



Universidades Lusíada

Ferreira, Rúben Mateus Oliveira

Localização de dispositivos móveis no interior de edifícios

<http://hdl.handle.net/11067/2359>

Metadados

Data de Publicação

2013

Resumo

A localização de um dispositivo móvel ou de um outro equipamento com sensores de rádio que suportam as comunicações Wi-Fi e/ou Bluetooth, poderá ser inferida através da análise da informação transmitida pelos dispositivos de redes sem fios. No entanto este tipo de sistemas de localização baseados em redes sem fios está sujeito aos fenómenos de variações de nível de sinal, que pode afetar de forma aleatória, todo o processo de localização. Nesta dissertação é proposto um mecanismo de localização ...

The location of a mobile device or another device equipped with sensors that support radio communications Wi-Fi and / or Bluetooth, can be inferred by analyzing the information transmitted by wireless devices. However, this type of location systems based on wireless networks is subject to phenomena of signal level variations that can affect randomly throughout the localization process. In this thesis we propose a location engine for mobile devices, in which the collection is made of the signal l...

Palavras Chave

Redes móveis, Comunicações

Tipo

masterThesis

Revisão de Pares

Não

Coleções

[ULF-FET] Dissertações

Esta página foi gerada automaticamente em 2024-04-19T19:56:12Z com informação proveniente do Repositório



UNIVERSIDADE LUSÍADA DE VILA NOVA DE FAMALICÃO

**LOCALIZAÇÃO DE DISPOSITIVOS MÓVEIS NO
INTERIOR DE EDIFÍCIOS**

Rúben Mateus Oliveira Ferreira

Dissertação para obtenção do Grau de Mestre

Vila Nova de Famalicão, 2013



UNIVERSIDADE LUSÍADA DE VILA NOVA DE FAMALICÃO

**LOCALIZAÇÃO DE DISPOSITIVOS MÓVEIS NO
INTERIOR DE EDIFÍCIOS**

Rúben Mateus Oliveira Ferreira

Orientador: Prof. Doutor Rui Gabriel de Azevedo Silva

Co- Orientador: Prof. Rui Manuel Dias Ferreira Lima

Dissertação para obtenção do Grau de Mestre

Vila Nova de Famalicão, 2013

Agradecimentos

Dedico esta dissertação para obtenção do Grau de Mestre a Susana Moreira, minha namorada, que sempre me apoiou e mostrou que era possível realizar este trabalho. Agradeço à minha família, principalmente aos meus pais, pois foram eles que me deram a possibilidade do acesso ao ensino superior e que estiveram sempre ao meu lado, tanto nas adversidades como nos bons momentos.

Ao meu orientador Prof. Doutor Rui Gabriel de Azevedo Silva e ao meu co-orientador Prof. Rui Manuel Dias Ferreira Lima, agradeço todo empenho, dedicação e disponibilidade que me prestaram ao longo destes meses e que foi crucial no desenvolvimento desta dissertação.

À instituição universidade Lusíada de Vila Nova de Famalicão e a todos os seus docentes que contribuíram para a minha formação.

Índice

AGRADECIMENTOS	III
ÍNDICE	IV
ÍNDICE DE FIGURAS	V
RESUMO	VI
ABSTRACT	VII
PALAVRAS-CHAVE	VIII
LISTA DE ABREVIATURAS.....	IX
1. INTRODUÇÃO.....	10
2. ESTADO DA ARTE	12
3. SERVLOC – LOCALIZAÇÃO DE DISPOSITIVOS.....	18
3.1. FERRAMENTAS DE DESENVOLVIMENTO	18
3.2. ARQUITETURA DO SISTEMA.....	21
3.3. PESQUISA E ARMAZENAMENTO DE DADOS PROVENIENTES DOS <i>ACCESS POINTS</i>	22
3.4. SERVIÇO DE LOCALIZAÇÃO	24
3.5. ALGORITMO DE LOCALIZAÇÃO	27
4. PROCESSAMENTO DA LOCALIZAÇÃO DO DISPOSITIVO MÓVEL.....	28
5. MONITORIZAÇÃO DO SISTEMA.....	34
6. SISTEMA DE INFORMAÇÃO DE SUPORTE AO SERVIÇO DE LOCALIZAÇÃO	35
7. SEGURANÇA.....	39
8. VALIDAÇÃO	40
9. CONCLUSÕES E PERSPETIVAS DE TRABALHO FUTURO.....	44
10. REFERÊNCIAS BIBLIOGRÁFICAS	45
11. ANEXOS	46

Índice de figuras

FIG. 1 - AUTO-CORRELAÇÃO ENTRE AS POTÊNCIAS DE SINAIS VINDOS DE UM AP	15
FIG. 2 - POSIÇÃO DE COM E SEM A PLANTA	17
FIG. 3 - MICROSOFT VISUAL C# 2010 EXPRESS, DESENVOLVIMENTO DA APLICAÇÃO SERVLOC BACKOFFICE	19
FIG. 4 - ECLIPSE V3.8, DESENVOLVIMENTO DA APLICAÇÃO "SERVLOC LOCATE ME"	19
FIG. 5 - NETBEANS IDE 7.0.2, DESENVOLVIMENTO DO WEB SERVICE	20
FIG. 6 - DIAGRAMA DE RELAÇÕES	21
FIG. 7 - APLICAÇÃO DE PESQUISA E ARMAZENAMENTO DE DADOS PROVENIENTES DOS ACCESS POINTS, "SERVLOC BACKOFFICE")	23
FIG. 8 - FUNÇÕES EXISTENTES NO SERVLOC WEBSERVICE.....	25
FIG. 9 - FRAGMENTO DE CÓDIGO DO SERVLOC WEBSERVICE	26
FIG. 10 - CÓDIGO DO ALGORITMO DE LOCALIZAÇÃO	27
FIG. 11 - SISTEMA DE AUTENTICAÇÃO DA APLICAÇÃO "SERVLOC LOCATE ME"	29
FIG. 12 - INTERFACE GRÁFICA DO PROCESSO DE LOCALIZAÇÃO.....	30
FIG. 13 - VISUALIZAÇÃO DA ÚLTIMA LOCALIZAÇÃO DOS "AMIGOS"	31
FIG. 14 - GESTÃO DE "AMIGOS" DO UTILIZADOR.....	32
FIG. 15 - SISTEMA DE LOCALIZAÇÃO EM MODO TREINO	33
FIG. 16 - APRESENTAÇÃO DO GRÁFICO DE FIABILIDADE DA APLICAÇÃO "SERVLOC LOCATE ME".....	34
FIG. 17 - APRESENTAÇÃO DO GRÁFICO DE ESTATÍSTICA "NÚMERO DE PEDIDOS POR ACCESS POINTS"	34
FIG. 18 - TABELAS DA BASE DADOS	35
FIG. 19 - RESULTADOS DO PRIMEIRO TESTE DE LOCALIZAÇÃO NO EDIFÍCIO Nº1	40
FIG. 20 - RESULTADOS DO SEGUNDO TESTE DE LOCALIZAÇÃO NO EDIFÍCIO Nº1	41
FIG. 21 - RESULTADOS DO TERCEIRO TESTE DE LOCALIZAÇÃO NO EDIFÍCIO Nº1.....	41
FIG. 22 - PEDIDOS E RESULTADOS ERRADOS DE LOCALIZAÇÃO NO EDIFÍCIO Nº1	42
FIG. 23 - RESULTADOS DO PRIMEIRO TESTE DE LOCALIZAÇÃO NO EDIFÍCIO Nº2	42
FIG. 24 - RESULTADOS DO SEGUNDO TESTE DE LOCALIZAÇÃO NO EDIFÍCIO Nº2	43
FIG. 25 - PEDIDOS E RESULTADOS ERRADOS DE LOCALIZAÇÃO NO EDIFÍCIO Nº2	43

Resumo

A localização de um dispositivo móvel ou de um outro equipamento com sensores de rádio que suportam as comunicações Wi-Fi e/ou Bluetooth, poderá ser inferida através da análise da informação transmitida pelos dispositivos de redes sem fios. No entanto este tipo de sistemas de localização baseados em redes sem fios está sujeito aos fenómenos de variações de nível de sinal, que pode afetar de forma aleatória, todo o processo de localização. Nesta dissertação é proposto um mecanismo de localização para dispositivos móveis, no qual é efetuado a recolha do nível de sinal de uma rede Wi-Fi e a deteção de outros dispositivos com suporte Bluetooth. É proposta uma solução para garantir a privacidade do utilizador, utilizando um algoritmo de encriptação para os dados que percorrem a rede durante o processo de localização. O grau de fiabilidade do sistema de localização foi verificado com a realização de testes em ambiente real, o que permitiu assegurar que este sistema é uma solução eficaz para a localização de um dispositivo. Este sistema possui a inovação de prever a localização de um dispositivo no interior de um edifício através da tecnologia Wi-Fi e também facilita o encontro de utilizadores que estejam próximos através da tecnologia Bluetooth.

Abstract

The location of a mobile device or another device equipped with sensors that support radio communications Wi-Fi and / or Bluetooth, can be inferred by analyzing the information transmitted by wireless devices. However, this type of location systems based on wireless networks is subject to phenomena of signal level variations that can affect randomly throughout the localization process. In this thesis we propose a location engine for mobile devices, in which the collection is made of the signal level of a Wi-Fi network and the detection of other devices supporting Bluetooth. It proposed a solution to ensure the privacy of the user, using an encryption algorithm to the data that traverses the network during the localization process. The degree of reliability of the location was verified with tests in a real environment, which has ensured that this system is an effective solution for the location of a device. This system has the innovation to predict the location of a device inside a building using Wi-Fi and also facilitates the gathering of users who are in the same building via Bluetooth.

Palavras-Chave

Android

Bluetooth

Encriptação

Localização

Privacidade

Windows

Wireless

Lista de Abreviaturas

AP	Access Point
GPS	Global Positioning System
GSM	Global System for Mobile Communications
MD5	Message-Digest algorithm 5
MAC	Media Access Control
PHP	Hypertext Preprocessor
RSS	Received Signal Strength
Wi-Fi	Wireless Fidelity

1. Introdução

Com a tecnologia Wi-Fi a tornar-se cada vez mais disponível em dispositivos móveis, esta apresenta um potencial cada vez maior para a sua utilização como uma tecnologia para a determinação de localização de objetos, por exemplo *smartphones*, PDA, PC. A localização no interior de edifícios de um dispositivo móvel através de uma rede Wi-Fi constituiu o principal objetivo para o trabalho apresentado nesta dissertação. Com este trabalho pretende-se ainda dar um contributo para permitir o anonimato dos utilizadores que usufruam deste sistema de localização, isto é, o processo de localização do dispositivo não receberá qualquer tipo de informação de dados que identifique indiretamente o dono do dispositivo. Toda a informação que comprometa a privacidade do dispositivo, que esteja a circular na rede, será completamente encriptada através do algoritmo de encriptação MD5.

Sendo o GPS um dos sistemas de posicionamento mais conhecidos, mas pelo facto do GPS não permitir a sua utilização no interior de edifícios, dada que este não consegue obter informações dos satélites (Bejuri, Mohamad, & Sapri, 2011), este sistema não pode ser utilizado nestas condições.

O conceito de localização de um dispositivo consiste na identificação do mesmo por via de um local físico, isto é, a localização do dispositivo será efetuada através da posição do equipamento num local físico, por exemplo o dispositivo **A** encontra-se na sala X. Para realizar o processo de localização podemos tirar partido de existência de uma infraestrutura de comunicação de redes sem fios. Um exemplo é a tecnologia Wi-Fi com os seus *access points* (APs), e o dispositivo terá também de possuir suporte à mesma tecnologia.

A utilização da tecnologia Wi-Fi em equipamentos móveis está cada vez mais difundida, tal como, locais com *access points* com esta tecnologia (A Europa gosta do Wi-Fi: novo estudo recomenda a disponibilização de mais espetro, 2013). Contudo, a localização de dispositivo no interior de um edifício apresenta um enorme problema devido à variação da potência do sinal (RSS) recebido, isto pelo facto de estarmos perante de alterações de objetos presente no local, de condições climatéricas, entre outras, o que leva ao aumento da probabilidade de erro na localização do dispositivo (Tse, 2005).

Os módulos de comunicação (Wi-Fi e Bluetooth) presentes num dispositivo móvel possuem *mac address* que identifica de forma clara todos os dispositivos. Por isso a utilização destes módulos para a localização do dispositivo colocam em causa a privacidade do seu dono (utilizador). A solução para minimizar este problema passa pela encriptação do *mac address* presente nos dispositivos, este processo é feito quando esta informação passa pela rede ao qual o dispositivo está ligado.

Neste sistema de localização usa-se a tecnologia *Bluetooth* para acrescentar informação acerca da localização de dispositivos vizinhos, sem interferir diretamente na localização do dispositivo, isto com a total privacidade dos mesmos, ou seja, os dispositivos vizinhos só serão localizados caso estes o permitem.

A arquitetura deste sistema de localização é constituída por duas aplicações e um serviço web. Uma das aplicações destina-se à gestão e armazenamento dos *access points* e dos locais envolvidos, ou seja, faz o rastreio dos sinais de rádio dos *access points* num local específico. A aplicação móvel tem como objetivo de capturar os sinais dos APs, enviando esta informação para o servidor web e por último apresentar o local em que o dispositivo se encontra. O algoritmo do sistema de localização está hospedado num servidor, acedido através de um *webservice*, que recebe e processa os sinais e os respetivos *mac address* dos APs. Toda esta informação é enviada pelo dispositivo móvel e fornece ao mesmo a informação do local onde se encontra.

A principal vantagem da utilização do processo de localização num *webservice* consiste na diminuição da carga de processamento e memória sobre o dispositivo móvel, pois estes geralmente possuem recursos limitados.

2. Estado da Arte

Este trabalho iniciou-se com uma pesquisa bibliográfica sobre publicações científicas que se debruçam sobre o problema de localização

No contexto de sistemas de localização, existem sistemas análogos com o aqui proposto, utilizando também infraestruturas de *access points* existentes num determinado local, para encontrar a localização de dispositivos móveis. Mas o sistema de localização aqui proposto apresenta um sistema de localização mais amplo, pelo facto de efetuar também a localização de dispositivos vizinhos ao dispositivo do utilizador, por exemplo, quando é efetuado um pedido de localização, o sistema de localização aqui apresentado efetua também a localização de dispositivos emparelhados com o dispositivo do utilizador.

Métodos que podem ser utilizados na localização de dispositivos no interior de edifícios através de uma rede Wi-Fi são o **Received Signal Strength Indicator (RSSI)**, o **Time of Arrival (ToA)**, o **Time Difference of Arrival (TDoA)** e o **Angle of Arrival (AoA)**.

O **RSSI** é utilizado o nível de potência de sinal de um dispositivo móvel em relação a um AP como informação de entrada para realizar o processo de localização. O cálculo para três RSSI de APs diferentes, é feito através da relação de três níveis de potências de sinal, obtendo-se assim um ponto estimado de localização (Carvalho, 2007). O **ToA** é idêntico ao anterior, no entanto, no ToA a distância é calculada através do tempo em que o sinal demora a percorrer entre o emissor e o recetor. Uma das desvantagens que este método apresenta é o que os equipamentos de rede utilizados têm de estar altamente sincronizados, caso contrário leva a erros de grandes proporções (Carvalho, 2007). O **TDoA** consiste numa adaptação do ToA, mas com a diferença de contabilizar o tempo que o sinal demora a percorrer entre o dispositivo e pelo menos três APs. Este método torna-se mais fiável do que o anterior, pois ao considerar também a diferença entre os tempos de chegada, o erro provocado será menor (Carvalho, 2007). No **AoA** a informação crucial para a localização do dispositivo móvel é o valor do ângulo de incidência em que o sinal é recebido. Este processo é realizado com sinais Wi-Fi de pelo menos de três APs diferentes, pois assim é possível efetuar o cruzamento dos três ângulos obtidos e deduzir qual a localização do dispositivo móvel (Carvalho, 2007).

Em relação a algoritmos de localização presentes no mercado, existem vários, nomeadamente, o projeto Cisco Wireless Location Services (Carvalho, 2007), o iFIND (Huang, Proulx, & Ratti, 2004), projeto desenvolvido na Universidade Rey Juan Carlos (Serrano, Cañas, Matellán, & Rodero, 2004), o Horus (Youssef & Agrawala, 2005) e o projeto desenvolvido na Universidade de Freiburg (Zhou, 2006).

O projeto *Cisco Wireless Location Services* (CWLS), desenvolvido em 2006 pela empresa Cisco (Carvalho, 2007), consiste num sistema de localização que funciona sem necessidade de qualquer instalação de software adicional nos dispositivos a localizar, mas necessita de hardware específico para o seu funcionamento. Possui ainda outros requisitos, tais como, os APs têm de ser Cisco e têm que estar preparados para a configuração *Cisco Wireless Control System* (WCS). Com esta configuração, este sistema de configuração têm possibilidade de utilizar os quatro métodos acima apresentados (RSSI, ToA, TDoA, AoA). Este sistema de localização efetua uma triangulação simples, isto é, não verifica certos fenómenos, como o de *multipath*¹ e a variação da atenuação do sinal tendo em conta a construção do edifício.

O *iFind* é uma aplicação desenvolvida pelo SENSEable City Laborator (Huang, Proulx, & Ratti, 2004) com implementação no campus do Massachusetts Institute of Technology (MIT). Esta aplicação permite que docentes e alunos localizem colegas de trabalho dentro do instituto. O processo de localização do *iFind* utiliza o *Place Lab*, que consiste numa applet em *Java*TM disponível em código aberto. O *Place Lab* é uma *framework* desenvolvida com a intenção de facilitar o desenvolvimento de aplicações e serviços de localização de variados dispositivos móveis, pois para além de funcionar com a tecnologia Wi-Fi, também utiliza a tecnologia Bluetooth, GPS e GSM. Esta *framework* possui vários algoritmos adaptados para GSM, Bluetooth e Wi-Fi, sendo que na tecnologia Wi-Fi o processo de localização é baseado em diagramas de Venn, sendo computacionalmente leve, pelo facto de ter uma baixa complexidade dos cálculos envolvidos. Por isso, para determinar em que área se encontra o dispositivo móvel, basta ter informação acerca da área de cobertura máxima de cada “AP” e a informação das coordenadas espaciais de cada AP em detetado pelo dispositivo móvel. A eficiência deste processo localização é melhorada com o aumento da densidade de AP’s.

¹ Fenómeno inerente à propagação das ondas Wi-Fi e que se caracteriza pelo facto das ondas Wi-Fi não percorrerem sempre o mesmo caminho numa comunicação entre dois dispositivos Wi-Fi.

O projeto *Robot localization using WiFi signal without intensity map*, desenvolvido na Universidade Rey Juan Carlos (Serrano, Cañas, Matellán, & Rodero, 2004), Espanha, foi concluído em 2004 e consiste num sistema de localização de dispositivos móveis em redes Wi-Fi no qual não necessita de recolher previamente o nível de sinal para todos os locais. Este sistema de localização é constituído por um modelo teórico de propagação das ondas Wi-Fi, capaz de localizar dispositivos móveis com uma precisão mais baixa que os sistemas com padrões de nível de sinal, mas com precisão semelhante às técnicas que utilizam mapa de nível de sinal rádio. Ainda neste sistema de localização usa uma estratégia de aproximação probabilística para determinar a posição do dispositivo. Este mecanismo constrói e atualiza uma grelha de probabilidade de encontrar o dispositivo em cada local. A seleção da posição é feita através do maior valor da probabilidade acumulada, isto de acordo com a regra de Bayes, que consiste numa regra matemática (Eq. 1). Este modelo declara como os dados atuais devem ser afetados pelos anteriores, sendo que neste caso, uma combinação dos novos dados sobre a nova localização do dispositivo móvel com os valores anteriores de localização do dispositivo.

$$P(R = r|e) = \frac{P(e|R=r) \times P(R=r)}{P(e)} \quad (\text{Eq. 1})$$

Sendo que $P(e)$ é a probabilidade do acontecimento anterior ocorrer e $P(R=r)$ é a probabilidade do acontecimento posterior acontecer (Eq.1), portanto $P(R=r|e)$ é a probabilidade que a variável aleatória R tenha o valor r tendo sido verificado o valor de e . O denominador é só uma constante de normalização que assegura que a probabilidade condicionada de $P(R=r|e)$ assume no máximo o valor 1. Assim sendo, a determinação de um nova posição do dispositivo móvel seria ajustada pela sua posição anterior. Neste sistema de localização é utilizado o modelo *breakpoint*, que consiste num modelo de propagação Wi-Fi que verifica apenas a distância percorrida pelas ondas Wi-Fi, não contabilizando assim o efeito que certos obstáculos têm na propagação dos sinais, como por exemplo: as paredes, mobiliário, entre outros. A utilização deste tipo de modelos permite que seja feita uma estimativa da potência de receção dos sinais Wi-Fi vindos de todos os APs para todo o espaço de localização.

O projeto Horus (Youssef & Agrawala, 2005) é um sistema de localização desenvolvido em 2005 na Universidade de Maryland, e tem o objetivo de obter uma relação perfeita entre a alta precisão de localização e os baixos requisitos computacionais. Através das técnicas de “location-clustering”, o algoritmo utilizado neste projeto torna-se

bastante leve, o que permite a localização de um grande número de dispositivos móveis em simultâneo. A alta precisão a que consegue chegar, deve-se em grande parte à identificação que o Horus faz às diferentes causas da variação dos sinais Wi-Fi e como ele as consegue resolver. O resultado principal é uma aplicação que é instalada nos dispositivos móveis dos utilizadores, onde se mede a potência de sinal recebido vindo de todos os APs que sejam detetáveis. O modo de funcionamento do sistema de localização do Horus possui duas fases, sendo que na primeira fase terá que passar por um processo de treino / calibração de modo a possibilitar a construção de um mapa rádio. A segunda fase usa os mapas para o processo de localização. De acordo com os autores, este sistema de localização usa técnicas probabilísticas e a sua implementação foi testada tanto no sistema operativo Windows como também em Unix. Os autores deste sistema de localização procuraram que mesmo sofrendo variações de potência ao longo do tempo, a auto-correlação das amostras recolhidas, é de 90% para um curto espaço de tempo (Fig. 1). Este sistema usa um modelo auto-regressivo para calcular a auto-correlação (Eq. 2) dos valores.

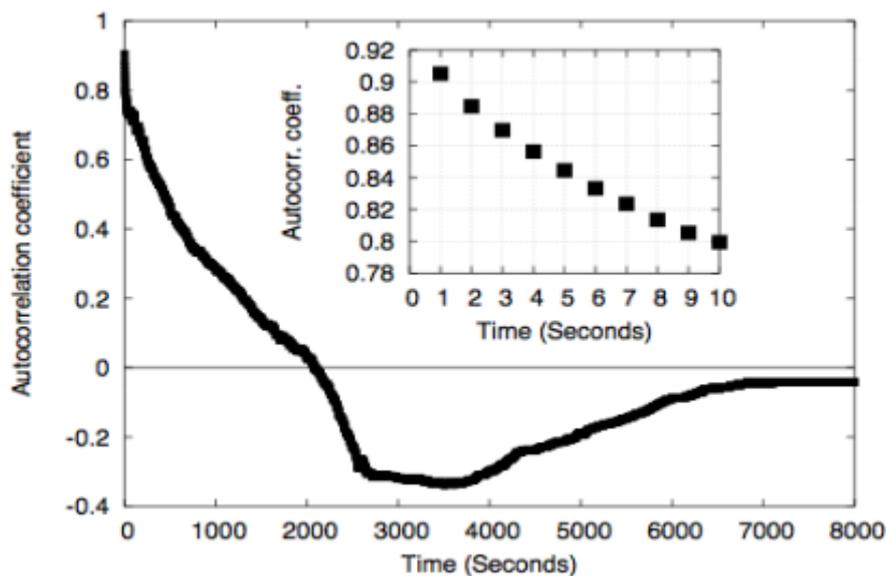


Fig. 1 - Auto-correlação entre as potências de sinais vindos de um AP

$$st = \alpha st - 1 + (1-\alpha)vt; \quad 0 \leq \alpha \leq 1 \quad (\text{Eq. 2})$$

Na equação (Eq. 2), a variável vt representa a variação do ruído e a st é uma série que representa as amostras de um *access point* (AP). Na fase de treino o valor de α é estimado e guardado como o parâmetro de distribuição para que durante a fase de localização, a distribuição Gaussiana seja adaptada com o seu valor. Com base neste

modelo, foi proposto um novo método (Eq. 3) para incluir o efeito da variância dos sinais Wi-Fi.

$$\chi = \frac{\sum_{i=1}^N p(i) \times i}{\sum_{i=1}^N p(i)} \quad (\text{Eq. 3})$$

Este sistema de localização utiliza um compensador (Eq.3) para a correção de variações, isto porque a potência dos sinais rádio recebidos pelo dispositivo não é contante para uma determinada posição. O funcionamento do compensador serve para detetar as variações ao estimar a distância entre duas localizações consecutivas.

Os autores para minimizar os requisitos computacionais utilizaram a técnica *Incremental Triangulation Clustering* (ITC), que consiste em dividir em diversas zonas toda a área do ambiente no qual o dispositivo a localizar se encontra, ou seja, cada zona será composta por um conjunto de APs. Assim sendo, são analisados apenas os sinais vindos dos APs estritamente necessários à localização não sendo necessário analisar todos os sinais Wi-Fi capturados pelo dispositivo móvel. Depois da análise dos sinais capturados é escolhido o AP com a maior potência de sinal recebida no dispositivo cliente e apenas os locais cobertos por este AP são pesquisados, escolhendo-se de seguida um segundo AP baseando-se no mesmo critério e apenas serão pesquisadas as áreas cobertas em comum pelo primeiro e segundo AP. O projeto Horus consegue localizar um dispositivo móvel numa rede Wi-Fi com um erro médio de 0,6 metros em 60% dos casos. A diminuição do erro médio é conseguido já que o sistema compara a probabilidade das estimativas das localizações caso a primeira estimativa tenha a maior probabilidade do que a segunda, a localização pára e é devolvida à primeira. Este mecanismo é processo recursivo, podendo verificar-se várias estimativas até se chegar ao resultado pretendido. De acordo com estas técnicas probabilísticas

O projeto desenvolvido na Universidade de Freiburg (Zhou, 2006), na Alemanha usa um sistema de localização de dispositivos móveis em redes Wi-Fi. Este projeto possui uma fase de treino, para obter o mapa de intensidade rádio, tal como os projetos mencionados anteriormente, mas com algumas diferenças. A implementação do sistema de localização num servidor e a previsão da localização do utilizador nos instantes seguintes. A previsão da localização é feita tendo em conta a potência dos sinais Wi-Fi dos APs que rodeiam o dispositivo, bem como acedendo ao histórico recente da posição do utilizador

para saber se o dispositivo tem estado parado ou se encontra em movimento. Este método permite corrigir os erros em relação a localizações realizadas através de potências de sinal muito semelhantes, mas que são imprescindíveis distinguir, como é o caso de estar do lado de dentro de uma sala ou do lado de fora junto à parede da mesma, ou até saber distinguir qual o andar em que o dispositivo se encontra. Para distinguir estes casos, os autores deste projeto incluíram a planta do edifício para saber quais as limitações nas trajetórias seguidas pelos dispositivos móveis. Como este mecanismo é baseado num método determinístico, implica fazer um estudo da planta do edifício e aplicação da mesma ao sistema de localização de uma forma manual, a qual se pode considerar como um inconveniente. Os autores deste projeto apresentaram uma possível solução para esta técnica que seria por encontrar uma forma do sistema de localização de analisar de uma forma automática a estrutura do edifício através de uma planta digitalizada. Os resultados obtidos provam que a utilização de um histórico recente do utilizador e a inclusão das plantas do edifício provoca (Fig. 2) uma melhoria da localização dos dispositivos móveis em movimento, reduzindo o erro médio existente em 8,4%, passando assim de 2,27 metros para 2,08 metros.

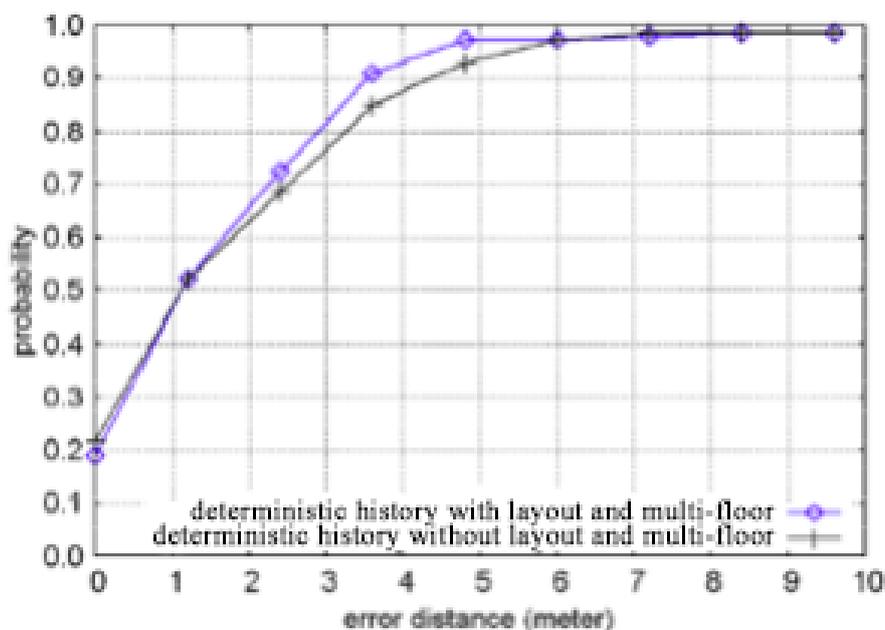


Fig. 2 - Posição de com e sem a planta

Desta maneira é possível criar algumas regras, que ajudam a minimizar os erros de localização do dispositivo móvel. Assim sendo uma das regras que este sistema de localização implementa é impor restrições na mudança de posição do utilizador conforme a velocidade que o mesmo consegue movimentar-se, isto é, o utilizador não pode mudar de

piso sem que este esteja numas escadas ou num elevador e também não pode atravessar paredes sem a existência de uma porta ou abertura.

As técnicas incorporadas neste projeto conseguem obter mais precisão, conseguindo um erro médio de 2,08 metros. Este sistema destaca-se dos outros, pelo facto de conseguir reduzir grande parte das falhas, que ocorrem quando os dispositivos parecessem estar a mover-se entre posições relativamente distantes e impossíveis de transpôr num curto espaço de tempo.

3. SERVLOC – Localização de dispositivos

Este capítulo tem o objetivo de descrever todo o processo de localização, desde da definição da localização dos locais físicos até a localização dos dispositivos móveis. O SERVLOC consiste num sistema para a localização de dispositivos móveis, com apoio de um *webservice*. A utilização do *webservice* tem o objetivo de diminuir a carga de processamento sobre o dispositivo móvel.

3.1.Ferramentas de desenvolvimento

Durante a realização deste trabalho foi necessário estudar algumas ferramentas de desenvolvimento, recorrendo a linguagens de programação distintas. Numa primeira fase do trabalho foi utilizada a linguagem de programação *c#* através do ambiente de desenvolvimento Microsoft Visual C# 2010 Express (Fig.3). Foi escolhida esta versão, pois revelou-se com maior compatibilidade com os sistemas utilizados nos dias de hoje. Com o decorrer do desenvolvimento, foram acrescentadas funcionalidades à aplicação através de utilização de bibliotecas para ajudar nas tarefas fundamentais. Foi adicionada a biblioteca “ManagedWifi.dll”, através do Visual C# 2010 Express, como objetivo de obter os dados (nível de sinal e *mac address*) de um *access point*. A biblioteca “mysql – conector – net-6.2.3” foi adicionada para efetuar a comunicação entre a aplicação em desenvolvimento (“Servloc BackOffice”) com uma base dados MYSQL.

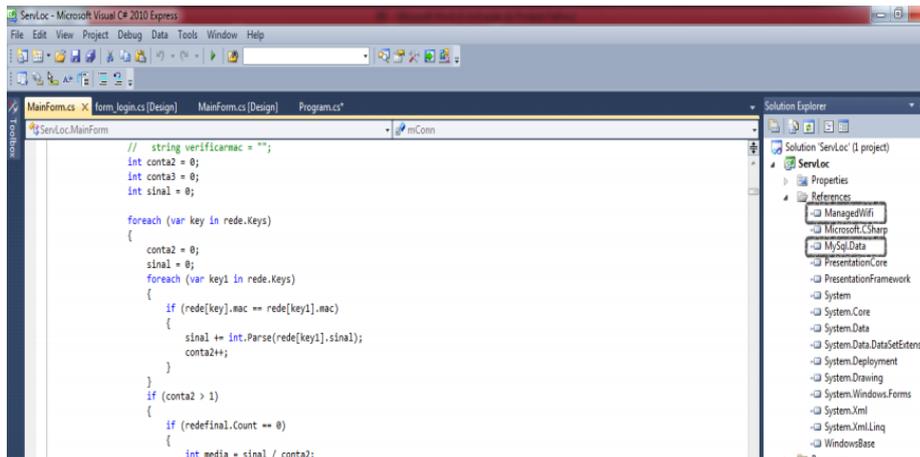


Fig. 3 - Microsoft Visual C# 2010 Express, desenvolvimento da aplicação ServLoc Backoffice

No desenvolvimento da aplicação para o dispositivo móvel, foi utilizada a linguagem de programação JAVA através do ambiente de desenvolvimento Eclipse v3.8 (IDE). Para a utilização deste ambiente do desenvolvimento da aplicação para o sistema operativo Android, sendo esta a aplicação que efetua a localização do dispositivo, foi necessário a instalação de um *plugin* que se intitula por Android Development Tools (ADT) v21.1.0-569685. Esta instalação é feita com o seguimento das etapas presente numa página web² destinada a programadores Android. Tal como na primeira fase do desenvolvimento foram utilizadas bibliotecas, sendo que neste caso consiste numa única biblioteca com o nome de “ksoap2-android-assembly-3.0.0-RC.4-jar-with-dependencies.jar” e tem por objetivo fornecer as primitivas para efetuar a comunicação do dispositivo móvel com o *webservice*.

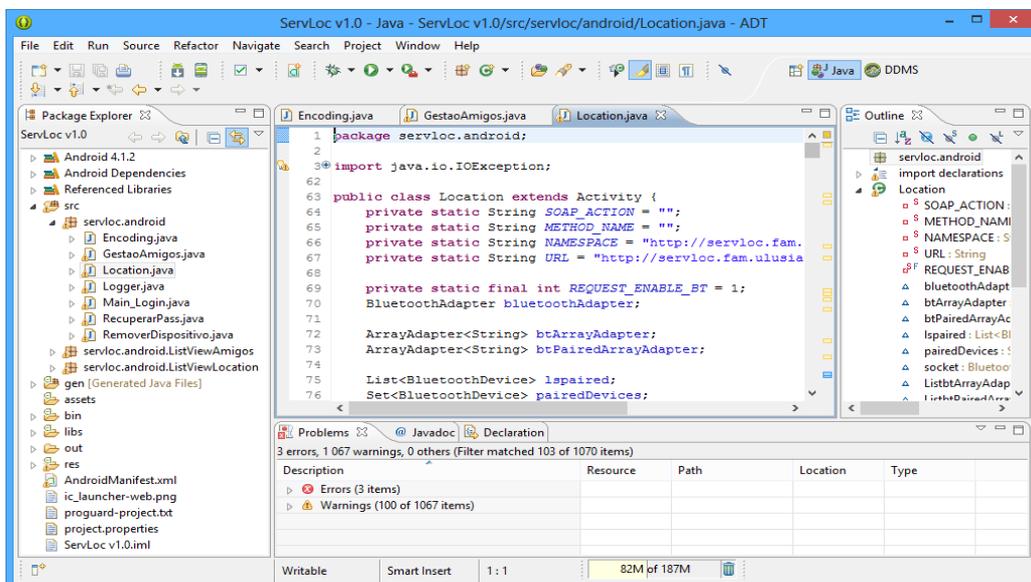


Fig. 4 - Eclipse v3.8, desenvolvimento da aplicação "Servloc Locate Me"

² <http://developer.android.com/sdk/>

Na Fig. 4, podemos visualizar o ambiente de desenvolvimento Eclipse para aplicação de localização, “Servloc Locate Me”, na qual apresenta alguns dos ficheiros construídos, nomeadamente, “Encoding.java” – tem a funcionalidade de encriptar os dados (*mac address*) que identifiquem o dispositivo móvel, “GestaoAmigos.java” – está responsável pela gestão (inserção e remoção) de dispositivos emparelhados, “Location.java” – é onde se realiza a localização do dispositivo, “Main_Login.java” – contem todo o processo de autenticação na aplicação de localização, “RecuperarPass.java” – tem a funcionalidade de recuperação da palavra-passe do utilizador para entrar na aplicação e “RemoverDispositivo.java” – neste ficheiro é efetuado a remoção do dispositivo da aplicação “Servloc Locate Me”.

Por último, na fase do desenvolvimento do *webservice* foi utilizada a linguagem de programação PHP, sendo esta a linguagem de código aberto com maior aceitação para servidores web, visto que se trata de um servidor LINUX. Para utilização desta linguagem foi utilizado o *software* NetBeans IDE 7.0.1, pois possui um conjunto de funcionalidades que ajudam durante o desenvolvimento, tais como, apresentação do código em diversas cores, a funcionalidade *intelli-sense* que ajuda o programador a visualizar as variáveis existentes no código.

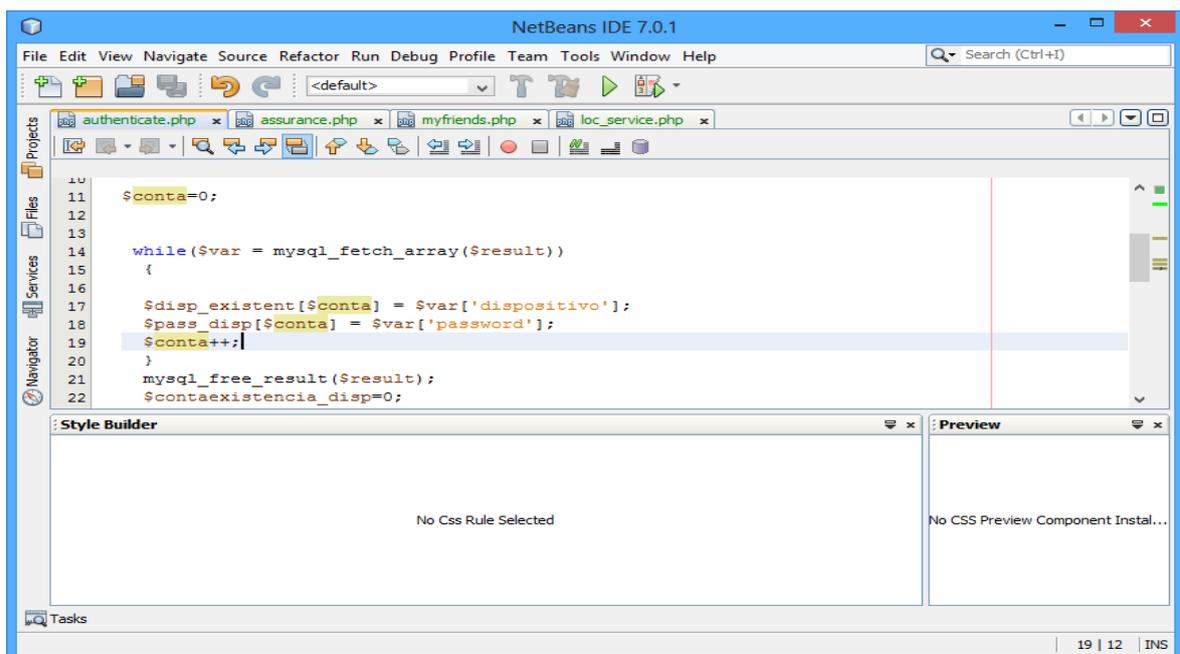


Fig. 5 - NetBeans IDE 7.0.2, desenvolvimento do web service

A Fig. 5 apresenta quatro ficheiros que são utilizados pelo *webservice*, correspondente a quatro funcionalidades distintas. O ficheiro “authenticate.php” faz a

verificação das credenciais inseridas na aplicação “Servloc Locate Me” pelo utilizador, o “assurance.php” manipula os dados (*mac address* dos *access points*, nível de sinal e o *mac address* encriptado do dispositivo), enviados pela aplicação de localização de forma no futuro obter uma estatística, o “myfriends.php” tem como funcionalidade de fazer a gestão dos amigo do utilizador da aplicação “Servloc Locate Me” e o “loc_service.php” é o ficheiro que na qual é registado todos os métodos utilizados pelo *webservice*.

3.2.Arquitetura do sistema

A fig. 6 contextualiza o tema do trabalho, apresentando a arquitetura do sistema usando um diagrama de blocos simplificado, de forma a apresentar as várias tecnologias utilizadas, interação entre aplicações e o funcionamento das mesmas.

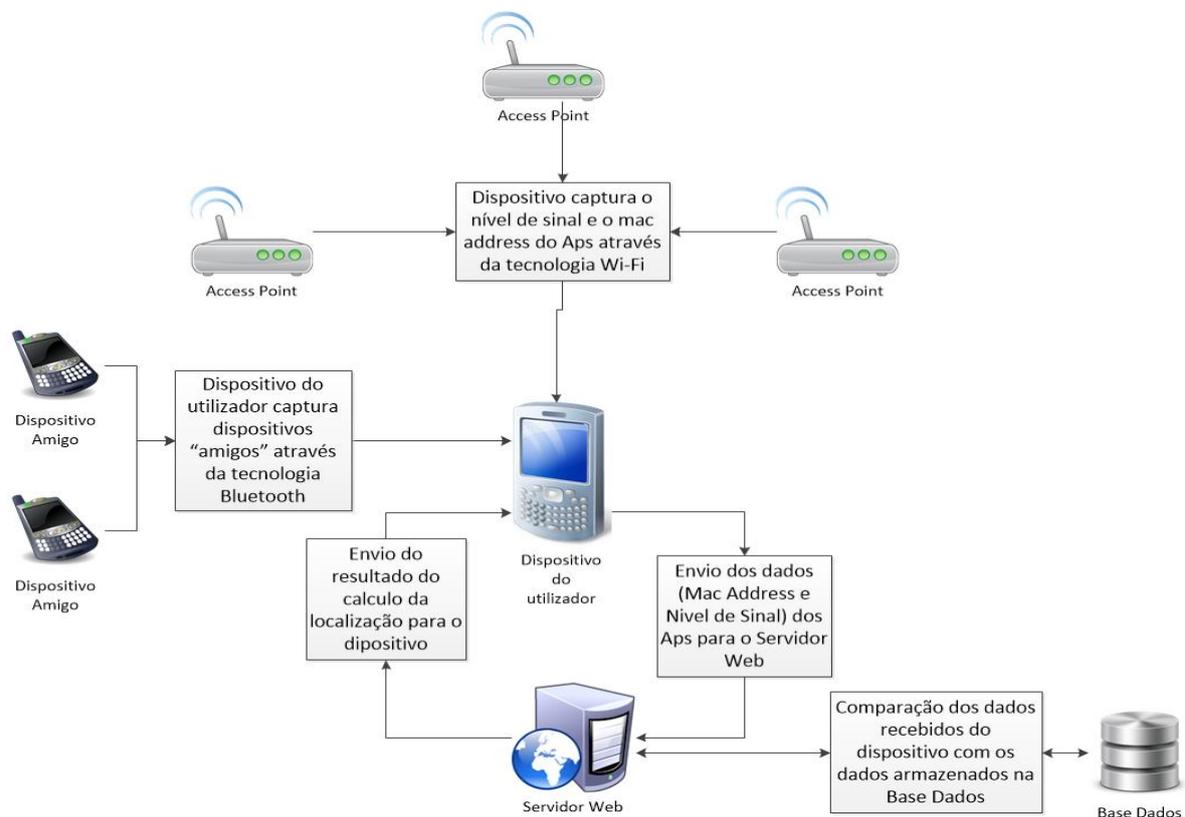


Fig. 6 - Diagrama de Relações

No diagrama apresentado na fig. 6, apresenta-se a arquitetura do sistema de localização. Este sistema caracteriza-se pela captação do nível de sinal e do *mac address* dos APs por parte de dispositivos móveis e também a deteção de dispositivos emparelhados com o mesmo. Após esta captura, os dados são encaminhados para o servidor, que usado em *webservice* interpreta os dados para obter a localização. Esta análise engloba a passagem dos dados recebidos por um algoritmo de localização que

efetua um processo de comparação de dados para inferir a localização do dispositivo. O resultado com a localização é enviado para o dispositivo móvel.

Todos os dispositivos que estejam emparelhados com o dispositivo do utilizador através da tecnologia *Bluetooth*, são considerados como dispositivos “amigos”, desta forma garante-se que o utilizador deu permissão aos dispositivos em questão para se poderem comunicar entre ambos. Neste trabalho, a utilização da funcionalidade de emparelhamento de dispositivos, permite localizar dispositivos que estejam na proximidade sem comprometer a privacidade dos mesmos.

Os dados recolhidos dos dispositivos “amigos” servem para saber a localização dos mesmos, isto é, o dispositivo do utilizador ao encontrar um dispositivo “amigo” durante o processo de localização diz-nos que a probabilidade do dispositivo “amigo” estar no mesmo local que o dispositivo do utilizador é elevada. Desta forma a localização de dispositivos que tem apenas a tecnologia *Bluetooth* é conseguida usando uma estratégia oportunista (Nguyen & Giordano, 2009) sobre dispositivos com tecnologia Wi-Fi.

3.3. Pesquisa e armazenamento de dados provenientes dos *Access Points*

Para armazenar de forma automática os dados dos *access points* de um certo espaço físico foi desenvolvido uma aplicação com o intuito de recolher: Nome do local, *mac address* e nível de sinal do *access point*. Toda esta informação serve para mapear a localização da infra-estrutura de comunicações.

Aplicação, designada por “ServLoc Backoffice”, funciona como um sensor para redes Wi-Fi e possui as seguintes funcionalidades:

- Multilinguagem;
- Utilização de credenciais para acesso;
- Filtragem da pesquisa por *mac address*;
- Adicionar e remover locais e *mac address* da base de dados;
- Verificação de *mac address* e locais já adicionados;
- A pesquisa dos *access points* e a respetiva associação a um espaço físico.



Fig. 7 - Aplicação de pesquisa e armazenamento de dados provenientes dos access points, "Servloc BackOffice")

O "Servloc Backoffice" (Fig. 7) permite escolher a visualização em duas línguas, português e inglês.

A utilização de credenciais nesta aplicação, é fundamental para tornar mais seguro o desvio dos dados armazenados na base dados com a localização dos *access points*, locais e das localizações, pois sem esta funcionalidade qualquer utilizador poderia alterar os dados em questão, o que levaria posteriormente a uma previsão errada no sistema de localização.

No "Servloc BackOffice" foi implementado um sistema de filtragem por *mac address* para ajudar o utilizador a encontrar o *mac address* que realmente pretende adicionar à base dados, por exemplo, o utilizador pode filtrar todos os *access points* que o respetivo *mac address* comece por FF-00, e de seguida adicionar à base dados.

O utilizador com o "ServLoc Backoffice" terá a facilidade de adicionar, remover e atualizar locais, bem como a facilidade de adicionar e remover os *mac address* encontrados pelo computador, evitando a inserção de dados diretamente na base dados. Para além disso, esta aplicação permite visualizar os nomes dos locais e os *mac address* de *access points* já adicionados pelo utilizador.

Por último, a aplicação usa um mecanismo para deteção dos *access points* e faz ainda a respetiva associação a um espaço físico. Esta será a funcionalidade mais importante da aplicação, pelo facto de interferir diretamente com o sistema de localização do dispositivo móvel. Esta funcionalidade consiste na associação do *mac address* e do nível de sinal dos *access points* encontrados a um local físico já adicionado na base dados pelo utilizador. Para tornar os dados recebidos (*mac address* e o nível de sinal) mais consistentes, a aplicação “Servloc BackOffice” possui a funcionalidade “Pesquisa Profunda”, sendo que esta faz um ciclo de dez pesquisas de *access points* e de seguida calcula a média do nível de sinal recebido, pois assim é absorvida as variações do nível de sinal.

3.4. Serviço de localização

Um *webservice* consiste num serviço web que permite às aplicações enviar e receber dados em formato XML. A utilização de um *webservice* torna-se numa vantagem, pelo facto de cada aplicação ser independente da linguagem de programação, o que é importante, em que foi desenvolvida uma aplicação para **Android** a comunicar com *webservice* para efetuar a localização do dispositivo móvel, no futuro podem ser desenvolvidas novas aplicações para diferentes sistemas operativos mantendo o algoritmo de localização.

Outra vantagem na utilização de um *webservice* é o reduzido consumo de processamento no dispositivo móvel, pelo facto do algoritmo do sistema de localização estar localizado no servidor remoto. A obrigatoriedade do acesso à Internet e o atraso no tempo de resposta (cerca de 5 segundos) a realizar as operações poderão ser consideradas como desvantagens.

Servloc Webservice

The screenshot displays a web service interface. On the left, a list of operations is shown with the 'scan' operation selected. On the right, the details for the 'scan' operation are displayed, including its name, binding, endpoint, SOAP action, style, and input/output parameters.

View the [WSDL](#) for the service. Click on an operation name to view its details.

- [scan](#)
- [bluescan](#)
- [blueread](#)
- [addbluefriends](#)
- [addbluefriendsauto](#)
- [senduser](#)
- [removeuser](#)
- [authentication](#)
- [sendmail](#)
- [lastaccessblue](#)
- [lastaccessbluefriends](#)
- [feedback](#)
- [orderlocal](#)
- [devicebyaps](#)
- [getlocates](#)
- [assurance](#)
- [updatedevicefriend](#)

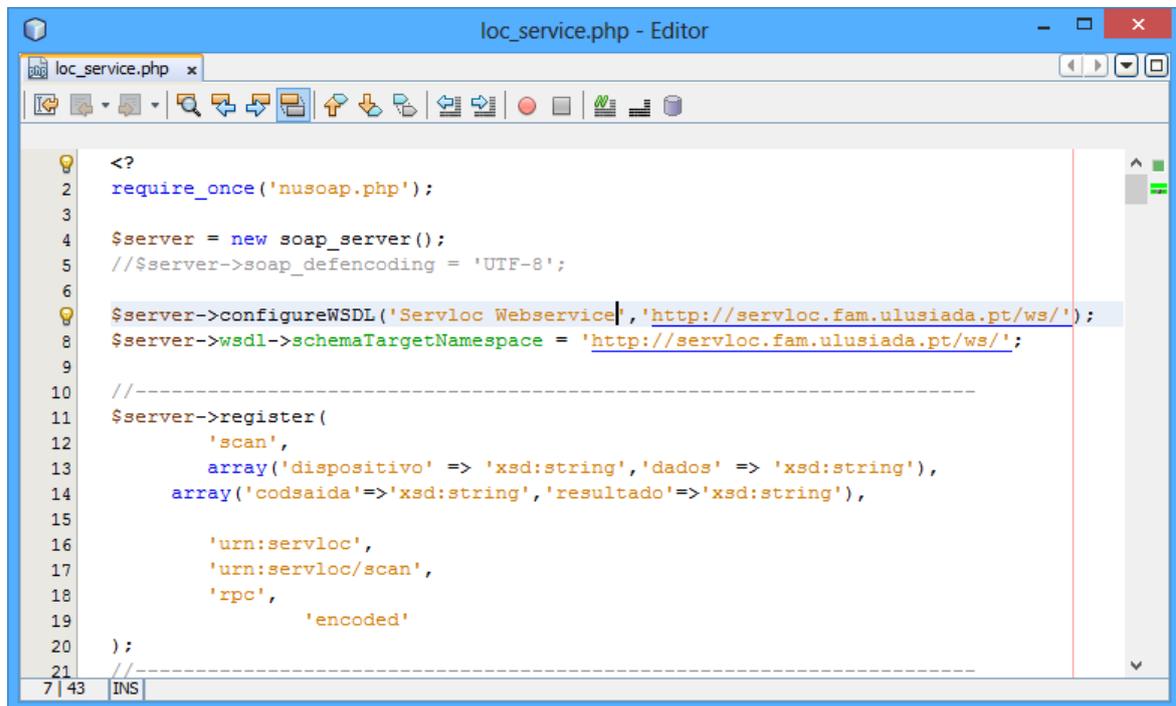
Close

Name: scan
Binding: Servloc WebserviceBinding
Endpoint: http://servloc.fam.ulusiada.pt/ws/loc_service.php
SoapAction: urn:servloc/scan
Style: rpc
Input:
use: encoded
namespace: urn:servloc
encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
message: scanRequest
parts:
dispositivo: xsd:string
dados: xsd:string
Output:
use: encoded
namespace: urn:servloc
encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
message: scanResponse
parts:
codsaida: xsd:string
resultado: xsd:string
Namespace: urn:servloc
Transport: http://schemas.xmlsoap.org/soap/http
Documentation:

Fig. 8 - Funções existentes no Servloc webservice

A Fig. 8 apresenta métodos com funcionalidades distintas que o *webservice* processa. Por exemplo, o método “scan”, sendo o cérebro do sistema de localização, recebe os dados (*mac address* encriptado da placa Wi-Fi do dispositivo, o nível de sinal e o *mac address* dos *access points*) da aplicação de localização, o “Servloc Locate Me”, em duas variáveis do tipo “string”, sendo a primeira (“dispositivo”) onde se armazena o *mac address* encriptado da placa Wi-Fi do dispositivo e a segunda (“dados”), onde se armazena o nível de sinal o *mac address* de todos os *access points*. A variável “dados”, armazena duas informações distintas separadas pelo caracter “;”. Quando a variável “dados” recebe “AA-AA-AA-AA-AA-AA;-57;BB-BB-BB-BB-BB-BB;-75”, esta informação é convertida para um *array* para que seja feita a interpretação correta dos dados. Após esta conversão, a identificação de um *access point* no *array* é feita através da paridade do índice do *array*, isto porque um índice par do *array* corresponderá ao nível de sinal que está associado ao *mac address* presente no índice anterior do *array*, por exemplo *array* no índice 0 terá “AA-AA-AA-AA-AA-AA” e no índice 2 terá “-57” e assim sucessivamente.

Para que o dispositivo móvel efetue a comunicação com o *webservice*, foi necessário criar um ficheiro desenvolvido em XML com o nome “loc_service.php” (Fig. 9), que contem vários métodos.



```
<?
2 require_once('nusoap.php');
3
4 $server = new soap_server();
5 // $server->soap_defencoding = 'UTF-8';
6
7 $server->configureWSDL('Servloc Webservice', 'http://servloc.fam.ulusiada.pt/ws/');
8 $server->wsdl->schemaTargetNamespace = 'http://servloc.fam.ulusiada.pt/ws/';
9
10 //-----
11 $server->register(
12     'scan',
13     array('dispositivo' => 'xsd:string', 'dados' => 'xsd:string'),
14     array('codsaida' => 'xsd:string', 'resultado' => 'xsd:string'),
15
16     'urn:servloc',
17     'urn:servloc/scan',
18     'rpc',
19     'encoded'
20 );
21 //-----
7 | 43 INS
```

Fig. 9 - Fragmento de código do Servloc webservice

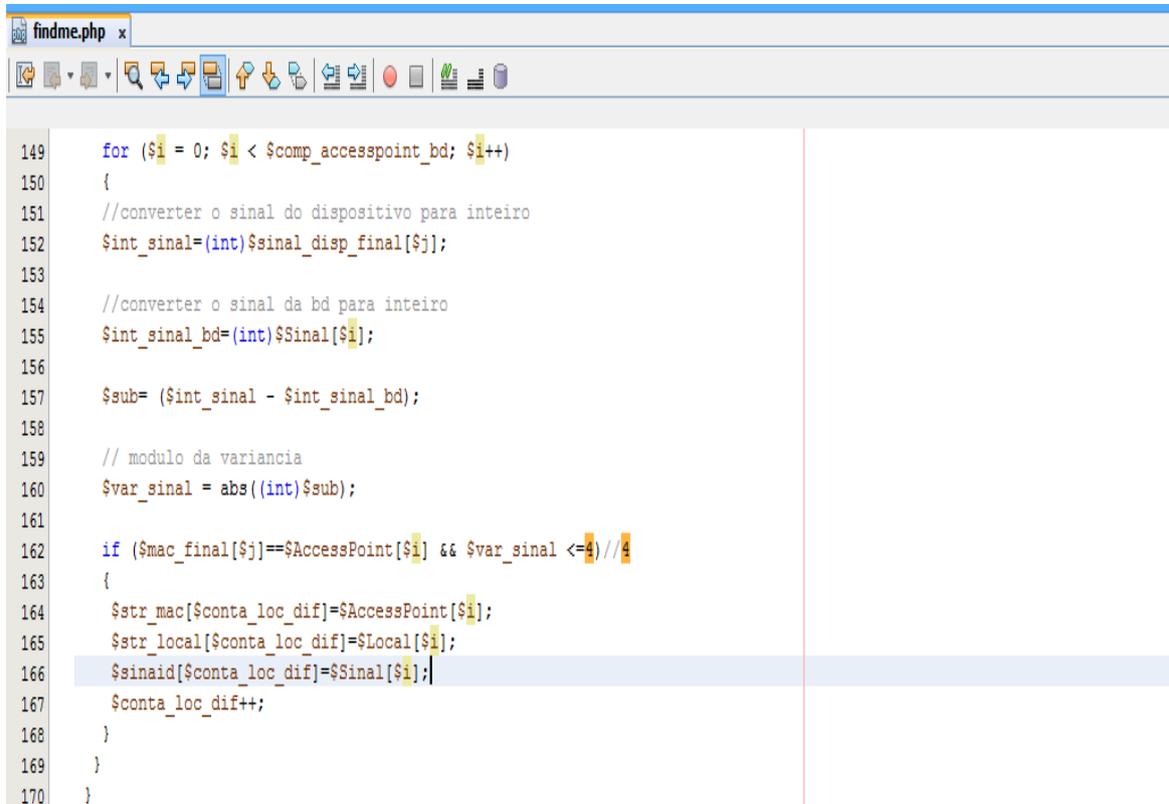
Para além do *webservice* determinar a localização, também permite o registo e remoção de dispositivos no sistema central, o sistema de autenticação dos mesmos e ainda a gestão de informação enviada pelo dispositivo móvel para ser colocada na base dados.

Este *webservice* está desenvolvido e apresentado numa página web³.

³ http://servloc.fam.ulusiada.pt/ws/loc_service.php

3.5. Algoritmo de localização

O algoritmo de localização tem como finalidade inferir a localização do dispositivo móvel, sendo que esta previsão é realizada pela comparação dos dados presentes na base dados com os dados enviados pela aplicação do dispositivo móvel (o Servloc Locate Me).

The image shows a screenshot of a code editor window titled 'findme.php'. The code is written in PHP and implements a location algorithm. It starts with a for loop that iterates through a list of access points. For each access point, it converts the received signal strength and the stored signal strength from the database into integers. It then calculates the absolute difference between these two values. If the received MAC address matches the access point's MAC address and the absolute difference is less than or equal to a threshold (4), it updates the local location information with the access point's name and signal strength. The code ends with a closing brace for the for loop.

```
149     for ($i = 0; $i < $comp_accesspoint_bd; $i++)
150     {
151         //converter o sinal do dispositivo para inteiro
152         $int_sinal=(int)$sinal_disp_final[$j];
153
154         //converter o sinal da bd para inteiro
155         $int_sinal_bd=(int)$Sinal[$i];
156
157         $sub= ($int_sinal - $int_sinal_bd);
158
159         // modulo da variancia
160         $var_sinal = abs((int)$sub);
161
162         if ($mac_final[$j]==$AccessPoint[$i] && $var_sinal <=4)//4
163         {
164             $str_mac[$conta_loc_dif]=$AccessPoint[$i];
165             $str_local[$conta_loc_dif]=$Local[$i];
166             $sinaid[$conta_loc_dif]=$Sinal[$i];
167             $conta_loc_dif++;
168         }
169     }
170 }
```

Fig. 10 - Código do algoritmo de localização

Quando o dispositivo móvel faz pesquisa dos *mac address* com os respectivos níveis de sinal, envia os mesmos através de uma variável de texto para o servidor. Através do webservice. O algoritmo obtém essa informação e efetua um processo de comparação com dados inseridos na base de dados pela aplicação “Servloc Backoffice”, apresentada no capítulo 3.3. Depois de realizado este processo será enviado para a aplicação do dispositivo móvel uma confirmação com a localização do dispositivo.

O pseudocódigo que se segue apresenta o sector mais importante do algoritmo de localização, ou seja, o momento de decisão:

```
for  $j = 0 \rightarrow comp\_mac\_disp\_final$  do  
  for  $i = 0 \rightarrow comp\_accesspoint\_bd$  do  
     $int\_sinal = sinal\_disp\_final[j]$   
     $int\_sinal\_bd = Sinal(i)$   
     $sub = int\_sinal - int\_sinal\_bd$   
     $var\_sinal = | sub |$   
    if  $mac\_final(j) = AccessPoint(i) \wedge var\_sinal \leq 4$  then  
       $str\_mac(conta\_loc\_dif) = AccessPoint(i)$   
       $str\_local(conta\_loc\_dif) = Local(i)$   
       $conta\_loc\_dif = conta\_loc\_dif + 1$   
    end if  
  end for  
end for  
for  $k = 0 \rightarrow comp\_local2$  do  
   $str\_comparar = str\_local(k)$   
   $conta\_precisao = 0$   
  for  $k2 = 0 \rightarrow comp\_local2$  do  
    if  $str\_comparar = str\_local(k2)$  then  
       $conta\_precisao = conta\_precisao + 1$   
    end if  
  end for  
   $str\_loc\_cont(k) = str\_comparar$   
   $str\_prec\_cont(k) = str\_precisao$   
end for
```

O pseudocódigo acima apresentado verifica se o *mac address* do *access point* encontrado pelo dispositivo está armazenado na base dados e se a diferença de nível de sinal entre ambos é inferior a quatro decibéis, caso afirmativo, o algoritmo encontrou a localização do dispositivo. A constante de valor quatro consiste na diferença de poder captação entre a placa de rede do computador e do dispositivo móvel, este valor foi definido depois da realização de testes ao algoritmo. Para mais detalhe sobre este algoritmo, o código encontra-se no Anexo 1.

4. Processamento da localização do dispositivo móvel

A aplicação de processamento da localização do dispositivo móvel, possui o nome “Servloc Locate Me”, tem a principal tarefa de permitir ao utilizador interagir no seu dispositivo móvel de forma a inferir a sua localização e dos seus “amigos”.

O “Servloc Locate Me” agrega, o sistema de autenticação, a apresentação de informação acerca dos *access points* encontrados, a localização do dono do dispositivo móvel, a última localização efetuada pelos dispositivos emparelhados (“amigos”), a encriptação de toda a informação relativamente aos dispositivos e ainda a pesquisa de dispositivos através da tecnologia *Bluetooth*.



Fig. 11 - Sistema de autenticação da aplicação "Servloc Locate Me "

O sistema de autenticação (Fig. 11) presente nesta aplicação foi construído de forma a garantir a existência de um utilizador por aplicação / dispositivo. Este sistema é constituído por três componentes: o registo, o início de sessão e a remoção do dispositivo na aplicação. Para o registo é necessário um e-mail válido, palavra-passe e *mac address* da placa Wi-Fi do dispositivo, que é obtido automaticamente e encriptado pela aplicação. Para a se efetuar o início de sessão o utilizador necessita de inserir palavra-passe de registo, pois a aplicação trata de obter e encriptar o *mac address* da placa de Wi-Fi e de seguida enviar estas informações para processamento no *webservice*. Este componente possui ainda um mecanismo para a recuperação da palavra-passe, sendo necessário inserir o e-mail de registo. Por último, para a remoção do dispositivo na aplicação, o utilizador terá de inserir a respetiva palavra-passe, para se confirmar a autenticação.



Fig. 12 - Interface gráfica do processo de localização

Numa primeira fase do processo de localização representada na fig. 12, esta aplicação efetua a deteção dos *access points* e apresenta ao utilizador as informações recolhidas, nomeadamente, o *mac address* e o nível de sinal. Estas informações são obtidas através da placa de Wi-Fi incorporada no dispositivo móvel. As mesmas informações são enviadas para o *webservice* juntamente com o *mac address* encriptado do dispositivo, embora esta informação não seja obrigatória para efetuar a localização do dispositivo, pois é apenas armazenada no histórico. Assim que a informação seja manipulada pelo *webservice*, a aplicação receberá a localização em forma de percentagem, por exemplo, 75% - Local 1 - 25 % Local 2. Esta informação é classificada graficamente com níveis de confiança, por exemplo, Local 1 com 75%. O mesmo resultado é apresentado um círculo preenchido com a cor verde, correspondente a um nível de confiança elevado. Se o Local 2 for apresentado por um círculo preenchido com a cor vermelha, é porque corresponde a um local com nível de confiança baixa.



Fig. 13 - Visualização da última localização dos "amigos"

Logo após da realização do processo de localização, são procurados dispositivos que estejam emparelhados com o dispositivo do utilizador. A pesquisa de apenas dispositivos emparelhados deve-se ao facto de manter a privacidade de dispositivos desconhecidos. A existência de emparelhamento entre dois dispositivos significa que ambos permitem partilhar informações entre eles, pois numa primeira ligação, a mesma teve de ser aceite por ambos dispositivos. Após ter encontrado os dispositivos emparelhados (“amigos”), a aplicação envia informações (*mac address* encriptado, nome ao qual o dispositivo está associado na placa *Bluetooth*) acerca do mesmo e o resultado da localização para o *webservice*, que por sua vez armazena na base dados. Posteriormente, estas informações serão agregadas ficando o utilizador como a última localização dos seus “amigos” (Fig. 13).



Fig. 14 - Gestão de "amigos" do utilizador

Para além da localização do dispositivo, o “Servloc Locate Me” permite a gestão dos “amigos” do utilizador (Fig. 14), sendo estes os dispositivos emparelhados. Esta gestão é feita de uma forma automática, na qual a aplicação adiciona todos os dispositivos que se apresentam na lista de dispositivos emparelhados à tabela “amigos do utilizador”. A inserção e a remoção de “amigos” será feita através das definições de emparelhamento do dispositivo.



Fig. 15 - Sistema de localização em modo treino

Por último, para posterior tratamento estatístico de confiabilidade do “Servloc Locate Me”, esta aplicação possui uma secção (Fig. 15) onde o utilizador antes de iniciar o processo de localização terá de escolher um local já existente na base dados correspondente ao espaço físico onde se encontra, um intervalo de tempo e repetição do processo de localização. A realização de várias amostragens permite obter dados mais realistas em relação à fiabilidade da aplicação, independentemente da hora do dia e de outras alterações às quais o espaço físico pode estar sujeito.

5. Monitorização do sistema

A página web foi idealizada com o intuito de apresentar de uma forma dinâmica a fiabilidade da aplicação de localização. O tratamento estatísticos dos dados recolhidos nos processos de localização e de “treino”, onde se armazena na base dados o local seleccionado pelo utilizador e o resultado final da localização, permite extrair um grau de fiabilidade da aplicação “Servloc Locate Me” (Fig. 16), que é apresentado sob a forma de uma página web.



Fig. 16 - Apresentação do gráfico de fiabilidade da aplicação "Servloc Locate Me"

Nesta página web são disponibilizados ainda os ficheiros necessários para a instalação de todas as aplicações (Servloc BackOffice e Servloc Locate Me) e ainda são apresentados gráficos com estatísticas obtidas através dos dados armazenados dos processos de localização (Fig. 17).

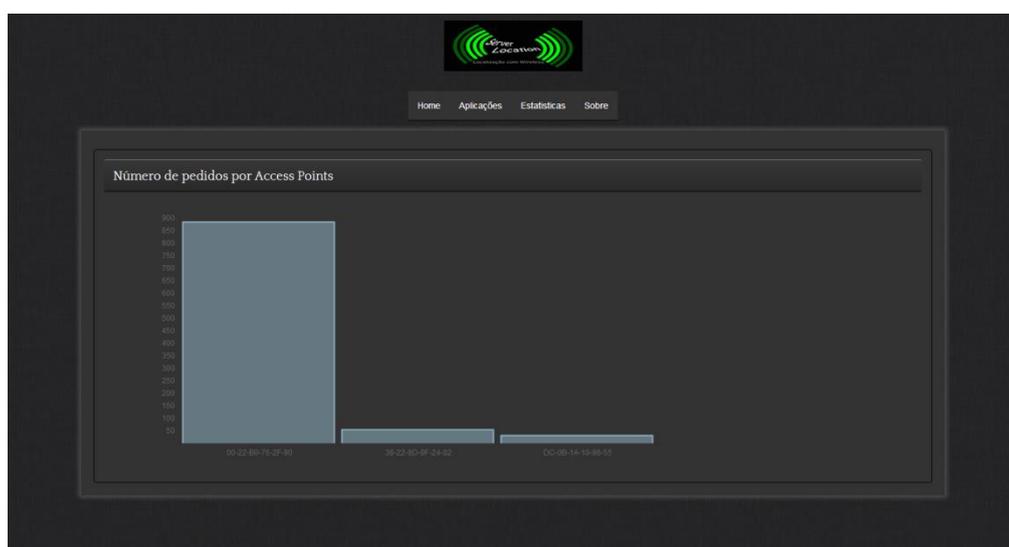


Fig. 17 - Apresentação do gráfico de estatística "Número de pedidos por Access Points"

6. Sistema de informação de suporte ao serviço de localização

A base de dados destina-se a armazenar, todos dados que as aplicações desenvolvidas neste trabalho recolhem.

Tabela	Ação	Registos	Tipo	Collation	Tamanho	Sobrecarga
<input type="checkbox"/> accesspoints	Visualizar Estrutura Procurar Inserir Limpar Eliminar	63	MyISAM	latin1_swedish_ci	3.7 KB	-
<input type="checkbox"/> assurance	Visualizar Estrutura Procurar Inserir Limpar Eliminar	526	InnoDB	ascii_general_ci	80.0 KB	-
<input type="checkbox"/> bluefriends	Visualizar Estrutura Procurar Inserir Limpar Eliminar	16	MyISAM	latin1_swedish_ci	3.5 KB	-
<input type="checkbox"/> devicebyaps	Visualizar Estrutura Procurar Inserir Limpar Eliminar	1,743	InnoDB	ascii_general_ci	272.0 KB	-
<input type="checkbox"/> lastaccessblue	Visualizar Estrutura Procurar Inserir Limpar Eliminar	3	MyISAM	latin1_swedish_ci	1.6 KB	128 Bytes
<input type="checkbox"/> locais	Visualizar Estrutura Procurar Inserir Limpar Eliminar	57	MyISAM	latin1_general_ci	3.6 KB	36 Bytes
<input type="checkbox"/> log	Visualizar Estrutura Procurar Inserir Limpar Eliminar	618	MyISAM	latin1_general_ci	90.5 KB	-
<input type="checkbox"/> servey	Visualizar Estrutura Procurar Inserir Limpar Eliminar	338	MyISAM	latin1_swedish_ci	11.8 KB	238 Bytes
<input type="checkbox"/> servey_nomes	Visualizar Estrutura Procurar Inserir Limpar Eliminar	338	MyISAM	latin1_swedish_ci	13.6 KB	-
<input type="checkbox"/> users	Visualizar Estrutura Procurar Inserir Limpar Eliminar	4	MyISAM	latin1_swedish_ci	2.5 KB	-

Fig. 18 - Tabelas da base dados

Na Fig.18 são apresentadas todas as tabelas utilizadas pelas aplicações (“Servloc BackOffice” e “Servloc Locate Me”), nomeadamente, a tabela “accesspoints”, a “assurance”, a “bluefriends”, a “devicebyaps”, a “locais”, a “log”, a “servey”, “servey_nomes” e a “users”.

Registo com os *access points* pesquisados.

A tabela “accesspoints” contém o *mac address* de todos *access points* encontrados pela aplicação “Servloc BackOffice”. Através desta tabela é possível ignorar *access points* desnecessários para a localização do dispositivo.

Campos da tabela:

- “id” - (numero inteiro auto incrementado);
- “mac” – (*mac address* dos routers existentes no local);

Registos dos armazenados pelo utilizador

A tabela “locais” tem como objetivo de armazenar a identificação de locais físicos, associando um nome definido pelo utilizador a cada local.

Campos da tabela:

- “id_local” - (número inteiro auto incrementado);
- “local” – (nome dos locais adicionados)

Registo de relacionamento entre *access points* e locais

A tabela “*servey*” faz uma relação das tabelas anteriores através dos id’s respetivos. Nesta tabela são armazenadas informações acerca da associação entre os *access points* e os locais.

Campos da tabela:

- “id” - (número inteiro auto incrementado);
- “id_local” - (id’s da tabela dos locais);
- “id_aps” – (id’s da tabela dos mac pesquisados);
- “sinal” - (nível de sinal de cada router);

Registo de decifragem do relacionamento entre *access points* e locais

A tabela “*servey_nomes*” possui a mesma função da tabela “*servey*” só que decifrada, ou seja, com os nomes e não com os id’s.

Campos da tabela:

- “Local” - (local);
- “AccessPoint” – (*mac address* do *access point*);
- “Sinal” - (sinal do *access point*);

Registos de armazenamento da resposta

Esta tabela “*log*” armazena a informação referente à resposta da localização enviada pelo *webservice*. Esta informação servirá para o utilizador visualizar as suas deslocações que foram identificadas neste sistema.

Campos da tabela:

- “id” - (número inteiro auto incrementado);
- “mac” – (*mac address* encriptado da placa Wi-Fi do dispositivo);
- “local” - (local do *AccessPoint*);
- “data_hora” – (data e hora)

Registos de armazenamento dos utilizadores da Aplicação

Na tabela “users” encontra-se a informação referente à autenticação do utilizador na aplicação. Através da consulta a esta tabela é que aplicação efetua o processo de autenticação do utilizador.

Campos da tabela:

- “id” - (número inteiro auto incrementado);
- “dispositivo” – (*mac address* encriptado da placa Wi-Fi do dispositivo);
- “email” – (e-mail para recuperação da palavra-passe);
- ”pass” – (palavra-passe de acesso à aplicação);

Registos de armazenamento de dispositivos Bluetooth encontrados

A tabela “bluefriends” é guardada a associação entre o dispositivo do utilizador e os dispositivos que estão emparelhados. Por isso, através desta tabela são definidos os “amigos” do dono do dispositivo na aplicação.

Campos da tabela:

- “id” - (número inteiro auto incrementado);
- “mac_dispositivo” – (*mac address* encriptado da placa Bluetooth do dispositivo do utilizador);
- ”amigos” - (*mac address* encriptado da placa Bluetooth do amigo);

Registo de armazenamento da última localização dos “amigos”

A tabela “lastaccessblue” armazena os dados da última localização encontrada dos “amigos” do utilizador.

Campos da tabela:

- “mac_bluetooth” – (*mac address* encriptado da placa de Bluetooth do dispositivo do amigo);
- ”nome” - (*mac address* encriptado da placa de Bluetooth do dispositivo móvel);
- ”local” - (local onde se encontra);
- ”data_hora” - (data e hora do último acesso);

Registo de dispositivos localizados por access points para tratamento estatístico

Na tabela “devicebyaps” são armazenados todos os *access points* encontrados pelos dispositivos, para posteriormente tratar estatisticamente estes dados, por exemplo através desta tabela pode-se obter informação do número de dispositivos que estiveram no alcance do *access point*.

Campos da tabela:

- “id” – (número inteiro auto incrementado);
- ”accesspoint” - (*mac address* do *access point*);
- ”sinalaccesspoint” - (nível de sinal capturado pelo dispositivo);
- “device” - (*mac address* encriptado da placa de Wi-Fi do dispositivo móvel);
- ”data_hora” - (data e hora de inserção da informação);

Registo de armazenamento com validação da localização

A tabela “assurance” armazena a informação que faz a validação da localização, armazenando o local definido pelo utilizador e o local enviado pela aplicação.

Campos da tabela:

- “id” – (número inteiro auto incrementado);
- ”locateM” - (nome do local mencionado pelo utilizador);
- ”locateF” - (nome do local enviado pela aplicação);
- ”datahora” - (data e hora de inserção da informação);

7. Segurança

Para segurar a privacidade do dono do dispositivo na troca de dados durante o processo de localização, foi utilizado o algoritmo de encriptação MD5 (Message-Digest algorithm 5), em que consiste num algoritmo de hash de 128 bits unidirecional desenvolvido pela RSA Data Security, Inc. Por ser um algoritmo unidirecional, uma palavra encriptada em MD5 não pode ser desencriptada novamente. Neste caso a identificação da palavra encriptada é feita através da comparação com outra palavra encriptada em MD5. Este tipo de encriptação é bastante utilizado em sistemas de autenticação. (Oliveira, 2012)

Este algoritmo de encriptação é utilizado para todos os dados que permite a identificação do dispositivo, nomeadamente, *mac address* da placa Wi-Fi e da placa Bluetooth. A implementação do algoritmo de encriptação MD5 é feita na aplicação “Servloc Locate Me”, pois assim a identificação do dispositivo nunca poderá ser enviada durante a troca de dados entre a aplicação e o *webservice*. A utilização deste tipo de encriptação na base dados torna os dados mais seguros, sendo impossível a sua desencriptação em possível ataque. Para mais detalhe sobre o algoritmo de encriptação, o código encontra-se no Anexo 2.

8. Validação

Depois de concluído todo processo de implementação prática, passamos à fase de testes das aplicações. A validação do sistema é um requisito importante para a sua qualidade em relação ao contexto em que vai operar. Pretende-se ainda fazer a deteção de falhas. Estas falhas serão detetadas através do processo de “Treino” existente na aplicação “Servloc Locate Me”, em que se o resultado da localização for diferente à localização selecionada pelo utilizador estamos perante uma falha na localização do dispositivo.

Para a realização dos testes foram utilizados dois dispositivos móveis, um **Samsung Galaxy Ace** com o sistema operativo **Android 2.3 Gingerbread** e um **Tablet Toshiba AT100** com o sistema operativo **Android 4.0 Ice Cream Sandwich**.

Teste em Edifício nº 1:

O primeiro edifício escolhido para teste é constituído por 7 divisões e por apenas um *access point*.

No primeiro teste foi escolhida a divisão “EDIFÍCIO Nº 1 QUARTO Nº 6” e teve um período de cinco horas. Durante este período foram registados 22 pedidos de localização, sendo que 12 destes pedidos tiveram um resultado acertado.

← T →		id	locateM	locateF	datahora
<input type="checkbox"/>	Editar	1	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 00:11:01
<input checked="" type="checkbox"/>	Editar	2	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 00:18:06
<input type="checkbox"/>	Editar	3	EDIFÍCIO Nº 1 QUARTO Nº 6	Não foi possível detectar a sua localização!!!!	2013-10-10 00:41:04
<input type="checkbox"/>	Editar	4	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 00:48:06
<input type="checkbox"/>	Editar	5	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 01:11:05
<input type="checkbox"/>	Editar	6	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 01:18:06
<input type="checkbox"/>	Editar	7	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 01:41:05
<input type="checkbox"/>	Editar	8	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 01:48:07
<input type="checkbox"/>	Editar	9	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 02:11:06
<input type="checkbox"/>	Editar	10	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 02:18:07
<input type="checkbox"/>	Editar	11	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 02:41:07
<input type="checkbox"/>	Editar	12	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 02:48:08
<input type="checkbox"/>	Editar	13	EDIFÍCIO Nº 1 QUARTO Nº 6	Não foi possível detectar a sua localização!!!!	2013-10-10 03:11:07
<input type="checkbox"/>	Editar	14	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 03:18:08
<input type="checkbox"/>	Editar	15	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 03:41:10
<input type="checkbox"/>	Editar	16	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 03:48:09
<input type="checkbox"/>	Editar	17	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 04:11:11
<input type="checkbox"/>	Editar	18	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 04:18:10
<input type="checkbox"/>	Editar	19	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 COZINHA	2013-10-10 04:41:18
<input type="checkbox"/>	Editar	20	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 04:48:10
<input type="checkbox"/>	Editar	21	EDIFÍCIO Nº 1 QUARTO Nº 6	Não foi possível detectar a sua localização!!!!	2013-10-10 05:11:29
<input type="checkbox"/>	Editar	22	EDIFÍCIO Nº 1 QUARTO Nº 6	EDIFÍCIO Nº 1 QUARTO Nº 6	2013-10-10 05:18:11

Fig. 19 - Resultados do primeiro teste de localização no edifício Nº1

No segundo teste foi escolhida a divisão “EDIFÍCIO Nº 1 QUARTO Nº 5” e teve um período de seis horas, entre as 23h00m e 6h00m do dia seguinte. Durante este período foram registados 85 pedidos de localização, sendo que 80 destes pedidos tiveram um resultado acertado.

				id	locateM	locateF	datahora
<input type="checkbox"/>	✎	Editar	✎	85	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:48:02
<input type="checkbox"/>	✎	Editar	✎	84	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:38:01
<input type="checkbox"/>	✎	Editar	✎	83	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:27:59
<input type="checkbox"/>	✎	Editar	✎	82	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:17:56
<input type="checkbox"/>	✎	Editar	✎	81	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:16:43
<input type="checkbox"/>	✎	Editar	✎	80	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:11:42
<input type="checkbox"/>	✎	Editar	✎	79	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:07:52
<input type="checkbox"/>	✎	Editar	✎	78	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:06:42
<input type="checkbox"/>	✎	Editar	✎	77	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 04:01:40
<input type="checkbox"/>	✎	Editar	✎	76	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:56:39
<input type="checkbox"/>	✎	Editar	✎	75	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:51:39
<input type="checkbox"/>	✎	Editar	✎	74	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:47:25
<input type="checkbox"/>	✎	Editar	✎	73	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:46:38
<input type="checkbox"/>	✎	Editar	✎	72	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:41:37
<input type="checkbox"/>	✎	Editar	✎	71	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 COZINHA	2013-10-11 03:37:24
<input type="checkbox"/>	✎	Editar	✎	70	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:36:37
<input type="checkbox"/>	✎	Editar	✎	69	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:31:36
<input type="checkbox"/>	✎	Editar	✎	68	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:27:17
<input type="checkbox"/>	✎	Editar	✎	67	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:26:35
<input type="checkbox"/>	✎	Editar	✎	66	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:21:35
<input type="checkbox"/>	✎	Editar	✎	65	EDIFÍCIO Nº 1 QUARTO 5	EDIFÍCIO Nº 1 QUARTO 5	2013-10-11 03:17:12

Fig. 20 - Resultados do segundo teste de localização no edifício Nº1

Para o último teste realizado neste Edifício foi escolhida a divisão “EDIFÍCIO Nº 1 QUARTO 1” e teve um período de cinco horas. Durante este período foram registados 31 pedidos de localização, sendo que 28 destes pedidos tiveram um resultado de localização correto.

				id	locateM	locateF	datahora
<input type="checkbox"/>	✎	Editar	✎	86	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 08:25:57
<input type="checkbox"/>	✎	Editar	✎	87	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 08:35:57
<input type="checkbox"/>	✎	Editar	✎	88	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 08:45:58
<input type="checkbox"/>	✎	Editar	✎	89	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 08:55:58
<input checked="" type="checkbox"/>	✎	Editar	✎	90	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 2	2013-10-12 09:05:59
<input type="checkbox"/>	✎	Editar	✎	91	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 09:15:59
<input type="checkbox"/>	✎	Editar	✎	92	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 09:26:00
<input type="checkbox"/>	✎	Editar	✎	93	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 09:36:00
<input type="checkbox"/>	✎	Editar	✎	94	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 09:46:01
<input type="checkbox"/>	✎	Editar	✎	95	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 09:56:01
<input type="checkbox"/>	✎	Editar	✎	96	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 10:06:02
<input type="checkbox"/>	✎	Editar	✎	97	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 10:16:02
<input type="checkbox"/>	✎	Editar	✎	98	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 10:26:02
<input type="checkbox"/>	✎	Editar	✎	99	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 10:36:03
<input type="checkbox"/>	✎	Editar	✎	100	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 10:46:04
<input type="checkbox"/>	✎	Editar	✎	101	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 10:56:04
<input type="checkbox"/>	✎	Editar	✎	102	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 11:06:04
<input type="checkbox"/>	✎	Editar	✎	103	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 2	2013-10-12 11:16:05
<input type="checkbox"/>	✎	Editar	✎	104	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 11:26:05
<input type="checkbox"/>	✎	Editar	✎	105	EDIFÍCIO Nº 1 QUARTO 1	EDIFÍCIO Nº 1 QUARTO 1	2013-10-12 11:36:06

Fig. 21 - Resultados do terceiro teste de localização no edifício Nº1

Através dos resultados obtidos do teste de localização no interior do edifício N°1, verifica-se que a probabilidade de erro é pequena, sendo que numa amostra de 138 pedidos de localização obteve-se 120 resultados de localização corretos, por isso a probabilidade de erro de localização neste edifício é de 0.13.

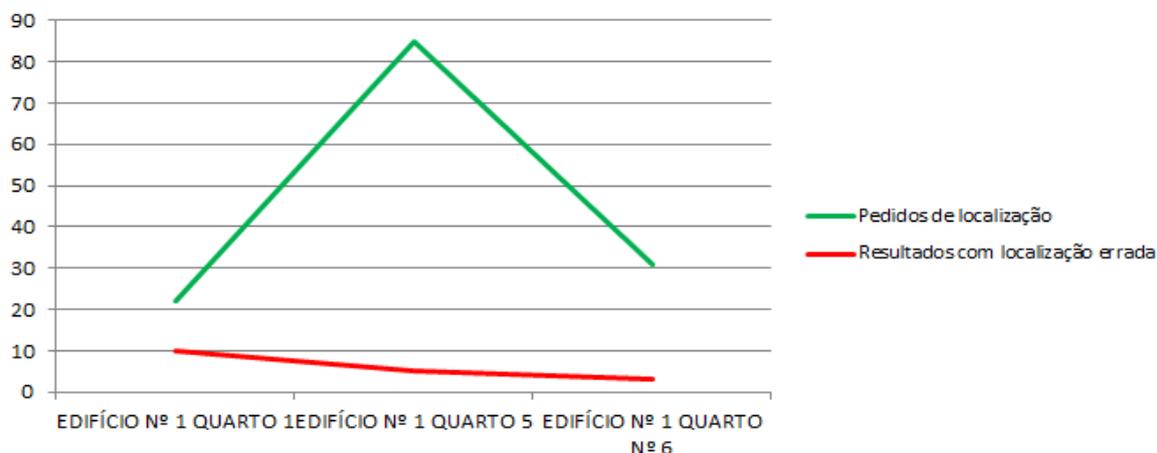


Fig. 22 - Pedidos e resultados errados de localização no edifício nº1

Teste em Edifício nº 2:

O segundo edifício escolhido para teste é constituído por 5 divisões e possui dois *access points*.

A divisão escolhida para o primeiro teste neste edifício foi “EDIFÍCIO N° 2 SALA DE ESTAR” e o período de teste foi de uma hora. Durante este período foram registados 60 pedidos de localização, sendo que 54 destes pedidos tiveram um resultado acertado.

	id	locateM	locateF	datahora
<input type="checkbox"/> Editar	466	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:18:01
<input type="checkbox"/> Editar	467	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 GARAGEM	2013-10-13 18:19:02
<input type="checkbox"/> Editar	468	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:20:03
<input type="checkbox"/> Editar	469	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:21:04
<input type="checkbox"/> Editar	470	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:22:04
<input type="checkbox"/> Editar	471	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:23:05
<input type="checkbox"/> Editar	472	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 GARAGEM	2013-10-13 18:24:06
<input type="checkbox"/> Editar	473	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:25:06
<input type="checkbox"/> Editar	474	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:26:07
<input type="checkbox"/> Editar	475	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:27:07
<input type="checkbox"/> Editar	476	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:28:08
<input type="checkbox"/> Editar	477	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:29:08
<input type="checkbox"/> Editar	478	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:30:09
<input type="checkbox"/> Editar	479	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:31:10
<input type="checkbox"/> Editar	480	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:32:10
<input type="checkbox"/> Editar	481	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 GARAGEM	2013-10-13 18:33:11
<input type="checkbox"/> Editar	482	EDIFÍCIO N° 2 SALA DE ESTAR	EDIFÍCIO N° 2 SALA DE ESTAR	2013-10-13 18:34:11

Fig. 23 - Resultados do primeiro teste de localização no edifício N°2

A divisão escolhida para o segundo teste neste edifício foi “EDIFÍCIO Nº 2 QUARTO 1” e o período de teste foi de uma hora. Durante este período foram registados 51 pedidos de localização, sendo que 28 destes pedidos tiveram um resultado acertado.

<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	563	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:23:23
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	562	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:22:15
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	561	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:21:14
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	560	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:20:13
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	559	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:19:13
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	558	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:18:11
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	557	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:17:10
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	556	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:16:06
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	555	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 GARAGEM	2013-10-14 22:15:05
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	554	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 QUARTO 1	2013-10-14 22:14:04
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	553	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 QUARTO 1	2013-10-14 22:13:03
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	552	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 QUARTO 1	2013-10-14 22:11:45
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	551	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 QUARTO 1	2013-10-14 22:10:44
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	550	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 QUARTO 1	2013-10-14 22:09:43
<input type="checkbox"/>		Editar		Inline Edit		Copiar		Remover	549	EDIFÍCIO Nº 2 QUARTO 1	EDIFÍCIO Nº 2 QUARTO 1	2013-10-14 22:08:41

Fig. 24 - Resultados do segundo teste de localização no edifício Nº2

Os resultados obtidos no teste de localização no interior deste edifício, foram menos satisfatórios que os resultados obtidos no edifício nº 1, porque numa amostra de 111 pedidos de localização obteve-se 82 resultados de localização acertados, o que apresenta uma probabilidade de erro de localização de 0.26.

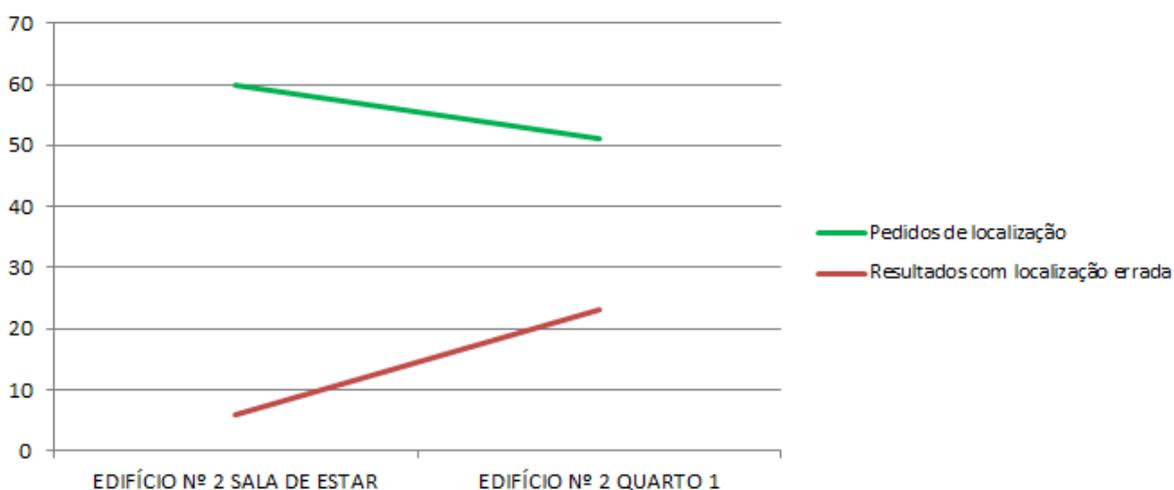


Fig. 25 - Pedidos e resultados errados de localização no edifício nº2

9. Conclusões e perspetivas de trabalho futuro

Foi implementado um sistema que tira partido de uma infraestrutura Wi-Fi existente para inferir a localização de dispositivos. Foi construído um protótipo do sistema de localização. Os resultados principais foram a criação de duas aplicações (“Servloc BackOffice” e “Servloc Locate Me”) de apoio para que o sistema seja minimamente automatizado e a criação de uma página *web* para informação de estatísticas. O "Servloc BackOffice" permite fazer o mapeamento dos locais em função nível do sinal dos pontos de acesso da rede Wi-Fi. O Servloc é um serviço web que é usado pelos dispositivos móveis para receber os pedidos de localização e enviar a respetiva resposta. Como as aplicações usam uma base dados comum, permitiu fazer a integração de informações que se encontram usualmente em plataformas distintas. A integração revelou-se fundamental para automatizar os procedimentos que compõem a arquitetura deste sistema.

No desenvolvimento deste sistema houve algumas dificuldades que se tiveram de ultrapassar, nomeadamente, a privacidade do utilizador, o desenvolvimento da aplicação de localização para o sistema operativo **Android**, a comunicação entre o *webservice* e a aplicação do dispositivo móvel e a variação do nível de sinal dos *access points*.

Apesar das variações do nível de sinal, o mecanismo de localização assegurou os requisitos mínimos para que um individuo se oriente dentro de um edifício. A privacidade do utilizador e a localização de dispositivos emparelhados com o dispositivo do utilizador foram também dois requisitos concluídos com sucesso.

Este sistema de localização pode ser utilizado tanto em locais privados como em locais públicos, como por exemplo casas, hospitais, escolas, centros comerciais, entre outras.

Perspetivas de trabalho futuro: Navegação no interior de edifícios para auxiliar o utilizador a encontrar o caminho até ao seu destino, ampliar os testes realizados com o protótipo e comparar com outros sistemas existentes.

10. Referências bibliográficas

- A Europa gosta do Wi-Fi: novo estudo recomenda a disponibilização de mais espectro.* (01 de 08 de 2013). Obtido em 20 de 09 de 2013, de EUROPA - European Union website, the official EU website: http://europa.eu/rapid/press-release_IP-13-759_pt.htm
- Bejuri, W. M., Mohamad, M. M., & Sapri, M. (2011). Ubiquitous Positioning: A Taxonomy for Location Determination on Mobile Navigation System. *Signal & Image Processing : An International Journal(SIPIJ) Vol.2, No.1.*
- Carvalho, J. F. (2007). Localização de Dispositivos Móveis em Redes Wi-Fi.
- Huang, S., Proulx, F., & Ratti, C. (2004). iFIND: A peer-to-peer application for real-time location monitoring on the MIT Campus.
- Moura, A. I. (2007). WBLs: um sistema de localização de dispositivos móveis em redes Wi-Fi.
- Nguyen, H. A., & Giordano, S. (2009). Routing in Opportunistic Networks. *IJACI*, 19-38.
- Oliveira, R. R. (2012). Criptografia simétrica e assimétrica: os principais algoritmos de cifragem.
- Serrano, O., Cañas, J. M., Matellán, V., & Rodero, L. (2004). Robot localization using. *Workshop de Agentes Físicos WAF'2004.*
- Tse, D. (2005). *Fundamentals of Wireless Communication.* New York.
- Youssef, M., & Agrawala, A. (2005). The Horus WLAN Location Determination System. *3rd International conference on Mobile Systems.*
- Zhou, R. (2006). Enhanced Wireless Indoor tracking System in multi-floor buildings . *12th International Conference of European University Information Systems.*

11. Anexos

Anexo 1 – Algoritmo de Localização

<?

```
//variavel do dispositivo
$dados_aps_sinal= $dados;

//separa os dados do dispositivo e coloca num vector
$dados_aps_sinal_sep= explode(';', $dados_aps_sinal);
$comp_dados = count($dados_aps_sinal_sep);
$conta_disp =0;

//dividir os dados mac e sinal
for ($i=0;$i<$comp_dados;$i+=2)
{
    $mac_disp[$conta_disp]=$dados_aps_sinal_sep[$i];
    $conta_disp++;
}

$conta_disp =0;

for ($i=1;$i<$comp_dados;$i+=2)
{
    $sinal_disp[$conta_disp]=$dados_aps_sinal_sep[$i];
    $conta_disp++;
}

$macfinal=$mac_disp;
$comp_macfinal =count($macfinal);

$sinal2=0;
$media=0;
$mac_final=NULL;
$indice_mac_final=1;

for ($l = 0; $l < $comp_macfinal; $l++)
{
    $conta_sinal =0;
    $sinal2=0;
    $comp_mac_final = count($mac_final);
    $conta_exist=0;

    for ($m = 0; $m < $comp_macfinal; $m++)
    {
        if ($macfinal[$l]==$macfinal[$m])
        {
            $sinal2= ((int)$sinal2 + (int)$sinal_disp[$m]);
            $conta_sinal++;
        }
    }

    if ($conta_sinal>=1)
    {
        if ($comp_mac_final==0)
        {
            (int)$media=($sinal2/$conta_sinal);
            $sinal_disp_final[0]=(int)$media ;
            $mac_final[0]=$macfinal[$l];
        }
    }
}
```

```

    }
    else
    {
        for ($m2 = 0; $m2 < $comp_mac_final; $m2++)
        {
            if ($macfinal[$l]==$mac_final[$m2])
            {
                $conta_exist++;
            }
        }

        if($conta_exist==0)
        {
            (int)$media=($sinal2/$conta_sinal);

            $sinal_disp_final[$indice_mac_final]=(int)$media;

            $mac_final[$indice_mac_final]=$macfinal[$l];

            $indice_mac_final++;
            break;
        }
    }
}

$conta_repit=0;
$comp_mac_final = count($mac_final);

for ($n = 0; $n < $comp_macfinal; $n++)
{
    for ($n2 = 0; $n2 < $comp_mac_final; $n2++)
    {
        if ($mac_final[$n2]==$macfinal[$n])
        {
            $conta_repit++;
        }
    }

    if ($conta_repit==0)
    {
        $sinal_disp_final[$indice_mac_final]=$sinal_disp[$n];

        $mac_final[$indice_mac_final]=$macfinal[$n];

        $indice_mac_final++;
    }
    $conta_repit=0;
}

$comp_mac_disp_final=count($sinal_disp_final);

mysql_select_db("servloc", $con);

$result = mysql_query("SELECT * FROM servey_nomes ORDER BY Sinal DESC,
local ASC");

$conta=0;

```

```

while($var = mysql_fetch_array($result))
{
    $Local[$conta] = ($var['Local']);
    $AccessPoint[$conta] = $var['AccessPoint'];
    $Sinal[$conta] = $var['Sinal'];
    $conta++;
}

mysql_free_result($result); //libertar os resultados da memória

$comp_accesspoint_bd=count($AccessPoint);
$comp_mac_disp_final=count($sinal_disp_final);

$conta_pres =0;
$conta_loc_dif=0;
$int_sinal=0;
$str_mac[0]=0;
$str_local=NULL;

for ($j = 0; $j <$comp_mac_disp_final; $j++)
{
    for ($i = 0; $i < $comp_accesspoint_bd; $i++)
    {
        //converter o sinal do dispositivo para inteiro
        $int_sinal=(int)$sinal_disp_final[$j];

        //converter o sinal da bd para inteiro
        $int_sinal_bd=(int)$Sinal[$i];

        $sub= ($int_sinal - $int_sinal_bd);

        // modulo da variancia
        $var_sinal = abs((int)$sub);

        if ($mac_final[$j]==$AccessPoint[$i] && $var_sinal <=4)//7
        {
            $str_mac[$conta_loc_dif]=$AccessPoint[$i];
            $str_local[$conta_loc_dif]=$Local[$i];
            $sinaid[$conta_loc_dif]=$Sinal[$i];
            $conta_loc_dif++;
        }
    }
}

$com_local2 =count($str_local);
$conta_precisao=0;

for ($k = 0; $k < $com_local2; $k++)
{
    $str_comparar = $str_local[$k];
    $conta_precisao=0;

    for ($k2 = 0; $k2 < $com_local2; $k2++)
    {
        if ($str_comparar==$str_local[$k2])
        {
            $conta_precisao++;
        }
    }

    $str_loc_cont[$k]=$str_comparar;
}

```

```

    $str_prec_cont[$k]=$conta_precisao;
}

mysql_select_db("servloc", $con);

$result = mysql_query("SELECT * FROM locais ORDER BY local ASC");

$conta=0;
$conta100=0;

while($var = mysql_fetch_array($result))
{
    $local_bd[$conta] = ($var['local']);
    $conta++;
}

$com_local_bd =count($local_bd);
$str_loc_bd[0]="0";
$str_cont_loc_bd[0]="0";
$conta_precisao2=0;

for ($k = 0; $k < $com_local_bd; $k++)
{
    $str_comparar2 = $local_bd[$k];
    $conta_precisao2=0;

    for ($k2 = 0; $k2 < $com_local2; $k2++)
    {
        if ($str_comparar2==$str_local[$k2])
        {
            $conta_precisao2++;
        }
    }

    if ($conta_precisao2!=0)
    {
        $str_loc_bd[$conta100]=$str_comparar2;
        $str_cont_loc_bd[$conta100]=$conta_precisao2;
        $conta100++;
    }
}

$com_str_loc_bd=count($str_loc_bd);
$conta_mac_encont =0;

if ($com_local2!=0)
{
    for ($j2 = 0; $j2 < $com_str_loc_bd; $j2++)
    {
        $str_cont_loc_bd[$j2] . "<br />";
        (int)$str_cont_loc_bd[$j2]= ($str_cont_loc_bd[$j2]*100/($com_local2))
;
        $str_loc_bd[$j2] ;
        $conta_mac_encont = $conta_mac_encont + $str_cont_loc_bd[$j2];
    }

    array_multisort($str_cont_loc_bd, SORT_DESC, $str_loc_bd);

    $final =(int)$str_cont_loc_bd[0] ."% --> " . $str_loc_bd[0] . ";" ;
}

```

```

    for ($o2 = 1; $o2 <$com_str_loc_bd; $o2++)
    {
        $final = $final . (int)$str_cont_loc_bd[$o2] . "% --> " .
        $str_loc_bd[$o2] . ";";
    }

    $resultado = $final;
}
else
{
    $resultado= "Não foi possível detectar a sua localização!!!!!";
}

mysql_close($con);

?>

```

Anexo 2 – Algoritmo de encriptação

```

package servloc.android;

import java.security.SecureRandom;

import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

public class Encoding {

    public static String encrypt(String seed, String cleartext) throws
Exception {
        byte[] rawKey = getRawKey(seed.getBytes());
        byte[] result = encrypt(rawKey, cleartext.getBytes());
        return toHex(result);
    }

    public static String decrypt(String seed, String encrypted) throws
Exception {
        byte[] rawKey = getRawKey(seed.getBytes());
        byte[] enc = toByte(encrypted);
        byte[] result = decrypt(rawKey, enc);
        return new String(result);
    }

    private static byte[] getRawKey(byte[] seed) throws Exception {
        KeyGenerator kgen = KeyGenerator.getInstance("AES");
        SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
        sr.setSeed(seed);
        kgen.init(128, sr); // 192 and 256 bits may not be available
        SecretKey skey = kgen.generateKey();
        byte[] raw = skey.getEncoded();
        return raw;
    }

    private static byte[] encrypt(byte[] raw, byte[] clear) throws
Exception {
        SecretKeySpec keySpec = new SecretKeySpec(raw, "AES");

```

```

        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, keySpec);
        byte[] encrypted = cipher.doFinal(clear);
        return encrypted;
    }

    private static byte[] decrypt(byte[] raw, byte[] encrypted) throws
Exception {
        SecretKeySpec keySpec = new SecretKeySpec(raw, "AES");
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE, keySpec);
        byte[] decrypted = cipher.doFinal(encrypted);
        return decrypted;
    }

    public static String toHex(String txt) {
        return toHex(txt.getBytes());
    }

    public static String fromHex(String hex) {
        return new String(toByte(hex));
    }

    public static byte[] toByte(String hexString) {
        int len = hexString.length()/2;
        byte[] result = new byte[len];
        for (int i = 0; i < len; i++)
            result[i] = Integer.valueOf(hexString.substring(2*i,
2*i+2), 16).byteValue();
        return result;
    }

    public static String toHex(byte[] buf) {
        if (buf == null)
            return "";
        StringBuffer result = new StringBuffer(2*buf.length);
        for (int i = 0; i < buf.length; i++) {
            appendHex(result, buf[i]);
        }
        return result.toString();
    }

    private final static String HEX = "0123456789ABCDEF";
    private static void appendHex(StringBuffer sb, byte b) {
        sb.append(HEX.charAt((b>>4)&0x0f)).append(HEX.charAt(b&0x0f));
    }
}

```