Confirmação Automática de Presenças com IoT

Tomás Peixoto Rito Maria

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Orientadores: Prof. José Manuel da Costa Alves Marques

Eng. Paulo Jorge Mendes Marques

Júri

Presidente: Prof. Francisco António Chaves Saraiva de Melo

Orientador: Prof. José Manuel da Costa Alves Marques

Vogal: Prof. Miguel Filipe Leitão Pardal

Setembro 2020

Agradecimentos

Para poder realizar esta dissertação, vários foram os desafios que tive de enfrentar ao longo destes últimos meses. Para tal, foi essencial o apoio de todos aqueles que me ajudaram a alcançar, finalmente, esta etapa. Expresso aqui a minha maior gratidão a todos.

Aos meus orientadores, Prof. José Manuel da Costa Alves Marques e o Engº. Paulo Jorge Mendes Marques, por toda o apoio demonstrado e por todos os conselhos que me disponibilizaram.

À Carolina, por toda a dedicação e ajuda que me deu ao longo deste processo, por nunca me ter deixado desistir e por todo o carinho demonstrado.

A todos os colaboradores da Link Consulting que tiveram a disponibilidade para testar a aplicação.

Aos meus tios, por toda a força que me deram e por todo o apoio incondicional, em todas as etapas da minha vida.

Aos meus pais e às minhas irmãs, pela paciência e solidariedade que tiveram comigo, em ajudar sempre que necessário.

Sem vocês nada seria igual.

Obrigado a todos.

Resumo

O Sistema de Gestão de Transporte de Doentes (SGTD) necessita de novos métodos para efetuar os registos das horas de entrada e saída de utentes das unidades de saúde. O sistema que existe atualmente não garante precisão nos registos, afetando assim a contabilização incorreta do serviço de transporte prestado.

A solução proposta neste relatório visa a modernização deste processo, através de um sistema de deteção automático que procura oferecer uma maior fiabilidade nos dados obtidos, ao mesmo tempo que diminui os recursos humanos necessários para realizar essa mesma atividade.

Neste trabalho são abordados temas relacionados com as tecnologias a utilizar para desenvolver a solução, como é que estas estão a ser aplicadas actualmente e como é que podem ser aplicadas de forma a garantir um funcionamento viável no futuro.

Foi implementado um sistema que, utilizando beacons de Bluetooth juntamente com uma aplicação móvel instalada nos smartphones dos utentes, é capaz de detetar a sua presença nas unidades de saúde. Esta solução mostrou-se capaz de monitorizar a presença dos utilizadores no local, mas devido a alguns problemas iniciais de implementação, não foi possível alargar as funcionalidades do sistema até ao limite.

Palavras-Chaves: SGTD; Bluetooth Low Energy; Beacons; Controlo físico de acessos

Abstract

The Sistema de Gestão de Transporte de Doentes (SGTD) needs new methods to record the hours of entry and exit of users of health units. The system that currently exists does not guarantee accuracy in the records, thus affecting the incorrect accounting of the transport service provided.
The solution proposed in this report aims to modernize this process, through an automatic detection system that seeks to offer greater reliability in the data obtained, while reducing the human resources necessary to carry out this same activity.
This work addresses topics related to the technologies to be used to develop the solution, how they are being applied today and how they can be applied in order to guarantee a viable operation in the future.
A system was implemented that, using Bluetooth beacons together with a mobile application installed on users' smartphones, is able to detect their presence in healthcare facilities. This solution proved to be capable of monitoring the presence of users on the site, but due to some initial implementation problems, it was not possible to extend the system's functionalities to the limit.

Keywords: SGTD; Bluetooth Low Energy; Beacons; Control of physical accesses

Índice

1 INTRODUÇÃO 1

1.1 OBJETIVOS 2

2 ESTADO DA ARTE 4

2.1 SISTEMA ATUAL - SGTD 4
2.2 API DO SGTD 6
2.3 SONHO – SISTEMA INTEGRADO DE INFORMAÇÃO HOSPITALAR 7
2.4 SISTEMAS DE PRESENÇAS 7
2.4.1 RFID 7
2.4.2 NFC 8
2.4.3 Biométrico 9
2.5 TECNOLOGIAS DE LOCALIZAÇÃO 10
2.5.1 GPS 10
2.5.2 Wi-Fi 11
2.6 BLUETOOTH LOW ENERGY 11
2.7 BLE BEACONS 13
2.7.1 iBeacon 14
2.7.2 Eddystone 15
2.7.3 Smart Places 15

3 SOLUÇÃO 18

3.1 TECNOLOGIAS 18
3.1.1 Bluetooth Low Energy 18
3.1.2 Wi-fi e GPS 18
3.2 DESCRIÇÃO FUNCIONAL 19
3.2.1 Registo e Autenticação de Utilizadores 19
3.2.2 Deteção de Beacons e de Localização 20
3.2.3 Registo de Horas no SGTD 20
3.2.4 Visualização de Histórico de Transportes 21
3.2 IMPLEMENTAÇÃO 22
3.2.1 Utilizador / Aplicação móvel 22
3.2.2 Servidor 26
3.2.3 Beacons 28
3.2.4 SGTD 29

4 AVALIAÇÃO 32

4.1 DESCRIÇÃO DO TESTE 32
4.2 Configuração do Beacons *32*
4.3 Transportes de Utilizadores *33*
4.4 Confirmação de localização dos utilizadores *34*
4.5 Teste 1 – Aplicação Desenvolvida em Ionic *34*
4.6 Teste 2 – Aplicação Nativa para Android *35*
4.6.2 Análise de Resultados *35*
4.7 Desenvolvimento Nativo para iOS *39*
5 CONCLUSÃO **42**
5.1 – Trabalho Futuro *43*
BIBLIOGRAFIA **48**

Índice de Figuras

Figura 1 – Interface SGTD 5
Figura 2 – Sistema de RFID [7] 8
Figura 3 – Sistema de Verificação de Presenças utilizando NFC [8] 9
Figura 4 – Esquema de Utilização do Wi-fi como sistema de Localização no Interior [11] 11
Figura 5 – Interações Entre os diferentes papéis nas comunicações Bluetooth [14] 12
Figura 6 – exemplo de Colocação dos Beacons num Sistema de Indoor Mapping 14
Figura 7 – Configuração de Beacons utilizando o Protocolo ibeacon 15
Figura 8 – Arquitetura da Solução 22
Figura 9 – Interfaces de Registo de Utilizador e de Autenticação 19
Figura 10 – Página De Próximo Transporte ..... **Erro! Marcador não definido.**
Figura 11 – Colocação dos Beacons no Edifício da Link Consulting 33
Figura 12 – Pseudo Código da Implementação das Componentes de Deteção de Beacons ..... **Erro! Marcador não definido.**
Figura 13 – Pseudo Código de Implementação da Componente de GPS 38

Índice de Tabelas

Tabela 1 – Resultados dos registos de deteções diários 36
Tabela 2 – Registos do dia detalhados 37
Tabela 3 – Exemplo de Deteções registadas durante a estadia de um utente numa unidade de saúde 44
Tabela 4 – Exemplo de Configuração a utilizar dentro de uma unidade de saúde 45

Lista de Abreviaturas

BLE – Bluetooth Low Energy
SGTD – Sistema de Gestão e Transporte de Doentes
SONHO – Sistema Integrado de Informação Hospitalar
RFID – Radio Frequency Identification
NFC – Near Field Communication
GPS – Global Positioning System
RSS – Received Signal Strength
AP – Access Point
SDK – Software Development Kit

1 Introdução

O registo da hora de entrada e de saída de utentes de unidades de saúde é feito manualmente por administrativos do local de destino do transporte, através do Sistema de Gestão de Transporte de Doentes (SGTD)[1]. Por este registo ser feito manualmente pelos administrativos, é possível que existam erros humanos que irão influenciar a contabilização do serviço de transporte. A contabilização acaba por não descrever fielmente a estadia dos utentes nas unidades de saúde. Esta atividade pode ser automatizada utilizando outras técnicas para o registo dos eventos de entrada e de saída.

Visto que, actualmente, a grande maioria da população possui e utiliza smartphones no seu quotidiano, é evidente que se pode tirar partido das capacidades dos mesmos, de forma a melhorar a dinâmica de execução do processo - tanto a nível do utilizador como da entidade prestadora de cuidados de saúde.

Neste projeto é proposta uma alternativa ao método atual, que passa pela utilização do *Bluetooth Low Energy* (BLE) [3], uma tecnologia de transferência de dados a curto alcance, com baixos consumos energéticos. Uma implementação desta especificação do standard Bluetooth encontra-se na maioria dos smartphones disponíveis no mercado, garantindo uma possível utilização a grande escala, permitindo assim uma maior evolução de serviços semelhantes no futuro.

A solução passa por criar uma aplicação móvel que os utentes possam descarregar nos seus smartphones e, nas unidades de saúde, instalar uma série de BLE Beacons em locais estratégicos. Estes dispositivos são escolhidos pela possibilidade de publicitar a sua identidade a um custo reduzido, podendo o sinal emitido pelos mesmos ser captado pelos smartphones, para que estes executem uma determinada ação desejada.

Com estas duas componentes a trabalhar em conjunto, é possível notificar o SGTD de forma automática através da aplicação móvel, sobre a entrada e saídas dos utentes, sem a necessidade de outros intervenientes, tornando o processo mais eficiente e preciso quanto aos tempos dos eventos.

Neste relatório será apresentado o sistema utilizado atualmente através do SGTD, as atividades que definem os vários processos e qual a desvantagem da abordagem que é seguida. Posteriormente, serão apresentados outros projetos que, de certa forma, apresentam semelhanças com a solução desejada, de forma a compreender melhor quais são os passos que devem ser dados para cumprir os objetivos. Será também explicado o funcionamento de várias tecnologias para perceber quais as vantagens e desvantagens de cada uma e escolher aquela que faz mais sentido, no âmbito do projeto. Por fim será apresentada uma solução inicial ao problema descrito que irá conter também uma arquitetura desejada para o sistema.

1.1 Objetivos

O desenvolvimento deste projeto tem objetivos que se propõe a cumprir, sendo estes explicados de seguida.

Precisão no lançamento de eventos: um dos fatores mais importantes a considerar quando abordamos este problema é a necessidade de que, as horas de entrada e de saída registadas no sistema, descrevam a situação real do percurso do utente dentro da unidade de saúde. Por isso é necessário implementar uma solução que seja capaz de aferir, com grande certeza, estes dados. Este objetivo pode ser comprometido devido à existência de situações onde a deteção de um Beacon deveria ter ocorrido mas não aconteceu, como o contrário, com a existência de uma deteção errada que não deveria ter ocorrido mas foi registada no sistema. Os dados que são guardados no sistema devem descrever a estadia do utente na unidade de saúde e devem ser analisados para perceber se existe coerência entre os mesmos.

Transparência: é desejado que o registo do utente no sistema não necessite de qualquer tipo de ação por parte dos possíveis intervenientes no processo de *check-in* e de *check-out* (administradores do local e utente). O processo deve ser executado automaticamente pelo sistema, mesmo em situações onde o utente tem o seu smartphone bloqueado (guardado no bolso, por exemplo). A única interação que deve existir é o aparecimento de uma notificação, que avise o utilizador que o evento realmente aconteceu e a que horas foi registado no sistema. Desta forma podemos obter um sistema onde os serviços são simples e de uso intuitivo. Visto que muitos dos possíveis utilizadores podem ser pessoas de idade considerável, muitas vezes com dificuldades a funcionar com certas tecnologias, existe a necessidade de criar um sistema acessível também para estes utilizadores.

Custos: os beacons são, de origem, um equipamento de custo reduzido. No entanto, precisam de ser instalados em locais estratégicos para que seja possível garantir a utilização do mínimo de recursos possíveis, sem comprometer a qualidade do serviço desejada. Para além disso, estes devem ser o único tipo de *hardware* a ser colocado no local, diminuindo a complexidade do sistema. Também é necessário ter em conta que, para os utentes, os custos em transferência de dados têm de ser mínimos, logo é necessário implementar uma arquitetura que diminua a quantidade de informação que tenha de ser trocada.

Segurança: o sistema tem de garantir que o registo da deteção efetuado é efetivamente realizado pelo utente em questão e não por outra pessoa. Como a solução proposta é suportada por comunicações *wireless*, é necessário ter isso em conta e assegurar que não existem ataques de eavesdrop (principalmente se as mensagens trocadas contiverem informação sensível, que deve ser confidencial). Da mesma forma que o modelo atual garante a autenticidade da entrada e saída do utente quando esta é marcada no sistema, também os novos métodos têm de garantir estes requisitos.

Multiplataforma: O serviço deve estar disponível para todos os utilizadores. Para isso é necessário certificar que é suportado por todo o tipo de smartphones e sistemas operativos, sejam eles iOS ou Android. Visto que o suporte ao Windows Phone está a terminar, este sistema operativo não é considerado para a solução [26].

2 Estado da Arte

Ao longo da evolução da sociedade e da tecnologia, foram aparecendo vários serviços que, para calcular as tarifas aplicadas à sua utilização, se baseiam no início e/ou fim de certas atividades. Quando estes eventos ocorrem, são guardados no sistema os dados relativos à data em que ocorrem e quem deu início à atividade. Podem observar-se exemplos destas aplicações em: serviços de hotelaria, em que a tarifa aplicada varia consoante o número de dias da marcação; serviços de táxis, cuja duração da viagem influencia o preço final; ou até mesmo no caso de aquisição de software, onde diferentes preços podem ser aplicados consoante o plano adquirido (diferentes planos podem consistir em diferentes períodos de tempo de aquisição), entre outros exemplos.

O SGTD necessita do mesmo tipo de informação para que o transporte possa ser corretamente valorizado e posteriormente pago às entidades transportadoras. O preço do transporte é calculado com base na distância percorrida pelo transporte, taxas fixas que podem ser aplicadas se for necessário, por exemplo, a aplicação de oxigénio ou um kit de parto e o tempo de duração no local de destino. Se o agrupamento for composto por mais que um utente, o tempo de duração no local é igual para todos os utentes, sendo calculado pela diferença entre a última hora de saída e a primeira hora de entrada.

Cada vez mais começam a ser desenvolvidos e a estar disponíveis no mercado sistemas que tornam este tipo de processos mais eficientes e automatizados. As empresas começam a procurar alternativas que consigam melhorar os seus métodos de negócio, de forma a conseguirem aumentar os lucros, mantendo a satisfação dos clientes elevada.

2.1 Sistema Atual - SGTD

Atualmente, é utilizado o SGTD [1], um sistema de informação desenvolvido pela Link Consulting, para controlar e gerir as atividades relacionadas com o processo de transporte de doentes.

Todo o processo, que assegura ao utente um meio de transporte em ambulância, passa por diversas fases. O SGTD contém uma interface gráfica (Figura 1), que é acedida através de um browser de internet, onde são executadas todas as fases. Inicialmente um médico, no papel de prescritor, elabora um pedido de transporte no sistema clínico. Neste registo, é guardado no sistema a informação pessoal e clínica do utente, bem como a informação necessária ao transporte: a data e hora nas quais se vai realizar, o local de destino e qual o motivo da deslocação. A requisição de transporte registada no sistema clínico é então importada para o SGTD para ser tratada - primeiro a nível administrativo, para verificação da informação, horário e local de destino e, numa segunda fase, para ser aprovada pelo diretor clínico do local de prescrição em causa. A partir do momento que é aprovada, a requisição de transporte passa a ser designada por Credencial de Transporte, passando a existir legitimidade para realizar o transporte. Uma Credencial de Transporte poderá conter uma ou mais prestações,

dependendo do tipo de tratamento/exame/consulta que o utente irá realizar. Uma prestação identifica um transporte que se vai realizar. Isto significa que, uma credencial pode conter vários transportes do utente, todos com o mesmo motivo médico. Estes transportes são efetuados em dias diferentes.

Periodicamente, o SGDT executa um algoritmo de agregação que cria agrupamentos a partir de todas as prestações já aprovadas existentes no sistema. Um agrupamento é um conjunto de utentes que podem ser transportados em conjunto, criados com base em critérios como intervalo de horário entre prestações, freguesia de origem do transporte, destino do transporte, etc. O objetivo é agrupar o maior número possível de utentes, tornando o transporte mais eficiente enquanto mantém as necessidades especificas de cada indivíduo. Estes agrupamentos são apresentados como sugestões às entidades transportadoras, que tomam a decisão de os realizar ou não. As entidades transportadoras podem rejeitar as propostas de agrupamentos apresentados pelo SGTD. Caso esta ação de rejeição não seja realizada, a uma hora predeterminada na véspera do transporte, as propostas são aceites tacitamente pela entidade transportadora. Após a aceitação do agrupamento, as entidades de transporte confirmam a realização do transporte. São então atribuídos, aos agrupamentos aceites, o quartel dos bombeiros que vai realizar o transporte, assim como o motorista e viatura. A partir desse momento, os agrupamentos passam a ser visíveis pelas entidades de destino do transporte.

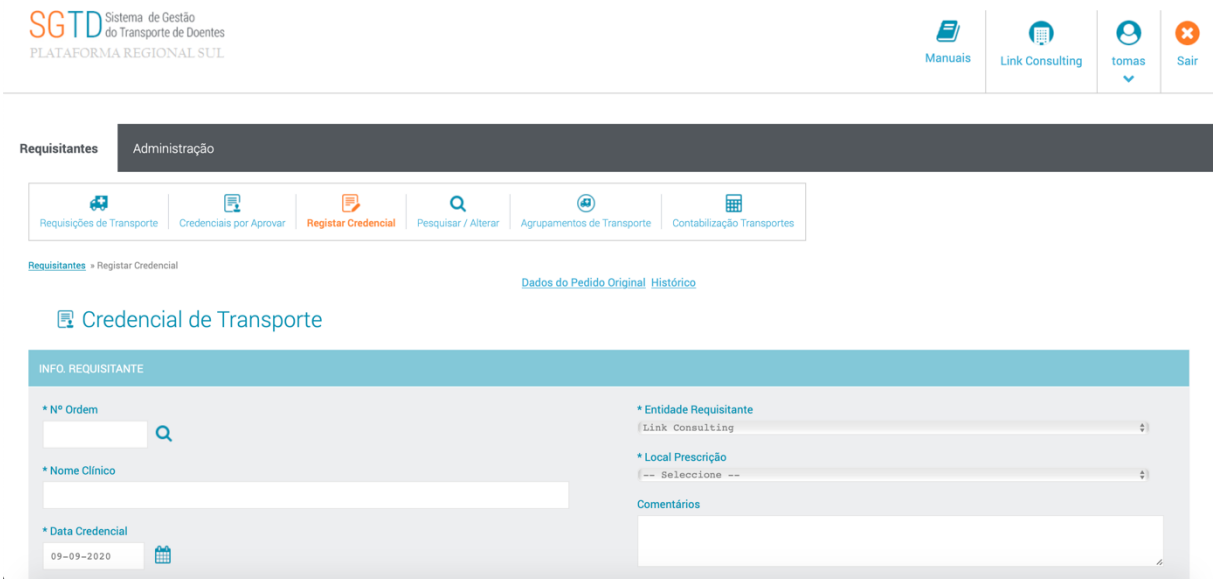

Figura 1 – Interface SGTD

Na chegada ao destino, a entidade prestadora do serviço de transporte dirige-se a um administrativo da unidade de saúde, que terá de inserir no sistema a hora de chegada ao local e a posterior hora de saída, quando estas se realizam. Esta informação pode ser substituída, caso o utente não compareça à consulta marcada, por uma nota de falta de comparência. Com a informação fornecida no tópico anterior, o algoritmo de contabilização do SGTD, responsável por valorar os agrupamentos de transporte, fica com toda a informação necessária para determinar os custos do agrupamento, encerrando desta forma, o processo.

Trata-se assim de um processo com vários passos e alguma complexidade e necessidade de intervenção manual, o que pode resultar numa gestão de horas de entrada e saída erradas. Estes dados vão resultar numa contabilização do serviço incorreta, sendo assim pagos valores que não descrevem o serviço prestado.

Apesar de não ser possível confirmar a existência de erros nas horas de entrada e de saída registadas, as unidades de saúde acreditam que estes erros efetivamente ocorrem. Não existindo, no sistema atual, maneira de avaliar o transporte e as horas de entrada e de saída, apenas com a utilização de um novo sistema podem ser obtidos números para suportar essas aferições.

Esta dissertação tem então como principal objetivo a automatização do processo de registo das horas de entrada e de saída de utentes, de forma a não ser necessário a inserção da informação no portal de forma manual por parte dos administrativos do local de destino do transporte.
Este método traz as seguintes vantagens: permite que as horas que são guardadas no sistema sejam mais precisas do que as inseridas manualmente pelos administrativos do local e retira do pessoal administrativo a tarefa repetitiva de registar a hora de entrada e de saída, para cada utente que entre na sua instituição via SGTD.

2.2 API do SGTD

O SGTD oferece uma API que contem três serviços que auxiliam na resolução do problema em questão, sendo estes: um pedido para retornar prestações, um para registar as horas de entrada de um utente no sistema e por fim outro para registar as horas de saída. A informação que é trocada neste pedido encontra-se no formato JSON.

No pedido de prestação, é enviado para o SGTD o número de telemóvel do utente e uma data. Com esta informação, o sistema consegue identificar se existem, para a data fornecida, prestações a executar, onde o utente é o detentor do contacto fornecido. O SGTD responde com uma lista contendo os identificadores de todas as prestações que encontrou e o local onde irão decorrer, sendo que esta lista pode estar vazia, caso o utente não tenha prestações para aquele dia.

Os pedidos de entrada e de saída funcionam da mesma forma. Estes contêm informação relativa a uma certa prestação (mais concretamente, o seu identificador) e uma hora. Com esta informação o SGTD guarda os dados relativos à entrada ou à saída (dependendo do pedido) para uma dada prestação.

É através destes três pedidos que é feita toda a comunicação necessária para solucionar o problema na sua forma mais básica. Estes pedidos são o pilar da comunicação que tem de existir entre a aplicação e o SGTD, uma vez que através da informação que é trocada, é possível implementar a solução sem necessitar de outras fontes ou mecanismos. No entanto, para garantir o correto funcionamento do serviço, é necessário criar medidas para que a utilização desta API seja efetuada na altura ideal. A aplicação não deve fazer um pedido de saída apenas porque detetou um beacon, pois estas situações podem acorrer variadas vezes, quando um utente se encontra numa unidade de saúde.

É necessário tomar decisões sobre quando é que estes eventos devem ser lançados, com o objetivo de mitigar ao máximo a existência de falsos positivos.

2.3 SONHO – Sistema Integrado de Informação Hospitalar

O SONHO [25] foi desenvolvido na década de 90, para dar suporte ao serviço administrativo dos hospitais, assegurando o controlo da produção e da faturação do mesmo. Uma das funcionalidades deste sistema é a de registar os movimentos do utente no local, efectuando também o registo das horas dos mesmos (como por exemplo, entradas numa especialidade).

Apesar de ser possível integrar o SGTD e o SONHO, para assim obter as informações sobre o utente e a sua estadia na unidade de saúde, observa-se que o SONHO não contém todos os dados necessários para aferir sobre a hora de entrada e de saída do utente. Um dos dados cruciais para este processo é a hora de saída do utente, dado esse que não encontramos no SONHO. Sem esta informação não pode ser efetuada a contabilização do serviço.

Apesar do SONHO não ser opção para a resolução do problema descrito, a sua utilização continua a ser benéfica, uma vez que contém informação que pode auxiliar na tomada de decisões em relação à entrada e saída dos utentes, assim como na confirmação da movimentação do utente dentro da unidade de saúde.

2.4 Sistemas de presenças

O uso de novas tecnologias em escolas e universidades, para controlar as presenças dos alunos começa a emergir. Utilizando diferentes abordagens, têm resultados significativos que ajudam nesta mudança de paradigma, tornando possível a adaptação do mesmo conhecimento para o caso em estudo. Os sistemas RFID, NFC e Biométrico são três exemplos de tecnologias utilizadas para este efeito.

2.4.1 RFID

Em várias universidades foram implementados métodos que utilizam *Radio Frequency Identification* (RFID) [6,7] para controlar as presenças dos alunos nas aulas, de forma a substituir os métodos tradicionais, por chamada ou por passagem de folha de presenças. Estes projetos tinham

como elementos principais etiquetas de RFID, que continham os identificadores de cada aluno, e um leitor que permitia ler as frequências de cada etiqueta. Estas componentes foram complementadas com um sistema de informação que guardava os logs das leituras efetuadas para depois serem processados devidamente. Um esquema desta arquitetura pode ser visualizado na Figura 2 que mostra a interação dos três componentes referidos anteriormente.

A utilização deste método demonstrou, em ambos os casos, melhorias na eficiência do processo de verificação de presenças, devido ao tempo que esta atividade demorou. Para além disso, como todos os dados são diretamente guardados numa base de dados, deixa de existir a necessidade de inserir manualmente as presenças no sistema, resultando numa alocação de recursos mais reduzida.

Apesar de o RFID ser uma tecnologia simples, com um processo de identificação trivial e de baixo custo, não é possível a sua utilização neste projeto devido à necessidade de distribuição de etiquetas para todos os utentes utilizadores do SGTD. Este processo poderia tornar-se dispendioso devido à elevada adesão ao serviço, uma vez que também seria necessário configurar cada uma das etiquetas individualmente, de forma a identificar cada um dos utentes.

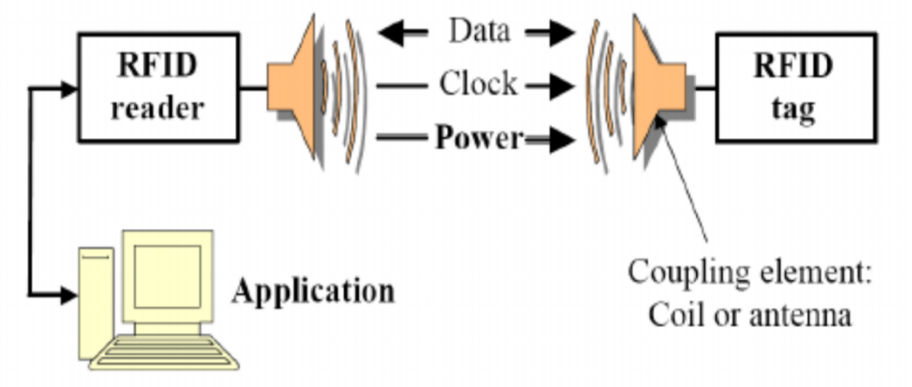

Figura 2 – Sistema de RFID [7]

2.4.2 NFC

Outra abordagem a este problema passa pela utilização de *Near Field Communication* (NFC). Um dos exemplos desta utilização é a universidade de Cartagena [8] que, com o objetivo de criar um ambiente tecnológico, permitiu que esta se tornasse uma universidade inteligente. Após a obtenção de resultados positivos no estudo, realizado para perceber o impacto do NFC na vida universitária, iniciaram-se as implementações para a verificação de presenças e os primeiros testes em aulas.

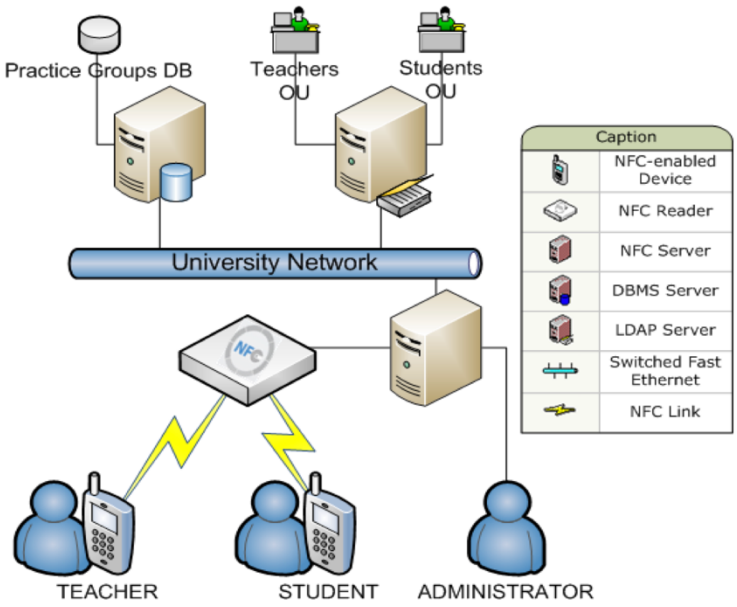

Figura 3 – Sistema de Verificação de Presenças utilizando NFC [8]

A componente NFC neste sistema é apenas composta por um leitor NFC e por uma aplicação móvel instalada nos smartphones, tanto dos alunos como dos professores. No início de cada aula, o professor faz *login* no seu telemóvel para poder informar o sistema, através do leitor de NFC, que vai dar início à aula, cuja localização fica também registada. Nesse momento, os alunos podem também fazer *login* e marcar a sua presença através do leitor. Na Figura 3 pode ser visualizado um esquema deste sistema, onde estão incluídos outros componentes que permitem o funcionamento do mesmo.

O sistema foi desenhado para que, em cada aula, um telemóvel apenas consiga marcar a presença uma única vez, o que garante que um aluno não pode fazer o registo duas vezes com o seu telemóvel. Ao mesmo tempo, pressupõe-se que os alunos não irão trocar telemóveis para marcar presenças. Para além do interesse por parte dos alunos e dos professores de adotar esta nova tecnologia, os resultados mostraram que existe realmente uma diminuição do tempo gasto, face ao método tradicional.

2.4.3 Biométrico

Atualmente, sistemas mais avançados tentam extrair informações diretamente de características únicas do corpo humano. Devido à singularidade de características como a impressão digital, a íris ou forma do rosto, podem ser implementados sistemas de reconhecimento que nos identificam rápida e automaticamente.

De uma forma geral, nestes casos, as características presentes numa pessoa são extraídas e guardadas numa base de dados. Estes dados começam por ser pré-processados para serem normalizados e, assim, existir consistência no processo de reconhecimento. Por fim, quando alguém

quer ser reconhecido pelo sistema, as suas características são novamente extraídas e utilizadas num processo de classificação, para identificar o utilizador.

No caso particular deste projeto, apesar da individualidade das características que iriam ser extraídas para este processo biométrico, seria necessário gravar a informação relativa a cada utente em alguma altura do tempo (numa primeira consulta, por exemplo). Para além disso, antes do processo de classificação, teria de existir um processamento das características, o que aumentaria significativamente a complexidade do sistema.

2.5 Tecnologias de Localização

As duas tecnologias presentes nos sistemas de localização atuais são o GPS e o Wi-fi. A primeira é bastante conhecia pela população. Começou a estar disponível para os condutores de veículos a motor, em dispositivos dedicados exclusivamente para navegação de estradas públicas. Com a evolução da tecnologia, o GPS começou a estar integrado nos smartphones e diversificou o seu âmbito para pedestres e velocípedes.

Apesar do Wi-fi ser conhecido maioritariamente para uso de redes locais sem fios, esta tecnologia pode ser utilizada em dois casos específicos, relacionados com a localização. O sistema de GPS aliado a esta tecnologia Wi-fi consegue aumentar a sua precisão que, na grande parte das situações, já é bastante elevada. Por outro lado, o uso de uma rede local sem fios permite criar um sistema de localização interior, que possibilita a um indivíduo, saber a sua localização dentro de um edifício.

2.5.1 GPS

Existem várias tecnologias para deteção de posição disponíveis no mercado. A mais usada é o *Global Positioning System* (GPS [9]), um sistema de navegação com base em satélites, que fornece posicionamento em tempo real. A localização é feita a partir da distância entre o recetor de GPS (que se encontra na terra) e três satélites que se encontrem na órbita terrestre.

Este posicionamento tem associado um intervalo de confiança de 95%, quando não existe qualquer barreira física entre o recetor de GPS e os satélites. Para o SGTD, seria necessário saber a localização de todos os estabelecimentos de prestação de cuidados de saúde para que, quando o utente chegasse e saísse do local, o sistema fosse notificado sobre cada evento. Também seria necessário decidir que ponto, no local, seria escolhido e ter em consideração que cada local tem a sua topologia e tamanho característicos. O posicionamento do utente no interior de uma unidade de saúde através do GPS também é pouco fiável, uma vez que este sistema está direcionado sobretudo para

uso exterior. Isto acontece porque o sinal recebido, quando se está no interior de um edifício, é atenuado, aumentando a margem de erro do posicionamento. Apesar deste erro não ser tão problemático como quando, por exemplo, se está a conduzir um automóvel, quando se fala em *indoor mapping*, a área de atuação é menor e o erro é mais problemático. Estes problemas tornam o GPS uma tecnologia pouco viável para resolver o problema em questão.

2.5.2 Wi-Fi

Outros estudos [10,11,12] também demonstram que o Wi-Fi pode ser considerado como tecnologia de localização para interiores. Este método depende do *Received Signal Strenght* (RSS), um sinal que o dispositivo (computador ou smartphone) recebe de um *Access Point* (AP). Para tal, também é necessário ter conhecimento sobre a planta da área em análise e onde estão localizados os APs. Através da triangulação do RSS, método semelhante, no cálculo de posicionamento, ao GPS, é possível determinar uma localização provável de um dispositivo numa determinada área. No esquema apresentado na Figura 4 estão representadas diversas zonas, bem como dois APs. Consoante as forças dos sinais recebidos por cada AP, é possível determinar a zona onde se encontra o dispositivo.

O Wi-Fi [24] funciona na mesma gama de sinais que o BLE (2.4GHz). Em contraste, os consumos energéticos são superiores: é necessário que os equipamentos (APs) estejam ligados à corrente, tornando o seu posicionamento no local uma tarefa mais elaborada.

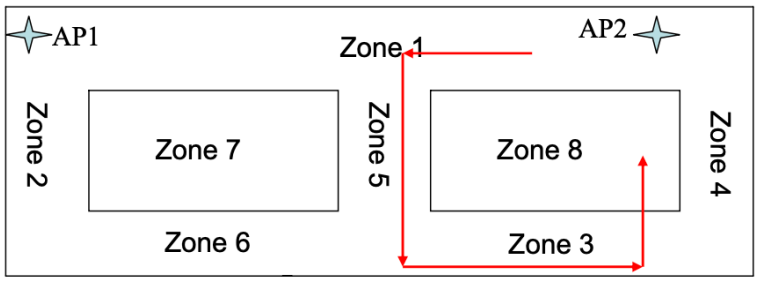

Figura 4 – Esquema de Utilização do Wi-fi como sistema de Localização no Interior [11]

2.6 Bluetooth Low Energy

O *Bluetooth Low Energy* foi desenvolvido pelo *Bluetooth Special Interest Group* para comunicações sem fios a curto alcance e foi incorporado nas especificações do Bluetooth versão 4.0. Este novo modelo de comunicação, semelhante ao standard Bluetooth, tem a particularidade de possuir menores consumos energéticos, bem como custos e taxas de transferência de informação menores.

Sendo o Bluetooth uma tecnologia ainda em crescimento (prevê-se que em 2021 esteja incorporado em mais de 41 mil milhões de dispositivos) [2], pode assumir-se que o *Bluetooth Low Energy* também fará parte do quotidiano e que o mercado da Internet das Coisas aumente.

O *Bluetooth Low Energy* [4,5] foi desenhado para transferir pequenos pacotes de informação, para que os dispositivos que o suportam possa informar sobre o seu estado. Opera na banda dos 2.4GHz, definindo assim 40 canais de frequências de rádio, separadas por 2MHz cada uma, sendo que 3 são para publicitar e as restantes para transferência de dados [14,15].

Na comunicação entre dispositivos através de BLE existem diferentes papéis para os participantes [13]: *Broadcaster:* periodicamente fazem broadcast de pacotes de dados definidos através dos canais específicos de publicidade; *Observer:* procuram encontrar pacotes que tenhas sido enviados pelos *advertisers*; Periférico: dispositivo que tem o papel de *slave* numa conexão com um dispositivo central; Central: dispositivo que tem o papel de *master* na conexão com um ou mais dispositivos periféricos. Na Figura 5 encontram-se as duas interações existentes entre dispositivos Bluetooth, assim como os papéis existentes em cada uma.

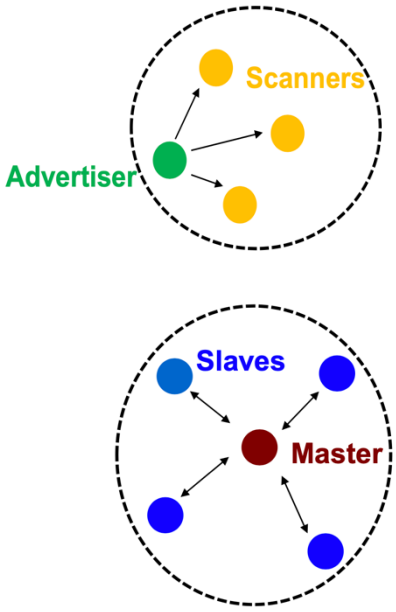

Figura 5 – Interações Entre os diferentes papéis nas comunicações Bluetooth [14]

O BLE faz com que qualquer objeto do quotidiano possa ser conectado ao smartphone, para assim se poder ganhar um novo controlo sobre o que está ao redor [14,16,17]. A lista de aplicações torna-se cada vez maior e é utilizada nas mais variadas situações, como as descritas de seguida. Em peças de vestuário o BLE pode ser utilizado para enviar informação captada por sensores embutidos para os smartphones. Isto implica a monitorização da respiração do utilizador, número de passos dados, pressão arterial, etc. Também em fechaduras inteligentes, que permitem que estas sejam (des)trancadas através de uma aplicação móvel, quando os dois equipamentos estão próximos e em smarthomes, um conceito cada vez mais presente como sendo o futuro no equipamento de casas. A

possibilidade de controlar cada equipamento presente na habitação através de smartphone começa a ganhar força com o aparecimento de produtos como lâmpadas, eletrodomésticos, portas e janelas inteligentes.

2.7 BLE Beacons

Os BLE Beacons são dispositivos eletrónicos que periodicamente, fazem *broadcast* de mensagens, através de sinais Bluetooth [18] (que seguem o protocolo BLE), que são detetadas por dispositivos que se encontram ao seu redor. Esta comunicação é unidirecional, pois os beacons não têm o papel de recetores, simplesmente fazem *broadcast* de informação, seguindo assim a topologia apresentada na secção 2.6 de *Broadcaster* e *Observers*.

As mensagens que transmitem são definidas pelo protocolo utilizado na comunicação. Alguns destes protocolos são o iBeacon desenvolvido pela Apple e o Eddystone desenvolvido pela Google. Ambos definem um pacote de informação que contém o identificador do beacons, enquanto que os restantes pacotes são específicos para cada protocolo.

Os sinais dos beacons podem ser detetados por smartphones, tablets ou computadores, desde que suportem BLE, sendo assim considerados como *observers* nesta comunicação. Os *observers* têm o papel de detetar sinais de beacons e executar um processo após essa deteção. Isto significa que os beacons sinalizam quando uma ação deve ser executada pelo *observer*. Para os processos não serem executados sempre que um dispositivo encontra um beacon, os *observers* podem ser configurados para filtrar os sinais provenientes dos beacons, através do seu identificador. Desta forma, apenas quando é detetado um beacon com um identificador específico é que um certo processo vai ser executado no *observer*.

Esta tecnologia pode ser aplicada em diversas situações e começa a estar presente no mercado:

Museu Guggenheim: através de beacons que se encontram por todo o museu, é possível receber, numa aplicação móvel, informação relativa a obras que se encontrem próximas do utilizador. Na aplicação estão guardados os identificadores dos beacons que se encontram no museu para assim, quando esta recebe sinal de um, apresentar a informação que está a transmitir.

Indoor Mapping: o *indoor mapping* é o conceito de navegação em espaços fechados. Este sistema pode ser implementado com o auxílio de um conjunto de beacons e uma aplicação móvel. Na Figura 6 podemos ver um local com quatro divisões e cada uma contém um beacon, representado a vermelho, que identifica essa mesma divisão. Com este conhecimento prévio do espaço físico, a aplicação, ao detetar um beacon, pode estimar em que local se encontra o utilizador, mostrando-lhe essa posição no mapa.

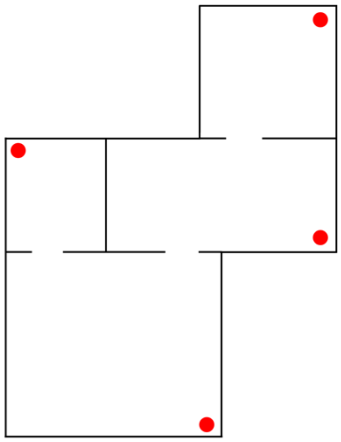

Figura 6 – exemplo de Colocação dos Beacons num Sistema de Indoor Mapping

2.7.1 iBeacon

O protocolo iBeacon [21] apresenta o conceito de uma região em torno do beacon, que permite aos smartphones determinarem quando entram ou saem do raio de alcance da região, ao mesmo tempo que estimam a proximidade ao beacon, através da força do sinal recebido. O identificador de uma região é definido por três campos que têm entre si uma relação hierárquica, e ao mesmo tempo são o identificador do beacon que a define:

UUID: composto por 16 bytes, deve ser específico para cada aplicação móvel.

Major: composto por 2 bytes, utilizado para aumentar e definir um uso particular da aplicação e assim ter maior precisão na identificação dos beacons.

Minor: composto por 2 bytes, possibilita uma precisão ainda maior para os usos na aplicação.

No exemplo da Figura 7 estão representadas três regiões distintas, onde as regiões 1 e 2 apresentam o mesmo valor de uuid. Possuíndo uma aplicação móvel que esteja configurada para receber apenas sinais dos beacons com este valor de uuid, o sinal da região 3 vai ser ignorado. Tanto na região 1 como na região 2, a aplicação vai executar um processo que é sinalizado pelos beacons. A diferença entre as duas regiões encontra-se no major, que é distinto. Esta separação permite que a aplicação móvel execute processos diferentes consoante a sub-região onde se encontra.

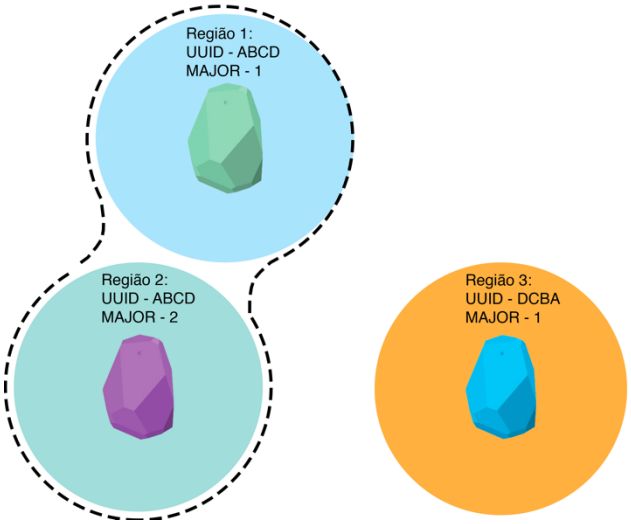

Figura 7 – Configuração de Beacons utilizando o Protocolo iBeacon

2.7.2 Eddystone

O formato Eddystone [22] pode ser detetado por dispositivos Android e iOS. Os pacotes podem incluir os seguintes *paylods*:

Eddystone-UID: um ID único e estático, que inclui 10 bytes referentes à identificação do serviço e 6 bytes referentes à instância. Este é o equivalente ao parâmetro UUID referido na secção anterior.

Eddystone-URL: um URL que pode ser diretamente gerado pelo utilizador e aberto num browser, com o intuito de apresentar um página.

Eddystone-TLM: contém informações sobre o beacon, como o estado da bateria e as coordenadas geográficas. Esta informação pode ser utilizada para fins de monitorização e manutenção.

Eddystone-EID: um identificador dinâmico que pode ser resolvido para um identificador estático.

Para um bom funcionamento do protocolo, os beacons têm de conter o Eddystone-UID ou o Eddystone-EID.Os outros dois campos são opcionais.

2.7.3 Smart Places

O Smart Places [27] é um projeto desenvolvido por Samuel Coelho com o objetivo de disponibilizar serviços, através de uma aplicação, baseados na proximidade do utilizador ao local dos mesmos. Estes serviços podem ser implementados em diversos locais, como restaurantes ou museus,

onde cada um é específico ao local. Desta forma, o utilizador utiliza apenas uma aplicação que integra os vários serviços.

O sistema é composto por:

- **Aplicação móvel para os utilizadores**: aplicação utilizada para aceder aos serviços disponibilizado por cada Smart Place.
- **Aplicação móvel para os proprietários:** aplicação utilizada pelos proprietários de um local onde estes podem configurar, listar, eliminar ou editar um Smart Place.
- **APIs de programador:** APIs utilizadas pelos programadores para criarem os serviços específicos a cada Smart Place
- **Servidor:** local onde podem ser encontrados os dados sobre cada Smart Place existente.
- **Beacons:** dispositivos utilizados para verificar a proximidade do utilizador ao local.

Cada Smart Place é criado por um proprietário através da aplicação móvel para proprietários. De seguida, é necessário colocar os beacons no local, cada um configurado com uma tag com uma finalidade específica.

Quando está na presença de um beacon, a aplicação móvel para utilizadores notifica o utilizador que este se encontra num Smart Place. A aplicação comunica uma primeira vez com o servidor para identificar o Smart Place a que o beacon pertence. Após este ser identificado, a aplicação comunica uma segunda vez com o servidor para assim apresentar ao utilizador os serviços que estão disponíveis no local onde este se encontra.

Neste projeto é importante observar a interação existente entre um beacon e uma aplicação móvel. Quando está na presença de um beacon, a aplicação móvel executa um processo onde inicia uma comunicação com o servidor, com o intuito de retornar informação sobre o local em que se encontra. Isto demonstra como é que os beacons podem ser utilizados juntamente com uma aplicação móvel de forma a detetar a presença de utentes dentro das unidades de saúde.

Também pode ser observado que o resultado final deste processo varia consoante o identificador do beacon. Isto significa que a aplicação móvel pode ter diferentes comportamentos ou executar diferentes processos consoante o identificador do beacon que detetou.

3 Solução

3.1 Tecnologias

Os principais sistemas operativos que encontramos instalados nos smartphones disponíveis no mercado são o Android e o iOS. Em ambas as plataformas podemos encontrar as tecnologias que foram discutidas na solução proposta. Contudo, devido a alguns métodos de segurança existentes no iOS, que não permitem a execução de tarefas por tempo indefinido, optou-se por utilizar exclusivamente o Android. Uma vez que a solução passa por manter o smartphone a detetar sinais de beacons por longos períodos de tempo, o iOS deixou de ser uma opção para realização de testes.

3.1.1 Bluetooth Low Energy

Sinal pelo qual os smartphones conseguem detetar os beacons que se encontram à sua volta. Este sinal foi configurado para que os smartphones detetassem apenas os sinais de beacons que fizessem parte deste projeto. A deteção de um beacon acontece apenas quando um smartphone entra no raio de alcance do sinal, acontecendo este evento apenas uma vez.

3.1.2 Wi-fi e GPS

Para garantir que o utilizador se encontra nas proximidades de um beacon, mesmo existindo uma deteção de sinal, são utilizadas informações como as redes Wi-fi que estão a ser detetadas pelo smartphone e as suas coordenadas GPS. Sabendo que redes se encontram no local onde o beacon foi colocado e através das suas coordenadas de GPS, podemos realizar uma comparação com os dados que são detetados pelo smartphone, em relação a cada um destes parâmetros, e perceber se a deteção ocorreu no local onde realmente o beacon se encontra instalado.

3.2 Descrição Funcional

O sistema desenvolvido é composto por vários processos que, em conjunto, permitem atingir os objetivos propostos para este projeto. Nesta secção serão descritos estes mesmos processos, assim como o papel que cada componente desempenha.

3.2.1 Registo e Autenticação de Utilizadores

Para restringir o uso do sistema apenas aos utilizadores da aplicação e do serviço de transporte gerido pelo SGTD, foi desenhado um processo de registo, que vai permitir a autenticação dos utilizadores nos outros processos existentes no sistema, que serão explicados adiante. Apenas os utilizadores com transportes agendados no SGTD é que conseguem efetuar o registo.

O processo de registo de utilizadores é iniciado na aplicação móvel. Na primeira execução da aplicação, após esta ser instalada, são apresentadas duas opções ao utilizador: para este iniciar sessão ou para efetuar o registo. Inicialmente, o utilizador deve escolher a opção de efetuar o registo, onde tem de inserir alguns dos seus dados pessoais, como o seu número de utente, uma palavra-chave e o seu número de telemóvel. Estes dados são enviados para o servidor, que confirma a existência de algum utilizador com o mesmo número de utente. Para além disso, o servidor também comunica com o SGTD para confirmar que o número de telemóvel fornecido também se encontra no sistema. Por fim, o utilizador é guardado na base de dados, onde a palavra-chave se encontra cifrada.

No fim deste processo, o utilizador pode autenticar-se, executando o início de sessão, tendo assim acesso aos serviços disponibilizados pelo sistema.

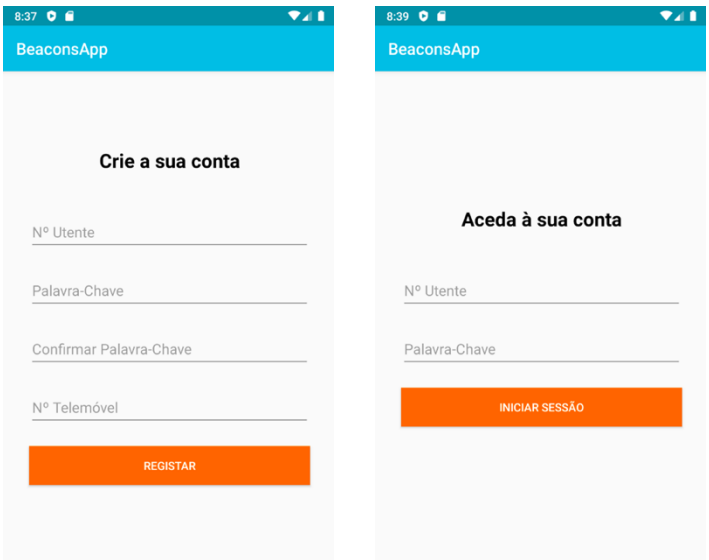

Figura 8 – Interfaces de Registo de Utilizador e de Autenticação

3.2.2 Deteção de Beacons e de Localização

Após o utilizador iniciar sessão, a aplicação começa o processo de deteção de beacons. Este processo está a ser executado independentemente do estado da aplicação, esteja esta em *foreground* ou em *background*. A partir desse momento, não é necessária mais nenhuma interação entre o utilizador e a aplicação. Nesse momento a aplicação móvel procura detetar beacons configurados com um identificador específico para este projeto. Isto permite que, apenas estes beacons despoletem o processo que é executado na sua deteção, não permitindo que tal aconteça com outros beacons pelos quais o utilizador possa passar.

No momento em que ocorre uma deteção, é iniciado o processo que regista no servidor a deteção efetuada pela aplicação. Este processo procura obter alguns dados sobre o local onde o utilizador se encontra, dados esses que incluem a sua coordenada geográfica, obtida através do sinal de GPS do smartphone, e as redes Wi-fi disponíveis no local. Após esta informação ter sido obtida, a aplicação envia um registo de deteção, composto pelo identificador do beacon, as coordenadas geográficas e as redes Wi-fi, para o servidor, que tem a função de validar os dados recebidos e guardá-los na base de dados.

A validação dos dados começa por verificar a existência de transporte para o utilizador no SGTD, pois caso este não exista, a deteção não é registada. Quando este existe, é retornado para o servidor o local de destino de transporte e prestação do mesmo. Com esta informação, o servidor consegue verificar se os dados enviados pela aplicação móvel permitem confirmar a presença do utilizador no local de destino de transporte. Em primeiro lugar, confirma-se que o beacon detetado pertence à unidade de saúde, através do identificador do mesmo. A seguir calcula-se a distância entre as coordenadas do local e as do utilizador. Por fim verifica-se se as redes Wi-fi que estão a ser detetadas pelo smartphone contêm a rede do local. Estas confirmações servem para aferir um grau de certeza em relação à deteção do beacon.

No servidor é criado um registo diário para cada utilizador com transporte agendado e que tenha detetado beacons. Este registo contém o identificador da prestação, o local de destino, a data do transporte e uma lista de deteções. Cada deteção contém a hora da deteção, o identificador do beacon detetado, as coordenadas de GPS do utilizador e a confirmação da deteção da rede Wi-fi local.

3.2.3 Registo de Horas no SGTD

O último processo a ser executado tem a função de registar as horas de entrada e de saída do utilizador no SGTD. Este processo é executado diariamente de madrugada, uma vez que existe uma maior probabilidade de não existirem utilizadores nas unidades de saúde, e é iniciado automaticamente pelo sistema. O servidor percorre todas os registos guardados no dia anterior, com o intuito de analisar as deteções efetuadas. Com esta análise, o servidor tem de ser capaz de aferir as horas de entrada e de saída do utente.

Para aferir sobre as horas de entrada, o servidor analisa as primeiras deteções registadas. O objetivo é perceber se existe confirmação da presença do utente através das suas coordenadas de GPS, no momento em que ocorreu a deteção. Para tal o servidor, ao verificar a primeira deteção, afere se existe confirmação de localização. Caso exista, o servidor envia para o SGTD as horas de entrada do utilizador, afirmando que existiu confirmação de localização. As horas de entrada são então registadas no sistema. Caso a confirmação não exista para esta deteção é então analisada a deteção seguinte. Neste caso é necessário garantir que o tempo que passou entre a primeira deteção e a segunda não é superior a 1 minuto. Se esta nova deteção apresentar confirmação de presença, é enviada para o SGTD a hora a que esta ocorreu. Este processo continua, até ser analisada uma deteção que tenha uma diferença temporal superior a 5 minutos, face à primeira deteção. Neste caso, é enviada para o SGTD a hora da primeira, mas aferindo que não existiu confirmação de presença. O intervalo de 5 minutos foi definido, para dar tempo ao smartphone de conseguir obter as coordenadas geográficas do utilizador.

A aferição das horas de saída é mais simples, uma vez que não é necessário informar o SGTD se existe confirmação de presença. Neste caso, a hora que é enviada para o SGTD é a hora da última deteção.

3.2.4 Visualização de Histórico de Transportes

Apesar da função principal da aplicação ser a deteção dos beacons e a transmissão de dados sobre o mesmo, foi implementada uma funcionalidade que informa o utilizador sobre transportes futuros e transportes já realizados. Com esta funcionalidade o utilizador pode visualizar a data e a hora do seu próximo transporte, assim como o local e o tipo de tratamento que se irá realizar. Para além disso o utilizador tem acesso ao seu histórico de viagem, onde pode consultar a que horas deu entrada e saída do local, bem como se estes valores foram obtidos de forma automática (através do sistema desenvolvido neste projeto) ou se foram obtidos da forma convencional.

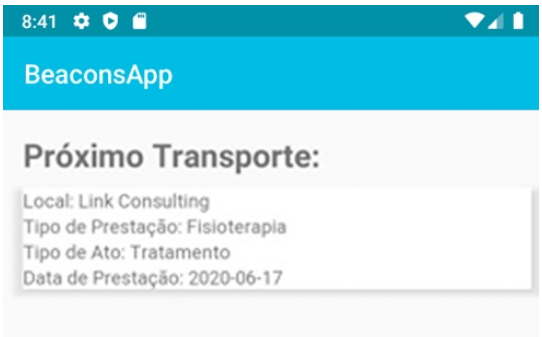

Figura 10 – Página De Próximo Transporte

3.2 Implementação

Esta implementação é composta por uma aplicação móvel, Servidor, Beacons e Sistema de Informação Externo (SGTD). Na Figura 9 pode ser observado um esquema da arquitetura do sistema e das interações existentes entre as várias componentes.

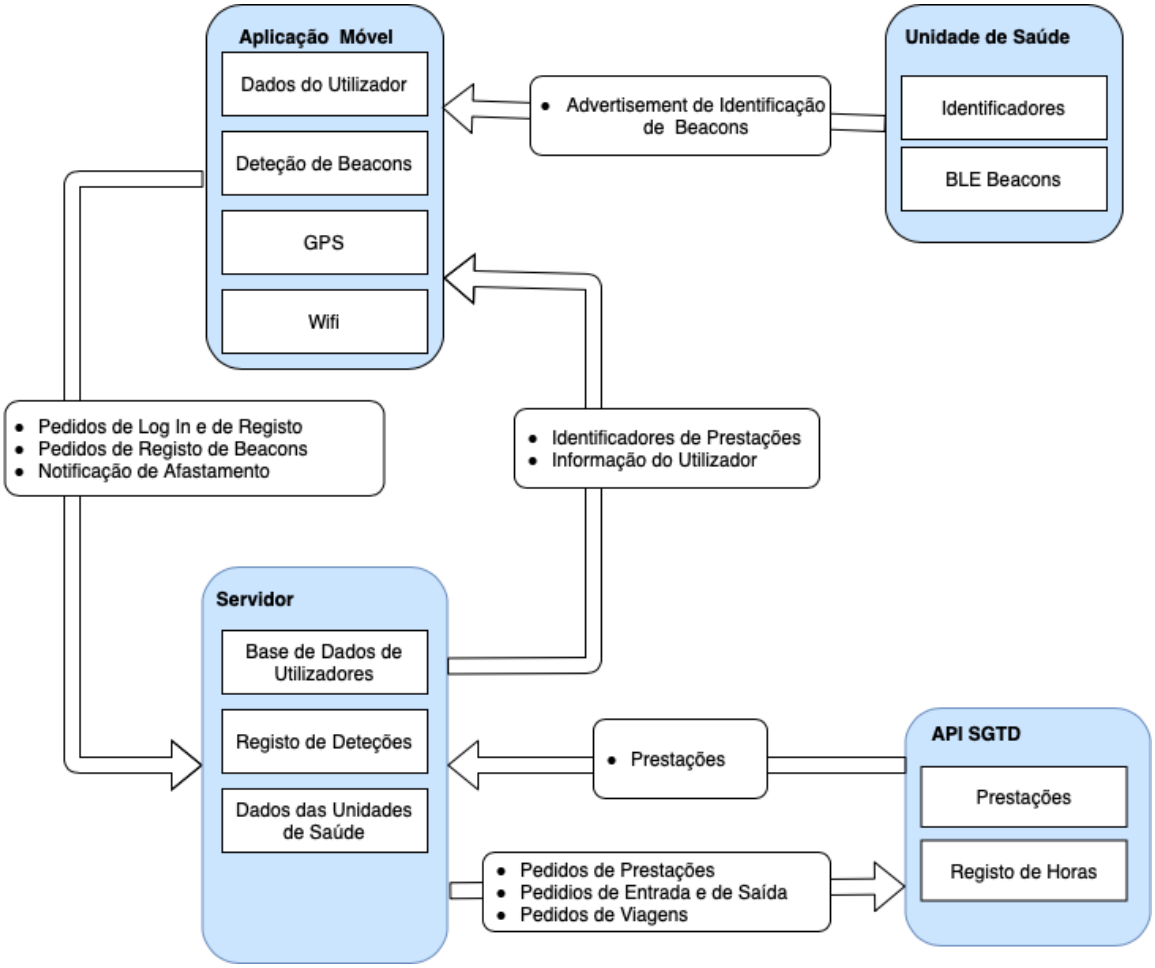

Figura 9 – Arquitetura da Solução

3.2.1 Utilizador / Aplicação móvel

Os smartphones Android contêm as tecnologias referidas anteriormente, necessárias para a implementação da solução. Inicialmente, foi eleita a Framework Ionic como ferramenta de desenvolvimento para Android. Esta Framework iria permitir desenvolver uma base de código executável por vários sistemas operativos, continuando a oferecer acesso a tecnologias nativas e com um desenvolvimento fácil e rápido. Contudo, apesar dos esforços para obter uma aplicação com níveis

de execução aceitáveis, tal não foi conseguido. Esta foi a razão pela qual o desenvolvimento da aplicação para Android foi de uma aplicação nativa, utilizando a linguagem de programação Java.

O grande desafio deste projeto recaíu na recolha contínua de dados de posicionamento do utilizador. Este processo deve ser executado, independentemente do estado da aplicação e do smartphone (excluindo a situação quando este está desligado). A aplicação pode encontrar-se em dois estados; *foreground,* quando a aplicação está a ser utilizada e podemos observar a sua interface no ecrã; *background*, quando não podemos visualizar a interface da aplicação, porque o utilizador está a utilizar outras funcionalidades do smartphone ou porque terminou o processo. A execução em *foreground* não cria nenhum constrangimento para o sistema operativo. A recolha de dados pode ser executada indefinidamente, uma vez que o sistema operativo assume que o utilizador tem consciência da utilização de recursos. Estando a aplicação em *foreground,* a prioridade do sistema operativo é disponibilizar os recursos necessários para o processo de recolha de dados. O mesmo não acontece quando a aplicação se encontra em *background*, onde o sistema operativo pode parar a execução da aplicação por esta não estar a ser utilizada pelo usuário. Esta gestão do sistema operativo faz com que seja necessário alterar as configurações do smartphone e desabilitar a opção de poupança de energia para esta aplicação.

Para detetar os beacons foi utilizada uma biblioteca chamada *Android Beacon Library* [28], desenvolvida pela *Radius Network*. Apesar desta biblioteca não ter sido implementada com o objectivo de detetar beacons configurados com o protocolo iBeacon, existem funcionalidades nesta biblioteca que o permitem fazer. O sistema operativo Android não tem funcionalidades que permitam realizar deteções do protocolo iBeacon, mas sim apenas de sinais BLE. Por isso seria necessário implementar toda a lógica para realizar a deteção deste protocolo. Comparada com outras bibliotecas disponíveis, esta permite detetar outro tipo de protocolos, não só o iBeacon, permite calcular a distância ao beacon, através da força do sinal recebido, mas acima de tudo, fornece serviços que permitam o funcionamento em background.

Monitoring e Ranging

A biblioteca utilizada no desenvolvimento da aplicação Android, para efetuar as deteções de beacons, disponibiliza dois processos de deteção, sendo estes o processo de *monitoring* e o processo de *ranging*.

No processo de *monitoring* as deteções são efetuadas quando o utilizador entra ou sai de uma região, ou seja, quando entra ou sai do alcance do sinal emitido pelo beacon. Este processo tem a vantagem de ser executado, independentemente do estado da aplicação.

O processo de *ranging* retorna uma lista de beacons que se encontram próximos do utilizador, assim como uma estimativa da distância, calculada através da força do sinal. Este processo apenas pode ser executado quando a aplicação está ativa, podendo o período entre leituras ser configurado, mas sendo geralmente um segundo entre leituras.

No contexto deste projeto, é necessário que as deteções ocorram quando a aplicação se encontra em background. Este objetivo apenas pode ser alcançado utilizando o processo de *monitoring*, uma vez que é este o processo que funciona independentemente do estado da aplicação, comportamento que não acontece no processo de *ranging*.

Para utilizar o processo de *monitoring* é necessário configurar a aplicação móvel com os identificadores de beacons que iram notificar a entrada e a saída da região. Visto que neste projeto existem vários beacons, cada um configurado com um identificador diferente, podemos utilizar o campo uuid na configuração da aplicação e detetar todos os beacons utilizados neste projeto, uma vez que o campo uuid está configurado em todos os beacons.

Contudo, existe um problema quando utilizamos esta abordagem de detetar os beacons apenas utilizando um dos seus campos uma vez que quando o processo de *monitoring* deteta a entrada e saída de uma região, apenas existe acesso ao campo configurado na aplicação, neste caso, o campo uuid.

Quando um beacon é detetado através do processo de *monitoring* a aplicação fica novamente ativa. Esta situação permite que processos como o de *ranging* possam ser utilizados neste momento. Por isso, a cada deteção utilizando o processo de *monitoring*, inicia-se a deteção utilizando o processo de *ranging*. Este processo permite obter toda a informação relativamente aos campos que constituem o identificador do beacon e não apenas o campo uuid.

```
public void onBeaconServiceConnect() {

          beaconManager.addMonitorNotifier(new MonitorNotifier() {

                    public void didEnterRegion(Region region) {
                              beaconManager.startRangingBeaconsInRegion(region);
                    }

          });

          beaconManager.addRangeNotifier(new RangeNotifier() {

                    public void didRangeBeaconsInRegion(beacon) {
                              sendRecord(beacons.getId1,
                                        beacons.getId2(),
                                        beacons.getId3());

                              beaconManager.stopRangingBeaconsInRegion(region);

                     }
          });


          beaconManager.startMonitoringBeaconsInRegion(region);

}
```

Figura 12 – Pseudo Código da Implementação das Componentes de Deteção de Beacons

Na Figura 12 pode ser observado o *pseudo* código implementado na aplicação móvel. O desenvolvimento consistiu na utilização de dois *listeners* que efetuam deteções de *monitoring* e de *ranging*. Cada um destes *listeners* entra em execução após serem executadas as funções *startMonitoringForBeaconInRegion*() e *startRangingForBeaconsInRegion*(), respetivamente. Desta forma, o processo de deteção de beacons é iniciado através da deteção por *monitoring*. Neste momento, a aplicação móvel apenas é sinalizada quando entra na região de um beacon que contém o campo uuid utilizado neste projeto. Ao entrar numa região, a aplicação executa o processo de *ranging*, onde podem ser obtidos mais dados sobre o beacon inicialmente detetado. Ao obter dados sobre este beacon, a aplicação termina novamente o processo de *ranging* e envia para o servidor os dados que obteve no momento da deteção. No processo de *ranging*, após 5 leituras consecutivas onde não são obtidos dados, o processo é então terminado, para evitar que a aplicação continue indeterminadamente neste estado de leitura.

Nos registos que se encontram no servidor relativamente às deteções efetuadas durante a fase de teste, pode ser observado que existem registos consecutivos com diferenças de tempo na ordem das milésimas de segundo. Este comportamento pode ser justificado devido a leituras consecutivas do processo de *ranging*. Uma vez que este processo é terminado quando existe uma resposta de sucesso por parte do servidor, até ser obtida esta resposta, podem ocorrer novas leituras, sendo enviados novos registos para o servidor. Este comportamento será abordado com mais profundidade na secção seguinte.

3.2.2 Servidor

A aplicação do servidor foi desenvolvida em Node.js devido às suas capacidades de processamento rápido e na forma como lida com pedidos simultâneos. Estas características permitem que múltiplos utilizadores enviem pedidos concorrentes e que estes sejam processados de forma eficiente e com tempos de resposta reduzidos.

A aplicação desenvolvida está instalada no Microsoft Azure. Este serviço de *cloud* permite que o servidor esteja disponível para qualquer utilizador da aplicação móvel, conseguindo criar testes reais de deteções de localização.

A função do servidor é de receber e processar os pedidos enviados pela aplicação móvel. Neste sentido foram criadas diversas APIs, que permitem ao utilizador registar-se no sistema, iniciar sessão e enviar os dados relativos à sua localização.

JSON Web Token

A aplicação de servidor está instalada na *cloud*, no serviço Microsoft Azure. Esta decisão surgiu com a necessidade de os serviços oferecidos pela aplicação necessitarem de estar disponíveis ao público, através de APIs. Contudo, estes serviços apenas devem estar disponíveis para os utilizadores da aplicação móvel desenvolvida. Para tal, foi desenvolvido um mecanismo de segurança para que o servidor rejeite outros pedidos externos.

O método desenvolvido consiste na utilização de JWT [29]. JWT é um *standard* que define um mecanismo seguro de transmissão de dados através de objetos JSON (token). Um dos cenários onde este standard é bastante útil é em situações de verificação de autorização, cenário esse que deve ser considerado pois as APIs do servidor são públicas.

O token é composto por três partes; *header*, *payload* e assinatura, onde cada uma se encontra cifrada utilizando Base64, separada por um ponto, tendo o seguinte formato xxx.yyy.zzz .

Header: contém informação sobre o algoritmo de assinatura e o tipo de token.

Payload: contém informações sobre o utilizador e outro tipo de dados, como a validade do token.

Assinatura: aplicação do algoritmo de assinatura, que se encontra no *header*, às outras componentes do token. Exemplo utilizando o algoritmo HMAC SHA256: HMACSHA256(base64Encode(header) + “.” + base64Encode(payload), segredo).

Todos os pedidos efetuados ao servidor devem conter este token, para confirmar se o emissor tem autorização de acesso, à exceção dos pedidos de registo e autenticação (início de sessão).

Contudo, o uso desta tecnologia como mecanismo de verificação de autorização acrescenta algumas vulnerabilidades, incluindo a possibilidade de ataques *man in the middle*. Estes tipos de ataques podem ocorrer para que o atacante obtenha o token que está a ser transmitido na comunicação

entre o cliente e o servidor. Ao obter o token, o atacante consegue iniciar uma comunicação com o servidor, fazendo-se passar pelo cliente e, assim, obter autorização para aceder aos recursos do servidor.

Desta forma existem duas implementações que são necessárias realizar, que tentam mitigar a existência deste problema. Em primeiro lugar os tokens que são gerados pelo servidor devem conter uma validade. Esta validade permite que, mesmo que o token tenha sido obtido por um atacante, este não seja possível de ser utilizado após expirar. A eficácia deste método varia consoante o tempo de validade do token, pois se o token se encontra válido por grandes períodos de tempo, maior será o tempo de utilização do mesmo por parte do atacante. Para este projeto, o valor do tempo de validade escolhido foi de um dia. Desta forma a aplicação pode requisitar o token no início de cada dia e o utilizador teria autorização para aceder ao servidor durante a sua estadia na unidade de saúde, independentemente da hora.

Uma vez que o token foi obtido por um atacante, existe uma grande probabilidade de o ataque ser executado novamente com sucesso. O atacante consegue obter o token porque o canal de comunicação utilizado nesta implementação entre o cliente e o servidor não é seguro. O protocolo de comunicação utilizado na implementação deste projeto é o protocolo HTTP, onde no cabeçalho da mensagem se inclui o token de autorização. Uma vez que o canal de comunicação não é seguro, os pedidos HTTP podem ser intercetados por um atacante que se encontre entre o cliente e o servidor que desta forma consegue obter o token e utilizá-lo em comunicações futuras.

Uma alteração que deveria ser implementada neste projeto é a utilização de um protocolo de comunicação mais seguro, o HTTPS. Este protocolo é uma implementação do HTTP utilizando uma camada de segurança adicional, que utiliza o protocolo SSL/TLS, sendo este um protocolo que visa fornecer segurança na comunicação numa rede de computadores. O uso do HTTPS permite realizar pedidos ao servidor, continuando a enviar o token de autorização no cabeçalho da mensagem, e mantendo o seu conteúdo confidencial, uma vez que estes conteúdos são cifrados e um atacante não os consegue visualizar. Desta forma garante-se que as comunicações efetuadas com o servidor, quando não são efetuadas sobre um canal de comunicação seguro, não estão sujeitas a ataques *man in the middle*, não permitindo ao atacante obter o token de autorização de um utilizador e fazer-se passar pelo mesmo para conseguir aceder aos serviços disponibilizados.

Registo e Início de Sessão

O registo e início de sessão são dois processos simples, implementados em praticamente todos os serviços disponíveis online e por isso já bastante usuais. O processo de registo visa guardar os dados que foram requisitados ao utilizador no sistema, para depois serem comparados com os dados que são recebidos quando o utilizador deseja autenticar-se. Caso seja efetuada a autenticação com sucesso, o servidor gera um JWT *token* que é enviado para a utilizador. A geração do *token* é feita utilizando dados do utilizador, que são cifrados pelo servidor. Apenas este consegue decifrar e verificar os dados que o constituem.

Registo de Deteções

Ao receber um pedido de registo de deteção, o servidor verifica a existência do *token* de autorização na mensagem. Este processo consiste na decifragem do *token* e verificação da sua validade e autenticidade. Os dados presentes no token devem pertencer ao utilizador, emissor do pedido. Caso um destes passos não seja executado com sucesso, é enviada uma resposta de erro.

Em caso de sucesso, o servidor cria um registo da deteção do utilizador que contém o beacon detetado, as coordenadas GPS, as redes Wi-Fi, bem como a data e a hora a que ocorreu a deteção.

3.2.3 Beacons

Neste projeto, cada beacon foi configurado com um ID distinto, que tem como objetivos identificar este projeto, assinalar diferentes espaços interiores que se encontram num edifício e reconhecer em que edifício os beacons se encontram. Dependendo do espaço ou do edifício onde se encontra o beacon, é definido um ID específico para cada caso.

O protocolo iBeacon é utilizado, uma vez que a aplicação móvel apenas tem de conter um identificador de beacons. Todos os beacons a utilizar são configurados com o mesmo uuid, de forma a representar este projeto. A aplicação apenas tem de conter este identificador e consegue detetar todos os beacons.

Uma vez que os beacons utilizados neste projeto são os Estimote Proximity Beacons 2014 [23], foi ponderada a utilização da tecnologia proprietária de processamento de sinal Bluetooth, que tem como objetivo efetuar a deteção de beacons configurados com um protocolo específico da Estimote. Esta tecnologia de processamento pode ser integrada na aplicação, utilizando o Estimote Proximity Software Development Kit (SDK). Cada beacon que é adquirido pode ser registado na conta Estimote do proprietário, num serviço chamado Estimote Cloud. Neste serviço é possível criar um *token*, que, mais tarde, vai ser configurado na aplicação móvel e utilizado para autenticar a aplicação, no momento em que se realiza uma deteção. Esta autenticação é feita pela Estimote Cloud, através da adição de uma camada de segurança. Foi, no entanto, verificado que, com a utilização do SDK e utilizando a configuração proprietária referida anteriormente, a ocorrência de deteções dos beacons era muito inferior às deteções que ocorriam quando os mesmos estavam configurados segundo o protocolo iBeacon. Por este motivo optou-se por manter a utilização do protocolo iBeacon.

3.2.4 SGTD

O SGTD é o local onde podemos encontrar toda a informação necessária sobre os utentes e os respetivos transportes. Neste projeto, o SGTD tem o papel de confirmar a identidade dos utilizadores e de retornar dados sobre os transportes dos mesmos.

Como foi referido na secção 2.2, o SGTD oferece um conjunto de APIs que podem ser utilizadas no desenvolvimento desta solução. Estas APIs seguem uma arquitetura REST e podem ser acedidas através de pedidos HTTP.

Validação do Número de Telemóvel

Ao realizar o registo, é pedido ao utilizador para fornecer o número de telemóvel. O SGTD contém um serviço que permite confirmar a existência do número de telemóvel no próprio sistema. Desta forma pode-se garantir que apenas os utilizadores conhecidos pelo SGTD, podem ter acesso à aplicação desenvolvida neste projeto.

Obter Prestação de Transporte

Um utilizador ao detetar um beacon, inicia um processo de registo dessa deteção no servidor. Para garantir que apenas os utilizadores, beneficiários do serviço de transporte, irão criar registos de deteções no sistema, é utilizado um serviço que verifica a existência de uma prestação para um determinado dia. Na resposta a esse pedido podemos obter o identificador da prestação, assim como o local de destino de transporte, caso esta exista. Este serviço utiliza o número de telemóvel do utilizador para confirmar a sua identidade.

Próximo Transporte e Histórico de Transporte

No SGTD existem dois serviços, com um carácter mais informativo para o utilizador, onde podem ser obtidos dados sobre os transportes realizados e a realizar. Apesar destes serviços não influenciarem a solução e o objetivo relacionado com as deteções, a sua utilização torna a aplicação móvel mais útil para o utilizador. Um utente que irá utilizar um transporte gerenciado pelo SGTD passa a ter acesso a dados sobre o seu próximo transporte, dados esses que incluem a data e hora do transporte, assim como o local de destino, que podem ser úteis para o utilizador.

Registo de Hora de Entrada e Hora de Saída

Estes dois pedidos servem para o sistema desenvolvido aferir a hora de entrada e de saída de um utente de uma unidade de saúde. Após ser realizado o transporte e o utilizador ter criado diversos registos de deteção de beacons, é necessário transpor esses dados para o SGTD e assim continuar o processo de gestão do transporte. Nestes pedidos são comunicados a prestação do transporte efetuado pelo utilizador e a horas de cada um dos eventos.

4 Avaliação

De forma a avaliar o funcionamento do sistema, foi desenhado um teste real no edifício da empresa Link Consulting. O objetivo deste teste era mostrar que as deteções continuavam a ocorrer mesmo num ambiente não controlado e que poderiam ser obtidos dados que conseguissem demostrar a atividade dos utilizadores ao longo do dia. Este teste foi executado pelos trabalhadores da Link Consulting ao longo de vários meses, onde foi possível realizar uma avaliação à performance do sistema e perceber quais os erros existentes.

4.1 Descrição do Teste

Este teste foi realizado no edifício da Link Consulting, local esse que permitiu colocar beacons em diferentes espaços, para assim observar a movimentação dos trabalhadores por todo o edifício. O método utilizado para a instalação dos beacons no local foi um beacon por piso.

Devido à existência de um acesso único a todos os pisos, garantiu-se que, desde a chegada de um trabalhador ao edifício até ao momento que chega à sua secretária, este iria aproximar-se de pelo menos dois beacons; um beacon que foi colocado na entrada do local e outro que se encontrava no piso onde se encontrava o trabalhador. Com esta simples deslocação, era possível aferir a hora de chegada do trabalhador assim como o fluxo existente dentro do edifício.

Os resultados obtidos seriam então confirmados pelos próprios trabalhadores. Este exemplo é capaz de simular a chegada e a saída de um utente de uma unidade de saúde.

4.2 Configuração do Beacons

Tal como descrito na solução, o objetivo dos beacons é de identificarem o serviço de localização (no caso do deste projecto), o local físico onde se encontra e uma secção específica desse local. Para tal, os beacons foram configurados com o uuid, um major e um minor de forma a cumprirem estes três objetivos.

O edifício da Link Consulting é constituído por sete pisos, sendo que, destes sete, foram escolhidos cinco pisos. No acesso a cada um dos cinco pisos foi colocado um beacon, estando todos configurados com o mesmo uuid. Um esquema da colocação dos beacons no local pode ser observado na Figura 11.

Ao compararmos este edifício com algumas unidades de saúde, sendo estes hospitais ou centros de saúde, observamos que em termos de dimensão, o edifico da Link Consulting é de menor

dimensão. Por este motivo, o valor do major de cada beacon não seguiu a configuração que foi proposta anteriormente, mas sim uma bastante mais simples que se adapta melhor às características do edifício. Existindo um beacon por cada piso, cada um foi configurado com um major igual ao número do piso onde se encontrava; o beacon que se encontrava no piso um foi configurado com o major 1, o que se encontrava no piso dois com o major 2 e daí por diante.

Por fim, como queremos identificar todos estes beacons como pertencentes ao edifício da Link Consulting, o mesmo valor de minor foi atribuído a todos eles.

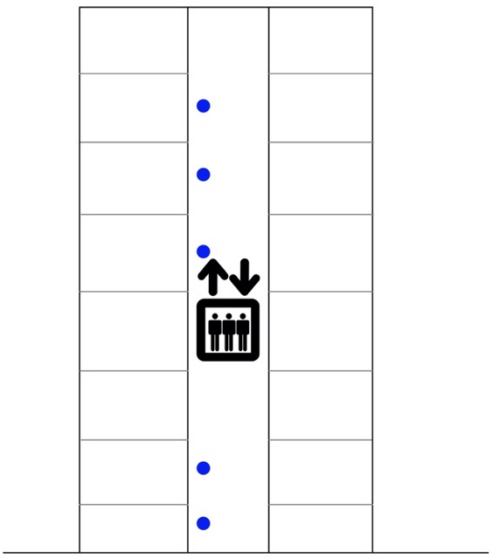

Figura 11 – Colocação dos Beacons no Edifício da Link Consulting

4.3 Transportes de Utilizadores

Para cada utilizador participante no teste, foram criadas prestações do serviço de transporte no SGTD. Estas prestações foram utilizadas para simular o transporte de um utente, e para cada utilizador foi gerada uma prestação diária, ao longo do período de teste.

As prestações de transporte são utilizadas como um acesso ao serviço de deteção, que apenas cria registos das mesmas no servidor quando existe uma prestação para o dia. Isto garante que uma simples deslocação de um utilizador a uma unidade de saúde (seja no acompanhamento de um familiar ou numa situação onde o serviço de transporte não está a ser utilizado) não gere, no servidor, registos de deteções de beacons.

Cada deteção que é feita por um utilizador é enviada para o servidor, que depois verifica com o próprio SGTD se existe alguma prestação para aquele dia. Caso exista, a deteção é registada ou, caso contrário, o processo é abortado.

4.4 Confirmação de localização dos utilizadores

De forma a confirmar a localização dos utilizadores, mesmo com a existência de deteções de beacons configurados para identificarem o edifício, foram utilizadas informações exclusivas à localização da Link Consulting. As características utilizadas neste processo de confirmação foram as coordenadas de GPS do edifício e a rede Wi-fi que se encontra instalada no mesmo.

Este processo é realizado através do registo destes dados no servidor e do local da prestação do serviço de transportes, que pode ser encontrado no SGTD. O utilizador ao efetuar uma deteção, envia para o servidor as coordenadas de GPS que estão a ser obtidas no smartphone, assim como uma lista das redes Wi-fi. O servidor pode assim comparar estes dados com os referentes ao local de prestação.

Na confirmação por GPS, o intervalo de certeza que justifica a presença do utilizador no local, deve variar consoante o edifício onde o sistema se encontra implementado. No caso da Link Consulting, que contém uma área de implantação relativamente pequena (entre os 550m^2 e 650m^2), o intervalo de certeza não deve ser superior a 250m. Este valor deve ter em conta o erro de posicionamento obtido na componente de GPS do smartphone devido à atenuação do sinal, uma vez que os utilizadores se encontram num local fechado. Em contrapartida, noutros locais onde a área de implantação é maior, o intervalo de certeza também deve ser maior. Este é um dos dados que tem de estar guardado no servidor relativamente às unidades de saúde.

Por fim, a cada registo de deteção enviado para o servidor, este compara a distância entre o ponto geográfico enviado pelo servidor e a coordenada referente ao destino de transporte. Caso esta distância seja inferior ao intervalo de certeza, o registo é efetuado. Caso a distância seja superior ao intervalo o registo pode ser assinalado, informando que a posição não foi confirmada, ou pode ser rejeitado.

Neste projeto, como era importante perceber o funcionamento da deteção de beacons e quantos registos eram efetuados ao longo do tempo, decidiu-se apenas assinalar as deteções que não respeitassem o intervalo de certeza. Se os registos fossem rejeitados, não era possível efetuar uma análise detalhada sobre a deteção de beacons.

4.5 Teste 1 – Aplicação Desenvolvida em Ionic

Tal como foi planeado na proposta para este projeto, foi desenvolvida uma primeira aplicação móvel utilizando a Framework Ionic. Este teste contou com a participação de nove trabalhadores da Link Consulting, que se voluntariaram para realizar esta atividade.

Esta solução esteve ativa durante um mês, tendo os utilizadores obtido zero deteções neste período. Neste caso, para além do mau funcionamento do processo de deteção de beacons, não foi possível perceber o comportamento dos sensores de GPS e de Wi-fi do smartphone.

O Ionic tem a particularidade de a implementação para uma plataforma, Android ou iOS, não tem de ser específica para uma determinada versão do sistema operativo. Uma das razões possíveis para o insucesso deste desenvolvimento poderá ter sido a falta de configurações efetuadas para criar uma aplicação executável para o smartphone. As aplicações desenvolvidas em Ionic, depois de criadas, podem ser importadas para os IDEs utilizados para o desenvolvimento de ambos os sistemas operativos, Android Studio e Xcode. Quando é importada para cada um dos IDEs, seria necessário configurar alguns parâmetros da aplicação para que esta funcionasse e fosse capaz de realizar as deteções dos beacons.

Para além disso, no momento em que foi implementada a aplicação, ainda eram parcos os conhecimentos relativos ao desenvolvimento de aplicações móveis. A falta de conhecimento relativo ao desenvolvimento de aplicações nativas agravou ainda mais este processo. Só depois de ter sido efetuado o teste e ter sido ponderada a abordagem que será descrita na seção seguinte, é que se entendeu as limitações da execução em background, situação que também será descrita na seção posterior. Apesar do mau funcionamento da aplicação ter sido descoberto numa fase inicial do teste, este continuou a ser executado ao mesmo tempo que se iam efetuando melhorias, que permitissem mitigar os erros que ocorriam.

4.6 Teste 2 – Aplicação Nativa para Android

Devido aos resultados insatisfatórios obtidos no Teste 1, uma nova estratégia foi desenhada de forma a ser possível obter deteções de beacons. Com esta abordagem esperava-se que a inexistência do *middleware* entre o código da aplicação e os sensores do smartphone trouxesse um aumento do desempenho no processo de deteção de beacons.

4.6.2 Análise de Resultados

Este teste teve a particularidade de ser efetuado durante a pandemia causada pelo Covid-19, reduzindo para apenas um, o número de trabalhadores da Link Consulting a realizar o teste. Contudo, o dia-a-dia deste trabalhador manteve-se semelhante ao dos tempos antes do vírus, pelo que se mantiveram as condições encontradas no primeiro teste.

Na Tabela 1 podem observar-se os resultados das deteções, por utilizador, que ocorreram durante o período de teste na Link Consulting. Neste teste, que teve duração de um mês, foram registados oito dias onde ocorreram deteções.

Data	Nº de deteções	Primeira Deteção	Hora de Entrada	Ultima Deteção	Hora de Saída
12/06/20	18	Entrada	09:34:43	Piso 3	12:02:16
18/06/20	26	Piso 3	11:58:38	Entrada	17:25:58
19/06/20	47	Entrada	10:04:10	Piso 3	18:51:53
07/07/20	5	Entrada	15:02:53	Piso 3	16:48:28
13/07/20	25	Entrada	09:41:14	Piso 3	17:07:13
14/07/20	19	Entrada	08:33:05	Piso 3	17:23:30
15/07/20	25	Entrada	08:37:50	Piso 3	17:39:45
16/07/20	30	Entrada	09:12:45	Entrada	12:33:20

Tabela 1 – Resultados dos registos de deteções diários

Através dos dados existentes na tabela é possível retirar algumas informações como as deteções que ocorreram durante um dia, o local da primeira e da última deteção, assim como as horas a que estas ocorreram.

A primeira deteção que ocorre num dia acontece maioritariamente no local que se encontrava sinalizado como sendo a entrada do edifício. Esta informação é nos dada pelo campo major configurado nos beacons. O resultado que se obteve na primeira deteção vai de encontro ao que era esperado deste teste, uma vez que a entrada do edifício é o local onde existe uma maior probabilidade de contacto entre o utilizador e os beacons. Estes resultados mostram que, na chegada do utilizador ao local de destino, a aplicação é capaz de realizar a deteção no local onde se encontra sinalizada a entrada, demonstrando assim a eficácia da aplicação no processo de entrada.

Uma vez que este teste teve a participação de um trabalhador da Link Consulting, esperava-se que a hora da primeira deteção ocorresse de manhã, num horário usual de início de trabalho. Dos resultados obtidos, em apenas seis dos oitos dias, a hora da primeira deteção se assemelhou ao horário do trabalhador. Isto significa que a aplicação móvel não foi sempre capaz de realizar deteções na chegada do utilizador ao edifício porque: estava em execução, mas não foi capaz de realizar a deteção quando o utilizador passa por um beacon, ou porque a sua execução foi interrompida. Estes dados foram confirmados pelo utilizador do teste, sendo que o mesmo afirmou que, nos primeiros dias, desligou o processamento da aplicação e que posteriormente voltou a iniciá-la. Estes resultados podem concluir que uma vez que a execução da aplicação é terminada pelo utilizador, esta não é capaz de registar novas deteções.

Relativamente à última deteção registada de cada dia, estas ocorreram maioritariamente no Piso 3, onde se encontrava o local de trabalho do utilizador. De forma semelhante à primeira deteção, esta era de prever que ocorresse maioritariamente na entrada do edifício, uma vez que este seria o local mais provável para a saída do utilizador. Apesar de a última deteção não ter ocorrido na entrada do edifício, este comportamento é justificado pelo facto de a aplicação não se encontrar constantemente a realizar *scans* aos beacons. Isto traduz num intervalo entre 5 e 10 minutos que pode ocorrer, entre eventos de *monitoring*.

Quando a aplicação se encontra em background, o tempo que ocorre entre scans de beacons é de 15 minutos, segundo as configurações standard. Uma vez que ocorre uma deteção no piso onde o utilizador trabalha, seriam necessários 15 minutos para obter uma deteção na entrada do edifício

quando o utilizador saísse do local, o que explica a falta desta deteção. É de notar que este intervalo pode ser configurado para ter um tempo menor, mas que os gastos de bateria seriam bastante superiores aos normais. Para este teste, verifica-se que, apesar da última deteção não ter ocorrido na entrada, ocorreu no piso do trabalhador, sendo que os gastos de bateria do smartphone não compensavam a existência desta deteção.

De forma semelhante às horas de entrada, as horas de saída ocorrem maioritariamente ao fim do dia, hora a que o trabalhador sai do local de trabalho. Mais uma vez, apesar de a última deteção não ter ocorrido no local de entrada, estas horas permitem perceber o funcionamento do sistema quando o utilizador sai do local, uma vez que a aplicação consegue realizar deteções que mostram essa movimentação.

Para analisar de forma mais detalhada as deteções que ocorrem ao longo do dia, é apresentada a Tabela 2 onde podem ser observadas as deteções reais que ocorreram num dia exemplo - dia 14/07/2020 - pelo utilizador de teste.

Ao serem analisadas as horas a que cada deteção ocorre, observa-se que existem vários registos consecutivos que apresentam o mesmo tempo de deteção ou que se distanciam entre si por meros segundos. Começando esta análise, observam-se os registos 1 e 2, onde ambos ocorreram no mesmo local e apresentam horas de deteção iguais. Estes resultados são o efeito da aplicação móvel, processo esse que foi descrito na seção 4.6.1. Uma vez que o processo de *monitoring* deteta uma entrada no alcance do sinal do beacon, é iniciado o processo de *ranging*. Este processo apenas é terminado quando é enviado um registo de deteção para o servidor e este responde com uma confirmação de sucesso. Contudo, a leitura de beacons que se encontram ao alcance do servidor não é terminada e, cada vez que um beacon é detetado, um novo registo é enviado para o servidor.

Registo	Hora de Deteção	Local de Deteção	Confirmação GPS
1	09:33:05	1	Sim
2	11:20:42	1	Não
3	11:20:42	1	Não
4	11:20:42	1	Não
5	11:20:59	1	Sim
6	11:23:35	1	Sim
7	11:23:42	1	Sim
8	11:59:57	7	Sim
9	11:59:57	7	Sim
10	13:04:11	1	Não
11	13:04:57	7	Não
12	13:04:57	7	Não
13	13:17:05	7	Sim
14	13:45:00	7	Sim
15	14:07:16	1	Sim
16	15:18:17	1	Não
17	18:23:14	7	Sim
18	18:23:30	7	Sim
19	18:23:30	7	Sim

Tabela 2 – Registos do dia detalhados

É necessário não terminar a deteção de beacons no momento em que é detetado o primeiro, uma vez que qualquer erro que possa ocorrer durante a comunicação com o servidor irá resultar numa deteção não registada. Assim, apesar de existirem mais registos relativos à mesma passagem do utilizador pelo beacon, garante-se a persistência de cada uma.

Por fim, um ponto que ficou aquém das expectativas desejadas foi a execução da componente de confirmação de localização, através dos dados obtidos pela componente de GPS. Na Tabela 2 são apresentadas as deteções onde o smartphone foi capaz de obter as coordenadas geográficas do utilizador. Das dezanove deteções que foram realizadas pelo utilizador, apenas em doze o smartphone foi capaz de enviar a latitude e longitude para o servidor. Na Figura 13 encontra-se o pseudo código da execução da componente de localização na aplicação móvel.

```
public void configureLocation(){

        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

        locationManager.requestLocationUpdates();
}

public void sendRecord(String uuid, String major, String minor) {

        GpsDTO gps = new GpsDTO();

        Location location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if(location != null) {
            gps.setLatitude(String.valueOf(location.getLatitude()));
            gps.setLongitude(String.valueOf(location.getLongitude()));

        }

}
```

Figura 13 – Pseudo Código de Implementação da Componente de GPS

Esta componente é iniciada na criação do ecrã principal da aplicação, pela função configureLocation(). Esta função tem o objetivo de criar um objeto do tipo LocationManager, objeto esse que fornece acesso aos serviços de localização do smartphone. Ao longo da execução da aplicação, vão sendo obtidas actualizações relativas à localização do utilizador, localização essa que fica registada no smartphone. Na função de envio do registo de deteção para o servidor, através da função *sendRecord(),* o objeto *LocationManager* executa o método *getLastKnownLocation()* para obter a última localização registada. Esta localização pode ter sido obtida recentemente pelo serviço, sendo que o utilizador deve encontrar-se ainda próximo da mesma, ou pode ter sido obtida à bastante tempo, fazendo com que a localização perca a validade uma vez que não existe certeza se o utilizador ainda se encontra perto do local ou não. Existem várias razões que fazem questionar a eficácia desta componente, começando pela escolha do fornecedor de localização (LocationProvider). Existem dois fornecedores que serviço que poderiam ter sido utilizados para obtenção de dados de localização sendo estes o GPS_PROVIDER e o NETWORK_PROVIDER.

O GPS_PROVIDER obtém a localização do utilizador através de satélites enquanto que o NETWORK_PROVIDER obtém esta informação através das torres de comunicação e redes de Wi-fi disponíveis. Neste projeto foi utilizado o GPS_PROVIDER por existir uma maior precisão nos dados obtidos. Contudo, uma vez que os utilizadores se encontram dentro de edifícios quando são realizadas deteções de beacons, nem sempre é possível obter atualizações da sua localização. Para tal, deveria ter sido utilizado o NETWORK_PROVIDER, uma vez que num local fechado, os sinais das torres de comunicação e das redes Wi-fi são facilmente obtidos. Outra razão que pode ter levado a esta obtenção de resultado foi a escolha do método utilizado para obter a localização. Como referi anteriormente, o método utilizado, *getLastKnownLocation()*, obtém a última atualização de localização do utilizador.

Estes dados encontram-se muitas vezes desatualizados, uma vez que o smartphone tem dificuldades em obter a informação dos satélites, não conseguindo manter a localização atualizada, tornando-se assim inválida. Em vez deste método, poderia ter sido utilizado uma função que procura retornar a posição atual do utilizador. Este método não foi escolhido e implementado na aplicação uma vez que o seu funcionamento em *background* tem tendência em falhar e a não conseguir retornar a localização do utilizador. Uma vez que o smartphone dos utilizadores se encontra bloqueado durante longos períodos de tempo, existe uma grande probabilidade de este método não retornar a localização. O intervalo de certeza, utilizado para confirmar a presença dos utilizadores no local, permitiu que todos os registos de deteção que continham informação sobre a localização fossem validados com sucesso. Esta validação é feita por uma biblioteca disponível para Node.js, que calcula a distância entre duas coordenadas geográficas. Existem várias bibliotecas com este propósito, mas neste projeto foi utilizada a “Geolib”.

4.7 Desenvolvimento Nativo para iOS

Após o sucesso da aplicação nativa para Android, quando comparada com o desenvolvimento em Ionic, iniciou-se o desenvolvimento uma aplicação nativa para iOS. O passo inicial foi começar a desenvolver um processo responsável pela deteção dos beacons, percebendo ao mesmo tempo que limitações do sistema operativo iam afetar o desenvolvimento.

Os testes que começaram a ser feitos foram realizados com uma aplicação que tinha como única função detetar beacons. Nesta fase, foram ignoradas todas as outras funcionalidades que podem ser encontradas na aplicação para Android como o registo de utilizadores e o processo de autenticação.

Observou-se que a aplicação era capaz de detetar beacons com bastante precisão, pois estas ocorriam sempre que o smartphone se aproximava destes dispositivos. É de notar, que este processo começou por ser testado com a aplicação em *foreground*.

Como já foi referido na secção 3.1, o desenvolvimento em iOS teve de ser abandonado devido à execução da aplicação quando esta se encontrava em *background*. Quando era aberta no smartphone outra aplicação, a execução era interrompida, pelo que deixavam de existir deteções de beacons. O mesmo acontecia quando o smartphone era bloqueado, tornando o desenvolvimento em

iOS irrelevante para este projeto. Em Android este problema foi contornado uma vez que se desativou a opção de poupança de bateria na execução de uma aplicação em específico, opção essa que não podemos encontrar no sistema operativo iOS.

No desenvolvimento para iOS existem os mesmos processos de deteção de beacons, processos esses que podem ser utilizados no desenvolvimento, ao importar a biblioteca CoreLocation. Estes processos assemelham-se aos encontrados no desenvolvimento para Android: o *monitoring* e *ranging*. O comportamento destes processos equipara-se ao comportamento em Android, pelo que a lógica de programação utilizada no desenvolvimento para iOS foi idêntica ao Android.

Por fim implementaram-se algumas definições para que a aplicação continuasse a ser executada, mesmo quando se encontrava em background, e para que as deteções de beacons continuassem a ocorrer.

Inicialmente, implementou-se o pedido de autorização para utilizar as funcionalidades de localização (onde se encontram também as funcionalidades de deteção de beacons). Estas autorizações permitem a utilização dos serviços de localização em qualquer momento, estando a aplicação a ser utilizada, mesmo quando o smartphone se encontra bloqueado.

Uma vez que esta configuração não melhorou o funcionamento em background, foi modificada uma configuração dos serviços de localização que permite a atualização da localização quando a aplicação se encontra em background. Esta configuração inclui a utilização dos serviços de Bluetooth.

Por fim, uma vez que nenhuma das tentativas anteriores apresentaram sucesso, foi criada uma tarefa na aplicação, com o único objetivo de continuar a execução da aplicação. Esta tarefa era iniciada quando a aplicação entrava em modo background e, a cada vez que o sistema operativo a terminava, era criada uma nova.

Apesar de não ter sido encontrada mais nenhuma configuração possível de ser realizada para o funcionamento da aplicação em *background*, o comportamento observado pode ter como origem uma configuração que não foi implementada. Apesar de estarem disponíveis no mercado aplicações móveis capazes de manter a sua execução em *background* por períodos de tempo indeterminados, não se conseguiu obter o comportamento desejado nesta aplicação.

5 Conclusão

Este projeto consistiu na análise e implementação do processo de registo de horas de entrada e de saída de utentes das unidades de saúde. A solução seria fundamental para automatizar este processo executado no SGTD, tornando-o assim menos sujeito a erro humano. O ponto principal a resolver recai na contabilização do serviço, pois se o registo das horas de entrada e de saída estiver incorreto também a contabilização estará. Não existindo, nos dias de hoje, um mecanismo que verifique as horas registadas e a estadia do utente na unidade de saúde, foi proposta a criação de um sistema capaz de registar as mesmas.

Para tal foi necessário estudar algumas tecnologias existentes, capazes de solucionar este problema, perceber as vantagens e desvantagens de cada uma e escolher aquela, ou aquelas, que se conseguiam adaptar melhor ao transporte do utente e à sua passagem pela unidade de saúde. Neste ponto foi claro que os BLE beacons estavam a mostrar bons resultados na literatura de referência, no que toca ao sistema de localização no interior e sistemas de deteção. Mesmo sendo tão promissores, continuam a existir duas tecnologias, mais populares e conhecidas pela população, que têm um grande impacto nos sistemas de localização; o GPS e o Wi-fi. Em conjunto, o uso destas três tecnologias é capaz de auxiliar na criação de um sistema com precisão e eficácia elevada, que deteta um utente numa unidade de saúde assim como a sua movimentação dentro da mesma.

A solução consiste assim numa aplicação móvel capaz de fazer a deteção de beacons que se encontram nas unidades de saúde, utilizando ao mesmo tempo o GPS e o Wi-fi para confirmar a presença do utilizador no local. O desenvolvimento desta aplicação mostrou ser um desafio devido a vários fatores. Inicialmente, a solução foi pensada para funcionar tanto em Android como em iOS. De forma a auxiliar o processo de desenvolvimento, existiu a necessidade de utilizar uma *framework* como o Ionic. Apesar do desenvolvimento de várias aplicações de teste, nunca foi possível obter bons resultados relativamente à deteção de beacons. Mesmo com a falta de sucesso nesta solução a mesma continuou a ser desenvolvida de forma insistente, tendo sido abandonada numa fase final, numa altura considerada tardia, face aos desenvolvimentos do projeto.

Este problema fez com que o desenvolvimento em Android nativo fosse apressado. Mesmo existindo uma melhoria substancial no comportamento da aplicação, houve uma grande perda de tempo no desenvolvimento em Ionic, não possibilitando a implementação e teste de outros mecanismos para a execução da aplicação em *background*. O método utilizado neste projeto é intrusivo em termos de experiência de utilizador e só deveria ser utilizado em último recurso.

Contudo os resultados obtidos foram promissores. A aplicação conseguiu detetar beacons de forma consistente, sendo que os registos podem descrever a movimentação do utilizador dentro de um edifício. No fim, podem ser afirmadas horas de entrada e de saída, do utilizador e do local, utilizando mecanismos de confirmação por GPS e Wi-fi para confirmar a sua presença.

Um dos grandes problemas que pode ser encontrado neste serviço é na adesão dos utentes à aplicação. Sem existir um benefício para a utilização da aplicação, dificilmente os utilizadores vão instalar e utilizar. Isto pode ser negado se o produto final for uma aplicação que é instalada uma vez e

que funciona durante um longo período, depois de o utilizador ter aberto a primeira vez. E nesta altura não é possível afirmar que a aplicação desenvolvida é capaz de funcionar dessa forma.

5.1 – Trabalho Futuro

Este projeto tem em vista a implementação do sistema nas unidades de saúde. Para tal, é necessário realizar uma avaliação ao trabalho desenvolvido e entender como este pode ser transferido para uma situação real. Para além disso, é essencial saber que componentes não foram implementadas, as razões para tal e a importância que estas podem vir a ter na execução dentro de uma unidade de saúde.

A componente de monitorização de beacon e a componente de validação de horas de entrada e saída foram inicialmente propostas para este relatório. No entanto, nenhuma destas componentes foi implementada. Isto deveu-se sobretudo ao atraso existente na implementação das deteções em *background*.

Algumas unidades de saúde têm um tamanho reduzido, o que facilita a gestão dos beacons no local, enquanto outras são compostas por várias alas. Quando maior for a área da unidade de saúde, maior será a gestão de todos os beacons que se encontram no local. Um dos problemas que pode ocorrer é a dificuldade de monitorizar os níveis de bateria de cada beacon. Para tal foi pensada uma componente que, através das deteções dos utentes, possibilitasse entender a existência de um beacon com algum problema de funcionamento. Esta análise poderia ser efetuada através de padrões encontrados nas deteções de um utilizador da seguinte forma:

- Existindo um conjunto de deteções efetuadas por utentes, encontrar os padrões dessas deteções. Estes padrões são constituídos por deteções onde os identificadores dos beacons são sempre constantes em relação à ordem pela qual são detetados.
- Ao longo da estadia do utilizador numa unidade de saúde, este vai realizando as deteções, sendo estas registadas no servidor.
- No fim, quando o utilizador sai do local, as deteções são analisadas e o sistema tenta perceber se existiu um beacon que deveria ter sido detetado pela aplicação, mas que não se encontra nos registos.
- A falta desta deteção notifica o gestor de beacons no local e este desloca-se até ao equipamento para realizar um teste e avaliar o estado do beacons.

Para o correto funcionamento desta componente existem alguns pressupostos que necessitam de ser cumpridos primeiro. Inicialmente, tem de existir um conjunto de registos criados pelos utilizadores para determinar os padrões de deteções. Por fim, tem de existir um género de *back-office* onde o gestor de beacons do local possa realizar a gestão dos equipamentos e receber as notificações do funcionamento

dos beacons. Este *back-office* pode ser implementado através de uma aplicação móvel própria para o gestor ou de uma aplicação web.

A escolha de beacons também é importante, no que diz respeito a estas atividades de gestão. Atualmente no mercado existem beacons, diferentes daqueles utilizados nesta solução, com a capacidade de se ligarem à internet. Esta ligação permite a troca de informação entre o gestor e os beacons. O gestor pode configurar os beacons remotamente sem ter de se deslocar para próximo dos mesmos, ao mesmo tempo que tem acesso a várias informações, como o estado da bateria, entre outras.

A componente de validação de horas de entrada e de saída visava a análise das deteções efetuadas pelo utilizador, com o intuito de aferir com maior certeza o momento da sua entrada e de saída. Uma vez que o servidor contém as deteções registas, possui informação que o torna capaz de analisar o percurso do utente na unidade de saúde e, no fim, decidir as horas de entrada e de saída que serão guardadas no SGTD. De forma a exemplificar a lógica de decisão do servidor, é apresentada a Tabela 1, onde é possível observar quatro registos de utilizadores e as várias deteções que podem ocorrer, onde cada deteção tem o identificador do beacon, estando este perto da entrada, saída ou no interior da unidade de saúde.

ID	Prestação	Registos	Entrada	Saída
R2	11:00	<S, 10:00>; <E, 10:01>; <I, 11:00>; <I, 11:30>; <S, 11:35>	10:01	11:35
R3	11:00	<S, 10:00>; <E, 10:30>; <I, 11:00>; <I, 11:30>; <S, 11:35>	10:30	11:35
R4	12:00	<E, 11:55>; <E, 11:57>; <I, 11:59>; <S, 13:00>	11:55	13:00
R5	13:00	<I, 13:00>; <I, 13:02>; <S, 13:30>	--	13:30

Tabela 3 – Exemplo de Deteções registadas durante a estadia de um utente numa unidade de saúde

Na tabela também são apresentadas as horas de entrada e de saída que seriam enviadas para o SGTD se o servidor não analisa-se os registos.

Analisando o registo R1 concluímos que o utente já se encontrava na unidade de saúde antes da hora de entrada. Este valor foi retirado do primeiro registo, que contém a etapa de entrada, <E,10:01>. O primeiro registo do utilizador ocorreu hás 10:00h, existindo neste caso, uma diferença de 1 minuto, comparando com a primeira deteção de entrada. Nesta situação podemos considerar a hora de entrada do utente como sendo hás 10:01h porque a diferença é mínima e pode ser desprezável. A hora de saída registada é equivalente à última deteção anotada na etapa de saída.

No registo R2 encontramos uma situação semelhante a R1. É possível entender que o primeiro registo da etapa de entrada ocorreu 30 minutos depois da primeira deteção (<S, 10:00>). Neste caso, a hora de entrada registada no SGTD não deve ser hás 10:30h, uma vez que existe uma grande diferença nos horários. Ao analisar os registos, o servidor deve escolher outra hora de entrada que se

adeque mais à presença do utente na unidade de saúde, notificando-o na aplicação móvel da inconsistência das deteções.

No registo R3 observamos que as horas de entrada e de saída estão corretas de acordo com as deteções que ocorreram. Neste exemplo pode ser realizada uma análise do percurso do utente e uma estimativa do tempo que este esteve presente na consulta. O percurso do utente é calculado através da sequência de deteções dos beacons e do local onde estes se encontram. A estimativa do tempo da consulta é dada pela deteção de um beacon intermédio (<I, 11:59>). A deteção que a sucedeu foi de um beacon de saída, uma hora depois, por isso é possível prever que este foi o tempo equivalente à duração da consulta.

Por fim, o registo R4 não apresenta nenhuma deteção de beacons de entrada. O sistema não consegue afirmar com certeza a hora de entrada do utente na unidade de saúde. O servidor, nestas situações, tem de notificar o utilizador através da aplicação que não consegue afirmar com precisão a sua hora de entrada.

A configuração dos beacons pensada para as unidades de saúde é distinta da configuração que foi implementada para o teste realizado na Link Consulting. A configuração pensada está desenvolvida em torno da definição de etapa, que surge com a necessidade de entender as ações de um utente dentro da unidade de saúde. Se existe a deteção de um beacon que está a identificar uma etapa de entrada, é de pressupor que o utente está a entrar na unidade de saúde.

Etapas	**UUID**	**Major**	**Minor**
Entrada	B9407F30-F5F8-466E-AFF9-25556B57FE6D	0 - 99	0 - 65535
Saída	B9407F30-F5F8-466E-AFF9-25556B57FE6D	100 - 199	0 - 65535
Intermédia	A5D67CE-1A2A-90BA-44CB-3567823545AB	200 - 65535	0 - 65535

Tabela 4 – Exemplo de Configuração a utilizar dentro de uma unidade de saúde

Um exemplo da configuração possível para uma unidade de saúde está apresentado na Tabela 2, sendo que as etapas estão representadas pelo campo major. Como existem 65535 valores que o campo major pode tomar, uma das soluções é atribuir 100 valores possíveis para as etapas de entrada e outros 100 para as de saídas, e os restantes para as etapas intermédias. Esta distribuição permite identificar 100 locais de entrada na unidade de saúde e 100 de saída. As etapas intermédias podem ser ainda mais especificadas, podendo então identificar diferentes alas com diferentes especialidades médicas, como fisioterapia, cardiologia, neurologia, etc. Os beacons pertencentes a cada etapa dos utentes têm de estar situados em locais apropriados aos seus objetivos.

Desta forma poderia ser desenvolvido um sistema que analisasse as deteções efetuadas pelo utente e, consoante a finalidade do transporte e motivo de deslocação, aferir um grau de certeza a cada deteção. Para tal, seria necessário conter no sistema, utentes que, tendo a mesma finalidade de

transporte, obtiveram deteção semelhantes. Assim, o sistema poderia ir aprendendo qual a movimentação habitual dentro de uma unidade de saúde e poderia aferir que deteções são coerentes nos registos dos utentes.

Uma outra configuração possível seria identificar todos os beacons que se encontram na mesma ala com o mesmo valor de major. A configuração seria mais simples pois todos os beacons colocados nessa mesma zona eram configurados da mesma forma e não seria necessário ter em atenção se o identificador utilizado era repetido. Contudo, a utilização do mesmo valor de major não permite identificar o mau funcionamento de um beacon específico, através da utilização do método de gestão referido mais acima. Sendo que todos os beacons de uma zona estariam configurados com o mesmo identificador, não seria possível perceber se um beacon específico estaria com alguma falha.

Apesar de um desenvolvimento curto, a aplicação mostrou-se capaz de obter deteções de forma consistente e precisa. No entanto, o resultado final pode ser melhorado, aumentando o grau de fiabilidade.

- **Melhoria do funcionamento em *background***: tal como foi dito, a solução para a execução em *background* é intrusiva e a aplicação deveria manter a execução durante períodos de tempo extensos. Este resultado pode ser obtido utilizando notificações *push* que no dia do transporte, despertam a aplicação. Desta forma os recursos não são desperdiçados no período que o utente não tem transporte.
- **Processo de saída da unidade de saúde**: neste momento as horas de entrada e de saída são enviadas para o SGTD através de um evento diário. Este evento pode ser substituído por um evento que é lançado, de acordo com a deslocação do utilizador e o seu afastamento em relação à unidade de saúde onde se encontra.
- **Sistema de gestão de fluxo**: com os vários registos de deteções existentes no sistema, é possível a criação de um sistema que analisa a movimentação dos utentes dentro das unidades de saúde. A movimentação está refletida nas deteções registadas pois estas contêm o identificador do beacon detetado. Ao ligar as deteções sucessivas, é possível recriar os passos do utente dentro da unidade de saúde e assim executar uma análise ao fluxo existente na mesma.

Bibliografia

[1] Link Consulting, http://www.linkconsulting.com/pt-pt/blog/whatwedo/sgtd/

[2] Bluetooth, <https://blog.bluetooth.com/bluetooth-low-energy-it-starts-withadvertising?_ga=2.35928611.249145918.1541338669-1324689682.1537743139>

[3] J. DeCuir, "Introducing Bluetooth Smart: Part 1: A look at both classic and new technologies.," in IEEE Consumer Electronics Magazine, vol. 3, no. 1, pp. 12-18, Jan. 2014, doi: 10.1109/MCE.2013.2284932.

[4] Faragher, R., Harle, R., "An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications," Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014), Tampa, Florida, September 2014, pp. 201-210.

[5] Gomez, C.; Oller, J.; Paradells, J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Sensors 2012, 12, 11734-11753.

[6] M. Kassim, H. Mazlan, N. Zaini and M. K. Salleh, "Web-based student attendance system using RFID technology," 2012 IEEE Control and System Graduate Research Colloquium, Shah Alam, Selangor, 2012, pp. 213-218, doi: 10.1109/ICSGRC.2012.6287164.

[7] Saparkhojayev, Nurbek & Guvercin, Selim. (2012). Attendance Control System based on RFID-technology. International Journal of Computer Science Issues. 9.

[8] M. V. Bueno-Delgado, P. Pavón-Marino, A. De-Gea-García and A. Dolón-García, "The Smart University Experience: An NFC-Based Ubiquitous Environment," 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Palermo, 2012, pp. 799-804, doi: 10.1109/IMIS.2012.110.

[9] El-Rabbany, A. (2002) *Introduction to GPS The Global Positioning System,* Artech House

[10] C. Lim, Y. Wan, B. Ng and C. S. See, "A Real-Time Indoor WiFi Localization System Utilizing Smart Antennas," in IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 618-622, May 2007, doi: 10.1109/TCE.2007.381737.

[11] Lun Lee, D., Chen, Q., *A Model-Based WiFi Localization Method*, Disponível em: <http://home.cse.ust.hk/~dlee/Papers/mobile/infoscale07-where-am-i.pdf> [Consultado em 01/03/19]

[12] Liu, H., Gan, Y., Yang, J., Sidhom, S., Wang, Y., Chen, Y., Ye, F. (2012) “Push the Limit of WiFi based Localization for Smartphones", *18th annual international conference on Mobile computing and networking*, Turquia, 2012

[13] Decuir, J. "Bluetooth 4.0: Low Energy"

[14] Martelli, F. "Bluetooth Low Energy"

[15] Gomez, C.; Oller, J.; Paradells, J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Sensors 2012, 12, 11734-11753.

[16] C. M. Yang et al., "Textile-based capacitive sensor for a wireless wearable breath monitoring system," 2014 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2014, pp. 232-233, doi: 10.1109/ICCE.2014.6775985.

[17] J. Decuir, "Introducing Bluetooth Smart: Part II: Applications and updates.," in IEEE Consumer Electronics Magazine, vol. 3, no. 2, pp. 25-29, April 2014, doi: 10.1109/MCE.2013.2297617.

[18] K. E. Jeon, J. She, P. Soonsawad and P. C. Ng, "BLE Beacons for Internet of Things Applications: Survey, Challenges, and Opportunities," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 811-828, April 2018, doi: 10.1109/JIOT.2017.2788449.

[19] Jergefelt, M. (2015) "An Internet of Pings: Enhancing the Web User Experience of Physically Present Patrons with Bluetooth Beacons", Vol. 1, Issue 2,

[20] Newman, N. Apple iBeacon technology briefing. J Direct Data Digit Mark Pract 15, 222–225 (2014). https://doi.org/10.1057/dddmp.2014.7

[21] Apple, Inc. "Getting Started with iBeacon"

[22] Google Developers. https://developers.google.com/beacons/eddystone

[23] Estimote, Inc. https://estimote.com/products/

[24] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison," in *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12-26, Feb. 2005, doi: 10.1109/MWC.2005.1404569.

[25] SONHO – Sistema Integrado de Informação Hospitalar. https://spms.minsaude.pt/product/sonho/

[26] MICROSOFT ANNOUNCED THE END OF SUPPORT FOR WINDOWS PHONE. LUMIA USERS ADVISED TO SWITCH. https://nokiamob.net/2019/01/21/microsoft-announced-the-end-of-support-forwindows-phone-lumia-users-advised-to-switch/

[27] Coelho, S. (2015) “Smart Places - A framework to develop proximity-based mobile applications”

[28] Android Beacon Library - https://altbeacon.github.io/android-beacon-library/index.html

[29] JSON Web Token - https://jwt.io