

Mestrado em Engenharia Informática

Dissertação

Detecção de actividade humana e representação em redes sociais

Duarte Raposo

draposo@student.dei.uc.pt

Orientador:

Jorge Sá Silva

Data: 31 de Agosto de 2012



Resumo

Actualmente existem inúmeros sensores presentes em dispositivos móveis por todo o mundo. Embora nos acompanhem no nosso quotidiano, não são aproveitados da melhor forma, isto é, não são utilizados com o intuito de recolher informação sobre o contexto onde nos inserimos, bem como as actividades que efectuamos.

O aparecimento das redes de sensores sem fios veio disponibilizar uma tecnologia de baixo custo e, para além disso, uma capacidade de adquirir diferentes dados do meio envolvente. O surgimento das redes sociais e dos mundos virtuais permitiu a partilha de experiências e de actividades pessoais, dispondo de milhões de utilizadores [1]. Torna-se assim possível e potencialmente interessante construir uma ligação entre o mundo real e o mundo virtual.

O presente Estágio/Dissertação propôs-se assim desenvolver uma plataforma intitulada *SocialSense*, que permita uma ligação entre o mundo real e o mundo virtual, através da conexão entre redes sociais com diferentes tecnologias, nomeadamente redes de sensores sem fios (*WSN*), dispositivos móveis e sistemas ubíquos, a partir de um *gateway*. Com base nesta plataforma desenvolvida, e ainda no âmbito deste estágio, foram construídas duas aplicações para dois projectos do Laboratório de Comunicações e Telemática: o projecto Avós e Netos e o projecto Youth5G.

O projecto Avós e Netos suporta a adopção de netos virtuais através de redes sociais como o *Facebook*, permitindo a monitorização dos netos por parte dos avós.

O projecto *Youth5G* destina-se a jovens que desejam encontrar locais nocturnos (cafés, eventos culturais, bares, etc) com uma maior afluência e, para tal, são utilizados dados colaborativos.

Além da construção da plataforma *SocialSense* e das duas aplicações, construí ainda uma aplicação *web*, uma aplicação *android*, participei na escrita de alguns artigos científicos, o meu trabalho serviu para a preparação de duas candidaturas Sapo *LABS*, e ajudei os meus colegas André Pinto e David Nunes na integração dos mecanismos de localização e de intermitência de actividade humana na plataforma.

Palavras-Chave

“Mundo Virtual”, “Mundo Real”, “Redes de sensores sem fios”, “Redes Sociais”, “Sistemas Ubíquos”

Agradecimentos

O verdadeiro valor das coisas, na maior parte das vezes, atribui-se mediante o esforço exigido e os obstáculos que se ultrapassaram para as podermos alcançar. Sem dúvida, é este o caso do presente projecto, foi exactamente todo o empenho e o longo caminho percorrido até então que lhe conferem o enorme valor que tem para mim. Há que frisar, que todo este esforço não era possível sem a colaboração de várias pessoas que estiveram ao meu lado, de uma forma ou de outra, e que foram fundamentais. Por isso, quero expressar aqui a minha gratidão.

Em primeiro lugar, quero agradecer ao Prof. Dr. Jorge Sá Silva, pelo incentivo, rigor e disponibilidade sempre demonstrado durante o desenvolvimento deste projecto.

Quero também agradecer aos meus colegas David Nunes e André Pinto, que desde o início foram meus parceiros na construção da plataforma *SocialSense* e que me ajudaram a definir o melhor rumo para o projecto.

A título pessoal, gostaria de agradecer com um especial e enorme carinho à minha Mãe, ao meu Pai e à minha grande Avó, que desde o início fizeram um esforço louvável, para que conseguisse chegar ao final deste percurso.

À minha querida namorada Andreia Barbas pela força, compreensão, carinho, dedicação e imensa paciência prestado todos os dias. Obrigado, por teres estado sempre ao meu lado, foste incansável.

E *last but not least* aos meus colegas e amigos Jorge Proença e Sérgio Reis que durante estes dois anos me acompanharam nesta trajectória, pelas longas noites passadas no DEI e por todo o companheirismo que me prestaram.

A todos(as), o meu muito obrigado!

Índice

Resumo.....	II
Palavras-Chave	II
Agradecimentos.....	IV
Índice	VI
Índice de Anexos	X
Índice de Figuras	XII
Índice de Tabelas	XVI
Índice de Gráficos	XVI
Acrónimos.....	XVIII
Capítulo 1 Introdução	2
1.1. Motivação	2
1.2. Objectivos	3
1.3. Estrutura do Documento.....	4
Capítulo 2 Tecnologias e Redes Sociais.....	6
2.1. Tecnologias	7
2.2. Redes Sociais.....	13
Capítulo 3 Estado de Arte	16
3.1. Projectos Desenvolvidos.....	16
3.1.1. Aplicações Semelhantes ao Projecto Avós e Netos(as) Virtuais	16
3.1.2. Aplicações Concorrentes ao Projecto <i>Youth5G</i>	23
3.2. Mecanismos de Localização	25
3.2.1. Localização <i>Indoor</i>	26
3.2.2. Localização <i>Outdoor</i>	28
3.3. Mecanismos de Intermittência	31
3.3.1. Redes <i>DTN</i>	31
3.3.2. Cadeias de <i>Markov</i>	32
3.4. Privacidade	35
3.4.1. Identificação Directa do Utilizador	35
3.4.2. Identificação Indirecta do Utilizador	36
3.4.3. Redes Sociais.....	37
3.4.4. Privacidade Associada ao Ambiente.....	37

Capítulo 4 Objectivos da Investigação e Método de Abordagem	40
4.1. Características e Requisitos da Plataforma <i>SocialSense</i>	40
4.2. Planeamento e Método de Trabalho	41
Capítulo 5	46
Estudos Preliminares	46
5.1. Modelo Inicial	46
5.1.1. Arquitectura Inicial	47
5.2. Trabalho Desenvolvido	48
5.2.1. Arquitectura Desenvolvida	48
5.2.2. <i>Facebook Publisher</i>	49
5.2.3. Aplicação Facebook	50
Capítulo 6 Projecto de Avós e Netos(as)	52
6.1. Descrição do Projecto	52
6.2. Especificação da Aplicação	53
6.2.1. Actores e Público-Alvo	54
6.2.2. Objectivos	54
6.2.3. Contexto	55
6.2.4. Capacidades	56
6.2.5. Requisitos Funcionais	56
6.2.6. Requisitos Não Funcionais	58
6.2.7. Requisitos de Qualidade	58
6.2.8. Restrições Técnicas	59
6.3. Arquitectura do Sistema	60
6.3.1. Descrição Geral	60
6.3.2. Plataforma	61
6.3.3. Base de Dados	66
6.4. Aplicação e Demonstração	68
6.4.1. Actividade	69
6.4.2. Mensagens	69
6.4.3. Videochamadas	69
6.4.4. Alertas	70
6.5. Principais Desafios	71
6.6. Conclusão	71
Capítulo 7	72

Projecto Youth5G	72
7.1. Descrição do Projecto	72
7.2. Especificação da Aplicação.....	74
7.2.1. Actores e Público-Alvo.....	74
7.2.2. Objectivos	75
7.2.3. Contexto.....	75
7.2.4. Capacidades	76
7.2.5. Requisitos Funcionais	76
7.2.6. Requisitos Não Funcionais	77
7.2.7. Requisitos de Qualidade	77
7.2.8. Restrições Técnicas.....	78
7.3. Arquitectura do Sistema	79
7.3.1. Descrição Geral.....	79
7.3.1. Plataforma.....	80
7.3.2. Aplicação.....	84
7.3.3. Base de Dados.....	88
7.4. Aplicação e Demonstração	90
7.4.1. Login e Registo.....	90
7.4.2. Mapa e Menu de Opções.....	91
7.4.3. Características do POI e Visualização de Eventos	91
7.4.4. Sistema de Incentivo e Feedback e Previsões.....	92
7.4.5. Inquérito aos Utilizadores	93
7.4.6. Modelo de Privacidade.....	93
7.5. Principais Desafios.....	94
7.6. Conclusão	94
Capítulo 8 Testes.....	96
8.1. Objectivos dos Testes.....	96
8.2. Projecto Avós e Netos	98
8.2.1. Testes de Requisitos	98
8.2.2. Testes de Performance.....	98
8.2.3. Testes de Funcionais	101
8.3. Projecto Youth5G.....	103
8.3.1. Testes de Requisitos	103
8.3.2. Testes de Performance.....	103

8.3.3. Testes Funcionais.....	106
Capítulo 9 Outros Trabalhos	110
Referências.....	112

Índice de Anexos

Anexo A “A Web Service-Based Framework Model for People-Centric Sensing Applications Applied to Social Networking”	116
Anexo B “Percontrol: Use of Neural Network and WSN to Control the User Profile in Pervasive Environments”	134
Anexo C “Quietly: A model privacy for pervasive and ubiquitous environments”	148
Anexo D “Proposta Sapo Labs 1”	150
Anexo E “Proposta Sapo Labs 2”	164
Anexo F “Mensagens de Comunicação do sistema Avós e Netos”	174
F.1 Descrição Geral	178
F.2 Especificação das mensagens	180
Anexo G “Mensagens de Comunicação do sistema Youth5G”	192
G.1 Descrição Geral	196
G.2 Especificação das Mensagens	198
Anexo H “Requisitos Funcionais do Sistema Avós e Netos”	214
H.1 Descrição	218
H.2 Requisitos Funcionais	219
Anexo I “Requisitos Funcionais do Sistema Youth5G”	242
I.1 Descrição	246
I.2 Requisitos Funcionais	247

Índice de Figuras

Figura 1 - Estrutura da Dissertação	5
Figura 2 - <i>RFID</i>	7
Figura 3 - Diferentes nós de redes de sensores sem fios.	8
Figura 4 - Smartphones, diferentes SO.....	9
Figura 5 - Smartshirt UA-E39.....	11
Figura 6 - WiiRemote, PlaystationMove e Kinect.....	12
Figura 7 – Arquitectura SoundSence	17
Figura 8 - Whozthat.....	20
Figura 9 - Serendipity.	22
Figura 10 - <i>Wlan Fingerprinting</i>	27
Figura 11 - Rede DTN.....	32
Figura 12 - Matrizes de Previsão	33
Figura 13 – Modelo Genérico de Privacidade	38
Figura 14 - Método de Trabalho SCRUM.	42
Figura 15 - Diagrama de Gant	44
Figura 16 - Gateway Híbrido Comunicação	46
Figura 17 - Tecnologias e Proxys interligados à plataforma <i>SocialSense</i>	47
Figura 18 - Arquitectura Inicial <i>SocialSense</i>	48
Figura 19 - Módulos desenvolvidos	49
Figura 20 - Publicação Facebook	50
Figura 21 - Aplicação <i>Facebook</i>	51
Figura 22 - Visão Geral do Sistema Avós e Netos.....	53
Figura 23 - Diagrama de Casos de Uso SocialSense Avós e Netos.....	57
Figura 24 - Arquitectura <i>SocialSense</i> da plataforma Avós e Netos.....	60
Figura 25 - Diagrama de Classes do sistema Avós e Netos.....	64
Figura 26 - Diagrama Entidade-Relacionamento do sistema Avós e Netos.....	67
Figura 27 - Acesso à aplicação Avós e Netos	68
Figura 28 - Menu Principal aplicação Avós e Netos.....	68
Figura 29 - Página de Actividade da Aplicação Avós e Netos	69

Figura 30 - Página de Videochamada da Aplicação Avós e Netos.....	70
Figura 31 - Página de Alertas da Aplicação Avós e Netos.....	70
Figura 32 - Visão Geral do Sistema Youth5G.....	73
Figura 33 - Diagrama de Casos de Uso da Aplicação Youth5G.....	76
Figura 34 - Arquitectura SocialSense da plataforma Youth5G	79
Figura 35 - Fórmula da previsão da afluência Youth5G	80
Figura 36 - Diagrama de classes da plataforma SocialSense Youth5G	82
Figura 37 - Diagrama de Classes da aplicação <i>Android</i> Youth5G	86
Figura 38 - Base de dados do sistema Youth5G	89
Figura 39 - Menu Inicial da Aplicação <i>Android</i> Youth5G.....	90
Figura 40 - Menu de Registo da Aplicação <i>Android</i> Youth5G	90
Figura 41 - Menu de Login da Aplicação <i>Android</i> Youth5G	90
Figura 42 - Mapa e Menu de Opções da Aplicação <i>Android</i> Youth5G.....	91
Figura 43 - Vista de detalhes de um POI da aplicação <i>Android</i> Youth5G.....	92
Figura 44 - Vista de detalhes de um evento da aplicação <i>Android</i> Youth5G	92
Figura 45 - Menu Principal da aplicação <i>Android</i> Youth5G	92
Figura 46 - Vista do Ranking da aplicação <i>Android</i> Youth5G.....	92
Figura 47 - Vista de envio de Feedback da aplicação <i>Android</i> Youth5G	92
Figura 48 - Vista de Previsão da aplicação <i>Android</i> Youth5G.....	92
Figura 49 - Vista inquéritos comentário da aplicação <i>Android</i> Youth5G	93
Figura 50 - Vista de inquéritos resposta rápida da aplicação <i>Android</i> Youth5G	93
Figura 51 - Disponibilização dos serviços aplicação <i>Android</i> Youth5G	93
Figura 52 - Deteção do ambiente aplicação <i>Android</i> Youth5G	93
Figura 53 - Cenário de Testes Avós e Netos	99
Figura 54 - Cenário de Testes Youth5G	104
Figura 55 - Autenticação Facebook OAuth.....	179
Figura 56 - Mensagens Padrão Sistemas Avós e Netos.....	179
Figura 57 - DS Set Activity	180
Figura 58 - DS Get Child Data.....	181
Figura 59 - DS Get Avatar Data.....	182
Figura 60 - DS Set Fall Detection	183

Figura 61 - DS Get Alerts.....	184
Figura 62 - DS Get Associated Chils	185
Figura 63 - DS Get Parent Skype ID.....	186
Figura 64 - DS Get Child Photo.....	187
Figura 65 - DS Get Location.....	188
Figura 66 - DS Get Child Skype ID.....	189
Figura 67 - DS Save Location	190
Figura 68 - DS Get Access Login.....	198
Figura 69 - DS Get Detail Poi	199
Figura 70 - DS Get Poi	200
Figura 71 - DS Get Feedback Status.....	201
Figura 72 - DS Get List Event.....	202
Figura 73 - DS Get List Poi.....	203
Figura 74 - DS Get Prediction	204
Figura 75 - DS Get Prevasive Services	205
Figura 76 - DS Get Ranking.....	206
Figura 77 - DS Get Survey Status.....	207
Figura 78 - DS New Password.....	208
Figura 79 - DS New User	209
Figura 80 - DS Set Data Sensor	210
Figura 81 - DS Set Feedback.....	211
Figura 82 - DS Set Survey.....	212
Figura 83 - DCU Efectuar Login.....	219
Figura 84 - DCU Visualizar Menu Inicial.....	221
Figura 85 - DCU Vista Mensagens.....	223
Figura 86 - DCU Vista Alarmes.....	226
Figura 87 - DCU Efectuar Videoconferência.....	228
Figura 88 - DCU Vista Actividade	231
Figura 89- DCU Vista Pedir Videochamada.....	234
Figura 90 - DCU Aceder a Mensagens	236
Figura 91- DCU Vista Iniciar Videochamada.....	238

Figura 92 - DCU Efectuar Autenticação.....	247
Figura 93 - DCU Efectuar Login.....	247
Figura 94 - DCU Efectuar Registo.....	250
Figura 95 - DCU Seleccionar POI	252
Figura 96 - DCU Atribuir Feedback	253
Figura 97 - DCU Visualizar Evento.....	255
Figura 98 - DCU Visualizar Características.....	256
Figura 99 - DCU Visualizar Menu Opções.....	256
Figura 100 - DCU Alterar Password.....	256
Figura 101 - DCU Visualizar Ranking	256
Figura 102 - DCU Realizar Inquérito	256
Figura 103 - DCU Navegar Mapa	256

Índice de Tabelas

Tabela 1 - Conjunto de Dados.....	36
Tabela 2- Testes de Requisitos Avós e Netos.....	98
Tabela 3 - Testes Funcionais Avós e Netos.....	102
Tabela 4 - Testes de Requisitos Youth5G.....	103
Tabela 5 - Testes Funcionais Youth5G.....	108
Tabela 6 - Lista de <i>Webservices</i> Sistema Avós e Netos.....	178
Tabela 7 - Lista de <i>Webservices</i> do sistema Youth5G.....	196

Índice de Gráficos

Gráfico 1 - Volume de negócio dos markets de smartphones.....	10
Gráfico 2 - Tempo de Resposta dos <i>Webservices</i> do Sistema Avós e Netos.....	99
Gráfico 3 - Volume de Dados dos <i>Webservices</i> do Sistema Avós e Netos.....	100
Gráfico 4 - Número de Pedidos Respondidos dos <i>Webservices</i> Avós e Netos.....	100
Gráfico 5 - Volume de Dados dos <i>Webservices</i> do Sistema Youth5G.....	105
Gráfico 6 - Tempo de Resposta dos <i>Webservices</i> do Sistema Youth5G.....	105
Gráfico 7 - Número de Pedidos Respondidos dos <i>Webservices</i> Youth5G.....	106

Acrónimos

A-GPS: Assisted Gps

AID: Anonymous Identifier

AODV: Ad hoc On-Demand Distance Vector

API: Application Programming Interface

APs: Access Points

ARM: Advanced Risc Machine

BSA: Basic Service Area

BSC: Base Station Controller

BSS: Basic Service Set

BTID: Bluetooth Identifier

CPU: Central Processing Unit

CSS: Cascading Style Sheets

DAO: Database Access Objects

DGPS: Differential Gps

DTN: Delay Tolerant Network

E-OTD: Enhanced Observed Time Difference

GPS: Global Positioning System

GSM: Global System for Mobile Communications

GTD: Geometric Time Difference

HTTP: HyperText Transfer Protocol

HTTPS: HyperText Transfer Protocol Secure

ID: Identifier

IP: Internet Protocol

IS: Identity Server

J2ME: Java 2 Platform Micro Edition

JSON: JavaScript Object Notation

LCT: Laboratório de Comunicações e Telemática

LMU: Location Measurement Unit

MAC: Media Access Control

MANET: Mobile Ad Hoc Network

MIT: Massachusetts Institute of Technology

OLSR: Optimized Link State Routing Protocol

OTD: Observed Time Difference

PC: Personal Computer

PHP: Hypertext Preprocessor

POI: Point of Interest

PPS: Precision Position System

RFID: Radio Frequency Identification

RSSI: Received Signal Strength Indicator

RTD: Real Time Difference

SDK: Software Development Kit

SMLC: Serving Mobile Location Center

SPS: Standard Position Service

TCP: Transmission Control Protocol

TDMA: Time Division Multiple Access

TH: Threshold

UDP: User Datagram Protocol

U-TDOA: Uplink Time Difference Arrival

WIFI: Wireless Fidelity

WLAN: Wireless Local Area Network

WSN: Wireless Sensor Network

XML: Extensible Markup Language

Capítulo 1

Introdução

Neste capítulo é apresentada a motivação que originou este trabalho, enquadrada na temática de interligação entre mundos virtuais e mundos reais e são descritos os desafios propostos. Seguidamente, são identificados os principais objectivos do presente estágio/dissertação, terminando com a estrutura desta dissertação.

1.1. Motivação

No mundo de hoje com as tecnologias de informação e as redes sociais é possível compartilhar informações, conhecimentos, interesses, experiências e actividades pessoais.

Este tipo de redes dispõem de milhões de utilizadores activos, que estão constantemente a enviar e partilhar uma imensidade de informação, tornando este tipo de redes muito apelativas.

As redes de sensores sem fios e outros dispositivos móveis evoluíram muito nos últimos tempos, fornecendo um leque alargado de sensores que estão sucessivamente a adquirir dados do mundo físico. Tem sido uma área que nos últimos anos teve um grande desenvolvimento, que proporcionou o fabrico de nós sensores cada vez mais pequenos e com mais capacidade energética. As empresas e universidades têm apostado na investigação, contribuindo assim, para a criação de protocolos de comunicação que oferecem a estas redes, os requisitos que elas necessitam.

Também os *smartphones* têm evoluído cada vez mais, tornando-se mais rápidos e mantendo a sua capacidade energética. Presentemente, existem muitos operadores que dispõem de pacotes de dados acessíveis, permitindo assim, que estejam em constante comunicação com a Internet. Posto isto, os *smartphones* juntamente com os seus sensores (*gps*, giroscópio e acelerómetro), serão um grande contributo para a aquisição de dados reais para um mundo virtual. Ainda assim, apesar de estarem presentes, não são devidamente utilizados para recolher informação de contexto ou de actividades onde estamos inseridos, permanecendo ainda muito aquém das suas potencialidades.

Este conjunto de tecnologias e redes sociais abrem assim uma oportunidade para a construção de sistemas que tem ao seu dispor uma grande quantidade de informação. Esta quantidade de informação analisada com os métodos correctos pode influenciar a vida das pessoas de um modo positivo e, ajudar em alguns problemas diários com a utilização dos dispositivos que já se encontram junto de nós. No entanto, existem ainda alguns problemas técnicos que necessitam de especial cuidado.

A dissertação vem assim aproveitar estas novas tecnologias e as redes sociais para criar uma plataforma que se denomina por *SocialSense*. Esta plataforma tem como objectivo geral efectuar a interligação de diferentes tecnologias como as *WSNs* (*Wireless Sensor Network*), sensores médicos, *smartphones*, entre outros. Este conceito será assim utilizado para a elaboração de dois projectos distintos Avós e Netos(as) e *Youth5G*.

1.2. Objectivos

Quando observo à minha volta o dia-a-dia das pessoas, encontro uma infinidade de sistemas possíveis de serem criados, com as tecnologias e dados já existentes. Estes sistemas podem solucionar problemas de uma sociedade através da utilização das várias tecnologias.

Assim, a plataforma *SocialSense* nasce com o objectivo de realizar a fusão entre tecnologias existentes, através de um sistema modular, contribuindo de forma positiva para a sociedade ubíqua através de dois projectos concretos.

O primeiro projecto designado “Avós e Netos” pretende apoiar instituições de contexto social, nomeadamente instituições de apoio e acolhimento a crianças e jovens socialmente vulneráveis. Pretende-se assim com o uso das tecnologias, que estes jovens tenham uma maior interacção com o mundo exterior, criando laços de afecto com pessoas externas à instituição através do apadrinhamento da criança/jovem. Este contexto serviu assim para criar os designados “Avós Virtuais” que apesar de não possuírem nenhum grau de parentesco se encontram nas suas casas ou lares a cuidar do neto adoptivo.

Para que este projecto se torna-se possível, construí assim uma aplicação na rede social *Facebook* que interagisse com a plataforma *SocialSense* e suas tecnologias. Deste modo, apesar dos avós estarem fisicamente longe dos netos adoptivos tornou-se possível que estes pudessem acompanhar e cuidar a sua vida dentro do espaço institucional.

Logo os objectivos principais para este projecto foram:

- Desenvolver uma plataforma central intitulada *SocialSense*;
- Desenvolver uma aplicação *web* para a rede social *Facebook*;
- Integrar os mecanismos de intermitência e localização criados pelos colegas de investigação David Nunes e André Pinto;
- Trabalhar em parceria com investigadores das Ciências Sociais.

Além do projecto Avós e Netos, ao longo do primeiro semestre foi efectuada uma candidatura a um projecto *SapoLabs*. Esta candidatura depois de aprovada, originou a construção de uma aplicação ubíqua, dentro da plataforma *SocialSense*.

Este projecto teve como principal temática a identificação de locais nocturnos com uma maior afluência, através da utilização de dados colaborativos. Através do feedback dado pelos utilizadores pretendia-se que estes pudessem identificar diferentes tipos de locais nocturnos como bares, discotecas, eventos culturais, etc. Para a Sapo, este projecto traria benefícios como publicidade dirigida através da aplicação e dados sobre os locais mais frequentados.

Durante o segundo semestre construí assim uma aplicação *Android*, que interagisse com a plataforma *SocialSense* e com a rede social *Foursquare*. Além do desenvolvimento da aplicação, foi necessário adicionar novos serviços à plataforma *SocialSense*. Um dos serviços que foi adicionado foi o serviço de privacidade.

Os objectivos principais para este projecto foram:

- Continuar a desenvolver e adaptar a plataforma *SocialSense*, através deste novo caso de estudo;
- Construir uma aplicação ubíqua que interaja com redes sociais como o *Foursquare*;

- Implementar os mecanismos de privacidade desenvolvidos pelo colega de investigação Valderi Leithardt;

1.3. Estrutura do Documento

Este documento encontra-se organizado em nove capítulos como podemos observar na Figura 1.

No Capítulo 1 é discutida a motivação subjacente a este estágio e os objectivos a atingir.

No Capítulo 2 são apresentadas diferentes tecnologias de aquisição de dados, que podem ser utilizadas/ligadas com a plataforma que construí e o significativo crescimento das redes sociais, como elas se inserem e interagem na vida das pessoas.

No Capítulo 3 é apresentado o estado de arte, onde são discutidos tópicos relevantes e inseridos no projecto como: uma análise comparativa, com semelhanças e diferenças das nossas aplicações com as existentes; diferentes métodos de localização *indoor* e *outdoor*, dada a sua importância no âmbito do vigente estágio; mecanismos de ligação intermitente; e, por fim, iremos abordar alguns métodos que garantem a privacidade dos utilizadores.

No Capítulo 4 iremos mencionar algumas características da nossa plataforma, bem como, o planeamento e método de trabalho seguidos durante o presente estágio/dissertação.

No Capítulo 5 serão apresentados alguns estudos iniciais sobre a plataforma *SocialSense*, bem como o protótipo implementado durante o primeiro semestre.

O Capítulo 6 apresenta o projecto *SocialSense* Avós e Netos. O capítulo inicia-se com uma breve descrição do projecto seguido da especificação, arquitectura do sistema, demonstração da aplicação, dificuldades encontradas e uma pequena conclusão.

O Capítulo 7 apresenta o projecto *SocialSense Youth5G*. Neste capítulo é apresentado uma breve descrição do projecto, seguindo-se da especificação, arquitectura, demonstração da aplicação, dificuldades encontradas e a conclusão.

Em penúltimo, encontra-se o Capítulo 8 que apresenta os objectivos da fase de testes, bem como os testes efectuados às duas plataformas.

Por fim, o Capítulo 9 apresenta algum trabalho adicional efectuado paralelamente à presente Dissertação/Estágio.

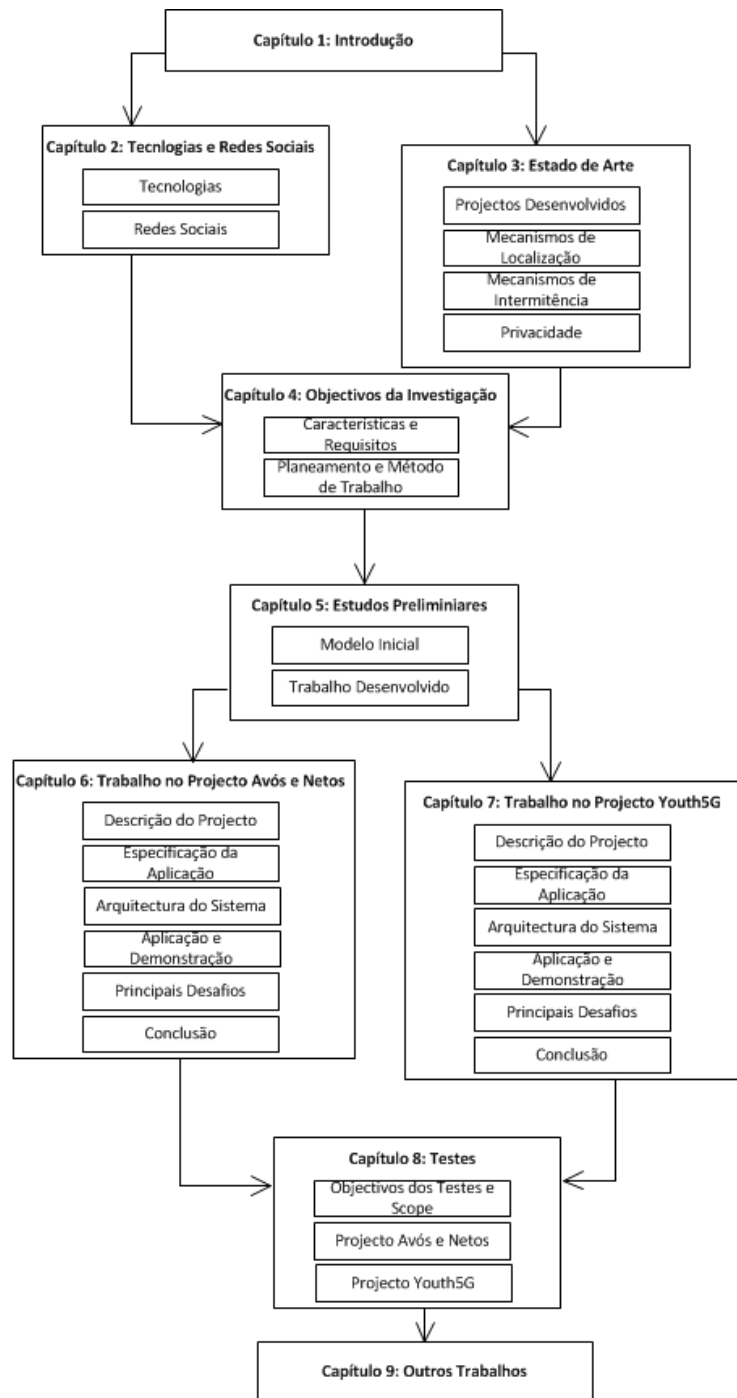


Figura 1 - Estrutura da Dissertação

Capítulo 2

Tecnologias e Redes Sociais

Nos últimos anos algumas tecnologias tiveram um grande avanço científico, sendo possível aplicá-las na vida real. Assim sendo, o conceito defendido por Mark Weiser “*Ubiquitous computing is the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user*”[2] de sistema ubíquos tornou-se cada vez mais evidente nos dias de hoje.

Uma destas tecnologias foram as redes de sensores sem fios que com o tempo se tornaram muito populares na comunidade, graças à sua possível aplicabilidade em diferentes áreas como: indústria, saúde, doméstica, etc. Devido à grande investigação nesta área os equipamentos têm-se tornado cada vez mais pequenos e os seus custos de produção diminuíram o preço final. Temos assim hoje em dia uma grande quantidade de sensores como: *WTNS, Wec, Telos, Spec, Ucla, PicoBeacon*, etc [3]. Assim, é possível desenvolver redes de sensores cada vez mais baratas, com uma grande quantidade de diferentes sensores, com baixo consumo energético e até com suporte de IP.

Com a evolução dos telemóveis e o aparecimento dos *smartphones* tornou-se possível que os seus utilizadores transportem uma grande quantidade de sensores (que nem sempre estão a ser utilizados), o que proporciona um grande espaço para o desenvolvimento de aplicações ubíquas. Estes *smartphones* têm tido um crescimento estrondoso, existindo nos dias de hoje a um preço relativamente acessível e fornecendo um vasto conjunto de aplicações. Este crescimento deu-se principalmente à grande popularidade de alguns sistemas operativos como o *Ios* e o *Android*, que melhoraram a interação com o utilizador e possuem actualmente diferentes *markets* na Internet. Através destes *markets* os utilizadores podem comprar uma grande quantidade de aplicações com diferentes utilidades.

Nos últimos anos assistimos também a um grande crescimento das redes sociais, que têm de dia para dia cada vez mais utilizadores, e se encontram constantemente a partilhar dados pessoais, de contexto, gostos, etc.

O conjunto destas tecnologias pode proporcionar assim uma nova experiência e tipos de aplicações que permitam aos utilizadores das redes sociais, combinar o mundo virtual com o mundo real, como é o caso dos projectos desenvolvidos nesta dissertação: *Youth5G* e *Avós e Netos(as) Virtuais*. Podemos assim, utilizar este conceito em áreas como a saúde e estudos sociais.

Este capítulo apresenta assim, um conjunto de tecnologias disponíveis e de grande importância para o presente estágio, assim como, a evolução das redes sociais nos últimos anos. Para finalizar, em cada secção será apresentada uma conclusão que relaciona os diferentes tópicos abordados com os projectos realizados.

2.1. Tecnologias

Como foi dito anteriormente existem vários tipos de tecnologias que podem ser usadas nos dias de hoje para adquirir dados do mundo físico. Iremos descrever algumas destas tecnologias bem como diferentes aplicabilidades que estas podem ter no mundo real.

A tecnologia *RFID* (*Radio Frequency Identification*) foi inventada após a segunda guerra mundial mediante os avanços na área do radar. No entanto, o seu grande crescimento foi por volta de 1970, onde um conjunto de investigadores em parcerias com universidades e algumas empresas fizeram um grande trabalho de investigação conseguido assim reduzir o seu custo de produção, tal como o seu tamanho.



Figura 2 - *RFID*

Fonte: http://pt.wikipedia.org/wiki/Identificação_por_radiofrequência

Recentemente, esta tecnologia é utilizada em diferentes aplicabilidades como: produtos alimentares, sistemas de pagamento nas auto-estradas, cartões de acesso, sistemas de segurança, identificação de animais, localização, etc.

O sistema é constituído por dois componentes: a *tag* e o leitor. A *tag* ou *transponder* está localizado no objecto que se pretende identificar, possuindo uma antena e um microchip que contem a informação de identificação do objecto. O leitor é geralmente maior e adquire os dados presentes na *tag*, podendo também escrever dados nesta.

Para o leitor poder ler as informações presentes na *tag*, este tem que estar dentro da sua zona e transmitir um sinal, que normalmente serve de alimentação para a *tag*. Quando a *tag* receber a energia ela “acorda” e retransmite os dados com o uso da energia recebida. Como podemos ver, este sistema permite que as *tags* estejam implementadas sem qualquer fonte de energia e podem na mesma comunicar os dados, quando é necessário, proporcionando um sistema auto-suficiente, que pode ser utilizado em muitas aplicações como a localização *indoor*.

Outras tecnologias que têm crescido nos últimos anos foram as redes de sensores sem fios. Presentemente, existe uma grande quantidade de sensores existentes no mercado para as mais variadas funções. Entre elas, sensores magnéticos, som, sensores de temperatura, acelerómetro, giroscópio, etc. Muitos destes sensores são utilizados em nós móveis, juntamente com tecnologias sem fios, desenvolvendo assim o princípio das redes sem fios. Estes conjuntos de pequenos nós, dispõem de capacidades de *sensing*, processamento e transmissão de dados através de uma ligação sem fios.

Comparando este tipo de redes, às redes actuais, facilmente verificamos que são inúmeras as diferenças entre elas. As redes de sensores sem fios, servem um género de aplicações diferentes e têm que se adaptar a elas, enquanto as redes actuais, servem praticamente o mesmo género de aplicações. Na construção destas redes temos que ter sempre em conta os problemas energéticos, escalabilidade, custos, mudanças de topologias bem como o fraco processamento e a pouca memória. Nos últimos anos, investigadores de todo o mundo têm tentado combater algumas destas falhas, criando novos protocolos e implementado mecanismos energéticos. Perante algumas dificuldades, os investigadores acreditam que algumas vantagens como flexibilidade, tolerância a falhas, baixo custo da tecnologia e o seu rápido desenvolvimento, fazem com que esta tecnologia promissora esteja um dia presente no nosso dia-a-dia de um modo ubíquo e autónomo.

Alguns destes problemas referidos, já apresentam nos dias de hoje, soluções vindas da área de investigação como os problemas energéticos. Alguns investigadores têm por exemplo estudado a capacidade de as redes de sensores sem fios poderem utilizar diferentes fontes de energia, como vibrações, explosões de energia e a construção de circuitos auto-alimentados, como o *chip* desenvolvido pela *STMicroelectronics*[4], que permite a captura da energia do ambiente. Como esta tecnologia tem baixos custos, um *chip* deste género está disponível para venda por preços que rondam os 0.60 dólares.

Em simultâneo, grupos como o *IEEE 802* desenvolvem *standards* de camada física e camada *mac* (*Media Access Control*) que diferem de muitos *standards* actuais. Um destes *standards* como, por exemplo, o 802.15.4 pretende fornecer baixas velocidades para dispositivos ubíquos e baixos custos, sendo a base para outros *standards* como o *ZigBee*[5].

Em termos de aplicação em diferentes áreas, nos dias de hoje áreas como a industrial, militar, localização, segurança, automóveis, agricultura, comercial e médicas contam com a presença deste tipo de redes. Na área industrial pode ser usado para controlo e monitorização de máquinas, na área militar para soluções de detecção de intrusos e na agricultura em monitorização dos diferentes elementos químicos presentes no solo.

Em suma, podemos dizer que no futuro devido à elasticidade desta tecnologia servir muitas áreas juntamente com a introdução do IPV6, será possível que cada nó sensor tenha uma identificação única, tornando-se presente no nosso dia-a-dia.



Figura 3 - Diferentes nós de redes de sensores sem fios.

Retirado de: <http://www.cister.isep.ipp.pt/wsn/>

Outro tipo de tecnologia que tem vindo a surtir um grande impacto e uma grande utilização são os *smartphones*. O primeiro *smartphone* IBM Simon [6], foi desenvolvido pela IBM e continha um calendário, livro de endereços, relógio, calculadora, bloco de notas, *e-mail*, faxes e jogos. Apesar do conceito de *smartphone* ser um pouco abstracto e não estar definido, o IBM Simon já não se encontra mais nessa definição comparado com o que hoje em dia estes novos *smartphones* possibilitam.

Actualmente, um *smartphone* é caracterizado como um dispositivo que além de oferecer as funções normais de um telemóvel, oferece um conjunto de recursos avançados através de um sistema operativo que, por sua vez, permite que os utilizadores criem e executem aplicações nele. Assim, se pensarmos em telemóveis e *PDA's*, o *smartphone* é a junção dos dois num único dispositivo.

Ao contrário dos telemóveis, os *smartphones* possuem um sistema operativo que disponibiliza uma *API* (*Application Programming Interface*) de programação[7], permitindo aos utilizadores a criação de diversas aplicações, e utilização por parte das aplicações do diverso *hardware* presente nos dispositivos. Os telemóveis, utilizam um *software* proprietário que não é reutilizável em diferentes dispositivos. Alguns dos mais importantes sistemas operativos de *smartphones* são o *Android*, *iOS*, *Windows Phone 7* e *Symbian* como podemos visualizar na Figura 4.



Figura 4 - Smartphones, diferentes SO.

Retirado de <http://blog.au-quotidien.net/actualite/cadeau-de-noel-quel-smartphone-offrir-pour-noel/>

Como estes sistemas operativos suportam aplicações criadas pelos utilizadores, as grandes companhias como a Nokia, Google, Apple e Microsoft, criaram mercados onde os utilizadores podem disponibilizar as aplicações gratuitamente ou pagas. Como podemos ver pelo Gráfico 1 o volume de negócios destes *markets* tem vindo a crescer nos últimos anos.

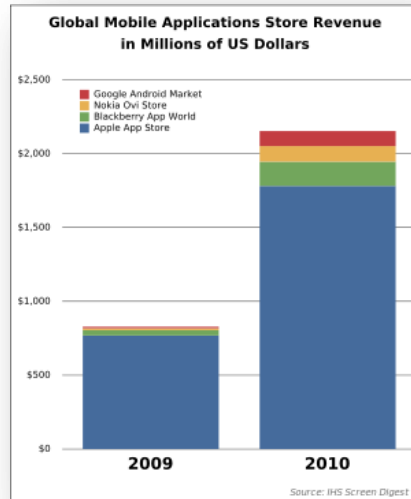


Gráfico 1 - Volume de negócio dos markets de smartphones.

Retirado de: http://en.wikipedia.org/wiki/File:Global_Mobile_Applications_Store_Revenue.svg

Estes *markets* têm uma grande vantagem para os investigadores sendo muitas vezes esquecida, nomeadamente, a possível distribuição de aplicações para estudos em grande escala, dado o seu grande número de utilizadores.

Estas aplicações permitem mudar o dia-a-dia dos utilizadores, fornecendo diferentes ferramentas com diversas utilidades, indo ao encontro das suas necessidades, libertando o utilizador dos pormenores (tratando apenas de o lembrar).

Uma diferença entre os telemóveis e os *smartphones*, encontra-se na presença de inúmeros sensores no *hardware* dos *smartphones*. Os sensores disponíveis neste tipo de dispositivos são normalmente, sensores de luminosidade, acelerómetro, giroscópio, proximidade, magnético, etc. Além da introdução destes sensores para conseguir uma maior interactividade com o utilizador, a utilização de arquitecturas já existentes como *ARM (Advanced Risc Machine)* [8], permitiu a simplificação das instruções do *cpu (Central Processing Unit)*, relativamente à arquitectura *x86*, originando a possibilidade de ter mais velocidade e menos energia gasta e, conseqüentemente, um menor aquecimento. Ainda assim, nos últimos anos, vimos a capacidade de armazenamento dos actuais *smartphones* aumentar para alguns GBs, permitindo o armazenamento de grandes quantidades de dados destes sensores. Relativamente à comunicação, actualmente, todos os modelos já dispõem de uma ligação *Wifi (Wireless Fidelity)* permitindo uma ligação à internet.

As capacidades que estes dispositivos transportam, permitem que se transformem numa ferramenta ideal para o sensoriamento no nosso dia-a-dia, recolhendo diferentes tipos de contextos, aplicações (como iremos apresentar mais à frente no projecto *CenceMe*[9] e construindo grandes redes colaborativas em que os utilizadores partilham os dados sobre os diferentes ambientes que estão inseridos.

No entanto, os *smartphones* também têm as suas desvantagens que apesar dos enormes esforços elas ainda continuam presentes. A maior desvantagem nestes sistemas continua a ser a duração da bateria, onde comparado com os telemóveis é muito inferior, durando por vezes dois ou três dias, dependendo da utilização. Apresenta-se como uma das soluções para este tema energético activar os sensores aquando necessário.

Outros dispositivos de monitorização têm aparecido nos últimos tempos como as *smartshirts*, que consistem num género de peça de roupa constituída por *E-Textils*[10]. Estes têxteis possuem dispositivos e ligações electrónicas intrínsecas, com uma grande flexibilidade, permitindo o seu uso no dia-a-dia como se fosse um tecido normal. Os *E-Textils*[10] permitem o fabrico de roupa com capacidades computacionais e dispositivos electrónicos embebidos.

Devido à grande utilidade deste tipo de tecidos, muitas empresas como *Numetrex*, *Adidas*, *SmartLife*, etc, têm criado camisolas para diferentes aplicações. Porém, estas camisolas diferem apenas nos sensores e funcionalidades implementadas, mantendo a mesma definição.

As *smartshirts* disponibilizam diferentes sensores como: medidores de frequência cardíaca; taxa de respiração; temperatura corporal; actividade; postura; etc. Como podemos constatar a tecnologia já se encontra tão avançada que muitos tipos de sensores podem ser instalados nas *smartshirts*. Em termos de comunicação, existem diferentes tecnologias que podem ser usadas como: *Wifi*; redes móveis; radiofrequência e *Bluetooth*. Algumas marcas desenharam até estas peças, de modo a poderem transportar *smartphones* dado o seu potencial como *gateway* desta tecnologia.

As áreas e aplicações que estas *smartshirts* podem servir são inúmeras, mas destacam-se como mais importantes a saúde e o desporto.

Em aplicações como a saúde podem ser utilizadas, para monitorizar doentes, através de vários sensores como frequência cardíaca, respiração, temperatura, actividade, postura. Imagine-se, por exemplo, um hospital onde os doentes que quisessem ser monitorizados tivessem vestido uma *smartshirt*. Ainda assim, poderia estar ligada a um *smartphone* através de *wifi*, de modo a que o(a) enfermeiro(a) ao mesmo tempo pudesse monitorizar todos os doentes. Se algum sensor detectasse algum valor irregular, poderia lançar um aviso e, deste modo, alertar o(a) enfermeiro(a) para uma situação de perigo.

Na área de desporto, esta tecnologia pode ser utilizada para medir a performance dos(as) atletas e, com isto, conseguir que estes(as) tenham um maior rendimento, como podemos ver na Figura 5, a utilização de uma *smartshirt* UA-E39.

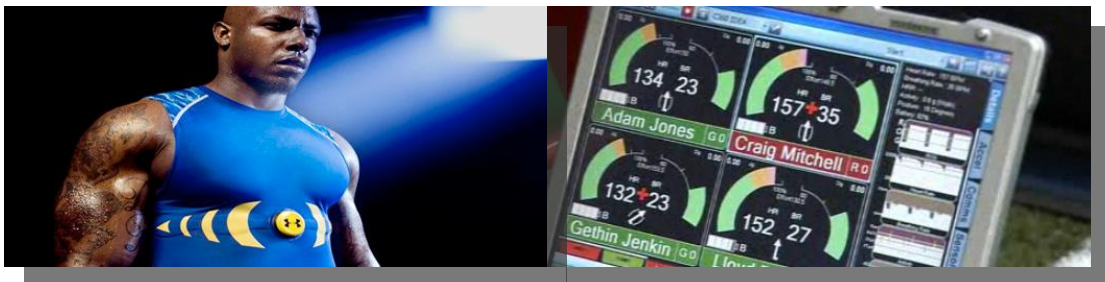


Figura 5 - Smartshirt UA-E39

Retirado de: <http://sc1sports.com/e39-under-armour-tracking-technology/>

Nos últimos anos, os controlos de detecção de movimento também tem tido uma grande evolução, devido às consolas de videojogos.

Quando lidamos com representações do mundo real, ou seja, mundos virtuais, muitas vezes a experiência vivida é muito importante, no sentido que o utilizador deve ter uma percepção o mais real possível desta interacção. Até há pouco tempo, essa interacção com os mundos virtuais era efectuada através de teclado e rato, utilizando combinações de teclas para simular o movimento do utilizador.

No entanto, como sabemos para conseguirmos reproduzir os movimentos que pretendemos, por vezes, necessitamos de horas até aprendermos as combinações necessárias das teclas, juntamente com alguns movimentos e cliques do rato. Esta interacção, apesar de ser a mais utilizada, não é nem de longe a que proporciona uma melhor experiência ao utilizador, uma vez que, perde a maior parte do tempo a aprender como traduzir movimentos em combinações de teclas.

Como a indústria de videojogos pretende que o utilizador tenha a melhor experiência possível, a ligação dos videojogos com o mundo virtual é inevitável. Aproveitando algumas tecnologias que se usam no mundo dos videojogos, bem como algum *know how* desta indústria para captar estes movimentos, poderá ser uma boa oportunidade para este projecto a utilização destas tecnologias.

Na indústria de videojogos o primeiro dispositivo que permitiu detecção de movimento, foi o *WiiRemote*, que através de um comando dispunha de uma câmara e acelerómetros que possibilitava obter os movimentos nos três eixos (vertical, lateral e profundidade)[11].

Aproveitando o grande sucesso da *Wii* que se destacou dos restantes concorrentes vendendo cerca de 70 milhões de unidades em quatro anos[12] e devido à sua interacção com os utilizadores, os restantes concorrentes lançaram outros controlos como o *PlaystationMove* e *Microsoft Kinect*. Este último, difere-se dos outros controlos devido à sua precisão e no quão realista ela transforma os movimentos do utilizador para o mundo virtual. Através da utilização de câmaras de vídeo, o controlo consegue fazer um reconhecimento visual, identificando o utilizador. Também a existência de um sensor de profundidade permite que o dispositivo tenha a percepção do ambiente do utilizador e, por fim, através de um microfone que utiliza várias matrizes é possível isolar as vozes de diferentes utilizadores para controlos de voz.



Figura 6 - WiiRemote, PlaystationMove e Kinect.

Retirado de: <http://otakrazys.blogspot.com/2011/03/playstation-move-vs-kinect-vs-nintendo.html>

Devido ao seu grande sucesso, a Microsoft disponibilizou um *SDK (Software Development Kit)* que permite o uso deste controlador em outras plataformas como, por exemplo, o *PC (Personal Computer)*. A meu ver a utilização deste controlador é uma mais-valia, visto que, pode ser utilizado em várias aplicações como, por exemplo, fazer a localização *indoor* de uma pessoa ou várias pessoas numa sala.

Em suma, todas as tecnologias apresentadas neste capítulo são relevantes para o presente estágio. Porém, apenas duas delas, nomeadamente, as redes de sensores sem fios e os *smartphones* foram utilizadas. Conforme será descrito em seguida, espera-se no futuro, dando continuidade às aplicações desenvolvidas nesta dissertação, integrar as restantes tecnologias e alargar o leque de serviços disponibilizados.

Tendo em conta a sua versatilidade, as redes de sensores sem fios (nomeadamente, o tamanho e a capacidade energética dos nós) foram a tecnologia escolhida no projecto “Avós e Netos(as) Virtuais” para disponibilizar o posicionamento *indoor* das crianças, para a construção de um mecanismo de alarmes e para efectuar videochamadas baseado na proximidade da criança ao computador da instituição.

No projecto *Youth5G* foram escolhidos os *smartphones* não só pelo seu grande uso no dia-a-dia dos utilizadores, mas também, porque contêm um grande conjunto de sensores úteis para adquirir dados. Estes sensores num futuro próximo servirão para adquirir diferentes dados como gostos musicais, actividade, localização e afluência de locais que serão utilizados para fazer publicidade dirigida por parte do operador Sapo.

Apesar de nem todas as tecnologias descritas terem sido utilizadas, como trabalho futuro algumas das restantes podem ser integradas nos projectos de diferentes modos: o *Kinect* para detecção de actividade humana em ambientes *indoor*; a *smartshirt* como dispositivo para monitorizar a actividade dos netos no seu dia-a-dia; e, por fim, o *rfid* como dispositivo de localização *indoor* ou como dispositivo de identificação de objectos ou locais.

Como Mark Weiser defende: “*Prototype tabs, pads and boards are just the beginning of ubiquitous computing. The real power of the concept comes not from any one of these devices; it emerges from the interaction of all of them*”[13] o poder da computação ubíqua está no conjunto das tecnologias e não numa só.

2.2. Redes Sociais

O aparecimento do conceito *Web 2.0* defende uma segunda geração da *web* como uma plataforma que possibilita um ambiente de interacção para os utilizadores. Este conceito surgiu perante o aparecimento das redes sociais, que motivam diariamente milhões de utilizadores a partilhar diferentes tipos de conteúdos.

Entendidos defendem que o conceito das redes sociais é definido por: “*Social networking websites are online communities of people who share interests and activities, or who are interested in exploring the interests and activities of others. They typically provide a variety of ways for users to interact, through chat, messaging, email*”[14].

Através deste novo conceito, foi possível que os utilizadores vissem a *web* de uma nova perspectiva, levando-os a querer partilhar informação, constantemente, com outros utilizadores, assim como criar relacionamentos.

A crescente vontade dos utilizadores em partilharem conteúdos na comunidade, fez com que os programadores das redes sociais implementassem um grande conjunto de ferramentas

que possibilitam a partilha de fotografias, interesses, aplicações, *hobbies*, dados pessoais e até a realização de eventos colectivos. Além disso, permitem a criação de grupos que, por sua vez, são constituídos por uma heterogeneidade de pessoas que partilham as mesmas ideias, gostos, etc. Actualmente a construção destes grupos, já originou revoltas sociais que provocaram, por exemplo, a queda do regime ditatorial do Egipto[15].

O grande crescimento das redes sociais, possuindo actualmente milhões de utilizadores, deve-se segundo o autor[14] a dois importantes factores:

- Em primeiro lugar, a capacidade para formar diferentes redes que permitem a partilha de dados pessoais, localização, ideais, interesses, etc;
- Em segundo lugar, a facilidade que os utilizadores possuem de partilhar conteúdos, não necessitando de perder tempo na construção de *websites*.

Como se pode ver, as redes sociais diferem de outras aplicações disponíveis na Internet, uma vez que, permitem uma maior facilidade de expressão, não prendendo o utilizador a detalhes técnicos, como acontecia até há pouco tempo.

Esta facilidade de expressão levou a muitos entendidos e críticos de áreas sociais a questionarem-se sobre o real valor destes relacionamentos e a experiência humana que é transmitida. Na minha opinião, o real valor destes relacionamentos está no fortalecimento de relações fora das redes sociais, ou seja, a possibilidade de virtualmente poder-se manter ligações que de outro modo seria impossível. Exemplos destes relacionamentos são: a possibilidade de pessoas que estão isoladas, por motivos de doença ou geográficos, possam manter um contacto com a família, e a ligação de serviços públicos e organizações que se encontrem afastadas da população.

Como resultado, estes novos relacionamentos criados pelas redes sociais diferem de outros que se encontram actualmente na Internet. Por exemplo, os utilizadores que dêem um *feedback* sobre um livro na *Amazon*, não têm qualquer ligação fora da rede ou em comum, dando apenas informação sobre aquele vendedor. Contrariamente, os utilizadores que formem um grupo e partilhem ideias sobre a construção de uma tecnologia, têm um relacionamento muito mais forte.

Como foi falado no Capítulo de Tecnologias, os dispositivos móveis como os *smartphones*, trouxeram um grande conjunto de funcionalidades aos utilizadores, assim como, uma grande mobilidade. Esta oportunidade trouxe às redes sociais o conceito de ubiquidade, permitindo que os utilizadores em qualquer lugar, a qualquer hora pudessem partilhar conteúdos.

Recentemente, existem inúmeras redes sociais, mas todas com particularidades distintas. Com base nestas diferenças a organização[16] criou um conjunto de categorias que identificam os diferentes tipos de redes sociais:

- **Baseadas no perfil do utilizador:** este tipo de redes sociais dá maior importância ao perfil do utilizador, fornecendo um espaço onde ele se possa descrever. Exemplos: *Facebook* e *MySpace*;
- **Baseadas em Conteúdos:** ao contrário do anterior, este tipo foca-se nos conteúdos deixando o utilizador em segundo lugar. Exemplo: *Flickr*;
- **White-Label:** baseado em comunidades, onde os utilizadores podem criar ou pertencer a comunidades. Exemplo: *WetPaint*;

- **Ambientes Virtuais:** os utilizadores interagem uns com os outros através de *avatars*. Exemplo: *SecondLife*;
- **Micro-blogging:** a interacção entre os utilizadores é feita através de pequenas mensagens de estado. Exemplo: *Twitter*.

Em conclusão, as redes sociais estão cada vez mais presentes na vida social das pessoas, bem como das instituições e organizações. Esta presença permitiu uma união e partilha de conteúdos como nunca se tinha visto, levando as pessoas a quererem em todo o lado a toda a hora partilhar conteúdos sobre si, sobre o que fazem, onde estão, etc.

Esta partilha de informação apresenta, por um lado, graves problemas em questões de privacidade, pois sem o utilizador se dar conta pode partilhar dados pessoais que poderão ser utilizados de um modo ilícito. Por outro lado, este conjunto de dados disponíveis, podem ajudar a resolver alguns problemas como no caso do projecto *Youth5G*, em que os gostos musicais das pessoas que utilizam a aplicação são necessários, para determinar que géneros de música o utilizador gosta mais.

Finalmente, estes projectos darão um contributo às redes sociais através de diferentes tecnologias, realizando a inserção de dados sobre os utilizadores de um modo autónomo.

Capítulo 3

Estado de Arte

O estado de arte desta dissertação pretende demonstrar alguns conceitos úteis que foram estudados e/ou utilizados no presente estágio.

Em primeiro lugar, será apresentado um conjunto de projectos desenvolvidos (académicos e comerciais) que se identificam de algum modo com os projectos Avós e Netos(as) e *Youth5G*. Assim será apresentado uma pequena descrição sobre cada um deles.

Em segundo lugar, serão abordados alguns mecanismos de localização existentes. Pretende-se mostrar o conjunto de técnicas de localização indoor e outdoor e as diferentes tecnologias utilizadas. Segue-se uma conclusão sobre as técnicas utilizadas no projecto *Youth5G* e Avós e Netos(as).

Em terceiro lugar, serão especificados alguns mecanismos de intermitência e a sua utilidade em aplicações e sistemas que nem sempre conseguem ter uma ligação activa.

Por último, será abordado um tópico sobre privacidade, que demonstrará alguns problemas existentes. Este tópico termina com uma pequena introdução a um novo modelo proposto por um colega de investigação e no qual também participei no âmbito deste estágio, que propõe um sistema de privacidade associada ao ambiente.

3.1. Projectos Desenvolvidos

Seguidamente, serão apresentados alguns trabalhos relacionados com o projecto Avós e Netos(as) Virtuais e *Youth5G*. O intuito deste capítulo é obter uma percepção dos trabalhos existentes, assim como, analisar os seus pontos fortes e pontos fracos. No final da apresentação dos sistemas/aplicações relacionados com cada projecto, será feita uma síntese de todos eles, demonstrando as principais potencialidades que terão para o presente estágio.

3.1.1. Aplicações Semelhantes ao Projecto Avós e Netos(as) Virtuais

3.1.1.1. *Sound Sense*

Um dos sensores utilizados nesta aplicação é o microfone. O microfone é um sensor capaz de detectar a actividade humana em diversificados contextos e está presente em quase todos os dispositivos móveis. A detecção do tipo de actividade e localização é possível através da captura e análise de sons do microfone.

A aplicação *SoundSense*[17] tem como principal objectivo disponibilizar uma *framework* que seja capaz de classificar os eventos adquiridos nos dispositivos móveis. Para isso, deverá ser escalável (cada pessoa tem os seus diferentes tipos de sons), deverá ser utilizada em diferentes contextos e deverá ser imperceptível.

Num mundo repleto de diferentes sons, um dos métodos que poderíamos pensar para classificar a sua diversidade, seria adquirir uma grande quantidade de amostras e treinar diferentes classificadores para todos os possíveis sons. Mas, a meu ver, parece-me que este procedimento é praticamente impossível dado cada pessoa estar exposta somente a um subconjunto de sons.

Ao criar este tipo de *frameworks* que capturam dados sobre os utilizadores, estes têm que estar convictos que a sua privacidade está salvaguardada. O modo como os autores combateram este problema foi correr a aplicação localmente, sem nenhuma interacção com servidores. Os sons provenientes do ambiente não serão guardados na íntegra, pelo contrário, apenas as características das amostras são guardadas pela aplicação.

Relativamente aos mecanismos de privacidade, os autores usaram um meio de protecção físico, ou seja, o microfone dos telemóveis apenas funciona a 4KHz e não a 8KHz sendo muita informação sobre o meio envolvente perdida.

Quando falamos em desenvolvimento de aplicações para dispositivos móveis, temos que ter em conta a sua fraca velocidade de processamento, dificuldades energéticas e memória. Ao invés, devemos ter sempre em pensamento que as aplicações não devem interromper o funcionamento normal do telemóvel.

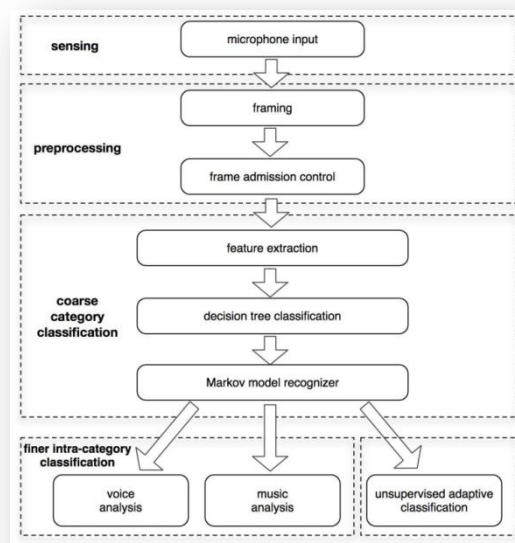


Figura 7 – Arquitectura SoundSense
Retirado de [17]

Como podemos ver na figura anterior, a arquitectura do *Soundsense* é composta pelas seguintes etapas: pré-processamento, classificação geral por categorias e finalmente classificação detalhada.

A função da primeira etapa (pré-processamento) começa com a segmentação de uma *stream* de áudio em várias *frames* de duração uniforme. Neste momento são extraídas as diferentes características de uma *frame* ou de uma janela para posterior classificação.

A segunda etapa tem como objectivo a classificação do som em diferentes categorias como: voz, música e som ambiente (qualquer som que não se identifica com os outros dois tipos).

A terceira etapa efectua uma análise detalhada ao som dentro daquela classificação. Quer isto dizer, que se fosse do tipo de música podíamos fazer uma análise de modo a verificar o género de música que era. Dentro desta etapa, quando o som é classificado como ambiente, a aplicação pede ao utilizador para colocar uma descrição do som.

Em termos de avaliação de recursos utilizados neste sistema, podemos dizer que o sistema preservou os recursos existentes, utilizando pouco processamento do *cpu* e pouca *off-the-shelf* memória, como consequência a duração da bateria permaneceu praticamente a mesma.

Quanto aos resultados obtidos, este modelo permitiu identificar som ambiente em 90% dos casos correctamente. Apesar da voz e música serem mais difíceis de identificar, devido a muitas músicas terem pequenos momentos de fala, o sistema classificou correctamente 80% destes sons. De um modo geral, o artigo diz que a *framework* é boa a descobrir eventos de longa duração, contudo, eventos de pequena duração por vezes são facilmente confundidos. Um dos métodos que permitiria suportar este tipo de eventos era através da diminuição do tempo de aquisição, mas como consequência iria gerar muitos falsos eventos.

Em conclusão, podemos dizer que o microfone é um bom método para identificar diferentes contextos de actividades em eventos de longa duração. Este método poderia por exemplo no futuro ser utilizado no sistema de Avós e Netos(as) para identificação dos diferentes locais onde o neto se encontra.

3.1.1.2. *CenceMe*

A aplicação *CenceMe*[9] utiliza os sensores dos telemóveis, para fornecer informações pessoais para as redes sociais. Estas informações são por exemplo: actividade, estado emocional, hábitos e dados ambientes(temperatura, ruído, etc).

Este sistema como muitos outros, tem cuidado com o modo como a aplicação gere os recursos dos telemóveis. Ao contrário do artigo anterior, onde a análise de dados era toda realizada nos dispositivos móveis, nesta aplicação devido aos problemas de recursos e devido à aplicação ser mais pesada em termos de processamento, foi decidido classificar uma parte desses dados em servidores *backend*.

À semelhança do artigo anterior é possível utilizar *tags* para identificar qualquer actividade, gesto ou som que possa ligar a um significado pessoal (ex. movimento que simboliza que o utilizador vai almoçar). Por esta razão, os actores decidiram que a saída do processo de classificação no dispositivo móvel se chamaria “*primitive*”, enquanto a saída dos servidores *backend* se chamaria “*facts*”. Estes “*facts*” são armazenados numa base de dados e posteriormente são utilizados para publicações.

O desenvolvimento da aplicação foi através de *J2ME (Java 2 Platform Micro Edition)*. Os autores encontraram alguns problemas no desenvolvimento devido aos poucos recursos disponibilizados, mecanismos de segurança e capacidade energética, alguns deles derivados ao sistema operativo *symbian* (Nokia N95).

O modo de simular a utilização desta aplicação foi através de um grupo de pessoas que durante três semanas transportaram os telemóveis.

Este sistema, pode ser muito útil para o desenvolvimento de futuros projectos como o Avós e Netos(as). Penso que, com ele poderemos tirar algumas orientações no que diz respeito às informações úteis do ambiente envolvente. O sistema consegue “perceber” um conjunto de estados do utilizador que, por vezes, podem ser interessantes para realizar uma monitorização.

3.1.1.3. *WhozThat*

O sistema de interacção das redes sociais e o mundo físico *WhozThat*[18] aborda a construção de novas ligações pessoais utilizando as redes sociais. Portanto, o seu principal objectivo é explorar a possibilidade de conhecer pessoas, que no dia-a-dia se encontram à nossa volta.

Note-se, que nos dias de hoje este tipo de aplicações passará a ser possível, graças ao crescimento das redes sociais como o *Facebook*. Estas redes permitem que esteja disponível na Internet várias informações sociais como: nome, fotografias, contacto pessoal, género, actividades, estado de relacionamento, *hobbies*, etc.

Os autores ao longo do artigo exemplificam este tipo de interacção através de alguns casos práticos. Por exemplo, quando uma pessoa se encontra num local público como um bar e à sua volta, até se encontram pessoas que partilham os mesmos gostos e interesse pelos mesmos *hobbies*, ou até conheçam amigos em comum. Como podemos observar, se existisse uma plataforma como a descrita até aqui, esta descoberta de novas amizades seria muito mais simples e interessante. Existem também mais contextos onde esta aplicação pode ser usada, como por exemplo, na identificação dos gostos pessoais, mais concretamente a utilização da aplicação para reproduzir listas musicais em diferentes ambientes, ao gosto de cada utilizador. Estes gostos musicais iriam ser retirados da informação presente nas diferentes redes sociais. Para finalizar, as diferentes aplicabilidades deste tipo de aplicações, o seu uso em termos comerciais, relacionando publicidade com os gostos dos utilizadores.

A troca deste conhecimento social é feita através da plataforma *WhozThat*[18], redes sociais e *smartphones*. De um modo geral, para existir esta interacção é necessário que os utilizadores se encontrem no mesmo local e tenham registo nas redes sociais. Assim sendo, os seus *smartphones* vão em primeiro lugar partilhar informação e em seguida utilizar as redes sociais para tirar informações de cada utilizador.

Especificamente, a identificação dos utilizadores nas redes sociais é feita através de um ID. Estes IDs são então enviados por *broadcast*, tanto por Bluetooth como por Wifi, seguindo-se o envio dos respectivos *IDs* por parte dos *smartphones* às redes sociais para efectuarem a procura de informações sobre os utilizadores. O protocolo segundo os autores é eficiente, extensível e apresenta gastos energéticos relativamente baixos.

Relativamente à aplicação musical, os autores utilizam o conhecimento que têm sobre os gostos musicais nas redes sociais dos utilizadores e através de *api's online* criam *playlist*. A maior dificuldade encontrada neste tipo de aplicação, deve-se à selecção dos gostos musicais do utilizador baseado no seu histórico. Os autores abordam que como trabalho futuro, existirá a necessidade de investigação deste tipo de algoritmos de selecção dos gostos musicais.

Como próximos passos, os autores abordam a capacidade de utilização de *multihop* na plataforma, de modo a implementar um mecanismo de *relay* entre os *smartphones*, através de *rebroadcast*, aumentando a área de procura. Quanto a este ponto encontro-me céptico devido à possibilidade de levantar problemas, por exemplo, a nível de energia.

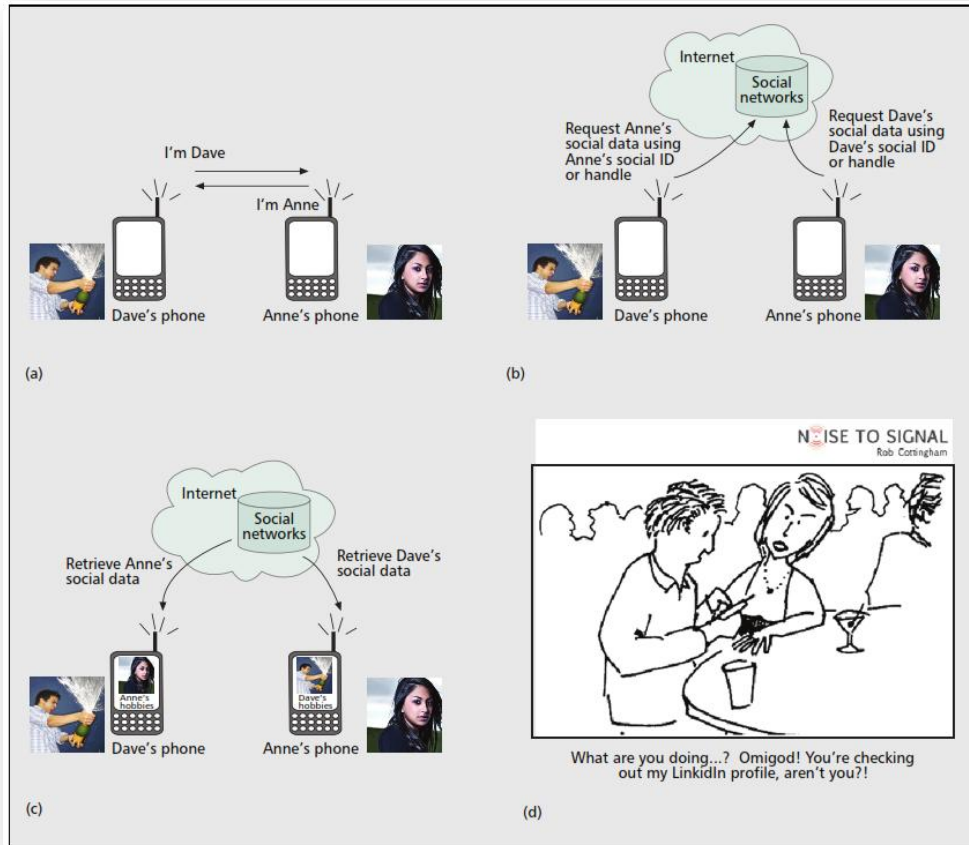


Figura 8 - Whozthat.

Retirado de [18].

3.1.1.4. *Serendipity*

Nos dias de hoje, onde a mobilidade das pessoas é cada vez mais importante, os dispositivos móveis como os *smartphones*, ganharam um lugar de destaque. O seu grande crescimento, permitiu a sua presença em praticamente todo o mundo e acessíveis a toda a gente. Além disso, vieram abrir novos horizontes, possibilitando um novo uso do seu conceito original como a possibilidade de utilização em outros tipos de aplicações.

O sistema *Serendipity*[19] utiliza esta nova abordagem aos dispositivos móveis como *smartphones* e do mesmo modo que os artigos anteriores, faz uma ligação social com ambientes virtuais. Ao contrário dos sistemas anteriores, que fundamentam o estudo com base nas redes sociais actuais, esta arquitectura assenta em redes sociais que permitem encontros entre pessoas. Deste modo, os autores desenvolveram uma arquitectura que muda a perspectiva dos dispositivos móveis, combinando as infra-estruturas de comunicação existentes com os sistemas sociais *online*.

Para fazer esta ligação utilizaram a tecnologia *Bluetooth*, que está praticamente presente em todos os dispositivos móveis. Esta tecnologia contém um mecanismo de descoberta de outros dispositivos na vizinhança, permitindo que o telemóvel funcione como um *gateway* para os sistemas online como o *Friendster*, *Monster* e *Match.com*.

Porém, no ano em que o artigo foi escrito não existia em grande escala o conceito de aplicações para os telemóveis. Assim, defendiam que a sua aplicação viria mudar este conceito, dado que, permitiria a interacção dos utilizadores com as redes sociais, mesmo sem estarem à frente de um teclado num *desktop* ou num *laptop*.

Em particular, foram desenvolvidos dois modelos para o sistema com aplicabilidades distintas, existindo também, um servidor chamado *Serendipity*, que contém os registos de todos os utilizadores.

O primeiro modelo *BlueAware*, consiste numa aplicação que corre em modo *background* nos *smartphones*, enviando um *btid* (*Bluetooth Identifier*), a todos os dispositivos à escuta na vizinhança e guardando em *log* de todos os *btids* que se encontrem na sua proximidade. Quando na vizinhança é encontrado um dispositivo que não está registado no *log*, o seu *btid* é enviado para o servidor para ser identificado. Além disso, este modelo é constituído por um interface que permite que os utilizadores alterem o *log*. Relativamente às questões energéticas, os autores identificaram um gasto de energia substancial nos dispositivos, devido aos contínuos *logs* e *scans*.

O segundo modelo designado por *BlueDar* consiste num equipamento que foi construído pelos autores, com a finalidade de fazer o rastreamento de dispositivos num local fixo, através dos *ids* que se encontrem na proximidade.

Como este sistema se baseia em relacionamentos entre pessoas e num *matching* entre elas através do perfil, interesses e *hobbies*, os autores desenvolveram um algoritmo que a partir de preferências definidas pelo utilizador permite efectuar esta comparação. O funcionamento deste algoritmo começa pelo servidor central, que contém as preferências de *matching*, semelhantes a redes sociais como o *LinkedIn* e *Match.com*. Adicionalmente, os utilizadores também utilizam pesos, para determinar que preferências terão mais importância na sua escolha.

O processo para efectuar este *matching*, começa pela recepção dos *bits* e valores máximos no servidor *Serendipity*. Seguidamente, é feita uma pesquisa por esse *btid* procurando o utilizador associado, se o utilizador existir, é chamado um *script* para calcular a similaridade. Supondo que, a pontuação está abaixo do valor máximo, é então enviado um alerta para o utilizador.

Para testar o sistema foi feita uma experiência com a duração de um ano. Nesta experiência participaram cerca de 100 indivíduos que transportaram o sistema *BlueAware* dentro de um laboratório do MIT (*Massachusetts Institute of Technology*), contrariamente, o dispositivo *BlueDar* foi instalado no bar.

Através do *Serenity*, conseguiram identificar a rotina dos utilizadores, bem como as pessoas que estavam em contacto, ou seja, na proximidade. Através desta informação sobre proximidade entre as pessoas, é possível determinar, por exemplo, o tipo de relação entre elas como base em distribuição de *gauss*.

Na apresentação dos resultados, os autores construíram gráficos (Figura 9) que permitem relacionar os diferentes períodos do dia com as diferentes pessoas que o utilizador interage, tanto no local de trabalho, como no bar. Repare-se, que com este tipo de análise é possível estudar comportamentos das pessoas individualmente, como de pessoas no seu colectivo, o que pode ser aplicado a várias áreas como à Sociologia ou à Psicologia.

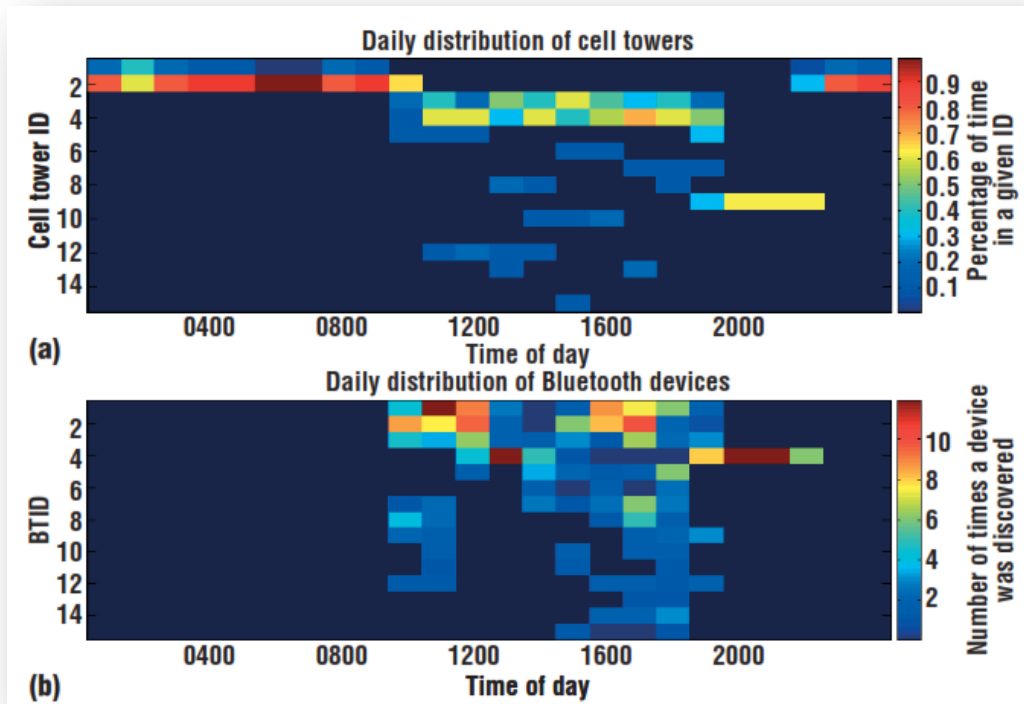


Figura 9 - Serendipity.

Retirado de [19].

Em conclusão, os autores defendem que este sistema pode ser utilizado em aplicações empresariais, *on-line* (redes sociais), comerciais e para localização. Em aplicações organizacionais pode ser útil para estudar os comportamentos dos trabalhadores. No entanto, a nível comercial poderia ser usado em publicidade direccionada aos gostos pessoais de cada utilizador. Por fim, em termos de localização, esta aplicação poderia ser usada para notificação de edifícios ou objectos à sua volta.

Numa análise conjunta aos quatro artigos, identifiquei alguns pontos semelhantes com o projecto Avós e Netos(as), como a detecção de proximidade no *SoundSense*[17], *Who's That*[18] e *Serendipity*[19], a interacção com as redes sociais e a preocupação em termos energéticos.

Contudo, existem pontos que diferem dos objectivos do corrente projecto. A plataforma a ser desenvolvida assenta num modelo para o suporte de várias tecnologias como medidores de tensão arterial, *smartshirts*, redes de sensores sem fios, etc.

Além disso, aplicações como o *Serendipity*[19] não utilizam as funcionalidades das novas redes sociais que apresentam um enorme potencial, face ao número de utilizadores disponíveis e quantidade de informação.

Em termos energéticos todas as aplicações têm este factor em consideração, no entanto, ao nível da privacidade isso já não acontece. Como podemos ver no caso do *Who's That* [18] o *id* (*Identifier*) da rede social que disponibiliza o acesso a todas as informações do utilizador, é passado entre os utilizadores sem qualquer mecanismo de segurança, ao contrário do sistema Avós e Netos(as).

Para finalizar este capítulo, nenhuma destas aplicações faz uma representação dentro das redes sociais. Com os modelos propostos nesta dissertação espera-se poder contribuir para que essa integração do mundo real com as redes sociais seja facilitada.

3.1.2. Aplicações Concorrentes ao Projecto *Youth5G*

Tal como o mercado dos *smartphones* cresceu, as redes sociais seguiram o mesmo caminho, dispondo de cada vez de mais utilizadores, que se encontram determinados e interessados em partilhar dados pessoais e do ambiente. Devido a estes factores, aplicações baseadas em dados colaborativos e redes sociais têm vindo a crescer.

Posteriormente, serão apresentadas algumas aplicações ubíquas que contêm semelhanças com o sistema *Youth5G*. Como aconteceu na secção anterior existiam muitas mais. No entanto, pretendeu-se escolher as mais significativas. Pretende-se assim, observar aplicações existentes no mercado, de modo a retirar alguns conceitos e realizar uma pequena análise.

3.1.2.1. *Waze*

No caso do *Waze*, a aplicação permite de modo colaborativo realizar uma navegação social, através dos vários utilizadores que usam o sistema. Para isto, os utilizadores podem inserir informações sobre o trânsito, disponibilizar informação do estado das vias, traçar rotas até ao local desejado, verificar o tráfego em cada via ou mesmo corrigir mapas.

Graças a estas funcionalidades, um dos principais incentivos que esta aplicação fornece ao utilizador, é assim, a possibilidade de traçar rotas através dos dados disponibilizados por outros utilizadores de forma colaborativa, que ao contrário das aplicações de navegação tradicionais, unicamente utilizam dados estáticos de navegação. Além disso, a aplicação permite também através das coordenadas *gps* (*Global Positioning System*) dos utilizadores actualizar estradas e mapas de um modo simples e económico.

3.1.2.2. *Foursquare*

O *Foursquare* enquadra-se noutra aplicação, mais virada para o domínio social e de partilha de informação sobre gostos pessoais e experiências, à semelhança do *Facebook*. Através desta aplicação, o utilizador pode partilhar as suas experiências, comentários e fotografias em termos de pontos de interesse geolocalizados e acompanhar os amigos que se encontrem em redor. Desta forma, o *Foursquare* ajuda o utilizador na escolha de diferentes locais (restaurantes, bares, etc), através da opinião dos utilizadores.

3.1.2.3. *GetGlue*

Para terminar, o *GetGlue* ao contrário dos anteriores, é uma aplicação que relaciona os gostos culturais dos diferentes utilizadores com o conceito de redes sociais, ou seja, uma rede social baseada em conteúdos. Através da partilha destes conteúdos e informações, os utilizadores podem acompanhar os gostos culturais dos seus seguidores mais propriamente, livros, filmes e músicas. Deste modo, o principal contributo desta aplicação para os utilizadores é a partilha e descoberta de novos elementos culturais.

Como resultado, o sucesso destas aplicações pode ser medido através do crescente número de utilizadores que aderem de dia para dia. Aproveitando o crescente interesse neste tipo de aplicações, a nossa proposta consiste em criar uma aplicação que, à semelhança das anteriores, utiliza dados colaborativos mas diferencia-se através de outro caso de estudo.

Pretende-se inclusivamente utilizar dados disponibilizados por redes como o *Foursquare* para retirar informações sobre os locais que serão utilizados no sistema *Youth5G*.

3.2. Mecanismos de Localização

Desde o início da descoberta do mundo, diversos navegadores criaram diferentes sistemas de coordenadas que permitissem localizar e representar locais.

Este capítulo pretende assim dar a conhecer alguns dos métodos de localização utilizados actualmente, tal como algumas das tecnologias importantes para as aplicações desenvolvidas neste estágio.

Antes de falar nas diferentes tecnologias utilizadas na localização, existem alguns métodos fundamentais que são importantes referenciar. Estes métodos são conhecidos por *lateration*, *proximity sensing*, *angulation*, *dead reckoning* e *pattern matching*.

No posicionamento do tipo *lateration* o dispositivo tem que estar no alcance da diferença entre 3 e 1 *base station* para ser possível efectuar o cálculo da posição. O método de *lateration* é dividido em dois tipos: circular *lateration* e *hyperbolic lateration*. O circular calcula a posição conhecendo a distância entre a *base station* e um ponto r_i (p.e. calculo pela força do sinal RSS), sendo i o número de *base stations* que estamos a considerar. Desenha-se assim uma circunferência em volta de cada *base station* à distância r , e calcula-se a posição recorrendo ao teorema de Pitágoras. No *hyperbolic lateration* o cálculo da posição é efectuado através de uma hipérbole e não por uma circunferência. Assim, não utiliza um valor absoluto para a distância mas sim uma diferença de distâncias, sendo necessário ter pelo menos 2 *base stations* para efectuar o cálculo.

Outro método, *proximity sensing* obtém a posição através do limite do modo de transmissão, ou seja, pelas ondas de rádio, por infravermelhos, por *bluetooth*, etc. O cálculo da posição é obtido derivando a posição da *base station*, que recebeu um sinal piloto¹ vindo do dispositivo.

Angulation é um método diferente dos anteriores, pois efectua o cálculo da posição com base no ângulo entre as diferentes *bases stations*. Para ser possível utilizar este método é necessário que as *base stations* estejam equipadas com *arrays* de antenas.

Um dos métodos mais antigos provenientes da navegação marítima é o *dead reckoning*. Este método calcula a posição de um objecto através da última posição conhecida e da direcção e velocidade do objecto. Podemos ver a utilização deste método, por exemplo, nos sistemas de navegação dos carros quando deixam de captar os satélites *GPS*.

No último método *pattern matching*, a posição é calculada com base em comparações de padrões, existindo dois métodos, um óptico e outro não óptico. No método óptico antes do cálculo da posição são tiradas fotografias da cena e depois são analisadas para chegar à posição do objecto. No meio não óptico são tidas outras propriedades como, por exemplo, as ondas radio para tirar um *fingerprint* do meio e depois comparado.

Após a descrição de alguns métodos de localização fundamentais, seguidamente, iremos mostrar algumas tecnologias que podem ser usadas para localização *indoor* e outras para localização *outdoor*.

¹ Sinal recebido em telecomunicações para supervisão, controlo ou sincronização.

3.2.1. Localização *Indoor*

Com a crescente necessidade de localização no interior dos edifícios, começam a aparecer alguns métodos de posicionamento *indoor*. Mais à frente, iremos apresentar alguns exemplos destes métodos de localização, começando pela tecnologia *wlan* e finalizando com a tecnologia de infra-vermelhos.

Uma das tecnologias cada vez mais promissora no interior dos edifícios é a *wlan*, que nos dias de hoje, equipa muitos dos dispositivos móveis que são comercializados. Deste modo, esta tecnologia possibilita um posicionamento *indoor*, sem custos adicionais de *hardware*.

O posicionamento *indoor* realizado por esta tecnologia é então assim constituído por *APs* (*Access Points*), que cobrem uma área chamada *BSA* (*Basic Service Area*) e servem um conjunto de terminais *BSS* (*Basic Service Set*). Por vezes, estes terminais encontram-se ligados entre si através de uma estrutura cablada, formando um *Extended Service Set*.

Para efectuar as observações do posicionamento são utilizadas os valores do *rss*, *snr* e *proximity sensing*. Considerando o pouco alcance desta tecnologia e a dificuldade de sincronização dos vários relógios, é impossível utilizar medições baseadas no tempo. Os valores do *rss* e *snr* são calculados através dos *beacons* enviados tanto por *uplink* (envio por parte do terminal de um *beacon* para os *aps*) ou por *downlink* (através da utilização da funcionalidade *passive scanning*).

Desta forma, os três métodos que podem ser utilizados para o posicionamento *indoor*, através desta tecnologia são: *proximity sensing*, *lateration* e *fingerprinting*.

No primeiro método *proximity sensing*, a posição do terminal é relacionada com a posição do *ap* que tem melhor qualidade de sinal, através de um mapeamento do *ap* com a sua localização. Este tipo de posicionamento é assim o mais simples, no entanto, sofre de perda de precisão.

Ao contrário do anterior, no método *lateration*, todos os *aps* necessitam de estar alinhados com o edifício, para ser possível efectuar o cálculo, feito através da derivada do valor do *path loss*. No entanto, um dos problemas deste método é a possibilidade de existir *multipath* em ambientes *indoor* (se não existir linha de visão), que origina uma perda de precisão do método.

Como podemos ver os métodos mais utilizados na localização *outdoor*, sofrem de alguns problemas na localização *indoor*. Assim, surge um novo método que tem convencido cada vez mais, designado por *fingerprinting*.

O *wlan fingerprinting* é um método constituído por duas fases: *online* e *offline*. Na fase *online*, identificada por uma fase de calibração e referenciação, são guardados os valores *rss* de uma zona num mapa, com diferentes orientações. Na fase *offline* os valores guardados no mapa anterior são comparados com o terminal e com isso, calcula-se a localização como podemos ver na Figura 10. Um dos pontos negativos para este tipo de localização, é o *overhead* que é provocado pela construção dos mapas na fase *online* e a sua repetição sempre que a rede se altera. Para resolver esse problema de *overhead*, por vezes são utilizados modelos matemáticos para construir os mapas.

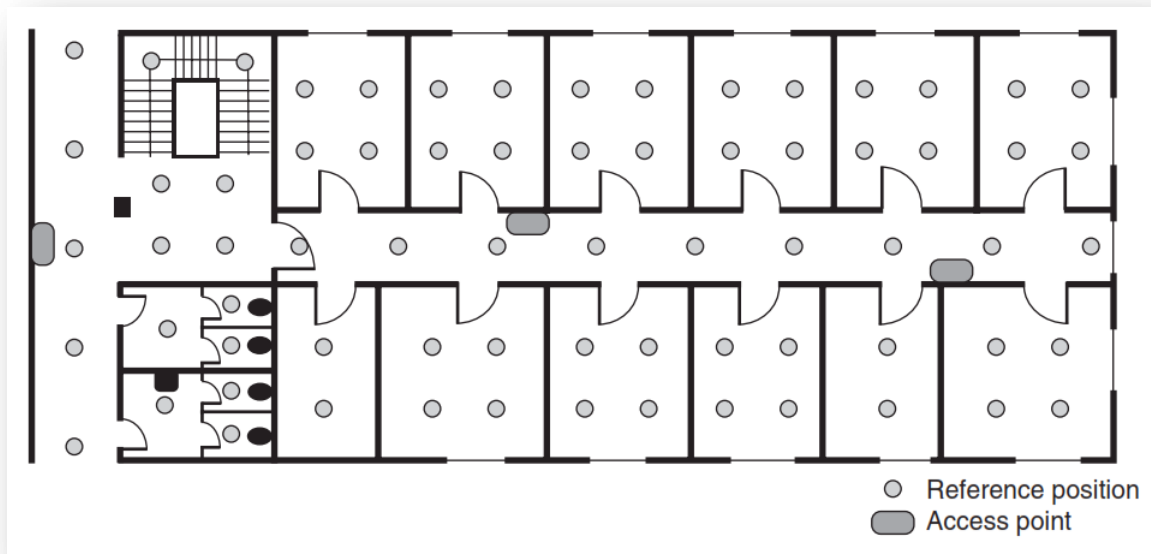


Figura 10 - *Wlan Fingerprinting*.

Retirado de [21].

Outra tecnologia que pode ser utilizada para a localização *indoor* é o *rfid*, mais especificamente, em posicionamento de proximidade. Esta tecnologia é composta por dois tipos de dispositivos (activo e passivo), que possuem alcances diferentes e não utilizam nenhum *beacon*, como a tecnologia anterior. O posicionamento é então efectuado através de qualquer informação que é transmitida entre o leitor e a *tag*.

Para a distribuição deste método em edifícios, são utilizadas topologias de rede e topologia de terminais. Na topologia de rede, os leitores são, por exemplo, colocados num corredor, ligados a um servidor e quando o utilizador passa por estes leitores (transportando uma *tag*), estes enviam os dados da *tag* como o *id* do utilizador e a sua posição para o servidor. De outro modo na topologia de terminal, o leitor pode estar integrado num dispositivo móvel e as *tags* estão espalhadas pelo edifício, permitindo que o leitor passe pelas diferentes *tags* e elas enviam os dados de localização para o dispositivo.

Este último método de topologia de terminal é o mais utilizado, devido à grande diferença entre os preços dos leitores que *rfid* (altos) e os preços das *tags* (baixos).

Outro tipo de tecnologia que também pode ser utilizada na localização *indoor* é a tecnologia de infravermelhos.

Os infravermelhos têm como vantagem o seu curto alcance dentro dos edifícios (alguns metros) e a particularidade de não penetrarem em paredes, permitindo assim restringir por exemplo, um por cada divisão, sem afectar a divisão vizinha. A desvantagem neste método consiste na necessidade da linha de visão entre o receptor e o emissor. Consequentemente, o método de localização utilizado é o de proximidade, à semelhança do modelo anterior.

Uma possível implementação desta tecnologia, seria através da disposição de vários aparelhos receptores em diferentes divisões, ligados a um servidor. Assim, o utilizador transportaria um emissor que enviaria o seu *ID* aos receptores onde estes iriam, por sua vez,

enviar os dados enviados pelo emissor ao servidor central, que faria no final o *matching* e calculava a posição.

Um ponto forte neste tipo de soluções é a eficiência energética, onde uma bateria consegue ter uma duração de aproximadamente um ano.

Posso concluir que apesar dos métodos aqui apresentados, esta é uma área actualmente em constante desenvolvimento, onde todos os dias se criam novos métodos juntamente com novas tecnologias. Das tecnologias aqui apresentadas, a meu ver, a que é mais promissora será a de localização através de *wlan*, devido à grande existência de terminais, em praticamente todos os edifícios.

3.2.2. Localização *Outdoor*

Actualmente os métodos de localização *outdoor* mais utilizados são os que utilizam satélites e antenas de redes móveis como o *GPS* e *GSM (Global System for Mobile Communications)*, respectivamente. Podemos encontrar estes sistemas pelas mais diversas áreas e em diferentes dispositivos como telemóveis, sistemas de navegação, sistemas de segurança, etc.

Satélite

A localização através de satélites é feita principalmente sobre o sistema americano *GPS*. O sistema é constituído por 24 satélites que circulam a cada 12 horas no espaço sideral e possui dois serviços um para civis e outro para militares, o *SPS (Standard Position Service)* e *PPS (Precision Position System)*. O *SPS* utilizado para civis dá informações sobre posição e tempo tendo inicialmente uma precisão de cerca de 100m[20] horizontalmente, o *PPS* dá informações da posição do satélite, velocidade e tempo tendo uma precisão inicialmente de cerca de 16m[20] horizontalmente.

Para efectuar o cálculo da localização, utiliza o método falado anteriormente de *lateration* juntamente com o método *code phase ranging*[21], onde cada terminal tem de medir a distância aos satélites e saber a posição dos satélites no tempo que o cálculo da posição está a ser efectuado. Ambos, são calculados através do sinal piloto transmitido pelos satélites. Primeiro, dois códigos de distância são emitidos para determinar o tempo de viagem, tendo em consideração a velocidade de propagação da luz. Secundariamente, o sinal contém uma mensagem de navegação com dados de assistência como órbita, correcções de relógio, parâmetros do sistema, que permite o receptor calcular a posição do satélite em cada momento.

O *GPS* diferencial ou *DGPS (Differential Gps)* é um mecanismo de localização que utiliza o *GPS* juntamente com um ponto de referência terrestre. Efectua o cálculo da posição através da diferença entre a distância e a posição conhecida, sendo para isso necessário instalar na terra estações de referência.

O sistema galileu é um sistema civil de posicionamento europeu financiado pela Agência Europeia, operado sobre controlo público. Ainda se encontra em fase de lançamento, sendo no futuro constituído por cerca de 27 satélites orbitando em três orbitas, com três satélites extra de reposição em caso de avaria. Na terra irá utilizar entre 15 e 20 estações de referência para a melhoria do cálculo da posição. Em comparação com o *GPS* este irá ter mais dois serviços, ou seja, quatro serviços no total, um serviço livre com cerca de 4 metros de precisão horizontal, um comercial com cerca de 1 metro de precisão horizontal, um público

com cerca de 6,5 metros horizontal e um para serviços de segurança com cerca de 4 metros de precisão horizontal.

Redes Móveis

Na localização através de redes móveis, podemos destacar quatro tipos diferentes de cálculo da localização com base nestas tecnologias. Os principais métodos são: *id* da célula em combinação com o *time advance*, *E-OTD* (*Enhanced Observed Time Difference*), *U-TDOA* (*Uplink Time Difference Arrival*) e *A-GPS* (*Assisted Gps*).

O primeiro método é baseado na técnica falada anteriormente de *proximity sensing*, ou seja, o cálculo da posição é efectuado através da derivação da posição da *base station* em relação à posição do dispositivo móvel. Para os valores serem mais precisos, é utilizado juntamente com o *cell id* o método *time advance*, que consiste no cálculo do tempo que um sinal demora a chegar desde o dispositivo móvel até à *base station*. Como podemos perceber, este tipo de mecanismos são muito úteis porque permitem o cálculo da posição de um dispositivo móvel, sem quaisquer custos.

O método *E-OTD* utiliza a técnica *lateration* falada anteriormente. Ao contrário do método anterior, em que não era necessário a instalação de nenhum componente adicional à rede, neste método é necessário a presença das *LMU* (*Location Measurement Unit*). Como o *E-OTD* foi desenhado para o *GSM* e este não utiliza sinais piloto, é usada a estrutura do *TDMA* (*Time Division Multiple Access*) para efectuar as medições de tempo. Como foi referido este método utiliza *lateration*, logo existem dois modos distintos que podem ser utilizados, sendo um pelo método de circular *lateration* e outro por *hyperbolic lateration*. No primeiro é necessário que os dispositivos móveis, estejam perfeitamente sincronizados com as *base stations*, enquanto no segundo as *base stations* têm que estar sincronizadas entre elas. Para o cálculo da posição os dispositivos móveis é calculado a diferença de tempo de chegada de sinais a partir de duas *base stations* diferentes, estas observações são denominadas de *OTD* (*Observed Time Difference*). Como não existe sincronismo entre as diferentes bases stations, o *LMU* que observa a estrutura *TDM* calcula o *offset* do tempo, chamado de *RTD* (*Real Time Difference*). No final, o dispositivo móvel calcula o *GTD* (*Geometric Time Difference*) e faz o cálculo da posição.

$$GTD = OTD - RTD$$

O *U-TDOA* como o *E-OTD* efectua o cálculo da posição do dispositivo móvel com base na técnica de *hyperbolic lateration*, através da observação da chegada dos dados nos *slots* temporais. Ao contrário do *E-OTD* que o cálculo da posição era efectuado pelo dispositivo móvel, no *U-TDOA* o cálculo é feito pela rede. Para isto ser possível, o dispositivo móvel necessita de estar ocupado. Além disso, é necessário a instalação de *LMU's* adicionais para que estas estejam a escutar o sinal, devido a ser usado *hyperbolic lateration* e ser necessário pelo menos três *base stations*. O processo do cálculo da posição é então o seguinte:

1. O dispositivo móvel necessita de estar em funcionamento;
2. O *SMLC* (*Serving Mobile Location Center*) precisa de descobrir a *base station* e o canal físico, para isso, pergunta ao *BSC* (*Base Station Controller*);
3. Necessita de identificar os *LMU's* nas imediações do dispositivo;
4. Configuração dos *LMU's* com método de medição, a janela temporal, o tempo de início, a duração e o canal;
5. Os *LMU's* escutam os dados que vêm do dispositivo e guardam os tempos de chegada;
6. Os resultados são enviados ao *SMLC* que estima a posição do terminal.

Como este método tem uma confiabilidade muito elevada é utilizado nos EUA como sistema de localização para chamadas de emergência, segurança, proteger estruturas críticas, etc.

O *GPS* é muito utilizado em todo o Mundo, mas nos dispositivos móveis apresenta alguns problemas de fiabilidade, consumo de energia e fraca cobertura *indoor*.

Assim, o *A-GPS* é um sistema que utiliza o *GPS* juntamente com o *DGPS* nas redes móveis, de modo a melhorar a precisão, redução de tempos de aquisição, menos consumo energético e aumenta a sensibilidade da recepção. Este sistema funciona através das redes móveis sendo preciso ter uma estação de referência para áreas com um raio de 200km. No cálculo da posição são efectuados os seguintes passos:

1. A estação de referência irá receber os sinais dos satélites que estão visíveis, efectuando a extracção dos dados presentes na mensagem de sinalização e calcula os dados de correcção *D-GPS*;
2. Em seguida periodicamente ou a pedido, o *SMLC* recebe os dados de suporte ao posicionamento;
3. Quando um dispositivo móvel quer calcular a sua localização, o *SMLC* envia-lhe os dados recebidos pela estação, contento os dados de assistência e de correcção;
4. O dispositivo utiliza assim os dados recebidos de modo a identificar os satélites que estão visíveis facilmente;
5. Dado o posicionamento poder ser efectuado pela rede ou pelo dispositivo móvel, os dados são enviados para o *SMLC* ou calculados pelo dispositivo, respectivamente.

Dos mecanismos apresentados neste capítulo de localização apenas dois serão utilizados nos dois sistemas da plataforma *SocialSense*.

Relativamente ao projecto Avós e Netos(as) o investigador André Pinto criou um mecanismo de localização *indoor* baseado em tri-lateração através do *RSSI* (*Received Signal Strength Indicator*). A tecnologia utilizada foram as redes de sensores sem fios, que através de pelo menos três nós sensores efectua o cálculo da posição de uma criança num ambiente *indoor*.

Finalmente, no projecto *Youth5G* para calcular o “local” onde utilizador se encontra será utilizado o *gps* do *smartphone* para efectuar o cálculo da posição.

3.3. Mecanismos de Intermitência

Nos últimos anos, temos visto uma grande evolução nas redes móveis, utilizando protocolos e tecnologias cada vez mais eficazes. Além disso, têm crescido de uma forma exponencial sistemas que utilizam estas redes para partilhar dados. Muitos destes sistemas têm que fornecer os dados com precisão e sem comprometer a eficiência energética dos dispositivos, encontrando-se muitas vezes em movimento.

No entanto, estes sistemas são muitas vezes susceptíveis a falhas na comunicação, devido ao congestionamento, áreas sem cobertura, perdas de sinal, etc. Neste tipo de problemas, por vezes é necessário recorrer a mecanismos de intermitência.

Este capítulo apresenta alguns mecanismos de intermitência que podem ser usados, de modo, a contornar a falta de ligação entre dispositivos móveis. Em primeiro lugar, o foco serão as redes *DTN* (*Delay Tolerant Network*), seguindo-se de outras técnicas como previsão através de cadeias de *Markov*. Embora estes mecanismos ainda não tenham sido implementados nas plataformas da dissertação, também participei na elaboração dos futuros modelos que serão integrados.

3.3.1. Redes *DTN*

Uma abordagem muito interessante neste campo são as redes *DTN*. Enquanto nas redes *MANET* (*Mobile Ad Hoc Network*) torna-se necessário que os nós móveis tenham sempre ligação, ou seja, tenham possibilidade de construir um caminho do nó de origem ao nó de destino, nas redes *DTN* isto nem sempre é possível.

Nas redes *MANET* com protocolos como o *OLSR* (*Optimized Link State Routing Protocol*) e o *AODV* (*Ad hoc On-Demand Distance Vector*), é sempre necessário conhecer a topologia da rede para garantir a entrega dos pacotes, apesar do *OLSR* ser proactivo e o *AODV* reactivo. No entanto, as redes *ad-hoc* físicas nem sempre conseguem cumprir este tipo de requisito, ou seja, nem sempre garantem que no mesmo instante todos os nós estão activos (seja por questões energéticas, seja pelo alcance da ligação) consigam construir um caminho do ponto de origem até ao ponto de destino.

Assim, as redes *DTN* permitem uma abordagem diferente, pois é possível que dois nós comuniquem sem pertencer à mesma rede. Para que isto seja possível, os nós têm que colocar em *buffer* os pacotes, quando não for possível fazer a entrega ou determinar o caminho.

Como podemos observar na Figura 11 não existe no tempo 8:00am, um caminho directo entre o nó de origem e o nó de destino. Se o encaminhamento fosse através dos protocolos falados anteriormente como o *AODV* e *OLSR*, os pacotes eram descartados, devido à necessidade de estes precisarem construir um caminho do ponto de origem S até ao ponto de destino D. Como neste tempo, este caminho não pode ser traçado, estes protocolos não conseguiriam entregar os pacotes.

Pelo contrário, nas redes *DTN* o encaminhamento dos pacotes entre S e D já seria possível, porque estas redes dispõem de mecanismos que quando um nó não consegue entregar ao próximo nó, ou não conhece o caminho de S a D, o nó coloca os pacotes no seu *buffer* até que surja uma oportunidade de entrega como podemos ver nos tempos seguintes 10:10pm e 11:00am.

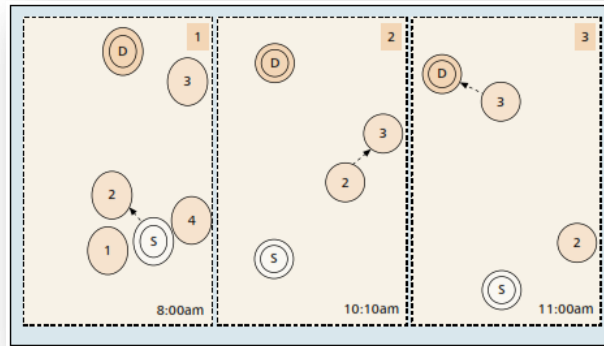


Figura 11 - Rede DTN.
Retirado de [22].

Ter um encaminhamento dos dados deste tipo, levanta alguns problemas com os quais não estamos habituados a estar confrontados como, por exemplo, o tempo de entrega dos pacotes. Consequentemente, as redes DTN não são capazes de ter uma conectividade constante, poucas perdas de dados e um baixo *delay*. Não quer isto dizer, que apesar de existirem muitos serviços e aplicações que não possam suportar este tipo de requisitos, que não existem outros que possam como: redes de sensores sem fios que quando não dispõem de energia, entram em modo *sleep* quebrando a conectividade, e redes para fins militares com grande mobilidade, onde nem sempre à conectividade entre soldados.

Apesar de os protocolos de encaminhamento mais utilizados como o *TCP* (*Transmission Control Protocol*) e o *UDP* (*User Datagram Protocol*) não serem adequados para este tipo de redes, os autores do artigo [22] apresentam uma grande quantidade de protocolos existentes para este tipo de redes e defendem que a escolha do protocolo, deve ser feito consoante o serviço/aplicação.

3.3.2. Cadeias de *Markov*

No artigo [23] os autores apresentam algumas técnicas para o envio de grandes quantidades de dados, sem comprometer a eficiência energética dos dispositivos e a precisão dos dados. Adicionalmente, apresentam alguns mecanismos como previsão de estados com base em localização quando a ligação não está disponível.

A *testbed* dos autores foi constituído por um conjunto de *smartphones* Nokia N95, que comunicam directamente para um servidor *back-end* e este comunica os dados para as redes sociais.

Nos mecanismos de intermitência, eles defendem que uma das possíveis estratégias que podem ser adoptadas prende-se com a criação de um previsor, que prevê o próximo estado a partir do actual. Este tipo de mecanismo pode ser aceite em alguns sistemas, mas em outros sistemas pode não ser aceite, devido à incerteza que têm.

Para fazer esta previsão inicialmente, eles utilizam matrizes de transição de modo a modelar os estados dos utilizadores e analisam estas matrizes através de cadeias de *Markov*. Estas matrizes guardam a probabilidade de transição entre os diferentes estados, sendo as probabilidades calculadas através da frequência com que os estados ocorrem. O cálculo das probabilidades é feito pelos *smartphones*.

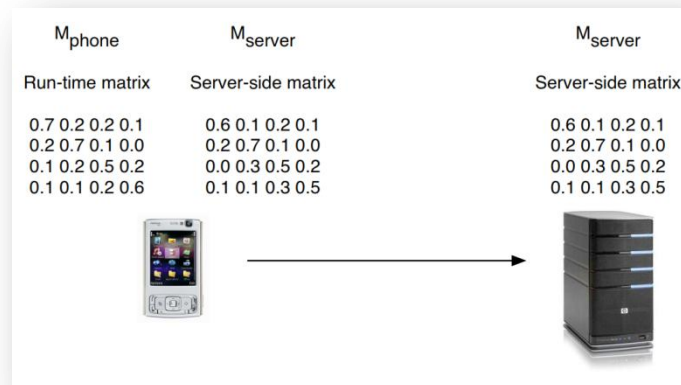


Figura 12 - Matrizes de Previsão
Retirado de [23].

Assim, em vez do telemóvel fazer o *upload* de um único valor, este passa a fazer o *upload* de uma matriz. Existem duas matrizes que são guardadas nos telemóveis, a primeira matriz é constantemente actualizada com novos dados enquanto a segunda matriz é uma cópia da matriz que se encontra nos servidores *back-end*. O envio da matriz para o servidor *back-end* é feito na inicialização do sistema ou por um tempo definido.

O grande problema deste mecanismo encontra-se no processo de envio da matriz para o servidor *back-end*, onde é necessário verificar se a primeira matriz é diferente da que se encontra nos servidores. Para calcular a diferença entre as matrizes os autores recorrem ao método da distância euclidiana e da distância de *Manhattan*, actualizando consoante o valor de Th (*Threshold*). Este valor define um máximo da distância calculada, assim a matriz é actualizada, se e só se, o valor da distância for maior ou igual ao limite.

$$Lx(M_{\text{phone}}, M_{\text{server}}) \geq Th$$

Como resultados, os autores do artigo registaram um decréscimo da precisão dos dados com o aumento do valor de Th superior a 1. Assim, utilizando o $Th=1$ eles encontraram um valor médio que permite menos actualizações com perda de pouca precisão. Outro ponto crítico neste modelo é a utilização do Modelo de *Markov*, que quanto mais tempo está desligado pior é a sua eficiência na previsão dos dados.

Até este ponto, todos os resultados foram com base na agregação dos dados de todos os utilizadores e não de um utilizador específico. Assim, eles apresentaram um método que relaciona a entropia² de um conjunto de dados com a precisão dos dados, ou seja, disponibiliza um nível de previsibilidade que pode ser usado para classificar o utilizador.

Por fim, os autores propuseram um último método associado a localizações. A ideia deste método é associar as matrizes de transição a um lugar específico e deste modo, prever com base na localização para onde os utilizadores se dirigem. Para efectuar este modelo de movimento de uma pessoa a andar num caminho, eles definiram que a pessoa só podia movimentar-se para caminhos adjacentes a ele. Para isto, é necessário representar os

² Permite formular uma teoria permitindo a compreensão das características gerais de um sistema parcial ou incompleto. Ou seja, permite atribuir valores probabilísticos quando algumas quantidades globais são conhecidas.

caminhos através de uma matriz com as probabilidades de ficar nesse caminho e de sair do caminho. Os grandes obstáculos deste modelo devem-se à impossibilidade de o utilizador saltar dois caminhos.

Enquanto este artigo[23] abordava um modelo onde a previsão dos dados era feito através de uma entidade central, o artigo [24] aborda a previsão do lado dos dispositivos móveis. No entender dos autores, a previsão baseada numa entidade central é problemática quando as bases de dados ficam comprometidas, acidentalmente, maliciosamente ou por outros motivos. Ainda assim, na aplicação deles nunca seria possível ter uma entidade central que consiga controlar todos os pontos de acesso.

Para isto, criaram um sistema *BreadCrumbs* que tem como objectivo acompanhar os movimentos dos utilizadores, e prever constantemente um modelo de mobilidade para ele. Este modelo de mobilidade será depois utilizado para gerar previsões de conectividades *connectivity forecasts* onde o utilizador se pode conectar.

Para a localização dos utilizadores é usada a tecnologia *PlaceLab* que fornece a localização através de uma base de dados. Significa isto que, as coordenadas em latitude e longitude são fornecidas com base no *MAC* do ponto de acesso, permitindo saber a posição do utilizador sem utilizar por exemplo *GPS* e deste modo poder prever os próximos “*connectivity forecasts*”.

Como no exemplo anterior, também foi utilizado um modelo de *Markov* de segunda ordem, devido a ser um modelo que tem pouco processamento, e permite que cada estado seja representado por duas coordenadas, a coordenada do último estado e a coordenada actual. Para não terem muitas posições com possível representação, foi aumentada a escala de modo a ter um nível menos pormenorizado, mas praticável e dentro do valor médio de erro da tecnologia *PlaceLab* (+/- 30 metros).

Em suma, podemos verificar que existem sistemas que necessitam de estar constantemente a enviar diferentes tipos de dados e por vezes uma das grandes dificuldades destes sistemas é a inexistência de uma ligação.

Neste capítulo encontrámos algumas soluções através de diferentes abordagens que já foram implementadas em sistemas existentes[23].

Este tipo de soluções são interessantes para os projectos *Youth5G* e *Avós e Netos(as)*, pois encaixam-se no tipo de sistema que adquire dados em tempo real e que necessita de estar constantemente a enviar dados, através de uma ligação que nem sempre pode estar disponível. Assim, aplicando este tipo de mecanismos conseguimos contornar alguns desses problemas.

3.4. Privacidade

O termo privacidade é um dos componentes essenciais da contemporaneidade, presente nos direitos dos cidadãos como forma de existir na sociedade de um modo anónimo.

Com o grande aparecimento de dispositivos móveis e redes de sensores sem fios, começam a surgir muitas aplicações que partilham os dados destas tecnologias nas redes sociais. No entanto, essa partilha de informação, se não for controlada com os mecanismos correctos, pode ser feita de um modo inseguro e pode vir a comprometer a segurança e privacidade do utilizador.

Nos últimos anos, houve um crescimento exponencial de aplicações que tentaram tirar partido das grandes quantidades de dados disponíveis em diferentes fontes. Porém, estas aplicações esquecem certos requisitos de privacidade essenciais como nos sistemas *WhozThat*[18] e *Serendipity*[19], colocando em causa a segurança do utilizador. Não só, algumas das redes sociais mais importantes como o *Facebook*[25] apesar de fornecerem mecanismos para assegurar a privacidade dos utilizadores, por vezes, não é suficiente.

Dependendo de cada sistema, cada aplicação deve ter diferentes abordagens relativamente à privacidade. Nos *mobile network systems* são definidos dois tipos de rede: sistemas que usam *peer to peer* (*WhozThat*[18]) e sistemas cliente-servidor (*SocialAware*[26]). Nos primeiros sistemas (*peer to peer*) as trocas de informações das redes sociais são efectuadas de dispositivo para dispositivo por *Bluetooth* ou *WiFi*, enquanto nos sistemas cliente-servidor existe uma entidade central para onde os dados dos dispositivos são enviados.

Como existem diferentes tipos de redes, existem também diferentes falhas de segurança e privacidade associados a cada tipo, sendo algumas comuns. Os autores[27] identificaram assim algumas falhas e dividiram-nas em três grupos. O primeiro grupo refere-se à identificação directa do utilizador, o segundo com a identificação indirecta e o terceiro com possíveis ataques como *eavesdropping*³, *spoofing*⁴, *replay*⁵ e *wormhole attacks*⁶.

Este capítulo irá apresentar algumas das falhas mencionadas, baseando-se em alguns projectos referidos anteriormente, mostrando que apesar das redes sociais terem fortes mecanismos de privacidade estes falham e mostram um modo alternativo de tratar a privacidade associada ao ambiente.

3.4.1. Identificação Directa do Utilizador

Este grupo de falhas refere-se a situações em que o utilizador pode ser identificado através de um identificador que de alguma forma compromete a privacidade do utilizador.

Algumas destas falhas estão presentes em sistemas como o *WhozThat*[18] ou no *SocialAware* [26], onde é necessário que seja transmitido entre os dispositivos *peer-to-peer* o *ID* da rede social para se poder partilhar os dados na rede. Se o envio do *ID* for em *cleartext*, sem nenhum tipo de encriptação, este sistema está susceptível a ataques do tipo *eavesdropping*, *spoofing*, *replay* e *wormhole attacks*).

Este tipo de ataques podem ser realizados, devido à captura do *ID* da rede social permitindo que um atacante faça-se passar pelo utilizador.

³ Técnica que consiste em escutar de modo secreto uma conversa privada sem o consentimento da pessoa.

⁴ Técnica que altera o pacote IP de modo a falsificar o remetente.

⁵ Um ataque que consiste em interceptar os dados na rede e depois retransmiti-los.

⁶ Este tipo de ataque consiste em interceptar os dados numa ponta da rede e retransmiti-los na outra ponta.

Uma forma de evitar este tipo de ataques é através da encriptação dos dados a partir de uma ligação segura *https* (garante confidencialidade e integridade), utilizar mecanismos de autenticação e utilizar certificados (garante autenticação e não repúdio). No entanto, existem sistemas em que não é possível o uso destas soluções, em grande parte devido ao peso computacional dos métodos de encriptação.

Existe outro tipo de soluções para combater estes problemas de privacidade e segurança. Os autores[27] criaram um sistema chamado *IS (Identity Server)*, para tentar combater os problemas relacionados com o anonimato. Sendo o envio da informação em *cleartext*, eles criam um *AID (Anonymous Identifier)*. Este identificador é gerado sempre que um dos dispositivos queira realizar uma comunicação e em vez de utilizar o identificador da rede social, utiliza este identificador. O *AID* é gerado pelo *IS* através do método de cifragem *SHA-1*, associando assim este *AID* ao dispositivo. O *IS* permite ainda que cada dispositivo seja identificado por vários *AID*, possibilitando que diferentes utilizadores possam obter informação. A utilização do *AID* garante privacidade na troca da informação dado que, os dispositivos não conseguem relacionar o *ID* da rede social com o *AID*. Além disso, a comunicação fica prevenida contra ataques de *spoofing* e *replay*, uma vez que, que os utilizadores já não comunicam através do *ID* da rede social mas sim, através do *AIS* que está em constante renovação.

Por fim, outra solução semelhante está presente no artigo *SmokeScreen*[28], onde os autores criaram um protocolo que emite identificadores em modo *broadcast*, para partilharem as informações pessoais através deles.

3.4.2. Identificação Indirecta do Utilizador

Quando abordam as questões relativas ao segundo grupo, ou seja, a identificação indirecta do utilizador, os autores[27] referem-se à possível identificação através de um conjunto de dados. Assim, apesar de o utilizador não ser identificado directamente este pode ser na mesma identificado indirectamente, como por exemplo através dos seus passatempos favoritos, locais preferidos, etc.

Name	Color	Letter	Number
Bill	Red	A	1
Fred	Green	A	2
Jon	Green	B	2
Joe	Red	C	1

Tabela 1 - Conjunto de Dados

Retirado de [27].

Como podemos ver na tabela em cima, é possível relacionar os utilizadores com os seus dados. Para resolver esta situação os autores estão a estudar a possibilidade de utilizar algoritmos de simplificação lógica como o algoritmo *Quine-McCluskey*[29].

Além disso, em sistemas como o *SocialAware*[26] é também possível através do *log* da data ou da localização construir um histórico e comprometer a privacidade do utilizador. Quando a comunicação é cliente-servidor este tipo de ataques são também possíveis, dado que os

sistemas muitas da vezes utilizam os nomes dos utilizadores e guardam as suas localizações, possibilitando, por isso, uma comparação de modo a identificar o utilizador e a comprometer a sua privacidade.

3.4.3. Redes Sociais

Algumas das redes sociais permitem o tipo de mecanismos falados anteriormente, como é o caso do *Facebook*, através do uso de certificados e ligações *https* (*HyperText Transfer Protocol Secure*), protegendo deste modo os dados entre a aplicação - servidor ou servidor - dispositivos.

Neste pequeno capítulo serão abordados outro tipo de problemas de privacidade referentes ao *Facebook* devido a esta ser a rede social utilizada pelo Avós e Netos(as).

Actualmente, o *Facebook* é uma das mais importantes redes sociais, e dispõe de uma quantidade enorme de utilizadores. Logo, tantos utilizadores, e tanta informação levantam grandes preocupações a nível de privacidade. Os utilizadores partilham muitos tipos de informação, desde a sua localização, o que fazem, o que gostam, a sua agenda, etc.

Esta informação sem os mecanismos correctos pode levantar grandes problemas, como por exemplo um utilizador partilhar a sua localização e dizer que irá sair de casa para ir de férias. Se não tivermos em atenção como a informação é partilhada pela rede social e de que maneira ela está exposta publicamente podia ser facilmente utilizada para cometer crimes.

Apesar do *Facebook* ter alguns mecanismos de segurança e de nós escolhermos com quem partilhamos a nossa informação isso, por vezes, não é suficiente. Nem sempre as pessoas têm a noção de como a informação é partilhada e não são avisados disso. Isto resulta um pouco da ideologia das redes sociais e dos próprios criadores [30].

Recorrendo ao exemplo anterior, podíamos por exemplo comentar a informação que íamos estar fora de casa com um amigo numa foto, e ele de seguida fazer uma *tag* para o álbum dele. Deste modo os amigos dele também já teriam essa informação.

Observando estas questões de falta de privacidade é importante criar alguns mecanismos pessoais como por exemplo, manter anonimato para uma profissão. Para manter este anonimato podemos seguir certas regras como não partilhar fotografias que mostrem a profissão ou escolher com quem a foto pode ser partilhada.

O livro [25] mostra também que apesar de o *Facebook* ter alguns mecanismos, e que estes podem ser configurados no perfil do utilizador, estes apresentam algumas falhas como:

- Quando um utilizador envia uma mensagem a outro utilizador, ele fica com acesso limitado e pode visualizar alguns dados do perfil;
- Quando um amigo responde na nossa barra de estados a qualquer mensagem que tenha sido colocada, os seus amigos podem observar essa actualização.

Em resumo, podemos concluir que apesar do *Facebook* ser uma grande rede social, existem alguns problemas graves de privacidade e que cabe ao utilizador ser prudente nos conteúdos que disponibiliza.

3.4.4. Privacidade Associada ao Ambiente

Grande parte das aplicações e sistemas apresentados anteriormente tratam a privacidade quase sempre do ponto de vista do utilizador, serviços e dispositivos. Para eles o utilizador está no centro e ele é a entidade máxima que decide o que quer ou não partilhar, ditando as regras, independentemente do ambiente que se encontra.

Segundo o autor [31] muitas das vezes isso é verdade mas existem outras situações, que dependendo do ambiente, deve ser este a ditar as suas próprias regras. Assim, o autor defende que a privacidade deveria também ser associada ao ambiente em que o utilizador se encontra, sendo que, cada ambiente deve definir as suas próprias políticas de privacidade.

Exemplos destes casos são por exemplo igrejas, bibliotecas, etc que possuem regras próprias ao ambiente que ditam a privacidade do utilizador.

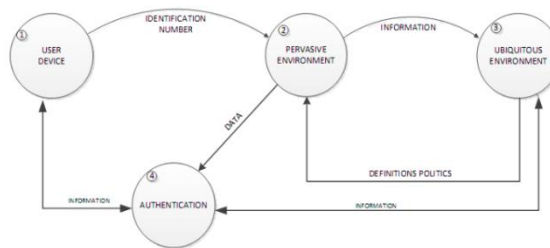


Figura 13 – Modelo Genérico de Privacidade
Retirado de [31].

Logo, o modelo proposto define que cada ambiente deve possuir um conjunto de serviços e que esses serviços devem conter um perfil máximo de “autorização”. Para isso, a taxonomia propõe que os utilizadores devem ter diferentes graus de autorização, consoante o ambiente. Os perfis são definidos por:

- *Bloqued*: Utilizador que tem o seu acesso bloqueado;
- *Guest*: Acesso controlado e limitado;
- *Basic*: Acesso controlado, mas com possibilidade de utilizar alguns serviços;
- *Advanced*: Acesso a todos os níveis anteriores;
- *Administrativ*: Acesso a todos os níveis anteriores e gestão do ambiente.

Como conclusão, o tema de privacidade está nos dias de hoje cada vez mais presente no nosso quotidiano. Fundamentalmente, pela entrada das tecnologias no nosso dia-a-dia em redes como o *Facebook*. Na implementação de sistemas como os desenvolvidos é necessário ter muito cuidado para que não se comprometa a privacidade dos utilizadores.

Tendo em vista os problemas que foram discutidos neste capítulo, no desenvolvimento dos dois projectos (*Youth5G* e *Avós e Netos(as)*) tivemos presentes algumas das soluções aqui apresentadas e até implementamos algumas delas, como o modelo de privacidade associado ao ambiente e regras de segurança que não permitam que o *ID* da rede social navegue em *cleartext* entre o cliente e servidor.

Capítulo 4

Objectivos da Investigação e Método de Abordagem

O grande crescimento das redes sociais originou o interesse de milhões de utilizadores na partilha de dados pessoais, interesses e opiniões. Também os grandes avanços nas tecnologias permitiram que os equipamentos se tornassem cada vez mais baratos e pequenos como redes de sensores, *smartphones*, *smartshirts*, bem como aparelhos de monitorização médica - medidores de tensão, ritmo cardíaco, etc.

Este conjunto de tecnologias e aplicações sociais até hoje têm estado divididas nos seus dois mundos, com excepção de alguns projectos como os referidos no Capítulo 3, que permitem a ligação de uma ou outra tecnologia às redes sociais.

Devido a esta falta de ligação, este estágio teve como objectivo a construção de uma *gateway* híbrida (plataforma *SocialSense*) que permita a ligação entre estes dois mundos. Pretendeu-se ainda que este modelo possibilitasse a construção de duas aplicações em outras áreas do saber como as Ciências Sociais e Comportamentais e empresas comerciais como a *Sapo*.

4.1. Características e Requisitos da Plataforma *SocialSense*

Dadas as diferentes possibilidades de aplicabilidade da nossa plataforma, neste projecto iremos focar-nos em dois casos específicos: nas áreas sociais através adopção/apadrinhamento de crianças pelas redes sociais; na área comercial através de um projecto *SapoLabs* que permite a identificação de locais e eventos nocturnos com uma maior afluência.

Um do objectivo do projecto Avós e Netos passa pelo uso das redes sociais para promover a questão da adopção de crianças desfavorecidas à distância, possibilitando-lhes uma melhor qualidade de vida, através do apadrinhamento. Pela mesma razão, gostaríamos de usar a grande disponibilidade que os idosos têm nos seus dias, oferecendo-lhes um meio que possibilita a monitorização dos seus netos. Este sistema permitirá em ambos os casos, a possibilidade de uma interacção com as crianças, sabendo por exemplo, se está a ser bem tratado, o seu estado de saúde, se está a ter uma boa integração no ambiente, isto tudo disponível na aplicação da rede social.

Por outro lado, o projecto *SapoLabs* além de permitir identificar locais com uma maior afluência, permite dirigir publicidade através da aplicação ubíqua e dos dados recolhidos pela aplicação.

A plataforma *SocialSense* têm assim como principais requisitos:

1. Construção do *gateway* que permita a interligação dos dois mundos;
2. Implementação de mecanismos de actividade;
3. Implementação de mecanismos de privacidade que assegurem uma monitorização anónima, garantindo os direitos das duas entidades;
4. Implementar mecanismos de segurança entre as ligações;
5. Implementar mecanismos de localização indoor;

4.2. Planeamento e Método de Trabalho

Para o desenvolvimento das duas aplicações e da plataforma foi utilizado um método de produção de *software* baseado em metodologias ágeis, devido à necessidade de uma sucessiva mutação e evolução da plataforma.

Sendo a plataforma modular e integrar diferentes tecnologias e métodos de outros projectos, é fundamental ser constantemente actualizada para suportar os diferentes mecanismos criados dentro do grupo de investigação. Deste modo, é garantido um desenvolvimento incremental interactivo, implementando diferentes versões da inicial, garantindo simplicidade no processo.

Ao longo do desenvolvimento existiram reuniões semanais (*face-to-face meetings*), com o intuito de conversar sobre o estado da aplicação, partilhar informação entre os diferentes elementos do grupo, decidir direcções a tomar e identificar respectivas soluções. Estas reuniões realizaram-se também com os orientadores do Projecto Sapó *Labs* e parceiros das áreas das Ciências Sociais

O método de trabalho utilizado dentro das metodologias ágeis foi o *SCRUM*, que permitiu que fossem feitos pequenos planeamentos a curto prazo e fossem definidos requisitos semanais em equipa.

Os elementos comprometidos com este projecto foram o Prof. Dr. Jorge Sá Silva, o aluno de doutoramento David Nunes, o aluno de mestrado André Pinto e eu. O Prof. Dr. Jorge Sá Silva foi o *product owner* do processo, assegurando todas as semanas o trabalho da equipa. O aluno David Nunes foi o *scrum master* que ajudou a remover alguns obstáculos na implementação do projecto, assim como, garantiu que o projecto se desenrolasse como previsto. Eu e os restantes elementos fizemos parte da *team*, onde a minha tarefa consistiu no desenvolvimento da plataforma *SocialSense (gateway)*, no desenvolvimento de uma aplicação *Android* e no desenvolvimento de uma aplicação *Facebook*. Foi minha tarefa ainda a integração de diferente módulo como o de localização e actividade desenvolvidos pelos colegas André Pinto e David Nunes.

O processo utilizado, representado pela Figura 14, inicia-se nas reuniões semanais entre os diferentes elementos, onde foram debatidos os requisitos da aplicação a serem implementados (*Product Backlog*).

O tempo médio de desenvolvimento destes requisitos foi de aproximadamente uma semana (*sprint*), onde no final dessa semana (na nova reunião), era feito o ponto de situação do processo e voltando a definir novos requisitos, para a semana seguinte.

Para organização de toda a documentação e gestão das diferentes tarefas foi criada uma plataforma *redmine*, onde cada elemento tinha as suas tarefas definidas. Além disso, para a gestão de todo o código de aplicações e *gateway* foi utilizado a plataforma *SVN*.

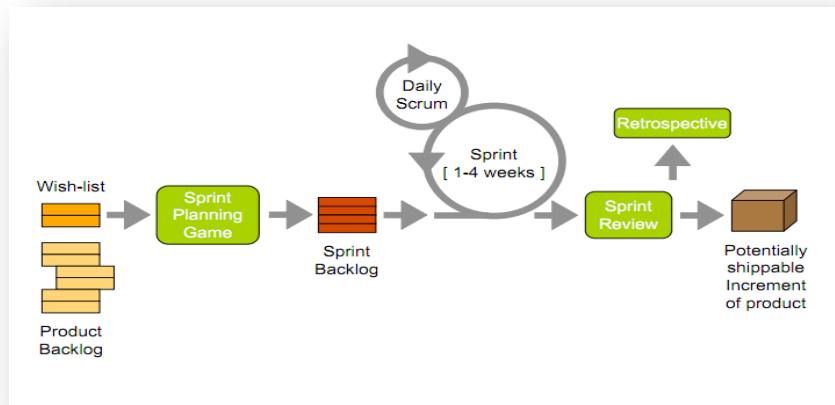


Figura 14 - Método de Trabalho SCRUM.

Retirado de [32].

A Figura 15 apresenta o diagrama de *gant* que mostra as diferentes tarefas efectuadas ao longo deste projecto.

Durante o primeiro semestre iniciei o presente estágio através do estudo e do levantamento das diferentes tecnologias existentes no mercado, que podiam ser úteis para este projecto, como *smartphones*, redes de sensores sem fios, *smartshirts*, etc. Avaliei ainda outros projectos semelhantes existentes. Seguidamente, iniciei o estudo de alguns mecanismos que no futuro irão ser implementados na plataforma como: privacidade, segurança e intermitência.

Posteriormente desenvolvi alguns conhecimentos sobre as diferentes linguagens/*API's* que utilizei na plataforma para os estudos preliminares definido no Capítulo 5. As linguagens foram: *JavaScript*, *Php (Hypertext Preprocessor)*, *Webservices*, *Java* e diferentes *API's (GoogleMaps e Facebook)*. Em simultâneo com esta tarefa ajudei a efectuar a primeira candidatura ao *SapoLabs*.

Em Novembro, com base nas linguagens estudadas, adicionei novas funcionalidades à plataforma *SocialSense*. Para além disso, criei uma pequena aplicação de teste para interacção com redes sociais, que permite a visualização da localização do utilizador através da *API GoogleMaps* e os seus dados pessoais.

Paralelamente participei na escrita do artigo que se encontra no Anexo A, através da aplicação criada nos estudos preliminares.

No final do primeiro semestre, no mês de Dezembro, elaborei o relatório intermédio.

No segundo semestre comecei a criar alguns documentos que especificavam os requisitos que a aplicação *Facebook* devia de ter para o projecto Avós e Netos. Estes requisitos encontram-se disponíveis no Anexo H.

Depois de concluir a especificação da aplicação, entrei numa nova fase do projecto onde desenvolvi a aplicação *Web*, bem como, os módulos correspondentes da plataforma *SocialSense*.

Terminado o desenvolvimento da aplicação *Facebook*, iniciei a integração no sistema do mecanismo de localização *indoor* desenvolvido pelo colega André Pinto e do mecanismo de actividade implementado pelo colega David Nunes.

Finalizado o projecto Avós e Netos, iniciei o projecto *Youth5G*. Seguindo os mesmos passos do projecto Avós e Netos, comecei por elaborar uma especificação com todos os requisitos que o sistema devia de suportar.

Terminada a especificação, iniciei o desenvolvimento da aplicação *Android* e dos novos serviços da plataforma *SocialSense*. Esta etapa teve um tempo de duração de um mês porque foi uma etapa muito exaustiva.

Como em Junho chegou ao laboratório o colega de investigação Valderi Leithardt da Universidade Federal do Rio Grande do Sul, foi decidido implementar o seu novo modelo de privacidade na plataforma, o que demorou cerca de uma semana.

Seguidamente, efectuei os testes que podemos visualizar no Capítulo 8.

Concluído o projecto *Youth5G* participei na elaboração de dois artigos, onde num artigo foi apresentada a plataforma *Youth5G*. No final do mês de Junho ajudei a realizar outra proposta ao concurso *Sapo Labs*.

Finalizei o meu trabalho através da escrita da presente dissertação.

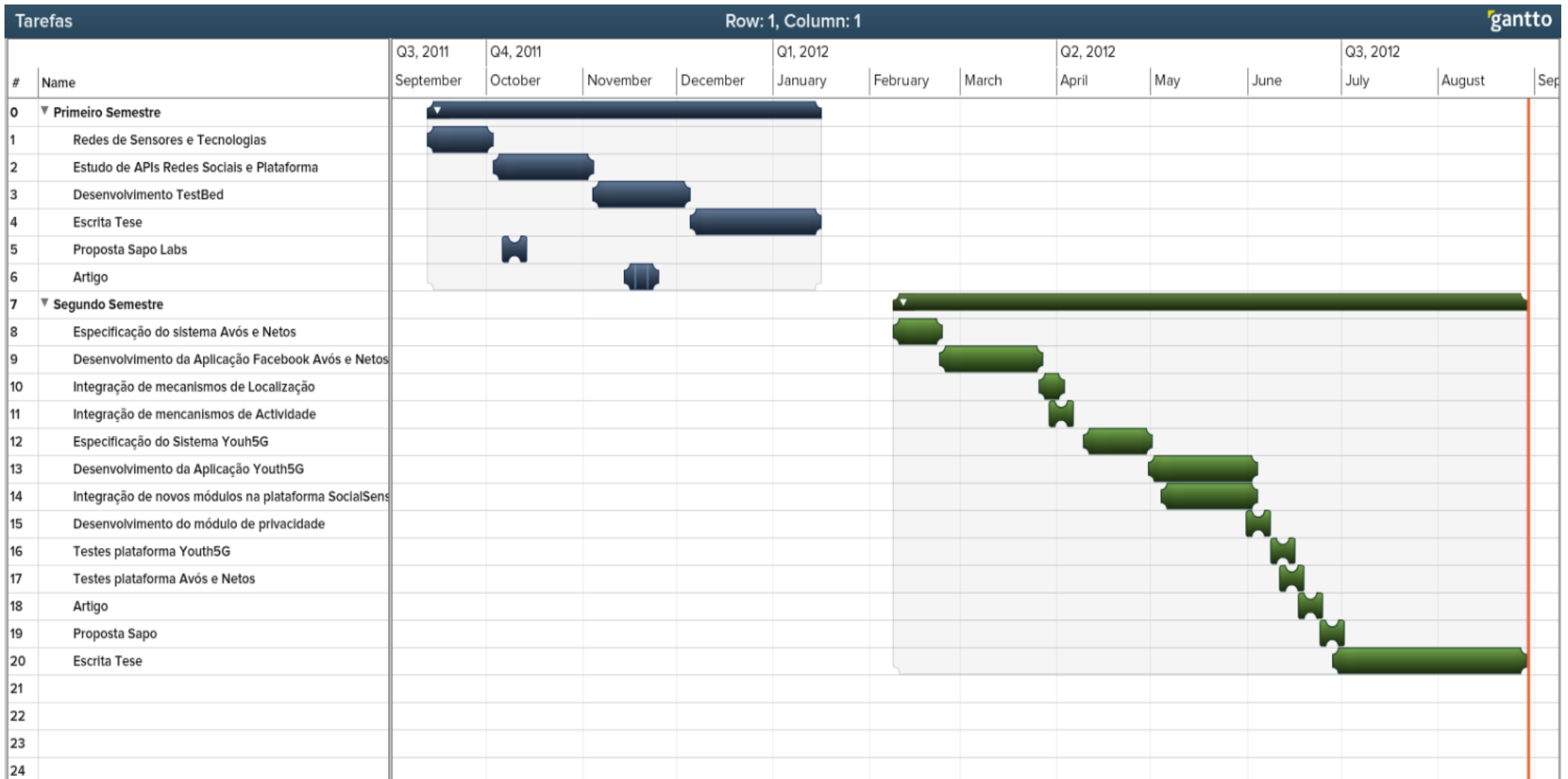


Figura 15 - Diagrama de Gant

Capítulo 5

Estudos Preliminares

Durante o primeiro semestre, apesar de todo o trabalho de leitura e análise (Capítulo 2) sobre as diferentes tecnologias e pesquisa de diferentes trabalhos na área, desenvolvi um pequeno protótipo de uma aplicação, para interacção entre o mundo virtual e o mundo real. Este protótipo serviu para assimilar os conhecimentos necessários, para no segundo semestre desenvolver a aplicação Avós e Netos e *Youth5G*.

Para permitir a introdução destas novas funcionalidades foram adicionados novos componentes, para realizar a integração destes dois mundos. Para uma melhor percepção do trabalho que foi efectuado ao longo deste primeiro semestre, neste capítulo irei apresentar as funcionalidades que foram implementadas.

5.1. Modelo Inicial

O modelo existente da plataforma *SocialSense* assenta sobre um *gateway* híbrido que permite um suporte proactivo e activo das redes de sensores sem fios. Quer isto dizer, que os nós sensores num modo pró-activo têm a capacidade de adquirir dados do meio ambiente e enviar os dados para o *middleware*. No modo reactivo irão reagir a comandos enviados pelo *middleware*.

Para fazer a interacção entre as redes sem fios e o *middleware* são utilizados *proxys* para proporcionar requisitos de escalabilidade, privacidade, segurança e mobilidade. Os *proxys* processam os dados vindos das redes de sensores enviando-os para o *middleware*. Deste modo, conseguimos manter a heterogeneidade das redes de sensores sem fios do lado dos *proxys*, mantendo um padrão de comunicação dos *proxys* para o *middleware* como podemos observar na Figura 16.



Figura 16 - Gateway Híbrido Comunicação

O *gateway* para além de suportar as redes de sensores sem fios, suporta também outros tipos de dispositivos (p.ex. dispositivos médicos). Como estes dispositivos necessitam de um dispositivo intermediário, podem ser utilizado *smartphones* ou computadores para fazer a ligação ao *middleware*.

Assim, o *gateway middleware* funciona como um ponto centralizado (servidor) que permite a ligação entre as diferentes tecnologias como *smartphones*, *proxys*, *smart-shirts* e medidores de tensão com as redes sociais (mundo virtual). De modo a modularizar e dividir estas diferentes tecnologias de aquisição de dados são utilizados *web services*, como por exemplo: *web service* de localização, *web service* de actividade, etc. Os *Restful Web Services* utilizam mensagens em XML (*Extensible Markup Language*) e JSON (*JavaScript Object Notation*) para comunicar.

Depois da chegada dos dados ao *gateway* são tratados, armazenados e enviados para as redes sociais como o *Facebook*.

Em suma, com a criação deste *gateway* híbrido é garantido a interoperabilidade das diferentes tecnologias com as redes sociais.

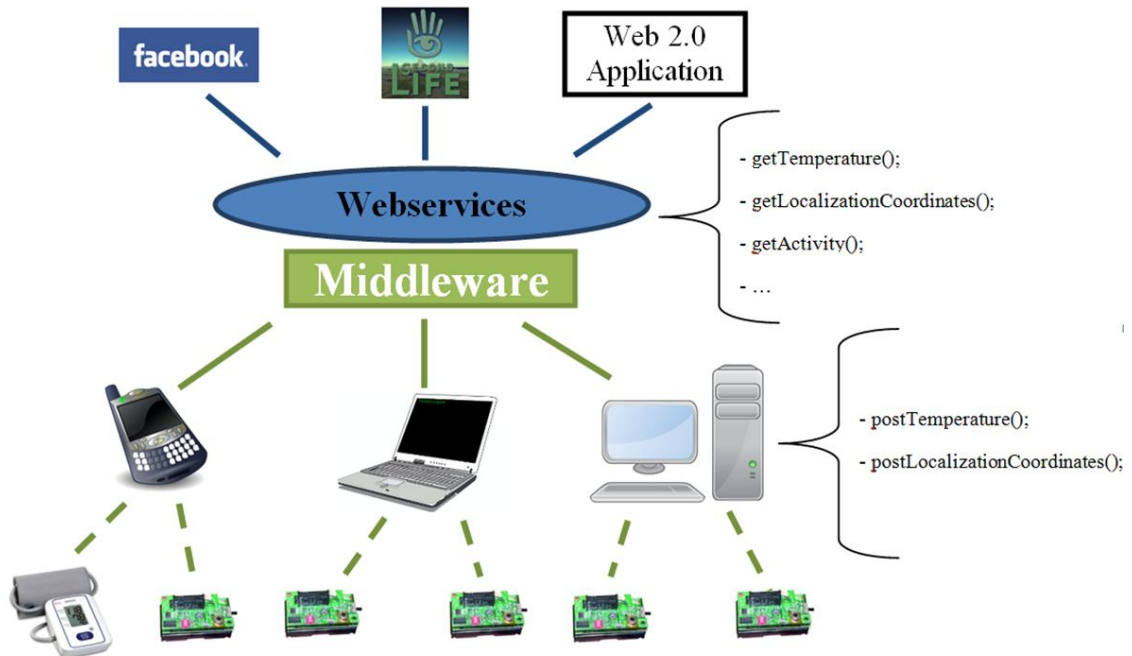


Figura 17 - Tecnologias e Proxys interligados à plataforma *SocialSense*

5.1.1. Arquitectura Inicial

Como foi dito anteriormente o *gateway* é o dispositivo que faz a “ligação”, entre o mundo real e o mundo virtual, redes sociais e aplicações, ou seja, a ligação entre as diferentes tecnologias com as aplicações como podemos ver na Figura 17.

Num momento inicial deste sistema era unicamente possível fazer o envio dos dados dos diferentes sensores para a rede social *Facebook* em forma de publicação. Como identificado no artigo [25], existem problemas graves de privacidade no *Facebook* através deste tipo de publicações. Estes problemas de privacidade serão assim resolvidos através da construção de uma aplicação para a rede social (Capítulo 6).

Como podemos observar na Figura 18, a arquitectura inicial do sistema era constituída por vários módulos. Os módulos que se encontram à esquerda comunicam com as tecnologias, e os módulos que se encontram à direita enviam os dados para as redes sociais (mundos virtuais).

O módulo *Inputs Adapters* recebe os dados provenientes de tecnologias como: *WSN*, medidores de tensão, *smartphones*, etc. Além disso, envia pedidos de controlo para as diferentes tecnologias e aplicações. Na outra ponta da Figura 18, existem diferentes *WebServices* representados por diferentes *URLs* como “rest/localization” e “rest/event” que permitem a interacção das aplicações com o *middleware*.

O núcleo do *gateway* era constituído pelo módulo *Query Manager*, *Services Repository* e *Output Adapters*. O primeiro módulo *Query Manager* tem como principal finalidade fazer a análise e

controlo dos diferentes dados que chegam periodicamente das aplicações. O segundo módulo *Services Repository* informa as diferentes aplicações sobre o conjunto de serviços que estão disponíveis no *gateway*, como por exemplo serviços de localização e actividade. Por último, o módulo *Output Adapters* permite adaptar o formato das diferentes saídas nos *webservices* consoante a aplicação.

Nos serviços que se encontram fora do núcleo do *gateway*, temos o serviço *Localization* e *Semantic Processing*. O primeiro serviço de localização tem como finalidade, poder associar uma coordenada de localização aos dados que vêm dos diferentes nós sensores e deste modo conseguir fazer a representação (p.e. num mapa) de um utilizador. Enquanto, o segundo serviço utiliza a semântica, para por exemplo, auxiliar algoritmos de detecção de actividade.

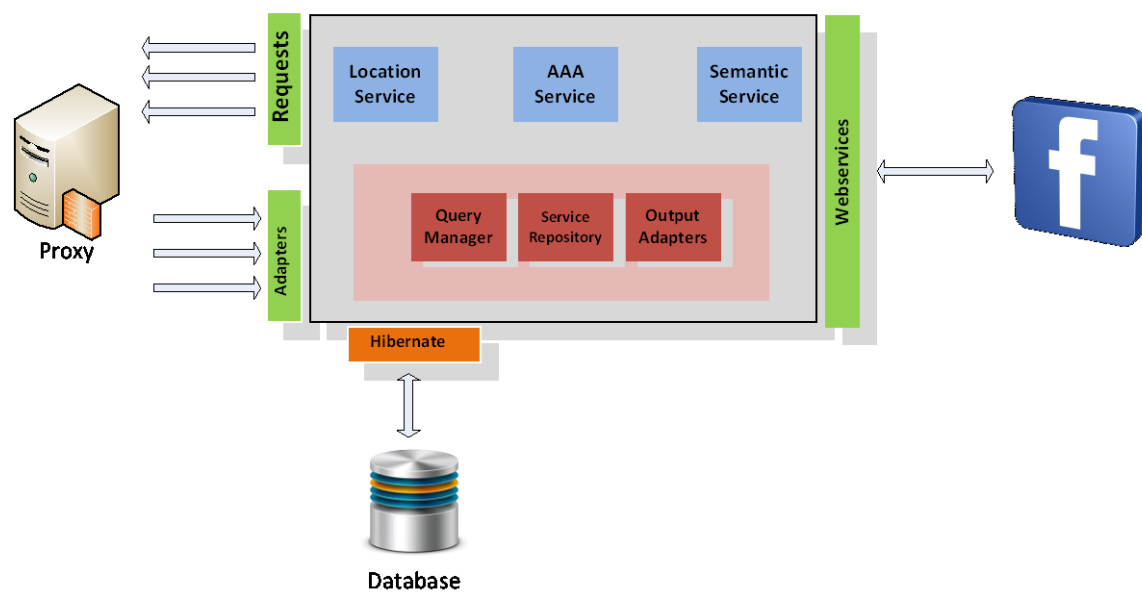


Figura 18 - Arquitectura Inicial *SocialSense*
Adaptado de [38]

5.2. Trabalho Desenvolvido

Como podemos verificar no capítulo anterior, o *middleware* dispunha de alguns serviços úteis que permitiam a troca de informação nos dois sentidos.

Ao longo do primeiro semestre foram adicionadas novos *webservices* e foi criado um primeiro protótipo (Figura 19) de modo a apreender alguns conhecimentos. Neste protótipo não só implementei os *webservices* de actividade, como também criei uma primeira aplicação *Facebook*, que permitisse a representação os dados numa rede social.

5.2.1. Arquitectura Desenvolvida

Na primeira fase de desenvolvimento da aplicação foram criados os seguintes serviços/módulos como representado na Figura 19:

- Desenvolvimento e implementação de um mecanismo que permitisse (*Facebook Publisher*) realizar actualizações periódicas no mural do *Facebook*;
- Desenvolvimento e implementação de alguns métodos de controlo sobre a base de dados;
- Implementação de um *webservice* que permitisse a detecção de actividade;
- Desenvolvimento e implementação de um *webservice* para enviar dados para a aplicação *Facebook*.

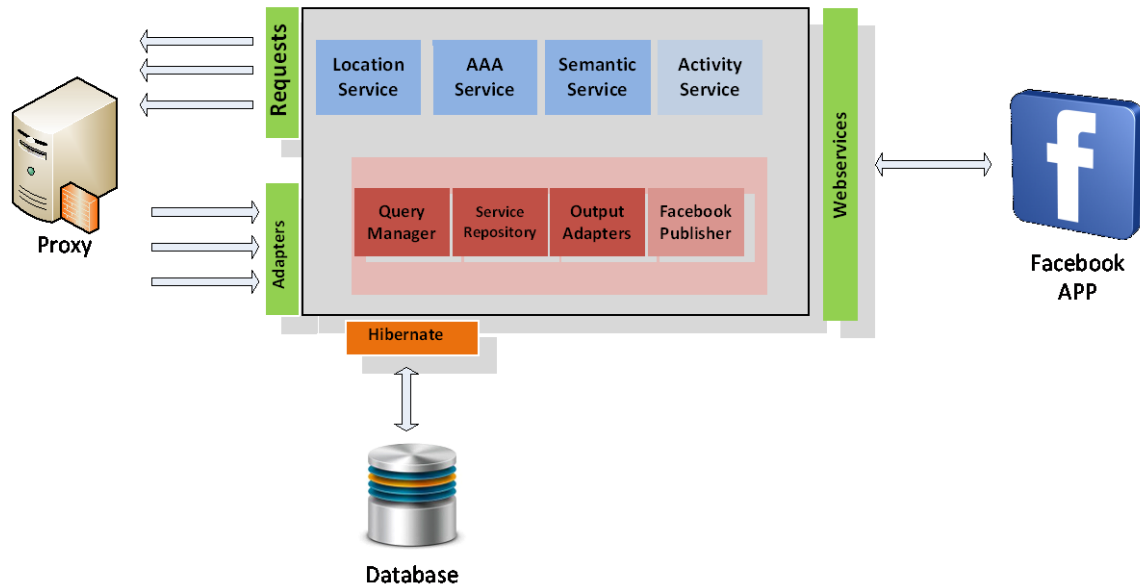


Figura 19 - Módulos desenvolvidos

5.2.2. Facebook Publisher

No modelo inicial, referente à plataforma *SocialSense* não existia nenhum método de envio de dados de forma periódica para a rede social *Facebook*, ou seja, os dados eram enviados no momento que eram recebidos vindos das redes de sensores sem fios. Este modo de envio originava bloqueios por parte do *Facebook*, uma vez que, existem mecanismos contra o envio de *spam*. Estes mecanismos eram accionados quando a nossa aplicação enviava cerca de 30 mensagens num dia.

Para resolver este problema foi criada uma actividade que inicia com o *middleware*, e contém um temporizador para cada utilizador. Posto isto, esta actividade envia publicações com os últimos valores recebidos (Figura 20).



Figura 20 - Publicação Facebook

Dado o *middleware* conter uma base de dados dos utilizadores, foi necessário ao longo do semestre criar algumas funções que permitissem a sua manipulação.

Para efectuar estes testes, foi implementada uma classe que permitiu receber dados das redes de sensores sem fios, sobre a actividade dos utilizadores. Este método permite assim, a identificação, de um modo simples, do movimento do utilizador (por exemplo, se este se encontra parado ou em movimento).

Por último, foi criado um novo *webservice* para permitir a comunicação ente a aplicação do *Facebook* e o *gateway middleware*, através de métodos *post* e *get*. Este novo *webservice* disponibiliza os dados dos utilizadores tais como a actividade, localização e temperatura.

5.2.3. Aplicação Facebook

Depois do desenvolvimento dos novos módulos no *middleware*, foi desenvolvida uma plataforma de teste que permite a representação das diferentes tecnologias nas redes sociais.

Esta comunicação foi implementada através da adição de um *webservice* no *gateway middleware* num novo *url* em "*rest/facebookmaps*". Este *webservice* faz a comunicação com a aplicação do *Facebook* através dos métodos *post* e *get* de *HTTP (HyperText Transfer Protocol)* utilizando *JSON*.

O *webservice* é assim responsável por disponibilizar os dados provenientes das diferentes tecnologias guardadas na base de dados, como localização, dados fisiológicos e actividade do utilizador.

Para representar o utilizador foi utilizado a *API* do *GoogleMaps*, para mostrar a localização e actividade através de um pequeno *avatar* como podemos ver na Figura 21. Adicionalmente, foi criado uma representação *indoor* do edifício, através da sobreposição de uma imagem (*Google Maps* não possui representação de espaços *indoor*). O pequeno *avatar* é representado por uma imagem, que se movimenta se estiver em actividade e fica parado se não estiver em actividade, através da linguagem *HTML5*. Para saber dados sobre o utilizador (como a temperatura), basta pressionar o *avatar* e é apresentado um balão que permite visualizar essa informação.



Figura 21 - Aplicação Facebook

Capítulo 6

Projecto de Avós e Netos(as)

Depois dos estudos preliminares que ocorreram no segundo semestre, surgiu o projecto Avós e Netos, que veio servir como um dos casos de estudo para a plataforma *SocialSense*.

6.1. Descrição do Projecto

O objectivo deste projecto consiste em criar um protótipo funcional de um sistema de monitorização, em tempo real ou desfasado, que permita o acompanhamento à distância de crianças desfavorecidas, com o intuito de apoiar um parceiro da área das Ciências Sociais.

O sistema irá permitir a monitorização das crianças dentro de um espaço comum, por exemplo numa escola ou numa comunidade. Essa monitorização será efectuada através de uma interface *web* que correrá como uma aplicação da rede social *Facebook*. Para realizar esta monitorização será utilizada a tecnologia de redes de sensores sem fios, sobre a forma de pelo menos um sensor sem fios, que mede vários tipos de sinais. A informação proveniente dos sensores sem fios será enviada para um computador na instituição (*proxy*), o qual integra e faz o pré-processamento da informação sensorial e envia os dados para um servidor que corre o *middleware* (Figura 22).

Este *middleware* será responsável pelo armazenamento da informação sensorial, pela informação de cada utilizador do sistema (adultos e crianças) e, pela disponibilização destes dados sob a forma de *Webservices*, que serão acedidos pela interface *Web* do sistema.

A interface *Web* apresentará a informação sob a forma de dados numéricos, mapas e “avatars” que irão representar a localização e actividade da criança. A localização e actividade serão detectadas através de módulos independentes que acedem à informação comunicando directamente com o *middleware* central. Dentro destes módulos, a informação sensorial é processada de forma a detectar eventos de interesse; por exemplo, súbitos aumentos de aceleração podem ser correlacionados com detecção de quedas. Finalmente, a interface *Web* irá permitir também uma interacção de longa distância entre as crianças e os adultos, sobre a forma de videoconferência, as quais são suportadas através do *software Skype*.

Assim, os objectivos deste projecto para o utilizador final, apresentam-se nos seguintes pontos: proporcionar a monitorização das crianças através de uma aplicação na rede social *Facebook*; proporcionar uma interacção simples e intuitiva da aplicação *Web*, dado os seus utilizadores serem pessoas idosas; permitir aos utilizadores um permanente contacto com as crianças a serem monitorizadas através do *Skype*.

Em resumo, os objectivos da aplicação serão a implementação de: um ou mais sistemas de monitorização; um ou mais sistemas de gestão de dados (*proxy*); um sistema de gestão servidor (*middleware*); um módulo de detecção de actividade; um módulo de localização; um módulo de interface *Web*.

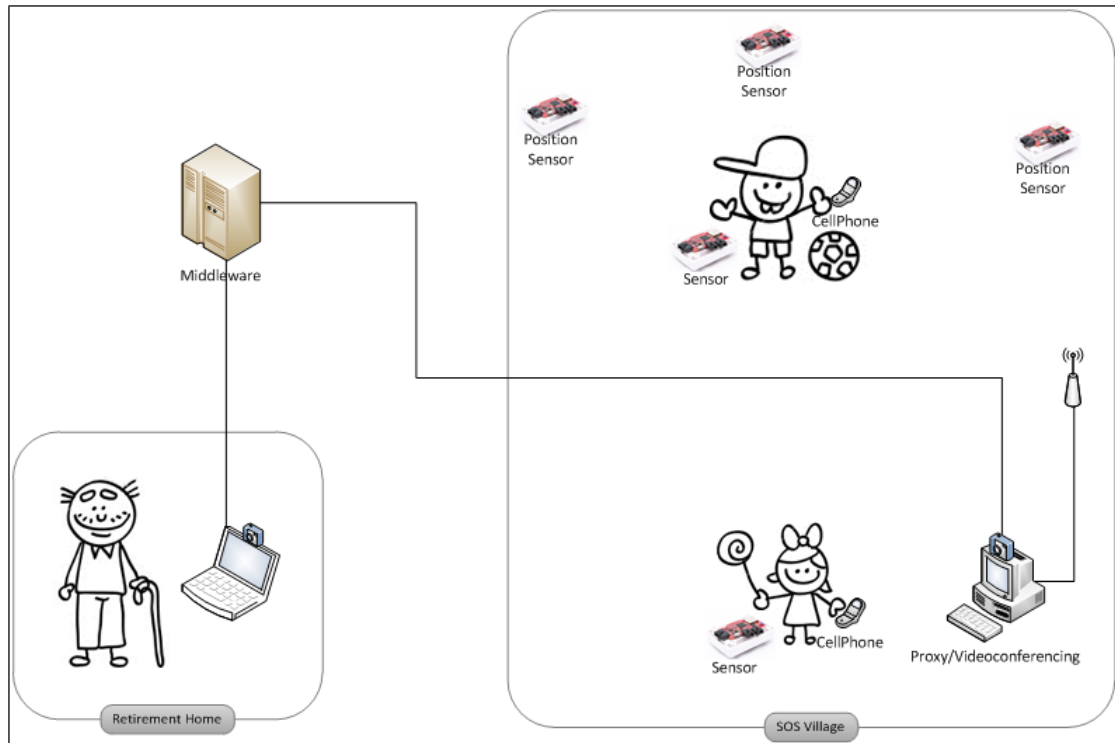


Figura 22 - Visão Geral do Sistema Avós e Netos

6.2. Especificação da Aplicação

A especificação deste projecto pretende fornecer uma visão geral do sistema a ser desenvolvido, incluindo definição, metas, objectivos, contexto e principais recursos.

Para especificar formalmente, serão definidos requisitos funcionais, requisitos não-funcionais, requisitos de qualidade e restrições técnicas. Para descrever os requisitos funcionais foram criados diagramas de Use-Case que se encontram disponíveis no Anexo H.

Esta especificação encontra-se organizada nas seguintes secções:

- Actores, metas que nos propomos atingir, os objectivos da aplicação Web, o contexto e as capacidades;
- Requisitos funcionais, que especificam as funcionalidades do sistema em através de modelos de casos de uso, especificando cada um em detalhe no Anexo H “Requisitos Funcionais do Sistema Avós e Netos”;
- Requisitos não funcionais, que especificam os requisitos do produto em requisitos de usabilidade e de desempenho;
- Requisitos de qualidade, que especificam a interface da aplicação, performance e manutenção;
- Restrições técnicas, que especificam algumas limitações no desenvolvimento da aplicação.

Finalmente, cada requisito terá dois tipos de propriedades associadas:

- MUST-Prioridade máxima (MUST);
- SHOULD-Prioridade média-alta (SHOULD).

6.2.1. Actores e Público-Alvo

Os actores definidos são compostos pelos dois extremos da aplicação: os avós e as crianças. Os avós são os principais actores, pois serão eles que vão interagir directamente com a maioria das funcionalidades da interface. As crianças são actores secundários cujas acções possíveis são: enviar pedidos de videoconferência e enviar/receber mensagens.

O público-alvo, desta aplicação está dividido da seguinte forma:

- *Stakeholders* internos:
 - Grupo de Investigação LCT-WSN: Todos os membros do grupo de investigação, mas designadamente o Professor Dr. Jorge Sá Silva, o doutorando David Nunes e o mestrando André Pinto.
- *Stakeholders* externos:
 - Parceiros das áreas das Ciências Sociais.

6.2.2. Objectivos

Os objectivos deste projecto, para o utilizador final, estão divididos nos seguintes pontos:

- Proporcionar um sistema de monitorização de crianças que detecta os seus movimentos através de um acelerómetro e permite a visualização da temperatura ambiente, humidade e localização;
- Proporcionar a integração de três dispositivos médicos, um oxímetro, um esfigmomanómetro e um espirómetro, cuja integração com o sistema depende da utilização de um telemóvel com tecnologia J2ME;
- Proporcionar a detecção de localização da criança, dentro de um espaço fechado (como uma escola);
- Proporcionar a detecção da actividade da criança (movimento ou não movimento);
- Proporcionar a interacção entre a criança adoptiva e o adulto, através de videoconferência. Ex: a criança poderá “pedir” uma videoconferência através de um botão no seu sensor *wireless*. Este pedido será detectado como um evento que poderá ser consultado pelo adulto através da página *Web*. O adulto poderá usar a mesma página *Web* para iniciar uma chamada de videoconferência para a instituição onde se encontra a criança ou a criança poderá aproximar-se do computador da instituição para iniciar a videochamada;
- Proporcionar a detecção de eventos de interesse, como quedas da criança.

Os objectivos deste projecto, a nível da aplicação, estão divididos nos seguintes pontos:

- Criar um sistema modular que permita a integração de diferentes tipos de sensores e de redes de sensores sem fios. Esta modularização será conseguida através de uma camada de abstracção que assume a forma de um sistema de gestão de dados (*proxy*), o qual corre num computador ligado a um dispositivo *sink*, responsável pela comunicação directa com os sensores sem fios. Os *proxies* comunicam com o *middleware* através de uma camada de *webservices* “inferior”, abstraindo assim *middleware* da comunicação directa com os sensores;
- Implementar um *middleware* central que faça a gestão da informação recebida dos *proxies* e que a armazene numa base de dados. Esta base de dados contém não só a informação sensorial, mas também informação referente ao funcionamento do sistema, como por exemplo um registo das crianças e dos adultos que o usam;
- Implementar um mecanismo de detecção de localização baseado em trilateração;
- Implementar um módulo de classificação de movimento ou não movimento, com base nos dados medidos pelo acelerómetro, bem como de detecção de acelerações súbitas seguidas de paragens bruscas, as quais são associadas com quedas da criança;
- Integrar três dispositivos médicos, um oxímetro, um esfigmomanómetro e um espirómetro, através de um telemóvel com tecnologia J2ME e um programa de *proxy* capaz de comunicar com o telemóvel;
- Implementar uma interface Web que mostre a informação sobre a localização da criança através de um mapa, bem como a sua actividade através de um pequeno avatar. Informação sobre os dados de temperatura, humidade e dos sensores médicos será representada sobre a forma de uma tabela com valores numéricos. A interface Web permitirá também consultar eventos de queda que tenham sido detectados, bem como pedidos de videoconferência por parte da criança. Finalmente, a interface deverá também permitir o início de uma videoconferência entre o adulto a instituição onde se encontra a criança.

Nesta dissertação foi objectivo implementar a estrutura base desta aplicação para que no futuro todas as funcionalidades sejam implementadas, e que sejam avaliados novos mecanismos de privacidade, segurança, intermitência e localização.

6.2.3. Contexto

Como referido anteriormente os actores da aplicação serão os avós adoptivos que se encontrarão na instituição a monitorizar os netos. Essa monitorização é efectuada através dos seguintes elementos:

- *Hardware*:

- Servidor: Este dispositivo é o responsável por disponibilizar o *middleware* que encarrega-se do controlo e gestão de todos os dados e mensagens trocadas entre os avós e crianças;
- *Proxy*: Dispositivo que realiza o pré-processamento e análise dos dados provenientes dos sensores, enviando o resultado para o *middleware*;
- Telemóveis: Estes dispositivos serão os responsáveis pela troca de mensagens escritas entre os avós e as crianças. Dentro da aplicação Web irá existir um pequeno *chat* onde os avós podem enviar mensagens para as crianças através de *sms* e os netos podem responder;
- Sensores: Esta tecnologia será a responsável pela aquisição de diferentes dados como humidade, temperatura e aceleração.
- Rede:
 - Internet: A Internet será a rede de comunicação entre os dispositivos do sistema como: *proxys*, *middleware* e a aplicação do *facebook*.
- *Software*:
 - *Skype*: Esta aplicação será a responsável pela videochamada que será iniciada pela aplicação web (avós) ou instituição (criança), permitindo a comunicação entre os dois actores;
 - *Aplicação Web*: A aplicação web será construída como uma aplicação *facebook*, sendo necessário existir um perfil nesta rede social para lhe poder ter acesso. Esta aplicação irá permitir aos avós utilizar as funcionalidades descritas no capítulo seguinte.

6.2.4. Capacidades

Esta subsecção fornece uma visão geral de alto nível das capacidades mais importantes do projecto *SocialSense* Avós e Netos. Note-se que serão disponibilizadas informações uteis para a compreensão dos requisitos, mas não contem requisitos específicos.

Assim, é apresentada uma lista das capacidades mais relevantes deste sistema:

- Representação da actividade da criança, em termos de movimento ou não-movimento;
- Representação da localização da criança, relativamente à instituição onde se encontra;
- Representação de eventos de queda e de pedidos de videoconferência despoletados pela criança;
- Permitir a realização de videoconferência entre o adulto e a instituição onde se encontra a criança;
- Permitir a troca de mensagens entre o adulto e a criança;
- Permitir pedidos de videoconferências despoletados pelas crianças.

6.2.5. Requisitos Funcionais

O diagrama de casos de uso seguinte resume os requisitos funcionais, de alto nível, para a aplicação web *SocialSense*. Todos os casos de uso e requisitos da aplicação estão disponíveis no Anexo H.

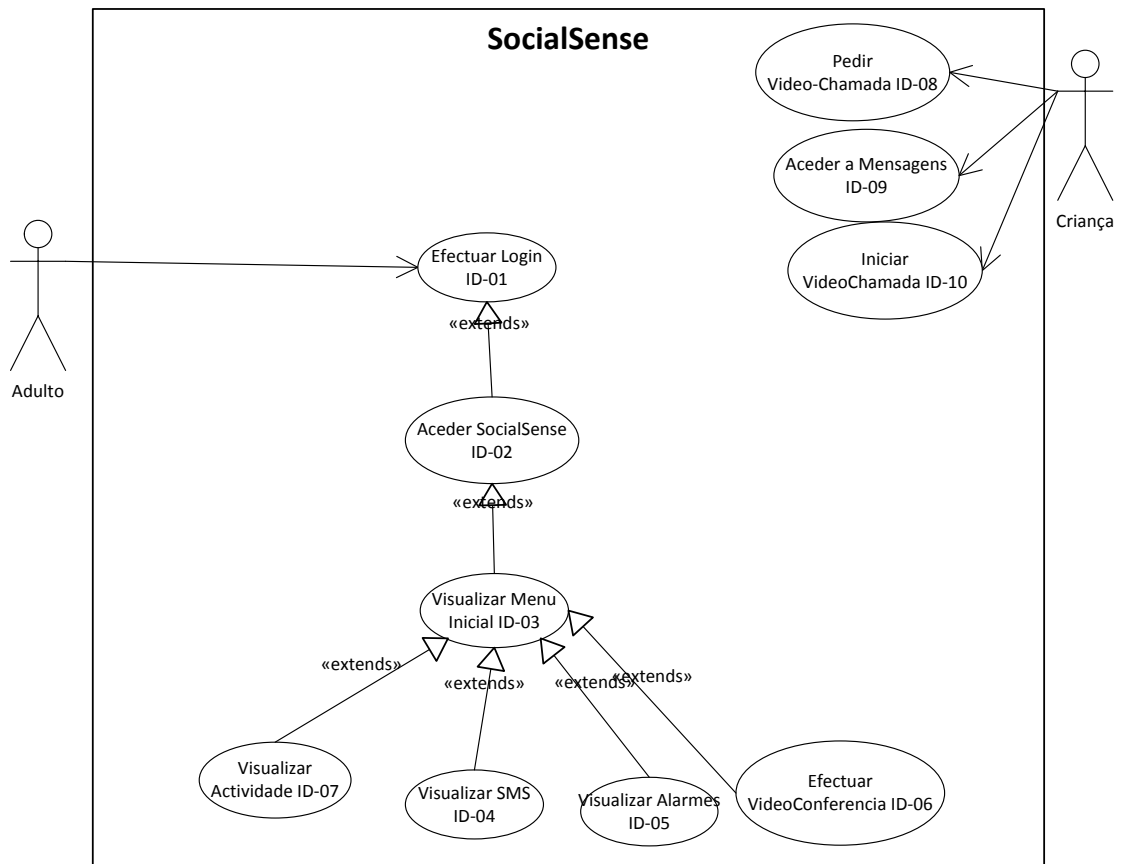


Figura 23 - Diagrama de Casos de Uso SocialSense Avós e Netos

6.2.6. Requisitos Não Funcionais

❖ Requisitos de Usabilidade

User Friendly: A aplicação *SocialSense* Avós e Netos tem o objectivo de ser o mais simples possível, uma vez que, como principais actores tem os idosos que, tendencialmente, têm uma menor apetência para este tipo de tecnologias.

✓ Prioridade – MUST

❖ Requisitos de Desempenho

Transacções processadas: O *middleware*, *proxys* e aplicação *web* têm de efectuar as transacções num tempo considerável pelo utilizador.

✓ Prioridade – MUST

Tempo de resposta ao utilizador: O tempo de resposta da aplicação ao utilizador tem de ser aceitável por parte do utilizador da aplicação.

✓ Prioridade – MUST

6.2.7. Requisitos de Qualidade

❖ Interface com o Utilizador

- A interface deve poder ser utilizada em várias plataformas como *pc*, *smartphones* e *tablets*;
- A interface deve ser *user-friendly* permitindo aos utilizadores navegar intuitivamente.

❖ Performance

- Os pedidos feitos pela aplicação *Web* ao *middleware* devem ser respondidos da forma mais rápida possível;
- A navegação entre páginas deve ser o mais fluida possível;
- Não devem ser utilizadas tecnologias web como o *flash* que tenham um processamento pesado.

❖ Manutenção

- O código fonte deve estar bem documentado para que no futuro outros *developers* possam continuar o projecto percebendo o comportamento das diferentes funções;

- A aplicação deve ser modular para permitir a adição de novos módulos no futuro;

6.2.8. Restrições Técnicas

❖ Limitações de Software

- A aplicação deve usar a *framework Dojo* para fazer animações de um modo simplista;
- A aplicação pode utilizar linguagens de programação como *PHP*, *Javascript*, *HTML* e *CSS (Cascading Style Sheets)*;
- A aplicação não pode utilizar linguagens de programação *Web* como o *Flash*.
- A plataforma *SocialSense* deve ser implementada em *Java* com o auxílio das bibliotecas *Hibernate* e *Jersey*.

6.3. Arquitectura do Sistema

Este capítulo tem como objectivo descrever a arquitectura *SocialSense* no âmbito do projecto Avós e Netos(as). Primeiramente, será apresentada uma descrição geral do *middleware*, mostrando os componentes que foram remodelados (note-se que esta descrição apenas fornece uma visão global da arquitectura). Em segundo, será apresentado o diagrama de classes, com todas as classes criadas para responder aos diferentes *proxys* e à aplicação *Facebook*. Por último, será mostrado o diagrama de entidade relacionamento da base de dados Avós e Netos(as).

6.3.1. Descrição Geral

Como foi dito no início do capítulo do sistema Avós e Netos(as), respeitante à arquitectura do programa, este seguiu os estudos preliminares efectuados ao longo do primeiro semestre.

No entanto, depois da especificação do caso de estudo Avós e Netos(as), surgiu a necessidade de tornar alguns componentes mais modulares e robustos. Seguidamente será apresentado o novo modelo com alguns dos novos componentes, que iremos descrever.

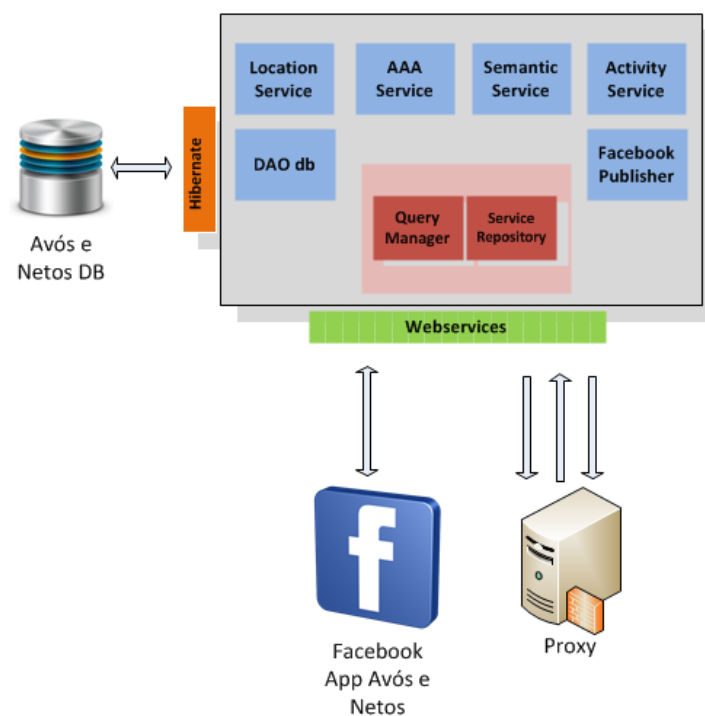


Figura 24 - Arquitectura *SocialSense* da plataforma Avós e Netos

Como podemos observar na Figura 24, foram efectuadas alterações em relação à plataforma inicialmente desenvolvida Figura 18.

Os *proxys* que antigamente comunicavam por *sockets* deixaram de comunicar através dos elementos *Adapters* e *Requests*, passando a comunicar através de *webservices*. Esta alteração deve-se à necessidade do requisito de modularidade, ou seja, a partir de agora cada *proxy* possui o seu próprio *webservice* para inserir dados na plataforma.

Além disso, foram adicionados novos módulos como *DAO db* e alterados alguns dos existentes como *Location Service*, *Query Manager*, *AAA Service*, *Activity Service* e *Facebook Publisher*.

O módulo *DAO db* surgiu da necessidade de ter um controlo das *queries* à base de dados. Como podemos visualizar na imagem, a base de dados é controlada através de um mapeamento objecto-relacional. Até este ponto, todos os *webservices* tinham as suas *queries*, o que originava por vezes repetição de código em *queries* iguais. Assim, através da inserção desse novo módulo, foi possível ter as *queries* associadas a cada objecto.

Dado o módulo *DAO db* ter sido inserido, o módulo *Query Manager* sofreu também algumas alterações. O módulo foi alterado ficando responsável pelo controlo das sessões *hibernate* através de um objecto *Utils*.

Apesar de existir um módulo *Location Manager*, este não suportava localização *indoor*. Como um dos requisitos especificados para o sistema Avós e Netos definia que a criança tem de ser monitorizada dentro de espaços *indoor*, foi efectuada a integração deste componente com o *proxy* desenvolvido pelo colega André Pinto.

O modo como o sistema realizava a autenticação dos utilizadores também foi alterada através do módulo *AAA Service*. Agora todos os pedidos vindos da aplicação são autenticados pelo próprio *Facebook*. Como descrito no Anexo F, a autenticação passou a ser efectuada através de *AOutb*.

Para além de todas as alterações a estes módulos foi também necessário inserir novos *webservices* para responder aos requisitos especificados. A lista dos *webservices* implementados está discriminada na Tabela 6.

6.3.2. Plataforma

Na página que se segue encontra-se o diagrama de classes da plataforma *SocialSense* Avós e Netos(as). A linguagem utilizada para o desenvolvimento da plataforma foi *Java*.

Em primeiro lugar, podemos observar um conjunto de classes com o nome *DAO* (*Database Access Objects*). Cada classe com este nome é constituída por um conjunto de operações de manipulação da base de dados (p.ex. inserir, apagar, actualizar). Estas classes representam o módulo *DAO db* e encontram-se dentro do *package DAO*.

Em segundo lugar, podemos visualizar um conjunto de classes que vai da *VideoConfImpl* até *SensorImplementation* e se encontram dentro do *package DatabaseConn*. Têm como principal funcionalidade fazer o tratamento dos pedidos que vêm dos *webservices*. Dentro deste grupo encontramos as classes responsáveis pela localização e actividade.

Em terceiro lugar, podemos visualizar as classes responsáveis por mapear os objectos na base de dados através do *Hibernate*, encontrando-se dentro do *package hibernateMaps*. Assim, cada entidade definida no modelo entidade relacionamento é representada por uma destas classes.

Em quarto, existe um conjunto de classes responsáveis por encapsular os pedidos e respostas efectuados através dos *webservices*, encontrando-se dentro do *package model*. Este conjunto de classes vai desde a classe *AlertData* até à classe *SkypeData*. No Anexo F podemos visualizar objectos *JSON* e *XML* dessas classes.

Em quinto e último lugar, encontramos as classes que representam os *webservices*, dentro do *package web*. Estas classes são as responsáveis por definir o *url* do *webservice*, assim como, os respectivos métodos *post* e *get*.

A classe *HibernateUtils* é a classe responsável pelo controlo das diferentes sessões, ou seja, representa o módulo *Query Manager*.

6.3.3. Base de Dados

Para guardar os dados vindos das tecnologias e da aplicação *Facebook* foi criada uma base de dados representada pela Figura 26.

Como referido anteriormente, a comunicação entre a base de dados e o *middleware* deu-se através da biblioteca *Hibernate*, permitindo um relacionamento objecto-relacional. A base de dados foi criada em *Postgres* e desenhada através do programa *PowerDesigner*.

As entidades criadas representam as crianças, os nós sensores, os avós adoptivos, os eventos dos sensores e localização.

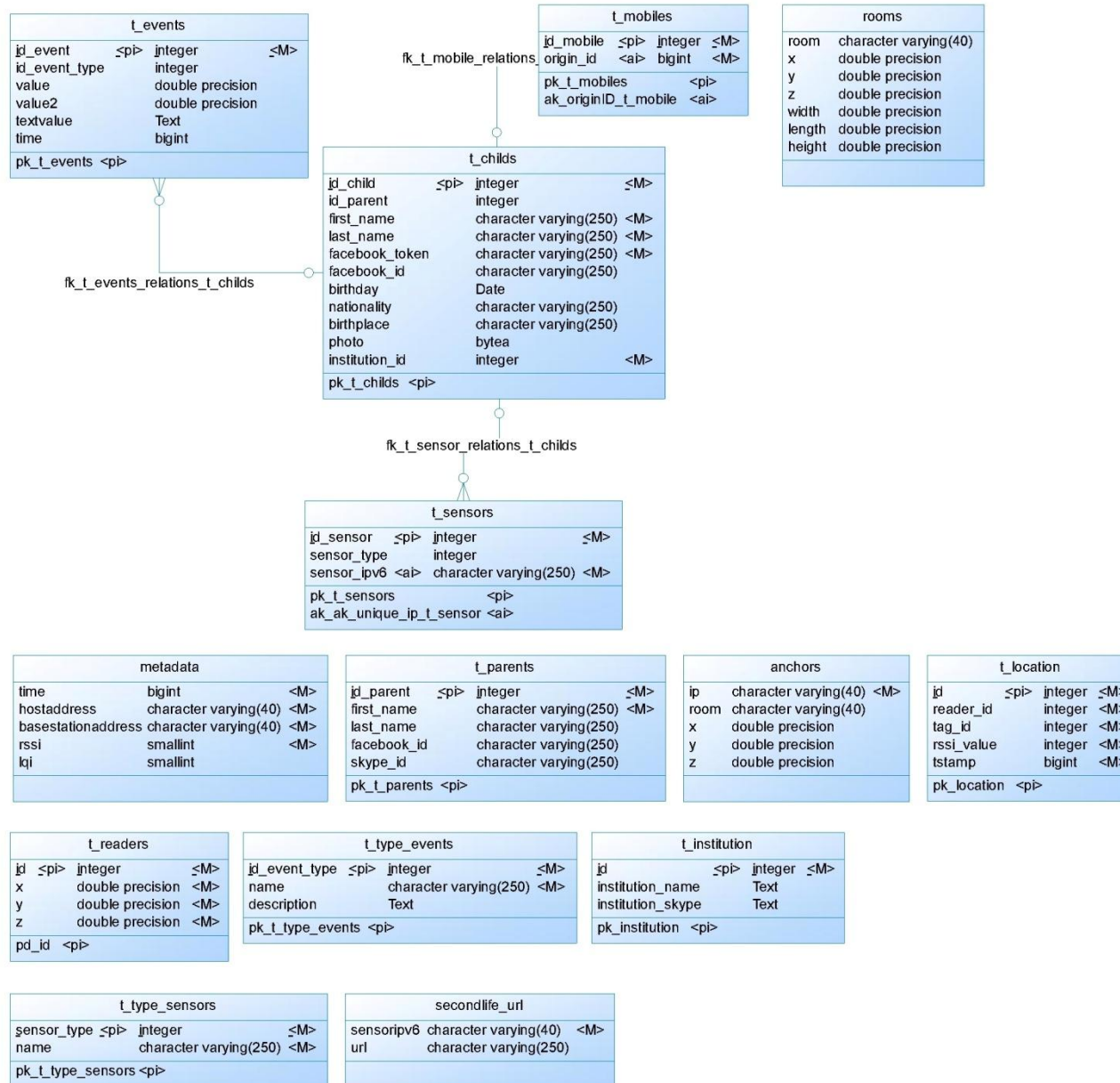


Figura 26 - Diagrama Entidade-Relacionamento do sistema Avós e Netos

6.4. Aplicação e Demonstração

Finalizado o processo de desenvolvimento da aplicação Web através da biblioteca *Dojo*, esta foi integrada com a rede social.

Como falado anteriormente, esta aplicação tem como finalidade proporcionar aos avós “virtuais” uma monitorização através das redes sociais. Assim, a primeira etapa necessária para o acesso à aplicação trata-se de um registo na rede social *Facebook*. Seguidamente, será adicionado esse registo à plataforma *SocialSense*.

Efectuado o registo na rede social e aceitando as permissões da aplicação, o avô poderá aceder à aplicação na rede social como é mostrado na Figura 27.

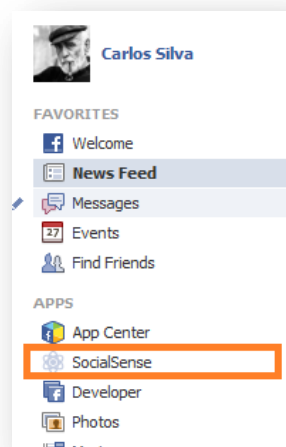


Figura 27 - Acesso à aplicação Avós e Netos

Depois de aceder à aplicação e esta carregar todos os elementos, o utilizador deparar-se-á com a página principal que contém o menu de acesso a todas as funcionalidades. Para aceder a estas funcionalidades, o utilizador terá que escolher o(a) neto(a) que pretende visualizar através de um menu que aparece ao clicar.

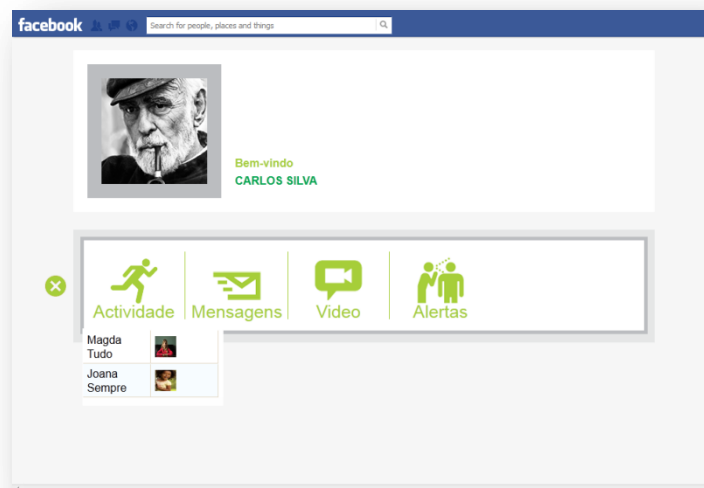


Figura 28 - Menu Principal aplicação Avós e Netos

6.4.1. Actividade

Como ficou definido nos requisitos presentes no Anexo H, a aplicação contém uma página de actividade. Esta página de actividade é responsável por disponibilizar os dados fisiológicos da criança, dados pessoais, actividade e localização.

Assim, quando o avô(ó) “virtual” deseja visualizar se a criança de encontra bem, este(a) acede a esta página, visualizando o local onde o(a) neto(a) se encontra e os dados fisiológicos como temperatura, actividade, etc.

A visualização da posição é efectuada através da *API* do *Google Maps*. Para a localização *indoor*, o espaço/sala necessita de estar registado na base de dados da plataforma *SocialSense* com uma coordenada de referência, para que o mecanismo de localização *indoor* calcule a posição do neto.

A actividade é representada através de um pequeno avatar que se anima com o movimento do(a) neto(a).

Toda a informação relativa à localização, temperatura e humidade é actualizada a cada dois segundos através de pedidos efectuados aos *webservices*.

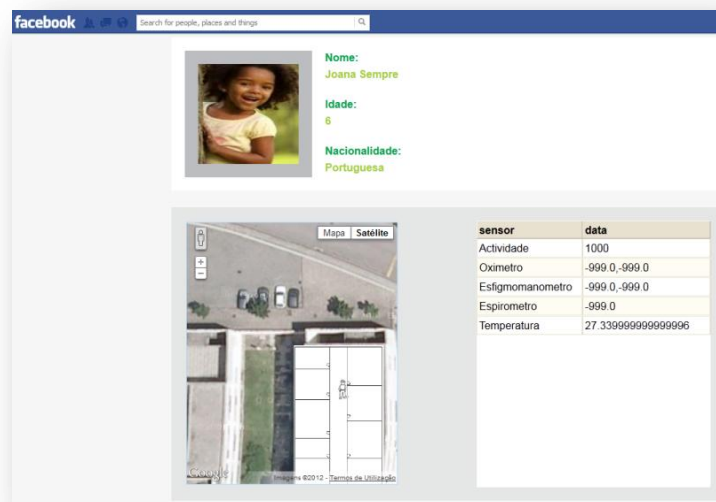


Figura 29 - Página de Actividade da Aplicação Avós e Netos

6.4.2. Mensagens

Como se poderá visualizar na fase de testes este requisito da aplicação não foi implementado, assim o menu não tem nenhuma acção relativa a esta funcionalidade.

6.4.3. Videochamadas

Para promover uma melhor interacção entre o(a) avô(ó) virtual e o(a) neto(a), a aplicação possibilita videochamadas através da utilização da aplicação *Skype*.

A videochamada tanto pode ser efectuada do lado do avô(ó) como do lado da criança na sua instituição. Por um lado, se o avô(ó) quiser efectuar uma videochamada dirige-se ao menu na aplicação e carrega no botão de chamar apresentado na Figura 30. Por outro lado, se a criança quiser falar com avô(ó), dirige-se ao computador da instituição, que automaticamente inicia a chamada através do mecanismos de detecção de proximidade.

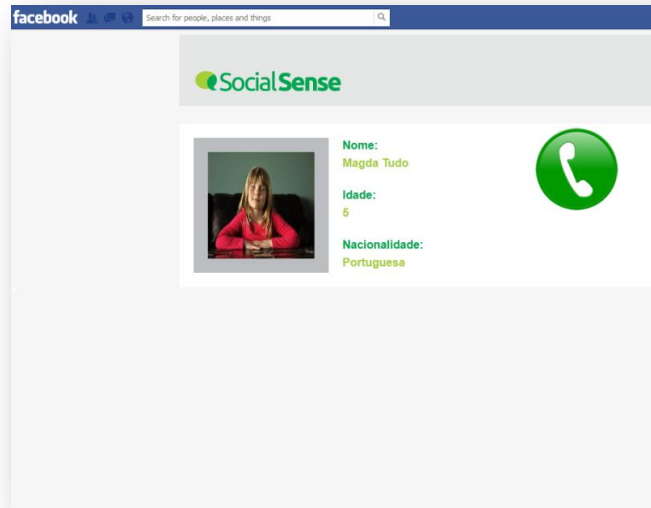
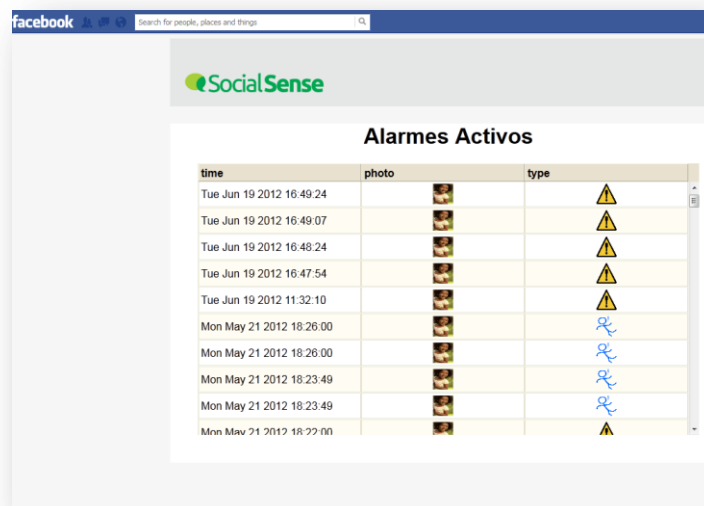


Figura 30 - Página de Videochamada da Aplicação Avós e Netos

6.4.4. Alertas

Esta página permite visualizar se a criança tem algum alerta. Os alertas são representados por símbolos e podem ser de dois tipos (pedido de contacto ou queda).

O pedido de contacto é efectuado através do botão do nó sensor que a criança transporta, ou seja, ao pressionar o botão do sensor é enviado um alerta para o *middleware* e depois apresentado na página. A queda é efectuada a partir do nó sensor, que através do seu acelerómetro detecta movimentos repentinos que são enviados para o *middleware* e depois apresentado na página em forma de queda (Figura 31).



6.5. Principais Desafios

Como em todos os projectos existem sempre desafios que têm que ser superados para que o projecto não falhe com o que está definido.

Assim, os principais desafios deste projecto estiveram presentes na implementação da aplicação *Web*. A implementação *Web* foi desenvolvida utilizando linguagens como o *HTML*, *PHP*, *javascript* e a biblioteca *Dojo*.

Tendo unicamente experiência em linguagens como *.net* e *HTML* em desenvolvimento *web*, desde cedo comecei a apreender os conceitos necessários para o desenvolvimento da aplicação *web*. Apesar da rápida compreensão de linguagens como o *PHP* e *Javascript* existiram muitas dificuldades relativamente à framework *Dojo*.

A *framework Dojo* apesar de ser muito conhecida e de ter alguma documentação, necessita de alguns conhecimentos sólidos em *Javascript* o que não era o meu caso. Assim, inicialmente tive muitas dificuldades na elaboração de certas funcionalidades na aplicação, mas que foram ultrapassadas para a fase final do projecto.

6.6. Conclusão

O principal objectivo do projecto Avós e Netos definido inicialmente era o de desenvolvimento de um sistema de monitorização, em tempo real ou desfasado, que permitisse o acompanhamento de crianças desfavorecidas em contextos sociais como, instituições de apoio e acolhimento a crianças socialmente vulneráveis, criando-se o termo “Avós Virtuais”.

Além disso, pretendia-se encontrar um parceiro da área das Ciências Sociais para criar um projecto conjunto.

Através de contactos efectuados no final do desenvolvimento do projecto, com investigadores do Centro de Estudos Sociais (CES) da Universidade de Coimbra através da Prof. Dr. Sílvia Portugal e aluna Joana Alves, foi encontrado um novo rumo para o projecto e criado uma parceria.

Na opinião da investigadora, o projecto no estado actual levanta muitos problemas de privacidade, dado o avô virtual não pertencer à família do neto adoptivo e o neto ser menor.

Assim, surgiu a ideia que esta aplicação poderia no futuro também ser adaptada para uso em perfis familiares de pessoas dependentes e com necessidade de ter um acompanhamento constante.

Capítulo 7

Projecto Youth5G

7.1. Descrição do Projecto

O objectivo deste projecto é criar uma aplicação ubíqua baseada em dados colaborativos. A aplicação permitirá às pessoas que frequentam ambientes nocturnos a identificação de locais com uma maior afluência, bem como, outras informações relativas a eventos que ocorram nesses locais.

O sistema será constituído por uma aplicação ubíqua que correrá no sistema operativo *Android* e por um servidor *backend* que integra os serviços *SocialSense*. A integração dos serviços *SocialSense* na aplicação *Youth5G* deve-se à possível utilização de mecanismos existentes na aplicação *Avós e Netos* (localização indoor e de proximidade), tal como, futuramente dados de outras tecnologias (*WSN*).

Deste modo, o servidor será o componente responsável pela resposta a qualquer pedido de informação vindo da aplicação ubíqua. Essa troca de informação será feita através de *Webservices REST*, que tratarão de realizar a autenticação dos utilizadores e responder aos pedidos, com base na informação que se encontra na base de dados e na *API Foursquare*. A *API Foursquare* permitirá obter os diferentes locais no país e os seus eventos. Assim, é seguido um modelo de modularidade já apresentado no projecto *SocialSense Avós e Netos*, que permite no futuro adicionar ou remover serviços, sem comprometer serviços existentes.

A aplicação ubíqua é responsável pela visualização da informação, pela interacção com o utilizador e pela recolha dos dados. Esta terá um mapa que permitirá ao utilizador visualizar os diferentes locais mostrando a afluência de pessoas nos eventos que ocorram naquela hora. Além disso, os utilizadores poderão participar e indicar a afluência de pessoas num determinado local, através de uma escala entre 0 (baixo) e 5 (alto). Para que os utilizadores estejam interessados em participar, a aplicação terá um sistema de incentivo na forma de ranking. Este sistema de ranking tem como objectivo a participação dos utilizadores de forma colaborativa ao indicarem o nível de afluência de um determinado evento. Isto é, os utilizadores que fornecerem informações verdadeiras ao sistema irão ter um acréscimo na sua pontuação e os utilizadores que fornecerem informações falsas ao sistema irão ver a sua pontuação decrescer. Os utilizadores que tenham a pontuação elevada irão poder ter ofertas como produtos, serviços e descontos Sapo no futuro.

Assim, os objectivos deste projecto para o utilizador final estão divididos nos seguintes pontos: proporcionar a visualização no mapa de locais/eventos; proporcionar um sistema que permita a visualização da afluência aos locais; permitir aos utilizadores mais participativos a possibilidade de ganhar ofertas Sapo.

Em suma o sistema será constituído por: uma aplicação *Android*; um servidor que fornece os serviços à aplicação; serviços na forma de *Webservices* que fornecem *POI's (Point of Interest)*, eventos, etc; uma *API* externa do *Foursquare*.

A imagem que se segue apresenta a o sistema:

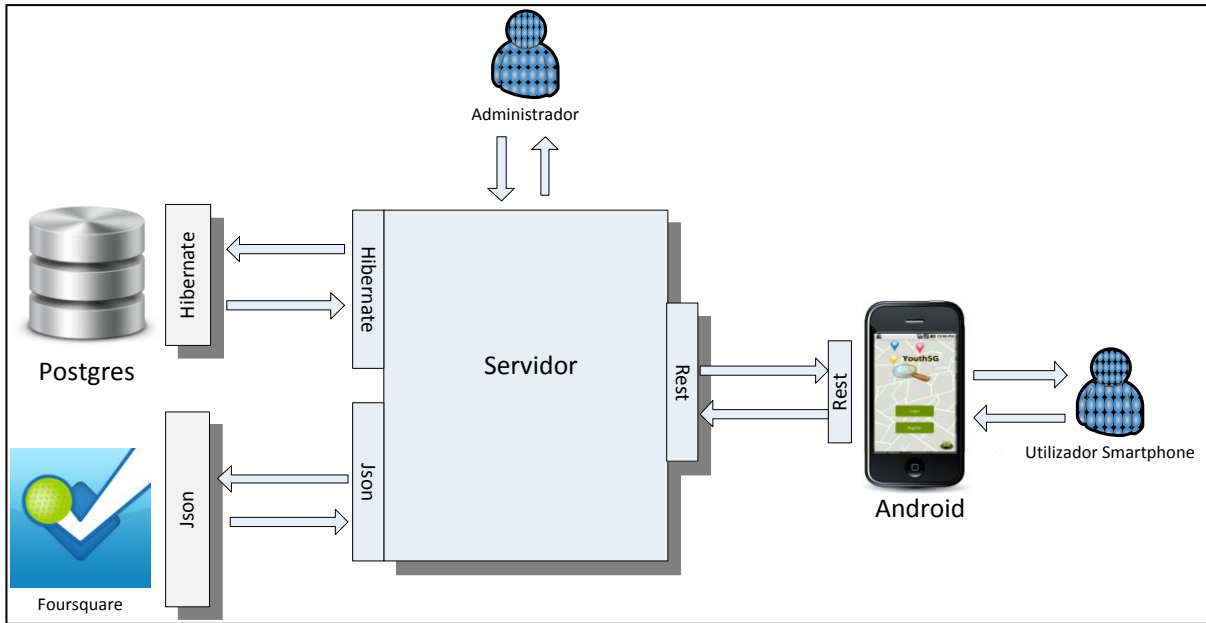


Figura 32 - Visão Geral do Sistema Youth5G

7.2. Especificação da Aplicação

Esta especificação documenta os requisitos do sistema da aplicação ubíqua *Youth5G*. Fornece uma visão geral do sistema a ser desenvolvido, incluindo definição, metas, objectivos, contexto e principais recursos. Para descrever o sistema e a interacção entre os vários actores foram criados requisitos funcionais apresentados através de diagramas *use-case* (Anexo I), requisitos não funcionais, requisitos de qualidade e restrições técnicas.

Esta especificação está organizada nas seguintes secções:

- Actores, metas que nos propomos atingir, os objectivos da aplicação *Android*, o contexto e as capacidades;
- Requisitos funcionais que especificam os requisitos funcionais do sistema em termos de modelos de casos de uso, especificando cada um em detalhe.
- Requisitos não funcionais que especificam os requisitos do produto em requisitos de usabilidade e de desempenho;
- Requisitos de qualidade que especificam a interface da aplicação, performance e manutenção;
- Restrições técnicas que especificam algumas limitações no desenvolvimento da aplicação.

Cada requisito apresentado será representado por uma prioridade do tipo:

- MUST- Prioridade máxima (MUST);
- SHOULD- Prioridade média-alta (SHOULD);

7.2.1. Actores e Público-Alvo

Os actores definidos nesta documentação são compostos pelos utilizadores da aplicação. Estes utilizadores são pessoas que se encontram entre os 14 e 35 anos de idade, de ambos os géneros e que gostem de frequentar eventos/ambientes nocturnos. Uma vez que a aplicação é baseada em dados colaborativos são os utilizadores da aplicação que têm o maior destaque, pois sem eles o sistema não conseguiria calcular a afluência de pessoas num determinado local.

O público-alvo deste sistema está dividido da seguinte forma:

- *Stakeholders internos*:
 - Grupo de investigação LCT-WSN: Todos os membros do grupo de investigação, designadamente o Professor Dr. Jorge Sá Silva, o doutorando David Nunes e o mestrando André Pinto;
- Sapo Labs: Os coordenadores do projecto da Sapo, designadamente o Pedro Torres e o Benjamin Júnior.
- Stakeholders externos:
- Utilizadores: Pessoas com um perfil entre os 14 e 35 anos, ambos os géneros e com uma vida social activa. O número de participantes que irão testar a nossa aplicação será entre 3 e 5.

7.2.2. Objectivos

Os objectivos deste projecto para o utilizador final estão divididos nos seguintes pontos:

- Criação de um sistema que permita visualizar POI (pontos de interesse), eventos, e a respectiva afluência recorrendo a uma escala apropriada;
- Permitir o anonimato nas operações efectuadas;
- Visualização das características dos diferentes POI e eventos;
- Permitir a participação do utilizador através da indicação da afluência a um evento;
- Motivar a utilização do sistema através de um ranking que incentive os utilizadores a participar e a fornecer informações válidas.

7.2.3. Contexto

Esta subsecção documenta o contexto do projecto a ser desenvolvido, em termos da sua interacção com os actores. Como referido anteriormente, os actores da aplicação serão pessoas com uma vida social activa, que frequentem eventos nocturnos(bares, discotecas, concertos, eventos culturais) e com idade compreendida entre os 14 e 35 anos de idade. As funcionalidades que o sistema disponibilizará são efectuadas pelos seguintes elementos:

- *Hardware:*
 - Servidor: Este dispositivo é o responsável por disponibilizar os serviços em forma de *WebServices REST*, gerir os dados de todos os utilizadores e pontos de interesse e realizar os pedidos à API do *Foursquare*.
 - *Smartphones:* Estes dispositivos serão os dispositivos responsáveis por disponibilizar a aplicação ao utilizador e um conjunto de dados de sensores como por exemplo o *GPS*.
- Rede:
 - Internet: A ligação entre o servidor e a aplicação ubíqua será realizada através de uma ligação de dados a partir das tecnologias Wifi, 2G, 3G, etc, de modo a permitir ao utilizador um grande nível de mobilidade.
- *Software*
 - Aplicação Ubíqua: A aplicação ubíqua será construída para o sistema operativo *Android 2.1*, devido à grande compatibilidade com as versões mais actuais. Para além disso, a aplicação ubíqua irá permitir ao utilizador usufruir das funcionalidades que este sistema pretende fornecer;
 - API *Foursquare*: Esta API será responsável por fornecer ao servidor um conjunto de informações como: pontos de interesse, eventos, fotografias, comentários, etc, dados pelos utilizadores do *Foursquare*.

7.2.4. Capacidades

Seguidamente será fornecida uma visão geral das capacidades mais importantes da aplicação *Youth5G*. Esta lista contém informações úteis para a compreensão dos seguintes requisitos, mas não contém requisitos específicos.

Assim é apresentada uma lista das capacidades mais relevantes deste sistema:

- Representação através de um mapa do local e dos pontos de interesse em torno do utilizador;
- Possibilidade de ver as características dos POIs, os seus eventos, atribuir *feedback*, e visualizar previsão de sua afluência;
- Possibilidade de traçar um trajecto até ao evento pretendido;
- Previsão da popularidade dos locais com base num histórico;
- Possibilidade de visualização do *ranking*.

7.2.5. Requisitos Funcionais

O diagrama de casos de uso da Figura 33 resume os requisitos funcionais, de alto nível, para a aplicação *Youth5G*. Todos os casos de uso e requisitos estão disponibilizados e descritos no Anexo I.

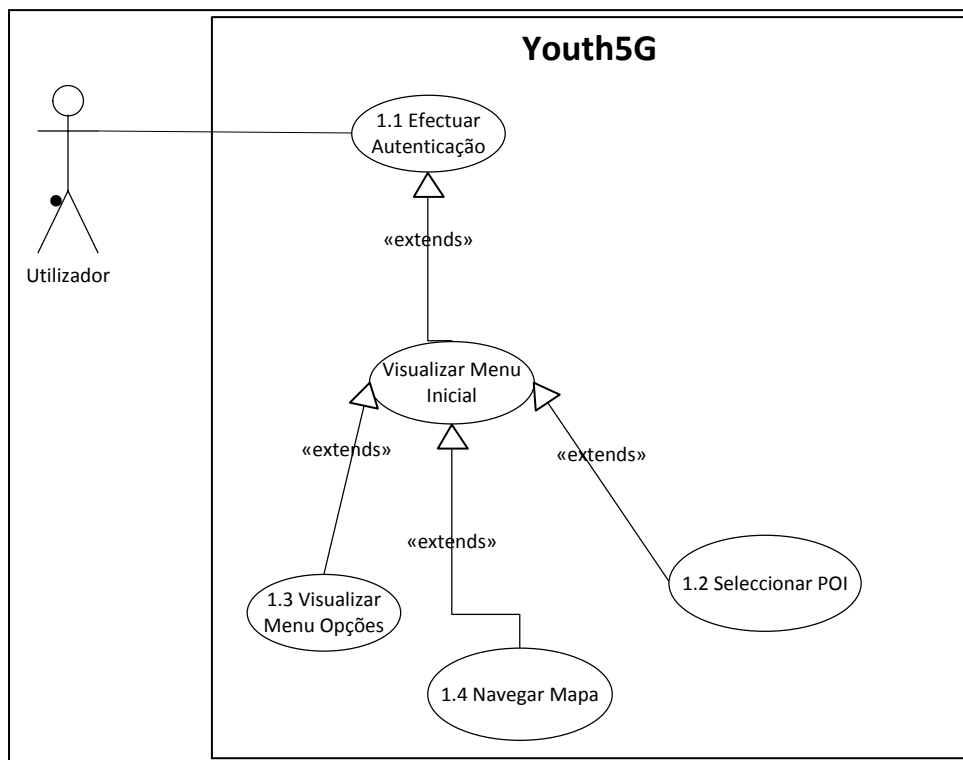


Figura 33 - Diagrama de Casos de Uso da Aplicação Youth5G

7.2.6. Requisitos Não Funcionais

❖ Requisitos de Usabilidade

- **Android:** Dado a aplicação ser concebida para ser executada no sistema operativo *Android*, esta deve obedecer às características como (Menus e Actividades) já existentes neste sistema operativo.
 - ✓ Prioridade – MUST
- **User Friendly:** A aplicação *Youth5G* deve disponibilizar a informação ao utilizador de um modo prático e objectivo.
 - ✓ Prioridade – MUST
- **Ajuda:** A aplicação deve conter um menu de ajuda que informe o utilizador sobre o modo de funcionamento da aplicação.
 - ✓ Prioridade – SHOULD

❖ Requisitos de Confiabilidade

- **Tempo médio de resposta:** A Aplicação deve esperar por uma resposta do servidor durante 10 segundos. Se passado este tempo não existir nenhuma resposta, a aplicação deve mostrar uma mensagem ao utilizador a informar que houve um problema na ligação.
 - ✓ Prioridade - MUST

❖ Requisitos de Desempenho

- **Transacções processadas:** O servidor e a aplicação ubíqua têm de efectuar as transacções num tempo considerável. Sempre que seja efectuada uma transacção a aplicação ubíqua deve mostrar uma animação.
 - ✓ Prioridade – MUST
- **Tempo de resposta ao utilizador:** O tempo de resposta da aplicação ao utilizador tem de ser aceitável por parte do utilizador da aplicação.
 - ✓ Prioridade - MUST

7.2.7. Requisitos de Qualidade

❖ Interface com o Utilizador

- A interface deve poder ser utilizada em várias plataformas *smartphones* e *tablets*;
- A interface deve ser *user-friendly* permitindo aos utilizadores navegar intuitivamente;

- A interface deve ser user-friendly permitindo aos utilizadores navegar nela intuitivamente;

❖ Performance

- Os pedidos feitos pela aplicação ao servidor devem ser respondidos da forma mais rápida possível;
- A navegação entre actividades deve ser fluida, bem como a navegação no mapa;
- Os pedidos devem ser feitos de um modo independente da interface

❖ Manutenção

- O código fonte deve estar bem documentado para que no futuro outros *developers* possam continuar o projecto percebendo o comportamento das diferentes funções;
- A aplicação deve ser modular para permitir a adição de novos módulos no futuro;

7.2.8. Restrições Técnicas

❖ Limitações de Software

- A Aplicação deve ser escrita em Java e deve seguir os parâmetros e regras definidos pela plataforma Android.

7.3. Arquitectura do Sistema

Este capítulo descreve a arquitectura da plataforma *SocialSense* depois da integração dos novos módulos do sistema *Youth5G*. Em primeiro, será apresentada uma descrição geral de alto nível da plataforma mostrando os componentes que foram adicionados (note-se que esta descrição apenas fornece uma visão global da arquitectura). Em segundo, será apresentado o diagrama de classes do sistema com uma pequena descrição. Em terceiro, será apresentado o diagrama de classes da aplicação *Android* com uma pequena descrição da estrutura seguida e do significado das classes. Por último, será apresentado o diagrama de entidade-relacionamento da base de dados.

7.3.1. Descrição Geral

Depois de terminada a integração do sistema Avós e Netos iniciou-se a implementação dos módulos necessários ao sistema *Youth5G*.

Além dos módulos desenvolvidos no projecto Avós e Netos, no projecto *Youth5G* foi necessário implementar módulos adicionais e alterar módulos existentes, como podemos visualizar pela .

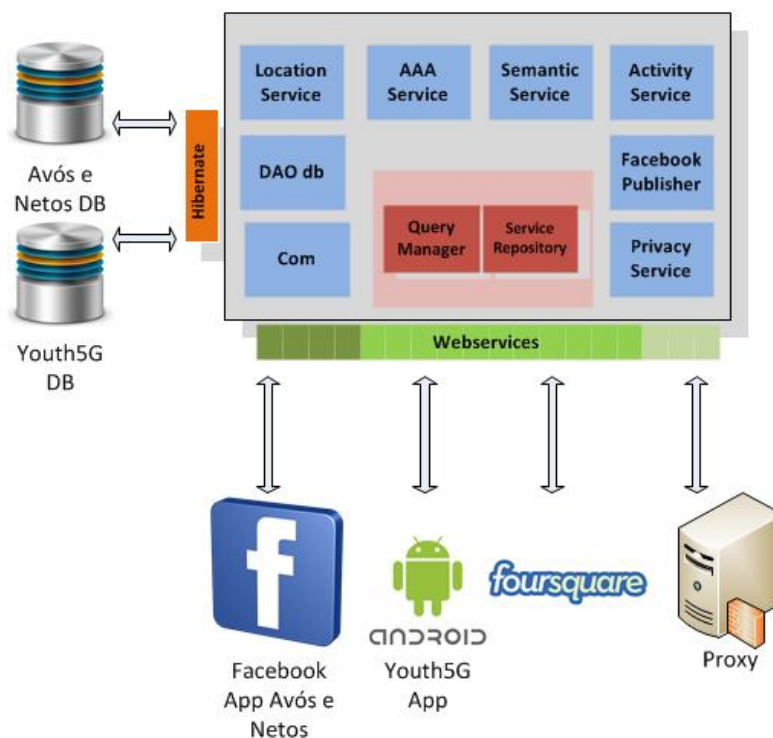


Figura 34 - Arquitectura SocialSense da plataforma Youth5G

Observando a nova arquitectura surgem dois novos módulos *Privacy Service* e *Com*. Além disso, verificamos que agora a plataforma possui uma nova base de dados e que comunica com a rede social *Foursquare* e com *smartphones Android*. Para tornar possível esta integração foi necessário adicionar novos *webservices* definidos na Tabela 7 e novos objectos Dao. Uma visão mais completa das classes implementadas é dada pela Figura 36.

Módulo de Privacidade

O controlo de privacidade associado ao ambiente (presente no estado de arte) foi implementado através do módulo *Privacy Service*. Este módulo é constituído por um conjunto de *webservice*s que verificam as permissões e serviços disponíveis para cada ambiente, permitindo à aplicação *Android* enviar os dados. Podemos visualizar o mecanismo de privacidade através da Figura 13.

Módulo Conn

Adicionalmente foi inserido o módulo *Com* que gere todos os pedidos recebidos e enviados pelos *webservice*s através de diferentes classes. É através deste módulo que são efectuados por exemplo o cálculo do ranking, o cálculo da popularidade e o cálculo da previsão.

O *ranking* é calculado através do desvio padrão da média de todos os feedbacks. Ou seja, para saber se um dado utilizador está a enviar um *feedback* “verdadeiro” é feito o cálculo da média dos *feedbacks* dos outros utilizadores e com isso verificado o seu *feedback*.

A popularidade de um determinado local (POI) é calculada através da média dos feedbacks submetidos pelos utilizadores naquela hora. Assim, de hora a hora este algoritmo corre e actualiza na base de dados a popularidade de cada POI. Se não existir nenhum feedback, a popularidade fica a zero e o POI é representado pela cor preta, como definido na especificação.

O cálculo da previsão da afluência de um determinado local é efectuado através de uma média aritmética ponderada. A utilização desta média possibilita que as semanas mais distantes tenham um peso menor do que as mais actuais.

$$\bar{p} = \frac{x_1p_1 + x_2p_2 + \dots + x_np_n}{p_1 + p_2 + \dots + p_n}$$

Figura 35 - Fórmula da previsão da afluência Youth5G

7.3.1. Plataforma

Uma das primeiras preocupações que tive desde o início deste projecto relacionou-se com a organização de todo o código, dado ser um projecto com muitas funcionalidades. Seguindo os mesmos padrões do projecto Avós e Netos, tentou-se que certas classes seguissem o mesmo padrão relativamente ao nome. Em seguida, iremos descrever de um modo geral algumas das classes presentes na Figura 36.

Em primeiro, visualizamos todas as classes Dao implementadas. Como anteriormente, as classes são as responsáveis pelas operações de manipulação da base de dados (p.ex. inserir, apagar, actualizar). Como podemos ver na Figura 36 foram implementadas doze novas classes do projecto *Youth5G*.

Em segundo, visualizamos todas as classes gerados pelo Hibernate para representar a manipulação das entidades da base de dados. Estas classes possuem todos os atributos das tabelas em forma de variáveis.

Em terceiro, visualizamos todas as classes *Com* que tratam os pedidos vindos de cada *webservice*. Como podemos ver o nome da classe é semelhante ao do *webservice*. Como referido anteriormente é nestas classes que são feitas algumas operações como o cálculo do *ranking*, o cálculo da popularidade e a previsão.

Em quarto, temos um conjunto de classes responsáveis por transportar os dados entre os *webservices* e o *smarphone*. Estas classes são designadas pela acção e pelo sentido da mensagem (p.ex. *RequestDetailEvent* e *ResponseDetailEvent*). Como podemos ver no Anexo G, as mensagens transferidas entre o smarphone e a plataforma são mensagens *JSON*.

Por último, encontram-se os diferentes *webservices* que recebem e enviam os dados. Cada *webservice* é identificado através de um *url* e definidos os respectivos métodos *get* e *post*.

7.3.2. Aplicação

Apesar do desenvolvimento de aplicações *Android* ser na linguagem *Java*, estas seguem um conjunto de definições e classes padrão próprios da API.

Assim, para o desenvolvimento de todas as vistas da aplicação foi necessário criar um conjunto de classes que estendessem da classe *activity*, como podemos ver na Figura 37. Estas actividades contêm os seus próprios métodos de manipulação, bem como *threads* que efectuem os pedidos à plataforma *SocialSense*.

De modo que o utilizador não permanecesse à espera de uma resposta com a *interface* bloqueada, todas as actividades efectuem os pedidos e aguardam a resposta a partir de uma *thread*. Além disso, estas actividades contêm um elemento da biblioteca chamado *Progressor* que efectua uma animação enquanto o pedido é feito. A *CommunicationClass* é a classe responsável por realizar todos os pedidos e aguardar a resposta.

Cluster

Como definido na especificação, o mapa devia de possuir um mecanismo de *cluster* para agrupar POIs quando ficasse com uma grande densidade. Dos vários algoritmos de *clustering* existentes o mais apropriado para os *smartphones* é *clustering* de grelha devido ao baixo poder de processamento. Deste modo a solução implementada é baseada numa grelha multidimensional, definida através da densidade de pixéis de cada ecrã do *smartphone* e pelo *zoom* do mapa. Assim, cada *POI's* do ecrã encontram-se dentro de uma célula, que se tiver mais do que um POI, transforma estes POIs num *cluster*. Os *clusters* vão ficando maiores na medida em que ao fazermos *zoom out*, convergem todos para uma única célula. As classes utilizadas para implementar este mecanismo foram assim o *GeoCluster*, *GeoBounds* e *GeoItems*.

Privacidade

Para implementar o mecanismo de privacidade associado ao ambiente foi necessário criar um listener *GeoLocationListener*, que detecta se um utilizador se encontra dentro ou fora de ambiente pervasivo. Ao entrar dentro de um ambiente pervasivo, a thread *GetPervasiveServices* pede à plataforma os serviços disponíveis naquele ambiente. Se existir algum serviço disponível naquele momento, são iniciados os listeners dos serviços correspondentes e enviados os dados para a base de dados.

7.3.3. Base de Dados

Apesar da base de dados existente da aplicação Avós e Netos, no início da especificação do projecto, achei necessário criar uma nova base de dados, de modo que cada projecto tivesse as suas entidades.

Assim, foi criado uma base de dados que permitisse representar os utilizadores, os locais, os eventos, o feedback e o serviço de privacidade, como podemos visualizar na Figura 38.

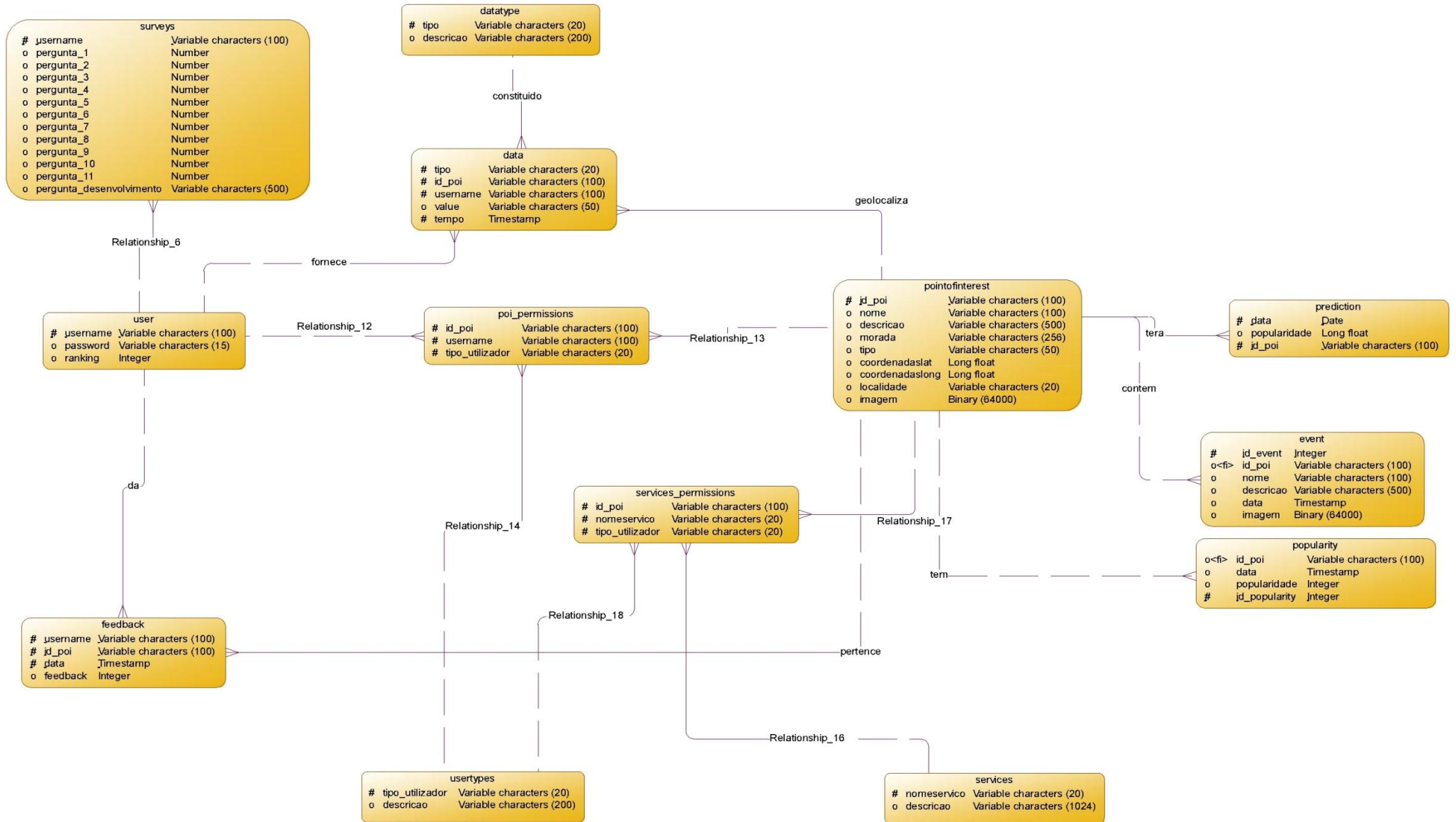


Figura 38 - Base de dados do sistema Youth5G

7.4. Aplicação e Demonstração

Depois de todo o processo apresentado anteriormente, ou seja, processo de especificação, requisitos e arquitectura segue-se a apresentação do produto final.

Como descrito anteriormente, o sistema *Youth5G* permite através de dados colaborativos a identificação de locais com uma maior afluência.

Assim para o utilizador poder utilizar a aplicação, deve em primeiro lugar instalar a aplicação no seu *smartphone* através do respectivo ficheiro *.apk*.

Depois da instalação, este inicia a aplicação através do item correspondente, sendo reencaminhado para o menu inicial Figura 39. A aplicação escolhe automaticamente o seu idioma através do idioma predefinido do sistema operativo *Android*, podendo ser Português ou Inglês.



Figura 39 - Menu Inicial da Aplicação *Android* Youth5G

7.4.1. Login e Registo

No menu inicial existem duas possibilidades, se o utilizador não tiver registado poderá efectuar um novo registo (Figura 40), caso contrário, pode efectuar o login e aceder à aplicação (Figura 41).

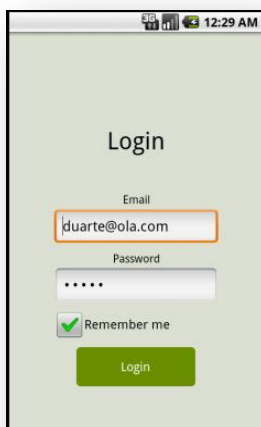


Figura 41 - Menu de Login da Aplicação *Android* Youth5G

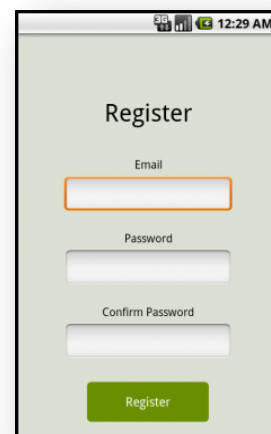


Figura 40 - Menu de Registo da Aplicação *Android* Youth5G

7.4.2. Mapa e Menu de Opções

Depois do utilizador efectuar a autenticação, é automaticamente redireccionado para o mapa, e serão carregados os respectivos *POIs* do local. Como podemos ver na Figura 42 a afluência dos locais é apresentada através de diferentes cores. Além disso, o mecanismo de *cluster* implementado permite uma melhor visualização dos locais com uma maior afluência.

Adicionalmente, é possível ter acesso a um conjunto de opções disponíveis carregando no botão de opções do *Android*. Neste conjunto de opções destacam-se as opções de mudar a *password*, mostrar *ranking*, realizar inquérito, mostrar ajuda e actualizar o mapa.

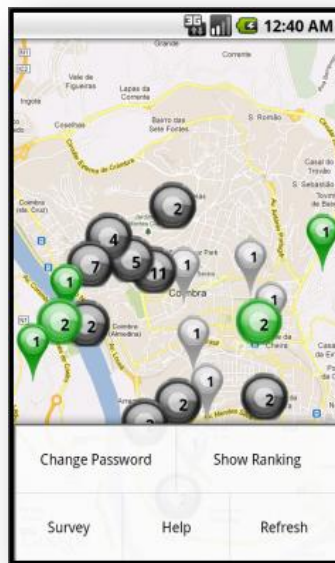


Figura 42 - Mapa e Menu de Opções da Aplicação *Android Youth5G*

7.4.3. Características do POI e Visualização de Eventos

Como descrito na especificação, quando o utilizador carrega num *POI* surge um menu (Figura 45) que permite visualizar eventos, visualizar características do *POI*, enviar *feedback* e ver a previsão da afluência para os próximos dias.

Ao escolher a opção de visualizar eventos, será mostrado ao utilizador uma lista com os diferentes eventos que irão ocorrer num determinado local. Adicionalmente, o utilizador também poderá escolher um determinado evento e visualizar detalhes sobre o evento (Figura 44).

Além disso, o utilizador pode também aceder aos detalhes de um determinado local, visualizando o nome, a morada, o tipo de estabelecimento, as coordenadas *gps*, a descrição e a foto (Figura 43). Nessa actividade, ao carregar no botão “*Go to POI*” o utilizador pode traçar uma rota do ponto onde se encontra até ao local do estabelecimento nocturno.

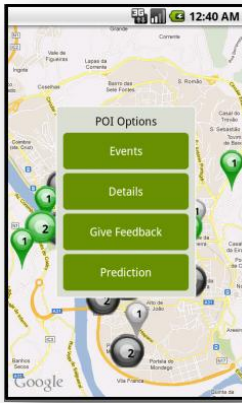


Figura 45 - Menu Principal da aplicação *Android Youth5G*



Figura 44 - Vista de detalhes de um evento da aplicação *Android Youth5G*

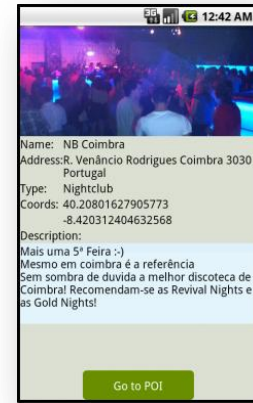


Figura 43 - Vista de detalhes de um POI da aplicação *Android Youth5G*

7.4.4. Sistema de Incentivo e Feedback e Previsões

Como foi mencionado anteriormente, a afluência de pessoas aos locais é calculado através de dados colaborativos, ou seja, cabe aos utilizadores do sistema enviarem o *feedback* sobre a afluência a um determinado local.

Assim, para o utilizador indicar a afluência de um local, acede ao menu principal carregando em cima do *POI*. Quando o menu abre, o utilizador escolhe a opção de submeter um *feedback*, aparecendo um novo menu com cinco opções (Figura 47).

Como os sistemas colaborativos necessitam que o utilizador tenha interesse em colaborar, foi criado um sistema de *ranking*, que beneficia os utilizadores que mais colaborarem. Através deste *ranking* o utilizador pode ganhar prémios e promoções disponibilizados pela Sapo.

Através dos dados submetidos pelos utilizadores, sobre os locais, o sistema contém um algoritmo que através do histórico calcula previsões para os dias seguintes. Para aceder a esta funcionalidade o utilizador apenas necessita de carregar no *POI*, esperar pelo menu e carregar na opção "*Prediction*".

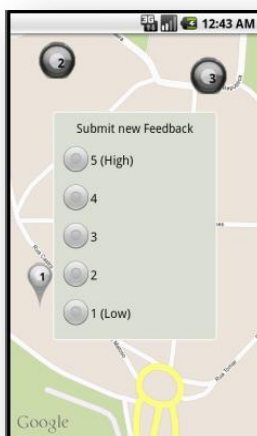


Figura 47 - Vista de envio de Feedback da aplicação *Android Youth5G*

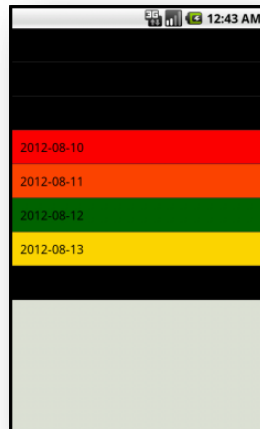


Figura 48 - Vista de Previsão da aplicação *Android Youth5G*



Figura 46 - Vista do Ranking da aplicação *Android Youth5G*

7.4.5. Inquérito aos Utilizadores

De modo a melhorar a aplicação no futuro dentro do projecto Sapo *Labs* foi criado uma funcionalidade que permite o utilizador realizar inquéritos. Os inquéritos são divididos em dois tipos de questões.

Em primeiro lugar, são apresentadas nove questões rápidas, onde a resposta é efectuada através de uma escala de um a cinco (Figura 50). Em segundo lugar, o utilizador pode deixar alguns comentários como melhorias e conselhos, através de um formulário disponibilizado (Figura 49).

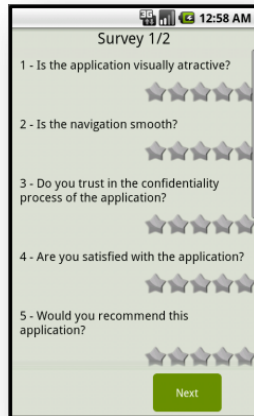


Figura 50 – Vista de inquéritos resposta rápida da aplicação *AndroidYouth5G*

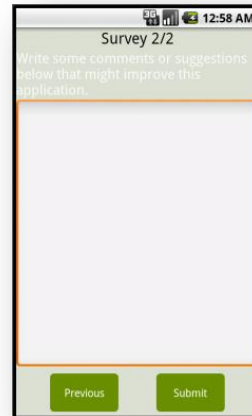


Figura 49 - Vista inquéritos comentário da aplicação *Android Youth5G*

7.4.6. Modelo de Privacidade

Como foi mencionado anteriormente, o colega do grupo de investigação Valderi Leithardt, propôs uma nova taxonomia de privacidade associada ao ambiente. Este modelo de privacidade foi assim implementado na aplicação gerindo a partilha de dados dos sensores, dentro de um ambiente como um bar, discoteca, etc. Como podemos ver na Figura 52 quando um utilizador entra num ambiente pervasivo, este indica os serviços disponíveis (Figura 51) e de seguida são enviados os dados.

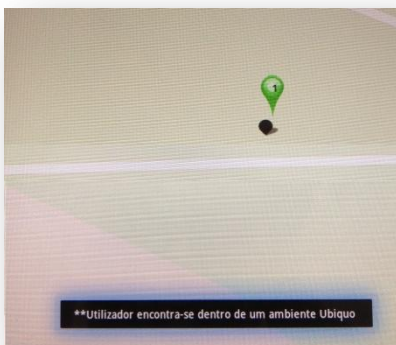


Figura 52 - Detecção do ambiente aplicação *Android Youth5G*

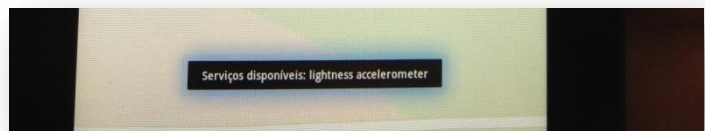


Figura 51 - Disponibilização dos serviços aplicação *Android Youth5G*

7.5. Principais Desafios

Ao longo do desenvolvimento desta aplicação existiram grandes desafios que tiveram que ser ultrapassados, para que a aplicação fosse construída e não falhasse os requisitos para o qual foi desenhada.

Assim, o principal desafio de todos foi a *API* disponibilizada pela *SapoLabs*. Como o projecto se encontravam dentro dos projectos Sapo, tínhamos ao dispor uma *API* que continham uma grande quantidade de informação geolocalizada ideal para a nossa aplicação.

Apesar de ter poucos conhecimentos com *webservices Soap*, desde muito cedo comecei a investigar a *API* e a tentar colmatar o risco do projecto falhar devido aos fracos conhecimento nesta área. Recorrendo à aplicação *SoapUI* comecei a efectuar alguns testes através dos ficheiros *WSDL* disponíveis no site.

Quando comecei a efectuar os primeiros testes apercebi-me que a documentação do *site*, estava desactualizada e que muitos dos serviços disponibilizados pela Sapo não se encontravam disponíveis.

Para tentar resolver a situação contactei o Dr. Pedro Torres que me reencaminhou o meu pedido para o *helpdesk* da Sapo. Apesar de todos os esforços nunca tive uma resposta concreta e nunca consegui colocar os *webservices* a funcionar, não por minha culpa, como o Dr. Benjamim Júnior e o Dr. Pedro Torres referiram numa reunião.

Para resolver esta situação foi assim necessário mudar a *API* do *SapoLabs* para a *API* do Foursquare.

7.6. Conclusão

O principal objectivo da elaboração deste projecto era o de construção de uma aplicação ubíqua que permitisse a identificação de eventos ou locais com maior afluência de forma colaborativa. Como podemos observar no capítulo seguinte o projecto foi finalizado com todas as funcionalidades especificadas.

No final do projecto os orientadores *SapoLabs*, o Dr. Benjamim Júnior e o Dr. Pedro Torres satisfeitos com o trabalho desenvolvido, apesar das dificuldades existentes e do tamanho do projecto, convidaram-nos para elaborar uma nova proposta conjunta.

Esta proposta conjunta aproveitou a aplicação criada através deste projecto mas em um contexto diferente, como descrito no Anexo E.

Apesar de ser uma proposta conjunta e com uma ideia inicial dos próprios orientadores *SapoLabs* conseguimos novamente um projecto na categoria três.

Por outro lado, o desenvolvimento deste projecto e implementação dos mecanismos de privacidade levou à elaboração de um artigo em conjunto com colegas da Universidade Federal do Rio Grande do Sul, que se encontra em aprovação. Este trabalho encontra-se descrito no Capítulo 9.

Capítulo 8

Testes

Como podemos verificar no passado [32] alguns defeitos no desenvolvimento de projectos tiveram no final custos elevados. Devido a falhas como estas, a fase de testes tem ganho de dia para dia uma maior importância no desenvolvimento de projectos de *software*. Este esforço visa aumentar a qualidade do produto e prevenir falhas irreparáveis durante o processo de desenvolvimento.

A fase de testes de um projecto de *software* é normalmente dividida em três níveis de testes constituídos por: testes unitários, testes de integração e testes de sistema.

Os testes unitários são o processo de teste dos componentes de um projecto de *software*, ou seja, nesta fase todas as funções, classes e atributos são testadas exaustivamente. Esta fase tem como vantagem ser muito eficaz pois revela possíveis defeitos no produto final e tem como desvantagem o “tempo”, pois é uma tarefa muito exaustiva.

Depois dos componentes individuais serem testados é fundamental verificar se eles funcionam como um todo, ou seja, como um sistema. Nesta fase espera-se que os componentes sejam testados em termos da sua interacção entre uns e outros.

Finalmente, depois dos componentes estarem integrados, segue-se os testes de sistema, que tem como finalidade testar o sistema em termos de requisitos. Nesta fase espera-se que o sistema cumpra todos os requisitos para o qual foi redesenhado inicialmente. Normalmente nesta fase são efectuados testes funcionais, testes de performance, etc.

Seguidamente, será apresentado a plano de testes e resultados obtidos dos dois projectos *SocialSense* Avós e Netos e *Youth5G*.

8.1. Objectivos dos Testes

Este plano de teste descreve o ambiente utilizado em cada teste, assim como, métricas, metas e ferramentas utilizadas na fase de teste dos dois projectos *SocialSense*.

Apesar das três fases de testes referidas em cima, o plano de testes efectuado apenas contempla os testes de sistema. Devido a serem muito exaustivos e terem um custo de tempo muito elevado, os testes unitários e testes de integração seriam impossíveis de realizar dentro do tempo previsto nos dois projectos *SocialSense*. Assim, os objectivos ao testar o sistema são:

- Garantir que os requisitos se encontram bem especificados, através de testes de requisitos;
- Garantir que todas as funcionalidades requeridas estão implementadas e a funcionar correctamente de acordo com os requisitos, através de testes funcionais;
- Garantir que o *software* cumpre os requisitos de performance e confiabilidade, através de testes de performance.

Como foi dito anteriormente, quanto mais cedo uma falha é detectada no ciclo de desenvolvimento menor será o custo da falha para o projecto. Os requisitos são a base da arquitectura do sistema que depois de “programada” chegará a um produto final. Assim,

depois da elaboração do documento de requisitos foi feito um pequeno teste nos dois projectos, de modo a verificar se os requisitos estavam descritos de um modo claro e objectivo, para que não existisse nenhum problema na fase de arquitectura do sistema.

A análise dos requisitos foi efectuada com base nestes parâmetros:

- Claro e Objectivo – O requisito é claro e o mais curto possível?
- Atómico – O requisito pode ser dividido em mais requisitos?
- Verificabilidade – O requisito pode ser testado?
- Priorizado – O requisito tem uma prioridade adequada?
- Válido - Se o requisito se encontra válido para seguir para arquitectura.

A performance do sistema apesar de não ser um requisito funcional tem uma grande importância. Quando um utilizador utiliza o sistema, ele não vai querer esperar muito tempo que uma certa funcionalidade esteja disponível, desistindo de usar o sistema se este demorar demasiado tempo a responder.

Relativamente à performance dos dois sistemas implementados os objectivos do teste são:

- Identificar possíveis perdas de mensagens;
- Identificar o volume de dados transmitidos em cada *webservice*;
- Identificar o tempo de resposta de cada *webservice* com um número diferente de utilizadores a aceder ao sistema.

Por último, os sistemas foram testados como um todo nos testes de sistema. Os testes que foram efectuados tiveram como base os requisitos especificados na fase inicial dos projectos. Para verificar se o produto cumpria os requisitos especificados, foram criados diversos testes que asseguraram a aprovação do requisito. Para que os testes validassem os requisitos correctamente foram criadas um conjunto de pré-condições, instruções e pós condições, que definem os parâmetros de teste e o resultado esperado do teste.

8.2. Projecto Avós e Netos

8.2.1. Testes de Requisitos

Em seguida, encontra-se o teste efectuado aos requisitos no projecto *SocialSense* Avós e Netos:

NºRequisito	Válido	Claro	Atómico	Prioritizado	Verificabilidade
ID-01					
ID-03					
ID-04					
ID-05					
ID-06					
ID-07					
ID-08					
ID-09					
ID-10					

Tabela 2- Testes de Requisitos Avós e Netos

Ao efectuar os testes segundo os parâmetros descritos anteriormente verificou-se que alguns requisitos necessitavam de ser revistos.

O requisito ID-01 não estava claro pois descrevia em demasia o processo de comunicação entre o *Facebook* e o servidor e não as funcionalidades em si. Seguiu-se o requisito ID-03 que se encontrava pouco objectivo, contendo muita informação e podendo ser expandido em mais casos de uso. Finalmente o requisito ID-07 também necessitava de ser clarificado, pois continha demasiadas funcionalidades.

8.2.2. Testes de Performance

Métricas e Cenário

Como foi referido previamente os testes de performance pretendem avaliar as seguintes métricas:

- Perda de mensagens;
- Volume de dados transmitidos;

- Tempo de resposta dos *webservices*;

Para conseguir testar o sistema com base nas métricas definidas foi necessário configurar um ambiente de testes como o representado:

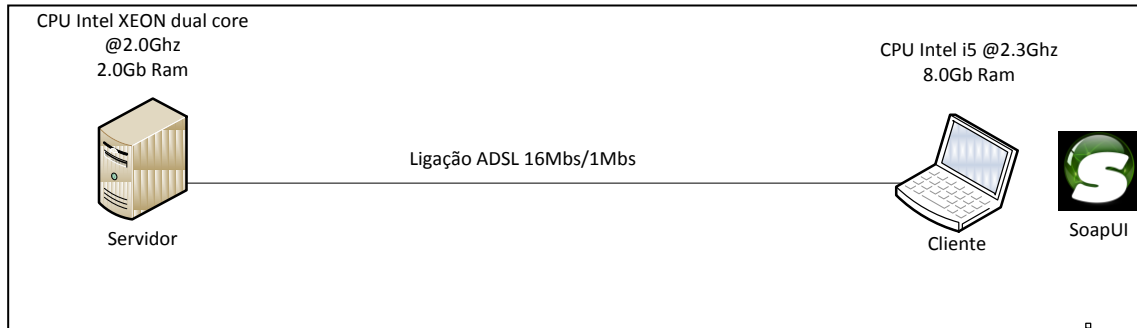


Figura 53 - Cenário de Testes Avós e Netos

Como podemos ver na imagem o simulador escolhido para a aplicação Avós e Netos a efectuar pedidos foi o *SoapUI*. O *SoapUI* é uma ferramenta de testes de *webservices*, que na sua nova versão possui suporte a *webservices REST*. Logo, para importar os pedidos dos *webservices* do sistema Avós e Netos foi gerado um ficheiro WADL através do seguinte link <http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/application.wadl>. Para realizar a captura de mensagens JSON vindas da aplicação *Facebook* para colocar na aplicação SoapUI, foi utilizado o *plugin Firebug* para o navegador *Firefox*.

Configurado o ambiente, foi necessário definir os restantes parâmetros do teste de simulação no SoapUI:

- Tempo de simulação: 60s;
- Número de Utilizadores: 5, 10, 30;
- Número de simulações: 3;
- Atraso do Teste: 1000;
- Aleatoriedade: 0.5;
- Estratégia: Simples.

Resultados

Seguidamente serão apresentados os resultados dos testes de *performance* efectuados.

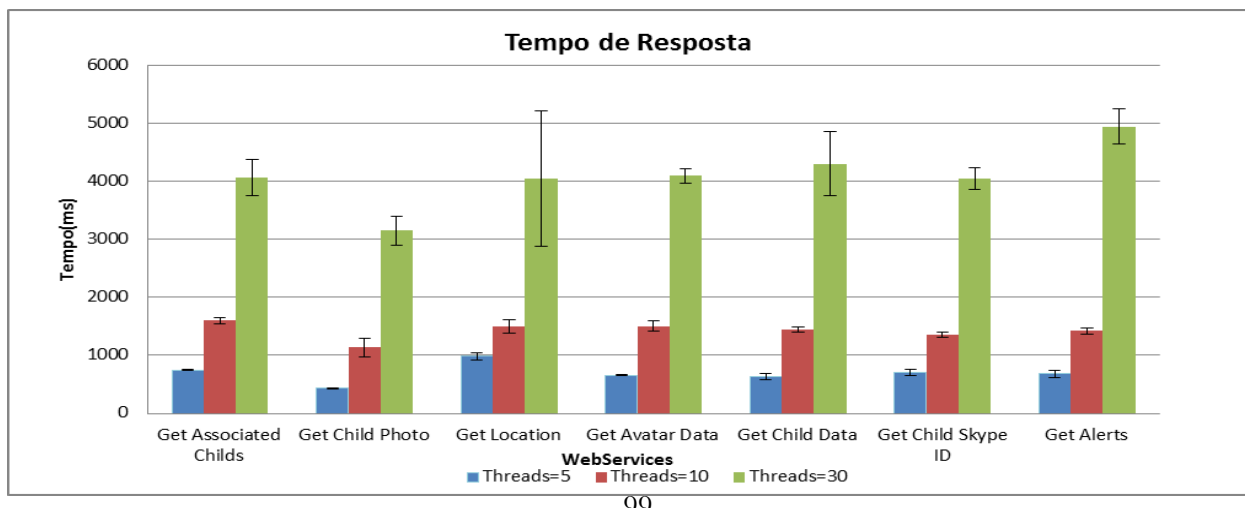


Gráfico 2 - Tempo de Resposta dos Webservices do Sistema Avós e Netos

Como podemos verificar no “Gráfico 2” o tempo de resposta de cada *webservice* aumentou consideravelmente com o número de pedidos em simultâneo. Por um lado, podemos visualizar que com cinco pedidos em simultâneo os *webservices* tiveram um tempo de resposta abaixo de um segundo. Por outro lado, com trinta pedidos em simultâneo o tempo aumentou para 5 segundos (máximo). Podemos também verificar que os tempos de resposta dos *webservices* são muito semelhantes, existindo diferenças pequenas.

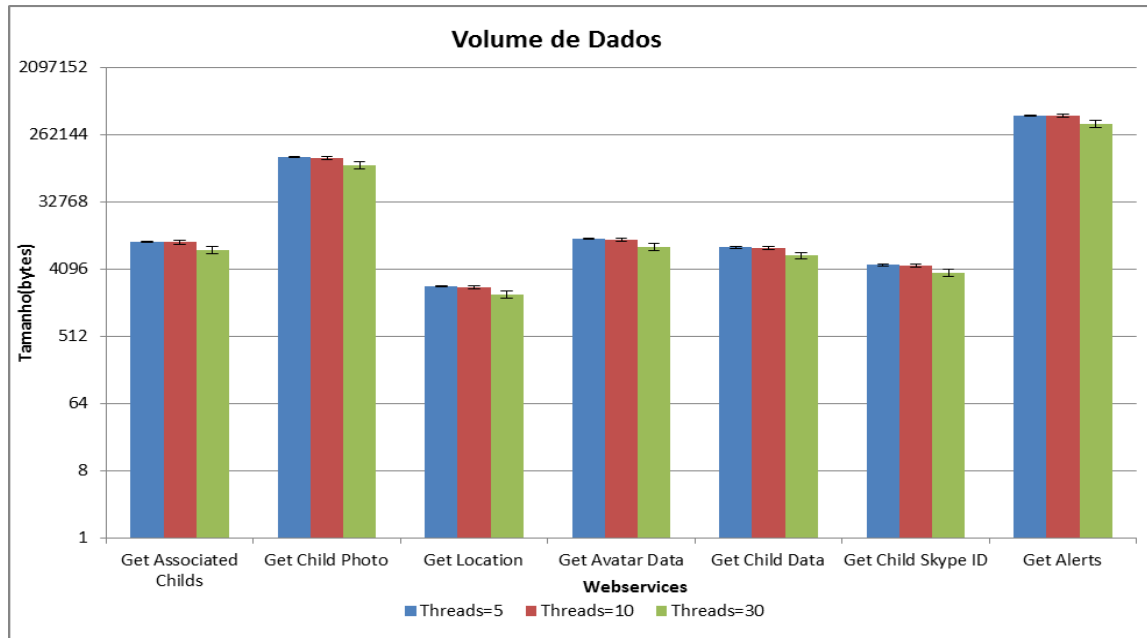


Gráfico 3 - Volume de Dados dos Webservices do Sistema Avós e Netos

O Gráfico 3 representa a quantidade de dados transmitidos por cada *webservice*. Como podemos observar os *webservices* que enviaram mais dados são o Get Child Photo e Get Alerts. Isto deve-se ao facto do primeiro transmitir as fotografias que têm um tamanho considerável e o segundo transmitir a lista de todos os alertas de uma criança. Através deste gráfico podemos verificar que apesar do volume de dados influenciar o tempo de resposta não é um factor directo.

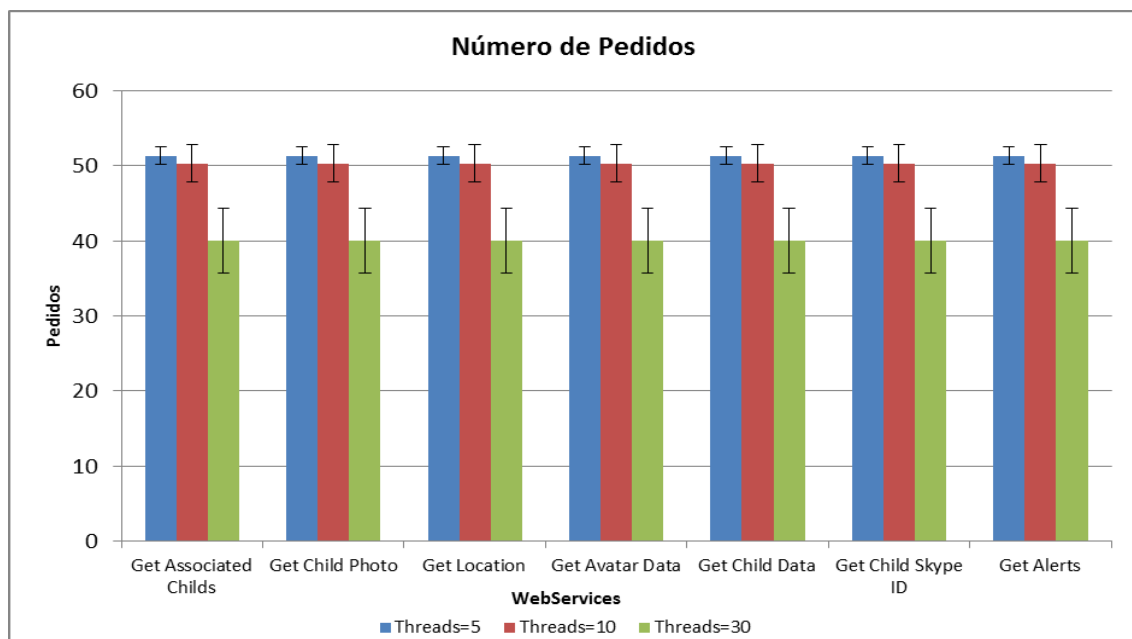


Gráfico 4 - Número de Pedidos Respondidos dos Webservices Avós e Netos

O último teste apresenta o número de mensagens tratadas por cada *webservice*. Como podemos visualizar quanto mais pedidos existem menos mensagens o sistema consegue responder no mesmo espaço de tempo. Indica assim uma tendência que com um maior número de pedidos, irão começar a ser descartados pacotes por *timeout*. Outra observação que podemos retirar deste teste é que até trinta utilizadores em simultâneo não existiu nenhuma perda de pacotes.

Em suma, podemos concluir que em termos de performance e confiabilidade o sistema está preparado para suportar trinta utilizadores em simultâneo, com o cenário implementado, garantindo respostas num tempo inferior a 5 segundos.

8.2.3. Testes de Funcionais

Na página seguinte é apresentada uma tabela que mostra os testes efectuados aos diferentes requisitos com os respectivos resultados dos testes funcionais.

Os requisitos que falharam nos testes funcionais dizem respeito aos requisitos funcionais SHOULD que não foram implementados como o requisito ID-04 de visualizar as mensagens e enviar mensagens do avô para a criança. Todos os restantes requisitos funcionais MUST foram implementados e passaram os testes funcionais, apresentando o comportamento esperado.

Nº	Requisito	Descrição	Pré-Condições	Instruções	Pós-Condições	Resultado
1	ID-01	Verificação se o Login é efectuado correctamente com um utilizador registado.	O utilizador já se encontra registado na rede social.	Entrar no Facebook com a conta chogathazard@gmail.com e password lctwsn	A aplicação inicia e vai até ao menu principal.	
2	ID-01	Verificação se o Login é efectuado correctamente com um utilizador não registado.	O utilizador não se encontra registado na rede social.	Entrar no Facebook com a conta draposo@gmail.com password 12345	A aplicação tem de apresentar novamente o menu de login.	
3	ID-03	Verificação dos botões do menu.	O utilizador encontra-se na página inicial. Adicionar crianças na base de dados.	Devem ser accionados todos os botões do Menu. Deve ser possível as crianças nas opções de actividade e videochamada.	O botão deve reencaminhar o botão para a respectiva página.	
4	ID-03	Verificação dos controlos do Menu.	O utilizador encontra-se na página inicial.	O menu deve poder ser minimizado e restaurado carregando no botão de minimização.	O menu minimiza e maximiza.	
5	ID-04	Visualizar Mensagens.	Devem existir mensagens na base de dados do servidor.	Devem poder ser seleccionadas mensagens. A página deve conter o menu de scroll.	As mensagens devem ser exibidas correctamente.	
6	ID-04	Visualização dos dados da criança.	Deve existir uma criança na base de dados com dados pessoais.	Ao abrir a página deve ser mostrado na parte superior a fotografia e nome da criança.	Os dados sobre a criança estarem visíveis na página.	
7	ID-04	Envio de Mensagens	Existir um número de telemóvel configurado na base de dados.	Entrar na página de envio de mensagens e enviar uma mensagem para a criança seleccionada.	A mensagem chegar ao telemóvel.	
8	ID-05	Visualização de alarmes da criança.	Devem existir alarmes na base de dados (alarmes de queda e alarmes de videoconferência)	Entrar na página de alarmes. Verificar se os alarmes se encontram na tabela. Verificar se existem diferentes tipos de alarmes.	Devem aparecer vários alarmes através de uma lista.	
9	ID-06	Testar Funcionamento Videochamada	A aplicação skype tem de estar instalada e configurada no sistema. A criança tem que ter um id skype na base de dados.	Entrar na página de videochamada e carregar no botão verde para chamar.	A chamada deve iniciar e começar a chamar.	
10	ID-07	Visualização da Actividade	Deve estar ligado o sistema de actividade.	Entrar na página e visualizar os dados de actividade na tabela e o movimento do avatar no mapa.	A tabela deve conter dados de actividade. Se o sistema de actividade estiver a detectar movimento, o avatar deve conter movimento.	
11	ID-07	Visualização da Localização e Mapa	Deve estar ligado o respectivo sistema de localização.	Entra na página de actividade e visualizar o mapa.	O mapa deve apresentar um avatar que se desloca como o utilizador dentro do edifício.	
12	ID-07	Visualização dos dados dos Sensores Médicos	Têm de existir dados de sensores médicos na base de dados.	Entrar na página de actividade e visualizar a tabela com os dados fisiológicos.	A tabela deve apresentar os dados do Oxímetro, Esfigmomanómetro, Espirómetro, Temperatura.	
13	ID-07	Visualização dos dados pessoais	Os dados pessoais da criança tem que se encontrar na base de dados.	Entrar na página de actividade e visualizar a tabela com os dados pessoais.	A tabela tem de conter o nome, idade e nacionalidade.	
14	ID-08	Verificação dos Alertas	Deve estar activo o sistema de actividade.	Carregar no botão do sensor Zolértia e visualizar a página de alarmes.	Na página de alarmes deve aparecer um novo alarme.	
15	ID-09	Verificação e Envio de um sms para a aplicação.	Deve estar inserido o número de telemóvel da criança na base de dados.	Enviar uma mensagem através do telemóvel da criança.	Deve aparecer uma nova mensagem na página de mensagens.	
16	ID-10	Iniciar Videochamada através de proximidade.	Deve estar inserido o id skype do avô na base de dados. O sistema de localização tem de estar activo.	Aproximar do computador da instituição.	Passado uns 5 segundos o computador começa a fazer a chamada.	

Tabela 3 - Testes Funcionais Avós e Netos

8.3. Projecto Youth5G

8.3.1. Testes de Requisitos

Os testes de requisitos efectuados ao projecto Youth5G encontram-se na tabela seguinte:

NºRequisito	Válido	Claro	Atómico	Prioritizado	Verificabilidade
ID-1.1.1	✓	✓	✓	✓	✓
ID-1.1.2	✓	✓	✓	✓	✓
ID-1.2.1	✓	✓	✓	✓	✓
ID-1.2.2	✓	✓	✓	✓	✓
ID-1.2.3	✓	✗	✓	✓	✓
ID-1.2.4	✓	✓	✓	✓	✓
ID-1.3.1	✓	✗	✓	✓	✓
ID-1.3.2	✓	✓	✓	✓	✓
ID-1.3.3	✓	✓	✓	✓	✓
ID-1.4	✓	✗	✓	✓	✓

Tabela 4 - Testes de Requisitos Youth5G

Na realização nos testes de requisitos encontrou-se algumas falhas na especificação dos requisitos.

Em primeiro lugar o requisito com o ID-1.2.3 não se encontrava claro ao descrever o modo como o utilizador interage, em vez disso, o caso de uso centrava-se mais em cálculos do que na interacção. Em segundo lugar, o requisito com o ID-1.3.1 encontra-se ambíguo, não especificando qual será o local onde a funcionalidade deverá estar inserida, dando a percepção de estar no menu inicial e não no menu do mapa. Por último, o requisito ID-1.4 apesar de especificar correctamente todas as funcionalidades encontra-se pouco objectivo.

8.3.2. Testes de Performance

Métricas e Cenário

À semelhança do projecto *SocialSense* Avós e Netos as métricas utilizadas para avaliar o sistema foram:

- Perda de mensagens;
- Volume de dados transmitidos;
- Tempo de resposta dos *webservices*;

Apesar do cenário de testes partilhar a mesma estrutura os elementos físicos são diferentes do projecto anterior, mantendo a aplicação de testes SoapUI.

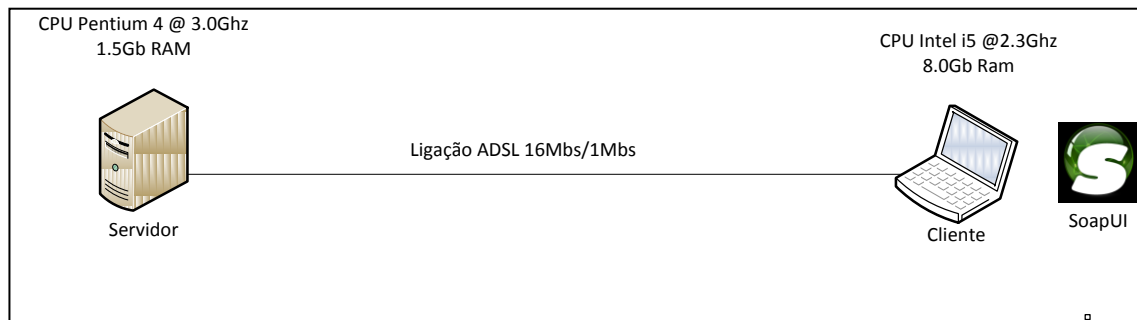


Figura 54 - Cenário de Testes Youth5G

Para importar os diferentes *webservices* para a aplicação *SoapUI* foi utilizado um ficheiro WADL gerado através do seguinte link <http://evfinder.dei.uc.pt:8080/youth5gserver/rest/application.wadl>. Para gerar as mensagens para o teste foi utilizado a ferramenta *wireshark* para efectuar a captura de pacotes entre a aplicação *Android* e o servidor.

Depois de configurar o ambiente, foi necessário definir os parâmetros de teste na aplicação SoapUI:

- Tempo de simulação: 120s;
- Número de Utilizadores: 3, 5, 10, 30;
- Número de simulações: 3;
- Atraso do Teste: 1000;
- Aleatoriedade: 0.5;
- Estratégia: Simples.

Resultados

Como podemos observar no gráfico seguinte (Gráfico 6), o tempo de resposta de cada *webservice* aumentou como era de esperado. No entanto existiram *webservices* que tiveram um aumento notório como é o caso do *webservice getlistpoi*. Este *webservice* teve este aumento exponencial porque depende de um pedido interno à API do *Foursquare*. Um melhoramento futuro pode ser efectuado de modo a rever o modo como o sistema efectua os pedidos ao *Foursquare*. É de referir que este teste mostra que o sistema implementado cumpre o requisito de confiabilidade inicialmente descrito, ou seja, responder a pedidos de três a cinco utilizadores num tempo máximo de dez segundos.

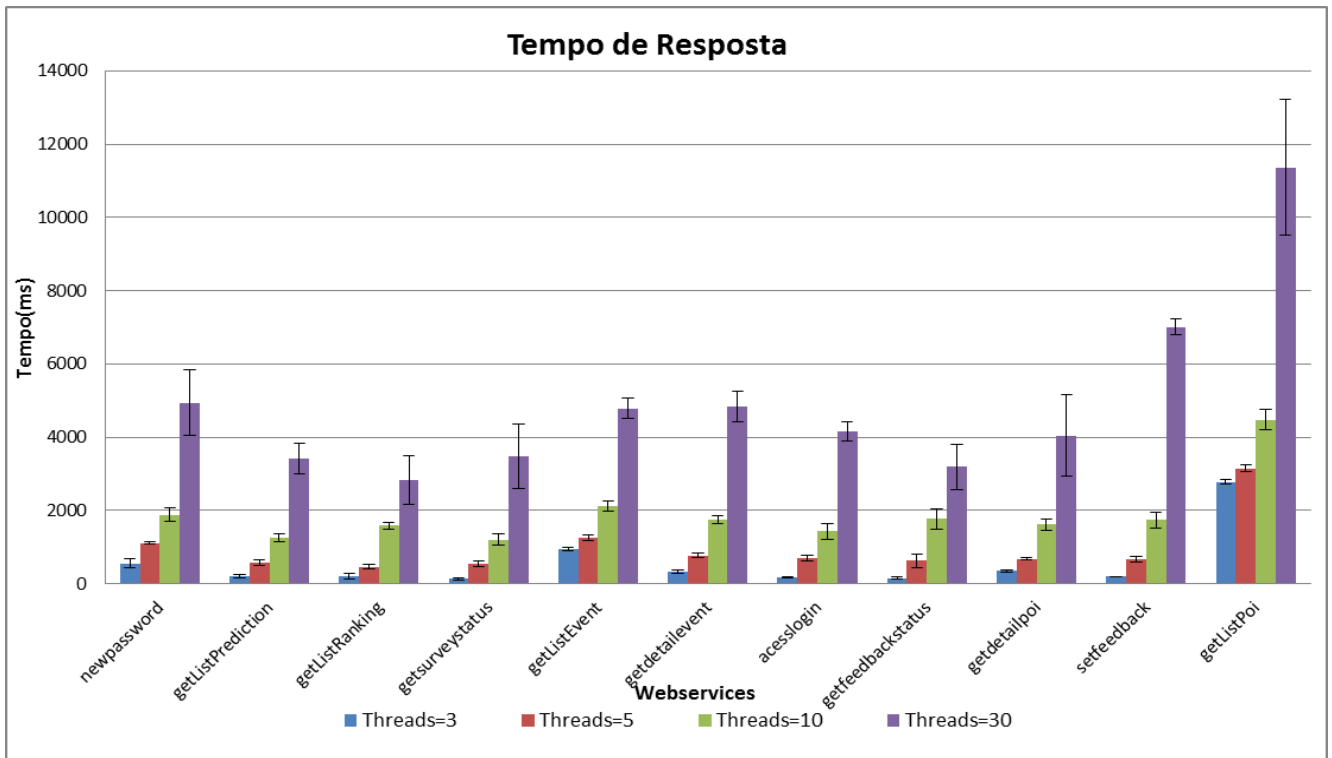


Gráfico 6 - Tempo de Resposta dos Webservices do Sistema Youth5G

Relativamente ao volume de dados, podemos observar (Gráfico 5) que com o aumentar do número de utilizadores em simultâneo existe um pequeno decréscimo na quantidade de dados enviados. Este decréscimo deve-se ao aumento do tempo de resposta de cada *webservice* como vimos no gráfico anterior. Particularmente confirma-se que o aumento do tempo de resposta do *webservice* *getListpoi*, não se deve ao volume de dados transmitidos mas sim a um factor interno.

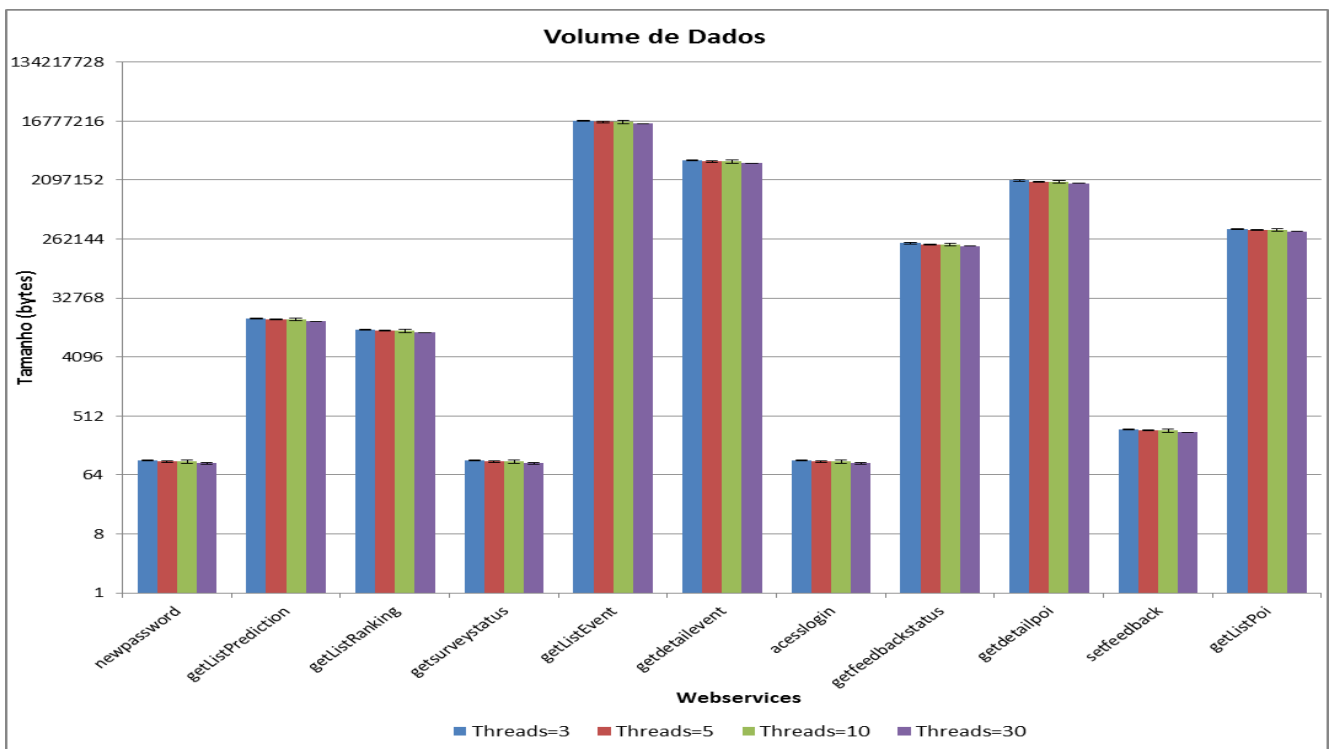


Gráfico 5 - Volume de Dados dos Webservices do Sistema Youth5G

O gráfico que se segue (Gráfico 7) representa o número de pedidos que foram respondidos pelos *webservices* do servidor *Youth5G*. Como podemos observar não existiram perdas de pacotes nas transacções ocorridas por *timeout* até trinta utilizadores em simultâneo. Por outro lado, através do gráfico consegue-se observar uma perda de capacidade de resposta no mesmo espaço de tempo (120 segundos).

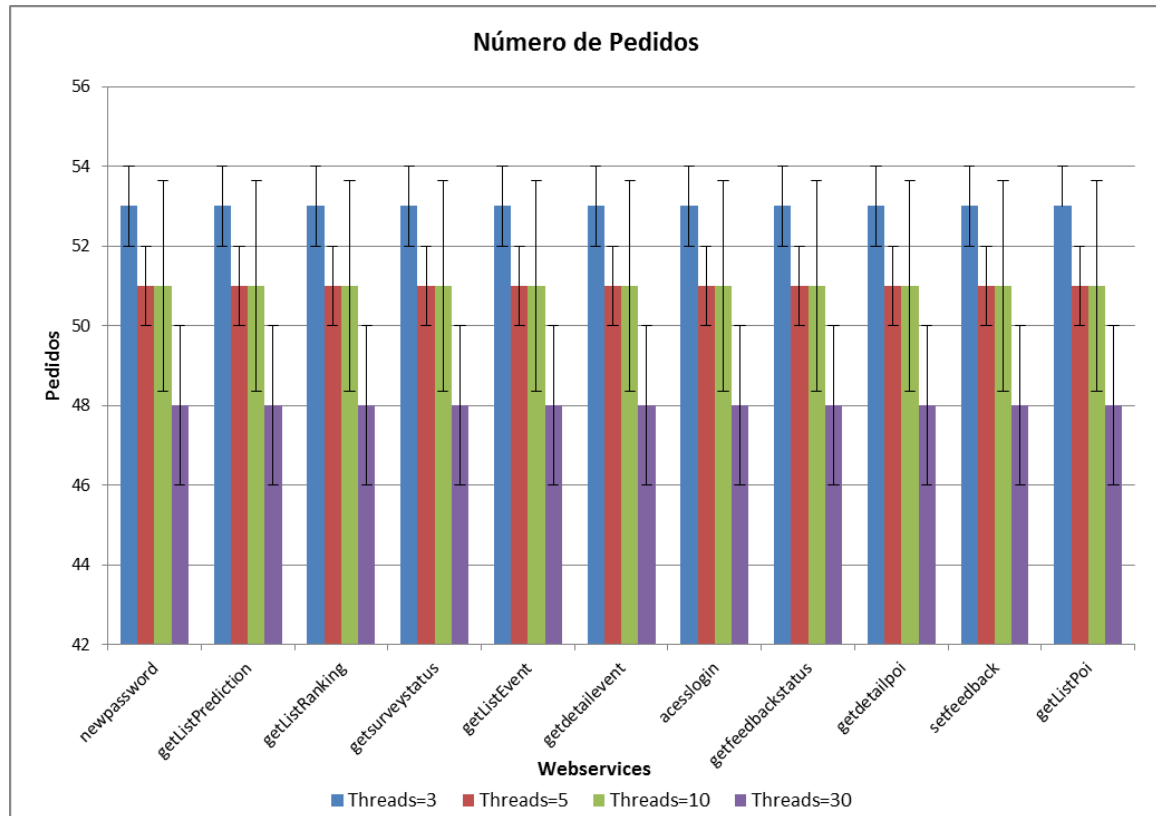















Gráfico 7 - Número de Pedidos Respondidos dos Webservices Youth5G

Pode-se assim concluir que o sistema consegue uma aprovação positiva em termos de performance e confiabilidade, apenas com algumas notas de melhoramentos quanto ao funcionamento interno de alguns *webservices*.

8.3.3. Testes Funcionais

Apesar do sistema *Youth5G* ter uma maior complexidade e número de requisitos, todos os requisitos funcionais propostos inicialmente na especificação passaram nos testes funcionais como podemos verificar na tabela que se encontra na página seguinte.

	Descrição	Pré-Condições	Instruções	Pós-Condições	Resultado
ID-1.1.1	Verificação se o Login é efectuado correctamente com um utilizador registado.	O utilizador de teste já tem de se encontra registado.	Entrar na aplicação Youth5G, aceder à actividade de login e realizar o teste com o email duarte@ola.com e password 123456.	A aplicação inicia e vai até ao menu principal mostrando o mapa.	
ID-1.1.1	Verificação se o Login é efectuado correctamente com um utilizador não registado.	O utilizador de teste não se pode encontrar registado na base de dados.	Entrar na aplicação Youth5G, aceder à actividade de login e realizar o teste com o email duarte.raposo@sapo.pt e password 123456.	A aplicação tem de apresentar novamente o menu de login.	
ID-1.1.1	Tentativa de Login sem ligação de dados.	Desligar a ligação de dados.	Entrar na aplicação Youth5G, aceder à actividade de login e realizar o teste com o email duarte@ola.com e password 123456.	A aplicação deve apresentar uma mensagem a informar que existe um erro na ligação.	
ID-1.1.2	Registo de um novo utilizador.	Não deve de existir nenhum utilizador na base de dados.	Entrar na aplicação Youth5G ,aceder à actividade de registo e realizar o registo com o email duarte2@ola.com	O botão deve reencaminhar o botão para a respectiva página.	
ID-1.1.2	Registo de um utilizador com um email existente.	Registar um utilizador com o email duarte@ola.com e pass 123456	Entrar na aplicação Youth5G, aceder à actividade de registo e realizar o registo com o email duarte@ola.com e pass 123456	Deve aparecer uma mensagem a dizer que o utilizador não existe.	
ID-1.2.1	Envio de um Feedback	Tem de existir pelo menos um POI no mapa, para poder efectuar a operação.	Seleccionar a opção de enviar feedback e submeter uma pontuação com o valor 2.	Deve aparecer uma mensagem a dizer que um novo feedback foi submetido.	
ID-1.2.1	Actualização de um Feedback	Tem de existir pelo menos um POI no mapa e uma submissão anterior, para poder efectuar a operação.	Seleccionar a opção de enviar feedback e mudar a opção já seleccionada com o valor 2 para o valor 3.	Deve aparecer uma mensagem a dizer que o feedback foi actualizado.	
ID-1.2.2	Verificação da lista de Eventos.	Tem de ser seleccionado um POI que contenha eventos disponíveis.	Seleccionar um POI no mapa e de seguida seleccionar a opção de visualizar eventos.	Deve ser apresentado uma actividade com uma lista de eventos.	
ID-1.2.3	Visualizar as previsões de afluência.	Têm de ser inseridos feedbacks na base de dados, para que o servidor possa calcular a afluência a um determinado local.	Seleccionar um POI no mapa e de seguida seleccionar a opção de previsões.	Deve ser apresentada uma lista com diferentes cores, indicando a afluência nos diferentes dias.	
ID-1.2.4	Visualizar as características de um POI.	Para realizar o teste é necessário que esteja pelo menos um POI representado no mapa.	Seleccionar o POI e de seguida seleccionar a opção de detalhes.	Deve ser apresentada uma nova actividade onde seja apresentado uma fotografia sobre o local, nome, morada, coordenadas gps e descrição.	
ID-1.2.4	Traçar rota até ao destino	Para realizar o teste é necessário que esteja pelo menos um POI representado no mapa.	Seleccionar o POI , seleccionar a opção de detalhes e carregar no botão de traçar rota.	Deve ser apresentada uma rota no mapa, desde o ponto onde o utilizador se encontra até ao local do POI.	
ID-1.3.1	Alterar Password	Para realizar o teste é necessário que o utilizador tenha feito login e se encontre na actividade do mapa.	Carregar no botão de opções do telemovel e seleccionar a operação dealterar a password. De seguida, preencher o campo com a password antiga e colocar a nova password nos campos seguintes.	Dever ser apresentada uma mensagem de confirmação a dizer que a password foi correctamente alterada.	
ID-1.3.2	Visualizar Ranking	Para realizar este teste é necessário criar alguns utilizadores e submeter alguns feedbacks relativamente a alguns pontos de interesse.	Encontrar-se na actividade do mapa e carregar no botão de opções do telemóvel seleccionando a opção de visualizar ranking.	Deve aparecer uma nova actividade, contendo a pontuação do utilizador e a posição do utilizador relativamente aos outros utilizadores através de uma lista.	






ID-1.3.3	Realizar Inquérito	Nenhuma.	Encontrar-se na actividade do mapa e carregar no botão de opções do telemóvel seleccionando a opção de realizar inquérito. De seguida responder aos grupo de respostas rápidas e deixar um comentário.	Deve aparecer uma nova actividade, com um primeiro grupo de perguntas de resposta rápida. A escala da resposta varia entre uma e cinco estrelas. No final encontra-se uma caixa de texto para o utilizador poder realizar um comentário. Depois de submeter o inquérito deverá ser apresentada uma mensagem de sucesso.	
ID-1.3.3	Testar a submissão do inquérito	Ter realizado um inquérito anteriormente.	Encontrar-se na actividade do mapa e carregar no botão de opções do telemóvel seleccionando a opção de realizar inquérito.	Deverá ser apresentada uma mensagem a informar que o inquérito já foi submetido.	
ID-1.4	Fazer Zoom	Nenhuma.	Encontrar-se na actividade do mapa e carregar nas setas de "menos" e "mais" que aparecem na parte inferior do mapa.	O mapa deverá realizar o zoom nos dois sentidos. Ao mesmo tempo devem ser redenhados os novos clusters.	
ID-1.4	Mover Mapa	Nenhuma.	Encontrar-se na actividade carregar no mapa e arrastar o dedo nas diversas direcções.	O mapara deverá actualizar o seu centro e mostrar os novos locais.	
ID-1.4	Abrir Cluster	Existir um local com vários POI's agrupados num cluster.	Carregar em cima do cluster.	Ao carregar em cima do cluster o mapa deve aproximar-se do local, diminuindo a escala.	

Tabela 5 - Testes Funcionais Youth5G

Capítulo 9

Outros Trabalhos

Durante o corrente estágio dissertação, paralelamente ao trabalho desenvolvido no estágio, foi elaborada duas propostas no âmbito do concurso *SapoLabs* e efectuadas várias publicações.

A primeira proposta do concurso *SapoLabs* foi aprovada na terceira categoria, mais especificamente, sem suporte financeiro mas com possibilidade de utilização das APIs do sapo, tendo início em Janeiro com o projecto *Youth5G*.

A segunda proposta *SapoLabs*, foi indicada através dos orientadores Sapo, devido aos bons resultados do projecto *Youth5G*. Esta proposta resultou na continuação sem suporte financeiro mas com possibilidade de utilização dos recursos Sapo.

Como resultado da primeira candidatura *Sapo*, na nossa primeira proposta foi efectuada a publicação de um artigo através da revista “<http://www.mdpi.com/journal/sensors>”, consistindo na apresentação da nossa plataforma baseada em *RESTful Web Services*, onde este estágio contribuiu no desenvolvimento da plataforma. Esta publicação foi aprovada.

Finalmente foi efectuada uma publicação para um jornal “<http://www.springer.com/computer/hci/journal/779>” com o artigo “*Quietly: A model privacy for pervasive and ubiquitous environments*”, onde foi apresentado o sistema *Youth5G* (desenvolvido por mim) através do modelo de privacidade proposto pelo colega de investigação Valderi Leithard. Este artigo encontra-se à espera de ser aprovado.

Referências

- [1] Nuno de Noronha. (22, July) Publico. [Online]. http://www.publico.pt/Tecnologia/facebook-ja-tem-mais-de-500-milhoes-de-utilizadores_1448243
- [2] (1993, March) Some Computer Science Issues in Ubiquitous Computing. [Online]. <http://www.ubiq.com/hypertext/weiser/UbiCACM.html>
- [3] Viktor K Prasanna Bhaskar, Bhaskar Krishnamachari Yang Yu, *Information Processing and Routing in Wireless Sensor Networks.*: World Scientific Publishing, 2006.
- [4] Sarah Clark. (2011, Nov.) NFC goes green: New ST chips use energy harvesting to replace the need for batteries. [Online]. <http://www.nfcworld.com/2011/11/08/311126/nfc-goes-green-new-st-chips-use-energy-harvesting-to-replace-the-need-for-batteries/>
- [5] ZigBee Alliance. ZigBee Specification Overview . [Online]. <http://www.zigbee.org/Specifications/ZigBee/Overview.aspx>
- [6] The BB and Iphone Zone. The World's First Smartphone – IBM's Simon. [Online]. <http://bb-iphone-zone.com/2011/the-worlds-first-smartphone-ibms-simon/>
- [7] Google. Android API Levels. [Online]. <http://developer.android.com/guide/appendix/api-levels.html>
- [8] Fabio Jordão. (2011, July) ARM versus x86. [Online]. <http://www.tecmundo.com.br/11914-arm-versus-x86.htm>
- [9] Emiliano Miluzzo, Nicholas D.Lane, and Kristóf Fodor. (2008) Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application.
- [10] Virginia Tech. e-textiles. [Online]. <http://www.ccm.ece.vt.edu/etextiles/>
- [11] Luiz. (2011, July) Como funciona o Wii Remote. [Online]. <http://vidageek.net/2009/07/01/como-funciona-o-wii-remote/>
- [12] Andre Yoskowitz. (2010, May) Wii sales top 70 million worldwide. [Online]. http://www.afterdawn.com/news/article.cfm/2010/05/10/wii_sales_top_70_million_worldwide
- [13] Mark Weiser. (91, Sep.) The Computer for the 21st Century. [Online]. <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>
- [14] Communities and Local Government, "Online Social Networks," London,.
- [15] Café com Notícias. (2011, Feb.) Reportagem especial - A revolução no Egipto e a força das redes sociais. [Online]. <http://cafecomnoticias.blogspot.com/2011/02/reportagem-especial-revolucao-no-egito.html>

- [16] Digizen. Social Networking. [Online]. <http://www.digizen.org/socialnetworking/>
- [17] Hong Lu and [et al]. (2009) SoundSense: Scalable Sound Sensing for People-Centric.
- [18] Aaron Beach and [et al] , "WhozThat? Evolving an Ecosystem for Context-Aware Mobile Social Networks," in *IEEE Network*, University of Colorado at Boulder, 2008.
- [19] Nathan Eagle and Alex Pentland, "Social Serendipity: Mobilizing Social Software," in *IEE Pervasive Computing*, 2005.
- [20] James R. Clynch. (2001, February) GPS Accuracy Levels. [Online]. <http://www.oc.nps.edu/oc2902w/gps/gpsacc.html>
- [21] Axel Kupper, *Location Based Services Fundamental and Operation*. England: Wiley, 2005.
- [22] San Diego Research Center Zhensheng Zhang, "Routing in Intermittent Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges," in *IEEE Communication Surveys*, 2006.
- [23] Mirco Musolesi and [et al] , "Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones," in *Pervasive*, Helsinki, Finland, 2010.
- [24] Anthony J. Nicholson and Brian D. Noble, BreadCrumbs: Forecasting Mobile Connectivity.
- [25] Angela Alcorn, *The (very) unofficial facebook privacy manual.*: Makeuseof.com.
- [26] Charles M. Gartrell, "SocialAware: Context-Aware Multimedia Presentation via Mobile Social Networks," University of Colorado at Boulder, 2008.
- [27] Mike Gartrell, and Richard Han Aaron Beach, "Solutions to Security and Privacy Issues in Mobile Social Networking," University of Colorado at Boulder,.
- [28] Angela Dalton, and Varun Marupadi Landon P. Cox, "SmokeScreen: Flexible Privacy Controls for Presence-Sharing," in *MobiSys'07*, San Juan, Puerto Rico, USA, 2007.
- [29] Literate Programs. Quine-McCluskey algorithm (Java). [Online]. [http://en.literateprograms.org/Quine-McCluskey_algorithm_\(Java\)](http://en.literateprograms.org/Quine-McCluskey_algorithm_(Java))
- [30] Emma Barnett. Facebook's Mark Zuckerberg says privacy is no longer a 'social norm'. [Online]. <http://www.telegraph.co.uk/technology/facebook/6966628/Facebooks-Mark-Zuckerberg-says-privacy-is-no-longer-a-social-norm.html>
- [31] Valderi R. Q. Leithardt and [et al] , "Quietly: A model privacy for pervasive and ubiquitous environments," *Personal and Ubiquitous Computing*.
- [32] Simson Garfinkel. (2012, Agosto) History's Worst Software Bugs. [Online]. <http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all>
- [33] Lara Srivastava ITU, "UBIQUITOUS NETWORK SOCIETIES: The Case of Radio Frequency Identification," 2005.

- [34] Simon Baker, Introduction to Agile and Scrum, 2006.
- [35] Pablo H. Ibarguengoytia-Gonzalez, Joaquín García-Hernández, Desús A.Pérez-Díaz Carlos F. García-Hernández, "Wireless Sensor Networks and Applications: A Survey," *International Journal of Computer Science and Network Security*, 2007.
- [36] Ph.D. Brian W. Pogue, "Smartphone Sensing," DARTMOUTH COLLEGE, 2011.
- [37] Stephanie Crawford. Como funciona o Microsoft Kinect. [Online]. <http://eletronicos.hsw.uol.com.br/microsoft-kinect2.htm>
- [38] David Nunes, "Integrating the Real and Virtual:Activity Detection ," Coimbra, 2011.

**Anexo A “A Web Service-Based Framework Model for
People-Centric Sensing Applications Applied to Social
Networking”**

Article

A Web Service-Based Framework Model for People-Centric Sensing Applications Applied to Social Networking

David Nunes *, Thanh-Dien Tran, Duarte Raposo, André Pinto, André Gomes and Jorge Sá Silva

Department of Informatics Engineering, University of Coimbra, Pólo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal; E-Mails: than@dei.uc.pt (T.-D.T.); draposo@student.dei.uc.pt (D.R.); afpinto@student.dei.uc.pt (A.P.); asng@dei.uc.pt (A.G.); sasilva@dei.uc.pt (J.S.S.)

* Author to whom correspondence should be addressed; E-Mail: dsnunes@student.dei.uc.pt; Tel.: +351-914-611-079; Fax: +351-239-701-266.

Received: / Accepted: / Published:

Abstract: As the Internet evolved, social networks (such as Facebook) have bloomed and brought together an astonishing number of users. Mashing up mobile phones and sensors with these social environments enables the creation of people-centric sensing systems which have great potential for expanding our current social networking usage. However, such systems also have many associated technical challenges, such as privacy concerns, activity detection mechanisms or intermittent connectivity, as well as limitations due to the heterogeneity of sensor nodes and networks. Considering the openness of the Web 2.0, good technical solutions for these cases consist of frameworks that expose sensing data and functionalities as common Web-Services. This paper presents our RESTful Web Service-based model for people-centric sensing frameworks, which uses sensors and mobile phones to detect users' activities and locations, sharing this information amongst the user's friends within a social networking site. We also present some screenshot results of our experimental prototype.

Keywords: people centric sensing; social networks; wireless sensor networks

27. 1. Introduction

When putting things in perspective and considering our civilization's history and existence so far, the Internet is nothing short of amazing. In the relatively short time it has existed it has already

transformed the way our world and society work, at a very fundamental level. It has also done so at a faster pace than any other massively important technological advancement. Not surprisingly, considering the social creatures we humans are, as our Internet evolved so did the means we use to communicate and interact with the people we deem close. Social networking bloomed and connected together an astonishing number of users. At the same time, sensing technologies like Wireless Sensor Networks (WSNs) are becoming more robust and viable while the sensor nodes themselves are becoming cheaper. Modern mobile phones and smartphones have also brought us mobility in our Internet access and are also a powerful computational resource, with many interesting capabilities. Mobile phones are also becoming even more powerful, common and cheaper. Never, in its entire history, did mankind have a collection of technologies that are capable of connecting and monitor such large numbers of users, simultaneously, at a global scale.

While the Web and its associated technologies will continue to evolve at increasingly faster paces, people will always live on the “real world”. Considering our social nature and the importance of “social media” throughout our history, the most likely path in our technological and social evolution is a merge between the real and the virtual: a society where people are connected not only by their physical proximity, but also by their virtual closeness. This implies the development of “people-centric” sensing systems that can acquire, process and share, in a private and secure manner, useful information about social network users. This new trend in our civilization’s evolution is still making its first steps. Our generation is on the verge of a technological revolution that will, in the next few decades, transform our society and open the path to many new market opportunities.

We are now seeing the first people-centric applications being released onto the market. In these systems, humans and their activities are the central focus of sensing and data visualization is focused towards the user’s friends and family, rather than research teams or technical staff. As the researchers of the MetroSense project suggest [1], people-centric sensing applications can be thought of as having a **personal**, **social**, or **public** focus. Applications with personal focus have been around for quite some time now, good examples are fitness and sports monitoring applications that use smart watches to acquire information regarding the user’s movement, heart rate and other vital signs and use this information to provide feedback on the user’s performance, like MapMyFitness [2] or Digifit [3]. Public-focused applications gather information about individual users within a metropolitan area and combine it to achieve high-level goals, such as managing the city’s resources. There are several projects on this area, namely the UCLA Urban Sensing initiative [4] and Wikicity [5]. People-centric sensing within a social context is perhaps the least explored of the three areas and the focus of our work. Applications in this area usually imply the analysis of the sensed data to detect social interactions and activities of people.

Although we already have the necessary base technology to develop these people-centric systems there are still many challenges associated with their development. Wireless Sensor Networks, for instance, are still mostly confined to research or industrial scenarios, isolated from the “real-world” and far away from useful “day to day” use. One can reason several possible causes that contributed to this situation. There are still financial and technological limitations in sensor node technology: nodes are computationally limited and still relatively expensive. Another reason is that development requires

a detailed understanding of low level protocol and network, therefore requiring a great amount of effort to build new applications. This is not a particularly favorable situation in a world where the Web 2.0 and user-generated applications and content are becoming ever more important. In addition, interoperability is another difficult problem created by the many independent proprietary data formats used by different sensor networks. Although it has been proven that it is possible to implement generic communication mechanisms (both IP protocol stack (6LowPAN) and web services) on the sensor nodes, the applicability and efficiency of these technologies still needs further validation from successful real world applications. Because of their limitations, allowing directly access to sensor nodes from the Internet brings many challenges and may not be a suitable solution for all WSN applications. Therefore, it is necessary to have an infrastructure that easily, securely, and efficiently supports the integration and interoperability of sensor networks with Web 2.0 environments.

On the other hand, social networks are still very “static”. Access to social networking nowadays is becoming increasingly more mobile. It is not uncommon to see people access their smartphones to share and discuss daily experiences shortly after their occurrence, updating thoughts and responding to feedback from their friends as the situation develops and the user’s life continues. Current users can announce social events to their group of friends, share experiences through photos and comments, and show their opinions and hobbies through “likes” and their own “private wall”. Despite the general public’s interest in these social services, their current functionality does not reflect the true dynamic of people’s relationships and personal lives. Instead of being pre-determined and unique events in time, social group activities can, in fact, happen very frequently and, most of the time, spontaneously. Current systems are not capable of providing this “real-time” component to social networking, which diminishes its true potential.

With these new social trends and current technological limitations in mind, we propose SocialSense, a system that combines social networking with mobile sensing technologies to expand and provide greater functionality to social networks and other web applications. In short, this system will focus on bringing a “real-time” component to social networking, by sharing and exploring real-time information about the users’ daily actions, social activities and visited locations. By using body-area networks and mobile phones, we intend to research and develop mechanisms that focus on “people-centric sensing” and provide a seamless connection between the virtual (social networks) and the real (the user’s daily lives). This connection brings greater value not only for the user’s social experience but also for the service provider, as the new types of information available can be mined and analyzed to create more personalized and interesting product advertisement mechanisms.

A system on a chip or “system on chip” (SoC or SOC) integrates several electronic system components into a single chip. Research on the topic of people-centric sensing is highly dependent on the development of integrated and flexible, yet cheap and comfortable sensors that function with a low energy consumption. This is a challenging task since current sensor nodes are still too bulky and expensive. We expect that this situation will change in the near future.

The remaining of this document is organized as follows: next section introduces an overview of our point of view on the research topic where we present our current model and working prototype. Section 3 details our current results while the last section concludes this article.

28. 2. Topic Overview and SocialSense's Model

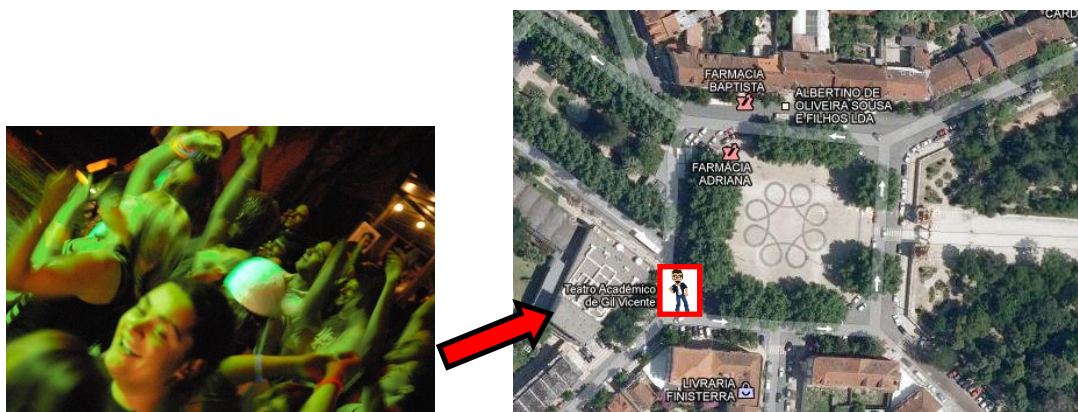
Bridging the real and virtual environments (Social Networks such as Facebook or Virtual Worlds such as Second Life and World of Warcraft) is a subject that has received a great deal of attention in recent scientific research, with some interesting experiences being performed in this area. The Beyond Broadcast conference, for example, took place not only in Cambridge, but also on an island in the 3-D virtual world Second Life. This was accomplished by streaming video and audio from the real-world conference to the Berkman Island in Second Life [6]. Another interesting experiment is described in [7] where a real life control panel was connected to a virtual control panel in Second Life in a way that, turning a knob on the real world one or pushing a button produced the exact same changes in the virtual replica. The work of Lifton, J. et al at the MIT media laboratory has proved to be much more complex and sophisticated. The authors coined the term "dual reality" [8,9] to indicate the ability to merge the real and virtual realities by using sensor networks. They designed several prototypes where they performed experiences in merging a real world location, the Media Lab's third floor, and virtual worlds, in this case Second Life. One of these prototypes is described in [10] where the authors present the ShadowLab, a Second Life map of the Media Lab's third floor animated by data collected from a network of several sensor/actuator nodes. Another dual reality implementation was the Ubiquitous Sensor Portal, a system designed for two-way cross-reality experience and communication. These portals stream information in both directions, from the user's environment to ShadowLab, and from ShadowLab into the real world.

There are also applications of sensor nodes as means of transmitting mobility into virtual worlds. In [11] a framework is proposed which maps a sensor node to an object in Second Life. The location of the sensor node is calculated by the framework and reflected on an avatar in Second Life which moves according to the real world movement of the node. The location of the node is calculated from the RSSI values from three or more fixed reference nodes, thus requiring a carefully designed WSN architecture.

However, none of the above works could be considered "people-centric", despite their connection to social networking. In order to better understand the applicability of people-centric sensing systems, let us consider a simple use-case. A social network user goes on a night-out with some of his friends to a local nightclub. He has an activity detection system that runs on his smartphone, possibility accompanied by several small body-area wireless sensors (in the form of a small bracelet, for example). The small sensors in his body directly communicate with the smartphone through a body-area wireless sensor network and share information regarding several conditions, such as his movement (accelerometer) and heartbeat. The sensors can also detect environment factors such as the temperature. The smartphone runs activity detection algorithms that combine the sensory information (for example, an increase in heart rate, temperature and movement) to discern what the user currently

doing, in this case, dancing. The smartphone accesses the Internet and communicates with the user's preferred social network to represent information about the user's current activities to his trusted group of friends. The information can be represented through a virtual representation of the user and the environment. If the user's friends also use the same system, it is possible to correlate their close physical proximity with activity detection and deduce that the group is currently engaging in a social group activity (they are all dancing in a nightclub). People who belong to this same group of friends who are still at home and were indecisive as to whether or not go to the nightclub can now have an idea of who is attending the social event and where it is happening, and use this information to make their decision.

Figure 1. Illustration of the use of activity detection with virtual social networks. A user's activities and location are represented in real time and become accessible amongst the user's group of friends.



The above use-case is a good example of how a people-centric sensing system could be a part of a regular person's social life. The term "people-centric sensing" is used by the MetroSense project [12] to describe a vision where the majority of network traffic and applications are related to sensor and actuator data, applied to the general populace. The MetroSense project envisions collaborative data gathering of sensed data by individuals, facilitated by sensing systems comprised of cheap and easily accessible mobile phones and their interaction with software applications. As a part of the MetroSense project, the authors of the work in [13] proposed the use of sensors embedded in commercial mobile phones to extrapolate the user's real-world activities that in turn can be reproduced in virtual settings. The authors' goal was to go further than simply representing locations or objects and provide virtual representations of humans, their surroundings and their social interactions. The system prototype implementation proposed was named CenceMe and allows members of social networks, namely Second Life, to process the information sensed by their mobile phones and use this information to extrapolate information about the user's surroundings and actions.

The state of the art in the area still represents only the first steps for these people-centric sensing systems. There is plenty of room for improvement in various areas, encouraging future research. In fact, creating such systems still presents several technical challenges that we intend to address with our research, namely:

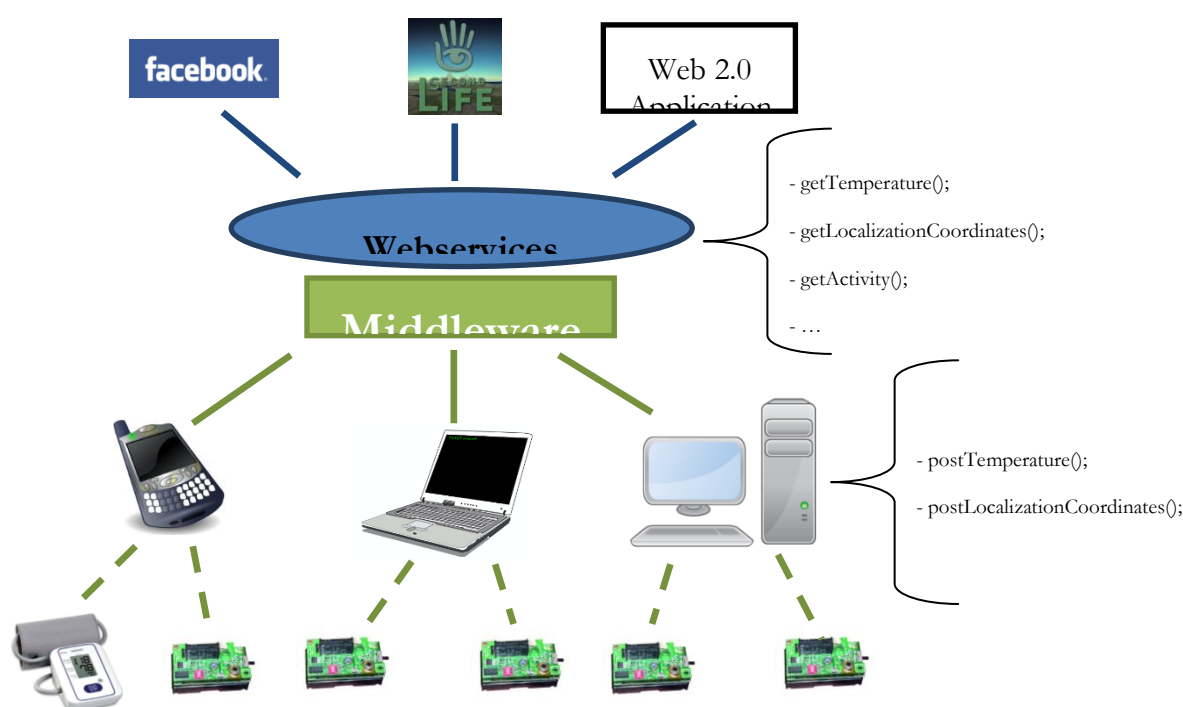
- **Privacy** Exchanging sensitive information such as user's vital signs, location and current activity is a task that has strict privacy requirements. Privacy is a requirement that affects all application layers: from the sensor's wireless communication to the exchange of middleware messages. During our research we intend to employ effective privacy mechanisms that ensure the safety of sensitive information.
- **Managing intermittent connectivity** Reliability of wireless sensor networks is very important in order to create systems that are robust and stable. However, due to the wireless nature of mobile phone and WSNs communication, there is always a risk of intermittent connectivity due to environment interference, power shortages or communication range. Robust people-centric sensing systems should be capable of maintaining an illusion of smooth and continuous people tracking. Offline mechanisms that can estimate a user's current activity and location even when there is a shortage of fresh data can maintain an illusion of continuous monitoring in adverse conditions. These offline mechanisms can also be employed in data mining and analysis, since their function is to find patterns in the data.
- **Detection of activity** With basis on previous work in the area, our research will attempt to improve and design resource aware algorithms that use both the mobile phone and wireless sensors and are directed specifically towards the representation of activities in social networking sites and virtual communities. At the same time, we will also seek interesting and engaging ways of representing user activities and environmental data, through the use of virtual representations of users (virtual avatars).

Aside from these technical challenges from an application point-of-view, research on the topic of people-centric sensing is also highly dependent on cheap and comfortable sensors that have a decent battery life. One of the current obstacles in our research is a lack of extremely lightweight and small sensors that can be used in a ubiquitous way, that is, the user forgets that he is using them. The area of SoC is extremely important for the development of new sensor devices that are both extremely flexible, comfortable and with a good battery-life. For example, in [14] the authors have published the design methodology and a demonstration of a silicon chip that has all the necessary characteristics for medical diagnostics, environmental monitoring and personal connectivity systems. More recently, the authors in [15] describe the design, implementation, and verification of a system-on-chip aimed to play the role of a general purpose processor for a wireless body area sensor network node. This a very interesting work, since body area sensor networks are the basis of people-centric sensing. Continuous research on the field brings further innovations; the work in [16] shows miniaturized "plug & play" sensor nodes that can collect vital physiological information and wirelessly transmit it to the system database.

We intend to design a model based on innovative solutions that effectively address all these technical challenges. Small and discreet wireless sensors allow for more diverse kinds of vital-signs data to be used, which permit the creation of accurate activity detection systems. Activity detection amongst groups of friends can be correlated to automatically create social events that can be shared in Social Networks. Localization techniques can be used to permit users to locate their friends in big social events. Offline inference techniques can provide an illusion of seamless connection between the

real and the virtual, by trying to infer current location and activity even when there is no fresh data due to a connectivity problem. All of these research approaches attempt to expand the features of social networking sites and increase their functionality beyond the simple sharing of photos and comments.

Figure 2. SocialSense's current architecture and some examples of used services.



The research herein proposed is based on a testbed already implemented in our labs. In our current model, Wireless Sensor Networks (WSNs) are data and functionality providers, managed by an integration middleware. This platform is called SocialSense and provides a set of services for consumption by external environments using web service APIs. This model facilitates the integration and interoperability of WSNs with virtual world environments, social networks and other web 2.0 applications. Our approach is based on a hybrid gateway-based model that supports both proactive and reactive communication. In proactive communication the sensor nodes periodically collect data and send it to the middleware for processing. In reactive communication, the end users send data/functionality requests to the middleware which then uses the sensor node's data to react accordingly. A simplified illustration of our current prototype's architecture is depicted in Figure 2. The major difference between our basic architectural solution and CenceMe [13] is our reliance on wireless sensors external to the mobile phone, a central middleware and the exposure of our system's data and functionality through web services. Wireless sensors are becoming increasingly smaller and comfortable to use, so their addition is an advantage over CenceMe, since we have more different

types of data that can be used to more accurately detect human activities. A centralized middleware allows us to have more control over client requests and data, facilitating the future implementation of robust privacy mechanisms. RESTful web services open our system's functionality to the public, promoting the re-use of wireless sensor deployments and the creation of new web applications by third-parties outside the scope of our project.

Our model connects the middleware with the sensor networks through several entities called "proxies". They receive data packets from the sensor nodes and perform earlier processing tasks, forwarding the information to the middleware for more advanced management processing. In addition, they may also handle specific commands and requests coming from the middleware to satisfy client applications, by performing queries to the sensor nodes. Proxies also include a few components that support mobility. One of the main advantages of this modular approach is that it can easily offer mobility, localization and scalability. Proxies are designed to be lightweight, so that they may be installed on both computers/laptops and modest or limited devices such as smartphones. Each proxy can handle communication with several wireless sensor networks and all proxies eventually connect to the central middleware service.

The middleware gateway exposes the functionality and data of sensor networks as web services. It is the central part of our framework, the basis for the integration and interoperability between sensor networks and other Web 2.0 environments. This approach facilitates the development of applications using sensor networks (users simply use a web service, instead of programming the wireless network), increases the diversity of possible web applications and promotes the reuse of WSN deployments and software code. In addition, data coming from the sensor networks can easily be stored in a cloud data warehouse for further processing. The middleware contains the necessary intelligence to process messages coming from different sensors and requests coming from different users and applications. It also contains the necessary interfaces to proactively send data to different Web 2.0 applications.

The activity detection module is responsible for processing sensor data and running algorithms that are capable of classifying current user activity. Current prototype versions of this module use accelerometer data to perform the detection of very simple activities, such as walking or running. These earlier versions were also implemented as part of the middleware service, but in the future the module will be moved onto a mobile proxy (possibility a smartphone) in order to promote mobility and scalability (signal processing tasks are to be performed by the user's own hardware). As our research advances we will explore the use of more types of data and attempt to design more efficient and accurate mechanisms for detecting activities.

Privacy is an area still under very active research and implementation. We are currently implementing a privacy mechanism based on TARF, a trust-aware routing framework for WSNs presented in [17]. TARF is an energy-efficient, highly scalable, and well adaptable solution that integrates both trustworthiness and energy efficiency in making routing decisions. TARF's privacy mechanism is based on a "Trust level" measurement that represents a node's opinion of his neighbor's level of trustworthiness. Neighbors with trust levels below a certain threshold are excluded as candidates for data routing operations. When an intruder forges the identity of a legitimate node, at

first, the intruder will be able to deceive its neighbors, but as the legitimate node broadcasts information about undelivered packets, the intruder's neighbors will start to downgrade its trust level and the intruder is eventually left out of the network. Our work intends on improving on TARF in a sense that TARF only allows for the downgrade of a neighbors' trust level, while we intend to create a more dynamic system that can both increase or decrease a node's trustworthiness depending on the situation. When thinking in terms of Social Network access there are also several important issues regarding privacy. Regarding Facebook, in particular, there are several efforts that try to expose its most relevant privacy and security flaws. For example, "The Very Unofficial Facebook Privacy Manual" [18] shows several privacy issues present Facebook, such as the fact that when you send a message to someone on Facebook they get limited access to your profile or when someone adds you as a friend and you haven't said yes or no to them, they can see your updates in their home feed. These types of issues can be avoided and circumvented for SocialSense data, since our solution is independent from any particular social network.

Intermittent connectivity and interference avoidance are areas that were already a subject of previous work within our research group. Namely, the work presented in [19] addresses interference issues on channels of IEEE 802.15.4 wireless networks. The study shows that it is difficult to predict the quality and stability of different wireless channels in sensor networks as the environment conditions greatly influence network performance and provides an experimental evaluation of the best channels for IEEE 802.15.4 networks. While the results and experience gained from our previous evaluations were already applied in our current prototype, research in intermittent connectivity will continue to be a part of future deployments.

SocialSense was also thought to be able to support the use of several medical devices. In our current implementation, we use a small heart beat monitor (also depicted in Figure 2), whose communication is based on the Bluetooth protocol. The use of these "external" devices requires an intermediate device (mobile phone or wireless node) that supports both the external protocol (Bluetooth) and communication with the middleware.

From the beginning, our vision was not to just propose a model for the integration of WSNs with web environments, but to expand this model towards people-centric sensing and to include mechanisms to provide location, vital signs and activity detection, using mobile phones and wireless sensor nodes as base hardware. We intend to offer a robust platform that offers environmental and personal information and allows this information to be combined in useful ways for both social networking users and service providers.

29. 3. Results and Discussion

Our most recent implementation of the model supports 6LowPAN as well as traditional WSNs. Communication between proxies and middleware is based on TCP/IP and RESTful web services, currently supporting XML and JSON as the data formats. REpresentational State Transfer (REST) [20] is a web service design idiom that uses a stateless client-server architecture in which the web services can be identified by URLs. Web service clients wanting to use these services have to resort to a small set of remote methods that are very similar to the ones present in HTTP, such as GET and POST.

The communication between proxies and sensor nodes is currently implemented independently using any protocols that are appropriate, in order to allow interoperability between different networks. Web services at the “bottom level” are used by the proxies for posting new data to the middleware. The web services provide basic functionality for storing data (which can be sensory or localization data) within the system in a private database. Examples of such “low level” web services called by proxy devices are the “postTemperature();” and “postLocalizationCoordinates();” services, although there are others of similar functionality. “Higher level” web services are used by web interfaces and applications to retrieve ready-to-use, processed data from the middleware. Our own web interface (that serves as a link between our framework and Facebook) uses several web services, such as “getTemperature();”, “getLocalizationCoordinates();” and “getActivity();” in order to update the user’s avatar position within the map of our testbed, as shown in Figure 3.

Figure 3. Current Web Interface displaying a user’s location and a simple avatar (circled red)



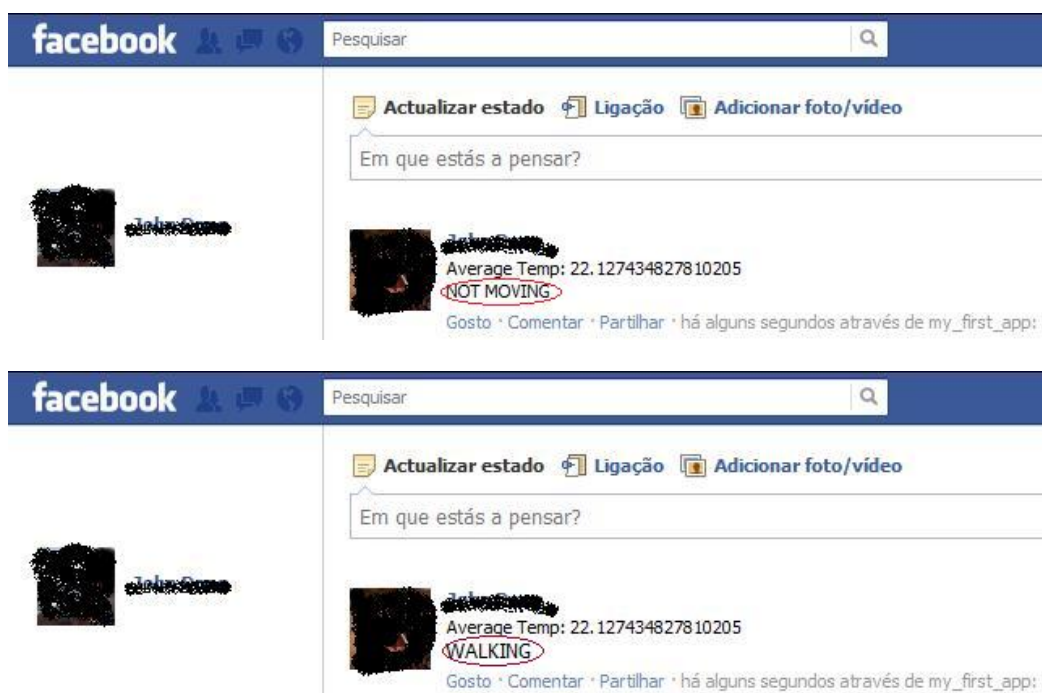
These web services are responsible for retrieving the most recent data from our database regarding a user’s current location, vital signs or activity. In addition to the web services provided by our own middleware, our current web interface also integrates with the Google Maps API in order to display the users’ location and activity by overlaying a map of our testbed and a small avatar. The interface allows users to select those who will have access to their own activity and location, and we implemented a login-based security mechanism to access data in our middleware. This login-based mechanism relies on Facebook’s own “access token” mechanism: when a user accesses our application, a web page asks for permissions to fetch various types of data (the user’s friends, Facebook ID, etc). In case the user accepts these terms, an “access token” is generated by Facebook and sent to our middleware by the user’s browser through HTTPS. This access token is then used by our middleware to communicate with Facebook and fetch the user’s Facebook ID, which is then compared to the existing IDs in our database. In case the ID exists on the database, the user is given access to sensory data, otherwise, our webservice returns nothing. Thus, only users registered within our system can fetch relevant data. The

middleware supports a localization services based on the RSSIs of the received wireless packets and a trilateration algorithm.

We developed a simple activity detection based on the data collected from the accelerometer sensor of Zolertia Z1 motes [21]. With the accelerometer's movement speed data, we perform simple signal frequency analysis tasks to classify the user's activity as "moving" or "not moving". Our approach is similar to the one described in [22], frequencies above the ones resulting from human walking motion are filtered out and a simple sum of the Fast Fourier Transformation coefficients is performed in order to get a rough idea of the intensity of the movements being experienced. This intensity is used to determine if the user is either walking or stopped. At the same time, localization can be used to further distinguish the type of movement; for example, if the user's movement's speed is less than 6 km/h then the system determines that the user is "walking". Otherwise, the user is running on foot or driving car, depending on the scenario. This classification scheme is extremely simple and results from our early efforts in establishing a working prototype. Future work will result in more advanced classification schemes.

In order to integrate activity detection with social networking, we developed a mashup application that combines the services provided by our platform (through the middleware web services) with the Facebook platform. The mashup periodically collects user activity and sensor data and publishes them into the user's personal wall. The results of this integration are shown in Figure 4.

Figure 4. Activity and sensory information are displayed on the Facebook user's personal wall.



The sensors used for our application were general-purpose Zolertia Z1 motes. These sensors are bulky and not wearable. Future iterations of our work would benefit from the use of more comfortable, smaller and application-specific sensors stemming from current research in the field of SoC. A good

example of an interesting sensor for people centric applications is the UP bracelet [23] from Jawbone, which provides motion sensing and heart beat monitoring. While the concept is very good, the bracelet still presents some limitations. We believe that people-centric sensing systems should focus on using common technology (mobile phones) and cheap and flexible technology, in order to promote the creation of interesting applications in a Web 2.0 context. The UP bracelet is quite expensive (aprox. 100 euros) and the diversity of measurements is low (only measures movement and heartbeat, which is quite common and does not justify the high pricing). It is also part of an integrated and closed monitoring system, thus limiting it's applicability in new people-centric systems. These limitations make it less interesting for both research and the general populace.

From our point of view, ideal sensors for people-centric sensing applications must be very comfortable and lightweight and provide as many measurements as possible. This is a problem, since many interesting vital signs can only be measured with cumbersome technology. To better explain our vision, let us consider a people centric sensing system that has “multiple” solutions. The “basic” solution has only the most important hardware (a bracelet with heart beat and accelerometer, for example) that acts as a direct bridge to the user's mobile phone. However, users can acquire more sensors that provide greater capability. A “sports” solution, for example, can provide more sensors for a better assessment of physical exercise. Oxygen saturation (SpO2) is an interesting measurement for this case, since it represents the amount of oxygen that dissolved or carried in the user's bloodstream. However, if we wanted to use this measurement, we would have to depend on cumbersome sensors that are too big to be comfortable and usable in a social context, as the one shown in Figure 5.

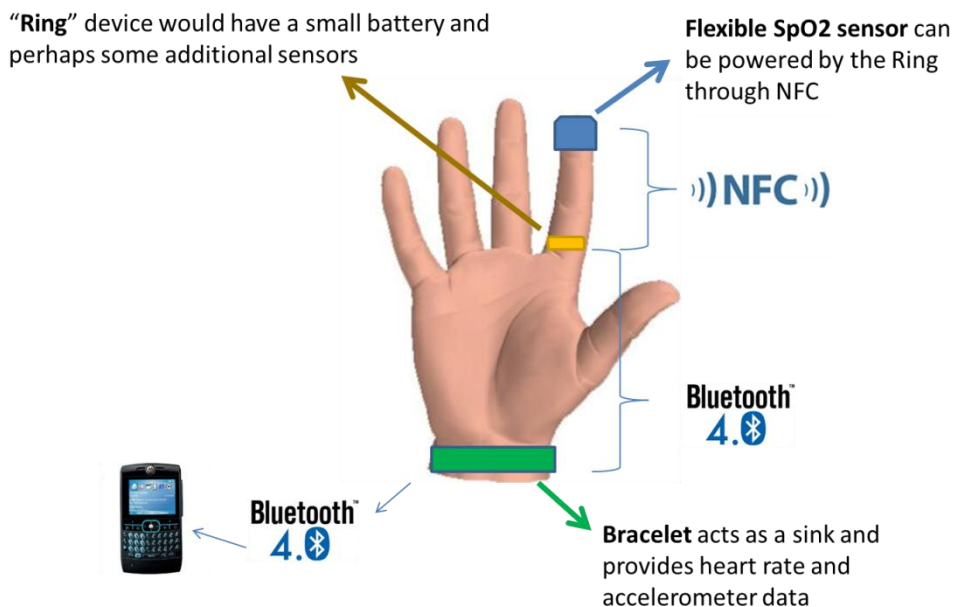
Figure 5. SpO2 sensor. The sensor's cable and size make it very uncomfortable to use in sports or social interactions [24].



From our understanding, there are several reasons that contribute to the SpO2 sensor's large size and cabling, namely their need for batteries. One solution to this problem could be the use of Near Field Communication (NFC) technologies in order to remove the need for both cabling and batteries, drastically reducing the sensor's overall size. Recently, STMicroelectronics has introduced a 16-kbit memory NFC chip designed to employ energy harvesting technology to power up small devices, thus eliminating the need for an internal power supply [25]. At the same time, NFC can be applied for wireless communication, thus removing the need for cabling. Thus, an interesting SoC challenge would be to combine NFC with a SpO2 sensor, providing a more comfortable solution for people-centric systems. In Figure 6 we present an interesting combination of sensor devices and

communication technologies that could provide an interesting solution for activity detection that is relatively comfortable, small and, perhaps, even “fashionable”:

Figure 6. Possible combination of hand-worn sensors that can be used in a people-centric sensing application.



Since NFC has a very short range (max. 20 cm, but typically only a very few cm), a “Ring” device could act as a wireless bridge between the SpO2 sensor and the bracelet. This would require embedding Bluetooth Low Energy and NFC onto a single device, while maintaining the overall price as affordable as possible. This solution would be extremely interesting from our point-of-view, since it would provide many different measurements while remaining relatively comfortable for the average user.

30. 4. Conclusions

Although the obtained results presented in this work are very simple, they prove that it is feasible to seamlessly merging physical and virtual world and develop new applications by accessing openly exposed sensory information through webservices. They also show that it is feasible to detect users' activity based on the data from sensors. This work represents our first steps into our research on people-centric sensing.

The sensors used on our application prototype were general purpose wireless sensors. These are not adequate for real-world deployments of people-centric sensing systems. Such systems benefit from small sensors that are both comfortable to wear and energy efficient. Future developments in the field of SoC will prove to be important for the development of sensors that are specifically oriented for these kinds of systems. We expect that future sensor technology will become cheaper and smaller, making people-centric systems more attractive to the general populace.

31. References

1. A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. s. Ahn, "The rise of people-centric sensing."
2. Mapmy: Fitness. [Online]. Available: <http://www.mapmyfitness.com/imapmy/wahoo/>
3. F. Zahradnik. Digifit: Wireless heart rate monitor plus fitness, training apps for iphone. [Online]. Available: <http://gps.about.com/od/mobilephonegps/a/digifit-iPhone-heart-rate-review.htm>
4. Urban sensing. [Online]. Available: <http://urban.cens.ucla.edu/>
5. F. Calabrese, C. Ratti, and K. Kloeckl, "Wikicity: Real-time location-sensitive tools for the city."
6. The conference goes virtual: Second life. 2008 Available online: <http://www.beyondbroadcast.net/blog/?p=37>
7. Real life control panel for second life. January 2004. Available online: <http://channel3b.wordpress.com/2007/01/24/real-life-control-panel-for-second-life/>
8. J. Lifton, "Dual reality: An emerging medium," Ph.D. dissertation, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences., 2007.
9. J. Lifton and J. A. Paradiso, "Dual reality: Merging the real and virtual," in in Proceedings of the First International ICST Conference on Facets of Virtual Environments (FaVE), Berlin, Germany, July 2009, pp. 27–29.
10. J. Lifton, M. Laibowitz, D. Harry, N.-W. Gong, M. Mittal, and J. A. Paradiso, "Metaphor and manifestation—cross-reality with ubiquitous sensor/actuator networks," *IEEE Pervasive Computing*, vol. 8, pp. 24–33, July 2009. Available online: <http://portal.acm.org/citation.cfm?id=1591886.1592128>
11. T. D. Tran and J. Silva, "A framework for integrating wsns and external environments," in 5th International Conference on Management and Control of Production and Logistics, Coimbra, Portugal, 2010.
12. Metrosense project. Available online: <http://metrosense.cs.dartmouth.edu/>
13. M. Musolesi, E. Miluzzo, N. D. Lane, S. B. Eisenman, T. Choudhury, and A. T. Campbell, "The second life of a sensor - integrating real-world experience in virtual worlds using mobile phones," in In Proc. of HotEmNets '08, 2008.

14. L. Wang, N. Aydin, A. Astaras, M. Ahmadian, P. A. Hammond, T. B. Tang, E. Johannessen, T. Arslan, S. P. Beaumont, B. W. Flynn, A. F. Murray e J. M. Cooper, “A SENSOR SYSTEM ON CHIP FOR WIRELESS MICROSYSTEMS”.
15. Z. Stamenkovic e G. & S. G. Panic, “A System-On-Chip for Wireless Body Area Sensor Network Node,” em *Proceedings of the 2008 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, IEEE Computer Society, 2008, 1-4*.
16. A. Saeed, M. Faezipour e M. & T. L. Nourani, “Plug-and-play sensor node for body area networks,” em *Life Science Systems and Applications Workshop, 2009. LiSSA 2009. IEEE/NIH, 2009*.
17. G. Zhan, W. Shi, and J. Deng, “Tarf: A trust-aware routing framework for wireless sensor networks.” in *EWSN*, ser. Lecture Notes in Computer Science, J. S. Silva, B. Krishnamachari, and F. Boavida, Eds., vol. 5970. Springer, 2010, pp. 65–80.
18. A. Alcorn. (2010, October) The Very Unofficial Facebook Privacy Manual . [Online]. Available: <http://netsavoir.com/2010/10/27/facebook-privacy-manual/>
19. T.-D. Tran, R. Silva, D. Nunes, and J. Silva, “Characteristics of channels of iee 802.15.4 compliant sensor networks,” *Wireless Personal Communications*, pp. 1–16, 2010, 10.1007/s11277-011-0395-3.
20. S. Tyagi. (2006, August) Restful web services. [Online]. Available: <http://www.oracle.com/technetwork/articles/javase/index-137171.html>
21. Zolertia. Z1 platform. [Online]. Available: <http://www.zolertia.com/ti>
22. E. Welbourne, J. Lester, A. Lamarca, and G. Borriello, “Mobile context inference using low-cost sensors,” in *In Proceedings of LoCA*. Springer-Verlag, 2005, pp. 254–263.
23. Jawbone webpage. Available online: <http://jawbone.com/up>.
24. Available online: <http://www.sensoronics.com/images/reusable-spo2-sensors-large.jpg>
25. S. Clark, “NFC goes green: New ST chips use energy harvesting to replace the need for batteries,” 8 November 2011. Available online: <http://www.nfcworld.com/2011/11/08/311126/nfc-goes-green-new-st-chips-use-energy-harvesting-to-replace-the-need-for-batteries/>

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).

**Anexo B “Percontrol: Use of Neural Network and
WSN to Control the User Profile in Pervasive
Environments”**

Percontrol: Use of Neural Network and WSN to Control the User Profile in Pervasive Environments

Valderi R. Q. Leithardt, Carlos O. Rolim, Anubis Rossetto, Cláudio F. R. Geyer
Federal University of Rio Grande do Sul
Institute of Informatics - GPPD
Porto Alegre – RS – Brasil
{valderi.quietinho, carlos.oberdan,anubis.rossetto,geyer}@inf.ufrgs.br

David Nunes, Duarte Raposo, Jorge Sá Silva
University of Coimbra
Department of Informatics Engineering
Pólo II, Pinhal de Marrocos, Coimbra 3030-290, Portugal
{dsnunes, draposo}@student.dei.uc.pt; sasilva@dei.uc.pt

Abstract: Wireless Sensor Networks (WSNs) can be considered a subclass of Ad-Hoc networks since they are composed by a great amount of sensor nodes that can communicate amongst themselves. Artificial Neural Networks (ANNs) are a powerful and flexible way to approach the problem of artificial intelligence, and can be used in many different application scenarios. WSNs and ANNs are used in many pervasive applications to capture and process data for control and managing tasks; with these facts in mind, this article's objective is to present a solution for a control system that uses WSNs and ANNs applied to pervasive environments.

Keywords: Pervasive Computing, Routing, Wireless Sensor Networks, Artificial Neural Networks.

I. INTRODUCTION

In 1991, Mark Weiser [1a], claimed that the most profound technologies would become so integrated onto our daily lives that it would be as if they just “disappeared”, since using them would become second-nature. With this idea in mind, Weiser proposed a new form of computation, pervasive computation.

By “disappearing” it is meant that the device becomes imperceptible; in other words, the user no longer needs to worry about the technological resource in use. From this perspective, pervasive computing frees users from worrying about secondary tasks (using the device's interface

correctly) and allows them to focus on their primary task (use the device as a tool to perform the desired action).

The development of microelectronics and wireless communication has greatly benefited the development and usage of Wireless Sensor Networks (WSNs), namely in a pervasive / ubiquitous environmental context.

WSNs are usually composed by a large number of wireless nodes spread through an area containing the events / phenomenon to be monitored [1]. A sensor node is a device with reduced size meant to be cheap and easily deployed in risky or difficult to access locations. Their small size severely limits their hardware capabilities; sensor devices usually present many limitations when it comes to battery, processing power, storage and communication interfaces [2]. These restrictions affect the amount of sensor nodes present in a given network, as well as the network's performance. This leads to a need to develop communication protocols that can handle the sensors' inherent limitations while being adequate for the application and associated routing mechanism. Being a sub-class of ad-hoc networks, routing is a task handled by the sensors themselves, and not managed by a specific routing device. By cooperating amongst themselves the sensors can carry data throughout the network in an energetically efficient way, through wireless radios, light pulses or ultrasound. Sensorial data is transmitted throughout the

network in “hops”; a sensor transmits its data to the next sensor in the network closer to the data’s final recipient, trusting that the transmission will occur smoothly, without interruptions. This transmission mechanism is useful to distribute the transmission’s energy costs throughout the whole network, avoiding the need to use high potency signals [1].

This makes the use of WSNs in pervasive/ubiquitous environments a more complex challenge, since these environments are populated with many other types of devices and communication mechanisms [7]. All this diversity in devices and communications is managed by the pervasive / ubiquitous application; an example of such an application is PerControle [13], a pervasive system for controlling user attendance through mobile devices and wireless networks.

This work’s main contribution is an analysis of the viability of using ANNs as a base technique to handle user profiles and manage devices on a pervasive environment that also uses WSNs to capture environmental data such as luminosity, temperature and movement. The ANNs and the WSNs are controlled by the pervasive system. An analysis comparing the performance of the system with distinct number of was performed. The results from this analysis are novel in the way that they confirm the viability of ANNs and can serve as initiative to further explore this concept in future research works in the area of pervasive computation.

The article is organized as follows: section II presents the Percontrol system and introduces several important concepts regarding ANNs and WSNs. Section III describes some of the most widely known concepts of routing in WSNs. Section IV gives an overview of the state-of-the-art in the area while section V presents the application scenario. Section VI contains information on the performed tests and shows some preliminary results. To finalize, section VII gives a conclusion and alludes to future work.

II. PERCONTROLE

PerControle [6], [13] is a system that manages and keeps track of user attendance in an automatic way; in other words, the system detects the entrance and exit of users within an environment, be it academic or entrepreneurial. PerControle also bolsters a user discovery and localization service, within the local environment, that is based on Bluetooth, WIFI and RFID. On an academic environment, the system controls student attendance without the teachers intervention; on an enterprise environment, it controls the entrance and exits times of the employees without

the need for their intervention. By resorting to certain aspects of pervasive computing, the system developed in [13] can guarantee reliability and transparency in the detection and localization of users, resulting not only in minimal impact on both the classroom activities or the workers’ service, but also avoiding potential forgetfulness of entrance and exit controls. The system identifies users through their mobile devices, and controls the environment based on their previously-defined profiles. The user / device presence on a certain location generates an event that is sent to a Webservice listener application. This listener application firstly identifies the user within the environment and then from configures the locale in accordance to the user’s profile stored in the system’s database. Environmental data such as temperature and luminosity are acquired through SunSpot wireless sensors [14] in conjunction with Arduino development kits [15]. In the work described in [13], an earlier version of this system was already being developed, but the management of different user profiles and devices on a single environment was yet to be supported. In order to support this feature, an artificial intelligence module was necessary which is now supported through the use of neural networks.

Thanks to its newfound intelligence, the system is now capable of adapting the environment to fit the demands of several users, simultaneously. The ANNs receive data from each user, perform pattern recognition and, by resorting to previous event data, make a decision to configure the environment in a way that is satisfying to most users.

Solving problems with ANNs is an attractive approach. According to [16], the way these problems are internally represented in the network and the natural parallelism associated with ANNs’ architecture create the possibility of a superior performance when compared with traditional machine-learning methods.

In ANNs, the usual procedure for problem-solving begins with a “learning” phase, where a set of examples are presented to the network from which are extracted the necessary characteristics to correctly represent the information provided [8]. These characteristics are then used to generate answers to the problem in question.

An artificial neural network is composed by several simple processing units. These units are connected through communication channels that have a certain “weight” associated to them. Each processing unit performs computational operations upon the data that is fed to it through its connections. The intelligent behavior that results from an artificial

neural network comes from the interactions between the network's processing units.

The most important property of ANNs is the ability to learn from their environment (that is, the input data) and improve their own performance. The entire "knowledge" contained within the network is stored in the "synapses" between the networks artificial neurons; in other words, the networks behavior is mostly conditioned by the weights associated to the connections between individual neurons. Hence, the development of a computational solution that not only infers a system's current state but also predicts its future condition seems to be the solution for the environment management problem previously mentioned. Considering the above characteristics, a neural network is a prediction mechanism that fits PerControle's needs, since it can use raw contextual data, learn from it and predict the configuration values that will better fit a future situation. The solution proposed in [17] was used as basis to adapt a neural network to the current application scenario. This effort demanded an analysis of the characteristics of WSNs in order to find the best configuration and techniques to associate ANNs parameters to Percontrole's WSNs.

The main components of sensor networks are the phenomenon, the sensor and the observer. The phenomenon is the event of interest to an observer, and it is monitored and processed by a WSN. Several different phenomena can be observed concurrently by a WSN. Monitoring is performed under certain criteria that determine the performance requirements for the sensor nodes to be used; this way, the acquisition and dissemination of information throughout the WSN is made in an effectively.

The sensor device is used to collect the data generated by the phenomenon and transmit it to the next node through wireless communication. Nodes respond to changes in the environment where they are currently operation, such as climatic changes, etc. The characteristics and capabilities of the sensor depend on the application and each node's sensibility may differ, depending on the distance to or the exposure time to the event being monitored. The observer is the element that receives the information collected by the WSN; depending on the application's needs, there may be more than one observer at any given time [10].

WSNs can be classified according to their configuration, sensing mechanisms, communication medium and processing type [2]. The type of routing is one of the most important research topics for reducing energy consumption and increase performance in wireless sensing applications.

III. ROUTING

Due to the peculiarities of their utilization (such as their limited battery life), communication between sensor nodes, deployed in an external environment, is subjected to many obstacles that may impede its effectiveness. Most applications that use WSNs communication are directed towards configurations with stationary characteristics, which differ from traditional Ad-Hoc networks, Routing can have its focus on addresses or data, the latter being a more recent approach whose main advantage is the fact that data aggregation manages to reduce the amount of packets exchanged in the network, and thus, reduce the overall energy consumption of the system [3].

Another approach to routing in WSNs is related to the TinyOS operating system, developed by the University of Berkeley. TinyOS does not use a complete addressing scheme like traditional IP-based Ad-hoc networks [3], [4]. Traditional Ad-hoc networks implement a routing protocol that constructs a "*Minimum Spanning Tree*" from an initial sink or access point [4]. The limited resources of a WSN stress the need to use simplified addressing schemes that reduce the size of network packets and the amount of RAM memory needed to store routing information.

Routing in WSNs can be categorized as: based on MAC (Medium Access Control), plane routing, hierarchical routing and geographical routing. MAC-based routing is based on conventional networks that applies techniques to minimize some of the most critical problems in WSNs. The main routing protocols that use medium access control according to [5] are: S-MAC (Sensor-MAC), ARC (Adaptive Rate Control), T-MAC (Time-out-MAC), B-MAC (Backoff-MAC), DE-MAC (Distributed Energy Aware MAC), TRAMA (Traffic Adaptive Multiple Access).

In plane routing, all nodes are considered as equals from a functional point-of-view; routing activity is shared fairly amongst all nodes in the network, without a hierarchical organization. Some plane routing protocols are presented in [2], [5] and [10]. SPIN is a protocol for WSNs that uses information about the amount of energy available in each sensor to manage routing operations. Negotiation protocols are used to disseminate information from a single node to all nodes in the network. Other routing protocols are Directed Difusion, SAR (Sequential Assignment Routing), Adaptative Local Routing Cooperative Signal Processing, Multi, PROC (Proactive Routing with

Coordination), SID (Source-Initiated Dissemination) and STORM/AD. The Mica Motes from University of Berkeley use the “TinyOS Becoming”, a routing protocol directed towards limited hardware to be used in stationary homogeneous networks, where sensory data is periodically sent to a basestation. This protocol assumes sensor nodes to have very limited processing and communication capabilities and presents a very compact implementation.

In hierarchical routing there are two main classes of nodes: source nodes and cluster heads. Source nodes simply collect and send data to their respective cluster head, which may then perform fusion / aggregation tasks before sending collections of data to the gateway [9]. All nodes within a class are considered equal from a functional perspective. Examples of these routing protocols are: LEACH (Low Energy Adaptive Clustering Hierarchy), TEEN (Threshold Sensitive Energy Efficient Sensor Network), APTEEN (Adaptive TEEN), SHARP Hybrid Adaptive Routing Protocol and PEGASIS (Power-Efficient Gathering in Sensor Information Systems). PEGASIS is a protocol for WSNs that is based on the concept of “currents”. Each node exchanges information only with its closest neighbors, forming a current between several nodes, whereas a single of these nodes is chosen to pass the collected information to the gateway.

Geographical routing uses geographical information to route data; this information usually includes neighbor node location [12]. Localization data can be referenced from a Global Position System (GPS) or from a local reference, valid for all the nodes in a network or just for a subset of neighbor nodes. The main geographical algorithms are LEACH-C, ICA (Inter Cluster Routing Algorithm), Geographic Routing without Location Information and GeoMote (Geographic Multicast for Network Sensors). In GeoMote there are three types of sensor nodes: GeoHosts (which produce data), GeoRouters (which forward data collected by the GeoHosts) and the GeoGateways (which act as data entry and exit points) [4]. There is also GEAR (Geographical and Energy Aware Routing) and GPSR (Greedy Perimeter Stateless Routing). Unlike the previous protocols, GPSR allows the addressing of a single node [9]. It uses two algorithms for data routing: when a single node identifies a neighbor that is closer to the final destination, the node sends data to that neighbor. If there is no closest neighbor, the data packet should be sent to a more distant node, in order to avoid areas of poor node-coverage. In these situations, the protocol builds a planar graph to identify the best neighbor to forward the data to [1].

In this section, a resume of some important routing protocols used in WSNs was given. An exhaustive comparative simulation of routing protocols for WSNs is presented in [19], which was used as theoretical basis to formulate the tests and obtain the preliminary results that serve as validation for the work here presented.

IV. STATE-OF-THE-ART

As previously mentioned, this article’s proposal refers to an implementation of ANNs in a pervasive control system for pervasive/ubiquitous environments in conjunction with WSNs (composed of SunSpot wireless sensors, for testing purposes). The system deals with concurrent data and adapts the environment’s configuration to match its users’ profile, according to the devices used to perform login onto the environment.

In the literature [20] we can find several academic works that focus on the detection of devices in many application areas within the domain of pervasive systems and wireless sensor networks, as introduced by Mark Weiser. The work in [20] shows a system that implements several integrated systems that allow the monitoring of patients in their own home. However, this work does not address the integration of different forms of communication within the same environment nor does it present mechanisms for handling user profiles. In Carvalho [21] an extensive proposal that focus on elderly users in ubiquitous environments is presented; the Internet is shown as a form of interaction but this work does not specify which languages, devices and forms of communication should be used. RIBEIRO [22] proposed a project that consisted on the creation of an artificial intelligence system capable of leading a robot through certain routes. The system was tested using a robot-vehicle that possessed freedom of movements (horizontal locomotion). The environment was scanned using sensors whose data was fed to the artificial intelligence, where an algorithm evaluated the situation and acted on the locomotion engines, making the robot avoid possible obstacles and follow the desired route. Despite being an innovative proposal, a target audience and usage scenario are not defined, hence not adapting to the proposed model. Fonseca [23] defines a solution that uses neural networks and RFID, partially achieve the desired results. Despite the idea of uniting neural networks and RFID in pervasive environments being valid and commendable, one of the weak points of this proposal was the language used, while the

existence of other devices and user profile management tasks are not considered.

Percontrole presents all the necessary features for pervasive / ubiquitous environment usage, considering different types of devices, and forms of communication. This work presents a significant improvement over previously presented versions in [6] and more recently in [13], while addressing all the primary objectives previously set. Table 1 presents the main characteristics and functionalities of the works researched in the available literature.

Table 1: Comparison between related research works.

Support and Control WORKS	Scope	Devices Transmission	Language	Public	Routing Protocol
ARAÚJO	ubiquitous computing	Not defined	Not defined	Not defined	Not defined
CARVALHO	ubiquitous computing	Environmental sensors, measuring devices, electronic devices, Internet	Not defined	Elderly people who require constant monitoring of their health	Not defined
RIBEIRO	Neural Networks	Sensor end-of-course and sonar	C/C++/C#, Java	Not defined	Not defined
FONSECA	Neural Networks, pervasive systems	RFID	Delphi	Laboratory of the University	Not defined
PERCONTROLE	Neural networks, systems, pervasive / ubiquitous	WiFi / RFID / Bluetooth Wireless Sensor Networks.	.Net	Educational and business environments	Zigbee

V. APPLICATION SCENARIO

The application scenario demonstrates the potential pervasive / ubiquitous computation has in improving efficiency in the workplace. It also attempts to illustrate different possible perspectives one can have on a single pervasive scenario. Initial versions of Percontrole did not anticipate the use of WSNs or ANNs [6], such versions only intended to automatize teacher’s student-attendance tasks in classrooms. Teaching is a teacher’s main occupation, while keeping track of student attendance is a necessary, but also rather inconvenient secondary task that is prone to errors, interrupts the class, wastes teaching time and may have to be performed multiple times in order to prevent students from leaving. Hence, Percontrole focuses on performing detection of people that enter the classroom environment,

hence performing student tracking automatically. This work proposes an extension of the work developed in [6] and [13], increasing the pervasive functionalities available in this user-tracking system, with the objective of increasing control over environmental conditions through the user’s mobile devices. Using SunSpot wireless sensors [14] and Arduino kits [15] Percontrole can sense and manage the temperature and luminosity of an environment; by using ANNs, the system can also attempt to adjust the values of these environmental properties to fit the individual preferences and the number of users in the environment, turning it into an intelligent location. The sequence diagram in Figure 1 shows the primary interactions between all parts of the system, as well as the messages that are exchanged since a user is detected until the environment adapts to his preferences. When the application detects the entrance of a device in the environment, a webservice that manages the associations between users and devices is accessed. The device is identified through its BDA (Bluetooth Device Adress), WiFi or RFID. The application maintains a module called BlueID, developed in .Net, which holds a list of all devices that were ever detected. Each time the application verifies the devices currently present in the environment, it performs a comparison with the previously stored list; newly detected devices generate an “entry” event while missing devices are associated with an “exit” event.

When accessed, the Webservice returns to the application the person’s username, associated device resources and personal preferences, through the HTTP protocol and an XML format message. The application also communicates with the SunSpot sensors to fetch the room temperature, luminosity, humidity or other environmental data that may be used at a later time. The following format was used to communicate with the sensor: messageID : sensorTypenn. Both the “messageID” and the “sensorType” are numerical values. The messageID field is used to associate the sensors response with the respective BlueID request, an important step since communication is asynchronous.

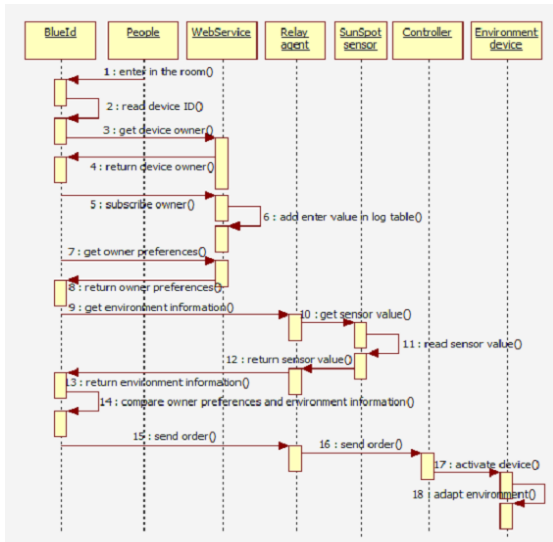


Figure 1: Sequence Diagram.

The sensorType represents the type of data being sent (luminosity, humidity, etc); the “\n” character is used to mark the end of a message, while the “:” character is used as a data separator. The current environmental state is compared with the user’s environmental preferences in order to decide what changes need to be made. After a decision is reached, the environment sends commands to the actuator controllers (connected through USB to an operating computer) in order to change the environmental characteristics (by turning the AC unit on and change the room temperature, for example). The extension of PerControl’s functionalities translated into a more complex architecture, as it is shown in Figure 2.

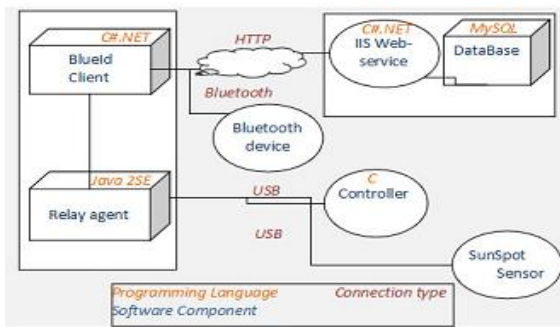


Figure 2: System’s Architecture

Initially, the prototype application and its respective transmitters were tested on a Windows operating system, an environment that benefited from the use of SunSpot sensors [14] and Arduino hardware [15]. There were many other advantages

that led us to choose the Arduino boards, namely the embedded input/output ports, low cost and strong modularity (using the appropriate “shield” extensions). The main idea behind the use of Arduinos was to test their viability for middleware development in pervasive environments, not excluding the possibility of having these boards completely replace the various individual sensors for an integrated, single-board solution connected to a computer. Figure 3 illustrates the functioning of our application.

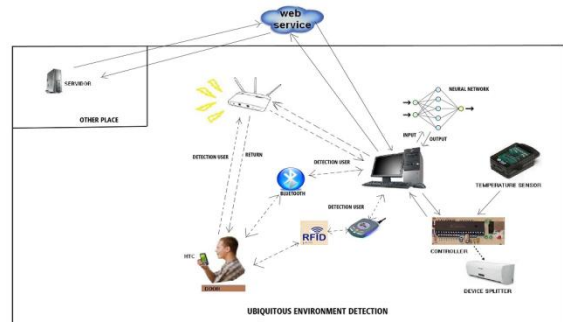


Figure 3: Functioning of Percontrol

From Figure 3, we can see that there will be a need to implement a central controller that will allow the exchange of profiles in the environment, according to the user’s preferences. The main challenge here is related to the number of parameters generated by the application while using WSNs and ANNs, which increase as the amount of sensor nodes increases. This results in larger energetic and resource demands, as well as in increased complexity for the neural network processing, which may compromise the ANN’s response time. An efficient monitoring of the network’s performance is necessary to guarantee a good quality of service. An example of this can be observed in the amount of time necessary to obtain information regarding a monitored environment; if it takes too long to obtain environmental information, this information may lose its value from an application perspective. Management of performance may provide means for the application to define proper quality metrics. These may be influenced by node density, exposition time, amount of dissipated energy and other factors, as specified in the description of the study presented in section 3. A mechanism that evaluates the level of importance of information is necessary for the management of quality of service. For instance, a sensor detecting a temperature of 20±C during spring is a normal occurrence. However, a the measurement of 50±C under the same circumstances is an

abnormal event, which turns this measurement into a relevant situation that requires an extra dose of attention; it could implicate an artificial intelligence mechanism that compares the abnormal value with values measured by other sensors to see if the information is reliable and to determine the proper course of action. Information that is of great importance to the normal function of the system should imply a greater effort for proper delivery. That is, the energy consumed in communications should vary depending on the importance of the data. Another relevant management aspect concerns the installation of ad-hoc networks in unknown areas, where the behavior of wireless communications can be highly unpredictable with high error-rates and considerable delays, which may compromise the value of the information to the application. Performance management usually includes the following sets of functions: quality assurance, performance monitoring, performance control, performance analysis [1]. The QoS management process begins with the detection of performance degradation and ends when the source of the problem is removed. In between, the process has many intermediary stages of situation analysis [18].

A need to perform a comparative study between routing protocols for use in diverse environments and with more wireless sensors arose. There are several techniques for service discovery; one of the most important ones is used by the Service Location Protocol (SLP) [19]. The technique consists on maintaining a directory that contains the services available to whom is offering them.

Hence, we performed some tests to validate our research and obtained some preliminary results that confirm the viability of our work and, consequently, the improvement that these new features bring over previous Percontrole versions. We expect these results to stimulate further research in the area.

VI. TESTS AND OBTAINED RESULTS

One issue with having multiple users on the same system is the problem of concurrent data; for example, the configuration of an air-conditioning unit may be influenced by every user that registers in its environment, since each person might have their own preferred temperature, whereas the temperature itself is constant throughout the whole environment. In order to bypass this problem, the decision-making process for selecting the best “average” temperature must take into account the individual preferences from all users within the environment. A widely used solution [16], [17], [11] that has shown great results is the use of AI, in particular ANNs [8]. A neural

network bases itself on real data that has occurred in previous moments and that has been stored within the system for posterior access and use.

The main objective of this work is not the choice of proper protocols or AI tools, but the creation of novel helper mechanisms for our pervasive application. Our choice for an AI mechanism resided on neural networks, while the routing mechanisms were TCP-IP and ZigBee. These choices were supported by other published works in the areas of routing protocols [19], artificial intelligence [11], and comparison and use of neural networks [8]. The neural network loads the entire history of a device being handled within the environment, and uses its historical data as “training”, in order to identify decision patterns that were assumed in a recent past. Considering our air-conditioning example, these patterns include the temperature that each person wants for a certain environment and what temperature was actually used when all users were taken into consideration. This type of analysis is crucial for the network’s decision-making process. Figure 4 presents part of the source code used to define the desired and assumed temperatures. These values are fixed for testing, but in a real scenario they are fetched from a database or an archive.

```
// Inicializa com
// 1 random
// 3 perception neurons (entradas)
// 6 hidden layer neurons
// 1 output neuron (saída)
net.Initialize(1, 3, 6, 1);

// Taxa de aprendizagem
net.LearningRate = 3;

// Iterações
iterations = 10000;

// Treina a rede
net.Train(input, output, TrainingType.BackPropogation, iterations);
}
```

Figure 4: Part of the neural network’s source code

The code shown in Figure 4 is used to train the neural network. It is possible to insert parameters that will associate this training with a specific user profile, impeding an external attacker to influence the process.

Following the training phase, the next step is to test the network to determine if it is apt to solve the problem of finding the ideal temperature from past event data. In order to perform the testing, a graphical interface was developed. The interface receives the values for current data and returns the ideal temperature estimation, as shown in Figure 5.

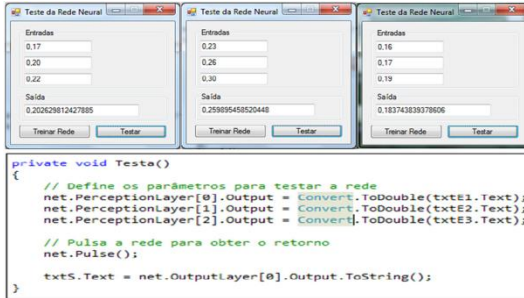


Figure 5: Training the ANNs

After the neural network’s training, we could identify the network’s response time after a user enters the environment, as shown in Figure 6.



Figure 6: Neural network’s performance and response time

Figure 6 presents the response times of the ANNs for the cases with 1, 3 and 10 distinct users identified by the pervasive environment, where the X axis represents the number of users and the Y axis represents the elapsed time. For a single user, the ANN took 3 seconds to process the information contained in the user’s profile, returning an average temperature with a value equal to the one defined by the user (since it is just a single person). For three users, the neural network took 5 seconds to respond, whereas for ten users it took 13 seconds. In Figure 7 a screen is shown, containing information on the users identified by the system, as well as on the devices associated with them.

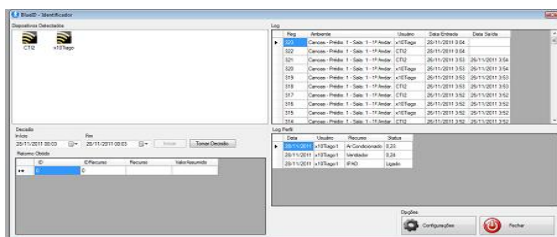


Figure 7: User identification screen.

To perform the identification of different environments we used an Arduino Duemilanove [15]. The Arduino Duemilanove (“2009”) is a microcontroller board based on the ATmega328, that

possesses 14 digital input/output pins (6 of which can be used as PWN outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The board contains the necessary assets to support the microcontroller and using it is as simple as connecting it to a computer with a USB cable or powering it with an AC-to-DC adapter or battery. With this board it was possible to detect devices via Bluetooth, WiFi and, after being integrated with an appropriate card reader, through RFID. The reader fetched the RFID a card’s serial number that can be cross matched with the user’s registration on the database. To this end, a RFID card reader model YHY502CTG was used in conjunction with the Arduino board. After obtaining the necessary application data and performing the necessary adjustments we validated our system using a didactic MultiPIC development Kit which possesses its own internal programmer. In other words, the MultiPIC microcontroller can be programmed directly on the board during the software’s development phase; there is no need to remove the controller and perform its programming on separate equipment. The didactic development Kit was connected a stepper motor that simulated a ventilator. The stepper motor can be put in action with different speeds; initially we defined 3 different speeds that corresponded to 3 different profiles. Figure 8 shows a picture of the assembled device.



Figure 8: Validation equipment

The software used to simulate a ventilator with 3 speeds that controlled the board and stepper motor was developed in C language using the development software from Microchip MTLab, and transferred to the microcontroller with the IC-Prog 1.06C, the software used to compile the source code onto the MultiPIC kit’s processor. On the performed tests the ANNs computed the average temperature from the user profiles and used current environmental information from the SunSpot sensors to correctly manage the ventilation system. From these tests we

conclude that PerControle managed the pervasive environment in a satisfactory manner and that the primary objectives of this research were met, although there is still much room for improvements.

VIII. CONCLUSION AND FUTURE WORK

As suggested in previous sections, the choice of a routing protocol should be based on the restrictions inherent to the observed phenomenon which define the monitoring environment and the minimum requirements for the sensing hardware. The main purpose of the work and results herein presented are to be used as basis for future research work that improves the area of pervasive / ubiquitous computation, namely in the use of ANNs with low response times in environments with thousands of users. These future tests should confirm that the routing protocol in use can be inserted into the BlueID module. The BlueID module should also be converted from .Net to C in order to avoid the need to use bulky language converting frameworks. The conversion to C also reduced the amount of energy required from sensors, since the number of message exchanges is reduced. Meanwhile, it will be also necessary to exhaustively study the functioning of the routing protocols that best adapt to the peculiarities of our pervasive system.

The performed tests confirm the viability of device detection with WiFi, Bluetooth and RFID, an improvement over previous PerControle versions. Nevertheless, there is still some latency in registering new devices on the system, which may be reduced by further adjusting the parameters sent to the ANNs. Still, the presented work represents a significant contribution since it covers different areas and technologies within pervasive computation. On top of several parameters and definitions still waiting implementation, future work needs to define a model and implement a mechanism for privacy control, for both users, their devices as well as for the pervasive environment itself.

ACKNOWLEDGMENTS

This work is associated with the UbiArch project - Ubiquitous Architecture for Context Management and Application Development of the Rio Grande do Sul's Federal University. We would like to thank the *Conselho Nacional de Desenvolvimento Tecnológico*

CNPq for its support. We would also like to thank Sun Microsystems for donating the SunSpot development kits, which allowed us to perform the necessary tests for our study. We kindly thank the students Sebastien Skorupski, Thiago Tomasel, Marcell Stein and Jeison Nunes da Silva which contributed towards the completion of this work.

REFERENCES

- [1] Loureiro, A. A. F. et al, *Redes de Sensores Sem Fio*. Disponível em: http://www.sensornet.dcc.ufmg.br/pdf/179_Loureiro_Nogueira_Ruiz_Mini_Nakamura_Figueiredo.pdf. Acessado em jun. 2008.
- [2] Luz, D.L., “Roteamento em Redes de Sensores”, IME -USP, São Paulo, 2004.
- [3] Cabrini, F.H., Kofuji S.T., “Roteamento em Redes de Sensores Sem Fim”, Laboratório de Sistemas Integráveis-USP, São Paulo, http://www.pad.lsi.u.br/humanlab/trabalhos/workshop_roteamento.pdf.
- [4] Lima, dos Santos, L. Um protocolo para descoberta e seleção de recursos grades móveis Ad Hoc, Tese de Doutorado, PUC RIO 2007, Pág. 27 e 28.
- [5] SILVA, R. ; LEITHARDT, V. R. Q. ; SILVA, S. J. GEYER, C. F. R. ; J. Rodrigues ; BOAVIDA, F. . A comparison of approaches to node and service discovery in 6lowPAN wireless sensor networks.. Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks. New York, NY, USA : ACM, 2009. v. 5. p. 44-49.
- [6] Ayres, C. ; LEITHARDT, V. R. Q. ; Everson Figueiró ; Hunder Correa ; Luis Henrique Leal Ries. Controle Acadêmico Pervasivo. In: XXI Salão de Iniciação Científica (UFRGS), 2009, Porto Alegre. Controle Acadêmico Pervasivo. Porto Alegre: Editora da UFRGS, 2009.

- [7] M. Weiser, "The computer for the twenty-first century." Scientific American, vol. 265, no. 3, pp. 94–104, Setembro 1991.
- [8] Carlos O. Rolim, Anubis Rossetto, Valderi R. Q. Leithardt, Cláudio F. R. Geyer "Análise de uma Rede Neural Híbrida como base para um Mecanismo de Predição de Situação". in Congresso da Sociedade Brasileira da Computação (CSBC 2012) Simpósio de COmputação Pervasiva Ubíqua. Curitiba, Brasil 2012.
- [9] Silva, R. N., Leithardt, V. R. Q., Silva, J. Sá, Geyer, C.F.R; Joel, J. R., Boavida, F. "A Comparison of Approaches to Node and Service Discovery in 6lowPAN Wireless Sensor Networks". Tenerife, The Canary Islands, Spain; The 5th ACM International Symposium on QoS and Security for Wireless and Mobile Networks.
- [10] Augusto, C. H. P., Carvalho, C. B., Silva, M. W. R., and de Rezende, J. F. - "REUSE: A Combined Routing and Link Scheduling Mechanism for Wireless Mesh Networks", to appear in Computer Communications (COMCOM), 2011. ISSN: 0140-3664. English, A4 size, 10 p.
- [11] LUDWIG JR., Oswaldo; COSTA, Eduard Montgomery Meira. Redes Neurais: fundamentos e aplicações com programas em C. 1 ed. Rio de Janeiro: Ciência Moderna Ltda, 2007.
- [12] SILVA, R. ; SILVA, S. J. ; GEYER, C. F. R. ; LEITHARDT, V. R. Q. ; BOAVIDA, F. . Use of GPS and 6LowPAN in Mobile Multi-Sink Wireless Sensor Networks - Issues and Feasibility.. In: Proceedings of the 8th International Information and Telecommunication Technologies Symposium (I2TS), 2009, Florianópolis - Brasil.
- [13] Leithardt, V. R. Q.; Rolim, C.; Rosseto, A.; Geyer, C.; Dantas, M. A. R.; Silva, J. Sa; Nunes, D. "Percontrol: A pervasive system for educational environments". In ICNC 2012, Hawaii US.
- [14] <http://www.sunspotworld.com/> - Ultimo acesso Maio 2012.
- [15] <http://www.arduino brasil.com/> - Ultimo acesso Maio 2012.
- [16] BRAGA, Antonio de Padua; LUDERMIR, Teresa Bernarda: "**Redes Neurais Artificiais: Teoria e aplicações**", (2007), Rio de Janeiro: LTC.
- [17] Rolim, C. O. ; Fernando Schubert ; Rossetto, Anubis G.M. ; LEITHARDT, V. R. Q. ; GEYER, C. F. R. ; Westphall, Carlos . Comparison of a Multi output Adaptative Neuro-Fuzzy Inference System (MANFIS) and Multi Layer Perceptron (MLP) in Cloud Computing. In: XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2012, Ouro Preto. X Workshop em Clouds e Aplicações - WCGA. Ouro Preto : SBC, 2012.
- [18] Pereira, V.; Silva, J.S.; Granjal, J.; Silva, R.; Monteiro, E.; Qiang Pan; , "A Taxonomy of Wireless Sensor Networks with QoS," New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on , vol., no., pp.1-4, 7-10 Feb. 2011.
- [19] LEITHARDT, V. R. Q. ; GEYER, C. F. R. ; Tavares . Estudo Comparativo entre Protocolos de Roteamento Utilizados em Redes de Sensores sem Fios. In: Congresso Trinacional da Fronteira, 2009, Foz do Iguaçu. IV Congresso da Acadêmia Trinacional da Fronteira. Foz do Iguaçu : PTI Itaipu Binacional.
- [20] ARAUJO, Regina Borges.(2003) Computação Ubíqua: Princípios, Tecnologias e Desafios. Departamento de Computação – Universidade Federal de S. Carlos (UFSCar). Disponível em http://im.ufba.br/pub/MAT570FG/LivroseArtigos/04_5_AraujoRB.pdf . Acessado em setembro 2011.
- [21] CARVALHO, Sergio Teixeira. COPETTI, Alessandro, LOQUES, Orlando Gomes. Sistema de

computação ubíqua na assistência domiciliar à saúde. Journal of Health Informatics (JHI), vol. 3, no. 2, p. 51-57, 2011.

[22] RIBEIRO, Dionisio R. de Souza, MACHADO, Kássio L. da Silva, PEREREIRA, Levi C. Azevedo. Rede Neural Perceptron embarcada em robótica. Instituto de Estudos Superiores da Amazônia. Belém, 2008.

[23] FONSECA, Lucas Oliveira. Sistema de localização RFID de usuários visando à racionalização de energia em ambientes inteligentes. Universidade de Brasília. Brasília, Fevereiro de 2011.

**Anexo C “Quietly: A model privacy for pervasive and
ubiquitous environments”**

Anexo D “Proposta Sapo Labs 1”

Proposta Labs.Sapo

Nome do projecto – Social Ubiquity – Youth5G

Abstract/Resumo – *“Resumo sucinto do que é o projeto, abordando os principais problemas científicos e tecnológicos a tratar e, quando relevante, deve incluir uma breve análise do estado da arte. Deverá ainda destacar os aspectos de inovação dos conceitos apresentados e os resultados expectáveis no âmbito desta proposta.”*

The Internet is one of mankind’s most recent advancements. Yet, in the short time it has existed, it has already transformed the way our world and our society work at a very fundamental level. It has also done so at faster pace than any other massively important technological advancement. As the Internet evolved, social networking bloomed and brought together an astonishing number of users. At the same time, sensing technologies like Wireless Sensor Networks are becoming more robust and viable while the sensor nodes themselves are becoming cheaper. Modern mobile phones and smartphones have also brought us mobility in our Internet access and are also a powerful computational resource, with many interesting capabilities. Mobile phones are also becoming increasingly more powerful, common and cheaper. Never, in its entire history, did mankind have such a collection of technologies that are capable of connecting such large number of users simultaneously, at a global scale.

On the other hand, while the Web and its associated technologies will continue to evolve at increasingly faster paces, the most likely path in our technological evolution is a merge between the real and the virtual: a society where people are connected not only by their physical proximity, but also by their virtual closeness. This new trend in our civilization’s evolution is still making its first steps. Our generation is on the verge of a virtual revolution that will, in the next few decades, transform our society and open the path to many new market opportunities.

With these technological and social trends in mind, we propose Youth5G, a system that combines social networking with mobile sensing technologies to expand and provide greater functionalities to social networks and other web applications. In short, this system will focus on bringing a “real-time” component to social networking, by sharing and exploring real-time information about the users’ daily actions, social activities and visited locations. By using body-area networks and mobile phones, we intend to research and develop mechanisms that focus on “people-centric sensing” and provide a seamless connection between the virtual (social networks) and the real (the user’s daily lives). This connection brings greater value not only for the user’s social experience but also for the service provider, as the newly available information can be mined and analyzed to create more personalized product advertisement mechanisms and improve client group classification tasks.

With Youth5G we propose to implement several specific applications. Our base system will use sensors and mobile phones to detect users' activity and localization and display this information on social networks (for example, Facebook), allowing for the automatic detection of social events. On top of this base system, we intend to implement specific services that take advantage of the system's capabilities. The "Nightclub Music" service combines localization with the users' musical interests to allow DJs to know the favorite musical genres of people currently frequenting their establishments and also to share their tracklists in social networks, allowing people to know what kind of music is passing on each local nightclub beforehand. This service could be used to expand a current music service, such as SAPO's Music Box. We also consider a "Targeted Advertising" service that combines localization, activity, social network relations and information about the user's interests to provide targeted marketing, which is intended to be more meaningful and effective than common advertisement techniques.

We believe that the Youth5G system is a flexible and promising new technology with many possible applications of great value for groups that are on the forefront of communication technologies, such as SAPO and the PT group.

Trabalho desenvolvido – *“As propostas de projeto que tenham por base a continuidade de projetos que já possuam alguma maturidade, nomeadamente através de componentes já desenvolvidos em contexto letivo ou através de projetos desenvolvidos, ou em curso, com outras fontes de financiamento, são valorizados pelo labs.sapo. Neste cenário é relevante identificar as mais valias que o trabalho desenvolvido representa para esta proposta e o modo como se podem complementar. É igualmente relevante distinguir outros trabalhos de investigação desenvolvidos na área em que a proposta de enquadra.”*

This SAPO proposal intends to extend the work done by our group in the European project "GINSENG" (<http://www.ict-giseng.eu>), by providing scalable and resilient solutions for social environments. Most of our previous work will be crucial to implement the proposed system that requires reliable Wireless Sensor Networking systems.

The research herein proposed is based on a testbed already implemented in our labs. In our current model, Wireless Sensor Networks (WSNs) are data and functionality providers, facilitated by an integration M2M platform. The M2M platform provides a set of services for consumption by external environments using web service APIs. This model facilitates the integration and interoperability of WSNs with virtual world environments, social networks and other web 2.0 applications.

Our current approach is already implemented and it is based on a hybrid gateway-based model that supports both proactive and reactive models of M2M communications. In proactive sensor networks, the sensor nodes periodically collect data and send it to the sink nodes for processing, storing and management tasks. In reactive models, end nodes or sink nodes will send data/functionality requests to

the sensor nodes, which will then react accordingly. A simplified illustration of our current prototype is depicted in Figure 1 of this proposal's "Youth5G_figures.pdf" attachment. The model uses multiple proxies/Gateways that mediate the access to the sensor nodes. One of the main advantages of this approach is that it can easily offer mobility, localization and scalability. The proxies/gateways are designed to be lightweight, so that they may be installed on modest or limited devices.

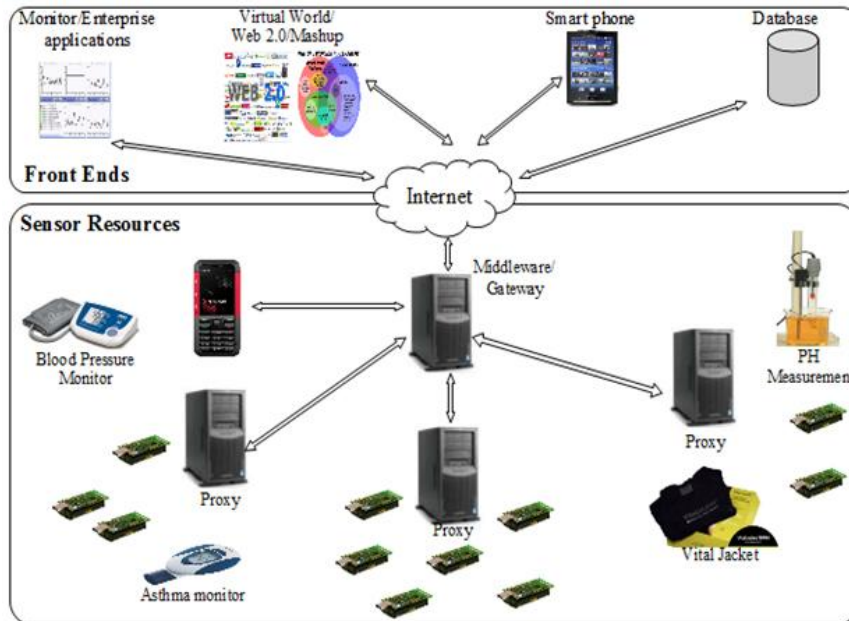


Figure 1 – Current prototype's model

Interoperability is a difficult problem when developing WSNs applications, due to the independent proprietary data formats used by different sensor networks. Although it has been proved that it is possible to implement both IP protocol stack (6LowPAN) and web services on the sensor nodes, the applicability and efficiency of these technologies still needs more investigation. Because of the limitations of sensor nodes in terms of computational resources and energy, allowing direct access to them from the Internet brings many challenges and may not be a suitable solution for all application scenarios. Therefore, it is necessary to have an infrastructure that easily, securely, and efficiently supports the integration and interoperability of sensor networks with other environments.

The members of this proposal have several work in this area and they have already presented 2 Internet drafts to IEEE 6LowPAN group:

M-K. Shin, T. Camilo, J. Sá Silva, D. Kaspar , "Mobility Support in 6LoWPAN", draft-shin-6lowpan-mobility-00, IETF Draft, June 2007

R. Silva, J. Sá Silva, draft-silva-6lowpan-mip6, "An Adaptation Model for Mobile IPv6 support in lowPANs", IETF Draft, May 2009

These drafts support the direct connection between mobile devices and Internet equipment by the use of IPv6. The mobile and IPv6 connections is crucial for the proposed devices.

The sensor nodes' mobility is supported by a proxy model proposed as part of one of our model, as described in the IETF drafts.

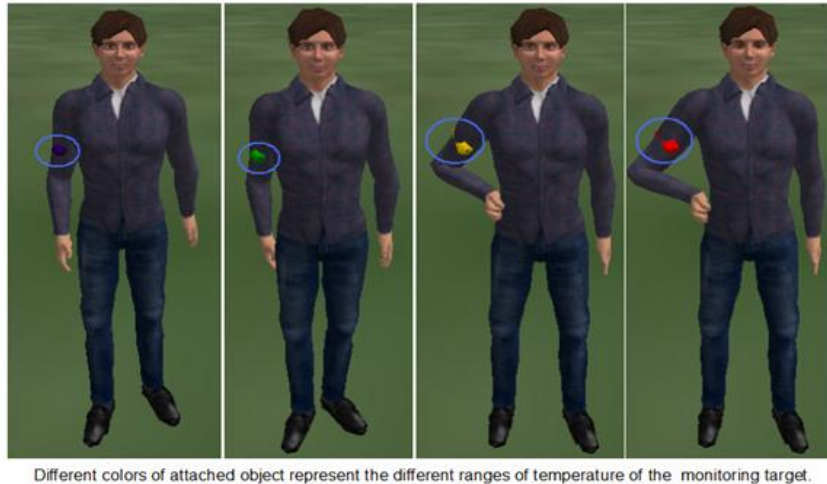
The middleware gateway exposes the functionality and data of sensor networks as web services. In a sense, the M2M middleware is the basis for the integration and interoperability between sensor networks and other Web 2.0 environments. This facilitates the development of applications using sensor networks, increases the diversity of possible applications, and promotes the reuse of WSN deployments and software code. In addition, the data coming from the sensor networks could easily be stored in a cloud data warehouse for further processing. The most recent implementation of our model supports 6LowPAN WSNs and uses the RSSIs of the received packets to perform localization. The communication between proxies and middleware is based on TCP/IP and Restful web services, currently supporting XML and JSON as the data formats. The communication between proxies and sensor nodes is implemented independently, using any protocols that are appropriate. It is based on open protocols such as IP or propriety ones. Similarly, the gateway middleware exposes the sensor's data and functionality through a set of services with JSON and XML formats. In addition, the middleware also supports localization services based on RSSI and a trilateration algorithm.

Youth5G was also thought to be able to support in the future the use of several medical devices. In our current implementation, we use a small heart beat monitor (also depicted in Figure 1), whose communication is based on the Bluetooth protocol. The use of these external devices requires an intermediate device that supports both the external protocol (Bluetooth) and communication with the middleware. In our model, both mobile phones (for increased mobility) and computers can serve as gateways for communication with these types of devices. For implementations that focus on activity detection and people-centric sensing, the mobile phone serves as the primary gateway for communication with virtual worlds and social networks, in order to promote mobility.

One of the main features of our system is the robustness of communication between small sensor devices. In this context, the FP7 GINSENG project (<http://www.ict-ginseng.eu>) became an important landmark in the field by proposing and developing a novel performance controlled WSN model to be deployed in critical industrial environments, like the refinery of Petrogal in Sines. The overall goal of GINSENG was a wireless sensor network that met application-specific performance targets that integrated with industry resource management systems. The project's resulting technologies were validated in real industry settings, where performance is critical.

To illustrate the proposed approach, two simple applications that integrate sensor networks with both Facebook and Second Life have already been implemented. These applications allow Facebook and Second Life users to easily access sensory information, about the physical world. In the first application, Facebook is employed to publish sensor data. In this simple demonstration, we publish the temperature data of a monitoring room in the wall of a Facebook account. This helps the account owner to instantly

know the temperature of the room and act appropriately in case it unexpectedly changes. The second application that we implemented was a mashup of WSNs with the Second Life Virtual World. It is similar to the first, but instead of displaying sensor data as plain text we used virtual objects. The temperature of a monitoring target (such as a person) was represented by changing the color of an object attached to an avatar. For example, if the temperature was between 10 and 19 degrees Celsius, the color of corresponding object would become green or, if the temperature was greater or equal 30 degrees Celsius, the color would be red. Figure 2 of this proposal's "Youth5G_figures.pdf" attachment shows some screenshots of this second application.



Different colors of attached object represent the different ranges of temperature of the monitoring target.

Figure 2 – Displaying the temperature sensed by sensor nodes using an object attached to an avatar in Second Life

Relação com outros projectos – *“Se este projeto se relacionar de alguma forma com outro projeto do universo PT ou outros projetos/produtos de outras empresas e instituições, deverão ser descritas de que forma ambos se aproximam e quais as mais valias resultantes para os envolvidos.”*

While our proposal is not a fruit of a previous relation with a SAPO's or PT's project, we believe that our idea has a lot of potential and can be easily applied into many of PT's current platforms. Our focus is on using body-area-networks, activity detection, localization and privacy mechanisms in conjunction with a social networking frontend. This frontend can be in the form of a popular social network (such as Facebook), part of a PT's platform (for example, a service provided as part of the MEO solution), or even a conjunction of the two.

Access to social networking nowadays is becoming mobile. It is not uncommon to see people access their smartphones to share and discuss their daily experiences shortly after they happen, updating their thoughts and responding to feedback from their friends as the situation develops. People-centric sensing systems gather and share several types of personal information. Information about a user's daily activities, social events and visited stores / locations can be very interesting for a group focused on

communication technologies, such as PT. It is possible, for example, to correlate information about the user's whereabouts with his hobbies and interests to determine the closest stores that might sell items of his interest. This would result in more personalized advertisement mechanisms that are both more useful for the user and more effective for the service provider. In the "Descrição" section, we discuss several usecases that explore how our idea might be of value to the PT group.

Descrição - *"Deve ser redigida explicando em detalhe o que se pretende fazer no projeto, quais os objectivos a alcançar e as razões da sua importância. Pretende-se que descreva qual é o problema a ser investigado, quais são os desafios, quais as abordagens e as respectivas metodologias para alcançar os objectivos. Deve ainda mostrar que as metodologias propostas são adequadas para alcançar os objectivos. Neste ponto, deve abordar os aspectos tecnológicos mais relevantes do projeto e também os aspectos organizativos necessários para a sua execução."*

Youth5G intends to define models for frameworks that integrate wireless sensor networks with mobile phones and expose their functionality as a service to Web 2.0 applications, namely to social networking websites. To apply our research work we plan to implement several functionalities which we will describe through several use-cases in the next paragraphs.

A social network user goes on a night-out with some of his friends to a local nightclub. He has an activity detection system that runs on his smartphone, possibility accompanied by several small body-area wireless sensors (in the form of a small bracelet, for example). The small sensors in his body directly communicate with the smartphone through a body-area wireless sensor network and share information regarding several conditions, such as his movement (accelerometer) and heart beat. The sensors can also detect environment factors such as the temperature. The smartphone runs our activity detection algorithms that combine the sensory information (for example, an increase in heart rate, temperature and movement) to discern what the user currently doing, in this case, dancing. The smartphone accesses the Internet and communicates with the user's preferred social network to represent information about the user's current activities to his trusted group of friends. The information can be represented through a virtual representation of the user and the environment. If the user's friends also use the same system, it is possible to correlate their close physical proximity with activity detection and deduce that the group is currently engaging in a social group activity (they are all dancing in a nightclub). People who belong to this same group of friends who are still at home and were indecisive as to whether or not go to the nightclub can now have an idea of who is attending the social event and where it is happening, and use this information to make their decision.

The integration of sensing systems with social networks opens many interesting possibilities other than the automatic detection and creation of social events. Tasks that involve the classification of users into groups, for example, can benefit from these technologies. Instead of creating user groups based on similar interests or other "static" user- entered information, we can correlate localization and activity to

find people that both share similar activities and are physically close. This is useful for improving “friend suggestions” or even for dating services. It is also useful for market research, as information about people’s location and activities can provide further insight on their product interests and client profile.

In this project, we also intend to implement more ambitious services that take advantage of the system’s basic functionalities: we suggest a service that combines localization with the users’ interests. Information such as the user’s current activity is usually too sensitive to be openly shared outside of the user’s trusted group. However, most social networking users openly share information about their interests, such as their favorite music genres. Thus, we will provide a “Nightclub Music” service that allows the local nightclub’s Disc Jockey to know the favorite musical genres of people currently frequenting his establishment. While the DJ would not be allowed to access individual information, he could still get a “general feeling” about the kind of audience that is currently enjoying his music, and adjust his performance accordingly. DJs could also share information regarding the tracklist currently playing in their nightclubs. By accessing their social network, users could not only know the establishments currently frequented by their friends, but also know what kind of music is playing. Thus, this service would also be particularly useful for people deciding which nightclub to go to. Since we have information about the users’ whereabouts we can also correlate the nightclub’s tracklist information with the users’ nights out and the people they were with.

People usually tend to associate music with emotionally rich moments or with people they feel strongly about. Our service will explore this concept and display a list of “the music you heard when you went out with your friend last night”, associated with a music service, such as SAPO’s Music Box, that would allow the user to listen to or even buy the songs. When we associate Music Box with a Facebook account it automatically discovers which of our friends also have access to the system. From here, SAPO’s Music Box allows us to create and share our own playlists and view our friend’s musical preferences. While the Social Networking integration itself was interesting, we found that the features it provided were rather simplistic. We believe that Music Box’s social networking features can be made much more attractive and useful with people-centric sensing systems. From our understanding, Youth5G could be integrated to expand the current Music Box service.

This could also be an interesting service for all those times when people listen to music they enjoy and wish to know the name of the song. We can easily imagine many other possible similar use cases. User’s musical preferences and localization could be used to suggest concerts or musical events, and provide links for the users to buy tickets. Our proposed system would be flexible enough to be applied in different contexts and could easily be extended to suggest more than just music.

Localization can also be correlated with the available stores in the area and with the user’s preferences and interests. Advertisements on the social network or the application portal would be targeted towards that particular user, showing advertisements of stores that are in both in his current physical location and in accordance to his defined interests. It is also possible to extrapolate more information about the user’s interests by analysing the locations he has visited: the stores he usually visits tells us the type of

products he usually buys, his workplace can tell us about possible work-related needs, his spare-time activities tells us more about his hobbies.

Let us consider a "Targeted Advertising" service that shows how we can apply these systems in targeted marketing. Like many other male teenagers of his age, John usually has a friendly football match with his friends every Thursday in a local football field. We cannot expect John to wear a complex activity detection system with many sensors while practicing sports, but systems limited to the John's smartphone and possibly a small wrist-strap are simple and practical enough to accompany the user for most of his day. Even if the user is not wearing his smartphone while practicing the sport, by knowing his location (the football field) and detecting an increase of his heart-rate (from running) we can extrapolate the user's physical activity and deduce that he is engaging in a sports match. After the match, John has a shower and is ready to go home. As he accesses his smartphone and logs in his social network, he is greeted with several suggestions for energy drinks, sodas or water from nearby supermarkets, as he might be thirsty from all the running. Since the system can also detect who else participated in the football match, it may also suggest footwear or other sports equipment as "better equipment for the next time you play with (one of the user's friends)".

The possibilities are endless. Online stores can associate what was bought by the user's friends and correlate this information with the user's hobbies and activities to present personalized advertisement suggestions. For example, our friend John might receive footwear suggestions based on the products bought by his friends. These suggestions appear associated with the user's social group activities, in the form of appellative announcements, such as "did you know that (one of the friends the user just played football with) recently bought a (model and brand) sneakers?"

The state of the art in the area still represents only the first steps for these kinds of systems. There is plenty of room for improvement in various areas, encouraging future research. In fact, creating such systems presents several technical challenges that we intend to address with our research, namely:

- **Privacy** – Exchanging sensitive information such as user's vital signs, localization and current activity is a task that has strict privacy requirements. Privacy is a requirement that affects all application layers: from the sensor's wireless communication to the exchange of middleware messages. During our research we intend to employ effective privacy mechanisms that ensure the safety of sensitive information.
- **Managing intermittent connectivity** – Reliability of wireless sensor networks is very important in order to create systems that are robust and stable. However, due to the wireless nature of mobile phone and WSNs communication, there is always a risk of intermittent connectivity due to environment interference, power shortages or communication range. Robust people-centric sensing systems should be capable of maintaining an illusion of smooth and continuous people tracking. Offline mechanisms that can estimate a user's current activity and location even when there is a shortage of fresh data can maintain an illusion of continuous monitoring in adverse

conditions. These offline mechanisms can also be employed in data mining and analysis, since their function is to find patterns in the data.

- **Detection of activity** – With basis on previous work in the area, our research will attempt to improve and design resource aware algorithms that use both the mobile phone and wireless sensors and are directed specifically towards the representation of activities in social networking sites and virtual communities. At the same time, we will also seek interesting and engaging ways of representing user activities and environmental data, through the use of virtual representations of users (virtual avatars).

Hence, this research proposal is not limited to simply developing an application; we intend to design a model based on innovative solutions that effectively address these technical challenges. Small and discreet wireless sensors allow for more diverse kinds of vital-signs data to be used, which permit the creation of more accurate activity detection systems. Activity detection amongst groups of friends can be correlated to automatically create social events that can be shared in Social Networks. Localization techniques can be used to permit users to locate their friends in big social events. Offline inference techniques can provide an illusion of seamless connection between the real and the virtual, by trying to infer current location and activity even when there is no fresh data due to a connectivity problem. All of these research approaches attempt to expand the features of social networking sites and increase their functionality beyond the simple sharing of photos and comments.

From the beginning, our vision was not to just propose a M2M model for the integration of WSNs with virtual world environments, but to expand this model towards people-centric sensing and include mechanisms to provide localization, vital signs and activity detection, using mobile phones and wireless sensor nodes as base hardware. We intend to offer to CSPs a M2M platform that combines in a rich environment correlated information of real and virtual environments. Such tool would be able to increase the customer insight from the CSP perspective as also to create a new set of software products based on the real/virtual world relationship.

Plano do projecto – *“Indique uma lista calendarizada de etapas e milestones que permitam aferir se os trabalhos do projeto estão a progredir face aos objectivos apontados na candidatura. Para cada etapa inclua uma designação (máx. 60 caracteres), uma data (mensal), e a descrição (máx. 300 caracteres) do que pode ser demonstrado ou reportado nessa data.”*

The project is prepared to run during 12 months.

- 1) Model development (3 months)
- 2) Implementation, model validation and assessment (6 months)
- 3) Complementary studies (3 months)

4) Project management, dissemination and exploitation (12 months – permanent)

Task 1 – Model development

By studying other projects in related areas the main aspects and capabilities that can and should be explored in the context of the proposed project will be identified. As a result, a white paper will be produced, that will discuss both relevant projects and the main players in the field and also discuss the potential of combining real information from wireless sensor networks with virtual models and social networks. The white paper will also identify the specific issues that will be explored in the project and the recommended methodology.

Still inside the context of this task, the general system architecture, paradigms, algorithms and mechanisms will be also specified. Although the proposed model will be detailed in the first two months of this task, it will be constantly refined during the project's lifetime.

Task 2 – Implementation, model validation and assessment

This task will lead to an implementation of the model developed and proposed in Task 1. This, of course, will require the setting up of project testbeds in University of Coimbra, and the implementation of several components of the model. The component implementation will be a continuation of our current work in the form of refinements on our current prototype. This includes sensor programming, improvement of the middleware to support more web applications and provide greater functionality and mobile phone programming. At the same time, we will work on creating web frontends that can be integrated with current social networks to allow access to our system and present sensory information in engaging ways (through virtual avatars and environments, for example).

This task also intends to evaluate the proposed model, from the functional to the performance-wise point of view. Functional validation will address issues pertaining to the compliance with the developed specifications. Performance validation will try to estimate the efficiency and effectiveness of the proposed mechanisms. An assessment of the complexity and feasibility of the proposed solution will also be an important outcome of this task. For this end, we intend to deploy a prototype application to be used within our University Campus that will allow us to test our ideas and validate our technical solutions.

Task 3 – Complementary studies

The results expected in this task will contribute to complement the model validation and assessment in task 2. This will allow the evaluation of the proposed model in more complex scenarios, beyond the ones implemented on the laboratorial testbeds.

Task 4 – Project management and dissemination

This task will provide the necessary project management and administration, in order to guarantee an adequate and timely achievement of results. It will also provide the dissemination and exploitation of project results.

Recursos necessários – *“Aqui deve especificar quais são os recursos necessários do labs.sapo para a execução do projeto que está a propor. Nos recursos deve indicar todo o tipo de suporte que tenha que ser fornecido pelo labs.sapo, nomeadamente: acesso a infraestrutura tecnológica, equipamentos, recursos humanos e dados. Os recursos humanos devem ser expressos em bolsas de investigação, sendo necessário especificar quais as milestones (descritas no plano do projeto) que serão executadas pelo bolseiro. Deverá ainda especificar a duração prevista para as bolsas.”*

Human Resources - this project requires one full-time research engineer. This researcher will carry out his/her work under contract, in order to guarantee an adequate degree of commitment and a minimum level of human resources fully dedicated to the project.

Missions - diffusion of project results not only through scientific communities but also through social communities. This step will produce reports, create webpages, and present technical papers in international meetings and conferences.

Overheads – 30% over the total amount, required by the host research centers.

Equipment – 2 PCs, 1 laptop PC, Sensor Nodes, Mobile phones

Resultados esperados – *“Após a conclusão do projeto quais os resultados esperados e de que forma estes constituem uma mais valia para as partes envolvidas. Se aplicável, poderá identificar potenciais modelos de negócio a explorar futuramente entre SAPO e a sua instituição.”*

By the project’s conclusion, we will have developed a working prototype of our people-centric sensing system together with scientific results in the form of models and technologies that were researched in the context of the previously presented technical challenges. This project’s resulting models,

technologies and architectures can also be applied in many other areas; for example, in health monitoring systems. Beyond the scientific value of these solutions, we believe that this project's use cases are of high value for SAPO and the PT group. Our prototype would be a novel system that goes beyond the current vanguard of social networking, bringing several technical features into play that can be useful for **data-mining** and **targeted advertising**. Real-time activity and localization can tell us a lot about the user's daily lives, preferences and frequented places. This information can be applied by CSPs in different ways for many commercial solutions, as described by the use-cases in the "Descrição" section.

In a way, the prospect of taking advantage of people-centric sensing systems can be a bit scary, but current trends point towards a future where such systems will most likely be very common. It is part of people's nature to gossip about the lives of others and share details about their personal lives, and they have been doing so for the past millennia. Our research proposal is innovative in the sense that it tries to anticipate this trend and harness these new market opportunities, allowing for more personalized advertisement systems that benefit both users and companies.

Anexo E “Proposta Sapo Labs 2”

Proposta Labs.Sapo

Nome do projecto – HappyHour.SAPO

Abstract/Resumo – *“Resumo sucinto do que é o projeto, abordando os principais problemas científicos e tecnológicos a tratar e, quando relevante, deve incluir uma breve análise do estado da arte. Deverá ainda destacar os aspectos de inovação dos conceitos apresentados e os resultados expectáveis no âmbito desta proposta.”*

(1500 caracteres)

A presente candidatura propõe-se implementar um sistema inovador que permitirá aos estabelecimentos comerciais, de uma forma simples e em tempo real, implementar políticas de marketing e divulgar os seus produtos a potenciais clientes que se encontrem nas proximidades. Para isso pretende-se criar micro-comunidades cruzando informação de geo-localização de utilizadores e os seus perfis de redes sociais.

Embora já existam algumas ferramentas que cruzam informação entre comunidades das redes sociais, a lógica de negócio do HappyHour.SAPO não está embebida em ferramentas como o FourSquare ou o Waze. O sistema proposto possibilita reunir numa base de dados central a geo-localização de utilizadores que instalem a aplicação HappyHour.SAPO nos seus dispositivos móveis. Esta base de dados irá permitir cruzar a informação geo-espacial com os gostos pessoais destes utilizadores, reconhecendo potenciais clientes nas imediações aos quais os estabelecimentos comerciais poderão divulgar promoções e actividades. Por exemplo, a aplicação HappyHour.SAPO permitirá aos bares aderentes divulgar a todos os utilizadores que se encontram a menos de 700 metros que nas próximas duas horas os preços das bebidas terão uma redução de 50%. Paralelamente, poderão também, em tempo real, reconhecer os gostos musicais dos clientes que se encontram nas proximidades e adaptar as suas músicas a estes perfis.

Trabalho desenvolvido – *“As propostas de projeto que tenham por base a continuidade de projetos que já possuam alguma maturidade, nomeadamente através de componentes já desenvolvidos em contexto letivo ou através de projetos desenvolvidos, ou em curso, com outras fontes de financiamento, são valorizados pelo labs.sapo. Neste cenário é relevante identificar as mais valias que o trabalho desenvolvido representa para esta proposta e o modo como se podem complementar. É*

igualmente relevante distinguir outros trabalhos de investigação desenvolvidos na área em que a proposta de enquadra.”

(2000 caracteres)

Foi já implementado um pequeno protótipo laboratorial, baseado apenas na divulgação de músicas, que se encontra pronto para demonstração local e que poderá também ser visualizado através de um pequeno vídeo colocado em: <http://www.youtube.com/watch?v=L-ORxaFcDy8&feature=plcp>. Em particular, a aplicação desenvolvida é baseada em *smartphones* com sistema operativo Android e na integração de uma interface Web com a popular rede social Facebook. A estrutura do protótipo implementado é apresentada na figura 1 em anexo.

O protótipo actual permitiu estudar também algumas das funcionalidades que serão necessárias na implementação do HappyHour.SAPO. Este protótipo baseou-se numa aplicação Android que é executada a partir do *smartphone* do utilizador comum e que representa num mapa (Google Maps) os bares que se encontram nas imediações. A cada um destes bares é associada a informação dos eventos, promoções e as *playlists* que poderão ser acedidas clicando em cima do respectivo *Point of Interest* (POI).

O HappyHour.PT recorrerá ainda a outras funcionalidades desenvolvidas no âmbito de projectos de investigação onde o grupo proponente participa ou participou. Por exemplo poderá utilizar-se os mecanismos de acesso ao meio em tempo real, os algoritmos de fiabilidade e os modelos de gestão da mobilidade desenvolvidos no projecto europeu FP7 GINSENG (<http://www.ict-ginseng.eu>) pelos investigadores proponentes da presente proposta. Poderão ainda ser utilizados modelos de localização precisa *indoor* desenvolvidos no âmbito de parcerias com empresas nacionais e ao abrigo de vários projectos QREN.

Relação com outros projectos – *“Se este projeto se relacionar de alguma forma com outro projeto do universo PT ou outros projetos/produtos de outras empresas e instituições, deverão ser descritas de que forma ambos se aproximam e quais as mais valias resultantes para os envolvidos.”*

(2000 caracteres)

A proposta HappyHour.SAPO oferece à SAPO.PT uma plataforma inicial, simples mas robusta, que pode ser lançada rapidamente no mercado e antecipar-se a potenciais empresas concorrentes. Não existem ainda ferramentas e sistemas que permitam aos estabelecimentos comerciais a divulgação de promoções, em tempo real, para clientes próximos e de acordo os seus perfis. Espera-se assim não só responder a uma oportunidade temporal sentida, mas também implementar uma plataforma base que poderá potenciar novos serviços, novos modelos de negócio e novas funcionalidades:

- A plataforma HappyHour.SAPO pode ser integrada com vários serviços SAPO. Poderá, por exemplo, ser associada à SAPO MusicBox para a venda de músicas.

- No caso da utilização do sistema HappyHour.SAPO em espaços de entretenimento nocturno, possibilitar aos estabelecimentos seleccionar as músicas de acordo com o perfil e os gostos dos utilizadores que se encontrem próximos.

- A SAPO.PT pode recorrer às base de dados da HappyHour.SAPO para também publicitar produtos seus de acordo com a localização, os gostos e actividades dos utilizadores.

- A plataforma HappyHour.SAPO permite aumentar os serviços SAPO.PT com uma plataforma que acreditamos ser interessante.

- A ferramenta proposta poderá ser utilizada para lançar um conjunto de outros serviços SAPO orientado a diferentes públicos alvo.

Descrição - *“Deve ser redigida explicando em detalhe o que se pretende fazer no projeto, quais os objectivos a alcançar e as razões da sua importância. Pretende-se que descreva qual é o problema a ser investigado, quais são os desafios, quais as abordagens e as respectivas metodologias para alcançar os objectivos. Deve ainda mostrar que as metodologias propostas são adequadas para alcançar os objectivos. Neste ponto, deve abordar os aspectos tecnológicos mais relevantes do projeto e também os aspectos organizativos necessários para a sua execução.”*

(6000 caracteres)

As Redes Sociais têm ganho muita popularidade e possuem actualmente milhões de utilizadores que, diariamente, partilham as suas experiências e planeiam as suas actividades pessoais e profissionais. Paralelamente, o avanço verificado nas tecnologias baseadas em *smartphones* faz prever para curto prazo a integração de sistemas inovadores na melhoria das interfaces e das funcionalidades das redes sociais de nova geração. A combinação destas tecnologias permite criar pontes entre os mundos reais e os mundos virtuais, e potenciar novos modelos de negócio.

Um dos desafios presente no sistema proposto é avaliar muito bem os benefícios oferecidos a cada um dos participantes alvo. Embora do ponto de vista dos estabelecimentos comerciais e da SAPO.PT as mais valias da solução HappyHour.SAPO sejam evidentes, será importante encontrar uma solução interessante para o utilizador final. O sistema HappyHour.SAPO não poderá ser encarado pelo utilizador final como "mais um serviço" de publicidade e de SPAM. Este terá de ser efectivamente do seu interesse. Para isso é necessário a participação activa dos estabelecimentos comerciais com a divulgação de promoções aliciantes e focadas, e que o HappyHour.SAPO seja complementado com outros serviços interessantes para o utilizador final. Para o desenho de tais serviços, os promotores da presente candidatura esperam contar também com o apoio dos profissionais da SAPO.PT.

Assim, o sistema HappyHour.SAPO permite:

- Oferecer aos utilizadores finais um sistema de informação sobre os estabelecimentos comerciais, respectivas promoções e programas de eventos, de acordo com os seus gostos e perfis. Embora o modelo de aplicação inicial tenha sido desenvolvido para o estabelecimentos de diversão nocturna, este poderá ter aplicação mais diversa como a restauração ou espaços comerciais.
 - Oferecer aos utilizadores a possibilidade de saber em tempo real quais as promoções dos estabelecimento próximos.
 - Oferecer ao utilizador a possibilidade de saber em tempo real o grau de afluência de cada espaço de diversão. Paralelamente permitirá dar a conhecer ao utilizador o nível de agitação do local, por exemplo se o local é um bar muito agitado onde as pessoas se encontram a dançar ou, se pelo contrário, é um bar mais calmo onde é possível conversar. Esta informação possibilitará ao utilizador escolher o local de acordo com as suas vontades do momento.
 - Oferecer aos estabelecimentos um modelo de negócio inovador e inexistente, que permite divulgar as suas promoções e actividades, em tempo real, de uma forma simples e direccionada através de *push-notifications*, a micro-comunidades que se encontrem nas imediações do local.
 - Permitir aos estabelecimentos comerciais implementar políticas inovadoras de marketing com o convite à participação activa dos utilizadores nas actividades do local, e que pode ser incentivada com prémios e brindes.
 - Proporcionar à SAPO.PT uma base de dados muito completa com perfis, gostos e comportamento de utilizadores.
 - Oferecer à SAPO.PT um novo modelo de negócio que poderá utilizar para fidelizar mais clientes, sejam estes estabelecimentos comerciais ou clientes finais. Por outro lado permitirá à SAPO.PT envolver-se no negócios das micro-comunidades criadas.
-
- Possibilitar à SAPO.PT estabelecer parcerias comerciais com os estabelecimentos comerciais.
 - Facilitar à SAPO.PT combinar o HappyHour.SAPO com outros serviços do universo SAPO.PT. Por exemplo, poderia ser possível em versões futuras associar as músicas escutadas em cada bar e em cada noite com o serviço SAPO MusicBox. É comum cada utilizador associar momentos bem passados e com valor emotivo às músicas que escutou num determinado momento e num determinado local. Em muitas das situações, o utilizador desconhece mesmo o nome e o autor da música. A SAPO.PT poderá, assim, tirar proveito dos registos do HappyHour.SAPO para "oferecer" aos seus clientes a música que eles procuram.
 - Oferecer à SAPO.PT um sistema simples, robusto e flexível que poderá ser facilmente adaptado a novos serviços e a outros públicos alvo no futuro.

A presente proposta combina duas fases distintas:

- Numa primeira fase, que é aquela que compreende a presente candidatura, pretende-se implementar, de uma forma pragmática, um sistema base (*field trial*) que permita aos estabelecimentos comerciais implementar políticas de divulgação de produtos e promoções a clientes que se encontrem nas proximidades e de acordo com os seus perfis.

- Numa segunda fase, após o término do projecto associado a esta candidatura, enriquecer a plataforma HappyHour.SAPO com novas funcionalidades, serviços e inovações tecnológicas. Estes serviços e inovações poderão incluir, por exemplo a inclusão de novos dispositivos como redes de sensores ou sistemas RFID, mecanismos inovadores de privacidade, detecção de actividade de utilizadores, partilha de gostos musicais entre utilizadores, associação de certos gostos a certos estabelecimentos nocturnos, entre outros.

Plano do projecto – *“Indique uma lista calendarizada de etapas e milestones que permitam aferir se os trabalhos do projeto estão a progredir face aos objectivos apontados na candidatura. Para cada etapa inclua uma designação (máx. 60 caracteres), uma data (mensal), e a descrição (máx. 300 caracteres) do que pode ser demonstrado ou reportado nessa data.”*

(2000 caracteres)

O plano proposto para a implementação da primeira fase do sistema HappyHour.SAPO tem a duração de 12 meses e compreende as seguintes etapas:

1) Reuniões com a equipa SAPO.PT para a especificação do sistema HappyHour.SAPO.

Data de início: Setembro 2012

Duração estimada: 1 mês

2) Desenvolvimento de um novo protótipo funcional, que compreenda as aplicações Facebook/Android do utilizador, a aplicação do estabelecimento comercial e a aplicação de gestão do sistema. Todas estas aplicações deverão ser integradas com os serviços SAPO.PT.

Data de início: Novembro 2012

Duração estimada: 3 meses

3) Implementação de novos algoritmos para detecção de actividade humana, localização e privacidade.

Data de início: Janeiro 2013

Duração estimada: 4 meses

4) Desenvolvimento de aplicações para iPhone.

Data de início: Março 2013

Duração estimada: 3 mês

5) Testes, Integração e avaliação.

Data de início: Junho 2013

Duração estimada: 3 meses

Recursos necessários – *“Aqui deve especificar quais são os recursos necessários do labs.sapo para a execução do projeto que está a propor. Nos recursos deve indicar todo o tipo de suporte que tenha que ser fornecido pelo labs.sapo, nomeadamente: acesso a infraestrutura tecnológica, equipamentos, recursos humanos e dados. Os recursos humanos devem ser expressos em bolsas de investigação, sendo necessário especificar quais as milestones (descritas no plano do projeto) que serão executadas pelo bolseiro. Deverá ainda especificar a duração prevista para as bolsas.”*

(2000 caracteres)

Recursos Humanos: Este projecto requer dois alunos de doutoramento com dedicação exclusiva. Este alunos poderão ser contratados directamente pela SAPO com vencimentos indexados às bolsas FCT e suportando o custo da propina anual de doutoramento no valor de 3.500€.

Valor estimado: 2 alunos x 12 meses x bolsa + 2 x propina anual = 2 x 12 x 980 + 2x 3500 = 23.520€ + 7.000€ = 30.520€

Paralelamente HappyHour.SAPO conta ainda com a participação de vários investigadores na Universidade de Coimbra, nomeadamente:

Jorge Sá Silva - Professor Auxiliar e investigador sénior, que coordenará o projecto.

Fernando Boavida - Professor Catedrático e investigador sénior.

David Nunes - aluno de doutoramento a desenvolver trabalhos relacionados com o tema do projecto.

Paulo Rupino - Professor Auxiliar e investigador sénior.

Luiz Correia - Professor visitante e investigador sénior.

Claudio Geyer - Professor visitante e investigador sénior.

Valderi Leinhart - aluno de doutoramento.

Vasco Pereira - Professor Auxiliar convidado e investigador sénior

Equipamento: 1 PC, 2 portáteis, 1 telemóvel Android, licenças MAC (programação iPhone) e 1 telemóvel iPhone

Valor estimado: 3.000€

Valor total para os 12 meses: 33.520 €

Resultados esperados – *“Após a conclusão do projeto quais os resultados esperados e de que forma estes constituem uma mais valia para as partes envolvidas. Se aplicável, poderá identificar potenciais modelos de negócio a explorar futuramente entre SAPO e a sua instituição.”*

(2000 caracteres)

Ao fim de um ano de trabalho o sistema será desenhado, implementado e testado com as funcionalidades propostas.

Anexo F “Mensagens de Comunicação do sistema Avós e Netos”

Avós e Netos

Mensagens de Comunicação

draposo@student.dei.uc.pt

Versão 0.1

DOCUMENTO APROVADO POR	Nome, email@student.dei.uc.pt
DATA	[dia] de [Mês] de [AAAA] (data da aprovação)

HISTÓRICO DE REVISÕES

VERSÃO	DATA	AUTOR	DESCRIÇÃO
0.1	2715/08/2012	Duarte Raposo	<ul style="list-style-type: none"> • Descrição das mensagens

F.1 Descrição Geral

Neste documento é descrito a troca de mensagens do sistema *SocialSense* Avós e Netos, entre a aplicação, servidor e proxys, através de diagramas de sequência e exemplos de mensagens no formato *JSON* e *XML*.

Esta especificação surge após a definição dos requisitos da aplicação, onde foram descritas as diferentes funcionalidades que o sistema deve fornecer. Estas funcionalidades usam *webservices* na troca de informação, através da biblioteca *jersey* em *java* (servidor) e *javascript/dojo* (aplicação web). A tabela seguinte mostra os diferentes *webservices* criados no servidor do sistema:

WebService	Método
URL/rest/activity	Set Activity
URL/rest/childpersonaldata	Get Child Data
URL/rest/facebookmaps	Get Avatar Data
URL/rest/postFallDetection	Set Fall Detection
URL/rest/getAlerts	Get Alerts
URL/rest/getAssociatedChilds	Get Associated Childs
URL/rest/getChildInfobyMoteID	Get Parent Skype ID
URL/rest/getChildPhoto	Get Child Photo
URL/rest/getLocation	Get Location
URL/rest/GetSkypeAccount	Get Child Skype ID
URL/rest/saveLocation	Save Location

Tabela 6 - Lista de *Webservices* Sistema Avós e Netos

Na definição dos *webservices* foi também necessário especificar algumas características da comunicação, como por exemplo a segurança. Assim, definiu-se que a comunicação entre a aplicação *Facebook* e os *webservices restfull* devia utilizar a autenticação *oauth* através de *access token* (como exemplificado na Figura 55). Adicionalmente, estabeleceu-se que a comunicação entre proxys e servidor não deveria conter nenhum tipo de segurança neste projecto, ficando assim como trabalho futuro.

Como podemos observar na Figura 56, o processo de comunicação foi efectuado através de uma mensagem que contém dois campos distintos. O primeiro campo “autenticação” é constituído pelo *acesstoken*, que garante não repúdio e autenticação. O segundo campo “dados” é o responsável pelo encapsulamento dos dados de envio e resposta, como exemplificado na Figura 56.

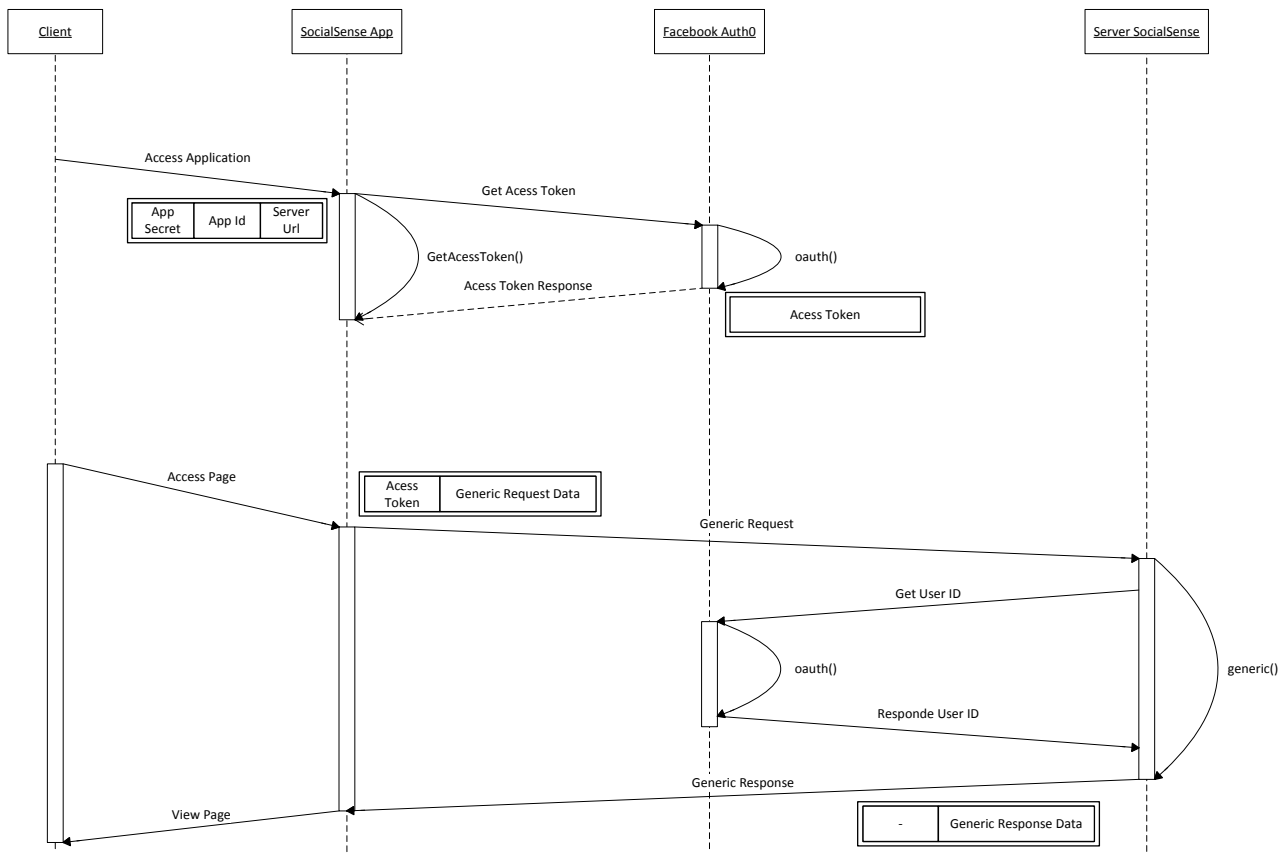


Figura 55 - Autenticação Facebook OAuth

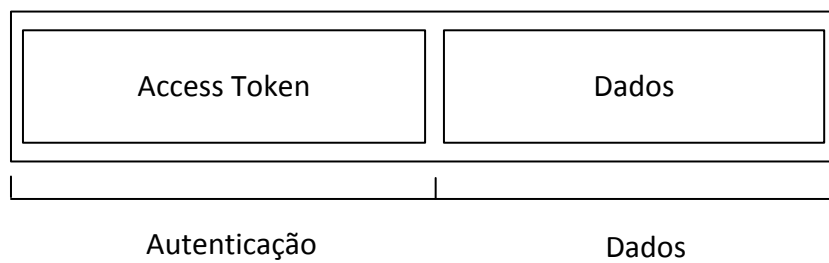


Figura 56 - Mensagens Padrão Sistemas Avós e Netos

F.2 Especificação das mensagens

Este capítulo especifica individualmente cada mensagem usada na comunicação entre o servidor/aplicação *Facebook* e servidor/*proxys*. Os diagramas que se seguem não representam a autenticação *OAuth*, dado já ter sido descrita em cima. Apresentam assim, unicamente as mensagens dos pedidos e a sua informação.

F.2.1 Set Activity

Como foi referido no documento de requisitos, mais especificamente, no requisito “Vista Actividade” o(a) avô(ó) poderá visualizar a actividade do neto(a) no mapa. Esta actividade é obtida através de sensores que o(a) neto(a) possui, sendo a sua informação transmitida através de módulos específicos, apelidados de *proxys*. A mensagem “Set Activity” é assim enviada através de um objecto *Sensor Data* transformado em *XML* entre o *proxy* e o servidor. Como podemos verificar na imagem em baixo esta mensagem, transporta o id do sensor, o tipo de dados, o valor, o tempo de aquisição e o endereço *IP*. Ao chegar a mensagem ao servidor, este insere os dados na base de dados e retorna uma mensagem de sucesso ou insucesso.

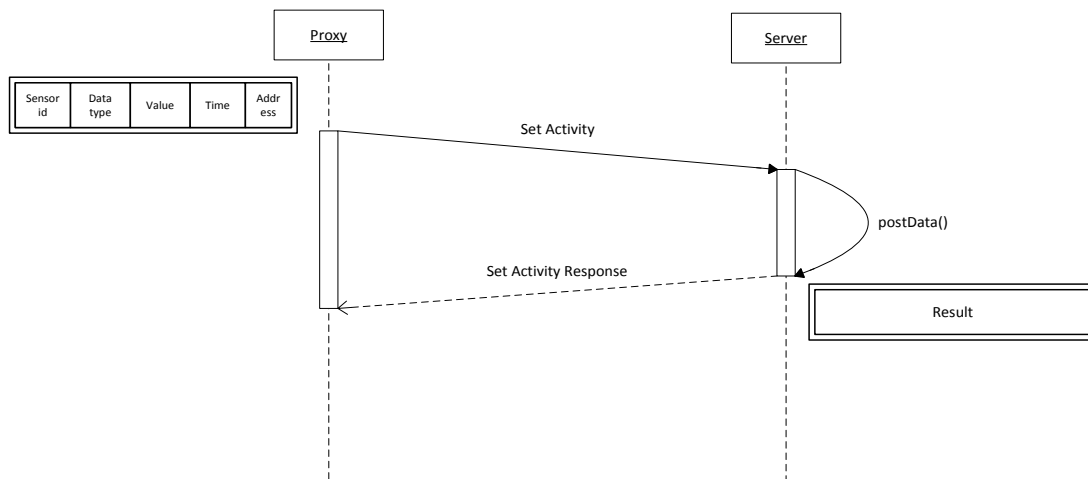


Figura 57 - DS Set Activity

Exemplo da mensagem XML enviada:

Mensagem Set Activity:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <sensordata>
    <sensorid>1</ sensorid >
    <datatype>temperature</ datatype >
    <value>21</ value >
    <time>1:00:12</ time >
    <hostaddress>null</ hostaddress >
  </ sensordata >
```

Mensagem Set Activity Response:

Data saved

F.2.2 Get Child Data

Por diversas vezes na aplicação *Facebook* é necessário apresentar informações pessoais sobre a criança, como por exemplo no requisito “Vista Actividade”.

Assim, foi criado no servidor um *webservice* que envie esta informação para a aplicação. O envio começa quando o servidor recebe uma mensagem do tipo “Get Child Data”, contendo os campos apresentados na **Erro! A origem da referência não foi encontrada..** Ao receber esta mensagem, o ervidor retira a informação da base de dados, referente ao id da criança fornecida na mensagem, enviando uma mensagem de resposta contendo o objecto *Child Personal Data* em formato JSON.

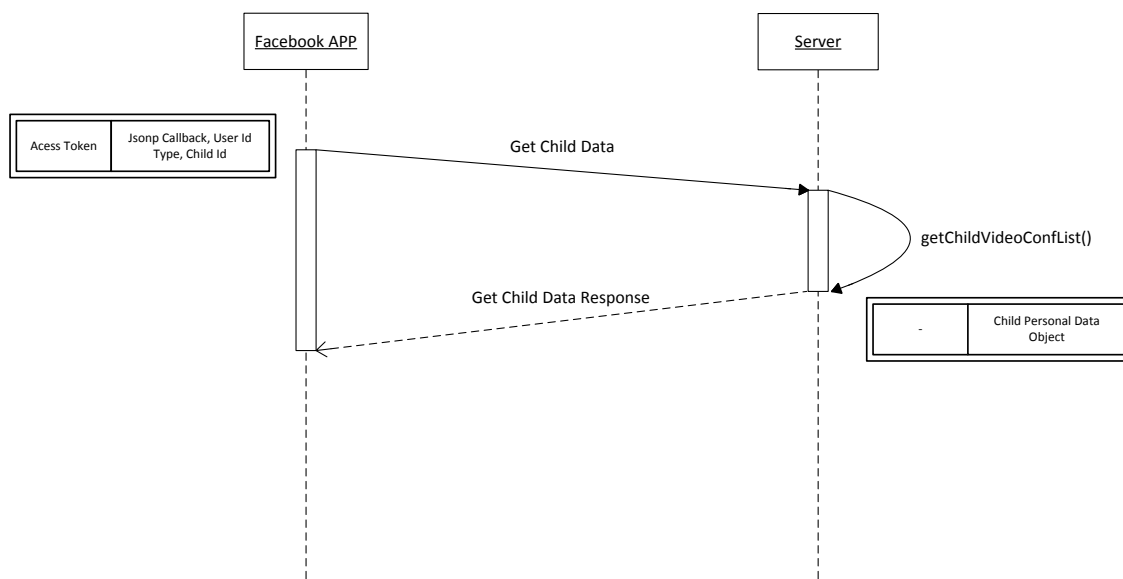


Figura 58 - DS Get Child Data

Exemplo da mensagem JSON enviada:

```
Mensagem Get Child Data:
GET
http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/childPersonalData?userid=AAABpEyAt7b0BADY2LrI77YhmKsSlqtqz05C0WaHsvkZA1LRROrNanN5PyJ7ZAtKTgJ23KCxAkf0KTKgzlqxmUfnrWLTTeIEQ6iALFKLQZDZD&useridtype=facebook&childID=10&jsonp_callback=dojo.io.script.jsonp_dojoloScript2._jsonpCallback
Mensagem Child Data Response:
dojo.io.script.jsonp_dojoloScript2._jsonpCallback({"birthday":"2006-10-05T00:00:00+01:00","firstName":"Magda","lastName":"Tudo","nationality":"Portuguesa"})
```


F.2.3 Get Avatar Data

Como especificado no requisito “Vista Actividade”, o(a) avô(ó) poderá visualizar diferentes tipos de dados fisiológicos sobre a criança (neto(a)). Assim, quando a mensagem “Get Avatar Data” chega ao servidor através do respectivo *webservice*, este irá procurar na base de dados os últimos dados fisiológicos da criança. Quando os dados forem encontrados, o servidor irá envia-los através de um objecto *WebAvatarInfo* no formato *JSON* para a aplicação *facebook*.

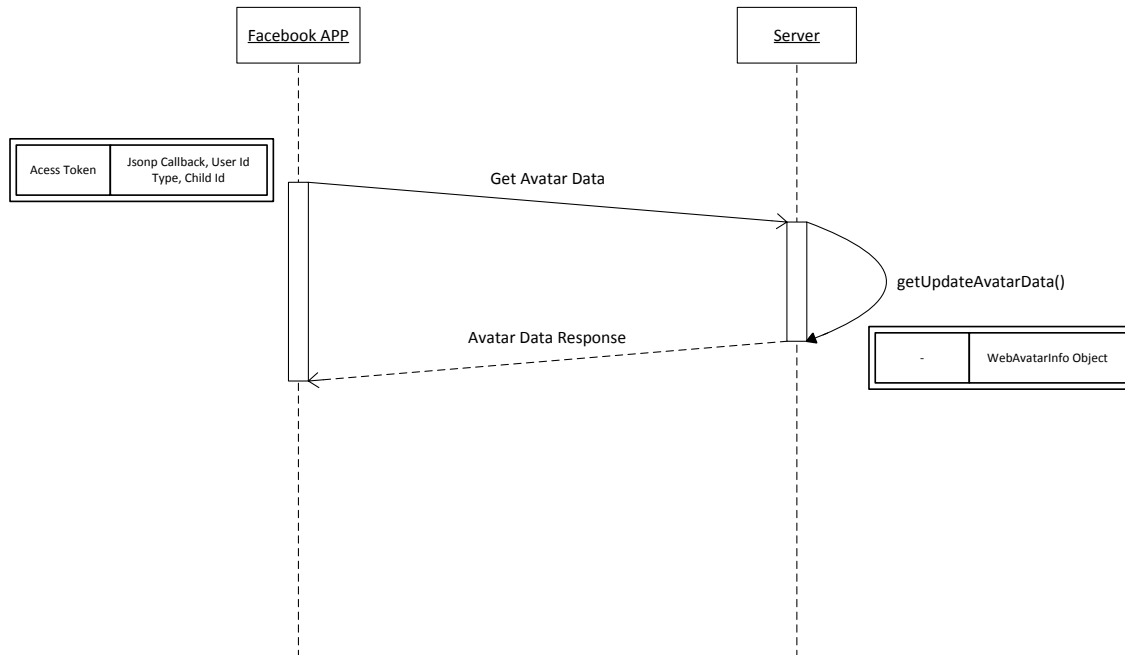


Figura 59 - DS Get Avatar Data

Exemplo da mensagem JSON enviada:

```
Mensagem Get Avatar Data:  
GET  
http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/facebookmaps?jsonp_callback=k=jQuery16107517602556849017_1345132059772&userid=AAABpEyAt7b0BADY2LrI77YhmKsSlqtqz05C0WaHsvkZA1LRROrNanN5PyJ7ZAtKTgJ23KCxAkf0KTKgzlqxmUfnrWLTTeIEQ6iALFKLQZDZD&useridtype=facebook&childID=10&_=1345132063193  
Mensagem Avatar Data Response:  
jQuery16107517602556849017_1345132059772({"activity":"-999","mobileMed_Oximeter":["-999.0","-999.0"],"mobileMed_Sphyg":["-999.0","-999.0"],"mobileMed_Spiro":"-999.0","temperature":"24.510000000000005"})
```

F.2.4 Set All Fall Detection

A mensagem “Set Fall Detection” representa um evento de queda num determinado sensor. Esta mensagem foi construída devido ao requisito “Vista Alarmes”. Este requisito especifica que o(a) avô(ó) poderá visualizar alguns tipos de alarmes, incluindo quedas. Assim, o *proxy* responsável pela detecção de quedas (acelerómetro) envia uma mensagem com o mote id para o respectivo *webservice*. Ao chegar esta mensagem ao servidor, este irá introduzir os dados na base de dados e enviar a confirmação para o *proxy*, como representado em baixo.

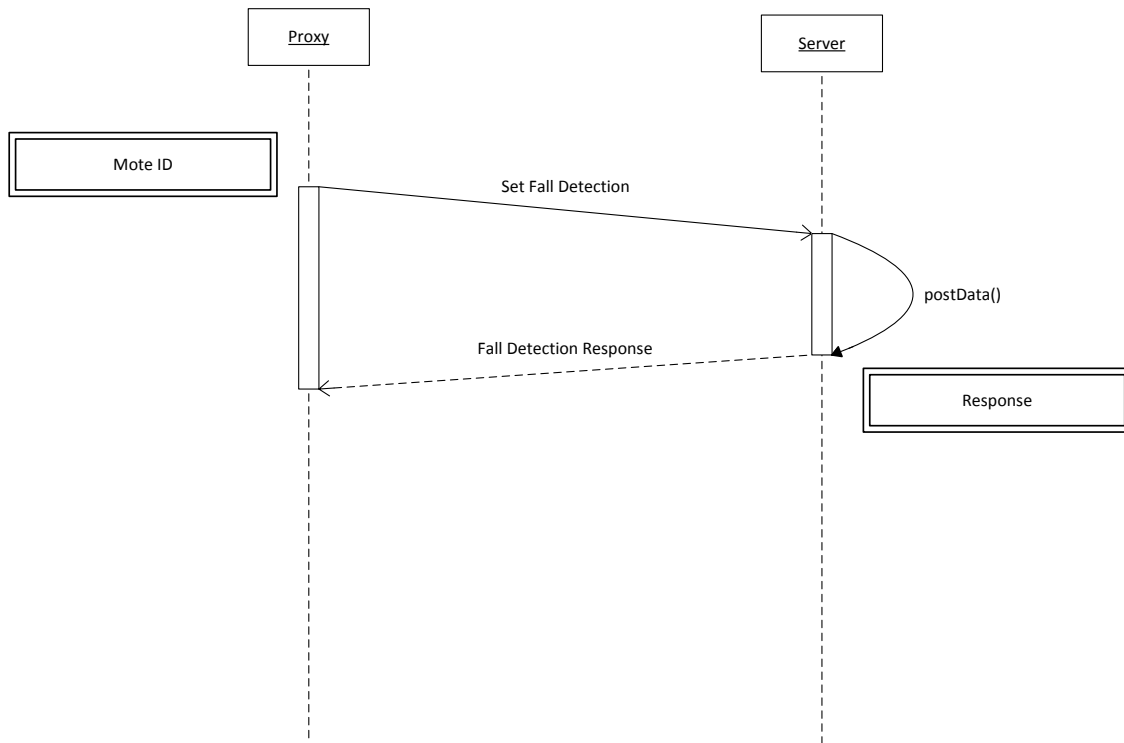


Figura 60 - DS Set Fall Detection

Exemplo da mensagem enviada:

```
Mensagem Set Fall Detection Status:  
3  
Mensagem Fall Detection Response:  
Data saved
```

F.2.5 Get Alerts

Como referido anteriormente, o(a) avô(ó) poderá visualizar diferentes alarmes especificados através do requisito “Vista Alarmes”, como por exemplo o alarme de queda e de pedido de videochamadas. Logo, quando o utilizador acede à página de alarmes, a aplicação envia uma mensagem “Get Alerts” para o *webservice* “URL/rest/getAlerts” no servidor. Quando esta mensagem chega ao servidor, este verificará na base de dados quais os alertas existentes, referentes ao *Child ID* passado na mensagem. Encontrados os alertas, o servidor irá enviar um objecto *Alert Data Object* no formato *JSON* para a aplicação.

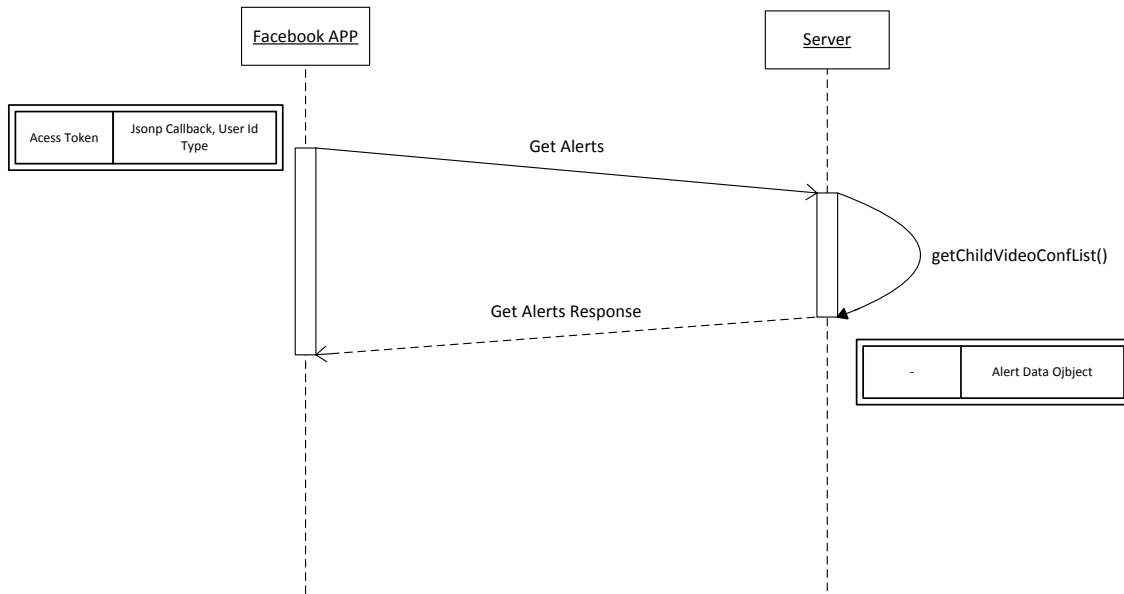


Figura 61 - DS Get Alerts

Exemplo da mensagem JSON enviada:

Mensagem Get Alerts:

GET

```
http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/getAlerts?userid=AAABpEyAt7b0BADY2LrI77YhmKsSlqtqz05C0WaHsvkZA1LRROrNanN5PyJ7ZAtKTgJ23K CxAKf0KTKgzlqxmUfnrWLTiEIEQ6iALFKLQZDZD&useridtype=facebook&jsonp_callback=dojo.io.script.jsonp_dojoloScript1._jsonpCallback
```

Mensagem Get Alerts Response:

```
dojo.io.script.jsonp_dojoloScript1._jsonpCallback({"alertData":[{"IDChild":"9","time":"1340120964467","typeevent":"sensor button"}]})
```

F.2.6 Get Associated Childs

No documento de requisitos foi especificado que a aplicação deveria de conter um menu que permitisse o utilizador aceder às diferentes funcionalidades de um modo simples. Dado o menu ter que apresentar algumas vezes os(as) netos(as) associados a cada avô(ó), foi criada a mensagem “Get Associated Childs”. Quando esta mensagem chega ao servidor, este irá procurar na base de dado quais os(as) netos(as) associados ao avô(ó) e retornar um objecto *Associated Child Data* em formato *JSON* para a aplicação contendo o nome dos netos.

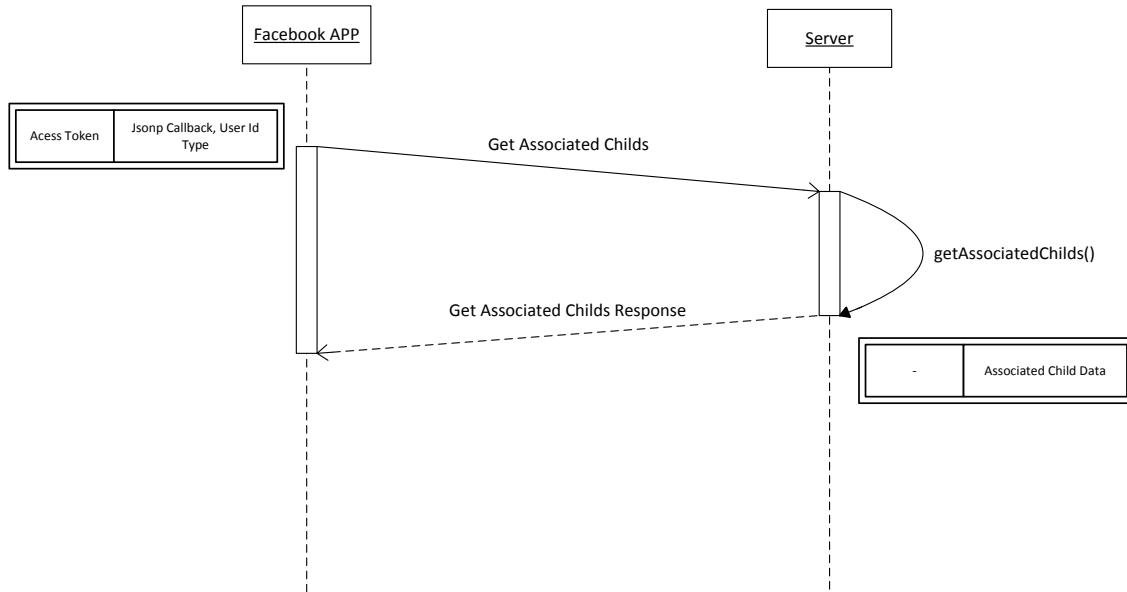


Figura 62 - DS Get Associated Childs

Exemplo da mensagem JSON enviada:

Mensagem Get Associated Childs:

GET
http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/getAssociatedChilds?userid=AAABpEyAt7b0BADY2LrI77YhmKsSlqtqz05C0WaHsvkZA1LRROrNanN5PyJ7ZAtKTgJ23KCxAkf0KTKgzlqxmUfnrWLTleIEQ6iALFKLQZDZD&useridtype=facebook&jsonp_callback=dojo.io.script.jsonp_dojoIoScript1._jsonpCallback

Mensagem Get Associated Childs Response:

```
dojo.io.script.jsonp_dojoIoScript1._jsonpCallback({"associatedChildData":{"firstName":"Magda","idChild":"10","lastName":"Tudo"},{"firstName":"Joana","idChild":"9","lastName":"Sempre"}}})
```

F.2.7 Get Parent Skype ID

O sistema *SocialSense* Avós e Netos permite através de redes de sensores sem fios iniciar videochamadas através da proximidade do(a) neto(a) ao computador da instituição. Para que isto seja possível foi criada a mensagem “Get Parent Skype ID” que tem como utilidade identificar o *Skype ID* do(a) avô(ô) através do sensor que o(a) neto(a) transporta. Assim, quando o(a) neto(a) se aproxima do computador da instituição, passados alguns segundos o *proxy* envia uma mensagem encapsulada num objecto *Skype Account* em formato *XML* com o id do nó sensor. Quando a mensagem chega ao servidor, este procura na base de dados o *Skype ID* retornando-o para o *proxy*.

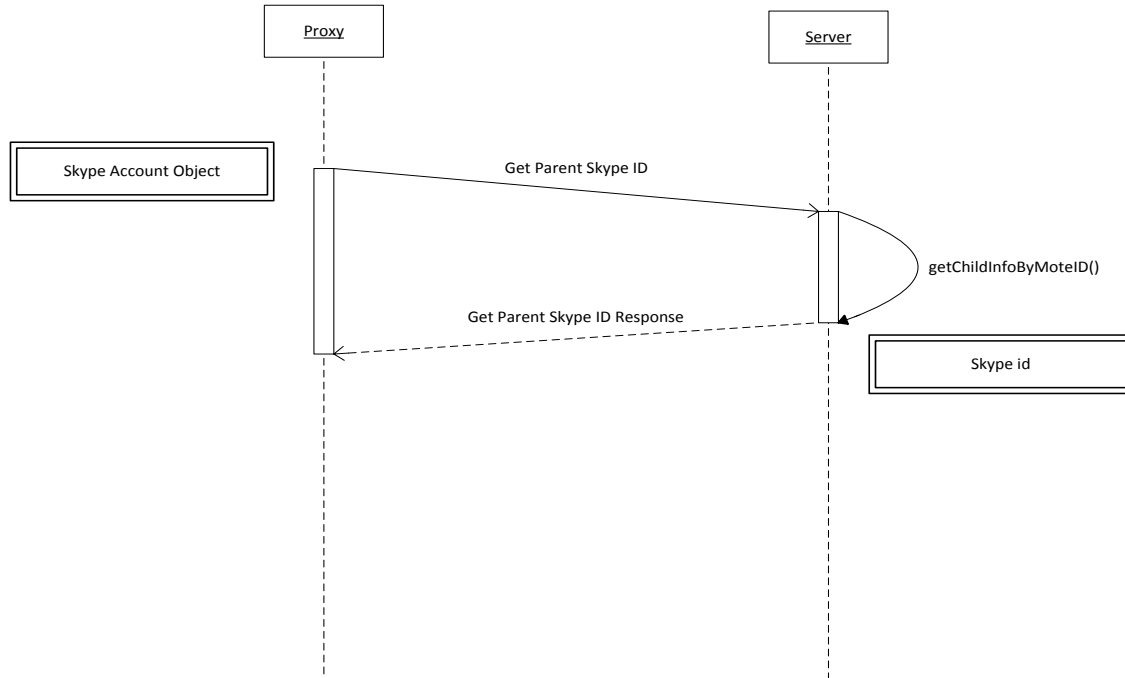
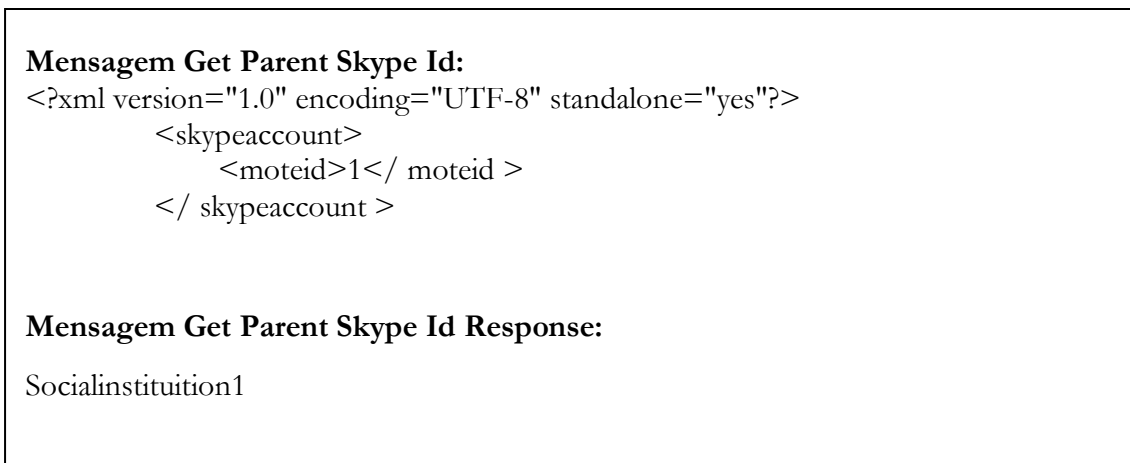


Figura 63 - DS Get Parent Skype ID

Exemplo da mensagem XML enviada:



F.2.8 Get Child Photo

Durante a aplicação, muitas das páginas necessitam de apresentar a fotografia do(a) neto(a) virtual. Assim, foi criada a mensagem “Get Child Photo” e respectivo *webservice* que retorna para a aplicação *Facebook* a imagem de um dado *Child ID*. O servidor ao receber esta mensagem verificará na sua base de dados qual o(a) neto(a) e respectiva foto, retornando a imagem em *array de bytes* para a aplicação.

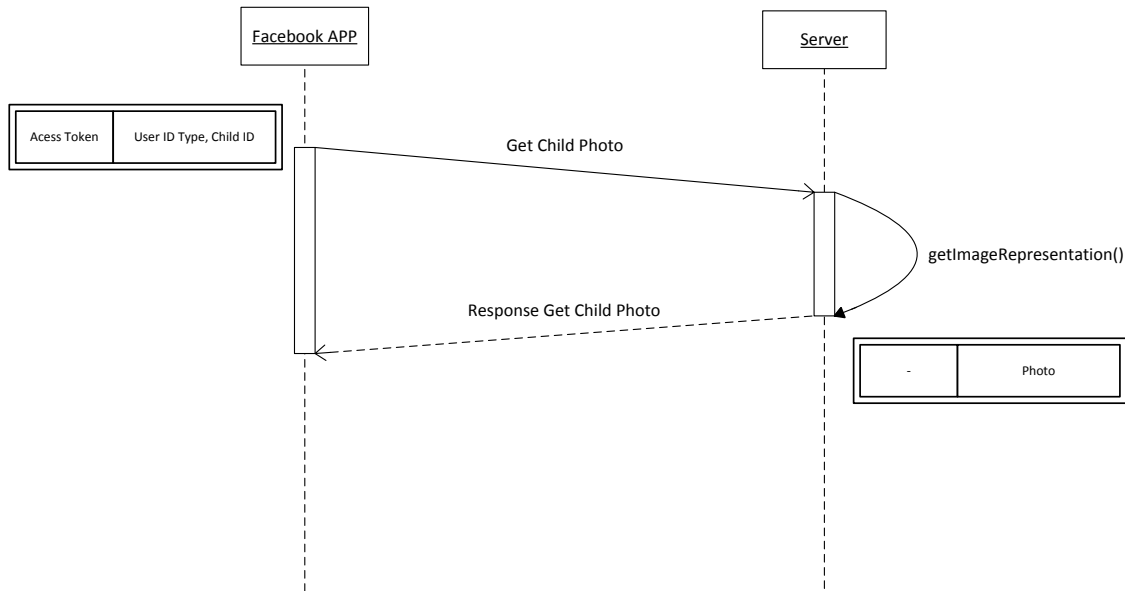


Figura 64 - DS Get Child Photo

Exemplo da mensagem JSON enviada:



F.2.9 Get Location

Um dos requisitos principais define que o(a) avô(ó) deve conseguir visualizar a localização do(a) neto(a) através de um mapa, que contém um pequeno avatar. Para isso, o sistema utiliza um mecanismo de localização através de redes de sensores sem fios. Logo, sempre que o utilizador pretende visualizar a actividade do(a) neto(a) na aplicação, é enviada uma mensagem “Get Location” para o *webservice* correspondente no servidor. Ao chegar esta mensagem ao servidor, será verificado na base de dados a última posição do neto e enviado para a aplicação através do objecto *Child Web Location* em formato *JSON*. Existirá um proxy de localização que é responsável por enviar os dados dos sensores para o servidor, num tempo de aquisição definido.

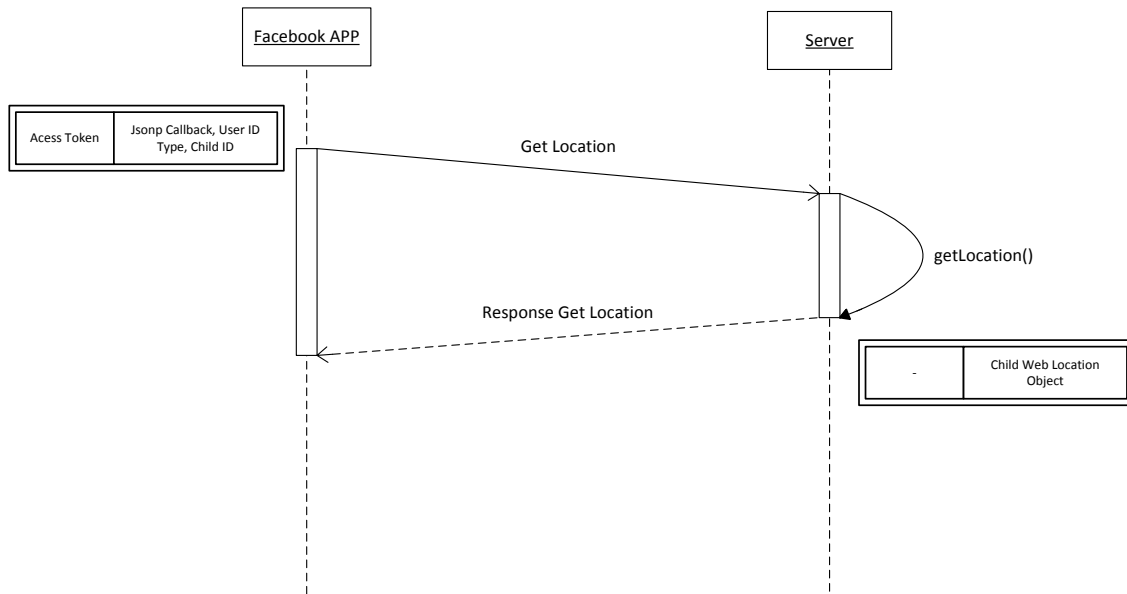


Figura 65 - DS Get Location

Exemplo da mensagem JSON enviada:

```
Mensagem Get Location:  
  
GET  
http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/getLocation?jsonp_callback  
=jQuery16106850857178974854_1345132757186&userid=AAABpEyAt7b0BADY2LrI  
77YhmKsSlqtqz05C0WaHsvkZA1LRROrNanN5PyJ7ZAtKTgJ23KCxAkF0KTKgzlqx  
mUfnrWLTiEIEQ6iALFKLQZDZD&useridtype=facebook&childID=10&_=1345132  
760448  
  
Mensagem Get Location Response:  
jQuery16106850857178974854_1345132757186(39.222;43.90)
```

F.2.10 Get Child Skype Id

Como foi explicado anteriormente na mensagem “Get Parent Skype ID” é possível que o(a) neto(a) inicie uma videochamada através de um mecanismo de proximidade que detecta a sua presença perto do computador. Isto é, este mecanismo só é accionado quando o(a) neto(a) pretende contactar o(a) avô(ó). Por outro lado, a aplicação *Facebook* permite também que o(a) avô(ó) possa contactar o(a) neto(a). Para isso, a aplicação envia uma mensagem “Get Child Skype Id” para o *webservice* “URL/rest/GetSkypeAccount”. Quando a mensagem chega ao servidor este procura na base de dados e envia para a aplicação através de um objecto *SkypeUser* em formato *JSON*.

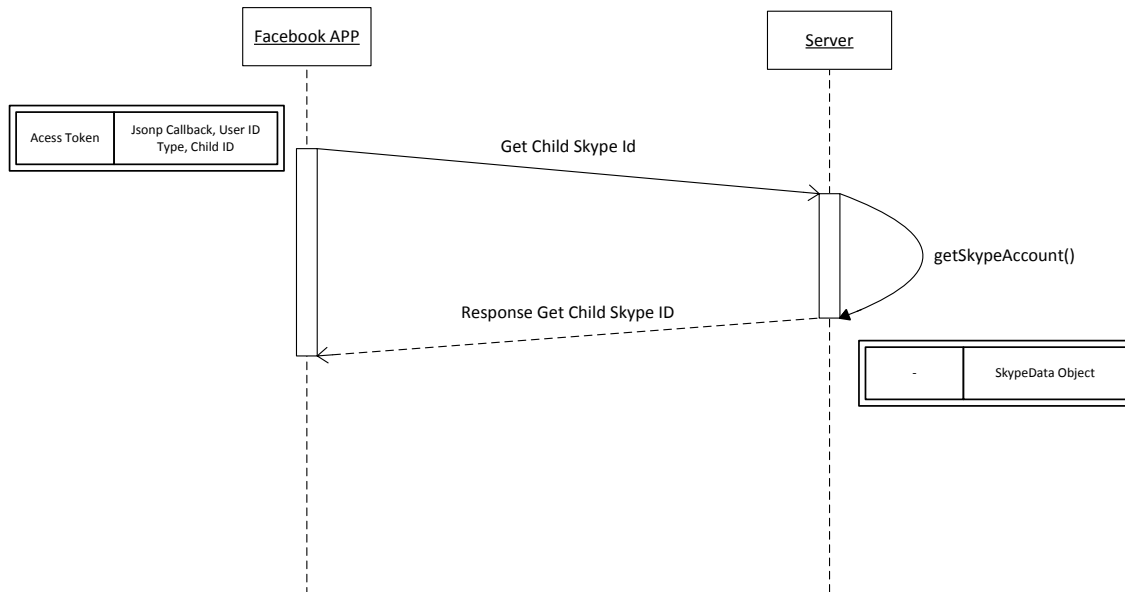


Figura 66 - DS Get Child Skype ID

Exemplo da mensagem JSON enviada:

Mensagem Get Child Skype ID:

GET

```
http://socialsense.dei.uc.pt:8080/MiddlewareService/rest/GetSkypeAccount?userid=AAABpEyAt7b0BADY2LrI77YhmKsSlqtqz05C0WaHsvkZA1LRROrNanN5PyJ7ZAtKTgJ23KCxAkf0KTKgzlqxmUfnrWLTTeIEQ6iALFKLQZDZD&childID=10&useridtype=facebook&jsonp_callback=dojo.io.script.jsonp_dojoIoScript3._jsonpCallback
```

Mensagem Get Child Id Response:

```
dojo.io.script.jsonp_dojoIoScript3._jsonpCallback({"skypeUser":"socialsenseinstitution2"})
```


F.2.11 Save Location

Na mensagem “Get Location”, foi descrito que a aplicação *Facebook* possui um requisito que permite a visualização da posição do(a) neto(a). Essa posição é inserida através de um *proxy*, que comunica com os diferentes sensores, e envia para o servidor a localização através de mensagens “Save Location”. Estas mensagens são enviadas através de *XML* representando um objecto *Tlocation*, que contem diferentes tipos de informação como o exemplo que é apresentado em baixo. Ao chegar esta mensagem ao servidor, este irá realizar alguns cálculos e guardar a posição na base de dados, enviando finalmente uma resposta de confirmação para o *proxy*.

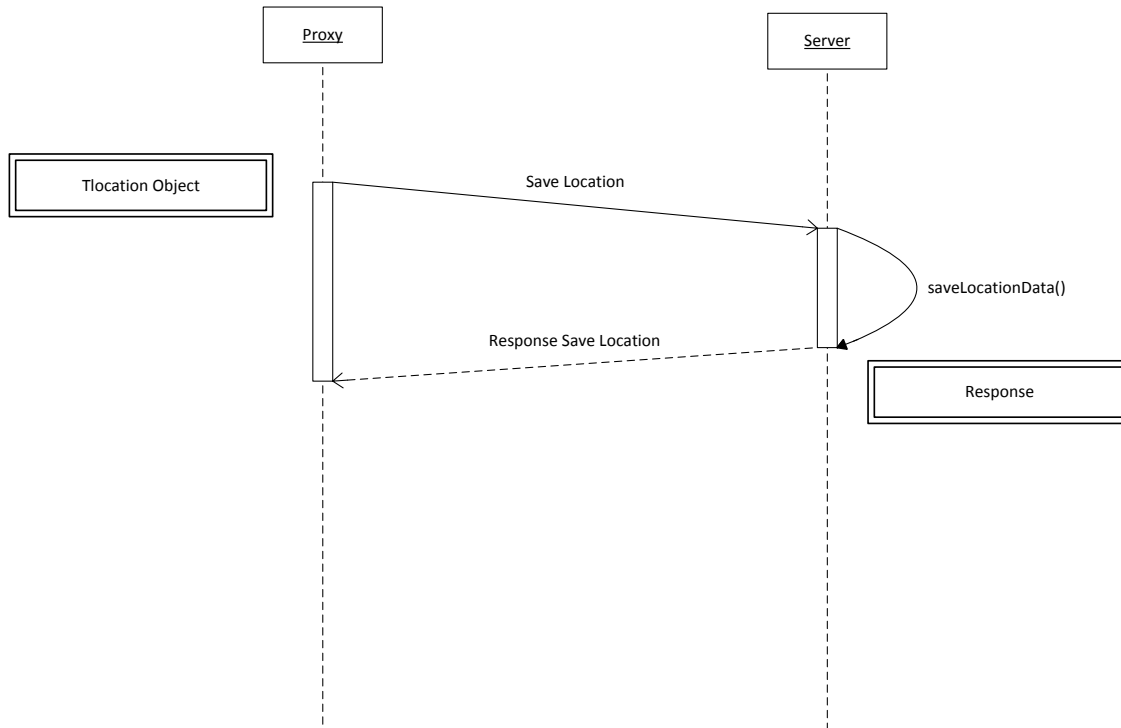


Figura 67 - DS Save Location

Exemplo da mensagem XML enviada:

```
Mensagem Save Location:  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
  <tlocation>  
    <id>1</ id >  
    <readerId>1</readerId >  
    <tagID>5</ tagID >  
    <rssValue>300.356</ rssiValue >  
    <tsamp>27291630</ tsamp >  
  </ tlocation >  
  
Mensagem Save Location Response:  
true
```


Anexo G “Mensagens de Comunicação do sistema Youth5G”

Youth5G

Mensagens de Comunicação

`draposo@student.dei.uc.pt`

Versão 0.1

Descrição da troca de mensagens da aplicação Youth5G

DOCUMENTO APROVADO POR	Nome, email@student.dei.uc.pt
DATA	[dia] de [Mês] de [AAAA] (data da aprovação)

HISTÓRICO DE REVISÕES

VERSÃO	DATA	AUTOR	DESCRIÇÃO
0.1	10/08/2012	Duarte Raposo	<ul style="list-style-type: none"> • Descrição das mensagens

G.1 Descrição Geral

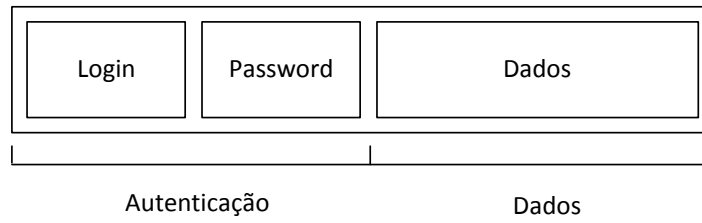
O objectivo deste documento é descrever a troca de mensagens existentes entre a aplicação *Youth5G* e o servidor, através de diagramas de sequência e exemplos de mensagens no formato *JSON*.

Depois de serem acordados os requisitos da aplicação foi essencial especificar quais seriam os *webservices* necessários para gerir a troca de informação e garantir o acordado no documento de requisitos. Assim, foi criada uma tabela que contém os *webservices* que seriam necessários, bem como, as respectivas mensagens de envio e de chegada.

WebService	Método
URL/Rest/newUser	SetNewUser
URL/Rest/newPassword	SetNewPassword
URL/Rest/acesLogin	GetAccessLogin
URL/Rest/getListPoi	GetListPOI
URL/REST/getDetailPoi	GetDetailPOI
URL/REST/getListEvent	GetListEvent
URL/REST/getDetailEvent	GetDetailEvent
URL/REST/sendFeedback	SetFeedback
URL/REST/getFeedbackStatus	GetFeedbackStatus
URL/REST/getListRanking	GetListRanking
URL/REST/getPreditionPoi	GetPreditionPOI
URL/REST/setInquerito	SetInquerito
URL/REST/getInqueritoStatus	getInqueritoStatus
URL/REST/getprevasiveservice s	GetPrevasiveServices
URL/REST/setsensordata	SetDataSensor

Tabela 7 - Lista de Webservices do sistema Youth5G

Seguidamente para garantir a integridade, não repúdio e autenticidade das mensagens foi decidido que a troca de informação deveria seguir um padrão definido por uma mensagem dividida em dois campos. Estes campos são chamados por: campo de autenticação e campo de dados.



Como podemos observar a mensagem está repartida em vários campos e cada um possui uma finalidade. O primeiro campo “autenticação” é o responsável por enviar os dados de *login* e *password* aquando um pedido por parte da aplicação *Android* para o servidor. O segundo campo “Dados” é o responsável pelo encapsulamento dos dados de resposta vindos do servidor para a aplicação *Android*.

G.2 Especificação das Mensagens

Este capítulo especifica individualmente cada mensagem usada na comunicação entre o servidor e a aplicação *Android*.

G.2.1 Get Access Login

A mensagem “Get Access Login” tem como principal finalidade enviar os dados de autenticação definidos pelo requisito efectuar login. Assim, a aplicação *Android* enviará uma mensagem com o respectivo *username* e *password* para o servidor. Chegada a mensagem ao *webservice* disponível em “/rest/accessLogin”, será efectuada a validação através da base de dados e enviada a resposta. A resposta irá no campo de dados da mensagem sem autenticação e poderá ser “OK”, “INVALID” ou “ERROR”.

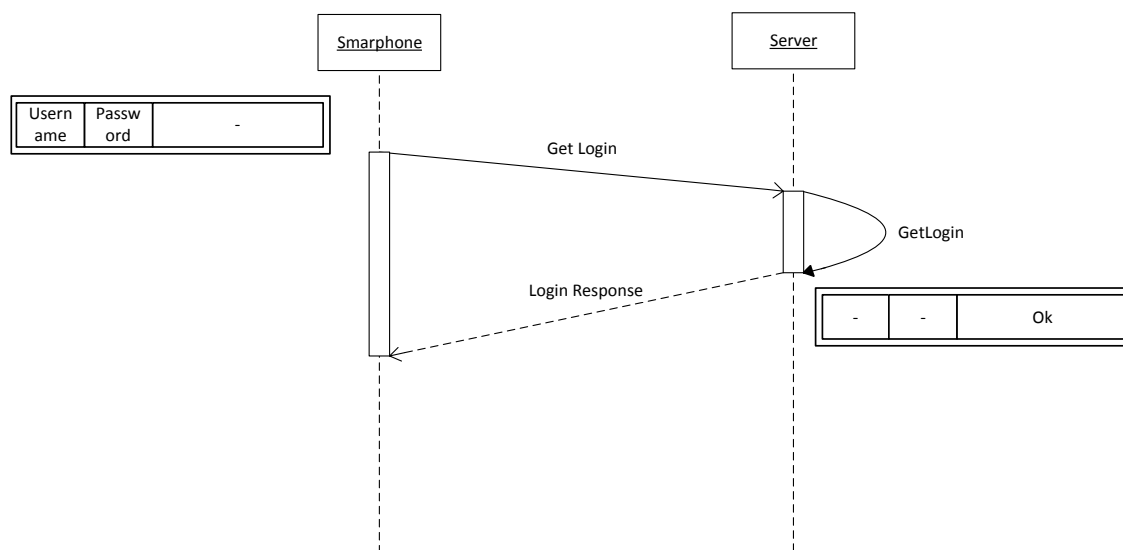


Figura 68 - DS Get Access Login

Exemplo da mensagem JSON enviada:

```
Mensagem Get Login:  
{ "password": "12345", "username": "duarte@ola.com" }  
Mensagem Login Response:  
ok
```

G.2.2 Get Detail Event

A mensagem “Get Detail Event” tem como objectivo recolher as características de um determinado evento. Quando o utilizador pretender aceder às características de um evento na aplicação *Android*, irá ser enviado para o servidor uma mensagem com o respectivo *username*, *password* e id evento. Quando esta mensagem chega ao *webservice* “/rest/getDetailEvent” será retornado para a aplicação os campos Data, Descrição, Id Evento, Id Poi e a Imagem do evento.

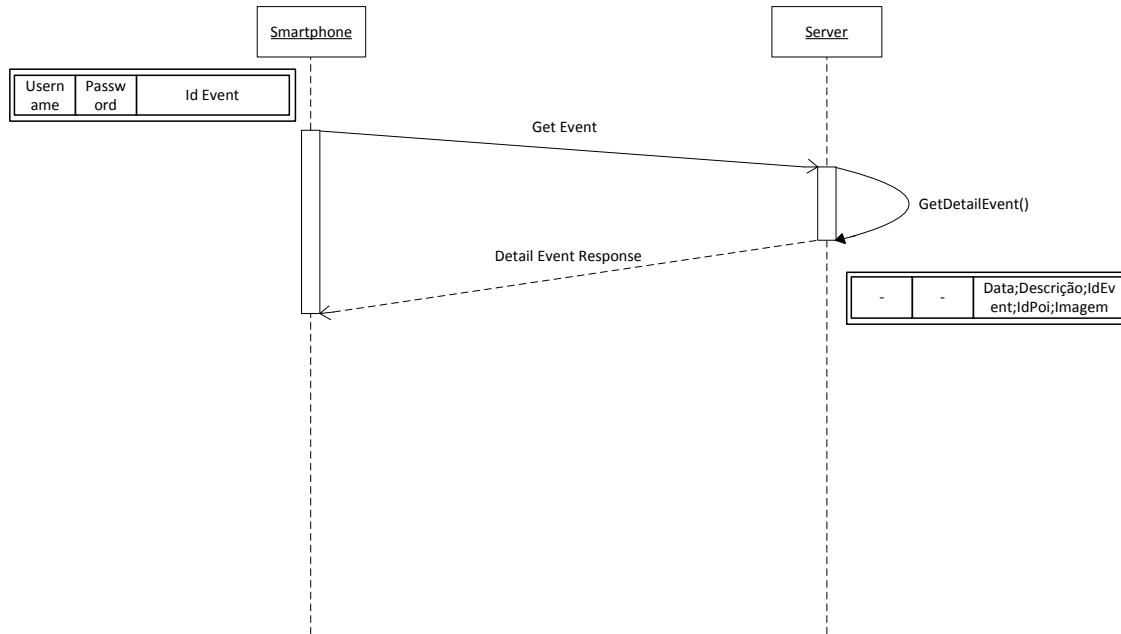


Figura 69 - DS Get Detail Poi

Exemplo da mensagem JSON enviada:

```

Mensagem Get Event:
{"id_event":56,"password":"12345","username":"duarte@ola.com"}
Mensagem Detail Event Response:
{"data":"2012-08-09 00:37:55.684","descricao":"Sempre a
bombar","idEvent":"56","idPoi":"56","imagem":"bytes"}
  
```

G.2.3 Get Detail POI

O requisito visualizar características *poi* permite que o utilizador visualize todos os dados de um local/ ambiente nocturno como foto, localização, etc. Logo, quando o utilizador pretende aceder a estas características, a aplicação *android* envia para o servidor uma mensagem com os dados de autenticação e o endereço do poi (id *foursquare*). Consequentemente, ao chegar a mensagem ao servidor irá ser pedida à API do *foursquare* uma imagem (se não estiver na base de dados) do local e de seguida será enviado para a aplicação *android*, as coordenadas GPS, a descrição, o id do poi e a imagem. Caso não exista nenhuma imagem no *foursquare* serão apenas enviados os dados sobre o poi.

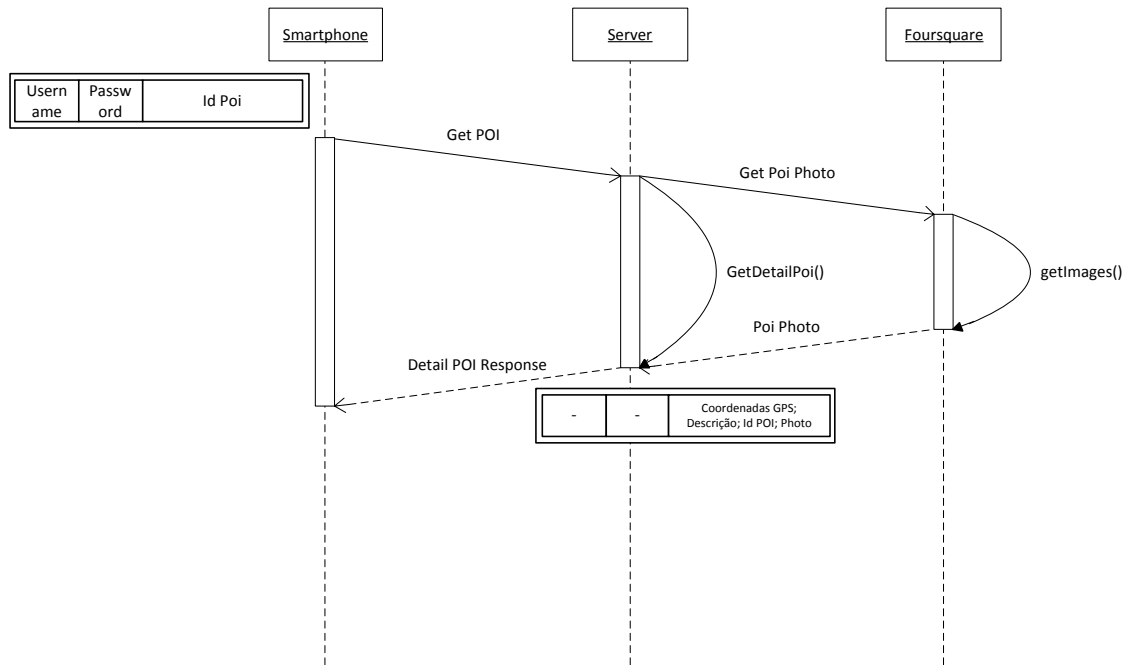


Figura 70 - DS Get Poi

Exemplo da mensagem JSON enviada:

```

Mensagem Get POI:
{"password":"12345","username":"duarte@ola.com","id_poi":"4cfd54d2fabc2d439c33ded2"}

Mensagem Detail POI Response:
{"coordenadaslat":"40.20801627905773","coordenadaslong":"-8.420312404632568","descricao":"Mais uma 5ª Feira :-)\nMesmo em coimbra é a referência\nSem sombra de duvida a melhor discoteca de Coimbra! Recomendam-se as Revival Nights e as Gold Nights!\n","idPoi":"4cfd54d2fabc2d439c33ded2","imagem":"bytes"}
  
```

G.2.4 Get Feedback Status

Esta mensagem está associada ao requisito atribuir *feedback*, que permite ao utilizador enviar o grau de popularidade de um determinado poi. Assim, esta mensagem possibilita à aplicação saber se foi enviado um feedback naquela hora. Para isso, é enviada uma mensagem para o servidor com os dados de autenticação e o id do poi. Ao chegar esta mensagem ao servidor, é procurado na base de dados se o utilizador já efectuou algum envio de previsão, retornando para a aplicação *android* o valor da escolha e o *status* da resposta. Ou seja, se o utilizador não tiver enviado nenhum feedback a mensagem retornará “-1”, caso contrário, retornará “1”, “2”, “3”, “4” ou ”5”.

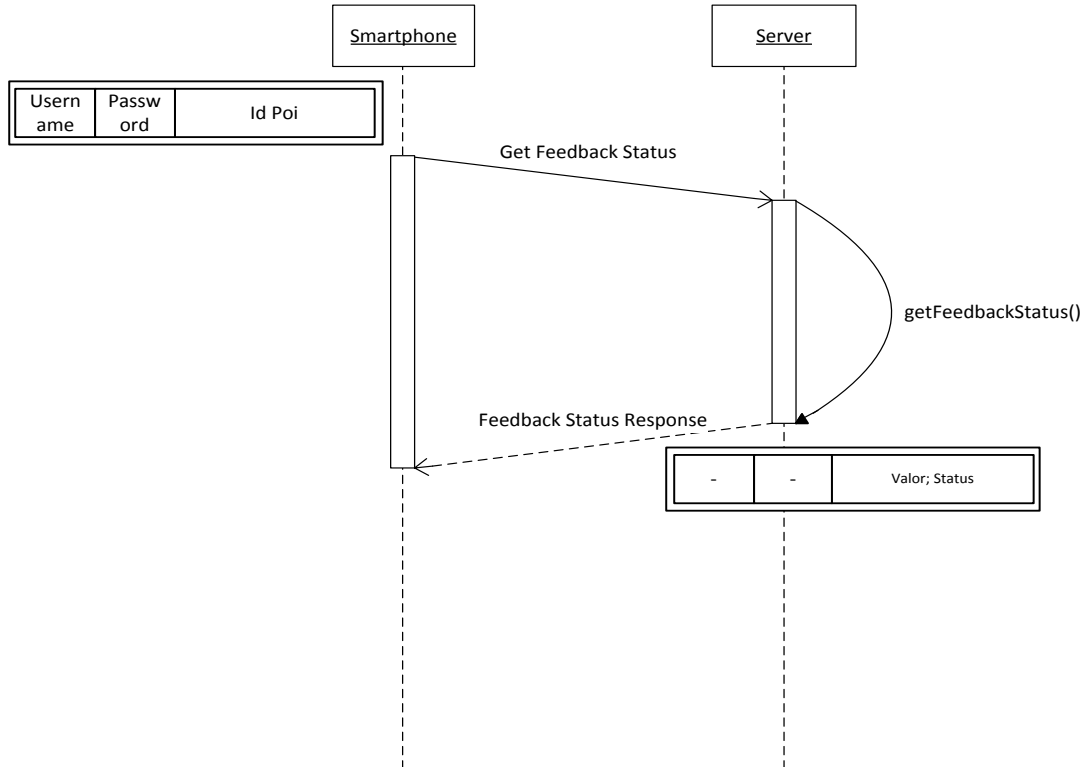


Figura 71 - DS Get Feedback Status

Exemplo da mensagem JSON enviada:

Mensagem Get Feedback Status:

```
{"password":"12345","username":"duarte@ola.com","id_poi":"4cfd54d2fabc2d439c33d  
ed2"}
```

Mensagem Feedback Status Response:

```
{"checkboxValue":"-1","status":"ok"}
```

G.2.5 Get List Event

A mensagem “get list event” é a mensagem responsável por disponibilizar uma lista de eventos na aplicação *Youth5G*. Para o utilizador poder visualizar os eventos que irão ocorrer num determinado local/poi, a aplicação enviará para o servidor os dados de autenticação e o Id do poi. Quando esta mensagem chega ao servidor, este retornará os dados de data, id do evento, id do poi, nome, que estão na base de dados.

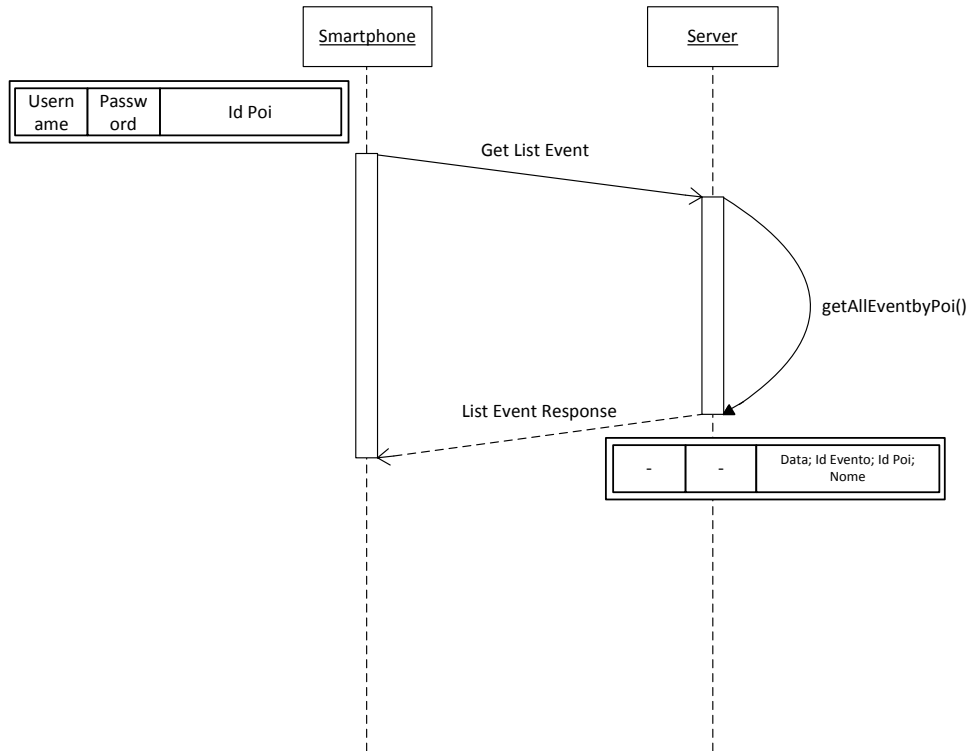


Figura 72 - DS Get List Event

Exemplo da mensagem JSON enviada:

```

Mensagem Get List Event:
{"password":"12345","username":"duarte@ola.com","id_poi":"4cfd54d2fabc2d439c33ded2"}

Mensagem List Event Response:
{"responseListEvent":[{"data":"2012-08-09T00:04:54.199+01:00","idEvento":"51","idPoi":"4cfd54d2fabc2d439c33ded2","nome":"Noite de Quinta"}, {"data":"2012-08-09T00:10:32.631+01:00","idEvento":"52","idPoi":"4cfd54d2fabc2d439c33ded2","nome":"Noite de Sexta"}]}
  
```

G.2.6 Get List Poi

Para o utilizador poder visualizar no mapa da aplicação *Youth5G* os diferentes *Pois* é necessário o envio da mensagem “get list poi”. Esta mensagem requer ao servidor uma lista com os diferentes *Pois* num raio de 5km a partir da posição cento do mapa. Para isso, é enviada uma mensagem para com os dados de autenticação do utilizador e as coordenadas do centro do mapa (latitude, longitude). Ao chegar esta mensagem o servidor irá fazer um pedido à *API* do *foursquare*, para que lhe seja enviado os últimos *Pois* disponíveis, actualizando assim a base de dados. Na mensagem de envio é retornado o nome, latitude, longitude, id do *poi* e popularidade.

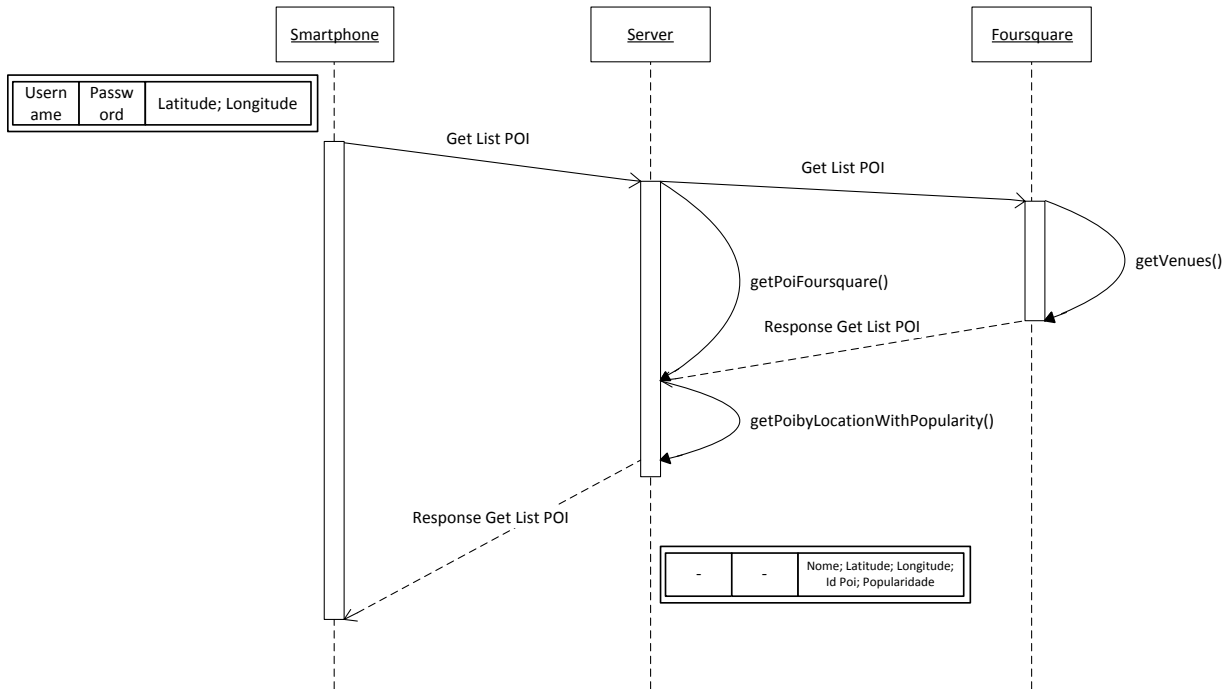


Figura 73 - DS Get List Poi

Exemplo da mensagem JSON enviada:

Mensagem Get List POI:

```
{ "password": "12345", "longitude": "-8.42", "username": "duarte@ola.com", "latitude": "40.21" }
```

Mensagem List POI Response:

```
{ "responseListPoi": [ { "nome": "Ilda Bar", "coordenadaslat": "39.25983086874046", "idPoi": "4fad8fafa4b0efef5e5ca80a", "popularidade": "-1", "coordenadaslong": "-8.58576040283596" }, { "nome": "Fino's", "coordenadaslat": "39.20794374730718", "idPoi": "4f611333e4b011d8502f3a76", "popularidade": "-1", "coordenadaslong": "-8.58576040283596" } ] }
```

G.2.7 Get Prediction

O requisito “visualizar previsões” permite ao utilizador da aplicação verificar a afluência de um local nos dias de semana que se seguem. Assim, a mensagem “get prediction” é enviada da aplicação para o servidor contendo os dados de autenticação e id do poi. Calculada a previsão do lado do servidor, será enviada uma mensagem contendo uma lista com o id do poi, data e popularidade.

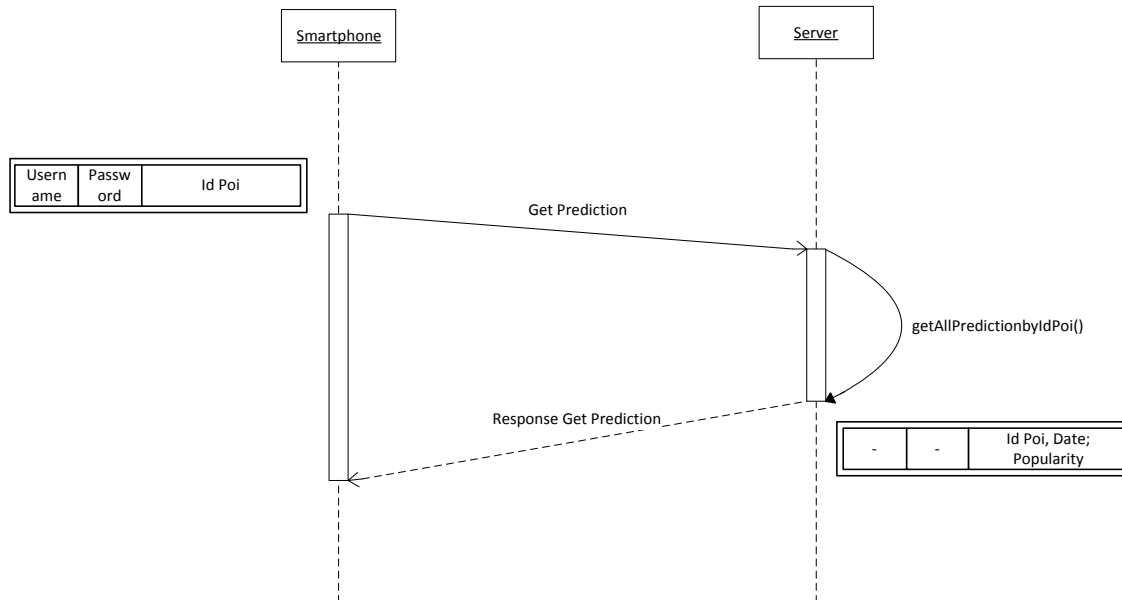


Figura 74 - DS Get Prediction

Exemplo da mensagem JSON enviada:

Mensagem Get Prevasive Services:

```
{"idPoi":"4cfd54d2fabc2d439c33ded2","password":"123456","username":"duarte@ola.com"}
```

Mensagem Response Get Prevasive Services:

```
{"responseListPrediction":[{"idPoi":"4cfd54d2fabc2d439c33ded2","data":"2012-08-10T00:00:00+01:00","popularity":"5.0"}, {"idPoi":"4cfd54d2fabc2d439c33ded2","data":"2012-08-11T00:00:00+01:00","popularity":"3.0"}, {"idPoi":"4cfd54d2fabc2d439c33ded2","data":"2012-08-12T00:00:00+01:00","popularity":"1.0"}, {"idPoi":"4cfd54d2fabc2d439c33ded2","data":"2012-08-13T00:00:00+01:00","popularity":"2.0"}]}
```

G.2.8 Get Prevasive Services

A mensagem “get prevasive services” é a mensagem responsável por identificar os serviços que um utilizador tem dentro de um ambiente. Assim, sempre que um utilizador entra num ambiente identificado no mapa, a aplicação ubíqua envia uma mensagem para o servidor com os dados de autenticação e id do poi, com a finalidade de identificar os serviços que o utilizador pode utilizar naquele ambiente. Ao chegar esta mensagem ao servidor este irá identificar os serviços e retorna-los para a aplicação ubíqua.

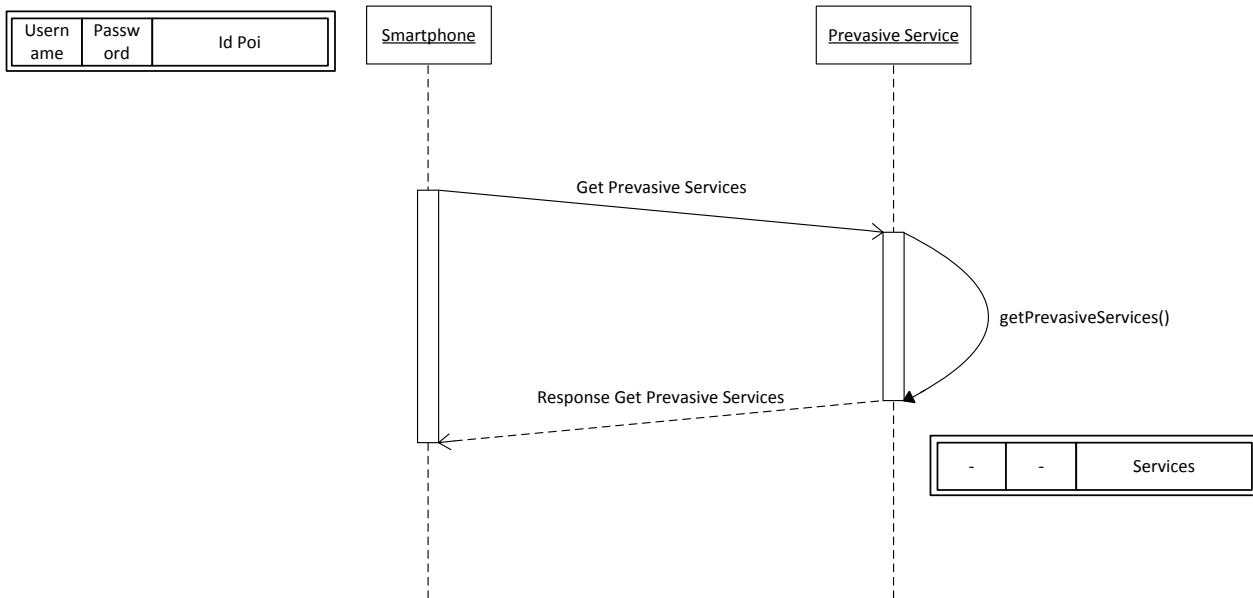


Figura 75 - DS Get Prevasive Services

Exemplo da mensagem JSON enviada:

Mensagem Get Prevasive Services:

```
{"password":"12345","username":"duarte@ola.com","id_poi":"4c4ae1089c8d2d7fcb5f286a"}
```

Mensagem Response Get Prevasive Services:

```
{"responsePrevasiveServices":[{"servicename":"lightness"}, {"servicename":"accelerometer"}]}
```


G.2.9 Get Ranking

Sendo a aplicação ser colaborativa, contém um mecanismo de incentivo em forma de *ranking* para incentivar o utilizador a participar na aplicação. A mensagem “get ranking” é a mensagem responsável pelo envio do ranking do utilizador para a aplicação. Quando o utilizador pretende saber qual o seu *ranking* depois de ter submetido alguns *feedbacks*, a aplicação *Youth5G* envia uma mensagem contendo os dados de autenticação. Ao chegar esta mensagem ao servidor, este irá procurar na base de dados o respectivo *ranking*, enviando de seguida, uma mensagem em forma de lista contendo todos os utilizadores e a sua pontuação.

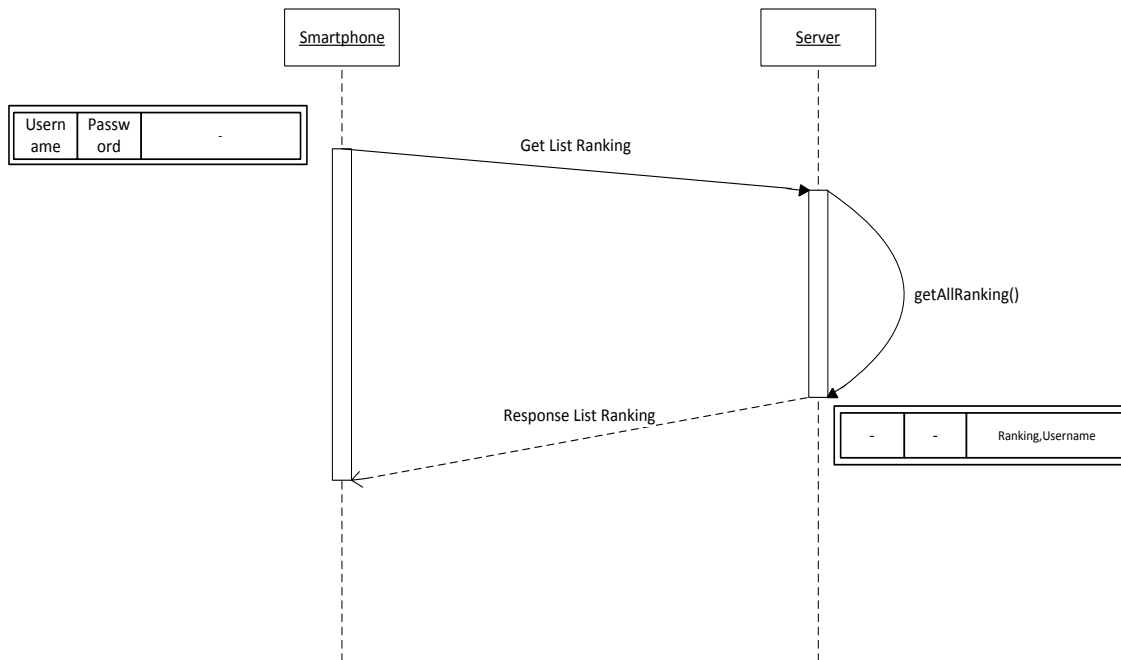


Figura 76 - DS Get Ranking

Exemplo da mensagem JSON enviada:

Mensagem Get Ranking:

```
{"password":"12345","username":"duarte@ola.com"}
```

Mensagem List POI Response:

```
{"responseListRanking":[{"ranking":"0","username":"duarte@ola.com"}, {"ranking":"0","username":"luis@gmail.com"}, {"ranking":"0","username":"duarte@ola.com"}]}
```

G.2.10 Get Survey Status

Como a opinião do utilizador foi um dos requisitos definidos para o desenvolvimento da aplicação, a mensagem “get survey status” permite identificar se um utilizador já efectuou o inquérito. Para isso, a aplicação envia para o servidor os dados de autenticação, aguardando uma mensagem de retorno que indica com “Ok” se o utilizador já realizou o inquérito.

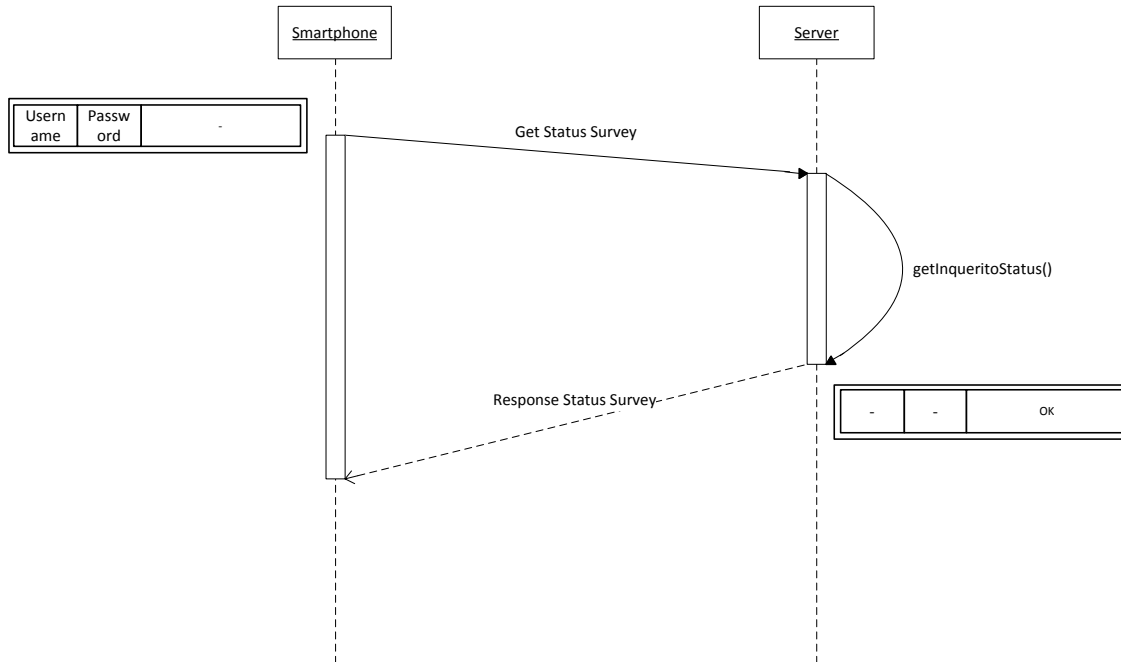


Figura 77 - DS Get Survey Status

Exemplo da mensagem JSON enviada:

Mensagem Get Ranking:

```
{"password":"12345","username":"duarte@ola.com"}
```

Mensagem List POI Response:

```
ok
```

G.2.11 New Password

Esta mensagem tem como objectivo implementar o requisito de “alterar password”. Como a aplicação permite o registo de novos utilizadores, se o utilizador o desejar, também poderá alterar a *password* criada inicialmente. Para que isto seja possível, a aplicação enviará uma mensagem ao servidor com os dados de autenticação e a nova *password*. Quando esta mensagem chega ao servidor, este alterará a *password* do utilizador na base de dados, enviando uma resposta de confirmação para a aplicação *Youth5G*.

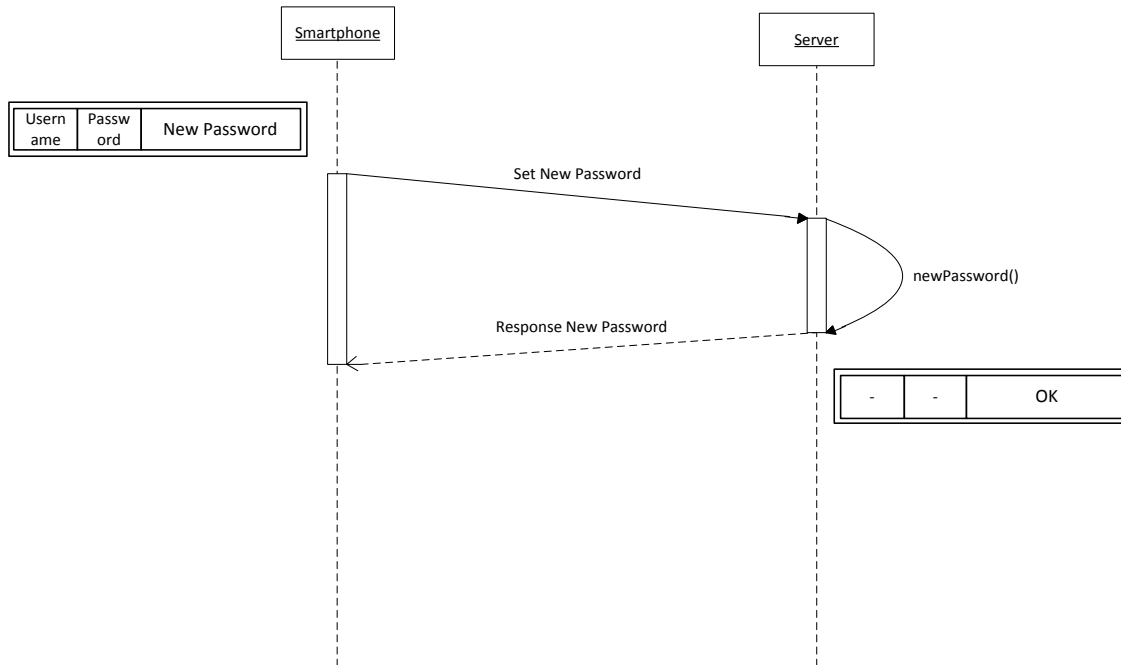


Figura 78 - DS New Password

Exemplo da mensagem JSON enviada:

<p>Mensagem Set New Password:</p> <pre>{"password":"12345","newpassword":"123456","username":"duarte@ola.com"}</pre> <p>Mensagem Response New Password:</p> <pre>ok</pre>

G.2.12 New User

Como foi dito anteriormente, a aplicação *android* permite o registo de novos utilizadores. Assim, quando um novo utilizador pretende efectuar um registo na aplicação, é enviado para o servidor uma mensagem com o seu *username* e respectiva *password*. Depois do servidor inserir este novo registo na base de dados com sucesso, será enviada uma mensagem com “OK” para a aplicação *android*.

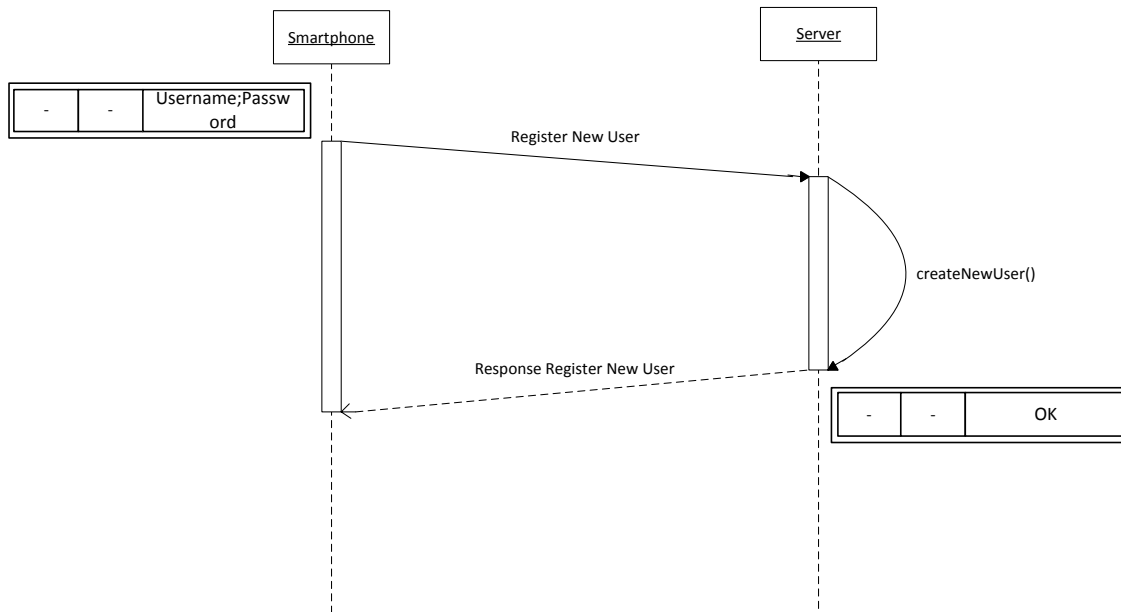


Figura 79 - DS New User

Exemplo da mensagem JSON enviada:

<p>Mensagem Register New User:</p> <pre>{"password":"12345","username":"duarte2@ola.com"}</pre> <p>Mensagem Response Register New User:</p> <pre>ok</pre>

G.2.13 Set Data Sensor

A mensagem “set data sensor” tem como finalidade o envio de dados dos sensores disponíveis nos *smartphones*. Como foi referido anteriormente, quando o utilizador entra num ambiente pervasivo, os dados que ele poderá partilhar são controlados através de um mecanismo de privacidade relativa ao ambiente. Ao receber os serviços do ambiente, a aplicação *android* irá activar os sensores e, conseqüentemente, será enviada uma mensagem que transporta os dados de autenticação e o dos sensores. Quando esta mensagem chega ao servidor este irá guardar os dados na base de dados e enviar uma mensagem de confirmação contendo “OK”.

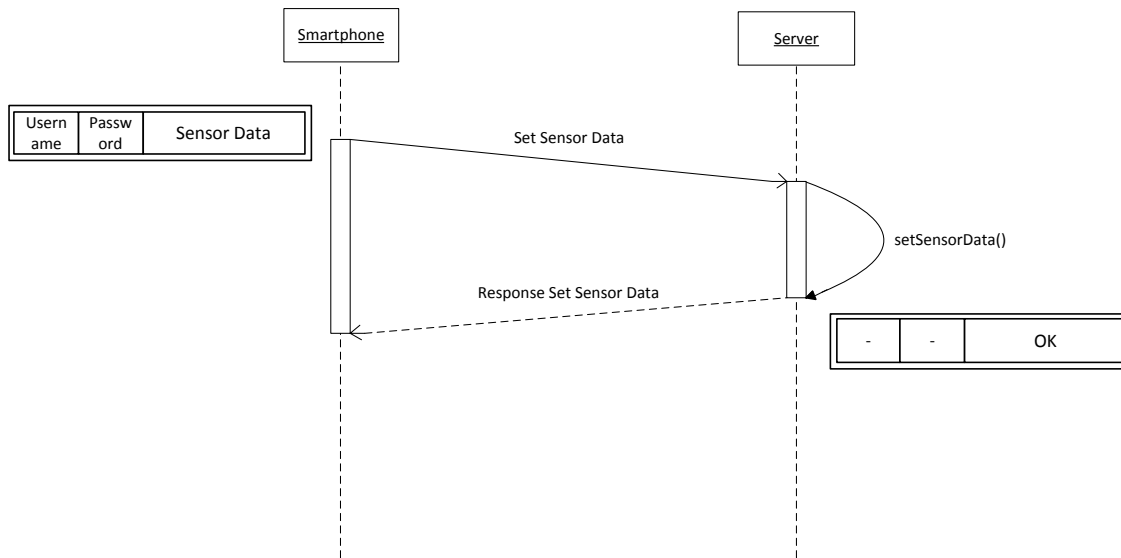


Figura 80 - DS Set Data Sensor

Exemplo da mensagem JSON enviada:

Mensagem Set Data Sensor:

```
{"data":[{"timestamp":12345667789,"value":1}, {"timestamp":12345667789,"value":2}, {"timestamp":12345667789,"value":3}, {"timestamp":12345667789,"value":4}, {"timestamp":12345667789,"value":5}, {"timestamp":12345667789,"value":6}, {"timestamp":12345667789,"value":7}, {"timestamp":12345667789,"value":8}, {"timestamp":12345667789,"value":9}], "username": "duarte@ola.com", "datatype": "accelerometer", "id_poi": "4c4ae1089c8d2d7fcb5f286a", "password": "12345"}
```

Mensagem Response Set Data Sensor:

```
ok
```

G.2.14 Set Feedback

Como foi especificado no documento de requisitos esta aplicação ubíqua é baseada em dados colaborativos para realizar o cálculo da afluência de pessoas a locais com eventos nocturnos. Assim, esta mensagem “set feedback” tem como objectivo o envio do *feedback* do utilizador relativo ao local onde se encontra. Para que isto seja possível, a aplicação envia uma mensagem contendo os dados de autenticação, id do poi e classificação, aguardando pela mensagem de resposta do servidor. Se a mensagem de resposta contiver “OK” é porque foi o primeiro *feedback* naquela hora, por outro lado, se a mensagem contiver “update” é porque o utilizador já continha um *feedback* enviado.

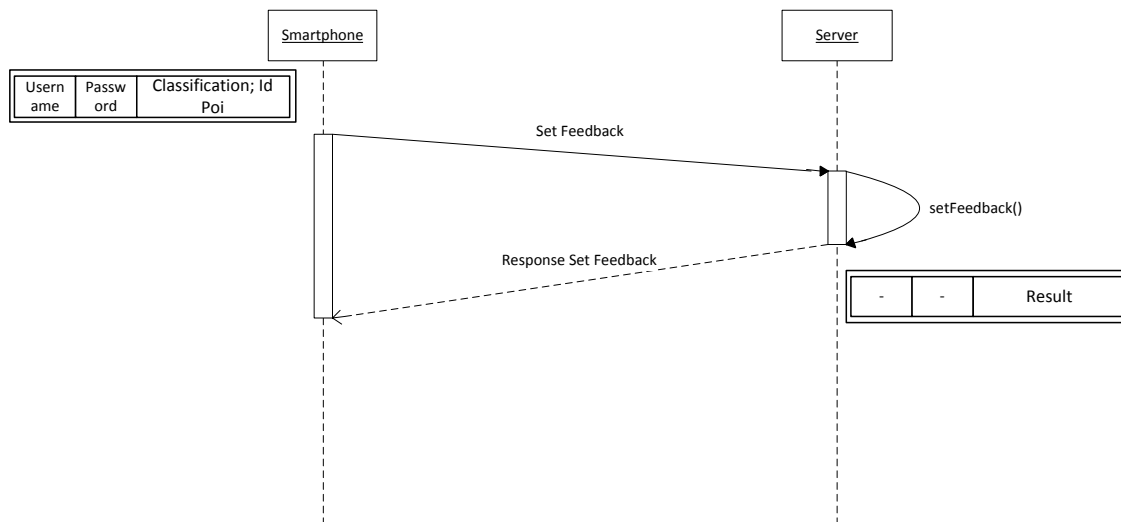


Figura 81 - DS Set Feedback

Exemplo da mensagem JSON enviada:

Mensagem Set Feedback:

```
{"password":"123456","username":"duarte@ola.com","classification":3,"id_poi":"4e8f24b89adfdc56c0d7987e"}
```

Mensagem Response Set Feedback:

```
update
```

G.2.15 Set Survey

Como referido no documento de requisitos a aplicação tem possibilita a realização de inquéritos aos utilizadores. Assim, quando um utilizador pretende enviar a sua opinião/sugestão relativa à aplicação *android*, é enviada uma mensagem com as respostas do inquérito e os dados de autenticação. Quando a mensagem chega ao servidor, este encarrega-se de guardar as respostas na base de dados e enviar uma mensagem de confirmação de sucesso com “OK”.

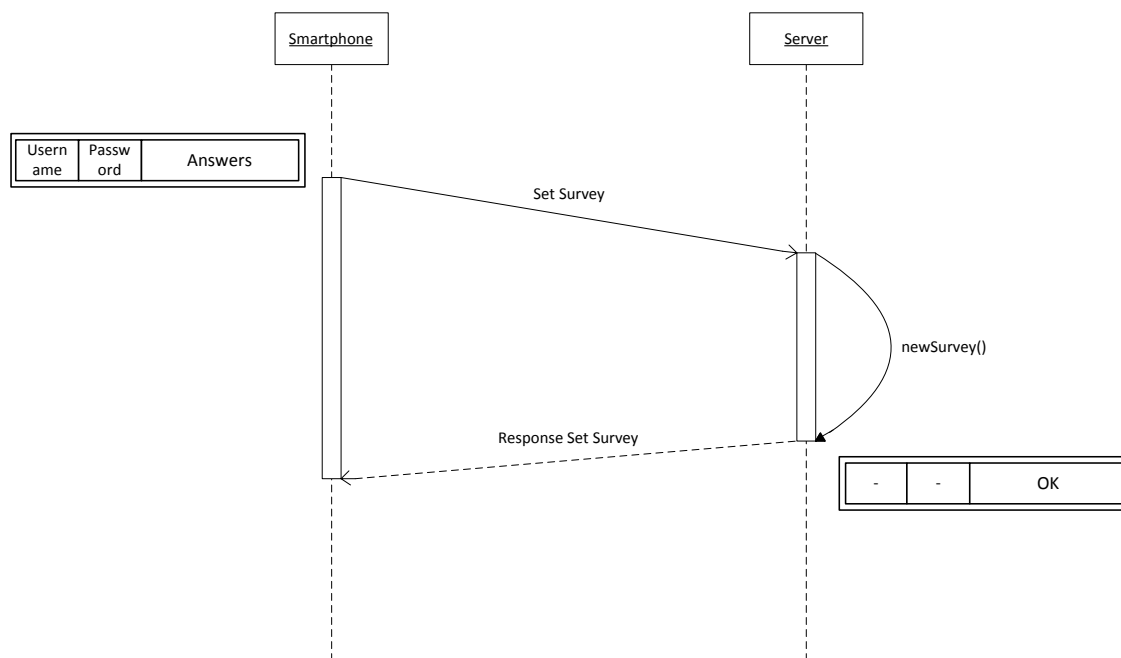


Figura 82 - DS Set Survey

Exemplo da mensagem JSON enviada:

Mensagem Set Survey:

```
{"pergunta_3": "3.0", "pergunta_4": "3.0", "pergunta_5": "3.0", "username": "duarte@ola.com", "pergunta_6": "3.0", "pergunta_1": "3.0", "pergunta_2": "3.0", "desenvolvimento": "olá", "pergunta_7": "2.0", "pergunta_8": "4.0", "password": "123456", "pergunta_9": "2.0"}
```

Mensagem Response Set Survey:

```
ok
```


Anexo H “Requisitos Funcionais do Sistema Avós e Netos”

Avós e Netos

Requisitos Funcionais

`draposo@student.dei.uc.pt`

27-07-2012

Versão 0.4

DOCUMENTO APROVADO POR	Nome, email@student.dei.uc.pt
DATA	[dia] de [Mês] de [AAAA] (data da aprovação)

HISTÓRICO DE REVISÕES

VERSÃO	DATA	AUTOR	DESCRIÇÃO
0.1	27/03/2012	Duarte Raposo	<ul style="list-style-type: none"> Draft.
0.2	30/03/2012	Duarte Raposo	<ul style="list-style-type: none"> Inserção de um esquema geral do sistema; Alterações na Visão Geral da Especificação; Índice de Figuras;
0.3	01/04/2012	Duarte Raposo	<ul style="list-style-type: none"> Inserção de um novo requisito
0.4	26/07/12	Duarte Raposo	<ul style="list-style-type: none"> Revisão do Português e correcção de alguns erros ortográficos;

H.1 Descrição

Este documento descreve todos os requisitos funcionais da aplicação Avós e Netos da plataforma *SocialSense*.

Os requisitos funcionais da aplicação estão descritos através de:

- Diagramas use-case: ilustram o actor e as suas acções ;
- Descrição do requisito;
- A prioridade de cada requisito: *Must, Should*;
- A complexidade do requisito: Baixa, Média e Alta;
- Actores e funções;
- Pré-condições: condições necessárias ao requisito;
- Pós-condições: o resultado esperado do requisito;
- Acção;
- Fluxo Básico de Eventos: o conjunto de eventos necessários na aplicação para efectuar as acções descritas;
- Fluxo Alternativo: descrição de outros fluxos diferente do básico;
- Fluxo de erro: fluxo de erro da aplicação;
- Frequência;
- Assunções;
- Requisitos especiais: algum requisito especial de *performance* ou segurança;
- Problemas;
- Cenários de Utilização: exemplo de um possível cenário de utilização real do requisito.

H.2 Requisitos Funcionais

Seguidamente, serão especificadas as funções externas, as responsabilidades associadas, e todos os casos de uso impulsionados por esses factores externos.

H.2.1 Avós

H.2.1.1 ID-01:Efectuar Login

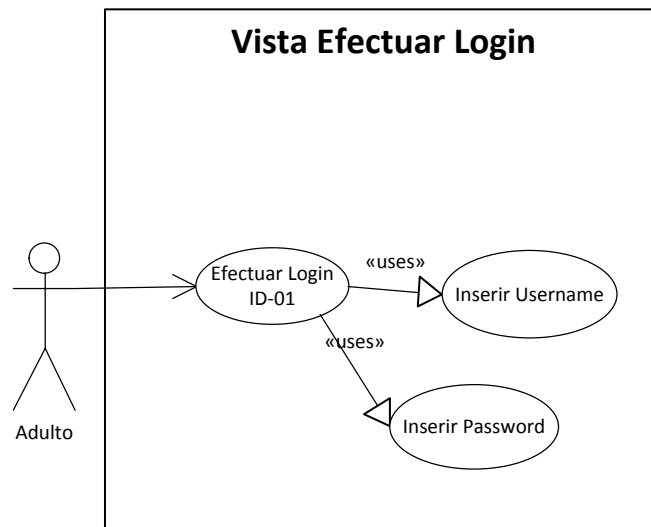


Figura 83 - DCU Efectuar Login

Descrição:

Para o utilizador aceder à aplicação, necessita de fazer *login* com o respectivo *username* e *password*, da conta do *facebook*. Quando este efectua o *login*, a aplicação irá recolher o *acesstoken* do respectivo utilizador e fazer um pedido ao *Facebook* para que este retorne o respectivo *FacebookID*. Este *FacebookID* é então confirmado através do seu envio da aplicação para o nosso *middleware*, que será confirmado através do registo presente na base de dados. Este menu de *login* como já pertence ao *Facebook* não necessita de ser implementado.

Prioridade: *MUST*

Complexidade: *Baixa*

Actores e funções:

- **Actor principal:** *Avós*.

Pré-condições:

- Esta ligação só pode ser iniciada por *https*.
- A conta de utilizador já tem de estar criada no sistema.

Pós-condições:

- **Sucesso:**
 - O utilizador efectua o *login* na aplicação e é redireccionado para o menu inicial da aplicação.
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema, consequentemente é apresentada uma mensagem de erro apropriada, a dizer que existiu uma falha na sua autenticação.

Acção:

- O utilizador pretende aceder ao sistema, para poder utilizar as suas funcionalidades e informações disponíveis, sendo necessário a sua autenticação no sistema.

Fluxo básico de eventos:

- O utilizador insere o *URL* da aplicação no *browser*;
- O *Facebook* mostra um formulário para inserir *login* e palavra-chave de utilizador;
- O utilizador submete os dados de *login*;
- A aplicação juntamente com o *Facebook* e o *middleware* autenticam o utilizador e ele entra na aplicação.

Fluxo alternativo:

- Falha de *login*:
 - O *Facebook* envia o utilizador, para uma página, para que este tente realizar *login* novamente;

Fluxo de erro:

- Se o utilizador não estiver registado é mostrada uma mensagem pela aplicação a dizer que existiu uma falha na autenticação.

Frequência:

- Baixa frequência. Sempre que o utilizador não tiver o *login* efectuado.

Assunções:

- Nenhuma.

Requisitos Especiais:

- **Performance:** Nenhuma.
- **Interface:** A interface será disponibilizada pelo *Facebook*, dado que a inserção dos dados de *login* são efectuados no seu formulário;
- **Segurança:** A senha digitada não deve ser exibida em texto simples.

Problemas:

- *Nenhum.*

Cenários de utilização:

- Por exemplo, os avós têm curiosidade acerca do dia do neto, nomeadamente se este se encontra bem. Para isso, entram na aplicação e verificam se existe algum alerta.

H.2.1.2 ID-03: Visualizar Menu Inicial

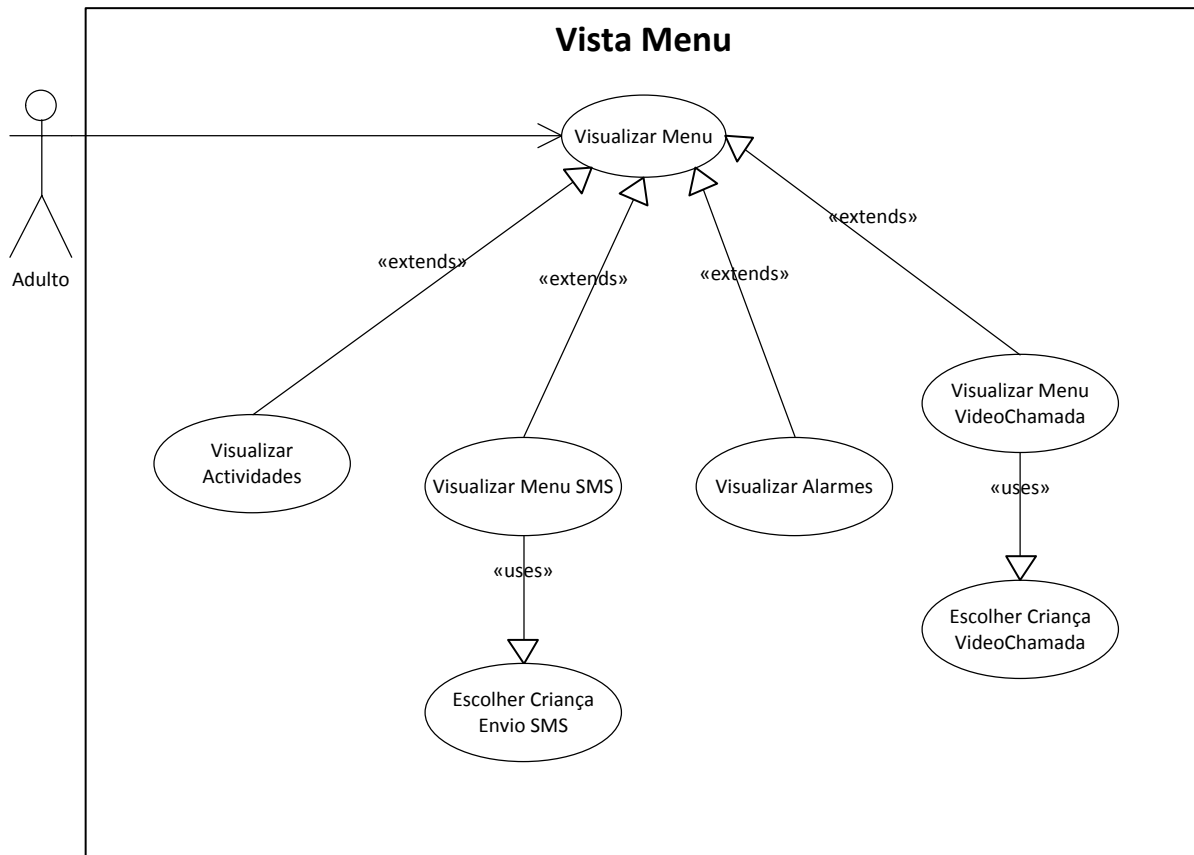


Figura 84 - DCU Visualizar Menu Inicial

Descrição:

Quando o utilizador entrar na página principal da aplicação encontrará um menu, que contém vários botões que permitem aceder a outras páginas (página das actividades, envio de *sms*, visualização de alarmes e videochamada).

Todos os botões ao serem clicados deverão apresentar uma lista, que permite escolher as crianças do respectivo adulto, à excepção do menu de alertas. Esta lista deve ser constituída pela foto da criança, bem como o seu nome.

O menu tem duas características importantes, que são a possibilidade de minimizar, ficar representado por um pequeno rectângulo e restaurar a sua posição inicial.

Além disso, deve conter duas setas em cada extremidade que possibilite a navegação entre os diferentes botões, na possibilidade de existência de um grande número.

Prioridade: MUST

Complexidade: Media-Alta

Actores e funções:

- **Actor principal:** Adulto.

Pré-condições:

- O menu deve ter um interface suave e muito intuitivo, não devendo apresentar nenhum obstáculo à navegação do utilizador;
- Deve ser feito através da *framework* Dojo;
- O utilizador tem que ter feito *login* e estar na página principal.

Pós-condições:

- **Sucesso:** O utilizador pode seleccionar as várias opções descritas em cima, como navegar para diferentes páginas, minimizar e restaurar o menu.
- **Falha:** Em caso de falha em alguma acção do menu, este deve ser novamente carregado de modo que permita a utilização das restantes acções.

Acção:

- O utilizador pretende navegar entre as páginas do sistema, para isso, selecciona as várias opções presentes no menu.

Fluxo básico de eventos:

- O utilizador entra na página principal;
- O utilizador navega no menu e clica em cima de uma opção;
- O utilizador é redireccionado para a página seguinte.

Fluxo alternativo:

- Falha na acção do botão: Se existir uma falha na acção do botão, o utilizador não deve navegar para nenhuma página, permanecendo assim na página principal.

Fluxo de erro:

- Não existirá nenhum fluxo de erro.

Frequência:

- Média: Sendo uma acção que permite a navegação deve ter uma frequência de acesso mediana.

Assunções:

- Assume-se que o utilizador tem uma experiência mínima na utilização do computador e possui um *browser* que suporte *JavaScript*;

Requisitos Especiais:

- **Performance:** Nenhuma;
- **Interface:** A interface tem que ter aspectos suaves e ser construída através da *framework* DOJO;
- **Segurança:** Não se aplica.

Problemas:

- Nenhum.

Cenários de utilização:

- O adulto encontra-se na página principal e pretende navegar até outro menu.

H.2.1.3 ID-04: Vista Mensagens

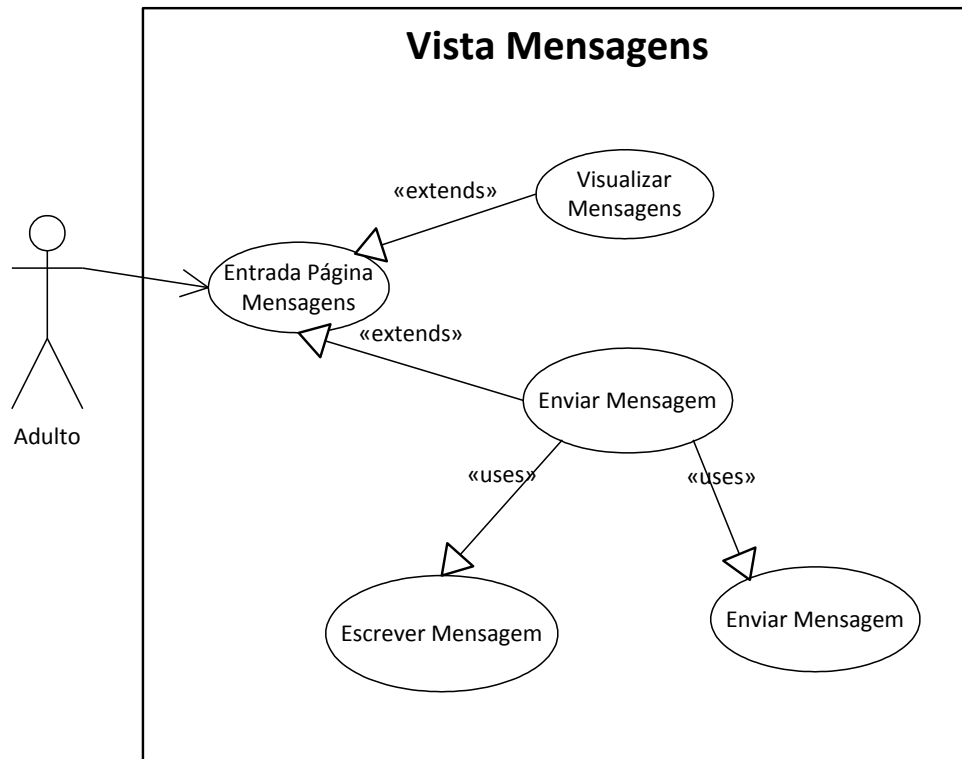


Figura 85 - DCU Vista Mensagens

Descrição:

Ao escolher a opção de “mensagens” do menu principal, o utilizador acede a uma página referente às mensagens armazenadas no sistema, trocadas entre ele mesmo e a criança

previamente seleccionada no menu. Neste ponto, o utilizador acede a uma tabela contendo os títulos e a data de todas mensagens armazenadas referentes a conversas com a criança seleccionada. Caso o número de mensagens seja demasiado elevado para as mostrar todas na mesma janela, o utilizador poderá fazer “scroll” pela lista, até encontrar a mensagem que pretende visualizar. O utilizador poderá fazer “hover” em cima de um determinado título, e a cor da linha correspondente deverá ser alterada para uma cor mais escura, voltando à cor original quando o utilizador fizer “hover” sobre outra linha, ou mover o rato para fora da tabela. De forma a ler a mensagem que pretende, o utilizador poderá “cliquear” em cima da linha com o título e data correspondentes. Ao clicar numa linha da tabela, a página *Web* deverá efectuar uma animação: no lugar onde se encontrava a tabela com os títulos, deverá aparecer um “quadrado” com o mesmo tamanho da tabela, com o conteúdo da mensagem seleccionada. Acima deste quadrado (aproximadamente no local onde se mostrava a primeira linha) deverá estar informação referente ao título e a data da mensagem. Acima da lista das mensagens deverá estar informação sobre a criança seleccionada (por exemplo, o nome e fotografia). A página deverá conter também um botão que permita aceder a uma a página de envio de mensagem. Ao clicar neste botão, o utilizador é encaminhado para uma página onde será apresentado uma caixa de texto para compor uma nova mensagem. Esta mensagem é enviada ao clicar num botão de envio de mensagem. O sistema ficará responsável por reencaminhar esta mensagem para um sistema de envio de mensagens SMS, sendo que, a mensagem irá ser enviada para o telemóvel da criança correspondente.

Prioridade: SHOULD

Complexidade: Elevada

Actores e funções:

- **Actor principal:** Adulto;

Pré-condições:

- O Adulto terá de ter feito login no sistema;
- O Adulto tem de seleccionar a opção “mensagens” no menu principal;
- O Adulto tem de seleccionar a criança que pretende, na lista que aparece abaixo da caixa “mensagens” do menu principal.

Pós-condições:

- **Sucesso:**
 - O utilizador selecciona a mensagem que pretende e lê o seu conteúdo;
 - O utilizador compõe uma mensagem e envia para a criança. A criança recebe a mensagem no seu telemóvel.
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema. É apresentada uma mensagem de erro apropriada;
 - O utilizador envia a mensagem, mas esta não é entregue no telemóvel da criança. Deve ser apresentada uma notificação.

Acção:

- O utilizador pretender entrar no sistema com o objectivo de aceder às mensagens registadas no sistema, sendo esse acesso feito através da sua autenticação no sistema e à selecção apropriada no menu.

Fluxo básico de eventos:

- O utilizador faz *login* na aplicação;
- O utilizador selecciona a opção “mensagens” e a criança que pretende no menu principal;
- O utilizador tem acesso a uma tabela contendo os títulos e as datas das mensagens;
- O utilizador selecciona uma mensagem e lê o seu conteúdo;
- O utilizador clica no botão de envio de mensagem;
- O utilizador tem acesso a um editor de texto onde pode compor a mensagem;
- O utilizador envia a mensagem, clicando no botão “enviar”;
- A mensagem é enviada para o telemóvel da criança;
- Visualização do estado de entrega da mensagem.

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de enviar a mensagem, logo deve de apresentar uma mensagem com essa informação;

Frequência:

- Baixa frequência.

Assunções:

- Assume-se que o cliente tem uma ligação segura para proceder à execução da aplicação;

Requisitos Especiais:

- **Performance:** As mensagens devem de ser entregues dentro de um tempo semelhante aos do envio através de um telemóvel.
- **Interface:** A interface tem de ter aspectos suaves e ser construída através da *framework* DOJO;
- **Segurança:** Só podem aceder a esta funcionalidade os utilizadores que tiverem efectuado o *login*.

Problemas:

- Quando uma mensagem não for entregue, deve ser apresentado ao utilizador uma mensagem.

Cenários de utilização:

- Um avô adoptivo acede à plataforma *SocialSense* e vai ler as últimas mensagens enviadas pelo seu “neto” de adopção. Decide responder, compondo e enviando uma nova mensagem.

H.2.1.4 ID-05: Vista Alarmes

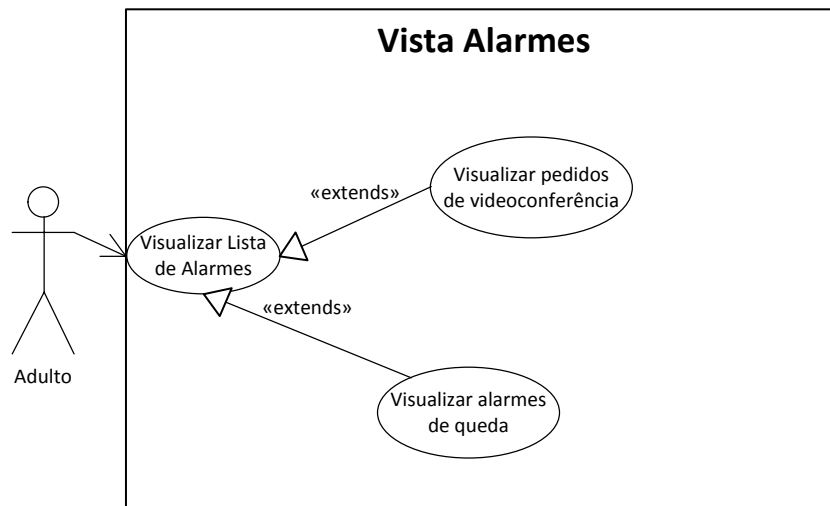


Figura 86 - DCU Vista Alarmes

Descrição:

Ao escolher a opção de “alarmes” no menu principal, o utilizador acede a uma página referente aos alarmes armazenados no sistema, referentes às crianças por ele adoptadas. Neste ponto, o utilizador visualiza uma tabela contendo o tipo, a data e o nome da criança de todos os alarmes presentes na base de dados. Caso o número de alarmes seja demasiado elevado para serem mostrados na mesma janela, o utilizador poderá fazer “scroll” pela lista. Os alarmes deverão estar organizados por data, sendo que os alarmes mais recentes são mostrados primeiro. Os tipos de alarmes mostrados poderão ser “alarme de queda” ou “pedido de videoconferência”, sendo que cada tipo de alarme deverá ter um ícone associado, que traduz visualmente o tipo de alarme.

Prioridade: MUST

Complexidade: Media

Actores e funções:

- **Actor principal:** Adulto

Pré-condições:

- O Adulto terá de ter feito *login* no sistema;
- O Adulto tem de seleccionar a opção “alarmes” no menu principal.

Pós-condições:

- **Sucesso:**
 - O utilizador consegue ter acesso a todos os alarmes;
 - O utilizador consegue apagar todos os alarmes;
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema. É apresentada uma mensagem de erro apropriada.

Acção:

- O utilizador pretender entrar no sistema, com o objectivo de aceder aos alarmes registados, sendo esse acesso feito através da sua autenticação no sistema e à selecção apropriada no menu.

Fluxo básico de eventos:

- O utilizador faz *login* na aplicação;
- O utilizador selecciona a opção “alarmes” no menu principal;
- O utilizador tem acesso a uma tabela contendo os tipos de alarme, datas e nome das crianças correspondentes;
- O utilizador faz “scroll” pela lista de alarmes activos.

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de enviar a mensagem, logo deve de apresentar uma mensagem com essa informação.

Frequência:

- Média.

Assunções:

- Nenhuma.

Requisitos Especiais:

- **Performance:** Os alarmes de queda devem ser detectados e mostrados com um atraso inferior a pelo menos 1 minuto;
- **Interface:** A interface tem que ter aspectos suaves e ser construída através da *framework* DOJO;
- **Segurança:** Esta funcionalidade só está acessível se os utilizadores efectuarem o login.

Problemas:

- O servidor de *middleware* não está disponível; os alarmes não são mostrados.

Cenários de utilização:

- Um avô adoptivo acede à plataforma *SocialSense* e vai consultar os alarmes activos, detectados pelo sistema. Verifica que existe um alarme de queda e acede ao menu “mensagens” para enviar uma mensagem ao “neto” adoptado a perguntar se está tudo bem.
Um avô adoptivo acede à plataforma *SocialSense* e vai consultar os alarmes activos, detectados pelo sistema. Verifica que existe um alarme de pedido de videoconferência e acede ao menu “videoconferência” para iniciar uma chamada para a instituição onde se encontra o “neto” adoptado.

H.2.1.5 ID-06: Vista Efectuar Videoconferência

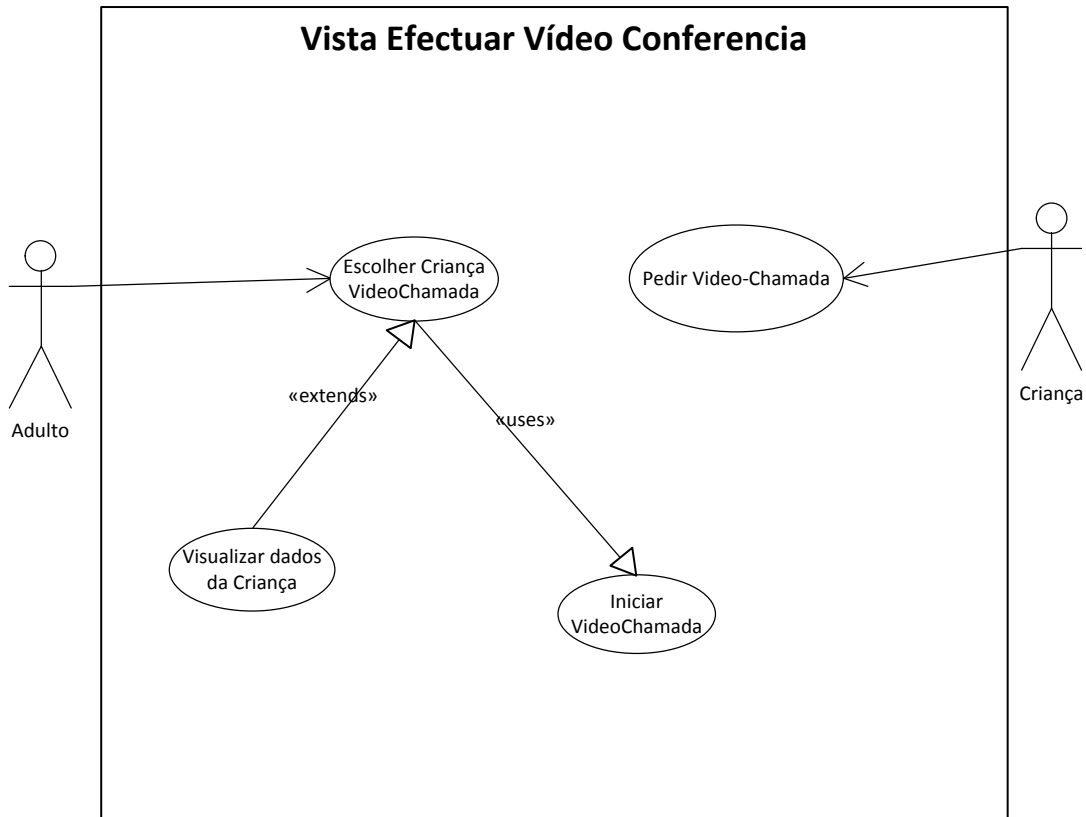


Figura 87 - DCU Efectuar Videoconferência

Descrição:

O utilizador acede ao menu de videochamadas após escolher a opção de "videochamada" no menu principal, seguindo-se a escolha da criança pretendida no menu descendente que aparece ao fazer *hover* com o rato. Ao escolher a criança, o utilizador é levado para uma página onde são apresentados alguns dados básicos da criança, tais como a sua fotografia,

nome, data de nascimento, naturalidade e nacionalidade. Estas informações deverão ser apresentadas numa "caixa" central. A página deverá também apresentar um botão de chamada, dentro da caixa onde são apresentadas as informações pessoais. De forma a seguir o padrão introduzido pelos telemóveis, o botão de chamar deverá ser um botão circular verde, com um ícone de um telefone. Ao pressionar este botão, o sistema acede à base de dados por forma a obter o contacto *Skype* da instituição onde se encontra a criança. A aplicação *Skype* inicia-se no computador do utilizador adulto, e uma chamada *Skype* é automaticamente iniciada para o contacto obtido. Esta inicialização é conseguida à custa de uma API *Javascript* fornecida pelo próprio Skype.

Prioridade: MUST

Complexidade: Elevada

Actores e funções:

- **Actor principal:** Adulto

Pré-condições:

- O Adulto terá de ter feito login no sistema;
- O Adulto tem de escolher a criança no menu principal;
- Necessária instalação e configuração da aplicação *Skype*.

Pós-condições:

- **Sucesso:**
 - O utilizador consegue ter acesso aos dados básicos da criança;
 - O utilizador consegue realizar a videochamada.
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema. É apresentada uma mensagem de erro apropriada;
 - O *Skype* está mal configurado e o utilizador não consegue realizar uma videochamada;
 - A criança encontra-se ausente e, por isso, apesar de todos os sistemas funcionarem, o utilizador não consegue comunicar.

Acção:

- O utilizador quer ver a criança, logo inicia uma chamada através do *press* no botão da videochamada;

Fluxo básico de eventos:

- O utilizador entra na aplicação;
- O utilizador escolhe a criança na opção de videochamada no menu;
- O utilizador entra na página de videochamada e carrega no botão;
- A aplicação abre-se e começa a chamar para a criança;
- A criança atende e o adulto fala com a criança;

- A chamada termina.

Fluxo alternativo:

- Falha na chamada:
- Se ninguém atender o adulto, este tem que voltar a carregar no botão na página de videochamada.

Fluxo de erro:

- Sendo a chamada efectuada pela aplicação *Skype*, qualquer erro relacionado com a chamada será tratado pela aplicação *Skype*.

Frequência:

- Média.

Assunções:

- Assume-se que o utilizador tem os dispositivos físicos necessários para efectuar a videochamada;

Requisitos Especiais:

- **Performance:** A ligação da internet deve ter velocidade suficiente para se conseguir efectuar a videochamada em perfeitas condições;
- **Interface:** A interface tem que ter aspectos suaves e ser construída através da *framework DOJO*;
- **Segurança:** Só podem aceder a esta funcionalidade os utilizadores que tiverem efectuado o *login*.

Problemas:

- A criança não estar disponível para atender a chamada;
- O *Skype* não estar disponível;
- Possível dificuldade de utilização do *Skype*.

Cenários de utilização:

- Uma criança pretende dizer ao avô(ó) que não se está a sentir bem. Logo, carrega no botão e o(a) avô(ó) inicia a videochamada.

H.2.1.6 ID-07:Vista Actividade

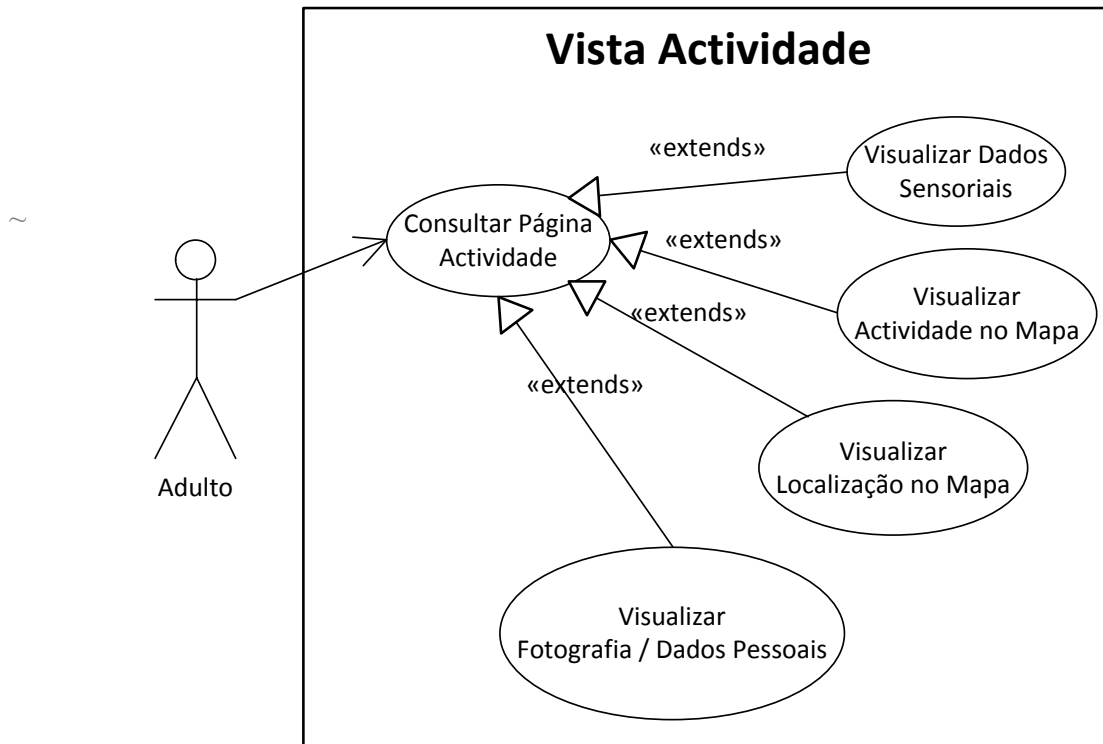


Figura 88 - DCU Vista Actividade

Descrição:

Ao escolher a opção de “Actividade” do menu principal, o utilizador acede a uma página referente à actividade, localização e valores sensoriais actualmente medidos pela componente de monitorização do sistema, referentes à criança previamente seleccionada no menu. Neste ponto, o utilizador acede a uma interface gráfica que apresenta várias componentes, uma para cada tipo de dados a ser mostrado. Uma das componentes representa alguns dados pessoais da criança em causa, tais como a sua fotografia, o seu nome, a data-de-nascimento, naturalidade e nacionalidade. Estas informações são apresentadas dentro de um “quadrado” na parte superior esquerda da página. Uma outra componente apresenta um mapa (*GoogleMaps* ou *Sapo Mapas*) onde se pode visualizar uma planta da *testbed* da aplicação. Nesta planta deverá poder ver-se um *avatar* (um pequeno “boneco” animado) cuja posição na planta da *testbed* reflecte a posição relativa da criança na instituição, em tempo real. O avatar também deverá alternar a sua animação, dependendo do sinal do acelerómetro ser classificado como “movimento” ou “não-movimento”. Este mapa de actividade / localização deverá ser apresentado na parte inferior direita da página. Ao lado do mapa, na parte inferior esquerda da página, deverá ser apresentada uma tabela contendo informação, em formato numérico, referente aos dados adquiridos pelos sensores sem fios e pelos dispositivos médicos ligados ao telemóvel com tecnologia J2ME. Os tipos de dados mostrados nesta tabela são os seguintes:

- Temperatura ambiente;

- Humidade ambiente;
- Oximeter bmp;
- Oximetry;
- Sphygmomanometer systolic;
- Sphygmomanometer diastolic;
- Spirometer.

Prioridade: MUST

Complexidade: Elevada

Actores e funções:

- **Actor principal:** Adulto

Pré-condições:

- O Adulto terá de ter feito login no sistema;
- O Adulto tem de seleccionar a opção “actividade” no menu principal;
- A criança tem que se encontrar com o sensor.

Pós-condições:

- **Sucesso:**
 - O utilizador consegue ter acesso ao mapa de actividade e localização;
 - O utilizador consegue ter acesso à fotografia e dados pessoais da criança;
 - O utilizador consegue ter acesso às informações sensoriais.
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema. É apresentada uma mensagem de erro apropriada;
 - O sistema de localização está inactivo devido a uma falha do sistema. O avatar não irá mover-se;
 - O sistema de actividade está inactivo devido a uma falha do sistema. O avatar não altera a sua animação (mas mantém a animação anterior);
 - Os sensores médicos não foram utilizados;
 - Os sensores *Zolertia* não enviam dados de temperatura e / ou humidade;
 - Os sensores médicos não foram utilizados durante algum tempo. Os valores correspondentes da tabela de dados sensoriais aparecem desactualizados;
 - Os sensores *Zolertia* não enviam dados de temperatura e / ou humidade. Os valores correspondentes da tabela aparecem desactualizados.

Acção:

- O utilizador pretender entrar no sistema, com o objectivo de aceder à actividade actual da criança, sendo esse acesso feito através da sua autenticação no sistema e à selecção apropriada no menu.

Fluxo básico de eventos:

- O utilizador faz login na aplicação;
- O utilizador selecciona a opção “actividade” no menu principal;
- O utilizador tem acesso a uma tabela contendo os dados sensoriais dos sensores da criança;
- O utilizador tem acesso aos dados básicos da criança, nomeadamente a sua fotografia, nome, data de nascimento, naturalidade e nacionalidade;
- O utilizador tem acesso a um mapa contendo a posição relativa e actividade da criança, e estas são actualizadas em tempo real

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de enviar a mensagem, logo deve de apresentar uma mensagem com essa informação.

Frequência:

- Alta frequência.

Assunções:

- Assume-se que a criança tem sempre o sensor consigo e em pleno funcionamento.

Requisitos Especiais:

- **Performance:** A actualização de actividade e localização, bem como dos valores sensoriais deve ser feita com um atraso inferior a 1 minuto.
- **Interface:** A interface tem que ter aspectos suaves e ser construída através da *framework* DOJO;
- **Segurança:** Só podem aceder a esta funcionalidade os utilizadores que tiverem efectuado o *login*.

Problemas:

- O servidor de *middleware* não está disponível; os valores de actividade, localização e valores sensoriais não são mostrados.

Cenários de utilização:

- Um(a) avô(ó) acede à plataforma *SocialSense* e vai consultar a actividade actual do neto adoptado.

H.2.2 Criança

H.2.2.1 ID-08: Pedir videochamada

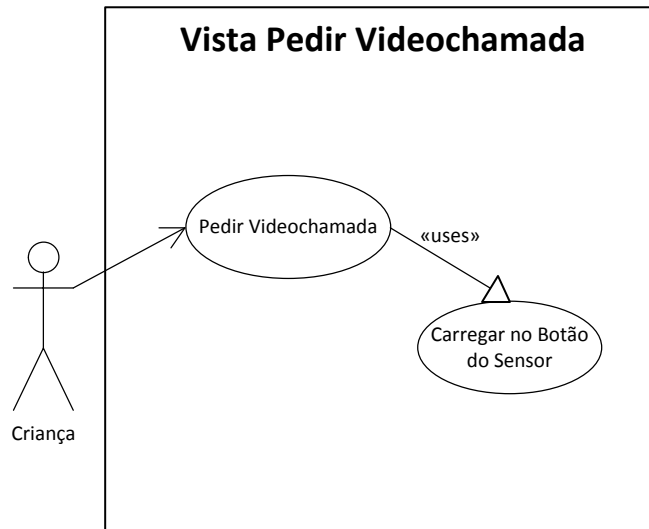


Figura 89- DCU Vista Pedir Videochamada

Descrição:

Para a criança poder fazer pedidos de videochamada, esta irá ter que carregar no botão do sensor *Zolértia*. Este sensor transportado pela criança será utilizado para adquirir dados como a aceleração e temperatura e para accionar chamadas de videoconferência para o(a) avô(ó).

A videoconferência irá utilizar a aplicação *Skype* dos dois lados, tanto da criança como do adulto. Para isso, tanto o adulto como a instituição da criança têm que ter um computador local com a aplicação *Skype* devidamente instalada/configurada.

Quando a criança carrega no botão irá ser enviada uma mensagem através do sensor *Zolértia* para o *proxy*, que por sua vez, envia para o *middleware*, seguindo-se uma notificação do *middleware* para a aplicação *Facebook*, de modo a informar o adulto.

Para finalizar o processo, o utilizador irá aceder à aplicação *Facebook* e ao visualizar a notificação presente na lista de alertas, iniciar-se-à a videochamada.

Prioridade: MUST

Complexidade: Media

Actores e funções:

- **Actor principal:** Criança.

Pré-condições:

- Os dados do utilizador e da criança terão que estar presentes no sistema. Ex: Skype Id;
- A instituição terá que ter um computador com *webcam*, tal como o adulto;
- A criança tem que transportar um sensor *Zolértia*.

Pós-condições:

- **Sucesso:**
 - ✓ A criança efectua o pedido através do sensor e o adulto contacta com a criança, através da visualização do alerta na aplicação *Facebook*.
- **Falha:**
 - ✓ A criança carrega no botão do sensor, mas como o adulto não se encontra a monitorizar naquele momento a criança, não inicia a chamada.

Acção:

- A criança pretende contactar o adulto e carrega no botão.

Fluxo básico de eventos:

- A criança carrega no botão do sensor;
- O sensor envia o evento para o *proxy* e o *proxy* para o *middleware*;
- O *middleware* grava o evento na base de dados;
- O adulto acede ao menu dos alarmes na base de dados e identifica um pedido de videochamada;
- O adulto vai ao menu de videoconferência e inicia a chamada para a criança;
- A criança encontra-se ao pé do computador com *webcam* da instituição.

Fluxo alternativo:

- O adulto não está presente: Se o adulto não estiver presente não será possível fazer a chamada, dado que o adulto não pode iniciar a videochamada;
- A criança não está presente: Se a criança não estiver presente no local, o adulto não pode fazer a videochamada;

Fluxo de erro:

- Se existir algum tipo de erro na aplicação *Web* será apresentada uma página com os erros;
- Se houver algum erro por parte do *Skype*, a própria aplicação *Skype* tratará o erro.

Frequência:

- Média.

Assunções:

- Assume-se que os dois actores têm computador com o equipamento necessário;

Requisitos Especiais:

- **Performance:** Necessária uma ligação à Internet que permita uma videochamada sem interrupções.

- **Interface:** A interface será semelhante à do restante website, pretende-se que se tenha cuidado na elaboração do interface para que, seja acessível a pessoas idosas.
- **Segurança:** A segurança da ligação é efectuada pela aplicação *Skype*.

Problemas:

- A possível ausência da criança ou do adulto.

Cenários de utilização:

- A criança sente-se doente e quer dizer ao adulto que não se está a sentir bem.

H.2.2.2 ID-09: Aceder a Mensagens

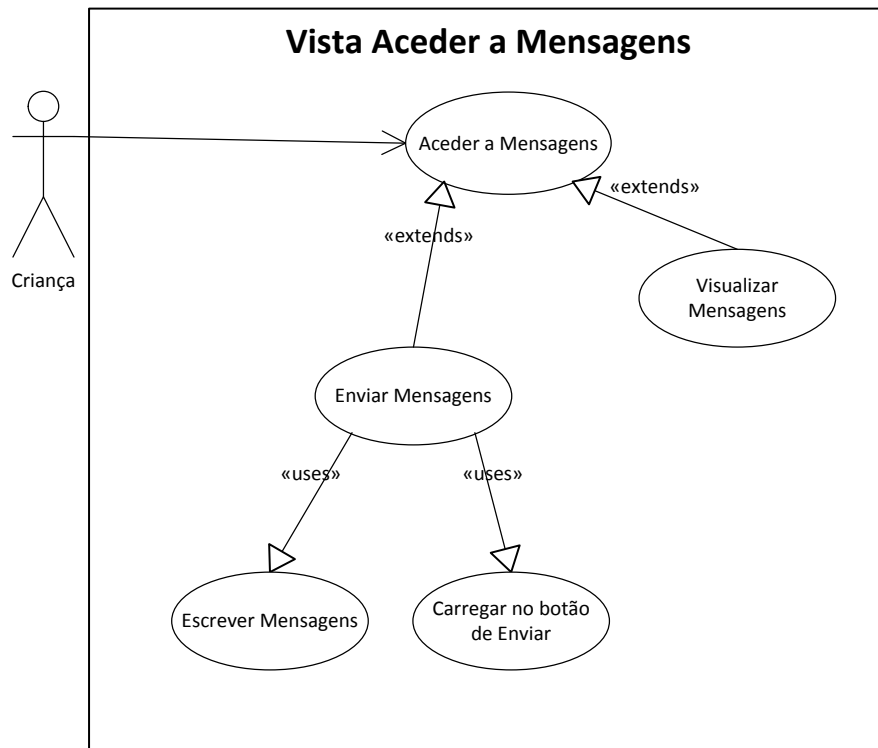


Figura 90 - DCU Aceder a Mensagens

Descrição:

Como a aplicação do *Facebook* permite o envio e recepção de mensagens, a criança poderá utilizar o telemóvel para receber mensagens do adulto, ou para enviar mensagens para o adulto. O tratamento destas mensagens é efectuado pelo *gateway* de sms que se encontra a correr no nosso *middleware*.

A presente vista descreve assim as acções básicas que o telemóvel precisa de implementar para o envio e recepção das mensagens. Como os telemóveis contêm a sua própria aplicação, este caso de uso não vai ser implementado.

O adulto pode enviar e receber as mensagens na aplicação Facebook como mostramos no caso de uso Figura 91.

Prioridade: NA

Complexidade: NA

Actores e funções:

- **Actor principal:** Criança.

Pré-condições:

- Existência de um telemóvel do lado da criança;

Pós-condições:

- **Sucesso:** NA
- **Falha:** NA

Acção:

- A criança envia uma mensagem. A criança escreve uma mensagem.

Fluxo básico de eventos:

- O adulto envia uma mensagem;
- A mensagem é enviada da aplicação para o *middleware*;
- O *middleware* envia a mensagem através do *gateway* de *sms*;
- A mensagem é entregue no telemóvel da criança;
- A criança visualiza a mensagem.

Fluxo alternativo:

- Nenhum;

Fluxo de erro:

- Não vamos tratar erros sobre o envio das mensagens.

Frequência:

- Média.

Assunções:

- Assume-se que a criança sabe ler e escrever, tal como, utilizar o telemóvel;

Requisitos Especiais:

- **Performance:** Nenhuma.
- **Interface:** A interface será a definida pelo telemóvel;
- **Segurança:** NA;

Problemas:

- Possíveis falhas com o envio das mensagens;

Cenários de utilização:

- O adulto quer informar a criança que não será possível ir busca-la à escola.

H.2.2.3 ID-10: INICIAR VIDEOCHAMADA

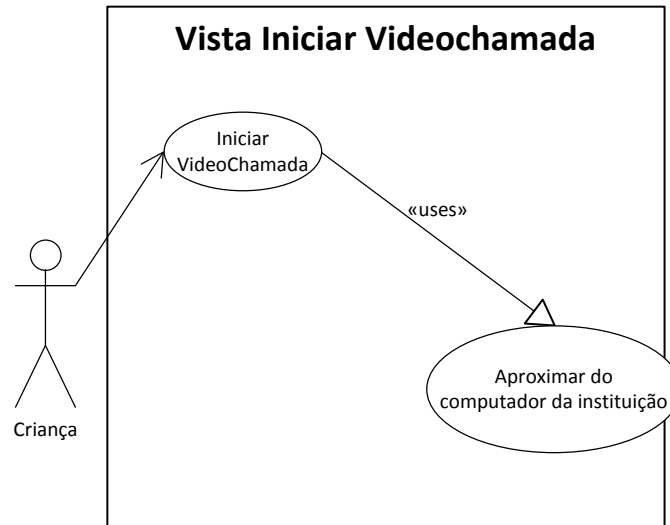


Figura 91- DCU Vista Iniciar Videochamada

Descrição:

Apesar de a criança poder fazer pedidos de videochamada, esta também pode iniciar a videochamada directamente para o seu tutor através do sensor que transporta. Esta videochamada é iniciada com base na distância que a criança se encontra do computador da instituição, ou seja, se a criança durante uns determinado segundos permanecer perto do computador este detecta a sua presença e inicia uma videochamada para o tutor através da aplicação *Skype*.

Prioridade: MUST

Complexidade: Media

Actores e funções:

- **Actor principal:** Criança.

Pré-condições:

- Os dados do utilizador e da criança terão que estar presentes no sistema. Ex: *Skype Id*;
- A instituição terá que ter um computador com *webcam*, bem como o adulto;
- A criança tem que transportar um sensor.

Pós-condições:

- **Sucesso:**
 - ✓ A criança inicia a videochamada para o seu tutor através da aproximação do sensor que transporta ao computador da instituição. A chamada inicia-se passado uns segundos, a videochamada inicia-se e o tutor atende.
- **Falha:**
 - ✓ A criança aproxima-se do computador da instituição, a chamada inicia-se mas o tutor não está presente, logo não pode atender.

Acção:

- A criança pretende contactar o tutor logo aproxima-se do computador da instituição, transportando o sensor.

Fluxo básico de eventos:

- A criança coloca o sensor;
- A criança aproxima-se do computador da instituição;
- A chamada inicia-se através da aplicação *Skype*;
- O tutor atende a chamada.

Fluxo alternativo:

- O tutor não está presente: Se o tutor não estiver presente não será possível atender a chamada;
- A criança não está presente: Se a criança não estiver presente no local, não é possível o adulto fazer a videochamada;

Fluxo de erro:

- A aplicação *Skype* é a responsável pela videochamada, se existir algum erro, será a aplicação *Skype* a tratar esse erro;

Frequência:

- Média.

Assunções:

- Assume-se que os dois actores têm computador com o equipamento necessário;

Requisitos Especiais:

- **Performance:** Necessária uma ligação à Internet que permita uma videochamada sem interrupções.
- **Interface:** Nenhum.
- **Segurança:** A segurança da ligação é efectuada pela aplicação *Skype*.

Problemas:

- A possível ausência do adulto.

Cenários de utilização:

- A criança sente-se doente e quer dizer ao adulto que não se está a sentir bem.

Anexo I “Requisitos Funcionais do Sistema Youth5G”

Youth5G

Documento de Requisitos

draposo@student.dei.uc.pt

02-06-2012

Versão 0.5

DOCUMENTO APROVADO POR	Nome, email@student.dei.uc.pt
DATA	[dia] de [Mês] de [AAAA] (data da aprovação)

HISTÓRICO DE REVISÕES

VERSÃO	DATA	AUTOR	DESCRIÇÃO
0.1	14/04/2012	Duarte Raposo	<ul style="list-style-type: none"> Draft.
0.2	18/04/2012	Duarte Raposo	<ul style="list-style-type: none"> Descrição do Projecto
0.3	18/04/2012	Duarte Raposo	<ul style="list-style-type: none"> Revisão dos Requisitos
0.4	19/04/2012	Duarte Raposo	<ul style="list-style-type: none"> Revisão dos Requisitos
0.5	25/04/2012	Duarte Raposo	<ul style="list-style-type: none"> Alteração para a API Foursquare.

I.1 Descrição

Este documento apresenta todos os requisitos funcionais da aplicação Youth5G através de diagramas Use-Case.

Os requisitos são descritos através dos seguintes critérios:

- Diagramas *use-case*: ilustram o actor e as suas acções;
- Descrição do requisito;
- A prioridade de cada requisito: *Must, Should*;
- A complexidade do requisito: Baixa, Média e Alta;
- Actores e funções;
- Pré-condições: condições necessárias ao requisito;
- Pós-condições: o resultado esperado do requisito;
- Acção;
- Fluxo Básico de Eventos: o conjunto de eventos necessários na aplicação para efectuar as acções descritas;
- Fluxo Alternativo: descrição de outros fluxos diferente do básico;
- Fluxo de erro: fluxo de erro da aplicação;
- Frequência;
- Assunções;
- Requisitos especiais: algum requisito especial de *performance* ou segurança;
- Problemas;
- Cenários de Utilização: exemplo de um possível cenário de utilização real do requisito.

I.2 Requisitos Funcionais

Esta subsecção descreve e especifica as funções externas, as responsabilidades associadas e todos os casos de uso impulsionados por esses factores externos.

I.2.1 Utilizador

I.2.1.1 ID 1.1 Efectuar Autenticação

Esta subsecção especifica o requisito de efectuar autenticação.

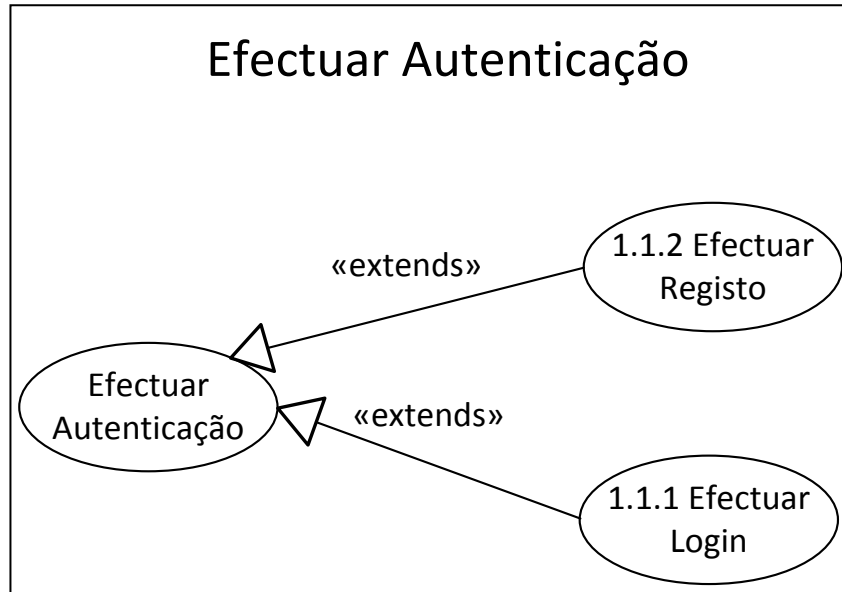


Figura 92 - DCU Efectuar Autenticação

ID 1.1.1 Efectuar login

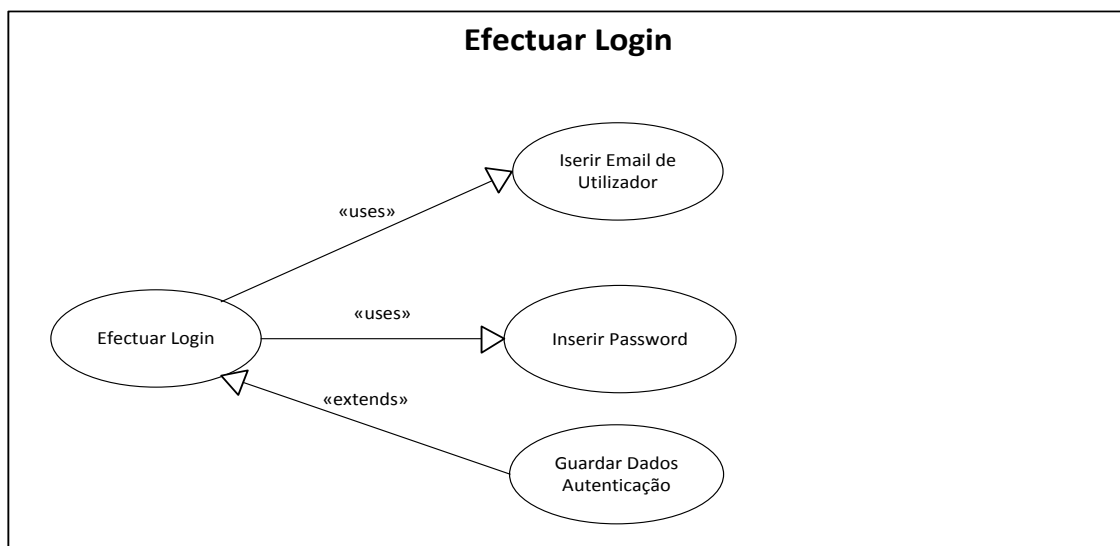


Figura 93 - DCU Efectuar Login

Descrição:

Ao executar a aplicação, o utilizador começará por efectuar *login* para aceder ao sistema. Assim, neste ponto o utilizador efectua o *login* com o seu endereço de *email* e respectiva palavra-chave utilizada no registo da aplicação.

Este menu de sistema terá de conter dois campos: um para o utilizador inserir o seu endereço de *email* e outro para inserir a sua palavra-chave, onde neste último campo a informação não deve ser apresentada em *cleartext*. Existirá também uma *checkbox* que ao estar seleccionada permitirá guardar os dados relativos ao nome do utilizador que pretende aceder à aplicação. Por fim, existirá um botão de *login* para que seja disparado todo o processo de autenticação.

Prioridade: MUST

Complexidade: Média-Baixa

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- A conta de utilizador já tem de estar criada no sistema.

Pós-condições:

- **Sucesso:**
 - O utilizador efectua o *login* na aplicação e é redireccionado para o menu inicial da aplicação.
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema, logo é apresentada uma mensagem de erro apropriada.

Acção:

- O utilizador pretende entrar no sistema, com o objectivo de aceder a informações disponíveis na aplicação, sendo esse acesso feito através da sua autenticação no sistema.

Fluxo básico de eventos:

- O utilizador executa a aplicação;
- A aplicação mostra o campo para inserir o nome de utilizador e palavra-chave;
- O utilizador tem a opção de guardar as informações de *login*;
- O utilizador submete os dados de *login*;
- O sistema valida os dados de *login* e o utilizador entra no menu inicial da aplicação.

Fluxo alternativo:

- **Falha de login:**

- O sistema rejeita o *login* do utilizador porque o email do utilizador ou a palavra-chave são inválidos;
- O sistema mostra uma mensagem de erro de *login* e mostra novamente o formulário de *login*.

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Baixa frequência: Sempre que o utilizador não tiver o *login* efectuado.

Assunções:

- Assume-se que o cliente tem uma ligação à rede externa (*internet*) para proceder à execução da aplicação.

Requisitos Especiais:

- ***Performance:*** Nenhum.
- ***Interface:*** A *interface* vai ter de ser um sistema simples e intuitivo onde o utilizador proceda ao seu login sem grandes complicações.
- ***Segurança:*** A palavra-chave inserida não deve ser exibida em *cleartext*.

Problemas:

- Nenhum.

Cenários de utilização:

Um utilizador tenta aceder à aplicação *Youth5G* com o intuito de visualizar um determinado POI com o seu respectivo evento. Para isso, é necessário proceder ao *login* na aplicação inserindo o seu endereço de *email* e a sua palavra-chave.

ID-1.1.2:Efetuar registo

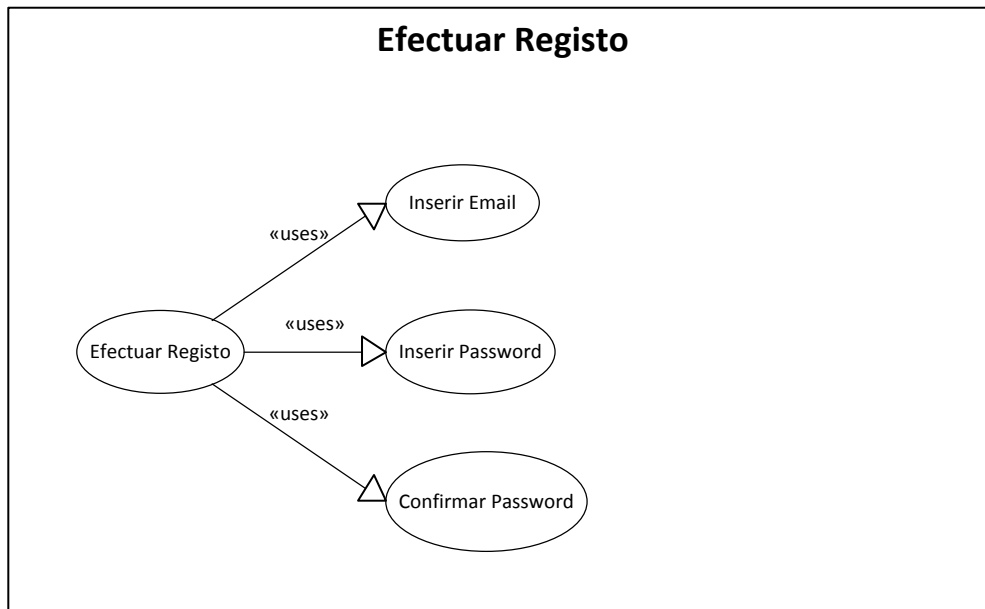


Figura 94 - DCU Efetuar Registo

Descrição:

Ao executar a aplicação, o utilizador começará por efetuar *login* na aplicação para aceder ao sistema. Caso o utilizador não esteja registado, existirá um campo no menu de *login* que vai permitir aceder ao menu de registo do sistema.

Este menu de sistema tem de conter três campos: um para o utilizador inserir o seu endereço de email e os restantes para inserir a sua palavra-chave e respectiva confirmação. Nestes dois últimos campos a informação não poderá ser apresentada ao utilizador em *cleartext*.

Se a operação de registo correr sem problemas, o utilizador será notificado e será reencaminhado directamente para o menu de login.

Prioridade: MUST

Complexidade: Média-Baixa

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade.

Pós-condições:

- **Sucesso:**
 - O utilizador efectua o registo na aplicação e é redireccionado para o menu de *login* da aplicação.
- **Falha:**
 - O utilizador é incapaz de prosseguir devido a uma falha do sistema, consequentemente, é apresentada uma mensagem de erro apropriada.

Ação:

- O utilizador pretende entrar no sistema com o objectivo de aceder a informações disponíveis na aplicação, sendo esse acesso feito através da sua autenticação no sistema. Para efectuar essa autenticação é necessário o utilizador que o utilizador esteja registado no sistema, caso contrário, as credenciais de login não serão validadas pelo sistema.

Fluxo básico de eventos:

- O utilizador executa a aplicação;
- O sistema mostra o campo para inserir nome de utilizador e palavra-chave;
- Caso o utilizador não esteja registado, este tem a opção de fazer o registo na aplicação, seleccionando a opção de registo;
- O utilizador insere os dados de registo no sistema e submete-os;
- O sistema valida os dados de registo e o utilizador é reencaminhado para o menu de login da aplicação.

Fluxo alternativo:

- ***Falha de registo:***
 - O sistema rejeita o registo do utilizador.
 - O sistema mostra uma mensagem de erro de registo e mostra novamente o formulário de registo.

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Baixa Frequência. Sempre que o utilizador iniciar pela primeira vez a aplicação.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- ***Performance:*** Nenhuma.

- **Interface:** A interface vai ter de ser um sistema simples e intuitivo onde o utilizador proceda ao seu registo sem grandes complicações.
- **Segurança:** A palavra-chave inserida não deve ser exibida em *cleartext*.

Problemas:

- Nenhum.

Cenários de utilização:

Um utilizador tenta aceder à aplicação *Youth5G* com o intuito de visualizar um determinado POI com o seu respectivo evento. Para isso é necessário proceder ao login na aplicação e caso o utilizador não esteja registado é necessário efectuar o registo de maneira a que os dados de *login* sejam validados pelo servidor.

I.2.1.2 ID 1.2 Seleccionar POI

Esta subsecção especifica o requisito de seleccionar POI.

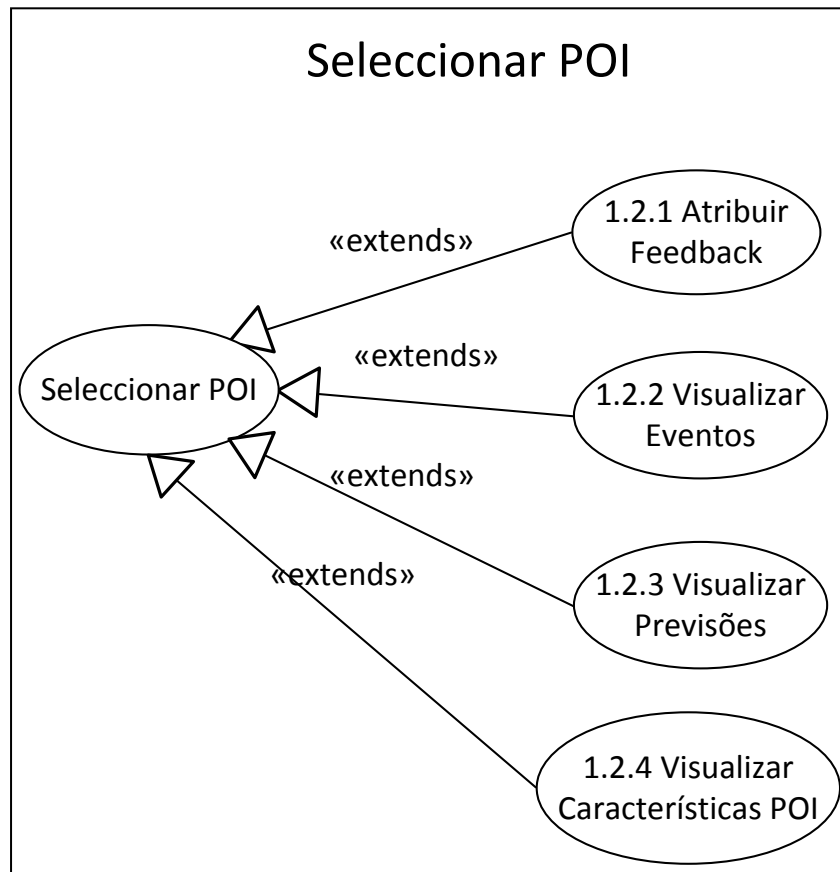


Figura 95 - DCU Seleccionar POI

ID 1.2.1 Atribuir Feedback

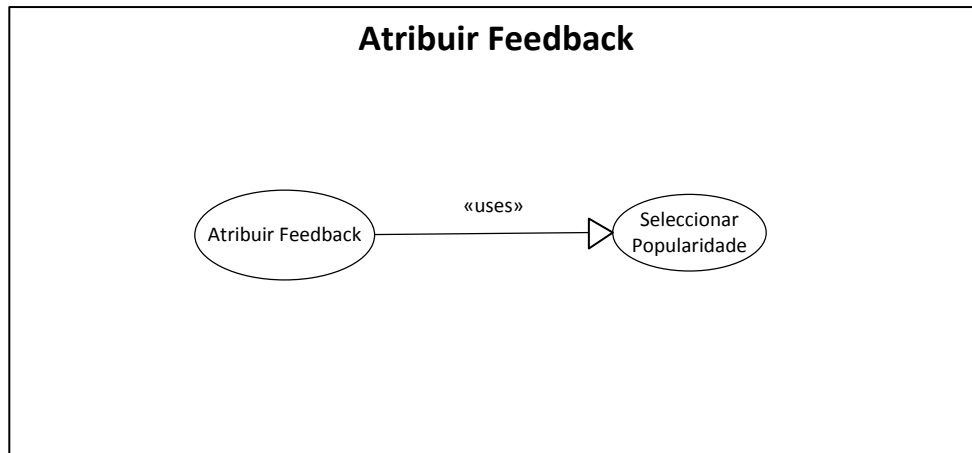


Figura 96 - DCU Atribuir Feedback

Descrição:

Quando a opção de atribuir *feedback* for seleccionada serão apresentadas as cinco opções de popularidade do local. Estas possuem valores de 1 (Baixo) a 5 (Alto), através de *radio buttons*.

Ao seleccionar uma das opções, o *feedback* é enviado para o sistema central e após a confirmação a aplicação informa o utilizador se o *feedback* foi enviado ou actualizado.

Após a submissão de uma opção, o sistema recalcula o *ranking* do utilizador.

Prioridade: *MUST*

Complexidade: *Média*

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O sistema guarda a opção seleccionada pelo utilizador.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que a opção não foi submetida.

Acção:

- O utilizador pretende enviar a popularidade do POI seleccionado anteriormente.

Fluxo básico de eventos:

- O utilizador selecciona o POI;
- O utilizador escolhe a opção de atribuir *Feedback*;
- O utilizador selecciona uma das 5 opções disponíveis.

Fluxo alternativo:

- **Actualização do Feedback:**
 - O utilizador pretende alterar o feedback fornecido anteriormente;
 - O utilizador altera a sua selecção;

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Média-Alta: Dependentemente da participação que cada utilizador tiver.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;
- **Interface:** A interface será a vista simples com um conjunto de “radio buttons” referentes à escala da avaliação;
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas: *Nenhum.***Cenários de utilização:**

O utilizador pretende atribuir um feedback sobre o POI que represente a popularidade do mesmo no presente dia. Este *feedback* é uma estimativa da densidade populacional do POI. O utilizador pode escolher entre cinco valores variando entre 1 (baixa densidade) e 5 (alta densidade).

ID 1.2.2 Visualizar Eventos

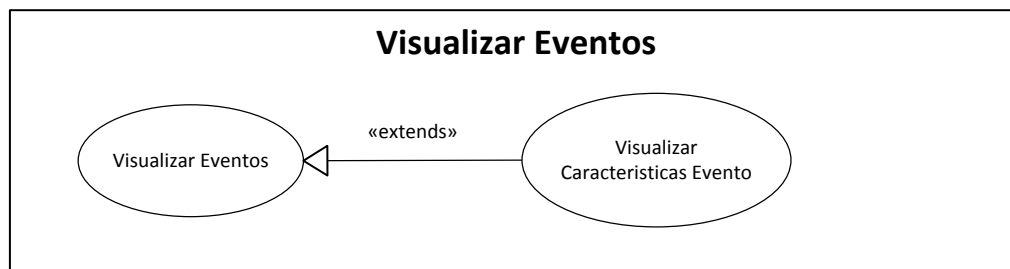


Figura 97 - DCU Visualizar Evento

Descrição:

Com esta opção o utilizador tem a possibilidade de visualizar os eventos futuros de um determinado POI e suas características.

Os eventos são apresentados através de uma lista de itens. Cada item contém o nome do evento no lado esquerdo, e o dia em que é realizado no lado direito. Caso o evento seja realizado no mês actual, é indicado apenas o dia. Caso o evento seja realizado num mês futuro, é indicado o dia e o mês.

O número de eventos a apresentar na lista não se encontra definido. Pode ser um número máximo estático e.g., 20 eventos.

O utilizador pode seleccionar um evento da lista. Com essa *selecção é apresentado uma vista contendo as características do evento seleccionado.*

Prioridade: MUST

Complexidade: Média

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador visualiza a lista de eventos disponíveis e opcionalmente, visualiza as características de um dos eventos.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que não é possível apresentar a informação desejada.

Acção:

- O utilizador pretende visualizar os eventos que vão ocorrer num determinado POI, assim como, os detalhes de cada evento.

Fluxo básico de eventos:

- O utilizador selecciona o POI;
- O utilizador selecciona a opção visualizar eventos;
- O utilizador escolhe um evento da lista apresentada;
- Observa as suas características.

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema é incapaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Média-Alta.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;
- **Interface:** A interface permitirá ao utilizador ver os diferentes eventos através de uma lista.
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

O utilizador visualiza a listagem de eventos de um determinado POI seleccionado anteriormente. A selecção de um evento apresenta uma vista contendo as características de um evento.

ID 1.2.3 Visualizar Previsões

Descrição:

Esta funcionalidade irá permitir ao utilizador, visualizar a afluência nos próximos dias a determinados locais, através do histórico de popularidade que se encontra no servidor.

Para isso, é necessário o utilizador seleccionar o POI em questão e dirigir-se à opção de previsão.

Como referido anteriormente a previsão é baseada no histórico de popularidade de um determinado POI, usando para o cálculo os dados das últimas 4 semanas. Este cálculo será efectuado do lado do servidor através de uma média aritmética ponderada, onde as semanas mais recentes tem um peso superior às semanas mais antigas.

Esta informação irá ser disponibilizada ao utilizador através de uma lista com diferentes cores, onde cada cor indica o grau de afluência a um determinado evento.

Prioridade: MUST

Complexidade: Média

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador consegue visualizar a previsão da afluência do local para os próximos dias.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que não é possível apresentar a informação desejada.

Acção:

- O utilizador pretende visualizar os dias em que o local costuma ter uma maior afluência;

Fluxo básico de eventos:

- O utilizador selecciona o POI;
- O utilizador selecciona a opção visualizar previsões;
- O utilizador visualiza as diferentes cores que indicam a afluência ao local;

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma exceção.

Frequência:

- Média.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;
- **Interface:** A interface permitirá ao utilizador ver a afluência ao local através de uma lista.
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

Por exemplo, um utilizador gostaria de num fim-de-semana, frequentar um bar bastante calmo. Assim, através da aplicação poderia visualizar os locais perto de onde se encontrava com menor frequência.

ID 1.2.4 Visualizar Características POI

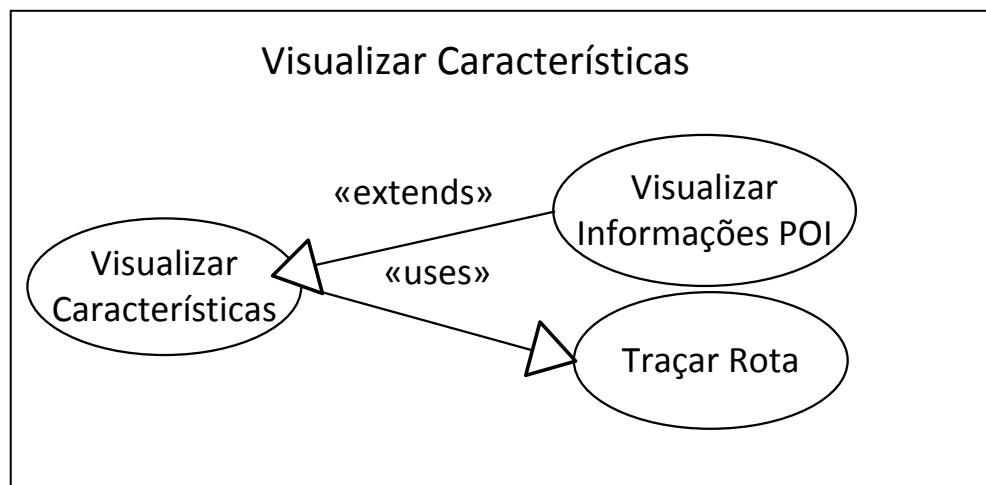


Figura 98 - DCU Visualizar Características

Descrição:

Esta funcionalidade permitirá ao utilizador visualizar informações diversificadas sobre o *POI* seleccionado e activar a navegação no mapa até ao ponto de interesse.

As informações fornecidas por esta funcionalidade serão uma fotografia sobre o local, o nome do local, a morada, o tipo de local, as coordenadas *GPS* e uma pequena descrição que será fornecida pelos utilizadores do *foursquare*.

Quanto à opção que permite a navegação do utilizador até ao *POI*, esta será accionada através de um botão que se encontra na parte inferior do ecrã. O trajecto irá ser fornecido por XML directamente da API do Google.

Prioridade: MUST

Complexidade: Alta

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador visualiza informações sobre o *POI* e consegue traçar um trajecto desde o ponto em que se encontra até ao local do *POI*.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que não é possível traçar o trajecto desejado.

Acção:

- O utilizador pretende saber algumas informações sobre o *POI* e o caminho para ir desde a posição onde está até ao local do *POI*.

Fluxo básico de eventos:

- O utilizador selecciona o *POI*;
- O utilizador selecciona a opção visualizar características;
- O utilizador visualiza a fotografia, morada, coordenadas *GPS*, etc ;
- O utilizador pede à aplicação para traçar o trajecto desde a sua posição actual até ao *POI*;

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma exceção.

Frequência:

- Média.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;
- **Interface:** A interface permitirá ao utilizador ver as diferentes informações do POI em forma de texto e através de uma fotografia;
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

O utilizador quer divertir-se e, para isso, gostava de ir para um bar onde esteja uma elevada afluência. Assim, observa no mapa um POI e vê suas características. Depois de ver as suas características e observar alguns comentários de outros utilizadores, decide que é o local que quer frequentar e para isso pede à aplicação para traçar a rota desde o local onde se encontra até ao local do POI.

I.2.1.3 ID 1.3 Visualizar Menu Opções

Esta subsecção especifica o requisito de visualizar menu opções.

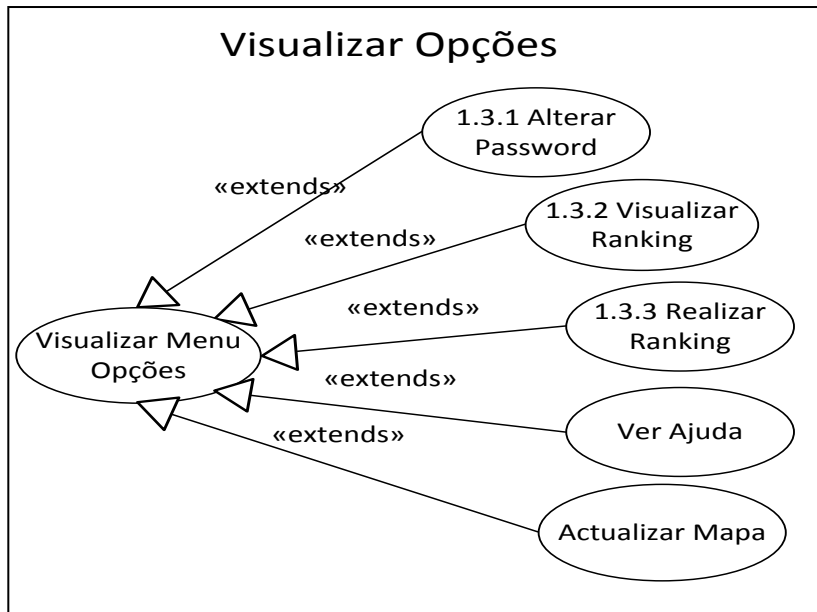


Figura 99 - DCU Visualizar Menu Opções

ID 1.3.1 Alterar Password

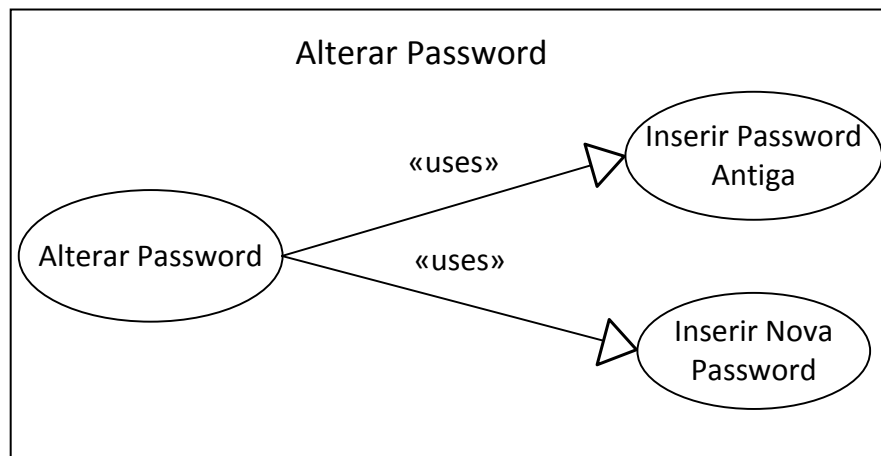


Figura 100 - DCU Alterar Password

Descrição:

Esta opção permite ao utilizador mudar a sua *password* quando deseja. Para isso, o utilizador inicia a aplicação e dirige-se à funcionalidade alterar a *password*, presente no menu de opções. Nesta funcionalidade o utilizador terá três caixas de texto, uma para a *password* antiga, uma para inserir a *password* nova e outra para confirmar a nova *password*.

Prioridade: SHOULD

Complexidade: Baixa

Actores e funções:

- **Actor principal:** Utilizador;

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador insere a *password* antiga e a nova recebendo uma notificação a dizer que a password foi alterada com sucesso.
- **Falha:**
 - O sistema não consegue alterar a *password* e a aplicação lança uma notificação que informa o utilizador a dizer que a password não foi alterada.

Acção:

- O utilizador deseja mudar a *password*;

Fluxo básico de eventos:

- O utilizador vai ao menu de opções;
- O utilizador selecciona a opção alterar *password*;
- O utilizador insere a *password* antiga e a nova *password*;
- A aplicação informa o utilizador do sucesso da operação;

Fluxo alternativo:

- **Falha na verificação da *password*:**
 - Se o utilizador não inserir correctamente a segunda nova *password*, o sistema lança um aviso e pede ao utilizador para voltar a inserir novamente.

Fluxo de erro:

- O sistema não realiza qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Baixa.

Assunções:

- Assume-se que o utilizador tem uma ligação á rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;
- **Interface:** A interface terá três caixas de texto que permitem ao utilizador alterar a password, estas caixas não devem mostrar a password;
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

O utilizador por algum motivo deseja alterar a sua *password*, para isso dirige-se a esta funcionalidade e altera a sua *password*.

ID 1.3.2 Visualizar Ranking

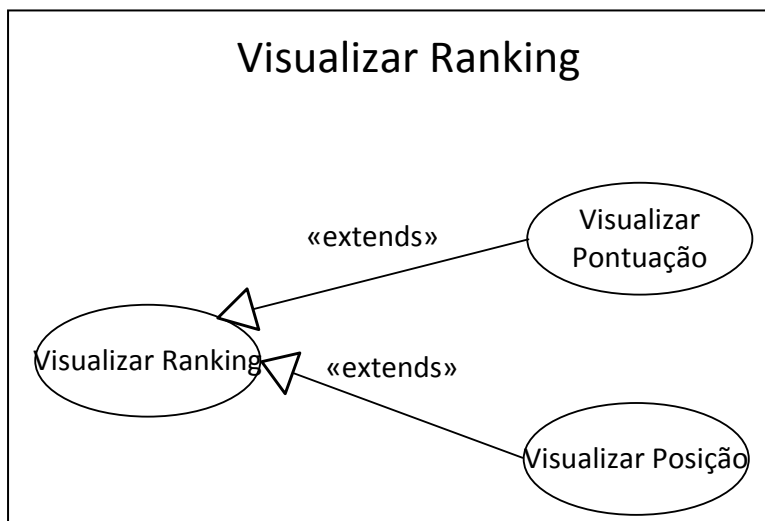


Figura 101 - DCU Visualizar Ranking

Descrição:

Como esta aplicação é baseada em dados colaborativos, onde os utilizadores podem colaborar enviando informações sobre a afluência aos diferentes locais, o *ranking* é um sistema de incentivo criado para que os utilizadores estejam constantemente a participar.

Deste modo esta funcionalidade permitirá ao utilizador ver a sua posição no *ranking* bem como a de outros utilizadores.

A visualização da informação será feita através de uma lista onde aparecerão todos os utilizadores e através de uma caixa de texto onde aparece a pontuação do utilizador.

Os utilizadores deverão todos iniciar com a pontuação a zero.

Prioridade: MUST

Complexidade: Média

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador consegue visualizar a sua pontuação e a pontuação dos outros utilizadores.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que não é possível apresentar a sua pontuação e a lista com as restantes pontuações.

Acção:

- O utilizador pretende verificar qual a sua posição no ranking e pontuação, depois de enviar alguns feedbacks sobre a afluência de um determinado POI;

Fluxo básico de eventos:

- O utilizador selecciona as opções;
- O utilizador selecciona a opção visualizar *ranking*;
- O utilizador visualiza a sua posição no ranking, assim como, a sua pontuação.

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Média.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;
- **Interface:** A interface permitirá ver o *ranking* do utilizador através de uma lista;
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password* para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

O utilizador encontra-se num evento nocturno e envia alguns *feedbacks* sobre a afluência naquele local. Depois de enviar esses *feedbacks* decide ver qual a sua pontuação.

ID 1.3.3 Realizar Inquérito

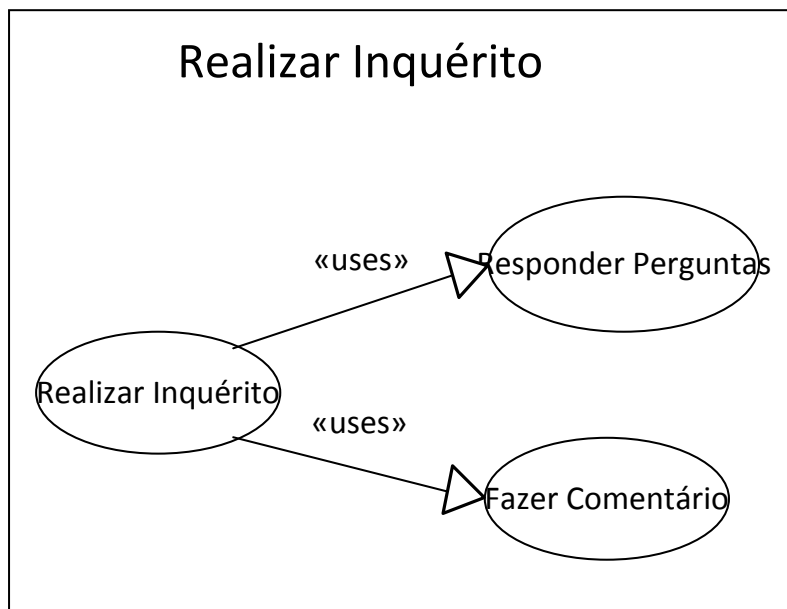


Figura 102 - DCU Realizar Inquérito

Descrição:

Como o sucesso do sistema depende em grande parte da sua aceitação pelo utilizador, será criada uma funcionalidade que permite realizar inquéritos na aplicação para recolher a opinião de cada utilizador.

Esta funcionalidade será constituída por um grupo de perguntas rápidas e um grupo de sugestões. O primeiro grupo terá cerca de nove questões que são apresentadas na aplicação em forma de texto, onde o utilizador responde através de uma escala entre 1 (mau) e 5 (muito bom). Adicionalmente, o último grupo será apresentada uma caixa de texto ao utilizador para escrever algumas sugestões de melhoramentos à aplicação.

Esta informação será enviada da aplicação ubíqua para o servidor, onde será analisada posteriormente pelo Administrador.

Prioridade: SHOULD

Complexidade: Média

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador realiza o inquérito e o inquérito é submetido com sucesso.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que não é possível submeter o seu inquérito.

Acção:

- O utilizador pretende enviar uma sugestão de melhoramento da interface;

Fluxo básico de eventos:

- O utilizador acede ao menu de opções;
- O utilizador selecciona a opção realizar inquérito;
- O utilizador responde aos dois grupos do inquérito;
- O utilizador submete o inquérito.

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema não é capaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma excepção.

Frequência:

- Média.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** Nenhum;

- **Interface:** A interface permitirá ao utilizador responder às perguntas através de uma escala entre 1 (mau) e 5 (muito bom), representado por estrelas.
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

O utilizador ao navegar na aplicação encontra um erro ou encontra um aspecto que desejava ver melhorado. Deste modo, este vai ao menu de opções e responde ao inquérito indicando os pontos que desejava que fossem melhorados.

I.2.1.4 Navegar Mapa

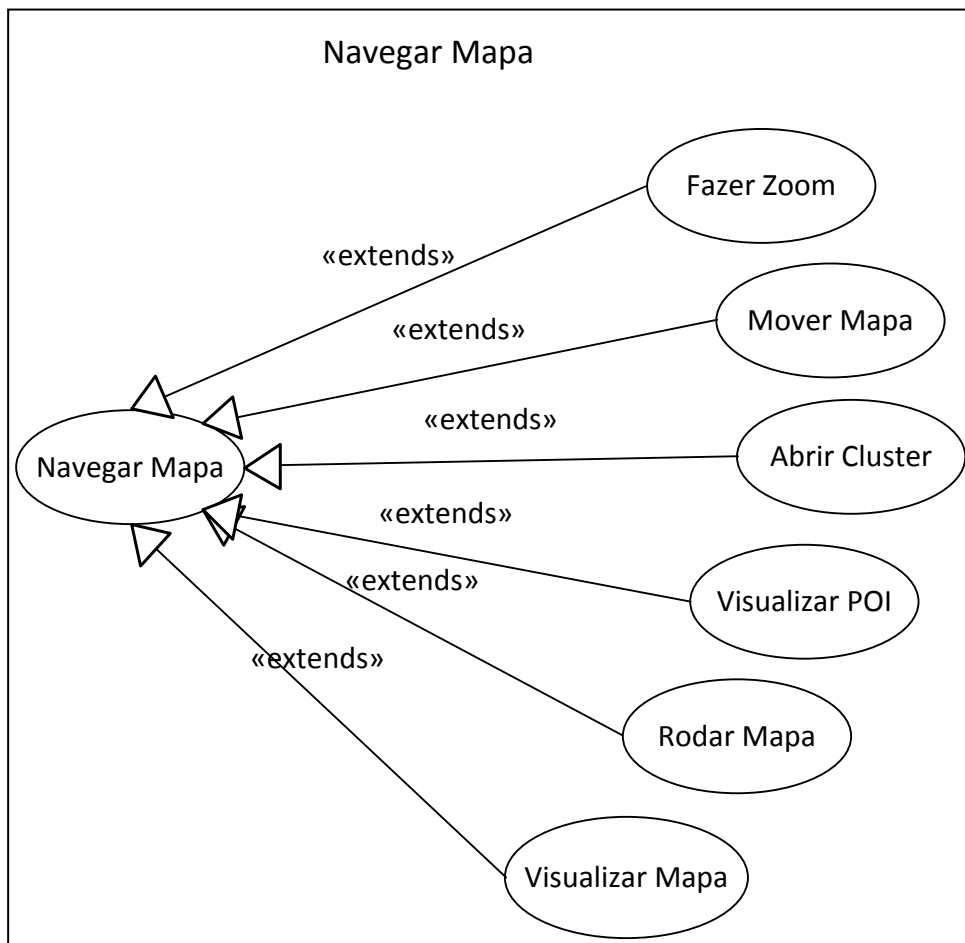


Figura 103 - DCU Navegar Mapa

Descrição:

Como foi dito anteriormente a aplicação ubíqua terá um mapa onde o utilizador poderá visualizar os diferentes POIs. Assim, através deste mapa o utilizador poderá ver os POI's que se encontram na sua zona e em outras zonas, recorrendo às funcionalidades de fazer *zoom*, mover o mapa, etc. O mapa será disponibilizado pela API do *Google Maps*.

Uma vez que existem diferentes níveis de afluência em cada evento de cada local, o mapa representa os POIs por diferentes cores. Estes POIs, ao serem desenhados, irão ter uma cor associada que discrimina em tempo real o tipo de popularidade que um determinado POI tem:

- Preto – não atribuído;
- Verde – Baixo;
- Amarelo – Médio;
- Vermelho –Alto.

Como existem situações em que existe um número elevado de *POIs* numa zona, a aplicação deverá recorrer a um mecanismo de *clustering* para reduzir o número de *POIs* em um *cluster*. Cada *cluster* deverá ter uma cor que represente a média de afluência de todos os *POIs* que contém, bem como, o número de *POIs* que se encontram naquele lugar.

Quando o utilizador carregar em um *POI* o mapa deve aumentar o *zoom* e ser centrado pela posição do *cluster*.

Prioridade: MUST

Complexidade: Muito Alta

Actores e funções:

- **Actor principal:** Utilizador

Pré-condições:

- O utilizador tem de estabelecer uma conexão à Internet, a fim de conseguir aceder a esta funcionalidade;
- O utilizador tem que ter feito *login*.

Pós-condições:

- **Sucesso:**
 - O utilizador consegue navegar pelo mapa e ver os POIs de cada local representado por *clusters*.
- **Falha:**
 - O sistema apresenta uma mensagem de erro informando o utilizador que não conseguiu encontrar nenhum POI para aquela localização.

Acção:

- O utilizador navega pelo mapa descobrindo locais novos que não conhece;

Fluxo básico de eventos:

- O utilizador faz a autenticação;

- O mapa é carregado;
- Os *POIs* são carregados;
- O utilizador navega pelo mapa.

Fluxo alternativo:

- Nenhum.

Fluxo de erro:

- O sistema é incapaz de completar qualquer uma das etapas atribuídas no fluxo básico ou alternativo, devido ao mau funcionamento e é mostrada uma exceção.

Frequência:

- Alta.

Assunções:

- Assume-se que o utilizador tem uma ligação à rede externa (internet) para proceder à execução da aplicação.

Requisitos Especiais:

- **Performance:** O mapa não deve carregar todos os elementos em um tempo aceitável;
- **Interface:** A interface tem que permanecer limpa, de modo a que o utilizador consiga visualizar os diferentes POIs sem se perder. As cores devem ser o mais suaves possível para a representação no mapa.
- **Segurança:** A troca de informação entre a aplicação e o servidor deve ser feita com o *username* e *password*, para assegurar a autenticidade dos dados.

Problemas:

- Nenhum.

Cenários de utilização:

O utilizador encontra-se numa localidade que não conhece e gostaria de ir mais os seus amigos divertir-se um pouco. Para isso inicia a aplicação no seu *smartphone* e visualiza os locais que têm uma maior afluência.