

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



**FEUP**

# **Um Exemplo de Computação Ubíqua em Serviços de Saúde Orientados ao Utente**

**Jailson Moreira**

VERSÃO PROVISÓRIA

Mestrado Integrado Engenharia Electrotécnica de Computadores

Orientador: Rosaldo J. F. Rossetti ( Dr.)

Julho de 2011



# Resumo

O Sistema de Saúde está a enfrentar uma mudança necessária, cuja renovação estimula o desenvolvimento de capacidades para conseguir a agilidade, interoperabilidade e eficiência imprescindíveis para melhorar a qualidade e a produtividade, reduzindo os custos associados. O modelo de atendimento centrado no indivíduo é para onde o sistema de saúde actual converge, conduzindo assim para um processo de atendimento distribuído à saúde .

Actualmente a humanidade enfrenta várias dificuldades inerentes a uma sociedade cada vez mais envelhecida, com uma maior predominância de doenças crónicas e dificuldades associadas como a mobilidade. Por exemplo uma das grandes epidemias que afectou e continua a afectar este planeta é a Diabete. De acordo com a Organização Mundial da Saúde, em 2006, havia cerca de 171 milhões de pessoas doentes da diabetes, e esse índice aumenta rapidamente. É estimado que em 2030 esse número dobre. A Diabetes Mellitus ocorre em todo o mundo, mas é mais comum (especialmente a tipo II) nos países mais desenvolvidos. O maior aumento actualmente é esperado na Ásia e na África, onde a maioria dos diabéticos será encontrada em 2030. O aumento do índice de diabetes em países em desenvolvimento segue a tendência de urbanização e mudança de estilos de vida.

Neste contexto, o avanço acelerado da tecnologia trouxe novos paradigmas para a área de computação e com eles vários benefícios para a sociedade. O paradigma da computação ubíqua traz inovações para aplicar a computação no dia-a-dia das pessoas sem que possa ser percebida. Para isso, utiliza-se da junção de diversas tecnologias já existentes como, por exemplo, a tecnologia móvel e sensores. Vários benefícios atingem a área médica, trazendo métodos novos e é nesta perspectiva que foi desenvolvido um protótipo de um sistema de monitorização de pacientes de uma forma discreta, com o objectivo de melhorar, rentabilizar e controlar os dados biométricos dos pacientes que sofrem de doenças como a diabetes e tem essa necessidade.

O sistema é composto por duas aplicações, uma móvel e outra Web, em que a móvel integra tecnologias como sistema operativo Android que é uma ferramenta importante para desenvolvimento de sistemas dessa natureza. A aplicação Web é fundamental para os profissionais de saúde pois é nela que acompanharão os dados biométricos enviados pelos pacientes. Os pacientes utilizam as duas aplicações para enviar os dados biométricos para serem guardados na base de dados e serem visualizados pelos profissionais de saúde e pelo próprio paciente.



# Abstract

The health system is facing an inevitable yet necessary change, the renewal of which stimulates the development of capabilities to achieve the agility, interoperability and efficiency that are essential to improving quality and productivity thus reducing associated costs. The patient-centred care model is where the current health system converges to, thus leading to a process of distributed health care services.

Nowadays, humankind faces several difficulties inherent to an increasingly aging society, with a predominance of chronic diseases and other problems associated with mobility. For instance, one of the greatest epidemics that has affected and still affects this planet is diabetes. According to the World Health Organization, in 2006 there were about 171 million people suffering from diabetes, and this value increases rapidly. It is estimated that by 2030 this value will double. Diabetes Mellitus occurs worldwide but is more common (especially the type II) in more developed countries. The largest increase is expected today in Asia and Africa, where most diabetics will be seen by 2030. The increased rate of diabetes in developing countries follows the trend of urbanization and changing lifestyles.

In this context, the rapid advancement in technology has brought new paradigms to the field of computing, as well as many benefits to the society. The paradigm of ubiquitous computing brings innovations to be applied to people's computing needs in their daily lives without it even being perceived. To achieve this, several existing technologies are combined such as, for instance, mobile technology and sensors networks. Several of these benefits then reach the medical field, bringing new methods to it, and it is in this perspective that a prototype of a system was developed for monitoring patients in a seamless way so as to improve, make it more cost effective and control the biometric data of patients that suffer from chronic diseases such as diabetes and are in such a need.

The system consists of two applications, one of which is mobile and the other Web-based, in which the former integrates the Android operating system which is a key tool for the development of systems of this sort. The Web-based application is critical to practitioners in the health care field since it is through it they will follow the biometric data uploaded by their patients. Patients use both applications to send biometric data to be stored in the database, as well as to be visualised by health professionals and by the patients themselves.



# Agradecimentos

Aproveito este espaço para agradecer a todas as pessoas que contribuíram, de alguma forma, para que a realização desta dissertação.

Agradeço ao Dr. Rosaldo J. F. Rossetti pela colaboração, e conselhos dados durante a realização deste trabalho.

Um especial agradecimento à minha família, que apesar da distância, me concedeu o seu apoio incondicional ao longo destes anos.

Uma palavra especial para a minha namorada, Maria Furtado, que esteve sempre presente ao longo da realização desta dissertação.

Por último, mas não menos importante, agradeço a todos aqueles que não foram mencionados, mas que de alguma maneira contribuíram para a elaboração deste trabalho.

Jailson Moreira

Julho, 2011





# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objectivos	1
1.2	Estrutura do Documento	2
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>5</b>
2.1	Computação Úbiqua	5
2.1.1	Características	5
2.1.2	Objectivos e relação com os utilizadores	6
2.1.3	Computação ubíqua vs computação móvel vs computação pervasiva	6
2.1.4	Desafios no desenvolvimento de sistemas ubíquos	7
2.1.5	Tecnologias de Comunicação	7
2.2	Computação Orientada a Serviços	9
2.2.1	Elementos da Computação Orientada a Serviços	10
2.2.2	Arquitectura Orientado a Serviço (SOA)	11
2.3	Computação Orientado a Agentes	13
2.3.1	Agente e Sistema Multiagente	13
2.4	Exemplos de Aplicações	14
2.4.1	Sistemas na área de Computação Ubíqua	14
2.4.2	Sistemas na área de Computação Orientado a Serviços	14
2.4.3	Sistemas na área de Computação Orientado a Agentes	15
2.4.4	Sistemas na área da Saúde	15
2.5	Doenças Crónicas: Diabetes e Obesidades	17
2.5.1	Breves Considerações	17
2.5.2	Obesidade	18
2.5.3	Hipercolesterolemia	19
2.5.4	Glicose	20
2.5.5	Hipertensão Arterial	20
2.5.6	Temperatura Corporal	21
2.6	Sumário	22
<b>3</b>	<b>Análise e Especificação de Requisitos</b>	<b>23</b>
3.1	Perspectiva geral do sistema	23
3.2	Características dos Utilizadores	24
3.3	Especificação detalhada dos Requisitos	24
3.4	Restrições	26
3.5	Casos de Utilização	26
3.6	Ferramentas de Desenvolvimento	28
3.6.1	Sistema Operativo Android	28

3.6.2	A Plataforma Eclipse . . . . .	31
3.6.3	Base de Dados . . . . .	33
3.7	Sumário . . . . .	39
<b>4</b>	<b>Implementação e Avaliação de Resultados</b>	<b>41</b>
4.1	Implementação da Base de Dados e Aplicação Web . . . . .	41
4.1.1	Ferramentas Utilizada . . . . .	41
4.1.2	Descrição de Permissões . . . . .	42
4.1.3	Modelos de Entidade e Associação . . . . .	43
4.1.4	Modelos Relacional . . . . .	44
4.1.5	Descrição dos Módulos do Sistema . . . . .	45
4.1.6	Definição de Vistas . . . . .	49
4.1.7	Decrição de páginas . . . . .	50
4.2	Implementação da Aplicação Móvel . . . . .	51
4.2.1	Visão geral . . . . .	52
4.2.2	Protocolo Comunicação TCP/IP . . . . .	53
4.2.3	Formato das Tramas e seus tratamentos . . . . .	53
4.2.4	Arquitectura Física . . . . .	55
4.2.5	Arquitectura Lógica . . . . .	56
4.3	Análise dos Resultados Obtidos . . . . .	57
4.3.1	Resultados da Base de Dados . . . . .	57
4.3.2	Resultados da aplicação Web . . . . .	59
4.3.3	Resultados e Análise da aplicação Móvel . . . . .	63
4.4	Sumário . . . . .	66
<b>5</b>	<b>Conclusões</b>	<b>67</b>
5.1	Síntese do Trabalho Desenvolvido . . . . .	67
5.2	Trabalhos Futuros . . . . .	68
<b>A</b>	<b>Interfaces e Procedimentos de Instalação</b>	<b>69</b>
A.1	Procedimentos da instalação do SDK . . . . .	69
A.2	Interfaces Adicionais . . . . .	70
	<b>Referências</b>	<b>75</b>

# Lista de Figuras

2.1	Relação entre a Computação Ubíqua, Pervasiva e Móvel [2]	6
2.2	Uma visão conceitual de como os elementos de SOC se inter-relacionam [8]	10
2.3	Medidor de IMC	18
2.4	Gráfico sobre o IMC [31]	19
2.5	Medidor de Colesterol e Glicose	20
2.6	Medidor de Pressão Arterial	21
2.7	Medidor de Temperatura: Termómetro	22
3.1	Arquitectura Geral do Sistema	24
3.2	Exemplo de registo em bloco de notas	25
3.3	Casos de utilização 1	27
3.4	Casos de utilização 2	27
3.5	Casos de utilização 3	27
3.6	Arquitectura do Android [36]	30
3.7	Arquitectura do Eclipse [37]	31
3.8	Representação da Plataforma em Camadas [38]	32
3.9	Cenário de Utilização de Plug-ins	33
4.1	Diagrama do Modelo Entidade e Associação	43
4.2	Diagrama de Arquitectura do Módulo Login	45
4.3	Diagrama de Arquitectura do Módulo Registo	46
4.4	Diagrama de Arquitectura do Módulo Perfil	47
4.5	Diagrama de Arquitectura do Módulo Ficha Clínica	48
4.6	Diagrama de Arquitectura do Módulo Medições	49
4.7	Arquitectura da Aplicação Móvel	52
4.8	Diagrama de distribuição de componentes	55
4.9	Diagrama de dependências dos pacotes de bibliotecas usados	56
4.10	Tabela pessoa da base de dados	57
4.11	Tabela ficha clínica da base de dados	57
4.12	Tabela medições da base de dados	58
4.13	Tabela da vista lista de pacientes da base de dados	58
4.14	Interface Inicial	59
4.15	Interface para Escolher Tipo de utilizador	59
4.16	Interface de Registo para Médico e Público	60
4.17	Interface de Registo para Paciente	60
4.18	Interface Principal do Médico	61
4.19	Interface Principal do Paciente	61
4.20	Interface histórico medições	62

4.21	Interface de Alertas . . . . .	62
4.22	Interface Inicial . . . . .	63
4.23	Interface erro login . . . . .	63
4.24	GUI Principal Paciente . . . . .	64
4.25	GUI Principal Médico . . . . .	64
4.26	Visualização do erro no envio de medições . . . . .	65
4.27	GUI de Confirmação dos valores a enviar . . . . .	65
4.28	GUI histórico medições Pressão Arterial . . . . .	65
4.29	GUI Alertas Colesterol . . . . .	65
4.30	GUI Alertas IMC . . . . .	65
4.31	GUI Alertas Temperatural . . . . .	65
4.32	GUI Lista Paciente . . . . .	66
4.33	GUI Lista Paciente . . . . .	66
A.1	Interface perfil do Médico . . . . .	70
A.2	Interface da ficha clinica de um paciente vista pelo medico Médico . . . . .	70
A.3	Interface de alerta vista pelo médico . . . . .	71
A.4	Interface das listas de ficha clinica de um médico . . . . .	71
A.5	Interface da média das medições . . . . .	72
A.6	Interface do historico medições do imc . . . . .	72
A.7	Interface de confirmação de envio . . . . .	72
A.8	Interface de envio de medições . . . . .	72
A.9	Interface da ficha clinica de um paciente . . . . .	73
A.10	Interface perfil do paciente . . . . .	73
A.11	Interface principal do publico . . . . .	73
A.12	Interface de alerta pressão arterial . . . . .	73

# Lista de Tabelas

2.1	Classificação de IMC [31]	19
2.2	Classificação das Sociedade Europeia de Hipertensão e de Cardiologia [34]	21
2.3	Tabela de Temperatura [33]	22
3.1	tabela de versões de Android [35]	29
3.2	comparação entre PostgreSQL e MySQL	35
4.1	Tabela de Associações	44
4.2	Modelo Relacional	44
4.3	Tabela Descritiva do Módulo LOGIN	46
4.4	Tabela Descritiva do Módulo Registo	47
4.5	Tabela Descritiva do Módulo perfil	47
4.6	Tabela Descritiva do Módulo Ficha Clínica	48
4.7	Tabela Descritiva do Módulo Medições	49
4.8	Trama enviada pelo cliente	53
4.9	Trama enviada pelo servidor	53
4.10	Trama enviada pelo cliente	54
4.11	Trama enviada pelo servidor	54
4.12	Trama enviada pelo cliente	54



# Siglas e Acrónimos

ADT	Android Development Tools
AVD	Android Virtual Devices
API	Application Programming Interface
CCA	Context Collecting Agent
CDMA EV-DO	Code Division Multiple Access Evolution-Data Optimized
CRA	Context Reasoning Agent
CORBA	Common Object Request Broker Architecture
EDGE	Enhanced Data rates for GSM Evolution
GSM	Global System for Mobile
IDEN	Integrated Digital Enhanced Network
IPA	Information Processing Agent
JADE	Java Agent Development Framework
FHSS	Frequency Hopping spread Spectrum Signaling
HDL	High Density Lipoproteins
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JDK	Java SE Development Kit
JRE	Java Runtime Environment
LDL	Low Density Lipoproteins
PANs	Wireless personal area networks
RSSF	Redes de Sensores Sem Fios
SGBDOR	Sistema Gestor de Banco de Dados Objecto Relacional
SMA	Sistemas Multi Agentes
SMC	Self-Managed Cell
SOA	Computação Orientada a Agentes
SOAP	SIMPLE OBJECT ACCESS PROTOCOL
SOC	Computação Orientada a Serviços
UMTS	Universal Mobile Telecommunications System
XML	Extensible Markup Language
WASP	Web Architectures for Service Platforms
WI-FI	<i>Wireless Fidelity</i>





# Capítulo 1

## Introdução

O Sistema de Saúde está a enfrentar uma mudança necessária, cuja renovação estimula o desenvolvimento de capacidades para conseguir a agilidade, interoperabilidade e eficiência, imprescindíveis para melhorar a qualidade e a produtividade, reduzindo os custos associados. O modelo de atendimento centrado no indivíduo é para onde o sistema de saúde actual converge, conduzindo assim para um processo de atendimento distribuído à saúde. Neste âmbito, a computação ubíqua é submersa no dia-a-dia do paciente através de serviços personalizados de atendimento. Estes serviços, oferecidos num cenário ubíquo, dotados de dispositivos electrónicos que podem prover a colaboração transparente entre os dispositivos do meio (sensores, equipamentos, utilizadores, etc.) a fim de arrecadar e fornecer informações a qualquer utilizador. Os utilizadores podem estar em qualquer lugar e em qualquer hora, que terão acesso a estas informações em qualquer dispositivo, mas não intrusivamente, e de forma invisível, a não ser quando é necessário. A tecnologia de agentes visa ser uma das fundamentais tecnologias usadas na computação ubíqua, para reconhecimento e análise de informação de contexto de ambientes distribuídos. Estes agentes de software transmitem a informação ao pessoal de saúde de uma forma mais rápida e precisa, permitindo uma tomada de decisão mais precisa e reduzindo os custos das transacções. Agentes podem usar e beneficiar das mais recentes tecnologias e modelos abertos, tal como SOA (*service-oriented architecture*) trazendo, assim, a promessa de facilidade de integração com componentes e sistemas legados. O alto custo das instituições de saúde e dos cuidados permanentes a pessoas com limitações, justifica cada vez mais a aparição de novas soluções de monitorização de pacientes que se revelem eficazes e de baixo custo. Com este tipo de monitorização, será possível ajudar pessoas a rentabilizarem o seu tempo e também controlar a sua saúde de um forma menos cansativa e stressante.

### 1.1 Objectivos

O objectivo principal deste trabalho foi conceber um sistema de monitorização de parâmetros vitais, viável e de baixo custo que permita tanto aos pacientes como aos profissionais de saúde me-

lhorarem, rentabilizarem o controlo e análise desses parâmetros enviados pelos pacientes. Também tem como objectivo “ libertar ” o paciente dos vários deslocamentos periódicos aos consultórios médicos a fim de disponibilizar estes mesmos registos. Tendo sempre em conta que contribuirá para que o sistema de saúde torne cada vez mais amigável do ambiente. Este Sistema vai servir de complemento a vários outros que existem mas que fazem monitorização de forma contínua, utilizando redes de sensores para recolha e envio de dados, em que os pacientes estão confinados as suas residências ou em hospitais. Mas especificamente, pretende-se com este projecto cumprir com os seguintes objectivos:

- Implementar duas aplicações: Uma aplicação Web **WebHealthSave** e outra Móvel **Health-SaveMobile** ambos para envio e visualização de dados;
- Criar uma Base de Dados para guardar os dados (parâmetros vitais, dados pessoais e clínicos dos utilizadores);
- Implementar Um Servidor que servirá de ligação entre aplicação móvel e Base de Dados.

## 1.2 Estrutura do Documento

Este relatório está organizado da seguinte forma:

- **Capítulo 2:** faz uma breve apresentação dos conceitos de computação, nomeadamente Computação Ubíqua, Orientada a Serviços e Orientada a Agentes. É também realizado um levantamento de sistemas que empregam um ou vários destes paradigmas na sua concepção, destacando-lhes a arquitectura. Faz um estudo sobre as doenças crónicas, com um nuançe maior sobre diabetes.
- **Capítulo 3:** faz a primeira abordagem ao sistema, ilustrando uma perspectiva geral do mesmo, especificando os objectivos, os respectivos requisitos e as restrições. E ainda faz uma apresentação sobre o sistema operativo Android, a plataforma eclipse, que são o sistema operativo e a plataforma de desenvolvimento utilizados neste projecto e a base de dados. Por fim mostra os diagramas de casos de utilização.
- **Capítulo 4:** são descritos os detalhes da implementação do protótipo, cujos requisitos e os casos de utilização foram especificados no capítulo anterior. A primeira parte da implementação é dedicada à aplicação Web. Na Secção 4.1.1 é apresentado as ferramentas que foram utilizados para a respectiva implementação, na Secção 4.1.2 é feita uma descrição das permissões dos utilizadores do sistema. Nas secções seguintes são apresentados um diagrama do modelo entidade-associação, as respectivas entidades e associações, o modelo relacional e, por último, as respectivas descrições dos módulos. Na segunda parte é feita a descrição da implementação da aplicação móvel. São descritos aspectos da fase de implementação e

instalação, designadamente a estrutura e dependências de código fonte e de módulos executáveis, assim como a sua respectiva instalação nas diferentes plataformas computacionais subjacentes.

- **Capítulo 5:** faz uma síntese do trabalho desenvolvido, sumariando as conclusões. São também apresentadas perspectivas de desenvolvimento futuro.



## Capítulo 2

# Revisão Bibliográfica

Neste Capítulo procura-se abordar toda a componente teórica sobre os sistemas de computação ubíqua, orientada a serviços e agentes, Android e outras soluções actuais utilizadas na monitorização remota de pacientes.

### 2.1 Computação Úbiqua

A Computação Ubíqua vem se tornando realidade, saindo dos laboratórios e se integrando na vida das pessoas sem que essas se apercebam. O termo Computação Ubíqua foi definido pela primeira vez pelo cientista chefe do Centro de Investigação Xerox PARC, Mark Weiser, como: “As tecnologias mais profundas e duradouras são aquelas que desaparecem. Elas dissipam-se nas coisas do dia-a-dia até tornarem-se indistinguíveis” [1].

#### 2.1.1 Características

A computação ubíqua é caracterizada pela presença de dispositivos portáteis, cada vez mais comuns devido aos avanços na fabricação de componentes electrónicos. Esses dispositivos apresentam uma grande capacidade de processamento e armazenamento de dados, recursos para comunicação sem fio, permitindo assim um aumento de utilização de equipamentos portáteis capazes de lidar com os dados multimédia. Esses dispositivos se vulgarizaram como *handhelds*, PDAs, e, agora, têm aparecido os *smart phones* e telemóveis de grande capacidade computacional. Além das funcionalidades originais, como capacidade de comunicação via telemóveis, tais dispositivos também possuem diversas funcionalidades e interfaces como GPS, rádio, TV e câmaras fotográficas digitais, sendo usados em aplicações que envolvem diversas áreas como indústria, medicina, uso pessoal, entre outros [2].

### 2.1.2 Objectivos e relação com os utilizadores

A Computação Ubíqua concebe novas maneiras de pensar acerca dos computadores. Estabelece novos paradigmas, que tenham em conta o mundo real em que vivemos e as tarefas que queremos completar, e que lhes permita integrarem-se de forma transparente. A computação ubíqua tem como finalidade estar em toda parte o tempo todo. Utilizadores do serviço, cujos objectivos e situações se podem alterar dinamicamente, exigem que a computação ubíqua tenha a capacidade para responder duma forma adaptada. Na maioria das vezes é o sistema que conclui por si mesmo as alterações, dando resposta de acordo com as mesmas. No entanto, o utilizador também pode reconfigurar activamente o sistema, de modo a adaptá-lo às novas condições. Assim se cria no sistema a necessidade de não só manter os objectivos do utilizador, e as tarefas e os recursos do sistema ao longo do tempo, mas também as condições ambientais. Simultaneamente, o sistema deverá ser capaz de compreender o raciocínio atrás dessas inferências referente às circunstâncias, permitindo-lhe ir aprendendo das ilações correctas e incorrectas. Para além de ser difícil recolher toda a informação necessária para garantir um comportamento adaptável, é um desafio fazê-lo imperceptivelmente, para além de criar um enorme fardo nos recursos do sistema como um todo [3].

### 2.1.3 Computação ubíqua vs computação móvel vs computação pervasiva

Computação ubíqua é a integração entre a mobilidade com sistemas e presença distribuída, em grande parte imperceptível, inteligente e altamente integrada dos computadores e suas aplicações para o benefício dos utilizadores. A computação ubíqua integra progressos na computação móvel e computação pervasiva. Enquanto a computação móvel centra-se no aumento de mobilidade de serviços entre ambientes, a pervasiva pressupõe a capacidade dos dispositivos computacionais serem embutidos no ambiente, caracterizado por um conjunto de entidades tais como dispositivos, utilizadores e redes, actuando “discretamente” e oferecendo omnipresença aos utilizadores, com o objectivo de integrar a computação com o ambiente físico no qual ela está imersa.



Figura 2.1: Relação entre a Computação Ubíqua, Pervasiva e Móvel [2]

Computação móvel por outro lado é a capacidade de um dispositivo computacional e os serviços associados ao mesmo serem móveis, permitindo este ser carregado ou transportado mantendo-se ligado à rede ou à Internet. Computação pervasiva define que os meios de computação estarão distribuídos no ambiente de trabalho dos utilizadores de forma perceptível ou imperceptível.

#### 2.1.4 Desafios no desenvolvimento de sistemas ubíquos

Abaixo estão listados alguns dos desafios de investigação ainda existentes. Estão divididos em algumas sub-áreas, como sugerido em [2]:

- **Monitorização:** Escolha e inclusão dinâmica dos contextos mais apropriados a cada aplicação; Técnicas para recolha de contextos físicos, lógicos e virtuais; Atribuição de semântica uniforme aos contextos utilizados; Identificação e escolha de fontes de contextos;
- **Modelação:** Modelo de arquitectura para sistemas cientes de contexto; Modelo para representação uniforme da sintaxe dos dados de contexto colectados; Modelo de armazenamento de dados contextuais; Modelo de comunicação adoptado entre diversos utilizadores ou aplicações;
- **Qualidade:** Qualidade de contexto (QoC); Qualidade de serviço (QoS); Qualidade das fontes de contexto; Gestão de aplicações cientes de contexto; Tratamento de falhas; Automação de tarefas; Utilização de algoritmos de aprendizagem; Identificação e tratamento de contextos individuais conflitantes; Identificação e tratamento de contextos colectivos conflitantes;
- **Segurança:** Segurança para troca de dados entre utilizadores e aplicações; Confiabilidade das fontes de contextos; Segurança da Informação de contexto.

#### 2.1.5 Tecnologias de Comunicação

Abaixo são discutidos brevemente algumas das tecnologias de comunicação mais populares, que tornaram possível a implementação de sistemas ubíquos.

##### 2.1.5.1 Dispositivos

A colocação de dispositivos em todo o ambiente do utilizador, fornecendo meios para que eles se interliguem, conversarem e trabalhem juntos é o novo paradigma que suporta a computação ubíqua. Em concepção, juntando os mundos reais e virtual, a construção de ubiquidade em serviços de informação e dispositivos é uma meta imprescindível [4].

##### 2.1.5.2 Redes sem fio

As tecnologias sem fio são uma realidade que cresce a cada dia, em ritmo acelerado, e que conquistam cada vez mais adeptos. As redes sem fio variam em seu modelo estrutural de diversas

formas. Podem ser implementadas de uma maneira mais complexa, baseadas em estações interligadas por um ou vários pontos de acesso, ou de maneira mais simples, onde estações podem comunicar entre si, num modelo ponto-a-ponto [5].

- **Wi-Fi**

Nome de uma popular tecnologia de rede sem fio, que usa ondas de rádio para fornecer Internet de alta velocidade sem fios e ligações de rede. A organização que possui o Wi-Fi (marca registada) define especificamente Wi-Fi como WLAN produtos que seja baseados na norma IEEE 802.11.

Inicialmente, Wi-Fi foi apenas usado no lugar do padrão 802.11b 2.4GHz. No entanto, a Wi-Fi Alliance tem ampliado o uso genérico do termo Wi-Fi para incluir qualquer tipo de produto ou de rede WLAN com base em qualquer das normas 802.11, inclusive 802.11b, 802.11a, banda dupla, e assim por diante, numa tentativa de impedir a confusão sobre a interoperabilidade de LAN sem fios.

Wi-Fi é suportado por muitas aplicações e dispositivos, incluindo consolas de videojogos, redes domésticas, PDAs, telemóveis, sistemas operacionais, e outros tipos de electrónica de consumo. Os produtos que são testados e aprovados como "Wi-Fi Certified" (marca registada pela Wi-Fi Alliance) são certificados como interoperáveis entre si, mesmo que sejam de fabricantes diferentes. Por exemplo, um utilizador com um produto certificado Wi-Fi pode usar qualquer tipo de ponto de acesso com qualquer outra marca de hardware do cliente que é também "Wi-Fi Certified". Produtos que passam esta certificação são obrigados a ter um selo de identificação em suas embalagens que os identifica como "Wi-Fi Certified" e indica a banda de frequência de rádio usada (2.5GHz para 802.11b, 802.11g ou 802.11n, e 5GHz para 802.11a) [6].

- **Bluetooth**

É uma tecnologia que permite uma comunicação simples, rápida, segura e barata entre computadores, *smart phones*, telemóveis, ratos, teclados, auriculares, impressoras e outros dispositivos, utilizando ondas de rádio no lugar de cabos. Assim, é possível fazer com que dois ou mais dispositivos comecem a trocar informações com uma simples aproximação entre eles.

Bluetooth é um padrão global de comunicação sem fio e de baixo consumo de energia que permite a transmissão de dados entre dispositivos compatíveis com a tecnologia. Para isto, uma combinação de hardware e software é utilizada para permitir que essa comunicação ocorra entre os mais diferentes tipos de aparelhos. A transmissão de dados é feita através de radiofrequência, permitindo que um dispositivo detecte o outro independentemente das suas posições, desde que estejam dentro do limite de proximidade.

Para que seja possível atender aos mais variados tipos de dispositivos, o alcance máximo do Bluetooth foi dividido em três classes:



- **Classe 1:** potência máxima de 100 mW, alcance de até 100 metros;
- **Classe 2:** potência máxima de 2,5 mW, alcance de até 10 metros;
- **Classe 3:** potência máxima de 1 mW, alcance de até 1 metro.

Isto significa que um aparelho com Bluetooth classe 3 só conseguirá se comunicar com outro se a distância entre ambos for inferior a 1 metro, por exemplo. Neste caso, a distância pode parecer inútil, mas é suficiente para ligar um auricular a um telemóvel de uma pessoa, por exemplo. É importante salientar, no entanto, que dispositivos de classes diferentes podem se comunicar sem qualquer problema, bastando respeitar o limite daquele que possui um alcance menor. A velocidade de transmissão de dados no Bluetooth é baixa: até a versão 1.2, a taxa pode alcançar, no máximo, 1 Mbps. Na versão 2.0, esse valor passou para até 3 Mbps. Embora estas taxas sejam curtas, são suficientes para uma ligação satisfatória entre a maioria dos dispositivos. Todavia, a busca por velocidades maiores é constante, como prova a chegada da versão 3.0, capaz de atingir taxas de até 24 Mbps.

Dispositivos Bluetooth operam na banda ISM (*Industrial, Scientific, Medical*) centrada em 2,45 GHz, que era formalmente reservada para alguns grupos de utilizadores profissionais. Nos Estados Unidos, a banda ISM varia de 2400 a 2483,5 MHz. Na maioria da Europa, a mesma banda também está disponível. A banda é dividida em 79 portadoras espaçadas de 1 MHz. Portanto, cada dispositivo pode transmitir em 79 frequências diferentes. Para minimizar as interferências, o dispositivo mestre, após sincronizado, pode mudar as frequências de transmissão de seus escravos para até 1600 vezes por segundo. Teoricamente sua velocidade pode chegar a 721 Kbps e possui três canais de voz [7].

- **Infravermelhos** Tecnologia Infravermelho usa luz difusa reflectida, em paredes, móveis, etc, ou uma luz dirigida, se uma linha de visão (LOS) existir entre o emissor e o receptor. O Emissor pode ser uma simples diodo emissor de luz (LED) ou diodos de laser. Fotodiodos agem como receptores. Os infravermelhos tem uma rápida propagação, mas em contrapartida tem uma largura de banda limitada devido as interferências das luzes do ambiente [5].
- **Transmissão rádio** As experiências de longo prazo, realizadas com transmissão de rádio para redes de área ampla (por exemplo micro-ondas) e telefones móveis constituem uma vantagem para transmissão de rádio. A transmissão de rádio pode cobrir áreas maiores e podem penetrar paredes, móveis, plantas, como já referido, podendo adquirir ainda cobertura adicional através da reflexão. Rádio normalmente não precisa de uma linha de visão (LOS) se as frequências não forem demasiado elevadas [5].

## 2.2 Computação Orientada a Serviços

A computação orientada a serviços representa uma nova geração de plataformas de computação distribuída. Como tal, engloba muitas coisas, incluindo o seu paradigma de próprio desenho

e princípios arquitecturais. Ou seja, representa um modelo distinto de arquitectura e de conceitos, tecnologias e *frameworks*. Para entender melhor a computação orientada a serviços, as Secções seguintes definem cada um desses elementos e conclui com uma secção que explica como eles se inter-relacionam conceitualmente e fisicamente.

## 2.2.1 Elementos da Computação Orientada a Serviços

De acordo com os conceitos apresentados em [8], abaixo descrevem-se os principais componentes de um sistema computacional baseados em serviços. A figura 2.2 ilustra as suas diversas relações.

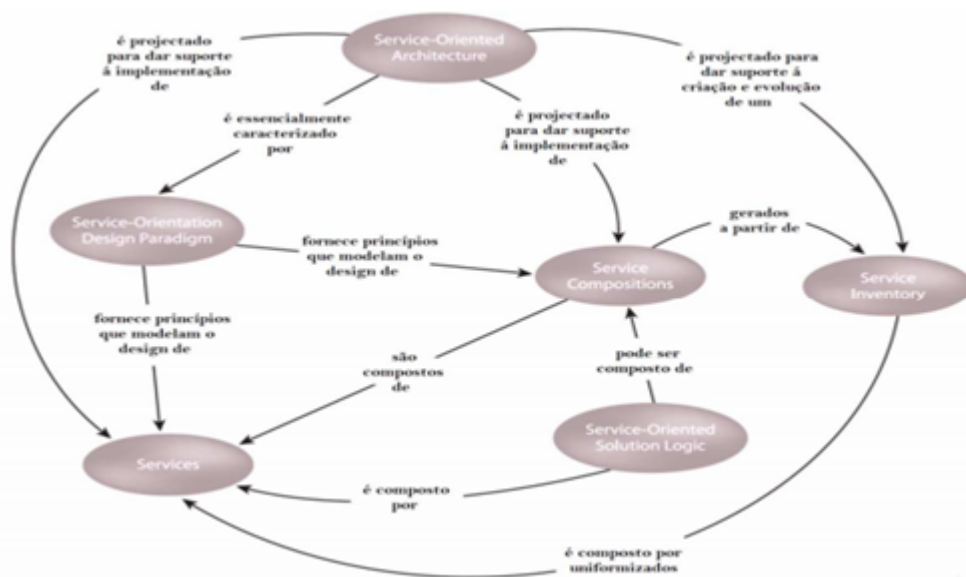


Figura 2.2: Uma visão conceitual de como os elementos de SOC se inter-relacionam [8]

- Arquitectura orientada a serviços estabelece um modelo arquitectónico que visa aumentar a eficiência, agilidade e produtividade de uma empresa de serviços como o principal meio através do qual a solução lógica é representada em apoio à realização dos objectivos estratégicos associadas com computação orientada a serviços. Arquitectura orientada a serviços é uma forma de arquitectura de tecnologia otimizada de apoio aos serviços, composições de serviço, e os inventários de serviços.
- Orientação a serviços é um paradigma de projecto composto por um conjunto específico de princípios de desenho. A aplicação destes princípios para a solução lógica resulta numa solução orientada ao serviço. A unidade mais fundamental da lógica de solução orientada a serviços é o serviço, portanto.

- A composição de serviços é composta de serviços que tenham sido especificados para fornecer a funcionalidade necessária para automatizar uma tarefa de negócio ou processo específico.
- Um inventário de serviço é uma colecção padronizada de forma independente e governada de serviços complementares dentro de um limite que representa uma sociedade ou um segmento significativo de uma empresa.
- Uma solução lógica orientada a serviços é implementada como serviços e composições de serviços elaboradas em conformidade com os princípios de desenho orientação a serviços.

### 2.2.2 Arquitectura Orientado a Serviço (SOA)

O constante crescimento do sector dos serviços no ambiente económico, traz com ele investimentos para a melhoria deste tipo de actividade, fazendo surgir a ciência de serviços. Os sistemas tradicionais cliente-servidor sempre focaram a interacção com o utilizador e os dados de forma centralizada, o que tornava difícil qualquer mudança estrutural pelo facto de exigir um grande esforço computacional, até mesmo nas novas aplicações clientes. A proposta da arquitectura SOA quebra este paradigma já que decompõe os sistemas em forma de serviços, permitindo que estes serviços sejam consumidos por aplicações clientes, através de uma comunicação padronizada. O conceito do que é uma arquitectura orientada a serviços ainda não é consensual, uma vez que o termo serviço não é um termo bem definido e se confunde com os conceitos de componentes e de contractos. Há diferentes interpretações do que é SOA, dependendo do interlocutor. Alguns exemplos são apresentados a seguir, com base no que é sugerido em [9].

- **Director de Negócios** “SOA é uma tecnologia que cria um ambiente de negócio ágil e provê vantagem competitiva ou maior valor acrescentado ao negócio”;
- **Gerente TI** “SOA é conjunto de processoss, estruturas e directrizes de governança que permite alinhar TI às necessidades do negócio”;
- **Arquitecto SW** “SOA é uma arquitectura de software baseada em padrões abertos que permitem integrar aplicações novas e existentes”;
- **Desenvolvedor** “SOA é um framework baseado em webservices que permite invocar objectos remotamente utilizando protocolo SOAP, baseado em XML” .

#### 2.2.2.1 O Conceito de Serviço

Diversas definições e conceitos sobre serviços apontam para um bem não tangível, existente apenas no espaço de tempo em que o produtor está a criar ou a construir e o consumidor está a usufruir. Esta definição é uma síntese das definições apresentadas a seguir, de acordo com [9]:

- “ É uma actividade ou série de actividades providas como a solução para os problemas do consumidor ”.

- “É toda a actividade económica cujo o resultado não gera algo físico, produto ou construção”.
- “É a experiência intangível praticada por um consumidor actuando como co-produtor”.

Serviços são módulos de negócio ou funcionalidades das aplicações que possuem interfaces expostas, e que são invocados via mensagens. Em SOA, recursos de software são agrupados como serviços bem definidos, auto-contidos, provendo funcionalidades e padrões de negócio, independentes do estado ou contexto de outros serviços. Há dois tipos de serviços: serviços de negócio e serviços técnicos (ou de infra-estrutura). Os serviços de negócio reflectem o conceito do negócio e estão tipicamente associados à execução de uma função de negócio de uma organização. Eles ampliam naturalmente a linguagem do negócio e podem se tornar a ponte de terminologia entre TI e os utilizadores do negócio durante a análise e o projecto. Os serviços técnicos ou de infra-estrutura são reutilizáveis por todos os processos de negócio. Eles são serviços de TI que devem ser nivelados por todas as linhas do negócio (por exemplo, serviços de segurança, de impressão, de log, de auditoria, de monitorização, de estatísticas em tempo de execução, de verificação de interfaces). Estes serviços tipicamente utilizam a infra-estrutura de serviços para prover alguma funcionalidade adicional, que não é focada no negócio [10]. De um ponto de vista puro de SOA, serviços técnicos não representam serviços, porque serviços deveriam representar sempre uma funcionalidade do negócio. Por outro lado, se temos a infra-estrutura de disponibilização de serviços, podemos utilizá-la para nos ajudar em outros aspectos. Na realidade, esta é uma questão de terminologia. O importante é deixar claro que estes serviços têm um propósito específico, a fim de que as equipas do negócio não tenham a impressão de que os serviços não necessariamente representam funcionalidades do negócio. É importante manter uma camada de abstracção clara que encapsula detalhes técnicos [11].

#### 2.2.2.2 Objectivos de SOA

SOA tem como objectivo permitir que as organizações realizem os seus negócios e tenham vantagens tecnológicas por meio da combinação de inovação de processos, eficácia na gestão e estratégia tecnológica, as quais circundam em volta da definição e reutilização de serviços. A melhoria da produtividade e agilidade, tanto para o negócio quanto para TI, é um dos benefícios mais importantes que SOA oferece e, conseqüentemente, a redução de custos no desenvolvimento e manutenção dos sistemas envolvidos [12].

#### 2.2.2.3 Características de SOA

Actividades de negócio são realizadas através de uma série de serviços que possuem maneiras bem definidas de “pedir” e “responder” as solicitações de informação. Desde que os serviços respondam de forma correcta aos comandos com a respectiva qualidade, não interessará o modo como este foi implementado. Isto significa que o serviço precisa de ser adequadamente seguro e confiável, além de rápido o suficiente, fazendo de SOA uma tecnologia ideal para ser utilizada num ambiente de TI que possua hardware e software de múltiplos fabricantes [9].

#### 2.2.2.4 Vantagens de SOA

A adopção de SOA, com todas as alterações que esta comporta, por parte dos vendedores e comunidades de utilizadores finais dentro da indústria de TI é um problema para qual é muito importante estabelecer o porquê desta adopção. SOA tem um enorme potencial arquitectural para oferecer, para qual reconhecem-se alguns benefícios, como sugerido em [13]:

- **Ubiquidade** Os serviços são acessíveis a partir de qualquer lugar e a qualquer momento;
- **Reutilização e Flexibilidade** Tem sido um dos maiores objectivos da engenharia de software à décadas, com variáveis graus de sucesso. Em SOA, a habilidade de compor novos serviços fora do contexto dos existentes apresenta possibilidade de os reutilizar e traz vantagens a uma organização que tem de ser ágil na resposta às necessidades de negócio. Neste sentido, existe uma redução de tempo e custos e aumento de qualidade no desenvolvimento de aplicações distribuídas através da reutilização de serviços;
- **Manutenção** A comunicação entre consumidores e provedores de serviços é baseada em interfaces. Isto ajuda imenso na manutenção porque são escondidos os detalhes de implementação;
- **Integração** A maioria das organizações tem uma infra-estrutura de aplicações altamente fragmentada, na qual foram criadas um vasto número de aplicações em múltiplas plataformas de programação e infra-estruturas de comunicação de vários vendedores. Neste contexto SOA traz muitas vantagens por ser um modelo de projecto baseado em interfaces, permitindo assim a integração das aplicações através de contratos entre consumidores e provedores de serviços.

## 2.3 Computação Orientado a Agentes

### 2.3.1 Agente e Sistema Multiagente

**Um Agente** é uma entidade de software que exhibe um comportamento autónomo, reactivo, pró-activo e com uma certa habilidade, geralmente orientado a objectivos, que está situado em algum ambiente sobre o qual é capaz de realizar acções para alcançar seus próprios objectivos de projecto e a partir do qual percebe alterações. Uma perspectiva baseada em agentes oferece uma gama de ferramentas, tecnologias e metáforas com potencial de melhorar consideravelmente a forma em que os indivíduos classificam e implementam diversos tipos de software. No entanto, o desenvolvimento de sistemas multiagente com um nível de complexidade elevada exige não só novos modelos e tecnologias, como também novas metodologias que apoiam os desenvolvedores na análise e desenho destes sistemas [14].

**Os Sistemas Multiagentes (SMA)** concentram-se no estudo de agentes autónomos em um universo multiagente. Para os SMA, o termo autónomo significa que os agentes têm uma existência própria, independente da existência de outros agentes. Usualmente, cada agente possui um

conjunto de capacidades comportamentais que definem sua competência, um conjunto de objectivos, e a autonomia necessária para utilizar suas capacidades comportamentais a fim de alcançar seus objectivos. Portanto, um agente é uma entidade computacional com um comportamento autónomo que lhe permite decidir suas próprias acções [14].

## 2.4 Exemplos de Aplicações

São muitas as aplicações que exploram o paradigma de Computação Ubíqua. Neste capítulo são apresentadas algumas dessas aplicações. As suas escolhas seguiram um conjunto de critérios considerados relevantes para os objectivos da dissertação. Os principais foram não só o facto de seus autores apresentarem uma arquitectura como suporte ao esclarecimento das respectivas aplicações, como também se ter levado em conta a quantidade de detalhes desta exposição. Foram agrupados em temas onde cada um se destaca. Assim, foram marcados como projectos na área da Computação Ubíqua (secção 2.4.1), da Computação Orientada a Serviços (secção 2.4.2), da Computação Orientada a Agentes (secção 2.4.3) e na área da Saúde (secção 2.4.4).

### 2.4.1 Sistemas na área de Computação Ubíqua

- Em [15] são apresentadas as actividades que integram o plano de acção "Rede de sensores sem fio na agropecuária". São actividades de monitorização e controlo de processos na agropecuária através do uso das inovadoras tecnologias de rede de sensores sem fio e a computação ubíqua. As redes de sensores sem fios (RSSF) ou WLAN são o resultado da rápida convergência de três tecnologias-chave: os circuitos digitais, a comunicação sem fio e os micro sistemas electromecânicos (*MEMS Micro Electro-Mechanical Systems*);
- EasyLiving [16] é um projecto da Microsoft Research, que define um conjunto de tecnologias destinadas à integração dinâmica de dispositivos, para o enriquecimento da experiência do utilizador na utilização do ambiente onde se encontra;
- Em [17] são apresentados três middleware para computação ubíqua, mais precisamente para mixed - reality, interacção de dispositivos e computação em casa. Utiliza Framework CORBA.

### 2.4.2 Sistemas na área de Computação Orientado a Serviços

- Em [18] é apresentada WASP - *Web Architectures for Service Platforms*, desenvolvida na University of Twente, Holanda, em que a arquitectura da plataforma é caracterizada particularmente pelo uso da tecnologia de distribuição de Web Services, e é executada sobre uma rede móvel 3G.
- Em [19] é apresentada uma arquitectura de middleware, orientada a serviços, desenvolvida na University of the Basques Country, uma extensão da *Jini Network Technology*, explorando a tecnologia Java. É constituída por: Control Resources, que inclui todos os dispositivos de

rede do ambiente controláveis; Context Resources, formada pelos sensores e outras fontes de informação ambiental da rede; Interaction Resources, que permite a interação de pessoas através de feedback ou comandos que permitam a conclusão dos seus objectivos; e pelo Ontology Manager, com função de gestão da base de dados de ontologias, importando dados ontológicos dos drivers, e criando uma primeira instância do Ontology Reasoner, de modo a permitir um comportamento inteligente e interacção proactiva dos aplicativos com os utilizadores.

### 2.4.3 Sistemas na área de Computação Orientado a Agentes

- Em [20] é apresentado o Global que é um modelo de educação ubíqua, descentralizado, baseado em sistemas multiagentes, formado por doze componentes. Interface com Utilizador, Persistência, Restritores e Proxys são componentes de suporte representados por API's. Os demais componentes são agentes de software. Esses agentes são executados de forma isolada, sendo que cada um disponibiliza um grupo específico de funcionalidades para o apoio da aprendizagem. Cada instância do Global, isto é, cada cópia sendo executada em um dispositivo, tem uma instância desses agentes em execução. O modelo não exige a execução de todos os agentes, sendo que a ausência de um compromete apenas as funcionalidades que dependem dele.
- Em [21] seus autores apresentam um sistema sensível ao contexto, usando múltiplos agentes para processar a informação, com o Context Collecting Agent (CCA), o Context Reasoning Agent (CRA) e o Information Processing Agent (IPA). O CCA recolhe informações de eventos relacionados com o utilizador de vários dispositivos e apresentados ao CRA, o qual tem armazenado os perfis e gera os correspondentes comandos que são enviados para o IPA, que traduz estes comandos para acções, mensagens ou algo mais complexo.
- Em [22] é apresentada a arquitectura dum sistema multiagente, dispendo a integração de dispositivos físicos de rede de uma forma muito rápida. *Java Agent Development Framework (JADE)* é o suporte desta arquitectura que divide-se em 3 grupos de agentes: Interface Agents, responsáveis pela transmissão de eventos de actualizações do sistema e recepção de acções de controlo dos utilizadores; Learning Agents, que envolvem diferentes tipos de algoritmos de aprendizagem, tais como lógica fuzzy, redes neuronais, árvores de decisão e, por último, Control/Utility agents, que controlam pontos no protocolo UPnP, passam mensagens de controlo aos correspondentes dispositivos software UPnP e estão à escuta dos eventos daqueles que estão registados.

### 2.4.4 Sistemas na área da Saúde

Sistema de informação em Saúde é um mecanismo de recolha, processamento, análise e transmissão da informação necessária para se organizar e operar os serviços de saúde e também para a investigação e o planeamento com vistas ao controlo de doenças. O propósito do Sistema de



Informação em saúde é seleccionar os dados pertinentes a esses serviços e transformá-los na informação necessária para o processo de decisões, próprios das organizações e indivíduos que planeiam, financiam, administram, provêem e avaliam os serviços de saúde. São funções do sistema de Informação em Saúde o planeamento, a coordenação e a supervisão dos processos de selecção, recolha, aquisição, registo, armazenamento, processamento, recuperação, análise e difusão de dados e geração de informação. [23] Portanto, trata-se de uma área onde computação ubíqua, SOA, e SMA aplicam-se muito bem.

- Em [24] é proposta a Self-Managed Cell (SMC), uma arquitectura modelo para aplicações ubíquas na saúde, em que para monitorizar os pacientes nas suas residências são utilizados sensores corporais. Essa arquitectura é essencialmente constituída por um conjunto extensível de serviços comunicação, feita pelo Barramento de evento, bem como a gestão e adaptação do controlo aos recursos dirigidos. Embora o conjunto de serviços possa modificar-se dependendo do contexto no qual o SMC é solicitado (p. ex., rede da área do corpo, sistema de controlo em casa, hospital), um número de serviços constituem a funcionalidade principal do SMC e sempre devem estar presentes. Descobridor de Serviços é responsável por manter a ligação na SMC. Este serviço interroga os novos dispositivos para estabelecer um perfil que descreve os serviços que eles oferecem e logo gera um evento que descreve a adição do novo dispositivo de outros componentes SMC para utilizar como apropriado. O Barramento de Eventos, lança as políticas de adaptação do comportamento do SMC. Serviço de Gestão, mantém objectos de adaptação para cada um dos componentes que podem ser alvo de acções de gestão.
- Em [25] é apresentado um *framework* para a monitorização de pacientes em casa com base em técnicas de computação pervasiva. O *framework* inclui componentes que permitem realizar o acompanhamento do paciente, permitindo a avaliação da sua situação médica e a geração de notificações de alarmes em momentos críticos. Um dos componentes principais do *framework* é um mecanismo de raciocínio, desenhado para identificar situações anormais relacionadas aos sinais vitais do paciente.
- Em [26] descreve-se uma *framework* da monitorização inteligente da saúde de uma pessoa em casa. Neste contexto, a computação pervasiva tem um papel importante que permite o tratamento de várias variáveis relevantes. A solução integra conhecimento médico, dados fisiológicos e comportamentais dos pacientes, e condições ambientais. A *framework* integra módulos da gestão de contexto, raciocinando e aprendendo. Um modelo lógico e as regras baseadas em recomendações médicas permitem analisar e identificar situações críticas do paciente. O exemplo de controlar a tensão arterial ilustra o uso desta proposta.
- Em [27] é apresentado O HealthShare, que é um software inovador que inclui a tecnologia necessária para a rápida troca de informações de Saúde e também um ambiente completo de desenvolvimento para minimizar o custo das soluções que atendam às necessidades de cada sistema envolvido. O HealthShare foi projectado para reduzir radicalmente o custo, o tempo



de desenvolvimento, e os riscos de construir e operar um sistema de troca de informações na área de saúde.

- Em [28] é apresentado o Wireless Remote Healthcare Monitoring with Motes que combina um monitor móvel (PDA) com a capacidade de remotamente e de qualquer lugar ter acesso à informação usando a Internet. O sistema usa tecnologia estabelecida para fornecer a monitorização de saúde rentável, eficiente e robusta que seria especialmente útil para cuidado continuado dos idosos.

## 2.5 Doenças Crónicas: Diabetes e Obesidades

Finalmente, é importante fazer-se uma breve descrição das doenças crónicas, com o objectivo de se compreender com que tipos de medições muitos desses pacientes geralmente se envolvem no seu dia-a-dia. Em muitos casos, tais medições devem ser realizadas pelo próprio paciente, como os níveis de glicose para os doentes que sofrem de diabetes, a fim de decidirem sobre a necessidade de se auto-ministrarem terapêuticas apropriadas para evitarem situações de risco. Entretanto, haverá outras doenças em que esta auto-monitorização por parte do paciente será necessária.

### 2.5.1 Breves Considerações

Segundo a Organização Mundial da Saúde, as doenças crónicas cuja declaração não é obrigatória, como as doenças cardiovasculares, a diabetes, a obesidade, o cancro e as doenças respiratórias, representam cerca de 59 por cento do total de 57 milhões de mortes em cada anuais e 46 por cento do total de doenças. Uma Doença crónica é uma doença cujo tempo de tratamento não é de todo curto, normalmente definido e apartir dos três meses .

As doenças crónicas não são de emergências médicas porque não colocam em risco a vida das pessoas num curto prazo. Porém, podem ser muito sérias e várias delas causam morte certa. As doenças crónicas possuem todas a condição em que um sintoma existe continuamente, e mesmo não pondo em risco a saúde física das pessoa, são extremamente incomodativo levando à disrupção da qualidade de vida e actividades da pessoas [29]. Um exemplo típico é a diabetes. Calcula-se que, em todo o mundo, existam 177 milhões de pessoas a soffrere de diabetes, sobretudo de tipo 2. Mais de mil milhões de adultos soffrem de excesso de peso. Destes, pelo menos 300 milhões são clinicamente obesos. Apesar de muito diferentes entre si, as doenças crónicas apresentam factores de risco comuns. São poucos e podem ser prevenidos como por exemplo:

- Colesterol elevado;
- Tensão arterial elevada;
- Obesidade;
- Tabagismo;
- Consumo de álcool.

## 2.5.2 Obesidade

A obesidade e a pré-obesidade são avaliadas pelo Índice de Massa Corporal (IMC). Este índice mede a corpulência, que se determina dividindo o peso (quilogramas) pela altura (metros), elevada ao quadrado, como indicado na expressão abaixo:

$$IMC = \frac{Peso}{Altura^2}$$

Existem vários tipos de aparelhos que servem para efectuar a respectiva medição, a figura 2.3 mostra um exemplo de medidor de IMC. Segundo a Organização Mundial de Saúde, considera-se que há excesso de peso quando o IMC é igual ou superior a 25 e que há obesidade quando o IMC é igual ou superior a 30. A obesidade é uma doença! Mais, é uma doença que constitui um importante factor de risco para o aparecimento, desenvolvimento e agravamento de outras doenças. Há tantas pessoas obesas a nível mundial que a Organização Mundial de Saúde (OMS) considerou esta doença como a epidemia global do século XXI. De acordo com a OMS, a obesidade é uma doença em que o excesso de gordura corporal acumulada pode atingir graus capazes de afectar a saúde. É uma doença crónica, com enorme prevalência nos países desenvolvidos, atinge homens e mulheres de todas as etnias e de todas as idades, reduz a qualidade de vida e tem elevadas taxas de morbilidade e mortalidade, acarretando múltiplas consequências graves para a saúde [30].



Figura 2.3: Medidor de IMC

### 2.5.2.1 Tipos de obesidade

- **Obesidade andróide, abdominal ou visceral** - quando o tecido adiposo se acumula na metade superior do corpo, sobretudo no abdómen. É típica do homem obeso. A obesidade visceral está associada a complicações metabólicas, como a diabetes tipo 2 e a dislipidémia e, a doenças cardiovasculares, como a hipertensão arterial, a doença coronária e a doença vascular cerebral, bem como à síndrome do ovário poliquístico e à disfunção endotelial (ou seja deterioração do revestimento interior dos vasos sanguíneos). A associação da obesidade a estas doenças está dependente da gordura intra-abdominal e não da gordura total do corpo.
- **Obesidade do tipo ginóide** - quando a gordura se distribui, principalmente, na metade inferior do corpo, particularmente na região glútea e coxas. É típica da mulher obesa.

### 2.5.2.2 Classificação da Obesidade

Um gráfico sobre o IMC é mostrado na Figura 2.4. As linhas tracejadas representam subdivisões dentro das classes principais, por exemplo, a magreza é dividida em "severa", "moderada" e "leve". Na Tabela 2.1 pode-se ver a classificação do imc de uma forma mais detalhada. [31]

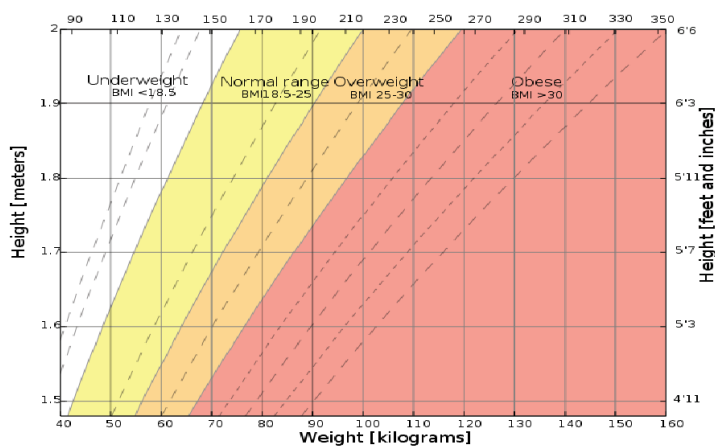


Figura 2.4: Gráfico sobre o IMC [31]

$IMC < 18 \frac{Kg}{m^2}$	Magresal
$IMC > 18 < 25 \frac{Kg}{m^2}$	Normal
$IMC > 25 < 30 \frac{Kg}{m^2}$	Excesso de Peso
$IMC > 30 < 35 \frac{Kg}{m^2}$	Obesidade moderada (grau I)
$IMC > 35 < 40 \frac{Kg}{m^2}$	Obesidade grave (grau II)
$IMC > 40 \frac{Kg}{m^2}$	Obesidade mórbida (grau II)

Tabela 2.1: Classificação de IMC [31]

### 2.5.3 Hipercolesterolemia

Manifesta-se quando os valores do **colesterol** no sangue são superiores aos níveis máximos recomendados em função do risco cardiovascular individual. O colesterol é indispensável ao organismo, quaisquer que sejam as células orgânicas que necessitem de regenerar-se, substituir-se ou desenvolver-se. No entanto, valores elevados são prejudiciais à saúde. Há dois tipos de colesterol: **O colesterol HDL (High Density Lipoproteins)**, designado por “bom colesterol”, é constituído por colesterol retirado da parede dos vasos sanguíneos e que é transportado até ao fígado para ser eliminado.

**O colesterol LDL (Low Density Lipoproteins)** é denominado “mau colesterol”, porque, quando em quantidade excessiva, ao circular livremente no sangue, torna-se nocivo, acumulando-se perigosamente na parede dos vasos arteriais. Quer o excesso de colesterol LDL, quer a falta de

colesterol HDL, aumentam o risco de doenças cardiovasculares, principalmente o enfarte do miocárdio.



Figura 2.5: Medidor de Colesterol e Glicose

#### 2.5.4 Glicose

O nome Glucose veio do grego glykys , que significa "doce", mais o sufixo -ose, indicativo de açúcar. Tem função de fornecedor de energia, participa das vias metabólicas, além de ser precursora de outras importantes moléculas.

A glicose, glucose ou dextrose, um monossacarídeo, é o carboidrato mais importante na biologia. As células a usam como fonte de energia e intermediário metabólico. A glucose é um dos principais produtos da fotossíntese e inicia a respiração celular em procariontes e eucariontes. É um cristal sólido de sabor adocicado encontrado na natureza na forma livre ou combinada. Juntamente com a frutose e a galactose, é o carboidrato fundamental de carboidratos maiores, como sacarose e maltose. Amido e celulose são polímeros de glucose. É encontrada nas uvas e em vários frutos. Industrialmente é obtida a partir do amido.

No metabolismo, a glucose é uma das principais fontes de energia e fornece 4 calorias de energia por grama. A glucose hidratada (como no soro glicosado) fornece 3,4 calorias por grama. Sua degradação química durante o processo de respiração celular dá origem à energia química (armazenada em moléculas de ATP - aproximadamente 30 moléculas de ATP por moléculas de glucose), gás e água [32].

#### 2.5.5 Hipertensão Arterial

Representa situações em que se verificam valores de tensão arterial aumentados. Para esta caracterização, consideram-se valores de tensão arterial sistólica (“máxima”) superiores ou iguais a 140 mm Hg (milímetros de mercúrio) e/ou valores de tensão arterial diastólica (“mínima”) superiores ou iguais a 90 mm Hg. Contudo, nos doentes diabéticos, porque a aterosclerose progride mais rapidamente, considera-se haver hipertensão arterial quando os valores de tensão arterial sistólica são superiores ou iguais a 130 mm Hg e/ou os valores de tensão arterial diastólica são superiores ou iguais a 80 mm Hg. Com frequência, apenas um dos valores surge alterado. Quando



Figura 2.6: Medidor de Pressão Arterial

apenas os valores da “máxima” estão alterados, diz-se que o doente sofre de hipertensão arterial sistólica isolada; quando apenas os valores da “mínima” se encontram elevados, o doente sofre de hipertensão arterial diastólica. A primeira é mais frequente em idades avançadas e a segunda em idades jovens. A hipertensão arterial está associada a um maior risco de doenças cardiovasculares, particularmente o acidente vascular cerebral [33].

### 2.5.5.1 Classificação da hipertensão arterial

A finalidade de classificações da tensão arterial é determinar grupos de pacientes que tenham características comuns, quer em termos de diagnóstico, de prognóstico ou de tratamento. Estas classificações são baseadas em dados científicos, mas são em certo grau arbitrárias. Diversas sociedades científicas têm suas classificações próprias. A Tabela 2.2 mostra a classificação da sociedade europeia de hipertensão e de cardiologia.

Categoria	PA diastólica (mmHg)	PA sistólica (mmHg)
Tensão Ótima	<80	<120
Tensão Normal	80-84	120-129
Tensão Normal Alta	85-89	130-139
Hipertensão grau 1	90-99	140-159
Hipertensão grau 2	100-109	160-179
Hipertensão grau 3	$\geq 110$	$\geq 180$
Hipertensão sistólica isolada	<90	$\geq 140$

Tabela 2.2: Classificação da Sociedade Europeia de Hipertensão e de Cardiologia [34]

### 2.5.6 Temperatura Corporal

“Os seres humanos podem passar sem água durante dias, em alguma ocasião, e sem comida durante várias semanas, em caso de necessidade, mas não podem passar sem aquecimento por mais de horas”. (ATKINSON, MURRAY, 1989). Segundo Perry Potter, temperatura corporal é o grau de calor que o corpo apresenta. É o equilíbrio entre o calor produzido e eliminado pelo corpo. Há uma variação precisa de temperatura dentro da qual as células funcionam com eficiência



Figura 2.7: Medidor de Temperatura: Termómetro

e a actividade enzimática é adequada. A média normal de temperatura para adultos considerados normais é de 36 °C a 37 °C graus, e é variável de acordo com o local em que for tomada como mostra a Tabela 2.3. Actividade enzimática é a capacidade das células em produzir enzimas be-

<b>Axilar</b>	36,0 °C a 37,0 °C
<b>Inguinal</b>	36,0 °C a 36,8 °C
<b>Buca</b>	36,8 °C a 37,0 °C
<b>Retal</b>	37,0 °C a 37,2 °C

Tabela 2.3: Tabela de Temperatura [33]

néficas ao organismo. Para manter-se em perfeito funcionamento, o ser humano necessita manter sua temperatura central dentro de uma faixa térmica estreita, por isso é dito que o homem é um animal homeotérmico. Se a temperatura corporal tornar-se excessivamente alta ou baixa, todos os sistemas do organismo são afectados, resultando na morte das células e consequentemente de todo sistema, a menos que a temperatura possa retornar a uma faixa mais normal [33].

## 2.6 Sumário

Através da apresentação de diversas informações de carácter mais geral pretendeu-se proporcionar ao leitor uma visão mais detalhada sobre cada um dos temas acima expostos no sentido de compreender o background teórico que suporta o desenvolvimento deste trabalho. Iniciou-se o capítulo com a abordagem a computação ubíqua dando ênfase a sua característica, a sua relação com a computação móvel e a pervasiva, os desafios que algumas subárea proporcionam e ainda as tecnologias de localização. Seguidamente foi descrito a computação orientada a serviços, analisando-se os elementos que a compõe, arquitectura orientado a serviços e arquitectura orientado a agentes. De seguida foi apresentado um conjunto de trabalhos que abordam cada um dos conceitos mencionados. Por último fez-se uma análise às doenças crónicas, dando ênfase à diabetes.

## Capítulo 3

# Análise e Especificação de Requisitos

Neste capítulo será descrito todo o trabalho desenvolvido para a criação e implementação do sistema de monitorização de pacientes cujo objectivo é permitir um fácil acompanhamento, por parte dos médicos, de parâmetros e sinais vitais dos pacientes.

### 3.1 Perspectiva geral do sistema

Este ponto tem por objectivo dar uma visão global do sistema, das suas interfaces externas e dos seus subsistemas principais. A arquitectura de um sistema caracteriza o relacionamento e a comunicação entre os diferentes módulos do sistema e possibilita visualizar o seu comportamento global de um modo simplista e de alto nível, além de identificar todos os processos e procedimentos chave essenciais ao seu funcionamento. Através deste tipo de arquitectura vislumbra-se o primeiro contacto do leitor com o sistema na sua globalidade, permitindo assim a partir da sua observação identificar e estruturar as principais actividades e intervenientes directamente relacionadas com o projecto. Este sistema foi projectado como uma arquitectura cliente -servidor composta por quatro componentes Aplicação Web, Aplicação Móvel, Servidores e Base de Dados. Estão ligados entre si através da internet, como mostra a Figura 3.1.

Está prevista a utilização desta plataforma no meio hospitalar (para os profissionais de saúde) e no caso do paciente pode ser usada em qualquer sítio desde que tenha a cobertura de Internet e disponha dos meios para a medição de dados biométricos em causa. Os profissionais de saúde também podem utilizar a respectiva plataforma fora do meio hospitalar. Hoje em dia existe uma grande percentagem de pessoas com doenças crónicas que mesmo tendo acompanhamento médico não fazem um controlo rigoroso sobre seus parâmetros vitais. Muitos realizam as várias medições que estão sujeitos anotando em bloco de notas "muita das vezes sem datas associadas", alguns memorizam e outros nem isso fazem.

Como visto na arquitectura geral do sistema, apresentada na Figura 3.1, tanto médico como paciente terão acesso a funcionalidades do sistema a partir dos seus terminais ligados à Internet.

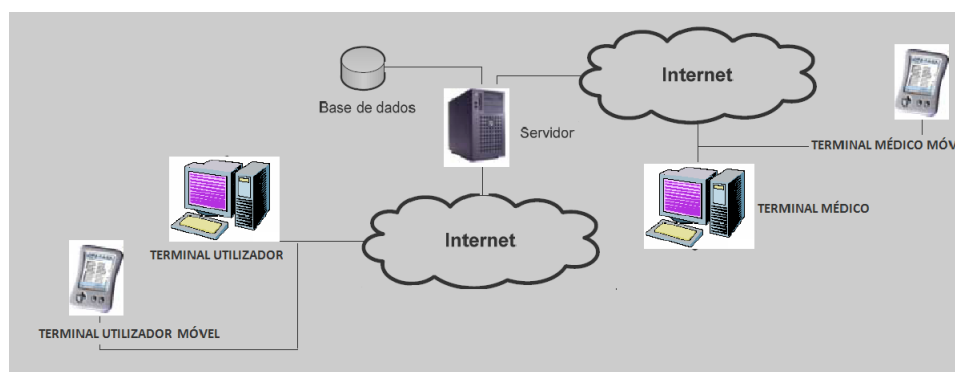


Figura 3.1: Arquitectura Geral do Sistema

Para isso, será implementada uma aplicação Web com os principais recursos identificados nos requisitos a seguir. De momento, é importante definir o conceito de aplicação Web, que neste projecto refere-se ao termo utilizado para designar, de forma geral, sistemas de informação projectados para utilização através de um navegador, na Internet ou em redes privadas ( intranet ). Trata-se de um conjunto de programas que é executado num servidor de HTTP (Web Host). O desenvolvimento da tecnologia Web está relacionado, entre outros factores, à necessidade de simplificar a actualização e manutenção mantendo o código-fonte em um mesmo local, de onde, ele é ligado pelos diferentes utilizadores.

Pode-se definir uma aplicação Web como uma aplicação de software que utiliza a Web, através de um browser como ambiente de execução. Uma Aplicação Web também é definida em tudo que se é processado em algum servidor, por exemplo: quando se entra em um e-commerce a página que se liga antes de vir até o navegador é processada num computador ligado à Internet que retorna o processamento das regras de negócio nele contido. Por isso se chama aplicação e não simplesmente sítio Web.

## 3.2 Características dos Utilizadores

Este Sistema permitirá quatro tipos de utilizadores: Pacientes, Profissionais de Saúde, Públicos e Administradores. Os Profissionais de saúde são também administradores, mas para realizarem o registo estes precisam de um código de aceitação que é fornecido pela administração. No entanto o profissional de saúde é que valida o registo dos pacientes activando as fichas clínicas então criadas. Quanto ao público estes estão sujeitos à validação por parte dos administradores. O uso da plataforma dependerá da forma como esta é utilizada pelos aplicativos, antevendo-se muito poucos ou mesmo nenhuns conhecimentos informáticos por parte dos seus utilizadores.

## 3.3 Especificação detalhada dos Requisitos

Foi possível definir os seguintes requisitos:





Figura 3.2: Exemplo de registo em bloco de notas

- **R.1** O sistema deve permitir o registo de utilizadores e a respectiva entrada no sistema utilizando os dados solicitados;
- **R.2** Todos os utilizadores devem ter uma username e password;
- **R.3** O sistema deve permitir ao utilizador seleccionar o tipo de dados biométricos a enviar e enviá-los;
- **R.4** Todos os dados biométricos enviados devem ter uma data e hora associada;
- **R.5** Os pacientes devem poder visualizar os históricos das medições;
- **R.6** Todos os pacientes devem ter uma ficha clínica e um médico associado;
- **R.7** O sistema deve ter sistemas de alertas em caso dos dados estarem fora do padrão;
- **R.8** O Sistema deve permitir acesso de utilizadores sem acompanhamento médico;
- **R.9** O Sistema deve permitir aos utilizadores consultarem as médias das medições;
- **R.9** O Sistema deve permitir aos utilizadores visualizarem e editarem os seus dados pessoais;
- **R.10** Devido ao conteúdo sensível dos dados deverão ser implementados modelos de segurança para proteger a confidencialidade e integridade dos dados;
- **R.11** O Sistema deve ter um sistema de alerta no caso do utilizador não enviar dados durante um período dentro do estipulado;
- **R.12** O Sistema deverá dispor de pelo menos dois tipos de interfaces distintas, uma para dispositivos móveis (telemóveis, PDAs, etc) e outras para PCs;

### 3.4 Restrições

Foi imposta as seguintes restrições:

- **Rt.1** O sistema deve ser desenvolvido em tecnologias open source e utilizando a linguagem Java; e/ou outra que permita desenvolvimento para a Web;
- **Rt.2** Apesar do sistema merecer um nível de segurança elevado, não terá esse grau uma vez que o tempo é muito curto para o efeito.

A fiabilidade de um sistema médico de monitorização de pacientes depende da sua capacidade de evitar falhas, para que possa fornecer um serviço contínuo e correcto. A segurança e igualmente importante, a privacidade e integridade deve ser assegurada.

### 3.5 Casos de Utilização

Os casos de utilização são as acções que devem suceder quando um actor interage com o sistema e que permite ao mesmo atingir o seu objectivo. Assim, cada caso de utilização é uma sequência de possíveis acções realizadas por um actor e o sistema numa determinada altura. Por um lado, os casos de utilização devem ser definidos para representar os objectivos do actor, e por outro lado, o caso de utilização deve representar as funções ou comportamentos do sistema que representa a interacção com o actor.

As Figuras 3.3, 3.4 e 3.5 mostram os diagramas de casos de utilização do sistema. A implementação foi baseada nesses diagramas e foi enfatizada na parte de perfis, envio de medições, alertas e históricos de pacientes, pois são as classes que importam para fazer com que as duas aplicações fiquem funcionais.

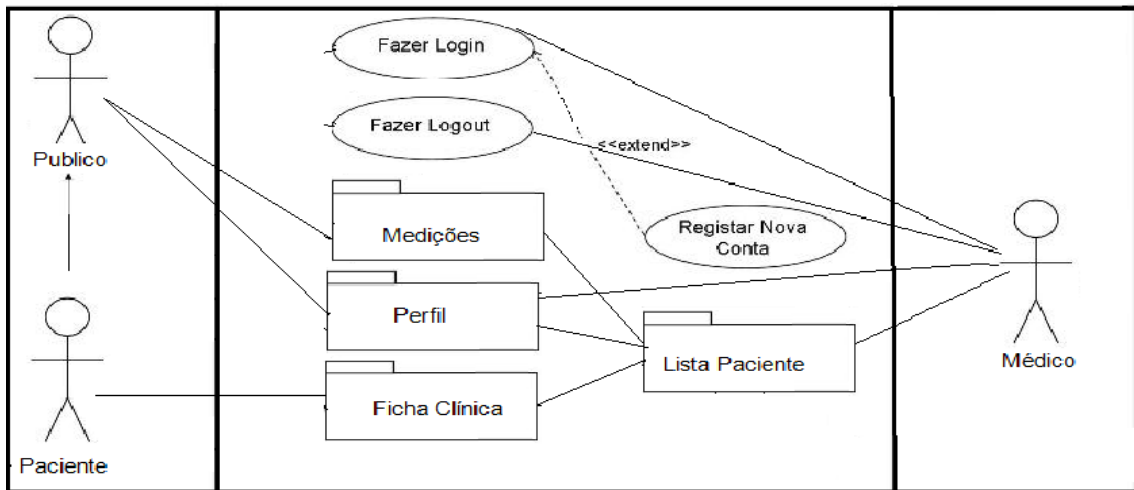


Figura 3.3: Casos de utilização 1

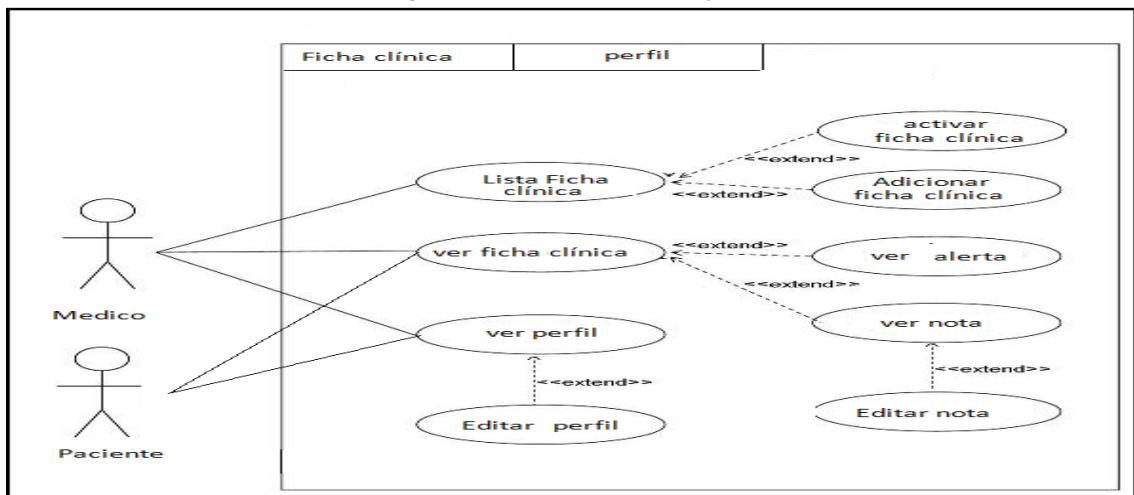


Figura 3.4: Casos de utilização 2

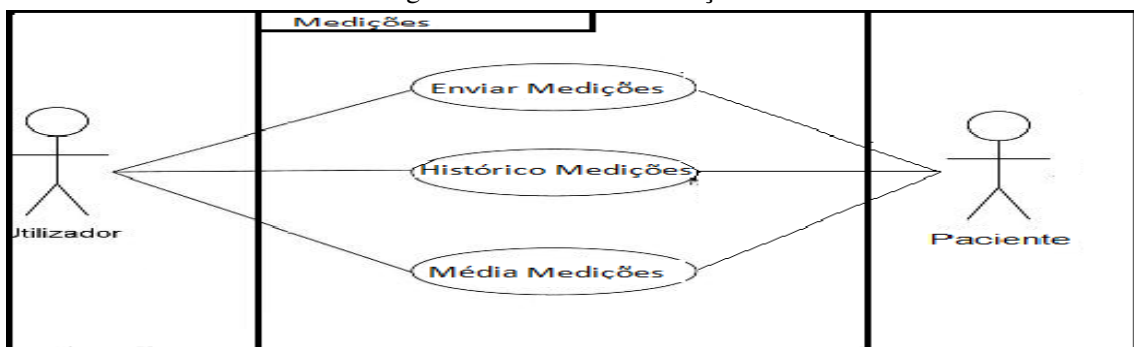


Figura 3.5: Casos de utilização 3

## 3.6 Ferramentas de Desenvolvimento

A seguir apresenta-se uma breve discussão sobre as ferramentas utilizadas no desenvolvimento deste projecto, nomeadamente o Sistema Operativo Android, os sistemas de bases de dados relacionais, e a plataforma de desenvolvimento Eclipse.

### 3.6.1 Sistema Operativo Android

O Android é um sistema operativo móvel baseado numa versão modificada do Linux. Foi originalmente desenvolvida por uma startup de mesmo nome, Android Inc. Em 2005, como parte de sua estratégia para entrar no espaço móvel, o Google comprou o Android e assumiu o seu trabalho de desenvolvimento (bem como seu desenvolvimento em equipa). O Google Android pretende ser livre e aberto, daí, a maioria do código do Android ser lançado sob a licença open-source Apache License, o que significa que quem quiser usar o Android pode fazê-lo descarregando o código fonte do Android. Além disso, os vendedores (normalmente os fabricantes de hardware) podem adicionar suas próprias extensões proprietárias para o Android e personalizar o mesmo para diferenciar seus produtos de outros. Esse modelo de desenvolvimento simples faz com que o Android seja muito atraente e tenha, portanto, despertado interesse de muitos vendedores. Este tem sido especialmente o caso para as empresas afectadas pelo fenómeno do iPhone, da Apple, um produto de enorme sucesso que revolucionou a indústria de *smartphones*. Essas empresas incluem também a Motorola e Sony Ericsson, que por muitos anos têm vindo a desenvolver seus próprios sistemas operativos móveis. Quando o iPhone foi lançado, muitos desses fabricantes tiveram que lutar para encontrar novas formas de revitalizarem seus produtos. A principal vantagem da adopção do Android é que ele oferece uma abordagem unificada para desenvolvimento de aplicações. Os desenvolvedores precisam apenas desenvolver para o Android, e sua aplicação deve ser capaz de funcionar como sistema operativo em vários dispositivos diferentes, contanto que os dispositivos utilizem o Android. No mundo dos *smartphones*, os pedidos são a parte mais importante da cadeia de sucesso. Os fabricantes de dispositivos, portanto, vêm o Android como a sua melhor esperança para desafiar a investida do iPhone, que já comanda uma grande parte das aplicações. [35]

#### 3.6.1.1 Versões de Android

O Android passou por um grande número de actualizações desde sua primeira versão. A Tabela 3.1 mostra as várias versões do Android e os seus nomes de código respectivos.

#### 3.6.1.2 Características do Android

Como o Android é de código aberto e disponível gratuitamente para os fabricantes, não há configurações fixas de hardware e software. No entanto, o Android em si suporta as seguintes funcionalidades:

- **Armazenamento** - Usa SQLite, um banco de dados relacional e leve, para armazenamento de dados;

Android Version	Release Date	Codename
1.1	9 February 2009	
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 October 2009	Eclair
2.2	20 May 2010	Froyo
2.3	6 December 2010	Gingerbread
3.0	22 February 2011	Honeycomb
3.1	10 May 2011	Honeycomb

Tabela 3.1: tabela de versões de Android [35]

- **Conectividade** - Suporta GSM / EDGE, IDEN, CDMA EV-DO, UMTS, Bluetooth (inclui A2DP e AVRCP), WiFi, LTE e WiMAX;
- **Mensagens** - Suporta SMS e MMS;
- **Navegador da Web** - Com base no WebKit que é open-source, em conjunto com o motor V8 JavaScript do Chrome;
- **Suporte a Media** - Inclui suporte para as seguintes medias: H.263, H.264 (no formato 3GP ou MP4 container), MPEG-4 SP, AMR, AMR-WB (no formato 3GP container), AAC, HE-AAC (no formato MP4 ou 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF e BMP;
- **Apoio Hardware** - Sensor Acelerómetro, camera, bússola digital, sensor de proximidade e GPS;
- **Multi-touch** - Suporta ecrãs multi-toque;
- **Multi-tasking** - Suporta aplicações multi-tasking; multi-tarefa;
- **Suporte a Flash** - Android 2.3 suporta Flash 10.1 por exemplo;
- **Tethering** - Suporta partilha de ligações à Internet como um ponto de acesso com fio ou sem fio

### 3.6.1.3 Arquitectura do Android

A fim de entender como funciona, a Figura 3.6 mostra as várias camadas que compõem o sistema operativo Android .

O Android é dividido em cinco secções e em quatro camadas principais, como descrevemos a seguir:

- **kernel Linux** - Este é o núcleo sobre o qual o Android é baseado. Essa camada contém todos os drivers de dispositivo para vários componentes de hardware de um dispositivo Android;

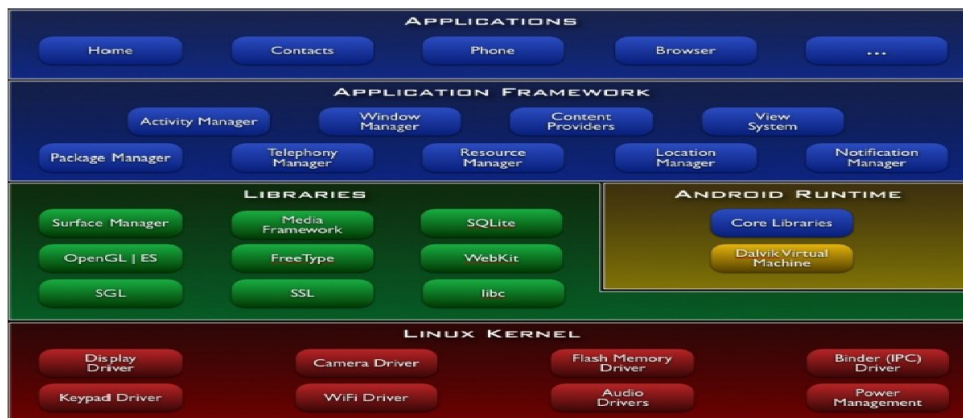


Figura 3.6: Arquitetura do Android [36]

- **Libraries-** Contêm todo o código que fornece as características principais de um sistema operativo Android. Por exemplo, a biblioteca fornece suporte de base de dados SQLite para que um aplicativo possa utilizá-lo para armazenamento de dados. A biblioteca WebKit fornece funcionalidades para navegação na Web;
- **Android Runtime** - Na mesma camada que as bibliotecas, o tempo de execução Android fornece um conjunto nuclear de bibliotecas que permitem aos desenvolvedores escrever aplicações Android utilizando a linguagem de programação Java. A Runtime Android também inclui a máquina virtual Dalvik, que permite a cada aplicativo do Android executar em seu próprio processo, com a sua própria instância da máquina virtual Dalvik ( aplicações Android são compiladas em executáveis Dalvik). Dalvik é uma máquina virtual especializada projectada especificamente para o Android e otimizada para vários dispositivos móveis com memória e CPU limitada;
- **Application Framework** - Expõe os vários recursos do sistema operativo Android para que os desenvolvedores de aplicações possam utilizá-los em suas aplicações;
- **Application** - Nesta camada superior, irá encontrar as aplicações que acompanham o dispositivo Android (tais como telefone, contactos, navegador, etc.), bem como as aplicações que descarregar e instalar a partir do Android Market. Qualquer aplicação desenvolvida para este sistema está localizada nessa camada.

Para o desenvolvimento em Android, pode-se usar um Mac, um PC com Windows ou uma máquina Linux. Todas as ferramentas necessárias são gratuitas e podem ser encontradas na Web. Ferramentas para o desenvolvimento de aplicativos Android são o **IDE Eclipse**, **SDK do Android**, e os **ADT**.

O Android SDK contém um depurador, bibliotecas, um emulador, documentação, exemplos de código e tutoriais. O SDK do Android faz uso do **Java SE Development Kit (JDK)**. Assim, o computador deve ter o JDK instalado.

O primeiro passo para o desenvolvimento de todas as aplicações é a obtenção do ambiente de desenvolvimento integrado (IDE). No caso do Android, o IDE Eclipse é recomendado como uma aplicação de desenvolvimento de software multi-linguagem, num ambiente caracterizado por um extensível sistema de plug-ins. Pode ser usado para desenvolver diversos tipos de aplicações, utilizando linguagens como Java, Ada, C, C ++, COBOL, Python, etc.

### 3.6.2 A Plataforma Eclipse

A plataforma Eclipse é uma plataforma para a integração de ferramentas de desenvolvimento open-source e possui uma arquitectura extensível baseada na utilização e desenvolvimento de plug-ins. Apresenta um conjunto de serviços e estruturas que representam os recursos mais comuns utilizados pela maioria dos desenvolvedores de ferramentas. Com a característica de trabalhar em uma plataforma aberta, onde novos recursos podem ser agregados a qualquer momento, a plataforma permite a interacção dos desenvolvedores com a plataforma de desenvolvimento de forma directa e intuitiva [37].

#### 3.6.2.1 Arquitectura

A Figura 3.7 mostra a estrutura básica da plataforma Eclipse. Fundamentalmente, o eclipse é uma estrutura voltada para plug-ins. Outros aplicativos podem actuar dentro desta estrutura básica para criar uma aplicação melhor e que cumpra com as necessidades do utilizador. Excepto ao pequeno núcleo do runtime, tudo em Eclipse é implementado como plug-ins. A Figura 3.8 mostra a representação da plataforma em camadas. De seguida apresenta-se a descrição de cada um dos seus componentes.

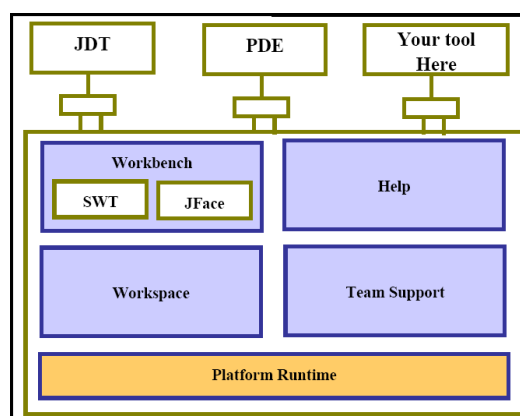


Figura 3.7: Arquitectura do Eclipse [37]

#### 3.6.2.2 Workspace do Eclipse

Responsável por administrar recursos do utilizador, que são organizados dentro de um ou mais projectos. As várias ferramentas “ligadas” ao IDE através dos plugins operam no ficheiros do

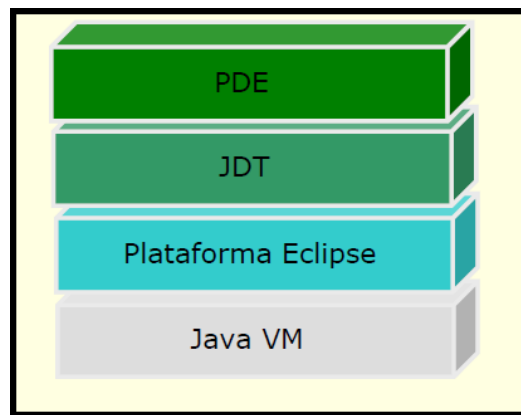


Figura 3.8: Representação da Plataforma em Camadas [38]

workspace do utilizar. O workspace consiste em um ou mais projectos onde cada projecto corresponde a uma pasta especificada pelo utilizador. Com o objectivo de minimizar a perda accidental de ficheiros o workspace contém um mecanismo que permite fazer "undo"ao workspace de modo a recuperar os ficheiros pretendidos. O workspace mantém também mecanismos que permitem guardar desde mensagens do compilador à informação sobre os breakpoints do debugger.

### 3.6.2.3 O Workbench

O workbench fornece a estrutura com a qual as ferramentas interagem com o utilizador, a janela que o utilizador “vê” quando a plataforma iniciada é o workbench; A sua API depende da API do JFace e da API do SWT. O workbench é constituído por editores, vistas e perspectivas. Os editores permitem ao utilizador abrir , editar e gravar objectos enquanto que as vistas providenciam informação sobre o documento a ser editado. São as perspectivas que definem o modo como os editores e as vistas são dispostas no ecrã.

As ferramentas de construção de componentes da interface gráfica do utilizador (GUI) podem ser divididas em: **O SWT (Standard Widget Toolkit)** e **Jface** . O SWT é um conjunto de ferramentas que possibilita a construção de uma aplicação independente da GUI do sistema de janelas do sistema operativo. Logo permite obter o mesmo aspecto visual tanto em máquinas que utilizem o Microsoft Windows como em sistemas que utilizem o Linux. O Jface é também uma ferramenta para controlar funções comuns da GUI. Tal como o SWT, é independente do sistemas de janelas tanto a sua API como a sua implementação. O Jface foi desenvolvido para funcionar com ou sem o SWT.

### 3.6.2.4 Java Development Tooling (JDT)

O JTD é uma ferramenta que adiciona à plataforma Eclipse a capacidade de desenvolver aplicações em Java e basicamente constitui um IDE para desenvolver programas Java. O JTD é implementado através de um conjunto de plug-ins que permitem ao utilizador, entre outras possibilidades, ter os ficheiros .Java e .class organizados por pastas, ter os projectos organizados por pacotes ,



adiciona capacidades de um editor de Java comum, como por exemplo colorir as palavras-chave e a sintaxe, ver os estados dos threads e das stack frames, etc. Os plug-ins que constituem o JTD são divididos em dois grupos: os que interferem e os que não interferem com a GUI, o que permite a utilização da plataforma Eclipse em sistema que não sejam baseados em interfaces gráficas com o utilizador.

### 3.6.2.5 Ambiente de Desenvolvimento de Plug-ins (PDE)

O Plug-in Development Environment (PDE) oferece ferramentas para criar, desenvolver, testar, depurar, construir e implantar plug-ins Eclipse, fragmentos, características, sites de actualização e produtos Rich Client Platform (RCP). O PDE também fornece ferramentas OSGi, que o torna um ambiente ideal para a programação de componentes, e não apenas para o desenvolvimento plug-ins Eclipse [37]. Algumas contribuições do PDE ao desenvolvimento de plug-ins podem ser vistas a seguir:

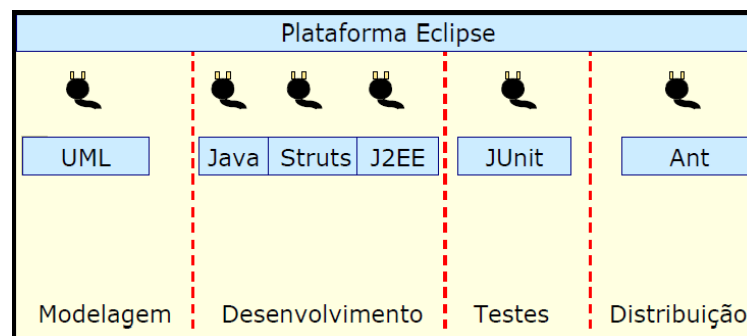


Figura 3.9: Cenário de Utilização de Plug-ins

### 3.6.2.6 Plataforma Runtime

É responsável por descobrir que plug-ins estão disponíveis na pasta de plug-ins do Eclipse para serem executados juntamente com o ambiente IDE.

### 3.6.3 Base de Dados

Como sugerido em [39], base de dados (BD ou DB, database) é uma entidade na qual é possível armazenar dados de maneira estruturada e com a menor redundância possível. Estes dados devem poder ser utilizados por programas, por vários utilizadores. Assim, a noção básica de dados é acoplada geralmente a uma rede, a fim de poder disponibilizar, conjuntamente, estas informações. Fala-se, geralmente, de sistema de informação para designar toda a estrutura que reúne os meios organizados para poder partilhar dados. Neste contexto a BD é uma colecção de informações organizadas de tal forma que um programa de computador pode seleccionar rapidamente os dados desejados. Assim, podemos pensar numa base de dados como um sistema de arquivamento electrónico.

As bases de dados tradicionais são organizadas por campos, registos e arquivos. Um campo é uma única parte de informação, um registo é um conjunto completo de campos, e um arquivo é uma colecção de registos. Um conceito alternativo em desenho de base de dados é conhecido como hipertexto. Numa base de dados Hypertext, qualquer objecto, seja ele um pedaço de texto, uma imagem ou um filme, pode ser ligado a qualquer outro objecto. Bases de dados de hipertexto são particularmente úteis para organizar grandes quantidades de informações díspares, porém, tais BD não são apropriados para análise numérica.

### 3.6.3.1 Gestão da Base de dados

Para obter informações duma base de dados, precisamos de um sistema de gestão da base de dados (SGBD). Como referido em [39], SGBD é uma colecção de programas que permite inserir, organizar e seleccionar dados numa base de dados. Tem-se notado porém que cada vez mais, o termo base de dados tem sido utilizado como sinónimo para sistema de gestão de base de dados. Algumas responsabilidades específicas do SGBD:

- permitir o acesso aos dados de maneira simples.
- autorizar um acesso às informações a múltiplos utilizadores
- manipular os dados presentes no bases de dados (inserção, supressão, modificação)

Os sistemas de gestão de bases de dados mais populares são os seguintes :

- Microsoft SQL server
- Oracle
- MySQL
- PostgreSQL

### 3.6.3.2 PostgreSQL

O PostgreSQL é uma SGBD relacional e orientado a objectos. É um software de livre distribuição e tem seu código fonte aberto. Oferece suporte a SQL de acordo com os padrões SQL92 / SQL99. Em termos de recurso pode ser comparado às melhores bases de dados comerciais existentes, sendo inclusive superior em alguns aspectos, como ao introduzir conceitos do modelo objecto - relacional que hoje estão disponíveis em algumas bases de dados comerciais. Arquitectura Básica do PostgreSQL é importante que haja a compreensão de sua arquitectura básica. Quando se abre uma sessão do Postgres, 3 processos são abertos:

- um processo daemon (*postmaster*);
- a aplicação do cliente (por exemplo, sua aplicação em PHP);
- um ou mais servidores da base de dados (processo *postgres*).

Um único processo *postmaster* gere as bases de dados existentes numa máquina. As aplicações do cliente(*frontend*) que desejam ligar determinada base de dados fazem chamadas a uma requisição do utilizador pela rede para o processo *postmaster*, que cria um novo processo-servidor (*backend*) e liga o processo-cliente ao servidor (*frontend*) e (*backend*) se comunicam sem a intervenção do *postmaster*. Sendo assim o *postmaster* o gestor das ligações do PostgreSQL. Agora faremos uma comparação entre o PostgreSQL e o MySQL ilustrada na Tabela 3.2 por serem soluções de código fonte aberta e ambas muito utilizadas em soluções de plataforma WEB como PHP e em sistemas como o Linux.

Tabela 3.2: comparação entre PostgreSQL e MySQL

	MySQL	PostgreSQL
Transações	Não	Sim
Lock linha	Não	Sim
Constraints	Não	Parcial
Programável	Parcial	Sim
password	Sim	Sim
Failsafe	Não	Sim
Hotback	Não	Não

- Transações: confirmação ou cancelamento de operações realizadas;
- Lock de linha: actualização bloqueia apenas as linhas a serem actualizadas;
- Constraints: chave primária e chave estrangeira para tabelas da base de dados;
- Programável: permite criar triggers(por exemplo);
- Password: validações de utilizador e password;
- Fail safe: segurança contra falhas(exemplo desligamento repentino do sistema);
- Hot backup: verifica se é possível realizar uma cópia consistente da base de dados enquanto transacções são efectuadas;

Apesar de tudo o MySQL tem a vantagem de possuir uma velocidade de acesso maior para grandes bases de dados. No entanto, as características de travamento de linhas, transacções e segurança contra falhas fazem do PostgreSQL uma opção mais segura e confiável.

### 3.6.3.3 Open Data Base Connectivity (ODBC)

Trata-se de um formato definido pela Microsoft que permite a comunicação entre clientes de base de dados e as Base de dados do mercado. O gestor de ODBC está presente nos sistemas Windows. No entanto, existem implementações em outras plataformas, incluindo as plataformas Unix e Linux. ODBC permite interrogar qualquer base de dados SQL. As bases de dados ou sistema de gestão de base de dados devem compreender as interrogações, e o aplicativo deve

enviar consultas em formato padrão ODBC. Embora o ODBC constitua uma interface com bases de dados, independente do SGBD, esta tecnologia ainda é uma solução proprietária da Microsoft. Isso se traduz por uma dependência de plataforma [40].

#### 3.6.3.4 Java DataBase Connectivity (JDBC )

Segundo autores em [41] JDBC é um conjunto de especificação de interfaces e classes em Java que padronizam a ligação com uma base de dados. É inspirado no bem sucedido padrão Microsoft de acesso a base de dados, ODBC, porém tem a vantagem de ser multi-plataforma. Além da independência da plataforma, Java também visa obter independência de base de dados. Isto significa que trocando a base de dados, espera-se pouca alteração na aplicação.

JDBC permite a utilização de caminhos alternativos para ligação a base de dados, ou seja podemos escolher diferentes drivers de diferentes tipos:

- Tipo 1: ligação através de uma ponte jdbc - odbc. Levando em consideração que a ponte jdbc-odbc já vem incorporada ao JDK, e juntamente com o driver ODBC , esta opção se torna a mais fácil de se encontrar;
- Tipo 2: o driver é obtido a partir de uma API nativa. Isto significa que o driver Java faz chamadas nativas a C ou C++ para subrotinas definidas pelo fornecedor da base de dados. Esta alternativa exige a instalação de software cliente;
- Tipo 3: através de um driver específico jdbc. Esta é a melhor opção, uma vez que ganha em desempenho e também é a opção multi-plataforma.

A JDBC utiliza a classe *java.sql.DriverManager* e duas interfaces *java.sql.Driver* e *java.sql.Connection*, para se conectar a uma base de dados:

- *java.sql.Driver*: proporciona à JDBC um ponto de partida para ligação à base de dados respondendo às requisições de ligação de *DriverManager* e fornecendo informações sobre a implementação em questão;
- *java.sql.DriverManager*: sua principal responsabilidade é manter uma lista de implementações de driver e apresentar a uma aplicação uma implementação que corresponde a uma URL requisitada;
- *java.sql.Connection*: usa-se a classe *Connection* para enviar uma série de dados de instruções SQL à base de dados e controlar o registo ou a interrupção dessas instruções.

#### 3.6.3.5 Os níveis ANSI/SPARC

Segundo autores em [39], a arquitectura ANSI/SPARC define níveis de abstracção para um sistema de gestão de bases de dados que são brevemente descritos a seguir:

- **Nível interno (ou físico) :** Os ficheiros são guardados em suporte de armazenamento informático e, a partir daí são manipulados pelo SGBD em execução no computador;

- **Nível conceptual** : chamado também MCD (modelo conceptual dos dados) ou MLD (modelo lógico dos dados). Define a organização das informações em tabelas de relacionamentos;
- **Nível externo** : define as vistas dos utilizadores, ou seja, como os dados são efectivamente visualizados para posterior utilização.

### 3.6.3.6 As características de um SGBD

De acordo com autores em [39] a arquitectura a três níveis definida pelo padrão ANSI/SPARC permite ter uma independência entre os dados e os tratamentos. Geralmente, um SGBD deve ter as características seguintes:

- **Independência física** : o nível físico pode ser alterado independentemente do nível conceptual. Isso significa que todos os aspectos materiais da base de dados não aparecem ao utilizador; trata-se simplesmente de uma estrutura transparente de representação das informações;
- **Independência lógica** : o nível conceptual deve poder ser alterado sem pôr em causa o nível físico, quer dizer que o administrador da base deve poder fazê-la evoluir sem necessariamente envolver os utilizadores;
- **Maneabilidade** : pessoas que não conhecem a base de dados devem ser capazes de fazer o seu pedido sem fazer referência a elementos técnicos da base de dados;
- **Rapidez dos acessos** : o sistema deve poder fornecer as respostas aos pedidos o mais depressa possível, o que implica algoritmos de pesquisa e consulta rápidos;
- **Administração centralizada** : o SGBD deve permitir ao administrador poder manipular os dados, inserir elementos, verificar a sua integridade de maneira centralizada;
- **Limitação da redundância** : o SGBD deve poder evitar, na medida do possível, informações redundantes, a fim de evitar por um lado um desperdício do espaço de memória, mas também erros;
- **Verificação da integridade** : os dados devem ser coerentes entre eles, ainda mais quando os elementos fazem referência a outros, em que estes últimos devem estar presentes;
- **Partilha dos dados** : o SGBD deve permitir o acesso simultâneo à base de dados por vários utilizadores;
- **Segurança dos dados** : o SGBD deve apresentar mecanismos que permitam gerir os direitos de acesso aos dados de acordo com os utilizadores.

### 3.6.3.7 Os diferentes modelos de bases de dados

As bases de dados apareceram no fim dos anos 60, numa época em que a necessidade de um sistema de gestão da informação flexível se fazia sentir. Existem cinco modelos de SGBD, diferenciados de acordo com a representação dos dados, que incluem :

- **o modelo hierárquico** : os dados são classificados hierarquicamente, de acordo com uma estrutura descendente. Este modelo utiliza apontadores entre os diferentes registos. Trata-se do primeiro modelo de SGBD ;
- **o modelo rede** : como o modelo hierárquico, este modelo utiliza apontadores para os registos. Contudo, a estrutura já não é necessariamente descendente;
- **o modelo relacional (SGBDR, Sistema de gestão de bases de dados relacionais)** : os dados são registados em quadros a duas dimensões (linhas e colunas). A manipulação destes dados faz-se de acordo com a teoria matemática das relações, conhecida como Álgebra Relacional;
- **o modelo dedutivo** : os dados são representados sob a forma de tabela, mas a sua manipulação faz-se por cálculo de predicados;
- **o modelo de objectos (SGBDO, Sistema de gestão de bases de dados objecto)**: os dados são armazenados sob a forma de objectos, quer dizer, de estruturas chamadas classes que apresentam dados membros. Os campos são instâncias destas classes, ou seja, objectos.

### 3.6.3.8 Diagrama do Modelo de Entidade e Associação

O **Diagrama do Modelo EA** é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstracção. Ele é a principal representação do Modelo de Entidades e Relação. É usado para representar o modelo conceitual do negócio. Não confundir com modelo relacional, que representam as tabelas, atributos e relações materializadas no base de dados.

### 3.6.3.9 Modelo Relacional para SGBD

O **modelo relacional** para gestão de bases de dados é um modelo de dados baseado em lógica e na teoria de conjuntos. Em definição simplificada, o modelo baseia-se em dois conceitos: conceito de entidade e relação. Uma **entidade** é um elemento caracterizado pelos dados que são recolhidos na sua identificação vulgarmente designado por tabela. Na construção da tabela identificam-se os dados da entidade a atribuição de valores a uma entidade constrói um registo da tabela. No fim dos anos 90, as bases relacionais tornaram-se as bases de dados mais comuns, com cerca de três quartos das bases de dados utilizadas na maioria das aplicações [39].

## **3.7 Sumário**

Ao longo deste capítulo foram apresentados as interfaces de comunicação e a visão de funcionamento de todo o sistema desenvolvido neste trabalho. Foram feitas a descrição dos requisitos do sistema e definição das suas restrições, sendo que algumas das restrições tecnológicas como o sistema operativo Android, plataforma Eclipse e sistema de gestão de base de dados foram analisadas de uma forma mais detalhada. Por fim foi abordada representação do sistema, apresentado com recurso a diagramas de casos de utilização UML das principais funcionalidades associadas à ferramenta desenvolvida.





## Capítulo 4

# Implementação e Avaliação de Resultados

Neste capítulo são descritos os detalhes da implementação do protótipo cujos requisitos e casos de utilização foram especificados no capítulo anterior. A primeira parte da implementação é dedicada à aplicação Web. Na secção 4.1.1 é apresentado as ferramentas que foram utilizados para a respectiva implementação, na secção 4.1.2 é feita uma descrição das permissões dos utilizadores do sistema. Nas secções seguintes são apresentados um diagrama do modelo entidade e associação as respectivas entidades e associações, o modelo relacional e por último as respectivas descrições dos módulos. A segunda parte é dedicada a aplicação Móvel.

### 4.1 Implementação da Base de Dados e Aplicação Web

#### 4.1.1 Ferramentas Utilizada

Para o desenvolvimento do sistema foram utilizadas as seguintes ferramentas:

- **PostgreSQL** é um sistema gestor de base de dados objecto relacional (SGBDOR), desenvolvido como um projecto de open source;
- **PHP (Hypertext Preprocessor)** é uma linguagem de script open source de uso geral, muito utilizada e especialmente adequada para o desenvolvimento de aplicações Web embutidas dentro do HTML;
- **HTML (HyperText Markup Language)** é uma linguagem de marcação utilizada para produzir páginas Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto do "casamento" dos padrões HyTime e SGML. HTML é a construção de blocos básicos de páginas Web. HyTime é um padrão para a representação estruturada de hipermédia e conteúdo baseado em tempo. Um documento é visto como um conjunto de

eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), ligados por hiperligações. O padrão é independente de outros padrões de processamento de texto em geral. SGML é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objects e para descrever as ligações;

- **Apache** O servidor Apache (ou Servidor HTTP Apache, Apache HTTP Server, ou simplesmente Apache) é o mais bem sucedido servidor Web livre.

#### 4.1.2 Descrição de Permissões

No que respeita aos tipos dos utilizadores foram definidas as seguintes permissões:

- **Utilizador (UTI):** Permissão associada a um pessoa que não está sendo acompanhado por um médico;
- **Utilizador (PUTI):** Permissão associada a um utilizador. No entanto, a informação; marcada com este nível está apenas disponível para o utilizador a quem a informação diz respeito. Por exemplo, no acto de ver histórico Medições, um utilizador terá acesso apenas às medições associadas ao seu id;
- **Paciente (PAC):** Permissão associada a um paciente registado no sistema de informação;
- **Próprio Paciente (PPAC):** A mesma situação que em PUTI, agora aplicada a um paciente;
- **Médico (MED):** Permissão associada a um médico registado no sistema de informação;
- **Próprio médico (PMED):** A mesma situação que em PUTI, agora aplicada a um médico;
- **Administrador (ADM):** Permissão associada a um gestor do sistema de informação. Tem licenças para interagir ao máximo nível com a base de dados;

### 4.1.3 Modelos de Entidade e Associação

A figura 4.1 ilustra a constituição do diagrama do modelo EA.

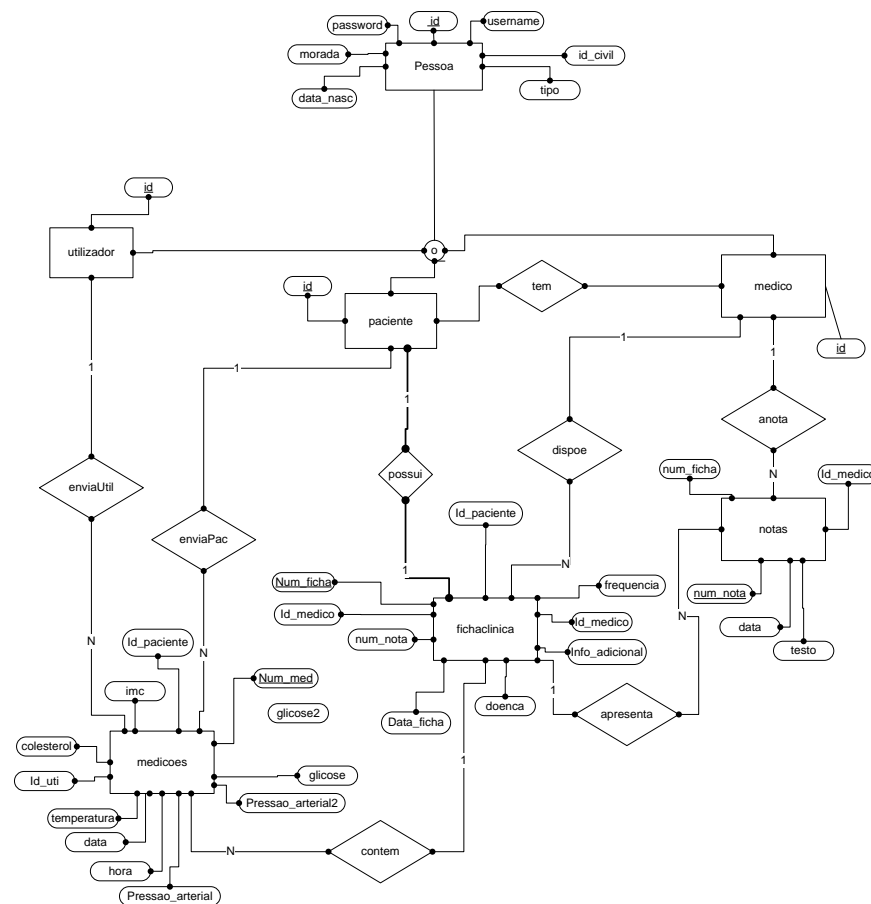


Figura 4.1: Diagrama do Modelo Entidade e Associação

#### 4.1.3.1 ENTIDADES

As entidades do módulo, com os seus respectivos atributos, são identificados abaixo:

peessoa(id,nome,data\_nasc,id\_civil,morada,telefone,username,password,tipo)

medico (id)

publico (id)

paciente (id , id\_medico, num\_ficha)

fichaclinica(num\_ficha, data\_ficha , id\_medico, id\_paciente, frequência, doença, info\_adicional, estado)

medicoes( num\_med, data, hora, num\_ficha, imc, temperatura, glicose, colesterol,pressao\_arteial, id\_paciente,id, pressao\_arteial2,glicose2 )

notas(num\_notas,testo,data\_notas)

#### 4.1.3.2 ASSOCIAÇÕES

A Tabela 4.1, a seguir, apresenta as associações do módulo, indicando a respectiva cardinalidade e participação.

Associações	Cardinalidade	Participações
apresenta(fichaclinica, medicoes)	1:N	total/parcia
dispoe(medico, fichaclinica)	1:N	total/parcia
possui(paciente, fichaclinica)	1:1	total/total
enviaPac(paciente, medicoes)	1:N	total/parcia
enviaPub(paciente, medicoes)	1:N	total/parcia
tem(medico, paciente)	1:N	total/parcia
anota(medico, notas)	1:N	total/parcia

Tabela 4.1: Tabela de Associações

#### 4.1.4 Modelos Relacional

A Tabela 4.2 determina o modo como cada registo de cada tabela se associa a registos de outras tabelas.

<b>fichaclinica</b>	num_ficha	infoAdional	frequencia	data_ficha	estado	doenca	# id_medico
<b>pessoa</b>	id	password	data_nasc	id_civil	morada	nome	username
<b>medicoes</b>	num_med	data	hora	... d_uti ...	glicose	id_pac	# num_icha
<b>notas</b>	# num_notas	# id_medico	#num_ficha				
<b>paciente</b>	# id	# id_medico	#num_ficha				

Tabela 4.2: Modelo Relacional

### 4.1.5 Descrição dos Módulos do Sistema

Nesta secção descrevem-se vários módulos definidos para a arquitectura da aplicação Web. Esses módulos são: **Login**, **Medicoes**, **Ficha Clínica** e **Perfil**. Cada um desses módulos mencionados irão ser descritas em conformidade com a seguinte estrutura:

- Serão identificadas três tipos de páginas por cores:
  - Azul:** Listagens e Visualizações
  - Verde:** Formulários
  - Laranja:** Acções
- Em cada página serão apresentadas três níveis de informação:
  - 1º Nível Nome da pagina
  - 2º NiveIndicação das tabelas e/ou vistas
  - 3º Nive Restrições de acesso a pagina

#### 4.1.5.1 Módulo Login

Trata da funcionalidade de autenticação e desautenticação no sistema. O ponto de partida do diagrama é a página «paginaComPermissao», uma página qualquer acedida por um utilizador do sistema com acesso à mesma. Daí, o utilizador tentará aceder a uma outra, «paginaSeguinte», à qual terá, ou não, acesso. A interpretação deste diagrama deve ser independente do nível de permissão atribuído a esse mesmo utilizador.

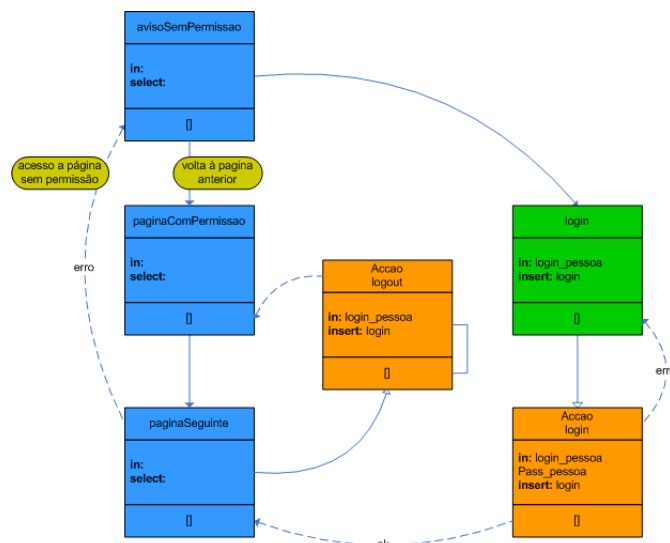


Figura 4.2: Diagrama de Arquitectura do Módulo Login

Página	Dados apresentados	Parâmetros recebidos
Login	username, password	username, password
Acção login		username, password

Tabela 4.3: Tabela Descritiva do Módulo LOGIN

#### 4.1.5.2 Registo

Este é o módulo que dá acesso aos utilizadores entrarem no sistema. Na primeira fase o utilizador escolhe o tipo de utilizador, ou seja, pode ser um Paciente, Médico ou uma pessoa que não tem um médico associado. Os pacientes só podem realizar o registo se tiverem um médico no sistema que já tenha criado uma ficha clínica para o efeito. Na segunda fase então preenche os dados solicitados e envia para ser validado. Depois de validados então estes podem efectuar os respectivos login.

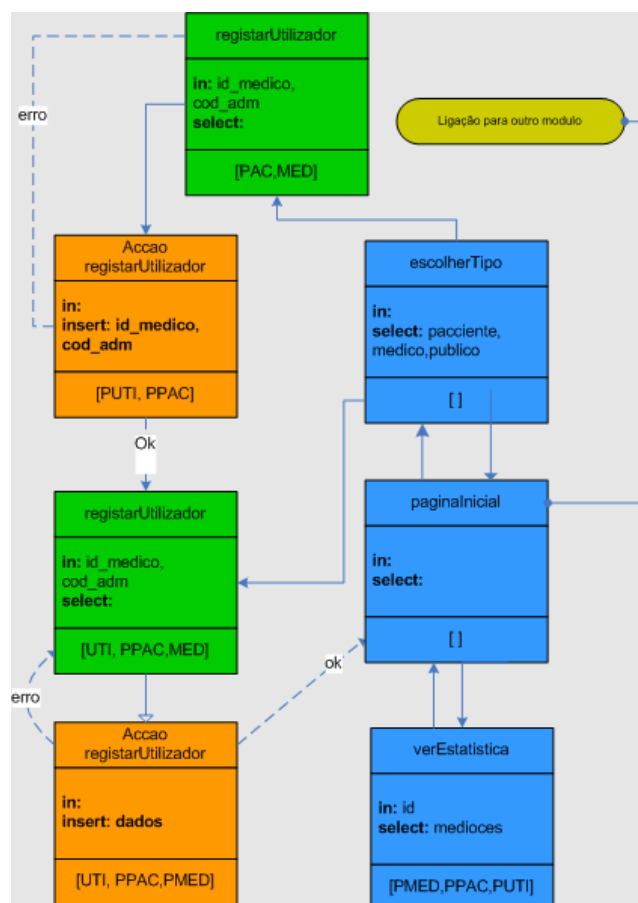


Figura 4.3: Diagrama de Arquitectura do Módulo Registo

Página	Dados apresentados	Parâmetros recebidos
escolhertipo	tipo	cod_adm
registro		nome, password, username morada, id_medico, num_ficha telefone, id_civil

Tabela 4.4: Tabela Descritiva do Módulo Registro

#### 4.1.5.3 Módulo Perfil

Trata das funcionalidades do perfil. Neste módulo o utilizador pode ver os seu dados pessoais e podem alterar alguns desses dados, isto porque existem dados que devido à sua unicidade não podem ser alterados.

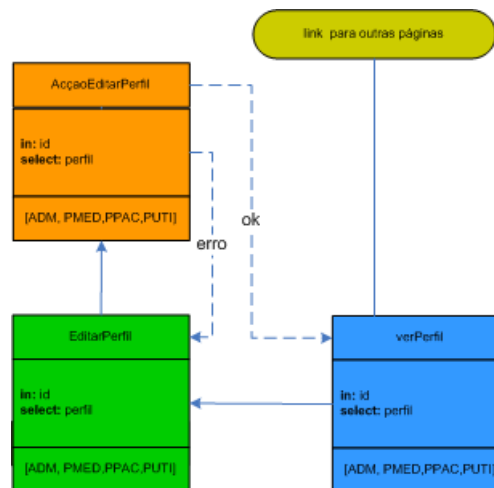


Figura 4.4: Diagrama de Arquitetura do Módulo Perfil

Página	Dados apresentados	Parâmetros recebidos
verperfil	nome, morada telefone, id_civil	id_pessoa
editarperfil	nome, morada telefone, id_civil	id_pessoa

Tabela 4.5: Tabela Descritiva do Módulo perfil

#### 4.1.5.4 Módulo Ficha Clínica

Trata das funcionalidades associadas a Ficha Clínica. Neste módulo os pacientes podem ver todo o conteúdo descrito na ficha e podem ver alertas. Também os médicos tem acesso a esses parâmetros e podem ver todas as fichas dos seus pacientes e também podem criar mais fichas clínicas.

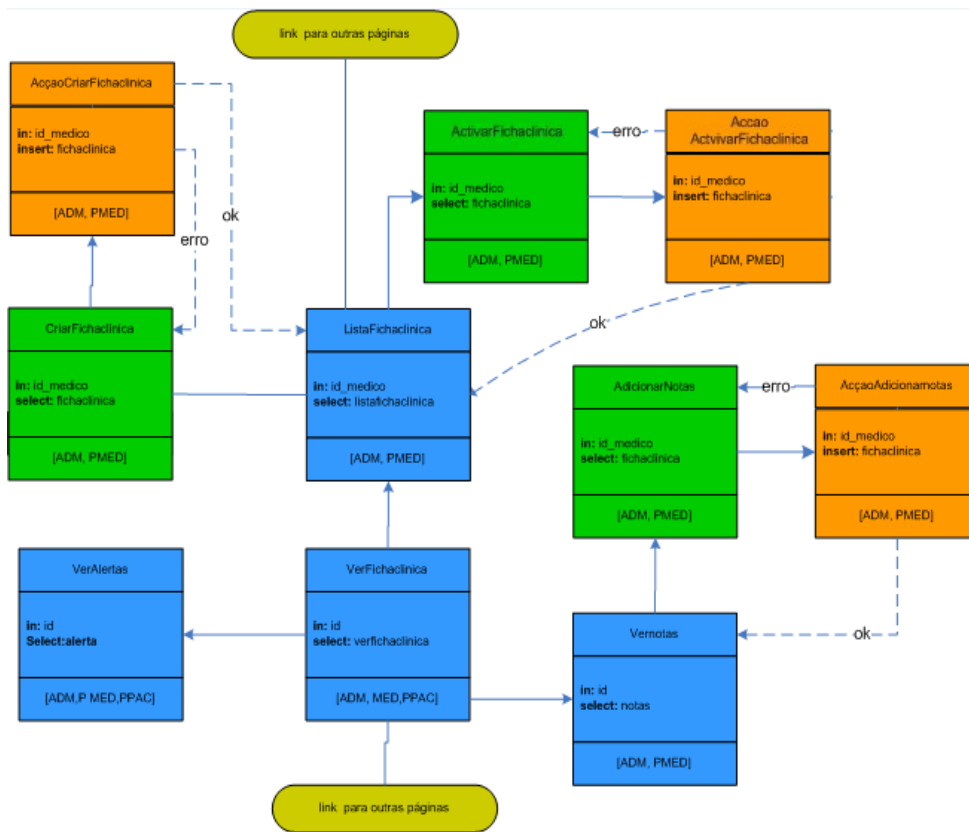


Figura 4.5: Diagrama de Arquitetura do Módulo Ficha Clínica

Página	Dados apresentados	Parâmetros recebidos
ListaFichaclinica	num_Ficha, data info_adicional,doença nome »pessoa	Um dos seguintes: id_medico, idp_aciente
verAlertas	medicoes diagnostico	id_medico idPaciente id_utilizador

Tabela 4.6: Tabela Descritiva do Módulo Ficha Clínica



#### 4.1.5.5 Módulo Medições

Trata das funcionalidades associadas ao tratamento de dados de Medições. Neste módulo pode-se enviar as medições, ver os históricos das mesmas e ainda ver as médias das medições. Este módulo pode ser acessado pelos médicos, pelos pacientes e utilizadores que não tenham médicos associados. Porém os médicos não tem acesso à funcionalidade enviar medições.

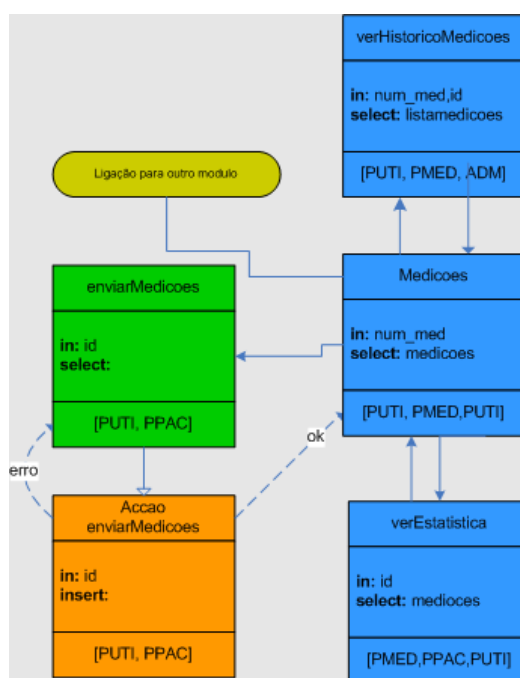


Figura 4.6: Diagrama de Arquitectura do Módulo Medições

Página	Dados apresentados	Parâmetros recebidos
historicomedicoes	num_med, tipo_medicoes data, hora	Um dos seguintes: id_pessoa
enviarmedicoes	tipo_medicoes, num_ficha	id_paciente, id_publico
verestatistica	tipo_medicoes, medias	id_pessoa

Tabela 4.7: Tabela Descritiva do Módulo Medições

#### 4.1.6 Definição de Vistas

Nesta secção estarão descritas algumas das vistas definidas para a base de dados desenvolvida. Será incluído o seu código SQL para melhor ilustração.

- **listapacientes** Pretende-se gerar uma vista que contém todos os pacientes com o respectivo médico associado.

```
CREATE VIEW listapaciente AS
```

```

SELECT pessoa.id, pessoa.nome, pessoa.data_nasc, pessoa.morada, pessoa.id_civil, pes-
sua.telefone, pessoa.password, pessoa.login, ficha_clinica.id_medico
FROM pessoa
JOIN paciente USING (id)
JOIN ficha_clinica ON ficha_clinica.id_paciente = paciente.id
ORDER BY paciente.id;

```

- **listamedicoes** Pretende-se gerar uma vista que contém todas as medições enviadas pelos pacientes.

```

CREATE VIEW listamedicoes AS
SELECT pessoa.id, pessoa.nome, medicoes.num_med, medicoes.imc AS peso, medicoes.temperatura,
medicoes.glicose, medicoes.colesterol, medicoes.pressao_artorial, medicoes.data, medicoes.hora,
medicoes.pressao_artorial2, medicoes.glicose2
FROM pessoa
JOIN paciente ON pessoa.id = paciente.id
JOIN medicoes ON medicoes.id_paciente = paciente.id
ORDER BY medicoes.num_med;

```

- **fichaclinica** Pretende-se gerar uma vista que contém todas as fichas clinicas dos pacientes.

```

CREATE VIEW fichaclinica AS
SELECT ficha_clinica.num_ficha, ficha_clinica.data_ficha, ficha_clinica.id_paciente, ficha_clinica.doenca,
ficha_clinica.frequencia, ficha_clinica.info_adicional, pessoa.nome, ficha_clinica.id_medico,
ficha_clinica.estado
FROM pessoa
JOIN ficha_clinica ON ficha_clinica.id_medico = pessoa.id;

```

#### 4.1.7 Descrição de páginas

Nesta secção descreve-se textualmente algumas das páginas apresentadas nos diagramas de arquitectura. Em todas as páginas estarão disponíveis caixas para autenticação no sistema e o retorno à página inicial assim como ligações a outras páginas.

- **fichaClinica** é a página principal dos pacientes e o médico também tem acesso a ela. Permite ver informações do paciente como o tipo de doença que este padece, o tipo de medições que o médico solicitou, as respectivas frequências de medições e o nome do médico e/ou paciente consoante o utilizador, se for médico então este vê o nome do paciente e vice-versa. Esta página ainda permite ver o estado da ficha o seu número e data em que foi gerada. Pode-se ver as alertas e notas no caso do médico;
- **historicoMedicoes** nesta página pode-se ver todas as medições realizadas pelo paciente e as respectivas datas e horas associadas. Tanto o paciente como o medico deste podem a visualizar;

- **enviarMedicoes** esta página contém sete tipos de medições e permite aos utilizadores com permissões de acesso escolher o(s) tipo(s) de medição(ões) a enviar. Os valores das medições são números reais e para separar as casas decimais deve-se utilizar ponto ' . '.
- **estatísticas** esta página mostra as médias das medições realizadas;
- **meuPerfil** é a página onde contém os dados pessoais dos utilizadores na qual estes podem fazer as respectivas actualizações dos dados;
- **listaPacientes** esta é a página principal dos médicos e apresentará a lista de todos os seus pacientes com os respectivos ID . Nesta página os médicos podem visualizar as fichas dos pacientes e visualizar os históricos de medições respectivos. Esta página é visualizada apenas pelo médico em causa;
- **listaFichaclinica** esta página contém a lista de todas as fichas clínicas criadas pelos médicos, ou seja, tem tanto as fichas que estão activas como as que não estão. Tanto umas como outras podem ser activadas ou desactivadas;
- **registarUtilizador** esta pagina permite a entrada de todos os utilizadores do sistema. Apresenta um conjunto de dados e todas elas são de preenchimento obrigatório.

## 4.2 Implementação da Aplicação Móvel

Nesta secção falaremos sobre o protocolo de comunicação TCP/IP e vamos mostrar os formatos das tramas enviadas do cliente para o servidor e vice-versa detalhando a forma como estas são tratadas ao longo dos seus percursos. Serão ilustrados alguns exemplos deste procedimento. Serão descritos aspectos da fase de implementação e instalação, designadamente a estrutura e dependências de código fonte e de módulos executáveis, diagramas de classes e componentes, assim como a sua respectiva instalação nas diferentes plataformas computacionais subjacentes. Para tal iremos necessitar das seguintes ferramentas :

- Eclipse;
- Java Runtime Environment ( JRE);
- Ferramentas de Desenvolvimento Android (ADT);
- Postgres JDBC Driver;
- Android SDK.

Eclipse é um IDE para programação Java; ADT (Ferramentas de Desenvolvimento Android) ao contrário, é um plugin para Eclipse que irá alargar a sua funcionalidade e permitir uma integração perfeita com as ferramentas necessárias para programar com Android. Finalmente temos

o emulador do Android, ou melhor, o SDK, que vai nos ajudar a testar nossas aplicações diretamente no nosso computador. Porém, para começar a desenvolver aplicações com Android no Eclipse, precisamos instalar um plug-in chamado Android Development Tools (ADT), que adiciona suporte integrado para projectos e ferramentas Android. Os procedimentos da instalação do SDK estão ilustradas em anexo.

#### 4.2.1 Visão geral

O sistema foi projectado como uma arquitectura Cliente - servidor, conforme esquematizado na Figura 4.7 Os sinais vitais do paciente são medidos pelos Pacientes e enviados do dispositivo móvel com a aplicação Android para ser guardados na base de dados e visualizados pelos responsáveis de saúde. A comunicação que ocorre entre o cliente e o servidor deve ser confiável. Ou seja, nenhum dado pode ser descartado e deve chegar no lado do cliente na mesma ordem em que o servidor enviou e vice-versa. Esta comunicação é feita através do protocolo TCP/IP. O servidor é responsável pela ligação à base de dados e fá-lo usando o Postgres JDBC Driver.

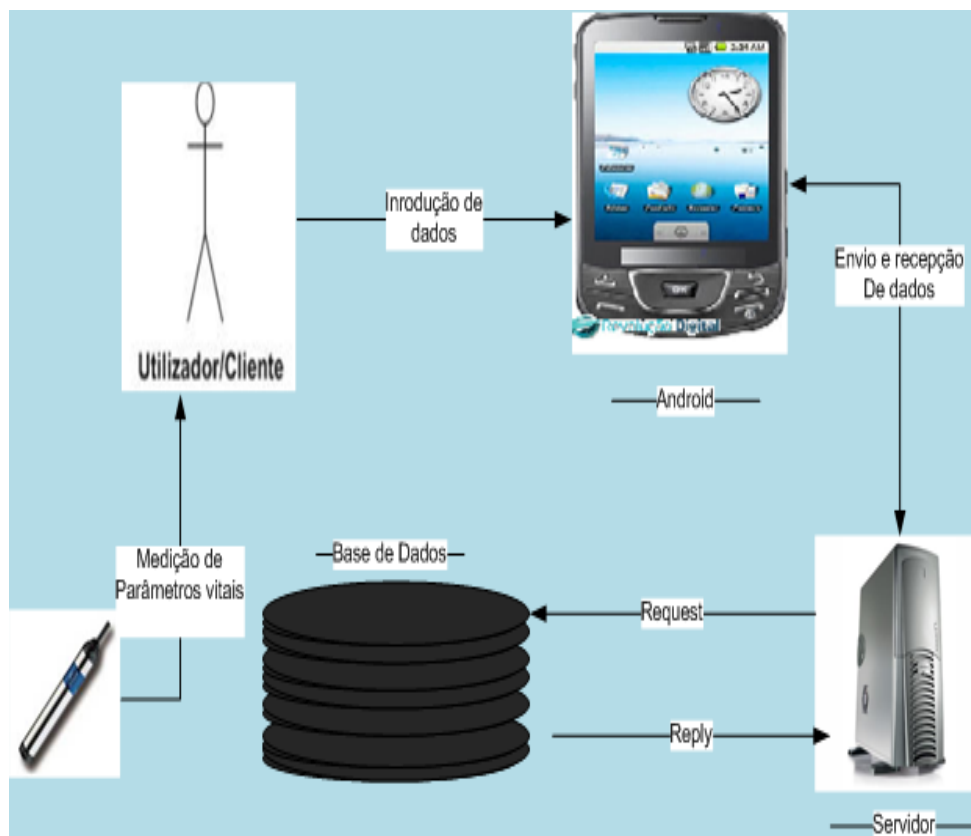


Figura 4.7: Arquitectura da Aplicação Móvel

### 4.2.2 Protocolo Comunicação TCP/IP

TCP oferece um confiável canal de comunicação ponto - a - ponto em que aplicações cliente - servidor comunicam-se uns com os outros na Internet. Para se comunicarem através de TCP, o programa cliente e o programa servidor estabelecem uma ligação um com o outro. Cada programa liga um socket à sua extremidade da ligação. O cliente e o servidor cada um lê e grava para o socket vinculado à ligação.

Um socket é um ponto final de um link de comunicação de duas vias entre dois programas em execução na rede. Classes Socket são usadas para representar a ligação entre um programa cliente e um programa servidor. O pacote java.net fornece duas classes - Socket e ServerSocket - que implementam o lado cliente da ligação e o lado servidor da ligação, respectivamente.

Normalmente, um servidor é executado em um computador específico e tem um soquete que está vinculado a um número de porta específico. O servidor apenas espera, ouvindo o socket para um cliente para fazer uma solicitação de ligação.

No lado do cliente, o cliente sabe o nome da máquina na qual o servidor está em execução e o número da porta onde o servidor está escutando. Para fazer uma solicitação de ligação, o cliente tenta encontrar o servidor na máquina do servidor e a porta. O cliente também precisa se identificar para o servidor para que ele se ligue a um número de porta local que irá utilizar durante esta ligação. Isso geralmente é atribuído pelo sistema.

Após a aceitação, o servidor recebe um novo socket ligado à mesma porta local e também tem o seu ponto de extremidade remoto definido como o endereço e a porta do cliente. Ele precisa de um novo socket para que ele possa continuar a escutar o socket original para solicitações de ligação atendendo às necessidades do cliente ligado.

### 4.2.3 Formato das Tramas e seus tratamentos

A trama enviada pelo cliente é um array de comprimento quatro em que a primeira posição contém o **ID** da trama, as duas posições seguintes contêm as **Tag1** e **Tag2** e a última posição contém o **Info**. **Info**, uma String que é uma Query para ser executada na base de dados. Enquanto que a enviada pelo servidor tem comprimento três, o ID, a **Tag1** e o **Info**, uma String que é o resultado da Query então executada na base de dados. Estas tramas estão ilustradas nas Tabelas 4.8 e 4.9, abaixo:

Tabela 4.8: Trama enviada pelo cliente

ID	Tag1	Tag2	Info
----	------	------	------

Tabela 4.9: Trama enviada pelo servidor

ID	Tag1	Info
----	------	------

Relativamente ao tratamento das tramas vamos mostrar dois exemplos, uma referente ao pedido de login e a outra sobre o envio de medições da parte de um paciente.

A trama enviada pelo cliente para o pedido de login tem esta informação no **Info** `"select * from pessoa where username='"+textOut.getText().toString()+"'and password='"+md5+"'"` e o formato que se segue:

Tabela 4.10: Trama enviada pelo cliente

0001	c.1	login	Info
------	-----	-------	------

Ao receber esta trama o Servidor verifica as duas tags e constata que se trata de um pedido login. Este faz o respectivo pedido à base de dados que lhe responde com uma das duas hipóteses: o resultado da query ou o erro da inexistência dos dados na tabela pessoa. O servidor envia uma de duas tramas ao cliente. A trama da Tabela 4.11 é enviada se o username e a password estiverem correctos.

Tabela 4.11: Trama enviada pelo servidor

0001	loginOK	Info
------	---------	------

O cliente por sua vez verifica a tag1 e envia uma mensagem de erro de entrada ou permite a entrada no sistema ao utilizador. Para o pedido de envio de medições efectuado pelo paciente a primeira trama enviada será para pedir o número de ficha clínica, uma vez que esta é necessária para inserção de dados na tabela medições da base de dados. Nesta trama o servidor usa tanto a

Tabela 4.12: Trama enviada pelo cliente

0101	num_ficha	envioMed1	Info
------	-----------	-----------	------

tag1 como a tag2 para saber como questionar a base de dados. Nas sequências seguintes as tramas são tratadas de forma idêntica à forma como foi tratada no caso de pedido de login.

#### 4.2.4 Arquitectura Física

O diagrama de distribuição de componentes apresentado na figura 4.8 reflecte a configuração dos nós de processamento e os seus componentes. A ideia é mostrar como foi delineada a arquitectura do sistema de software na perspectiva dos seus componentes digitais, demonstrando as suas dependências, e ainda identificar quais os componentes que são instalados em cada nó computacional.

Pode-se observar que o Servidor necessita do Java Run Enviroment (JRE) de modo a ser executado assim como o PostgreSQL JDBC Driver para ligação à base de dados.

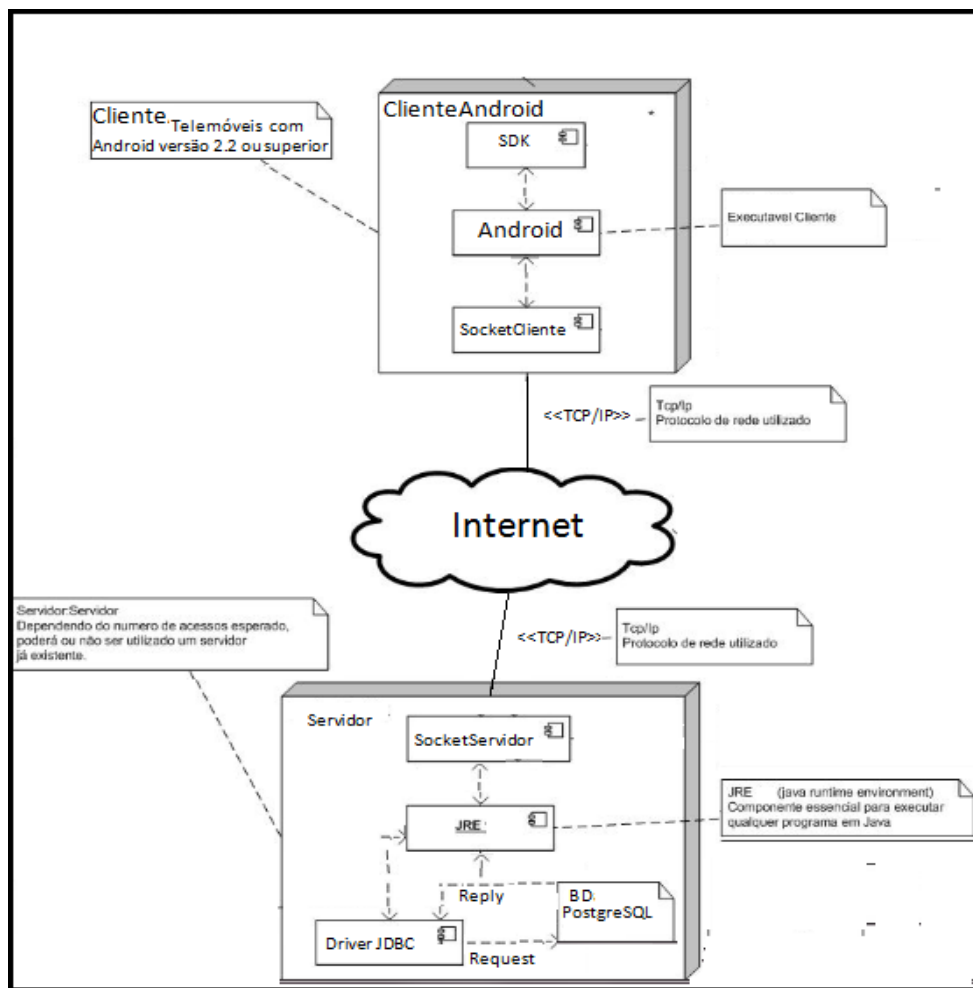


Figura 4.8: Diagrama de distribuição de componentes

#### 4.2.5 Arquitectura Lógica

Para desenvolver o sistema recorreremos à linguagem de programação orientada a objectos JAVA. Foram utilizadas algumas bibliotecas da distribuição da API do java e também bibliotecas Android. A Figura 4.9 ilustra as bibliotecas utilizadas, garantindo uma melhor compreensão.

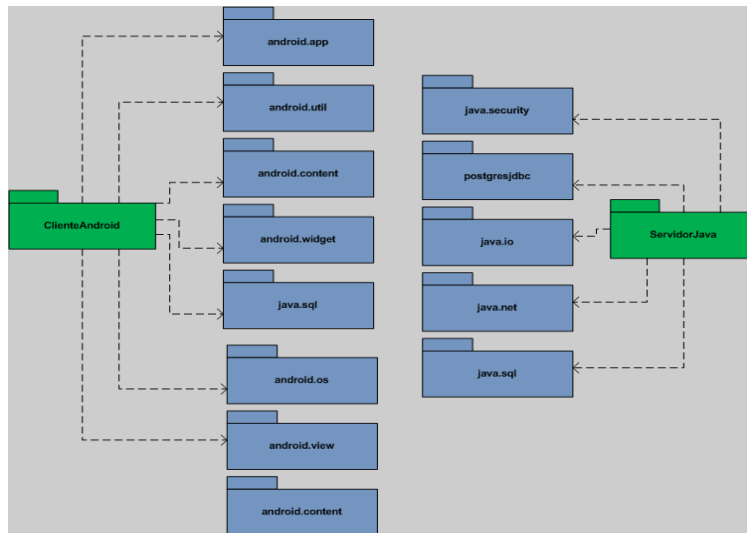


Figura 4.9: Diagrama de dependências dos pacotes de bibliotecas usados



## 4.3 Análise dos Resultados Obtidos

Ao longo de todo o trabalho desenvolvido e apresentado neste documento, os resultados obtidos a partir da aplicação web, aplicação móvel e da base de dados permitiram atingir os objectivos inicialmente propostos. Esta secção vai mostrar os resultados dos testes efectuados ao sistema no seu todo.

### 4.3.1 Resultados da Base de Dados

Nesta secção vamos mostrar algumas das tabelas e vistas criadas na base de dados. A figura 4.10 mostra-nos o conteúdo da tabela pessoa.

Ações	id	nome	data_nasc	morada	id_civil	telefone	password	username	tipo
<a href="#">Editar</a> <a href="#">Deletar</a>	1	Angelo da Silva Mendes	1990-12-12	rua da frente	126789888	967028830	d04500dbac51e00df4a6cc0657866b07	Angelo Mendes	medico
<a href="#">Editar</a> <a href="#">Deletar</a>	3	Armando Silva	1990-12-12	rua da frente	234567309	967899999	d04500dbac51e00df4a6cc0657866b07	Arm Silva	medico
<a href="#">Editar</a> <a href="#">Deletar</a>	15	Jailson Duete Martins Moreira	1981-06-13	costa cabral	234567309	963239908	81d09db52d040c20036dbd9313ad055	Jailson Moreira	paciente
<a href="#">Editar</a> <a href="#">Deletar</a>	16	Maria Furtado	1990-12-12	rua do medo	212333444	967028830	d04500dbac51e00df4a6cc0657866b07	Maria	qpessoa
<a href="#">Editar</a> <a href="#">Deletar</a>	18	Maria dos Santos Furtado	1990-12-12	rua da frente	666666666	967899999	d04500dbac51e00df4a6cc0657866b07	Maria Furtado	qpessoa
<a href="#">Editar</a> <a href="#">Deletar</a>	19	Octavio Francisco Semedo	1981-06-13	rua da miseria	6667889	963239908	d04500dbac51e00df4a6cc0657866b07	Octavio semedo	paciente
<a href="#">Editar</a> <a href="#">Deletar</a>	20	Carlos Tavares Martins	1990-12-12	rua da frente	212331444	967028830	d04500dbac51e00df4a6cc0657866b07	Carlos Martins	paciente
<a href="#">Editar</a> <a href="#">Deletar</a>	2	Antonio Semedo da Nole	1990-12-12	rua da crise	11123	967028830	d04500dbac51e00df4a6cc0657866b07	Antonio Nole	medico
<a href="#">Editar</a> <a href="#">Deletar</a>	21	Jose Jorge Borges	1990-12-12	rua da frente	98765789	967654334	d04500dbac51e00df4a6cc0657866b07	Jose Borges	paciente
<a href="#">Editar</a> <a href="#">Deletar</a>	22	Izequiel pereira Veiga	1981-06-13	rua da crise	673775647	967899999	d04500dbac51e00df4a6cc0657866b07	Izequiel Veiga	paciente
<a href="#">Editar</a> <a href="#">Deletar</a>	24	Patricio Lima	1990-12-12	rua da comedia	98099899	88987789	d04500dbac51e00df4a6cc0657866b07	Patricio Lima	paciente
<a href="#">Editar</a> <a href="#">Deletar</a>	23	Odaley Pereira	1980-12-12	rua da comedia	679779889	967899999	d04500dbac51e00df4a6cc0657866b07	Odaley Pereira	paciente

Figura 4.10: Tabela pessoa da base de dados

A figura 4.11 mostra-nos o conteúdo da tabela ficha clínica.

Ações	num_ficha	data_ficha	id_medico	id_paciente	frequencia	doenca	info_adicional	estado		
<a href="#">Editar</a> <a href="#">Deletar</a>	5	2011-05-12	3	16	peso semanal	pressão e glicose	3vezes por dia	Diabete	pressão e glicose e peso	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	6	2011-05-14	2	19	IMC semanal	pressão e glicose	3vezes por dia, co...	Diabete	todas	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	7	2011-05-15	2	20	IMC semanal	pressão e glicose	3vezes por dia, co...	Diabete	todas	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	8	2011-06-07	2	21	IMC semanal	Pressao e Glicose	3vezes ...	Diabete	todas	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	9	2011-06-07	2	22	IMC semanal	Pressao e Glicose	3vezes ...	Diabete	todas	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	10	2011-06-07	2	23	IMC semanal	Pressao e Glicose	3vezes ...	Diabete	todas	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	11	2011-06-07	2	24	IMC semanal	Pressao e Glicose	3vezes ...	Diabete	todas	ACTIVO
<a href="#">Editar</a> <a href="#">Deletar</a>	12	2011-06-15	2	NULL	Pressao: 3 vezes por dia Glicose: 3 vezes por ...			Diabete Milutus	Devido ao seu estado de saúde deve realizar as ...	INACTIVO

Figura 4.11: Tabela ficha clínica da base de dados

A figura 4.12 mostra-nos o conteúdo da tabela medições.

Ações	num_med	data	hora	num_ficha	imc	temperatura	glicose	colesterol	pressao_arterial	id_paciente	id	pressao_arterial2	glicose2
Editar Deletar	4	2011-05-12	12:20:22+01	5	7.8	NULL	NULL	NULL	NULL	15	NULL	NULL	NULL
Editar Deletar	7	2011-05-12	12:58:51+01	5	27.9	NULL	NULL	NULL	NULL	15	NULL	NULL	NULL
Editar Deletar	8	2011-05-12	13:02:07+01	5	45	NULL	NULL	NULL	NULL	15	NULL	NULL	NULL
Editar Deletar	5	2011-05-12	12:27:59+01	5	32.4	NULL	10.9	NULL	88	15	NULL	125.5	6.5
Editar Deletar	3	2011-05-12	10:24:33+01	5	23.5	NULL	6.5	NULL	82	15	NULL	124	8
Editar Deletar	6	2011-05-12	12:27:59+01	5	32.4	90	10.8	NULL	88	15	NULL	125.5	7
Editar Deletar	9	2011-05-14	13:09:29+01	6	25	37	10	8	125	19	NULL	84	8
Editar Deletar	10	2011-05-14	13:12:35+01	6	34.5	37.3	6.4	5.8	123.3	19	NULL	81.3	7.7
Editar Deletar	11	2011-05-14	13:12:35+01	6	34.5	37.3	6.4	5.8	123.3	19	NULL	81.3	7.7
Editar Deletar	12	2011-05-14	15:06:16+01	6	25	38.7	6.9	8.7	133.5	19	NULL	88.4	6.5
Editar Deletar	13	2011-05-14	15:06:16+01	6	25	38.7	6.9	8.7	133.5	19	NULL	88.4	6.5
Editar Deletar	14	2011-05-14	15:06:16+01	6	25	38.7	6.9	8.7	133.5	19	NULL	88.4	6.5
Editar Deletar	15	2011-05-15	09:53:37+01	7	24	38	5	4.6	130	20	NULL	87	7.7
Editar Deletar	16	2011-05-15	11:23:40+01	NULL	25	38	8	5	132	NULL	18	87	7.5
Editar Deletar	17	2011-05-15	12:52:40+01	5	24	NULL	NULL	8	NULL	15	NULL	NULL	NULL
Editar Deletar	18	2011-05-15	12:53:21+01	5	NULL	38	8	NULL	NULL	15	NULL	NULL	7

Figura 4.12: Tabela medições da base de dados

A figura 4.13 mostra-nos o conteúdo da vista lista de pacientes.

id	nome	data_nasc	morada	id_civil	telefone	password	username	id_medico
15	Jailson Duete Martins Moreira	1981-06-13	costa cabral	234567309	963239908	81dc9bdb52d04dc20036dbd8313ed055	Jailson Moreira	3
19	Octavio Francisco Semedo	1981-06-13	rua da miseria	6567889	963239908	d04500dbac51e00df4a6cc0657886b07	Octavio semedo	2
20	Carlos Tavares Martins	1990-12-12	rua da frente	212331444	967028830	d04500dbac51e00df4a6cc0657886b07	Carlos Martins	2
21	Jose Jorge Borges	1990-12-12	rua da frente	98765789	987654334	d04500dbac51e00df4a6cc0657886b07	Jose Borges	2
22	Izequiel pereira Veiga	1981-06-13	rua da crise	673775647	967899999	d04500dbac51e00df4a6cc0657886b07	Izequiel Viega	2
23	Odaley Pereira	1960-12-12	rua da comedia	879779999	967899999	d04500dbac51e00df4a6cc0657886b07	Odaley Pereira	2
24	Patricio Lima	1990-12-12	rua da comedia	98098899	88987789	d04500dbac51e00df4a6cc0657886b07	Patricio Lima	2

Figura 4.13: Tabela da vista lista de pacientes da base de dados

### 4.3.2 Resultados da aplicação Web

Para iniciar qualquer actividade neste sistema é necessário estar inscrito no mesmo. As interfaces que se seguem vão retratar esta etapa de registo que tem como ponto de partida a interface inicial da aplicação, ilustrada na Figura 4.14. Nesta página encontram-se opções de registo de novos utilizadores e de entrada na aplicação.



Figura 4.14: Interface Inicial

Carregando sobre registar, o utilizador escolherá o tipo (paciente, médico ou público) como mostra a Figura 4.15



Figura 4.15: Interface para Escolher Tipo de utilizador

De seguida mostra-se as interfaces de registo dos três utilizadores, começando pela do médico que para fazer o registo precisa de um código fornecido pela administração. Este código é utilizado na interface anterior ao do registo que poderá ser vista na figura em anexo. Após essa validação, então este poderá realizar o respectivo registo preenchendo o formulário apresentado na Figura 4.16. Esta interface também é usada pelo público para realizar o respectivo registo que depois de realizado terá de ser validado pelo administrador.

Registrar	
Username	<input type="text"/>
Password	<input type="password"/>
Repetir Password	<input type="password"/>
Nome	<input type="text"/>
Morada	<input type="text"/>
N.º ID Civil	<input type="text"/>
Data Nascimento	<input type="text"/>
Telefone	<input type="text"/>

Figura 4.16: Interface de Registo para Médico e Público

Quanto ao Paciente é necessário que o médico lhe forneça o seu número de ficha clínica e o número do próprio médico, para que possa realizar o registo como mostra a Figura 4.17

Registrar	
Username	<input type="text"/>
Password	<input type="password"/>
Repetir Password	<input type="password"/>
Nome	<input type="text"/>
Morada	<input type="text"/>
N.º ID Civil	<input type="text"/>
Data Nascimento	<input type="text"/>
Telefone	<input type="text"/>
ID do Médico :	<input type="text"/>
Nº de Ficha :	<input type="text"/>

Figura 4.17: Interface de Registo para Paciente

Concluído o processo de registo e a respectiva validação, mostraremos agora as interfaces principais para cada um dos utilizadores começando pela do Médico que é a lista dos pacientes representada na Figura 4.18. Nesta página o médico terá a lista de todos os seus pacientes, podendo consultar as fichas clínicas dos pacientes e ainda monitorar os dados biométricos então enviados. Ainda nesta página há links para outras interfaces como perfil e listas de fichas clínicas activas e inactivas.



Figura 4.18: Interface Principal do Médico

A interface principal dos pacientes é o da ficha clínica como podemos ver na Figura 4.19. Nesta página o paciente pode ver o seu processo clínico delineado pelo médico, consultar alertas e os valores das medições. Ainda há links para outras interfaces.

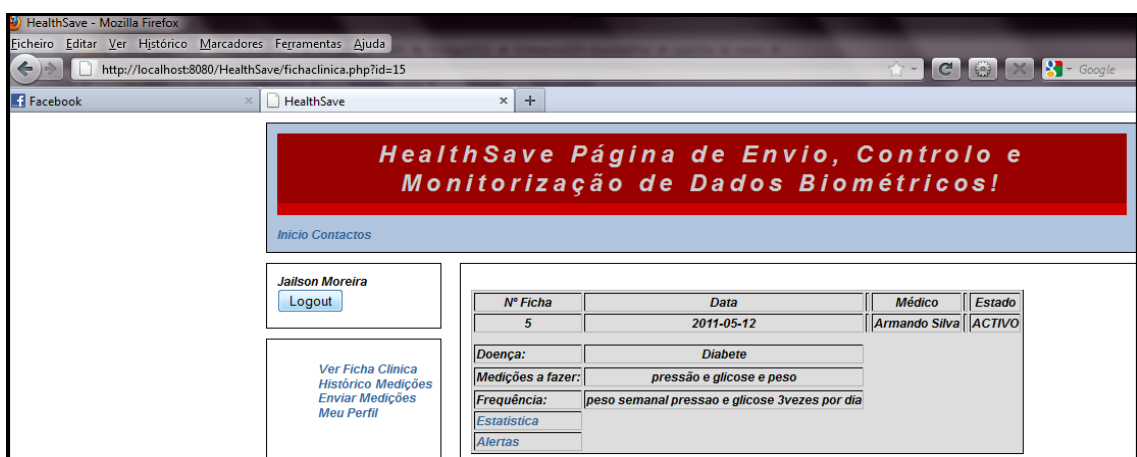


Figura 4.19: Interface Principal do Paciente

A interface principal do público é o histórico das medições, que é uma interface similar à

interface vista pelo paciente e pelo médico relativo ao histórico medições ilustrada na Figura 4.20. As demais interfaces referentes ao utilizador público encontram-se em anexo.

IMC [Kg/m <sup>2</sup> ]	Glicose A1C [mmHg]	Glicose 2h [mmHg]	Temperatura [°C]	PA Sistólica [mmol/l]	PA Diastólica [mmol/l]	Colesterol [mmol/l]	Data	Hora
27	12	8	38	127	83	5	2011-06-14	17:50:02+01
23	5	6	39	120	79	8.7	2011-06-09	14:42:33+01
23.9	7.7	8.9	39.5	108	162	5.9	2011-06-08	23:22:44+01
39.8	6.5	7.5	38	134	90		2011-06-03	14:14:42+01
23.4			38.4				2011-06-02	17:44:15+01
								<a href="#">Seguinte</a>

Figura 4.20: Interface histórico medições

Para a aplicação Web mostraremos a última interface que é o de alertas como se pode ver na Figura 4.21. As restantes interfaces podem ser vistas em anexo. Esta página dá aos utilizadores um diagnóstico relativo às últimas medições, para que estes possam tomar as respectivas medidas de acordo com o seu estado actual.

IMC	Glicose A1C / Glicose 2h	Temperatura	Colesterol	PA Sistólica / PA Diastólica
Sobrepeso	Glicose Alta	Febre Moderada	Nivel Desejavel	Pressão Ótima

Figura 4.21: Interface de Alertas

### 4.3.3 Resultados e Análise da aplicação Móvel

Foram realizados testes em telemóveis com SO Android versão 2.2 e funcionou perfeitamente. No entanto as imagens apresentados são dos testes realizados no emulador do SDK. É de se salientar que apesar de não termos realizado mais testes em telemóveis com versões do SO Android superior ao 2.2 esta aplicação deve funcionar nestes dispositivos. Em versões inferiores não é garantido que estas funcionem, entretanto.

A aplicação móvel apresenta a interface inicial, ilustrada na Figura 4.22. A imagem da Figura 4.23 mostra-nos o resultado dos testes quando o utilizador introduz valores inválidos do username e/ou da password.

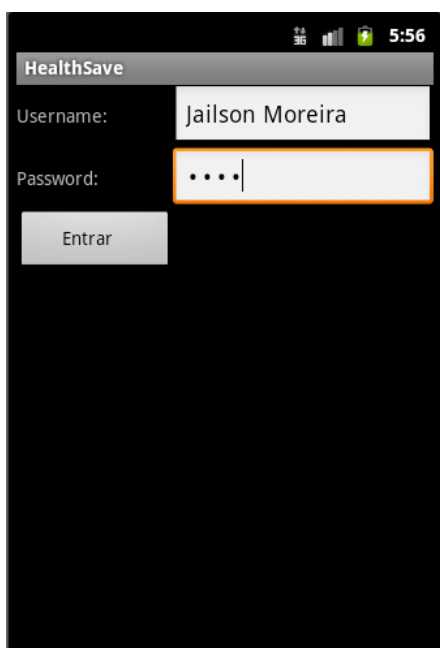


Figura 4.22: Interface Inicial

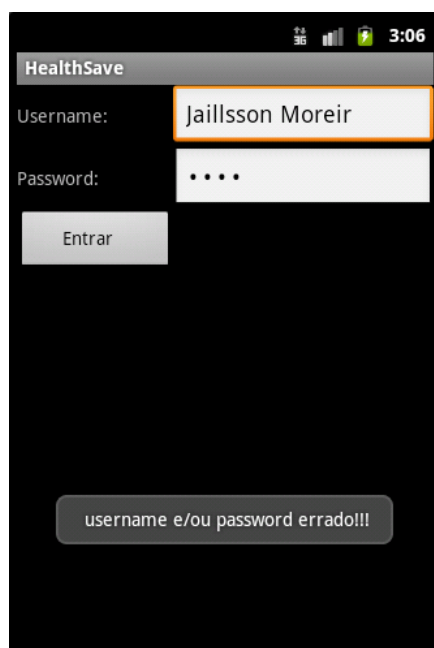


Figura 4.23: Interface erro login

As interfaces principais do paciente e médico são apresentados nas Figuras 4.24 e 4.25, respectivamente. Quanto à do público é igual à do paciente com a excepção do botão que dá acesso à ficha clínica. De resto as interfaces são iguais.

O paciente e o público, a partir da interface principal, têm acesso às seguintes funcionalidades: enviar medições, histórico medições, estatísticas, alertas e perfil. O paciente ainda terá acesso à sua ficha clínica. O médico poderá ver o seu perfil e lista dos pacientes. Apresentaremos as interfaces de enviar medições e as suas nuances, o histórico de medições e as alertas, enquanto as outras poderão ser vistas em anexo.



Figura 4.24: GUI Principal Paciente

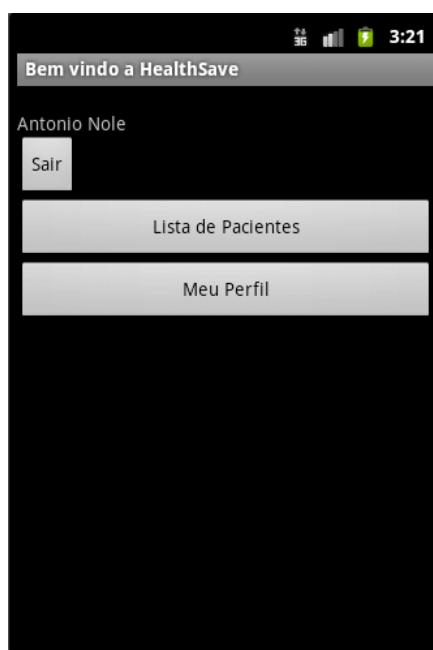


Figura 4.25: GUI Principal Médico

O envio das medições é uma das principais funcionalidades da aplicação. A interface foi feita da forma mais intuitiva possível, fornecendo as informações necessárias para realização da mesma. A Figura 4.26 mostra o caso em que o utilizador introduz dados incorrectos para o envio. Sempre que tal acontece o campo com valores incorrectos ficam com a cor vermelha e é mostrada uma mensagem sinalizando o respectivo erro. Não havendo erros, então é solicitada ao utilizador uma confirmação dos valores introduzidos numa outra interface como mostra a Figura 4.27. Em anexo encontram-se mais imagens relativas ao teste de envio de medições.

O utilizador pode ver o histórico de medições dos diferentes tipos de medições separadamente, ou seja poderá ver só as medições relativas a Tensão Arterial ou IMC, etc. A interface apresenta os valores e as respectivas datas e horas, conforme mostra a Figura 4.28



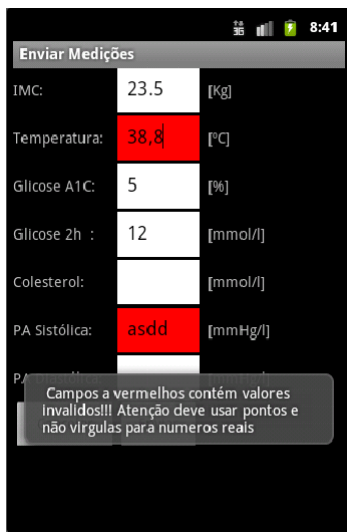


Figura 4.26: Visualização do erro no envio de medições



Figura 4.27: GUI de Confirmação dos valores a enviar

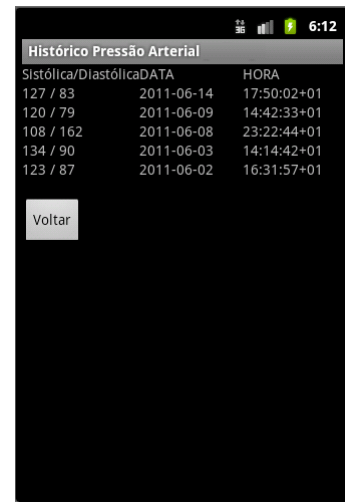


Figura 4.28: GUI histórico medições Pressão Arterial

Assim como o histórico de medições, as visualizações dos alertas estão individualizadas para os diferentes tipos de medições. Foram definidos sistemas de cores para melhor sinalizar os diferentes estados em que as medições podem estar. A cor verde é para os valores de medições que se encontram num nível Bom, a cor laranja para valores que estejam num nível entre os limites inferior do Bom e os limites superior do Mau e a cor e vermelha é para valores que estejam num nível Mau. Essas sinalizações podem ser vistas nas Figuras 4.29, 4.30 e 4.31.



Figura 4.29: GUI Alertas Colesterol



Figura 4.30: GUI Alertas IMC

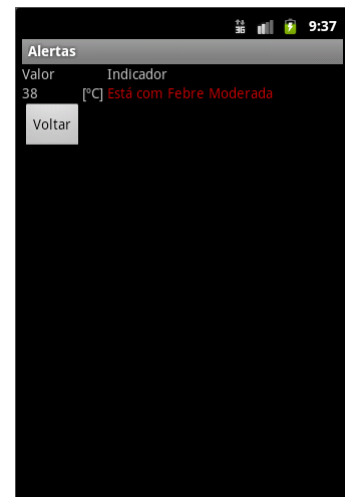


Figura 4.31: GUI Alertas Temperatural

O Médico poderá visualizar a lista de todos os seus pacientes como mostram as Figuras 4.32 e 4.33 e ter acesso ao histórico de medições, cuja interface é similar à apresentado na Figura 4.28.

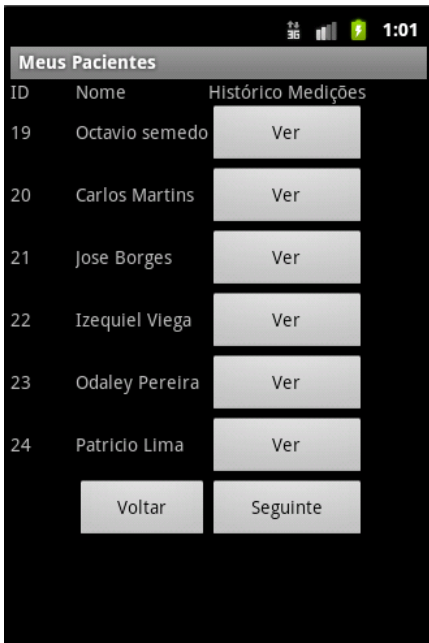


Figura 4.32: GUI Lista Paciente



Figura 4.33: GUI Lista Paciente

## 4.4 Sumário

Neste capítulo foi descrito todo o processo da implementação das duas aplicações e da base de dados. Também foi apresentado um conjunto de interfaces resultante da implementação, realçando o bom funcionamento do sistema.

## Capítulo 5

# Conclusões

Neste capítulo vai se fazer a síntese do trabalho elaborado neste documento, assim como das conclusões. São também apresentadas perspectivas de desenvolvimento futuro para este projecto.

### 5.1 Síntese do Trabalho Desenvolvido

O uso da inovação tecnológica na medicina vem se tornando cada vez mais de extrema importância nos avanços científicos da mesma. Essa inovação traz a capacidade de médicos exercerem cirurgias, exames, consultas, diagnósticos, monitorização de uma forma diferente e todos nós beneficiamos com isso.

O futuro da tecnologia médica, a julgar por seu progresso acelerado nos últimos anos, nos faz prever que a cada dia irão surgir novos equipamentos, novos aparelhos e novos recursos de diagnósticos e terapêuticos. O importante é saber quando utilizá-los e ter uma noção clara das suas indicações, suas limitações, seus riscos e da relação custo - benefício em cada caso em particular. Nem todos têm o poder financeiro para adquirir certos equipamentos, fazendo com que muitos países ou cidades ainda estejam muito atrás nessa corrida tecnológica, não podendo beneficiar das inovações. Isso faz com que nós, estudantes de tecnologia em geral, possamos desenvolver várias ferramentas ainda não existentes em nosso “mundo”, podendo beneficiar milhares de pessoas a baixo custo. Este projecto foi realizado com este propósito, trazer uma melhoria para área médica na perspectiva do utente.

Este trabalho foi desenvolvido por etapas, em que a primeira foi realizar estudos sobre os paradigmas que o sistema solicitava para melhor percepção das suas características. Na segunda etapa foram definidos e analisados os requisitos exigíveis a uma plataforma deste tipo, impondo algumas restrições e limitações para o sistema. Em seguida efectuamos a implementação do protótipo, comprovando-se que seu uso é uma mais valia para as pessoas com necessidade de controlar os seus dados vitais com ou sem acompanhamento médico, como os doentes crónicos.

Para desenvolver um sistema de monitorização dos dados vitais de pacientes é necessário ter em consideração diversos aspectos: fiabilidade, custo, segurança e usabilidade. Durante a elaboração deste projecto todos esses aspectos foram tidos em consideração. No entanto, o aspecto mais importante foi manter o baixo custo do protótipo. A segurança foi tida em conta aquando da análise de requisitos, mas não foi explorada na prática durante a implementação da aplicação Web.

De um modo geral, os objectivos para a elaboração deste trabalho foram cumpridos. O sistema mantém a sua abordagem baixo custo e é possível monitorizar dados vitais do paciente através das aplicações móvel e Web. O sistema apresenta uma interface gráfica para envio e visualização de dados de forma simples e intuitiva. No entanto houve algumas funcionalidades em relação a alertas que não foram implementadas e que serão sugeridas como melhorias do sistema em trabalhos futuros.

## 5.2 Trabalhos Futuros

O objectivo do trabalho apresentado foi essencialmente atingido. Constatase, todavia, que existe ainda trabalho a desenvolver. Sendo um sistema de monitorização algo complexo, existe ainda muito a fazer para transformar este sistema, num sistema com maior fiabilidade e melhor usabilidade. Considera-se que o sistema poderá ficar mais completo com a introdução de outras funcionalidades como marcação de consultas e o seu respectivo atendimento da parte do médico, tudo a partir da aplicação Web.

As questões de segurança são uma preocupação constante quando se trata de dados sensíveis como é o caso neste sistema. Tornar o sistema mais seguro com encriptação dos dados enviados deve ser uma evolução a ter em conta.

Uma melhoria a ter em conta também é o modo de funcionamento do sistema de alerta. Um modo de avisar tanto o médico como o paciente, através de SMS quando o paciente demorar tempo excessivo sem enviar medições para base de dados é algo a se implementar.

Também melhorar o sistema de apoio às pessoas sem acompanhamento médico, em casos em que os valores de medições estejam com variações significativas, avisando-as de riscos iminentes.

Uma outra melhoria importante é desenvolver uma funcionalidade em que haverá uma interligação de dados entre as instituições de saúde e as farmácias, de forma a que os pacientes possam adquirir os seus medicamentos com base na receita passada pelo médico. Os farmacêuticos terão acesso a essas receitas de modo a poderem confirmar a sua veracidade. Trata-se, portanto, de um sistema com elevado potencial e um grande número de extensões possíveis, não obstante deixando de mencionar a sua importância social na área da saúde.

## Anexo A

# Interfaces e Procedimentos de Instalação

### A.1 Procedimentos da instalação do SDK

**Sistemas Operativo :** Windows 7

**Ambientes de Desenvolvimento:** Eclipse IDE

Após o download do SDK, descompactou-se o arquivo .zip em uma pasta apropriada . Por padrão, os arquivos do SDK são descompactados numa pasta nomeada `android_sdk_<platform><release><build>`. O directório contém os subdirectórios tools.

Na Propriedades do Meu Computador, no friso Avançado, adicionou-se o caminho completo para o directório tools/. em Variáveis de Ambiente, e no Path em Variáveis do Sistema.

Agora para começar a desenvolver aplicações com Android no Eclipse, precisamos instalar o ADT, que adiciona suporte integrado para projectos e ferramentas Android.

1. Iniciou o Eclipse, Help > Software Updates > Find and Install..
2. Search for new features to install > Next.
3. New Remote Site.
4. nome para o site remoto (Ex.: Android Plugin) e informe a seguinte URL:
5. OK.
6. verá o novo site adicionado e seleccionado na lista de busca. > Finish.
7. O plugin ADT não é assinado, mas aceita-se a instalação > Install All.
8. Reiniciou o Eclipse.
9. Depois de reiniciar, actualizou-se as preferências do Eclipse para apontar para o directório do SDK:

- Window > Preferences > Android
- Localize o directório do SDK > SDK Location.
- Apply > OK.

## A.2 Interfaces Adicionais

Aqui vão estar as restantes interfaces da aplicação Web e da aplicação móvel

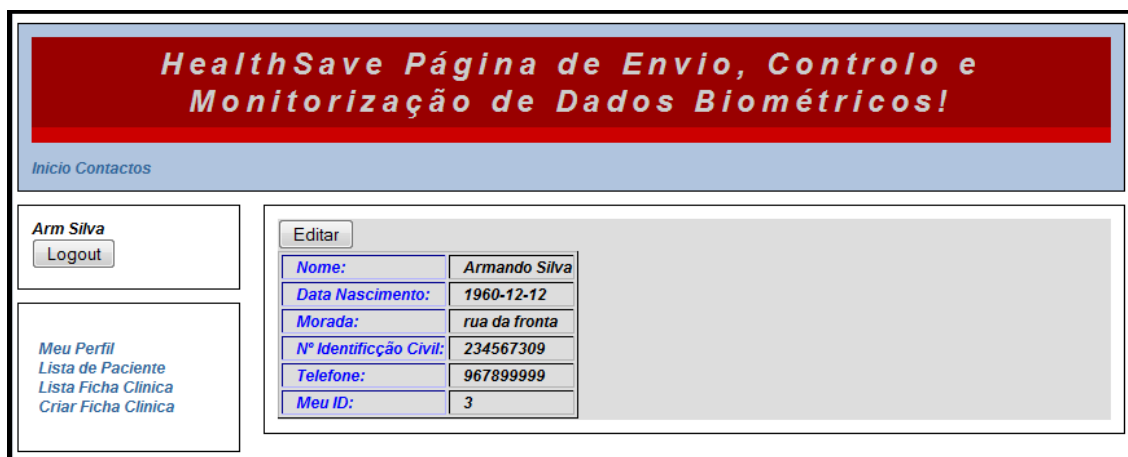


Figura A.1: Interface perfil do Médico



Figura A.2: Interface da ficha clinica de um paciente vista pelo medico Médico

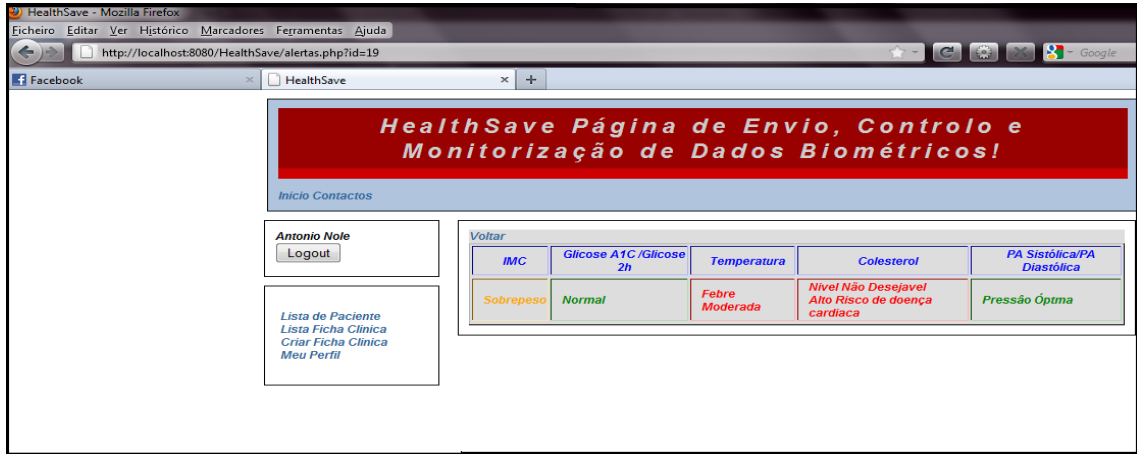


Figura A.3: Interface de alerta vista pelo médico



Figura A.4: Interface das listas de ficha clinica de um médico

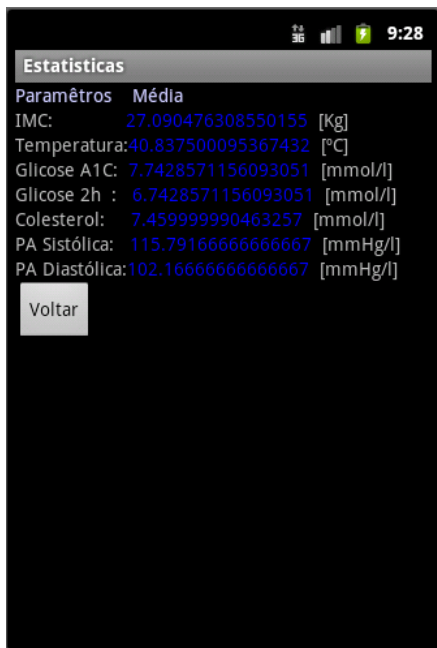


Figura A.5: Interface da média das medições

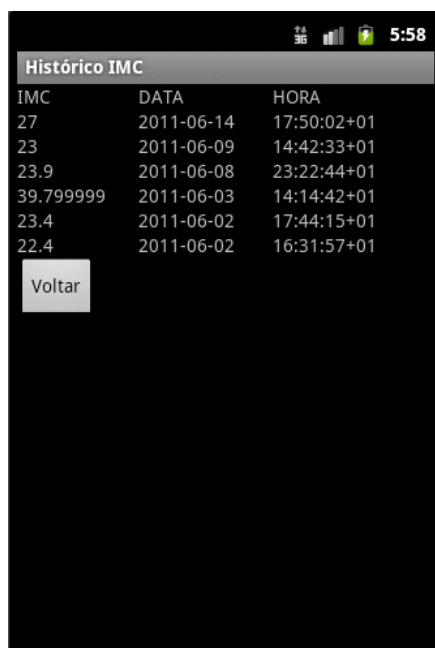


Figura A.6: Interface do historico medições do imc



Figura A.7: Interface de confirmação de envio

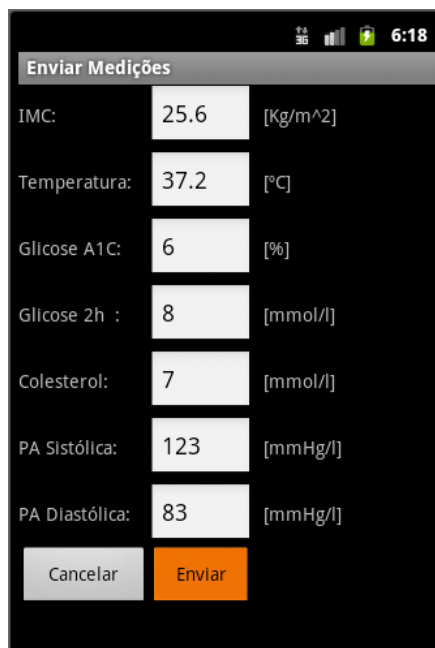


Figura A.8: Interface de envio de medições



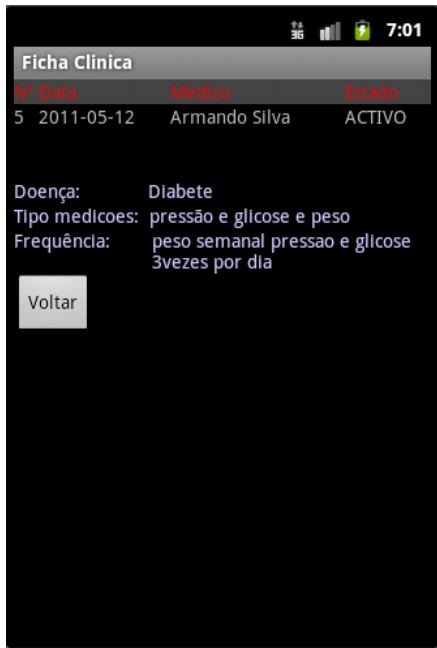


Figura A.9: Interface da ficha clinica de um paciente



Figura A.10: Interface perfil do paciente



Figura A.11: Interface principal do pu-blico

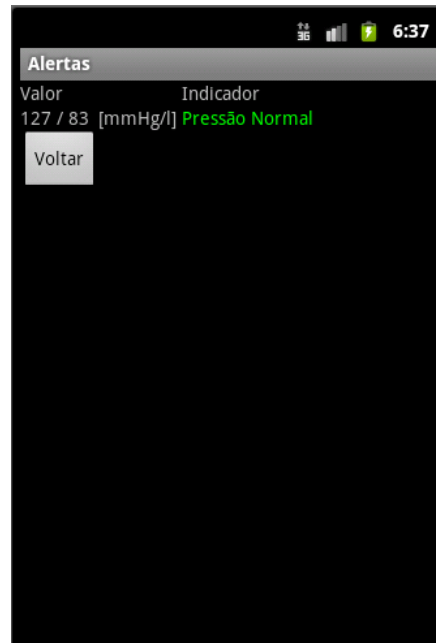


Figura A.12: Interface de alerta pressão arterial



# Referências

- [1] Mark Weiser. The computer for the 21st century. 1991.
- [2] Daniele Dutra; Márcia Abech. Infra-estrutura de sistemas ubíquos. 2008.
- [3] Guruduth Banavar; Abraham Bernstein. Software infrastructure and design challenges for ubiquitous computing applications. 2002.
- [4] M. Weiser Russell, D. M. The future of integrated design of ubiquitous computing in combined real e virtual worlds. 1998.
- [5] Jochen Schiller. *Mobile communications*. Pearson Education Limited, Second edição, 2003.
- [6] <http://pt.wikipedia.org/wiki/Wifi>, acessido a última vez em 28 de Junho de 2011.
- [7] <http://pt.wikipedia.org/wiki/Bluetooth>, acessido a última vez em 28 de Junho de 2011.
- [8] Thomas Erl. *SOA Principles of Service Design*. Pearson Education, 2008.
- [9] Rodolpho Ugolini Neto. Arquitetura orientada a serviços – soa infra-estrutura para a inovação. 2006.
- [10] M. MARKS, E. A. ; BELL. Service-oriented architecture: a planning and implementation guide for business and techonology. 2007.
- [11] N. M. JOSUTTIS. Soa in practice: The art of distributed system design. 2007.
- [12] S. CARTER. The new language of business: Soa e web. 2007.
- [13] P Elfatry, A. ; Layzell. Negotiating in service-oriented environments. 2004.
- [14] Paulo Roberto Ferreira Júnior. Coordenação de sistemas multiagentes actuando em cenários complexos. 2008.
- [15] André Torre Neto. Redes de sensores sem fio e computação ubíqua na agropecuária. 2000.
- [16] Paramvir Bahl ; Venkata N. Padmanabhan. Enhancements to the radar user location and tracking system. 2000.
- [17] Tatsuo Nakajima; Kaori Fujinami; Eiji Tokunaga; Hiroo Ishikawa. Middleware design issues for ubiquitous computing. 2004.
- [18] P. ; Guizzardi G. ; Ferreira Pires L. ; Goncalves Filho J. Rios, D. ; Dockhorn Costa. Using ontologies for modeling context-aware services platforms. 2003.

- [19] M.; Lafuente Salvador, Z.; Larrea. Smart environment application architecture, a. pervasive computing technologies for healthcare. 2008.
- [20] Jezer Machado de Oliveira; Solon Rabello; Jorge Luis Victória Barbosa. “um modelo multi-agente descentralizado para ambientes de educação ubíqua. 2006.
- [21] Dong Wang; Xiu5Feng Wang. Multi-agent based architecture of context-aware systems. 2007.
- [22] W.H.; Salcic Z.; DeSouza N.; Ramkumar S. Wang, K.I.5K.; Abdulla. Multiagent control system with mobile ubiquitous platform for ambient intelligence intelligent environments. 2008.
- [23] Alexandra Marinelli. Espaços arquitetônicos e virtuais dos serviços de saúde suportados pela telemática. páginas 62–63, Fevereiro 2006.
- [24] N. Dulay; E. Lupu; A. Schaeffer Filho; S. Keoh; M. Sloman; K. Twidle; J.Sventek; S. Heeps; S. Strowes. Amuse: Autonomic management of ubiquitous e-health systems, journal article concurrency and computation: Practice and experience wiley doi. 2007.
- [25] Douglas Faria Moreira Mareli. Sistema de computação pervasiva para tele-saúde.
- [26] Alessandro Copetti; O. Loques ; J.C.B. Leite. Intelligent context-aware monitoring in home care. 2008.
- [27] Intersystems do brasil: “healthshare”.
- [28] Einstein Lubrin; Elaine Lawrence; Karla Felix Navarro. Wireless remote healthcare monitoring with motes. 2005.
- [29] [http://pt.wikipedia.org/wiki/Doença\\_crônica](http://pt.wikipedia.org/wiki/Doença_crônica), acessido a última vez em 28 de Junho de 2011.
- [30] <http://pt.wikipedia.org/wiki/Diabetes>, acessido a última vez em 28 de Junho de 2011.
- [31] [http://pt.wikipedia.org/wiki/Índice\\_de\\_massa\\_corporal](http://pt.wikipedia.org/wiki/Índice_de_massa_corporal), acessido a última vez em 28 de Junho de 2011.
- [32] <http://pt.wikipedia.org/wiki/Glicose>, acessido a última vez em 28 de Junho de 2011.
- [33] <http://pt.wikipedia.org/wiki/Febre>, acessido a última vez em 28 de Junho de 2011.
- [34] [http://pt.wikipedia.org/wiki/Hipertensao\\_arterial](http://pt.wikipedia.org/wiki/Hipertensao_arterial), acessido a última vez em 28 de Junho de 2011.
- [35] <http://http://pt.wikipedia.org/wiki/Android>, acessido a última vez em 28 de Junho de 2011.
- [36] Elder Elisandro Schemberger; Ivonei Freitas; Ramiro Vani. Plataforma android. 2009.
- [37] Zhihui Yang; Wayne Zage; Dolores Zage. Plataforma eclipse de desenvolvimento e integração.

- [38] Leandro Daflon. Introdução à plataforma eclipse.
- [39] <http://pt.kioskea.net/contents/bdd/bddtypes.php3>, acessido a última vez em 28 de Junho de 2011.
- [40] Open database connectivity - odbc. <http://www.anaesthetist.com/mnm/sql/odbc.htm>, acessido a última vez em 28 de Junho de 2011.
- [41] Professor Esmael H. F. Santos. Interface com banco de dados java data base connectivity - jdbc. Abril 2005.