhantes, assim cada caso deve ser estudado em particularmente no que toca à decisão da tecnologia a abordar. Neste projeto em específico e tratando-se de apenas um programador a questão da eventual portabilidade foi um ponto fulcral no momento da escolha.

2.5 Enquadramento do trabalho

O objetivo da presente dissertação é o desenvolvimento de uma aplicação que facilmente seja escalável e com um componente técnica bastante própria.

Considerando o estado da arte apresentado anteriormente, verificou-se que existe diversas soluções de localização e sistemas já desenvolvidos mas que funcionam de formas bastante independentes. Tomando como exemplo as soluções da *estimote* para a aplicação ter o módulo de navegação e proximidade ativo será necessário dois tipos de *beacons* diferentes no mesmo espaço.

A *ShowMe* pretende centralizar todos os serviços que possam ser oferecidos pelos beacons, apenas com um tipo de *beacon* e numa só aplicação. Esta aplicação foi desenhada não só a pensar nas funcionalidades atuais, bem como nas futuras e por isso é possível afirmar que todas as componentes até agora estudadas nesta dissertação (notificação, navegação, vista em tempo real da localização do utilizador e rastreio de *assets*) são possíveis incorporar na aplicação futuramente. O facto de a aplicação não ter um Design direcionado para uma área de negócio específica, pois todas as imagens e *icons* são configuráveis no servidor, torna-a escalável, como pequenos reajustes, a outros negócios, como por exemplo, auxílio e navegação num museu, navegação e gestão de *stocks* de um armazém.

De uma perspetiva mais tecnológica a presente dissertação descreve uma aplicação que otimizou e recriou formas de comunicação entre o dispositivo inteligente e os emissores de sinal, neste caso em particular os *beacons*, inovando ainda no método de navegação indoor com o auxílio do *smartphone*.

3. Metodologia e planeamento

O presente capítulo pretende descrever a metodologia de trabalho utilizada neste, bem como o planeamento ao nível de *timings* de cada requisito.

3.1 Metodologia

Este projeto, tal como a maioria dos projetos IT, adotou uma metodologia ágil [57]. Dentro das várias metodologias ágeis existentes, o *Kanban* foi a escolhida pois para além de ser rapidamente implementada, tem princípios adequados ao projeto, que são:

- Visualizar facilmente o fluxo de trabalho;
- Limitar o trabalho em curso;
- Tendo em conta cada coluna do *Kanban*, é bastante claro através de uma simples consulta qual o ponto de situação de cada requisito;
- Conhecendo o fluxo *Kanban* permite medir mais eficazmente os *timings* necessários para cada tarefa;
- Permite ao final de cada conjunto de requisitos avaliar processos de melhoria para os próximos requisitos, validando quais as maiores falhas anteriores.

Esta metodologia de trabalho é ótima não só para visualmente se verificar o ponto de situação do projeto, em relação a uma eventual previsão inicial, bem como para identificar qual ou quais os requisitos que estão atualmente a ser trabalhados [58]. Apesar desta metodologia permitir que sejam atribuídas tarefas para o mesmo período temporal, permite também configurar um limite máximo de requisitos por programador.

Esta regra é uma regra fundamental nesta metodologia pois se efetivamente a tarefa está marcada como iniciada este evento deverá realmente acontecer.

O *Kanban* quando utilizado corretamente, isto é, com todas as etapas bem definidas e claras para todos os intervenientes, torna a tarefa de prever o fim do requisito mais fácil, pois com a experiência da utilização desta metodologia de trabalho os tempos gastos em cada etapa vão ser mais precisos.

Por último, mas não menos importante, esta metodologia permite no final de cada conjunto de requisitos fazer uma introspeção e verificar o que é possível melhorar, tanto ao nível de previsões, como ao nível de desenvolvimento e até mesmo adequar o ciclo de vida de um requisito, aumentando, alterando ou removendo etapas definidas na metodologia, “quadros” *Kanban* (na Figura 10 apresenta-se o *Kanban* associado ao ciclo de vida de cada requisito neste projeto).

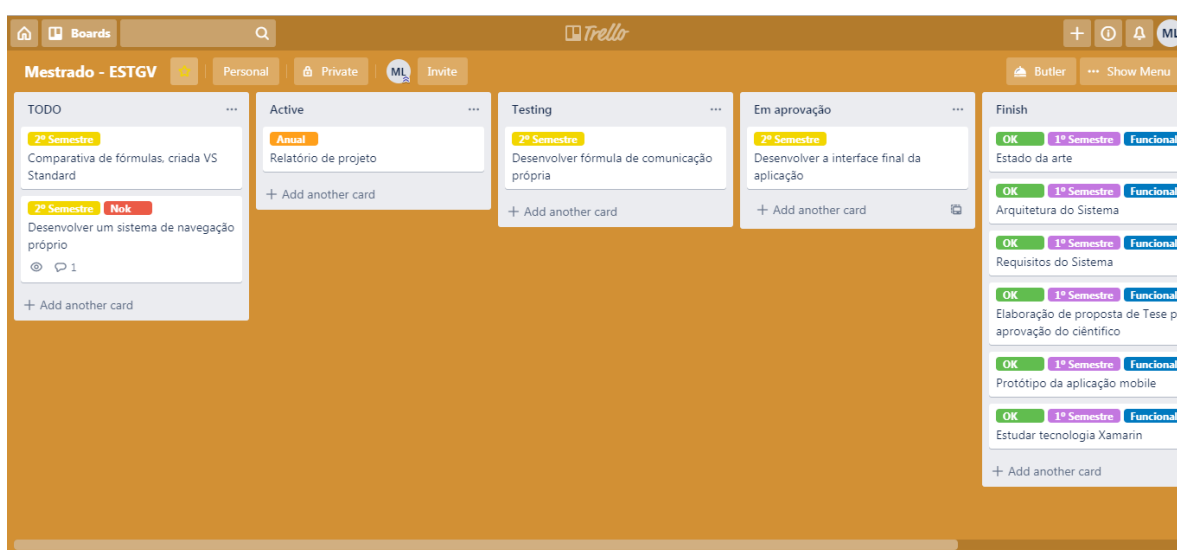


Figura 10: Ciclo de vida de cada requisito

Como é possível visualizar neste projeto optou-se por apenas cinco momentos para um requisito, visto que se trata de um trabalho individual esta gestão acabar por ser mais fácil.

O primeiro “quadro”, com o nome *TODO*, indica requisitos que ainda não foram iniciados. O segundo, *Active* são os requisitos que estão neste momento a ser trabalhados, aqui é importante colocar um nome genérico, pois nem tudo que se faz num projeto IT é desenvolvimento. A terceira etapa garante os testes antes da conclusão/aprovação final do requisito.

O quarto passo, quando aplicável, significa que o requisito está em aprovação de terceiros, por exemplo a ser validado pelo(s) orientador(es) ou a ser aprovado pelo Conselho Técnico-Científico da ESTGV. O último ponto do ciclo de vida de um requisito é a conclusão do mesmo.

Importante citar que cada requisito está rotulado com etiquetas, que definem a sua data e sua tipologia (por exemplo um requisito funcional não precisa de passar por testes unitários). Paralelamente, e para que se entenda que este ciclo de vida não é unidirecional, é importante referir que um requisito que esteja em aprovação ou em fase de testes e que não seja validado

com sucesso, poderá voltar para o TODO, com comentários alusivos ao porquê deste retrocesso e com a “etiqueta” – NOK (Figura 11).

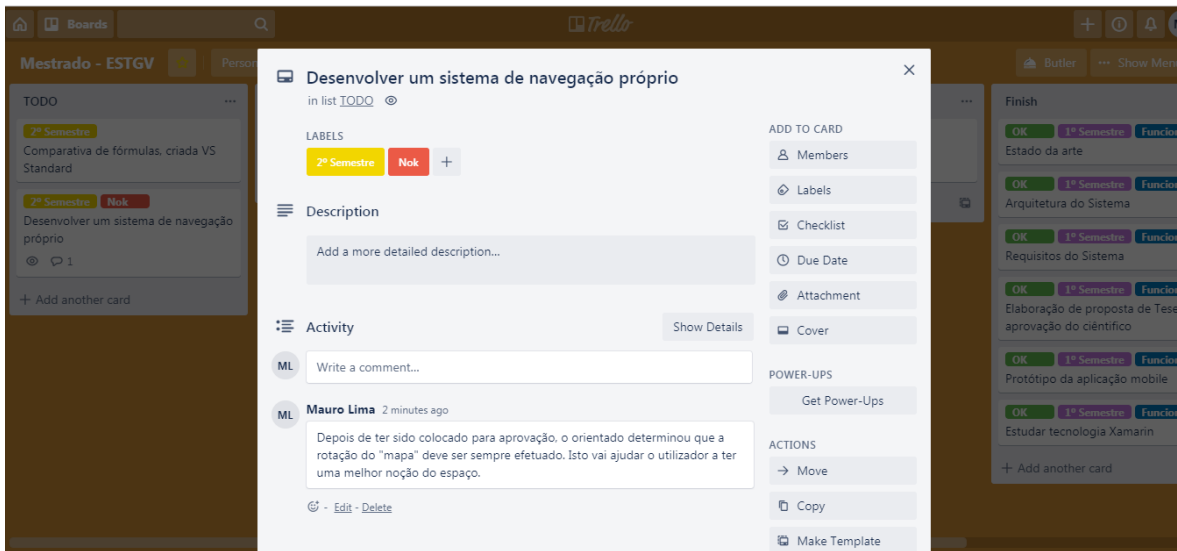


Figura 11: Requisito com NOK e o respetivo descritivo

3.2 Planeamento

O planeamento deste projeto foi dividido em dois semestres letivos. O primeiro semestre compreendido entre os meses de Novembro 2018 a Abril de 2019 e o segundo semestre de Abril de 2019 a Novembro de 2019.

3.2.1 Primeiro semestre

Neste primeiro semestre foi definido que o aluno iria ter em conta as seguintes tarefas:

- Estado da arte;
- Definir a arquitetura do sistema;
- Definir os requisitos do sistema;
- Apresentar uma proposta de Tese de mestrado para aprovação do Conselho Técnico-Científico da ESTGV;
- Protótipos da aplicação móvel;
- Estudar a tecnologia *Xamarin*, utilizada para o desenvolvimento da aplicação;

3.2.2 Segundo semestre

No segundo semestre foi definido que o aluno iria ter em conta as seguintes tarefas:

- Identificar eventuais problemas da fórmula de comunicação e implementar um sistema de melhoria à mesma;

- Desenvolver um sistema de navegação simples e eficiente para a aplicação;
- Desenvolver uma interface final intuitiva e apelativa para o utilizador;
- Terminar o relatório de trabalho;

4. Requisitos

Este capítulo pretende abordar os requisitos do sistema, funcionais e não funcionais. Para um melhor enquadramento de cada requisito é também efetuado um levantamento dos *stakeholders* da aplicação *ShowMe*.

4.1 Stakeholders

Os *stakeholders* são agentes que interagem de forma direta ou indireta com o sistema [59]. De uma forma geral estes podem ser englobados em:

- Responsáveis pelo desenvolvimento do produto
- Financiadores do produto
- Utilizadores do produto

De entre os *stakeholders* identificados, um produto não necessariamente pressupõem a participação de todos. No caso deste projeto aqui descrito, não existem financiadores para o produto. Neste projeto, os agentes são apresentados no quadro 2.

Quadro 1: Stakeholders e as suas responsabilidades

Stakeholders	Descrição	Responsabilidade
Orientadores do Projeto	Prof. Doutor José Morgado; Prof. Doutor Rui Duarte	Debate sobre os requisitos funcionais e não funcionais da aplicação. Monitorizar e aprovar os processos de desenvolvimento e utilização
Desenvolvedor do produto	Mauro Lima	Responsável por implementar e descrever a aplicação

Utilizador	Clientes da superfície comercial que utilizem a aplicação <i>ShowMe</i>	N/A
------------	---	-----

4.2 Requisitos funcionais

Os requisitos funcionais pretendem descrever as funcionalidades exatas do sistema, bem como documentar comportamentos do sistema perante uma qualquer interação [60].

Depois da definição dos objetivos da aplicação e do problema estar bem clarificado iniciou-se o levantamento dos requisitos funcionais da aplicação *ShowMe*. Tipicamente esta fase é discutida com o cliente, no entanto, como não é aplicável neste caso, este debate foi feito entre o aluno e os orientadores. Assim sendo foram identificados os seguintes requisitos funcionais:

Permitir a escolha de categorias favoritas

Descrição Geral: A aplicação deverá permitir ao utilizador escolher quais as categorias do seu interesse.

Stakeholder(s): Utilizador da aplicação

Pré-Condição: N/A

Fluxo de eventos: 1. Entrar na aplicação; 2. (opção A) Caso seja a sua primeira entrada na aplicação este menu será apresentado de imediato e terá de escolher a(s) sua(s) categoria(s) favoritas; 2. (opção B) Caso não seja a primeira entrada na aplicação, clicar no botão Conta para aceder à sua listagem e editar as suas categorias.

Notificações de alerta

Descrição Geral: A aplicação deverá notificar o utilizador de mensagens configuradas pelas lojas com novos produtos/promoções/etc.

Stakeholder(s): Utilizador da aplicação

Pré-Condição: O utilizador deverá ter sempre configurado as suas categorias favoritas, pois é com base nestas que receberá as notificações.

Fluxo de eventos: 1. Permitir a execução da aplicação em *background*.

Gerar prémio e navegar para a loja

Descrição Geral: A aplicação permite o utilizador gerar um cupão de desconto (prémio) das lojas aderentes à aplicação *ShowMe* e levar o utilizador até a loja escolhida para reclamar o seu prémio ou então armazenar o mesmo.

Stakeholder(s): Utilizador da aplicação;

Pré-Condição: N/A

Fluxo de eventos A: 1. O utilizador deverá clicar no menu desconto de lojas; 2. Escolher uma das lojas listadas; 3. Clicar no botão para gerar prémio; 4. Voltar atrás e o cupão ficará armazenado na sua listagem;

Fluxo de eventos B: 1. O utilizador deverá clicar no menu desconto de lojas; 2. Escolher uma das lojas listadas; 3. Clicar no botão para gerar prémio; 4. Navegar para a loja; 5. Clicar no botão de reclamar o prémio; 6. Apresentar o copão a um colaborador da loja; 7. O colaborador visualiza o cupão dentro da aplicação e efetua o desconto indicado no cupão. 8. O Cupão é removido automaticamente da lista de cupões do utilizador, depois deste o gerar para mostrar ao colaborador;

Fluxo de eventos C: 1. Clicar no botão para listar os meus descontos; 2. Clicar no cupão que deseja utilizar; 3. Clicar no botão para navegar para a loja e reclamar o prémio; 4. Navegar para a loja; 5. Clicar no botão de reclamar o prémio; 6. Apresentar o copão a um colaborador da loja; 7. O colaborador visualiza o cupão dentro da aplicação e efetua o desconto indicado no cupão. 8. Cupão é removido automaticamente da lista de cupões do utilizador;

Troca de prémio

Descrição Geral: A aplicação permite o utilizador depois de gerar um cupão de desconto (prémio) das lojas aderentes da aplicação *ShowMe* efetuar a troca por outro que mais lhe agrade.

Stakeholder(s): Utilizador da aplicação;

Pré-Condição: Deverá ter efetuado já a geração do prémio, passo 1, 2 e 3 do requisito anterior

Fluxo de eventos A: 1. Clicar no botão para trocar por outro cupão; 2. Selecionar um cupão da listagem apresentada; 3. Clicar no botão “trocar” para efetuar a troca.

Fluxo de eventos B: 1. Clicar no botão para listar “os meus descontos”; 2. Clicar no cupão que deseja trocar; 3. Clicar no botão para trocar por outro cupão; 4. Selecionar um cupão da listagem apresentada; 3. Clicar no botão “trocar” para efetuar a troca.

4.3 Requisitos não funcionais

Os requisitos não funcionais de um sistema indicam quais as propriedades e restrições do sistema desenvolvido, como por exemplo linguagem em que é implementada, seguranças da aplicação, desempenho e até mesmo respeitar uma determinada legislação [60].

Esta secção visa identificar e descrever os requisitos relacionados com usabilidade, performance, portabilidade, segurança e obrigatoriedades da aplicação *ShowMe*. Tendo em conta o cenário atual do mercado de aplicações móveis, estes pontos devem ser bastante claros, pois são os primeiros a passarem por fazer de aprovação do utilizador e dos espaços comerciais e que ditará o sucesso ou não desta aplicação.

Interação humano-computador

Título: Interface intuitiva;

Requisitos Descrição: A interface da aplicação deve ser bastante simples e intuitiva, proporcionando ao utilizador uma experiência bastante agradável e com uma curva de

aprendizagem bastante curta. Todas as funcionalidades devem ser de rápido acesso, pois esta aplicação trata-se de um instrumento secundário do espaço comercial.

Finalidade: Para que seja uma aplicação que abranja um público diverso, deverá possuir uma interface bastante simples e intuitiva.

Título: Contextualização de conteúdos

Requisitos Descrição: A aplicação não deverá incluir lojas não aderentes, bem como garantir que caso uma loja aderente não tenha cupões disponíveis essa indicação seja dada ao utilizador.

Finalidade: Este requisito pretende principalmente não criar uma falsa esperança no utilizador, oferecendo assim uma experiência agradável tanto ao nível da confiança como da interação.

Performance

Título: Fluida e eficiente

Requisitos Descrição: A aplicação *ShowMe* não deverá sofrer de qualquer quebra ao nível do tempo de resposta às ações do utilizador. Tendo em conta a conjuntura atual do mercado não é aceitável uma aplicação sofra constantemente de problemas de performance.

Finalidade: Oferecer ao utilizador uma experiência fluida, garantindo que esta não causa qualquer impacto negativo no utilizador (stress/fadiga/etc.).

Título: Precisão de comunicação

Requisitos Descrição: Sendo este um dos principais requisitos da aplicação, comunicação entre cliente/loja, esta tem de garantir eficiência no que diz respeito à localização do utilizador e a forma como é notificado.

Finalidade: Garantir que o cliente recebe as notificações de lojas em seu redor e de apenas aquelas do seu interesse.

Título: Precisão de método de navegação

Requisitos Descrição: Este requisito pretende que o algoritmo de navegação da *ShowMe* não seja suscetível a erro, necessitando de uma eficiência de navegação bastante próxima da excelência.

Finalidade: Dado se tratar de um dos mais importantes requisitos do sistema, é imperativo direcionar o utilizador para a loja que este efetivamente pretende visitar.

Título: Tamanho da aplicação

Requisitos Descrição: A aplicação não deverá exceder os 100Mb de ocupação de espaço, devido a restrições das variadas lojas de aplicativos para *smartphone*, uma vez que quando este valor é ultrapassado sugere a instalação da aplicação utilizando uma rede Wi-Fi, caso o utilizador esteja a utilizar dados móveis.

Finalidade: Este tipo de alertas, para um público-alvo específico, mais velho ou com menor tráfego mensal no seu tarifário, pode ser um entrave para a instalação da aplicação. A primeira iteração com a *ShowMe* não deverá desde logo ser um entrave para o cliente.

Portabilidade

Título: Compatibilidade com várias plataformas

Requisitos Descrição: Este requisito pretende garantir que a aplicação é compatível com as duas principais plataformas do mercado IOS e *Android*.

Finalidade: Como cada vez existe mais uma grande disparidade no mercado em relação ao sistema operativo utilizado, é importante abranger uma grande parte do mercado. Assim esta aplicação foi desenvolvida com a tecnologia *Xamarin* [61], capaz de ser instalada em múltiplas plataformas.

Segurança

Título: Segurança do utilizador

Requisitos Descrição: Este requisito não funcional pretende que na aplicação todas as proteções de dados do cliente estejam asseguradas.

Finalidade: Dado que cada vez mais se fala em proteção de dados, e dado a alteração de lei recente do RGPD (Regulamento geral sobre a proteção de dados), na aplicação *ShowMe* optou-se por implementar uma forma diferente de identificação do utilizador. Na primeira ligação do utilizador à aplicação é atribuído um *token* único (número) armazenado no seu telefone. Assim cada vez que este se ligar à aplicação este *token* será lido e carregado toda a sua informação, sem necessitar de colocar a informação habitual: nome, idade, email, género, etc.

Obrigatoriedades

Título: *Smartphones* com placas BLE/Bluetooth

Requisitos Descrição: Este ponto é o único requisito exigido ao cliente, é que o seu *smartphone* tenha uma placa BLE ou Bluetooth para a comunicação com a arquitetura da aplicação.

Finalidade: Sem esta tecnologia o utilizador não conseguirá interagir com sistema e portanto nem será permitida a instalação da aplicação, importante citar que hoje em dia esta tecnologia está presente em qualquer *smartphone* e que já está implementada nos telemóveis desde o ano de 2000 [62]. Durante a realização da presente dissertação não foi possível identificar qualquer *smartphone* sem esta tecnologia.

5. Arquitetura do Sistema

O presente capítulo descrever a arquitetura do sistema de uma perspetiva geral e completa. De um ponto de vista mais detalhado será especificada a arquitetura de comunicação entre o utilizador e o Beacon, bem como a arquitetura do sistema de navegação da aplicação.

5.1 Especificação geral do sistema

O cliente ao entrar no espaço comercial e ao ligar a aplicação esta inicia desde logo a sua monitorização constante de quais os *beacons* válidos em sistema e quais os que efetivamente estão ao seu alcance e a que distancia (em metros). O cliente inicia sempre a sua experiência no ecrã de preferências, isto permite desde logo atualizar os seus gostos e nesta mesma iteração usufruir destas novas configurações. O utilizador pode também consultar todas as lojas do sistema, e verificar a mensagem que cada uma delas tem para si. A loja tipicamente irá colocar uma frase apelativa que desperte o interesse do utilizador levando a que este gere de imediato um cupão de desconto.

Depois deste desconto ser gerado, o cliente pode desde logo dirigir-se para a loja para usufruir do prémio ou guardá-lo para uma posteridade. Se pretender a sua utilização imediata a *ShowMe* poderá auxiliar o utilizador no caminho até à loja. O utilizador pode ainda efetuar a troca deste ou de outros cupões que já tenha por um outro do seu interesse.

De uma perspetiva de uma utilização mais passiva, o cliente pode simplesmente ligar a aplicação e durante a sua visita será notificado, tendo em conta os seus interesses, de eventuais promoções temporárias das lojas aderentes à *ShowMe*.

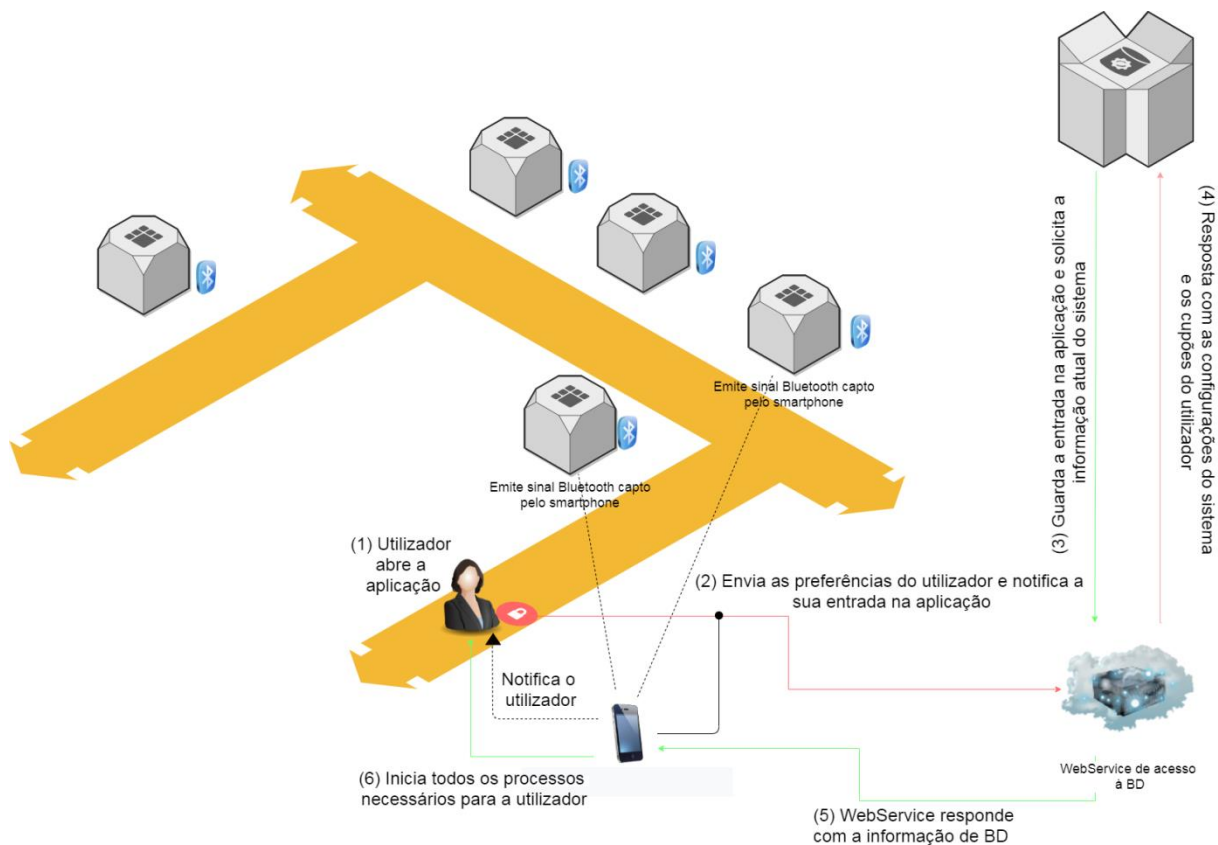


Figura 12: Arquitetura geral do sistema

Na Figura 12 está representada uma vista de alto nível do sistema em que se pode claramente identificar três pontos essenciais para o funcionamento da aplicação *ShowMe*. O envio dos interesses do utilizador e a sua entrada em sistema, a receção das configurações do sistema e os respetivos cupões do cliente, bem como a componente de notificação de uma determinada promoção consoante a proximidade do utilizador do *beacon* em questão.

O facto de consumir diversa informação armazenada em BD, permite reajustar as configurações da aplicação a cada momento que o utilizador abra a aplicação ou depois de X tempo de utilização. As configurações que podem ser ajustadas são: quais os *Beacons* instalados e as suas notificações ou cupões, resultantes de configurações de temporizadores que estão relacionados com o cálculo da posição da informação, novas configurações do sistema, etc.

Depois de carregada toda a informação recolhida na BD e com a posição atual do utilizador é possível iniciar-se a navegação para a loja seleccionada, quando necessário.

A última componente do sistema, que consiste em notificar o utilizador de eventuais promoções do seu interesse tem um funcionamento bastante moderado, isto é, no caso de o utilizador se manter nas proximidades de um *Beacon*, apenas será notificado uma vez da mensagem configurada. Assim, é garantido que esta notificação apenas acontece em *beacons* que estejam categorizados como um ponto de interesse do utilizador. Ainda no contexto das notificações, estas são de dois tipos: uma notificação nativa do *Android* normal e em 20% dos

casos pode ser gerado um *pop-up* dentro da aplicação, sendo que esta janela é gerada de forma aleatória e com a possibilidade de sucesso de apenas 20% (valor que pode ser ajustado para cada *beacon*).

5.2 Arquitetura da aplicação

O Presente capítulo pretende detalhar a arquitetura do sistema de módulos específicos deste projeto: a comunicação entre o utilizador e o *beacon* e a funcionalidade de navegação dentro de um edifício.

5.2.1 Comunicação entre utilizador e *beacon*

Na Figura 13 está representada um panorama referente ao módulo de comunicação da aplicação *ShowMe*.

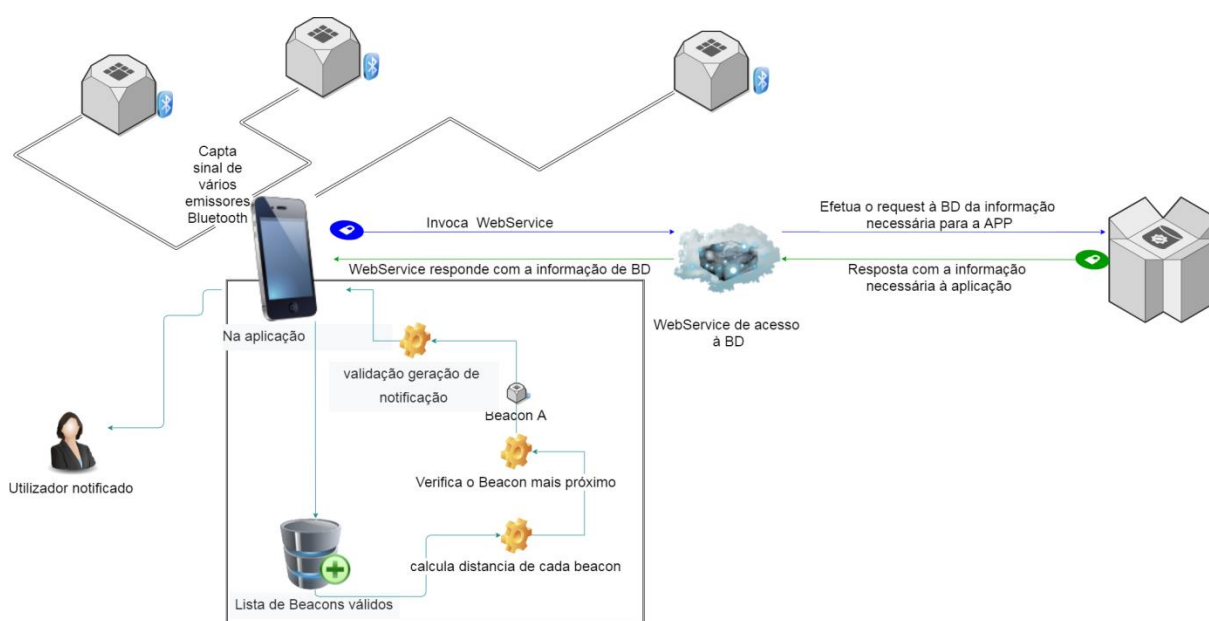


Figura 13: Arquitetura método de comunicação da aplicação

Como referido anteriormente existe numa primeira fase uma recolha de informação configurada na BD. Para este módulo importa ter em conta a fórmula de cálculo da distância entre o utilizador e o *beacon* configurada, bem como os *beacons* ativos e disponíveis em sistema.

Depois da informação proveniente da BD ser lida pela aplicação, é feita uma verificação a cada segundo de quais os dispositivos de proximidade com que o BLE do *smartphone* consegue comunicar. Estes são automaticamente validados com a listagem de *Beacons* ativos e configurados na BD. Todos os outros que não estiverem na listagem, por uma questão de segurança, são automaticamente descartados.

Depois de efetivamente ser verificado quais os *beacons* ativos e com quais existem comunicação é efetuado o cálculo da distância do utilizador para com cada um deles. Depois do valor da distância, d , estar disponível para todos os *beacons*, o sistema verifica qual o que está mais perto do utilizador e utiliza essa informação também para verificar se o valor está dentro de um limite considerado para gerar uma notificação, quando existente. Este limite de distância é um valor configurado na BD.

Estas validações anteriormente definidas são cíclicas e em constante avaliação enquanto o utilizador usa o sistema.

5.2.2 Navegação

Na Figura 14 está representada uma vista do componente de navegação do sistema desenhado durante a execução do presente projeto.

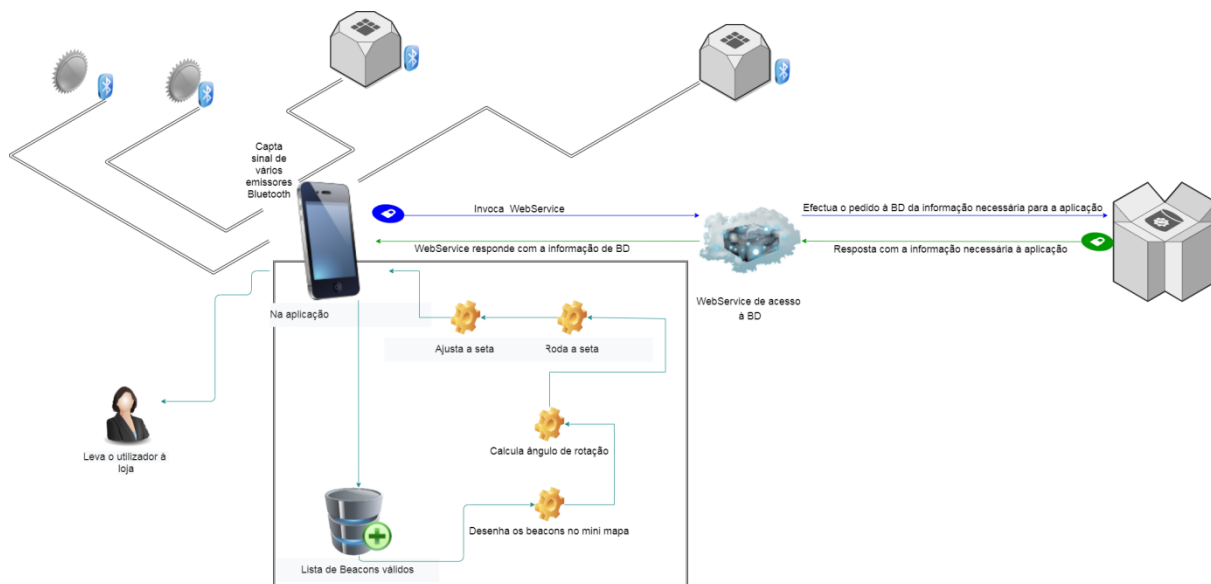


Figura 14: Arquitetura método de navegação da aplicação

Tal como capítulo anterior, para o correto funcionamento do método de navegação também é necessário configurar informação na BD. Torna-se necessário verificar a posição configurada no eixo $[x, y]$ de cada *beacon* e, depois dessa informação obtida, cada *beacon* é posicionado num mapa construído na janela da aplicação. De seguida é calculado o centro do eixo $[x, y]$ em relação aos *beacons* já criados e esse valor será a posição fictícia do utilizador nesse mini mapa. Esta posição, do utilizador, é sempre fixa, e estará representada por uma seta que indicará qual a direcção que o utilizador tem de tomar para o *beacon* seleccionado. O ângulo de rotação da seta é calculado tendo em conta o centro do eixo em relação à posição do *beacon* seleccionado, fazendo com esta gire corretamente na direcção da loja seleccionada (Figura 15).

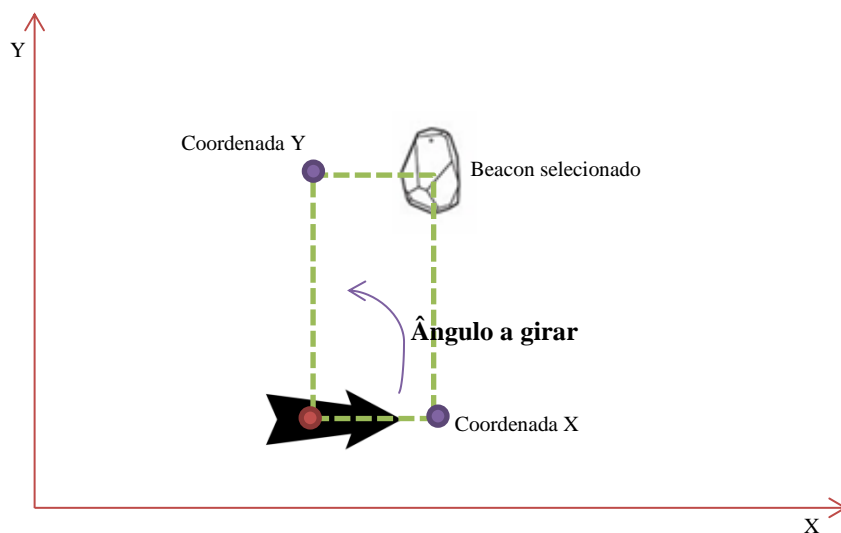


Figura 15: Funcionamento do método de rotação da seta da navegação

Importante referir que esta seta é reajustada sempre que a direção e ângulo do *smartphone* se altere em relação ao valor inicial. Importante citar que o valor inicial da seta é sempre o mesmo, desenhada na horizontal e só depois é que é calculado o ângulo a girar. Este calculo é quase imediato e por isso imperceptível para o utilizador.

Por último o utilizador será notificado que chegou à loja selecionada, quando a aplicação receber um valor de distância do *beacon* selecionado que esteja a menos de X metros de distância (sendo este um valor parametrizado).

6. Implementação

O presente capítulo pretende descrever toda a implementação efetuada durante a realização deste projeto, tanto ao nível do método de comunicação, bem como relativamente à componente de navegação da aplicação e o seu funcionamento. Será ainda demonstrado de uma perspetiva mais técnica o digrama de desenvolvimento, de classes e o modelo relacional da base de dados.

6.1 Diagrama de desenvolvimento

O diagrama de Desenvolvimento (Figura 16), visa elucidar o tipo de ligação e o protocolo utilizado para ser possível uma conexão viável entre a aplicação móvel e a BD.

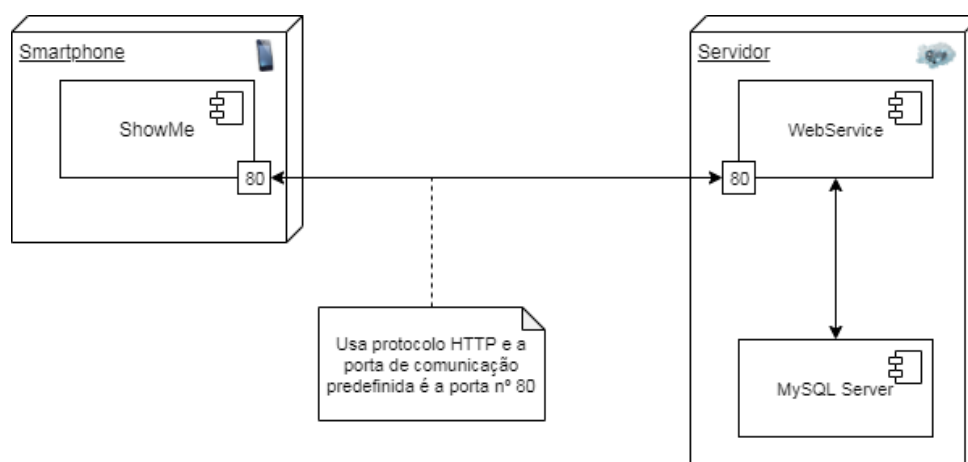


Figura 16: Diagrama de desenvolvimento

Neste diagrama é possível distinguir o nó “Smartphone” e o nó “Servidor”, sendo que este último possui a componente BD e um *Webservice*. Estes dois nós comunicam através do protocolo HTTP que utiliza a porta 80 como predefinido.

O “Smartphone” é o dispositivo inteligente onde a componente referente à aplicação *ShowMe* está instalada. É possível instalar a mesma num outro dispositivo, no entanto como referência foi utilizado o *smartphone*. A componente *WebService*, dentro do nó “Servidor”, é um serviço de integração de sistemas que permite a ligação à BD e efetuar algumas operações, utilizadas pela componente atual, mas também com escalabilidade para componentes futuras. O componente MySQL é a Base de dados relacional que suporta todo o armazenamento de dados da aplicação *ShowMe*.

6.2 Módulo de sistema de autenticação

Na aplicação *ShowMe* a privacidade do utilizador é uma mais-valia a ter em conta, assim como é importante garantir que o cliente rapidamente começa a experiência.

Assim no desenvolvimento desta dissertação o sistema de reconhecimento de utilizador passa apenas por a atribuição de um ID que é gerado na sua primeira entrada na aplicação. Este identificador é armazenado na memória do telefone do cliente (*localStorage*) para ser reaproveitado cada vez que este inicia a aplicação. Desta forma a *ShowMe* garante que todos os cupões de desconto gerados pelo cliente se encontram sempre disponíveis para utilização (Figura 17).

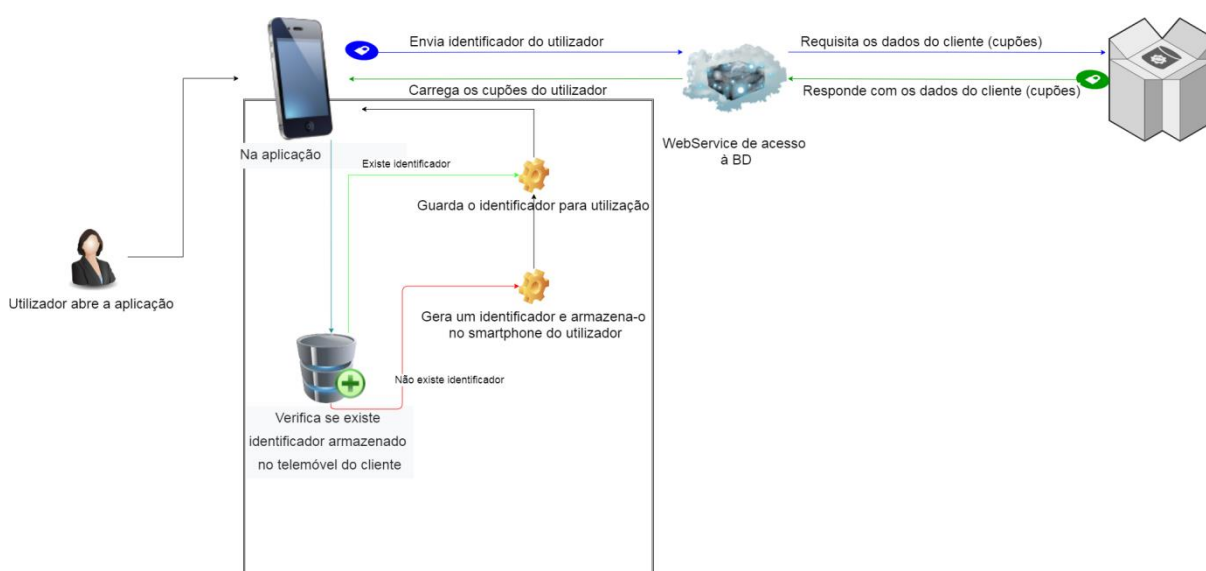


Figura 17: Diagrama de sistema de autenticação

6.3 Diagrama de classes

O diagrama apresentado na Figura 18, visa descrever as classes utilizadas pela aplicação *ShowMe*, descrevendo o impacto que cada um tem no funcionamento da aplicação móvel.

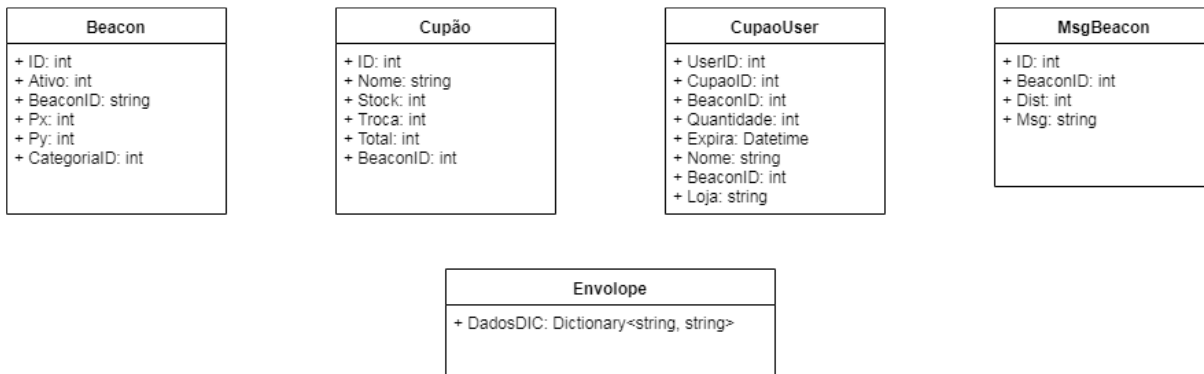


Figura 18: Diagrama de classes

A classe Beacon é onde se encontra toda a informação do beacon propriamente dita tendo como principal função representar a estrutura e funcionalidades associadas ao próprio Beacon. A classe cupão é em todo semelhante à anterior mas com o intuito de representar um cupão de desconto da aplicação.

Já a classe CupaoUser é uma classe com objetivo mais relacional e representa o desconto ganho de cada utilizador. A classe que estrutura as mensagens de notificação para o utilizador associada a um Beacon tem a designação de MsgBeacon.

Por último, surge a classe Envelope, que é uma classe mais utilizada na comunicação com o *Webservice* e indicará qual a instrução que este vai executar e qual o seu conteúdo.

Todas estas classes permitem à aplicação encapsular os conjuntos de objetos ou dados frequentemente utilizados.

6.4 Base dados

Na aplicação *ShowMe*, como em quase todas as aplicações, a grande lógica em termos de negócio começa a ser construída a partir da Base dados. Para tal, e dada a relação lógica existente entre as várias tabelas, foi construída uma base de dados relacional tendo em conta os requisitos referidos no Capítulo 4.

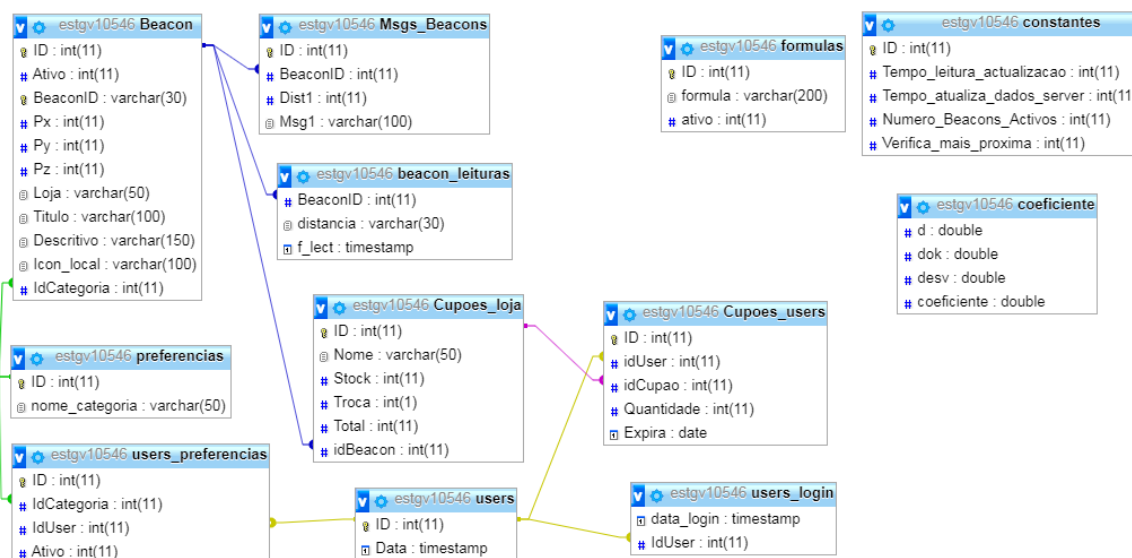


Figura 19: Modelo relacional da BD

Tal como na perspectiva da utilização do sistema, em BD as duas tabelas fundamentais são a tabela *Beacon* e *Users*, pois é a partir destas que praticamente todas as relações iniciam.

Começando pela tabela *Beacon*, considerando o diagrama apresentado na Figura 19, esta pode ter várias leituras a si associadas que vão sendo registadas ao longo do tempo e podem vir a ser úteis no futuro quando se desejar estatísticas das lojas mais visitadas. Uma outra relação desta classe prende-se com as notificações configuradas para os utilizadores. Neste contexto, cada *beacon* pode ter uma ou mais mensagens de notificação para o utilizador que podem chegar a variar com a distância a que este se encontra do *beacon*. Outra relação importante é a com a tabela *Cupoes_loja*, que representa quais e quantos cupões a loja tem disponível naquele instante.

Relativamente à tabela *users*, esta não tem qualquer informação pessoal do utilizador, pois tal como dito anteriormente, esta aplicação não requer autenticação através da introdução de utilizador e palavra-passe, conforme será apresentado na secção 6.5. Partindo desta tabela é possível identificar que também ela está diretamente relacionada com diversas outras tabelas tais como, *users_login* que indica a data e hora que determinado utilizador entrou na aplicação, a tabela *preferencias*, que tem todas as categorias mais relevantes para o utilizador e, por último, a tabela *cupoes_user* que indica quais os cupões que o utilizador tem naquele momento. Sobre esta última é importante referir que toda a gestão de *stock* de cupões é feita através de dois *triggers* presentes nesta BD. Um com a funcionalidade de subtrair os cupões disponíveis de cada loja assim que é atribuído um prémio ao utilizador ou quando troca o prémio, e outro que serve para incrementar os cupões devolvidos pelo utilizador nessa mesma troca.

Por último existem três tabelas auxiliares que apesar de não terem qualquer tipo de relação com outras, são importantes de uma perspectiva de escalabilidade da aplicação: a tabela *fórmulas* que indica a fórmula ativa para o cálculo da comunicação. Esta tabela contém constantes importantes no que toca ao refreshamento de dados utilizados pela aplicação e

ainda amostras de medições para cálculo do coeficiente de acerto da fórmula de comunicação, conforme será apresentado na secção 6.5. Em relação às constantes pode ser interessante serem alteradas de acordo com o ambiente, estratégia de negócio e dinâmica de interação com o utilizador que a empresa que compra o produto pretende adotar. Relativamente às amostras de medições de *beacons* e a sua margem de erro, tal como é indicado na secção 6.5, quantas mais existirem mais otimizado será o sistema de comunicação.

De forma a tornar a leitura deste relatório mais fluida, toda a informação relativa a cada campo de cada tabela pode ser consultada no Anexo 1 deste documento.

6.5 Módulo de comunicação da aplicação

Como dito anteriormente o *beacon* emite uma série de informação que outro dispositivo inteligente consegue receber e interpretar.

É através destes outputs emitidos pelo *beacon* que a aplicação *ShowMe*, depois de efetuados os cálculos necessários, vai ser capaz de interpretar qual o *beacon* que está mais próximo do utilizador. Assim, pode-se afirmar que o *RSSI* e a Potencia de Transmissão (*TPower*) são os dois dados importantes para o cálculo da proximidade do utilizador em relação ao *beacon*. O *RSSI* fornece a força do sinal que o dispositivo receptor consegue ler, sendo este valor sempre negativo e varia de -26 (o utilizador encontra-se a alguns centímetros do beacon) a -100 (utilizador encontra-se a 40/50 metros de distância do beacon).

A *TPower* representa a potência de transmissão, ou seja, a energia a que o transmissor emite o seu sinal. Este valor tipicamente varia entre -40 *dBm* e +4 *dBm*, influenciando o alcance do sinal, ou seja, quanto maior o poder de transmissão do *beacon*, maior será também o seu alcance. No entanto, é também necessário ter em conta que quanto maior a *TPower* do *beacon*, maior será o gasto de bateria, e por isso a sua durabilidade é menor.

Estas duas variáveis de entrada, juntas, fornecem à aplicação um valor de distância, *d*. Existem diversas fórmulas que podem ser utilizadas, no entanto, a fórmula mais comum na literatura é dada pela equação 6-3 [63]:

$$d = \frac{10^{(TPower-RSSI)}}{10 \times N_{[2,4]}} \quad (6-3)$$

Importante citar que o valor *N* é uma constante de acerto que varia tipicamente entre 2 e 4.

Este valor *d*, é o valor da distância que o utilizador ou dispositivo inteligente está do emissor. Como a aplicação *ShowMe* pretende não só se destacar-se pelo aspeto funcional e a sua relação com o cliente, mas também pela capacidade de comunicação. Tal é conseguido através de uma melhoria introduzida na fórmula de comunicação referida anteriormente. Em determinados ambientes, fatores como o número de outros aparelhos eletromagnéticos e quantidade de pessoas concentradas num mesmo espaço, verificou-se que a taxa de

assertividade da fórmula genérica não era totalmente eficiente, necessitando sempre de cerca de 4 ou 5 amostras para obter um valor de distância aproximado ao valor real.

Posto isto, foi identificada a necessidade de efetuar um acerto à distância calculada, inserindo um coeficiente de acerto na equação. Este coeficiente necessitou primeiramente de uma fase de treino que pretende que sejam medidas e armazenadas amostras de distâncias, desvios e coeficientes. De seguida é calculado a média dos coeficientes armazenados e definido o coeficiente auxiliar da fórmula inicial. De seguida descreve-se este processo.

1º Passo – Medir e armazenar amostras

Efetuar um pré-processamento em que seja efetuado a medição manual (coluna *dok* da tabela seguinte) em comparação com a medida fornecida pela fórmula (coluna *d*, da tabela), ou seja, se a fórmula indicar que estamos a 5 metros do *Beacon* e após a medição estivermos apenas a 4 metros (coluna *dok* da tabela), existe um desvio de 1 metro na fórmula, representado na coluna *desv* da tabela 2. Com os valores de *d* e *dok* é possível calcular o erro e o respetivo coeficiente necessário para acertar a fórmula da aplicação que está representado na última coluna da tabela abaixo ilustrada. Para uma visualização da tabela com mais informação, consultar o Anexo 3.

<i>d</i>	<i>dok</i>	<i>desv</i>	Coeficiente = D/Dok
8,95	7,81	1,14	1,14
7,13	6,03	1,1	1,18
2,12	1,12	1	1,89
9,44	7,68	1,76	1,22
9,57	8,36	1,21	1,14

Quadro 2: Amostras de medições de distância entre Beacon e o Smartphone

De seguida é apresentado um gráfico (Figura 20) onde se verifica que a fórmula com o acerto do coeficiente calculado (passo 2), a verde, efetivamente se aproxima com a curva da distância real (a preto) em relação ao valor da distância calculada pela fórmula genérica de cálculo entre o *beacon* e o utilizador.

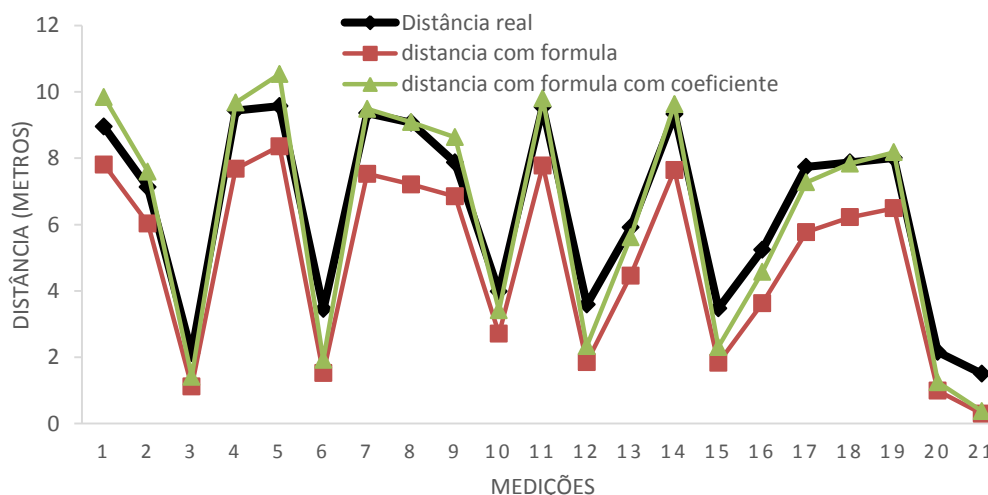


Figura 20: Gráfico de curvas de medições

2º Passo – Calcular média de coeficientes:

Depois de efetuado este pré-processamento, é calculado uma média dos coeficientes que irá auxiliar a fórmula de comunicação a tornar-se mais precisa, calculada da seguinte forma (6-4):

$$Coef_{\alpha} = \frac{\sum_{i=1}^N (d - d_{ok})}{N} \quad (6-4)$$

Em que α representa o coeficiente a calcular e N o número de registos totais armazenados em BD.

3º Passo – Acertar valor calculado:

Depois do valor de coeficiente obtido tendo em conta as amostras recolhidas, este será adicionado à fórmula inicial para que o cálculo da distância seja o mais eficiente possível, esta fórmula é representada da seguinte forma (6-5):

$$d = \frac{10^{(TPower-RSSI)}}{10 \times N_{[2,4]}} \times Coef_{\alpha} \quad (6-5)$$

Importante citar que o valor do coeficiente pode ser constantemente otimizado, visto que estas medições se encontram em base de dados e cada vez que a aplicação é iniciada é calculado o coeficiente tendo em conta as amostras atuais da BD, ou seja, quantas mais amostras existirem melhor será o acerto a considerar para o parâmetro médio.

6.6 Módulo de navegação da aplicação

Tal como já foi referido, hoje em dia existem imensas formas de navegação *indoor* utilizando tecnologias como *NFC*, *Wi-Fi* e *Beacons* [48] [31] [64]. No entanto, a maioria das vezes os sistemas de navegação são implementados com a troca frequente de sinais, ou seja, a aplicação tem três parâmetros de entrada: local a navegar, posição atual (dispositivo mais perto) e próximo dispositivo.

Utilizando como exemplo a tecnologia *Wi-Fi*, uma aplicação deverá estar em contínua análise do sinal de comunicação mais forte para obter a posição atual e, sabendo que tem como destino o *AP X*, a aplicação tem de calcular o melhor trajeto verificando os *AP's* da proximidade e identificar qual o mais adequado a ser o próximo ponto de deslocação do utilizador, e assim sucessivamente até chegar ao *AP* de destino.

Com a tecnologia *Beacon* a maioria dos sistemas de navegação implementados hoje em dia funcionam exatamente da mesma forma que os do *Wi-Fi* alterando apenas o protocolo de

comunicação. Enquanto que com o *Wi-Fi* é possível questionar-se o avultado investimento para um sistema com diversas comunicações, uma vez que necessita de largura de banda adequada, com os *Beacons* quanto maior for o número de comunicações menor é a durabilidade de cada aparelho.

O *NFC* tal como dito no capítulo do estado da arte, também já utilizado como forma de navegação é a tecnologia com mais limitações dentro das referidas anteriormente, pois é preciso um grande nível de proximidade entre o dispositivo emissor e recetor, e mais suscetível a erro por se tratar de uma navegação feita ponto a ponto, ou seja, caso o utilizador falhe um dos pontos do trajeto já não é possível voltar a enquadrar, contrariamente aos métodos referidos anteriormente.

Apesar de tudo estas formas de navegação podem efetivamente ser legítimas, no entanto facilmente se percebe que existe imensas operações até chegar ao destino final. Essas operações podem trazer lentidão à aplicação, quando não implementada corretamente, e até mesmo requerer um maior esforço por parte do utilizador da aplicação.

Com vista a poupar recursos tanto do cliente como da entidade responsável pelo espaço comercial, bem como fornecer uma experiência mais eficaz e agradável ao cliente, na aplicação *ShowMe* foi desenvolvido um método de navegação específico.

Resumidamente, o sistema de navegação começa por uma configuração dos *beacons* ao nível do seu posicionamento considerando um eixo $[x,y]$ em que o ponto $(0,0)$ é a origem. É a desta origem que é calculada, em metros, a posição de cada *beacon* no eixo.

Estas configurações são utilizadas para depois efetuar a construção de um “mapa” em pequena escala para o utilizador (Figura 21).



Figura 21: Mini mapa da aplicação ShowMe

Nota: Este mapa será representado com quadrantes pois é importante ter esta ideia visual para o que se segue.

Este mapa é construído cada vez que o utilizador pretenda navegar para uma loja, garantindo que a configuração atual do sistema está sempre refletida nesse mesmo mapa. Depois de este mapa estar devidamente desenhado é adicionado uma seta ao centro do ecrã com o objetivo de direcionar o utilizador para a loja certa. Esta seta irá X graus de acordo com as seguintes fórmulas (6-6,6-7):

$$X_c = \frac{Max(x) + Min(x)}{2} \quad (6-6)$$

$$Y_c = \frac{Max(y) + Min(y)}{2} \quad (6-7)$$

O valor $[X_c; Y_c]$ fornece à aplicação o centro do sistema de eixos (x, y) , de seguida é necessário calcular a tangente entre o valor do centro de eixos e o *Beacon* que se pretende navegar $[P_x; P_y]$, da seguinte forma (6-8):

$$\tan \theta = \frac{\sin(P_y - Y_c)}{\cos(P_x - X_c)} \quad (6-8)$$

Por último é necessário calcular efetivamente o angulo a girar da seguinte forma (6-9):

$$\alpha = \frac{\tan(\theta) \times 180}{\pi} \quad (6-9)$$

A seta centrada anteriormente irá girar α graus, indicando ao utilizador em que direcção este se deve dirigir. É importante referir que o *Beacon* neste sistema de navegação irá ter um papel de objetivo a atingir indicando ao utilizador que chegou à loja pretendida.

Por último, falta apenas citar que esta seta será reenquadrada cada vez que o utilizador gire o seu telemóvel, para tal foi utilizado o sensor *Compass* (Bússola) do telemóvel de forma a esta se manter sempre na direcção que se pretende, a loja seleccionada.

6.7 Módulo funcional da aplicação

Tendo em conta a capacidade e o potencial de crescimento da tecnologia *beacon*, esta tese foca-se no desenvolvimento de uma aplicação que centralize os dois principais conceitos que estão subjacentes à tecnologia: a orientação das pessoas através dos *beacons* e da sua posição atual, e ao mesmo tempo oferecer uma plataforma de notificações, promoções e cupões escalável a qualquer negócio. Não obstante a lógica de escalabilidade no desenvolvimento da aplicação *ShowMe*, a prova de conceito foi orientada ao contexto de utilização mais usual: um centro comercial. É importante provar a adequação da aplicação neste cenário, visto que se trata de um espaço em que todos os pontos referidos anteriormente podem ser englobados e aproveitados por um espaço comercial. Neste contexto físico, por norma de grandes

dimensões, é essencial determinar a posição atual do cliente em relação aos *Beacons*. Assim, torna-se vital, conjugar a posição do cliente fornecida pelo sistema de cálculo da aplicação, com uma série de notificações do seu interesse.

A aplicação móvel funciona quase como um navegador de interesses num dado local. Para a aplicação funcionar corretamente, o cliente apenas terá de ter o bluetooth ligado no seu dispositivo, bem como permitir que seja notificado pela aplicação *ShowMe*.

O funcionamento da aplicação móvel é simples, tendo em conta que foi desenhada considerando o perfil dos utilizadores. Quando o cliente entra pela primeira vez na aplicação, é solicitado que sejam selecionadas categorias do seu interesse, conforme apresentado na Figura 22.

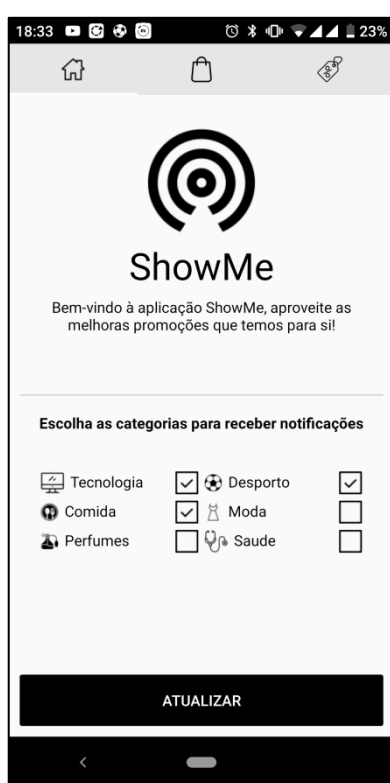


Figura 22: Seleção de categorias de interesse

Assumindo que o cliente já possui os seus interesses aquando da primeira entrada na aplicação, como por exemplo desporto, tecnologia, leitura. O cliente entra no espaço comercial com a aplicação instalada e a correr em background e será apenas notificado de promoções, novos produtos etc, das lojas da tipologia que definiu (Figura 22). Quando este efetivamente abrir a aplicação, serão listadas todas as lojas aderentes (Figura 23), podendo o utilizador selecionar uma do seu interesse e solicitar a geração de um cupão de desconto como é possível observar-se na Figura 24.

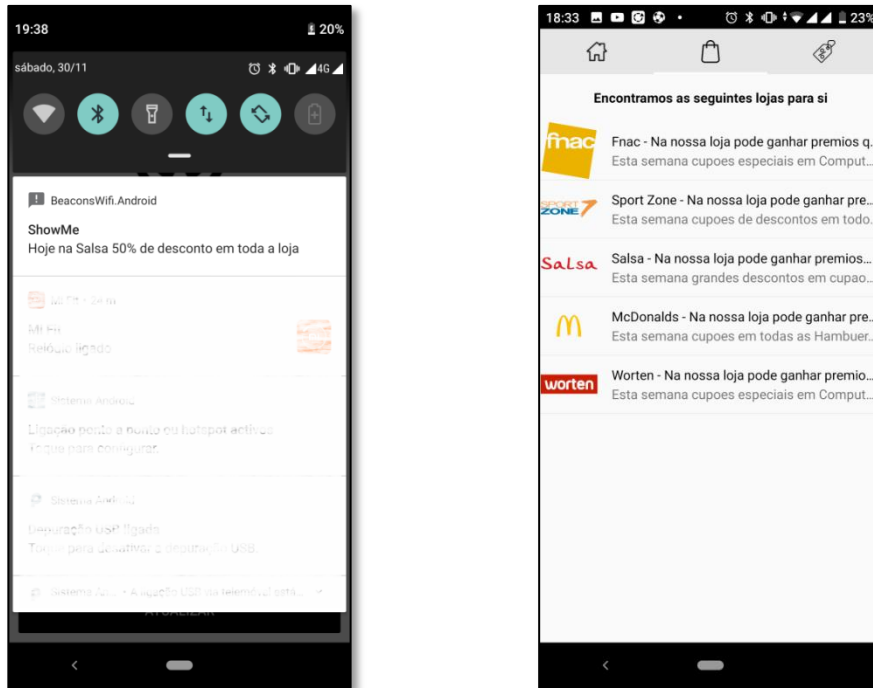


Figura 23: Notificação de promoções e listagem todas as lojas aderentes para geração de cupão

Quando o utilizador seleccionar uma loja e gerar o respetivo cupão de desconto, serão disponibilizadas várias opções de uso, “ir à loja reclamar o prémio”, “trocar por outro cupão” ou simplesmente guarda o seu cupão para mais tarde, retrocendo para o menu principal (seta).

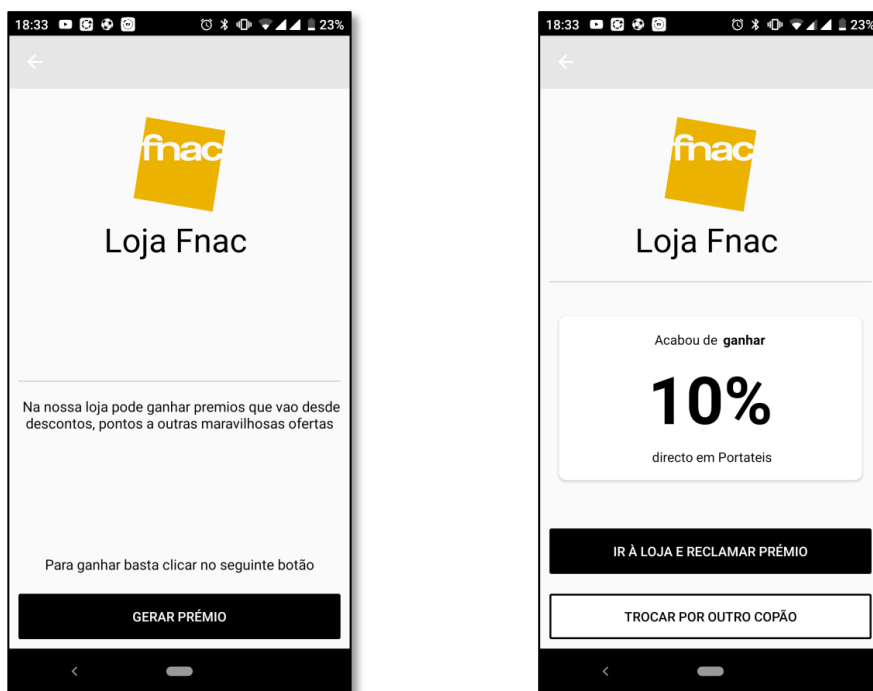


Figura 24: Geração de prémio e opções de uso disponibilizadas

Ir à loja reclamar o prémio

No caso de o utilizador selecionar a opção de reclamar o seu prémio, este será automaticamente redirecionado para a loja acionando um elemento orientador que está relacionado com a sua posição e a localização dos *beacons* mapeados no edifício. Ao chegar à loja é apresentada uma nova opção para efetivamente reclamar o seu prémio. Caso utilize o cupão, este ficará indisponível para outras utilizações (Figura 25).

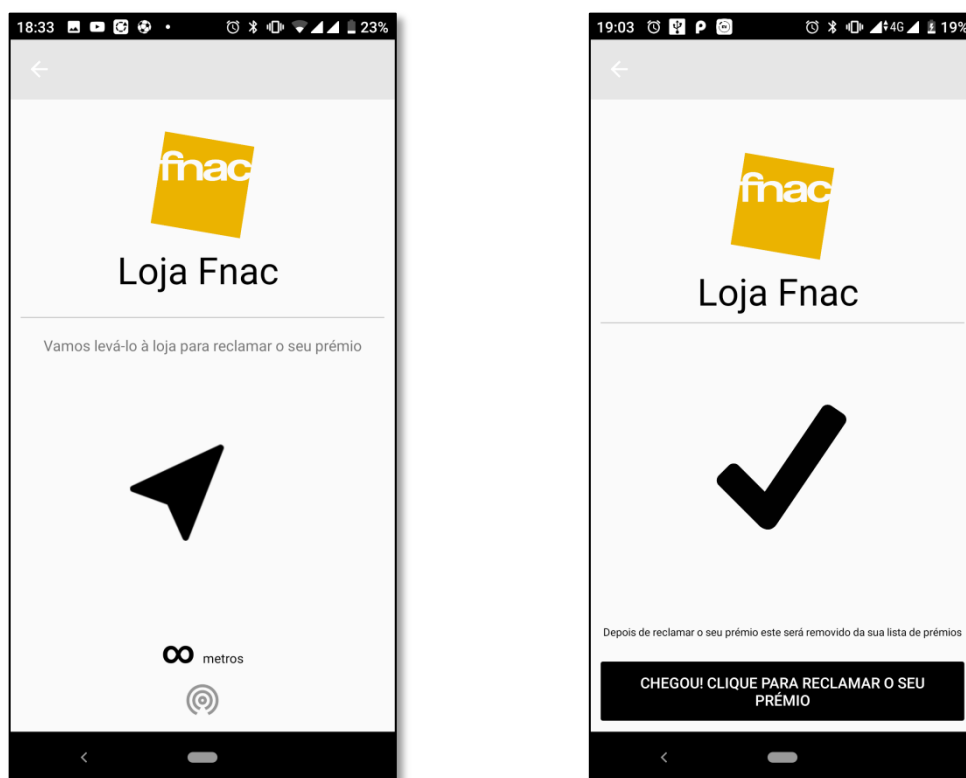


Figura 25: Direcionar o utilizador para loja e possibilitar reclamar o seu prémio

Importante ainda citar, que o simbolo referente ao infinito será alterado ao longo do tempo. Este simbolo é o valor por defeito de campo para quando não existe comunicação entre o *smartphone* e o beacon da loja selecionada. Isto acontece pois não é possível indicar com precisão a que distância estes se encontra até efetivamente haver comunicação entre ambos.

Caso o utilizador reclame efetivamente a utilização do seu cupão, será apresentada uma nova janela com todas informações da loja e da promoção para que o funcionário depois de validar a credibilidade do cupão atribua o desconto ao cliente (Figura 26).



Figura 26: Reclamar o prémio do cupão

Na presente janela da Figura 26 é também visível um identificador que está relacionado com o cupão gerado, este número é único e por isso é garante que o cupão em causa apenas é descontado uma única vez.

Trocar por outro cupão

A opção de trocar o cupão ganho por um outro do interesse do cliente, é mais uma forma que poderá beneficiar não só o utilizador, mas também a própria loja posicionando-se sempre numa perspetiva de ajudar o cliente a comprar o que pretende, impedindo que estes possam usar cupões da concorrência que sejam do seu interesse. Assim, caso o utilizador pretenda trocar o seu cupão será apresentada uma lista de diversos cupões compatíveis com o que possui atualmente, permitindo o cliente livremente escolher um deles para efetivar a troca (Figura 27).

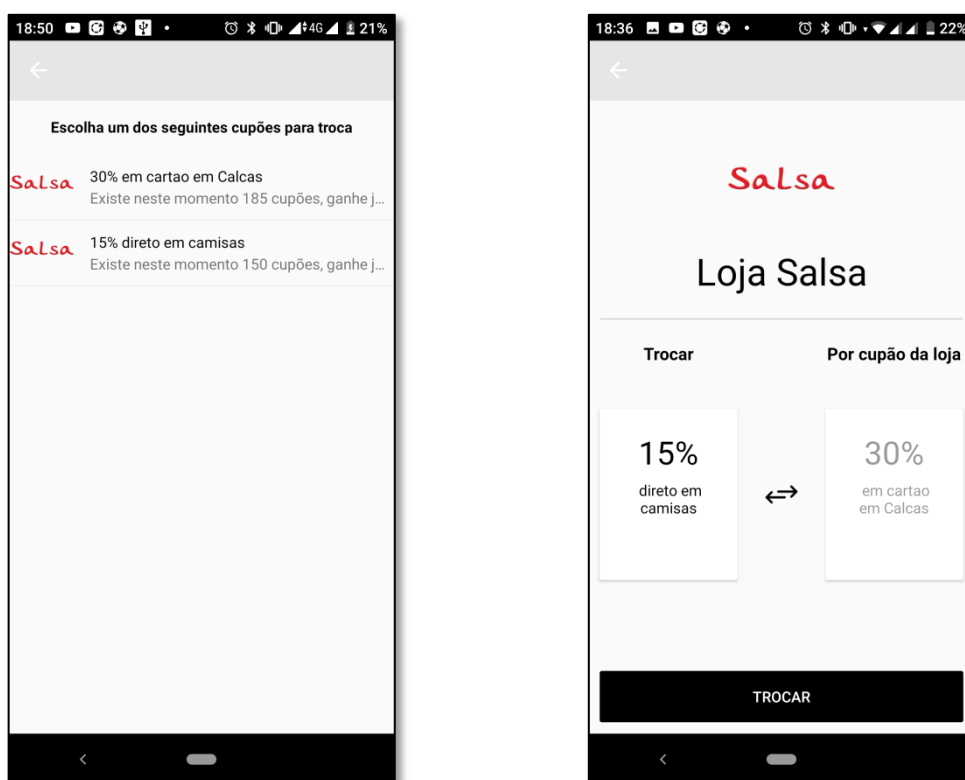


Figura 27: Lista de prémios para troca e efetivação da troca

Guardar cupão

Importante citar que caso o utilizador pretenda guarda o cupão para utilizar mais tarde, quer o original quer o novo (depois da troca) basta apenas recuar para o menu principal e este será adicionado à sua listagem de descontos (Figura 28).

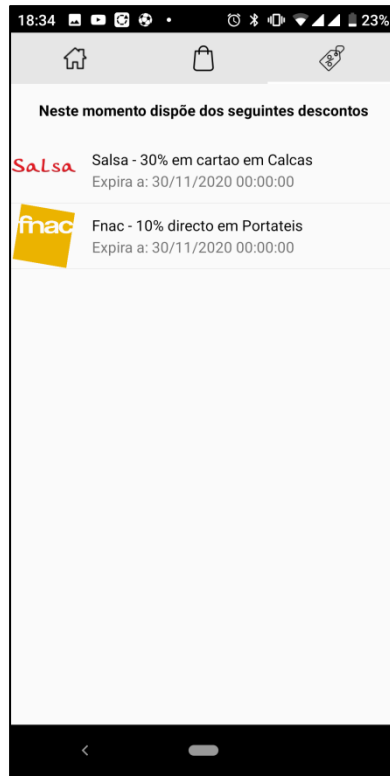


Figura 28: Lista “Meus descontos”

7. Conclusão e trabalho futuro

No presente capítulo será efetuada uma reflexão final sobre o trabalho realizado durante esta dissertação, bem como indicadas soluções de evolução ao sistema na sua totalidade de forma a torna-lo mais completo e eficaz tanto para o cliente como para os espaços comerciais

7.1 Conclusão

A presente dissertação, em conjunto com primeiro ano do Mestrado, fez com que o enriquecimento de matérias lecionadas, principalmente na área do desenvolvimento para dispositivos móveis, fosse notória mostrando também que é necessário uma enorme persistência e dedicação para atingir o objetivo final com a qualidade pretendida.

Este projeto demonstrou ser possível utilizar a tecnologia BLE/Bluetooth para estimar a localização de uma forma assertiva de um dispositivo inteligente dentro de edifícios, bem como demonstrar a implementação de um sistema de navegação *indoor* simplificado tirando partido do algoritmo de angulação.

Foi também garantido que, através de uma tecnologia recente mas com um custo reduzido de implementação, é possível aproximar as pessoas dos espaços comerciais e das lojas, oferecendo promoções de compra de produtos tendo como base os interesses dos utilizadores. Este complemento de *marketing* de proximidade faz acreditar que a presente implementação do projeto possa vir a ser utilizado por um grande número de espaços comerciais.

7.2 Trabalho futuro

Para o trabalho futuro, começa-se por referir funcionalidades a adicionar ao sistema. Uma funcionalidade que daria uma força adicional à aplicação era a implementação de um *backoffice* de gestão de todas as notificações e configurações do espaço comercial, que iria permitir fechar o ciclo de desenvolvimento do produto. Para além disso, podia ser adicionada ao sistema uma visão em tempo real da aglomeração de pessoas nos vários sectores do edifício, que iria permitir ajudar o departamento de segurança do espaço comercial a prevenir e identificar eventuais problemas que possam ocorrer naquele edifício. Neste contexto, poderia também efetuar-se a contagem de pessoas que entram numa loja, medindo-se assim o tráfego de pessoas.

Por último o sistema de comunicação e navegação podia passar a ter em conta a variável z do eixo $[x, y, z]$ para a identificação do piso atual, tornando-se uma mais-valia em edifícios com que os pisos baixos e que o sinal de um *beacon* pode passar para um piso superior ou inferior. Ainda da perspetiva dos espaços comerciais, era interessante oferecer uma ferramenta de sugestão de descontos que causariam maior interesse ao utilizador. Esta ferramenta devia ter como base o *Machine Learning* de forma a que este módulo fosse constantemente evoluindo. Já da perspetiva do utilizador, esta aplicação pode ser evoluída no que toca à geração de cupões de desconto, esta geração podia começar a ser feita automaticamente tendo em conta a área de localização do cliente, ou seja, em caso do utilizador estar na zona de restauração ser gerado um cupão de desconto de um determinado restaurante.

Em suma, é possível concluir que esta aplicação poderá tomar vários sentidos, tantos quantos os espaços comerciais necessitarem ao longo do tempo.

Referências

- [1] Nasa, “www.nasa.gov,” 30 12 2018. [Online]. Available: https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html. [Acedido em 30 12 2018].
- [2] H. Fredrick, 16 12 2018. [Online]. Available: <https://itstillworks.com/doesnt-gps-work-inside-building-18659.html>.
- [3] Texas Instruments, “www.ti.com,” [Online]. [Acedido em 25 06 2019].
- [4] Forbes, “<https://www.forbes.com/sites/homaycotte/2015/09/01/beacon-technology-the-what-who-how-why-and-where/#1ec4b92d1aaf>,” [Online]. [Acedido em 20 11 2019].
- [5] Grand view research, “<https://www.grandviewresearch.com/press-release/global-bluetooth-beacons-market>,” [Online]. [Acedido em 02 10 2019].
- [6] WordStream, 28 11 2018. [Online]. Available: <https://www.wordstream.com/blog/ws/2018/10/04/beacon-technology>.
- [7] Ibeacon, 17 11 2018. [Online]. Available: <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>.
- [8] C. Ranasinghe e C. Kray, “Location Information Quality: A Review,” 2018.
- [9] X. Y. Y. Z. Y. L. L. C. Yapeng Wang, “Bluetooth positioning using RSSI and triangulation methods,” *Consumer Communications and Networking Conference*, p. 837–842, 2013.
- [10] a. N. A. H. C. E. a. U. O. O S Oguejiofor, “Trilateration Based localization Algorithm for Wireless Sensor Network.,” pp. 21-27, 2013.
- [11] N. S. E. N. S. S. E. F. Nor Aida Mahiddin, “Indoor Position Detection Using WiFi and Trilateration Technique,” *International Conference on Informatics and Applications*, p. 362–366, 2012.
- [12] K. Kaemarungsi, “Efficient design of indoor positioning systems based on location fingerprinting,” *International Conference on Wireless Networks, Communications and Mobile Computing*, p. 181–186, 2005.
- [13] G. S. Tareq Alhmiedat, “An Indoor Fingerprinting Localization Approach for ZigBee Wireless Sensor Networks,” p. 190–202, 2013.
- [14] X. Ding, L. Gao e Z. Wang, “Modified Fingerprinting Algorithm for Indoor Location”.
- [15] S. S. Chawathe, “Beacon Placement for Indoor Localization using Bluetooth,” *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, 2008.
- [16] S. Goswami, “Indoor Location Technologies,” 2012.
- [17] J. Hightower e G. Borriello, “Location Sensing Techniques,” *IEEE Computer*, 2001.

- [18] F. Subhan, H. Hasbullah, A. Rozyyev e S. T. Bakhsh, “Indoor positioning in Bluetooth networks using fingerprinting and lateration approach”.
- [19] G. Trein, N. Singh e P. Maddila, “ Simple approach for indoor mapping using low-cost accelerometer and gyroscope sensors,” 2013.
- [20] E. I. Chang e R. P. Lippmann, “Using Genetic Algorithms to Improve Pattern Classification Performance”.
- [21] A. P. M. d. Araújo, “Proposição de modelo de risco de crédito para uma empresa,” 2016.
- [22] M. Y. Umair e R. Venkata, “An enhanced K-Nearest Neighbor algorithm for indoor positioning systems in a WLAN,” 2015.
- [23] A. Kataria e M. D. Singh, “A Review of Data Classification Using K-Nearest Neighbour”.*International Journal of Emerging Technology and Advanced Engineering*.
- [24] itchannel, “<https://www.itchannel.pt/news/a-fundo/queremos-apostar-em-projetos-nas-areas-de-mobilidade-localizacao-indoor-em-tempo-real-e-iot->,” 2018. [Online].
- [25] Bluetooth, “<https://www.bluetooth.com/specifications/bluetooth-core-specification/>,” [Online].
- [26] H. Karimi, “Indoor wayfinding and navigation,” 2015.
- [27] E. Perahia e R. Stacey, Next generation wireless LANs: throughput, robustness, and reliability in 802.11, 2008.
- [28] R. Mühlbauer, “Análise da Conectividade e Recepção de Beacons em”.
- [29] B. S. Boes, “System and method for reducing signal interference between bluetooth and WLAN communications”.
- [30] F. J. Villanueva, J. D. Gazzano, D. Villa, D. Vallejo, C. Mora, C. G. Morcillo e J. C. López, “Distributed architecture for efficient indoor localization and orientation,” 2013.
- [31] K. O. V. C. M. N. A. Busra Ozdenizci, “Development of an indoor navigation system using NFC technology,” *Proceedings - 4th International Conference on Information and Computing*, p. 11–14, 2011.
- [32] A. Paus, “Near Field Communication in Cell Phones,” 2007.
- [33] Q. Ladetto e B. Merminod, “Digital magnetic compass and gyroscope integration for pedestrian navigation”.*9th Saint Petersburg International Conference on Integrated Navigation Systems*.
- [34] V. Otsason, A. Varshavsky, A. Lamarca e E. D. Lara, “Accurate GSM Indoor Localization,” *Pervasive and Mobile Computing*, 2007.
- [35] Y. Tian, “Practical indoor localization system using GSM”.
- [36] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai e M. Wiseman, “Indoor location sensing using geomagnetism,” *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011.
- [37] D. Navarro e G. Benet, “Magnetic map building for mobile robot localization purpose,”

Referências

- IEEE Conference on Emerging Technologies and Factory Automation*, 2009.
- [38] Galván-Tejada, G.-V. Galván-Tejada, Delgado-Contreras e B. RF, “Infrastructure-Less Indoor Localization Using the Microphone, Magnetometer and Light Sensor of a Smartphone,” 2015.
- [39] M. Yoshino, S. Haruyama e M. Nakagawa, “High accuracy positioning system using visible LED lights and image sensor” *IEEE Radio and Wireless Symposium*.
- [40] T. Azevedo, “Roteamento ZigBee”.
- [41] Zigbee, “https://zigbee.org/zigbee-products-2/#zigbeecertifiedproducts/?view_30_search=end%20point&view_30_page=1,” [Online].
- [42] Y. Zhao, L. Dong, J. Wang, B. Hu e Y. Fu, “Implementing indoor positioning system via ZigBee devices,” *IEEE*, 2008.
- [43] H. Bhatt e B. Glover, *RFID Essentials*, 2005.
- [44] D. Gislason, *Zigbee Wireless Networking*.
- [45] L. Ni, Y. Liu, Y. C. Lau e A. Patil, “LANDMARC: indoor location sensing using active RFID,” *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, 2003.
- [46] R. O. Mitchell, H. Rashid, F. Dawood e A. AlKhalidi, “People tracking and location based services via integrated mobile and RFID systems,” 2013.
- [47] Indoo.rs, “<https://indoo.rs/>,” [Online]. [Acedido em 15 05 2019].
- [48] Estimote, “<https://estimote.com/>,” [Online]. [Acedido em 25 04 2019].
- [49] Infsoft, “<https://www.infsoft.com/technology/positioning-technologies/wi-fi>,” 05 10 2019. [Online].
- [50] Android, “developer.android.com,” [Online]. [Acedido em 29 08 2019].
- [51] Apple, “<https://developer.apple.com/>,” [Online]. [Acedido em 30 08 2019].
- [52] Microsoft, “azure.microsoft.com,” [Online]. [Acedido em 02 09 2019].
- [53] JQuery mobile, “<https://demos.jquerymobile.com/1.2.1/docs/pages/phonegap.html>,” [Online]. [Acedido em 05 09 2019].
- [54] Microsoft, “<https://docs.microsoft.com/en-us/xamarin/cross-platform/deploy-test/memory-perf-best-practices>,” [Online]. [Acedido em 03 04 2019].
- [55] LinkedIn, “<https://www.linkedin.com/jobs/xamarin-jobs/>,” 30 04 2019. [Online].
- [56] altexsoft, “<https://www.altexsoft.com/blog/mobile/13-apps-made-with-xamarin-cross-platform-development-in-practice/#ca-mobile-provides-a-safe-and-native-mobile-banking-experience>,” [Online]. [Acedido em 29 10 2019].
- [57] Mestra informática, “<http://www.mestrainfo.com.br/site/software-metodologia-agile/>,” [Online].
- [58] J. Boeg, *Kanban em 10 Passos*.

- [59] A. Söderholm, H. G. Gemünden e G. M. Winch, *International Journal of Project Management*.
- [60] E. Figueiredo, “https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/req-funcional-rnf_v01.pdf,” [Online]. [Acedido em 27 10 2019].
- [61] Microsoft, “<https://docs.microsoft.com/pt-pt/xamarin/xamarin-forms/deploy-test/index>,” [Online]. [Acedido em 03 11 2019].
- [62] Bluetooth, “<https://www.bluetooth.com/about-us/our-history/>,” 11 11 2019. [Online].
- [63] Iotandelectronics, “iotandelectronics.wordpress.com,” 13 12 2018. [Online]. Available: <https://iotandelectronics.wordpress.com/2016/10/07/how-to-calculate-distance-from-the-rssi-value-of-the-ble-beacon/>.
- [64] Infsoft, “<https://www.infsoft.com/>,” [Online]. [Acedido em 19 11 2019].
- [65] X. Y. Y. Z. Y. L. e. L. C. Yapeng Wang, “Bluetooth positioning using RSSI and triangulation methods,” *IEEE 10th Consumer Communications and Networking Conference*, 2013.
- [66] M. V. D. Vincent Pierlot, “Triangulation Algorithms for 2D Positioning,” [Online]. Available: <http://www.telecom.ulg.ac.be/triangulation/#biblio-14>.
- [67] L. M. Ni, Y. Liu, Y. C. Lau e A. P. Patil, “LANDMARC: Indoor Location Sensing Using Active RFID”.

ANEXO 1

Atributos da tabela Beacon

Atributos	Descrição
ID	Serve para representar consecutivamente o <i>beacon</i> , ou seja atribui-se incrementalmente um número ao <i>beacon</i> .
Ativo	Representa se o Beacon está ativo ou não e se efetivamente é utilizado pela aplicação.
BeaconID	Este campo representa o <i>MacAdress</i> do Beacon
Px	O valor Px representa a posição do Beacon um eixo X.
Py	O valor Py representa a posição do Beacon um eixo Y.
Pz	O valor Pz representa a posição do Beacon um eixo Z, importante citar que neste momento este campo não é utilizado, mas no futuro em caso de o mapa da aplicação ser em 3D este atributo é fundamental
Loja	Este atributo indica o nome da loja em que este está instalado
Titulo	Este campo representa uma mensagem de boas vindas da loja na aplicação <i>ShowMe</i>
Descritivo	O atributo descritivo é essencialmente utilizado como uma mensagem informação de promoções e cupões que está acontecer num determinado espaço temporal
Icon_local	O campo Icon_local irá ditar o caminho no servidor aonde está o logótipo da loja que o Beacon representa.
IdCategoria	Este campo representa a que Categoria a loja está associada (categorias representadas no anexo 8)

Atributos da tabela beacon_leituras

Atributos	Descrição
BeaconID	Este campo indica o número do <i>beacon</i> que registou a leitura.
distancia	Este atributo tem como objetivo armazenar a quantos metros de distâncias estava o utilizador no momento da leitura.
f_lect	Data e hora a que a leitura ocorreu.

Atributos da tabela constantes

Atributos	Descrição
ID	Serve para representar consecutivamente as constantes do sistema, ou seja atribui-se incrementalmente um número à constante, sendo que a com maior valor é a que se considera como constantes ativas do sistema.
Tempo_leitura_atualizacao	Este campo representa de quanto em quanto tempo a aplicação regista as leituras (representadas no anexo 2).
Tempo_atualiza_dados_server	Este atributo representa de quanto em quanto tempo a aplicação refrescar os seus dados com o que está armazenado na BD.
Numero_Beacons_Activos	Indicador relativa a quantidade de Beacon ativos no sistema.
Verifica_mais_proxima	Atributo que indica de quanto em quanto tempo é avaliado o Beacon que está mais próximo do utilizador.

Atributos da tabela Cupoes_loja

Atributos	Descrição
ID	Serve para representar consecutivamente os cupões do sistema, ou seja atribui-se incrementalmente um número ao cupão.
Nome	Representa o nome ou descritivo do cupão.
Stock	Este atributo representa a quantidade de cupões deste tipo que são possíveis gerar.
Troca	Este indicar permite saber se o cupão é um cupão apto para troca.
idBeacon	Este campo indica a qual Beacon pertence este cupão.

Atributos da tabela Cupoes_users

Atributos	Descrição
ID	Serve para representar consecutivamente os cupões do utilizador, ou seja atribui-se incrementalmente um número ao cupão do utilizador.
idUser	Representa o dono do cupão, ou seja, o utilizador a que este pertence.
idCupao	Representa qual o cupão propriamente dito que o utilizador, ou seja, o elo de ligação com os cupões do sistema (anexo 4).

Quantidade	Indica a quantidade deste tipo de cupões que o utilizador possuiu.
Expira	Representa a data em que o cupão expira, importante citar que o facto de o utilizador receber um novo cupão deste tipo faz com que esta data seja atualizada e assim tem mais cupões mas também mais tempo para os usar (tipicamente 1 ano)

Atributos da tabela formulas

Atributos	Descrição
ID	Serve para representar consecutivamente as fórmulas do sistema, ou seja, atribui-se incrementalmente um número à fórmula.
formula	Representa a fórmula de cálculo utilizada pela aplicação na componente de comunicação.
ativo	Indicar que representa se a fórmula está ou não ativa, importante citar que o sistema está preparado para que apenas esteja uma esteja marcada como ativa.

Atributos da tabela Msgs_Beacons

Atributos	Descrição
BeaconID	Serve para representar a que <i>beacon</i> a mensagem está relacionada.
Dist1	Indicador de distância a que a mensagem deve ser despoletada.
Msg1	Este atributo representa a mensagem com que o utilizador será notificado.

Atributos da tabela preferencias

Atributos	Descrição
ID	Serve para representar consecutivamente as categorias do sistema, ou seja atribui-se incrementalmente um número à categoria.
Nome_categoria	Descritivo da categoria propriamente dita.

Atributos da tabela users

Atributos	Descrição
ID	Serve para representar consecutivamente o utilizador, ou seja atribui-se incrementalmente um número ao utilizador.
Data	Representa a data de criação do utilizador.

Atributos da tabela users_login

Atributos	Descrição
data_login	Este campo tem como objetivo representar a data a que o login de um determinado utilizador foi efetuado.
IdUser	Este campo é o responsável pela relação desta tabela para com a tabela utilizadores e representa nada mais que o próprio utilizador.

Atributos da tabela users_preferencias

Atributos	Descrição
ID	Serve para representar consecutivamente uma preferência do utilizador, ou seja atribui-se incrementalmente um número à preferência do utilizador.
IdCategoria	Representa a categoria que o utilizador tem como favorita.
IdUser	Este campo indica a qual utilizador pertence esta informação.
Ativo	Indica se a preferência do utilizador está ou não ativa.

ANEXO 2

	Nativa	Web	Híbrida	Cross Platform
Custo e portabilidade	Elevado, quando se pretende desenvolver para mais que um sistema operativo.	Médio, pois apesar de se tratar de uma tecnologia bastante abrangente no mercado é necessário um grau de performance da aplicação bastante elevado e por isso necessário developers com bastante background.	Médio/Baixo, pois trata-se de uma tecnologia com bastante abrangência de mercado e capaz de efectuar o deploy para vários sistemas operativos.	Baixo, pois a para além da compatibilidade com os vários sistemas operativos é também esta compatível com diversas linguagens de programação (no caso específico do <i>Xamarin</i>).
Funcionalidades nativas	Acesso a todas as funcionalidades do sistema operativo e <i>smartphone</i> .	O acesso a estas funcionalidades é bastante restrito.	O acesso às funcionalidades nativas depende sempre da <i>framework</i> em utilização e por isso poderá haver limite de algumas funcionalidades em detrimento de outras	Do que foi possível experienciar todas as funcionalidades nativas são facilmente acessíveis. Paralelamente não foi encontrado nenhum relato de limitações.
Consistência da interface	Nas aplicações existem componentes de interface uniformizados por sistema operativo.	As <i>frameworks</i> web permitem alcançar uma interface bastante semelhante à nativa.	As <i>frameworks</i> web permitem alcançar uma interface bastante semelhante à nativa.	As aplicações desenvolvidas nesta tecnologia permitem alcançar uma interface bastante semelhante à nativa.
Distribuição	As lojas de aplicações móveis têm benefícios no que toca ao <i>marketing</i> customizado, no entanto existe uma série de restrições que têm de ser compridas para ter a aplicação disponível em loja.	Não tem qualquer restrição no entanto também não tem qualquer benefício de <i>marketing</i> .	As lojas de aplicações móveis têm benefícios no que toca ao <i>marketing</i> customizado, no entanto existe uma série de restrições que têm de ser compridas para ter a aplicação disponível em loja.	As lojas de aplicações móveis têm benefícios no que toca ao <i>marketing</i> customizado, no entanto existe uma série de restrições que têm de ser compridas para ter a aplicação disponível em loja.
Performance	O código nativo acede directamente a todas as funcionalidades do sistema operativo e do <i>smartphone</i> e por isso a sua performance é notória.	Performance pode ser um problema quando existe uma má ligação à internet.	Quando se trata de uma aplicação complexa o facto de existir um sistema de multicamadas pode prejudicar a performance da aplicação.	Apesar de não ser uma aplicação com acesso nativo a todas as funcionalidades a interpretação do acesso às mesmas não é sentida pelo utilizador. O aluno para além de comprovar isso na utilização da sua aplicação, pode também confirmá-lo pela frequente utilização da aplicação CA Mobile.
Resultado (em pontos)	2,4 pontos	0,8 pontos	1,1 pontos	2,6 pontos

Nota: O resultado foi efetuado com uma média dos resultados anteriormente assinalados em cada casuística.

Verde (3 pontos) – Bom ; Amarelo(1 pontos) – Médio ; Vermelho(0 pontos) – Mau;

ANEXO 3

d	dok	desv	Coeficiente = D/Dok
8,95	7,81	1,14	1,14
7,13	6,03	1,1	1,18
2,12	1,12	1	1,89
9,44	7,68	1,76	1,22
9,57	8,36	1,21	1,14
3,45	1,52	1,93	2,26
9,36	7,53	1,83	1,24
9,07	7,21	1,86	1,25
7,87	6,85	1,02	1,14
3,98	2,71	1,27	1,46
9,54	7,78	1,76	1,22
3,59	1,85	1,74	1,94
5,91	4,46	1,45	1,32
9,33	7,64	1,69	1,22
3,47	1,83	1,64	1,89
5,24	3,63	1,61	1,44
7,74	5,77	1,97	1,34
7,87	6,22	1,65	1,26
8,01	6,49	1,52	1,23
2,16	0,99	1,17	2,18
1,5	0,3	1,2	5