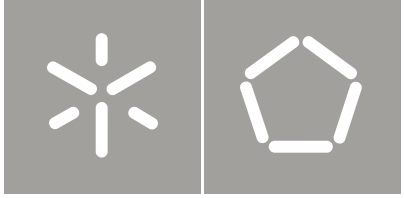
Universidade do Minho
Escola de Engenharia

Ilídio Miguel Pereira da Silva

Avaliação da Tecnologia Bluetooth como Sensor da Mobilidade Urbana

Avaliação da Tecnologia Bluetooth como Sensor da Mobilidade Urbana
Ilídio Miguel Pereira da Silva
UMinho | 2011

Outubro de 2011



Universidade do Minho
Escola de Engenharia

Ilídio Miguel Pereira da Silva

Avaliação da Tecnologia Bluetooth como Sensor da Mobilidade Urbana

Dissertação de Mestrado
Engenharia de Comunicações

Trabalho efectuado sob a orientação do
Professor Doutor Adriano Moreira

Outubro de 2011

Dedicado a todas as pessoas que me acompanharam

durante o meu percurso académico.

Agradecimentos

Em primeiro lugar, agradeço aos meus pais, bem como à minha irmã e avós por toda a motivação que me transmitiram, durante a realização deste trabalho.

Gostaria também de agradecer a toda a minha restante família, que carinhosamente sempre me apoiou, em especial à minha tia Margarida, ao meu tio Vítor e ainda ao meu tio Agostinho que, durante todo o meu percurso académico, muito me ajudaram com conselhos e orientações úteis.

Agradeço ao Professor Doutor Adriano Moreira todo o tempo que generosamente disponibilizou para me ajudar na realização deste trabalho de mestrado, e pelo seu empenho na minha melhor orientação, transmitindo-me sempre os seus pontos de vista, o seu conhecimento e fornecendo-me valiosas ideias para a conclusão deste trabalho.

Um agradecimento especial para o Hélder Lemos, o Ângelo Conde, o Carlos Sousa e ainda para o Miguel Almeida, com os quais tive o prazer de trabalhar durante parte deste trabalho. Quero ainda agradecer todo o apoio e incentivo, mostrado por todos os meus amigos, docentes e colegas de curso. Afirmo ainda aqui um especial agradecimento aos meus amigos, Marco Costa, Filipe Miranda e ainda ao Sébastian Leroux, que com as suas abordagens e opiniões me esclareceu perante dúvidas ou dificuldades encontradas.

Não posso terminar, sem agradecer a todos os lojistas que gentilmente cederam parte do seu espaço de trabalho para ser possível realizar este trabalho, referindo especialmente o Sr. Miguel e o Sr. Alves. Agradeço ainda a todos aqueles que se mostraram disponíveis para colaborarem neste projecto, através da instalação da aplicação Epi, bem como na revisão desde documento.

Toda a vossa amizade, indicações, incentivos e conselhos, ajudaram-me a concluir este trabalho, e por consequência, o meu curso.

Para todos vocês um grande e forte abraço, e o meu **Muito Obrigado**!

Resumo

A Tecnologia Bluetooth está presente na vida das pessoas. Dispositivos como telemóveis, PDAs, computadores portáteis e mais recentemente veículos, são apenas alguns exemplos onde se encontram presentes interfaces Bluetooth.

Uma vez que esta tecnologia foi desenvolvida para comunicação entre dispositivos, que normalmente se encontram em ambientes de mobilidade reduzida, como por exemplo um escritório ou uma sala, seria interessante conhecer o comportamento da tecnologia num ambiente maior mobilidade dos dispositivos.

Este é um dos objectivos deste projecto de dissertação, avaliar o comportamento da tecnologia Bluetooth num ambiente que inclua elevada mobilidade dos dispositivos. Nesse sentido, foram realizados alguns testes, neste tipo de ambiente, que tinham por finalidade determinar a probabilidade de detecção de um dispositivo. Paralelamente ao estudo anterior, também foi feito um outro com o objectivo de conhecer os padrões de comportamento espacial das pessoas numa determinada área geográfica. Estes padrões são identificados através da análise de dados recolhidos no local do teste, salvaguardando sempre a privacidade das pessoas.

O trabalho desenvolvido permitiu ainda construir e integrar um módulo de recolha de informações, relacionadas com a tecnologia Bluetooth, numa aplicação de difusão de mensagens.

Abstract

Bluetooth is a technology is more and more present in people's lives. Devices like mobile phones, PDAs, laptops and more recently vehicles are just some examples where Bluetooth interfaces are present.

Since this technology was developed for communication between devices that typically operate in environments with limited mobility, such as an office or a living room, it would be interesting to know, the behavior of the technology, in another environment. One scenario that offers greater mobility for the devices.

This is the purpose of this dissertation project, to evaluate the performance of the Bluetooth technology in an environment that includes high mobility. In this sense, were performed some tests in this kind of environment whose purpose was to determine the probability of a detection device.

In parallel with the previous study was also made another, whit the purpose to know the standards of the spatial behavior of people in a given geographical area. These patterns are identified through analysis of data collected at the test site, always preserving the privacy of people.

With this work, it was possible to build and integrate a module for collecting information related to Bluetooth technology, in an application of broadcast messages.

Índice Geral

ÍNDICE DE FIGURAS ..... **XIII**

ÍNDICE DE TABELAS ..... **XVII**

ÍNDICE DE GRÁFICOS ..... **XIX**

LISTA DE ACRÓNIMOS ..... **XXIII**

1. INTRODUÇÃO **1**

1.1. CONTEXTUALIZAÇÃO DO TEMA E MOTIVAÇÃO **1**

1.2. OBJECTIVOS PROPOSTOS **2**

1.3. ABORDAGEM DE INVESTIGAÇÃO **3**

1.4. ORGANIZAÇÃO E ESTRUTURA DO DOCUMENTO **5**

2. COLLABORATIVE SENSING NETWORKS **7**

2.1. CENÁRIOS DE APLICAÇÃO **8**

2.2. DESENVOLVIMENTO DE SISTEMAS **10**

2.2.1. REQUISITOS E DESAFIOS PARA OS SISTEMAS 12

2.2.2. SEGURANÇA DOS SISTEMAS 15

2.3. PROJECTOS EM DESENVOLVIMENTO **18**

2.3.1. *URBAN SENSING* - UCLA 18

2.3.2. CARTEL 18

2.3.3. METROSENSE 19

2.3.4. AROUND KNOWLEDGE 19

3. TECNOLOGIA DE COMUNICAÇÃO BLUETOOTH **21**

3.1. INTRODUÇÃO **21**

3.1.1. EVOLUÇÃO DA TECNOLOGIA BLUETOOTH 23

3.2. PROCESSO DE DESCOBERTA DE DISPOSITIVOS **25**

3.2.1. PROCESSO DE CONEXÃO E TOPOLOGIAS DE REDE 29

3.3. *STACK DO* BLUETOOTH 31

3.4. PROCESSO DE DESCOBERTA DE SERVIÇOS 34

4. ESPECIFICAÇÃO E DESENVOLVIMENTO DA SOLUÇÃO 37

4.1. CENÁRIO DE OPERAÇÕES 38

4.2. INFORMAÇÃO A RECOLHER 39

4.3. HARDWARE PARA INSTALAÇÃO DA APLICAÇÃO 41

4.3.1. HARDWARE A UTILIZAR 42

4.3.2. LINGUAGEM DE PROGRAMAÇÃO E API A UTILIZAR 45

4.4. ARQUITECTURA DA APLICAÇÃO DE *SCAN* 47

4.4.1. CLASSE *BLUETOOTHSCANS* 49

4.4.2. CLASSE *BLUETOOTHCOM* 51

4.4.3. CLASSE *SCANEDBTDEVICE* 53

4.4.4. CLASSE *STORAGESCANBT* 53

4.4.5. ARMAZENAMENTO DOS DADOS RECOLHIDOS 54

4.4.6. FUNCIONAMENTO EM "MODO INDEPENDENTE" DA APLICAÇÃO 55

5. IMPLEMENTAÇÃO E TESTES 57

5.1. IMPLEMENTAÇÃO E CENÁRIO DE TESTES 57

5.2. TESTES REALIZADOS 60

5.2.1. TESTE SEM MOBILIDADE 60

5.2.2. TESTES COM MOBILIDADE 61

5.3. PROCESSOS PARA ANÁLISE DOS RESULTADOS 63

5.3.1. PROCESSO DE ANÁLISE POR LOCAL 63

5.3.2. PROCESSO DE ANÁLISE GLOBAL 69

6. RESULTADOS OBTIDOS 79

6.1. TESTE SEM MOBILIDADE 79

6.2. TESTES COM ELEVADA MOBILIDADE 82

6.2.1. TESTE 1 – ANÁLISE DE RESULTADOS 83

6.2.2. TESTE 2 - ANÁLISE DE RESULTADOS 91

6.2.3. TESTE 3 - ANÁLISE DE RESULTADOS 99

6.2.4. CÁLCULO DAS PROBABILIDADES DE DETECÇÃO 106

6.2.4.1 Análise por Local *106*

6.2.4.2. Análise Global *110*

6.3. PROCESSOS ANALÍTICOS 114

6.3.1 PROCESSO ANALÍTICO DE ANÁLISE POR LOCAL 115

6.3.2 PROCESSO ANALÍTICO DE ANÁLISE GLOBAL 122

6.4 COMPARAÇÃO ENTRE OS PROCESSOS DE ANÁLISE 127

6.4.1 ANÁLISE POR LOCAL 128

6.4.2 ANÁLISE GLOBAL 130

7. INTEGRAÇÃO NA APLICAÇÃO EPI 133

7.1. APLICAÇÃO EPI 133

7.1.1. NOVAS FUNCIONALIDADES DESENVOLVIDAS 135

7.1.2. SERVIDOR 140

7.1.3. PROTOCOLOS DE COMUNICAÇÃO 145

7.2. MÓDULO BLUETOOTH 146

7.2.1. CLASSE *BLUETOOTHCOM* 147

7.2.2. CLASSE *BLUETOOTHSCANS* 147

7.3. RESULTADOS 148

8. CONCLUSÃO 155

8.1. TRABALHO REALIZADO 155

8.2. TRABALHO FUTURO 157

REFERÊNCIAS BIBLIOGRÁFICAS 159

ANEXOS 163

A1. TABELAS BLUETOOTH 164

A2. CLASSES DA APLICAÇÃO DE *SCAN* 165

A3. CLASSES DO MÓDULO BLUETOOTH 167

A4. LISTAGEM DO SERVIDOR 168

Índice de Figuras

Figura 1- Principais redes "base" de uma *Colaburative Sensing Network*. 8

Figura 2 - Cenários de aplicação das *Collaburative Sensing Networks*.....9

Figura 3 - Requisitos e desafios no desenvolvimento dos sistemas.....13

Figura 4- Diferentes Versões da Tecnologia Bluetooth. 23

Figura 5 - Esquema de dispositivos com a tecnologia Bluetooth activa.....25

Figura 6 - Banda de Frequências da Tecnologia Bluetooth. Fonte [25].....26

Figura 7 - Divisão da Banda de Frequências na Tecnologia Bluetooth. Fonte [19]. 27

Figura 8 - *Stack* da Tecnologia Bluetooth. 32

Figura 9 - Exemplo de uma conexão a um serviço usado o SDP. 35

Figura 10 - Localização geográfica dos pontos de recolha. 38

Figura 11 - Dispositivos considerados para a instalação da aplicação. 42

Figura 12 - Diagrama de Classes da Aplicação.....47

Figura 13 - Exemplo do conteúdo do ficheiro com as amostras recolhidas.....54

Figura 14 - Aspecto do menu da aplicação.....55

Figura 15 - Esquema do ambiente do teste.....58

Figura 16 - Croquis da rua Gil Vicente. 59

Figura 17 - Cenário do teste sem mobilidade.....61

Figura 18 - Fluxograma do processo de análise por local.....64

Figura 19 - Esquema do processo de análise por local - Iteração 1.....66

Figura 20- Esquema do processo de análise por local - Iteração 2. 66

Figura 21- Esquema do processo de análise por local - Iteração 3. 68

Figura 22- Esquema do processo de análise por local - Iteração 4. 68

Figura 23- Esquema do processo de análise por local - Iteração 5. 69

Figura 24 - Exemplo de sequências de detecção 1. 70

Figura 25 - Diagrama de Árvore Para Todas as Sequências de Detecção. 71

Figura 26 - Exemplo de Sequências de Detecção 2. 73

Figura 27 – Fluxograma do processo de análise global. 74

Figura 28 – Fluxograma do processo de análise global para as sequências completas. ...75

Figura 29- Fluxograma do processo de análise global para as sequências incompletas...77

Figura 30: Área de cobertura do *scan.* 115

Figura 31: Exemplos de distâncias percorridas. 118

Figura 32: Forma para determinar o ângulo θ 121

Figura 33: Aspecto do menu da aplicação de simulação. 123

Figura 34: Fluxograma para o algoritmo da sequência (1 → 2 → 3). 124

Figura 35: Fluxograma para o algoritmo da sequência (3 → 2 → 1). 125

Figura 36: Introdução dos dados na aplicação para simulação. 126

Figura 37: Situações de falha não detectáveis. 129

Figura 38: Valores da probabilidade de detecção global. 131

Figura 39: Exemplos de percursos realizados por dispositivos. 131

Figura 40 - Princípio de funcionamento da aplicação Epi. 135

Figura 41 – Nova interface gráfica do Epi. 137

Figura 42 - Digrama de Blocos da aplicação Epi. 138

Figura 43 - Resposta à função F5. 143

Figura 44 - Tabelas da base de dados do servidor. 144

Figura 45 - Formato de mensagens entre clientes Epi. 145

Figura 46 - Diagrama de Classes do Módulo Bluetooth. 146

Índice de Tabelas

Tabela 1 - Bluetooth Protocol *Stacks*. 46

Tabela 2 – Resultados do teste sem mobilidade. 81

Tabela 3 - Informações Gerais dos 3 Testes Realizados. 82

Tabela 4 - Informações Sobre os Dispositivos Detectados. 83

Tabela 5 - Análise por local – Geral. 107

Tabela 6 - Análise por local - veículos. 108

Tabela 7 - Análise por local - pessoas. 109

Tabela 8 - Análise Global - Geral. 111

Tabela 9 - Análise Global - Veículos. 112

Tabela 10 - Análise Global - Pessoas. 113

Tabela 11: Resumo dos resultados obtidos na análise por local. 128

Tabela 12 - Valores de Tempo predefinidos nos temporizadores. 139

Tabela 13 - Epi: Informações do estudo. 148

Tabela 14 - Banda de Frequências e Canais RF. Fonte [18]. 164

Tabela 15 - Protocolos e Camadas da *Stack* Bluetooth Usadas pelos Perfis. Fonte [17]. 164

Índice de Gráficos

Gráfico 1 - Frequência de detecção em ambiente sem mobilidade. 80

Gráfico 2 - Teste1: Frequências de Detecção – Geral. 83

Gráfico 3 - Teste1: Frequências de Detecção – Veículos. 84

Gráfico 4 - Teste 1: Frequências de detecção - Pessoas. 84

Gráfico 5 - Teste 1: Histograma de Frequências de Detecção – Geral. 85

Gráfico 6 - Teste 1: Histograma de Frequências de Detecção - Veículos. 85

Gráfico 7 - Teste 1: Histograma de Frequências de Detecção – Pessoas. 86

Gráfico 8 - Teste 1: Detecções Totais por Dia. 86

Gráfico 9 - Teste 1: Detecções Diferentes por Dia – Veículos. 87

Gráfico 10 - Teste1: Detecções Acumuladas por Hora Numa Semana. 88

Gráfico 11 - Teste1: Classes dos Dispositivos Detectados. 89

Gráfico 12 - Teste 1: Número de Serviços Bluetooth dos Dispositivos. 90

Gráfico 13 - Teste1: Serviços Detectados por Dispositivo. 90

Gráfico 14 - Teste 2: Gráfico de Frequências de Detecção – Geral. 91

Gráfico 15 - Teste 2: Gráfico de Frequências de detecção – Veículos. 92

Gráfico 16 - Teste 2: Gráfico de Frequências de Detecção – Pessoas. 92

Gráfico 17 - Teste 2: Histograma de Frequências – Geral. 93

Gráfico 18 - Teste 2: Histograma de Frequências – Veículos. 93

Gráfico 19 - Teste 2: Histograma de Frequências – Pessoas. 94

Gráfico 20 - Teste 2: Detecções totais por dia. 94

Gráfico 21 - Teste2: Detecções Acumuladas por Hora. 95

Gráfico 22 - Teste2: Dispositivos diferentes detectados por hora. 96

Gráfico 23 - Teste 2: Classes dos dispositivos detectados. 97

Gráfico 24 - Teste 2: Número de serviços Bluetooth dos dispositivos. 97

Gráfico 25 - Teste1: Serviços Detectados por Dispositivo. 98

Gráfico 26 - Teste 3: Gráfico de Frequências – Geral 99

Gráfico 27 - Teste 3: Gráfico de Frequências - Veículos 100

Gráfico 28 - Teste 3: Gráfico de Frequências - Pessoas 100

Gráfico 29 - Teste3: Histograma de Frequências – Geral. 101

Gráfico 30 - Teste3: Histograma de Frequências – Veículos. 101

Gráfico 31 - Teste3: Histograma de Frequências – Pessoas. 101

Gráfico 32 - Teste3: Detecções totais por dia. 102

Gráfico 33 - Teste3: Detecções acumuladas por hora numa semana. 103

Gráfico 34 - Teste 3: Classes dos Dispositivos detectados. 104

Gráfico 35 - Teste 3: Número de serviços Bluetooth dos dispositivos. 105

Gráfico 36 - Teste3: Serviços Detectados por Dispositivo. 105

Gráfico 37 - Probabilidades de detecção por local – Geral 107

Gráfico 38 - Probabilidades de detecção por local - Veículos. 108

Gráfico 39 - Probabilidades de detecção por local – Pessoas 109

Gráfico 40 - Probabilidades de detecção globais - Geral. 111

Gráfico 41 - Probabilidades de detecção globais - Veículos. 112

Gráfico 42 - Probabilidades de detecção globais - Pessoas. 113

Gráfico 43: Gráfico da função $\boldsymbol{PDetecção}\nu$. 117

Gráfico 44: Função $\boldsymbol{Pdetecção\theta}$, ν, para três ângulos diferentes. 119

Gráfico 45: Função $\boldsymbol{Pdetecção\theta}$, ν , para duas velocidades diferentes. 120

Gráfico 46: Probabilidades de Detecção Global obtidas para as diferentes simulações.127

Gráfico 47 - Epi: Gráfico de Frequências.....149

Gráfico 48 - Epi: Histograma de Frequências.....150

Gráfico 49 - Epi: Detecções Totais por Dia.....150

Gráfico 50 - Epi: Detecções Acumuladas por Hora. 151

Gráfico 51 - Epi: Classes dos Dispositivos Detectados. 152

Gráfico 52 - Epi: Número de Serviços Bluetooth. 152

Gráfico 53 - Epi: Serviços Detectados por Dispositivo. 152

Lista de Acrónimos

AP – *Adopted Protocols*

API – *Application Programming Interface*

DoS – *Denial-of-Service*

FHSS – *Frequency Hopping Spread Spectrum*

GSM – *Global System for Mobile Communications*

HCI – *Host Controller Interface*

HTTP – *Hypertext Transfer Protocol*

IEEE – *Institute of Electrical and Electronics Engineers*

IP – *Internet Protocol*

L2CAP – *Logical Link Control and Adaptation Protocol*

LAN – *Local Area Network*

MAC – *Media Access Control*

OBEX – *Object EXchange*

PDA – *Personal Digital Assistant*

PPP – *Point-to-Point Protocol*

RFCOM – *Radio Frequency Communications Protocol*

RSSI – *Received Signal Strength Indicator*

SDP – *Service Discovery Protocol*

TCP – *Transmission Control Protocol*

TCS – *Telephony Control protocol Specification*

UCLA – *University of California, Los Angeles*

UDP – *User Datagram Protocol*

UUID – *Universally Unique Identifier*

1. Introdução

Neste capítulo é feita uma pequena contextualização do tema abordado na dissertação e as razões que motivaram à sua realização. Apresenta-se os objectivos que se esperam alcançar com a sua realização. É ainda dada a conhecer a abordagem de investigação adoptada de modo a descrever como foi desenvolvido todo este trabalho. Por fim, na última secção do capítulo, descreve-se como este documento se encontra organizado, para que o leitor se consiga orientar melhor.

1.1. Contextualização do Tema e Motivação

Geralmente, quando usamos um telemóvel ou um dispositivo electrónico móvel usamo-lo para complementar as nossas necessidades de comunicação. E se fosse possível, enquanto o utilizamos, contribuir para conhecer melhor alguns dos nossos comportamentos, hábitos pessoais, sociais ou até mesmo urbanos. Por exemplo, avaliar o estado da nossa saúde e qualidade de vida, interessante não? Tudo isso pode ser feito sem grandes custos ou complicações, recorrendo às *Collaborative Sensor Networks.*

Neste tipo de redes o objectivo principal é a recolha de informações, referentes a diversas tecnologias de comunicação, que estão presentes nos diferentes dispositivos electrónicos que as pessoas utilizam. Normalmente, estas informações são usadas para estudar os diferentes hábitos de mobilidade e alguns aspectos da qualidade de vida de uma determinada população, associada a um local ou região.

Neste projecto de dissertação, é dado um forte destaque à tecnologia de comunicação Bluetooth. Esta tecnologia está presente na maior parte dos dispositivos

electrónicos que as pessoas utilizam, leva a que a mesma seja uma "forte concorrente" para ser utilizada neste tipo de redes. Neste projecto são realizados alguns estudos com esta tecnologia a fim de se perceber a sua utilização numa *Collaborative Sensing Network,* ou num sistema que tenha por base uma rede deste tipo.

O desenvolvimento deste projecto incorre sobre um tema recente, que apresenta vários desafios interessantes, abordando conteúdos referentes a algumas unidades curriculares do curso de Eng.ª de Comunicações. É um projecto de dissertação rico em conteúdo científico e com uma forte componente prática.

1.2. Objectivos Propostos

O principal objectivo a alcançar, nesta dissertação, é a construção de um módulo para a recolha de algumas informações relacionadas com a tecnologia Bluetooth. Este módulo será integrado numa outra aplicação já desenvolvida, de seu nome Epi. Esta aplicação tem como objectivo a difusão epidémica de mensagens de texto entre os seus utilizadores, onde o "módulo Bluetooth" deve actuar de forma transparente, do ponto de vista dos utilizadores.

O tema principal desta dissertação são as *Collaborative Sensing Networks* e como a tecnologia Bluetooth é utilizada para a construção do módulo, foi definido um outro objectivo. Pretende-se também realizar um estudo do comportamento da tecnologia Bluetooth, num ambiente de elevada mobilidade, por exemplo, as ruas de uma cidade, nomeadamente ruas utilizadas por pedestres e automóveis. Este estudo visa determinar a probabilidade de detecção de um dispositivo Bluetooth, numa determinada área. Esta tecnologia quando desenvolvida, não foi desenhada para ser utilizada neste tipo de cenário, portanto, este estudo é relevante pois não existem muitos estudos sobre esta matéria.

Outro objectivo é analisar, de uma forma indirecta, como as pessoas (que transportam e/ou utilizam os seus dispositivos) se distribuem geograficamente e temporalmente num determinado local.

A metodologia utilizada, para alcançar os objectivos propostos, é a análise de algumas das soluções já desenvolvidas, relacionadas com o tema, de forma a encontrar uma solução adequada para o problema em mãos.

Resumindo, no final deste projecto de dissertação pretende-se então alcançar os seguintes objectivos:

- Criação de um módulo, para recolha de informação relacionada com a tecnologia Bluetooth;
- Analisar como as pessoas se distribuem geograficamente e temporalmente num determinado local, tendo em conta a informação recolhida;
- Calcular a probabilidade de detecção de um dispositivo Bluetooth, num ambiente de elevada mobilidade.

1.3. Abordagem de Investigação

Para atingir os objectivos descritos foi necessário realizar todo um conjunto de tarefas para a solucionar o problema em mãos. Nesta secção descreve-se resumidamente, como foi realizado o desenvolvimento de todo este projecto de dissertação. Esta descrição é feita pela ordem que as tarefas foram executadas.

Como já foi referido anteriormente as *Collaborative Sensing Networks* "operam" recorrendo ao uso de uma ou várias tecnologias de comunicação, como é o caso da tecnologia Bluetooth. Para se conhecer melhor este tipo de redes foi realizada uma pequena investigação acerca das mesmas. Esta investigação permitiu conhecer melhor

os cenários de implementação e requisitos necessários ao desenvolvimento um sistema para este tipo de redes.

Após a realização da investigação o próximo passo foi compreender o funcionamento da tecnologia Bluetooth, e detectar que informações se poderiam recolher, de modo a estudar a mobilidade das pessoas numa determinada zona geográfica.

Depois de examinados alguns artigos relativos á tecnologia verificou-se que, havia pouca informação sobre o comportamento da tecnologia Bluetooth em ambientes exteriores, nomeadamente em espaços de elevada mobilidade. Definiu-se outro objectivo: determinar a probabilidade de detecção de um dispositivo num ambiente de elevada mobilidade.

Agora precisava-se de encontrar um local para se realizar um estudo relativo á mobilidade das pessoas, este local deveria oferecer condições de elevada mobilidade. E concluiu-se que o melhor cenário era uma rua da cidade.

Identificada a informação a recolher e o local para realização do estudo, era necessário o desenvolvimento de uma solução que proporcionasse a recolha das informações. A solução encontrada foi o desenvolvimento de uma aplicação que realizasse o *scan* de dispositivos Bluetooth que se encontram num local e armazenasse as informações recolhidas.

Desenvolvida a aplicação, deu-se início a um conjunto de “testes de campo” no cenário escolhido. Paralelamente a estes testes, optou-se por realizar um outro teste, num ambiente interior e sem mobilidade, ambiente este onde o uso da tecnologia Bluetooth é mais comum. A realização deste teste era importante porque poderiam, mais tarde, ser comparados os resultados para os dois diferentes cenários de teste.

Depois de analisada a informação recolhida de todos os testes, ficou-se a conhecer alguns parâmetros da mobilidade das pessoas e veículos, para aquela rua. Utilizando as informações recolhidas pela aplicação de *scan*, foi ainda possível determinar a probabilidade de detecção dos dispositivos detectados.

O trabalho desenvolvido até este ponto ajudou à concretização de um outro objectivo, a integração de um módulo de *scan* Bluetooth, na aplicação Epi.

A aplicação Epi faz de parte um sistema de recolha de informação, relativo à mobilidade das pessoas. Este sistema já se encontrava em desenvolvimento no grupo de sistemas ubíquos da Universidade do Minho (Ubicomp), no âmbito do projecto SUM (*Sensing and Understanding human Motion dynanics*). Esta aplicação, quando instalada permite aos seus utilizadores, enviar mensagens de texto, de forma epidémica. No capítulo 7, a aplicação Epi é explicada em detalhadamente.

Realizadas algumas alterações na aplicação de *scan* desenvolvida, esta pode ser integrada como um módulo na aplicação Epi. Assim, paralelamente ao envio de mensagens por parte dos utilizadores, é recolhida informação acerca dos dispositivos Bluetooth que se encontram na proximidade do utilizador.

Posteriormente, as informações recolhidas são enviadas para o servidor do sistema. Fizeram-se algumas alterações no servidor do sistema, pois ele não foi desenhado para receber informações relativas aos dispositivos Bluetooth.

1.4. Organização e Estrutura do Documento

O documento está organizado em oito capítulos. A ordem dos capítulos é a ordem da realização das tarefas. Cada capítulo está dividido em diferentes secções, para que seja mais fácil ao leitor identificar e consultar.

Neste capítulo 1, é feita uma introdução, de modo a dar a conhecer, de uma forma sucinta, o tema que a dissertação aborda, os objectivos que se prevêem alcançar, a abordagem de investigação adoptada e ainda como o presente documento está organizado.

O segundo capítulo apresenta uma pequena abordagem relativa ao funcionamento das *Collaborative Sensing Networks*, os requisitos a ter em conta no desenvolvimento deste tipo de redes de informação e alguns projectos em desenvolvimento.

No terceiro capítulo é feita uma descrição geral da tecnologia Bluetooth, indicando alguns aspectos fundamentais, para a especificação de uma solução que se adeqúe às necessidades do projecto em mãos.

O quarto capítulo apresenta e explica a solução encontrada para a implementação deste projecto. Apresenta-se as diferentes opções consideradas para a instalação da solução e a arquitectura da mesma.

No capítulo 5, é descrita a implementação da solução e os testes realizados. É descrito também o ambiente de teste mais aprofundadamente e ainda são explicados os processos de análise desenvolvidos, para determinar as probabilidades de detecção dos dispositivos.

Já o capítulo 6 apresenta os resultados obtidos na realização dos testes. É feita uma interpretação e análise individual dos resultados obtidos, para cada um dos testes realizados. Aqui são ainda dados a conhecer os valores obtidos para a probabilidade de detecção dos dispositivos.

De seguida, no capítulo 7, é descrito a arquitectura do módulo desenvolvido e a sua inserção na aplicação Epi. Apresentam-se as novas funcionalidades criadas para a aplicação e ainda os resultados obtidos resultantes da inserção módulo na aplicação Epi.

Por fim, o oitavo capítulo indica as conclusões e sugestões para complementar o trabalho no futuro. □

2. Collaborative Sensing Networks

Antes de clarificar a solução encontrada para os desafios exigidos por este tema dissertação, convém compreender o que são as *Collaborative Sensing Networks*.

O cerne, a essência principal, deste tipo de redes é a partilha de informação, para determinados cenários de estudo, feita de forma livre e espontânea por parte de quem a disponibiliza ou de quem colabora.

Uma *Collaborative Sensing Network* ou *Collaborative Sensor Network* tem a "missão" de, num futuro próximo, fazer uma espécie de "ponte" entre um mundo virtual e o mundo real [5].

Como é referido em [6], existem centenas de milhões de automóveis (podendo estes estar equipados com dispositivos de comunicação) e mais de um milhar de milhão de pessoas que possuem dispositivos de comunicação, carros e seres humanos podem vir a fazer parte da maior e mais dinâmica rede de sensores em todo o mundo, já nos próximos anos.

Este tipo de redes tem por objectivo funcionar como meio de recolha de informação, acerca de algo que nos rodeia. Elas "escutam" determinado parâmetro ou grandeza física, de um ambiente onde "estamos inseridos" ou frequentámos. Todo este processo de recolha é denominado de "*sensing*".

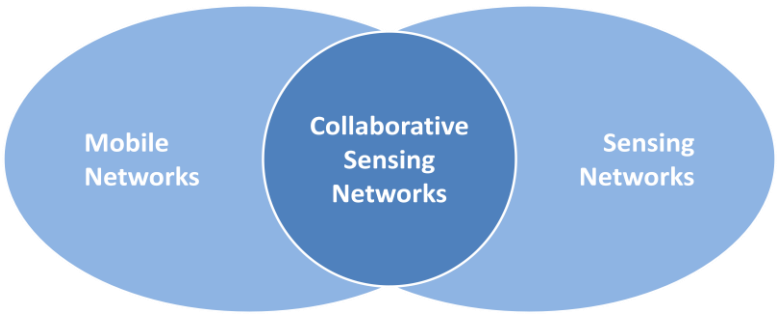


Figura 1- Principais redes "base" de uma *Colaburative Sensing Network*.

O que distingue estas redes das normais redes de sensores é que, o "*sensing*" não se encontra limitado a determinado espaço ou área. As actuais redes móveis possuem aqui um papel importante, dado que providenciam a mobilidade das diferentes fontes que disponibilizam a informação, podendo estas cobrir diferentes áreas geográficas em instantes de tempo diferentes. Deste modo, é monitorizada uma área geográfica muito maior, com menos fontes de informação (sensores) do que numa normal rede sensores (Figura 1).

2.1. Cenários de Aplicação

Nos dias de hoje, é grande o número e tipo de dispositivos computacionais móveis ou portáteis que são usados em ambiente pessoal, social e urbano (Figura 2). Para além dos tradicionais telemóveis, a variedade de dispositivos inclui PDAs, câmaras fotográficas, sistemas de navegação, leitores de música, auriculares, etc..

Uma vez que a mobilidade destes dispositivos está fortemente associada à mobilidade dos seus utilizadores (donos) durante as suas actividades diárias, a obtenção de dados sobre a mobilidade destes dispositivos permite obter indirectamente determinado tipo de informações úteis acerca dos seus utilizadores ou do ambiente em redor dos mesmos. Sabendo agora que alguns destes dispositivos integram ainda um conjunto de sensores (que permitem obter informação acerca da temperatura, movimento, luminosidade, ritmo cardíaco, humidade, etc.), a quantidade de informação

que pode ser recolhida permite abrir a porta a vários "cenários" de estudo ao nível pessoal, social e urbano. Estas informações, depois de analisadas e processadas, podem ser disponibilizadas a todo um conjunto de pessoas que as pretendam consultar.

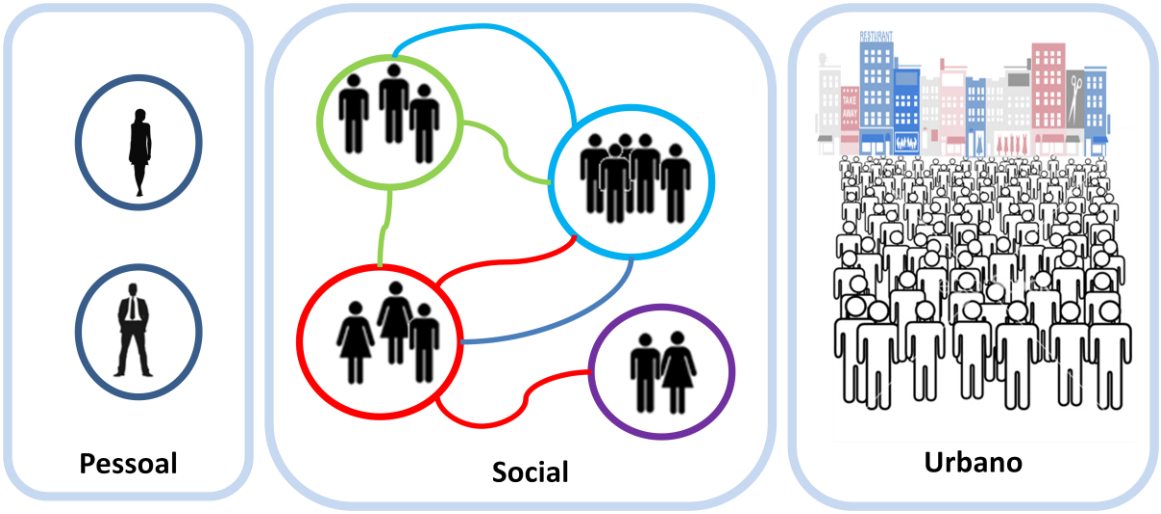


Figura 2 - Cenários de aplicação das *Collaburative Sensing Networks*.

Tendo em conta os três "cenários" ou categorias de estudo apresentados, podem ser desenvolvidos vários tipos de aplicações. Ao nível pessoal, as aplicações mais interessantes a serem desenvolvidas são, talvez, as aplicações de monitorização médica. Com este tipo de aplicações, facilmente se poderia observar o "espaço pessoal" de determinado paciente. Como é referido em [1], poderíamos obter informação acerca dos valores de açúcar no sangue ou ritmo cardíaco, que depois poderiam ser monitorizados pelo próprio paciente ou por outra pessoa encarregue do seu cuidado. Um outro exemplo, que se enquadra nesta categoria, é também o controlo e monitorização de uma "casa inteligente", onde o dono da mesma poderia obter várias informações relevantes acerca da sua habitação.

Se nos focarmos agora ao nível social, também existem aplicações interessantes a desenvolver. Neste cenário, a informação recolhida é partilhada apenas por um determinado conjunto de pessoas ou amigos, possivelmente por todos aqueles que contribuem com parte da informação. Aqui, poderíamos incluir aplicações de partilha de

fotografias, aplicações de partilha de um itinerário ou percurso, pequenas aplicações de partilha de áudio/vídeo.

Por fim, pensando num nível urbano, categoria onde o tema desta dissertação está mais próximo, a informação e os dados obtidos implicam a partilha dos mesmos com todo o público em geral ou com uma entidade centralizada. Uma aplicação interessante seria a integração destas redes, por exemplo, com uma aplicação semelhante à do *Google StreeView* [2]. Em vez de imagens ou informação acerca de determinado local serem actualizados em intervalos de tempo longos, por profissionais, poderiam ser actualizadas diariamente através de informação "produzida" por um telemóvel de um morador ou visitante no local. Um outro exemplo seria, numa aplicação parecida com a anterior, se esta mostrasse o valor da temperatura para uma determinada zona de determinada cidade, indicasse os níveis de poluição do ar e sonora, entre outros. Outra solução seria, criar uma aplicação que permitisse avaliar o comportamento do tráfego automóvel e pedestre para determinada data ou local. Neste nível, também poderiam ser englobadas algumas aplicações destinadas aos outros dois cenários, no entanto, neste caso, seria feita uma análise global dos resultados produzidos por essas aplicações.

2.2. Desenvolvimento de Sistemas

Podemos consideras as *Collaborative sensing* ou *Collaborative Sensor Networks*, como um conjunto de tarefas a realizar, como refere Nicholas et al. em [2]. As tarefas são as seguintes:

- Formulação da questão: o sistema, dependendo do tipo de informação que necessita de recolher, questiona os dispositivos para o envio de dados.
- Recolha da informação: após questionados, os dispositivos enviam a informação para o sistema, onde o mesmo a deve armazenar.
- Análise da informação: com a informação recolhida o sistema, recorrendo a um algoritmo de análise dos dados, fórmula os resultados.

- Partilha e apresentação dos resultados: tendo em conta os resultados obtidos, os mesmos tem de ser apresentados, de uma forma apelativa, para os utilizadores do sistema.

Outro aspecto a reflectir, no desenvolvimento destes sistemas, refere-se à forma como a informação pode ser recolhida. Assim sendo, temos basicamente duas opções: uma é a recolha de dados de forma participativa, a outra é a recolha de dados de forma oportunística.

Na primeira opção, as pessoas que possuem os dispositivos de "*sensing" (*por exemplo o seu próprio telemóvel), optam pelo envio de determinado tipo de informação útil para o sistema, sem compromissos ou interesses pessoais ou financeiros. Os sistemas que optem por este tipo de abordagem têm de se concentrar na construção de ferramentas e mecanismos que permitam às pessoas partilhar, publicar, investigar e consultar as informações recolhidas bem como os resultados formulados pelo sistema, de forma a incentivar as pessoas a participarem no projecto em causa. As pessoas, ao partilharem a informação, fazem-no de forma consciente esperando no final obter algo em troca.

Relativamente à segunda opção, com a recolha de informação oportunística, as pessoas não estão conscientes dos sistemas que se encontram a recolher informação dos seus dispositivos de comunicação. Neste caso, não interessa o estado ou o tipo do dispositivo utilizado pelas pessoas, o que interessa é que a informação que seja recolhida seja útil, ou preencha alguns dos requisitos do cenário de estudo em causa. Podendo até a tarefa da formulação da questão aos dispositivos não ser realizada, uma vez que os mesmos estão alheios à recolha de informação. O desafio principal desta abordagem é saber se a informação obtida de determinado dispositivo, comunga com os requisitos da aplicação desenvolvida.

Se com a recolha de dados participativa o tipo de informação obtida pode ser mais ampla e mais objectiva, também é necessário o desenho de uma aplicação mais

complexa e apelativa, ou até mesmo a utilização de dispositivos adaptados para estes sistemas (quer a nível de software ou hardware), para uma maior colaboração das pessoas ou utilizadores no envio da informação.

Por outro lado, com a recolha oportunística, as aplicações de envio ou recolha de dados não necessitam de ser tão apelativas e complexas, podendo até nem serem necessárias. Como consequência, o universo de dispositivos para recolha de informação pode ser maior, no entanto a informação recolhida nem sempre é útil ou é insuficiente para o propósito do sistema em causa.

Depois disto, facilmente se chega à conclusão que, para se optar por uma abordagem ou outra, devemos ter em conta o propósito da aplicação desenvolvida e para que cenário ou categoria de estudo (pessoal, social ou urbano) se destina.

Como é referido em [2], o modelo participativo deve ser utilizado quando existe um conjunto de pessoas interessadas nos resultados a ser formulados por determinado sistema. O número desse conjunto de pessoas deve ser maior ou igual ao número de dispositivos de recolha de informação.

Já o modelo oportunístico é o ideal para casos onde as aplicações dos sistemas não necessitem de uma intervenção directa por parte das pessoas na recolha de informação e onde a recolha possa ser feita de forma automática pelo sistema.

Como exemplo para o modelo participativo podem ser referidos os projectos levados a cabo pela universidade de *Los Angels* (UCLA)[10] e para o modelo oportunístico o projecto CarTel [6]. Ambos os projectos são apresentados mais à frente.

2.2.1. Requisitos e Desafios para os Sistemas

No desenvolvimento de um sistema que opere sobre uma *Collaborative Sensing Network*, existe uma série de requisitos funcionais que devem ser abordados e levados

em conta (Figura 3). Requisitos como a transparência de dados, a heterogeneidade, a escalabilidade, a privacidade e segurança, entre outros, são muito importantes e devem ser bem analisados quando se desenvolve um sistema ou aplicação que tenha como base este tipo de redes [5]. No entanto, garantir alguns destes requisitos pode tornar-se num verdadeiro desafio.

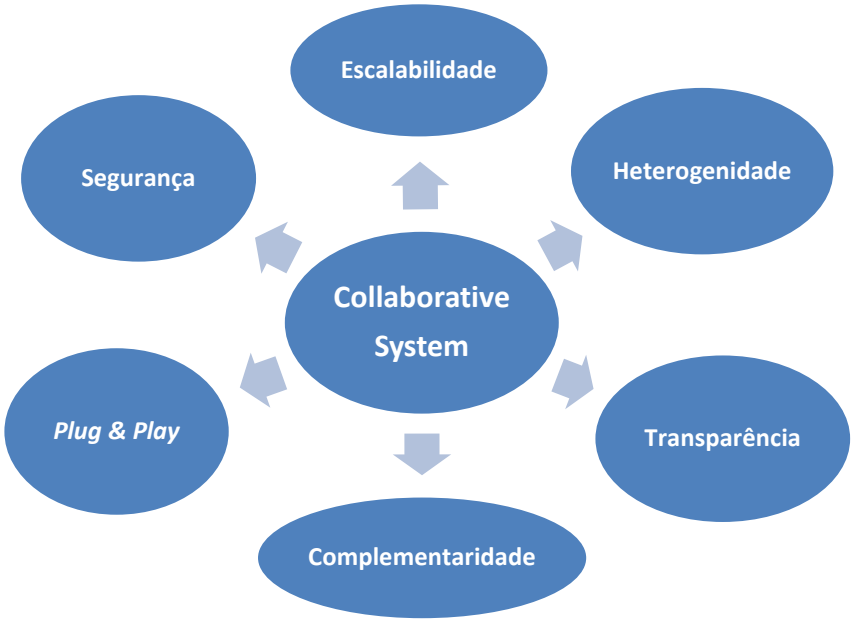


Figura 3 - Requisitos e desafios no desenvolvimento dos sistemas.

Um sistema deste tipo deve ser escalável, isto é, se o número de dispositivos aumentar, com o sistema em funcionamento, o mesmo deve conseguir efectuar a recolha de dados de todos eles. Ou então, se os utilizadores pretenderem consultar os resultados formulados pelo sistema, o mesmo deve garantir um funcionamento efectivo. Esta é uma característica indispensável para uma abordagem participativa na recolha de informação dos dispositivos.

É esperado que o sistema também possa dar garantias de alguma autonomia. A heterogeneidade está, de certo modo, ligada á autonomia destes sistemas. Nem todos os dispositivos ou fontes de informação operam da mesma forma ou tem características

de funcionamento iguais. Nestes sistemas, deverão existir garantias que mesmo os dispositivos não tão recentes ainda possam aceder aos serviços disponibilizados.

Este requisito deve ser tido mais em conta, quanto maior for o "*target*" de dispositivos para que o sistema foi desenvolvido. Se um sistema pretende recolher informações ou disponibilizar a consulta de informação de diversos dispositivos (telemóveis, computadores portáteis, etc.) aos seus utilizadores, devem dar mais garantias de Heterogeneidade. Desta forma os serviços disponibilizados pelo sistema têm de estar sempre disponíveis, independentemente do hardware/fabricante dos dispositivos, dos seus sistemas operativos e ainda das redes ou protocolos de transporte a que os mesmos estão ligados com o servidor do sistema.

A questão da transparência em termos de desempenho e no acesso à informação a recolher é outro desafio, a acrescentar a este tipo de sistemas. As funcionalidades dos sistemas devem ser apresentadas aos utilizadores, de uma forma global, através de uma aplicação com uma interface de interacção apelativa, sem grande complexidade de utilização. Baixar a complexidade de utilização pode ajudar a cativar e aumentar o número de utilizadores do sistema, e consecutivamente a quantidade de informação recolhida pelo sistema ser maior.

No desenvolvimento dos sistemas um requisito do tipo "*Plug & Play*" seria também bastante desejável. O processo de tornar a informação recolhida disponível para outros, deve ser o mais simples possível. Os donos dos dispositivos de onde está a ser recolhida a informação, devem obedecer ao mínimo possível de regras ou políticas de troca de informação impostas pelo sistema ou aplicação do mesmo [5].

A complementaridade com serviços já existentes, em outro tipo de sistemas, é outra funcionalidade a ter em conta. Por exemplo, as operadoras de telemóveis poderiam desenvolver um sistema de "*sensing"* usando a sua arquitectura de

funcionamento e infra-estruturas para providenciar a recolha de informações. Desta forma, poderiam também fornecer um novo leque de serviços aos seus utilizadores.

Outro desafio importante a considerar é o desenho de novos modelos de tráfego de dados para estas redes [5]. Dado que estas novas aplicações e serviços a desenvolver levam a um aumento dos utilizadores na rede, novos requisitos podem ser exigidos às redes que suportam o transporte destes dados, nomeadamente para as redes móveis ou redes *wireless*.

Já foram apresentados os requisitos, que devem ser levados em conta no desenvolvimento de sistemas para uma *Collaborative Sensing Networks*. Contudo, existe um requisito que apresenta o maior dos desafios para estas redes a segurança deste tipo de sistemas. Como se trata de um assunto mais complexo, será discutido em seguida com maior detalhe.

2.2.2. Segurança dos Sistemas

Ao contrário das normais redes de sensores, as *Collaborative Sensing Networks* estão fortemente associadas à mobilidade dos dispositivos e, como já foi referido, as actuais redes móveis podem servir de "suporte" a estas redes. Sendo assim, alguns dos problemas de segurança aqui presentes assemelham-se bastante aos das redes móveis actuais.

Como é referido em [8], sistemas que não tentam levar em conta as questões da segurança e privacidade são inadequados para o propósito destas redes e não estariam em funcionamento muito tempo.

Dado que as fontes de informação estão na posse de "entidades estranhas" ao sistema, torna-se necessário garantir a integridade da informação recolhida. Também

para as fontes de informação (dispositivos das pessoas), é fundamental a garantia da preservação da sua privacidade a quando do envio dos dados para o sistema. Soluções que envolvam privacidade, devem ser desenvolvidas tendo sempre em conta a integridade dos dados [8].

Por exemplo, não deve ser possível para os gestores e utilizadores da aplicação, obter informações acerca das fontes (dispositivos) de onde o sistema recolheu informação. Uma solução para evitar este problema seria, por exemplo, ser possível ao sistema identificar apenas o *access point* (AP) de onde a informação é proveniente, e não o dispositivo pessoal que enviou a informação para o mesmo *access point* [8].

Devido à constante mobilidade dos dispositivos (sensores), a topologia da rede está constantemente a mudar, algo que não acontece em outros tipos de redes, podendo facilitar assim a introdução de fontes de informação não confiáveis, gerando um problema de segurança preocupante.

Imagine-se um sistema participativo de recolha de informação, onde um dispositivo esteja a submeter informação para o sistema. Essa informação poderia facilmente ser capturada e alterada, por outro dispositivo e só depois enviada para o sistema, sem que este último detecte que informação que a disponibilizou foi alterada (ataque "*man-in-the-middle*").

Como são as pessoas que têm em sua posse as fontes de informação do sistema, ainda é mais "apetecível" para algum atacante realizar tal ataque.

As redes de comunicação tradicionais poderiam recorrer a protocolos de *routing* seguros para tentar contornar o problema anterior, mas nas *Collaborative Sensing Networks*, o mesmo já não acontece. O ideal aqui, para o sistema, é tentar garantir a credibilidade da informação recebida, em vez de se concentrar na segurança da rota por onde passa a informação.

A solução para este problema talvez passe pelo uso de redundância [8]. O sistema poderia recorrer a algoritmos estatísticos para estimar a validade da informação recebida.

Um outro aspecto interessante a reflectir é a forma como este tipo de redes deve responder, perante um ataque de DoS (*Denial of Service* – negação de serviço). Como estamos perante uma rede de elevada mobilidade dos seus nós, um atacante que pretenda realizar um ataque de DoS tem, de certa forma, a tarefa facilitada, pois o sistema não tem qualquer controlo sobre os dispositivos.

Prefigure-se um sistema que realiza a recolha de informação de forma oportunística. Se um atacante, noutro tipo de redes, realizar um ataque de DoS poderá enviar grandes quantidades informação para o sistema, de forma a degradar o desempenho do mesmo. Neste caso a estratégia de ataque pode até ser o oposto, ou seja, um ataque de DoS poderia ser conseguido impedindo os dispositivos de enviarem informação, ou de responderem a solicitações do sistema. Isso levaria o sistema a não produzir resultados correctos ou a produzir resultados de baixa precisão, levando os utilizadores a deixar de utilizar os serviços do sistema.

Como se pode agora concluir, um sistema a operar numa *Collaborative sensor Networks*, tem pela frente vários desafios em questões de segurança, sendo que os principais são: privacidade; autenticidade dos dados; leituras de fontes de informação confiáveis; integridade do sistema.

Se o sistema conseguir dar uma resposta aceitável à maior parte destas questões, incentivará mais pessoas a colaborarem com os seus dispositivos na recolha de informações e aumentará o número de utilizadores do sistema de "*sensing"*.

2.3. Projectos em Desenvolvimento

Actualmente, redes com este tipo de características estão a ser utilizadas e desenvolvidas ao nível académico, em pequenos projectos de algumas organizações /empresas ou então servem pequenos grupos sociais. Apesar de economicamente não apresentarem grandes lucros, espera-se que venham a ter a adesão das grandes empresas.

Existem já alguns projectos que começam a dar os primeiros passos nesta área. Em seguida são dados a conhecer alguns deles.

2.3.1. *Urban Sensing* - UCLA

Nos Estados Unidos, a Universidade da Califórnia em *Los Angels* (UCLA) tem vindo a implementar uma série de projectos na área do *Urban sensing* [10]. Alguns dos projectos aqui em desenvolvimento são: *Cyclesence* (ajudar os ciclistas a encontrarem bons percursos na cidade e ajudar a melhorar os mesmos); *Dietsense* (aquisição e análise de dados acerca das escolhas nutricionais das pessoas); *Family Dynamics* (análise da dinâmica física das famílias); *Networked Naturalist* (realizar recolha de dados para investigação ecológica); *Garbage Watch* (coleccionar imagens de lixo de determinados locais para determinar onde devem ser colocados novos ecopontos), etc.

2.3.2. CarTel

Um outro exemplo é o projecto de nome CarTel, que tem vindo a ser desenvolvido pelo MIT [11]. Neste projecto o objectivo é recolher, processar, disponibilizar e visualizar a informação recebida por parte de dispositivos móveis colocados em vários automóveis, nas cidades de Boston e Seattle nos Estados Unidos da América (EUA). Essa informação recebida é encaminhada para um servidor central onde, depois de processada, pode ser obtida informação acerca do tráfego automóvel (atraso

e filas de trânsito, condição do pavimento das estradas), monitorização de redes *Wi-Fi* urbanas, análise sobre hábitos de condução.

2.3.3. MetroSense

MetroSense (*Secure People – Centric Sensing at Scale*) é o nome de um projecto global que, em conjunto com vários outros projectos, se dedica a desenvolver novas aplicações para integrar os novos dispositivos móveis, tendo em conta os requisitos e as características exigidas para este tipo de redes [9]. As empresas e organizações envolvidas no projecto são a NOKIA, a INTEL, a *National Science Foundation* (NSF), a MOTOROLA e ainda o instituto para a Segurança, Tecnologia e Sociedade (*The Institute for Security, Technology, and Society* - ISTS) do colégio de *Dartmouth* nos EUA.

2.3.4. Around Knowledge

A *Around Knowledge* é uma empresa portuguesa, recentemente criada, que desenvolve soluções para o estudo do seguimento/localização de pessoas, aplicações móveis, Bluetooth marketing, etc.

Esta empresa desenvolveu recentemente o projecto BIPS, que tem como objectivo a realização do "seguimento" de pessoas, em determinadas áreas geográficas ou dentro de espaços fechados, mais propriamente centros comerciais. Este projecto é útil para os donos de espaços comerciais que pretendam saber, por exemplo, os percursos que as pessoas mais frequentam bem como as lojas mais visitadas ou ainda o tempo de espera em determinadas filas [26]. Esta solução permite economizar até 89% do preço actual deste tipo de estudos.

As informações recolhidas para análise da mobilidade das pessoas, são recolhidas por intermédio das ondas electromagnéticas que os dispositivos móveis, que as pessoas transportam consigo, transmitem. Todo este processo é feito garantindo o anonimato das pessoas que transportam os dispositivos, não sendo nunca possível identificar a

pessoa A, B ou C, que esteve no local X, Y ou Z. Este processo é semelhante ao que é aplicado no desenvolvimento deste projecto de dissertação, como é explicado nos próximos capítulos. As principais tecnologias utilizadas para leitura dos sinais de rádio são: Bluetooth, GSM e WiFi.

Esta empresa tem vindo a ser agraciada com alguns prémios, onde se destaca o primeiro lugar obtido no *MIT ISCTE-IUL 200k Venture Competition*, com o projecto referido anteriormente, o BIPS. Para mais informações, acerca desta empresa, pode ser consultada a referência [12]. □

3. Tecnologia de Comunicação Bluetooth

Antes de se avançar para a especificação de uma solução concreta para o problema em mãos, é necessário entender como funciona a tecnologia de comunicação Bluetooth.

No presente capítulo, é explicada a tecnologia de uma forma geral e são esclarecidos alguns aspectos relativos à mesma, fundamentais para se encontrar a solução correcta. Alguns conceitos referidos neste capítulo são necessários para a compreensão da solução encontrada, e serão explicados nos capítulos seguintes.

3.1. Introdução

Uma característica comum à maioria dos dispositivos utilizados na recolha de informação numa *Collaborative Sensing Network* é disporem de uma interface de comunicação baseada na tecnologia Bluetooth. Este é também o caso de alguns dispositivos que equipam os automóveis mais recentes. O facto de se ter escolhido esta tecnologia está relacionado com um conjunto de vantagens que a mesma apresenta, em relação a outras tecnologias de comunicação sem fios e que fazem a mesma ser tão popular nos dias de hoje.

Em primeiro lugar, trata-se de uma tecnologia que comunica por meio de ondas de radiofrequência, resistindo razoavelmente às interferências do meio de comunicação.

Não necessita que os dispositivos envolvidos na rede estejam em linha de vista, podendo ainda ser integrada essa mesma rede por vários dispositivos multiponto (tipicamente um máximo de 7). Os vários protocolos da tecnologia Bluetooth foram construídos de forma a suportar comunicação de voz e de dados, podendo também integrar facilmente outros protocolos, como por exemplo o TCP. A inserção de novos dispositivos na rede é feita de forma pouco complexa e automática. Os micro-chips ou o hardware que implementam as interfaces são de baixo consumo e custo.

Apesar de esta tecnologia apresentar todas estas vantagens, também apresenta alguns problemas. A velocidade de transferência de dados não é muito elevada, quando comparada com outras tecnologias de comunicação sem fios, como é o caso da tecnologia 802.11b. Possui um curto alcance, no entanto a tecnologia tem vindo a sofrer algumas alterações e o alcance e as taxas de transferências aumentado ligeiramente.

Outros dois aspectos relevantes são: os fracos mecanismos de segurança que a tecnologia apresenta na suas primeiras versões, e ainda o número limitado de dispositivos que se podem conectar na rede ao mesmo tempo.

Esta tecnologia foi criada e dada a conhecer pela empresa *Ericsson,* no ano de 1994. A tecnologia encontra-se fortemente "ligada" à Bluetooth SIG (*Special Interest Group*) [14], que é uma organização sem fins lucrativos, iniciada por cinco companhias, que se encarrega de administrar todo o que se encontra relacionado com o desenvolvimento da tecnologia Bluetooth, nomeadamente, a gestão e actualizações de protocolos e normas, bem como o anúncio do lançamento de novas versões. Actualmente esta organização já conta com mais de 14.000 companhias como membros, que ajudam a desenvolver e promover a tecnologia, em que algumas delas fabricam dispositivos/hardware que recorrem à mesma. O *Institute of Electrical and Electronics Engineers* (IEEE) refere a tecnologia no standard 802.15 (standard relativo às "*Personal Area Networks*" – PAN), com a "referência" IEEE 802.15.1.

3.1.1. Evolução da Tecnologia Bluetooth

A tecnologia Bluetooth tem vindo a ser desenvolvida basicamente ao longo da última década e começou a fazer notar-se mais após o lançamento da sua versão 1.0 no início de 1999, como se pode observar pela Figura 4.

Até à versão 1.1, e dado esta ser uma tecnologia recente, apresentava muitos problemas na conexão entre dispositivos, e os fabricantes dos mesmos tinham dificuldades em solucioná-los. Até esta versão a tecnologia frequentemente ia sofrendo alterações na sua arquitectura e desenho.

A partir da versão 1.2, lançada nos finais de 2003, já se havia corrigido a maior parte dos bugs existentes nas versões anteriores e começaram a surgir no mercado vários dispositivos que disponibilizavam esta tecnologia. As taxas de transmissão foram aumentadas para cerca de 1Mbit/s (efectivamente <800kbps), foi melhorada a interface de controlo HCI (*Host Controller Interface*) e ainda melhorados os mecanismos de conexão e descoberta de dispositivos.

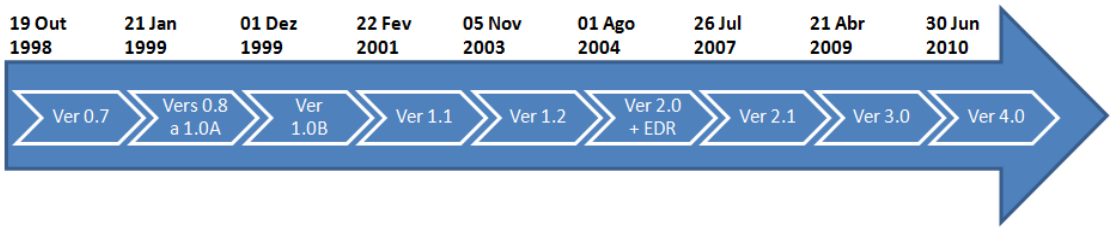


Figura 4- Diferentes Versões da Tecnologia Bluetooth.

Em Agosto de 2004, foi lançada a versão 2.0 + EDR (*Enhanced Data Rate)*. Aqui as principais alterações foram o aumento das taxas de transmissão, para valores na ordem dos 3Mbit/s, contudo na prática os valores rondavam os 2.1Mbit/s. Este aumento foi conseguido devido a melhoramentos na técnica de modulação dos sinais.

Cerca de três anos depois era disponibilizada a versão 2.1, tendo com principal melhoria o processo de emparelhamento de dispositivos (*pairing*), e também o aumento da segurança nas conexões. Nesta versão foi ainda adicionado "outro aperfeiçoamento" o EIR (*Extended Inquiry Response*) que viria a complementar o processo de descoberta de dispositivos, possibilitando a recolha de mais informações acerca do dispositivo descoberto, antes de se estabelecer uma conexão. Desde esta versão os fabricantes dos dispositivos poderiam decidir que perfis suportava o dispositivo de acordo com a(s) aplicações que os mesmos pretendiam desenvolver.

Nos primeiros meses de 2009 ficou disponível a versão 3.0 ou 3.0+HS (*High Speed*). Agora a tecnologia apresentava taxas de transferência bastante superiores às das versões anteriores. Desta vez é possível alcançar as taxas de transmissão entre os 24Mbit/s e os 54Mbit/s, garantindo um a taxa mínima de 11MBits/s. Isto é possível porque foi introduzida uma camada de adaptação para a especificação IEEE 802.11, ou seja para as redes WiFi, aumentando-se assim as taxas de transmissão. Aqui também foi operacionalizada a retro-compatibilidade com versões anteriores. Caso o dispositivo não possua placa de rede WiFi, para se fazer uso da nova camada de adaptação, serão usadas as taxas de transmissão anteriores. Foi ainda realizado outro aperfeiçoamento, a inclusão do EPC (*Enhanced Power Control*), responsável por diminuir interferências e interrupções nas comunicações.

Por fim, a versão mais recente da tecnologia é a versão 4.0, lançada a 30 de Junho de 2010. Esta versão também se designa de Bluetooth LE (*Low Energie)*, pois foi optimizado o consumo de energia, que até aqui não tinha sofrido melhoramentos significativos. Com esta redução no consumo de energia, abre as portas para que um maior número de dispositivos, que consomem pouca energia, também possa usufruir da tecnologia. Nesta versão as características para as taxas de transferência e largura de banda mantêm-se as mesmas que na versão anterior. Foi contudo melhorado o raio de alcance, podendo atingir agora uma média de 60 metros.

3.2. Processo de Descoberta de Dispositivos

Na Figura 5 estão representados um conjunto de cinco dispositivos e todos eles possuem a tecnologia Bluetooth integrada no seu hardware. As circunferências em redor de cada dispositivo representam o alcance das ondas de rádio irradiadas pelo dispositivo (todos eles encontram-se configurados, com o seu "modo de descoberta"[1] activo). O alcance de cada dispositivo é variável, dependendo da classe de potência do dispositivo Bluetooth. Actualmente estão definidas 3 classes de potência (classe 1 a classe 3) cujos alcances são de aproximadamente 100, 10 e 1 metros, respectivamente. Quanto maior a sua potência, mais energia o dispositivo consome.

No dispositivo A não se encontra representada circunferência, pois ele está a realizar o *scan*/varrimento dos dispositivos que se encontram na sua área de cobertura.

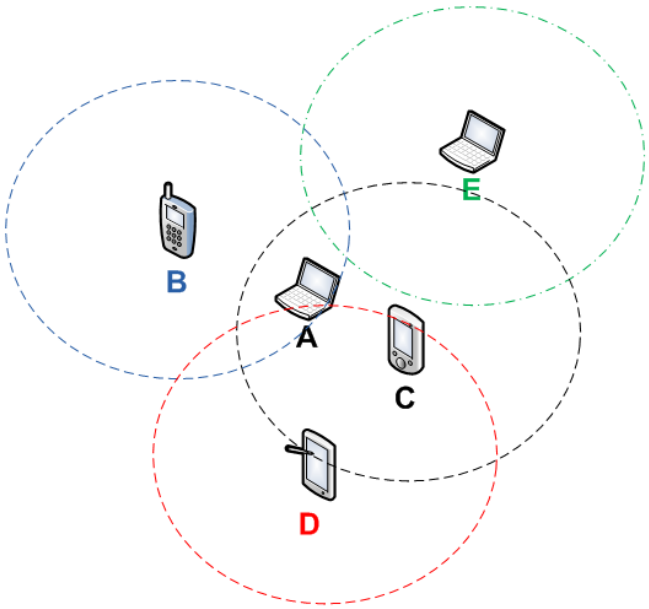


Figura 5 - Esquema de dispositivos com a tecnologia Bluetooth activa.

Do que está representado na figura, facilmente se percebe que o dispositivo A apenas conseguirá detectar os dispositivos B, C e D e que, o dispositivo E não tem uma potência de sinal suficiente para ser detectado por A. A seguir descreve-se como funciona todo este processo de descoberta.

[1] O "modo de descoberta" é um parâmetro, configurável no dispositivo Bluetooth, que indica se o mesmo responde ou não a tentativas de descoberta, por parte de outro dispositivo Bluetooth.

A tecnologia Bluetooth funciona por intermédio de ondas de rádio (ou seja, os dispositivos não necessitam de se encontrar em "linha de vista" para comunicarem), na banda de frequências entre os 2.402Ghz e os 2.480Ghz (Figura 6). Sendo assim, cada dispositivo funciona e realiza todas as suas comunicações para troca de "mensagens" com uma determinada frequência, dentro desta banda.

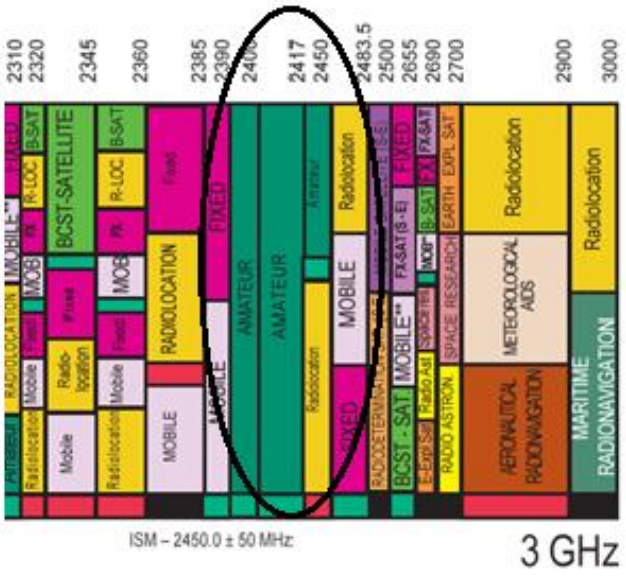


Figura 6 - Banda de Frequências da Tecnologia Bluetooth. Fonte [25].

Para ser possível a um dispositivo descobrir os restantes dispositivos que se encontram em seu redor, este faz o *broadcast* de uma mensagem e fica à espera de respostas. Se fizermos uma analogia com a Figura 5, o dispositivo A envia um conjunto de mensagens a questionar quem se encontra à sua volta. Este processo é conhecido como processo de inquirição ou *inquiry process*. A mensagem enviada não identifica o dispositivo A, apenas contêm um "código de acesso" que é reconhecido pelos restantes dispositivos.

Os dispositivos nas imediações de A que receberam a mensagem, caso estejam configurados em modo de descoberta (em termos técnicos, *inquiry scan mode*), respondem com uma outra mensagem para A indicando a sua presença. Neste caso em

particular a Figura 5, o dispositivo que realiza a procura (dispositivo A) irá receber, muito provavelmente, três mensagens dos dispositivos B, C e D. Cada resposta recebida por A contêm o endereço MAC e um número (indicando a classe do dispositivo), relativo ao dispositivo que foi descoberto. Estas informações serão analisadas e explicadas, em maior detalhe, numa das secções do capítulo seguinte. O processo de troca de mensagens até pode ser fácil de compreender, no entanto o mesmo é realizado de uma forma mais complexa.

Como já foi referido, a tecnologia Bluetooth opera na banda dos 2.4Ghz[2] e cada dispositivo divide essa mesma banda em 79 intervalos ou canais com 1Mhz de largura de banda para cada canal, onde cada um deles se destina a ser usado por um dispositivo para estabelecer uma conexão (ver Figura 7). Todos estes 79 canais são usados para estabelecer conexões, no entanto, existem 32 deles que para além deste objectivo também são utilizados no processo de detecção de dispositivos. Estes 32 canais estão divididos em dois grupos, com 16 canais cada grupo.

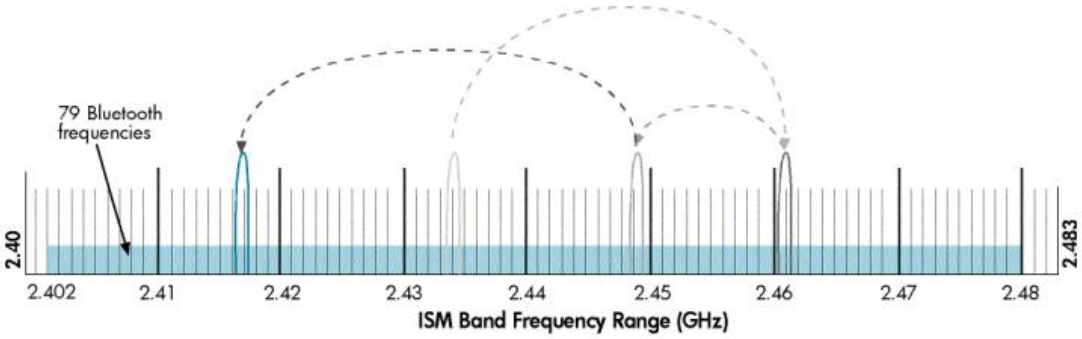


Figura 7 - Divisão da Banda de Frequências na Tecnologia Bluetooth. Fonte [19].

Após percorrer todo o conjunto A e enviar uma mensagem em cada frequência desse conjunto, o dispositivo que realiza a procura, volta à frequência inicial do conjunto

[2] Esta banda é ligeiramente diferente em alguns países (ver a Tabela 14 do Anexo 1).

e repete todo o processo. Isto é realizado cerca de 256 vezes[3]. Terminado o conjunto A o dispositivo de procura avança para o conjunto B efectuando novamente o mesmo processo. A mudança de frequências é feita 3200 vezes por segundo. Todo este processo de procura leva o seu tempo, sendo assim a Bluetooth SIG recomenda que para um dispositivo poder encontrar todos os dispositivos ao seu redor deve levar cerca de 10,24 segundos. Senão vejamos:

Tempo de espera num canal = $625\ \mu s$

Número total de canais para um conjunto = 16

Número de repetições = 256

$$625\ \times 16\ \times 256 = 2{,}56\ segundos$$

Sabendo agora que se muda de conjunto 4 vezes (duas iterações por conjunto), em cada procura:

$$2{,}56\ \times (2 \times 2) = 10{,}24\ segundos$$

Os dispositivos que não se encontram a realizar uma procura também realizam o mesmo processo de mudança de frequência, no entanto fazem-no com uma cadência muito menor do que o dispositivo de procura. Aqui a mudança de frequência é realizada a cada 1,28 segundos.

Como o dispositivo de procura (dispositivo A) se encontra a mudar de frequência e os restantes também (dispositivos B, C, D e E), todos estes acabarão por "encontrar" o dispositivo A numa mesma frequência (desde que a potência de sinal dos mesmos alcance o dispositivo A, o que não acontece com o dispositivo E). Ao receberem a mensagem de A, os dispositivos B, C e D, enviam uma mensagem como resposta de volta para A. No entanto esta resposta não é enviada imediatamente a seguir à recepção da mensagem de A, uma vez que dois dispositivos diferentes poderiam receber mensagem de A na mesma frequência e isso causava uma "colisão". Assim sendo, neste

[3] A partir da versão 2.1 do Bluetooth (desde 2004) foi introduzido o modo EDR (*Enhanced Data Rate*), que veio aumentar as velocidades de transmissão para cerca de 3 vezes, levando este processo a ser realizado mais rapidamente.

caso os dispositivos B, C, e D aguardam um pequeno tempo, calculado aleatoriamente, antes de enviarem a resposta para A, de modo a evitarem o problema anterior.

Esta troca de frequências ou salto de frequências é realizada através de um algoritmo de descoberta próprio, recorrendo à técnica de modulação de sinais FHSS (*Frequency Hopping Spread Spectrum*). É precisamente este algoritmo que tem vindo a ser alvo de estudos de forma a torná-lo mais rápido e eficiente para reduzir o tempo de procura por dispositivos. Esta técnica de modulação reduz ainda as perdas de informação (perdas de pacotes de dados) nas comunicações realizadas, bem como as interferências que sejam causadas por outras tecnologias, que operem na mesma banda de frequências como é o caso do WiFi.

Segundo os autores de [16], o tempo médio previsto para a descoberta de um dispositivo é de cerca de 5 segundos. No entanto, às vezes também pode levar entre 10 a 15 segundos. Tendo em conta a solução a desenvolver, para este trabalho, este processo torna-se crítico. Para ambientes de elevada mobilidade, quanto mais tempo demorar o processo de descoberta, menor a probabilidade de descoberta de um determinado dispositivo. Daí a importância do algoritmo de descoberta neste processo.

Feita esta breve explicação do processo de descoberta de dispositivos Bluetooth, convém lembrar que todo este processo atrás descrito é somente para a descoberta de novos dispositivos. O processo de descoberta efectuado ainda não conduz à "conexão definitiva" entre dispositivos.

3.2.1. Processo de Conexão e Topologias de Rede

O processo de conexão de dispositivos pode ser dividido em duas fases. A fase de inquisição (*inquiry*), que compreende todo o processo de descoberta anteriormente explicitado, e a fase de paginação ou emparelhamento. A fase de paginação não é aqui muito explicada ao detalhe, pois para o desenvolvimento deste projecto de dissertação não é necessário estabelecer uma "conexão definitiva" entre dispositivos.

Qualquer dispositivo que se pretenda conectar a outro, deve estar no chamado *modo page* enquanto que os dispositivos que aceitam as conexões devem-se encontrar no modo *page scan*. No contexto da Figura 5, se A se quiser conectar a B, A deve estar no modo *Page* e B no modo *Page Scan*. Para esta conexão se poder realizar com sucesso, o dispositivo A tem de conhecer o "número de identificação" (endereço de 48 bits) do dispositivo B. Este número pode ser do conhecimento de A de várias formas: através do processo de procura, através de um anterior processo de procura cujo mesmo ficou armazenado em A ou ainda sendo introduzido pelo fabricante de A. Quando A obtém o numero de identificação de B, o dispositivo A liga-se a B e dão início a uma troca de pequenas mensagens de forma a acordarem alguns "parâmetros" para a comunicação entre os dois. A partir deste ponto a ligação fica estabelecida, ou até que seja terminada por um dos dispositivos ou que algum deles não esteja mais na área de alcance do outro. Posteriormente, para uma percepção mais detalhada do processo de conexão, aconselha-se a leitura das referências [16] e [17].

No que diz respeito às topologias de rede na tecnologia Bluetooth são identificados dois tipos de redes, e as *Piconet* as *Scatternet*. Uma *Piconet* é uma rede que é composta por um máximo de oito dispositivos, onde um destes dispositivos é denominado de *Master* e os restantes sete de *Slave*. O *Master* possui um papel importante na rede, ele é o dispositivo responsável por criar a rede e ainda gerir todo o processo sincronização e de troca de informação entre os dispositivos, uma vez que só existe troca de informação entre *Master* e *Slaves*.

Neste tipo de redes também existem os dispositivos do tipo *Parked*, que são dispositivos que não tem uma conexão activa com o *Master*. Para um dispositivo do tipo *Parked* poder comunicar com o *Master* algum dos sete *Slaves* tem de mudar para modo *Parked*, para o novo dispositivo poder comunicar. Numa única *Piconet*, em teoria podem estar até cerca de 200 dispositivos em modo *Parked*.

Numa *Piconet* podem ainda encontrar-se dispositivos do tipo *Standby*. Estes dispositivos encontram-se num estado de espera onde periodicamente "acordam", para realizarem uma pesquisa, por "mensagens de *inquiry*", nos 32 canais de frequências que

são utilizados no processo de detecção de dispositivos, mostrando-se assim disponíveis para uma conexão com outro dispositivo.

As *Scatternet* são redes compostas por duas ou mais *Piconets*, onde os *Slaves* de uma *Piconets* podem estar associados a uma ou mais *Piconets*. Um dispositivo *Master* numa *Piconets* pode também ser *Slave* em outra.

3.3. *Stack do* Bluetooth

Como em todas as tecnologias existem vários protocolos envolvidos no estabelecimento de uma conexão entre dois ou mais dispositivos, onde os mesmos podem actuar ao nível de diferentes camadas de rede. É normal representá-los através de uma pilha protocolar ou *stack,* onde cada protocolo desempenha uma função específica no contexto da conexão a realizar. Em [16], os autores referem a *stack* como um conjunto de *device drivers*, bibliotecas de desenvolvimento e ferramentas para uso do utilizador, desenvolvidas por uma organização (fabricante do dispositivo, por exemplo), que permite aos mesmos criar facilmente aplicações que tirem partido das funcionalidades do dispositivo. Isto também acontece com a tecnologia Bluetooth.

A *stack* da tecnologia Bluetooth, representada na Figura 8, pode ser dividida em dois níveis principais: nível dos protocolos de transporte e nível dos protocolos de *midleware*. Entre estes dois níveis existe a ainda interface HCI (*Host Controller Interface*) com o objectivo de fornecer uma espécie de interface de controlo sobre as camadas mais baixas, nomeadamente para acesso ao estado do hardware e registos de controlo do mesmo.

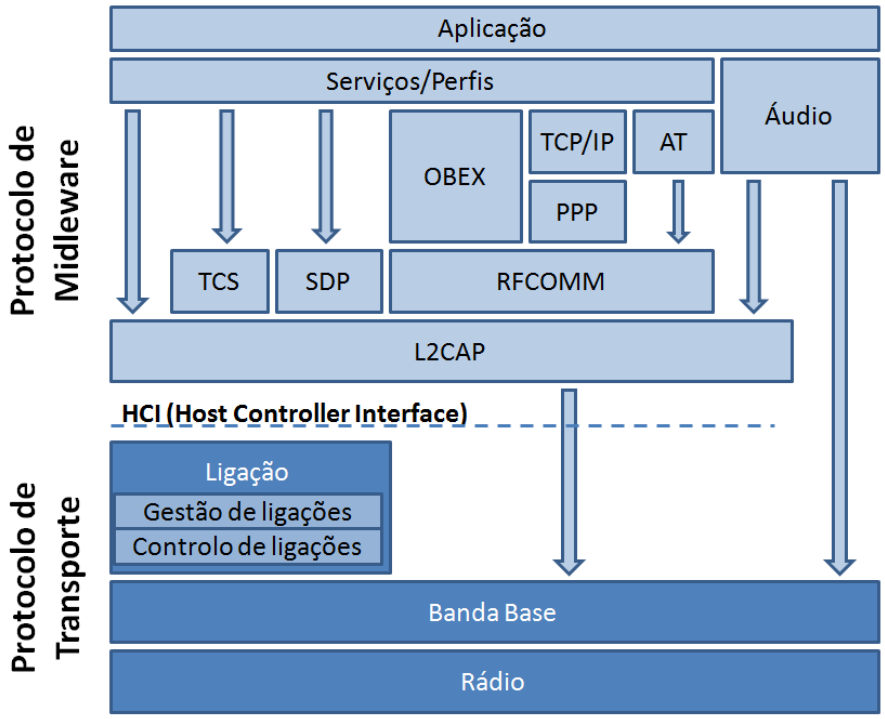


Figura 8 - *Stack* da Tecnologia Bluetooth.

No primeiro grupo, o protocolo de transporte actua ao nível de três camadas: a camada física, a camada de banda base e a camada de ligação. Estas são as camadas nativas da tecnologia Bluetooth e encontram-se implementadas em todos os dispositivos Bluetooth.

Na camada física está definido tudo que esteja ligado ao hardware da tecnologia Bluetooth. Aqui estão englobadas as questões relativas a modulações de sinal e potência de sinal.

A camada de banda base trata de parâmetros essencialmente ligados à formação e interpretação dos pacotes (construção das mensagens), enviados ou recebidos pelo dispositivo.

Já a camada de ligação trata de aspectos ligados ao estabelecimento de conexões, configurações do dispositivo, gestão de potência do mesmo e escalonamento de tráfego.

No segundo nível (protocolos de *midleware*), a *stack* é composta por um conjunto de diferentes protocolos, desenhados tendo em conta requisitos para as aplicações que necessitem de utilizar dispositivos Bluetooth. Neste nível, os diferentes sistemas operativos organizam os protocolos um pouco à sua forma, podendo por vezes retirar o controlo de alguns deles ao próprio utilizador e o mesmo ser "obrigado" a utilizar uma aplicação própria do fabricante desse dispositivo, para poder usufruir de todas as potencialidades do dispositivo. Na *stack* da Figura 8 estão representados os protocolos principais existentes neste nível mas podem existir mais.

Já foi referido anteriormente que a tecnologia Bluetooth está desenhada para a transmissão de mensagens (pacotes) de dados ou de voz, isto é visível na *stack* da Figura 8. Uma nova aplicação que é desenhada para operar recorrendo a esta tecnologia pode utilizar serviços/perfis que realizam a transmissão/tratamento de pacotes de dados ou então utilizar pacotes de áudio. Para além destes dois tipos de pacotes a tecnologia possui ainda os seus pacotes de controlo.

As aplicações de transmissão de áudio não necessitam obrigatoriamente de um tratamento por parte de um protocolo deste nível, podendo mesmo actuar directamente sobre as camadas mais baixas da *stack*, nomeadamente recorrendo directamente à camada de Banda Base.

Relativamente às aplicações de troca de dados, por vezes têm de recorrer a um vasto conjunto de protocolos intermédios antes de interagirem com as camadas mais baixas da *stack*. Por exemplo, uma aplicação de troca de mensagens de texto, que use o protocolo TCP/IP, pode operar sobre a tecnologia Bluetooth de forma fácil ou sem grandes complicações. A aplicação pode utilizar os protocolos PPP (*point-to-point protocol*), RFCOM (*Radio Frequency Communications Protocol*) e L2CAP (*Logical Link Control and Adaptation Protocol*), como se pode observar no esquema da *stack*.

Neste nível, na *stack* destacam-se ainda os protocolos OBEX (*OBject EXchange*), SDP (*Service Discovery Protocol*), TCS (*Telephony Control protocol Specification*) e AP (*Adopted Protocols*).

O primeiro protocolo deste grupo, o OBEX, é um protocolo especialmente desenhado para rapidamente se poderem realizar transferências de ficheiros entre dispositivos Bluetooth. Um outro protocolo o TCS, é mais utilizado pela tecnologia quando é necessário realizar um controlo e gestão das chamadas de voz entre dispositivos. O protocolo SDP destina-se a realizar a descoberta de serviços existentes em outros dispositivos. Ele é utilizado no desenvolvimento do projecto em mãos e o seu funcionamento é explicado em maior detalhe na próxima secção.

Um outro protocolo da *stack* é o RFCOM, é um protocolo baseado em *streams* de dados e oferece as mesmas garantias de serviço que o protocolo TCP, das redes IP. Foi desenhado para emular o protocolo RS-232. Como é referido em [16], comparado com o protocolo TCP tem um reduzido número de portas de ligação o que traz algumas restrições no fornecimento dos serviços das aplicações.

Quase todos os protocolos deste nível recorrem ao protocolo L2CAP, pois ele serve de protocolo de transporte para quase todos os restantes protocolos deste nível. Ele é o responsável por realizar o interface, fazer uma espécie de adaptação, para os protocolos de mais alto nível. Este protocolo, fazendo de novo uma analogia com as redes IP, pode ser comparado ao protocolo UDP (*User Datagram Protocol*) em alguns aspectos, mas existem pequenas diferenças que o tornam mais completo que o UDP, onde a sua principal característica é poder ser configurado para diferentes modos de operação [16].

3.4. Processo de Descoberta de Serviços

Um serviço ou perfil diz respeito a uma determinada aplicação instalada num dispositivo, que se encontra em execução no mesmo, com o objectivo de realizar um conjunto de acções a pedido de outro dispositivo remoto ou fornecer determinado tipo de informação a outro dispositivo remoto. Ou seja, um serviço ou perfil actua como uma

"norma" de como os dispositivos envolvidos na comunicação devem realizar determinada tarefa. Isto é útil quando se encontram em comunicação dois dispositivos de fabricantes diferentes, por exemplo.

Um serviço é uma funcionalidade que um dispositivo dispõe. São exemplos de serviços as seguintes funcionalidades: *Cordless Telephony*, *LAN Acces*, *File Transfer*, etc. Para cada serviço ou perfil são usados determinados protocolos da *stack* do Bluetooth como se pode ver pela Tabela 15 do anexo A1, adaptada de [17].

No que diz respeito à descoberta de serviços de dispositivos, a tecnologia Bluetooth recorre a um protocolo especifico o SDP (*Service Discovery Protocol)*. Vai então ser feita uma sucinta explicação do protocolo, para se entender melhor o processo de descoberta de serviços. Convêm referir, que a descoberta de serviços só pode ser realizada caso os processos de descoberta de dispositivos e de paginação ou emparelhamento, explicados anteriormente, tenham sido realizados.

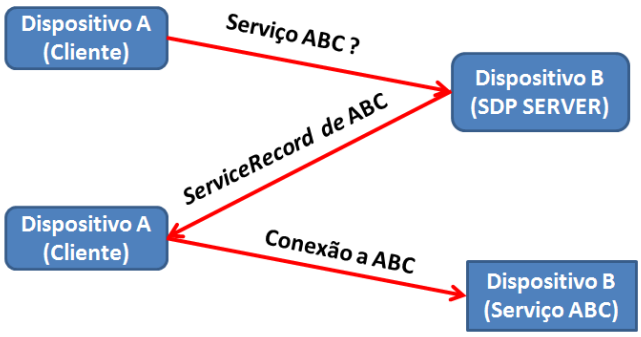


Figura 9 - Exemplo de uma conexão a um serviço usado o SDP.

Os dispositivos Bluetooth possuem uma aplicação servidora em execução, com o objectivo de fornecer informação acerca dos serviços fornecidos pelo dispositivo. As aplicações Bluetooth a correr no dispositivo local, são então registadas nessa aplicação servidora. Suponha-se agora que se tem dois dispositivos Bluetooth, o dispositivo A e o

dispositivo B. Após o dispositivo A detectar o dispositivo B na sua proximidade, o dispositivo A indica ao dispositivo B o serviço que pretende utilizar. Por sua vez o dispositivo B tem uma aplicação servidora SDP activa, onde se encontra lá uma lista de todas a aplicações activas que o dispositivo B suporta bem como os serviços requeridos pelas mesmas. Em seguida, dispositivo B devolve a A um registo com informações acerca do serviço pretendido por A. O dispositivo A após verificar esse registo, conecta-se a B na porta respectiva desse serviço, como ilustra a Figura 9 (adaptada de [16]).

O registo que o dispositivo B devolve ao dispositivo A tem o nome de *ServiceRecord* que contém uma lista de parâmetros que descrevem o serviço pretendido. Um desses parâmetros é o *ServiceID* que contém um número, o UUID (*Universally Unique Identifier*), que não é nada mais do que um numero unívoco que identifica o serviço. Outro parâmetro importante é o *ServiceClassIDList* que funciona como uma espécie de segundo UUID, para quando se têm duas aplicações diferentes no mesmo dispositivo que utilizem os mesmos serviços, como por exemplo serviços de *File Transfer*, ou ainda quando se tem uma aplicação que utilize vários serviços. Nesta lista encontra-se ainda mais parâmetros sendo um deles o conjunto de protocolos da *stack* do Bluetooth que são necessários utilizar na disponibilização do serviço pretendido. Quem pretender criar um serviço deve regista-lo no servidor SDP criando um novo *ServiceRecord* para o serviço em questão.

Cada dispositivo Bluetooth pode ter um servidor activo e um cliente de serviços a correr ao mesmo tempo, onde aqui é usado o protocolo L2CAP (como se pode ver pela *stack* do Bluetooth), para coordenar este processo. O protocolo SDP também é útil para se descobrir se determinado serviço ainda se encontra activo. □

4. Especificação e Desenvolvimento da Solução

Para que os objectivos propostos para este projecto sejam alcançados é necessário desenvolver uma solução, que satisfaçam os mesmos. Esta solução tem de fornecer um conjunto de informações para que com as mesmas seja possível responder ao problema em mãos. O problema implícito neste projecto é a avaliar a descoberta de dispositivos em ambientes de elevada mobilidade.

A solução encontrada passa pelo desenvolvimento de uma aplicação de *scan* que recolhe algumas informações acerca dos dispositivos *Bluetooth,* que se encontrem em determinada área geográfica. As informações são recolhidas através de um *scan,* realizado por um determinado período de tempo. Para que seja possível recolher este conjunto de informações, a abordagem adoptada, foi a instalação de alguns componentes de recolha em diferentes locais, localizados ao longo de uma rua. O objectivo desta abordagem era conseguir que os dispositivos fossem detectados em todos estes locais.

Neste capítulo, é indicada e explicada a solução encontrada para ultrapassar os problemas encontrados. Também é descrito o cenário escolhido, onde a solução deve ser implementada e testada, uma vez que o mesmo tem influência na forma como a solução deve ser construída, tendo em conta o problema em mãos.

Serão mencionados alguns conceitos, necessários para o desenvolvimento da solução, nomeadamente a forma como as informações são recolhidas e para que

componente deve ser desenvolvida a aplicação de *scan*. A leitura do capítulo anterior é recomendável para a percepção de alguns dos conceitos.

4.1. Cenário de Operações

Para se encontrar uma solução correcta para este problema, é necessário conhecer bem o ambiente ou o cenário, onde a mesma solução tem de ser implementada. Assim sendo, e dado que um dos objectivos é o estudo de como a tecnologia Bluetooth se comporta num ambiente de elevada mobilidade, nada melhor do que ter como cenário de implementação a rua de uma cidade. As ruas de uma cidade são o cenário ou ambiente ideal para o estudo que se pretende realizar, uma vez que são local de passagem para muitas pessoas e veículos, e onde a mobilidade dos mesmos é elevada. Foi então escolhida uma rua da cidade de Guimarães, a rua Gil Vicente, como se pode ver na Figura 10. Esta rua de sentido único, possui várias lojas de comércio que atraem um grande número de pessoas.

Figura 10 - Localização geográfica dos pontos de recolha.

Nesta rua foi definido um conjunto de três locais, para recolha de informação. A escolha deste conjunto, só foi possível devido à gentileza alguns lojistas, que se mostraram interessados em colaborar com o projecto. Assim, os lojistas cederam parte do seu espaço comercial para a instalação do material necessário à recolha de informação.

É necessário agora encontrar uma estratégia, para recolher essas informações e também definir e identificar quais os dispositivos a utilizar na recolha. Em termos de logística, os locais encontrados não ofereciam ligação a nenhum tipo de rede informática, ou seja, toda a informação que foi recolhida teve de ser armazenada no local e analisada posteriormente, no final do teste. A portabilidade do material a instalar nos locais, era também um aspecto a ter em conta, uma vez que não se pretendia incomodar o normal funcionamento dos estabelecimentos comerciais, enquanto os testes estavam a decorrer. Este ambiente ou cenário de teste, também traz algumas complicações para a análise das informações recolhidas, mas isso é explicado no próximo capítulo.

4.2. Informação a Recolher

Para se saber quando determinada pessoa ou veículo passou por determinado local, é preciso recolher e armazenar informação relativa ao dispositivo Bluetooth associado a essa pessoa ou veículo. A informação recolhida deve identificar o dispositivo e o seu local de passagem. Convém agora explicar que a informação a recolher não identifica a pessoa ou o veículo que transporta o dispositivo Bluetooth, mantendo assim salvaguardada "a condição" de privacidade, para pessoas ou veículos, das quais a informação seja eventualmente recolhida. A partir deste ponto, neste documento, e de forma a se tornar mais fácil a referência a estas informações, é designado pelo nome de amostra todo o conjunto de informações recolhidas que identifique um dispositivo Bluetooth, numa localização específica e num momento específico. Põe-se agora a questão: que informação recolher? O que é necessário para identificar um dispositivo Bluetooth?

Explicado (no capítulo anterior), todo o processo de descoberta ou detecção de dispositivos é mais fácil introduzir a descrição dos atributos, que se pretendem obter de cada dispositivo. As principais informações a "recolher", que constituem uma amostra, para cada dispositivo são:

- O seu endereço MAC;

- O “nome” do dispositivo;
- A data, hora e local, quando o dispositivo foi detectado;
- A classe/categoria do dispositivo;
- Os serviços/perfis *Bluetooth* que o dispositivo detectado suporta;
- A potência do sinal detectado (opcional).

O endereço do dispositivo ou endereço *MAC (Media Access Control)*, não é nada mais do que um número de 48 bits que serve para identificar univocamente um dispositivo. Este número é atribuído ao dispositivo na altura de fabrico do mesmo pelo seu fabricante. Este endereço é usado como referência ao dispositivo e não pode ser alterado ou modificado.

O atributo, nome do dispositivo é um nome que o utilizador do dispositivo Bluetooth pode “personalizar” ao seu gosto, de forma a tornar mais fácil a identificação do dispositivo quando o utilizar. Este atributo por si só não seria muito relevante para a identificação de dispositivos, pois poderíamos encontrar dois ou mais dispositivos com o mesmo nome e isso poderia provocar alguma “confusão”, daí o uso do atributo anterior na identificação.

A data, a hora e o local são informações que não são fornecidas pelos dispositivos detectados, mas são definidas pela aplicação de detecção quando um dispositivo é encontrado.

Associada também à identificação de dispositivos, existe ainda o tipo ou a classe/categoria a que o dispositivo pertence. A tecnologia Bluetooth divide os dispositivos em vários tipos de classes [21], as principais (*Major Device Classes*) e onde por sua vez estas classes também dividem os dispositivos, em sub-grupos mais específicos (*Minor Device Classes*). Alguns exemplos de *Major Device Classes* são: *Miscellaneous, Computer, Phone*, *LAN /Network Access point*, *Audio/vídeo*, *Toy*, etc. Para a *Major Class Phone*, por exemplo, existem as seguintes *Minor Classes: Cellular,* Cordless, *Smart phone*, etc.

Por fim, os serviços ou perfis dos dispositivos Bluetooth e a potência do sinal recebido, designada por RSSI (*Received Signal Strength Indicator*), são também informações interessantes a recolher. Os serviços permitem descrever de uma forma geral as potencialidades do dispositivo em questão, bem como, o tipo de aplicações podem ser usadas pelos mesmos. A classificação dos serviços suportados pelo dispositivo também se encontram divididas em diferentes classes como acontece com o parâmetro da classe/categoria, explicado anteriormente. Já através da potência do sinal pode ser estimada a distância a que determinado dispositivo está do dispositivo de *scan* e assim determinar a localização do mesmo, aquando do momento da sua detecção. Contudo por questões de desenho da aplicação de *scan*, explicadas na secção 4.3.2, o atributo RSSI não é utilizado.

4.3. Hardware para Instalação da Aplicação

Uma outra questão que deve ser colocada, para o desenvolvimento desta aplicação de *scan* é: Onde vai ser instalada a aplicação? Que hardware se deve utilizar? A resposta a estas questões, têm influencia no desenho da aplicação, assim sendo convém analisar bem todas as hipóteses existentes.

Como a tecnologia *Bluetooth* tem vindo a ser desenvolvida, desde o seu início, de forma a realizar a comunicação sem fios entre dispositivos que se encontrem nas proximidades uns dos outros, possui algumas características que podem ser um problema no desenvolvimento desta aplicação. Sendo assim, a aplicação deve ter em conta, alguns requisitos, que limitam as escolhas do hardware a utilizar. Os requisitos identificados são os seguintes:

- Detectar os dispositivos, havendo o mínimo de interacção possível entre o dispositivo da aplicação e o dispositivo detectado;
- Recolher as informações necessárias de um dispositivo, num curto espaço de tempo;
- Armazenar ou transferir as informações relativas aos dispositivos já detectados, sem que se comprometa o processo de *scan* de novos dispositivos.

- Uma vez iniciada a aplicação, convêm funcionar durante um período de tempo longo, evitando que a mesma seja reiniciada.

4.3.1. Hardware a Utilizar

Dependendo da capacidade computacional dos dispositivos a utilizar, para implementar a aplicação, os requisitos referidos anteriormente e as condições logísticas do cenário de implementação da solução, torna-se necessário identificar quais os dispositivos que mais se adaptam a eles.

Para a implementação e instalação da aplicação, em termos de hardware, foram identificados 3 dispositivos (ver Figura 11):

- *Router*;
- Computador portátil;
- Placa electrónica/ microcontrolador.

Figura 11 - Dispositivos considerados para a instalação da aplicação.

A primeira opção, router ou encaminhador, consiste em adaptar um router WiFi, substituindo o seu software por um outro que implemente a aplicação pretendida. Esta opção, parece ser a mais viável em termos de portabilidade, uma vez que os routers são mais compactos em tamanho que os portáteis e com maior capacidade de processamento que os microcontroladores. No entanto necessitariam de algum

hardware adicional como *dongles Bluetooth* e *pen drives*/cartões de memória, pois os mesmos possuem pouca capacidade de armazenamento de dados.

Relativamente à segunda opção, com uso de computadores portáteis não seria necessário hardware adicional e eventualmente a aplicação poderia realizar operações computacionais mais complexas. A única desvantagem é de facto a comodidade de transporte, ou seja a portabilidade do hardware.

Por fim a última opção, o uso do dispositivo electrónico/microcontrolador (por exemplo, a plataforma electrónica de prototipagem *Arduino*), apresenta também uma reduzida capacidade de processamento e seriam também necessários mais recursos, tal e qual como nos *routers*. A grande vantagem é que dos três dispositivos este cativa mais, por apresentar maior portabilidade, tornado a sua instalação, no local, um pouco mais cómoda, o que é importante para o propósito da aplicação.

No que diz respeito ao consumo energético, o router e o micro controlador não podem funcionar com baterias, tendo os mesmos de se encontrar ligados à corrente eléctrica, uma vez que o seu funcionamento tem de ser constante por um considerável período de tempo. Aqui, os computadores portáteis têm vantagem em relação aos outros dois dispositivos porque, em caso de falha de corrente eléctrica, possuem baterias de maior duração, e isso garante o funcionamento da aplicação por mais alguns minutos, podendo assim salvaguardar-se o funcionamento da aplicação.

Analisados os três dispositivos, talvez o router fosse uma solução que se adaptava bem ao problema em mãos. Instalava-se uma pen ou um *dongle* Bluetooth no *router* e conectava-se o mesmo a alguma rede, com acesso à Internet, para envio dos dados para um servidor central. Se não houvesse rede ou ligação à Internet, poder-se-ia armazenar os dados num disco externo ou pen drive. Até aqui, não haveria nenhum problema em utilizar os routers para implementar a aplicação de *scan*. Por outro lado,

um dos requisitos a ter em conta na implementação da aplicação, quando se forem realizar os testes, é a sincronização temporal de todos os locais de recolha. Por exemplo, se num dos locais falhasse a corrente eléctrica ou o router fosse desligado, ainda que por breves instantes, e reiniciado de seguida, perderia a sua sincronização temporal em relação aos outros locais de recolha. Assim, não se torna muito fiável a utilização deste dispositivo para o propósito em causa.

O outro dispositivo não escolhido, a placa electrónica, facilmente se poderia afirmar que seria a solução mais adequada e óbvia. Mas a placa também apresenta o mesmo problema que foi mencionado para o router. No entanto o problema poderia ser evitado, acrescentando um módulo de bateria à placa. A desvantagem no uso deste dispositivo seria a integração dos vários módulos de hardware na placa (relógio, bateria, modulo de hardware Bluetooth, modulo para leitor de cartões de memoria, etc.), e a correcta gestão de todos os módulos, por parte do microcontrolador.

Assim sendo, ficou decidido que o dispositivo a utilizar seria o computador portátil. Este dispositivo já dispõe de uma maior capacidade de processamento e armazenamento, e se o problema mencionado ocorrer, basta reiniciar o mesmo não havendo problemas de sincronização. O computador cumpre razoavelmente todos os requisitos necessários indicados anteriormente.

A solução encontrada para o problema hardware pode ficar resolvida com o uso do computador portátil, mas este ainda não seria o ideal para este problema. O ideal seria o desenvolvimento de uma placa electrónica, ou seja a criação de um dispositivo electrónico de raiz, adequado para os requisitos indicados anteriormente. Teria de ser um dispositivo que estivesse equipado de raiz com um módulo Bluetooth, suporte para armazenamento de dados, suporte para interface de rede, bateria e micro controlador com capacidade de processamento adequada. Este caminho não foi escolhido, uma vez que o desenvolvimento de hardware não se encontra no âmbito desta dissertação.

4.3.2. Linguagem de Programação e API a Utilizar

Escolhido o dispositivo onde a aplicação vai ser instalada, nomeadamente o computador portátil, e as informações que se pretendem obter, deve agora ser analisada uma forma de integrar tudo isto. A forma encontrada foi o desenvolvimento de uma aplicação de *scan*. Considerando isto, foram então levantadas outras questões: Que linguagem de programação utilizar e qual o sistema operativo a utilizar? Como ler as informações do "interface Bluetooth", presente no computador?

Relativamente à leitura das informações, a resolução encontrada foi o uso de uma API (*Application Programming Interface*), que disponibilize funções de leitura das informações pretendidas. Uma API é um conjunto de funções, que permitem aceder ou simplificam o acesso às funcionalidades de um determinado dispositivo ou software instalado no computador.

Como já foi referido na secção 3.3, o acesso ao hardware Bluetooth é feito por intermédio de uma *stack*. Contudo, cada sistema operativo pode organizar o acesso à *stack* Bluetooth de uma forma particular, e isto pode levar, a quem desenhe aplicações deste tipo, a não ter acesso ou não poder explorar na totalidade as potencialidades do dispositivo. Este aspecto foi notado na aplicação desenvolvida, pois foi usada a *stack* da *Microsoft* e a mesma não devolve informações acerca do RSSI dos dispositivos.

Na Tabela 1, está presente um pequeno resumo que indica que protocolos, usados na tecnologia Bluetooth, estão acessíveis, tendo em conta algumas APIs e *stacks* de sistemas operativos.

Analisando qual a linguagem de programação a utilizar, existem várias soluções a considerar. Feita uma pesquisa, a maior parte das aplicações desenvolvidas, para uso da tecnologia Bluetooth, recorrem ao uso das linguagens *Python* e JAVA. Isto ocorre talvez porque são as linguagens que actualmente possuem APIs mais "simpáticas" de utilizar e gratuitas, para aceder às funcionalidades do dispositivo Bluetooth. Contudo, existem

também algumas aplicações desenvolvidas em C++ e nas linguagens .NET, como é o caso do VB.net e do C# (C *sharp*).

API/Bluetooth *Stack*	RFCOMM	L2CAP	SCO	SDP	HCI
Microsoft (Windows XP)	X				
Widcomm/Broadcom (Windows XP)	X	X	X	X	X
Microsoft (Windows Vista/7)	X	X	X	X	
BlueZ (GNU/Linux)	X	X	X	X	X
OS X	X	X			X
PyBlueZ (Windows XP)	X				
PyBlueZ (GNU/Linux)	X	X	X	X	X

Tabela 1 - Bluetooth Protocol *Stacks*.

Contudo nesta matéria existe uma "condicionante". Um outro objectivo desta dissertação é a integração desta aplicação de *scan* numa outra já desenvolvida, a aplicação Epi. A aplicação Epi, encontra-se elaborada recorrendo à linguagem C#. Assim por esta razão, para o desenvolvimento da aplicação de *scan*, foi escolhida a linguagem de programação C#. A API a utilizar é a API do projecto 32feet.NET v3.0 (*Personal Area Networking for .NET*)[22].

As bibliotecas desta API visam facilitar o acesso à *stack* da tecnologia Bluetooth, pois suportam principalmente as *stacks* da *Microsoft* e da *Broadcom/Widcomm*, entre outras. Esta é a API escolhida pois é bastante completa e porque actualmente não existem muito mais soluções "abertas" à comunidade, tendo em conta a linguagem de programação a utilizar e o ambiente onde a mesma vai ser integrada, ou seja uso da linguagem *C#* no sistema operativo *Windows*.

4.4. Arquitectura da Aplicação de *Scan*

Conhecida a API e a linguagem de programação a utilizar foi desenhada a aplicação. Nesta secção é dado a conhecer a forma como foi construída a aplicação e explicada a sua arquitectura. A aplicação foi construída de modo a ser facilmente integrada em outras aplicações. A aplicação actua como um módulo, ou seja, extensão para outras aplicações que pretendam realizar *scan*/detecção de dispositivos Bluetooth e tenham como base a linguagem de programação C# (C *sharp*). Esta pode ainda ser utilizada de forma independente, isto é, sem ser integrada como parte de outra aplicação, uma vez que possui um pequeno interface, em modo texto (linha de comandos), para que o utilizador possa interagir com a mesma.

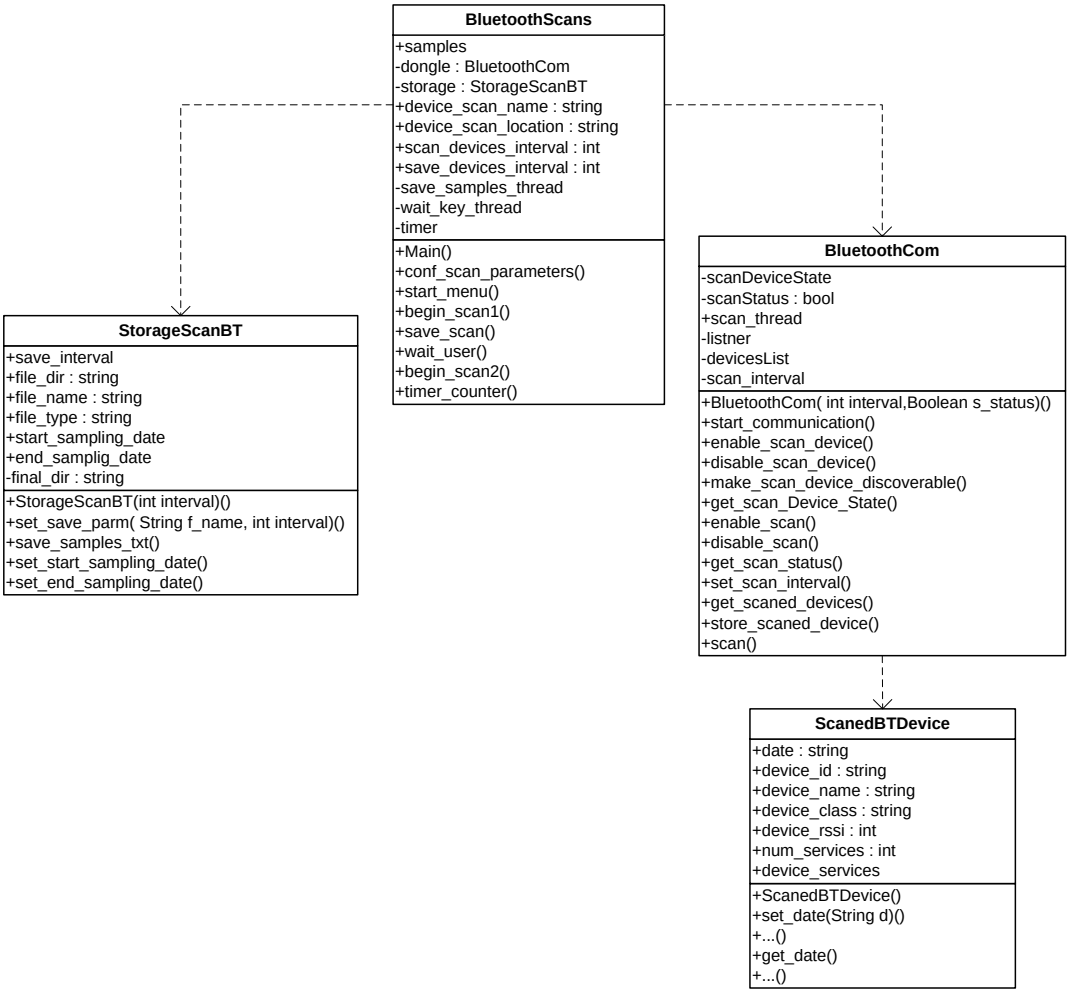


Figura 12 - Diagrama de Classes da Aplicação.

As funcionalidades desenvolvidas para esta aplicação são: alteração de parâmetros de *scan*, realização de um *scan* por tempo indeterminado e ainda realização de *scan* por um período de tempo específico.

A arquitectura da aplicação encontra-se assente num conjunto de quatro classes, que actuam em conjunto de forma a corresponder aos requisitos da aplicação em mãos. As classes que constituem a aplicação estão representadas na Figura 12, através do diagrama de classes da aplicação. Na figura são apresentados os principais métodos e parâmetros das classes. Como se pode visualizar, a aplicação é composta por 4 classes: *BluetoothScans, BluetoothCom, ScanedBTDevice* e *StorageScanBT.* No anexo A2 encontram-se representadas estas quatro classes em com maior detalhe, uma vez que na Figura 12 apenas se pretende indicar as variáveis e os métodos mais relevantes de cada classe.

A classe *BluetoothScans* coordena todo o processo de detecção com o auxílio das restantes classes. Esta classe pode ser considerada a classe principal da aplicação e ainda tem a função de ser a "intermediária", para que outras aplicações ou os utilizadores (quando usada individualmente) possam tirar partido das funcionalidades da aplicação.

As classes *BluetoothCom* e *ScanedBTDevice* coordenam todo o processo de detecção de dispositivos Bluetooth e armazenam os resultados na classe *BluetoothScans*, para mais tarde a classe *StorageScanBT* ter acesso aos mesmos.

Por fim, a classe *StorageScanBT* é responsável por organizar e armazenar os dados recolhidos em memória de forma definitiva, criando ficheiros com os dados dos *scans* (amostras) para análise posterior, uma vez que não é requisito da aplicação fazer a análise dos dados. Se olharmos para esta arquitectura, pode-se ver que se por alguma razão se precisasse de enviar as amostras dos *scans* pela rede, facilmente se poderia criar uma nova classe e integrá-la na arquitectura da aplicação sem grandes complicações. Neste caso, a nova classe a ser criada apenas teria de ter acesso aos dados referente às amostras armazenadas na classe *BluetoothScans*. Em seguida é descrita cada uma das classes de forma individual, usando numa abordagem *top-down*.

4.4.1. Classe *BluetoothScans*

A "classe principal" da aplicação tem o nome de *BluetoothScans,* e é a responsável pela instanciação e gestão dos objectos globais das diferentes classes, mais especificamente das classes *BluetoothCom* e *StorageScanBT*, por intermédio das varáveis de classe *dongle* e *storage*, respectivamente. Esta classe proporciona o acesso a todas as funcionalidades da aplicação desenvolvida. As suas funcionalidades são: alteração de parâmetros de *scan*, realização de um *scan* por tempo indeterminado e ainda, realização de *scan* por um período de tempo específico.

Para ser possível tirar partido destas funcionalidades é necessário recorrer a um conjunto de variáveis e funções que convêm explicar. Observando novamente a Figura 12, vê-se que a classe contém uma variável de nome *samples*. Esta variável do tipo *List,* vai armazenar todos os dispositivos que foram detectados. Assim, a classe *BluetoothCom* tem acesso à variável para armazenar os dispositivos encontrados, preenchendo cada elemento da lista com os dados referentes a um dispositivo (amostra). A variável *samples* é então usada também pela classe *StorageScanBT*, onde a mesma retira os diferentes elementos da lista e os armazena em disco, tendo em conta um determinado intervalo de tempo. Resumindo, a classe *BluetoothCom* armazena os dados dos *scans* em *samples* (amostras) e a classe *StorageScanBT* retira os mesmos guardando-os de forma definitiva em disco.

Para controlo do tempo de procura de dispositivos e ainda para controlo do intervalo de gravação das amostras em disco, a classe usa as variáveis *scan_devices_interval* e *save_samples_interval*, respectivamete. A primeira variável indica o período de *scan* dos dispositivos, ou seja, o tempo em que o hardware Bluetooth do dispositivo de *scan*, vai realizar a procura por novos dispositivos. Este processo de procura encontra-se explicado na secção 3.2. Já a segunda variável, diz respeito ao período em que as amostras vão ser retiradas da lista de amostras e armazenadas em ficheiro. Estas variáveis representam tempos que estão definidos em segundos e são utilizadas pelas *threads* de controlo de *scan* e de controlo de gravação.

Estas *threads* são explicadas mais à frente, por agora vai ser analisado o que fazem as funções da classe. Contudo, na classe existem mais duas variáveis que convém ainda apontar: *device_scan_location* e *device_scan_name*. A primeira serve para indicar o local onde é realizado o *scan* e a segunda descreve o nome do dispositivo de *scan*, pois poderiamos ter vários dispositivos com a aplicação instalada no mesmo local a realizar scans.

Nesta classe as principais funções que implementam as funcionalidades da aplicação são: *save_scan()*, *conf_scan_parameters()*, *begin_scan()*, *wait_user()*, *begin_scan2()*, *timer_counter()* e *start_menu()*.

A primeira função, *save_scan(),* é invocada sempre que se pretenda gravar para o disco as amostras que se encontrem na lista de amostras, referenciada pela variável *samples*.

Com a segunda função, *conf_scan_parameters()*, são alterados os parâmetros de *scan*, como por exemplo os tempos *scan_devices_interval* e *save_samples_interval*. Esta função é utilizada principalmente quando a aplicação está a funcionar sem ser parte integrante de outra, ou seja, serve para o utilizador definir os parâmetros de controlo do *scan*.

Assim como a função anterior, a função *start_menu()*, também só é utilizada quando o utilizador usa a interface de texto da aplicação. Esta função lista as várias opções que o utilizador tem ao seu dispor na aplicação.

As funções *begin_scan()* e *begin_scan2()*, dão início a todo o processo de *scan*. Estas duas funções definem o modo como o *scan* é feito, sem temporizador (uso da primeira função) ou com temporizador (uso da segunda função). A primeira destas duas funções lança uma *thread* ficando à espera que o processo de *scan* seja interrompido de forma explícita (por ordem do utilizador). Já a segunda função lança uma *thread* que espera determinado tempo antes de interromper o processo de *scan*, ou seja, o utilizador tem de esperar que o processo de *scan* termine.

Por fim, nesta classe estão ainda implementadas um conjunto de 4 *threads*, mais especificamente uma *thread* para a gravação das amostras, uma *thread* para lidar com o temporizador, uma *thread* para listar os dispositivos encontrados no ecrã e outra thread para esperar que o utilizador prima uma tecla para terminar o *scan* actual. Contudo nem sempre são criadas todas estas threads. A *thread* de gravação de amostras é a única do conjunto das quatro que é sempre criada, com as outras isto já não acontece. A *thread* que lida com o temporizador, só é criada quando se pretende realizar um *scan* por determinado período de tempo, pois é necessário o controlo de um temporizador. As últimas duas threads descritas, só são criadas quando a aplicação funciona em modo individual, onde é necessário o utilizador ter um controlo de como está a decorrer o processo de *scan*.

4.4.2. Classe *BluetoothCom*

Para ser realizado *scan* de dispositivos é essencial o controlo do hardware Bluetooth na máquina onde se encontra instalada a aplicação. Esse é o objectivo desta classe. Esta classe usa a API do projecto *32feet*, já discutida na secção 4.3.2. Aqui está presente um conjunto de variáveis de classe privadas que orientam o controlo do hardware. As variáveis são as seguintes: *scanDeviceState*, *scanStatus*, *scan_interval*, *listner*, *devicesList* e *scan_thread*.

A primeira variável, *scanDeviceState*, indica o estado actual do dispositivo Bluetooth que a aplicação está a utilizar. Este estado pode ser *PowerOff* (desligado), *Connectable* ou *Discoverable*. No primeiro caso, o dispositivo está desligado, no segundo caso, o dispositivo está conectável mas não pode ser descoberto por outros dispositivos (está em modo oculto, explicado na secção 3.2), no terceiro caso o dispositivo pode ser descoberto por outros dispositivos e aceitar conexões.

Caso seja necessário saber se o dispositivo já se encontra a realizar um *scan*, existe a variável *scanStatus,* que indica se o *scan* se encontra a decorrer ou não.

O tempo de *scan*/procura por dispositivos vizinhos está referenciado pela variável *scan_interval*, o valor desta variável é exactamente o mesmo que o da variável *scan_devices_interval,* da classe *BluetoothScans*.

As variáveis *listner* e *devicesLisner* têm o objectivo de implementar, respectivamente, um *socket* cliente e um *Array,* com informação à cerca dos dispositivos detectados.

Por fim existe a variável *scan_thread* que, como o nome indica, controla a *thread* de *scan*. Esta classe possui uma *thread* em funcionamento para permitir que, ao mesmo tempo que o dispositivo Bluetooth esteja a realizar um *scan*, possa ser possível obter informação acerca do mesmo, ou até poder parar o *scan* sem ter que desligar o dispositivo Bluetooth. Conclusão, sempre que seja necessário terminar um determinado *scan,* basta terminar a *thread* ficando o dispositivo ainda activo. Esta *thread* é definida por intermédio da função *scan()* desta classe.

A função *scan()* recorre a um método da API Bluetooth utilizada, de seu nome *discovery_devices()*. Este método realiza o processo de descoberta de dispositivos (*inquery process*), num determinado tempo definido na variável de classe *scan_interval*, onde aqui apenas obtém o endereçao fisico (endereço MAC) de cada dispositivo encontrado. O método recorre também ao protocolo SDP, contatando individualmente cada dispositivo encontrado, de forma a obter as restantes informações do mesmo.

Esta classe contém ainda um conjunto de outras funções, que podem ser divididas em dois grupos. Um grupo implementa o controlo dos parâmetros de funcionamento do dispositivo Bluetooth e o segundo grupo implementa o controlo da acção de *scan*.

Para terminar a descrição desta classe, falta apenas explicar como são organizadas e guardadas as funções relativas a um dispositivo detectado. Cada vez que um dispositivo é detectado ele possui um conjunto de atributos que é necessário armazenar e "organizar". Esses atributos são organizados num objecto do tipo *ScanedBTDevice* e posteriormente são inseridos na lista de dispositivos detectados, através da variável global *samples*, presente na classe *BluetoothScans*.

4.4.3. Classe *ScanedBTDevice*

Como se referiu, na secção anterior, quando um novo dispositivo Bluetooth é detectado é necessário armazenar um conjunto de informação relativa ao mesmo. A informação recolhida (amostra), é então organizada no seguinte conjunto de variáveis: *date*, *device_id*, *device_name*, *device_class*, *device_rssi*, *num_services* e *device_services*. Estas variáveis não necessitam de grande explicaçao pois cada uma armazena informaçao, acerca das informações descritas na secção 4.2. Convém esclarecer que a variável *num_services* diz respeito ao número de serviços disponíveis (que o dispositivo encontrado possui) e a variavel *device_services* é uma lista com o nome desses serviços.

As funções que esta classe implementa, são apenas funções de "*get*" e "*set*" para cada uma das variáveis indicadas atrás, que permitem o acesso por parte de outras classes às mesmas.

4.4.4. Classe *StorageScanBT*

A última classe que falta analisar da aplicação desenvolvida é a classe *StorageScanBT*. O objectivo principal desta classe é armazenar as informações relativas aos dispositivos encontrados. Estas informações são retiradas da lista de dispositivos através da variável *samples* presente na classe *BluetoothScans*, onde posteriormente são armazenadas num ficheiro de texto no disco. A forma como os dados estão organizados nesse ficheiro é descrita em maior detalhe na seguinte secção deste capítulo.

As variáveis desta classe que convém referir são as seguintes*: file_dir, file_name, start_sampling_date, end_sampling_date.* A primeira variável indica a directoria onde se vai encontrar armazenado o ficheiro de texto, o segundo indica o nome do ficheiro das amostras e a terceira e quarta variáveis indicam o tempo (data e hora) de início do processo de scan e o tempo (data e hora) de fim do processo de *scan*.

As funções da classe também estão ligadas com o processo de gravação do ficheiro texto e ainda com a obtenção do tempo de gravação inicial e final.

4.4.5. Armazenamento dos Dados Recolhidos

Uma das coisas mais importantes que tem de ser definida na construção da aplicação é a forma como os dados vão ser armazenados e vão estar disponíveis para análise futura. A presente aplicação armazena os dados num ficheiro de texto, com a estrutura presente na Figura 13.

Como se pode observar da figura, o ficheiro no início contém um conjunto de linhas de texto que são como uma espécie de resumo do *scan* que foi realizado. Lá encontra-se descrito o nome do dispositivo de *scan*, a localização onde foi realizado o *scan*, a data e hora de início e fim do *scan*, a duração temporal de todo o conjunto de *scans* para se obter as amostras, o número de dispositivos encontrados (aqui duas detecções do mesmo dispositivo contam como duas detecções diferentes), o número total de *scans* realizados e ainda o tempo em segundos que demorou cada *scan*.

Em seguida a este pequeno resumo, o ficheiro lista todos os dispositivos detectados (amostras). Neste momento, sempre que seja necessário analisar os dados basta utilizar este ficheiro, convertê-lo para um formato que a ferramenta de análise escolhida reconheça, podendo assim serem analisados.

```
::::: ApplicationScanBT Samples File :::::

Scan Device Name:unknown
Scan Location:unknown
Scan Start Date:06-04-2011 15:59:47
Scan End Date:06-04-2011 16:02:40
Scan Duration:0:0:2:53
Num scanned devices:40
Num of Bluetooth scans:34
Time of Discovery:3

-------------- Scanned Devices ----------------

Device Addr:00185D009C59
Device Scan Date:06-04-2011 16:00:10
Device Name:00:18:5D:00:9C:59
Device Class:Computer
Device Num Services:2
Device Services:Capturing, Telephony,

Device Addr:00185D009C59
Device Scan Date:06-04-2011 16:00:14
Device Name:UM-virtual-machine-0
Device Class:Computer
Device Num Services:2
Device Services:Capturing, Telephony,
```

Figura 13 - Exemplo do conteúdo do ficheiro com as amostras recolhidas.

4.4.6. Funcionamento em "Modo Independente" da Aplicação

Se a aplicação estiver a ser executada sem ser parte integrante de outra, ela possui um pequeno interface do tipo texto (estilo linha de comandos), para o utilizador poder controlar e configurar a mesma como desejar. As funcionalidades da mesma estão acessíveis por um pequeno menu, como se pode ver pela Figura 14.

Com a primeira funcionalidade, alteração dos parâmetros de *scan* (*Setting Scan Parameters*), o utilizador pode definir como o *scan* vai ser realizado. O utilizador aqui define o nome do dispositivo que vai realizar o *scan*, o nome do local onde vai ser feito o *scan*, o nome do ficheiro onde vão ficar armazenadas os resultados (amostras) do *scan*, o intervalo tempo de gravação dos resultados e ainda o tempo de cada *scan* para detecção de novos dispositivos.

Figura 14 - Aspecto do menu da aplicação.

A funcionalidade *Begin New Scan*, inicia a realização de um *scan* por tempo indeterminado, ou seja, depois de iniciado um *scan* o mesmo só pode ser parado por acção do utilizador. No momento em que este desejar terminar o *scan*, basta apenas pressionar a tecla Esc.

Por fim, a última funcionalidade (*Begin New Timed Scan*), inicia a realização de um *scan*, como a anterior, no entanto aqui o *scan* para após terminado um período de tempo definido pelo utilizador no início do *scan*.

Esta funcionalidade foi pensada para situações em que se necessite de realizar um *scan* com uma duração específica e o utilizado não possa estar, nessa altura no local do *scan*, para terminar o mesmo. □

5. Implementação e Testes

Como já foi explicada a solução encontrada, para resolver o problema em mãos, é agora dado ênfase á implementação da solução e descrição dos testes que foram realizados com a mesma.

Neste capítulo, é referido como foi instalada a solução nos locais definidos para os testes, e como foram realizados os mesmos. É ainda descrita a forma como os resultados obtidos serão posteriormente analisados, nomeadamente com a explicação de dois processos de análise desenvolvidos.

5.1. Implementação e Cenário de Testes

Na secção 4.1, já foi realizada uma breve apresentação do cenário de teste onde solução iria ser implementada. Ficou decidido que, para o propósito deste projecto, o melhor cenário de teste seria a rua de uma cidade. Neste caso a rua Gil Vicente, da bela cidade de Guimarães. Após se definir qual a rua a utilizar, passou-se à instalação da solução, ou seja, da aplicação de *scan* em três locais da rua. Os locais escolhidos foram três lojas da rua, uma vez que o material a instalar teria de ficar salvaguardado. Foram apenas definidos três locais de recolha de informação dado que, não se encontraram mais lojistas interessados em colaborar com o projecto.

Para a instalação desta aplicação de *scan,* nos três locais, foi decidido também que se iria utilizar como hardware computadores portáteis, e eventualmente caso os mesmos não tivessem hardware Bluetooth de raiz, se recorreria a *dongles* Bluetooth

(ver a secção 4.3.1). Nos três locais ficaram instalados os computadores, aos quais se ligou um cabo USB e na outra extremidade do cabo foi colocado um *dongle* Bluetooth v2.0. Este *dongle* foi estrategicamente posicionado nas montras das lojas, de modo a que o mesmo ficasse o mais próximo possível da rua.

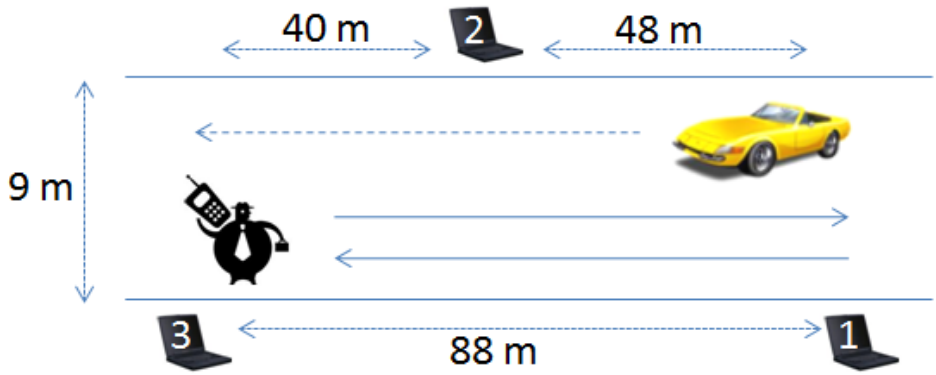


Figura 15 - Esquema do ambiente do teste.

À primeira vista, esta rua pode parecer um local que não apresenta grandes problemas para o que se pretende testar. Contudo, como o ambiente de teste não é "controlado" existem alguns aspectos a ter atenção. Numa rua as pessoas podem-se movimentar livremente, em qualquer direcção, e consequentemente os dispositivos que transportam consigo, o que leva a uma análise posterior dos dados bastante mais complicada. O ideal seria mesmo as pessoas deslocarem-se de forma a que realizassem sempre um único percurso e sempre numa "direcção paralela" aos pontos de recolha, como se pretende explicar com o esquema da Figura 15. Isso acontece para a maior parte dos veículos que se deslocam na rua. Este facto é importante pois assim era quase garantida a passagem das pessoas em todos os locais de detecção.

Mas este não é o único problema neste cenário de teste. Por exemplo, a rua não tem só acesso nas suas extremidades, as pessoas ou veículos também podem surgir em qualquer ponto do cento da rua (repare-se no esquema da Figura 16). A existência de uma rua mais pequena e um parque de automóveis, no centro da mesma faz, com que possam vir a ser detectados dispositivos que não passem por todos os locais de

detecção. Neste caso, estes dispositivos podem não ter potência de sinal suficiente, para serem detectados por todos os pontos de detecção.

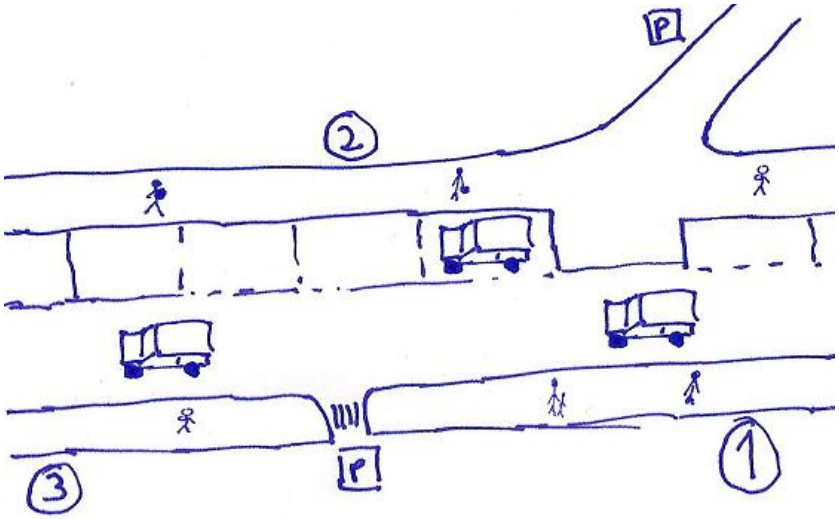


Figura 16 - Croquis da rua Gil Vicente.

Um outro problema que pode acontecer é o de uma pessoa ou veículo, activar ou ligar o seu dispositivo Bluetooth depois de já ter passado por um dos locais de detecção. Estes aspectos têm de ser levados em conta, na análise dos resultados, neste ambiente de teste "não controlado".

Para além dos factores referidos anteriormente, que condicionam a posterior análise dos dados, existe ainda outro conjunto de factores que dificultam as transmissões sem fios (a atenuação, a interferência, o multi-percurso (*Multipath*), o ruído, etc.), ainda para mais num ambiente tão hostil como são as ruas de uma cidade.

A tecnologia Bluetooth, que funciona por intermédio de ondas de rádio, pode até estar imune a alguns destes factores e até ter mecanismos que reduzem o impacto destes nas suas comunicações. Contudo, a tecnologia opera na banda dos 2.4GHz (ver secção 3.2), e todos os sinais electromagnéticos transmitidos nesta banda sofrem grandes atenuações em contacto com as moléculas de agua. Como o cenário de teste é

ao ar livre, as condições atmosféricas (imagine-se um dia de chuva ou nevoeiro) também podem constituir uma "outra barreira" na realização destes testes.

5.2. Testes Realizados

Escolhido o ambiente de teste e implementado o sistema de *scan* de dispositivos era a altura de dar início aos testes. Nesta secção são explicados como foram realizados os testes e qual o objectivo da sua realização. No próximo capítulo serão apresentados os resultados obtidos para cada um dos testes.

Foram realizados vários testes com o propósito de se analisarem vários parâmetros. Apesar de um dos objectivos a alcançar ser analisar o comportamento da tecnologia Bluetooth em ambientes de elevada mobilidade, também é importante saber como se comporta a mesma num ambiente de baixa mobilidade ou mobilidade nula. Esta tecnologia foi projectada e criada para trocar informação entre dispositivos que se encontrassem na proximidade uns dos outros, sendo este o ambiente mais comum de uso da tecnologia. Desta forma, durante a realização dos testes na rua, fez-se um pequeno teste num ambiente sem mobilidade.

5.2.1. Teste sem Mobilidade

Para este teste foi criada uma configuração de teste sem mobilidade, num espaço interior, sem obstáculos entre os diferentes dispositivos como se apresenta na Figura 17. Todos os dispositivos a serem detectados no teste são de classe 2, ou seja têm apenas uma potência de transmissão que permite um raio de operação de 10 metros. A realização deste teste pode ser entendida como um conjunto de três sub-testes (sub-testes 1, 2 e 3), onde se faz variar a duração do tempo de *scan* (tempo de procura por dispositivos), sendo os tempos utilizados: 3, 5 e 10 segundos. A aplicação realizou um determinado número de *scans* por cada sub-teste, para encontrar os dispositivos vizinhos e a duração de cada sub-teste foi de 5 minutos.

Para este teste foi utilizado um computador portátil, para realização dos *scans*, com a aplicação devidamente instalada. Tratando-se este de um ambiente de teste "controlado", foram também ainda utilizados mais dois computadores portáteis, com a interface de Bluetooth activa em modo visível e ainda um *smartphone,* também com a interface activa em modo visível. Estes últimos três dispositivos foram colocados a diferentes distâncias do computador portátil onde está instalada a aplicação, como se vê pela Figura 17. Os sub-testes 1, 2 e 3 tiveram como tempo de *scan* 3, 5 e 10 segundos, respectivamente. Com a realização deste teste era pretendido obter as probabilidades de detecção dos dispositivos, para os diferentes tempos de *scan*.

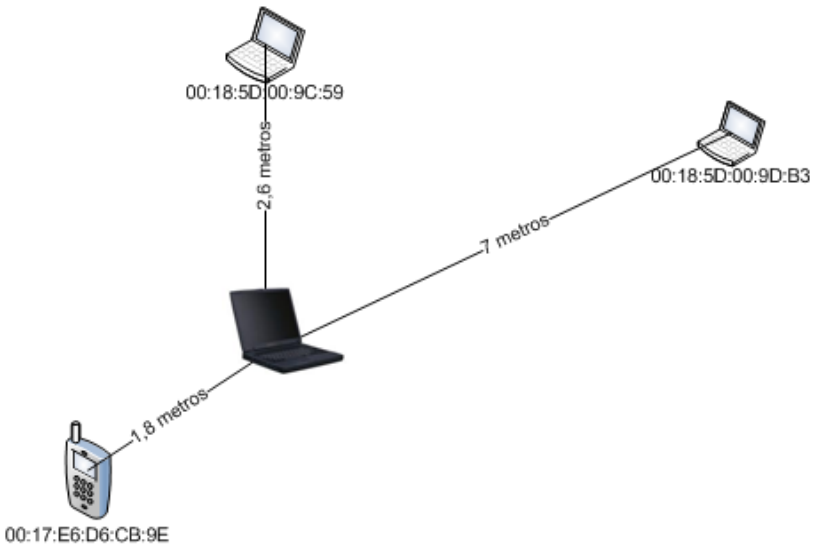


Figura 17 - Cenário do teste sem mobilidade.

5.2.2. Testes com Mobilidade

Após se ter instalado, nos respectivos locais, todo o material para realização dos *scans* era agora possível dar início ao teste. Para este tipo de teste, ou seja, com mobilidade, apenas estava previsto um único teste. No entanto, depois de analisados os resultados deste e os resultados do teste sem mobilidade, achou-se importante realizaram-se mais dois testes neste cenário. O motivo para esta decisão deve-se à diferença obtida na probabilidade de detecção, quando no teste sem mobilidade, se alterou o período de *scan* de dispositivos, de 3 para 5 segundos, como se vai poder

avaliar da análise dos resultados no capítulo 6. Totalizou-se assim um conjunto de três testes num ambiente com mobilidade.

Ficou também definido que todos estes testes, com mobilidade, deveriam ter a duração de cerca de sete dias. Esta longa duração dos testes deve-se ao facto de mais tarde se poder estudar o "padrão de mobilidade" das pessoas que frequentaram a rua. Os testes após serem iniciados, apenas seriam interrompidos passado esse período de tempo de cerca uma semana, de modo a que o estudo a ser realizado fosse o mais fiável possível da realidade presente no local. Convém ainda referir que todos os portáteis em todos os locais de detecção se encontravam sincronizados, ou seja possuíam a mesma data e hora local. Este factor é importante, para mais tarde na análise das amostras não haver discrepâncias temporais, entre locais, nas amostras recolhidas.

Destes três testes, no teste 1 e no teste 2, o tempo de *scan* de dispositivos foi configurado para três segundos e no teste 3 para cinco segundos. O objectivo da realização do teste 2 foi verificar a consistência de resultados em comparação com os resultados do teste 1. Assim, o tempo de descoberta de dispositivos (tempo de *scan*) foi de 3 segundos, para o teste 1 bem como para o teste 2.

No teste 3, foi aumentado este tempo de *scan*, para se estudar se haveria um aumento no número de detecções de cada dispositivo, como ocorreu no teste sem mobilidade. A realização destes três testes também permite verificar se o padrão de mobilidade das pessoas, na rua, é semelhante em períodos de tempo diferentes.

No final da realização destes testes, era pretendido obter as probabilidades de detecção dos dispositivos, realizando uma análise dos dados por local e outra análise global englobando todo o conjunto dos locais de *scan*. Para cada análise foi desenvolvido um processo de análise de resultados, apresentado na próxima secção.

5.3. Processos para Análise dos Resultados

Nesta secção do documento é apresentado todo o processo de análise de resultados para os testes com mobilidade. Estes foram desenvolvidos para tornar a análise das probabilidades de detecção mas fácil de realizar, tentando assim contornar os problemas que o cenário de teste com mobilidade traz (ver a secção 5.1).

Foi definido que seriam utilizados dois processos de análise, um processo de análise definido para realizar o estudo das detecções por local e um outro processo para estudo global das detecções (conjunto dos três locais de *scan*). Para a implementação prática do processo de análise recorreu-se à linguagem de programação *C#*, dado que a mesma já tinha sido utilizada para desenvolver a aplicação de *scan*.

5.3.1. Processo de Análise por Local

Com o processo de análise por local, é pretendido realizar uma análise individual das probabilidades de detecção de dispositivos, para um determinado local.

Foi então desenvolvido um processo de análise, cujo objectivo é obter um valor relativo à probabilidade de detecções falhadas, para um determinado local. Para se obter esta probabilidade, é necessário saber o número de falhas na detecção de todos os dispositivos. Como não se tem cem por cento de certeza de quantos dispositivos passam num local, construiu-se um algoritmo que visa obter dois parâmetros, o número total de detecções e o número total de detecções falhadas.

O algoritmo encontra-se representado na Figura 18. Como entrada de dados temos dois conjuntos de listas de dados: uma lista contém os endereços MAC, de todos dispositivos detectados em todos os locais, e a outra lista com todas as amostras das detecções de todos os locais.

Por cada local de detecção, o algoritmo selecciona, um endereço MAC da primeira lista (lista 1) e vai cruzando essa informação com cada endereço da segunda lista (lista 2), até serem verificados todos os endereços MAC desta primeira lista. Se o

endereço MAC seleccionado da lista 1 coincidir com algum endereço MAC da lista 2, o algoritmo avança para o cálculo da condição de detecção falhada. Se os endereços das listas não coincidirem, é seleccionado um outro endereço MAC da lista 2 e volta-se a verificar a condição anterior.

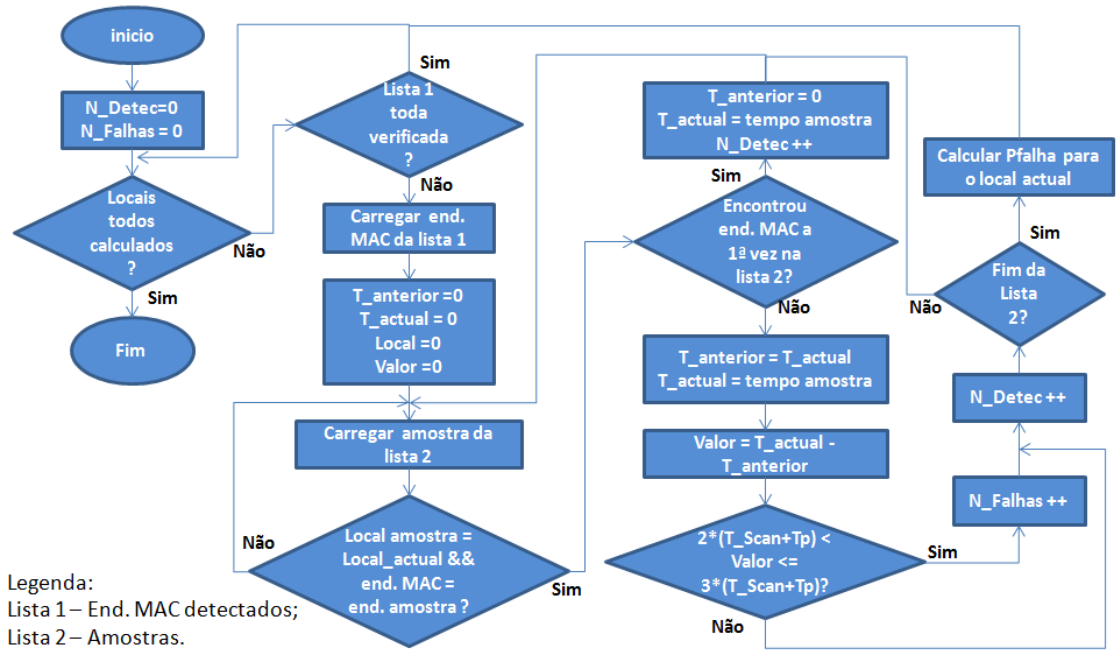


Figura 18 - Fluxograma do processo de análise por local.

A principal dificuldade, na definição deste processo de análise, é saber quando existe uma falha numa detecção. Como já foi explicado, quando um dispositivo é detectado é recolhida um conjunto de informação acerca do mesmo, que se designa de amostra. Um dos parâmetros dessa amostra é data/hora de detecção do dispositivo. Assim sendo, de cada vez que um dispositivo for detectado com um intervalo de tempo entre duas detecções maior do que um determinado valor, está-se perante uma falha na detecção.

O cálculo da condição de detecção falhada, permite encontrar o número de detecções e o número de falhas de detecção, necessários para determinar a probabilidade de falha (P_{Falha}), apresentada na Equação 1. Esta probabilidade é calculada

após serem verificados todos os endereços MAC desta primeira lista, e após todas as amostras para o local de cálculo terem sido verificadas.

$$P_{Falha} = \frac{Falhas}{Falhas + Detecções} \qquad (\boldsymbol{Eq.1})$$

A condição de detecção falhada é descrita por uma expressão matemática, pois a mesma deve levar em conta todo um conjunto de parâmetros. Ocorre uma falha sempre que o intervalo de tempo que decorre entre duas detecções consecutivas ($t_i - t_{i-1}$), do mesmo dispositivo verifica a seguinte condição:

$$2 * (T_{Scan} + T_P) < t_i - t_{i-1} \leq \alpha * (T_{Scan} + T_P) \quad \forall\, \alpha \geq 3 \qquad (\boldsymbol{Eq.2})$$

$t - tempo\ da\ detecção$;

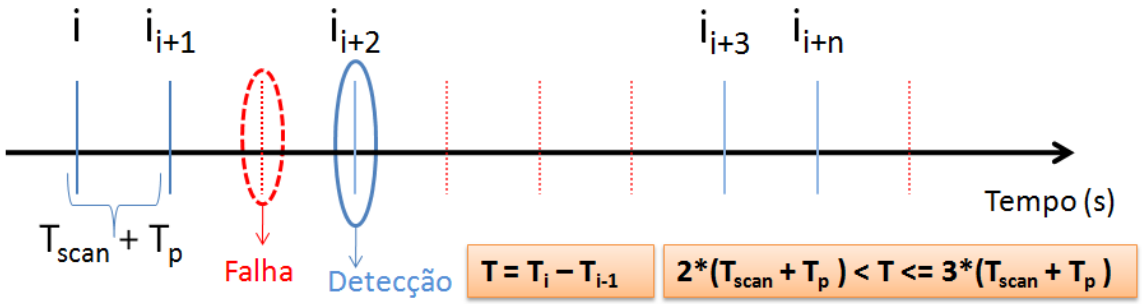
$T_{Scan} -\ tempo\ de\ scan\ dos\ dispositivos$;

$T_p\ -\ tempo\ de\ processamento\ de\ dados$;

$\alpha - limite$ máximo $de\ falha.$

O número total de falhas de detecção, para um dispositivo, é dado pelo número de casos em que o tempo da detecção menos o tempo da detecção anterior, esteja dentro de um intervalo de tempo (Δ). Nesse intervalo de tempo, o valor mínimo é duas vezes o tempo de *scan* adicionado ao tempo de processamento de dados, por parte da aplicação. O valor máximo volta a ser o tempo de *scan* adicionado ao tempo de processamento, multiplicado por um parâmetro α, que representa um valor para o limite máximo de falha. Este parâmetro tem de ser um valor maior ou igual a 3. Este valor máximo garante que Δ seja mínimo, ou seja, este é o "tempo mais curto" que pode ocorrer entre duas detecções, levando assim a que esta condição de detecção seja a mais rígida possível. O parâmetro α, utilizado para os cálculos dos resultados apresentados (no capítulo 6), tem o valor de 3. Contudo, no algoritmo desenvolvido ele é configurável, assim como o número de locais de detecção e o tempo de *scan,* tornado desta forma o algoritmo mais dinâmico, adaptável a outros cenários de teste.

Um Dispositivo – Iteração 1:



$$2 * (T_{Scan} + T_p) < t_i - t_{i-1} \leq \alpha * (T_{Scan} + T_p) \quad \forall \alpha \geq 3$$

Exemplo:

T_{scan} = 3seg T_p = 0.20 seg T_{i-1}= 0 seg T_i= 2seg

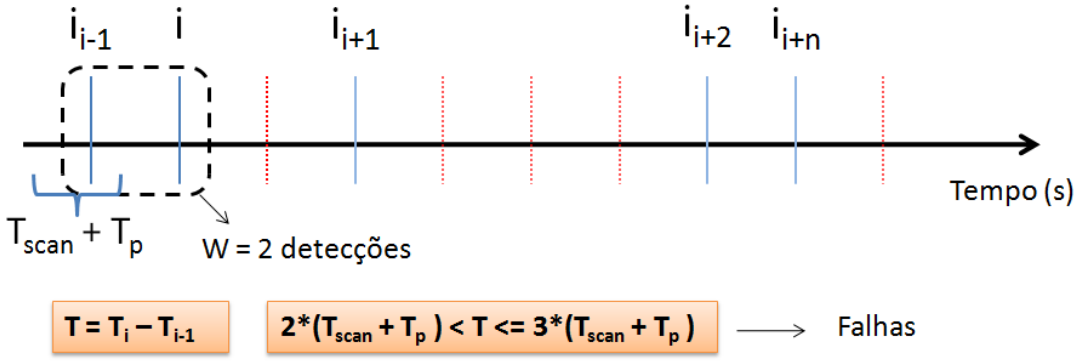
$2*(T_{scan} + T_p) < T <= 3*(T_{scan} + T_p) \Leftrightarrow 6.4 < T <= 9.6$

$T = T_i - T_{i-1} = 2 - 0 =$ 2 seg

Detecções = 1
Falhas = 0

Figura 19 - Esquema do processo de análise por local - Iteração 1.

Um Dispositivo – Iteração 2:



Exemplo:

T_{scan} = 3seg T_p = 0.20 seg T_{i-1}= 2 seg T_i= 4.5 seg

$2*(T_{scan} + T_p) < T <= 3*(T_{scan} + T_p) \Leftrightarrow 6.1 < T <= 9.15$

$T = T_i - T_{i-1} = 4.5 - 2 =$ 2.5 seg
Detecções = 2
Falhas=0

Figura 20- Esquema do processo de análise por local - Iteração 2.

Para uma melhor percepção deste processo de análise são utilizados os esquemas da Figura 19 à Figura 23, onde é simulado o funcionamento do processo, para um dispositivo, num determinado local. Estes esquemas representam um conjunto de iterações do processo para um determinado dispositivo, tendo sido atribuído ao parâmetro α o valor 3, como foi referido anteriormente.

Nos esquemas é assumido que a linha temporal representa as amostras correspondentes á lista 2 (lista das amostras) e que já foi extraído um endereço MAC, da lista 1, para o qual são realizadas as iterações.

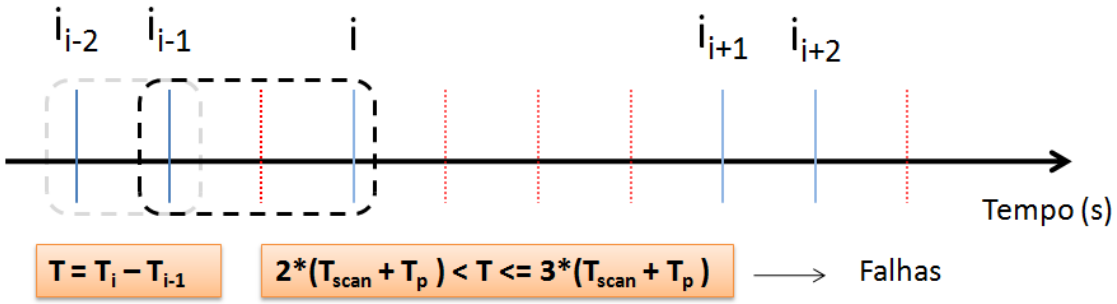
No primeiro esquema, na Figura 19, está esquematizada a primeira iteração. Como se pode ver, no eixo dos tempos estão representadas um conjunto de linhas "a cheio", que representam um conjunto de amostras para o endereço MAC extraído.

Nesta primeira iteração do processo, o número de detecções validas, para o dispositivo em análise é de 1. Uma vez que a primeira amostra (i), representada na linha temporal, correspondente ao endereço a ser encontrado. Desta forma, a primeira detecção não conta como falha, e sendo esta primeira iteração do processo, o número de detecções do dispositivo é igual a um.

Avançando para a segunda iteração, na Figura 20, o mesmo dispositivo, voltou a ser detectado num instante de tempo superior. Aplicando a condição de detecção falhada, verifica-se que o valor da detecção actual menos a anterior encontra-se dentro do intervalo de valores válidos, daí contar como mais uma detecção correcta.

Na próxima iteração (Figura 21), a iteração 3, se aplicarmos a condição de detecção falhada verifica-se que o valor encontrado encontra-se dentro do "intervalo de falha" então, neste caso, podemos afirmar que entre o tempo da detecção actual e o tempo da detecção anterior, existiu uma detecção falhada.

Um Dispositivo – Iteração 3:



Exemplo:

T_{scan} = 3seg T_p = 0.20 seg T_{i-1}= 4.5 seg T_i = 11 seg

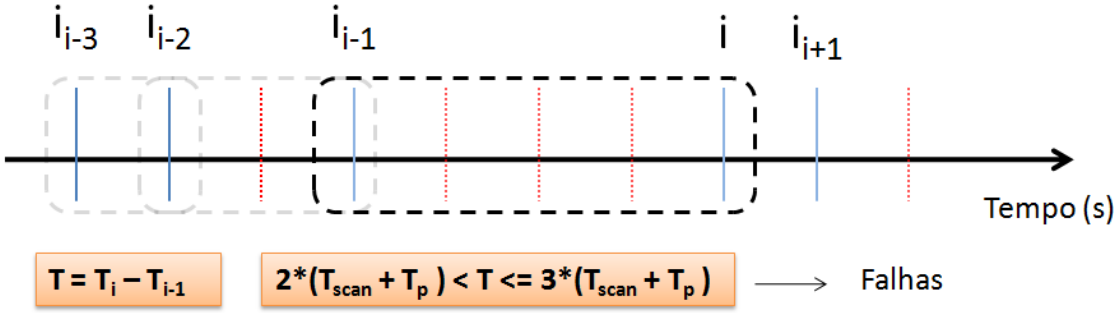
$2*(T_{scan} + T_p) < T <= 3*(T_{scan} + T_p)$ ⇔ **6.1 < T <= 9.15**

T = $T_i - T_{i-1}$ = 11 - 4.5 = **6.5 seg**
Detecções = 3
Falhas=1

FALHA !

Figura 21- Esquema do processo de análise por local - Iteração 3.

Um Dispositivo – Iteração 4:



Exemplo:

T_{scan} = 3seg T_p = 0.20 seg T_{i-1} = 11 seg T_i = 23 seg

$2*(T_{scan} + T_p) < T <= 3*(T_{scan} + T_p)$ ⇔ 6.1 < T <= 9.15

T = $T_i - T_{i-1}$ = 23 – 11 = **12 seg**
Detecções = 4
Falhas = 1

Figura 22- Esquema do processo de análise por local - Iteração 4.

Seguindo para a Figura 22, verifica-se que o valor do tempo de detecção actual menos o tempo de detecção anterior está acima do intervalo de valores, logo também não estamos perante uma falha.

Por fim, na última iteração, Figura 23, o intervalo de valores também é respeitado não apresentando aqui falha alguma.

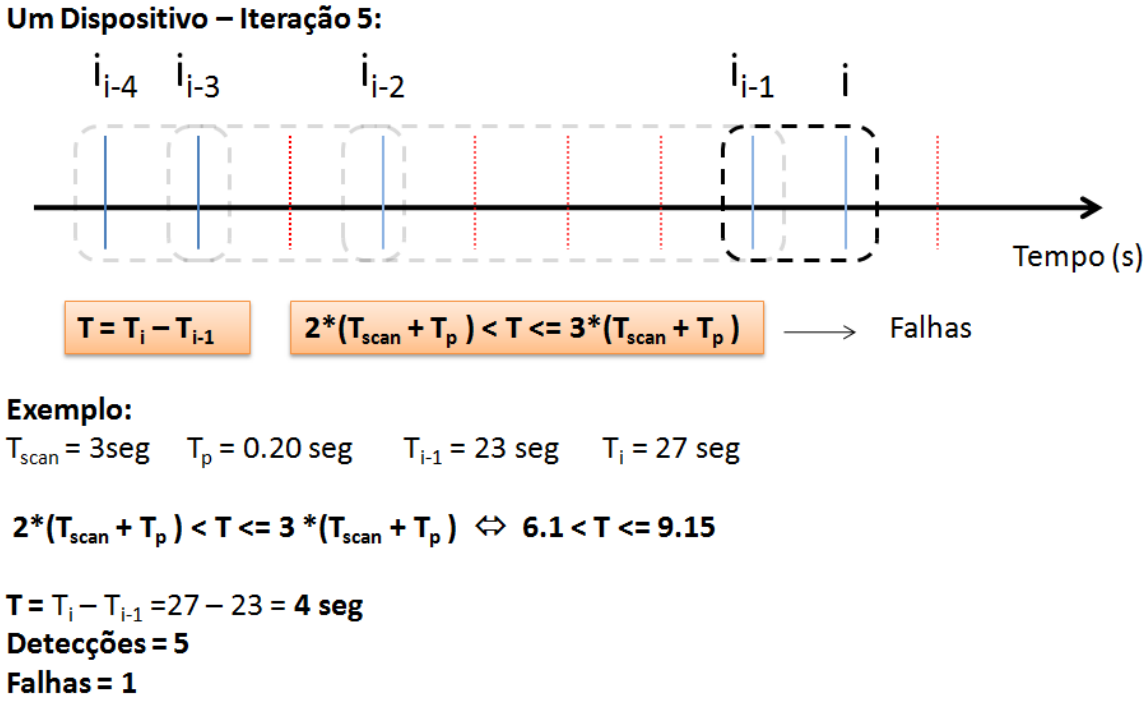


Figura 23- Esquema do processo de análise por local - Iteração 5.

5.3.2. Processo de Análise Global

Este processo de análise destina-se a realizar uma análise global das amostras recolhidas. Com este processo é pretendido obter a probabilidade de detecção de um dispositivo, tendo em conta todos os pontos de detecção utilizados no teste.

No esquema da Figura 24, encontram-se representadas três linhas temporais, cada uma dizendo respeito a um local de detecção. Em cada uma das linhas temporais,

estão assinaladas as detecções de um determinado dispositivo que ocorreram em determinado instante de tempo. Inicialmente o dispositivo A é detectado no local 1 e em seguida nos locais dois e três.

Vamos agora focar a nossa atenção no dispositivo C. Reparando bem, este dispositivo é detectado a meio da linha temporal do local 1, em seguida é detectado no local 2 três vezes consecutivas, e por fim no local 3.

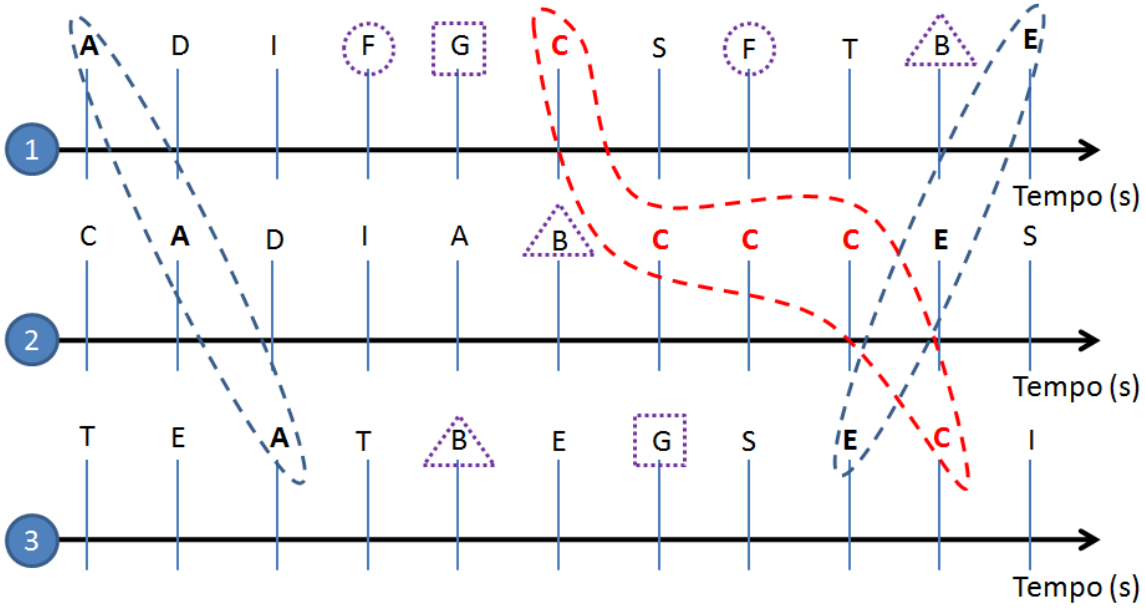


Figura 24 - Exemplo de sequências de detecção 1.

Agora observe-se novamente a Figura 15. Na rua onde foram realizados os testes, o único sentido que os veículos podem tomar é do local de detecção 1 ao local de detecção 3. Já as pessoas podem tomar este sentido ou o inverso, ou seja, do local de detecção 3 para o local de detecção 1. Para uma detecção poder ser considerada com cem por cento de sucesso, os dispositivos tem de ser detectados em todos os pontos de detecção (que neste caso em particular eram 3). Contudo, pode haver casos onde o dispositivo não seja detectado em todos os locais e no entanto correspondeu a uma pessoa ou veículo que passou pela rua. Voltando a visualizar a Figura 24, verifica-se que os dispositivos F e G também foram detectados mas não em todos os locais, como os

dispositivos A, C e E. Desta figura pode-se ainda concluir que o dispositivo E não seria um veículo, pois ele foi detectado inicialmente no local 3, depois no 2 e por fim no 1. Os veículos não podem circular neste sentido.

Sendo assim, é necessário definir quando um dispositivo foi detectado com sucesso ou não. Para tal foram definidos dois tipos de sequências de detecção, as sequências de detecção completas e incompletas.

Uma sequência de detecção diz-se completa, quando o dispositivo é detectado em todos os locais, de forma sequencial (caso dos dispositivos A e E) ou ordenada (caso do dispositivo C). Já os dispositivos F e G realizaram sequências de detecção incompletas.

O número de sequências totais possíveis que podem ocorrer, para os testes realizados, ou seja, com três locais de detecção, ainda é considerável. Todas as sequências que podem ocorrer encontram-se representadas pelo diagrama de árvore, da Figura 25.

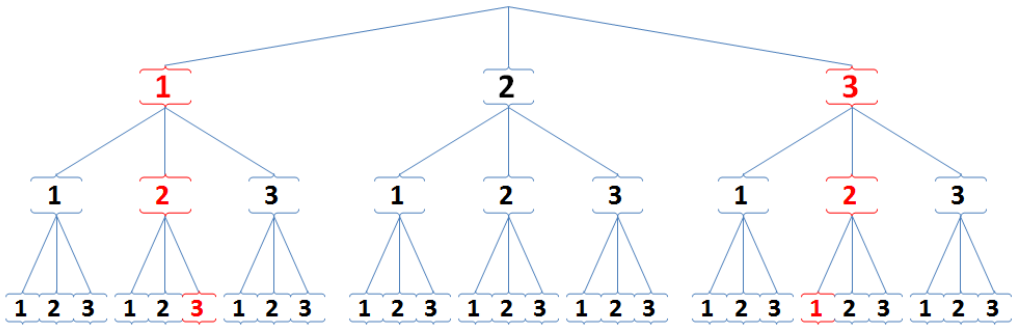


Figura 25 - Diagrama de Árvore Para Todas as Sequências de Detecção.

Contudo, nem todas as sequências apresentadas na Figura 25, são aceites como válidas, como por exemplo a sequência (1→1→1). Em particular, para estes testes realizados as sequências de detecção completas são todas as sequências do tipo: (1 → 2 → 3) ou (3 → 2 → 1).

De forma a reduzir os casos de sequências incompletas de detecção, é assumido que nenhum dispositivo pode ser inicialmente detectado no centro da rua, sem ter sido detectado antes nos pontos 1 ou 3, ou seja, nos pontos de detecção das extremidades da rua. As sequências incompletas, para o teste realizado são as seguintes: (1→3), (1→2), (3→2) e (3→1).

Em todas as outras sequências que não se verifique o que foi referido, as mesmas não são consideradas para este processo de análise.

Reparando agora em outro dispositivo da Figura 24, o dispositivo B, também se está perante um caso de sequência de detecção completa, mas desta vez a mesma não deve ser considerada válida. Recordando mais uma vez à Figura 15, observe-se as distâncias de um local a outro. Como a maior distância entre locais é de 48 metros, que correspondente à distância do local 1 ao local 2, esta vai ser a nossa distância de referência. Assumindo que uma pessoa se desloque a 2 m/s, facilmente se verifica que para uma pessoa se deslocar de um local a outro, leva cerca de 24 segundos. Vamos designar este tempo de Δt.

Sabido isto, considere-se um dispositivo que foi detectado num local e só passado um elevado período de tempo foi detectado em outro, continuando o mesmo a realizar uma sequência de detecção completa. A questão que se coloca aqui é se esta sequência, que levou um maior período de tempo a ser completa, deve ser considerada válida. A resposta para a questão é não. Esta sequência não deve ser considerada como válida.

Este dispositivo B foi detectado no local 3 e no local 2, a detecção no local 1 só ocorre muito mais tarde. Isto pode indicar, por exemplo, que o dispositivo passou uma primeira vez no local do teste (passagem no sentido de 3 para 1), não sendo detectado no local 1, e só após uma segunda passagem pelo local do teste, ter sido detectado no local 1.

Para se evitar situações deste tipo, o tempo de detecção de um dispositivo entre dois locais, terá de obedecer à seguinte condição:

$$T_{detecção\ entre\ 2\ locais} \leq \Delta t + T_{Scan} + T_P \qquad (\boldsymbol{Eq.3})$$

$\Delta t - tempo\ máximo\ para\ uma\ pessoa/veículo, se\ deslocar\ de\ um\ local\ a\ outro;$

$T_{Scan} -\ tempo\ de\ scan\ dos\ dispositivos;$

$T_p\ -\ tempo\ de\ processamento\ de\ dados.$

Na condição anterior, para cálculo dos testes realizados, foi definido o valor de dois segundos para o T_P.

Examinemos agora na Figura 26 o dispositivo B. A detecção deste dispositivo do local 3 para o local 2, obedece à condição anterior, até aqui tudo bem. Contudo, a detecção do mesmo dispositivo do local 2 para o local 1 não respeita a condição anterior. Assim sendo não se trata de uma sequência de detecção completa. Em vez disso é assumido, pelo processo de análise, como uma sequência de detecção incompleta, neste caso a sequência incompleta (3→2).

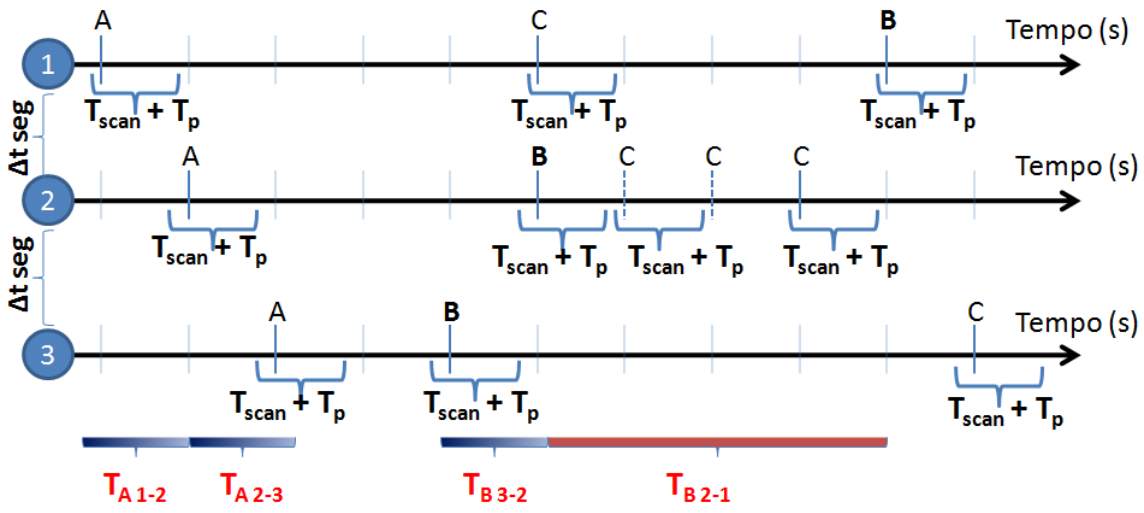


Figura 26 - Exemplo de Sequências de Detecção 2.

O algoritmo de análise para este processo encontra-se representado pelo fluxograma da Figura 27. Neste processo de análise, inicialmente é calculado o número de sequências completas e o número de sequências incompletas, onde posteriormente é calculada a probabilidade de detecção, através da seguinte equação:

$$P_{Dectecção} = \frac{Num.Seq.Completas}{Num.Seq.Completas + Num.Seq.Incompletas} \qquad (\boldsymbol{Eq.4})$$

Para a obtenção do número de sequências completas e incompletas foram elaborados dois fluxogramas, reapresentados nas Figura 28 e Figura 29. Estes dois fluxogramas, como já aconteceu no caso do processo de análise por local, têm como entrada de dados duas listas: uma lista contém os endereços MAC de todos os dispositivos detectados e outra lista todas as amostras das detecções realizadas. O algoritmo o que faz é "cruzar os dados" de uma e de outra lista, onde no final obtêm-se os valores para o número de sequências completas, no caso do primeiro algoritmo, e para o número de sequências incompletas no caso do segundo.

Este algoritmo do processo de análise global foi também implementado em linguagem *C#*. O algoritmo é dinâmico, ou seja, adaptável à mudança do número de locais de teste, bem como dos valores dos parâmetros da equação 3 (Eq.3).

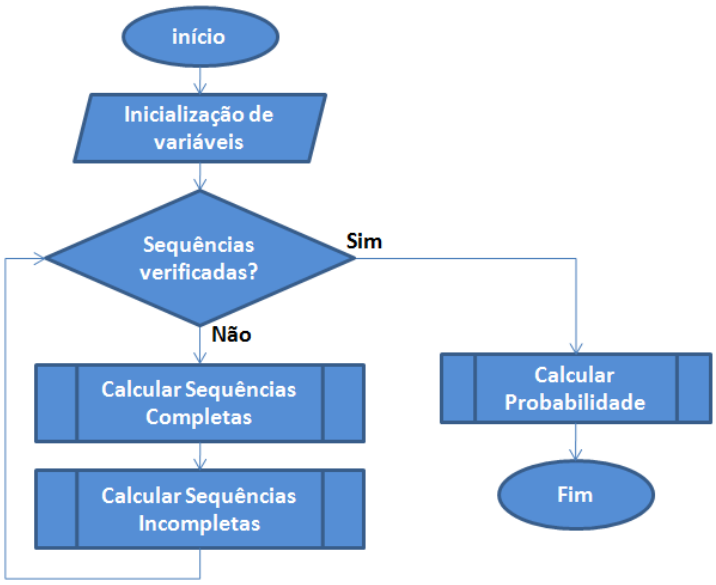


Figura 27 – Fluxograma do processo de análise global.

Analisando o fluxograma da Figura 28, podemos ver que o algoritmo vai recolhendo um endereço MAC detectado e verificando quando o mesmo aparece na lista de amostras. Ao mesmo tempo vai actualizando um conjunto de quatro variáveis que dizem respeito ao dispositivo detectado. Nestas, duas delas representam o tempo de detecção do presente na amostra anterior e o tempo de detecção presente na amostra actual. As outras duas representam o local de detecção da amostra anterior e o local de detecção da amostra actual.

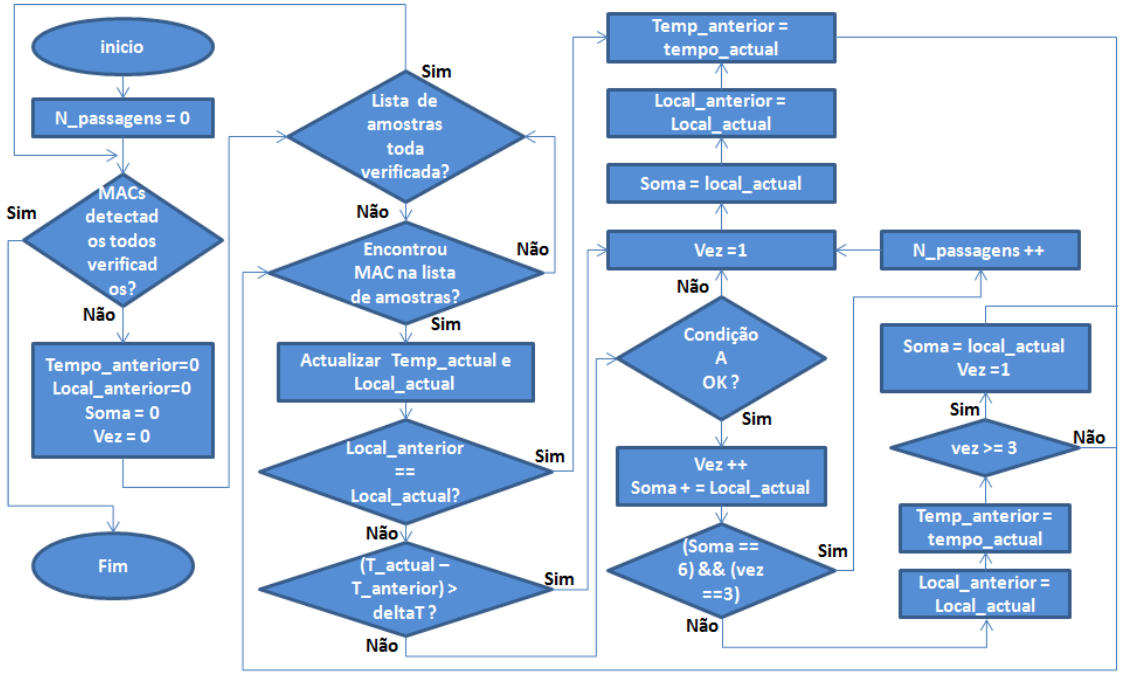


Figura 28 – Fluxograma do processo de análise global para as sequências completas.

Está ainda presente neste algoritmo, a variável "*soma*" e a variável "*vez*". Na variável "*soma*", é armazenada a soma dos locais onde o dispositivo foi detectado. Para estes testes em particular, "*soma*" não deve ultrapassar o valor máximo de seis. Pois cada local é identificado com um número, como aqui existiam 3 locais: 1+2+3=6 ou 3+2+1=6. Estas "somas" representam uma sequência de detecção completa. Sempre que o valor máximo seja ultrapassado, o valor de soma é actualizado para o número do local actual.

Por sua vez, só a variável "*soma*" não chegaria para fazer um controlo correcto das sequências de detecção válidas. A variável "*vez*" também é necessária na execução do objectivo deste algoritmo. Se analisarmos bem um dispositivo que realizasse, por exemplo, a sequência (1→2→1→2), sempre que a equação 3 fosse verificada também se obteria para "*soma*", o valor de seis. No entanto, neste caso não se está perante uma sequência de detecção completa. Deste modo, sempre que a condição A se verificar, a variável "*vez*", é incrementada. Se o valor da variável "*vez*" atinge o valor três, estamos perante uma sequência de detecção completa, que pode ser considerada. Na prática, esta variável "*vez*", indica-nos o número de locais diferentes para os quais o dispositivo já foi detectado de modo a realizar uma sequência completa.

A condição A não se encontra representada no fluxograma da Figura 28, mas é apresentada de seguida:

$$\left((\boldsymbol{local}_{\boldsymbol{actual}} > local_{\boldsymbol{anterior}}) \cap (\boldsymbol{local}_{\boldsymbol{actual}} - \mathbf{1} = \boldsymbol{local}_{\boldsymbol{anterior}})\right) \cup \left((\boldsymbol{local}_{\boldsymbol{actual}} < local_{\boldsymbol{anterior}}) \cap (\boldsymbol{local}_{\boldsymbol{anterior}} - \mathbf{1} = \boldsymbol{local}_{\boldsymbol{actual}})\right)$$

A primeira parte da condição ocorre quando um dispositivo realiza o percurso do local 1 para o local 3 e a segunda parte, quando o dispositivo realiza o percurso inverso.

Para a análise das sequências incompletas, o algoritmo elaborado, representado na Figura 29, é semelhante ao algoritmo das sequências completas. Neste algoritmo, também para cada endereço MAC que foi detectado, é verificada a sua existência na lista de amostras. Cada vez que o mesmo for encontrado, é avaliada a condição da equação 3. Isto é possível por intermédio do mesmo conjunto de quatro variáveis, as variáveis dos tempos de detecção anterior e actual e ainda as variáveis do local actual e local anterior de detecção. Sempre que a condição da Equação 3 não se verifique, o algoritmo testa se a última detecção do dispositivo foi feita num dos locais da periferia do teste. No caso dos testes elaborados, no local 1 ou no local 3.

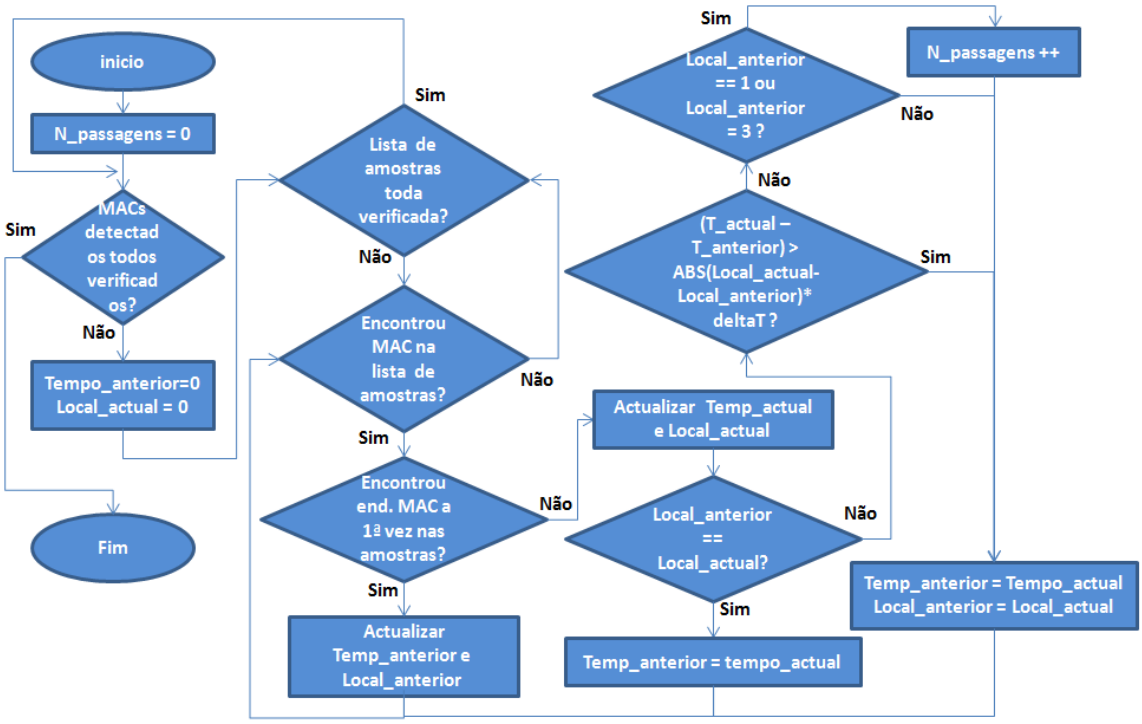


Figura 29- Fluxograma do processo de análise global para as sequências incompletas.

Assim sendo este algoritmo também demonstra alguma dinâmica e flexibilidade de cálculo, uma vez que se adapta facilmente a um teste com mais locais de detecção.

Este algoritmo desenvolvido para as sequências incompletas, se analisado de perto, verifica-se que devolve um valor para o número total de passagens que os dispositivos realizaram pelo local de teste. O número de passagens, não é nada mais do que o número de sequências completas e incompletas realizadas por todos os dispositivos. Se repararmos num outro aspecto, quer as sequências completas, quer as sequências incompletas têm algo em comum, a detecção de um dispositivo em pelo menos um local da periferia. No caso das sequências incompletas os dispositivos podem ser detectados em um local (por exemplo, sequência (1→2)) ou nos dois locais da periferia (por exemplo, sequência (1→3)). Já para as sequências completas os dois locais da periferia estão sempre presentes.

Assim, no final de se obter o número de passagens, para se obter o número de sequências incompletas é necessário subtrair, ao número de passagens, o número de sequências completas.

Neste ponto, após a execução dos dois algoritmos e de se saber os valores para o número de sequências completas e incompletas, pode agora ser calculada a probabilidade de detecção de acordo com a equação 4.

Para o próximo capítulo está reservada a apresentação dos resultados obtidos para todos os testes realizados, onde foram usados para análise dos mesmos todos estes algoritmos apresentados nesta secção do documento.

6. Resultados Obtidos

Após apresentada a descrição de todos os testes realizados e os processos de análise desenvolvidos, vamos dar a conhecer os resultados que se obtiveram na sua realização.

Como foi explicado, o principal objectivo a alcançar é determinar a probabilidade de detecção de um dispositivo, que use a tecnologia Bluetooth num ambiente de elevada mobilidade. A par deste objectivo também é pretendido retirar conclusões acerca da distribuição geográfica e temporal das pessoas num determinado local. Para tal foi encontrada uma solução e realizado um conjunto de quatro testes. Um dos testes foi feito num ambiente sem mobilidade e os restantes três num ambiente de elevada mobilidade, na rua Gil Vicente da cidade de Guimarães.

Este capítulo esta dividido em duas secções principais onde a primeira relata os resultados do teste sem mobilidade e a segunda os resultados dos testes com mobilidade.

6.1. Teste sem mobilidade

Este teste foi realizado num ambiente interior, onde os dispositivos não ofereciam mobilidade. Foi colocado um dispositivo (computador portátil) num espaço amplo, realizando, de forma contínua, um conjunto de *scans*. O objectivo dos *scans* é obter a probabilidade de detecção de outros três dispositivos, que se encontram nas proximidades, como se pode ver na Figura 17, na secção 5.2.1. Este teste foi dividido por

um grupo de três sub-testes (sub-testes 1, 2 e 3), onde se fez variar a duração do tempo de *scan*.

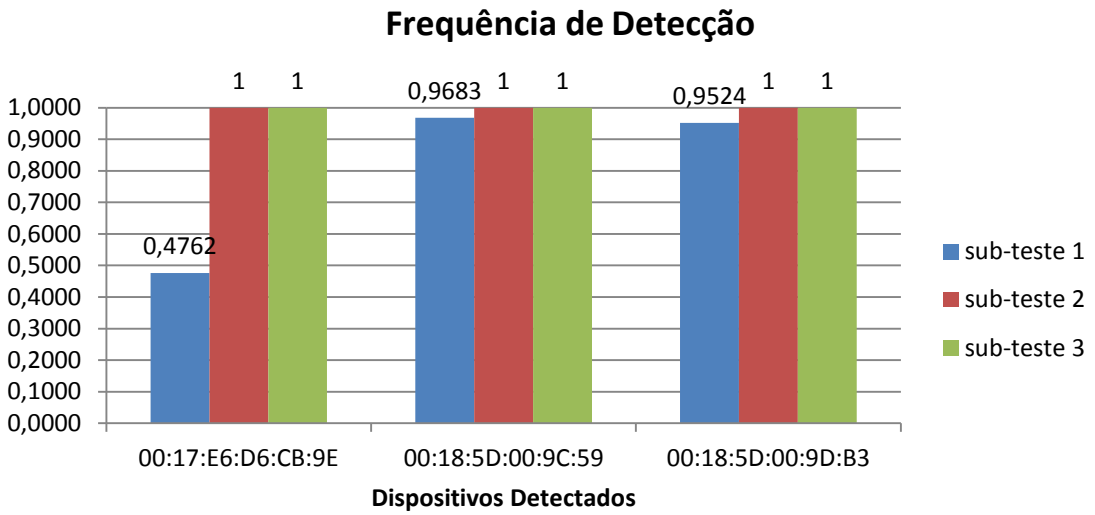


Gráfico 1 - Frequência de detecção em ambiente sem mobilidade.

Como se pode observar no Gráfico 1, onde se encontram representadas as frequências de detecção para cada um dos três dispositivos, sendo os valores normalizados, todos os dispositivos são detectados quase tantas vezes como o número de *scans* realizados.

Feita a análise do Gráfico 1, verifica-se uma excepção no sub-teste 1. O dispositivo com o endereço MAC "00:17:E6:D6:CB:9E" é detectado cerca de metade das vezes do que seria de esperar. Esta excepção pode ter ocorrido devido ao tempo curto de *scan* no sub-teste 1, que é de três segundos.

Ao compararmos os outros dois dispositivos, também se vê que existe uma ligeira diferença, nos resultados para o sub-teste 1. O dispositivo com o endereço MAC 00:18:5D:00:9C:59, foi detectado mais vezes que o dispositivo com o endereço MAC 00:18:5D:00:9D:B3.

Sub-Teste	Tempo de *scan* (Seg.)	Nº de *scans*	Nº de Detecções Total	Probabilidade De Detecção (%)
1	3	63	160	84,66
2	5	54	162	100,00
3	10	27	81	100,00

Tabela 2 – Resultados do teste sem mobilidade.

Esta diferença nas detecções, deve-se ao facto da distância a que os mesmos se encontram do "ponto de *scan*", uma vez que o dispositivo com o endereço 00:18:5D:00:9C:59 é detectado mais vezes e se encontra mais próximo deste. Nos outros dois sub-testes realizados o número de detecções é sempre igual entre os três dispositivos. Outro aspecto a referir é que com a diminuição do tempo de *scan,* diminui-se o número de detecções para cada dispositivo, como se esperava.

Do resultado deste teste conclui-se que o ideal será ter um tempo de *scan* configurado por volta dos 5 segundos, para que possam ser detectados com aproximadamente 100% de sucesso, todos os dispositivos. Esta análise dos resultados foi o que motivou a realização de mais dois "testes de campo".

Na Tabela 2 pode observar-se o número total de detecções para cada sub-teste e ainda a probabilidade de detecção obtida para cada um. A probabilidade de detecção foi determinada pela seguinte equação:

$$P_{Detecção} = \frac{Num.Detecções}{Num.Scans \times Num.Dispositivos} \qquad (Eq.5)$$

Como era, ainda assim, de esperar o teste 1 obteve a probabilidade de detecção mais reduzia, apresentando um valor bem satisfatório. Da tabela verifica-se ainda que o sub-teste 2 obtém o número mais elevado de detecções.

6.2. Testes com elevada mobilidade

Relativamente aos testes com mobilidade, tiveram como cenário uma rua da cidade de Guimarães. Foram colocados, em três pontos da rua, um conjunto de computadores portáteis que realizaram um conjunto de *scans* dos dispositivos que passavam pelos locais (ver a Figura 15).

Neste ambiente, foram realizados um conjunto de três testes. Estes foram realizados em períodos de tempo diferentes com a duração de cerca de sete dias. Na Tabela 3 pode-se visualizar as informações gerais, acerca dos mesmos. O terceiro teste teve a duração de menos um dia e a duração do intervalo de *scan* de dispositivos foi de três segundos nos testes 1 e 2, e de cinco segundos no teste 3.

Os resultados dos testes vão ser descritos detalhadamente, elaborando-se uma secção para cada tipo de teste.

Teste	Duração	Data de Início	Data de Fim	Intervalo de *Scan*	Nº de Amostras	Nº Disp. Detectados
1	7 Dias	23-03-2011	30-03-2011	3 Seg.	73288	3800
2	7 Dias	29-06-2011	06-07-2011	3 Seg.	76285	4131
3	6 Dias	09-07-2011	16-07-2011	5 Seg.	57653	3072

Tabela 3 - Informações Gerais dos 3 Testes Realizados.

Para a análise destes resultados, foram definidas três categorias de estudo: Geral – esta categoria engloba todos os dispositivos detectados; Veículos – esta categoria engloba todos os "veículos" detectados; Pessoas – classe que engloba todas as "pessoas" detectadas. Na Tabela 4 encontra-se um resumo dos dispositivos detectados, por categoria.

Como não é possível identificar na informação recolhida (relativa às classes dos dispositivos), uma classe de dispositivos do tipo veículo, foi assumido que um dispositivo

pertence a esta categoria caso a sua classe seja uma das três seguintes: *AudioVideoHandsFree*, *AudioVideoHeadset* e *AudioVideoVcr*. Todas as restantes classes de dispositivos detectadas foram assumidas como pertencendo à categoria do tipo "pessoas".

Teste	Duração	Intervalo de *Scan*	Nº Disp. Detectados	Nº Veículos Detectados	Nº Pessoas Detectadas
1	7 Dias	3 Seg.	3800	836	2964
2	7 Dias	3 Seg.	4131	824	3307
3	6 Dias	5 Seg.	3072	518	2554

Tabela 4 - Informações Sobre os Dispositivos Detectados.

6.2.1. Teste 1 – Análise de Resultados

Uma das primeiras conclusões a retirar da análise de resultados do teste 1, é que a maioria dos dispositivos encontrados apenas foi detectada de uma a três vezes. Isso é evidente pela observação dos gráficos 2 ao 7.

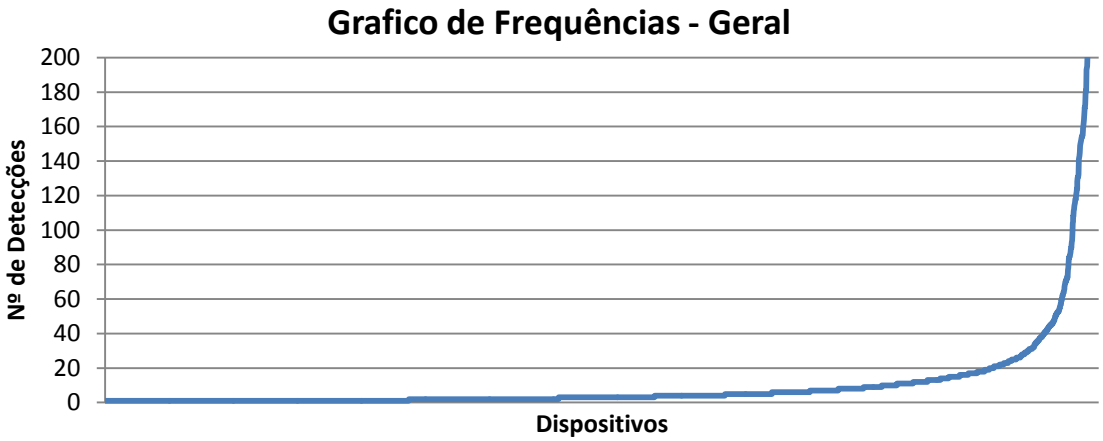


Gráfico 2 - Teste1: Frequências de Detecção – Geral.

Analisando o Gráfico 2 verifica-se um crescimento mais lento da curva de frequências no início até cerca das 25 detecções, onde depois o seu crescimento

aumenta mais rapidamente. Isto acontece devido ao elevado número de dispositivos que foram detectados menos do que 25 vezes.

O mesmo comportamento verifica-se para o estudo da categoria dos veículos (Gráfico 3) e das pessoas (Gráfico 4). Para os veículos, o crescimento da curva é bastante lento no início, e a partir das 20 detecções também aumenta mais rapidamente. Já no caso da categoria das pessoas, só por volta das 30 detecções é que o crescimento da curva é mais acentuado.

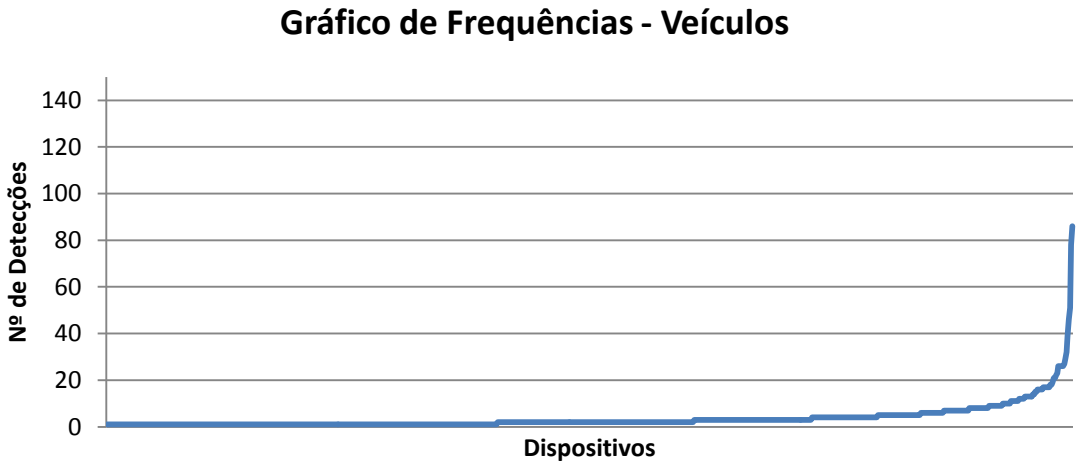


Gráfico 3 - Teste1: Frequências de Detecção – Veículos.

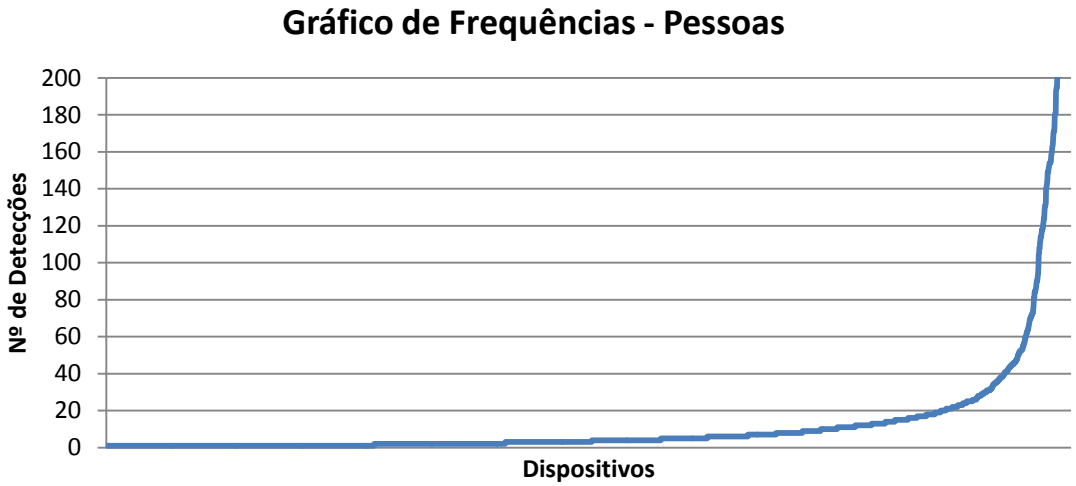


Gráfico 4 - Teste 1: Frequências de detecção - Pessoas.

Esta visualização torna-se mais clara observando o Gráfico 5, Gráfico 6 e Gráfico 7, onde as frequências de detecção estão agrupadas num histograma por classes. Aqui optou-se por "partir" a classe "1-5" em valores individuais, uma vez que a grande parte dos dispositivos detectados pertencia a esta classe.

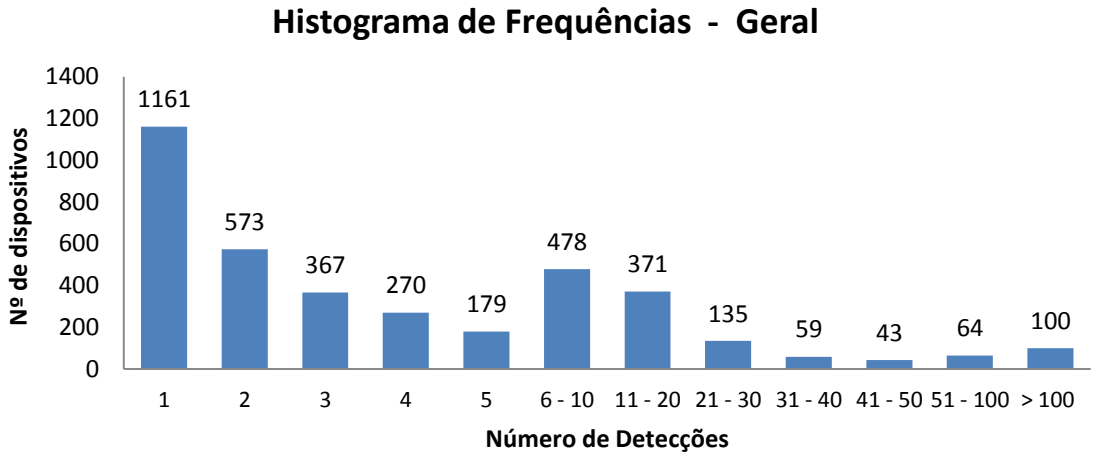


Gráfico 5 - Teste 1: Histograma de Frequências de Detecção – Geral.

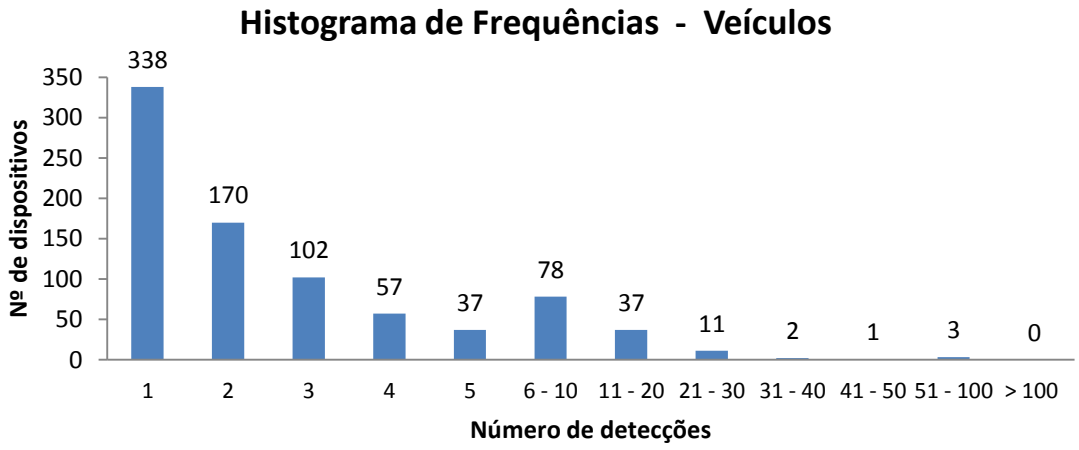


Gráfico 6 - Teste 1: Histograma de Frequências de Detecção - Veículos.

Dos 3800 dispositivos diferentes detectados neste teste, 1161 foram detectados uma vez, ou seja, 2639 foram detectados mais de uma vez o que representa cerca de 69,4% de todas as detecções, um valor bastante aceitável. No Gráfico 5, verifica-se também que o número de dispositivos detectados vai diminuindo em relação ao número

de detecções. Um aspecto interessante a realçar, que não era bem visível no Gráfico 2, é o facto de 100 dispositivos terem sido detectados mais de 100 vezes, ainda que todos correspondam à categoria das pessoas.

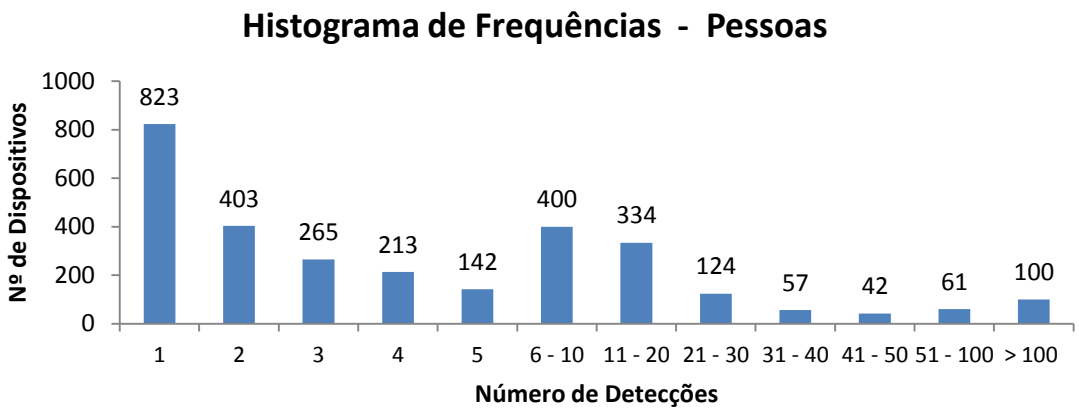


Gráfico 7 - Teste 1: Histograma de Frequências de Detecção – Pessoas.

Uma outra análise elaborada aos resultados deste teste foi relativamente ao número de detecções totais por dia. Visualizando o Gráfico 8 constata-se que as pessoas tem uma afluência à rua de forma relativamente constante, independentemente do dia da semana.

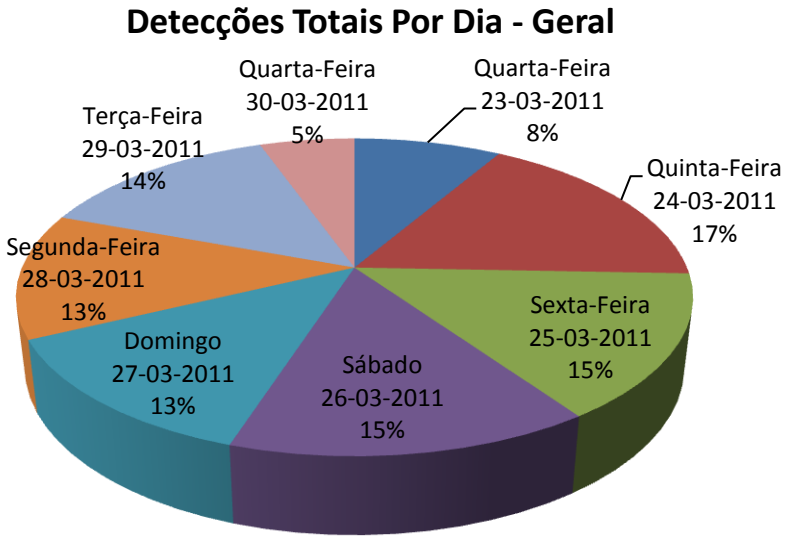


Gráfico 8 - Teste 1: Detecções Totais por Dia.

A realização das detecções começou numa quarta-feira às 12h20m tendo terminado o teste na quarta-feira da semana seguinte à mesma hora. Sendo assim, e considerando as amostras detectadas no dia 23 e no dia 30 como um único dia, verifica-se que há uma distribuição equivalente de resultados. No entanto, na quinta-feira dia 24, foi o dia em que mais dispositivos se detectaram na rua.

A título de curiosidade, para este primeiro teste, foram também calculados o número de veículos diferentes detectados por dia. Comparando os resultados obtidos com os do gráfico anterior os mesmos não diferem muito. Mas na Terça-Feira dia 29, registou-se o maior número de detecções para veículos diferentes.

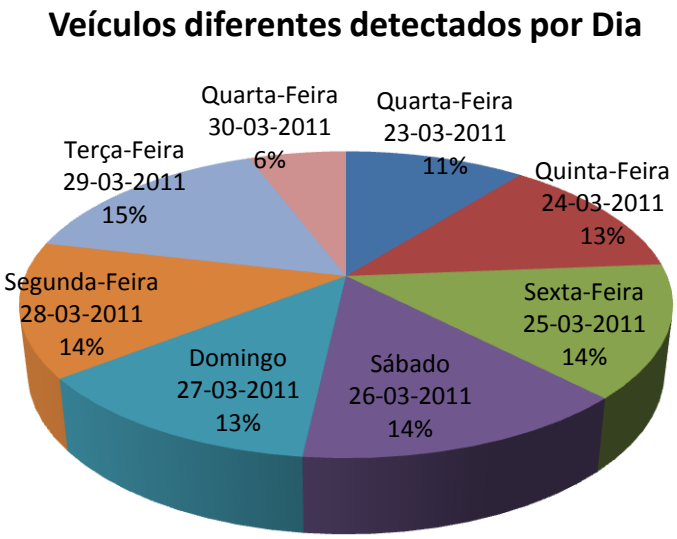


Gráfico 9 - Teste 1: Detecções Diferentes por Dia – Veículos.

Destes dois últimos gráficos que foram apresentados podem ser conhecidos "dois padrões " de mobilidade para esta rua. Como o Gráfico 8 representa o número total de detecções, pode-se afirmar que na quinta-feira foi o dia onde as pessoas passaram mais tempo na rua. Se fosse feito outro estudo, semelhante ao do Gráfico 8, mas desta vez calculando o numero de detecções individuais, poderia descobrir-se também qual o dia onde mais pessoas estiveram na rua. Isso foi feito para o Gráfico 9, mas para a classe dos veículos. Da análise deste gráfico, pode-se afirmar que a terça-feira é dia em que a rua teve mais movimento, ao nível dos veículos. Claro está que

estes “dois padrões” decifrados, não provam, por exemplo, que a rua tem mais tráfego automóvel à terça-feira. Para tal teriam de ser realizados testes consecutivos com uma duração maior.

No Gráfico 10 encontra-se a distribuição de todas as detecções realizadas por hora, acumuladas ao longo da semana. Analisando o gráfico, vê-se perfeitamente que existe um “padrão” que comunga com o que seria expectável. Se olharmos para as primeiras horas do dia, observamos um número de detecções bastante reduzido, à excepção da primeira hora do dia que ultrapassou as 1000 detecções. A estas horas a maioria do comércio encontra-se fechado. Depois, no início da manhã, sobe o número de detecções a partir das 7h00 até cerca das 10h00, e a partir desta hora o número estabiliza nas 5000 detecções. Durante a tarde o número cai ligeiramente e estabiliza por volta das 4500 detecções por hora. No final da tarde, e até as 20h00 o número de detecções decresce de forma acentuada passando das 5358 para as 1779. No entanto, no início da noite, e até às 23h00, voltam a subir obtendo-se aqui os valores mais altos do dia, para depois das 23h00 decrescerem acentuadamente, como no final da tarde.

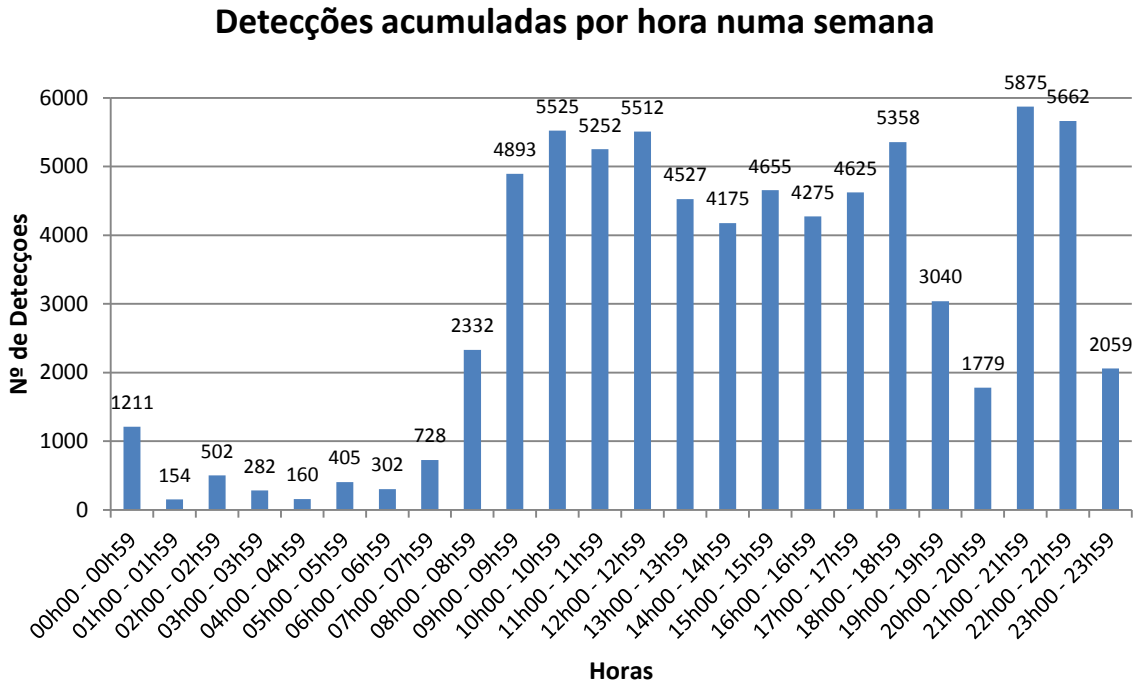


Gráfico 10 - Teste1: Detecções Acumuladas por Hora Numa Semana.

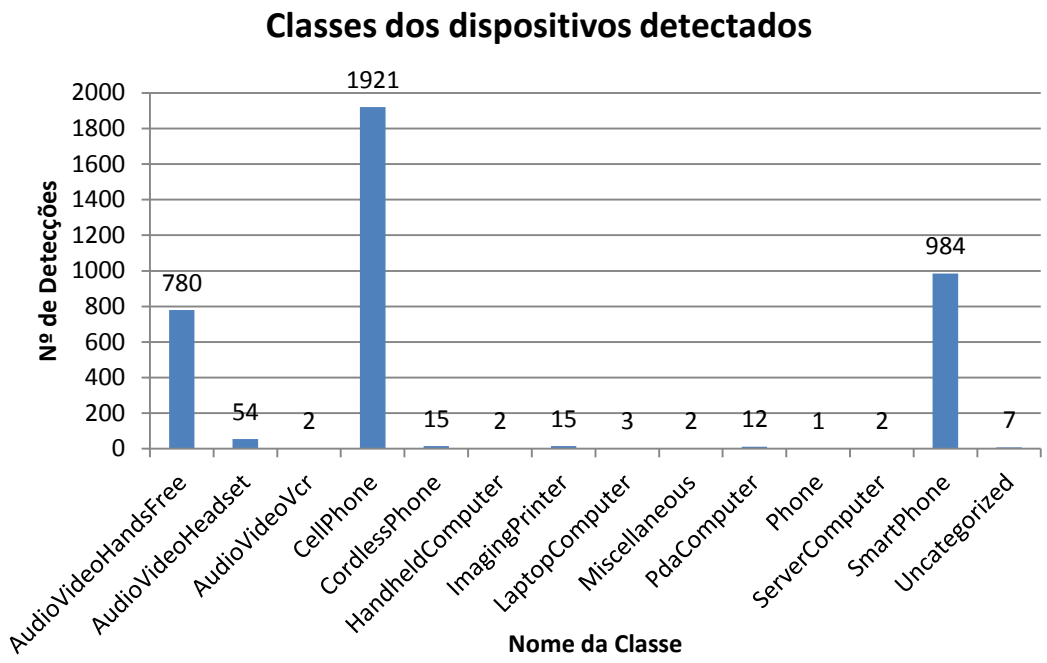


Gráfico 11 - Teste1: Classes dos Dispositivos Detectados.

As outras duas análises realizadas dizem respeito ao tipo de dispositivos encontrados e ao número de serviços presentes nos mesmos. Observando o Gráfico 11 verifica-se que três classes de dispositivos dominam sobre todas as outras, sendo elas: *AudioVideoHandsFree*, *CellPhone* e ainda *SmartPhone*.

Estes resultados também não são de estranhar pois são os dispositivos mais portáveis e cómodos, que as pessoas podem utilizar num ambiente de elevada mobilidade, como são o caso das ruas de uma cidade.

Relativamente ao número de serviços Bluetooth encontrados nos dispositivos, da visualização do Gráfico 12 conclui-se que a maior parte dos mesmos, ou seja 54% possui 4 serviços disponíveis, enquanto apenas 2% dos mesmos possuem 5 serviços. Não foram detectados dispositivos com mais de 5 serviços neste teste 1.

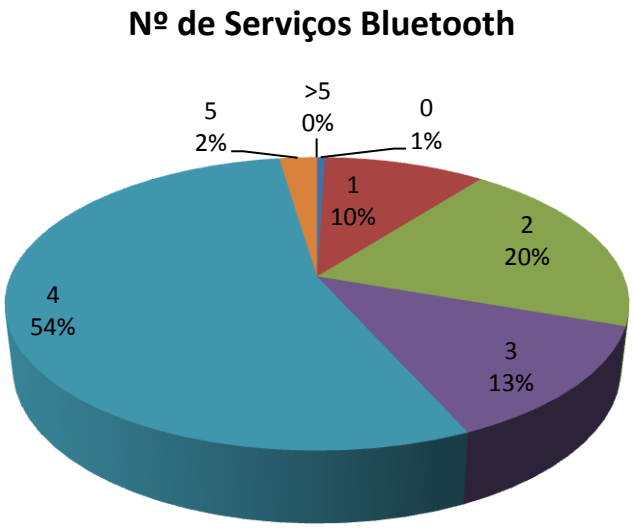


Gráfico 12 - Teste 1: Número de Serviços Bluetooth dos Dispositivos.

Para terminar a análise do teste 1, apenas falta referir que serviços foram detectados por dispositivo. Observando o Gráfico 13, constata-se que existem quatro serviços que estão presentes na maior parte dos dispositivos, sendo eles os serviços de: *Capturing*, *Network*, *ObjectTransfer* e *Telephony*. Ou seja, serviços que disponibilizam funcionalidades de fotografia/imagem, conexões de rede, transferência de objectos (por exemplo, ficheiros) e telefonia.

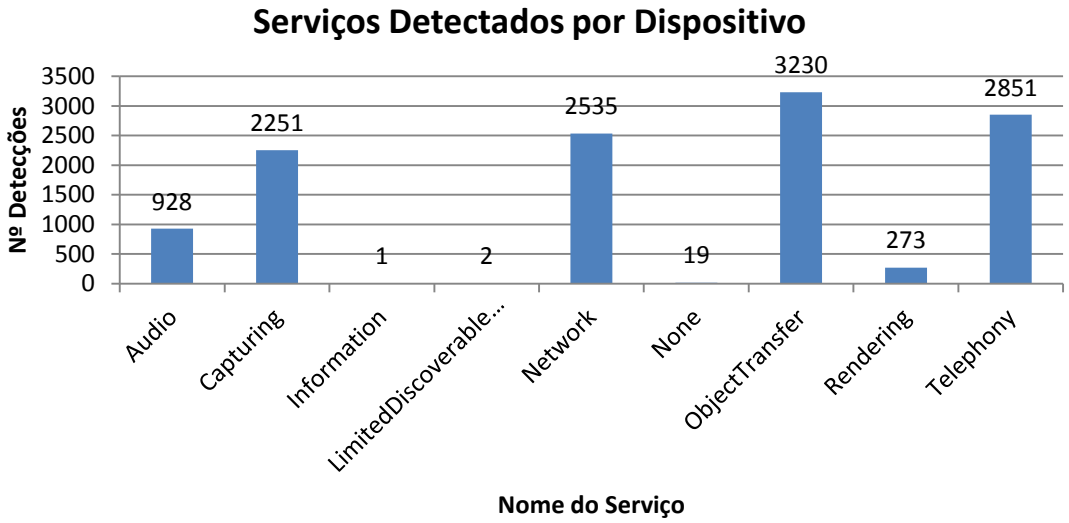


Gráfico 13 - Teste1: Serviços Detectados por Dispositivo.

6.2.2. Teste 2 - Análise de Resultados

Neste teste 2 as condições da realização do teste, foram exactamente as mesmas que as do teste 1. Apenas se realizou num período de tempo diferente.

Da análise das frequências de detecção, visíveis no Gráfico 14, nota-se que a partir das 20 detecções a curva de frequências cresce mais lentamente que a do teste 1. Isto acontece devido ao maior número de dispositivos detectados no teste 2. Neste teste também existem mais dispositivos detectados, um maior número de vezes.

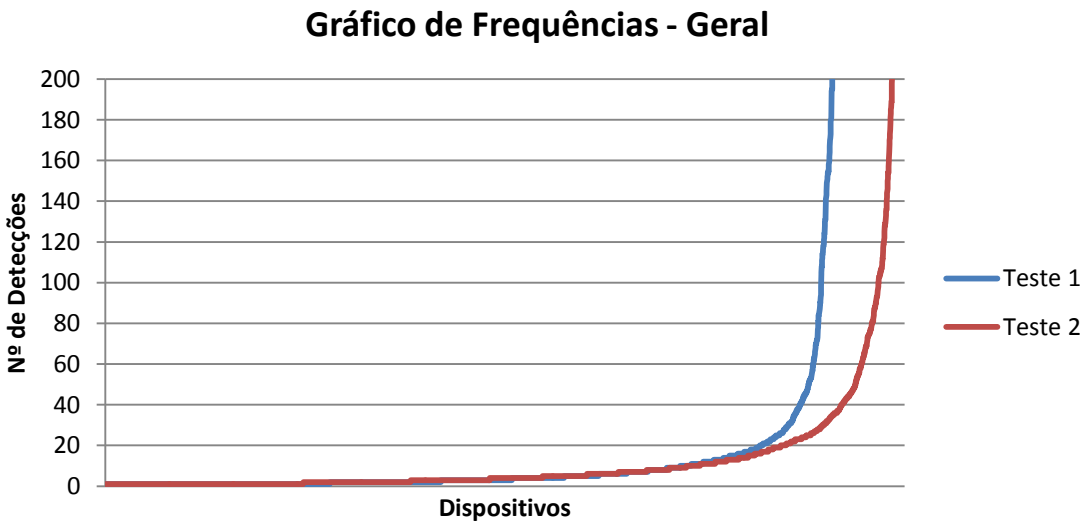


Gráfico 14 - Teste 2: Gráfico de Frequências de Detecção – Geral.

Na análise das categorias dos veículos e das pessoas, a história também não é muito diferente. Para os veículos (Gráfico 15), vê-se um crescimento da curva ligeiramente mais rápido que no teste anterior. Esta ligeira diferença também é justificada pelo menor número de detecções para a categoria dos veículos. Olhando agora para a categoria das pessoas (Gráfico 16), o crescimento da curva de frequências volta a ser mais lento quando comparado com o teste 1, evidenciando um aumento no número de dispositivos detectados.

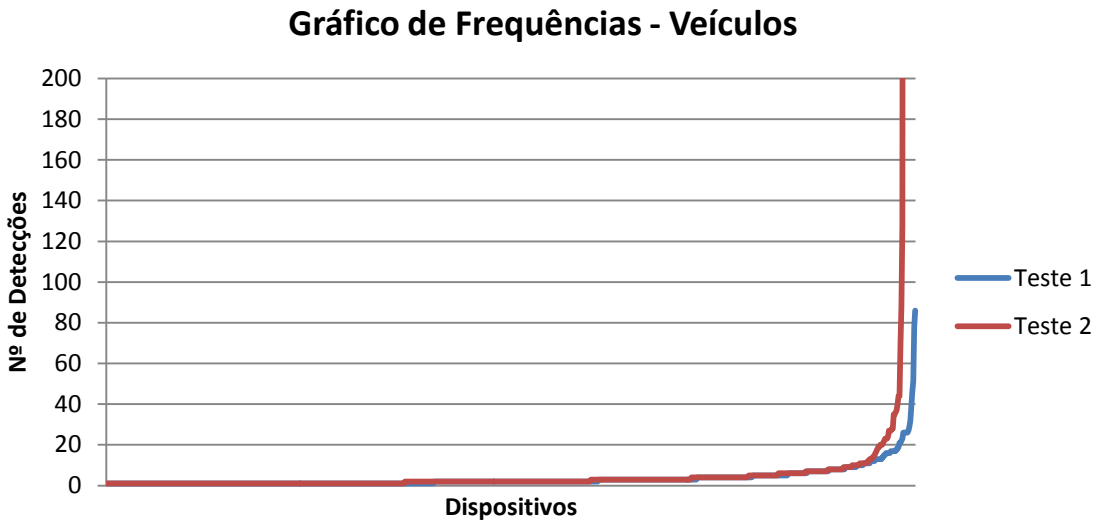


Gráfico 15 - Teste 2: Gráfico de Frequências de detecção – Veículos.

Gráfico 16 - Teste 2: Gráfico de Frequências de Detecção – Pessoas.

Visualizando agora o Gráfico 17, volta-se a verificar que a maior parte dos dispositivos foram detectados, de uma a cinco vezes e ainda que 3109 dispositivos foram detectados mais de uma vez, o que representa cerca de 75,24% de todos os dispositivos detectados. Este valor ainda é mais interessante que os valores apresentados no teste 1. Isto deve-se ainda ao facto de mais dispositivos, correspondentes à categoria das pessoas, terem sido detectados 20 ou mais vezes, como se comprova com o Gráfico 19.

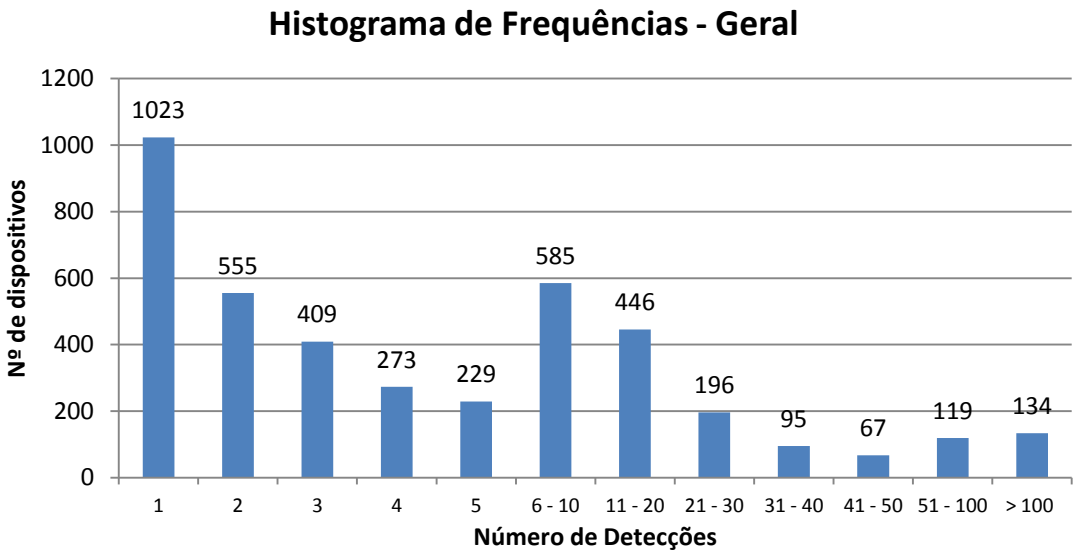


Gráfico 17 - Teste 2: Histograma de Frequências – Geral.

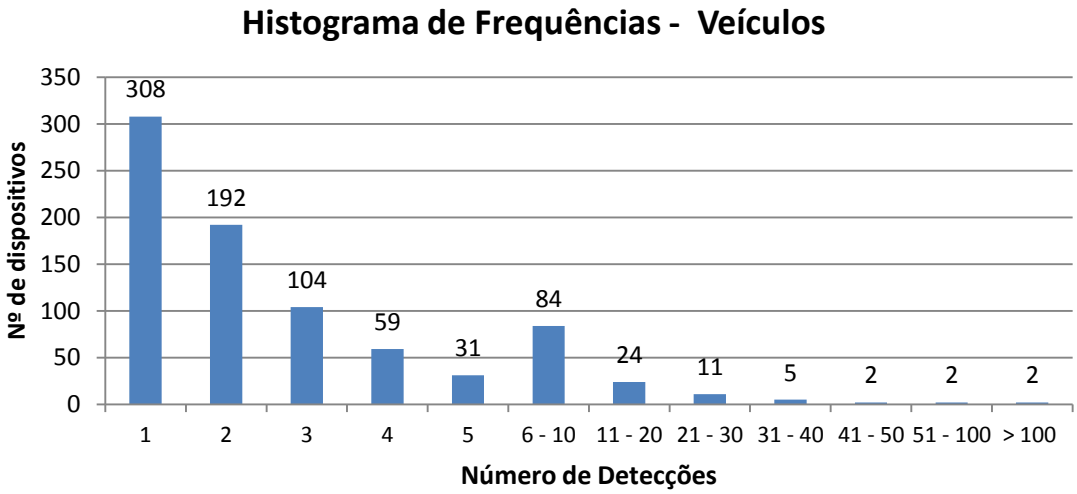


Gráfico 18 - Teste 2: Histograma de Frequências – Veículos.

Em contrapartida no Gráfico 18, correspondente à categoria dos veículos, verifica-se que muito poucos dispositivos foram detectados mais de vinte vezes. Fenómeno já verificado no teste 1. Estes veículos podem ter estacionado perto de um dos locais de *scan*, daí este elevado número, para tão poucos veículos. No caso da categoria das pessoas, as pessoas que são detectadas mais de 20 vezes deverão correspondem a pessoas que trabalham nas lojas ou pessoas que realizaram compras

nas lojas, passando mais tempo nas mesmas, e consequentemente os seus dispositivos foram detectados mais vezes.

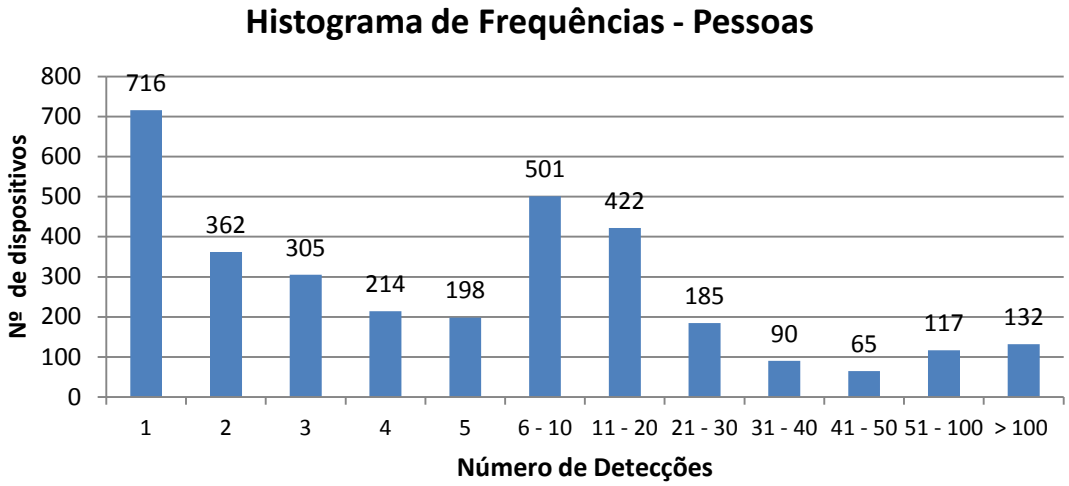


Gráfico 19 - Teste 2: Histograma de Frequências – Pessoas.

Em seguida é analisado o número de detecções totais realizadas por dia. O teste 2 teve início na quarta-feira dia 29-06-2011 às 16h44m e terminou na quarta-feira seguinte no dia 6-07-2011 às 16h43m. O dia em que se verificou um maior número de detecções foi na quinta-feira, com 17% das detecções totais (Gráfico 20). Um facto curioso é que já no teste 1 a quinta-feira tinha sido o dia com maior número de detecções, exactamente com a mesma percentagem de valores (ver Gráfico 8). Verificou-se também que no sábado é um dia de grande movimento na rua, como demonstram os resultados dos testes 1 e 2.

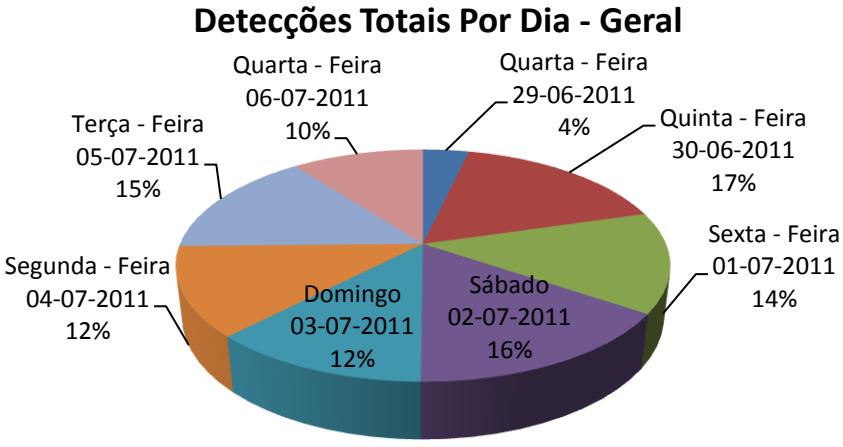


Gráfico 20 - Teste 2: Detecções totais por dia.

Relativamente à análise das detecções por hora (Gráfico 21), verifica-se a existência de um "padrão" semelhante ao do teste 1.

As detecções começam a diminuir durante as primeiras horas do dia, até cerca das 7 horas, como seria de esperar, uma vez que o comércio na rua se encontra fechado de madrugada. Depois das 7 horas da manhã até às 13 horas, o número de detecções sobe até atingir o pico máximo, de 6887 detecções. De seguida o número de detecções cai ligeiramente até às 15 horas, para voltar a subir até às 17 horas. As detecções vão decrescendo até às 21 horas, subindo para as 3133 detecções às 23 horas sendo o pico mais alto da noite para este teste 2. Este padrão de resultados apresentado no teste 2 é semelhante ao do teste 1, sendo a principal diferença a hora de pico das detecções. No teste 1 o pico máximo era às 22 horas com 5875 detecções (ver Gráfico 10), no teste 2 o pico máximo ocorre às 12 horas com 6887 detecções.

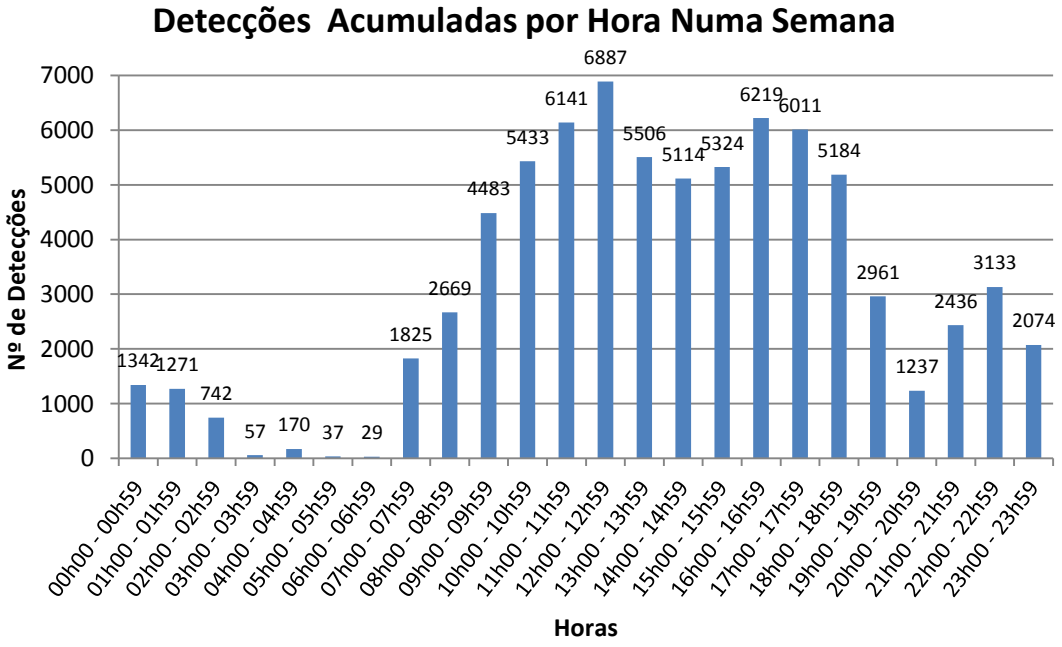


Gráfico 21 - Teste2: Detecções Acumuladas por Hora.

Por curiosidade, foi também analisado o número de dispositivos diferentes detectados por hora, para se perceber se a detecção do mesmo dispositivo, tinha algum

"peso" na análise anterior. Constatando os valores apresentados pelo Gráfico 22, isso não acontece. Este gráfico apresenta um padrão de resultados muito parecido com o anterior.

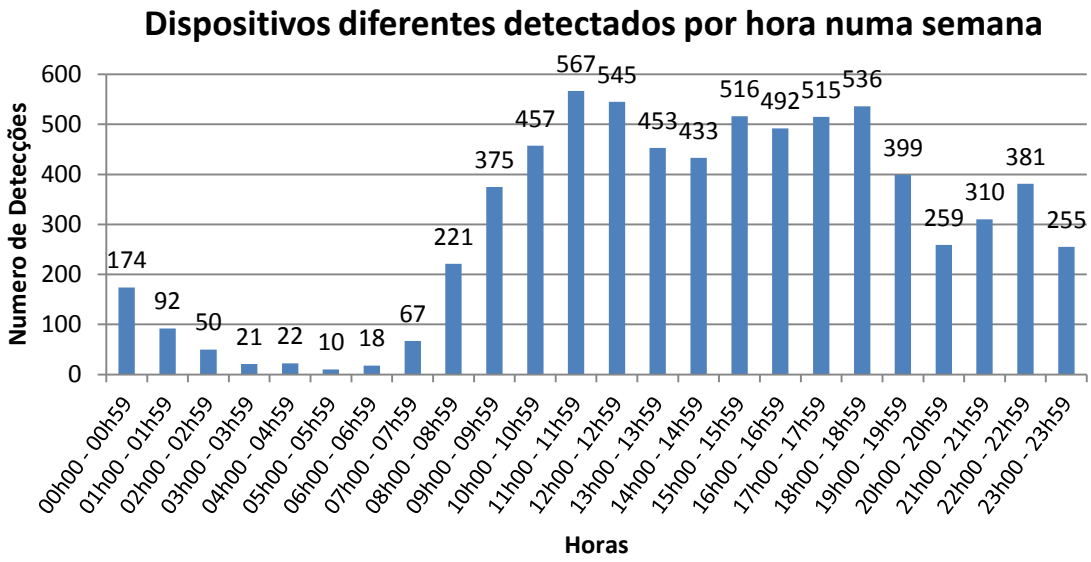


Gráfico 22 - Teste2: Dispositivos diferentes detectados por hora.

As classes de dispositivos que mais se detectaram (Gráfico 23), neste teste 2, foram: *AudioVideoHandsFree*, *CellPhone* e ainda *SmartPhone*. Estas foram as classes que também tinham sido mais detectadas no teste 1. Depois destas, as classes *AudioVideoHeadset* e *CordlessPhone* são as mais detectadas, com 53 detecções no caso da primeira e 18 no caso da segunda. Neste gráfico ainda estão descritas as classes 1036 e 1776, que devem corresponder a um tipo de dispositivo com uma versão mais antiga da tecnologia Bluetooth, e em que a API usada na aplicação de *scan* não conseguiu decifrar, o "nome técnico" das mesmas. Feita uma pesquisa constatou-se que a classe 1036 deve pertencer a um dispositivo de áudio/vídeo, e que a classe 1776 a um dispositivo de *Imaging* com funcionalidades de visualização, gravação de imagem, scanner e impressão.

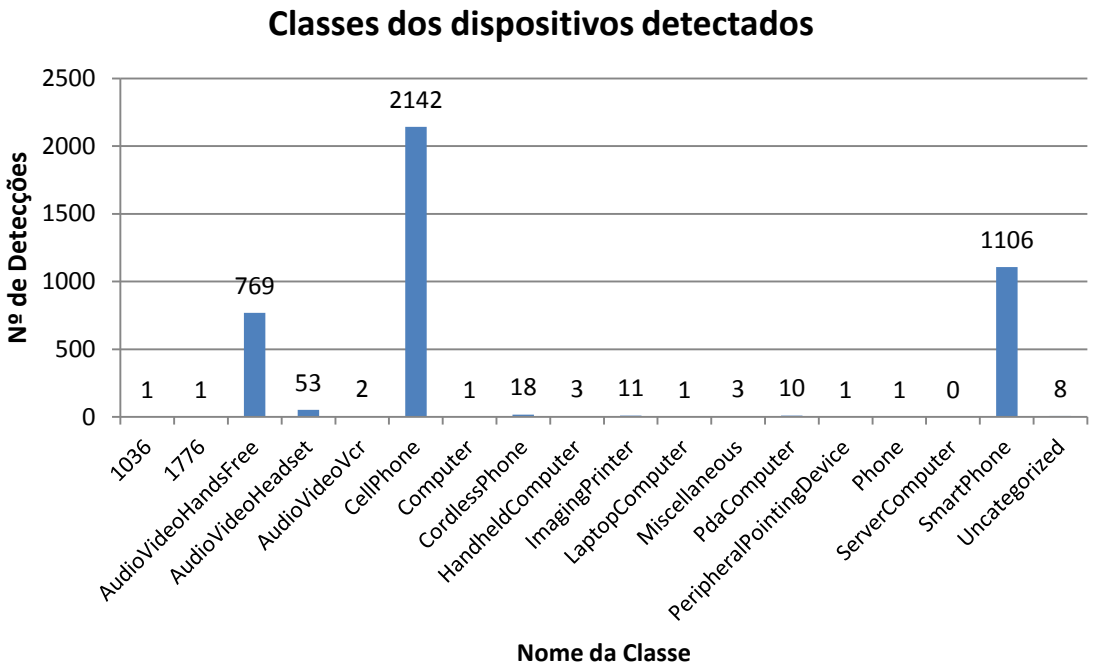


Gráfico 23 - Teste 2: Classes dos dispositivos detectados.

Da análise do número de serviços detectados nos dispositivos (Gráfico 24), conclui-se que o número de serviços presentes nos mesmos aumentou relativamente. Neste teste 2, aumentaram as detecções de dispositivos com 3 e 5 serviços. Contudo, a maior parte dos dispositivos detectados ainda possui 4 serviços com 54% de todos os dispositivos detectados.

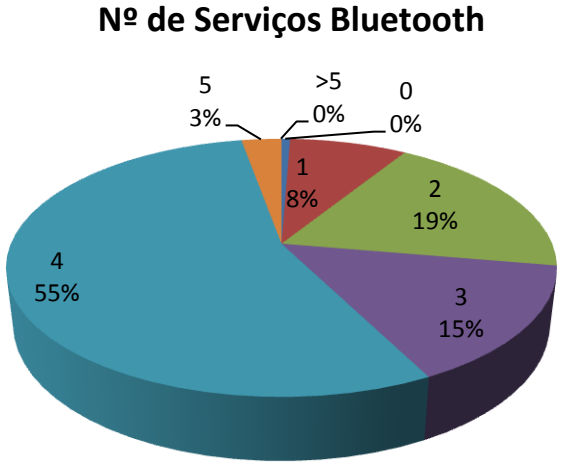


Gráfico 24 - Teste 2: Número de serviços Bluetooth dos dispositivos.

Relativamente aos serviços por dispositivo encontrados (Gráfico 25), a tendência é semelhante à do teste anterior. Aqui os serviços mais detectados são também os serviços de: *Capturing*, *Network*, *ObjectTransfer* e *Telephony*.

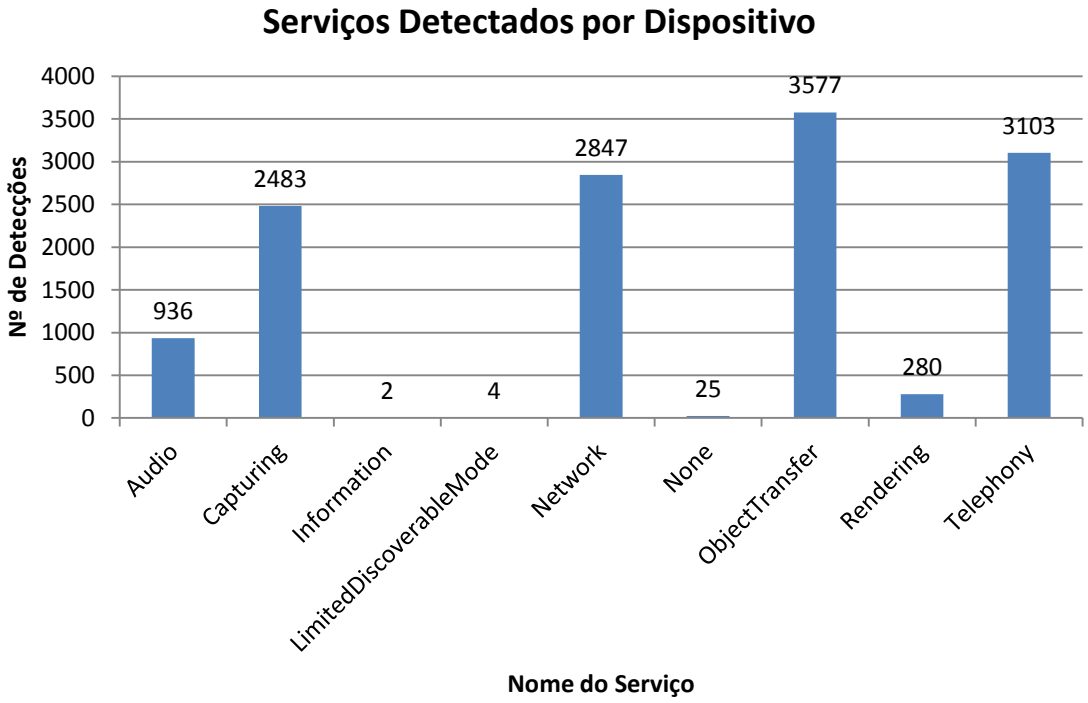


Gráfico 25 - Teste1: Serviços Detectados por Dispositivo.

6.2.3. Teste 3 - Análise de Resultados

Para completar a análise de resultados dos testes com elevada mobilidade, apenas falta a análise dos resultados para o teste 3. Na execução deste teste 3, foi alterado um parâmetro na aplicação de *scan* desenvolvida. O parâmetro alterado foi o tempo de *scan*, ou seja, o tempo de procura por dispositivos. O tempo de *scan* neste teste foi modificado para cinco segundos, enquanto que nos testes 1 e 2 o seu valor era de três segundos.

Convém ainda explicar que a duração do teste 3 foi mais curta que a duração dos outros dois testes. Este problema aconteceu devido a uma falha de energia, numa das lojas onde se tinha instalado a aplicação de *scan*, fazendo com que se desligasse o computador que fazia os *scans*. Como este problema ocorreu no último dia do teste, optou-se por excluir este último dia da análise de resultados, ficando assim o teste 3 com a duração de seis dias em vez de sete.

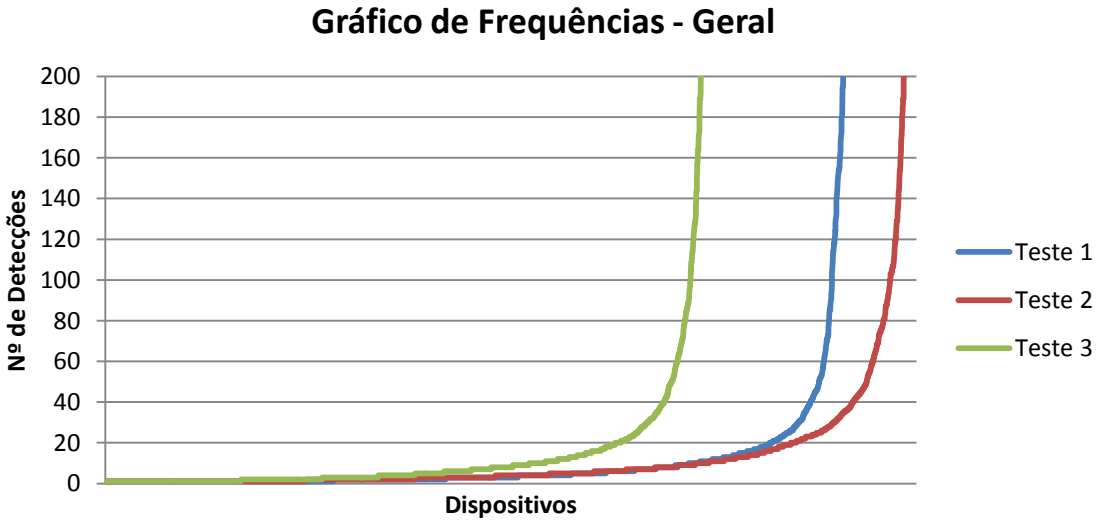


Gráfico 26 - Teste 3: Gráfico de Frequências – Geral

Relativamente à análise das frequências para o teste 3 verifica-se, por intermédio do Gráfico 26, que a curva tem um crescimento mais acelerado em relação aos outros testes realizados. Apesar da sua inclinação só se acentuar a partir das 30 detecções, até

esse valor a curva já apresenta um crescimento maior do que os valores que foram obtidos nos restantes testes.

No que concerne à categoria dos veículos (Gráfico 27) e das pessoas (Gráfico 28), o crescimento da curva de frequências é mais lento do que no caso dos outros dois testes. Este acontecimento deve-se ao facto de no teste 3 ter tido uma duração mais curta e como consequência, foram detectados menos dispositivos, por oposição ao teste 2 que foi o teste onde se verificaram mais detecções.

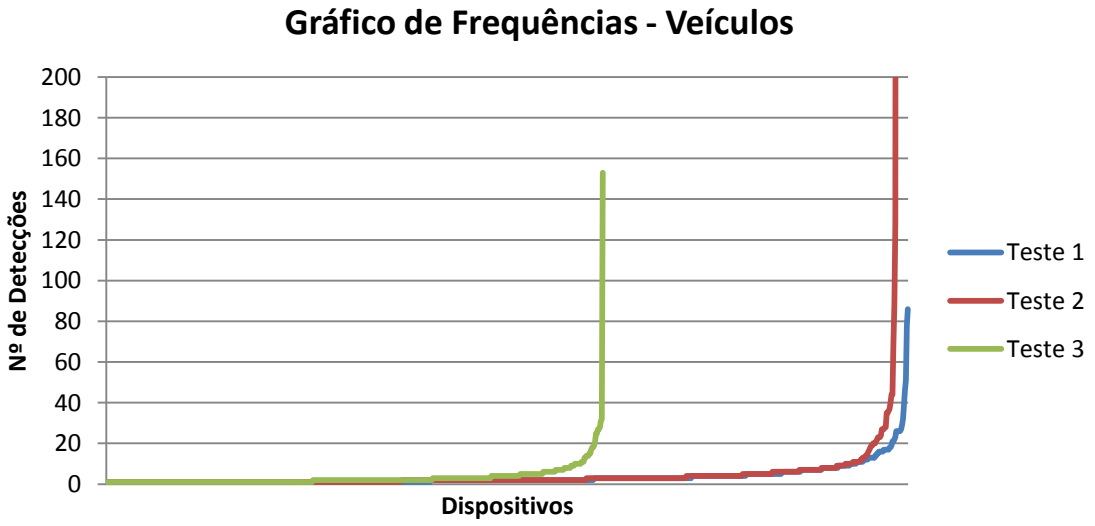


Gráfico 27 - Teste 3: Gráfico de Frequências - Veículos

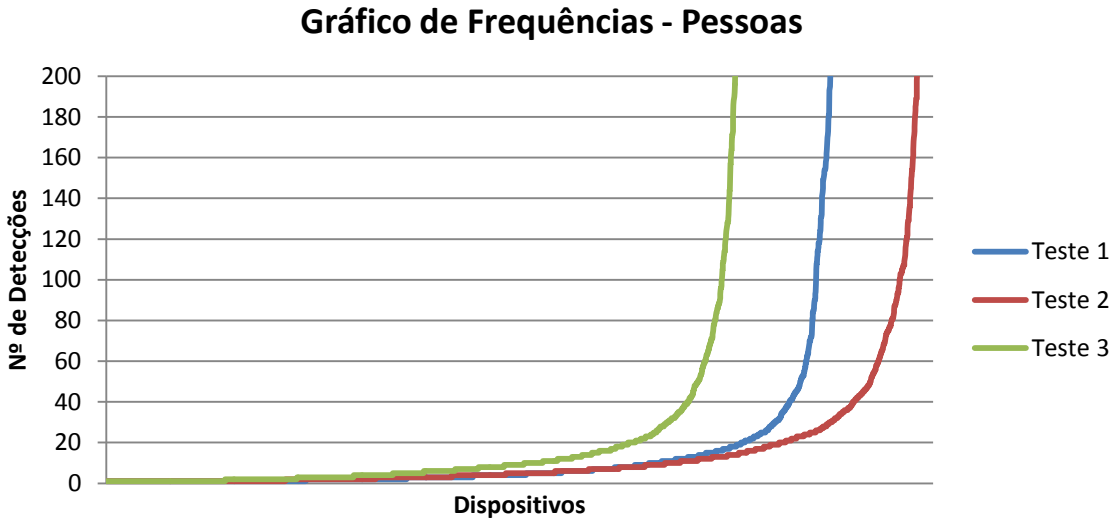


Gráfico 28 - Teste 3: Gráfico de Frequências - Pessoas

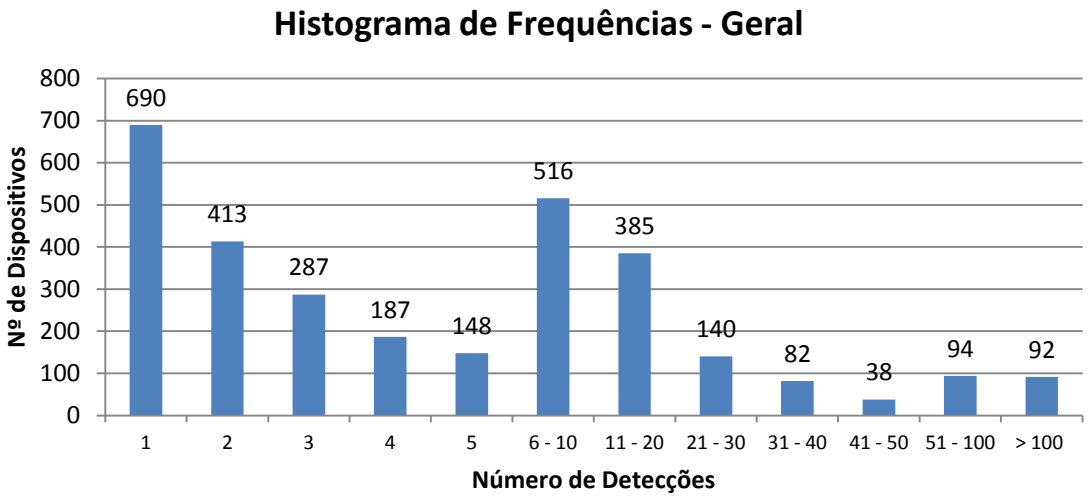


Gráfico 29 - Teste3: Histograma de Frequências – Geral.

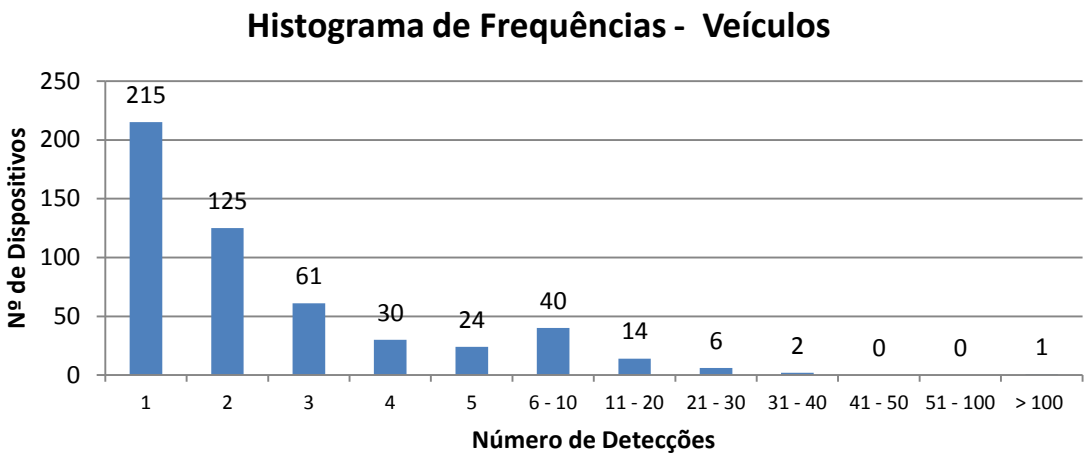


Gráfico 30 - Teste3: Histograma de Frequências – Veículos.

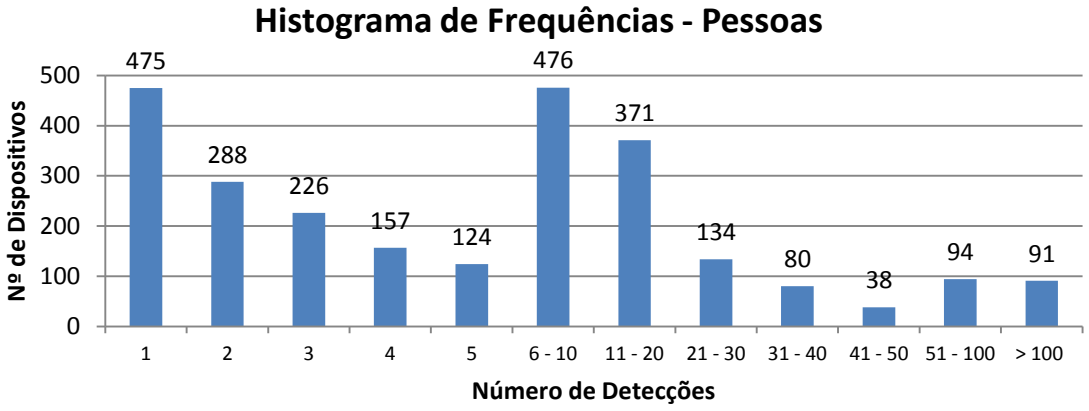


Gráfico 31 - Teste3: Histograma de Frequências – Pessoas.

Visualizando agora os histogramas de frequências para o teste 3, volta-se a constatar, como nos testes anteriores, que a classe de 1 a 5 dispositivos é a que mais vezes foi detectada. Desta vez, neste teste, 77,54% dos dispositivos foram detectados mais de uma vez (ver Gráfico 29). Comparado com os outros dois testes este valor ainda é mais elevado, contudo não se pode esquecer que este teste teve a duração de menos um dia.

Relativamente aos veículos (Gráfico 30), os resultados são semelhantes aos dos testes anteriores. Destaca-se aqui apenas que um único dispositivo foi detectado mais de cem vezes, isto pode indicar que algum veículo estacionou perto de algum dos locais de *scan* obtendo-se para o mesmo um número elevado de detecções.

Na categoria das pessoas (Gráfico 31) é notório que existiu um aumento no número de dispositivos que foram detectados mais de trinta vezes, a destacar aqui os 91 dispositivos que foram detectados mais de cem vezes.

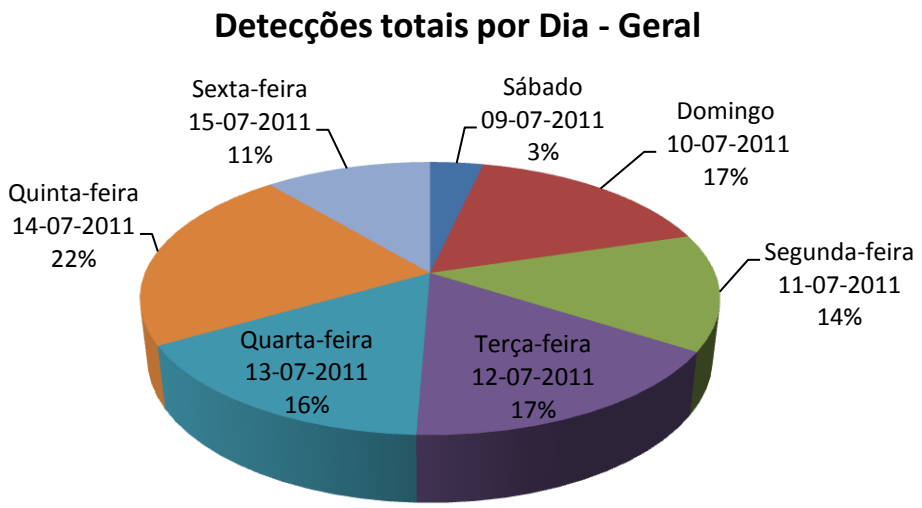


Gráfico 32 - Teste3: Detecções totais por dia.

O teste 3, teve início no sábado dia 09-07-20011 às 18h50m e terminou numa sexta-feira dia 15-07-2011 à mesma hora. Durante o seu período de execução o dia no qual se verificaram mais detecções foi na quinta-feira dia 14, com 22% de todas as detecções (Gráfico 32). Este é o valor mais alto obtido em todos os testes realizados.

Neste teste também se verifica uma maior diferença no número de detecções para os diferentes dias da semana.

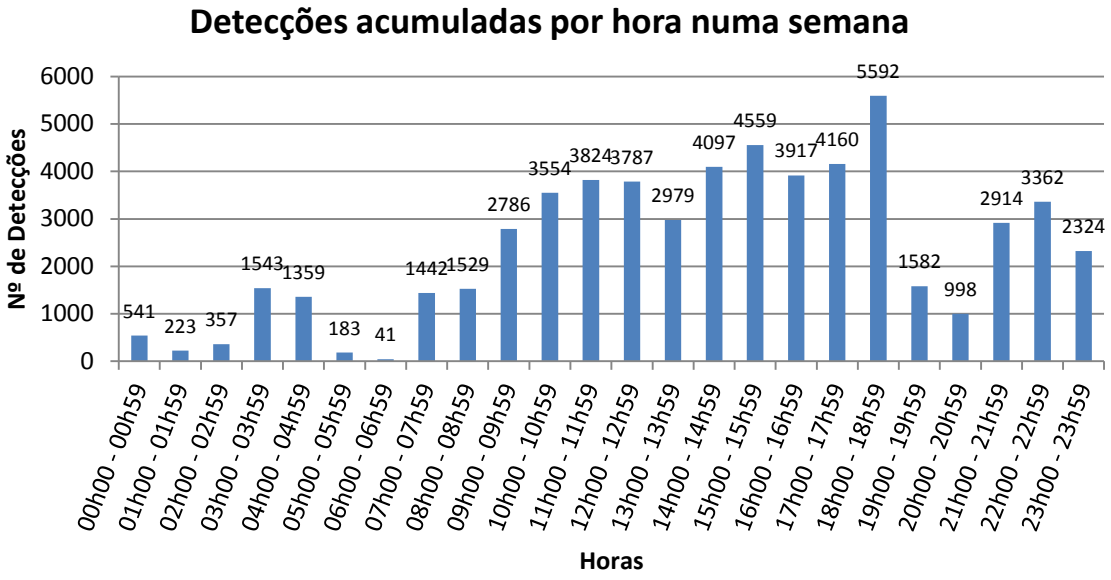


Gráfico 33 - Teste3: Detecções acumuladas por hora numa semana.

Na análise do número de detecções acumuladas por hora (Gráfico 33), também se verifica a existência de um padrão de resultados semelhante ao teste 1 e 2. Se forem analisadas só as primeiras horas do dia até as 7 horas da manhã, verifica-se a existência de um considerável número de detecções, cerca de 1500, entre as 3 horas e as 5 horas. Facto curioso para esta hora do dia.

O número de detecções vai subindo sempre a partir das 7 horas da manha, até estabilizar nas 3800 detecções, entre as 11 horas e as 13 horas, para depois baixar para menos de 3000. Aspecto justificável porque engloba a hora de almoço.

Após a hora de almoço, durante a tarde, as detecções atingem o pico máximo às 18 horas, com o valor de 5592, para em seguida caírem abruptamente ate à hora de jantar. Das 18 horas às 20 horas é verificada uma queda de 4594 detecções.

Durante a noite, o maior número de detecções ocorre às 23 horas, com cerca de 3362 detecções.

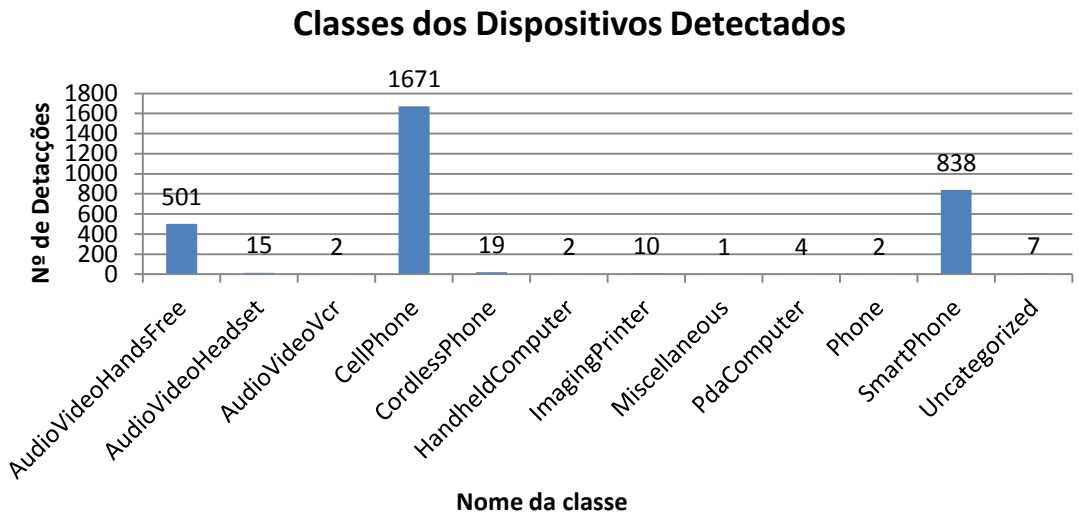


Gráfico 34 - Teste 3: Classes dos Dispositivos detectados.

Para concluir a análise dos resultados do teste 3, apenas falta referir as classes de dispositivos encontradas, número de serviços disponíveis nos dispositivos e dizer qual o nome dos serviços detectados. No primeiro caso, no Gráfico 34, volta-se a verificar o esmagador domínio das classes *AudioVideoHandsFree*, *CellPhone* e ainda *SmartPhone*, conforme já tinha acontecido nos testes anteriores. Com a classe *CellPhone* a ser a mais detectada.

Em relação ao número de serviços Bluetooth presentes nos dispositivos (Gráfico 35), também se volta a verificar que é mais comum encontrar dispositivos com 4 serviços diferentes e que não foi detectado nenhum dispositivo com mais de cinco serviços.

Por fim, o nome dos serviços detectados nos dispositivos também não apresentam mudanças os mais detectados continuam a ser os serviços de: *Capturing*, *Network*, *ObjectTransfer* e *Telephony*.

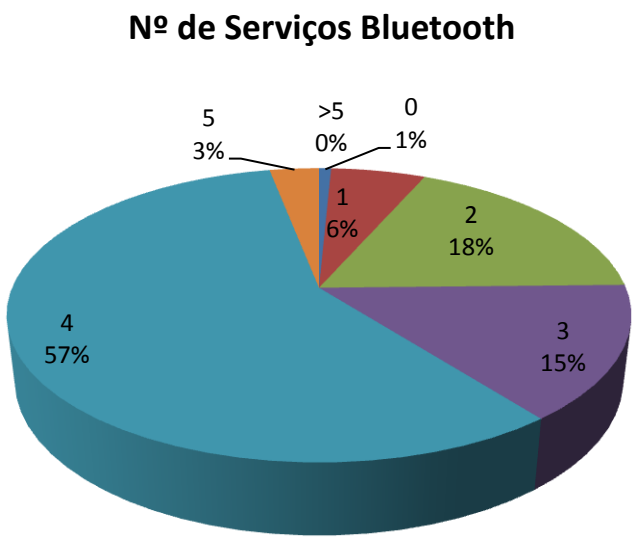


Gráfico 35 - Teste 3: Número de serviços Bluetooth dos dispositivos.

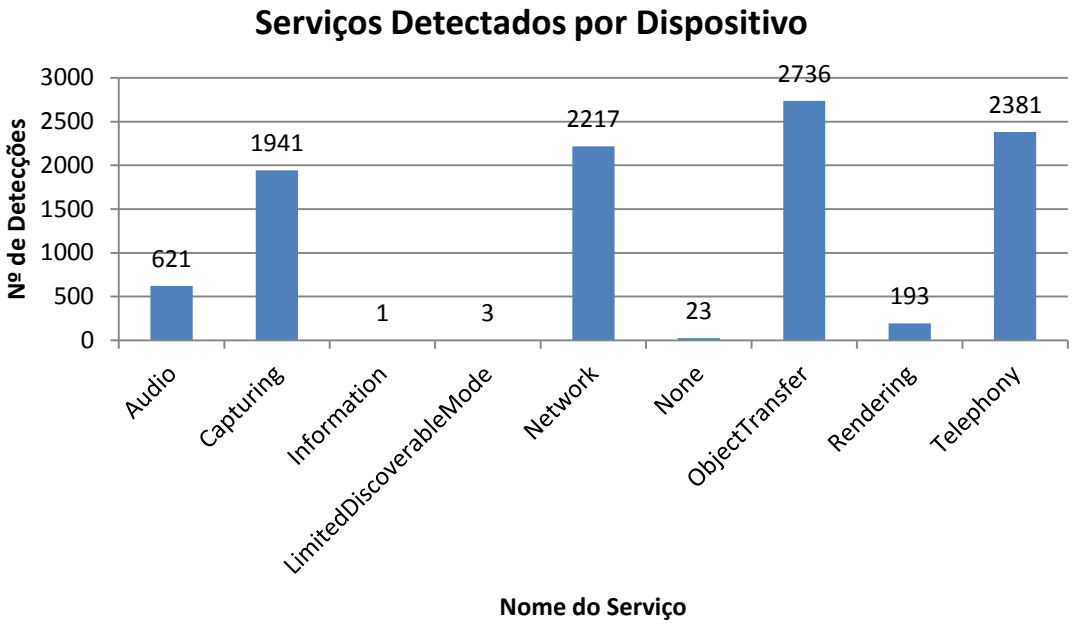


Gráfico 36 - Teste3: Serviços Detectados por Dispositivo.

6.2.4. Cálculo das Probabilidades de Detecção

O cálculo das probabilidades de detecção teve como base os processos de análise descritos no capítulo 5. Aqui são analisadas as probabilidades de falha para cada local de detecção, bem como, a análise em conjunto dos três locais de detecção. Similarmente são analisados à parte todo o conjunto de dispositivos e os dispositivos que dizem respeito à categoria dos veículos e das pessoas.

No cálculo das probabilidades de falha aplicado no processo na análise por local é utilizado o processo de análises descrito na secção 5.3.1. Para a análise global é utilizado o processo de análise apresentado na secção 5.3.2.

6.2.4.1 Análise por Local

Para se determinar a probabilidade de detecção de um dispositivo num local, requer que se obtenha uma forma de calcular o número de vezes que um dispositivo não foi detectado.

O objectivo do algoritmo apresentado na secção 5.3.1 é tentar solucionar esse problema. Este algoritmo no final da sua execução apresenta o número de falhas para um determinado local, sabendo-se assim o número de vezes que um dispositivo não foi detectado. Após esse número ser calculado basta aplicar a equação 2, obtendo-se assim uma probabilidade de Falha para o local em questão.

Depois de calculada a probabilidade de falha é muito mais fácil determinar a probabilidade de detecção. Esta probabilidade é dada pela equação 6:

$$P.Detecção_{num\ local} = 1 - P_{Falha} \qquad (\boldsymbol{Eq.6})$$

Os resultados obtidos para a probabilidade de detecção, para todos os locais e para todos os testes realizados, encontram-se descritos na Tabela 5 e no Gráfico 37.

Local	Teste	Nº de *Scans*[4]	Total de Detecções	Dispositivos Detectados	Nº de Falhas	Prob. De Falhas (%)	Prob. De Detecção (%)
1	1	N/D	47.801	2.804	2.458	4,89	95,11
	2	138.947	37.445	3.222	2.044	5,18	94,82
	3	N/D	25.674	2.141	833	3,14	96,86
2	1	N/D	11.874	1.529	598	4,79	95,21
	2	149.234	22.728	2.447	1.329	5,52	94,48
	3	N/D	20.654	2.040	936	4,34	95,66
3	1	N/D	13.613	2.659	563	3,97	96,03
	2	151.546	16.112	2.663	693	3,81	96,19
	3	N/D	11.325	2.205	331	2,84	97,16

Tabela 5 - Análise por local – Geral.

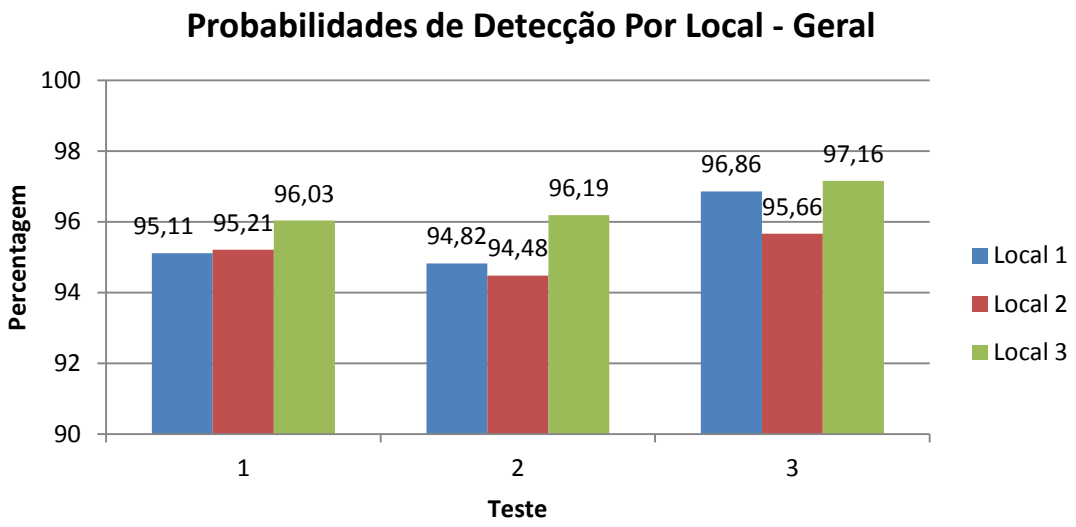


Gráfico 37 - Probabilidades de detecção por local – Geral

Observando a coluna da probabilidade de falhas, na Tabela 5, verifica-se que os valores obtidos são bastante satisfatórios. Verifica-se que a probabilidade de falha que se obteve mais elevada foi de 5,52%. Todos estes valores, tendo em conta o algoritmo desenvolvido, indicam que a probabilidade de detecção dos dispositivos por local é elevada. Concentrando as atenções no Gráfico 37 nota-se que o local 3, é o local onde o

[4] O número de *scans* não apresentados deve-se a dois aspectos. No caso do teste 1, ainda não se encontrava implementada, na aplicação de *scan*, a contagem do número de *scans*. No caso do teste 3, os valores não são apresentados devido à ocorrência do problema da falha de energia.

sucesso na detecção de um dispositivo é mais elevado. Convêm, recordar que neste teste 3, o tempo se *scan* dos dispositivos era mais elevado, era de 5 segundos.

Local	Teste	Nº de *Scans*	Total de Detecções	Veículos Detectados	Nº de Falhas	Prob. De Falhas (%)	Prob. De Detecção (%)
1	1	N/D	1.243	596	17	1,35	98,65
	2	138.947	2.099	627	122	5,49	94,51
	3	N/D	597	311	15	2,45	97,55
2	1	N/D	519	202	44	7,82	92,18
	2	149.234	840	372	36	4,11	95,89
	3	N/D	684	266	17	2,43	97,57
3	1	N/D	1.351	586	33	2,38	97,62
	2	151.546	872	465	16	1,8	98,2
	3	N/D	482	317	8	1,63	98,37

Tabela 6 - Análise por local - veículos.

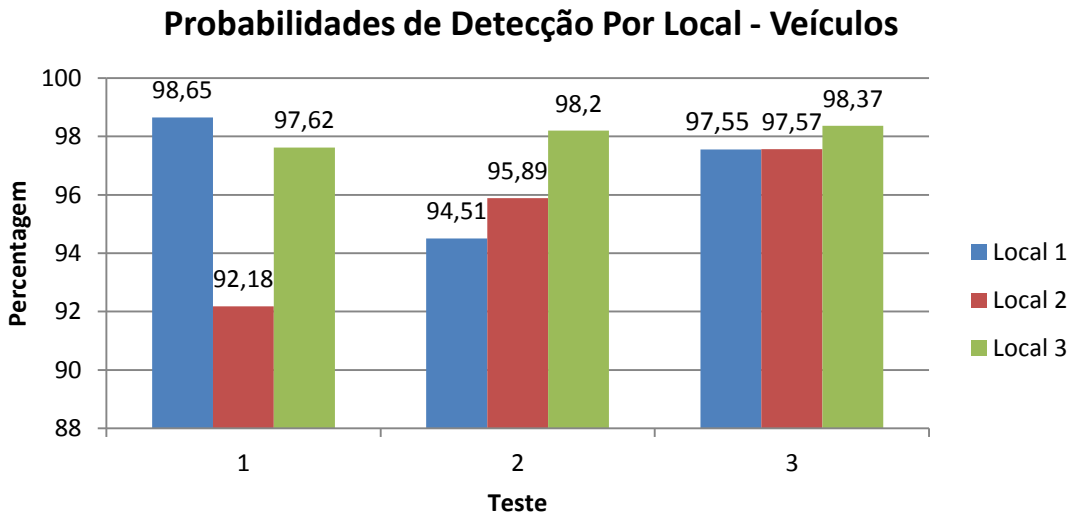


Gráfico 38 - Probabilidades de detecção por local - Veículos.

Do gráfico observa-se também que no teste 1 a variação da probabilidade de detecção, entre os locais de detecção, é menor que nos restantes testes.

Considerando agora apenas a categoria dos veículos, observa-se que no teste 1 a discrepância entre as probabilidades de detecção é mais acentuada (Gráfico 38). No local 2 obtém-se unicamente 92,18%, o valor mais baixo de probabilidade de detecção, deste estudo. Assinalar que no teste 3 todos os valores de detecção são muito

semelhantes. No que concerne aos veículos detectados (Tabela 6), o local 2 também é o que apresenta sempre os menores valores. Isto talvez aconteça, devido ao facto de o local de detecção 2 ser um local mais afastado da estrada, uma vez que aqui o passeio é mais largo e ainda existem locais de estacionamento.

Local	Teste	Nº de *Scans*	Total de Detecções	Pessoas Detectadas	Nº de Falhas	Prob. De Falhas (%)	Prob. De Detecção (%)
1	1	N/D	46.558	2.208	2.439	4,98	95,02
	2	138.947	35.346	2.595	1.919	5,14	94,86
	3	N/D	25.077	1.830	816	3,15	96,85
2	1	N/D	11.355	1.327	553	4,64	95,36
	2	149.234	21.888	2.075	1.293	5,58	94,42
	3	N/D	19.970	1.774	919	4,4	95,6
3	1	N/D	12.262	2.073	526	4,11	95,89
	2	151.546	15.240	2.198	623	3,92	96,08
	3	N/D	10.843	1.888	320	2,87	97,13

Tabela 7 - Análise por local - pessoas.

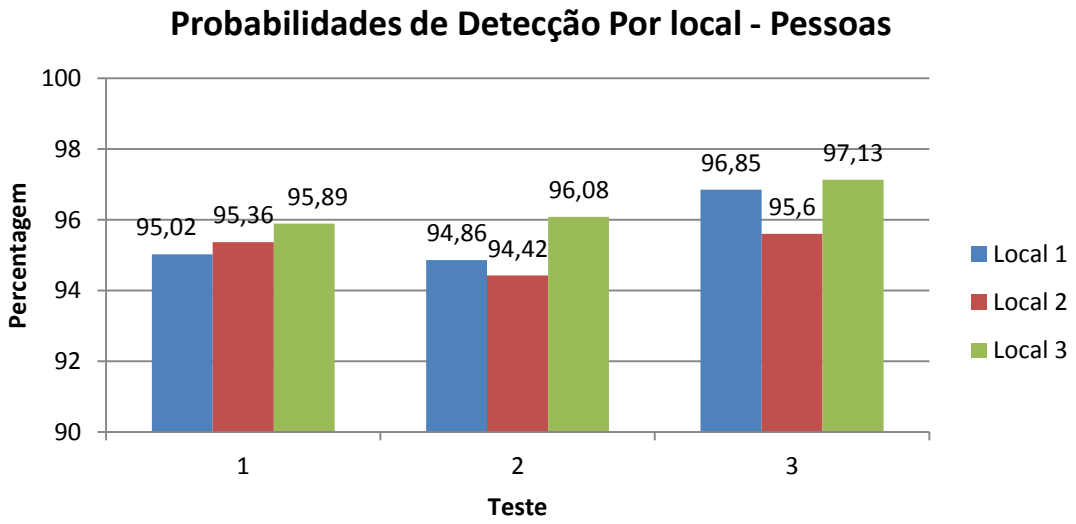


Gráfico 39 - Probabilidades de detecção por local – Pessoas

Para terminar a análise por local constata-se que para a categoria das pessoas, os valores de probabilidade obtidos (Gráfico 39) são semelhantes aos obtidos quando se consideram todos os dispositivos detectados. Na Tabela 7 verifica-se que a probabilidade de detecção das pessoas, em todos os locais, ronda quase sempre os

mesmos valores. Com excepção do local 2, que no teste 1 a probabilidade de pessoas detectadas foi bastante mais baixo que nos restantes locais.

6.2.4.2. Análise Global

Como já foi referido anteriormente, na secção 5.3.2, esta análise global tem o objectivo de determinar uma probabilidade de detecção para todos os dispositivos que foram detectados, obedecendo a um padrão de detecção ou sequência de detecção específica. Para os testes realizados foram então definidos dois tipos de sequências, as sequências completas e incompletas. Após determinado o número de sequências para cada tipo, seria possível chegar ao valor da probabilidade de detecção correcta, dado pela Equação 4. Para a determinação do número de sequências completas e de sequências incompletas recorreu-se ao algoritmo apresentado na Figura 27.

Um parâmetro importante usado no algoritmo, para o cálculo destes números de sequências, é o tempo assumido para um dispositivo se deslocar de um local de detecção a outro, seja o mesmo da categoria das pessoas ou veículos. Para esse parâmetro (Δt) presente na Equação 3, foi atribuído um diverso conjunto de valores (24,60,120,300,600 e 1800) de forma a estudar se as probabilidades de detecção sofriam grandes alterações com a variação do mesmo.

Examinando a Tabela 8, encontramos os valores obtidos para as probabilidades de detecção. Comparando cada teste realizado, verifica-se que a probabilidade de detecção toma valores relativamente próximos uns dos outros, para os diferentes testes. O mesmo acontece quando se fez variar o valor de Δt, onde as probabilidades variam sempre muito pouco, com excepção do Δt de 24 segundos. Este factor indica a estabilidade deste processo de análise.

Se olharmos de novo para a Tabela 8, constata-se, como seria de esperar, o aumento do número de sequências completas, quando se aumenta o valor para Δt.

Δt (seg)	Nº Seq. Incompletas			Nº Seq. Completas			Prob. Detecção(%)		
	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3
24	1.216	1.496	1.324	317	447	307	20,68	23,01	18,82
60	1.848	2.111	1.774	869	1.314	879	31,98	38,36	33,13
120	1.938	2.179	1.804	986	1.543	1.034	33,72	41,46	36,43
300	2.011	2.257	1.867	1.067	1.648	1.145	34,67	42,20	38,01
600	2.069	2.296	1.934	1.116	1.725	1.176	35,04	42,90	37,81
1800	2.199	2.417	2.041	1.175	1.810	1.245	34,83	42,82	37,89

Tabela 8 - Análise Global - Geral.

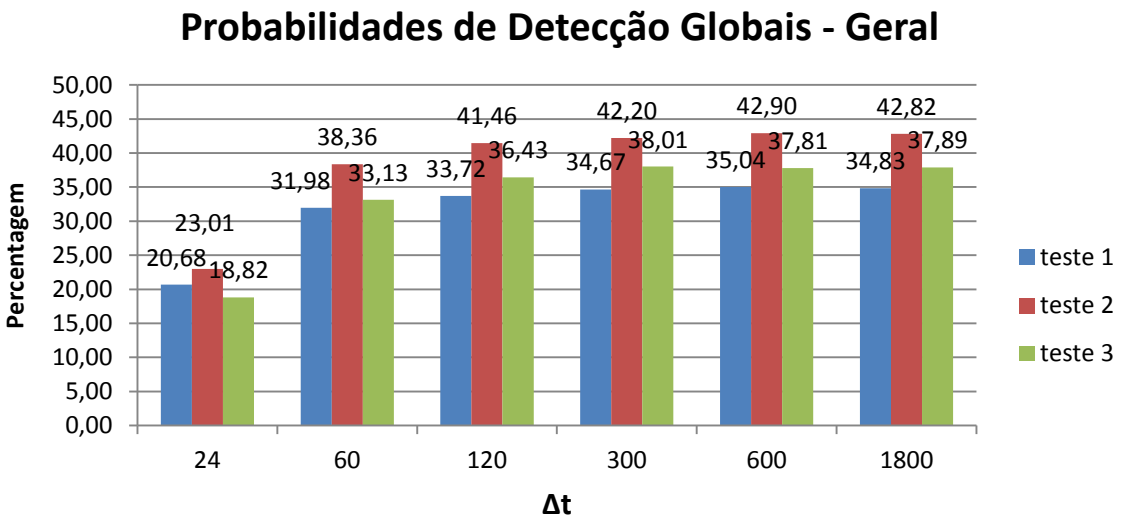


Gráfico 40 - Probabilidades de detecção globais - Geral.

Com um Δt de 24 segundos, o teste 2 obteve a probabilidade de detecção mais elevada com um valor cerca de 23,01% (Gráfico 40), nos restantes dois testes também se obtiveram valores semelhantes. O mesmo se verifica para os outros valores de Δt, onde o teste 2 continua sempre com a probabilidade mais elevada.

De todos os testes realizados, o valor máximo da probabilidade de detecção ronda os 43%. Este valor indica que nem metade dos dispositivos que foram detectados,

realiza uma sequência de detecção completa, ou seja, não passam por todos os locais de teste. Este baixo valor tem a ver com o ambiente de teste, que como já foi referido, apresenta várias condicionantes ao processo de cálculo de uma probabilidade de detecção.

Nesta análise de detecção global, um dos aspectos mais sugestivos será "observar" o comportamento dos dispositivos da categoria dos veículos. Uma vez que estes apenas se podem deslocar num único sentido, e neste caso realizando a sequência (1→2→3) (ver Figura 15), é quase como certo que passem em todos os pontos de detecção. Por outro lado, existe uma garagem de estacionamento de veículos entre os locais de detecção 1 e 2, o que pode condicionar a não passarem no local 3, não realizando assim a sequência de detecção completa (1→2→3).

Δt (seg)	Nº Seq. Incompletas			Nº Seq. Completas			Prob. Detecção(%)		
	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3
24	399	311	155	77	140	49	16,18	31,04	24,02
60	416	317	158	94	170	53	18,43	34,91	25,12
120	418	321	160	94	173	54	18,36	35,02	25,23
300	427	325	168	94	173	55	18,04	34,74	24,66
600	436	330	174	94	174	55	17,74	34,52	24,02
1800	449	349	186	95	178	57	17,46	33,78	23,46

Tabela 9 - Análise Global - Veículos.

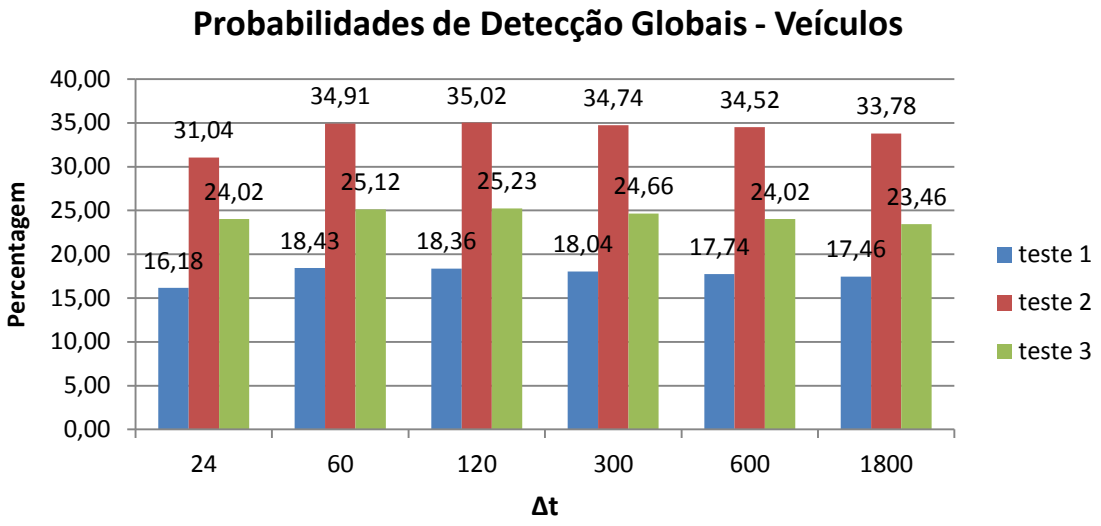


Gráfico 41 - Probabilidades de detecção globais - Veículos.

A Tabela 9 apresenta os resultados referentes à categoria dos veículos. Aqui, mais uma vez se destaca que ao variar o Δt de 24 para 60 segundos, o número de sequências completas aumenta, enquanto que para outros valores de Δt, o número tende a estabilizar.

Relativamente aos valores das probabilidades de detecções obtidas (Gráfico 41), novamente o valor mais alto obtido ocorre no teste 2 (cerca de 35%), sendo este valor cerca de 10% mais elevado do que o do teste 3 e cerca de 15% mais elevado do que o do teste 1. O valor de probabilidade mais alto foi de 35,02%, para o teste 2 com um Δt de 120 segundos. Ao contrário do que seria de se esperar, pelas razões anteriormente indicadas, esta categoria também apresenta valores de probabilidade de detecção relativamente baixos.

Δt (seg)	Nº Seq. Incompletas			Nº Seq. Completas			Prob. Detecção(%)		
	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3
24	817	1.185	1.169	240	307	258	22,71	20,58	18,08
60	1.432	1.794	1.616	775	1.144	826	35,12	38,94	33,82
120	1.520	1.858	1.644	892	1.370	980	36,98	42,44	37,35
300	1.584	1.932	1.699	973	1.475	1.090	38,05	43,29	39,08
600	1.633	1.966	1.760	1.022	1.551	1.121	38,49	44,10	38,91
1800	1.750	2.068	1.855	1.080	1.632	1.188	38,16	44,11	39,04

Tabela 10 - Análise Global - Pessoas.

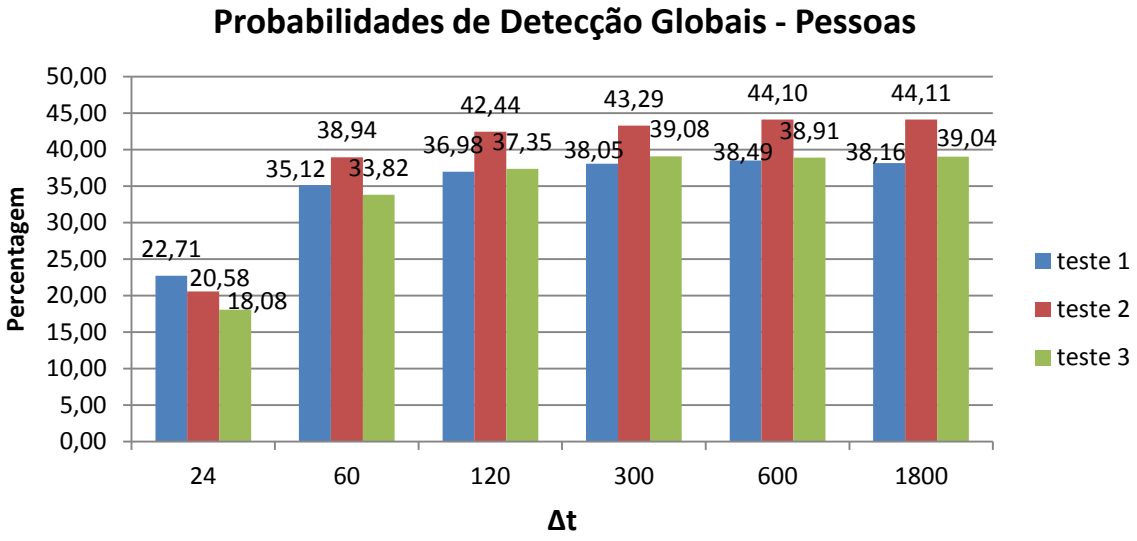


Gráfico 42 - Probabilidades de detecção globais - Pessoas.

Na categoria das pessoas observa-se um facto curioso (Gráfico 42), o teste 2 contínua a ser o teste onde se obtém a probabilidade mais elevada com 44,11%. Contudo quando o Δt é de 24 segundos, o teste 1 apresenta a probabilidade de detecção mais elevada (22,71%). Aqui o número de sequências completas também aumenta com o aumento dos valores do Δt (ver

Δt (seg)	Nº Seq. Incompletas			Nº Seq. Completas			Prob. Detecção(%)		
	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3
24	817	1.185	1.169	240	307	258	22,71	20,58	18,08
60	1.432	1.794	1.616	775	1.144	826	35,12	38,94	33,82
120	1.520	1.858	1.644	892	1.370	980	36,98	42,44	37,35
300	1.584	1.932	1.699	973	1.475	1.090	38,05	43,29	39,08
600	1.633	1.966	1.760	1.022	1.551	1.121	38,49	44,10	38,91
1800	1.750	2.068	1.855	1.080	1.632	1.188	38,16	44,11	39,04

Tabela 10).

Falta apenas referir que nesta análise global, o facto do teste 3 realizar um *scan* de dispositivos de 5 segundos em vez de 3 segundos, como nos outros testes, não trás grandes vantagens no aspecto da probabilidade de detecção de dispositivos. Neste estudo, um valor mais elevado no tempo de *scan* implicou apenas um maior tempo entre duas detecções consecutivas.

6.3. Processos Analíticos

Os processos de análise explicados até esta secção são processos de análise meramente experimentais, ou seja, o tratamento e análise dos dados foram baseados na informação recolhida no local de teste. Contudo, é importante estudar, um processo analítico para obtenção das probabilidades de detecção. Para se poder comprar se os resultados obtidos experimentalmente estão de acordo com o estudo analítico. Para o estudo analítico, desenvolveram-se dois processos. O objectivo destes processos analíticos é obter uma expressão que, traduza a probabilidade de detecção de um dispositivo.

6.3.1 Processo analítico de análise por local

Se considerarmos a passagem de um dispositivo (pessoa ou veículo), por um local de detecção, tendo em conta um ambiente de mobilidade, verifica-se que o sucesso na sua detecção depende essencialmente de três parâmetros: a distância percorrida pelo dispositivo dentro da área de *scan* ($\boldsymbol{d}$); o tempo de exposição ao *scan* ($\boldsymbol{t}$); a velocidade de passagem ($\boldsymbol{v}$). Estes 3 parâmetros estão todos relacionados por uma equação:

$$d = v \times t \qquad (\boldsymbol{Eq.\ 7})$$

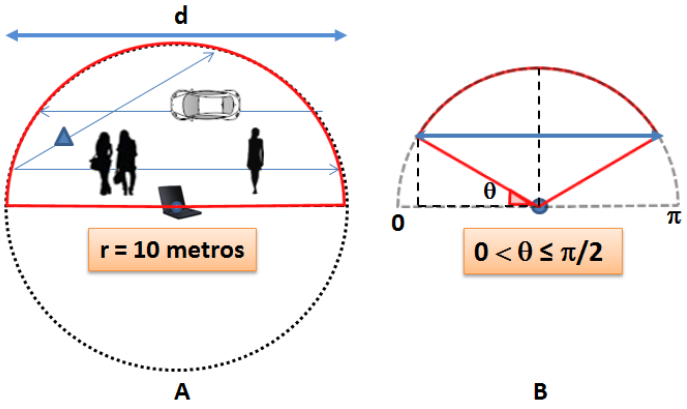


Figura 30: Área de cobertura do *scan*.

Os dispositivos podem realizar vários percursos dentro da área do *scan* representada na Figura 30-A, pela circunferência a tracejado onde o seu raio é de 10 metros (dado que são utilizados dispositivos Bluetooth de classe 2). Assumindo que todos eles passam numa trajectória rectilínea (no semi-eixo da circunferência superior, ou seja do lado da rua), a uma distância média do ponto de *scan*, os dispositivos percorrem uma distância média determinada pela equação 8. O cálculo desta distância média é feito da seguinte forma:

$$\bar{d} = \frac{1}{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} 2r \cos\theta \, d\theta = \left(\frac{1}{\frac{\pi}{2}}\ 2r\right) \sin\theta \Big|_0^{\frac{\pi}{2}} = \left(\frac{1}{\frac{\pi}{2}}\ 2r\right)\left(\sin\left(\frac{\pi}{2}\right) - \sin(0)\right)$$

$$\bar{d} = \frac{4r}{\pi} \qquad (\boldsymbol{Eq.8})$$

Substituindo esta expressão da distância média (*Eq. 8*) na Equação 7, verificamos que o tempo médio de passagem também pode ser obtido:

$$\bar{t} = \frac{4r}{\pi v} \qquad (\boldsymbol{Eq.9})$$

Encontrada a expressão, que nos faculta o tempo médio de passagem, podemos estabelecer uma relação desta com uma outra expressão. Uma expressão que traduza a probabilidade de detecção um dispositivo.

Sabendo experimentalmente, através do teste realizado sem mobilidade, que se a exposição de um dispositivo a um *scan* for superior ou igual a cinco segundos, a sua probabilidade de detecção é de 100%, pode-se estimar, grosseiramente, que a probabilidade de detecção para os restantes tempos de *scan* aumenta linearmente. Deste modo, pode ser definia uma expressão, que traduz a probabilidade de detecção:

$$P_{Detecção}(\mathrm{t}) = \begin{cases} \dfrac{\mathrm{t}}{5}\,, & 0 \leq \mathrm{t} < 5 \text{ seg.} \\ 1\,, & \mathrm{t} \geq 5 \text{ seg.} \end{cases} \qquad (\boldsymbol{Eq.10})$$

Associando a expressão do tempo médio (*Eq. 9*) com a anterior, é obtida a expressão da probabilidade de detecção, em função da velocidade de passagem do dispositivo (v):

$$P_{Detecção}(\nu) = \begin{cases} \dfrac{4\mathrm{r}}{5\pi\nu}\,, & \nu \geq \left(\dfrac{4\mathrm{r}}{5\pi}\right) \\ 1\,, & 0 < \nu < \left(\dfrac{4\mathrm{r}}{5\pi}\right) \end{cases} \qquad (\boldsymbol{Eq.11})$$

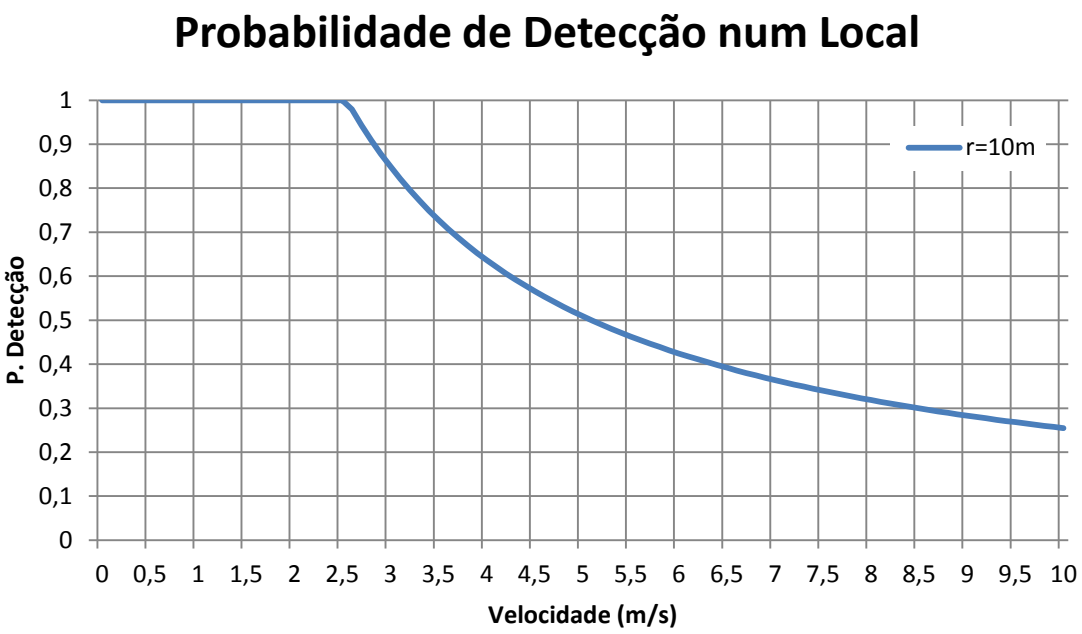


Gráfico 43: Gráfico da função $P_{Detecção}(v)$.

Os valores da probabilidade de detecção, tendo em conta a velocidade a que o dispositivo se move, considerando uma área de *scan* de 10 metros (raio=10m), estão representados no Gráfico 43.

No processo de *scan* de um dispositivo, constata-se também que quanto mais longe se passa do ponto de *scan*, menor é a distância percorrida pelo dispositivo dentro da ”área de *scan”*. A utilização da Equação 11, para determinar a probabilidade de detecção de um dispositivo, não leva este aspecto em conta, uma vez que só depende da velocidade com que o dispositivo passa, dentro da “área de *scan”*. Deste modo, a sua utilização é mais adequada, quando se pretende estudar a detecção de dispositivos, que se possam deslocar por toda a área de *scan*. Como é o caso dos dispositivos transportados pelas pessoas. Observando o Gráfico 43, verifica-se que a probabilidade de detecção é de 100%, caso uma pessoa se desloca com uma velocidade de 2 m/s.

Até este ponto, este processo analítico de análise, só teve em conta a velocidade de passagem dos dispositivos pela área de *scan*. Uma outra forma para se analisar a

probabilidade de detecção de um dispositivo, num local, é considerando o comprimento do percurso que ele efectua dentro da "área de *scan*". Variando o ângulo θ, representado na Figura 30-B, sabe-se a distância percorrida pelo dispositivo, dentro da "área de *scan*". Esta distância é dada pela seguinte expressão:

$$d = 2r\cos(\theta) \qquad (\boldsymbol{Eq.12})$$

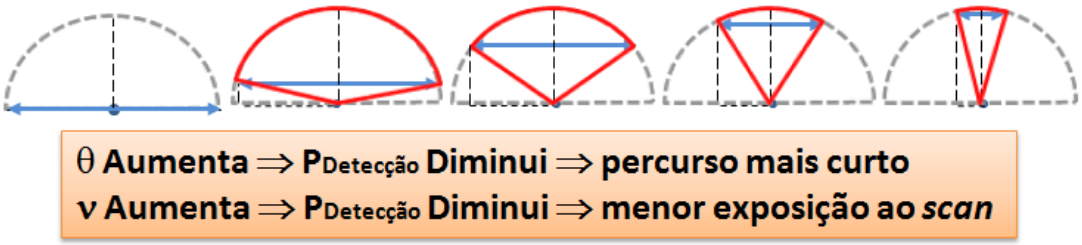


Figura 31: Exemplos de distâncias percorridas.

Na Figura 31, pode-se verificar que quando o ângulo θ aumenta a distância percorrida é menor. Deste modo, mostra-se que, a função da probabilidade de detecção de um dispositivo não depende só da velocidade, mas também da distância percorrida pelo dispositivo dentro da "área de *scan*", que varia consoante o ângulo θ. Assim, recorrendo à Equação 12 e à Equação 7, pode-se definir uma nova expressão:

$$t = \frac{d}{v} = \frac{2r\cos(\theta)}{v} \qquad (\boldsymbol{Eq.13})$$

Substituindo a equação anterior na Equação 10, obtêm-se uma nova função para a probabilidade de detecção:

$$P_{Detecção}(\theta, \nu) = \begin{cases} \dfrac{2r\cos(\theta)}{5\nu}, & \nu \geq \left(\dfrac{2r\cos(\theta)}{5}\right) \\ 1, & \nu < \left(\dfrac{2r\cos(\theta)}{5}\right) \end{cases} \qquad (\boldsymbol{Eq.14})$$

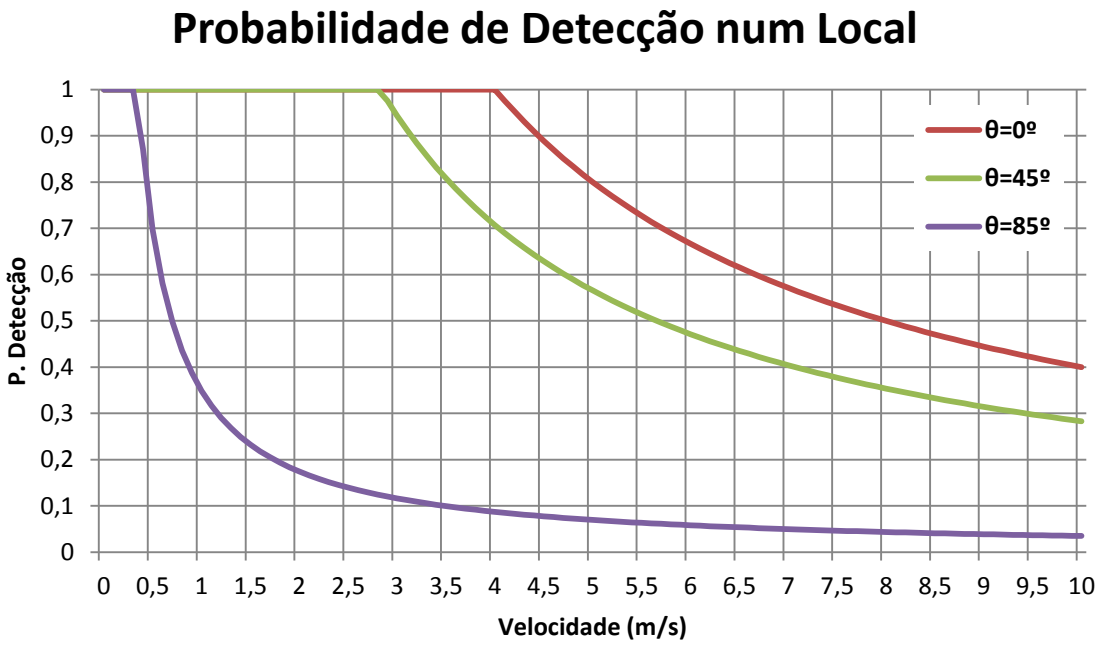


Gráfico 44: Função $Pdetecção(\theta, v)$, para três ângulos diferentes.

Nos Gráfico 44 e Gráfico 45 está representada a função anterior. Ao visualizar-se o Gráfico 44 confirma-se o que é explicado na Figura 31. Sempre que o ângulo θ aumenta, a probabilidade em detectar um dispositivo é menor, à medida que a velocidade de passagem do mesmo aumenta.

Para um ângulo de passagem de 45°, implica um dispositivo realizar um percurso de 14,1 metros (Eq. F) dentro da "área de *scan*". A probabilidade de detecção é bastante próxima da que foi obtida pela equação 11, quando se considerava apenas a velocidade para a probabilidade de detecção.

Também se verifica, no Gráfico 44, que se um dispositivo realizar um percurso de 20 metros dentro da "área de *scan*", ou seja, deslocar-se paralelamente e muito próximo do ponto de *scan*, mesmo com uma velocidade de 8 m/s (28,8 km/h), ainda se obtém uma probabilidade de 50% para a sua detecção.

Por outro lado, quando o dispositivo realiza um percurso curto, dentro da "área de *scan*" (por exemplo, com um ângulo de 85°, percurso de 1,7 metros), mesmo que passe a uma

velocidade quase nula (1 m/s) a sua probabilidade de detecção já é menor do que 50%, cerca de 39%.

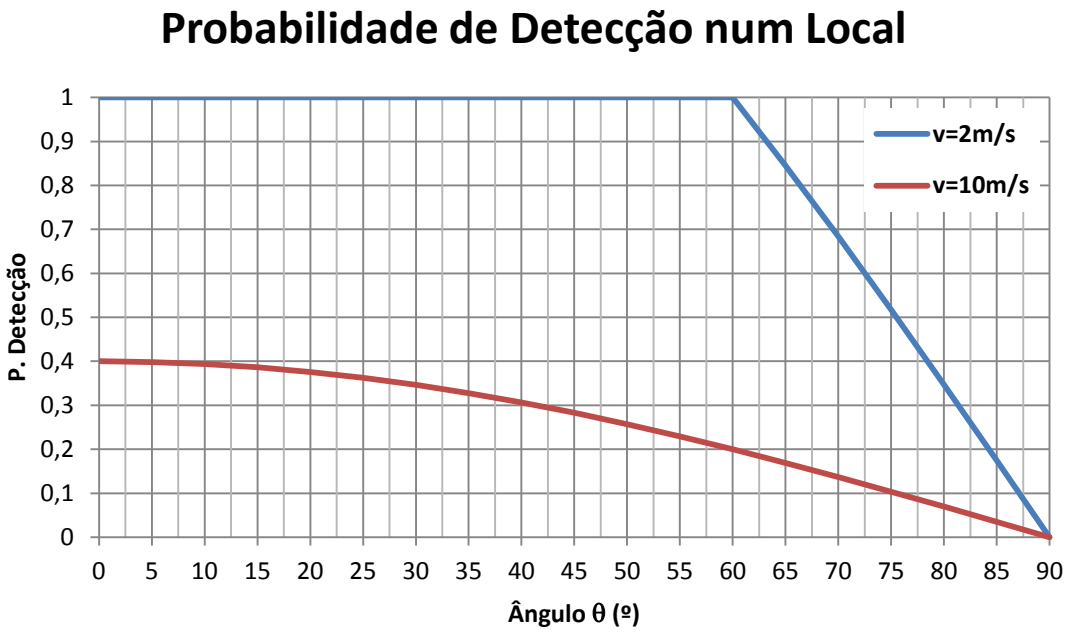


Gráfico 45: Função $Pdetecção(\theta, v)$, para duas velocidades diferentes.

Observando agora o Gráfico 45, pode-se verificar o mesmo comportamento do Gráfico 44. Contudo, neste gráfico, é apresentada a probabilidade de detecção tendo em conta duas velocidades. A velocidade de 2m/s, que é assumida como a velocidade de um dispositivo transportado por um pedestre, ou seja, uma pessoa, e a velocidade de 10 m/s (cerca de 30Km/h), que é assumida como a de um dispositivo transportado por um veículo, se o mesmo estiver equipado com interface Bluetooth.

Da Equação 14 e observando o Gráfico 45, pode-se determinar uma probabilidade de detecção para os dispositivos, que se desloquem apenas numa parte da "área de *scan*". Por exemplo, neste cenário de teste utilizado, uma rua constituída por 2 passeios para pedestres e uma estrada no centro, para os veículos circularem, é possível determinar um valor analítico para uma probabilidade de detecção de um veículo, que circule na estrada.

Sabendo que a largura da rua é de cerca de 9 metros e assumindo que a estrada passa pelo meio da mesma (o que não é bem verdade, pois num dos lados da rua existe

passeio e locais de estacionamento), é possível determinar o valor analítico da probabilidade de detecção para os veículos (ver Figura 32).

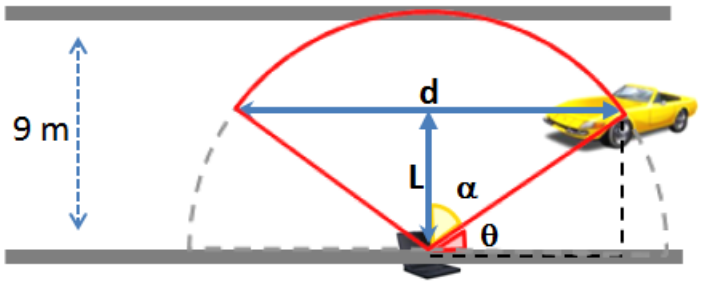


Figura 32: Forma para determinar o ângulo θ .

Admitindo que a estrada passa a 4 metros ($L = 4$) do "ponto de *scan*", o ângulo θ de passagem dos veículos é de 24°. Este ângulo pode ser determinado através da seguinte expressão:

$$L = r\cos(\alpha) \leftrightarrow \alpha = \cos^{-1}\left(\frac{L}{r}\right)$$

$$\alpha = \cos^{-1}\left(\frac{4}{10}\right) \cong 66°$$

$$\theta = 90° - \alpha = 24°$$

Deste modo e consultando novamente o Gráfico 45, verifica-se que o valor analítico, para a probabilidade de detecção de um veículo que passe na estrada é de cerca de 36%. Um outro exemplo é o de cálculo da probabilidade para uma pessoa que se desloque no passeio oposto ao do ponto de *scan*, ou seja, o passeio a seguir à estrada. Se o passeio estiver a uma distância de oito metros do ponto de *scan* ($L = 8$), e assumindo que uma pessoa se desloca no mesmo a uma velocidade de 2m/s, verifica-se que probabilidade de detecção ($\theta = 53°$) é de 100%.

6.3.2 Processo analítico de análise global

Este processo analítico desenvolvido tem como objectivo obter uma probabilidade de detecção, para o conjunto dos três locais de *scan*. Como já tinha sido realizado anteriormente no processo experimental de análise global (ver secção 5.3.2), para um dispositivo ser detectado nos três locais, este tem de realizar um conjunto de sequências de passagem. Estas sequências de passagem que um dispositivo pode realizar são várias, dependendo do número de locais onde ele é detectado. As sequências possíveis de realizar, já explicadas na secção 5.3.2, são as seguintes:

- Sequências completas: (1 → 2 → 3) ou (3 → 2 → 1);
- Sequências incompletas: (1 → 2), (1 → 3), (3 → 2) ou (3 → 1).

Como não é possível saber o percurso que um dispositivo realiza, sem termos dados experimentais, foi criada uma aplicação (Figura 33), que simula a passagem de um conjunto de n dispositivos, tendo em conta os percursos que estes podem realizar. Para tal, é assumido que um dispositivo realiza uma das sequências acima indicadas.

Após realizada a simulação de todos os dispositivos, é obtido um número de sequências completas e incompletas que são usadas para calcular a probabilidade de detecção global. Esta probabilidade é calculada usando a Eq. 4, a relembrar:

$$P_{Dectecção} = \frac{Num.Seq.Completas}{Num.Seq.Completas + Num.Seq.Incompletas}$$

A aplicação desenvolvida permite, realizar várias vezes este processo, isto é, podemos simular a passagem de n dispositivos, N vezes. Como resultado é obtido um conjunto de N valores para a probabilidade de detecção, ou seja, um valor para cada simulação da passagem de n dispositivos.

Por exemplo, imagine-se que se pretende realizar a simulação da passagem de 1000 dispositivos (n=1000) na rua e realizar esta simulação 500 vezes (N=500). O programa realiza 1000 iterações (para simular a passagem dos 1000 dispositivos), encontrando no final o valor para a probabilidade de detecção, valor este que é armazenado. Depois, realizaria todo o processo anterior mais 499 vezes. No final, a aplicação, executa a média aritmética dos 500 valores de probabilidades encontrados e devolve a probabilidade final, que corresponde á probabilidade de detecção global.

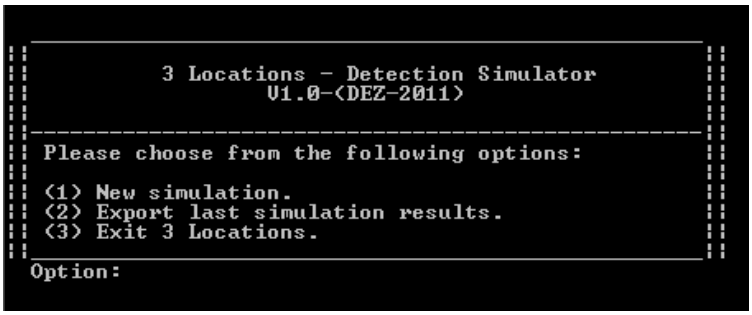


Figura 33: Aspecto do menu da aplicação de simulação.

A decisão se um dispositivo é detectado num local, é feita por intermédio de uma função, que se pode designar de "função de decisão". A "função de decisão", tem o objectivo de gerar, de forma automática, um valor aleatório (0 ou 1), tendo em conta uma probabilidade atribuída. A probabilidade atribuída é a probabilidade de sair o valor '1', que representa a probabilidade de detecção num local. Se a função devolver o valor '0', indica que o dispositivo não foi detectado. Se devolver o valor '1', indica que o dispositivo foi detectado.

A simulação da passagem de um dispositivo é realizada segundo um de dois algoritmos. Para um dispositivo realizar uma sequência completa, isto é, ser detectado em todos os locais, existem 2 casos possíveis. Se o dispositivo realiza a sequência (1 → 2 → 3) é usado o algoritmo representado pelo fluxograma da Figura 34. Caso realize a sequência (3 → 2 → 1) é usado o algoritmo representado pelo fluxograma da Figura 35.

Para se seleccionar qual o algoritmo a utilizar, a aplicação, recorre á "função de detecção". Se a função gerar o valor '0' é usado o da Figura 34, se gerar o valor '1' é usado o da Figura 35. Aqui a probabilidade assumida para a função de detecção gerar um valor, para escolher um dos dois algoritmos, é de 50%. Este valor foi escolhido dado que os dispositivos podem iniciar o seu percurso na rua, começando pelo local 1 ou começando pelo local 3.

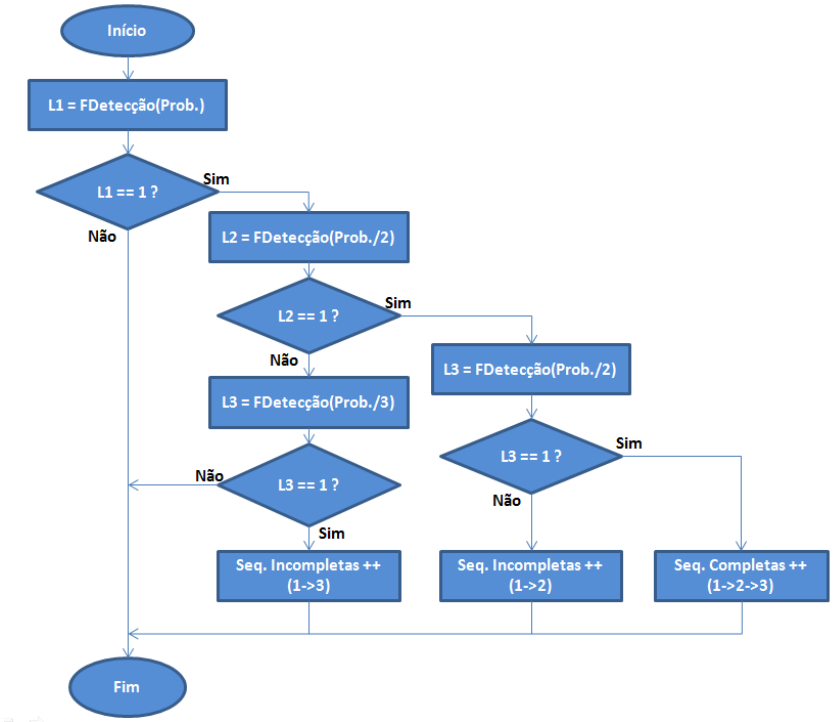


Figura 34: Fluxograma para o algoritmo da sequência (1 → 2 → 3).

Estes algoritmos têm ainda outra particularidade, a alteração do valor da probabilidade de detecção num local, consoante os percursos que um dispositivo pode realizar entre 2 locais. Consideremos as sequências (1 → 2 → 3) e (1 → 3).

Se voltarmos a observar a Figura 16, da secção 5.1, verifica-se que se um dispositivo realizar a sequência (1 → 2 → 3), durante a sua passagem do local 1 para o 2

ele tem duas opções de percurso. O dispositivo pode seguir em frente (o que seria preferível para os testes) ou então também pode ir pela rua de cima entre o local 1 e 2.

Neste caso a probabilidade de detecção em 2, é assumida como metade da probabilidade de detecção num local, uma vez que havia dois percursos possíveis para o dispositivo realizar. O mesmo se passa do local 2 para o 3, o dispositivo pode virar para o parque de estacionamento, antes de ser detectado no local 3.

Por outro lado, se estivermos a simular uma sequência de passagem (1 → 3), o dispositivo tem três percursos possíveis de realizar. Ele pode seguir sempre em frente do local 1 para o 2, pode virar para a rua de cima ou pode ainda virar para dentro do parque de estacionamento. Neste caso a probabilidade de detecção por local é dividida por 3.

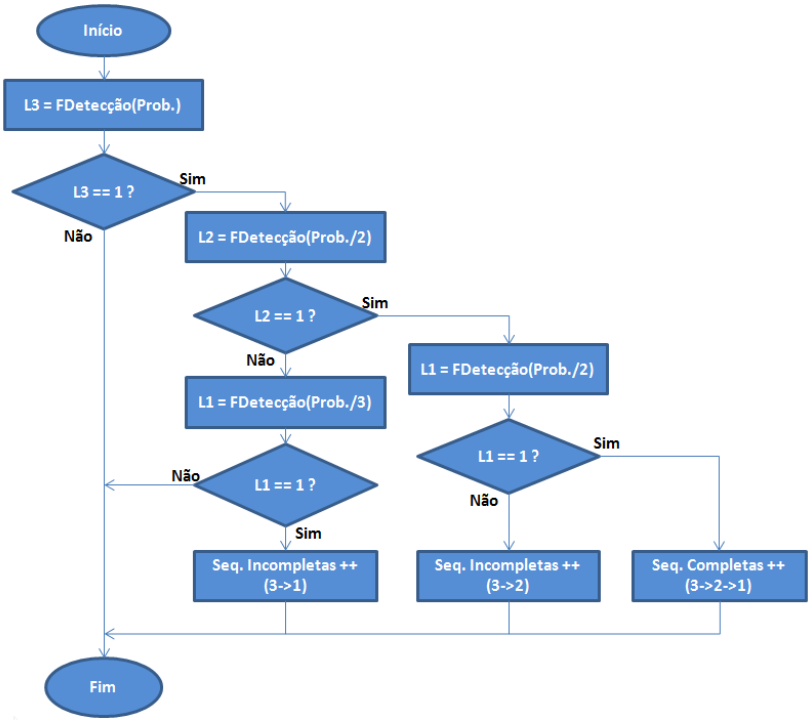


Figura 35: Fluxograma para o algoritmo da sequência (3 → 2 → 1).

No final da simulação da passagem de todos os n dispositivos, é calculada a probabilidade de detecção global, correspondente a uma simulação, conforme já foi

explicado. Depois de todas as N simulações terem sido realizadas, é calculado o valor médio da probabilidade de detecção global.

Este processo de análise da probabilidade geral faz uma simulação de todos os percursos possíveis de realizar pelos dispositivos, não fazendo distinção entre categorias, ou seja, entre dispositivos transportados por pessoas ou veículos. Como a maior parte dos dispositivos detectados são da categoria correspondente às pessoas, será usada, para a função de decisão, a probabilidade teórica calculada na secção anterior, dada pela Equação 11. A equação 11 determina que uma pessoa que se desloque a 2m/s a sua probabilidade de detecção é de cerca de 100% (ver Gráfico 43). Na simulação realizada, esta probabilidade, foi assumida como a probabilidade de detecção num local e foram realizadas 5000 simulações (N=5000) para 4000 (n=4000) passagens de dispositivos, como se pode ver pela Figura 36.

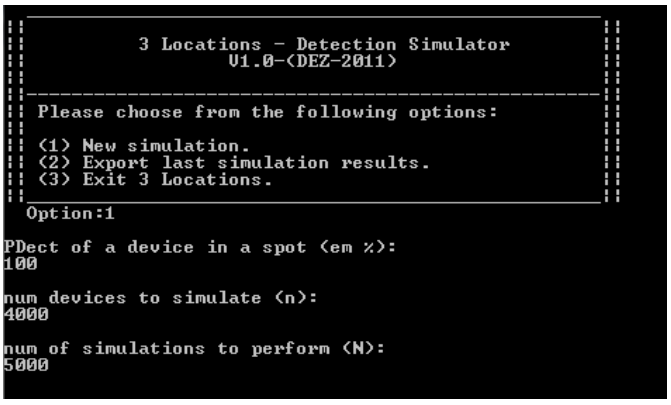


Figura 36: Introdução dos dados na aplicação para simulação.

A probabilidade final obtida para a probabilidade de detecção global foi de 37,5%, como pode ser observado no Gráfico 46. O intervalo de valores, para a probabilidade de detecção, obtido nas simulações situa-se entre os 33,8% e os 40,7%.

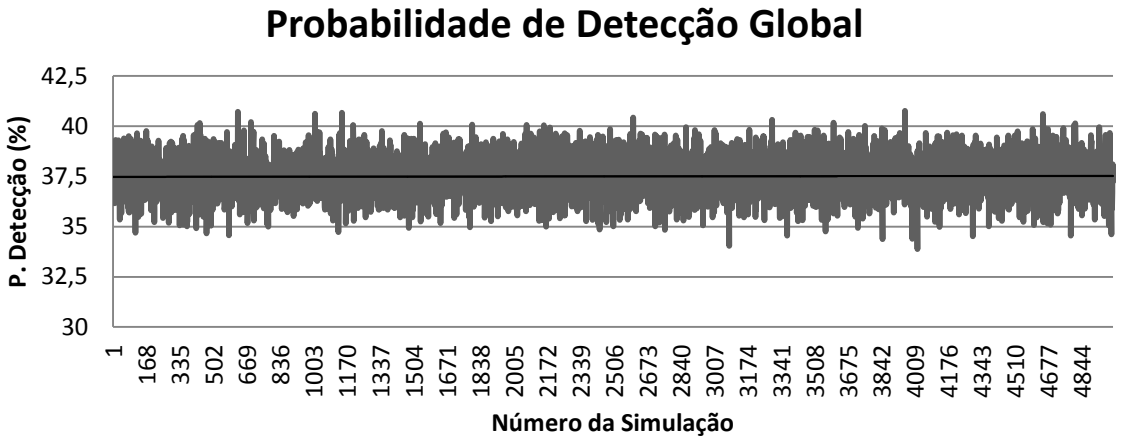


Gráfico 46: Probabilidades de Detecção Global obtidas para as diferentes simulações.

Esta foi a solução encontrada para se poder simular analiticamente a probabilidade de detecção nos 3 locais. O desenvolvimento de um processo analítico que recorresse ao uso de expressões (como foi feito para o estudo analítico por local), para determinar a probabilidade de detecção também foi considerado. O desenvolvimento de uma expressão analítica para a situação em mãos, implicava um estudo mais profundo sobre probabilidades condicionadas, que não se conseguiu realizar. Deste modo, não se conseguiu chegar ainda a uma expressão analítica para a probabilidade de detecção. A simulação foi aqui apresentada, pois tem em conta aspectos que teriam de ser considerados, caso se utilizasse um processo de análise que recorresse ao uso de expressões analíticas.

6.4 Comparação entre os processos de análise

Nesta secção é feita uma comparação dos processos de análise analíticos e experimentais. Serão ainda analisados os diferentes resultados obtidos nos mesmos. Inicialmente é feita uma análise por local e depois a análise global.

6.4.1 Análise por Local

Encontrados os resultados para os processos de análise analítico e experimental é agora possível tirar algumas conclusões. Na Tabela 11 encontra-se o resumo dos resultados obtidos nos dois processos.

Tipo Análise		Probabilidade de Detecção (%)		
Experimental		Classes de Estudo		
Local	Teste	Geral	Veículos	Pessoas
1	1	95,11	98,65	95,02
	2	94,82	94,51	94,86
	3	96,86	97,55	96,85
2	1	95,21	92,18	95,36
	2	94,48	95,89	94,42
	3	95,66	97,57	95,6
3	1	96,03	97,62	95,89
	2	96,19	98,2	96,08
	3	97,16	98,37	97,13
Analítica		N/D	36	100

Tabela 11: Resumo dos resultados obtidos na análise por local.

Comparando os dois resultados obtidos para a análise por local, constata-se que para a classe das pessoas, o processo experimental e o analítico apresentam a mesma conclusão. Isto é, os dispositivos que as pessoas transportam se realizarem um percurso dentro da "área de *scan*", têm uma grande probabilidade de serem detectados. Pode mesmo afirmar-se que a sua probabilidade de detecção se situa entre os 94% e os 100%.

Já para a classe dos veículos isto não se verifica. Apesar do processo experimental indicar que a sua probabilidade de detecção se situa entre os 92% e os 98%, o processo analítico apenas traduz, que a sua probabilidade de detecção é de 36%. Neste caso, o processo analítico é possível que esteja mais próximo da realidade.

Ao analisar-se o processo experimental de detecção por local (ver secção 5.3.1), verifica-se que ele pode não contabilizar alguns casos de falhas nas detecções de dispositivos, uma vez que, para se detectar uma falha é necessário que ocorram pelo menos duas detecções desse dispositivo.

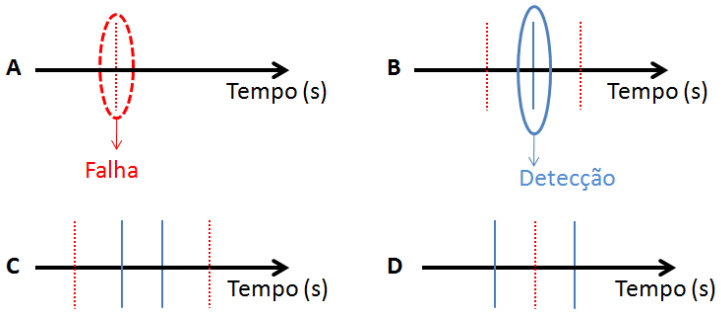


Figura 37: Situações de falha não detectáveis.

Observando a Figura 37 isto facilmente é compreensível. Na figura estão representadas quatro situações A, B, C e D, onde em cada linha temporal estão representadas as detecções de um dispositivo. Por exemplo, o processo experimental da análise por local, não iria detectar as falhas para a situação A, B e C. Só na situação D iria ser detectada uma falha.

Para a classe das pessoas, estas situações de falha, também se podem verificar mas não são muito relevantes, já que uma pessoa realiza percursos mais lentos do que um veículo dentro da "área de *scan*". Deste modo, a probabilidade de serem detectadas num local é maior.

No caso dos veículos este aspecto é importante. Estes passam mais rapidamente pela "área de *scan*" e podem estar mais susceptíveis à ocorrência de uma falha na sua detecção. Apesar destas situações de falha não derem detectáveis, não se pode afirmar que o processo de análise experimental seja deficiente, uma vez que estas situações não

se podem detectar na prática. Somente, se fosse possível, estar a contabilizar fisicamente todos os dispositivos com interface Bluetooth activo, que passassem na "área de *scan"*.

Um outro aspecto que pode ocorrer, pelo método do processo de análise experimental, é o estacionamento de um veículo dentro da "área de *scan*" (uma vez que a rua possui locais de estacionamento). Naturalmente, pelo processo experimental, que o número de falhas para o mesmo seria menor, dado que o memo passaria mais tempo na "área de *scan"*, aumentando assim a sua probabilidade de detecção.

São estes dois factores que levam a que a probabilidade de detecção dos veículos seja mais elevada, quando usado o processo experimental.

6.4.2 Análise Global

Comparando os valores obtidos para a probabilidade de detecção global, usado o processos analítico (que neste caso é uma simulação) e experimental, verifica-se que foram obtidos valores similares para a probabilidade de detecção global.

No processo experimental (secção 6.2.4.2) são determinadas várias probabilidades, onde é levado em conta o tempo de passagem de um dispositivo entre locais (Δt). Verificou-se que a partir de um Δt de 120 segundos, na categoria geral (onde se estuda a probabilidade de detecção de todos os dispositivos detectados), o valor da probabilidade situa-se entre os 34% e os 43%.

Já para o processo analítico, descrito na secção 6.3.2, os valores para a probabilidade de detecção situam-se entre os 33,8% e os 40,7%.

Sobrepondo os intervalos de valores obtidos, pelos dois processos (analítico e experimental), para a probabilidade de detecção (ver Figura 38) verifica-se que se obtêm um valor médio para a probabilidade de detecção de 37,35%. Este valor médio obtido é

muito próximo do valor médio que se obteve na simulação realizada, que foi o valor de 37,5% para a probabilidade de detecção global.

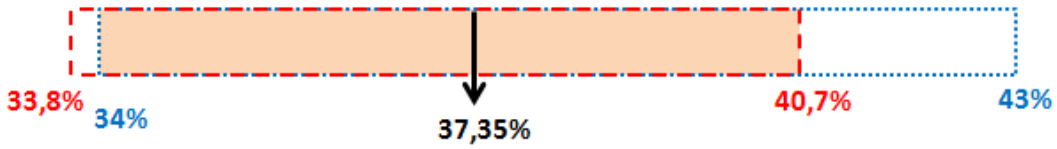


Figura 38: Valores da probabilidade de detecção global.

Neste "processo analítico" como este valor é simulado, e não é assumido nenhum Δt para um dispositivo se deslocar de um local a outro.

Se observarmos a Figura 39, vemos um conjunto percursos possíveis para um dispositivo realizar na prática. Os percursos a linha tracejada, não são assumidos por nenhum dos dois processos de análise, nem pelo experimental nem pelo analítico (simulação), uma vez que esse percurso levaria à realização de duas sequências não válidas no estudo, as sequencias (2 → 1) e (2 → 3).

As linhas a ponteado representam, para o processo de análise analítico, duas sequências incompletas (1 → 3) e (1 → 2).

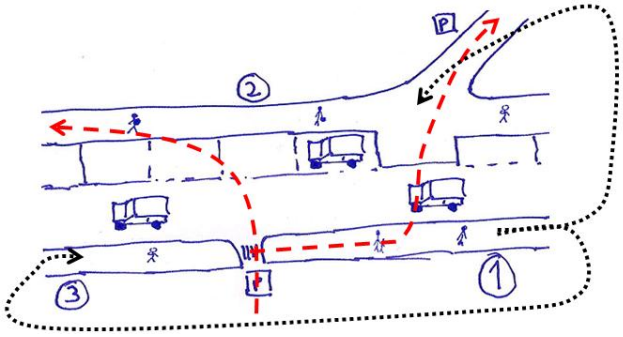


Figura 39: Exemplos de percursos realizados por dispositivos.

No entanto, para o processo experimental, que leva em conta um Δt para um dispositivo se deslocar de um local ao outro, elas podem não ser consideradas como sequências

incompletas. Se não respeitarem a condição do limite de tempo entre locais, não entrando assim para o cálculo da probabilidade de detecção global.

Desta forma e como o processo analítico (simulação), não leva em conta nenhum Δt para um dispositivo se deslocar de um local a outro, faz com que o número de sequencias de passagem completas e incompletas seja mais elevado.

Como os resultados obtidos nos dois processos de análise estão próximos, a conclusão a que se chega é que a probabilidade de detecção de um dispositivo pelos três locais de *scan* é relativamente baixa. Este fenómeno, já seria de se esperar, uma vez que o local de teste possui, para os dispositivos, várias “escapatórias de percurso”.

Era interessante realizar este teste num local, que oferecesse elevada mobilidade para os dispositivos e onde os mesmos tivessem obrigatoriamente, de se deslocar pelos três locais de *scan*, de modo a se poderem verificar que diferenças existiam para os valores da probabilidade de detecção. □

7. Integração na Aplicação Epi

Este capítulo realiza uma breve explicação do processo de integração do módulo Bluetooth, desenvolvido para a aplicação Epi. Depois de realizado todo o trabalho apresentado nos capítulos anteriores, a aplicação de *scan* de dispositivos construída foi adaptada para que fosse possível "encaixar" parte da mesma, como um módulo, na aplicação Epi.

Ao longo das próximas secções é feita uma pequena descrição da aplicação Epi. Na primeira secção são descritas as novas funcionalidades introduzidas e modificações que foram efectuadas na aplicação do Epi, protocolo de comunicações e servidor. Nas restantes secções é explicado como foi construído o módulo Bluetooth, e ainda apresentada uma análise dos dados recolhidos com o módulo Bluetooth.

7.1. Aplicação Epi

O Epi é uma aplicação que se encontra actualmente em desenvolvimento no seio do grupo Ubicomp da Universidade do Minho, no âmbito do projecto SUM (*Sensing and Understanding human Motion dynamics*).

Esta aplicação proporciona aos utilizadores de redes WiFi a troca de mensagens de texto entre utilizadores, que se encontrem próximos**,** mesmo sem conexão à Internet**.**

As mensagens trocadas num determinado local são armazenadas e novamente difundidas em outros locais para onde quer que o Epi se desloque, criando assim uma espécie de difusão epidémica das mesmas. A Figura 40, retirada de [23], pretende ilustrar a arquitectura e o princípio de funcionamento da aplicação Epi.

A aplicação não permite enviar mensagens a um utilizador em específico, dado que o envio é feito em *broadcast,* na rede IP em que o utilizador se encontra ligado. No processo de troca de mensagens, um utilizador é considerado na vizinhança de outro se obedecer a uma condição estabelecida por uma função de proximidade.

Esta função é uma operação matemática efectuada sobre duas assinaturas[5] de rádio: a assinatura de rádio anexada à mensagem na origem e uma outra assinatura de rádio recolhida no instante da recepção da mensagem. A assinatura de rádio é a lista dos *Access Points* WiFi que se encontram na vizinhança quando é efectuada a recolha.

Os utilizadores podem fazer uso das funcionalidades da aplicação quando se encontram ligados a uma rede WiFi. O tipo de rede WiFi que o utilizador usa é indiferente, pode estar ligado a uma rede em modo *ad-hoc* (que pode ser criada pelo próprio) ou a uma rede em modo infra-estrutura (normalmente a um *Access Point* - AP). Em qualquer um dos casos, o utilizador não necessita de ter acesso à Internet.

Quando recebe ou envia uma mensagem, realiza a recolha de uma assinatura de rádio que é armazenada. Mais tarde, quando a aplicação se encontra na presença de uma ligação à Internet, a informação do ambiente de rádio armazenada é enviada para um servidor.

[5] Uma assinatura é o conjunto de informação que é recolhido do ambiente de rádio, num determinado instante ou intervalo de tempo. Por exemplo, uma assinatura Bluetooth pode conter várias amostras.

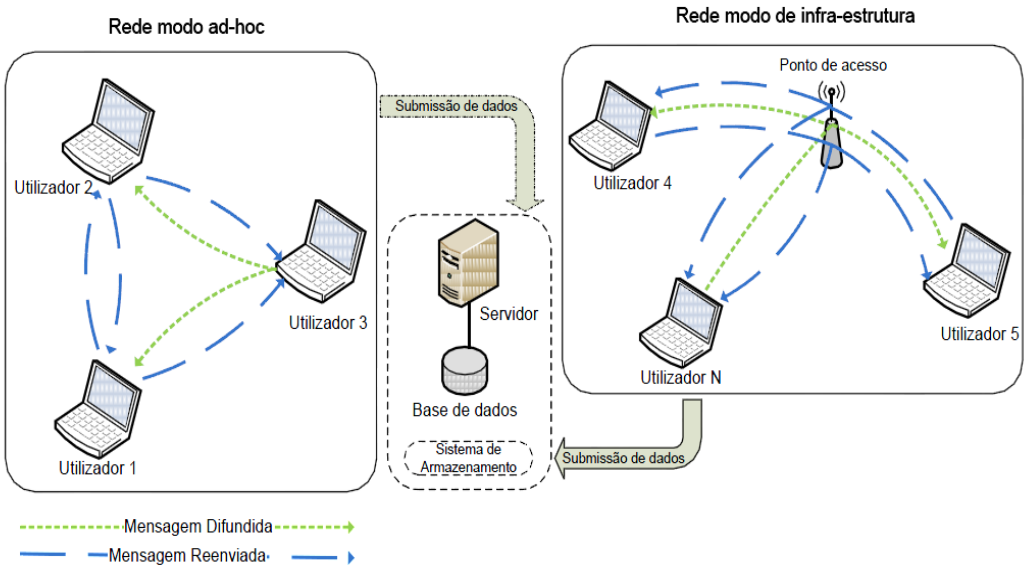


Figura 40 - Princípio de funcionamento da aplicação Epi.

Relativamente à questão da privacidade da informação, armazenada e enviada para o servidor, não são recolhidos dados relativos ao conteúdo das mensagens de texto trocadas entre utilizadores. Nem mesmo o nome de utilizador que envia uma mensagem é armazenado. No entanto é recolhido um valor, recorrendo a uma função de *Hash*, determinado pelo texto da mensagem. Este valor é usado para detectar a difusão sucessiva da mesma mensagem (efeito epidemia).

A restante informação enviada juntamente com a assinatura de rádio é a informação temporal e o endereço físico da placa de rede de onde foi efectuada a recolha da assinatura de rádio. Este é o único dado armazenado que pode ser associado ao utilizador.

7.1.1. Novas Funcionalidades Desenvolvidas

O objectivo desta dissertação, no que concerne à aplicação Epi, é a inclusão de um novo módulo que realize uma recolha (apenas periódica), de um novo tipo de assinatura de rádio. No âmbito desta dissertação, o novo módulo a incluir é um que

realize a recolha de informação relativa a dispositivos Bluetooth que se encontrem na vizinhança. A este novo módulo é dado destaque na próxima secção.

A par deste módulo foi também desenvolvido um outro módulo, no âmbito de uma outra dissertação, que realiza a recolha de informações relativas a estações WiFi.

As informações recolhidas por estes dois novos módulos, em conjunto com o módulo já existente (modulo de recolha de informação dos APs vizinhos), são periodicamente submetidas para um servidor, onde são armazenadas numa base de dados para depois serem consultadas e analisadas.

Os detalhes da arquitectura original da aplicação Epi, função de proximidade e módulo de recolha de informação APs na vizinhança, encontram-se no documento de dissertação *Difusão Epidémica de Mensagens em Hotspots WiFi* [23].

Os detalhes do módulo que realiza a recolha de informações relativas a estações WiFi, encontram-se no documento de dissertação *Descoberta Passiva de Estações em Redes 802.11* [24].

Com a inclusão destes novos módulos na aplicação, foi necessário realizar algumas alterações no código da mesma e consequentemente no protocolo de comunicação com o servidor, de forma a suportar os novos tipos de assinaturas de rádio.

No servidor foram também necessárias alterações para suportar as modificações introduzidas pelo novo protocolo, e consequentemente na base de dados, para armazenar os dados referentes aos novos tipos de assinaturas de rádio recolhidas.

Uma vez que foram realizadas todas estas alterações na aplicação, achou-se que seria interessante desenvolver uma nova interface gráfica. Apesar de se incluírem novos módulos de recolha de informação (neste caso os módulos do Bluetooth e das estações

WiFi), estes são transparentes do ponto de vista do utilizador. Com a nova interface gráfica, pretende-se oferecer uma melhor *user experience*, e desta forma cativar mais utilizadores a instalar e usar a aplicação. O novo interface gráfico foi desenvolvido no âmbito de outra dissertação de mestrado. A janela principal da nova interface gráfica está ilustrada na Figura 41.

Figura 41 – Nova interface gráfica do Epi.

Também se achou interessante adicionar uma nova funcionalidade, que permita a um utilizador saber a lista de utilizadores com os quais interagiu mais vezes. Essa lista é armazenada na estrutura de armazenamento de ”vizinhos”(EAV).

Para suportar esta nova funcionalidade, foi necessário adicionar um novo campo às mensagens do protocolo usado na comunicação entre clientes. Este novo campo é o endereço MAC da placa de rede dos utilizadores, pois só assim é possível identificar com que utilizadores interagiu cada utilizador.

Depois das alterações efectuadas ao projecto Epi, a aplicação apresenta a estrutura representada pelo diagrama de blocos da Figura 42, em que as alterações efectuadas estão assinaladas a negrito. Pode-se observar que a aplicação contínua a ser controlada, tendo como ”pivô”, o Gestor Principal. Este bloco é o responsável por fazer a ligação entre os diferentes interfaces da aplicação, o interface do utilizador, o interface

de rede (WiFi) e ainda o novo interface Bluetooth. Este último foi o interface onde foi realizada a inclusão do novo módulo de recolha de assinaturas de rádio Bluetooth. O módulo de recolha de assinaturas de rádio de estações WiFi foi incluído no interface de rede, uma vez que usa o mesmo interface WiFi que os restantes.

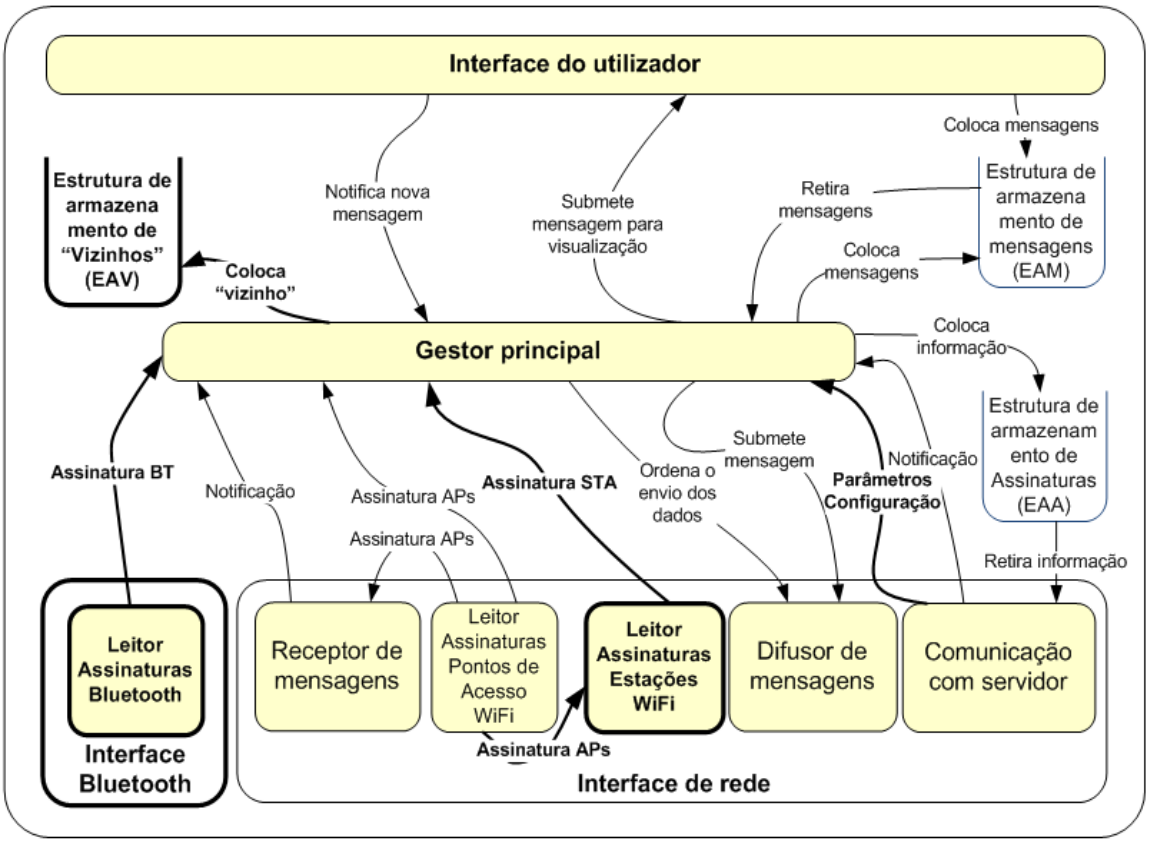


Figura 42 - Digrama de Blocos da aplicação Epi.

Ao iniciar a aplicação, o Gestor principal é responsável por iniciar um conjunto de objectos, a thread que vai receber as comunicações de outros clientes e ainda três temporizadores. Os temporizadores são responsáveis por periodicamente chamar as funções de difusão das mensagens armazenadas, comunicação com o servidor e recolha de assinatura de rádio das redes WiFi na vizinhança.

De forma a chamar os novos módulos, foram adicionados dois novos temporizadores no início de execução da aplicação, que periodicamente chamam uma função responsável pela recolha das novas assinaturas de rádio. A informação das recolhas é depois armazenada na estrutura de assinaturas (EAA).

O período de tempo predefinido para cada temporizador está representado na Tabela 12, no entanto estes podem ser alterados numa das operações que acontece entre cliente e servidor. Sempre que um cliente comunica com o servidor efectua três operações distintas. Primeiro, verifica se existe uma nova versão da aplicação disponível. Segundo, verifica os parâmetros de configuração. Por fim, verifica se tem assinaturas de rádio armazenadas, que em caso positivo são imediatamente enviadas uma a uma para o servidor. As três operações são efectuadas pela ordem respectiva sempre que é aberta uma ligação ao servidor, e se alguma delas falha as seguintes já não acontecem. A comunicação com o servidor é realizada periodicamente ou quando o utilizador verifica se existe uma actualização disponível da aplicação.

Temporizador	Primeira Execução (Seg.)	Período Das Próximas Execuções (Seg.)
Módulo APs	8	1800
Módulo Sta WiFi	15	1800
Módulo Bluetooth	10	1800
Comunicação Servidor	5	3600
Difusão de Mensagens	900	900

Tabela 12 - Valores de Tempo predefinidos nos temporizadores.

A operação do cliente contactar o servidor para verificar os parâmetros de configuração impostos pelo servidor é também uma nova funcionalidade adicionada. Esta também é transparente para o utilizador. Os parâmetros de configuração incluem o período de tempo dos temporizadores, a activação e desactivação dos novos módulos e o valor de configuração da função de proximidade. Para a função de proximidade, os valores predefinidos mantêm-se a 20[23] e os novos módulos estão activados.

7.1.2. Servidor

O servidor do Epi, é um Servlet[6] a correr num servidor *Apache Tomcat*, que recebe pedidos HTTP POST dos clientes para efectuar operações específicas. O servidor identifica a interacção que o cliente quer efectuar através do valor do campo de função (”**f**”) no corpo do pedido POST.

De forma a suportar a recepção de novas assinaturas e a verificação dos parâmetros de configuração, foram adicionadas três novas funções ao servidor às duas funções **f1** e **f2** que já existiam [23]. As funções **f3** e **f4** para a recepção de assinaturas de rádio Bluetooth e assinaturas de rádio das estações WiFi, respectivamente. A função **f5** para efectuar a verificação dos parâmetros de configuração.

Os pedidos HTTP POST para cada uma das novas funções é processado da seguinte forma:

Função F3

O corpo POST neste pedido enviado ao servidor deverá incluir os campos:

- **f** - identifica a função que se pretende invocar, que neste caso é ”f3”;
- **ats** - instante de tempo no cliente em que o pedido HTTP POST é criado, o formato é do tipo “yyyy:MM:dd:HH:mm:ss; ±xx:zz”, em que yyyy representa o ano, MM representa o mês, dd representa o dia, mm representa os minutos, ss representa os segundos e ±xx:zz representa o desfasamento horário existente em várias zonas do mundo;
- **sgt** - identifica o tipo de assinatura de rádio a que se refere o pedido, “1” para a assinatura de rádio das redes WiFi, “2” para a assinatura de rádio Bluetooth e “3” para a assinatura de rádio das estações WiFi;
- **sts** - instante de tempo em que a assinatura rádio foi recolhida, o formato é do tipo “yyyy:MM:dd:HH:mm:ss”, em que yyyy representa o ano, MM representa o mês, dd representa o dia, mm representa os minutos, ss representa os segundos;

[6] É um módulo que estende a funcionalidade de um servidor Web, através de módulos de aplicação implementados em linguagem Java.

- **clm** - endereço MAC da máquina de onde a mensagem foi enviada, o formato é ``XX:XX:XX:XX:XX:XX'', onde XX representa um byte em formato hexadecimal (00 a FF);
- **has** - transporta o *Hash* da mensagem enviada ou recebida. É representado por uma *string* com 32 caracteres hexadecimais. No caso do pedido não se referir a uma mensagem, este campo deve conter a *string* "nomsgnomsgnomsgnomsgnomsgnomsgno";
- **mts** - instante de tempo em que a mensagem foi criada, o formato é do tipo "yyyy:MM:dd:HH:mm:ss; ±xx:zz", em que yyyy representa o ano, MM representa o mês, dd representa o dia, mm representa os minutos, ss representa os segundos e ±xx:zz representa o desfasamento horário existente em várias zonas do mundo;
- **mac** - lista com os dispositivos Bluetooth. Este campo será representado por uma *string* com a lista de endereços MAC, separados pelo carácter ";". Cada endereço MAC deve ser transmitido na forma "XX:XX:XX:XX:XX:XX", onde XX representa um byte em formato hexadecimal (00 a FF);
- **dev** - lista com o nome dos dispositivos Bluetooth, separados pelo carácter ";";
- **tds** - lista com o instante de tempo de descoberta de cada dispositivo, separados pelo carácter ";". O formato de cada instante de tempo é do tipo "yyyy:MM:dd:HH:mm:ss", em que yyyy representa o ano, MM representa o mês, dd representa o dia, mm representa os minutos, ss representa os segundos;
- **cls** – lista com o nome das classes dos dispositivos, separados pelo carácter ";";
- **srv** – lista com o nome dos serviços disponibilizados por cada dispositivo, separados pelo carácter ";". Para cada dispositivo os serviços encontram-se separados pelo carácter ",";
- **rsi** – lista com os valores de RSSI de cada dispositivo, separados pelo carácter ";". Este campo será representado por uma *string*, onde caso o seu valor seja igual a *Null*, indica que esse dispositivo foi o que realizou o *scan* dos restantes.

Para este pedido o servidor deve responder com uma das seguintes mensagens:

- **001;record_id** - significa que o pedido foi aceite e que os respectivos dados foram armazenados na base de dados com o número do registo representado por *record_id*;
- **100;description** - significa que o pedido não foi aceite devido a algum erro interno do servidor, o erro é descrito por *description*;
- **101** - Significa que o pedido não respeita o formato definido e que, por isso, os dados não foram armazenados na base de dados;
- **102;description** - significa que o pedido respeita o formato definido mas que não foi possível armazenar estes dados na base de dados, sendo o erro descrito por *description*.

Função F4

- **f** - identifica a função que se pretende invocar, que neste caso é “f4”;
- **ats**, **sgt**, **sts**, **clm**, **has** e **mts** - estes campos nesta função são comuns aos campos da função f3 descrita em cima;
- **ngm** - lista com as estações WiFi detectadas. Este campo será representado por uma *string* com a lista de endereços MAC, separados pelo carácter “;”. Cada endereço MAC deve ser transmitido na forma “XX:XX:XX:XX:XX:XX”, onde XX representa um byte em formato hexadecimal (00 a FF).

Para este pedido o servidor deve responder com mensagens, que são iguais às da função f3.

Função F5

O corpo POST neste pedido enviado ao servidor inclui apenas o campo de função ”f” com o valor **f5**. A este pedido, o servidor deverá responder um conjunto de parâmetros separados por ”**#**”, em que cada parâmetros é constituído pelo seu nome e

valor, como ilustra a Figura 43. Os nomes dos parâmetros introduzidos nesta versão reconhecidos pelo cliente são os seguintes:

- **simthreshold** - Valor da métrica associado à função de proximidade;
- **mod_wifista** - Activação/Desactivação do módulo de recolha de assinaturas de rádio de estações WiFi, "*enable*" para activar e "*disable*" para desactivar;
- **mod_bt** - Activação/Desactivação do módulo de recolha de assinaturas de rádio Bluetooth, "*enable*" para activar e "*disable*" para desactivar;
- **int_commsrv** - Valor em segundos para o temporizador que chama a função para comunicação com o servidor;
- **intscan_wifista** - Valor em segundos para o temporizador que chama a função de recolha de assinaturas de rádio de estações WiFi;
- **intscan_bt** - Valor em segundos para o temporizador que chama a função de recolha de assinaturas de rádio Bluetooth;
- **intscan_wifiaps** - Valor em segundos para o temporizador que chama a função de recolha de assinaturas de rádio das redes WiFi.

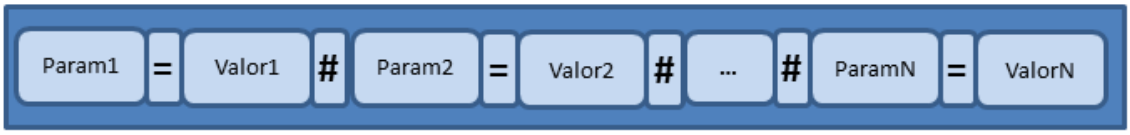


Figura 43 - Resposta à função F5.

Quando o cliente consegue estabelecer comunicação com o servidor, nas operações de verificação da versão da aplicação e parâmetros de configuração, o servidor faz consultas à base de dados onde é guardada essa informação. Na operação em que o cliente envia uma a uma, as assinaturas de rádio que têm armazenadas na EAA, o servidor recebe a informação de cada assinatura, verifica-a e depois armazena-a na base de dados.

A base de dados também sofreu alterações para suportar o armazenamento das novas assinaturas e parâmetros de configuração. O modelo da base de dados encontra-se ilustrado na Figura 44.

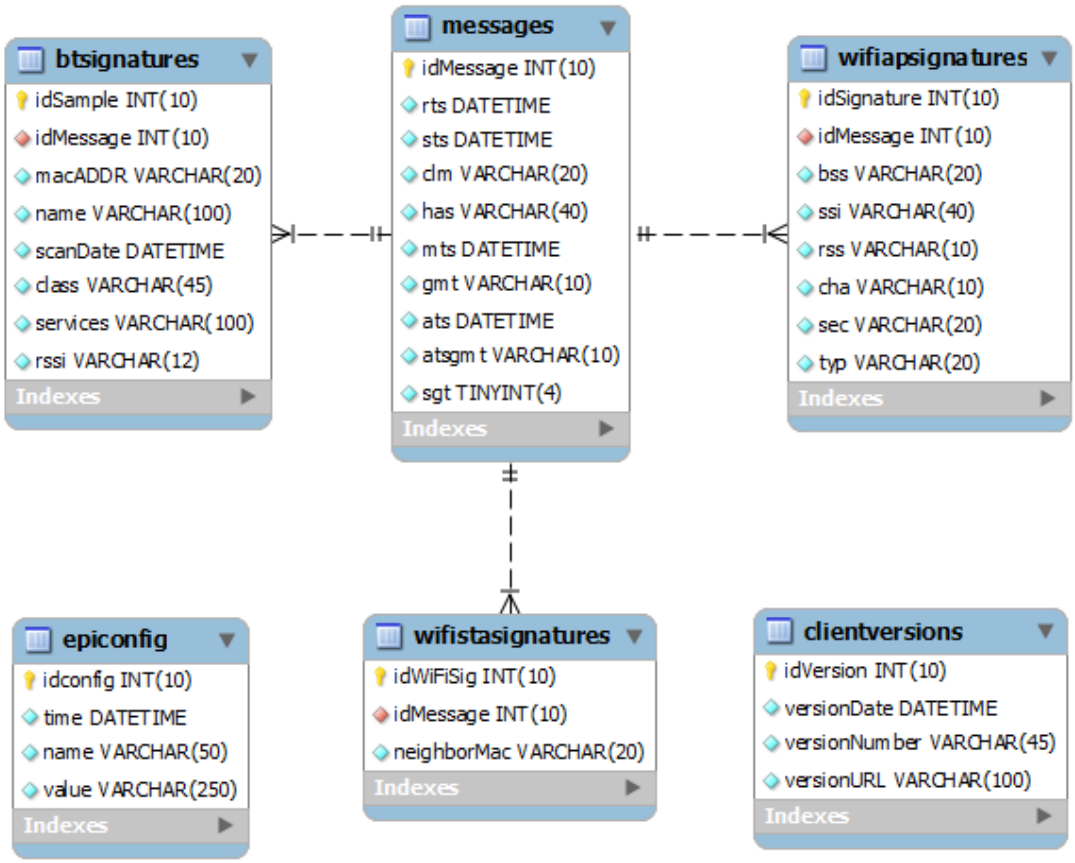


Figura 44 - Tabelas da base de dados do servidor.

Como se pode ver na figura, a base de dados é composta por um conjunto de seis tabelas. A tabela *clientversions*, armazena informação relativa à versão da aplicação cliente que os utilizadores usam. Já tabela *epiconfig* armazena informação relativa ao parâmetros de configuração da aplicação cliente. Esta foi construída de forma a manter um histórico das configurações, introduzidas pelo administrador do lado do servidor. As tabelas *btsignatures*, *wifistsignatures* e *wifiapsignatures*, armazenam as assinaturas de rádio que foram recolhidas pelos 3 módulos na aplicação cliente.

Por fim, *Messages* é a tabela que guarda as informações relativas às mensagens trocadas pelos utilizadores, que servem como suporte para o envio das informações de

recolha das assinaturas de rádio. Nesta tabela existem dois campos importantes o *idMessage* e o *has*. O primeiro campo, para além de chave primaria nos registos desta tabela, serve para referenciar nas tabelas *btsignatures*, *wifistsignatures* e *wifiapsignatures*, qual a mensagem que enviou a respectiva assinatura. O campo *has*, contém o *hash* do texto da mensagem que originou a recolha da assinatura de rádio. O conteúdo do campo também permite identificar se a assinatura de rádio é proveniente de uma mensagem ou de uma recolha periódica.

7.1.3. Protocolos de Comunicação

Nas mensagens enviadas entre clientes, o endereço de destino dos pacotes é o endereço de *broadcast* e o protocolo de transporte usado é o UDP. Na Figura 45 está representado o formato de mensagem enviada ao nível de aplicação.

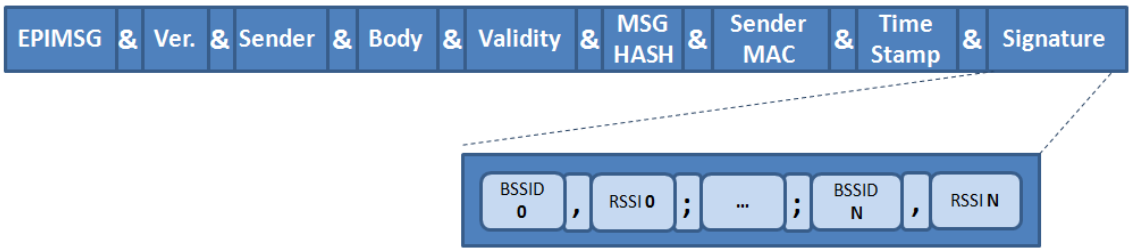


Figura 45 - Formato de mensagens entre clientes Epi.

O campo "Sender MAC" contém o endereço MAC da placa de rede do utilizador que enviou a mensagem. Este campo é a única alteração efectuada ao protocolo original [23].

A troca de mensagens entre cliente e servidor é diferente, o protocolo usado ao nível de transporte é o TCP. A listagem do anexo A4 mostra o exemplo do envio de uma assinatura de rádio de estações Bluetooth, do cliente ao servidor. A informação enviada pelo cliente ao servidor (linhas 10-21 da listagem), não tem uma ordenação específica. Trata-se apenas de uma colecção de informação com pares nome - valor. A colecção de dados esperada pelo servidor varia consoante a função, como já foi descrito na secção 7.1.2.

7.2. Módulo Bluetooth

O módulo Bluetooth que foi inserido, na aplicação Epi, é parte da aplicação que foi desenvolvida para realizar o *scan* dispositivos, relatado nos capítulos anteriores.

As informações que o módulo recolhe são exactamente as mesmas que já haviam sido definidas no desenvolvimento da aplicação de *scan*. Essas informações encontram-se descritas na secção 4.2, deste documento. Sempre que um utilizador tiver o seu hardware Bluetooth activo no computador são realizados os *scans*. Caso contrário o modulo não é activado.

Nesta aplicação Epi, o módulo Bluetooth é composto por duas classes: a classe *BluetoothScans* e *BluetoothCom*. Na Figura 46 encontra-se o diagrama de classes deste módulo. No Anexo A3 encontram-se representadas estas duas classes com maior detalhe, uma vez que na Figura 46 apenas se pretende indicar as variáveis e os métodos mais relevantes de cada classe. Nas secções seguintes é descrita cada classe.

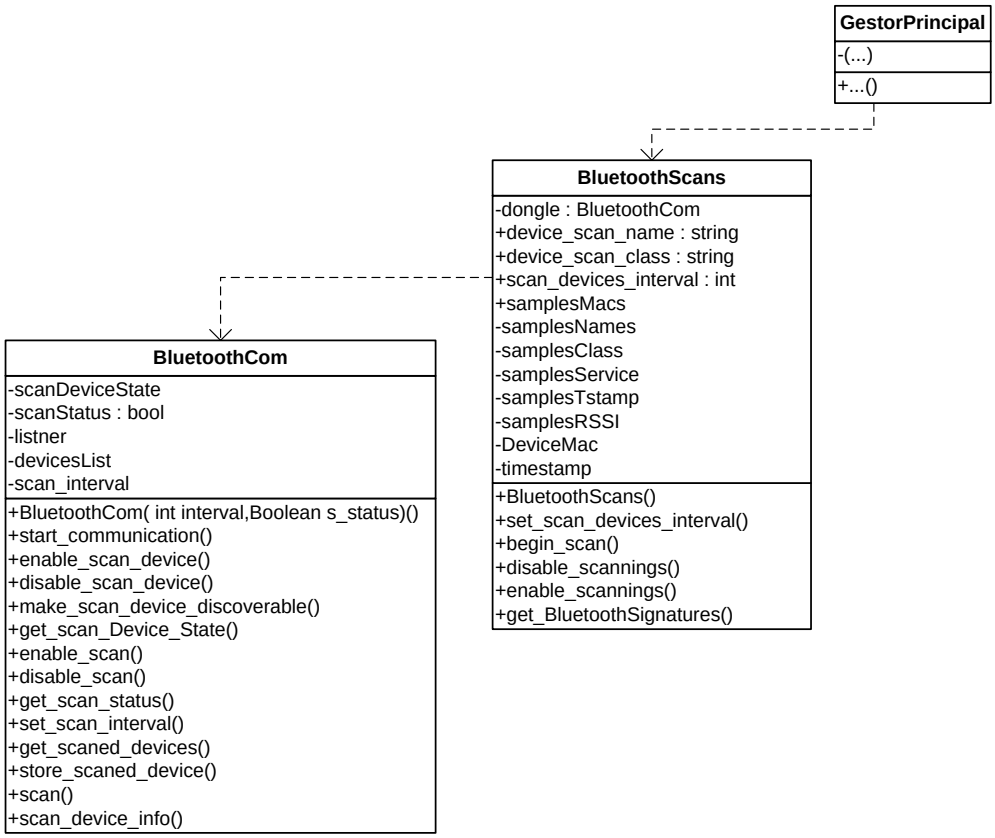


Figura 46 - Diagrama de Classes do Módulo Bluetooth.

7.2.1. Classe *BluetoothCom*

A classe *BluetoothCom* é basicamente igual à mesma que já foi explicada na secção 4.4.2, como tal não é aqui realizada toda a sua discrição. Na aplicação de *scan* desenvolvida, quando se realizava um *scan,* ao recolher uma amostra[7] era criado um objecto do tipo *ScanedBTDevice*, e lá eram armazenadas as informações da mesma onde posteriormente eram colocadas numa lista na classe *BluetoothScan.*

Agora, neste módulo Bluetooth, as informações do *scan* são recolhidas e colocadas directamente num conjunto de listas na classe *BluetoothScans*, onde existe uma lista para cada parâmetro das amostras, como é explicado mais á frente. Nesta classe foi adicionado ainda o método *scan_device_info()*, que devolve informações relativas ao dispositivo que realiza o *scan*. Foi eliminada a variável *scan_thread*, uma vez que na classe *GestorPrincipal* da aplicação Epi, já existe um temporizador que acciona uma *thread* para o processo de *scan*, não sendo necessário a criação de uma nova para o mesmo propósito.

7.2.2. Classe *BluetoothScans*

A classe *BluetoothScans,* sofreu algumas alterações que convêm explicar. Como já foi dito na secção 7.1.1, o bloco (classe) *GestorPrincipal* da aplicação Epi, lança um conjunto de três temporizadores que activam as *threads* dos diferentes módulos de recolha de informação. Quando a *thread* correspondente aos *scans* Bluetooth é lançada, um objecto do tipo *BluetoothScans()* (anteriormente já instanciado pela classe *GestorPrincipal*) chama o método *begin_scan()* da classe *BluetoothScans*, para iniciar um novo *scan* de dispositivos. Este *scan* é feito com o auxílio dos métodos da classe *BluetoothCom*.

Relativamente às informações que são recolhidas, cada parâmetro de uma amostra é inserido na sua respectiva lista (*samplesMacs*, *samplesNames*, etc.).

[7] Uma amostra é um conjunto de informações relativas a um dispositivo Bluetooth. Uma assinatura Bluetooth pode conter várias amostras.

Cada vez que é realizado um *scan*, é armazenado nas variáveis *timestamp* e *DeviceMAC*, a data/hora da realização desse *scan* bem como o endereço MAC do dispositivo que realizou os *scans*.

Quando o *scan* terminar, a execução volta para a classe *GestorPrincial*, que chama o método *getBluetoothSignatures()* da classe *BluetoothScans*, de forma a obter as amostras recolhidas, para se construir uma assinatura e depois enviar os dados para o servidor (caso tenha ligação com o mesmo) ou armazena-los na lista EAA para enviar mais tarde.

No final de todo este conjunto de procedimentos, o temporizador da classe *GestorPrincipal* aguarda um determinado período de tempo (configurável), até ordenar o lançamento da *thread*, para a realização de um novo *scan* de dispositivos Bluetooth.

7.3. Resultados

Construído e integrado o módulo Bluetooth na aplicação Epi, foi possível realizar um pequeno estudo da informação recolhida. Contudo, como a nova versão da aplicação apenas foi disponibilizada cerca de duas semanas antes da realização desta análise, ainda poucas pessoas tinham instalado a aplicação. Este estudo serve mais para justificar a correcta integração do módulo e que tipos análises se podem realizar tendo em conta os dados recolhidos.

A análise aqui apresentada teve em conta 15 dias de funcionamento da aplicação, deste o dia 29-09-2011 às 20 horas até ao dia 14-10-2011 às 19 horas e 59 minutos. Na Tabela 13 encontra-se um resumo de informações relativas ao estudo.

Duração	Data de Início	Data de Fim	Duração de cada *Scan*	Nº de Amostras	Nº Disp. Detectados
15 Dias	29-09-2011	14-10-2011	10 Seg.	1318	77

Tabela 13 - Epi: Informações do estudo.

Relativamente à mobilidade do cenário onde decorreu a recolha de informação, pode dizer-se que é de mobilidade nula ou baixa, dado que esta aplicação tem como *target* computadores portáteis e os mesmos não costumam estar presentes em cenários ou ambientes de elevada mobilidade.

No Gráfico 47 pode ser observado que a frequência de detecção de um dispositivo tem o mesmo comportamento que apresentava nos testes realizados anteriormente, apesar do cenário de recolha de amostras ser diferente.

A curva começa por crescer lentamente, o que indica que um elevado número de dispositivos foi detectado poucas vezes. A partir de cerca das 25 detecções a curva apresenta um crescimento quase exponencial, mas não tanto como no caso dos testes em ambiente de mobilidade elevada. Isso também já seria de se esperar uma vez que o *target* da aplicação Epi é para computadores portáteis e as pessoas não utilizam este dispositivo quando em movimento como é o caso dos telemóveis, por exemplo.

Gráfico 47 - Epi: Gráfico de Frequências.

O histograma representado no Gráfico 48 indica que é mais frequente um dispositivo ser detectado uma ou duas vezes. Os dispositivos que apresentam um elevado número de detecções são, muito provavelmente, os dispositivos que tinham a aplicação Epi instalada e a interface Bluetooth activa no seu computador, para puderem realizar os *scans*.

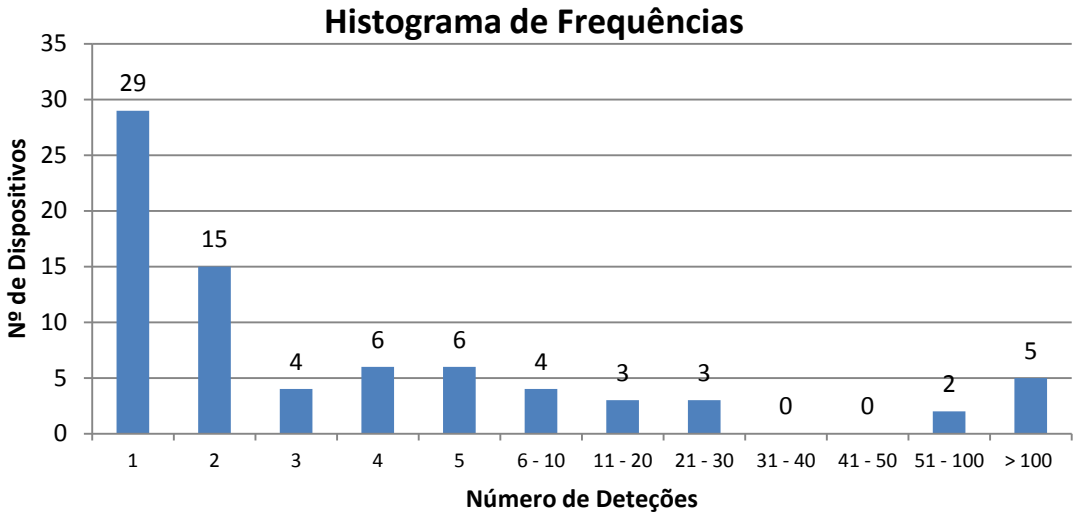


Gráfico 48 - Epi: Histograma de Frequências.

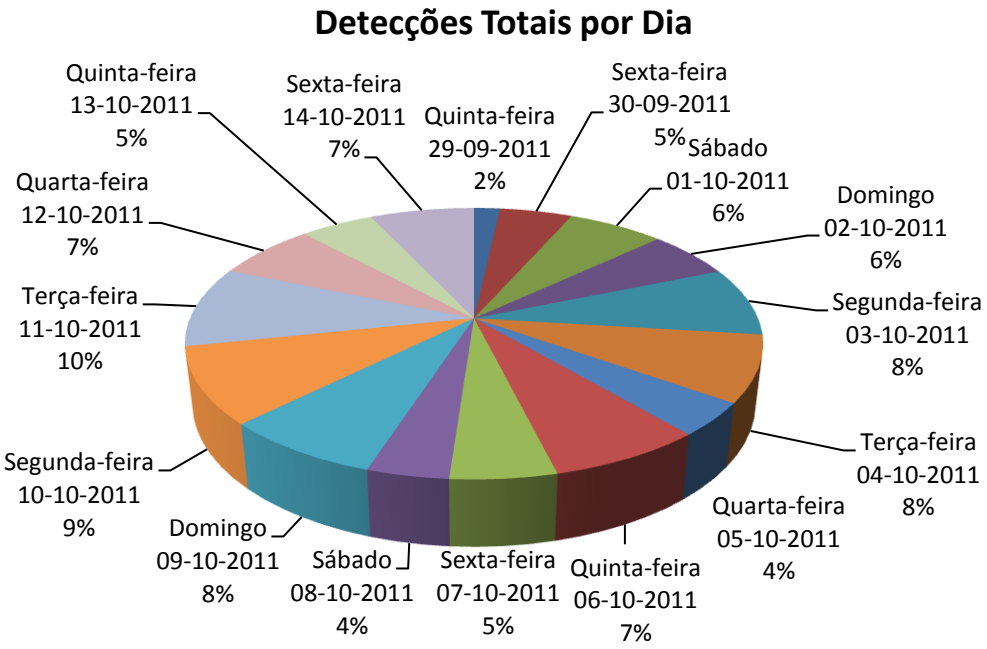


Gráfico 49 - Epi: Detecções Totais por Dia.

Analisando agora o número de detecções por dia (Gráfico 49), durante os quinze dias de recolha de informação, verifica-se que o número de detecções é constante. Aqui, os dias com mais detecções foram a terça-feira dia onze com 10% de detecções e a segunda-feira dia dez, com 9% de detecções. Os restantes dias apresentaram todos cerca de 8% de detecções.

Um outro estudo elaborado está presente no Gráfico 50 e diz respeito ao número de detecções acumuladas por hora. O número de detecções começa a cair no início da primeira hora do dia, com 65 detecções, até por volta das sete horas da manhã, não sendo aqui verificada nenhuma detecção até as oito horas da manhã. Durante a manha e até às duas primeiras horas da parte da tarde o crescimento é mais lento, não alcançando sequer as 65 detecções. Da parte da tarde, às 14horas temos o pico de detecção máximo do dia com 119 detecções, até às 19 horas, onde se desce para as 66 detecções. Até ao final do dia o número de detecções estabiliza, subindo para cerca de oitenta.

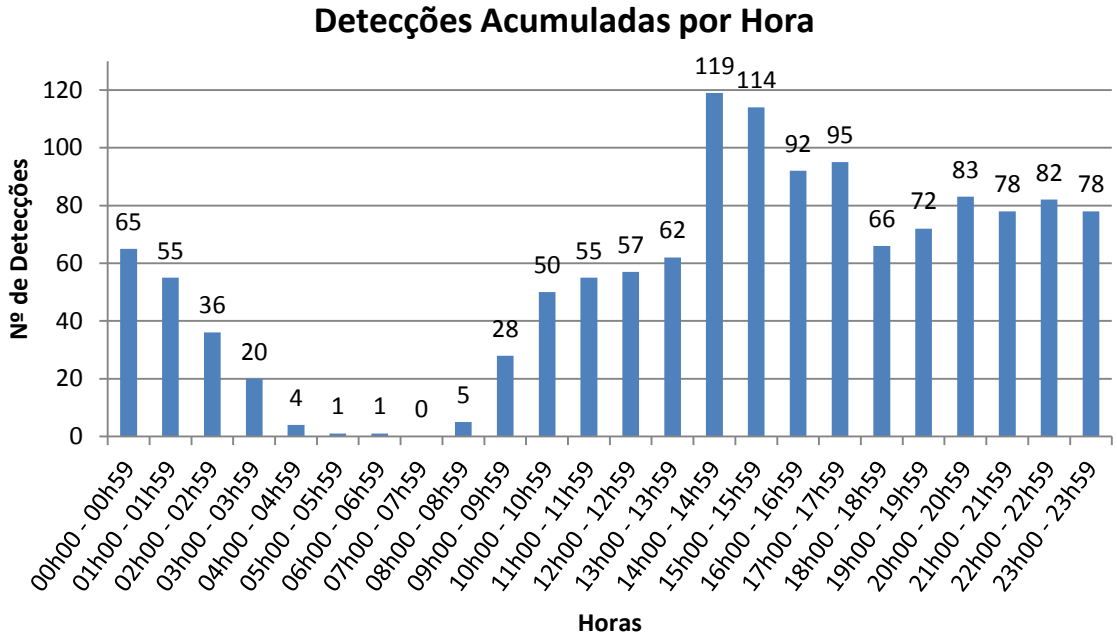


Gráfico 50 - Epi: Detecções Acumuladas por Hora.

Para as classes dos dispositivos detectados, também não se verificam grandes novidades. A classe mais detectada foi a dos computadores portáteis com 29 detecções. No entanto, se associarmos as classes *CellPhone* e *SmartPhone* obtêm-se um número total de detecções de 38 detecções, um valor mais elevado que as classes *DesktopComputer* e *LaptopComputer* juntas, com 37 detecções no total.

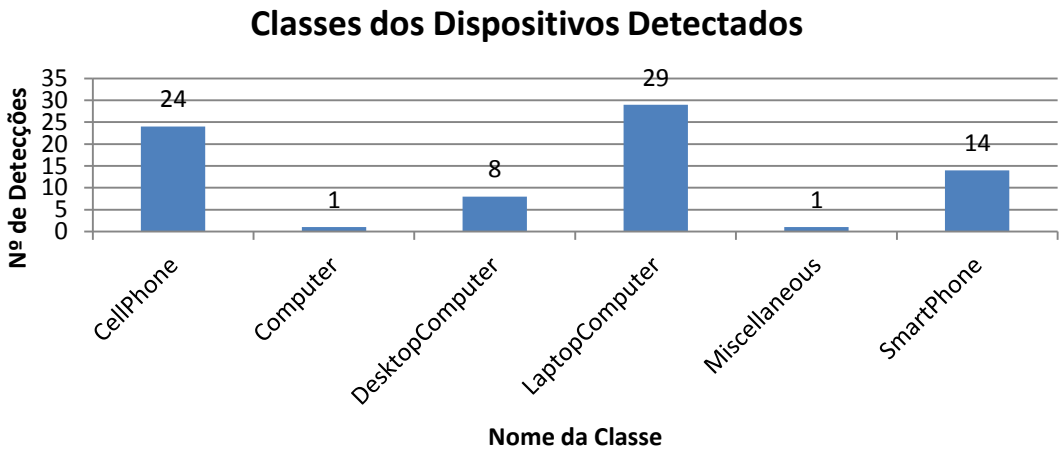


Gráfico 51 - Epi: Classes dos Dispositivos Detectados.

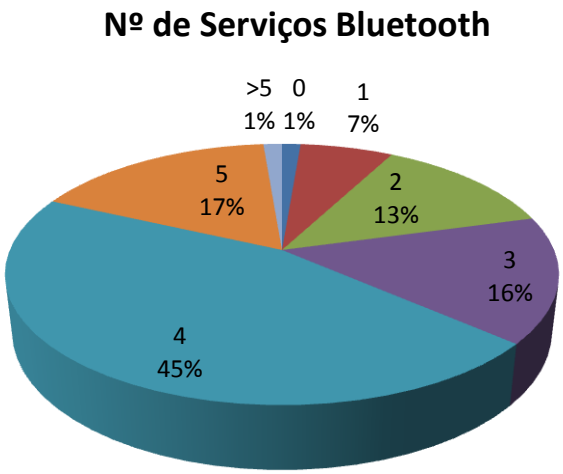


Gráfico 52 - Epi: Número de Serviços Bluetooth.

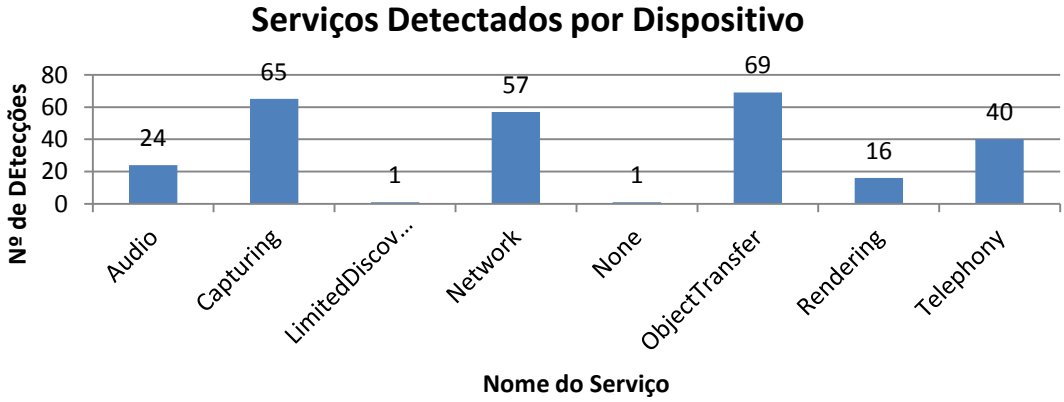


Gráfico 53 - Epi: Serviços Detectados por Dispositivo.

O número de serviços Bluetooth detectados pode ser visto no Gráfico 52. Mais uma vez se determina, que é mais comum encontrar dispositivos Bluetooth com quatro serviços disponíveis. Um dado curioso é que desta vez, é mais comum encontrar dispositivos com cinco serviços, aspecto que nos testes de campo não se verificava tanto.

Relativamente aos serviços disponibilizados (Gráfico 53), os serviços de transferência de objectos, *Capturing* e *Network* lideram. Os serviços de *Telephony* decresceram ligeiramente, aspecto já esperado dado o *target* para que esta aplicação foi desenvolvida. □

8. Conclusão

Apresenta-se de seguida a análise de todo o trabalho realizado com este projecto de dissertação. Ao longo de duas secções é descrita a sua elaboração e ainda o que poderá ser desenvolvido no futuro para complementar o projecto de dissertação.

8.1. Trabalho Realizado

No final da realização deste projecto de dissertação é feito um "balanço" do trabalho elaborado. Examinando os três objectivos que haviam sido propostos para esta dissertação pode-se afirmar que todos eles foram alcançados.

O primeiro objectivo era a criação de um módulo que recolhesse um conjunto de informações relacionadas com a tecnologia Bluetooth. Este módulo foi elaborado sem grandes complicações ou problemas e foi conseguida a sua integração na aplicação Epi.

Com este módulo foi recolhida alguma informação, que posteriormente foi analisada, e cujos resultados se apresentam na secção 7.3. Contudo, a informação recolhida não foi muita, podendo-se concluir que existem poucos utilizadores da aplicação Epi que possuem a sua interface Bluetooth activa, no seu computador. O facto de haver pouca informação pode ser justificado pela divulgação recente da aplicação Epi, proporcionando pouco tempo aos utilizadores para a instalarem.

Outro objectivo era desenvolver um estudo que apurasse como as pessoas se distribuem geograficamente e temporalmente num determinado local. Nesse sentido foi

construída uma aplicação de *scan* de dispositivos. Na implementação da aplicação de *scan,* foram realizados vários testes de onde se recolheram várias informações.

Da análise das mesmas foram identificados vários "padrões de mobilidade" das pessoas, que reflectem de uma forma exacta o que se pode fisicamente observar no local. Este estudo poderia ser ainda mais extenso, dado que vários outros "parâmetros de mobilidade" poderiam ser analisados e apresentados. Desta forma se conclui que com a tecnologia Bluetooth, dispõe-se de uma ferramenta com elevado potencial para a realização de estudos deste tipo.

Por fim, o último objectivo, e talvez o mais relevante de todos, era o cálculo da probabilidade de detecção de um dispositivo Bluetooth num ambiente que oferecesse elevada mobilidade. Para tal usaram-se os dados recolhidos pela aplicação e *scan* e foram desenvolvidos dois processos de análise, resultando dois estudos: a probabilidade de detecção por local e a probabilidade de detecção global, ou seja, tendo em conta um conjunto de locais de *scan*.

Se tivermos em conta a abordagem do estudo por local, constata-se que a tecnologia Bluetooth é uma boa solução para quando se pretende realizar a detecção de dispositivos de uma forma pontual, uma vez que as probabilidades de detecção foram elevadas: por exemplo, contar as pessoas que entram numa loja ou saber quanto tempo passa em média cada pessoa, num determinado local.

Caso se pretenda usar agora a tecnologia, por exemplo, para determinar um percurso realizado por uma pessoa ou detectar dispositivos a uma velocidade média ou alta, ela não é uma boa opção, se considerarmos as probabilidades de detecção obtidas no caso do estudo global. Se fosse aumentado o número de locais de detecção talvez se obtivessem melhores resultados.

Nesta matéria não se tem conhecimento de outros estudos do mesmo tipo, o que seria interessante para se poderem comparar os resultados.

8.2. Trabalho Futuro

Como em qualquer outro estudo desta natureza existe sempre trabalho futuro a ser realizado, e este não é excepção. Neste caso, para este estudo seria necessário realizar pequenos ajustes na aplicação de *scan*. Nomeadamente a revisão do processo de armazenamento de valores relativos aos *scans*, em caso de falha ou mau funcionamento da aplicação.

Também seria interessante a realização de mais alguns testes de campo, pelo menos mais dois.

O primeiro seria realizar um outro conjunto de *scans*, substituindo os *dongles* onde está implementada a versão 2.0 de Bluetooth, por *dongles* com a versão 3.0. Depois determinar se existiriam diferenças no número de dispositivos detectados e consequentemente nas probabilidades de detecção. Este teste ainda esteve previsto mas depois, devido à falta de tempo, não se chegou a realizar.

O segundo teste seria a implementação da aplicação de *scan* num cenário que mostrasse, também, elevada mobilidade, mas desta vez que não se realizasse num ambiente exterior, como é um caso de uma rua. Para esta situação, o cenário ideal seria um *shopping* ou centro comercial.

Um outro aspecto que se poderia ter em conta seria a montagem e construção de um dispositivo próprio para a realização dos *scans*.

Em relação à análise de resultados, era também interessante desenvolver ou encontrar um método diferente, do processo de análise global, para calcular as probabilidades de detecção e no final serem comparados os resultados. Talvez a partir do estudo das probabilidades da detecção por local se conseguisse encontrar um método diferente e obter um valor para as probabilidades de detecção.

No que diz respeito à aplicação Epi, é um projecto que se encontra em desenvolvimento, como tal está a ser desenvolvido mais trabalho. De salientar que a inclusão de novas funcionalidades para os utilizadores da aplicação deve ser um assunto a reflectir, uma vez que o desenvolvimento de novas funcionalidades cativaria mais utilizadores para o uso da aplicação. Deste modo mais dados seriam gerados e assim poderiam ser realizadas mais análises.

Considerando o servidor, deve ainda ser implementada uma interface que disponibilize informação, em tempo real dos dados recolhidos ao administrador do servidor. □

Referências Bibliográficas

[1] *Wireless Urban Sensing Systems .* **Mani Srivastava, Mark Hansen, Jeff Burk, et al.** Los Angels : Center for Embebbed Networked Sensing Systems, UCLA and ICSI Center for Internet Research , Technical Report April 2006.

[2] "*Urban Sensing: Opportunistic or Participatory?*", **Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, Andrew T. Campbell**, Presented at *Workshop on Sensing on Everyday Mobile Phones in Support of Participatory Research*, Sydney, November 6, 2007.

[3] *Real-Time Ubiquitous Urban Sensing and Modelling.* **Musolesi, Mirco.** Cambrige, UK : Computer Laboratory, University of Cambrige.

[4] *Sensing in Rich Bluetooth Enviroments.* **Marion Hermersdorf, Heli Nyholm, Jukka Perkiö, et al.** Finland : Nokia Research Center and Helsinki Institute for Information Technology, in Proceedings of WSW'06 at SenSys'06. SenSys'06

[5] *A CommomSence Approach to Real-world Global Sensing.* **Srdjan Krco, Mattias Johansson, Vlasios Tsiatsis.** : Ericsson, in 7th International Conference on AD-HOC Networks & Wireless, September 10 - 13, 2008

[6] *CarTel: a distributed mobile sensor computing system*. **Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden**. 2006. In Proceedings of the 4th international conference on Embedded Networked Sensor Systems (SenSys'06). ACM, New York, NY, USA, 125-138. DOI=10.1145/1182807. 1182821

[7] *Challenges to building Bluetooth-based sensing solutions*. **Chieh-Yih Wan and Sai Prasad.** 2009. In Proceedings of the Fourth International Conference on Body Area Networks (BodyNets '09). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Article 6, 9 pages.DOI=10.4108/ICST.BODYNETS2009.6022 http://dx.doi.org/10.4108/ICST.BODYNETS2009.6022

[8] "*People-Centric Urban Sensing: Security Challenges for the New Paradigm*", **Peter Johnson, Apu Kapadia, David Kotz, Nikos Triandopoulos**, Dartmouth Technical Report TR2007-586, February 2007.

[9] Metro Sence. 2011. Online. Available: http://metrosense.cs.dartmouth.edu/

[10] Urban Sensing. 2011. Online. Available: http://urban.cens.ucla.edu/

[11] Cartel. 2011. Online. Available: http://cartel.csail.mit.edu/doku.php

[12] Around Knowledge. 2011. Online. Available: http://www.aroundknowledge.com/

[13] Bluetooth. 2011. Online. Available: http://www.bluetooth.com/

[14] Bluetooth SIG, The Bluetooth Specification v2.0 EDR. 2011. Online. Available: http://www.bluetooth.org/Technical/Specifications/adopted.htm

[15] Wikipedia, Bluetooth. 2011. Online. Available: http://en.wikipedia.org/wiki/Bluetooth

[16] Albert S. Huang, Larry Rudolph, **Bluetooth Essentials for Programmers**, Cambridge University Press, 2007.

[17] David Kammer, Gordon McNutt, Brian Senese, **Bluetooth - Application Developer's Guide**, Syngress, 2002.

[18] Houda Labiod, Hossam Afifi, Costantino De Santis, **Wi-Fi, Bluetooth, Zigbee and WiMAX**, Springer, 2007.

[19] Bluetooth Wireless Technology Basics, Hewlett-Packard (HP), 05/2004.

[20] Jochen H. Schiller, **Mobile Communications**, Pearson Education, 2003.

[21] Bluetooth, BaseBand. 2011. Online. Available: https://www.bluetooth.org/Technical/AssignedNumbers/baseband.htm

[22] 32Feet. 2011. Online. Available: http://32feet.codeplex.com/

[23] C. A. L. Sousa, **"Difusão Epidémica de Mensagens em Hotspots WiFi"**, Ph.D. Dissertation, Universidade do Minho, Guimarães, Portugal, Nov. 2010

[24] Hélder Tavares de Lemos, **"Descoberta Passiva de Estações em Redes 802.11"**, Ph.D. Dissertation, Universidade do Minho, Guimarães, Portugal, Out. 2011

[25] The Radio Spectrum. 2011. Online. Available: http://en.wikipedia.org/wiki/File:United_States_Frequency_Allocations_Chart_2003_-_The_Radio_Spectrum.jpg

[26] Radar de Negócios. 2011. Online. Available: http://www.faroldeideias.com/arquivo_farol/index.php?programa=Radar&id=1029

Anexos

A1. Tabelas Bluetooth

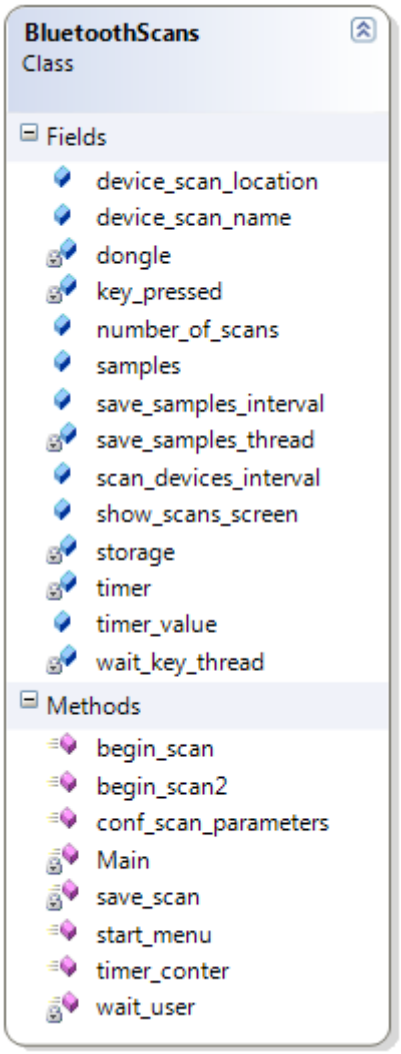
Países	Banda de Frequências (MHz)	Canais RF	
Europa* e Estados Unidos da América	2400 – 2483.5	f = 2402+k	k = 0,…,78
França	2446.5 – 2483.5	f = 2454+k	k = 0,…,22
Espanha	2445 – 2475	f = 2454+k	k = 0,…,22

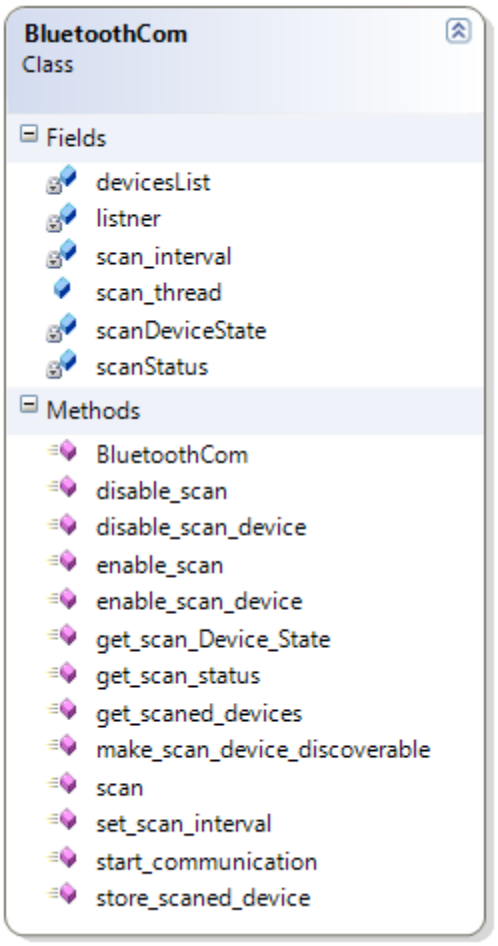
Tabela 14 - Banda de Frequências e Canais RF. Fonte [18].

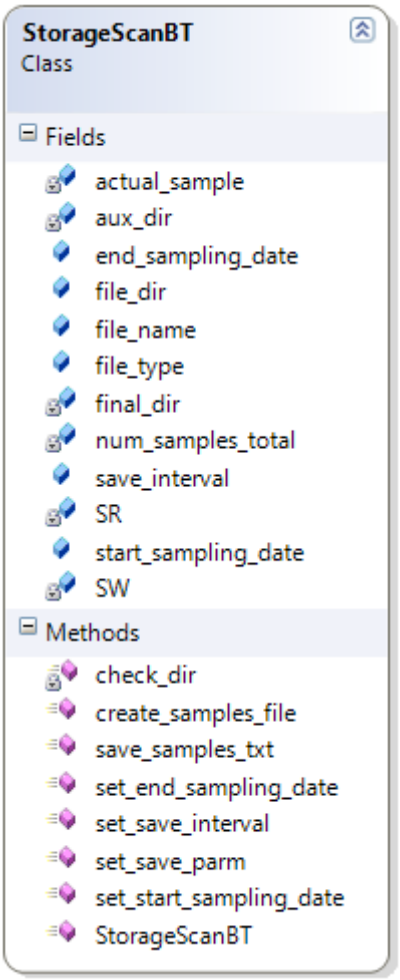
Perfil	Camadas Inferiores	L2CAP	SDP	RFCOM	PPP	OBEX	TCS
Service Discovery Application	X	X	X				
Cordless Telephony	X	X	X				X
Intercom	X	X	X				X
Serial Port	X	X	X	X			
Headset	X	X	X	X			
Dial-up Networking	X	X	X	X			
FAX	X	X	X	X			
LAN Access	X	X	X	X	X		
Generic Object Exchange	X	X	X			X	
Object Push	X	X	X			X	
File Transfer	X	X	X			X	
Synchronization	X	X	X			X	

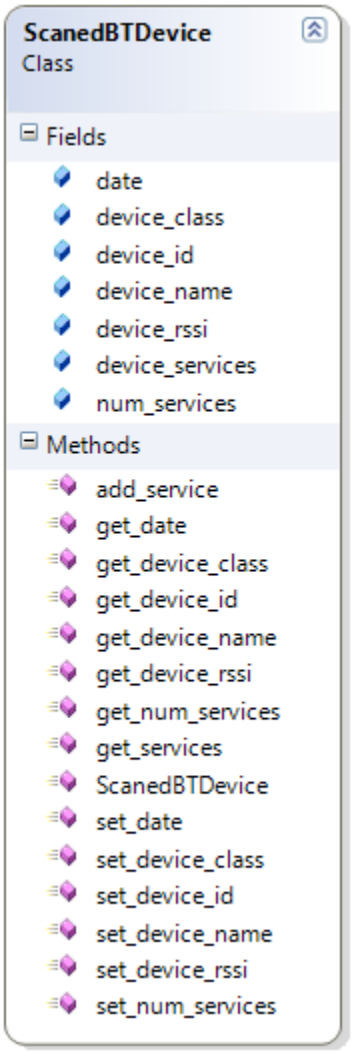
Tabela 15 - Protocolos e Camadas da *Stack* Bluetooth Usadas pelos Perfis. Fonte [17].

A2. Classes da Aplicação de *Scan*

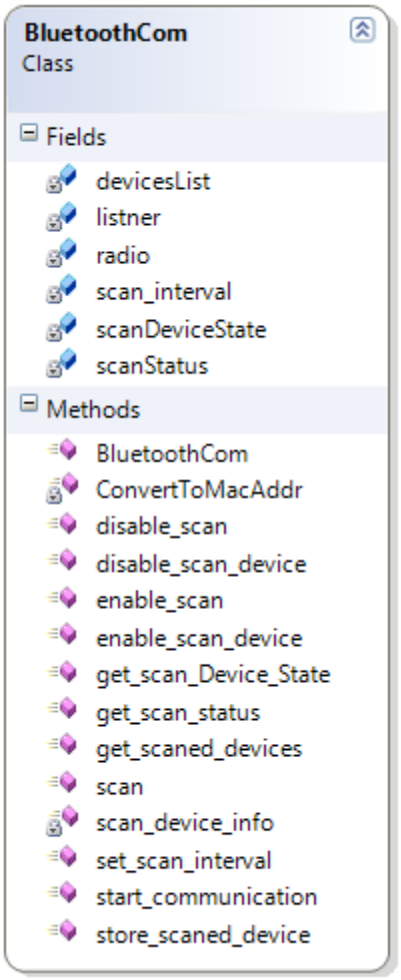


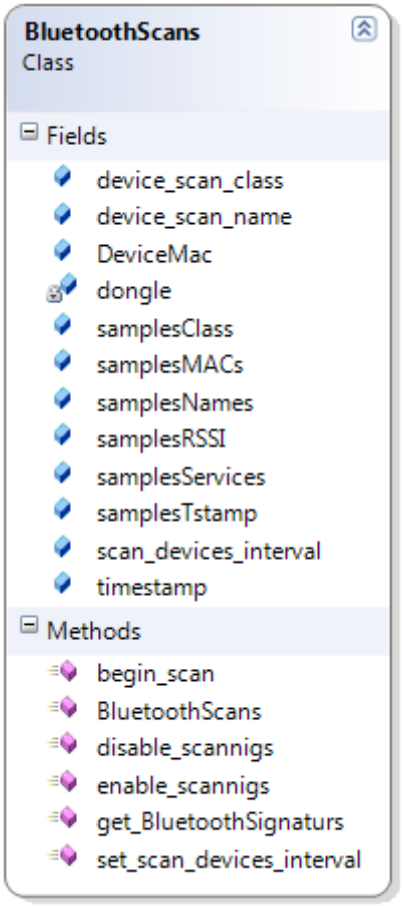



StorageScanBT
Class
Fields
actual_sample
aux_dir
end_sampling_date
file_dir
file_name
file_type
final_dir
num_samples_total
save_interval
SR
start_sampling_date
SW
Methods
check_dir
create_samples_file
save_samples_txt
set_end_sampling_date
set_save_interval
set_save_parm
set_start_sampling_date
StorageScanBT


ScanedBTDevice
Class
Fields
date
device_class
device_id
device_name
device_rssi
device_services
num_services
Methods
add_service
get_date
get_device_class
get_device_id
get_device_name
get_device_rssi
get_num_services
get_services
ScanedBTDevice
set_date
set_device_class
set_device_id
set_device_name
set_device_rssi
set_num_services

A3. Classes do Módulo Bluetooth





A4. Listagem do Servidor

```
1  Cliente -> Servidor
2  POST /epiServer2/epiS HTTP/1.1
3  Content-Type: application/x-www-form-urlencoded
4  Host: epi.dsi.uminho.pt:8080
5  Content-Length: 338
6  Expect: 100-continue
7  Servidor -> Cliente
8  HTTP/1.1 100 Continue
9  Cliente -> Servidor
10 f=f3&
11 ats=2011%3a10%3a13%3a18%3a42%3a22%3b%2b01%3a00&
12 sgt=2&clm=00%3a25%3aD3%3aB0%3aCA%3a2E&
13 has=nomsgnomsgnomsgnomsgnomsgnomsgno&
14 mts=2011%3a10%3a13%3a17%3a44%3a20%3b%2b01%3a00&
15 sts=2011%3a10%3a13%3a17%3a44%3a20&
16 mac=00%3a25%3aD3%3aB0%3aCA%3a2E&
17 dev=DINH-PC&
18 tsd=2011%3a10%3a13%3a17%3a44%3a20&
19 cls=LaptopComputer&
20 srv=Network%2c+Rendering&
21 rsi=Null
22 Servidor -> Cliente
23 HTTP/1.1 200 OK
24 Server: Apache-Coyote/1.1
25 Content-Type: text/html
26 Content-Length: 10
27 Date: Thu, 13 Oct 2011 17:42:25 GM
```