



# **DESENVOLVIMENTO DE HMI E BASE DE DADOS PARA DISPOSITIVO MÉDICO DE MONITORIZAÇÃO DE SINAIS VITAIS**

Dissertação apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de mestre em Engenharia Biomédica.

**Tiago André Vicente Carvalho**

**2007108287**

**Orientador na FCTUC: Prof. Dr. Jorge Landeck**

**Supervisor na Exatronic: Eng. André Eiras dos Santos**

Coimbra, Julho 2012

Aos meus pais e irmã,

## **Agradecimentos**

A realização deste mestrado só foi possível graças à colaboração e ao contributo de várias pessoas, às quais gostaria de exprimir o meu mais profundo agradecimento.

Inicialmente queria agradecer ao Prof. Doutor Jorge Landeck pela orientação e pela disponibilidade demonstrada.

Queria agradecer à Exatronic e aos seus colaboradores pelo acolhimento, apoio e convívio oferecidos desde o primeiro dia.

Um obrigado especial ao Eng. André Santos, na função de meu supervisor, pela orientação, pelo incentivo constante na procura de excelência e pelo entusiasmo transmitido durante a realização deste trabalho.

Ao Eng. Carlos Pereira pela disponibilidade, apoio e conhecimentos técnicos disponibilizados.

Aos meus companheiros de sala na Exatronic, pela amizade, convívio e ajuda partilhada ao longo deste ano.

A todos os meus colegas de cursos e amigos o meu muito Obrigado pelos ótimos momentos, pela amizade e pela ajuda disponibilizada ao longo destes anos.

## Resumo

O aumento da esperança média de vida associado a uma melhoria dos cuidados de saúde tem contribuído para o envelhecimento da população e para o aumento do número de indivíduos com doenças crónicas.

Estas circunstâncias criam uma sobrelotação dos centros responsáveis pela prestação desses cuidados, diminuindo a qualidade do serviço fornecido. No entanto, estas mesmas circunstâncias criam novas oportunidades de negócio com o desenvolvimento de dispositivos e serviços médicos, procurando melhorar a qualidade de vidas destes indivíduos e diminuir as necessidades dos serviços de saúde.

É com base nestes pressupostos que se iniciou o desenvolvimento do “Exa Allinone”, um dispositivo médico de monitorização de sinais vitais.

Nesta primeira fase, procedeu-se ao desenvolvimento de uma interface gráfica *user-friendly* e de uma base de dados adequada. Este processo teve em conta as diretivas europeias para o correto desenvolvimento de dispositivos e *software* médico (Diretiva 93/42/CEE; Diretiva 2007/47/CE; Norma IEC 62304), necessitando para isso a adoção de um modelo de desenvolvimento de *software*, neste caso o modelo AGILE.

O *Software* foi desenvolvido tendo em conta quatro utilizadores diferentes com permissões de utilização diferentes. Tendo em conta os conhecimentos de cada utilizador também se procedeu ao desenvolvimento de dois modos de apresentação da informação recolhida, um modo de apresentação para utilizadores com formação na área da saúde e outro para utilizadores sem essa formação.

Esta aplicação foi, assim, desenvolvida na linguagem C# com o apoio do ambiente de desenvolvimento integrado Visual Studio®, sendo o *software* compatível com o sistema operativo Windows® Embedded CE 6.0. Durante o desenvolvimento usou-se uma *mainboard* Micro 2440 da FriendlyArm, com ecrã táctil de 7”.

Após a integração da interface e da base de dados, realizaram-se testes ao sistema, dos quais resultou a elaboração de um documento descritivo dos resultados obtidos. A par deste desenvolvimento, foi criado um manual de utilização da aplicação.

**Palavras-chave:** Dispositivo Médico; *Software* Médico; Monitorização de sinais vitais; Interface gráfica; Base de dados; *Ambient Assisted Living*; IEC 62304; Diretiva 93/42/CEE; Diretiva 2007/42/CE; Engenharia de Software.

## Abstract

The increased life expectancy associated with the improvement of the health care treatments has led to the ageing of the population and an increased number of individuals with chronic diseases.

These circumstances are leading to an overcrowding of the health care centers, decreasing the effectiveness of service provided. Meanwhile, these circumstances are creating new opportunities for the development of medical solutions and services, targeting an improvement in the life quality of these individuals and easing the requirements on the health care service.

It is based upon these assumptions that the development of the “Exa Allinone” device started, a medical device for the monitoring of vital signs.

On this first stage, a user-friendly graphical interface and an appropriated database were developed. This development took in account the European directives for the correct development of medical devices and *software* (Directive 93/42/CEE; Directive 2007/47/CE; international standard IEC 62304), being necessary the adoption of a software development method, in this case, the AGILE method.

The *software* was developed taking in account the identification of four different users with different associated permissions. The knowledge of each user was taken into account, which leads to the development of two different presentation modes, one for users with health care knowledge and the other for users without that knowledge.

Therefore, the application was developed in C# programming language with the support of the integrated development ambient Visual Studio<sup>®</sup>, being compatible with the Windows Embedded 6.0 operating system.

After the integration of the interface with the database, several tests on the system were conducted. These testes led to the elaboration of a document with the obtained results. A user’s manual was also developed, explaining the usage of the application.

**Keywords:** Medical device; Medical *software*; Vital signs monitoring; graphical interface; Database; *Ambient Assisted Living*; IEC 62304; Directive 93/42/CEE; Directive 2007/42/CE; Software Engineering.

## Índice

<b>Agradecimentos</b> .....	<b>i</b>
<b>Resumo</b> .....	<b>ii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Lista de Tabelas</b> .....	<b>vi</b>
<b>Lista de Figuras</b> .....	<b>vii</b>
<b>Acrónimos e Definições</b> .....	<b>x</b>
<b>Capítulo 1. Introdução</b> .....	<b>1</b>
1.1. Contextualização .....	2
1.2 Enquadramento .....	2
1.3 Objetivos .....	4
1.4 Organização .....	4
<b>Capítulo 2. Gestão do Projeto</b> .....	<b>7</b>
2.1 Exatronic .....	8
2.2 Calendarização do projeto.....	8
<b>Capítulo 3. Conhecimentos adquiridos</b> .....	<b>11</b>
3.1 Estado de Arte.....	12
3.2 Parâmetros Fisiológicos .....	12
3.2.1 Sinais vitais.....	12
3.2.2 Temperatura corporal .....	13
3.2.3 Frequência cardíaca .....	14
3.2.4 Frequência respiratória .....	15
3.2.5 Pressão sanguínea .....	15
3.2.6 ECG .....	16
3.2.7 Oximetria de Pulso .....	17
3.3 Engenharia de Software.....	18
3.4 <i>Software</i> para Dispositivos Médicos.....	20
3.5 Ambiente de Desenvolvimento .....	24
3.5.1 Sistema de Desenvolvimento (SDK) .....	24
3.5.2 Sistema Operativo (SO).....	25
3.5.3 Linguagem <i>CSharp</i> .....	25
3.5.4 Sistema de aquisição .....	26
3.5.5 Outras Aplicações .....	27
<b>Capítulo 4. Ciclo de Vida do Desenvolvimento de Software</b> .	<b>29</b>
4.1 Descrição do Projeto .....	30
4.1.1 Definição de ambientes de operação e de Utilizadores .....	30
4.2 Análise de Requisitos.....	31
4.2.1 Casos de Uso.....	32
4.2.2 Esquema de apresentação "Normal" .....	35
4.2.3 Esquema de apresentação "Avançado" .....	36

---

4.3 Arquitetura do <i>Software</i> .....	37
4.4 Arquitetura da Interface gráfica .....	38
4.4.1 Administrador.....	39
4.4.2 Paciente.....	40
4.4.3 Convidado.....	40
4.5 Arquitetura da Base de Dados .....	41
4.5.1 Conceito ACID .....	44
<b>Capítulo 5. Resultados e Testes .....</b>	<b>45</b>
5.1 Resultados.....	46
5.2 Teste do <i>software</i> .....	48
5.2.1 Testes Unitários.....	48
5.2.2 Testes de integração .....	48
5.2.3 Testes finais.....	49
<b>Capítulo 6. Conclusão .....</b>	<b>51</b>
6.1 Trabalho Futuro .....	53
<b>Bibliografia.....</b>	<b>54</b>
<b>Anexos .....</b>	<b>56</b>
Anexo I: Empresas e produtos concorrentes .....	57
Anexo II: Análise de Requisitos de Software .....	59
Apêndice A: Desenho Inicial da Interface Gráfica.....	72
Apêndice B: Análise de Riscos .....	75
Anexo III: Desenho da Arquitetura do <i>software</i> .....	78
Anexo IV: Interface Gráfica .....	84
Anexo V: Testes finais .....	93
Anexo VI: Manual de utilizador .....	101

## Lista de Tabelas

Tabela 1 – Equipa do projeto. ....	2
Tabela 2 - Valores Normais de Temperatura corporal por local de medição e por idade .....	14
Tabela 3 - Valores normais de Freq. Cardíaca em repouso por idade .....	14
Tabela 4 - Valores normais para a frequência respiratória em repouso por idade .....	15
Tabela 5 -- Valores normais de Pressão Arterial por idade .....	16
Tabela 6 - Tarefas do ciclo de vida do processo de desenvolvimento de Software .....	21
Tabela 7 – Exemplo da Classificação atribuída a alguns dispositivos médicos de acordo com a diretiva 93/42/EEC. ....	23
Tabela 8 - Elementos necessários para o desenvolvimento do trabalho .....	24
Tabela 9 - Sinais fisiológicos adquiridos pela placa de aquisição .....	26
Tabela 10 - Equipamentos semelhantes ao dispositivo em desenvolvimento .....	57
Tabela 11 - Valores padrão para algumas variáveis consideradas .....	82
Tabela 12 - Lista de testes finais realizados a um utilizador com permissão "Convidado" .....	93
Tabela 13 - Lista de testes finais realizados a um utilizador com permissão "Paciente" .....	94
Tabela 14 - Lista de testes finais realizados a um utilizador com permissão "Administrador" .....	95
Tabela 15 - Lista de testes finais realizados por elemento: Teclado.....	98
Tabela 16 - Lista de testes finais realizados por elemento: Modo de apresentação "Normal" .....	99
Tabela 17 - Lista de testes finais realizados por elemento: Modo de apresentação "Avançado" .....	99
Tabela 18 - Identificação dos símbolos.....	103



## Lista de Figuras

Figura 1 - Percentagem da população total com idade igual ou superior a 65 anos. ....	3
Figura 2 - Percentagem de pessoas impossibilitadas de desenvolver as suas atividades diárias por problemas físicos, mentais, doença ou limitação, por idade. EU, 2001 .....	3
Figura 3 - Diagrama de Gantt - Planeamento Inicial.....	9
Figura 4 - Diagrama de Gantt - Planeamento real.....	9
Figura 5 - Representação de um batimento cardíaco normal .....	17
Figura 6 - Representação gráfica dos custos de alteração no SW nas diferentes fases do processo .....	18
Figura 7 - Representação esquemática do modelo em cascata .....	19
Figura 8 - Representação do ciclo de vida do Software em conformidade com o modelo AGILE e com a norma IEC - 62304. ....	22
Figura 9 - SDK Micro2440 da FriendlyARM. Esquerda: Touchscreen usado; Direita: placa Micro2440 .....	24
Figura 10 - Diagrama Casos de Uso – Inicial.....	33
Figura 11 - Diagrama casos de usos – Final.....	34
Figura 12 – Modo de apresentação “Normal” (desenho inicial) .....	35
Figura 13 – Modo de apresentação “Avançado” (desenho inicial) .....	36
Figura 14 – Arquitetura do sistema.....	37
Figura 15 – Esquema das janelas associadas ao Administrador .....	39
Figura 16 - Esquema das janelas associadas ao Utilizador Comum .....	40
Figura 17 – Esquema das janelas associadas ao Convidado .....	41
Figura 18 - Desenho da base de dados - Inicial.....	42
Figura 19- Desenho da base de dados - Final.....	43
Figura 20 – Modo de apresentação “Normal”. Desenho inicial.....	46
Figura 21 - Modo de apresentação "Normal" atual. ....	46
Figura 22 – Modo de apresentação “Avançado”. Desenho inicial. ....	47
Figura 23 - Modo de apresentação "Avançado" atual.....	47
Figura 24 – Esquema de apresentação “Normal” .....	68
Figura 25 – Esquema de apresentação “Avançado”.....	69
Figura 26 - Modo de apresentação "Normal" .....	72
Figura 27 - modo de apresentação "Avançado" .....	72
Figura 28 - Separador "Sistema" .....	73
Figura 29 - Separador "Valores Padrão" .....	73
Figura 30 - Separador "ECG" .....	73
Figura 31 - Separador "Dados" .....	74

Figura 32 - Separador "Adicionar Utilizador" .....	74
Figura 33 - Desenho de alto nível da arquitetura do sistema .....	78
Figura 34 - Relação entre janelas da Interface Gráfica .....	79
Figura 35 - Desenho da base de dados.....	81
Figura 36 - janela de autenticação.....	84
Figura 37 - Janela Inicial do administrador. Apresenta-se a seleção de pacientes. ....	84
Figura 38 - Modo de apresentação "Avançado". Apresenta-se a representação de gráficos e a seleção da variável.....	85
Figura 39 - Modo de apresentação "Avançado". Variáveis numéricas.....	85
Figura 40 - Esquema de apresentação "Básico". Observa-se o feedback visual desenvolvido. ....	85
Figura 41 - Modo de apresentação "Normal". Janela de confirmação da requisição de ajuda.....	86
Figura 42 - Separador "opções do sistema" .....	86
Figura 43 - Separador "Gestão de pacientes" .....	86
Figura 44 - Adição de novos pacientes/administradores .....	87
Figura 45 - Lista de Alarmes associados a um utilizador .....	87
Figura 46 - Adição de um novo Alarme.....	87
Figura 47 - Adição de um novo alarme - Seleção dos dias a repetir o alarme .....	88
Figura 48 - Adição de um novo alarme - Seleção do parâmetro a medir .....	88
Figura 49 - Adição de um novo alarme - Adição de uma nota.....	88
Figura 50 - Separador "Alteração dos dados do paciente/administrador" .....	89
Figura 51 - Separador "Alteração da password de utilizador/administrador" .....	89
Figura 52 - Separador "Gestão de Administradores" .....	89
Figura 53 - Separador "Alteração da data/hora" .....	90
Figura 54 - Separador opções de aquisição de Temperatura e saturação de Oxigénio... ..	90
Figura 55 - Separador opções de aquisição de Pressão Sanguínea .....	90
Figura 56 - Separador opções de aquisição da frequência respiratória .....	91
Figura 57 - Separador opções de aquisição de ECG.....	91
Figura 58 - Separador opções de aquisição do valor de elevação do segmento ST .....	91
Figura 59 - Separador definição dos valores padrão dos sinais vitais (valores usado no esquema básico) .....	92
Figura 60 - Teclado desenvolvido – Letras Maiúsculas .....	92
Figura 61 - Teclado desenvolvido – Minúsculas .....	92
Figura 62 - Teclado desenvolvido – Números e símbolos .....	92
Figura 63 – Janela de Login .....	101

Figura 64 - Janela principal do Utilizador "Paciente" - modo de apresentação "Normal" .....	102
Figura 65 - janela de Opções .....	103
Figura 66 - menu de alteração dos dados pessoais do utilizador .....	104
Figura 67 - Janela principal do utilizador "Administrador" .....	105
Figura 68 - Modo de apresentação "Normal" .....	106
Figura 69 - modo de apresentação "Avançado" .....	106
Figura 70 - Seleção do sinal a apresentar nos gráficos .....	107
Figura 71 - janela de opções - Sem paciente selecionado .....	107
Figura 72 - Gestão de pacientes .....	108
Figura 73 - Adição de pacientes .....	108
Figura 74 - Gestão de Administradores .....	109
Figura 75 - Janela principal do utilizador "Convidado" .....	110
Figura 76 - Janela de opções - Utilizador convidado .....	110
Figura 77 - Opções de aquisição.....	111
Figura 78 - Opções de aquisição - Valores padrão .....	112
Figura 79 - Gestão de Alarmes.....	112
Figura 80 - Adição de alarmes .....	113
Figura 81.a - (esquerda): Menu seleção de repetição; 81.b - (direita): Menu seleção de sinal .....	113
Figura 82 - Adição de comentário .....	114
Figura 83 - gestão do envio de dados .....	114
Figura 84 - gestão de números de telefone.....	115
Figura 85 - Alterar data e hora.....	115
Figura 86 - Teclado - Maiúsculas .....	115
Figura 87 - Teclado - minúsculas .....	116
Figura 88 - Teclado - Símbolos e números .....	116
Figura 89 - Colocação do sensor de SPO2.....	116
Figura 90 - ECG de 3 eletrodos - locais de colocação periféricos .....	117
Figura 91 - ECG de 3 eletrodos - locais de colocação no tronco .....	117
Figura 92 - locais para colocação do eletrodo do peito .....	118

## Acrónimos e Definições

<b>IHM</b>	Interface Homem-Máquina
<b>DM</b>	Dispositivo Médico
<b>AAL</b>	<i>Ambient Assisted Living</i>
<b>SW</b>	<i>Software</i>
<b>IEC</b>	<i>International Electrochemical Commission</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>SDK</b>	<i>Software Development kit</i>
<b>BPM</b>	Batimentos por minuto
<b>mmHg</b>	Milímetros de mercúrio

## **Capítulo 1. Introdução**

No presente capítulo será exposta a contextualização do projeto desenvolvido, a motivação para a sua concretização assim como os objetivos que se ambicionam atingir.

### 1.1. Contextualização

O projeto proposto pela Exatronic compreende o desenvolvimento de um Interface Homem-Máquina (IHM) *user-friendly* e de uma base de dados para um Dispositivo Médico (DM) de monitorização de sinais vitais, inserido no conceito de Ambient Assisted Living (AAL). O desenvolvimento desta aplicação subentende a sua adequação às normas europeias vigentes para o desenvolvimento de DMs e de *software* (SW).

Tendo por base o *know-how* da empresa na conceção e desenvolvimento de dispositivos com eletrónica integrada e a criação recente de uma nova área de negócio – Exa4life, este projeto visa desenvolver um produto com características que permitam a sua integração no mercado da saúde.

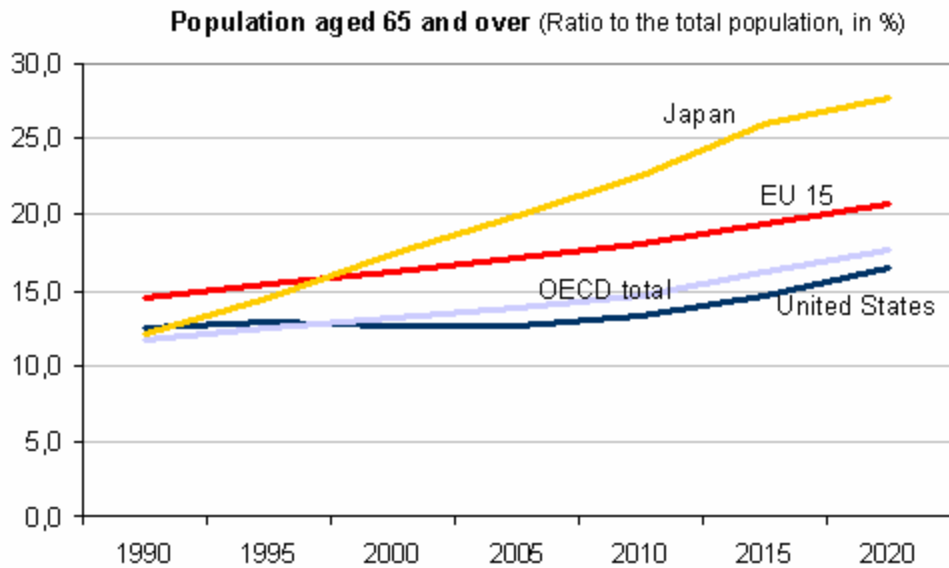
A Tabela 1 resume os intervenientes responsáveis pela realização do projeto, que visa a obtenção do grau de Mestre em Engenharia Biomédica.

**Tabela 1 – Equipa do projeto.**

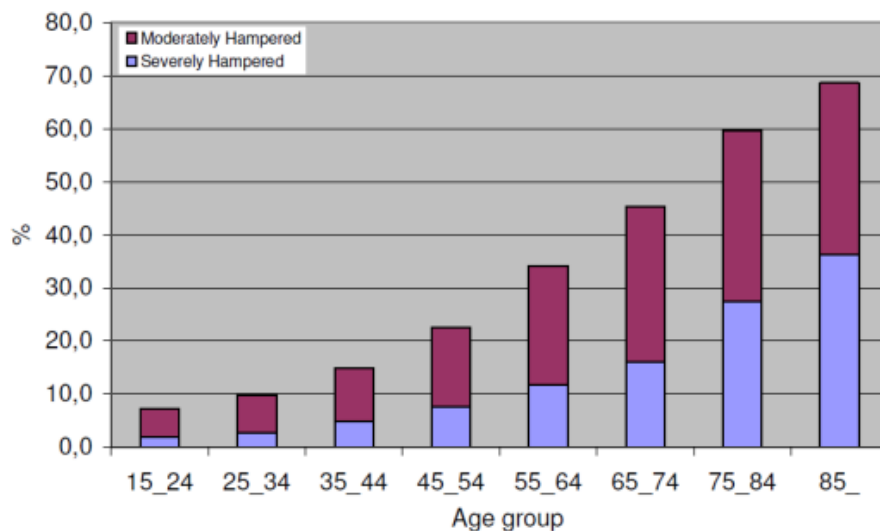
Nome	Papel desempenhado no projeto	Contactos
<b>Tiago André Vicente Carvalho</b>	Estudante responsável pela execução do Projeto	tpontocarvalho@gmail.com
<b>Prof. Dr. Jorge Landeck</b>	Orientador do Projeto na FCTUC	jlandeck@fis.uc.pt
<b>Eng. André Eiras dos Santos</b>	Supervisor do Projeto na Exatronic	asantos@exatronic.pt
<b>Prof. Miguel Morgado</b>	Responsável pela coordenação dos Projetos de MIEB	miguel@fis.uc.pt

### 1.2 Enquadramento

O aumento da esperança média de vida associado a uma melhoria dos cuidados de saúde tem contribuído para o envelhecimento da população (Figura 1) e para o aumento do número de indivíduos com doenças crónicas (Figura 2). Esta alteração demográfica traz implicações tanto no mercado de trabalho, com a diminuição de indivíduos em idade ativa, como na eficiência dos serviços de saúde com a sobrelotação destes, devido ao aumento da população com doenças limitadoras e/ou crónicas. [1]



**Figura 1 - Percentagem da população total com idade igual ou superior a 65 anos. [1]**



**Figura 2 - Percentagem de pessoas impossibilitadas de desenvolver as suas atividades diárias por problemas físicos, mentais, doença ou limitação, por idade. EU, 2001 [1]**

No entanto, esta alteração demográfica também oferece novas oportunidades de mercado, com o desenvolvimento de produtos e serviços que permitam aumentar a qualidade de vida da população envelhecida e/ou doente e solucionar os problemas mencionados. [1]

Por tudo isto surgem os dispositivos para AAL que visam ajudar os indivíduos portadores de alguma forma de deficiência e o grupo cada vez maior de idosos a viver mais tempo nos locais de que gostam, enquanto mantêm o nível de vida, autonomia e segurança. Através, por exemplo, da monitorização da saúde e da atividade, torna-se essencial providenciar acesso a sistemas de apoio social, médico e de emergência, facilitar o contacto social, e continuar o tratamento médico no seu ambiente doméstico. [1]

Assim, o dispositivo a desenvolver procura inserir-se nesta categoria de dispositivos, procurando oferecer uma interface de fácil utilização para a monitorização de sinais vitais por períodos de tempo mais ou menos longos, por pessoas formadas ou não na área da saúde. Os dados assim adquiridos permitirão a criação de historiais clínicos mais completos, para uma maior disponibilidade de dados e um diagnóstico mais fundamentado, permitindo, desta forma, a identificação de situações anormais e a consequente emissão de alarmes, procurando apostar numa medicina preventiva. [1]

### **1.3 Objetivos**

Este projeto incide no desenvolvimento de um SW para o dispositivo de monitorização de sinais vitais, sendo constituído por duas partes distintas, a base de dados e a interface gráfica. O seu desenvolvimento será orientado pela Norma 62304 da *International Electrochemical Commission* (IEC). Esta norma apresenta um conjunto de atividades necessárias para um correto desenvolvimento e manutenção de um *software* médico.

Para melhor se compreender os temas abordados na norma e definir uma metodologia para o desenvolvimento do *software*, estudou-se o conceito de Engenharia de Software. A adoção de uma metodologia para o desenvolvimento de *software* permite definir um conjunto de passos necessários para o correto desenvolvimento de *software* médico e planear o trabalho a realizar.

### **1.4 Organização**

Esta dissertação está organizada em capítulos, cujos conteúdos se apresentam resumidamente:

#### **Capítulo 1: Introdução**

Neste capítulo é contextualizado o projeto e são apresentados todos os intervenientes assim como também é descrito o enquadramento e enumerados os objetivos propostos. Finalmente, é apresentada a estrutura organizacional de toda a dissertação.

#### **Capítulo 2: Gestão do Projeto**

No presente capítulo é descrita resumidamente a organização que acolheu o projeto (Exatronic) e a contribuição do mesmo para a instituição acolhedora. É também apresentada a calendarização inicial e final das macro tarefas realizadas ao longo do projeto.

#### **Capítulo 3: Conhecimentos Adquiridos**

Neste capítulo são descritos os conhecimentos adquiridos ao longo do desenvolvimento do projeto. Estes conhecimentos são descritos resumidamente e contemplam: Estudo do Mercado; estudos dos valores padrão dos sinais fisiológicos recolhidos; Introdução à Engenharia de *Software*; Desenvolvimento de *Software* Médico; Ambiente de desenvolvimento e suas ferramentas.

#### **Capítulo 4: Ciclo de Vida do Desenvolvimento de Software**



No capítulo considerado, serão abordadas algumas das fases do ciclo de vida do desenvolvimento de *software*: Análise de requisitos de SW; Desenho da Arquitetura.

### **Capítulo 5: Resultados e Testes Finais**

Neste capítulo apresenta-se o resultado final do desenvolvimento do SW, apresentando-se, sumariamente, a interface desenvolvida e os testes a que o SW foi sendo sujeito durante o desenvolvimento.

### **Capítulo 6: Conclusão**

Neste capítulo é feita uma análise dos conhecimentos adquiridos e são apresentadas algumas sugestões para trabalho futuro.



## **Capítulo 2. Gestão do Projeto**

As secções que constituem este capítulo incluem uma breve descrição da Exatronic, empresa que permitiu o desenvolvimento do projeto, a lista das principais tarefas necessárias à concretização do projeto com êxito, juntamente com a calendarização inicialmente prevista e a calendarização que foi efetivamente cumprida.

### **2.1 Exatronic**

A Exatronic, que conta com 17 anos de atividade, reúne competências para, de uma forma vertical, conduzir processos de investigação e desenvolvimento tecnológico; criar produtos personalizados; incrementar valor e otimizar processos de industrialização. O *core business* da empresa abrange a conceção, desenvolvimento, prototipagem, industrialização e fornecimento de soluções e produtos com eletrónica.

Em dezembro de 2008 a Exatronic foi certificada pela Norma Portuguesa (NP) 4457:2007 em Gestão de Investigação, Desenvolvimento e Inovação (IDI), sendo a primeira Pequena e Média Empresa (PME) do setor da eletrónica a conseguir esta distinção.

Tendo por base os princípios orientadores em Inovação & Desenvolvimento Tecnológico (I&DT), na segunda metade de 2010, surge associada à Exatronic uma *start-up* orientada para o sector da saúde – a Exa4Life cuja motivação será o desenvolvimento de soluções biomédicas com eletrónica integrada e que permitam a melhoria da qualidade de vida do ser Humano.

Dos vários prémios atribuídos à Exatronic destaca-se o reconhecimento atribuído pelo Instituto de Apoio às Pequenas e Médias Empresas e à Inovação (IAPMEI) como PME EXCELÊNCIA 2009, em Julho de 2009. Distinção obtida também no ano de 2011.

No primeiro trimestre de 2010, a Exatronic obteve a certificação pela ISO 13485 relativo ao requisito normativo para desenvolver e fabricar dispositivos médicos com eletrónica integrada.

### **2.2 Calendarização do projeto**

No sentido de obter uma linha orientadora para a realização do projeto e para que se consigam atingir os objetivos definidos para o projeto, realizou-se um planeamento das tarefas a realizar e dos seus prazos de conclusão. Este planeamento foi realizado tendo em conta a Norma IEC - 62304:2006, onde se encontram especificadas as tarefas necessárias ao correto desenvolvimento de *software* médico.

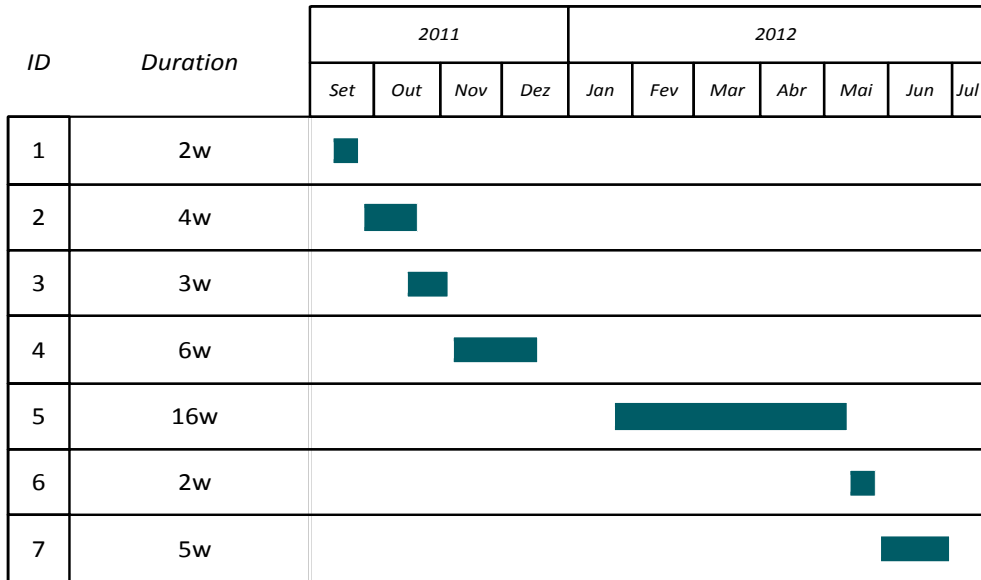
Assim, identificaram-se as seguintes tarefas principais:

1. Familiarização e perceção das metodologias de trabalho da Exatronic;
2. Estudo das diretivas e normas europeias para o desenvolvimento de dispositivos médicos,
3. Planeamento do desenvolvimento, análise de requisitos do SW;
4. Desenho de arquitetura;
5. Desenvolvimento do *Software*;
6. Elaboração e revisão da documentação de suporte;

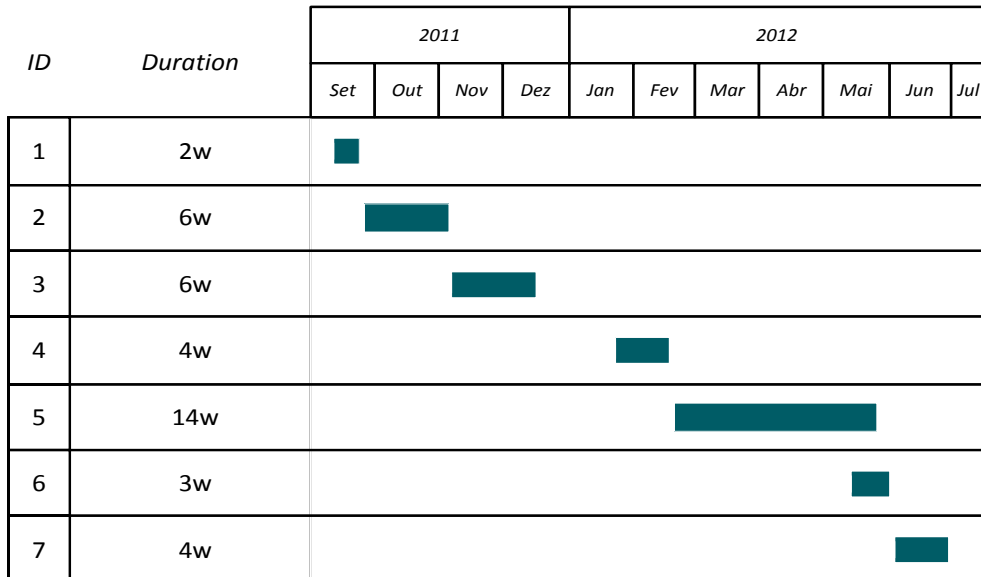
7. Elaboração da Tese.

Nos seguintes diagramas de Gantt (Figura 3 e 4) apresentam-se o planeamento inicial e final do projeto, respetivamente.

O projeto apresentado teve início a 12 de Setembro de 2011 e terminou a 30 de Junho de 2012 (data em que terminou a redação da tese de mestrado).



**Figura 3 - Diagrama de Gantt - Planeamento Inicial**



**Figura 4 - Diagrama de Gantt - Planeamento real**



## **Capítulo 3. Conhecimentos adquiridos**

As secções que compõem este capítulo apresentam alguns aspetos e considerações que foram tomadas antes do início do desenvolvimento da aplicação e que levaram à determinação da lista de requisitos para o seu desenvolvimento.

### **3.1 Estado de Arte**

Uma vez que este projeto envolve o desenvolvimento de um sistema de monitorização para ser usado em ambiente habitacional, seria interessante conhecer alguns dispositivos já disponíveis no mercado e algumas das suas características.

Assim, o estudo de mercado realizado junto de algumas empresas produtoras/fornecedoras de equipamentos médicos, revelou que existe uma vasta oferta de produtos para o meio hospitalar (serviços de cuidados gerais, centros cirúrgicos e serviços de cuidados de ambulatório), assim como o surgimento de produtos destinados ao ambiente habitacional, inseridos no conceito de AAL.

Em relação ao primeiro caso, dispositivos para meio hospitalar, verifica-se a integração de diversos métodos de aquisição num único dispositivo; a apresentação dos dados adquiridos através de valores numéricos de grandes dimensões e do recurso a gráficos. Em alguns dispositivos, verifica-se o recurso a tecnologia *wireless* para a comunicação com uma unidade central e a integração de vários dispositivos. Observou-se também, em alguns casos, a possibilidade de guardar os dados recolhidos até um máximo de 24h.

Em relação aos dispositivos para meio habitacional, observa-se a existência de um dispositivo distinto para cada sinal vital adquirido, facilitando a interação do utilizador com estes. Estes dispositivos comunicam por tecnologia *wireless* com um dispositivo central, comunicando este, por sua vez, com um servidor central onde a informação recolhida é guardada e processada, podendo ser acedida através de uma página web.

Uma vez que a aplicação a desenvolver é para ser usada num dispositivo cujo principal meio de interação com o utilizador será um *touchscreen*, estudaram-se aplicações desenvolvidas para dispositivos móveis, e a forma como estas apresentavam a informação.

Alguns destes dispositivos podem ser consultados no Anexo I.

### **3.2 Parâmetros Fisiológicos**

Nesta secção apresenta-se uma breve descrição dos parâmetros fisiológicos que o sistema é capaz de obter, do seu método de obtenção, assim como a gama de valores considerados normais para uma pessoa saudável.

#### **3.2.1 Sinais vitais**

Os sinais vitais são dos parâmetros fisiológicos mais importantes e mais vulgarmente medidos. Estes sinais são indicativos da saúde geral e do bem-estar de um doente e incluem [2]:



- Frequência cardíaca;
- Frequência respiratória;
- Pressão sanguínea;
- Temperatura corporal.

Existem outros sinais fisiológicos possíveis de serem medidos, no entanto, não são reconhecidos como "sinais vitais" e/ou encontram-se em discussão para inclusão na lista anterior. É de referir que estes implicam o uso de material mais especializado, a formação de um profissional ou apresentam uma natureza subjetiva. Entre estes sinais vitais encontram-se:

- Oximetria de pulso (SPO<sub>2</sub>);
- Nível de glicose no sangue;
- Espirometria;
- Dor;
- Características da pupila;
- Entre outros.

Dos parâmetros fisiológicos anteriormente apresentados, o dispositivo de monitorização permite a observação de:

- Sinais fisiológicos vitais:
  - Temperatura corporal;
  - Pressão sanguínea;
  - Frequência cardíaca, através do módulo de ECG e do módulo de SPO<sub>2</sub>;
  - Frequência respiratória, através do módulo de ECG.
- Sinais fisiológicos não vitais:
  - SPO<sub>2</sub>;
  - ECG;
  - Valor de elevação do segmento ST (parâmetro obtido pela análise do ECG).

Para cada um dos sinais fisiológicos considerados existe um conjunto de valores padrão, no entanto, estes valores são baseados em dados estatísticos, não tendo em conta as variações associadas a cada individuo. É de referir que estes valores também variam de autor para autor.

### **3.2.2 Temperatura corporal**

A temperatura é considerada um sinal vital uma vez que reflete a condição interna dos processos fisiológicos no corpo humano. A temperatura interna do corpo é de 37°C e não varia mais de 0.77°C, sendo mais alta do que na pele. Em contraste a temperatura da pele aumenta e diminui de acordo com a temperatura ambiental. [2,3]

A temperatura corporal é medida com um termómetro, um sensor de temperatura ou um sensor de infravermelhos, e esta pode ser medida em 4 locais diferentes: na boca, no reto, na axila e no ouvido, sendo que a temperatura medida varia ligeiramente dependendo do local de medição. [2,3]

Apresentam-se na Tabela 2, os valores normais de temperatura, por idade e por local de medição:

**Tabela 2 - Valores Normais de Temperatura corporal por local de medição e por idade [2]**

Idade	Temperatura (°C)	Local
<b>Recém Nascido</b>	35.5 - 39.5	Axilar
<b>1 ano</b>	37.7	Oral
<b>3 anos</b>	37.2	Oral
<b>5 anos</b>	37.0	Oral
<b>Adultos</b>	37	Oral
	36.4	Axilar
	37.6	Retal
<b>&gt;70 anos</b>	36.0	Oral

### 3.2.3 Frequência cardíaca

A frequência cardíaca representa o número de batimentos cardíacos por unidade de tempo, normalmente expressos em batimentos por minuto (bpm). A sua variação pode ser representativa de diversas situações, tais como: necessidade de absorver oxigénio, necessidade de maior fluxo sanguíneo, situações de *stress*, problemas hormonais (hipotiroidismo), entre outras. [2,3]

Este parâmetro pode ser obtido através do tratamento de sinais elétricos captados por ECG, através de pletismografia ótica e de palpação. [2,3]

Apresenta-se a seguir na Tabela 3, os valores normais de frequência cardíaca em repouso e por idade: [2]

**Tabela 3 - Valores normais de Freq. Cardíaca em repouso por idade [2]**

Idade	Pulso (batidas/min)
<b>Recém Nascido</b>	100 - 170
<b>1 ano</b>	80-170
<b>3 anos</b>	80-130
<b>6 anos</b>	75-120
<b>10 anos</b>	70-110
<b>14 anos</b>	60-110
<b>Adultos</b>	60-100

### 3.2.4 Frequência respiratória

A frequência respiratória corresponde ao número de vezes que ocorre um ciclo respiratório por minuto, isto é, uma inspiração e uma expiração. Este parâmetro é possível de se determinar, entre outras formas, através da variação do valor da impedância torácica durante o ciclo respiratório, medido através dos eletrodos de ECG. [2,3]

Uma forma de se verificar que uma pessoa se encontra em dificuldade respiratória é através da avaliação da saturação em oxigênio do sangue (SPO<sub>2</sub>). Se este valor for inferior a 90% indica hipoxia e requer uma avaliação imediata. [3]

Apresenta-se na Tabela 4 os valores normais da frequência respiratória em repouso e por idade:

**Tabela 4 - Valores normais para a frequência respiratória em repouso por idade [2]**

Idade	Ciclos respiratórios (ciclos/min)
<b>Recém Nascido</b>	30-50
<b>1 ano</b>	20-40
<b>3 anos</b>	20-30
<b>6 anos</b>	16-22
<b>14 anos</b>	14-20
<b>Adultos</b>	12-20

### 3.2.5 Pressão sanguínea

A pressão sanguínea é a medida da pressão exercida pela passagem do sangue nas paredes dos vasos sanguíneos durante a sístole e a diástole. Este valor é medido em milímetros de mercúrio (mmHg). Num adulto saudável, a pressão máxima (sistólica) é aproximadamente 120 mmHg, enquanto a mínima (diastólica) é de 80 mmHg. Outro valor tido em conta é a onda de pulso, que se caracteriza pela diferença entre a pressão sistólica e a pressão diastólica. [2,3]

A hipotensão ou pressão sanguínea baixa, é caracterizada por uma pressão sanguínea abaixo do valor normal. No entanto só se considera que a pressão sanguínea está demasiado baixa se houverem sintomas, tais como: tonturas e/ou desmaios. [2]

A hipertensão ou pressão sanguínea elevada, é caracterizada por uma pressão sanguínea acima do valor normal. Hipertensão persistente é um dos fatores de riscos para acidentes vasculares cerebrais, insuficiência cardíaca e doenças renais. A hipertensão grave e súbita pode ainda provocar alterações de comportamento, náuseas, tonturas e vertigens. [2,3]

Apresenta-se a seguir, Tabela 5, os valores normais da pressão sanguínea em repouso e por idade:

**Tabela 5 -- Valores normais de Pressão Arterial por idade [2]**

Idade	Sistólica (mmHg)	Diastólica (mmHg)	Média (mmHg)
<b>Recém Nascido</b>	65-95	30-60	80/60
<b>Criança</b>	65-115	42-80	90/61
<b>3 anos</b>	76-122	46-84	99/65
<b>6 anos</b>	85-115	48-64	100/56
<b>10 anos</b>	93-125	46-68	109/58
<b>14 anos</b>	99-137	51-71	118/61
<b>Adulto</b>	100-140	60-90	120/80
<b>Idosos</b>	100-160	60-90	130/80

### 3.2.6 ECG

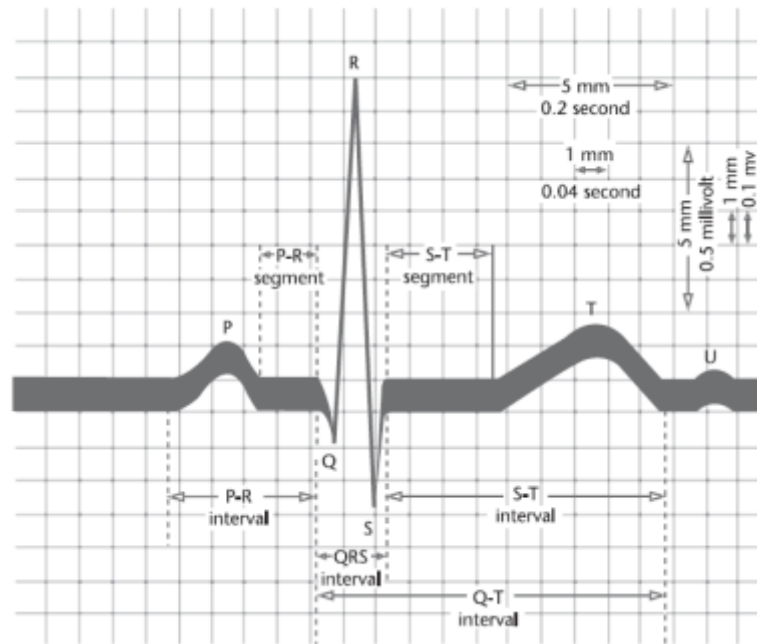
O ECG consiste na representação da atividade elétrica do coração captada através de eletródos colocados na superfície do corpo. É obtido através de um eletrocardiógrafo e é usado para verificar se a função cardíaca está normal ou se existe alguma variação que possa ser associada com uma doença cardiovascular. [2,3]

Um ECG pode ser obtido através da aplicação de até 10 elétrodos, para um ECG de 12-lead. O termo "lead" é usado para referir a posição dos elétrodos e o modo como a detecção da atividade elétrica do coração está ser medida, sendo que cada *lead* permite ver uma parte do coração. Assim, é possível classificar os *leads*, quanto ao modo de aquisição, em: [5]

- **Lead Bipolar** - medição da diferença de potencial entre lados opostos do coração;
- **Lead Unipolar** - através da medição da variação de um único elétrodo em relação a um potencial de referência.

Através do estudo da periodicidade, dos valores temporais e de amplitude dos diversos segmentos e pontos identificáveis num ECG, é possível a identificação de diversas condições cardíacas, tais como arritmias e isquemia do miocárdio. Por exemplo, na análise do segmento ST, a elevação ou depressão deste segmento pode indicar isquemia, alterações da frequência cardíaca, entre outras razões. [5]

Apresenta-se na Figura 5 a representação de um batimento cardíaco normal. Na mesma Figura é também possível observar alguns dos parâmetros analisados num ECG. O processamento dos sinais adquiridos pelos elétrodos, assim como a análise dos segmentos e pontos do ECG não serão abordados, pois não fazem parte dos objetivos do projeto.



**Figura 5 - Representação de um batimento cardíaco normal [4]**

### 3.2.7 Oximetria de Pulso

A oximetria de pulso é um método útil e não invasivo para se inferir o estado cardiorrespiratório do doente. Este método permite determinar a saturação em oxigénio do sangue arterial, assim como a frequência cardíaca. Estas medidas são obtidas através de um oxímetro de pulso, cuja operação se baseia em dois princípios: na variação da absorção da luz consoante o grau de oxigenação da hemoglobina, permitindo o cálculo da saturação em oxigénio do sangue, e na variação pulsátil do valor de luz absorvida, permitindo o cálculo da frequência cardíaca. [6]

Uma pessoa saudável tem um valor de SPO2 compreendido entre os 95 e os 99% de saturação. [6]

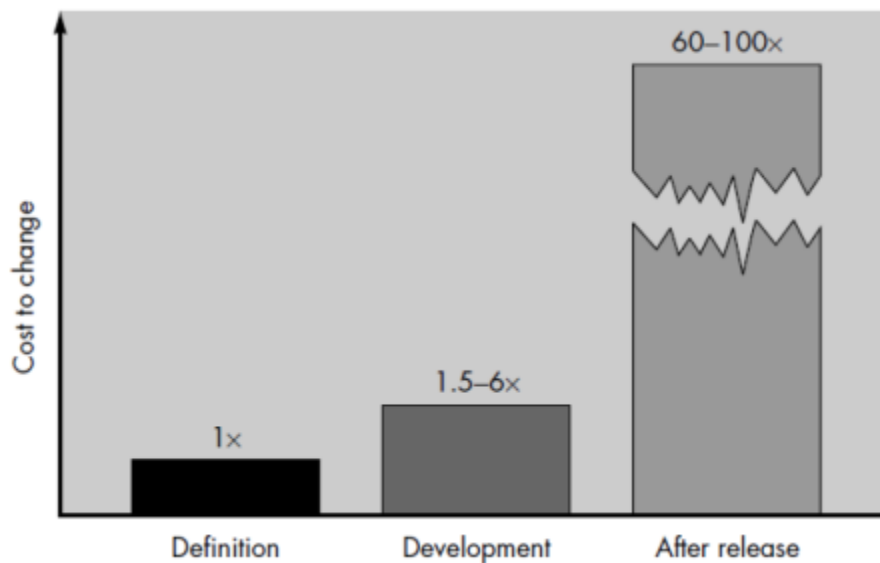
Existem alguns fatores que podem afetar as medições obtidas por este método, tais como: a luz ambiente, o movimento do paciente, uma função cardíaca alterada, assim como uma vasoconstrição e/ou uma hemoglobina anormal, a temperatura do local de medição, e/ou o grau de limpeza que o próprio fotorreceptor apresenta. [6]

### 3.3 Engenharia de Software

O desenvolvimento de *Software* nos últimos 50 anos sofreu uma grande evolução. Maioritariamente devido ao aumento da performance do *hardware*, com o aumento da memória física e da capacidade de armazenamento, mas também com mudanças na forma de desenhar e pensar o desenvolvimento de *software* e a arquitetura deste. [7,8]

Este projeto assenta no desenvolvimento de SW o que atualmente é considerado por grande parte dos autores como uma tarefa complexa e importante. Com esta complexidade surgiu a necessidade de criar a disciplina de Engenharia de Software, que foi descrita em 1993 pela IEEE, como "a aplicação de um processo sistemático, disciplinado, e quantificado ao desenvolvimento, operação e manutenção de SW". [8]

Um dos principais objetivos da Engenharia de *Software* é reduzir o tempo de desenvolvimento e custos, pois só assim é que um projeto se torna viável. Para tal, uma das metas passa por reduzir o número de alterações em fases avançadas e prever possíveis erros no início do desenvolvimento, pois como mostra a Figura 6, quanto mais tarde for efetuada a alteração, maior será o custo dessa alteração. [7,8]



**Figura 6 - Representação gráfica dos custos de alteração no SW nas diferentes fases do processo [8]**

Deste modo, o projeto vai ter uma maior incidência na tarefa de definição de requisitos do SW evitando erros ao longo do desenvolvimento, diminuindo assim os custos de engenharia através da diminuição do tempo dispensado para o seu desenvolvimento. [8]

Os fatores essenciais que se devem ter em conta durante todo o desenvolvimento são as seguintes [8]:

- **Flexibilidade**, de modo a tornar fácil a adaptação a novos requisitos que possam surgir.

- **Fiabilidade**, em todo o SW de forma a garantir que o número de erros seja reduzido e não ponha em causa o bom funcionamento deste.
- **Usabilidade**, através de uma interface *user friendly* e muito intuitiva.
- **Segurança**, de forma a garantir que todos os intervenientes com o SW não sejam lesados.
- Um desempenho adequado a todas as partes do projeto.

Para garantir que estas características são cumpridas é defendido por grande parte dos autores que se siga uma metodologia de desenvolvimento. Uma das primeiras a surgir foi o modelo em cascata (Figura 7). Este modelo consiste na realização do processo de desenvolvimento uma única vez.

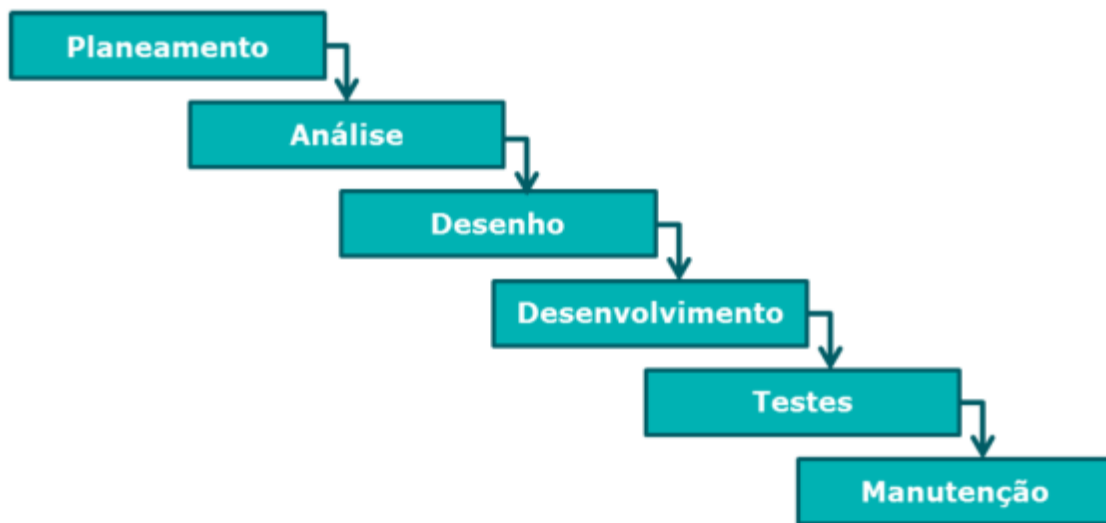


Figura 7 - Representação esquemática do modelo em cascata [8]

Atualmente, os processos de desenvolvimento mais usados seguem modelos incrementais e/ou evolutivos, mais flexíveis que o modelo anterior, e que procuram obter sistemas finais mais sólidos e com maior qualidade, valorizando:

- **Indivíduos e interações;**
- **SW Funcional;**
- **Colaboração com o cliente;**
- **Resposta às alterações;**
- **Testes unitários contínuos.**

De entre os métodos para desenvolvimento de SW, não existe um método que seja melhor que os outros, cada um tem as suas vantagens e desvantagens, sendo que cada programador e empresa se identificará melhor com um ou com outro. Assim, dos diversos modelos existentes, o modelo AGILE foi escolhido para o desenvolvimento do projeto, tanto pelas suas características como por já ser aplicado na empresa.

Este é um modelo de desenvolvimento iterativo e incremental que procura encorajar uma resposta rápida e flexível a alterações no desenvolvimento do projeto. [7]

Este modelo procura [7]:

- **Encorajar equipas e atitudes que facilitem a comunicação entre pessoas;**
- **Enfatizar a rápida entrega de *software* operacional;**
- **Adotar o cliente como parte da equipa de desenvolvimento;**
- **Reconhecer que o planeamento tem as suas limitações e que um projeto deve ser flexível.**

### **3.4 *Software* para Dispositivos Médicos**

Na Europa, todos os dispositivos médicos que tenham por objetivo a entrada no mercado têm de cumprir as disposições presentes na Diretiva 2007/47/CE, que veio complementar a Diretiva 93/42/CEE.

Assim, e de acordo com a definição presente nesta diretiva, um “dispositivo médico” é: [10,11]

*“Qualquer instrumento, aparato, aplicação, software, material ou outro artigo, quer seja usado sozinho ou em combinação, incluindo o software cuja intenção do fabricante é de usa-lo especificamente para diagnóstico e/ou fins terapêuticos e que seja necessário, para a sua correta aplicação, ser usado por seres humanos com o intuito de:*

- *Diagnóstico, prevenção, monitorização, tratamento ou alívio de doenças;*
- *Diagnóstico, monitorização, tratamento, alívio ou compensação de um ferimento;*
- *Investigação, substituição, modificação, ou suporte de um processo anatómico ou fisiológico;*
- *Controlo de conceção;*

*e que não atinja a sua ação principal no corpo humano por meios farmacológicos, imunológicos ou metabólicos, mas que possa ser assistido na sua função por tais meios.”*

Como mencionado anteriormente, este projeto tem como objetivo o desenvolvimento de *Software* para um sistema de monitorização remota de sinais vitais. Assim sendo, e de acordo com a Diretiva 2007/47/CE, todo o *Software* usado em dispositivos médicos é, ele próprio, um dispositivo médico. [10,11]

Assim, a grande diferença entre um *Software* para aplicação médica e um *Software* comum são os requisitos regulamentares que o primeiro tem que cumprir, assim como uma documentação mais exaustiva do seu desenvolvimento. [11]

Para que estes requisitos sejam cumpridos, surgem as normas, que fornecem as linhas orientadoras para o desenvolvimento de *Software*. Neste projeto, a norma a seguir será a IEC 62304:2006 – *Software life cycle processes*. Esta norma aconselha a adoção de um processo de desenvolvimento de *software* (*Software Development Life Cycle Model*) e identifica o número mínimo de tarefas para que o *software* desenvolvido seja de confiança e capaz de produzir resultados seguros e fidedignos. O número de tarefas exigidas por esta norma varia com a classificação de segurança atribuída ao *software*. Assim, na Tabela 6, encontram-se listadas as tarefas indicadas na norma, que coincidem com as macro tarefas



consideradas pela maioria das metodologias de desenvolvimento, assim como uma pequena descrição do que se pretende em cada tarefa. [9]

**Tabela 6 - Tarefas do ciclo de vida do processo de desenvolvimento de Software [8,9]**

<b>Tarefa</b>	<b>Descrição</b>
<b>Planeamento do desenvolvimento</b>	Elaboração de um plano de desenvolvimento para conduzir as atividades das fases de desenvolvimento de acordo com o objetivo, a magnitude e as condições de segurança para o desenvolvimento do SW.
<b>Análise de requisitos</b>	Recolher e descrever todas as características, funcionalidades, <i>inputs</i> e <i>outputs</i> , requisitos de segurança, entre outros que se ache relevante para o desenvolvimento do SW, documentando estes pontos.
<b>Desenho de arquitetura</b>	Transformação da análise de requisitos numa arquitetura, descrevendo a estrutura e identificando os itens do SW.
<b>Desenho detalhado</b>	Refinar a arquitetura do SW e definição de algumas funções, de forma a ter uma perceção mais detalhada de como será o SW.
<b>Implementação e Verificação de unidades do software</b>	Estabelecer estratégias, métodos e procedimentos para a verificação. Definir critérios de aceitação ao nível: da memória, dos dados e o controlo do seu fluxo, do tratamento de falhas, de autodiagnóstico, bem como da sequência de eventos.
<b>Integração e testes de integração</b>	Integrar o SW e testá-lo de forma a verificar se esta a atingir os objetivos ou não.
<b>Testes ao sistema</b>	Estabelecer um plano de testes para o sistema executar cada uma dessas tarefas e verificar a integridade e funcionalidade do sistema.
<b>Lançamento do SW</b>	Antes do lançamento do SW o fabricante tem que assegurar que toda a verificação do mesmo foi feita, bem como avaliar todos os resultados dessa verificação.

Para a definição do método de desenvolvimento de *Software*, as tarefas indicadas pela norma foram cruzadas com o modelo AGILE, obtendo-se o esquema da Figura 8.



**Figura 8 - Representação do ciclo de vida do Software em conformidade com o modelo AGILE e com a norma IEC - 62304. [8]**

Após o lançamento do SW, o desenvolvimento não termina, mas inicia-se uma nova fase de desenvolvimento de Software. De forma a atingir os requisitos do cliente, de segurança e desempenho novas versões serão desenvolvidas.

### **Classificação de Segurança**

Para que os dispositivos se encontrem em conformidade com as diretivas e as normas, é necessário proceder à sua classificação de segurança, tendo como base os possíveis efeitos no doente, no operador ou noutras pessoas resultante de um acidente com o uso do dispositivo. [9;10]

De acordo com a Diretiva 93/42/CEE, os dispositivos médicos podem ser classificados em quatro classes: I (risco reduzido ou nulo), IIa, IIb, III (risco elevado). Este dispositivo, após estudo das regras para classificação de dispositivos médicos presentes no anexo IX da diretiva, foi classificado como classe IIa [10].

Esta classificação teve em conta os pontos apresentados na regra 10 [10]:

*"Dispositivos ativos destinados a diagnóstico estão na classe IIa:*

...

*- Se estão destinados ao diagnóstico direto ou à monitorização de processos fisiológicos vitais, a menos que destinados especificamente à monitorização de parâmetros fisiológicos vitais, onde a natureza da variação é tal que pode resultar em perigo imediato para o doente, por exemplo variações na performance cardíaca, respiração, atividade do SNC, sendo neste caso classificados como Classe IIb. ..."* (traduzido da regra 10 do anexo IX da Diretiva 93/42/EEC).

Na tabela 7, apresentam-se alguns dispositivos médicos classificados de acordo com esta diretiva [13].

**Tabela 7 – Exemplo da Classificação atribuída a alguns dispositivos médicos de acordo com a diretiva 93/42/EEC. [13]**

Classe IIa	Classe IIb
<ul style="list-style-type: none"> <li>• Eletrocardiógrafos</li> <li>• Eletroencefalogramas</li> <li>• Termómetro eletrónico</li> <li>• Estetoscópio eletrónico</li> <li>• Equipamento eletrónico para medição da pressão sanguínea</li> </ul>	<ul style="list-style-type: none"> <li>• Monitorização em cuidados intensivos</li> <li>• Dispositivos de alarme</li> <li>• Sensores biológicos</li> <li>• Analisadores da presença de gases no sangue em cirurgias abertas ao coração</li> <li>• Monitores de apneia, inclui monitores de apneia em cuidados domiciliários</li> </ul>

Sendo o *Software* em desenvolvimento considerado um dispositivo médico, de acordo com o afirmado anteriormente, também este terá de ser classificado quando à sua segurança, como declarado na norma IEC 62304. Assim, e de acordo com esta norma, o SW pode ser classificado em [9]:

- **Classe A: Nenhum perigo ou ferimento para a saúde é possível.**
- **Classe B: Ferimentos ligeiros são possíveis.**
- **Classe C: Ferimentos graves ou morte são possíveis.**

Uma vez que o Software a ser desenvolvido só terá de apresentar os resultados obtidos pela placa de aquisição e armazenar estes dados numa base de dados, não havendo a aplicação de qualquer tipo de energia ou produto no utilizador. Como tal, não se espera que o SW represente algum tipo de perigo para o utilizador. Tendo em conta este pressuposto, o SW foi classificado com Classe A.

Como afirmado anteriormente, a norma IEC: 62304 indica o número mínimo de tarefas a desenvolver para que o SW seja considerado seguro e de confiança, sendo que este número varia com a classificação de segurança atribuída. Assim, para o desenvolvimento se encontrar em conformidade com a norma, é necessário desenvolver os seguintes documentos: [9]

- **Plano de desenvolvimento de *software*;**
- **Análise de requisitos do *software* (Anexo II);**
- **Implementação e verificação das unidades de *software*.**
- **Libertação do *Software* (*Software release*);**
- **Análise de Riscos – em conformidade com a norma ISO 14971 – *Application of risk management to medical devices* (anexo II);**

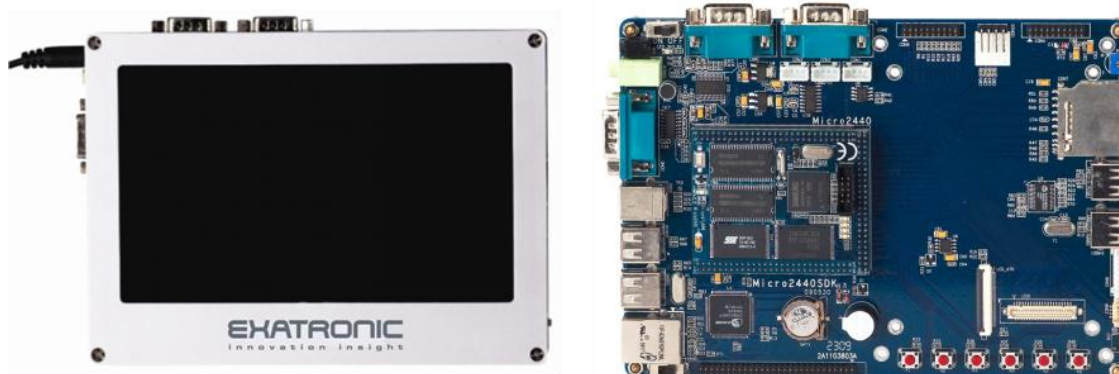
Apesar de a norma não exigir a documentação do desenho de arquitetura do *software*, nem a verificação das unidades de *software* implementadas, uma forma simplificada destes documentos foi desenvolvida (Capítulo 4.3 Arquitetura de *software* e Anexo VI: Testes finais, respetivamente) para orientação e verificação da evolução da implementação da aplicação.

### 3.5 Ambiente de Desenvolvimento

Neste capítulo será apresentada uma breve descrição do *hardware* e *software* usados durante o desenvolvimento do projeto.

#### 3.5.1 Sistema de Desenvolvimento (SDK)

Para o desenvolvimento do SW foi usado o sistema Micro2440 da *FriendlyARM*. Os fatores que levaram à escolha deste SDK foram o baixo custo, as dimensões reduzidas e a elevada conectividade, com portas USB, COM, RJ45 e ranhura para cartões SD.



**Figura 9 - SDK Micro2440 da *FriendlyARM*. Esquerda: *Touchscreen* usado; Direita: placa Micro2440 [14]**

Este sistema possui uma CPU ARM9 a 400MHz, memória RAM de 64MB (34 MB livres para o sistema operativo e aplicações), 256 MB de NAND Flash e 2MB de NOR Flash. Este sistema tem suporte para os sistemas operativos Linux 2.6, Android e WinCE 5 e 6. [14]

Este sistema possui ainda diversos elementos integrados, como portas USB e portas série, botões físicos, saída áudio, *buzzer* e ecrã-táctil. [14]

Assim, para o desenvolvimento do projeto, alguns elementos serão necessários, identificados na Tabela seguinte

**Tabela 8 - Elementos necessários para o desenvolvimento do trabalho**

Elemento	Funcionalidade
<b>Ecrã táctil 7" resistivo</b>	Mostra toda a informação do SW Interface táctil
<b>Conexão USB</b>	Sincronização e transferência de dados do SDK com o PC
<b>Porta COM</b>	Comunicação com a placa de aquisição
<b><i>Buzzer</i></b>	Emissão de alertas sonoros
<b>Porta RJ45</b>	Comunicação remota
<b>Ranhura para cartão USB /porta USB</b>	Comunicação com a base de dados

### 3.5.2 Sistema Operativo (SO)

A identificação do SO com que se irá trabalhar ao longo do projeto é da maior relevância, visto que a linguagem de programação escolhida e as ferramentas de desenvolvimento necessárias estão dependentes do SO escolhido. Assim, o SO foi escolhido entre os sistemas suportados pelo sistema de desenvolvimento: Windows CE 5.0 e 6.0, Linux 2.6 e Android, sistemas operativos suportados pelo friendlyARM.

Tendo em conta que o Windows CE 5.0 é uma versão anterior ao Windows CE 6.0, e que este último apresenta melhorias tanto ao nível da performance como da segurança, sem indicação de contrapartidas. O Windows CE 6.0 foi escolhido em detrimento do Windows CE 5.0. [15]

Numa pesquisa inicial, observou-se que os requisitos mínimos, de memória RAM e memória Flash, para um sistema Android são de 128MB RAM e de 256MB Flash, requisitos esses, que se revelam superiores aos disponibilizados pelo sistema de desenvolvimento (64 MB RAM, 256 MB Flash). Em relação ao Windows CE 6.0 refere-se a necessidade de pelo menos 300KB. Em relação ao Linux não há a indicação de um valor mínimo. [16]

Uma característica que diferencia o Windows CE 6.0 do Linux 2.6, é o facto de o Windows ser um sistema operativo em tempo real, desenvolvido para garantir disponibilidade, isto é, frequência com que o sistema responde a um pedido em tempo útil; e confiança, a frequência com que estas repostas estão corretas. No entanto esta característica não é estritamente necessária pois não se espera que o *software* a ser desenvolvido necessite de realizar operações em tempo real, por exemplo, análise de sinais fisiológicos para a emissão de alertas. Além disso, é possível adicionar suporte para operações em tempo real ao Linux. [16]

Uma opinião presente em diversos fóruns é a de que o Linux é normalmente utilizado em *headless devices*, dispositivos sem interface humano, e que este é mais vocacionado para comunicação em rede. No entanto é possível encontrar suporte para o desenvolvimento de aplicações para *headed devices*, dispositivos com interface humano, assim como é possível encontrar drivers para aumentar a conectividade do Windows CE. Outra opinião presente em fóruns da especialidade é que o custo de desenvolvimento em Windows ou Linux pode ascender ao mesmo valor, apesar do Linux ser *open source*. Isto deve-se ao facto de certos *codecs* e *drivers* para Linux terem de ser comprados, sendo, no entanto, disponibilizados de forma gratuita para Windows. [17; 18]

Assim, não havendo uma característica que diferencie de forma clara um dos sistemas operativos, o Windows CE 6.0 será o escolhido, pois já existe *know-how* na empresa.

### 3.5.3 Linguagem CSharp

*CSharp* (C#) é uma linguagem de programação desenvolvida pela Microsoft com a intenção de ser simples, moderna, de propósito geral (*general-purpose*) e *type-safe*. Procura aliar uma produtividade elevada ao poder da linguagem C, sendo suportada pelo *.NET Frameworks' Common Language Runtime*. [19]

Esta linguagem tem um conjunto de características que, além de aumentarem a produtividade do programador, permitem também a construção de aplicações mais robustas e duradouras. Entre essas características encontram-se: *garbage collection*, libertação automática de espaço em memória que não está a ser utilizado; *Exception handling*, permite detectar, identificar e atuar sobre erros que possam acontecer durante a execução do código; *type-safe*, identificando erros durante o desenvolvimento do código. [19]

Esta linguagem faz uso de uma vasta biblioteca disponibilizada pela Microsoft®, a *.NET Framework*. Esta biblioteca disponibiliza métodos para o desenho de interfaces gráficas, conectividade a bases de dados, comunicação em rede, entre outros, úteis para o desenvolvimento de aplicações para ambiente Windows. No entanto, e uma vez que estamos a desenvolver para um dispositivo móvel, não iremos usar esta vasta biblioteca, mas sim uma versão reduzida desta, a *.NET Compact Framework*. No desenvolvimento da aplicação usar-se-á a *.NET Compact Framework 3.5*, a última versão desenvolvida deste *framework*. [19]

Resumindo, esta linguagem foi escolhida pela sua semelhança com a linguagem de programação JAVA, lecionada ao longo do curso; devido à facilidade de comunicação entre o SDK de desenvolvimento e o IDE (*Visual Studio 2008*) usado no desenvolvimento da aplicação; devido à existência de *know-how* na empresa, e pelo facto de não haver suporte para a linguagem JAVA no Windows CE.

#### 3.5.4 Sistema de aquisição

O sistema de aquisição comunica com o sistema principal através de uma porta série (RS232), fornecendo em tempo real os seguintes sinais fisiológicos:

**Tabela 9 - Sinais fisiológicos adquiridos pela placa de aquisição**

Informação gráfica	Informação numérica
<ul style="list-style-type: none"><li>• <b>2 ondas de ECG (leads diferentes)</b></li><li>• <b>1 onda referente ao segmento ST do ECG</b></li><li>• <b>1 onda de SpO2 (plestimograma)</b></li><li>• <b>1 onda de respiração</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Frequência cardíaca</b></li><li>• <b>Frequência respiratória</b></li><li>• <b>Valor de SpO2</b></li><li>• <b>Temperatura</b></li><li>• <b>Pressão sanguínea</b><ul style="list-style-type: none"><li>○ <b>Pressão sistólica</b></li><li>○ <b>Pressão diastólica</b></li><li>○ <b>Pressão média</b></li></ul></li></ul>

Esta placa possui também um módulo para a aquisição da pressão sanguínea de forma invasiva, no entanto os dados obtidos por este módulo não serão processados.

### 3.5.5 Outras Aplicações

Para realizar algumas tarefas pontuais, foi necessário recorrer a outras aplicações.

- Para a comunicação entre o computador de desenvolvimento e o SDK Micro2440 usou-se o *Microsoft® ActiveSync*.
- Para o desenho gráfico dos botões usados no interface gráfico usou-se o *Inkscape®*;
- Para o desenho da base de dados usou-se o *Visual Paradigm for UML*;
- Para o desenho da arquitetura usou-se o *Microsoft® Visio*.





## **Capítulo 4. Ciclo de Vida do Desenvolvimento de Software**

Como foi apresentado anteriormente, o desenvolvimento da aplicação vai seguir os requisitos presentes na norma IEC 62304 em combinação com o modelo AGILE, de acordo com o apresentado na Figura 12. Neste capítulo vão ser descritas algumas fases do ciclo, entre as quais se destacam a análise de requisitos e o desenho da arquitetura.

#### **4.1 Descrição do Projeto**

Como foi referido anteriormente, este projeto pretende o desenvolvimento de um interface gráfico e de uma base de dados integrada para um dispositivo de monitorização de sinais vitais. Espera-se que esta interface seja capaz de apresentar os parâmetros fisiológicos obtidos pela placa de aquisição, que permita o armazenamento destes dados na base de dados, e a transferência destes dados em tempo real, em indeferido ou através da realização de relatórios para um serviço remoto, passando estes dados a estar acessíveis numa página web.

##### **4.1.1 Definição de ambientes de operação e de Utilizadores**

Tendo em conta uma melhor compreensão dos requisitos a desenvolver, procurou-se primeiro identificar os principais locais de operação, bem como os principais utilizadores do dispositivo.

Assim, este *software* foi pensado tendo em conta os seguintes locais de operação:

- **Hospitais públicos:**
  - Sistema de ambulatório – monitorização à distância da evolução do doente;
  - Realização de rastreios.
- **Clinicas/lares privados:**
  - Sistema de ambulatório – monitorização à distância da evolução do doente;
  - Realização de rastreios;
  - Monitorização de emergência.
- **Ambiente habitacional:**
  - Monitorização de emergência;
  - Responsabilização dos utilizadores para o estado da sua saúde.

A partir destes locais de operação, foi possível identificar quatro tipos principais de utilizadores do dispositivo: Utilizador profissional, utilizador comum, utilizador remoto e utilizador ocasional, descritos de seguida.

##### **Utilizador profissional:**

Considera-se que este utilizador tem algum tipo de formação na área da saúde. Desta forma, todos os dados adquiridos pela placa de aquisição serão disponibilizados ao utilizador, tanto em formato numérico como gráfico, procurando que a apresentação destes seja semelhante aos dos monitores de sinais presentes normalmente nos hospitais – apresentar a informação com recurso a algarismos de fácil visibilidade e a possibilidade de apresentar a informação em gráficos.

### **Utilizador comum:**

Considera-se que este utilizador não tem qualquer tipo de formação na área da saúde. Assim, apenas os sinais vitais e o SpO2 (sinal de fácil leitura, compreensão e obtenção) serão apresentados ao utilizador, oferecendo-se *feedback* visual e sonoro quanto ao estado do sinal medido. Os restantes dados adquiridos, mas não apresentados, serão guardados na base de dados.

### **Utilizador remoto:**

Considera-se que este utilizador esteja registado na base de dados do dispositivo e que esteja a aceder ao mesmo a partir de um dispositivo remoto (página *web* ou semelhante). Este utilizador poderá requerer o envio dos dados armazenados no dispositivo e, dependendo da permissão associada, alterar os seus dados ou de outro utilizador.

### **Utilizador ocasional/convidado:**

Considerou-se este utilizador para permitir o acesso à aplicação por utilizadores não registados na base de dados ou, caso ocorra algum erro com a base de dados, não estando esta disponível. Assim, o utilizador poderá usar o sistema de aquisição normalmente, no entanto, os dados obtidos não serão guardados.

## **4.2 Análise de Requisitos**

Os requisitos de *software* representam um conjunto de características que o SW terá de ter quando terminado, devendo também descrever as entradas e saídas do SW, o comportamento deste face a estas entradas e saídas, e o comportamento face a possíveis erros durante o seu funcionamento.

Em relação aos atores considerados, verificou-se que a principal diferença estava na forma como os dados adquiridos são apresentados e no nível de *feedback* oferecido. Assim, em vez de se criar duas aplicações distintas, uma para utilizadores profissionais, e uma para utilizadores comuns e ocasionais, optou-se por criar duas formas distintas de apresentar a informação. Assim, desenvolveu-se o modo de apresentação "Normal", mais indicado para utilizadores sem conhecimento da área da saúde, e um modo de apresentação "Avançado", mais indicado para utilizadores formados na área da saúde, com uma apresentação semelhante a um monitor de sinais vitais.

Os esquemas de apresentação mencionados serão abordados posteriormente.

De seguida apresentam-se os requisitos considerados para cada utilizador.

### 4.2.1 Casos de Uso

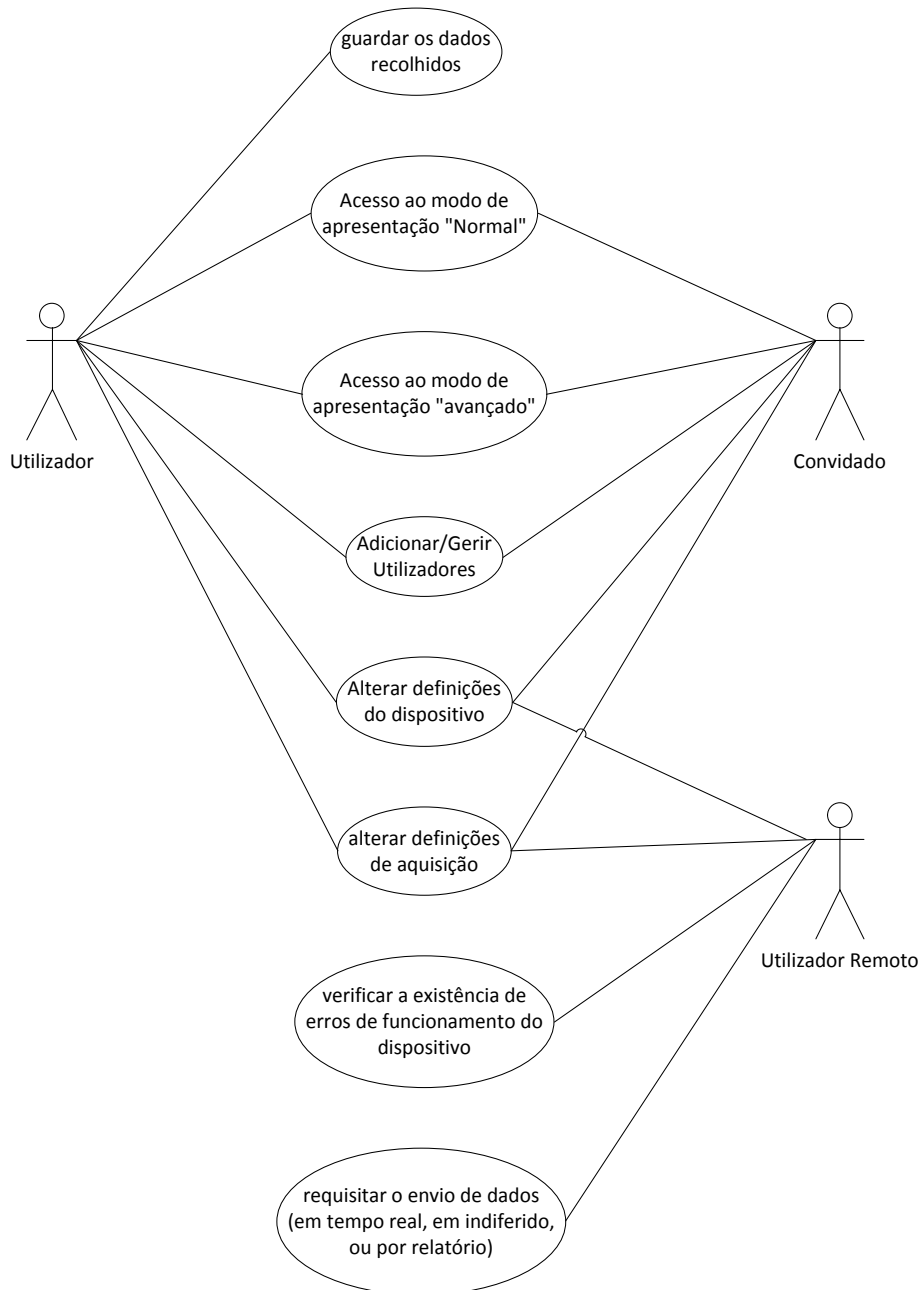
Em engenharia de *software*, os casos de uso consistem num conjunto de passos e interações entre atores/utilizadores e o sistema, no sentido de se atingir um objetivo. O utilizador pode ser humano ou um sistema externo.

Tendo em conta os utilizadores identificados para operarem o dispositivo, decidiu-se criar uma lista de permissões, de forma a restringir o acesso do utilizador a certas opções do dispositivo. Assim, cada utilizador teria uma das seguintes permissões:

- **Utilizador** – Uma vez que, inicialmente, os requisitos do utilizador comum e utilizador profissional eram muito semelhantes, diferindo apenas no modo de apresentação, optou-se por agrupar estes utilizadores na mesma permissão, permitindo a alteração entre o modo de apresentação;
- **Convidado** – Visto que os dados obtidos por este utilizador não serão guardados na base de dados, seria necessário identificar este utilizador;
- **Utilizador Remoto.**

Apresentam-se na Figura 10 os requisitos considerados para cada permissão. Como será apresentado de seguida, esta lista de permissões irá sofrer algumas alterações.

NOTA: De forma a tornar a leitura dos casos de uso mais acessíveis, os requisitos referentes a cada modo de apresentação foram agrupados nos seguintes requisitos: "Acesso ao modo de apresentação "Normal"" e "Acesso ao modo de apresentação "Avançado". Os requisitos referentes a cada modo de apresentação serão abordados de no capítulo 4.2.2. - Esquema de apresentação "Normal" e no capítulo 4.2.3 - Esquema de apresentação "Avançado".



**Figura 10 - Diagrama Casos de Uso – Inicial**

Após implementação parcial destes requisitos, verificou-se, que a interface estava confusa e que seria necessário restringir o acesso a certas opções, levando a uma alteração dos requisitos considerados. Assim, decidiu-se alterar a lista de permissões, dividindo a permissão "Utilizador" em "Administrador" e "Paciente", como se pode ver de seguida:

- **Administrador** – será o responsável pelo dispositivo, este utilizador poderá adicionar novos administradores e/ou pacientes.
- **Paciente** – será o individuo cujos sinais vitais estarão a ser monitorizados. É de referir que, em ambiente habitacional, o utilizador do dispositivo e a pessoa que está a ser monitorizada podem ser a mesma pessoa.
- **Convidado**

• **Utilizador Remoto.**

A alteração dos requisitos associados a cada permissão pode ser observada na Figura 11.



**Figura 11 - Diagrama casos de usos – Final**

### 4.2.2 Esquema de apresentação "Normal"

Como apresentado anteriormente, este modo de apresentação foi pensado para ser usado por utilizadores sem formação na área da saúde, e que, portanto, estejam pouco familiarizados com os valores normais para cada um dos sinais fisiológicos recolhidos. Assim, e no sentido de fornecer ao utilizador alguma forma de *feedback*, criou-se um código de cores indicativo da condição do sinal medido, bem como uma indicação sonora se este se encontrar alterado. Este esquema procura apresentar apenas a informação relativa aos sinais vitais (frequência cardíaca e respiratória, pressão sanguínea e a temperatura) e à SpO2, de uma forma simplificada e *user-friendly*, procurando-se transmitir a informação sob a forma de imagens e com o recurso a botões de fácil acessibilidade.



**Figura 12 – Modo de apresentação "Normal" (desenho inicial)**

Apresentam-se listados de seguida os requisitos considerados para este modo de apresentação:

- Indicação do valor dos sinais vitais medidos:
  - Frequência Cardíaca;
  - Frequência respiratória;
  - Pressão sanguínea;
  - Temperatura;
  - SpO2 (apesar de não ser considerado um sinal vital, a informação que transmite é de fácil compreensão e obtenção).
- Identificação do sinal vital medido com o recurso a ícones;
- Apresentação da informação em formato numérico;
- Fornecer *feedback* visual (codificação por cor) quanto ao estado do parâmetro fisiológico medido;
- Fornecer *feedback* sonoro se houver alteração significativa do parâmetro fisiológico medido;
- Iniciar/terminar medição da pressão sanguínea;
- Fornecer um botão para o pedido de apoio;
- Possibilidade de definir os valores padrão para os valores fisiológicos.

### 4.2.3 Esquema de apresentação “Avançado”

Como referido anteriormente, este modo de apresentação de informação foi pensado para utilizadores com formação na área da saúde. Assim, procurou-se fornecer um ambiente semelhante a outros monitores de sinais vitais mais comuns em ambiente profissional, apresentando-se a informação tanto em formato numérico como em formato gráfico. Neste modo de apresentação procurou-se apresentar toda a informação que a placa de aquisição pode adquirir, disponibilizando-se, assim, todos os sinais fisiológicos obtidos numa só janela. Dada a restrição de espaço para a apresentação da informação em formato gráfico, optou-se por apresentar apenas dois gráficos, sendo que a informação contida nos gráficos é escolhida pelo utilizador.



**Figura 13 – Modo de apresentação “Avançado” (desenho inicial)**

Apresentam-se de seguida os requisitos considerados para este modo de apresentação:

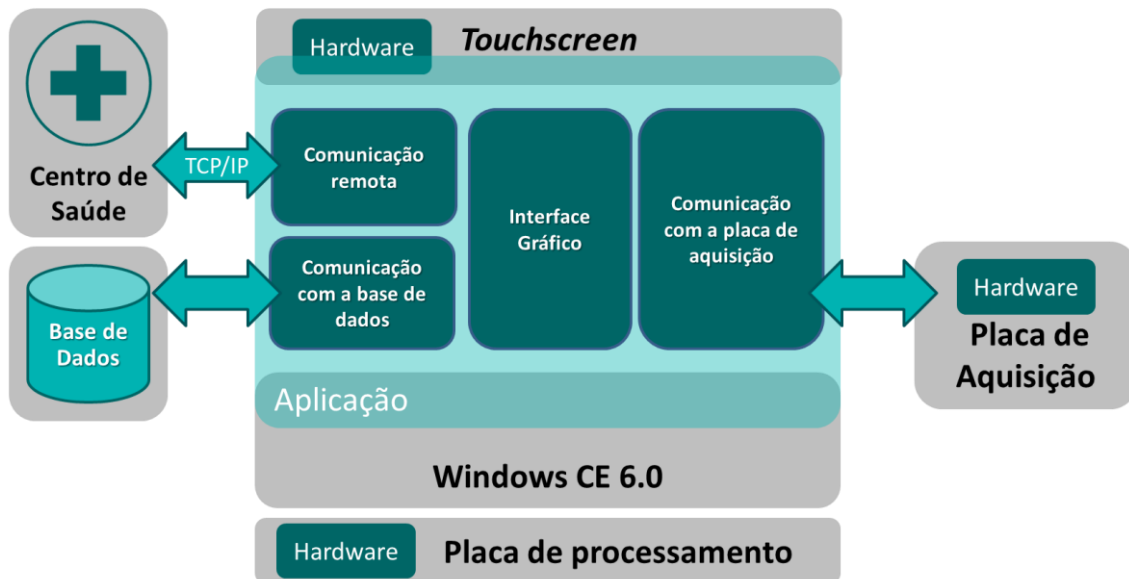
- Apresentação da informação em formato numérico e de gráficos.
  - Frequência cardíaca;
  - Frequência respiratória;
  - Pressão sanguínea;
  - Temperatura;
  - Valor de elevação do segmento ST;
  - Forma de onda do ECG;
  - Forma de onda da respiração;
  - Plestimografia;
  - Forma de onda da elevação do segmento ST.
- Seleção da informação a apresentar no gráfico;
- Iniciar/terminar medições de pressão sanguínea.



### 4.3 Arquitetura do Software

Apesar de a norma não exigir a documentação da arquitetura de *software*, esta é apresentada de forma a se conhecerem as interações entre os diversos componentes do sistema.

A arquitetura do *Software* consiste numa representação de alto nível dos componentes do *Software*, as suas propriedades externas e o relacionamento entre si e entre outros *softwares*. A arquitetura do *software* apresenta-se na Figura 14 .



**Figura 14 – Arquitetura do sistema**

Esta arquitetura pode ser separada em parte física e parte lógica, sendo a parte física composta por:

- Ecrã tátil;
- Placa de aquisição;
- Placa de processamento

E a parte lógica da arquitetura constituída por:

- Aplicação central;
- Base de dados;
- *Firmware* da placa de aquisição de sinal;
- Servidor remoto.

Relativamente à aplicação central, esta é, por sua vez, dividida em 4 partes funcionais, identificadas na Figura 18:

- Interface gráfica:
  - Responsável por processar os *inputs* do utilizador e por apresentar a informação recolhida pela placa de aquisição.
- Comunicação com a base de dados:

- Esta biblioteca fornece os métodos para controlo, introdução e procura de informação na base de dados.
- Comunicação com a placa de aquisição:
  - Esta biblioteca fornece os métodos para a receção e descodificação de mensagens emitidas pela placa de aquisição, bem como da codificação e envio de mensagem para esta.
- Comunicação com o servidor remoto.
  - Esta biblioteca fornece os métodos para a conexão com o servidor remoto, assim como outros métodos necessários para que a comunicação ocorra sem erros.

Tendo em conta o objetivo do projeto, procedeu-se apenas ao desenvolvimento da Interface gráfica e comunicação com a base de dados, ficando a comunicação com a placa de aquisição e o servidor remoto para trabalho futuro.

#### 4.4 Arquitetura da Interface gráfica

Procede-se agora a uma análise detalhada da aplicação central. Esta é composta, até ao momento, pela Interface gráfica mais duas bibliotecas (ficheiros .dll), uma contendo os métodos necessários para realizar operações sobre a base de dados, e a outra contendo objetos para serem utilizados na interface gráfica. Esta estrutura foi adotada porque permite o agrupamento do código por função lógica, facilitando assim possíveis atualizações do *software* e o reaproveitamento do código.

Relativamente à interface gráfica, esta está dividida em janelas, tendo cada janela duas classes associadas, uma classe responsável pelo formato e disposição dos componentes (botões, imagens, entre outros) na janela, sendo esta classe gerada automaticamente pelo IDE, e a segunda classe responsável pela codificação do comportamento da janela aos *inputs* do utilizador.

Durante o desenvolvimento, optou-se por se usar um número reduzido de janelas e de se modificar a informação contida nestas ao longo da interação do utilizador com a aplicação, visto que este método é computacionalmente menos exigente que a criação de janelas, diminuindo o tempo de resposta do interface.

Dadas as limitações gráficas da *framework* usada para o desenvolvimento da aplicação, em relação à usada para o desenvolvimento de aplicações para computadores, houve a necessidade de desenvolver um conjunto de controlos entre os quais botões e legendas (*labels*), que fossem mais atrativos para o utilizador.

Após o *login* do utilizador, este é identificado como Paciente ou Administrador, sendo os seus dados pessoais, definições da aplicação e de aquisição, carregados de seguida. Se o utilizador não pretender fazer o *login* ou não for possível aceder à base de dados, o utilizador tem hipótese de iniciar a aplicação como Convidado.

Apresentam-se de seguida, as janelas e separadores a que os utilizadores têm acesso, dependendo da permissão associada

### 4.4.1 Administrador

Como apresentado anteriormente, o administrador será a pessoa responsável pela adição de novos administradores e/ou pacientes. Este utilizador terá acesso a todas as pessoas registadas na base de dados e terá também acesso aos dois modos de apresentação de informação, podendo alternar livremente entre os dois. No entanto, estes modos, assim como os separadores "Dados" e "Sensores", só serão apresentados após a seleção de um Paciente, visto o administrador realizar as aquisições em nome desse utilizador, e portanto usar as definições deste.

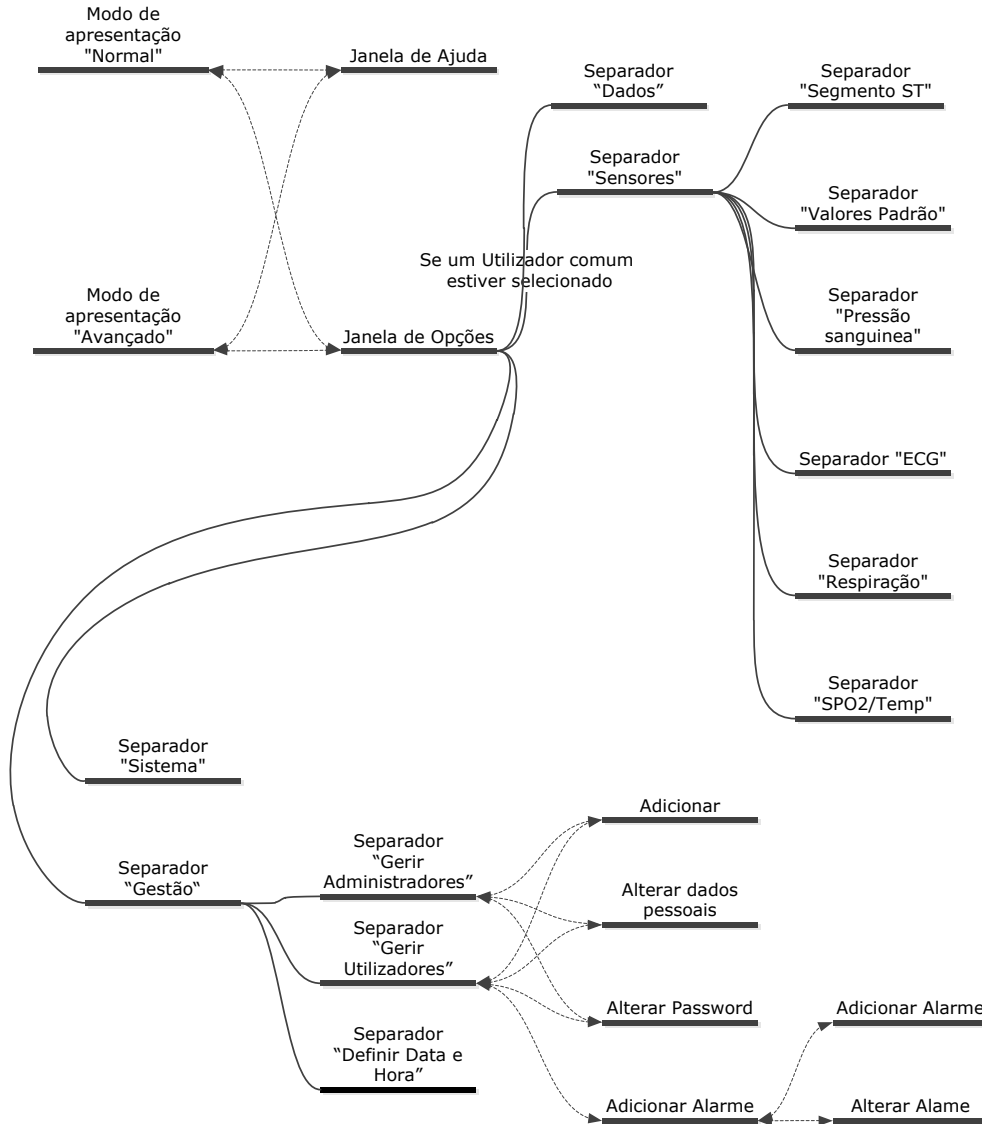
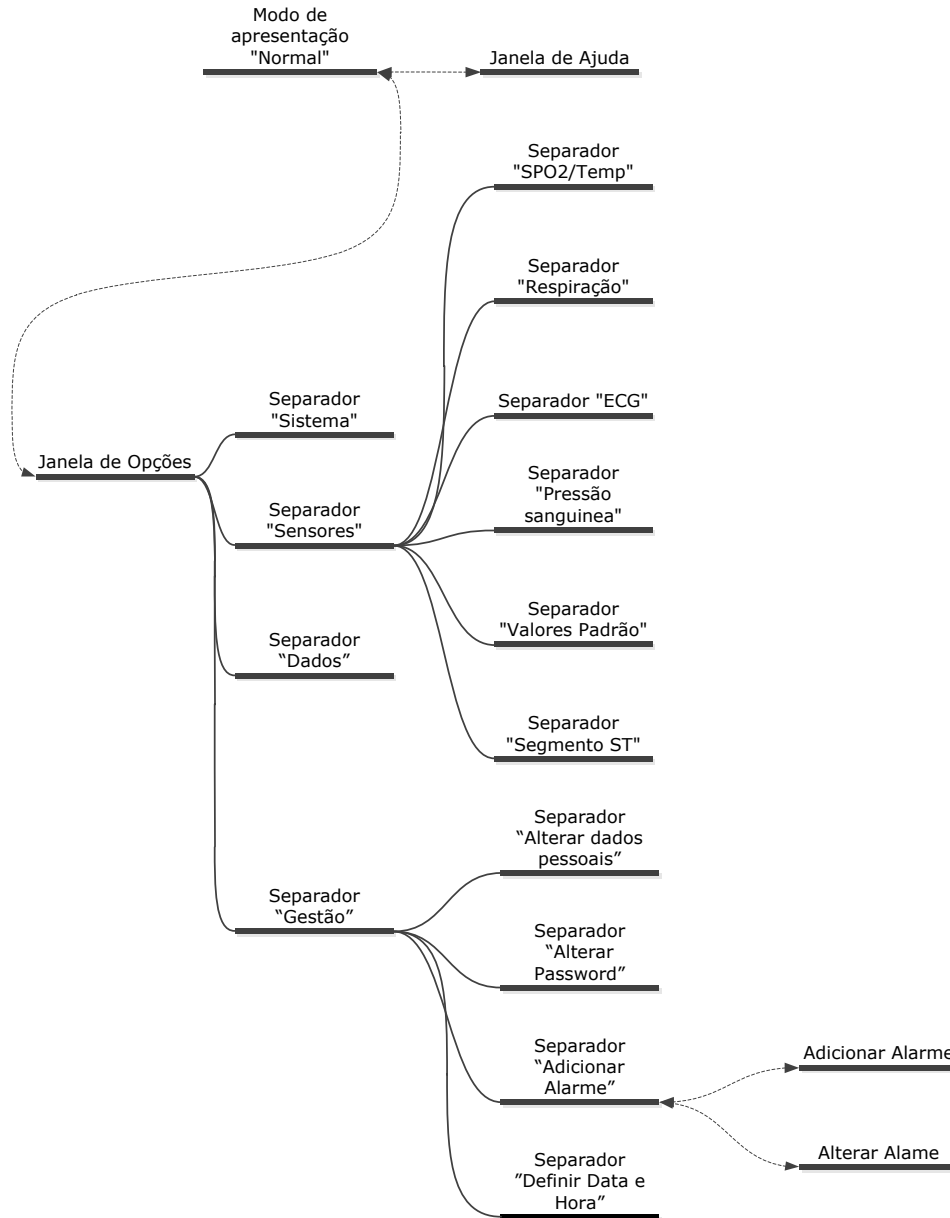


Figura 15 – Esquema das janelas associadas ao Administrador

#### 4.4.2 Paciente

Este utilizador terá acesso apenas ao modo de apresentação normal, à sua informação pessoal, não tendo acesso a informação sobre os restantes utilizadores do dispositivo.

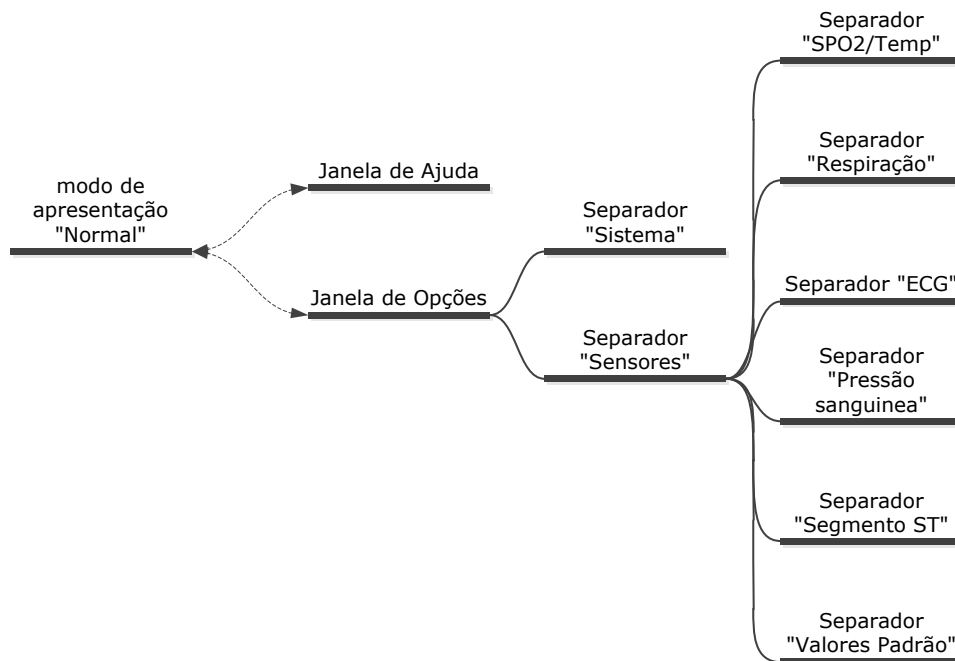


**Figura 16 - Esquema das janelas associadas ao Utilizador Comum**

#### 4.4.3 Convidado

O utilizador enquanto "Convidado" tem apenas acesso ao esquema de apresentação "Normal". Durante a iniciação do esquema de apresentação, um conjunto de valores padrão para as definições da aplicação e da placa de aquisição serão carregados, podendo o utilizador alterar estes valores na janela de opções.

No entanto, e como dito anteriormente, a aplicação só será usada para a visualização dos dados adquiridos.



**Figura 17 – Esquema das janelas associadas ao Convidado**

#### 4.5 Arquitetura da Base de Dados

A base de dados foi desenvolvida em Microsoft SQL Compact 3.5, fazendo uso da linguagem SQL, uma das linguagens mais usadas para armazenamento de dados. Esta base de dados foi escolhida pelas seguintes razões: [20;21]

- Encontra-se integrada com o IDE em uso;
- Foi desenvolvida tendo em conta o *framework* usado (.NET Compact Framework 3.5);
- Foi desenvolvida tendo em conta aplicações que vão correr em dispositivos móveis;
- Requisitos de memória reduzidos (2-3MB de memória física);
- A base de dados reside num único ficheiro e pode gerir ficheiros até 4GB;
- Está em conformidade com o conceito ACID (abordado de seguida) promovendo a salvaguarda dos dados armazenados.

A arquitetura da base de dados foi criada usando-se o Visual Paradigm for UML, tendo sido implementada mais tarde na base de dados referida anteriormente.

As Figuras 18 e 19 representam, respetivamente, a arquitetura inicial e a arquitetura atual da base de dados. As modificações apresentadas deveram-se a necessidades que surgiram durante o desenvolvimento da aplicação e à modificação de alguns requisitos.



Figura 18 - Desenho da base de dados - Inicial

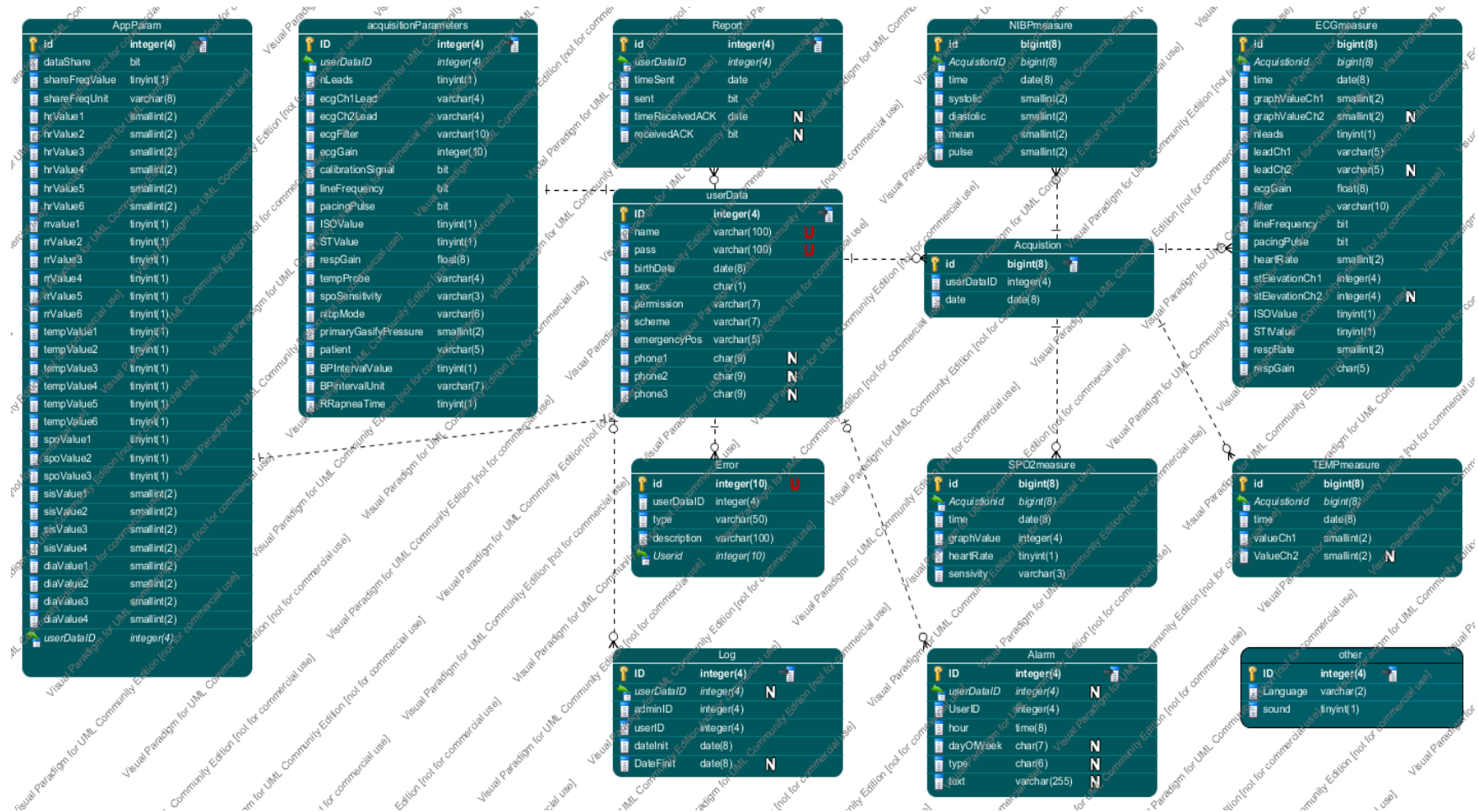


Figura 19- Desenho da base de dados - Final

Apresenta-se de seguida uma breve descrição das tabelas presentes na arquitetura atual:

- **UserData** – dados do utilizador do dispositivo (nome, idade, password);
- **Report** – guarda informação relativa ao envio e à receção de relatórios;
- **AppParameters** – guarda os dados usados para o feedback visual dos sinais fisiológicos medidos;
- **AcquisitionParameters** – guarda os dados referentes à configuração da placa de aquisição (ganhos, sensibilidades, entre outros);
- **Log** – identificar que pacientes foram selecionados durante a sessão de um administrador, a que horas e por quanto tempo;
- **Acquisition** – indica a data em que cada aquisição ocorreu, relacionando cada aquisição com o utilizador que a realizou;
- **TempMeasure** – guarda os dados recolhidos pelos sensores de temperatura;
- **SPOMeasure** – guarda os dados recolhidos pelo sensor de SpO2;
- **NIBPMeasure** – guarda os dados recolhidos pelo sensor de pressão sanguínea;
- **ECGMeasure** – guarda os dados recolhidos pelo módulo de ECG;
- **Error** – guarda informação dos erros que ocorrem durante o funcionamento da aplicação;
- **Other** – indica a linguagem usada da última vez que o programa correu;
- **DeviceData** – informação do dispositivo (identificação, sensores disponíveis)
  - Uma vez que, até à data ainda não se tinham definido os dados necessários para a identificação dos dispositivos, esta tabela não foi criada.

#### 4.5.1 Conceito ACID

ACID (atomicidade, consistência, isolamento, durabilidade) refere-se a um conjunto de propriedades que garantem que operações na base de dados ocorrem de forma segura. [22]

**Atomicidade** – requer que cada transação seja “tudo ou nada”. Isto é, se uma parte da transação falhar, toda a transação deve falhar, deixando a base de dados no estado anterior à transação.

**Consistência** – toda a informação escrita na base de dados deve ser válida de acordo com todas as regras definidas.

**Isolamento** – Apesar de poderem estar a ocorrer várias transações ao mesmo tempo, uma transação não pode ter conhecimento da outra.

**Durabilidade** – assim que a transação for corretamente executada (*committed*), a transação deve permanecer nesse estado, mesmo após falha energética ou a ocorrência de erros.



## **Capítulo 5. Resultados e Testes**

Utilizando os requisitos apresentados no Anexo II: Análise de requisitos e os diversos pontos apresentados anteriormente, procedeu-se ao desenvolvimento de uma versão beta da interface gráfica e base de dados do dispositivo de monitorização de dados vitais.

Assim, apresenta-se de seguida o resultado final do *software* desenvolvido, bem como os testes que foram realizados para verificação dos requisitos identificados. Uma breve descrição da utilização do *software* é descrita no anexo VII: Manual de utilização.

### 5.1 Resultados

No final do desenvolvimento da versão beta da aplicação, conseguiu-se obter, em relação à interface gráfica, um resultado muito semelhante ao pensado inicialmente, como se pode observar nas Figuras 20 e 21, referentes ao modo de apresentação "Normal". A principal diferença encontra-se no botão de *logout*, introduzido devido à alteração dos requisitos.



Figura 20 – Modo de apresentação "Normal". Desenho inicial.



Figura 21 - Modo de apresentação "Normal" atual.

Em relação ao modo de apresentação "Avançado" houve algumas alterações em relação ao inicialmente desenhado, como se pode observar pelas Figuras 22 e 23. Algumas dessas alterações são:

- **Adição do valor da elevação do segmento ST aos dados numéricos;**
- **Substituição da cor de fundo do gráfico por uma cor sólida, facilitando a visualização do gráfico;**
- **Remoção dos parâmetros de aquisição;**
- **Adição de uma escala aos gráficos (dependo do sinal fisiológico apresentado).**



Figura 22 – Modo de apresentação "Avançado". Desenho inicial.



Figura 23 - Modo de apresentação "Avançado" atual.

As Figuras 21 e 23 correspondem às janelas principais da aplicação e representam apenas algumas das janelas desenvolvidas para a aplicação. As restantes janelas e separadores desenvolvidos podem ser observados no Anexo IV: Interface Gráfico.

## 5.2 Teste do *software*

Existem diversas estratégias que podem ser usadas para testar o *software* desenvolvido. Por um lado pode-se esperar até que o *software* esteja completamente implementado para então se proceder aos testes da aplicação. No entanto esta aproximação não oferece bons resultados. Por outro lado, podem-se realizar testes diários ou sempre que alguma parte do sistema é desenvolvido. A estratégia de teste normalmente escolhida fica entre os dois casos anteriores, apostando-se numa visão incremental de testes, começando pelos testes de unidades individuais do programa (testes unitários), movendo-se para o teste de integração das unidades anteriores (testes de integração), culminando com testes a todo o sistema construído (testes finais).

Uma breve descrição de cada um dos testes considerados anteriormente será apresentada nos pontos seguintes.

### 5.2.1 Testes Unitários

Os testes unitários focam-se em pequenos módulos que são necessários para o desenvolvimento do projeto, como por exemplo métodos/funções ou classes específicas, sendo testados fora da aplicação principal. Assim, estes testes são usados para observar se o módulo em questão está a funcionar de acordo com o esperado, antes de se proceder à sua integração na aplicação. Como estes testes são muito específicos, pontuais e informais não necessitam de ser documentados. [7].

No caso particular deste projeto, testes deste tipo foram aplicados durante o desenvolvimento dos diversos elementos gráficos, criados propositadamente para o interface gráfico deste DM, e durante o teste de comunicação da base de dados.

### 5.2.2 Testes de integração

Os testes de integração são realizados durante a junção de módulos. Apesar dos módulos terem sido previamente testados quanto às suas funcionalidades, não há garantias de que quando os módulos comunicarem entre si se obtenha o resultado esperado, isto é, uma subfunção de um módulo pode ter um efeito adverso noutro módulo, informação pode ser perdida ou transmitida de forma errada. Assim, é necessário que, à medida que se vai construindo o *software*, se vá realizando testes para descobrir erros que tenham surgido da comunicação entre os módulos. [7]

Neste projeto optou-se por uma "Integração Cima-Baixo" (*Top-down Integration*), isto é, procedeu-se à integração dos módulos começando por uma classe principal à qual se foi progressivamente adicionando novas funções e classes, testando cada nova iteração. Desta forma, como o programa foi sendo construído e testado em pequenos incrementos, os erros são mais facilmente identificados e corrigidos e a comunicação entre módulos é testada de uma forma mais completa. [7]

Como descrito estes testes foram sendo realizados ao longo do desenvolvimento da aplicação, tendo, por vezes, levado à reestruturação da ideia inicial.

### 5.2.3 Testes finais

Estes testes, ao contrário dos testes anteriores, são realizados sobre uma versão da aplicação que se considera terminada. Com estes testes pretende-se compreender se os requisitos apresentados inicialmente foram implementados, se a aplicação se está a comportar como esperado e se houve algum erro que não foi identificado nos testes anteriores. Apesar de não ser exigido pela norma IEC 62304, procedeu-se à documentação dos testes finais realizados à aplicação de forma a comprovar que os requisitos considerados inicialmente foram implementados.

Assim, o documento desenvolvido foi dividido em secções, tendo em conta a organização dos requisitos apresentados no Anexo II: Análise de Requisitos. Para a descrição dos testes elaborou-se um documento com os seguintes campos:

- **Teste** – Designação e descrição do teste a realizar;
- **Resultado esperado** – Descrição do resultado esperado ao realizar o teste;
- **Resultado obtido** – Descrição do resultado obtido aquando da realização do teste.

O documento elaborado pode ser observado no Anexo VI: Testes Finais.



## **Capítulo 6. Conclusão**

No início do projeto, o estudo dos dispositivos existentes no mercado, permitiu concluir que há uma inexistência, ou muito baixo número, de dispositivos médicos para utilização em ambiente doméstico, assim, como uma falta de continuidade e acompanhamento após a medição do sinal vital. Este dispositivo procura explorar e responder a essa possível necessidade do mercado.

Assim, durante o desenvolvimento do presente trabalho, procurou-se que a interface desenvolvida assentasse em duas características principais, a facilidade de utilização (*user-friendly*) e a usabilidade. Assim, espera-se que estas características, aliadas a uma integração de diversos módulos de aquisição num único dispositivo, e a uma aposta na óptica do utilizador comum, isto é, o desenvolvimento de uma interface tendo em conta o utilizador sem formação na área da saúde (oferecendo e acompanhamento), sejam fatores de diferenciação dos restantes dispositivos e interfaces presentes no mercado.

Assim, a primeira fase do projeto passou pela definição dos requisitos para o dispositivo a desenvolver. Durante esta fase houve a identificação dos ambientes de utilização e futuros utilizadores do dispositivo, assim como estudo das diretivas e normas necessárias para o correto desenvolvimento e certificação de dispositivos médicos. Durante esta fase realizaram-se algumas reuniões com os engenheiros da empresa de forma a identificar uma primeira versão dos requisitos. Também durante esta fase, desenvolveu-se o desenho da arquitetura do *software* tendo em conta todos os requisitos levantados.

A segunda fase passou pela implementação dos requisitos identificados, isto é, o desenvolvimento da interface e da aplicação. Para a realização desta fase houve a necessidade de aprender uma nova linguagem de programação, a linguagem C#. Para a rápida adaptação a esta nova linguagem, contribui o conhecimento adquirido sobre a linguagem JAVA. A aliar a esta nova linguagem houve o desenvolvimento das competências adquiridas com a linguagem de SQL.

Na terceira e última fase procedeu-se à integração dos diversos componentes da interface gráfica entre si e com a base de dados, e aos testes do sistema. Pode-se concluir que a integração entre estes elementos foi executada com sucesso, apesar da necessidade de corrigir alguns erros detetados.

Pode-se concluir que os objetivos propostos para este projeto foram atingidos, isto é, o desenvolvimento da interface gráfica e da base de dados foi realizado com sucesso, havendo ainda algum trabalho a realizar para que a libertação do dispositivo no mercado seja possível. Este trabalho é indicado de seguida, no capítulo 6.1 Trabalho Futuro



### 6.1 Trabalho Futuro

O *software* desenvolvido durante este projeto, interface gráfica e base de dados, representa apenas metade do dispositivo médico. Assim, como trabalho futuro, e para que uma primeira versão do dispositivo médico possa ser libertada, será necessário:

Desenvolver o protocolo de comunicação com o sistema de comunicação e a sua integração com a interface gráfica;

Estudar e desenvolver uma forma de comunicação remota com o dispositivo Médico (ex: *Web Service*) e a sua integração com o *software* desenvolvido;

Desenvolver um *software* para desktop que permita a análise dos dados adquiridos e guardados no cartão de memória e/ou transferidos;

Testar a interface gráfica quanto à sua usabilidade e obter feedback dos utilizadores principais.

A Interface gráfica e a base de dados foram desenvolvidas tendo em conta a integração futura do sistema de aquisição e do *Web Service* (ou solução semelhante), no entanto, torna-se complicado prever se as decisões tomadas serão as mais corretas no sentido de facilitar a integração. Assim, o *software* desenvolvido poderá vir a sofrer alterações durante a integração dos novos elementos.

## **Bibliografia**

1. Report: Europe is facing a demographic challenge – Ambient Assisted Living offers Solutions; March 2006
2. Ladner, S. C. (2002). *Fundamentals of nursing: Standards and Practices*. Delmar/Thomson Learning.
3. Mahadevan, S., & Garmel, G. M. (2005). *An Introduction to Clinical Emergency Medicine*. Cambridge.
4. Clifford, Gari D.; Azuaje, Francisco; McSharry, Patrick E.; *Advanced Methods and tools for ECG Data Analysis*; Artech House, 2006.
5. ADINSTRUMENTS;  
<http://www.adinstruments.com/solutions/research/Electrophysiology-and-Cell-Biology/ECG,-EKG/>. [Online] [Citação: 22 de Março de 2012]
6. Gonçalves, Emeline Alves; Project Report – Monitorização remota de sinais vitais; 10 de Setembro de 2007
7. Pressman, Roger S., PH.D.; *Software Engineering – A Practitioner’s Approach*; Seventh Edition; McGrawHill
8. Pereira, Carlos. “Desenvolvimento de uma Interface Gráfica e de uma Base de Dados para um Dispositivo Médico de Electroterapia”. Coimbra, Junho 2011
9. IEC 62304:2006 – Medical Device software – Software life cycle processes
10. Council Directive 93/42/EEC, 14 June 1993 – Concerning Medical Devices
11. Directive 2007/47/EC of the European Parliament and of the council, 5 September 2007 – amending Council Directive 90/385/EEC on the approximation of the laws of the Member States relating to active implantable medical devices, Council Directive 93/42/EEC concerning medical devices and Directive 98/8/EC concerning the placing of biocidal products on the market
12. ISO 14971:2007 – Medical Devices – Application of risk management to medical devices
13. MEDDEV 2.4/1 Rev. 9, June 2010 – Guidelines Relating to The Application of the council Directive 93/42/EEC on Medical Devices; European Commission
14. Friendly ARM mini2440 – User’s Manual
15. e-com Systems. <http://www.e-consystems.com/WindowsCE5vs6.asp>. [Online] [Citação: 2 de Fevereiro de 2012]
16. Nawatt labs.  
[http://nawattlabs.com/index.php?option=com\\_content&view=article&id=18:windows-linux&catid=9:blog&Itemid=5](http://nawattlabs.com/index.php?option=com_content&view=article&id=18:windows-linux&catid=9:blog&Itemid=5). [Online] [Citação: 2 de Fevereiro de 2012]

17. Stackoverflow. <http://stackoverflow.com/questions/326516/windows-ce-vs-embedded-linux>. [online] [Citação: 3 de Fevereiro de 2012]
18. BEC Systems. <http://bec-systems.com/site/462/embedded-linux-versus-windows-ce>. [Online] [citação: 3 de Fevereiro de 2012]
19. C# language specification ECMA-Internation. 4th edition june 2006
20. Microsoft. <http://technet.microsoft.com/en-us/library/ms172037.aspx> [Online] [Citação: 21 de Março de 2012]
21. Microsoft. <http://technet.microsoft.com/en-us/library/ms171986.aspx> [Online] [Citação: 21 de Março de 2012]
22. Microsoft. <http://technet.microsoft.com/en-us/library/aa998171%28v=EXCHG.65%29.aspx> [Online] [Citação: 14 de Junho de 2012]

# **Anexos Confidenciais**