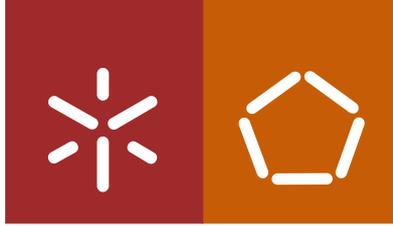


Universidade do Minho
Escola de Engenharia

João Henrique Ribeiro Valente

Monitorização de ECG de Pacientes em Mobilidade



Universidade do Minho
Escola de Engenharia

João Henrique Ribeiro Valente

Monitorização de ECG de Pacientes em Mobilidade

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica

Trabalho efetuado sob a orientação de:
Alexandre Santos
Joaquim Macedo

outubro de 2014

ESTA DISSERTAÇÃO NÃO FOI ESCRITA AO ABRIGO DO NOVO
ACORDO ORTOGRÁFICO.
ESTE DOCUMENTO FOI ESCRITO EM L^AT_EX.

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE
APENAS PARA EFEITOS DE INVESTIGAÇÃO MEDIANTE
DECLARAÇÃO ESCRITA DO INTERESSADO QUE A TAL SE
COMPROMETE;

Universidade do Minho, ___/ ___/ ____

Assinatura: _____

AGRADECIMENTOS

A presente secção serve para expressar o meu agradecimento a todas as pessoas que de alguma forma contribuíram para a realização deste projecto.

Assim, gostaria de agradecer aos professores Alexandre Santos e Joaquim Macedo, pela orientação e supervisão deste projecto e pela disponibilidade demonstrada ao longo de todo o trabalho.

Aos meus amigos e colegas João Gaspar e Igor Vasilevskiy, estudantes de Medicina, ao Nuno Alves, à Priscila Alves, e ao Ricardo Marques dos cursos de Informática Médica, Bioengenharia e Engenharia Informática, respectivamente, agradeço todo o apoio e colaboração prestados.

Ao Rui Couto, aluno do curso de Engenharia Informática agradeço a disponibilidade para me auxiliar no *debugging* do código em *Android* na aplicação móvel.

À equipa do *BITalino* agradeço por me ter facultado algumas informações relevantes para o desenvolvimento do projecto.

Aos meus pais e aos meus irmãos, agradeço do fundo do coração toda a sua paciência, carinho, apoio e compreensão durante a execução deste projecto.

Ao meu padrinho, Luís Gonçalves, um agradecimento especial pela partilha de opiniões e apoio ao longo deste percurso.

Aos meus colegas de curso que me acompanharam em todo o percurso académico agradeço a sua amizade e companheirismo.

Resumo

Devido à actual conjuntura económica e social, é necessário adoptar medidas ao nível das instituições de saúde que visem uma redução de custos. Estas medidas devem fazer-se acompanhar de uma política de qualidade na prestação de cuidados de saúde. Tem de se verificar o cumprimento dos requisitos necessários para um tratamento adequado dos pacientes e uma diminuição dos erros cometidos pelos profissionais de saúde.

Deslocações regulares às Unidades de Saúde traduzem-se num maior congestionamento dos serviços das várias unidades, conduzindo a uma menor Qualidade do Serviço prestado ao paciente.

Os últimos desenvolvimentos nos campos da tecnologia de sensores biomédicos e dispositivos móveis, combinados com maiores capacidades das comunicações sem fios, têm tornado possível o surgimento de novos paradigmas de monitorização de saúde. Este tipo de paradigmas visa uma melhoria da qualidade de vida dos pacientes e uma diminuição da afluência às Unidades de Saúde.

As funcionalidades cada vez maiores dos dispositivos móveis como os *Personal Digital Assistants (PDAs)* e os *smartphones* fazem deles um componente vital em sistemas de monitorização de saúde.

Neste trabalho, define-se uma arquitectura de um sistema de monitorização remota do sinal de Electrocardiograma (*ECG*) de pacientes em mobilidade.

Para este projecto recorre-se ao *BITalino* como plataforma de aquisição de sinais vitais.

A utilização das tecnologias no contexto da saúde implica o cumprimento de elevados padrões de segurança.

Deve ser possível a transmissão de informação dos vários utentes sem que esta esteja sujeita a possíveis ataques informáticos (ataques do tipo *man in the middle* e *DOS (Denial of Service)* ou *DDOS (Distributed Denial of Service)* ao servidor).

Abstract

Due to the current economic and social context, it is necessary to adopt measures at the healthcare providers' level which presuppose a cost reduction.

This leverage in costs ought to be accompanied with a healthcare quality policy.

This policy determines the accomplishment of the necessary requirements for a proper treatment of patients and a reduction of the errors committed by the healthcare personnel.

Regular commuting to the Healthcare Units corresponds to a major congestion of the services of the several units which yields to a decrease in the quality of the service provided to the patient.

The most recent developments in the core fields of Biomedical Sensor Technology and handheld devices, associated with greater capacities of wireless communications, have made possible the appearance of new Healthcare Monitoring paradigms whose purpose is to improve the quality of life of the patients and to reduce the overall congestion of the Healthcare Units.

The increased functionalities of the mobile devices such as *Personal Digital Assistants (PDAs)* and *smartphones* make themselves a vital component in innovative systems of Healthcare Units.

In this Thesis, an architecture of a remote monitoring system of the Electrocardiogram Signal (*ECG*) of patients in mobility is defined, using *BITalino* as the Vital Sign Acquisition platform.

The recurrent use of technologies in the Healthcare context implies the fulfillment of high Security Standards.

This way data transmission of the several patients is possible without being too susceptible to man in the middle attacks or denial of service (*DoS*) and distributed denial of service (*DDoS*) attacks at the server side.

CONTEÚDO

1	INTRODUÇÃO	1
1.1	Definição do Problema	1
1.2	Motivação	2
1.3	Objectivos	3
1.4	Metodologia	5
1.5	Resultados Obtidos	6
1.6	Estrutura do documento	7
2	ESTUDO DOS SINAIS VITAIS	9
2.1	Sistema Cardiovascular	9
2.2	Características e Morfologia do Sinal de ECG	10
2.3	Monitor de Holter	16
2.4	Ritmo Sinusal Normal	17
2.5	Arritmias Cardíacas	18
2.5.1	Bradycardia Sinusal	19
2.5.2	Taquicardia Sinusal	19
2.5.3	Fibrilhação Auricular (AF)	20
2.5.4	Arritmia Sinusal	20
2.5.5	Bloqueio de Ramo	21
2.5.6	Flutter Auricular (AFL)	21
3	TÉCNICAS DE AMOSTRAGEM, FILTRAGEM E COMPRESSÃO	23
3.1	Amostragem	23
3.2	Subamostragem	23
3.3	Filtragem	24
3.3.1	Flutuação da linha de base	25
3.3.2	Interferência da Rede Eléctrica	26
3.3.3	Ruído de origem muscular	28
3.3.4	Detecção do complexo QRS	30
3.4	Compressão	31
4	ALGORITMOS PARA AQUISIÇÃO E PROCESSAMENTO	33
4.1	Algoritmos de Redução de dados	33
4.1.1	Algoritmo <i>Turning Point</i>	33
4.1.2	Algoritmo AZTEC (Amplitude-Zone-Time-Epoch-Coding)	36
4.1.3	Algoritmo de CORTES (Coordinate-Reduction-Time-Encoding-System)	40
4.2	Detecção do complexo QRS	42
4.2.1	Algoritmo de <i>Pan-Tompkins</i>	42
4.2.2	Algoritmo MOBD (Multiplication Of the Backward Difference)	45

4.2.3	Algoritmo de <i>Okada</i>	47
4.3	Compressão	51
4.3.1	Compressão no domínio dos tempos	51
4.3.1.1	Codificação de Huffman	51
4.3.1.2	<i>Lempel-Ziv-Welch (LZW)</i>	52
4.3.1.3	<i>Differential Pulse Code Modulation (DPCM)</i>	52
4.3.2	Codificação por Transformada	54
4.3.2.1	Transformada de <i>Fourier</i> discreta	54
4.3.2.2	Transformada rápida de <i>Fourier</i>	54
4.3.2.3	Transformada do Cosseno Discreta	54
4.3.2.4	Transformada de <i>Wavelet</i> Discreta	55
4.3.2.5	Transformada rápida de <i>wavelet</i>	56
5	PROJECTOS RELACIONADOS	57
5.1	Sistemas de aquisição de Biosinais	57
5.1.1	BITalino	57
5.1.2	e-Health Sensor Platform	60
5.1.3	Sensium	61
5.1.4	Pluto	61
5.1.5	CPOD (Crew Physiologic Observation Device)	62
5.1.6	Shimmer	63
5.1.7	Zephyr BioHarness	64
5.1.8	AMON	65
5.2	Sistemas de Monitorização Remota	67
5.2.1	CodeBlue	67
5.2.2	MobiHealth	68
5.2.3	Sensium	69
5.2.4	Vital-Jacket	70
5.2.5	HM4All	71
5.2.6	Lifeguard	72
5.2.7	UbiMon	73
5.2.8	<i>AlarmNet</i>	74
5.2.9	<i>Aid-N</i>	75
5.3	Análise das características dos diferentes sistemas	77
6	DESENVOLVIMENTO DO PROTÓTIPO DO SISTEMA	79
6.1	Arquitectura do Sistema	79
6.1.1	Contextualização	79
6.1.2	Módulo de Aquisição	80

6.1.2.1	API do BITalino	82
6.1.3	Módulo de Processamento	83
6.1.4	Módulo de compressão de dados	84
6.1.5	Módulo de transmissão de dados	84
6.1.6	Módulo de <i>log</i>	84
6.1.7	Servidor Central	85
6.1.8	Estação de Visualização	86
6.2	Plataformas de teste e de dados biométricos	87
6.2.1	Plataformas de teste	87
6.2.2	Plataformas de dados biométricos	88
6.2.3	Comparação dos Formatos de Ficheiros dos <i>Datasets</i> com o formato <i>EDF</i>	89
6.3	Implementação do Projecto	91
6.3.1	Análise <i>MATLAB</i>	91
6.3.2	Análise em plataforma móvel	97
7	CONCLUSÕES E TRABALHOS FUTUROS	109
7.1	Conclusões	109
7.2	Trabalhos Futuros	110
	Referências Bibliográficas	112

LISTA DE FIGURAS

Figura 1	Representação esquemática do sistema de condução cardíaco [1].	9
Figura 2	Configuração geral das 3 derivações de Einthoven (Triângulo de Einthoven) [2].	10
Figura 3	ECG de 12 derivações [3].	11
Figura 4	Disposição das 6 derivações de Wilson.	12
Figura 5	Utilização de um ECG de 12 derivações numa unidade de saúde [3].	12
Figura 6	Disposição dos eléctrodos sobre o coração [4].	13
Figura 7	Disposição dos eléctrodos segundo Einthoven [4].	13
Figura 8	Disposição dos eléctrodos nos dedos [4].	14
Figura 9	Registo da derivação II apresentando a onda P, o complexo QRS e as ondas T e U [5].	14
Figura 10	Medição dos vários parâmetros de um sinal de ECG. (Fonte: Hewlett Packard Company, Palo Alto, Calif.)	15
Figura 11	Monitor de Holter [6].	16
Figura 12	Ritmo Sinusal Normal [7].	17
Figura 13	Bradicardia Sinusal [8].	19
Figura 14	Taquicardia Sinusal [9].	20
Figura 15	Fibrilhação Auricular [8].	20
Figura 16	Arritmia Sinusal [10].	21
Figura 17	Flutter Auricular [8].	22
Figura 18	Características dos filtros ideal, de <i>Butterworth</i> , <i>Chebyshev</i> tipo I e <i>Chebyshev</i> tipo II [11].	25
Figura 19	Exemplo de um Filtro Notch Adaptativo [12].	29
Figura 20	Sinal de ECG com e sem Ruído.	30
Figura 21	Detecção do Complexo QRS através de filtros passa-banda com larguras de banda nominais de $\pm 2.5\text{Hz}$, $\pm 8.3\text{Hz}$ e $\pm 11\text{Hz}$ [12].	31
Figura 22	Algoritmo <i>Turning Point</i> [13].	34
Figura 23	Algoritmo <i>Aztec</i> [13].	37
Figura 24	Algoritmo <i>AZTEC</i> (continua na Figura 25).	38
Figura 25	Algoritmo <i>AZTEC</i> (continuação).	39
Figura 26	Algoritmo de <i>CORTES</i> [13].	42

Figura 27	Algoritmo <i>MOBD</i>	46
Figura 28	BITalino: Versão <i>Board</i> [14].	59
Figura 30	BITalino: Versão <i>Freestyle</i> [15].	59
Figura 31	e-Health Sensor Platform [16].	60
Figura 32	<i>e-Health Sensor Platform</i> com <i>Arduino</i> (à esquerda) e <i>Raspberry-Pi</i> (à direita) [16].	60
Figura 33	Sensium [17].	61
Figura 34	Pluto [18].	61
Figura 35	CPOD [19].	62
Figura 36	Shimmer [20].	63
Figura 37	BioHarness [21].	64
Figura 38	AMON [22].	65
Figura 39	CodeBlue [23].	68
Figura 40	Sensium [24].	69
Figura 41	LifePebble [25].	69
Figura 42	Vital-Jacket [26] e [27].	70
Figura 43	HM4All [28].	72
Figura 44	Lifeguard [19].	73
Figura 45	AlarmNet [29].	74
Figura 46	Arquitectura do Sistema.	79
Figura 47	Esquema Relacional da Base de Dados.	85
Figura 48	Sinal de <i>ECG</i> original/ não filtrado.	91
Figura 49	Zoom de 2 períodos do sinal original.	92
Figura 50	Sinal de <i>ECG</i> subamostrado por um factor de 2.	92
Figura 51	Zoom de 2 períodos do sinal subamostrado.	93
Figura 52	Sinal de <i>ECG</i> sem Flutuação de Linha de Base.	94
Figura 53	Zoom de 2 períodos de <i>ECG</i> sem Flutuação de Linha de Base.	94
Figura 54	Sinal de <i>ECG</i> sem Flutuação de Linha de Base e ruído de <i>EMG</i>	95
Figura 55	Zoom de 2 períodos do sinal de <i>ECG</i> sem Flutuação de Linha de Base e ruído de <i>EMG</i>	95
Figura 56	Sinal de <i>ECG</i> após os vários estágios de filtragem.	96
Figura 57	Zoom de 2 períodos do sinal de <i>ECG</i> após os vários estágios de filtragem.	96
Figura 58	Sinal de <i>ECG</i> filtrado com a delineação das ondas <i>P</i> , <i>Q</i> , <i>R</i> , <i>S</i> e <i>T</i>	97
Figura 59	Método <i>Acquire()</i>	100
Figura 60	Configuração geral dos eléctrodos.	101
Figura 61	Geração dos gráficos na aplicação <i>Android</i>	101
Figura 62	<i>Activity</i> de <i>Login</i>	102
Figura 63	<i>SplashScreen</i>	102
Figura 64	<i>Intent</i> para permitir a utilização do <i>Bluetooth</i>	103
Figura 65	Dispositivo de aquisição e aplicação móvel.	103
Figura 66	Interface Gráfica da Aplicação desenvolvida em <i>Android</i>	104

Figura 67 Diagrama da Aplicação móvel desenvolvida. . 104

LISTA DE TABELAS

Tabela 1	Parâmetros do sinal de <i>ECG</i> normal. (Adaptado de [30], [31] e [32]).	17
Tabela 2	Tabela de códigos <i>ASCII</i> [33].	53
Tabela 3	Coeficientes das <i>wavelets</i> de <i>Daubechies</i> para $N=2$, $N=4$, $N=8$ e $N=12$ [11].	56
Tabela 4	Características dos Sistemas de Aquisição . . .	66
Tabela 5	Características de sistemas de monitorização. .	76
Tabela 6	Características principais dos sensores de Electrocardiograma (<i>ECG</i>) e de Acelerometria de cada sistema	77
Tabela 7	Estrutura dos pacotes de dados do <i>BITalino</i> . . .	81

LISTA DE ALGORITMOS

1	Algoritmo Turning Point	35
2	Algoritmo de <i>Okada</i>	50

LISTA DE ACRÓNIMOS

3G	Terceira Geração
AVL	augmented Voltage Left
AVR	augmented Voltage Right
AVF	augmented Voltage Foot
AC	Alternate Current
A/D	Analogue-to-digital
ADC	Analogue-to-digital Converter
ADT	Android Development Tools
AES	Advanced Encryption Standard
AF	Atrial Fibrillation
AFL	Atrial Flutter
AHA	American Heart Association
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AV	Auriculo-Ventricular
AZTEC	Amplitude-Zone-Time-Epoch-Coding
BAN	Body Area Network
BPM	batimentos por minuto
BSN	Body Sensor Network
CDMA	Code division multiple access
CMRR	Common Mode Rejection Ratio
CORTES	Coordinate-Reduction-Time-Encoding System
CR	Compression Ratio
CRC	Cyclic Redundancy Check
CSE	Common Standards for quantitative Electrocardiography
DC	Direct Current

DCT Discrete Cosine Transform
DOS Denial-of-Service
DDOS Distributed Denial-of-Service
DFT Discrete Fourier Transform
DPCM Differential Pulse Code Modulation
DWT Discrete Wavelet Transform
FFT Fast Fourier Transform
FIFO First In, First Out
ECG Electrocardiograma
EMG Electromiografia
EDA Electrodermal Activity
EDF European Data Format
FIR Finite Impulse Response
FSK Frequency-shift keying
FWT Fast Wavelet Transform
GSM Global System for Mobile Communications
GPRS General Packet Radio Service
GPS Global Positioning System
GSR Galvanic Skin Response
HTTP Hypertext Transfer Protocol
IDE Integrated Development Environment
IEEE Institute of Electrical and Electronics Engineers
IIR Infinite Impulse Response
ISM Industrial, Scientific and Medical
ISR Interrupt Service Routine
IT Instituto de Telecomunicações
LAN Local Area Network
LED Light Emitting Diode
LSB Least Significant Byte

LZW Lempel-Ziv-Welch
MAC Media Access Control
MOBD Multiplication of the Backward Difference
MSB Most Significant Byte
ONU Organização das Nações Unidas
OSI Open Systems Interconnection model
PC Personal Computer
PDA Personal Digital Assistant
PIR Passive Infrared
PRD Percent Root-Mean-Square Difference
RAM Random-Access Memory
ROM Read-Only Memory
SA Sino-Auricular
SBC Sub-Band Coding
SD Secure Digital
SDK Software Development Kit
TP Turning Point
UMTS Universal Mobile Telecommunications System
URL Uniform Resource Locator
USART Universal Synchronous/Asynchronous Receiver/Transmitter
VLA Voltage Left Arm
VLC Variable Length Coding
VLL Voltage Left Leg
VRA Voltage Right Arm
WI-FI Wireless Fidelity
WLAN Wireless Local Area Network
WSN Wireless Sensor Network

INTRODUÇÃO

1.1 DEFINIÇÃO DO PROBLEMA

Um dos principais problemas com que se deparam os profissionais de saúde, tanto em Portugal, como noutros Países, tem a ver com os custos do sector que resultam da prestação de cuidados aos utentes nas unidades de saúde centrais e alas de internamento.

Uma grande afluência a estas unidades contribui para o aumento do volume da despesa, assim como a ocorrência de erros por parte dos profissionais de saúde.

Existem cada vez mais pacientes crónicos e um maior envelhecimento da população, a par de uma maior esperança média de vida.

Segundo o Boletim da Organização das Nações Unidas (*ONU*) de 2001 [34], o número estimado de pessoas com idade igual ou superior a 60 anos era de 605 milhões em 2000. Prevê-se que este número passe para cerca de 2 mil milhões em 2050 .

Muitos dos utentes que recorrem diariamente às Instituições de Saúde, geralmente não possuem um quadro clínico que justifique a sua deslocação a este tipo de instituições.

Este tipo de ocorrência traduz-se num maior congestionamento dos serviços, afectando a Qualidade do Serviço prestado e acarretando mais custos de transporte para o paciente.

Por outro lado, grande parte das alas de internamento destas unidades encontram-se sobrelotadas.

Neste caso, é necessário libertar camas para ser possível a prestação de serviços de saúde a pacientes com um quadro clínico grave.

Os pacientes que tenham sido submetidos a um regime de internamento médico e com um determinado historial clínico, requerem um acompanhamento clínico ajustado às suas necessidades.

Este acompanhamento pode passar pela monitorização à distância, no seu domicílio de alguns sinais vitais que sejam relevantes para aferição da existência de um novo quadro clínico passível de readmissão a nível hospitalar.

Tendo em conta estes pressupostos, torna-se imprescindível reduzir a afluência de utentes às unidades de saúde como forma de reduzir custos desnecessários.

Assim, a melhor solução para o problema passa por um acompanhamento médico do paciente no domicílio, através da monitorização remota dos seus sinais vitais.

Esta monitorização deve ser realizada com recurso a software e a tecnologias de informação adequados para este tipo de situação, respeitando sempre a confidencialidade da informação.

1.2 MOTIVAÇÃO

O maior desenvolvimento dos processos de comunicação sem fios para transmissão de dados a curto e longo alcance tem vindo a permitir uma maior mobilidade e interconectividade dos vários dispositivos biomédicos. Tal permite a verificação do estado de um paciente sem a necessidade de este se encontrar hospitalizado. Neste caso, o paciente pode continuar com a sua rotina diária enquanto é realizada essa verificação.

O sector da saúde depende da tecnologia e dos novos avanços tecnológicos, juntamente com um certo factor de inovação, para fornecer aos médicos e restantes profissionais de saúde um conjunto de alternativas para detecção e tratamento de certas patologias.

Os ataques informáticos efectuados com base nas vulnerabilidades dos sistemas estão a tornar-se cada vez mais sofisticados.

Pode-se considerar que as técnicas criptográficas existentes permitem preservar a integridade da confidencialidade dos dados relativos aos pacientes.

A sua integridade pode ser preservada através da encriptação desse tipo de informação e pela securização das comunicações.

Esta securização das comunicações pode ser efectuada por recurso a protocolos de comunicação implementados sobre determinados protocolos de segurança já existentes.

A confidencialidade dos dados pode ser garantida pela anonimização dos dados relativos a cada paciente do lado da aplicação cliente.

No mercado das aplicações móveis tem-se verificado uma grande expansão e desenvolvimento, o que torna este sector adequado ao desenvolvimento de soluções na área da saúde devido ao seu carácter de transversalidade.

Actualmente, existe uma grande incidência de doenças cardiovasculares em todas as faixas etárias. Este tipo de patologia resulta de hábitos alimentares incorrectos e de um estilo de vida sedentário, o que requer a monitorização contínua desses pacientes, dado que apresentam uma grande taxa de mortalidade.

Em Portugal, mais de 30 % dos óbitos registados devem-se à prevalência das doenças cardiovasculares.

O número crescente nas últimas décadas deste tipo de pacientes e dos pacientes crónicos em geral tem-se traduzido em mais custos para o sector da saúde devido a um maior congestionamento das instalações hospitalares.

De modo a reduzir esta afluência às Unidades de Saúde, propõe-se neste trabalho, nos casos em que tal seja possível, que se efectue a

monitorização remota de certos parâmetros vitais. A necessidade de permanência dos pacientes nas unidades de saúde pode ser avaliada com base na triagem dos mesmos em função do carácter de urgência.

1.3 OBJECTIVOS

Monitorização de ECG de Pacientes em Mobilidade

1. Estudo e Exploração de Plataformas de Aquisição de Sinais Vitais

O *BITalino* foi a plataforma de aquisição de sinais vitais seleccionada para desenvolvimento do projecto, uma vez que tem por base a modularidade e reúne os requisitos necessários para medição dos valores de *ECG* e de acelerometria em tempo real. Permite também a visualização dos dados através de uma *API* gráfica (*Python, Java, LabView, Microsoft Visual C#* ou *Matlab*). Um dos objectivos deste projecto prende-se com a análise e exploração de algumas plataformas de aquisição de sinais vitais e a realização de um estudo comparativo entre as varias plataformas, tomando o *BITalino* como plataforma base para essa mesma comparação.

2. Análise e eventual utilização de técnicas de Amostragem, Filtragem e Compressão

Os sinal de *ECG* é um sinal analógico sendo necessário proceder à digitalização e quantização deste tipo de sinais uma vez que os computadores e os dispositivos móveis operam no domínio discreto.

As técnicas de amostragem e de subamostragem podem ser aplicadas ao sinal original para efeitos de compactação desse mesmo sinal e redução da banda de frequências necessária para o transmitir.

Essa compactação é efectuada por redução do número de amostras presente no sinal original através de técnicas como a decimação.

Este tipo de sinais é susceptível a ruído proveniente de várias fontes, nomeadamente da rede eléctrica, da deslocação dos eléctrodos relativamente à sua posição inicial e de certos artefactos originados pela contracção muscular. É necessário proceder a uma filtragem do sinal previamente à fase de detecção do complexo QRS e da onda P.

A compressão de dados é essencial para sinais fisiológicos continuamente recolhidos por aplicações de monitorização no contexto da saúde móvel (*mobile health*) de modo a prolongar o

tempo de vida da bateria através da redução dos custos de transmissão e da banda do sinal necessária para o transmitir.

No contexto dos biosinais considerados no âmbito deste projecto, pode-se considerar uma compressão sem perdas ou uma compressão com perdas de informação.

A compressão sem perdas de informação não é geralmente utilizada.

Esta apresenta poucas perdas de informação relativamente ao sinal original juntamente com baixos rácios de compressão.

No caso da compressão com perdas, ocorre perda significativa da informação, no entanto, consegue-se atingir rácios de compressão bastante superiores.

Este último tipo de compressão, pode ser utilizado desde que se garanta que grande parte da informação clinicamente relevante não é eliminada por este processo de compactação de dados.

3. Detecção de Situações Críticas a partir do Sinal de ECG.

Alguns dos parâmetros a avaliar para detecção de situações críticas são:

- A presença ou a ausência de Ondas P. A onda P pode-se encontrar ausente na Fibrilhação Auricular, no ritmo idioventricular e no ritmo juncional.
- A duração do complexo *QRS*. No caso do Bloqueio de Ramo, que corresponde a uma anomalia cardíaca, a duração dos complexos *QRS* é superior a 100 ms. O valor normal para esta duração encontra-se referenciado na Tabela 1.
- O duração do segmento *PR*. Um tempo de despolarização das aurículas longo pode significar que existe um problema de condução cardíaco desde o nó sino-auricular que corresponde ao local onde tem origem o estímulo eléctrico até ao nó auriculo-ventricular que permite a propagação desse mesmo estímulo à aurícula esquerda.

4. Desenvolvimento de uma aplicação protótipo de monitorização do ECG em mobilidade

Como forma de exemplificar uma possível solução para o problema proposto, será desenvolvida uma aplicação móvel para monitorização remota do *ECG* em mobilidade.

A definição de uma primeira arquitectura do sistema será objecto de estudo tal como apresentado na secção Metodologia do presente documento.

1.4 METODOLOGIA

Como forma de tentar solucionar o problema acima descrito, foi desenvolvida uma aplicação móvel para o sistema *Android* para detecção de situações críticas/anomalias no sinal de *ECG* dos pacientes e notificação dos médicos e enfermeiros que acompanham o processo clínico do Paciente caso se confirme a existência de uma situação crítica ou urgente.

No contexto da aplicação afiguram-se 3 cenários possíveis:

- Pacientes sem grandes sintomas e que necessitem de consultas de rotina.
- Pacientes com historial significativo do ponto de vista cardíaco.
- Reabilitação de Pacientes que tenham tido recentemente alta hospitalar e se encontrem numa fase de recuperação.

Para os 3 cenários estabelece-se que o paciente se encontra no seu domicílio com um conjunto de sensores que monitorizam a sua actividade cardíaca.

No âmbito deste projecto, seleccionou-se a placa de aquisição de sinais vitais *BITalino* por possuir um sensor de *ECG*, ser modular, permitindo assim acrescentar outro tipo de sensores ao kit original e possuir uma boa relação em termos de qualidade/preço.

A informação que é recolhida pelos sensores é transmitida à aplicação móvel através da tecnologia *Bluetooth*, que é uma tecnologia de comunicação sem fios de curto alcance.

A aplicação móvel efectua um pre-processamento, através de filtragem e amostragem e efectua uma análise do sinal tendo em consideração alguns parametros para determinar se existe algum tipo de anomalia no *ECG*.

A informação contida na aplicação móvel é enviada para um servidor central onde é registada sob a forma de uma base de dados clínica/hospitalar.

Antes de a informação ser registada, é realizada uma triagem para avaliar se a situação do paciente é uma situação não-crítica, crítica ou urgente, procedendo-se à normalização deste tipo de informação.

O sistema a ser desenvolvido recorrerá a algumas tecnologias sem fios (*wireless*) já existentes.

A utilização das tecnologias no contexto da saúde implica o cumprimento de elevados padrões de segurança para que seja possível a transmissão de informação dos vários utentes sem que esta esteja sujeita a possíveis ataques informáticos.

Para a implementação do projecto recorreu-se à ferramenta *MATLAB* de modo a que seja possível desenvolver um algoritmo que defina as etapas de amostragem, filtragem e compressão.

Para tal, serão utilizados dois *Datasets* do *PhysioNet*: um orientado para a filtragem do ruído presente no sinal original (*Dataset* proveniente da base de dados *ECG-ID Database* do *PhysioNet* [35]) e outro orientado para a detecção de arritmias (*Dataset* proveniente da base de dados *MIT-BIH Arrhythmia Database*).

Alguns algoritmos convencionais como o *Turning Point*, *CORTES*, *AZTEC*, *Pan-Tompkins* e *Okada* serão sujeitos a análise e teste para verificar o seu comportamento em dispositivos móveis.

1.5 RESULTADOS OBTIDOS

Inicialmente foi utilizada a plataforma de testes *MATLAB* onde foram testados vários tipos de filtros tais como um filtro passa-alto de Butterworth com frequência de corte aos 0.5 Hz e um filtro de Chebyshev tipo I e um filtro de Chebyshev tipo II, ambos com frequência de corte aos 0.5 Hz e um *ripple* máximo de 3 dB. Também se experimentou reduzir algumas interferências de sinal com um filtro de Savitzky-Golay, no entanto, este filtro induz alguma distorção no sinal. Por fim, recorreu-se a alguns filtros do Algoritmo Pan-Tompkins e concluiu-se que estes apresentam um melhor resultado à saída, atenuando a Flutuação de Linha de Base e a Interferência do sinal de *EMG*. Para remoção da interferência de rede eléctrica de 50 Hz utilizou-se um filtro *notch* também designado de rejeita-banda. Neste processo de desenvolvimento de uma estratégia de filtragem, utilizou-se o *DataSet ECG-ID Database* do *PhysioNet*.

Numa segunda fase migrou-se para a plataforma *Android*. A aplicação desenvolvida recebe os dados do canal de *ECG* do dispositivo de aquisição seleccionado (*BITalino* - versão *plugged*). A comunicação com esse dispositivo é efectuada através do protocolo de transmissão de dados a curta distância Bluetooth, através das classes de código desenvolvidas para a parte de Comunicações. Com a Biblioteca *achartengine* foram gerados gráficos na aplicação *Android*. A aplicação *Android* da Figura 61 representa em termos gráficos o sinal de *ECG*. Este sinal de *ECG* corresponde ao sinal original sem nenhum estágio de filtragem a nível digital, sendo que o *BITalino* implementa algum tipo de filtragem por *hardware*. Utilizaram-se alguns indivíduos tendo sido colocados os eléctrodos na zona palmar das mãos para poder gerar os gráficos na aplicação móvel a partir dos dados dos sinais de *ECG* de cada um desses indivíduos. Foram desenvolvidas classes de código para subamostrar o sinal original e para realizar filtragem digital do sinal. Também se procurou através de determinadas técnicas de *Thresholding* detectar algumas arritmias no sinal de *ECG*.

1.6 ESTRUTURA DO DOCUMENTO

Este documento é composto por 7 capítulos.

O presente capítulo diz respeito à Introdução que é constituída pela definição do problema, pela motivação, pelos objectivos, pela metodologia e pela estrutura do documento.

O Capítulo 2 refere-se ao Estudo dos Sinais Vitais onde é efectuada uma breve caracterização do sistema de condução cardíaco, características e morfologia do sinal de *ECG*, e patologias associadas.

O Capítulo 3 aborda algumas técnicas de amostragem, filtragem e compressão geralmente utilizadas no âmbito dos sinais biomédicos.

No Capítulo 4 são descritos alguns algoritmos convencionais para monitorização do sinal de electrocardiograma.

No Capítulo 5 é realizada uma descrição dos vários projectos relacionados e é efectuada uma análise comparativa dos mesmos.

No Capítulo 6 designado de Desenvolvimento do Projecto estabelecem-se os requisitos do Sistema, é definida a arquitectura do sistema e descrevem-se os passos da implementação do projecto.

No Capítulo 7 são apresentadas as conclusões do projecto juntamente com algumas sugestões de trabalhos futuros.

ESTUDO DOS SINAIS VITAIS

2.1 SISTEMA CARDIOVASCULAR

O coração pode ser considerado uma bomba electromecânica, ou seja, um sistema que gera de forma automática um impulso e o transmite às células cardíacas levando à sua contracção e relaxamento e permitindo que o sangue se distribua por todo o organismo.

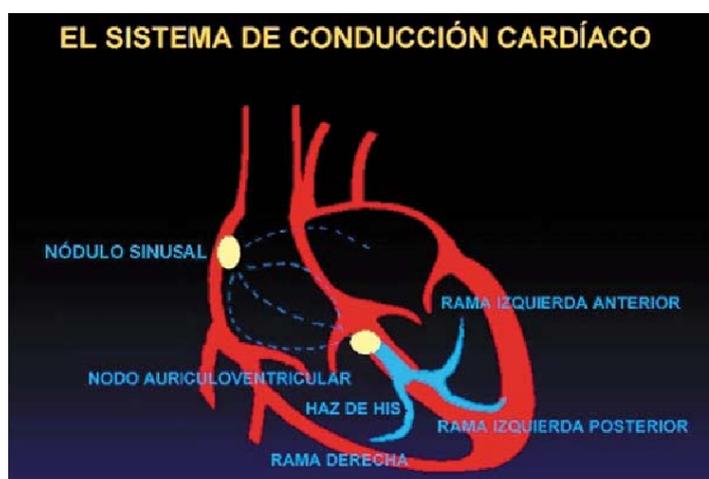


Figura 1: Representação esquemática do sistema de condução cardíaco [1].

Na Figura 1 observa-se uma representação cardíaca com as suas câmaras, aurículas e ventrículos e as válvulas que as separam e as comunicam com a artéria aorta e com a artéria pulmonar.

O nó sinusal, que se situa na parte posterior e superior da aurícula direita muito próximo do local de junção da veia cava superior, designa-se de marcapasso cardíaco uma vez que a despolarização das suas células ocorre mais rapidamente que as outras células cardíacas.

Este impulso eléctrico formado no nó sinusal distribui-se pelas aurículas, levando à sua contracção.

O impulso alcança o nó aurículo-ventricular situado sob a inserção da válvula septal e da válvula tricúspide e de seguida chega a uma pequena estrutura denominada de feixe de His. Este tipo de estrutura divide-se nos ramos direito e esquerdo que se subdividem, por sua vez, numa rede de feixes encarregados de transmitir este impulso às células musculares dos ventrículos (a rede de Purkinje) [1].

2.2 CARACTERÍSTICAS E MORFOLOGIA DO SINAL DE *ecg*

O *ECG* é um sinal bioeléctrico que regista a actividade eléctrica do coração em função do tempo, sendo uma ferramenta básica de diagnóstico crucial na avaliação das funções do coração e das patologias que lhe estão associadas.

De modo a registar uma onda de *ECG*, é efectuado um registo diferencial entre 2 pontos da superfície corporal de um determinado indivíduo, sendo cada um desses registos designado de derivação.

Einthoven definiu inicialmente um conjunto de 3 derivações:

$$I = VLA - VRA \quad (1)$$

$$II = VLL - VRA \quad (2)$$

$$III = VLL - VLA \quad (3)$$

onde *VRA* = *Voltage Right Arm*, *VLA* = *Voltage Left Arm*, e *VLL* = *Voltage Left Leg*.

Estas três derivações encontram-se representadas na Figura 2.

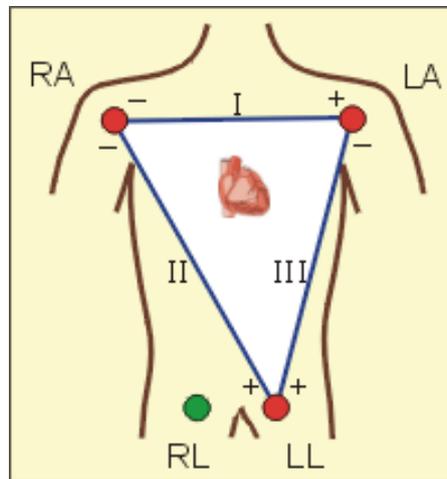


Figura 2: Configuração geral das 3 derivações de Einthoven (Triângulo de Einthoven) [2].

A derivação I corresponde à colocação do eléctrodo positivo no braço esquerdo e do eléctrodo negativo no braço direito (Equação 1). A derivação II corresponde à colocação do eléctrodo positivo na perna esquerda e do eléctrodo negativo no braço direito (Equação

2). A derivação III corresponde à colocação do eléctrodo positivo na perna esquerda e do eléctrodo negativo no braço esquerdo (Equação 3).

Para o conjunto de 3 derivações acima descritas verifica-se a relação $II = I + III$, pelo que esta derivação contém uma fracção importante da informação presente no sinal de ECG e é muito utilizada a nível clínico.

Geralmente, nas unidades de saúde a actividade eléctrica do coração é registada a partir de um ECG com 12 derivações devido ao facto de esta se encontrar melhor caracterizada por um maior conjunto de derivações e devido ao facto de a qualidade do sinal ser superior neste caso. Estas 12 derivações são constituídas pelas 3 derivações de Einthoven (derivações I, II e III), pelas 6 derivações de Wilson (V1 a V6) e pelas 3 derivações aumentadas de Wilson, aVR , aVL e aVF (Figura 3.)

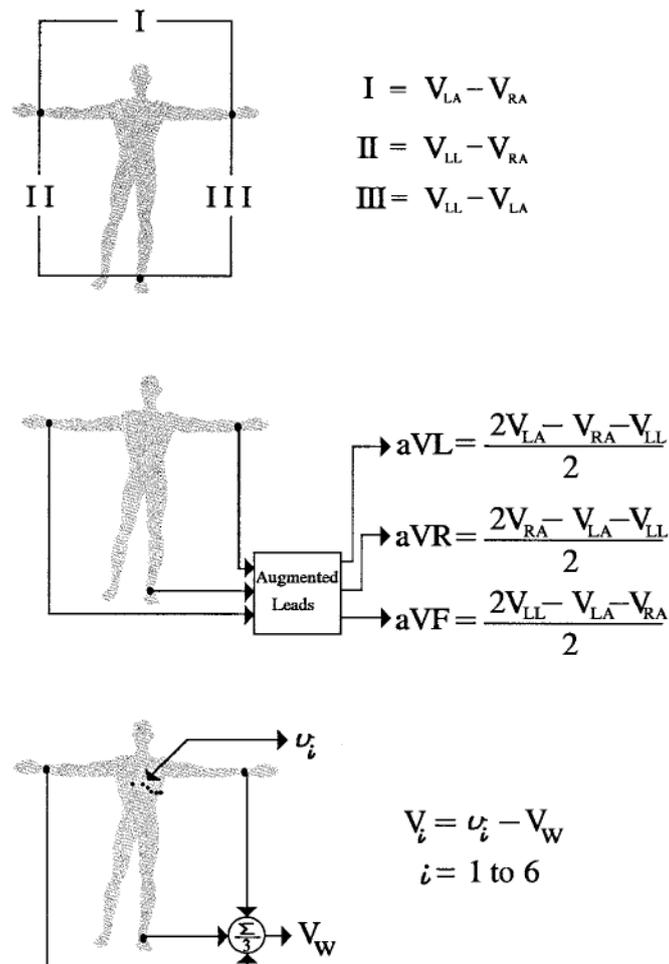


Figura 3: ECG de 12 derivações [3].

As 6 derivações de Wilson definidas por V₁, V₂, V₃, V₄, V₅ e V₆ dispõem-se no paciente de acordo com o esquema da Figura 4. A derivação V₁ é colocada no quarto espaço intercostal (entre as costelas 4 e 5) do lado direito do esterno. A derivação V₂ à semelhança de V₁ é colocada no quarto espaço intercostal (entre as costelas 4 e 5) mas do lado esquerdo do esterno. A derivação V₃ é colocada entre as derivações V₂ e V₄. A derivação V₄ é colocada no quinto espaço intercostal (entre as costelas 5 e 6) na linha médio-clavicular. A derivação V₅ é colocada na mesma direcção de V₄, na linha axilar anterior esquerda. A derivação V₆ é colocada na mesma direcção de V₄ e V₅ na linha médio-axilar.

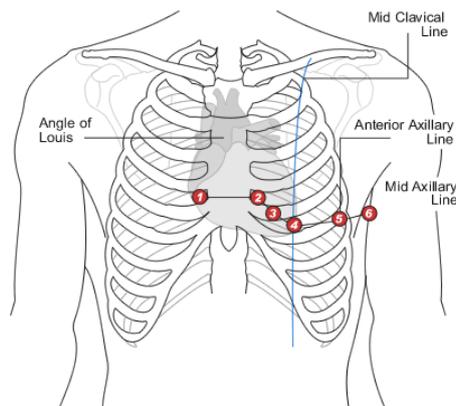


Figura 4: Disposição das 6 derivações de Wilson.

Na Figura 5 encontra-se representado um ECG de 12 derivações. Este tipo de ECG não costuma ser utilizado para monitorização contínua.

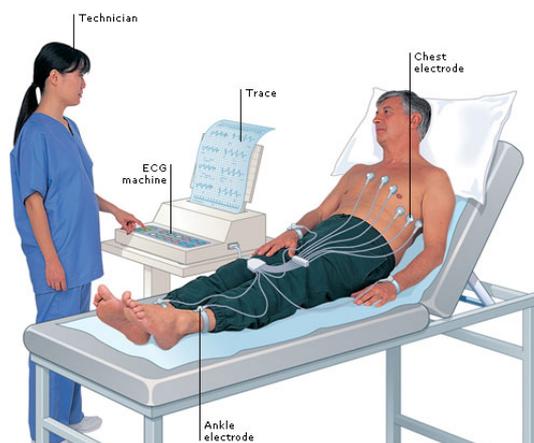


Figura 5: Utilização de um ECG de 12 derivações numa unidade de saúde [3].

Para monitorização em tempo real de um paciente costuma utilizar-se sistemas com 3 ou 5 derivações, sendo mais frequente o uso do ECG de 3 derivações.

No ECG de 3 derivações, de acordo com a informação constante em [4], a disposição dos eléctrodos pode ser uma das 3 que se seguem (Figura 6, 7 ou 8):



Figura 6: Disposição dos eléctrodos sobre o coração [4].

Os eléctrodos de ECG podem ser colocados sobre o coração numa configuração triangular com o eléctrodo positivo do lado esquerdo, o negativo do lado direito e o neutro mais acima e ao centro tal como representado na Figura 6.



Figura 7: Disposição dos eléctrodos segundo Einthoven [4].

Também podem ser colocados segundo a configuração de Einthoven (eléctrodo positivo no braço direito, eléctrodo negativo no braço esquerdo e neutro na perna esquerda) 7.



Figura 8: Disposição dos eléctrodos nos dedos [4].

Alternativamente podem ser colocados nos dedos de acordo com a disposição em 8 ou na zona palmar das mãos (eléctrodo positivo na parte direita da palma da mão direita, neutro na parte esquerda da palma da mesma mão e eléctrodo negativo na parte direita da palma da mão esquerda).

O sinal de *ECG* é constituído por uma onda *P* seguida de um complexo *QRS* e das ondas *T* e *U*, como apresentado na Figura 9.

A presença da onda *P* no sinal de *ECG* deve-se à activação das aurículas por um impulso cardíaco com origem no nodo auriculoventricular (*AV*). O complexo *QRS* relaciona-se com a excitação a nível dos ventrículos. A onda *T* corresponde a uma repolarização do sinal. Relativamente à onda *U*, não se verifica em todos os indivíduos e pensa-se que esta seja um potencial de repolarização.

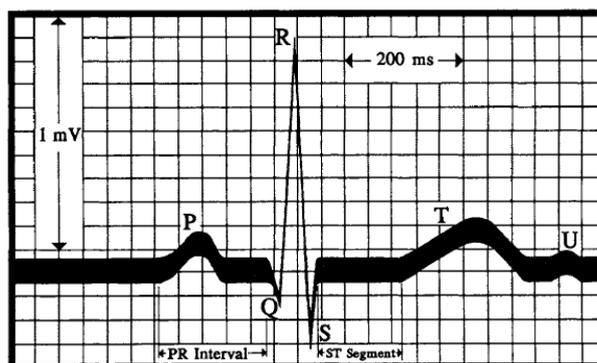


Figura 9: Registo da derivação II apresentando a onda *P*, o complexo *QRS* e as ondas *T* e *U* [5].

Os sinais de ECG encontram-se geralmente no intervalo de ± 2 mV e requerem uma largura de banda de 0.05 a 150 Hz.

A Figura 10 apresenta o ECG de um batimento cardíaco. Encontram-se também presentes na Figura os tipos de medições que podem ser efectuadas em cada uma das ondas constituintes do ECG, para classificação de cada tipo de batimento e ritmo cardíaco.

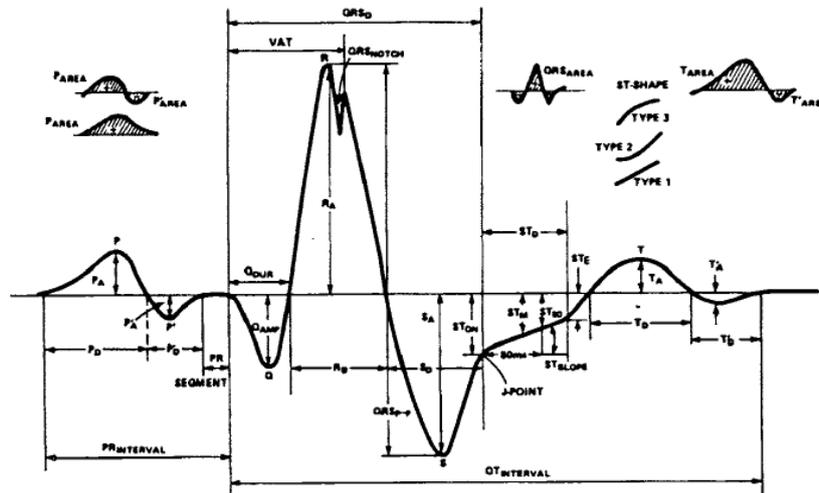


Figura 10: Medição dos vários parâmetros de um sinal de ECG.
(Fonte: Hewlett Packard Company, Palo Alto, Calif.)

O Electrocardiograma (ECG) de um indivíduo encontra-se geralmente contaminado pelo mais variado tipo de interferências (interferências da rede eléctrica, flutuação de Linha de Base e ruído originado pela contracção muscular).

Estas têm de ser removidas ou atenuadas de modo a que seja possível efectuar a sua interpretação clínica ou a detecção automática de algumas arritmias através da utilização de *software*.

2.3 MONITOR DE HOLTER

O monitor de Holter (Figura 11) é um dispositivo de diagnóstico que permite a monitorização e o armazenamento dos dados do *ECG* e do batimento cardíaco de um determinado paciente durante um período de 24 h.

Após esse período de 24 h, este é devolvido a um profissional de saúde para análise e interpretação dos dados obtidos.

Este sistema não é por isso um sistema de monitorização em tempo real, fazendo a análise dos dados *off-line*.

Caso haja a necessidade de alguns pacientes serem monitorizados continuamente devido a problemas cardiovasculares é fundamental a existência de uma plataforma que autonomize este processo e que permita a sua monitoração à distância.

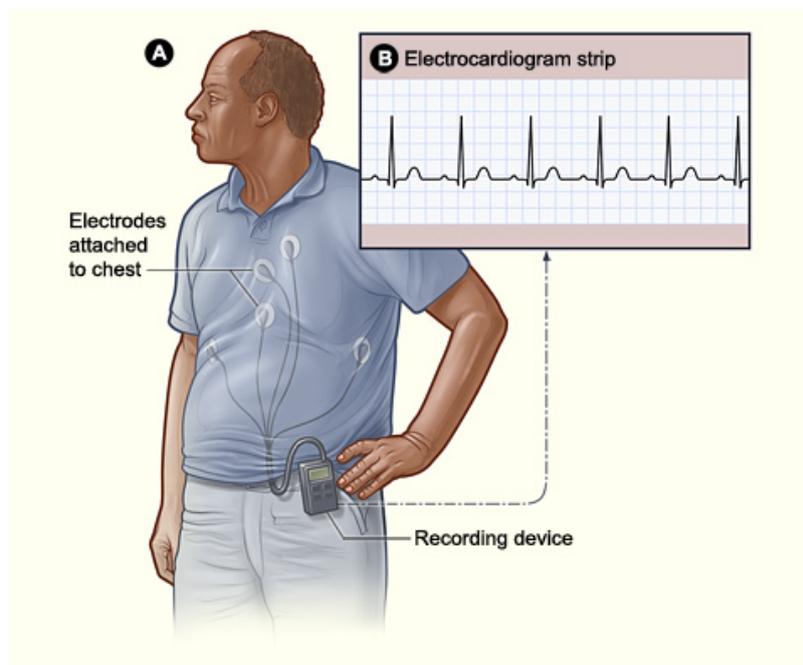


Figura 11: Monitor de Holter [6].

2.4 RITMO SINUSAL NORMAL

O Ritmo Sinusal Normal (Figura 12) caracteriza-se por uma frequência de 60 a 100 *bpm* em fase de repouso. O sinal eléctrico do coração tem origem num impulso eléctrico gerado no nó sino-auricular (SA) também designado de nó sinusal. Por este motivo, o ritmo normal assume também a designação de sinusal.

As 3 deflexões, *P*, *QRS* e *T*, características de um sinal de *ECG*, apresentam-se de modo sequencial normal e a duração dos intervalos de cada um dos segmentos do sinal de *ECG* encontra-se bem definida.



Figura 12: Ritmo Sinusal Normal [7].

Apresenta-se na tabela 1 os valores padrão da duração e amplitude de cada uma das ondas e segmentos presentes no sinal de *ECG*:

Tabela 1: Parâmetros do sinal de *ECG* normal. (Adaptado de [30], [31] e [32]).

Parâmetros	Valores Normais
Ritmo Cardíaco	60-100 <i>bpm</i>
Onda <i>P</i>	Amplitude: 0.20-0.30 mV Duração: 0.09-0.13 s
Complexo <i>QRS</i>	Amplitude: 1.10-2.10 mV Duração: 0.08-0.12 s
Onda <i>R</i>	Amplitude: 1.10-2.10 mV
Onda <i>Q</i>	Amplitude: 0.25 x onda <i>R</i> Duração: < 0.04 s
Onda <i>T</i>	Amplitude: -0.5 mV Duração: 0.16 s
Intervalo <i>PQ</i> ou <i>PR</i>	Duração: 0.12-0.20 s
Intervalo <i>QT</i>	Duração: 0.35-0.42 s
Intervalo <i>ST</i>	Duração: 0.27-0.33 s

2.5 ARRITMIAS CARDÍACAS

Podemos definir as arritmias cardíacas como todo o ritmo diferente do ritmo sinusal normal.

Este é o ritmo gerado no nodo sinusal com uma frequência entre 60 e 90 batimentos por minuto.

Define-se automaticidade como a propriedade de algumas células cardíacas para gerar de forma espontânea estímulos capazes de se transmitir às células vizinhas.

Esta capacidade depende da existência de uma corrente de despolarização diastólica espontânea, que alcança automaticamente o potencial limiar e gera um potencial de acção.

No coração humano existem diversas estruturas cardíacas automáticas, mas em condições normais a corrente diastólica do nodo sinusal é a mais rápida, alcança o potencial limiar e gera um potencial de acção que se transmite ao resto das células cardíacas, originando o ritmo sinusal normal.

A alteração do automatismo das células cardíacas pode explicar a génese de múltiplas arritmias cardíacas.

Estas podem originar-se quando a corrente de despolarização diastólica das células automáticas não sinusais ultrapassa a corrente sinusal, determinando a ocorrência uma arritmia rápida.

Por outro lado, caso se verifique a depressão do automatismo sinusal, isto determina que o ritmo cardíaco seja gerado por outra estrutura automática. Este tipo de estrutura designa-se de nodo auriculo-ventricular e é responsável por um ritmo cardíaco mais lento que o ritmo sinusal.

Define-se a excitabilidade como a capacidade de as células cardíacas para responder a um estímulo efectivo, procedente de células automáticas. Estas originam um potencial de acção que se verifica após um período refractário absoluto a qualquer tipo de estímulo aplicado. A esse período refractário segue-se um período relativo durante o qual os estímulos de grande magnitude podem gerar potenciais de acção capazes de se propagar às células vizinhas.

Teríamos então uma fase de recuperação da excitabilidade, em que estímulos de magnitude limiar geram um novo potencial de acção.

Nesta fase existe um curto período, denominado de supernormal, em que se poderia gerar um potencial de acção com estímulos inferiores ao limiar.

Existem situações patológicas nas quais se podem gerar pós-potenciais responsáveis por um potencial de acção prematuro que, ao propagar-se por todo o coração, determina o aparecimento de uma arritmia.

2.5.1 *Bradicardia Sinusal*

A bradicardia sinusal (Figura 13) caracteriza-se pelo batimento lento do coração.

Esta patologia deve-se essencialmente a um ritmo cardíaco inferior a 60 *bpm*.

O seu traçado de *ECG* apresenta, à excepção da frequência cardíaca, as mesmas características do ritmo cardíaco normal.

As condições verificadas neste caso são:

- Presença do complexo *QRS*.
- Duração do intervalo *PR* entre 0.12 s e 0.22 s.
- Duração da onda *P* entre 0.06 s e 0.1 s.
- Frequência cardíaca inferior a 60 *bpm*.
- Intervalo entre 2 ondas *P* consecutivas inferior a 1.1 vezes a média dos intervalos entre ondas *P*.



Figura 13: Bradicardia Sinusal [8].

2.5.2 *Taquicardia Sinusal*

A Taquicardia Sinusal (Figura 14) é uma anomalia cardíaca que se caracteriza pelo batimento rápido do coração, o que corresponde a uma frequência cardíaca superior a 100 *bpm*.

A detecção deste tipo de anomalia depende da verificação das seguintes condições:

- Existência de complexo *QRS*.
- Intervalo *PR* compreendido entre 0.12 s e 0.22 s.
- Duração da onda *P* entre 0.06 s e 0.1 s.
- Frequência cardíaca superior a 100 *bpm*.
- Intervalo entre 2 ondas *P* consecutivas menor que 1.1 vezes a média dos intervalos entre as ondas *P*.

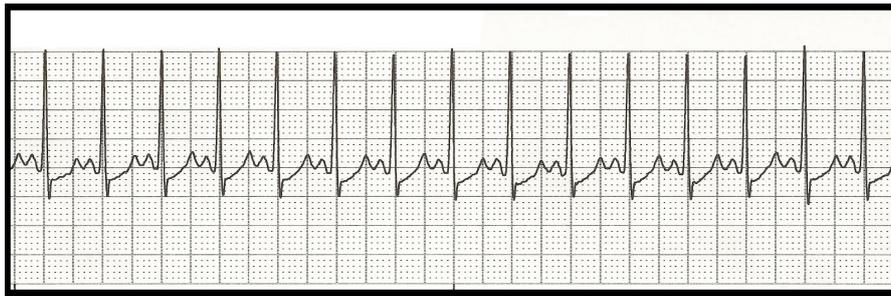


Figura 14: Taquicardia Sinusal [9].

2.5.3 Fibrilhação Auricular (AF)

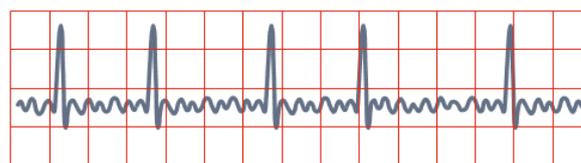
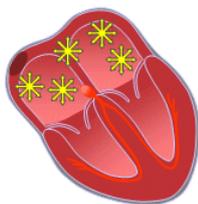
Na fibrilhação auricular (Figura 15), as aurículas contraem-se aproximadamente 400 a 500 vezes por minuto.

Uma vez que a maioria dessas contrações auriculares é fraca e incompleta, as ondas *P* não são registadas no electrocardiograma (ECG).

As contrações auriculares são quase sempre bloqueadas no nó AV, o que resulta numa resposta ventricular irregular.

Para detecção deste tipo de condição cardíaca, os seguintes pressupostos têm de se verificar:

- Duração do complexo QRS inferior a 0.06 s ou superior a 0.1 s
- Ausência de onda *P*
- Presença de ritmo variável ao longo de 10 s



Baseline irregular, ventricular response irregular

Figura 15: Fibrilhação Auricular [8].

2.5.4 Arritmia Sinusal

A arritmia sinusal (Figura 16) deve-se a bloqueios na condução aurículo-ventricular com origem no nó sinoatrial (SA).

As condições necessárias à confirmação da presença desta arritmia no sinal de *ECG* são as seguintes:

- Presença do complexo *QRS*.
- Tempo de duração do segmento *PR* entre 0.12 s e 0.22 s.
- Duração da onda *P* entre 0.06 s e 0.1 s.
- Frequência cardíaca inferior a 60 *bpm*.
- Intervalo entre 2 ondas *P* consecutivas superior a 0.9 vezes a média dos intervalos entre ondas *P*.

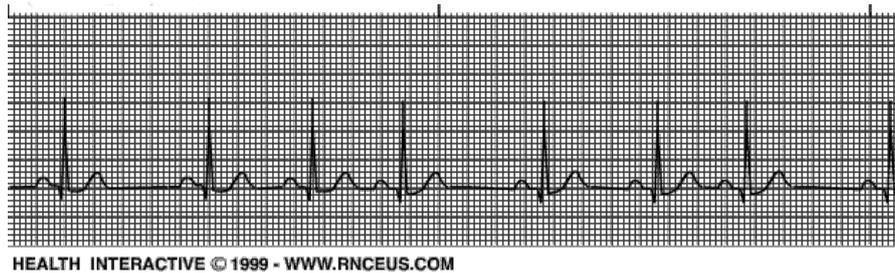


Figura 16: Arritmia Sinusal [10].

2.5.5 Bloqueio de Ramo

O Bloqueio de ramo é caracterizado por um atraso na activação de um dos ventrículos (direito ou esquerdo) originando o bloqueio do ramo associado a esse ventrículo.

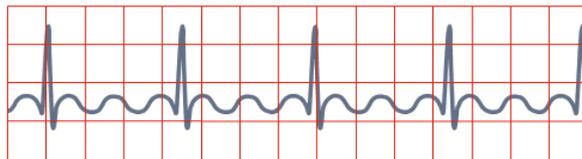
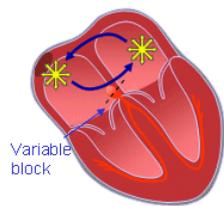
Esta patologia cardíaca traduz-se num complexo *QRS* com uma duração superior ou igual a 0.12 s e numa onda *T* com polarização normal.

2.5.6 Flutter Auricular (AFL)

O *Flutter* Auricular (Figura 17) caracteriza-se por um conjunto de ondas regulares em “dente de serra” (ondas *F*) que substituem a linha isoelectrica que existe geralmente entre o final da onda *T* e o início da onda *P*, deixando de ser possível distinguir a onda *P* no sinal de *ECG*.

Esta patologia quando associada a outras patologias do foro cardíaco como a síndrome de Wolff-Parkinson-White (*WPW*) pode originar a morte súbita.

A frequência auricular desta patologia possui um valor aproximado definido entre 220 *bpm* e 400 *bpm*.



Rapid flutter waves, ventricular response irregular

Figura 17: Flutter Auricular [8].

TÉCNICAS DE AMOSTRAGEM, FILTRAGEM E COMPRESSÃO

3.1 AMOSTRAGEM

O teorema de *Shannon-Nyquist* estabelece que a frequência de amostragem deve ser duas vezes superior à componente de maior frequência de um sinal, sendo este valor apenas um mínimo teórico, uma vez que este teorema só é válido para um intervalo de amostragem infinito.

A nível europeu, foi definido em 1990 pelo *Common Standards for quantitative Electrocardiography (CSE)*, que a frequência mais alta do sinal de ECG é de 150 Hz.

A *American Heart Association (AHA)* recomenda frequências de amostragem 2 ou 3 vezes superiores a esse valor, existindo vários estudos que sugerem que deve ser utilizada uma frequência de pelo menos 500 Hz para amostrar correctamente o sinal [36].

3.2 SUBAMOSTRAGEM

Amostragem Adaptativa

Esta técnica de amostragem, permite reduzir o consumo de energia a nível da transmissão de dados através da selecção inteligente das frequências de amostragem a utilizar com um determinado sinal.

O sinal original é dividido em dois segmentos, um de baixa frequência (ondas *P* e *T*) e um de alta frequência (complexo *QRS*). O número de amostras e consequentemente a frequência de amostragem aplicados são superiores no caso do complexo *QRS*. No caso das componentes de baixa frequência, é possível utilizar frequências de amostragem inferiores. Isto advém do facto de estas últimas componentes possuírem diversas amostras com informação redundante. Esta informação redundante revela-se desnecessária para a reconstrução do sinal. Pode ser eliminada recorrendo a uma frequência de amostragem inferior.

O sinal original é comparado com um determinado *threshold* de tensão. Se o sinal se encontrar acima dele é aplicada uma frequência de amostragem superior, ou inferior caso contrário.

O ajuste da frequência de amostragem também pode ser efectuado por recurso a um dicionário de códigos de um método de compressão já existente que procede a uma análise das redundâncias do sinal [37].

3.3 FILTRAGEM

Filtragem [11] é um processo de selecção, ou supressão de determinadas componentes em frequência de um sinal.

Esta pode ser efectuada recorrendo a filtros analógicos e/ ou filtros digitais.

No caso da plataforma de aquisição de biosinais seleccionada (*BITalino*), o sensor de *ECG* dispõe de um conjunto de filtros que realizam a filtragem por hardware no dispositivo.

O sensor de *ECG* presente no *BITalino* possui um filtro passa-banda entre 0.5 Hz e 40 Hz assim como um amplificador com um valor de *CMRR* (*Common Mode Rejection Ratio*) de 110 dB.

Este integra um amplificador de instrumentação com um ganho de 20 dB e alguns amplificadores operacionais para realizar a filtragem passa-banda e a amplificação do sinal filtrado.

A filtragem passa-banda é implementada recorrendo a um filtro passa-baixo *Butterworth* de 4ª ordem com frequência de corte aos 40 Hz e um ganho de 100 dB e um amplificador de instrumentação com *coupling AC* para rejeitar as tensões de entrada *DC* com frequência de corte aos 0.5 Hz.

O circuito de filtragem analógico incluído no sensor de *ECG* apresenta deste modo um valor de rejeição de modo comum (*CMRR*) de 110 dB para um *Ganho* ≥ 10 .

Os filtros digitais também podem atenuar as componentes em frequência de um sinal conforme se pretenda.

A banda de transição 18 que corresponde a uma gama de frequências que permite a transição da banda de passagem para a banda de rejeição (o ganho do filtro passa de um valor de 1 dB para um valor muito próximo de 0) deveria ser nula no caso ideal, passando-se quase que instantaneamente da banda de passagem para a banda de rejeição, sendo possível uma resposta mais rápida por parte do filtro.

No entanto, não é possível desenvolver filtros com este tipo de resposta.

Os filtros digitais podem dividir-se em filtros de Resposta a Impulso Finita (*FIR*) e em filtros de Resposta a Impulso Infinita (*IIR*).

Os filtros *FIR* caracterizam-se por serem filtros não recursivos, possuem um número de impulsos finito e são definidos pela Equação 4:

$$y(n) = \sum_{k=0}^N b(k)f(n-k) \quad (4)$$

Os filtros *IIR* são filtros recursivos, possuem um número infinito de impulsos e são definidos pela Equação 5:

$$y(n) = \sum_{k=0}^N b(k)f(n-k) + \sum_{k=1}^M a(k)y(n-k) \quad (5)$$

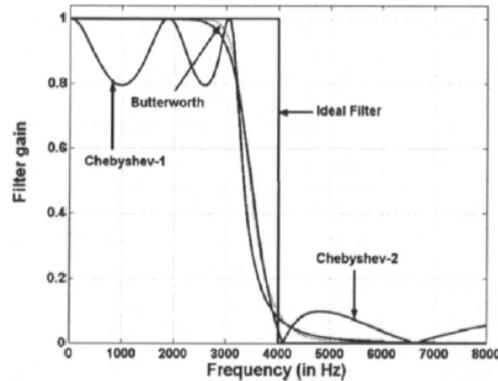


Figura 18: Características dos filtros ideal, de *Butterworth*, *Chebyshev* tipo I e *Chebyshev* tipo II [11].

O Electrocardiograma (*ECG*) de um indivíduo encontra-se geralmente contaminado pelo mais variado tipo de interferências. Estas têm de ser removidas ou atenuadas de modo a que seja possível efectuar a sua interpretação clínica ou a detecção automática de algumas arritmias através da utilização de *software*.

As principais interferências presentes neste tipo de sinal são a interferência da rede eléctrica, a flutuação de linha de base e o ruído causado pelas contracções a nível muscular.

3.3.1 Flutuação da linha de base

Para remoção da flutuação da linha de base, Hamilton[38] recorre a um filtro passa-alto de acordo com a Equação 6 que é a uma equação de diferenças :

$$y(nT) = y(nT - T) - x(nT)/32 + x(nT - 16T) - x(nT - 17T) + x(nT - 32T)/32 \quad (6)$$

A flutuação de linha de base consiste de componentes de baixa frequência originadas pela transpiração, respiração ou movimentos corporais. Estas componentes sobrepõem-se ao segmento *ST* do sinal de *ECG* no espectro das frequências, o que dificulta a correcta análise e interpretação dos dados originais.

Deste modo, torna-se necessário proceder à remoção deste tipo de interferências.

Para se eliminar este tipo de interferência recorre-se às técnicas de filtragem linear e de ajuste polinomial.

Atendendo a que a componente de mais baixa frequência do *ECG* (definida pelo ritmo cardíaco mais lento, neste caso *40 bpm*) é de aproximadamente 0.67 Hz, deve-se definir um filtro passa-alto com uma frequência de corte mais baixa (cerca de 0.5 Hz) para que essa componente apareça no espectro de frequências com um baixo grau de atenuação, ao mesmo tempo que se remove as frequências em torno dos 0 Hz.

Esta frequência de corte deve ser seleccionada de modo a que a informação do sinal de *ECG* permaneça sem distorções ao mesmo tempo que é removida ao máximo a flutuação da linha de base.

De modo a evitar o desfasamento entre o sinal actual e o sinal original, que pode induzir várias alterações a nível das relações temporais no ciclo cardíaco, o filtro a ser aplicado deve ser um filtro de fase linear como é o caso de um filtro de resposta a impulso finita (*FIR*).

Os filtros *FIR* quando comparados com os filtros *IIR*, apresentam uma ordem de complexidade muito superior para o mesmo nível de desempenho. Deste modo, o atraso provocado por este tipo de filtros é muito superior ao dos filtros *IIR*, dado que estes últimos possuem uma resposta aproximadamente linear.

Em alternativa, pode-se realizar uma primeira decimação do sinal seguida de uma interpolação. O processo de decimação remove o conteúdo de alta frequência, podendo utilizar-se um filtro passa-baixo para produzir na saída um valor aproximado da flutuação da linha de base. Sobre este valor é realizada uma interpolação à frequência de amostragem original e que é posteriormente subtraído ao sinal original.

Na técnica de ajuste polinomial, os pontos de interesse que se situam sobre a curva do sinal de *ECG* são aproximados por funções polinomiais de modo a identificar os complexos QRS e o segmento PQ.

Através da utilização de polinómios de terceiro grau da série de *Taylor* é possível identificar os complexos QRS através da aproximação por *splines* cúbicas (polinómios do 3º grau) de conjuntos sucessivos de três nós.

3.3.2 Interferência da Rede Eléctrica

Hamilton[39] sugere a utilização de um de 2 fitros para remoção da interferência de rede eléctrica aos 50 Hz ou 60 Hz: Um filtro de *notch* adaptativo ou um filtro de *notch* não adaptativo.

Os filtros de *notch* são um caso particular dos filtros rejeita-banda.

Os filtros adaptativos são uma classe de filtros especial que vai actualizando o valor do filtro à medida que este prossegue.

Esta actualização é realizada mediante a diferença entre a entrada e a saída e através da função erro, sendo que dos coeficientes anteriores se obtêm os novos coeficientes do filtro.

Os coeficientes do filtro são geralmente recalculados para alterar a frequência *notch* de um filtro *notch* de acordo com a frequência extraída do sinal de interferência de rede eléctrica.

Com um filtro notch estático, não adaptativo, seria necessário rejeitar uma maior gama de frequências para acomodar a variabilidade da frequência deste sinal de interferência.

O filtro adaptativo segue a frequência da interferência pelo que a banda de rejeição pode ser muito mais estreita, retendo por conseguinte mais informação do sinal de ECG.

O filtro de *notch* não adaptativo para uma frequência de interferência de rede eléctrica de 50 Hz é definido pela função de transferência apresentada na Equação 7. Na Equação 7, T é o período de Amostragem.

$$H(z) = \frac{1 - 2\cos(2\pi \cdot 50 \cdot T)z^{-1} + z^{-2}}{1 - 2r\cos(2\pi \cdot 50 \cdot T)z^{-1} + r^2z^{-2}} \quad (7)$$

Este filtro possui um zero no círculo unitário aos 60 Hz e um pólo para o mesmo ângulo com um raio r .

r assume um valor entre 0 e 1, sendo que a largura de banda do filtro *notch* diminui à medida que esta constante se aproxima do valor unitário.

A função de transferência acima referida traduz-se na equação 8. Trata-se de uma equação de diferenças em que $N = 2\cos(2\pi 50T)$.

$$y(t) = rN \cdot y(t - nT) - r^2 y(t - 2nT) + x(t) - N \cdot x(t - nT) + x(t - 2nT) \quad (8)$$

Os filtros notch são geralmente utilizados para supressão da frequência de interferência de rede eléctrica (50Hz ou 60Hz), no entanto induzem distorção no sinal uma vez que removem essa frequência do sinal original [40].

Para que essa distorção seja aceitável a banda de frequências do filtro *notch* tem de ser bastante estreita, o que determina que a filtragem seja ineficaz nos casos em que ocorre um maior desvio em frequência da interferência de rede eléctrica [40].

A rede eléctrica pode induzir uma interferência sinusoidal de 50 Hz ou 60 Hz no sinal original, possivelmente acompanhada de alguns

dos seus harmónicos, sobrepondo-se às ondas de baixa amplitude do sinal.

Para remover esta interferência pode-se aplicar um filtro *notch* ao sinal de entrada.

Este filtro é definido por um par de zeros complexos-conjugados que se situam no interior do círculo unitário a uma determinada frequência ω_0 , sendo as frequências mais próximas desta frequência de interferência atenuadas.

Este filtro pode ser melhorado através da colocação de um par de pólos sobre uma linha que atravesse um dos raios da circunferência (a distância dos pólos ao centro da circunferência determina a largura de banda do filtro *notch*).

Uma vez que não é possível desenvolver um filtro linear e invariante no tempo para remoção do ruído sem originar ecos, pelo facto de a um aumento da resolução em frequência corresponder um decréscimo da resolução no tempo pode-se alternativamente desenvolver um filtro não linear que gera uma curva sinusoidal aproximada, $v(n)$, do sinal de interferência eléctrica original, $x(n)$, para posterior remoção.

A sinusóide $v(n) = \sin(\omega_0 n)$ ajusta-se ao sinal de interferência da rede eléctrica através da utilização da função de erro, $e(n) = x(n) - v(n)$ que depende da componente contínua (DC) do sinal, sendo que a componente DC da função de erro pode ser removida por via da sua primeira diferença.

O valor de $v(n)$, é actualizado por um valor, α , positivo ou negativo, de acordo com o sinal da primeira diferença.

O sinal de saída obtém-se por subtracção do valor total aproximado da interferência ao sinal de entrada.

3.3.3 Ruído de origem muscular

O ruído de origem muscular consiste de componentes de baixa frequência que se sobrepõem ao complexo *PQRST* no domínio das frequências.

Para atenuação e supressão do ruído de *EMG*, Ahlstrom et al.[12] sugerem a utilização de um filtro passa-baixo de *Lynn* [41] de 2ª ordem.

Este filtro tem como base a função de transferência definida na Equação 9 com m igual ao número de zeros múltiplos da frequência da interferência.

$$H(z) = \frac{(1 - z^{-m})^2}{(1 - z^{-1})^2} \quad (9)$$

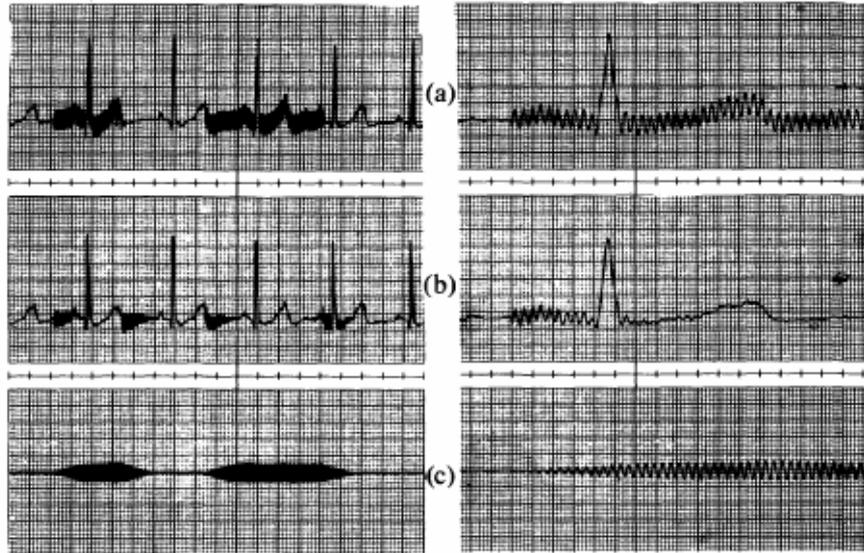


Figura 19: Exemplo de um Filtro Notch Adaptativo [12].

O que resulta na equação de diferenças descrita na Equação 10:

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - mT) + x(nT - 2mT) \quad (10)$$

No caso da equação 9 são colocados m zeros igualmente espaçados no círculo unitário e um pólo de cancelamento para $z=1$ (correspondendo a uma frequência de 0 Hz).

Hamilton [38] propõe a utilização de um filtro passa-baixo de *Lynn* de acordo com a Equação 11 que corresponde a uma equação de diferenças onde T é o período de amostragem e n é um número inteiro:

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T) \quad (11)$$

Para remover este tipo de ruído pode ser implementado um filtro passa-baixo com uma resposta em frequência do tipo da resposta a impulso Gaussiana.

A função de $\beta(n)$ em $h(k, n) \sim e^{-\beta(n)k^2}$ define a largura da função Gaussiana que estabelece que os segmentos com uma pequena taxa de variação sofrem uma filtragem significativa ao passo que os complexos *QRS* ao qual correspondem maiores taxas de variação do sinal são pouco filtrados.

A Figura 20 representa um sinal de *ECG* com e sem Ruído de modo a poder ser realizada uma distinção entre os dois tipos de sinal.

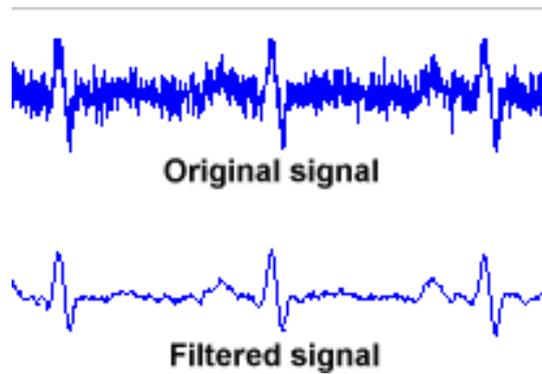


Figura 20: Sinal de *ECG* com e sem Ruído.

3.3.4 Detecção do complexo QRS

Para detecção do complexo QRS, Ahlstrom et al. [12] propoem a utilização de um filtro passa-banda de *Lynn* [41] de ordem 2 com a função de transferência definida na Equação 12 (para detecção do complexo QRS a frequência central geralmente utilizada em filtros analógicos é de 17 Hz). Na Equação 12 m corresponde ao número de zeros que se encontram no círculo unitário.

$$H(z) = \frac{(1 - z^{-m})^2}{(1 - 2(\cos\theta)z^{-1} + z^{-2})^2} \quad \text{com } \theta = 60^\circ, 90^\circ \text{ ou } 120^\circ \quad (12)$$

Este tipo de filtros passa-banda recorre a coeficientes inteiros, de modo a facilitar a sua implementação, pelo que o coeficiente $2\cos\theta$ so é um valor inteiro para valores de θ de 0, 60, 90, 120 ou 180. Os pólos em 0 e 180° correspondem a filtros passa-baixo e passa-alto, ao passo que polos em 60, 90 e 120° correspondem a filtros passa-banda com frequências centrais de $1/6T$, $1/4T$ e $1/3T$, respectivamente.

Se $\theta = 60^\circ$ a função de transferência passa a ser a que se encontra apresentada na Equação 13

$$H(z) = \frac{(1 - z^{-m})^2}{(1 - z^{-1} + z^{-2})^2} \quad (13)$$

A Equação 13 resulta na Equação 14 que é uma equação de diferenças.

$$y(nT) = 2y(nT - T) - 3y(nT - 2T) + 2y(nT - 3T) - y(nT - 4T) + x(nT) - 2x(nT - mT) + x(nT - 2mT) \quad (14)$$

Dependendo da frequência de amostragem, f_s previamente utilizada para amostrar o sinal, é possível determinar o número de zeros que o filtro deve ter de modo a que a frequência central seja de 17 Hz. Dado que a frequência central, f_c é definida pela equação $f_c = f_s/m$ em que f_s corresponde à frequência de amostragem e m corresponde ao número de zeros do filtro, é possível redefinir o número de zeros, m como sendo $m = f_c/f_s$. Como se pretende que $f_c = 17 \text{ Hz}$ tem-se que $m = 17/f_s$ [12].

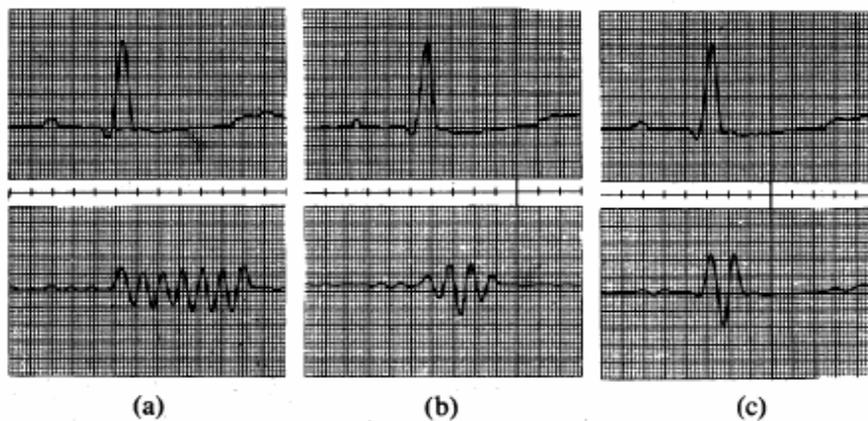


Figura 21: Detecção do Complexo QRS através de filtros passa-banda com larguras de banda nominais de $\pm 2.5 \text{ Hz}$, $\pm 8.3 \text{ Hz}$ e $\pm 11 \text{ Hz}$ [12].

3.4 COMPRESSÃO

A compressão de sinais biomédicos tem como objectivo a redução do espaço de armazenamento ou da banda de transmissão de um sinal sem perda de informações clinicamente relevantes, o que pode ser conseguido por eliminação das redundâncias do sinal, podendo esta ser efectuada com ou sem perda de informação.

Na compressão sem perdas, o sinal original pode ser totalmente recuperado, conseguindo-se rácios geralmente baixos (na ordem de 2 para 1). Na compressão com perdas de informação, uma vez que os dados são quantizados é possível atingir rácios de compressão

mais elevados (na ordem de 10 para 1). Este tipo de compressão é o mais comumente utilizado devido à possibilidade de rácios de compressão superiores e pelo facto de não introduzir degradações significativas no sinal codificado.

A nível da compressão de dados, existem 2 medidas importantes:

- Diferença Percentual do valor Quadrático médio, *PRD*(*Percentual Root Mean Square Difference*)

$$PRD = \sqrt{\frac{\sum_{n=0}^{N-1} [x(n) - x_{rec}(n)]^2}{\sum_{n=0}^{N-1} x^2(n)}} * 100 \quad (15)$$

onde N é o número total de amostras do sinal de *ECG* original, $x(n)$ e $x_{rec}(n)$ é o sinal de *ECG* reconstruído.

- Razão de Compressão (*CR*)

$$CR = \frac{\text{número de bits antes da compressão}}{\text{número de bits depois da compressão}} \quad (16)$$

No Capítulo designado de Algoritmos para Aquisição e Processamento são detalhadas algumas das técnicas de compressão de dados quer no domínio dos tempos quer no domínio das frequências.

Alguns exemplos destas técnicas de compressão são a Codificação de Huffman, Lempel-Ziv-Welch (*LZW*), e a codificação por transformada (Transformada de Fourier, Transformada do Cosseno, Transformada por *Wavelet*) Quando aplicável recorre-se às transformadas rápidas correspondentes.

ALGORITMOS PARA AQUISIÇÃO E PROCESSAMENTO

Neste capítulo são apresentados e analisados vários algoritmos importantes para o tratamento de *ECG* em mobilidade. Analisam-se algoritmos para redução de dados, detecção do complexo *QRS* e de compressão.

4.1 ALGORITMOS DE REDUÇÃO DE DADOS

4.1.1 Algoritmo Turning Point

O algoritmo *Turning Point* (Figura 22) processa 3 pontos de dados num determinado instante de tempo: um ponto de referência (y_0) e 2 pontos de dados consecutivos (y_1 e y_2).

O algoritmo tenta reter y_1 ou y_2 dependendo de qual é o ponto que preserva o declive dos 3 pontos originais.

Se $(y_1 - y_0)(y_2 - y_1) \geq 0$, é seleccionado o ponto y_2 , caso contrário é escolhido o ponto y_1 .

Neste método, apenas as amplitudes é que são armazenadas mas não os seus índices de posição, por isso existem alguns erros locais [13].

O algoritmo *Turning Point* (*TP*) reduz a quantidade de dados em 50% (redução de 2 para 1) em tempo real sem erros significativos.

O algoritmo de redução de dados *Turning Point* é de fácil implementação e requer uma pequeno número de computações.

Este algoritmo analisa as tendências dos pontos da amostra e armazena apenas um de cada par de pontos consecutivos.

O algoritmo *Turning Point* retém os pontos máximos e mínimos em que o sinal do declive do sinal original se altera (*turning points*).

O algoritmo *Turning Point* passa para metade a frequência de amostragem de um modo eficiente, ao mesmo tempo que preserva as principais características do sinal [42].

O algoritmo *Turning Point* é um método de decimação adaptativa que se baseia no declive resultante de duas amostras consecutivas do sinal, reduzindo a frequência de amostragem de um sinal de *ECG* por um factor de dois.

Neste método, para uma certa amostra, verifica-se qual das duas amostras seguintes origina um declive mais acentuado, sendo a outra amostra descartada.

O rácio de compressão originado por este método é de 2 para 1. Este pode ser inferior devido ao facto de as amostras que não foram

descartadas não se encontrarem igualmente espaçadas. Para determinar o intervalo de tempo correspondente a esse espaçamento podem ser necessários alguns *bits* adicionais [3].

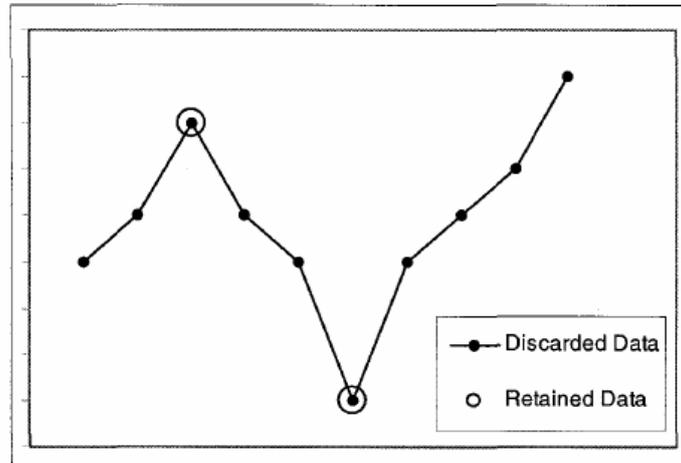


Figura 22: Algoritmo *Turning Point* [13].

Algorithm 1 Algoritmo Turning Point

var:

```
Double [] Ecg_original, TP
Int X0, X1, X2
Int S1, S2
Int i
```

Início

```
Função sign(x){
    Se(x==0) então {
        retorna 0
    }
    Senão{
        Se(x>0) então {
            retorna 1
        }
        Senão{
            retorna -1
        }
    }
}
```

```
Carregar o sinal original Ecg_original
Definir o array TP com metade do tamanho de Ecg_original
```

```
X0=Ecg_original[0]
TP[0]=X0
```

```
i=1
```

```
Enquanto(i<length(Ecg_original)){
```

```
    X1=org[i]
    X2=org[i+1]
```

```
    S1=sign(X1-X0)
    S2=sign(X2-X1)
```

```
    Se(!S1 || (S1+S2)) então {
```

```
        X0=X2
```

```
    }
```

```
    Senão{
```

```
        X0=X1
```

```
    }
```

```
    TP[i]= (Double) X0
    i++
```

```
}
```

Fim

4.1.2 Algoritmo AZTEC (*Amplitude-Zone-Time-Epoch-Coding*)

O algoritmo *AZTEC* (Figura 23) foi originalmente desenvolvido para pré-processar os sinais de *ECG* para análise de ritmo.

O *AZTEC* é um programa de pré-processamento que é activado por uma interrupção que converte as tensões de entrada na forma digital.

Este algoritmo recebe um conjunto de amostras que são posteriormente convertidas em *plateaus* ou em declives.

Os *plateaus* definem-se como zonas da representação gráfica com uma dada inclinação inicial e que estabilizam em torno de um determinado valor.

O *threshold* deste algoritmo é um valor de tensão diferencial.

Os *plateaus* *AZTEC* (linhas horizontais) são gerados por interpolação de ordem zero.

Quando surge um sinal de frequência e amplitude mais elevadas tal como o complexo *QRS* as amostras de tensão irão variar rapidamente e serão formadas linhas de curta duração.

Considera-se que uma série de linhas, cada uma contendo 4 amostras ou menos é adequadamente representada por uma taxa constante de alteração de tensão, ou declive, no caso de a diferença de tensão entre linhas adjacentes não mudar de sinal.

O declive é terminado por uma linha com mais de 4 amostras ou por uma mudança de sinal.

Apesar de o *AZTEC* fornecer um rácio de redução de dados elevado, o sinal reconstruído tem baixa fidelidade principalmente devido à descontinuidade (quantização em forma de degrau) das ondas [13].

O algoritmo *AZTEC* fornece uma redução de cerca de 20 para 1 por produção de *plateaus* e declives para codificar o sinal de *ECG*.

Esta técnica de redução de dados introduz descontinuidades e distorção significativas no sinal reconstruído.

O algoritmo *AZTEC* pega nos dados de *ECG* e substitui as sequências dos pontos da amostra por *plateaus* e declives.

Este algoritmo apresenta uma compressão de dados significativa, uma vez que apenas são transmitidos 2 parâmetros: a amplitude e o comprimento de um *plateau*.

O *AZTEC* recorre a 3 variáveis para produzir um valor de saída: o *threshold* da largura da abertura V_{th} , a duração máxima de um *plateau* V_e e o *threshold* da duração do *plateau* *PNT* que despoleta a produção de um declive[42].

O método *AZTEC* foi inicialmente desenvolvido como *software* de pré-processamento para monitorização em tempo real de *ECGs*. Provou ser útil para efeitos de análise automática e detecção de complexos *QRS*. Todavia não se revelou adequado para a visualização correcta desses sinais devido ao facto de o sinal reconstruído possuir forma de escada.

Este método baseia-se no facto de o sinal de *ECG* consistir de *plateaus* e inclinações. Considera-se que existe um *plateau* se o valor do sinal permanecer sensivelmente o mesmo para um número de amostras superior a 3. O *plateau* é codificado pela sua duração (número de amostras) e por um valor de amplitude fixa. Caso contrário, existe uma inclinação ou declive, sendo este armazenado pela sua duração e pela amplitude do último ponto da amostra.

Nesta técnica, recorre-se a interpolação linear para reconstruir o sinal original, o que resulta num sinal descontínuo.

Apesar de este método produzir um elevado rácio de compressão, a qualidade do sinal reconstruído é muito baixa. Foram propostos vários métodos alternativos entre os quais se destaca o método de *CORTES* que se assume como um misto entre o método *AZTEC* e o método do Ponto de Viragem (*Turning Point*) [3].

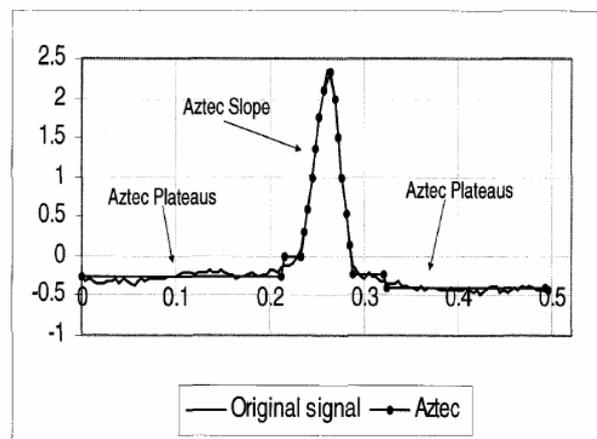


Figura 23: Algoritmo *Aztec* [13].

Nas Figuras 24 e 25 encontra-se representado um fluxograma do algoritmo *AZTEC*. Para que o programa não entre num ciclo infinito, define-se um condicional a seguir à entrada de amostras no sistema. No caso de existirem amostras é executado o algoritmo, caso contrário, este não é executado.

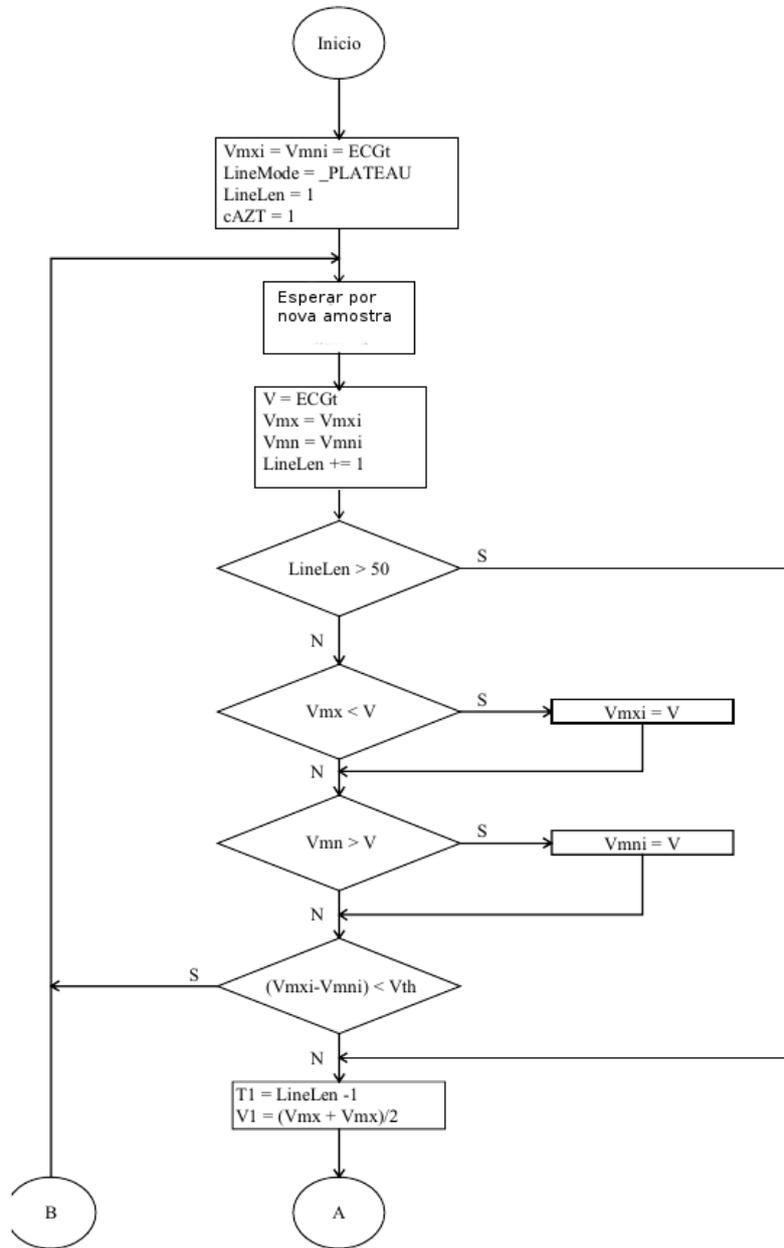


Figura 24: Algoritmo AZTEC (continua na Figura 25).

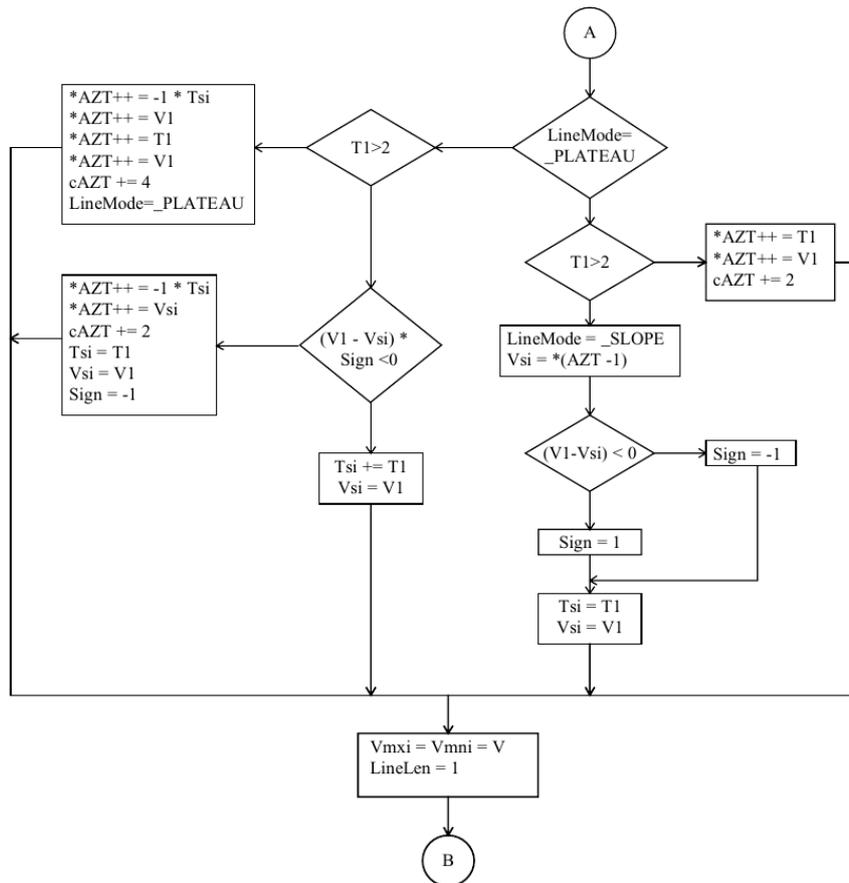


Figura 25: Algoritmo AZTEC (continuação).

4.1.3 Algoritmo de CORTES (Coordinate-Reduction-Time-Encoding-System)

O algoritmo CORTES (Figura 26) é uma combinação dos algoritmos AZTEC e *Turning Point*.

O algoritmo *Turning Point* preserva atributos importantes dos complexos QRS.

No entanto, fornece a mesma quantidade de redução de dados para áreas de sinal de mais baixa informação, tais como as ondas P e T.

Por outro lado, o AZTEC tem boas propriedades de redução de dados, adapta o seu modo de codificação à natureza da região do sinal e elimina ondas pequenas de linha de base.

O CORTES aplica o algoritmo *Turning Point* às regiões de alta frequência (complexos QRS) e o algoritmo AZTEC às regiões de frequência mais baixa e às regiões isopotenciais do sinal de ECG, onde ocorre grande redução de dados com pouca distorção.

O algoritmo CORTES é um algoritmo apropriado para a análise de ECG em tempo real em microprocessadores [13].

O algoritmo CORTES (*Coordinate-Reduction-Time-Encoding System*) aplica o algoritmo *Turning Point* às componentes clinicamente significativas de frequência mais elevada do sinal e o algoritmo AZTEC às regiões isopotenciais do ECG em que ocorre uma grande redução de dados com pouca distorção do sinal.

O algoritmo de redução de dados CORTES é um híbrido dos algoritmos *Turning Point* e AZTEC que foi concebido com o propósito de beneficiar dos pontos fortes de cada um deles. Baseia-se nas seguintes observações:

- O *Turning Point* produz uma boa qualidade de reconstrução do sinal para uma redução de dados de 2:1.
- O AZTEC descarta dados não significativos em termos clínicos no sinal de ECG com uma redução de dados tipicamente superior que 10:1 na região isoeleétrica.

À medida que o sinal de ECG é amostrado, o CORTES aplica em simultâneo o algoritmo AZTEC à região isoeleétrica do sinal de ECG e o algoritmo *Turning Point* às regiões clinicamente significativas de frequência mais elevada do sinal de ECG.

O CORTES utiliza um *threshold* de comprimento de linha, V_{ln} que é determinado experimentalmente.

Quando é gerado um *plateau* AZTEC, o CORTES armazena os dados AZTEC se o *plateau* for maior que V_{ln} ou guarda os dados *Turning Point* se o *plateau* for menor ou igual a V_{ln} .

Apenas são gerados os *plateaus* AZTEC; não são produzidos declives.

De modo a preservar a morfologia do complexo QRS, o CORTES usa o algoritmo *Turning Point* em vez dos declives AZTEC para codificar as ondas de frequências mais elevadas.

De modo a acelerar o processo, os dados provenientes do *Turning Point* são armazenados directamente no *array* de redução.

Usando 2 apontadores de *array* $P1$ e $P2$, o *CORTES* determina a diferença entre dados permanentes e probatórios.

Um marcador identifica as transições entre os dados *AZTEC* e *Turning Point*.

O marcador pode ser qualquer valor binário diferente do nível de saída do conversor *A/D*.

Se for utilizado um conversor de 8 *bits* com um microprocessador, tem de ser sacrificado um nível de quantização, reduzido a resolução em amplitude de 256 para 255 níveis.

O primeiro índice do *array* *CORTES* toma o mesmo valor que o marcador, $P1$ toma o valor 1 e $P2$ toma o valor 2.

São armazenados pontos sucessivos como dados *TP* probatórios, avançando o apontador $P2$ até ser encontrado um *plateau* *AZTEC*.

Se a duração do *plateau* for superior a V_{ln} é guardado um *plateau* *AZTEC* e os apontadores movem-se.

Se o *plateau* for menor que V_{ln} os dados *TP* passam a permanentes.

Depois de os dados serem armazenados permanentemente, o algoritmo começa a produzir outro *plateau* ao mesmo tempo que armazena dados *TP* num estado probatório.

Embora o *CORTES* tenha mais lógica de decisão que o *AZTEC*, o fluxo de dados é mais pequeno que o *AZTEC* uma vez que não existe codificação do declive.

A terminação dos *plateaus* e as decisões em qual delas armazenar são muito simples e rápidas.

Tal como o *AZTEC*, o *CORTES* utiliza V_e , a duração máxima de um *plateau* e V_{th} , o *threshold* da largura de abertura.

Em vez de *PNT*, o *threshold* do declive, o *CORTES* usa V_{ln} , o *threshold* de comprimento que determina se se armazena os dados do *plateau* ou os dados provenientes do algoritmo *Turning Point*.

Embora o algoritmo *CORTES* preserve a morfologia do complexo *QRS*, existem ainda descontinuidades na região isoelectrica, por isso é desejável algum *smoothing*.

Os *plateaus* *AZTEC* são convertidos em sequências de pontos de dados discretos e realiza-se interpolação entre cada par de pontos *Turning Point*.

Para filtrar o sinal de *CORTES* reconstruído aplica-se *smoothing* parabólico ao sinal [42].

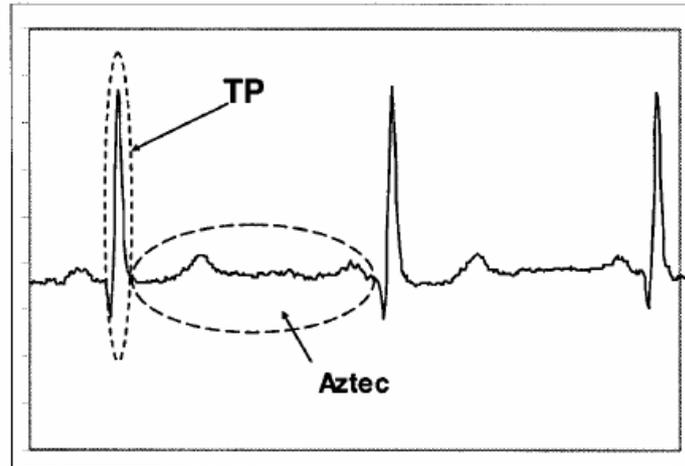


Figura 26: Algoritmo de *CORTES* [13].

4.2 DETECÇÃO DO COMPLEXO QRS

4.2.1 Algoritmo de Pan-Tompkins

No algoritmo de *Pan-Tompkins*, de modo a atenuar o ruído, o sinal original passa por um filtro passa-banda digital composto por filtros passa-baixo e passa-alto em cascata.

Filtro passa-baixo:

O filtro passa-baixo que constitui o filtro passa-banda definido por *Pan* e *Tompkins* é um filtro de 2ª ordem com a função de transferência apresentada na Equação 17:

$$H(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2} \quad (17)$$

que se traduz na Equação 18 que é uma equação de diferenças :

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T) \quad (18)$$

O ganho deste filtro é de 36 dB e possui uma frequência de corte de aproximadamente 11 Hz. Apresenta um atraso de processamento de 6 amostras.

Filtro passa-alto:

O filtro passa-alto resulta da subtração da saída de um filtro passa-baixo de 1ª ordem de um filtro passa-tudo (ou seja, as amostras do sinal original).

A função de transferência deste filtro é apresentada na Equação 19 que corresponde à Equação de diferenças 20:

$$H(z) = \frac{(-1 + 32z^{-16} + z^{-32})}{(1 + z^{-1})} \quad (19)$$

o que se traduz na equação de diferenças:

$$y(nT) = 32x(nT - 16T) - [y(nT - T) + x(nT) - x(nT - 32T)] \quad (20)$$

Este filtro tem um ganho de 32 dB, frequência de corte próxima dos 5 Hz e apresenta um *delay* de 16 amostras.

O próximo processo após a filtragem é a diferenciação, seguida de potenciação por um factor de 2 e posteriormente por um integrador de janela móvel.

Diferenciação:

Durante a fase de derivação é obtida informação acerca do declive do complexo QRS.

Após a filtragem, o sinal é diferenciado para fornecer a informação do declive do complexo QRS.

Para tal, recorre-se a uma derivada de 5 pontos com a função de transferência apresentada na Equação 21:

$$H(z) = (1/8T)(-z^{-2} - 2z^{-1} + 2z^1 + z^2) \quad (21)$$

A Equação 21 traduz-se na equação de diferenças apresentada na Equação 22.

$$y(nT) = (1/8T)[-x(nT - 2T) - 2x(nT - T) + 2x(nT + T) + x(nT + 2T)] \quad (22)$$

Apresenta um *delay* de 2 amostras.

Potenciação por um factor de 2

Após a diferenciação, o sinal é elevado ao quadrado ponto a ponto.

Esta operação que é uma transformação não linear é descrita pela Equação 23:

$$y(nT) = [x(nT)]^2 \quad (23)$$

O processo de potenciação por um factor 2 intensifica o declive da resposta em frequência da derivada. Permite também a eliminação de falsos positivos causados pelas ondas T com energias espectrais mais elevadas que o usual ao tornar todos os pontos de dados positivos.

Integrador de Janela móvel

O integrador de janela móvel produz um sinal que inclui informação sobre o declive e a largura do complexo QRS. Este determina-se a partir da Equação 24 onde N é o número de amostras na largura da janela de integração:

$$y(nT) = (1/N)[x(nT - (N - 1)T) + x(nT - (N - 2)T) + \dots + x(nT)] \quad (24)$$

O complexo QRS é detectado através de um conjunto de vários thresholds aplicados à resposta do integrador e ao sinal filtrado por um filtro passa-banda:

No sinal à saída do integrador de janela móvel aplicam-se os *thresholds* especificados nas Equações 25 e 26:

$$ThresholdI1 = NPKI + 0.25(SPKI - NPKI) \quad (25)$$

com $SPKI = 0.125PEAKI + 0.875SPKI$ (se $PEAKI$ for um máximo do sinal) e $NPKI = 0.125PEAK1 + 0.875NPKI$ (se $PEAKI$ for o máximo do sinal de ruído).

$$ThresholdI2 = 0.5ThresholdI1 \quad (26)$$

O primeiro *threshold*, $ThresholdI1$ é usado para uma primeira análise do sinal.

O segundo *threshold*, $ThresholdI2$ é usado se o complexo QRS não for detectado dentro de um certo intervalo de tempo. Nesta situação é usada uma técnica de *searchback* para tentar detectar o complexo QRS. $PEAKF$ é o máximo do sinal.

Quando o complexo QRS é detectado a partir do segundo *threshold*,

$$SPKI = 0.25PEAKI + 0.75SPKI \quad (27)$$

No sinal à saída do filtro passa-banda aplicam-se os *thresholds* especificados nas Equações 28 e 29:

$$ThresholdF1 = NPKF + 0.25(SPKF - NPKF) \quad (28)$$

com $SPKF = 0.125PEAKF + 0.875SPKF$ (se $PEAKF$ for um máximo do sinal) e $NPKF = 0.125PEAKF + 0.875NPKF$ (se $PEAKF$ for o máximo do sinal de ruído).

$$ThresholdF2 = 0.5ThresholdF1 \quad (29)$$

O primeiro *threshold*, $ThresholdF1$ é usado para uma primeira análise do sinal.

O segundo *threshold*, $ThresholdF2$ é usado se o complexo QRS não for detectado dentro de um certo intervalo de tempo de modo que é

necessária uma técnica de *searchback* para tentar detectar o complexo QRS.

$PEAKF$ é o máximo do sinal.

Quando o complexo QRS é detectado a partir do segundo *threshold*,

$$SPKF = 0.25 PEAKF + 0.75 SPKF \quad (30)$$

Para ritmos cardíacos irregulares, os *thresholds threshold I1* e *threshold F1* são reduzidos para metade de modo a aumentar a sensibilidade da detecção e evitar perder alguns batimentos do sinal [43].

O algoritmo de *Hamilton-Tompkins* recorre a um pré-processador baseado num filtro passa-banda. Este filtro passa-banda efectua filtragem linear e não linear sobre um sinal de ECG. Isto é utilizado para obter informação sobre o declive, a duração e a amplitude do complexo QRS juntamente com um estágio de decisão otimizado para detecção do mesmo.

Um filtro passa-banda limita a frequência da banda de passagem ao intervalo de 5 a 12 Hz para atenuação do ruído.

Posteriormente, o declive acentuado da onda R é salientado através da diferenciação e potenciação por um factor de 2.

O sinal resultante é submetido a *smoothing* por utilização de integração de uma janela deslizante. Isto permite obter uma estimativa da energia num curto espaço de tempo que é posteriormente usada como uma estatística de detecção [44].

4.2.2 Algoritmo MOBD (*Multiplication Of the Backward Difference*)

O algoritmo MOBD (*Multiplication of the backward difference*) (Figura 27) aplica filtragem não linear para detectar o complexo QRS.

Este quantiza o sinal de ECG num menor número de *bits* antes de aplicar a transformada não linear MOBD.

O quantizador atenua as ondas de baixa frequência e baixa amplitude P e T e põe em evidência os complexos QRS.

Posteriormente a primeira diferença (*backward difference*) é elevada ao quadrado para destacar as características de alta frequência do sinal como o complexo QRS.

Os sinais que são caracterizados por uma amplitude elevada e por um declive acentuado, tais como o complexo QRS, são ainda mais destacados através da multiplicação de várias amostras sucessivas da *backward difference* ou primeira diferença do sinal.

Esta multiplicação resulta numa *backward difference* ou primeira diferença igual a 0 para as ondas P e T quando a transformada MOBD é aplicada [44].

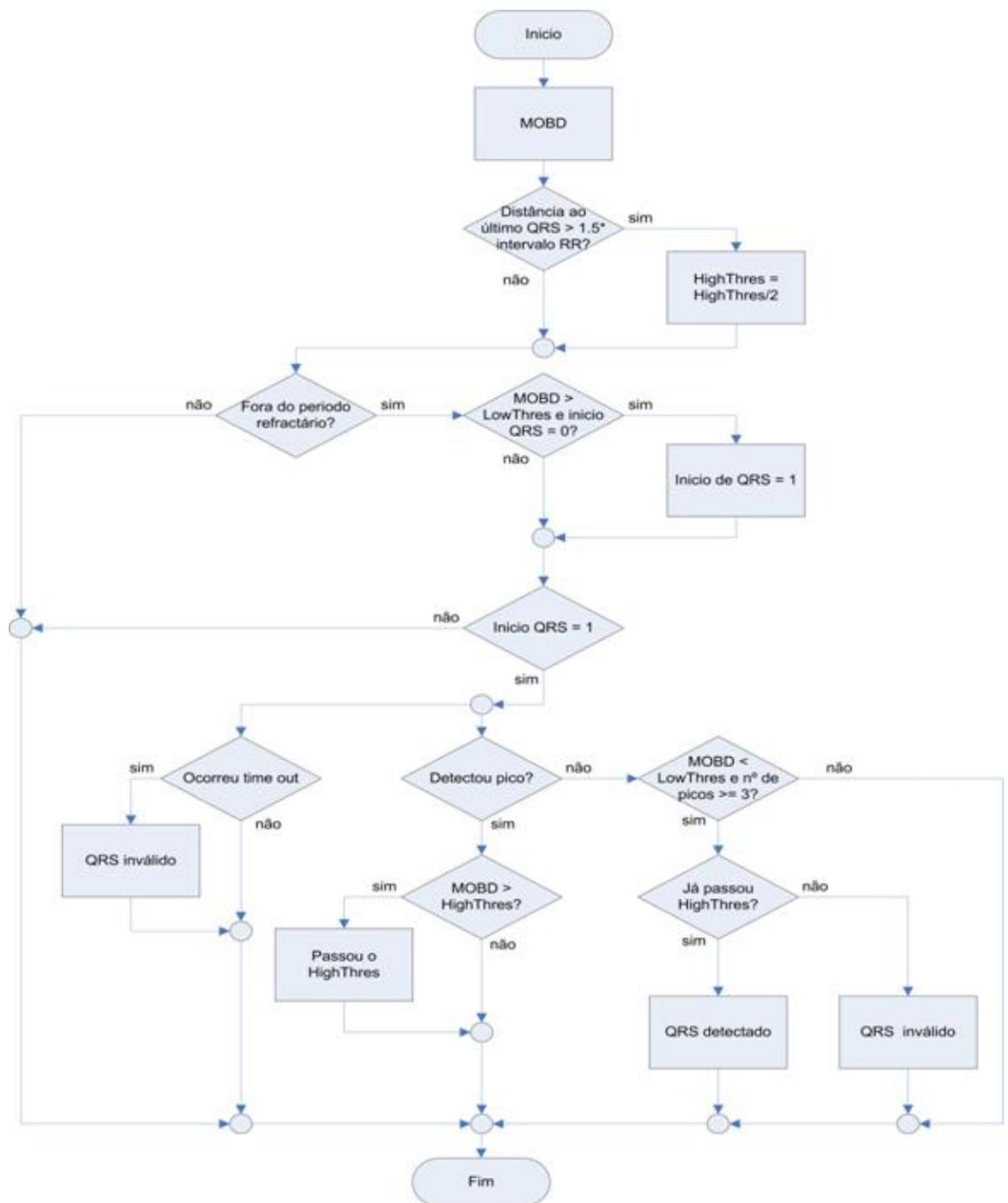


Figura 27: Algoritmo MOBD.

4.2.3 Algoritmo de Okada

O algoritmo de Okada (Algoritmo 2) recorre a técnicas de filtragem no domínio digital para detecção do complexo *QRS*.

Como um primeiro passo, as componentes de alta frequência são separadas do sinal de *ECG* amostrado.

O *ECG* é amostrado à frequência de 1000 amostras/s.

Este algoritmo computa a diferença entre uma média móvel de 3 pontos do sinal original e a saída de um filtro passa-baixo, o que determina uma operação de filtragem passa-banda do sinal original.

Posteriormente, essa diferença é elevada ao quadrado.

Para destacar o complexo *QRS*, o sinal resultante é multiplicado por uma versão de si próprio filtrada não-linearmente.

Os extremos do sinal resultante correspondentes a falsos positivos que ocorrem devido a uma grande flutuação de linha de base são removidos por procura de extremos locais na saída do filtro de média móvel de 3 pontos [44].

1. As médias de cada 4 amostras são calculadas em tempo real e armazenadas, sendo o sinal reamostrado a 250 Hz.

Posteriormente, os valores armazenados são submetidos a *smoothing* através de uma média móvel de 3 pontos ponderada com os coeficientes 1,2,1.

O sinal resultante é representado por $\{y_{1i}\}$ ($i = 1, 2, \dots, n$).

2. Determina-se a saída do seguinte filtro passa-baixo:

$$y'_{1i} = \frac{1}{2m+1} \sum_{k=i-m}^{i+m} y_{1k} \quad (31)$$

De seguida, calcula-se o quadrado das diferenças entre $\{y_{1i}\}$ e $\{y'_{1i}\}$ ($i = m+1, \dots, n-m$) do seguinte modo:

$$y_{2i} = (y_{1i} - y'_{1i})^2 \quad i = m+1, \dots, n-m \quad (32)$$

3. As componentes de alta frequência dos complexos *QRS* surgem por um longo período de tempo desde o *onset* até ao fim quando comparados com os das outras ondas.

Deste modo é determinada a onda modificada $\{y_{3i}\}$ como:

$$y_{3i} = \frac{y_{2i}}{64} \left\{ \sum_{k=i-m}^{i+m} \frac{y_{2k}}{64} \right\}^2 \quad i = m+1, m+2, \dots, n-m \quad (33)$$

Em $\{y_{3i}\}$, os segmentos com complexos *QRS* são destacados, mas os máximos de curta duração em $\{y_{2i}\}$ são atenuados.

Existem casos no entanto, em que alguns eventos tais como a flutuação da linha de base muito acentuada num conjunto de estágios distintos, originam máximos adicionais em $\{y_{3i}\}$.

4. Para solucionar este problema coloca-se o peso em $\{y_{3i}\}$. Deste modo, determina-se $\{y_{4i}\}$ como se apresenta na Equação 34:

$$y_{4i} = k.y_{3i} \quad i = m + 1, m + 2, \dots, n - m \quad (34)$$

em que

$$k = 1 \quad se(y_{1i} - y_{1i-m})(y_{1i} - y_{1i+m}) > 0 \quad (35)$$

$$k = \phi \quad se(y_{1i} - y_{1i-m})(y_{1i} - y_{1i+m}) < 0 \quad (36)$$

Neste caso, y_{1i-m} e y_{1i+m} são os pontos localizados m pontos à frente e m pontos atrás do ponto y_{1i} , respectivamente.

m deve ser escolhido de modo a que um segmento de $2m + 1$ pontos sucessivos corresponde à média da duração das ondas Q , R e S .

Deste modo, o sinal de $(y_{1i} - y_{1i-m})(y_{1i} - y_{1i+m})$ é positivo para os complexos QRS e negativo para a flutuação gradual, uma vez que o elementos dos complexos QRS (onda Q , onda R , onda S) são simétricos relativamente ao eixo dos tempos ao passo que a onda de flutuação gradual não o é.

5. Procurar na onda $\{y_{4i}\}$ os pontos que ultrapassam o valor de *threshold* h definido como:

$$h = (\max\{y_{4i} | i = m + 1, m + 2, \dots, n - m\}) / 32 \quad (37)$$

Os pontos que forem encontrados são divididos em segmentos de 144 ms.

A onda $\{y_{4i}\}$ é convertida numa onda rectangular $\{y_{5i}\}$, sendo que esses segmentos colocados em *ON* e os restantes colocados em *OFF* [45].

Numa fase inicial, o sinal de *ECG* é sujeito a *smoothing* através de um filtro de média móvel de 3 pontos apresentado na Equação 38:

$$Y_0[n] = \frac{1}{4}(X[n - 1] + 2X[n] + X[n + 1]) \quad (38)$$

A saída deste filtro passa depois através de um filtro passa-baixo:

$$Y_1[n] = \frac{1}{2m+1} \sum_{k=n-m}^{n+m} Y_0[k] \quad (39)$$

A diferença entre a entrada e a saída deste filtro é elevada ao quadrado, de modo a suprimir as ondas de baixa amplitude relativamente ao pico R :

$$Y_2[n] = (Y_0[n] - Y_1[n])^2 \quad (40)$$

Esta diferença elevada ao quadrado é posteriormente filtrada de modo a ampliar a área do complexo QRS relativamente às outras características do sinal de ECG .

O *threshold* de amplitude é calculado como uma fracção do sinal de ECG :

$$X_{th} = \beta \max\{X[n]\} \quad (41)$$

onde β é a percentagem do sinal de ECG a ser removida e $0 \leq \beta \leq 1$.

Posteriormente à aplicação de um *threshold* de amplitude é calculada e aplicada a primeira derivada para acentuar o declive do complexo QRS .

Para determinação da primeira derivada, *Okada* recorreu a um esquema de primeiras diferenças (*backward difference*) [46] apresentado na Equação 42:

$$Y[n] = X[n] - X[n-1] \quad (42)$$

Algorithm 2 Algoritmo de *Okada*

```
var:

Double [] T0, Y0
Double [] Medias4Pontos, MediasMoveis, MediaMovel3Pontos
Double [] y1Filtrado, y1
Double MediaActual
Double SomaY1, SomaY2
Int i, k, m

Início

Carregar o sinal original (T0,Y0)

MediaActual, SomaY1, SomaY2, k = 0
m=3

Para {i = 0 até i < length(Y0), passo 4}

    MediaActual=(y0(i)+y0(i+1)+y0(i+2)+y0(i+3) /4
    Medias4Pontos=Medias4Pontos + MediaActual

Para (j = 0 até j < length(Medias4Pontos), passo 3){
    MediaMovel3Pontos
        =(Medias4Pontos(j)+2*Medias4Pontos(j+1)+Medias4Pontos(j+2))/3
    MediasMoveis=MediasMoveis + MediaMovel3Pontos
}

y1 = MediasMoveis

Para {k= 1 até k < length(y1)}
    SomaY1=SomaY1+y1(k)
    y1Filtrado = y1Filtrado + SomaY1/(2m+1)
}

Para (k= 1 até k < length(y1), k++){
    SomaY2=SomaY2+y2(k)/64
}

Para (k= 1 até k < length(y1)){
    Para (i=m+1 até i <= length(y1)-m, i++){
        y2(i)=(y1(i)-y1Filtrado(i))^2
        y3(i)=y2(i)*(SomaY2^2)/64

        Se ((y1(k)-y1(k-m))(y1(k)-y1(k+m)) > 0) então {
            k=1
        Senão {
            k=0
        }

        Para (l=m+1 até l < length(y1)-m, m++){
            y4(l)=k*y3(k)
            h=max(y4)/32
        }
    }
}

Fim
```

4.3 COMPRESSÃO

4.3.1 Compressão no domínio dos tempos

4.3.1.1 Codificação de Huffman

A codificação de *Huffman* [11] cria um código de comprimento variável (VLC) (*Variable Length Coding*) para cada símbolo da fonte de modo que o número de *bits* no código é aproximadamente inversamente proporcional à probabilidade de ocorrência desse símbolo.

O código de *Huffman* para uma fonte de informação é gerado em 2 passos:

- **Redução da fonte:**

- **Passo 1:** Os símbolos são organizados com probabilidade decrescente de ocorrência.
- **Passo 2:** Depois de os símbolos se encontrarem ordenados, é criado um novo símbolo por combinação dos 2 símbolos menos prováveis.
- **Passo 3:** Junta-se novamente os 2 símbolos menos prováveis e reordena-se as novas probabilidades. Repete-se este processo até a fonte mais recente ter apenas 2 símbolos.

- **Atribuição de código:**

- **Passo 1:** A seguir ao processo de redução da fonte, inicia-se o processo de atribuição de código. Aos 2 primeiros símbolos atribuem-se os códigos 0 e 1.
- **Passo 2:** A probabilidade à qual se atribuiu o código 0 foi obtida com base em 2 probabilidades do passo anterior. Deste modo, os códigos atribuídos a estas probabilidades são 00 e 01.
- **Passo 3:** A outra probabilidade foi obtida por junção de 2 probabilidades anteriores. Neste passo, o código atribuído a estas probabilidades é 10 e 11.
- **Passo 4:** Uma das outras probabilidades foi obtida por combinação de 2 probabilidades do passo anterior. Neste passo, os códigos atribuídos a estas probabilidades são 010 e 011.

- **Passo 5:** Repete-se o mesmo processo até se ter códigos *Huffman* para todos os símbolos da fonte original.

A descodificação do código de *Huffman* é única para uma determinada tabela de *Huffman*; não é possível obter resultados diferentes.

4.3.1.2 Lempel-Ziv-Welch (LZW)

O método de compressão *LZW* [47] é um método de compressão com base num dicionário que começa com a sua inicialização para todos os caracteres do código *ASCII* (Tabela 2).

O primeiro símbolo da sequência a comprimir encontra-se no dicionário uma vez que este já foi inicializado.

Neste método de compressão, os símbolos vão sendo adicionados e concatenados um a um, a uma *string* *I*.

Sempre que um símbolo é concatenado à *string* *I*, realiza-se a procura desta *string* no dicionário, continuando o processo até *I* não existir no dicionário.

Ao adicionar o próximo símbolo *x* isto faz com que a pesquisa no dicionário falhe. A *string* *I* encontra-se no dicionário mas a *string* *Ix* (símbolo *x* concatenado a *I*) não.

O codificador coloca na saída o apontador do dicionário que aponta para a *string* *I*, guarda a *string* *Ix* na próxima entrada do dicionário disponível e inicializa a *string* *I* com o símbolo *x*.

4.3.1.3 Differential Pulse Code Modulation (DPCM)

A técnica de compressão de dados *DPCM* consiste na remoção da componente previsível de cada amostra de um sinal por predição linear, em que uma determinada amostra, $x(n)$, é calculada a partir das amostras anteriores por recurso a um preditor linear apresentado na Equação 43:

$$\hat{x}(n) = \sum_{k=1}^N a_k x(n-k) \quad (43)$$

onde $\hat{x}(n)$ corresponde a um valor aproximado de $x(n)$ para o instante de tempo discreto n , sendo $\{a_k\}$ o peso do preditor.

Nesta técnica de compressão, a sequência de erro $e(n) = x(n) - \hat{x}(n)$ é quantizada e posteriormente codificada por entropia através da atribuição de códigos de comprimento variável de acordo com a frequência de ocorrência, sendo que o dicionário correspondente se baseia na codificação *Huffman* que atribui códigos mais pequenos ou mais longos aos valores que ocorrem com probabilidades mais elevadas ou mais baixas, respectivamente, produzindo resultados de

Tabela 2: Tabela de códigos ASCII [33].

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

compressão bastante próximos do valor de entropia da sequência de erro.

Neste método de compressão utiliza-se cerca de 1 *bit* por amostra para representar o sinal de ECG.

Esta técnica de compressão de dados pode ser realizada sem perdas se não se proceder à quantização das sequências de erro, conduzindo no entanto a menores rácios de compressão.

4.3.2 Codificação por Transformada

4.3.2.1 Transformada de Fourier discreta

A Transformada de Fourier Discreta (*DFT*) de um sinal é definida pela Equação 44:

$$f(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} F(k) W_N^{-kn} \quad 0 \leq n \leq N-1 \quad (44)$$

com $f(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) W_N^{kn} \quad 0 \leq k \leq N-1$ em que $f(k)$ é o coeficiente da *DFT* de ordem k e $W_N = e^{-\frac{j2\pi}{N}}$.

4.3.2.2 Transformada rápida de Fourier

A *FFT* é um algoritmo computacionalmente eficiente para o cálculo da *DFT*.

Para que o algoritmo possa ser usado o número de amostras do sinal tem que ser uma potência de base 2 ou seja é necessário considerar r^t tal que o número de pontos no tempo seja 2^t .

4.3.2.3 Transformada do Cosseno Discreta

A transformada do cosseno discreta (*DCT*) [11] unidimensional, para uma sequência de entrada $f(n)$ é definida como na Equação 45:

$$F(k) = \alpha(k) \sum_{n=0}^{N-1} f(n) \left[\cos \frac{(2n+1)\pi k}{2N} \right] \quad 0 \leq k \leq N-1 \quad (45)$$

$$\text{com } \alpha(k) = \begin{cases} \sqrt{1/N} & k = 0 \\ \sqrt{2/N} & 1 \leq k \leq N - 1 \end{cases}$$

A entrada é um conjunto de N amostras $f(n)$ e a saída é um conjunto de N coeficientes da transformada DCT .

Quando os dados de entrada consistem de quantidades correlacionadas, a maior parte dos N coeficientes da transformada derivados da DCT tomam o valor de zero ou números pequenos e apenas alguns possuem valores elevados (geralmente os primeiros).

A compressão com a DCT é efectuada por quantização dos coeficientes.

4.3.2.4 Transformada de Wavelet Discreta

A transformada de Wavelet [11, 3] resulta da combinação das técnicas de transformada e de decomposição por sub-banda.

Este tipo de transformada inclui um filtro FIR passa-baixo $h[.]$ e um filtro FIR passa-alto $g[.]$, sendo os seus coeficientes calculados recursivamente a partir das Equações 46 e 47 para uma sequência discreta de N pontos $c_{0,k}$, com $k \geq 0$ e $k \leq N - 1$:

$$c_{j+1,k} = \sum_m c_{j,m} h[m - 2k] \quad 0 \leq k \leq N/2^{j+1} - 1 \quad (46)$$

$$d_{j+1,k} = \sum_m c_{j,m} g[m - 2k] \quad 0 \leq k \leq N/2^{j+1} - 1 \quad (47)$$

Apresenta-se na Tabela 3 com os coeficientes das *Wavelets* de *Daubchies* para $N=2$, $N=4$, $N=8$ e $N=12$:

A primeira *wavelet* que se encontra na Tabela 3 é designada de *Wavelet* de *Haar* [11].

A transformada de *wavelet* é implementada com recurso a bancos de filtros de sub-banda (*SBC*) que consistem de um filtro passa-baixo seguido de um filtro passa-alto [3].

Neste método de compressão, o sinal original é amostrado a uma determinada frequência, sendo posteriormente sujeito a filtragem do tipo passa-baixo e passa-alto, obtendo-se dois subsinais, cada um com uma frequência de amostragem de $f_s/2$, podendo ocorrer a divisão sucessiva desses subsinais nas suas componentes de baixa e alta frequência até o domínio da frequência se encontrar suficientemente dividido.

	n	h[n]		n	h[n]
N=2	0	0.70710678	N=12	0	0.111540743350000
	1	0.70710678		1	0.494623890398000
N=4	0	0.482962913144		2	0.751133908021000
	1	0.836516303737		3	0.315250351709000
	2	0.224143868042		4	-0.226264693965000
	3	-0.129409522551		5	-0.129766867567000
N=8	0	0.230377813308		6	0.097501605587000
	1	0.714846570552		7	0.027522865530000
	2	0.630880767939		8	-0.031582039318000
	3	-0.027983769416		9	0.000553842201000
	4	-0.18703481171		10	0.004777257511000
	5	0.030841381835		11	-0.001077301085000
	6	0.032883011666			
	7	-0.010597401785			

Tabela 3: Coeficientes das *wavelets* de *Daubechies* para N=2, N=4, N=8 e N=12 [11].

Os subsinais resultantes podem ser codificados por *DPCM*, por entropia ou pela transformada de modo a atingir-se o melhor rácio de compressão possível para cada banda de frequências.

A reconstrução do sinal é efectuada a partir dos subsinais codificados que são sujeitos a interpolação seguida de filtragem (filtro passa-baixo no caso dos subsinais de baixa frequência e filtro passa-alto no caso dos outros sinais), sendo que o sinal reconstruído corresponde à soma dos valores de saída de cada um dos filtros [3].

4.3.2.5 Transformada rápida de wavelet

A transformada rápida de *wavelet* permite reduzir o tempo de computação e de determinação dos seus coeficientes, à semelhança do que ocorre na *FFT*.

PROJECTOS RELACIONADOS

Neste capítulo são apresentados e analisados alguns dos projectos que se relacionam com este trabalho, estabelecendo uma divisão entre os sistemas de aquisição e os sistemas de monitorização propriamente ditos. Consta deste capítulo uma tabela com as características de alguns sistemas de aquisição (Tabela 4), uma tabela com as características gerais de alguns sistemas de monitorização (Tabela 5) e uma tabela com as características principais dos sensores de *ECG* e de *acelerometria* para os vários sistemas (Tabela 6) juntamente com uma análise comparativa das mesmas.

5.1 SISTEMAS DE AQUISIÇÃO DE BIOSINAIS

5.1.1 BITalino

BITalino [48, 49, 15] é um projecto português desenvolvido por elementos do Instituto Superior Técnico (*IST*) e do Instituto Superior de Engenharia de Lisboa (*ISEL*) em parceria com o Instituto de Telecomunicações (*IT*). Consiste num sistema de aquisição de sinais vitais modular e de baixo custo, com base na plataforma de *hardware* Arduino.

O BITalino foi criado com o objectivo de integrar uma plataforma biométrica para reconhecimento em tempo real de cada indivíduo a partir dos seus dados de *ECG*, obtidos por sensorização [48].

Esta plataforma veio recentemente a permitir um conjunto mais vasto de aplicações. Pode ser utilizada para auxiliar actividades de sala aula, para fazer a *interface* com outros dispositivos ou para realizar prototipagem rápida de aplicações no âmbito da computação fisiológica [49].

A versão actual desta plataforma integra múltiplos sensores fisiológicos, nomeadamente sensores de *Electrocardiografia (ECG)*, *Electromiografia (EMG)*, *Actividade Electrodermal (EDA)*, e *Acelerometria (ACC)*. Possui também um Sensor de Luz (*LUX*) e um Díodo Emissor de Luz (*LED*) que funciona como actuador e permite a sincronização com outros dispositivos (como um ecrã de computador ou uma câmara de vídeo).

O sensor de *ECG* inclui um Amplificador de Instrumentação com acoplamento *AC* para rejeitar as tensões de entrada *DC*, seguido de um filtro passa-baixo Butterworth de 4^a ordem, sendo a largura de banda deste sinal de 0.5-40 Hz [49].

Na primeira versão do sistema, foi utilizado um filtro passa-baixo de Kaiser entre 2.5 e 30 Hz para filtrar este mesmo sinal [48].

O acelerómetro (ADXL335), por sua vez, permite adquirir valores em cada um dos 3 eixos (x, y e z) entre -3G e +3G com uma largura de banda de 0.5-50 Hz.

Para além deste conjunto de sensores, o sistema é constituído por um módulo de processamento, por um módulo de gestão da energia e por um módulo de transmissão com *Bluetooth* v2.0 de classe II (alcance até 10 m).

O módulo de processamento é constituído pelo microprocessador AVR de 8 bits e 8 MHz ATmega328P que inclui um conversor analógico-digital (ADC) de aproximações sucessivas com uma resolução de 10 bits e por um *Multiplexer* analógico de 6 canais que recebe valores de tensão de entrada entre 0 e 3.3 V.

O módulo de gestão de energia é alimentado por uma Bateria de Iões de Lítio com uma Tensão Nominal de 3.7 V e uma capacidade padrão de 320 mAh e possui um regulador de baixa tensão de 3.3 V.

O sistema apresenta uma autonomia de ≈ 4 horas para uma capacidade da bateria de 320 mAh, ≈ 6 horas para uma capacidade de 400 mAh [48] e ≈ 17 horas para uma capacidade de bateria de 850 mAh [15].

O BITalino possui 6 portas analógicas de entrada, 4 a 10 bits e 2 a 6 bits, e 8 portas digitais a 1 bit (4 de entrada e 4 de saída).

O *firmware* define alguns parâmetros do microcontrolador como a frequência de amostragem, a taxa de transmissão de dados ou *baud rate* e o protocolo de comunicação.

A frequência de amostragem possui um valor pré-definido de 1000 Hz, podendo assumir também os valores de 1, 10 ou 100 Hz; a taxa de transmissão de dados (Baud Rate) apresentada pelo sistema é de 115200 símbolos por segundo.

Relativamente ao modo de funcionamento do sistema, os canais de entrada analógicos e digitais são continuamente amostrados, os dados são organizados num conjunto de *bytes* e estes pacotes de dados são enviados através do *USART* incluído no microcontrolador.

Estes pacotes de dados integram um identificador da sequência a transmitir e um CRC para detectar possíveis erros na mensagem e recuperar de inconsistências dos pacotes de dados, ambos de 4 bits, e detêm um tamanho que varia entre 3 bytes e 8 bytes.

O microcontrolador ATmega328P possui 2 *Timers*, um de 16 bits e um de 8 bits, nos quais foram implementadas *ISRs* (*Interrupt Service Routines*), sendo que a primeira, quando invocada define a frequência de amostragem, amostra os canais e preenche um *buffer* circular do tipo *FIFO*, ao passo que a segunda rotina obtém as amostras do *buffer*, organiza os dados em pacotes e calcula o CRC antes de enviar os pacotes através do *USART* que está incluído no microprocessador, e consequentemente para o módulo *Bluetooth* [49].

Este sistema pode existir sob a forma de uma de 3 versões:

- A versão *Board* (Figura 28) em que os sensores se encontram ligados de forma irreversível a uma placa base que integra um módulo da unidade de controlador, um módulo de gestão de energia e um módulo *Bluetooth*.

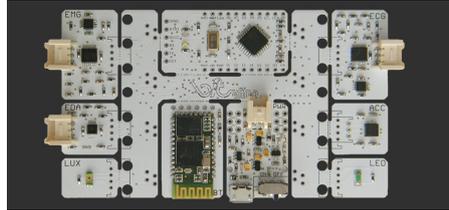


Figura 28: BITalino: Versão *Board* [14].

- A versão *Plugged* (Figura 29) em que os sensores se encontram separados da placa base e aos quais se adicionam cabos *RJ22* para os ligar à placa base.

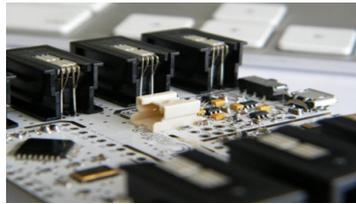


Figura 29: BITalino: Versão *Plugged* [15].

- A versão *Freestyle* (Figura 30) inclui os vários módulos presentes na versão *Board* sem estarem conectados entre si permitindo a combinação dos vários módulos conforme pretendido.

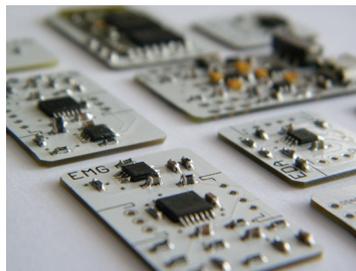


Figura 30: BITalino: Versão *Freestyle* [15].

5.1.2 e-Health Sensor Platform

A placa de sensores *e-Health Sensor Platform* 31 da *Cooking Hacks* (divisão de *open Hardware* do Libelium) é uma plataforma de sinais vitais que permite desenvolver aplicações médicas e biométricas de monitorização. Integra 10 tipos de sensores diferentes: sensor de pulso, saturação de oxigénio no sangue (SpO_2), *airflow* (respiração), temperatura corporal, electrocardiograma (ECG), glucómetro, resposta galvânica da pele (GSR), pressão sanguínea, posição do paciente (acelerómetro) e sensor de electromiografia (EMG).

Esta placa pode ser acoplada a uma placa *Arduino* ou a uma placa *RaspberryPi* 32. Neste último caso, é necessário interpor uma placa adaptadora entre a placa *RaspberryPi* e a placa *e-Health*.

A informação biométrica recolhida pode ser enviada por utilização de qualquer uma das opções de conectividade disponíveis: *Wi-Fi*, *3G*, *GPRS*, *Bluetooth*, *802.15.4* e *ZigBee*.

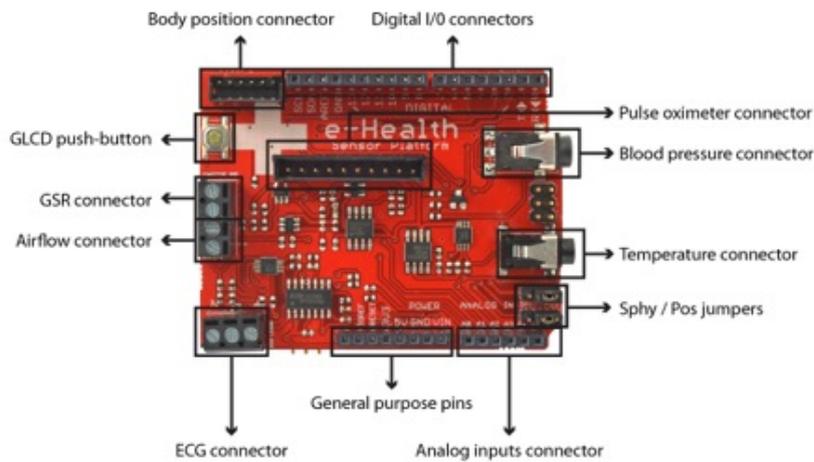


Figura 31: *e-Health Sensor Platform* [16].

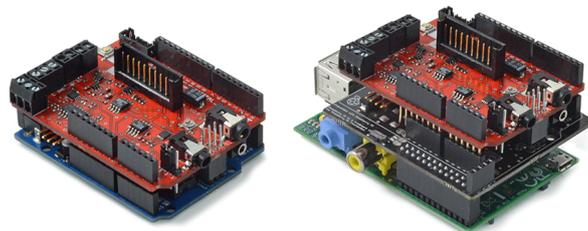


Figura 32: *e-Health Sensor Platform* com *Arduino* (à esquerda) e *Raspberry-Pi* (à direita) [16].

5.1.3 *Sensium*

A placa *Sensium 40* foi desenvolvida pela *Queen Mary University London* e foi o primeiro *System-on-chip* concebido especificamente para a monitorização de sinais vitais.

O *chip* inclui um transdutor de baixo consumo, um protocolo *MAC*, um microprocessador digital *8051-eWarp* de *1 MHz*, periféricos de input/output, memória *RAM* de *64 KB*, um conversor analógico-digital de *10 bits* e algumas interfaces de sensores padrão.

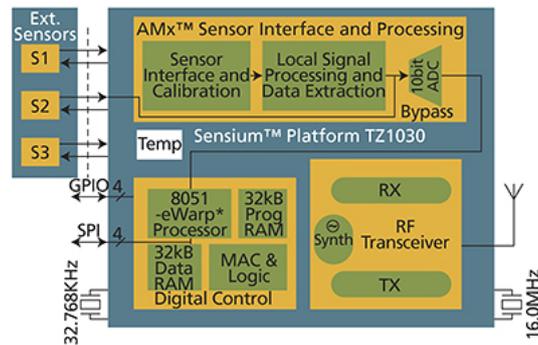


Figura 33: Sensium [17].

5.1.4 *Pluto*

O *Pluto 34* é um *mote* desenvolvido no âmbito do projecto *Code-Blue* da Universidade de *Harvard* que se baseia no *mote Tmote Sky* e integra um microcontrolador *TI MSP430*, um rádio *ChipCon CC2420* e uma bateria de iões de lítio com uma capacidade de *120 mAh*.



Figura 34: Pluto [18].

5.1.5 CPOD (Crew Physiologic Observation Device)

O dispositivo CPOD (Figura 35) integra o projecto *Lifeguard* da NASA e é um sistema de monitorização de parâmetros fisiológicos concebido com o objectivo de monitorizar o estado de saúde dos astronautas durante as suas missões espaciais.

Este pode guardar os parâmetros fisiológicos até 9 horas numa memória flash de 32 MB e pode efectuar o download de dados gravados ou em tempo real para um computador através de uma ligação *standard* RS232.

Alternativamente, o CPOD pode enviar informação em tempo real através das tecnologias *Bluetooth* ou *FSK* (*Frequency Shift Keying*) de 916 MHz para outro dispositivo.

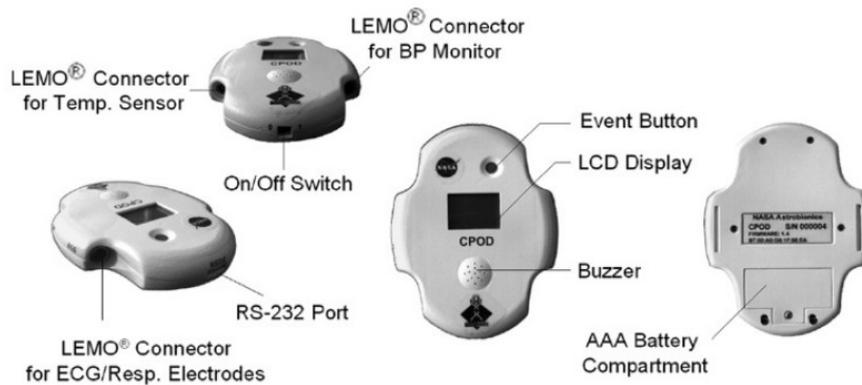


Figura 35: CPOD [19].

5.1.6 Shimmer

Shimmer (Figura 36) é uma plataforma de sensores sem fios capaz de registar e transmitir dados fisiológicos e de movimento em tempo real.

Esta plataforma permite fazer de *interface* com outros sensores sem fios: *ECG*, *EMG*, *GSR*, acelerómetro, giroscópio e um sensor de vibração *PIR* (*Passive Infra Red*).

O *Shimmer* transmite os dados em tempo real através da tecnologia *Bluetooth* ou do protocolo da *IEEE 802.15.4*.

IEEE 802.15.4 é um *standard* que especifica a camada física e controlo de acesso aos media no modelo *OSI* para *LR-WPANs* (*Low-Rate Wireless Personal Area Networks*).

Também possui um cartão *SD* para armazenamento local.



Figura 36: Shimmer [20].

5.1.7 Zephyr BioHarness

Zephyr BioHarness (Figura 37) consiste num dispositivo colocado numa fita, ao qual se ligam eléctrodos de modo a avaliar alguns dos parâmetros fisiológicos de um paciente tais como:

- ECG
- Batimento Cardíaco
- Frequência Respiratória
- Movimento (Acelerómetro)

A transmissão de dados neste dispositivo realiza-se através da Tecnologia *Bluetooth*.



Figura 37: BioHarness [21].

5.1.8 AMON

AMON [22] é um dispositivo capaz de monitorizar vários parâmetros fisiológicos e de gerar alertas quando necessário (Figura 38).

Este sistema foi desenvolvido com o objectivo de monitorizar pacientes em casa ou num hospital com elevado risco cardíaco e/ ou respiratório.

O dispositivo integra sensores de pressão sanguínea, temperatura, SpO_2 e ECG.

Inclui também um acelerómetro de 2 eixos que correlaciona a actividade do utilizador com os sinais vitais medidos.

O sistema incorpora um dispositivo GSM que pode transmitir e receber dados para troca de informação com o centro médico, onde os médicos analisam os dados recebidos.

O protótipo inclui *software* para processamento em tempo real e análise dos sinais vitais medidos.

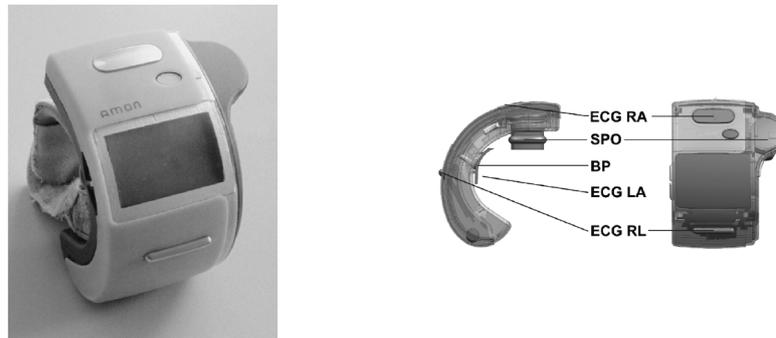


Figura 38: AMON [22].

Tabela 4: Características dos Sistemas de Aquisição

	<i>BITalino</i>	<i>eHealth</i>	<i>Shimmer</i> ¹	<i>BioHarness</i>	<i>AMON</i>
Protocolo de Comunicação	Bluetooth	Wi-Fi 3G GPRS Bluetooth 802.15.4 ZigBee	Bluetooth IEEE 802.15.4 - rádio TI CC2420	Bluetooth	GSM/UMTS
Alcance	<i>ND</i>	<i>ND</i>	<i>ND</i>	91.44 m-304.80 m	<i>ND</i>
Banda ISM	2.4 GHz	<i>ND</i>	2.4 GHz	2.4 GHz	<i>ND</i>
Microcontrolador	ATmega328P	ATmega328 ²	TI MSP430	<i>ND</i>	AT91R40807
Frequência de operação	8 MHz	16 MHz ²	8 MHz	<i>ND</i>	40 MHz
Memória	RAM: 2 KB ROM: 1 KB Flash: 32 KB	RAM: 2 KB ROM: 1 KB Flash: 32 KB	RAM: 16 KB ROM: 2 KB Flash: 256 KB	<i>ND</i>	RAM: 136 KB ROM: <i>ND</i> Flash: 4 MB
Resolução	10 bits	10 bits ²	12 bits	<i>ND</i>	<i>ND</i>
Fonte de energia	Bateria de Iões de Lítio	<i>ND</i>	Bateria de Iões de Lítio	Bateria de Iões de Lítio	Bateria de Iões de Lítio
	3.7 V	5 V	3.6 V	<i>ND</i>	1.8 V-3.6 V
	320 mAh	<i>ND</i>	450 mAh	<i>ND</i>	1.25 Ah
Velocidade de transmissão	<i>ND</i>	<i>ND</i>	250 kbps (rádio) 721 kbps - 2.0 Mbps (Bluetooth)	<i>ND</i>	<i>ND</i>
Dimensões	105 mm x 60 mm	<i>ND</i>	65 mm x 32 mm x 12 mm	<i>ND</i>	<i>ND</i>
Sensores	EMG ECG EDA ACC LUX	Pulso SpO ₂ <i>Airflow</i> Temperatura corporal ECG Glucómetro GSR Pressão Sanguínea Acelerómetro EMG	Acelerómetros Giroscópio Sensor de Pressão ECG EMG	Ritmo Cardíaco Ritmo respiratório Acelerómetro ECG	Pressão Sanguínea SpO ₂ ECG Acelerómetro Ritmo cardíaco Temperatura da pele
Peso	30 g	<i>ND</i>	31 g	<i>ND</i>	286 g
Frequência de amostragem	1 Hz, 10 Hz, 100 Hz ou 1000 Hz	<i>ND</i>	125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz ou 8000 Hz	<i>ND</i>	<i>ND</i>

1 Shimmer3 + Shimmer EXG

2 Arduino Uno Rev. 3

5.2 SISTEMAS DE MONITORIZAÇÃO REMOTA

5.2.1 CodeBlue

CodeBlue [50, 51] é um projecto que foi desenvolvido por elementos da Universidade de Harvard e da Universidade de Boston em colaboração com o *Boston Medical Center*, com o *Brigham and Women's Hospital*, com o *Spaulding Rehabilitation Hospital* e com a *Johns Hopkins University*. O CodeBlue consiste numa plataforma de *hardware* e de *software* para redes de sensores médicos, oferecendo um conjunto de protocolos para detecção de dispositivos (*device discovery*), um esquema de encaminhamento multi-salto baseado nos métodos de publicação/subscrição e uma *interface* que permite aos prestadores de cuidados de saúde obter informação de certos grupos de pacientes a partir da realização de *queries* ao sistema. (Figura 39)

Para além de permitir a monitorização em tempo real dos sinais vitais de cada paciente, o CodeBlue possui também um sistema de localização baseado em radiofrequência designado de MoteTrack com o objectivo de fornecer uma localização precisa e exacta dos pacientes e dos prestadores de cuidados médicos, o que se revela útil no caso de hospitais de grandes dimensões.

A versão actual deste sistema integra um oxímetro de pulso, um sensor de *ECG* de 2 derivações, um acelerómetro tri-axial, um giroscópio uni-axial e um sensor de *EMG*.

Na primeira versão deste sistema os sensores de *ECG* eram alimentados por *motest* Mica2/MicaZ ou Telos. Estes *motest* que utilizavam pilhas alcalinas AA foram substituídos por um *mote* de dimensões mais reduzidas desenvolvido no âmbito deste projecto, o *mote* Pluto, que se baseia no *mote* Tmote Sky e incorpora um microprocessador TI MSP430 e um rádio ChipCon CC2420.

Os sensores de *ECG* desenvolvidos incorporam o amplificador de instrumentação CMOS INA321 da Texas Instruments que amplifica o sinal diferencial por um factor de 5 e filtra uma grande parte do ruído de modo comum. As versões mais recentes do sistema incluem também um conjunto de amplificadores operacionais que reduzem a quantidade de ruído presente no sinal e actuam como um filtro passa-baixo, ao mesmo tempo que realizam uma maior amplificação do sinal.

Um filtro passa-alto de *feedback* corrige dinamicamente as mudanças *DC* que possam ocorrer ao longo do tempo.

O sinal de *ECG* já filtrado é encaminhado para um *ADC* onde é amostrado a uma frequência configurável (tipicamente de 120 Hz).

O acelerómetro (LIS3Lo2AQ) pode adquirir em cada um dos 3 eixos entre -2G e +2G ou entre -6G e +6G e amostra o sinal a 100 Hz.

A versão actual do sistema inclui uma bateria de lítio com uma capacidade de 120 mAh.

O sistema apresenta uma autonomia de cerca de 5 horas.

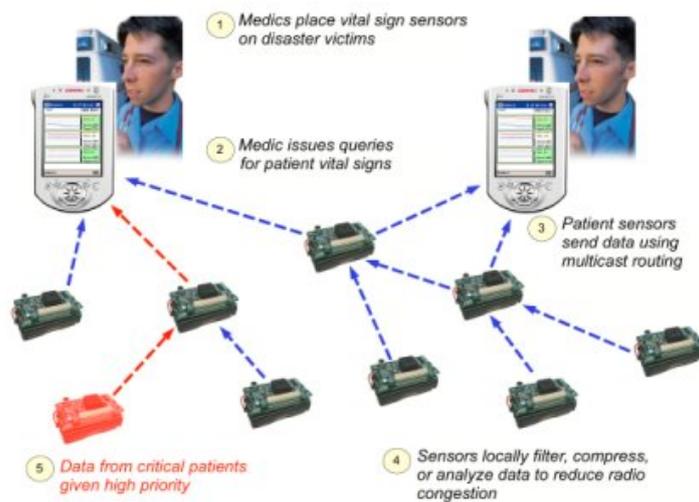


Figura 39: CodeBlue [23].

5.2.2 MobiHealth

O projecto Mobihealth [52, 53] foi desenvolvido pela Universidade de Twente em parceria com a Ericsson GmbH, tendo iniciado em Maio de 2002. Contempla a criação de um sistema de monitorização de sinais vitais baseado numa rede de área corporal (Body Area Network, BAN). Permite também a sua transmissão em tempo real aos prestadores de cuidados de saúde através de redes públicas sem fios.

Os sensores são responsáveis pelo processo de aquisição de dados conduzindo a que os sinais relativos ao movimento dos pacientes, actividade muscular, pressão sanguínea ou ECG sejam primeiro convertidos em sinais eléctricos, sendo posteriormente amplificados, acondicionados, digitados e comunicados dentro da BAN.

A *Mobile Base Unit, MBU* é o principal componente da BAN, actuando como uma *gateway* que agrega as medições dos sinais vitais (comunicação intra-BAN baseada em redes sem fios como o Bluetooth e o Zigbee) e as transmite ao resto do sistema (comunicação extra-BAN baseada em GPRS e UMTS). Os sinais medidos são transmitidos ao centro de cuidados médicos onde são consultados em tempo real por uma equipa médica.

5.2.3 Sensium

O projecto Sensium [54] foi desenvolvido pela Queen Mary University London e foi o primeiro System-on-chip concebido especificamente para a monitorização de sinais vitais (Figura 40).

O *chip* inclui um transdutor de baixo consumo, um protocolo *MAC*, um microprocessador digital 8051, periféricos de input/output, memória, um conversor analógico-digital e algumas interfaces de sensores padrão.

Este sistema consome 2.5 mA de uma fonte de 1 V quando activo e recebe ou transmite informação a uma distância superior a 3 m, consumindo adicionalmente 0.5 mA quando o microprocessador se encontra activo e o seu relógio interno se encontra sincronizado a 1 MHz.

Este *System-on-chip* tem sido utilizado como uma plataforma de rede de sensores corporais wireless juntamente com um monitor de dimensões reduzidas (o Life Pebble (Figura 41)) que monitoriza continuamente o ECG, o ritmo cardíaco, a actividade física (acelerómetro triaxial) e a temperatura da pele.

A antena do Life Pebble permite que os sinais vitais sejam transmitidos a uma frequência de 915 MHz e a uma distância de entre 5 a 10 metros.

Para minimizar a potência de processamento, foram implementados filtros de Butterworth e IIR.

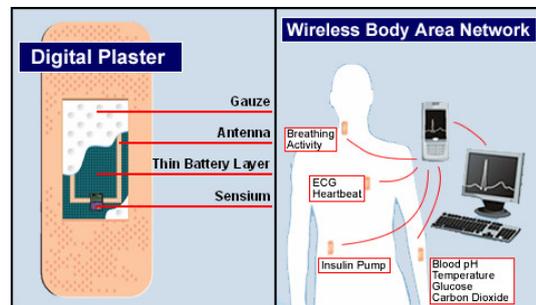


Figura 40: Sensium [24].



Figura 41: LifePebble [25].

5.2.4 *Vital-Jacket*

O projecto *Vital-Jacket* [55] foi desenvolvido pelo Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e é um sistema que permite monitorizar os sinais vitais dos pacientes na Cardiologia e no Desporto, através de um casaco e de uma *T-shirt* que integram os vários sensores dos sinais que se pretende analisar (Figura 42).

Este sistema permite a aquisição dos sinais de ECG, temperatura, respiração, movimento/queda, postura e saturação de oxigénio para além de algumas variáveis psico-sociais como o pânico, a entrega de medicação, os hábitos de actividade física e a localização do paciente por exemplo.

O sistema possui um ECG de 1, 3 ou 5 derivações e um acelerómetro triaxial.

A informação é transmitida em tempo real para um *PDA*, para um computador pessoal ou alternativamente para um sistema de informação de cardiologia através das redes móveis *wireless LAN*, *GPRS* ou *UMTS* e simultaneamente recolhida para um cartão de memória *SD* que pode ser utilizado para armazenamento offline.



Figura 42: *Vital-Jacket* [26] e [27].

5.2.5 *HM4All*

O projecto *HM4All* [56] foi desenvolvido pela Universidade do Minho em parceria com o Instituto Politécnico do Cavado e do Ave. Consiste de um Sistema de Monitorização de Sinais Vitais que tem por base redes *Zigbee* espacialmente distribuídas e que foi aplicado à ala de internamento de um hospital privado Português (Figura 43).

O *HM4All* (*Health Monitor for All*) baseia-se nas tecnologias de redes de sensores sem fios, *Wireless Sensor Network* (*WSN*) e protocolos base e permite a monitorização remota dos sinais de *ECG*, *SpO2* e temperatura da pele.

Os pacientes que se encontram em reabilitação no seu domicílio também podem ser monitorizados por prestadores de saúde especializados no hospital através de *PDA*s.

Os sensores *Zigbee* enviam de modo contínuo a informação relativa aos sinais vitais de cada paciente para um dos coordenadores de rede através de uma ou mais redes de sensores sem fios aplicadas à ala hospitalar de uma Unidade de Saúde.

Os dados são enviados através de uma interface em série para a *Gateway Zigbee-to-WiFi*, onde são processados e enviados, através da infra-estrutura *WiFi* para o Servidor de Dados.

Os sinais vitais dos pacientes podem ser apresentados em qualquer browser que se encontre a funcionar em qualquer uma das estações de monitorização.

Neste sistema todos os sensores têm por base o módulo *ZigBee JN5139-M01* da *Jennic*.

Este sistema inclui um sensor de *ECG* que se baseia no amplificador de instrumentação *INA118* da *Texas Instruments*.

Neste caso, são utilizados filtros analógicos para remover as componentes fora da banda 0.5 Hz-40 Hz e um filtro de *Notch* analógico adicional para reduzir o ruído que surge associado aos 50 Hz da rede de alimentação.

Um conversor analógico digital de 12 *bits* é utilizado para amostrar o sinal a 200 Hz.

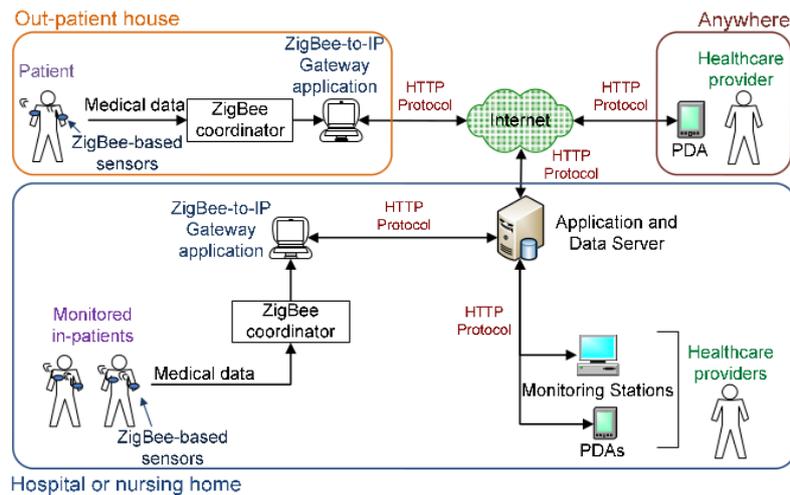


Figura 43: HM4All [28].

5.2.6 Lifeguard

O projecto Lifeguard[57, 58] desenvolvido pela NASA é um sistema de monitorização de parâmetros fisiológicos inicialmente concebido para monitorizar o estado de saúde dos astronautas durante as suas missões espaciais 44.

Consiste de uma unidade de pequenas dimensões designada de *Crew Physiological Observation Device (CPOD)* responsável pela recolha e armazenamento ou transmissão via tecnologias sem fios da informação recolhida.

Os biosensores registam os sinais fisiológicos de ECG, ritmo cardíaco, pressão sanguínea e temperatura para cada utilizador.

O sistema também monitoriza a saturação de oxigénio no sangue (SpO_2) e regista a actividade do paciente através dos acelerómetros.

O CPOD pode guardar os parâmetros fisiológicos até 9 horas numa memória flash de 32 MB. Permite também efectuar o download de dados gravados ou em tempo real para um computador através de uma ligação *standard RS232*.

Alternativamente, o CPOD pode enviar informação em tempo real através das tecnologias de *Bluetooth* ou de *FSK* de 916 MHz para outro dispositivo.

A configuração deste dispositivo não permite acessos múltiplos.

O sistema apresenta uma autonomia em modo de operação contínuo de 9 horas.

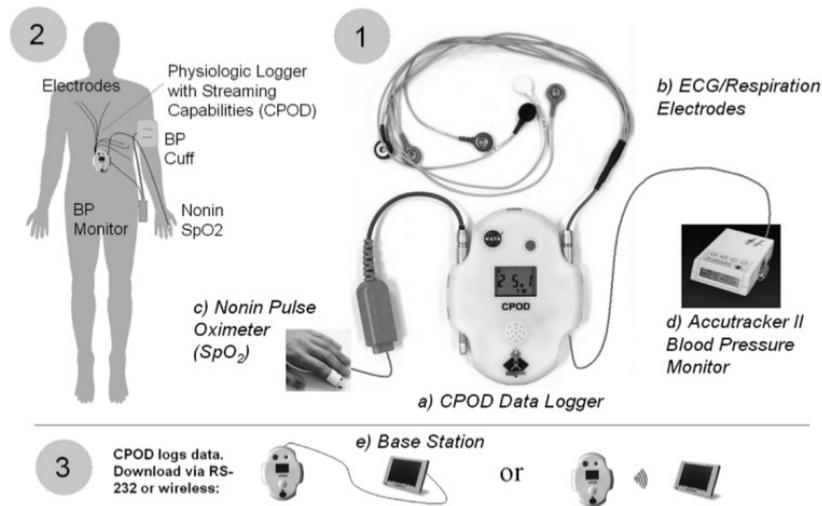


Figura 44: Lifeguard [19].

5.2.7 UbiMon

O projecto UbiMon[59, 60] foi desenvolvido pelo Imperial College London. Este projecto pretende viabilizar a gestão contínua dos pacientes sob as suas condições fisiológicas naturais de modo a que a ocorrência de certos artefactos esporádicos que ameaçam a vida do paciente possam ser detectados.

O projecto de monitorização ubíqua para sensores usáveis e implantáveis (*UbiMon*) do *Imperial College of London* visa fornecer um sistema de monitorização contínuo e não invasivo para pacientes de modo a capturar eventos transitórios mas fatais.

A estrutura básica do *UbiMon* baseia-se na concepção de uma rede de sensores corporais (*BSN*).

O sistema consiste de 5 componentes principais, nomeadamente os nodos *BSN*, a unidade de processamento local, o servidor central, a base de dados dos pacientes e a estação de visualização.

Com a actual estrutura *UbiMon*, um número de biossensores sem fios incluindo faixas de ECG de 3 ou 2 derivações e sensores de saturação do oxigénio (SpO_2) foi desenvolvido.

Para facilitar a incorporação da informação do contexto, o estado de alerta devido ao contexto baseado em sensores, incluindo acelerómetros, e sensores de temperatura e condutância da pele são também integrados no nodo *BSN*. Para além disso, um cartão *flash BSN* compacto é desenvolvido para *PDA*s, em que os sinais são recolhidos, apresentados e analisados pelo *PDA* [61].

5.2.8 AlarmNet

AlarmNet [29] é uma rede de sensores sem fios que permite a monitorização em residências de apoio a pessoas com um determinado grau de incapacidade.

O sistema integra sensores de infravermelhos para detecção de movimento e de poeiras e sensores de temperatura, luminosidade, acelerometria, *ECG*, ritmo cardíaco e SpO_2 .

Estes sensores baseiam-se na plataforma de *hardware* *MICAz*.

O sistema possui esquemas e protocolos de segurança que asseguram a privacidade dos pacientes ou residentes.

Permite a recolha de dados para análise em tempo real por um profissional de saúde.

Na Figura 45 apresenta-se a arquitectura e a topologia do sistema (à esquerda) e um exemplo de aplicação (à direita).

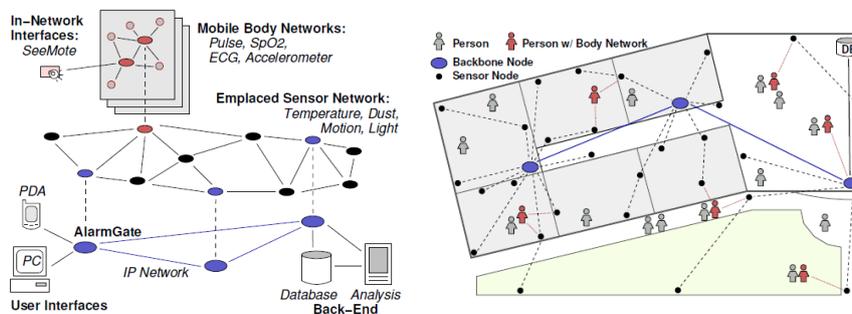


Figura 45: AlarmNet [29].

5.2.9 Aid-N

Este sistema não é para ser implementado em casa, no entanto a sua arquitectura é similar a este tipo.

É um sistema com o objectivo de realizar triagem e monitorização de vítimas de acidentes, ou catástrofes, de modo a não haver necessidade de um médico fazer uma triagem exaustiva a cada paciente, podendo assim focar os seus esforços nos feridos mais graves.

A arquitectura do sistema consiste em sensores de tensão arterial e *EKG*, que são colocados nos pacientes e ligados a um dispositivo que transmite a informação recebida para um computador local que estará a ser supervisionado por um profissional de saúde.

Na Tabela 5 apresentam-se as características dos sistemas de monitorização acima referidos. Esta tabela foi realizada com base em [62], [49], [15], [50], [63], [56], [64], [54], [57], [65], [59], [60].

Tabela 5: Características de sistemas de monitorização.

	<i>CodeBlue</i>	<i>HM4All</i>	<i>Sensium</i>	<i>Lifeguard</i>	<i>UbiMon</i>
Protocolo de Comunicação	IEEE 802.15.4 - Chipcon CC2420	IEEE 802.15.4/ZigBee	Bluetooth ZigBee	Bluetooth RS-232	Bluetooth/ Wi-Fi 3G/ GPRS IEEE 802.15.4/ ZigBee
	2.4 GHz	2.4 GHz	915 MHz	ND	2.4 GHz
Microcontrolador	TI MSP430F1611	JN5139	8051-eWarp	PIC16LF877	TI MSP430
Frequência de operação	8 MHz	16 MHz	1 MHz	7.3728 MHz	ND
Memória	RAM: 10 KB ROM: ND Flash: 48 KB + 256 B	RAM: 96 KB ROM: 192 KB Flash: ND	RAM: 64 KB ROM: ND Flash:	ND ND 32 MB	ND ROM: 4 KB Flash: 128 KB
Resolução do sistema	12 bits	12 bits	10 bits	12 bits	10 bits
Fonte de energia	Bateria de Iões de Lítio	Bateria de Iões de Lítio	Bateria flexível impressa	2 pilhas AAA	Pilha de Relógio
	3.6 V	3.6 V	1 V	3 V	3.3 V
	120 mAh	ND	ND	ND	ND
Velocidade de transmissão	250 kbps	250 kbps	ND	ND	250 kbps
Dimensões	57 mm x 36 mm x 16 mm	18 mm x 30 mm	ND	129 mm x 100 mm x 20 mm	25 mm x 6 mm
Sensores	Acelerómetro	ECG SpO2 Ritmo cardíaco	ECG Ritmo Cardíaco Acelerómetro Temperatura da pele Respiração	ECG Ritmo Respiratório SpO2 Pressão Sanguínea Temperatura da Pele Ritmo Cardíaco	ECG SpO2 acelerómetros sensores de temperatura Ritmo Cardíaco
Peso	30.5 g	ND	ND	ND	ND

Tabela 6: Características principais dos sensores de ECG e de Acelerometria de cada sistema

	Frequência de amostragem		Gama de variação	
	Sensor de ECG	Acelerómetro	Sensor de ECG	Acelerómetro
BITalino	1000Hz	100 Hz ou 1000 Hz	ND	+/- 3g
CodeBlue	Configurável (120 Hz)	100 Hz	ND	2g/6g
HM4All	200 Hz	ND	ND	ND
Sensium	ND	ND	1-5 mV	ND
Lifeguard	256 Hz	ND	+/- 12 mV	+/- 10g
Vital Jacket	500 Hz	ND	ND	ND
Shimmer	ND	ND	ND	+/- 1.5g-6g

5.3 ANÁLISE DAS CARACTERÍSTICAS DOS DIFERENTES SISTEMAS

As tabelas 4 e 5 colocam em evidência as características dos sistemas de aquisição e de monitorização sujeitos a análise.

Desta análise constata-se que a nível dos protocolos de comunicação, os mais utilizados são o *Bluetooth* e o GSM/GPRS e UMTS(3G).

Utiliza-se o protocolo *Bluetooth* para transmissão de dados a curta distância (comunicação entre os sensores e o dispositivo móvel).

Para comunicação de dados a longa distância (comunicação entre o dispositivo móvel e o servidor central)recorre-se aos protocolos GSM/GPRS e UMTS(3G).

A banda *ISM* tipicamente utilizada é de 2.4 GHz.

No caso dos sistemas analisados verifica-se que a frequência de operação geralmente utilizada pelos microcontroladores é de 8 MHz.

Os sistemas possuem uma resolução de 10 ou de 12 *bits*, permitindo representar um número máximo de $2^{10} = 1024$ ou $2^{12} = 4096$ valores respectivamente de cada vez que os sinais são amostrados.

A fonte de energia predominantemente utilizada, na maior parte dos sistemas é a bateria de iões de lítio.

Este tipo de bateria permite armazenar o dobro de uma bateria de hidreto metálico de níquel (*NiMH*) e três vezes mais que uma bateria de níquel cádmio (*NiCd*).

Neste caso, não é necessário carregar a bateria até ao total da sua capacidade ou descarregá-la totalmente, ao contrário da bateria de *NiCd*, o que se assume como uma característica importante para o tipo de sistemas que se pretende implementar.

De acordo com a informação presente na tabela 6, no *BITalino* a frequência de amostragem pode assumir as frequências de 1, 10, 100 ou 1000 Hz. No projecto *CodeBlue* a frequência de amostragem do sen-

tor de *ECG* é de 120 Hz (configurável), sendo de 200 Hz no *HM4All*, de 256 Hz no *LifeGuard* e de 500 Hz no *VitalJacket*.

O acelerómetro é amostrado a 100 Hz no *CodeBlue*.

DESENVOLVIMENTO DO PROTÓTIPO DO SISTEMA

A Figura 46 apresenta a arquitectura geral do sistema que se pretendeu estudar e desenvolver. Nela podemos identificar o paciente, o módulo de aquisição, o dispositivo móvel detido pelo paciente, o servidor onde se encontra alojada a base de dados e a estação de visualização dos dados dos vários pacientes.

6.1 ARQUITECTURA DO SISTEMA

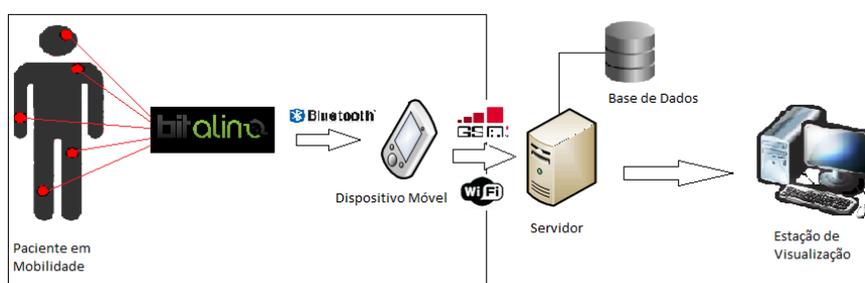


Figura 46: Arquitectura do Sistema.

Dos vários componentes da arquitectura presentes na Figura 46 foi utilizado o BITalino como dispositivo de aquisição, implementou-se a parte das comunicações entre o BITalino e o dispositivo móvel e foram gerados gráficos na aplicação móvel a partir da biblioteca *achartengine*. Foram também efectuados vários testes no sentido de implementar o módulo de processamento e de detecção de Anomalias Cardíacas e o módulo de *log* do sistema. Para efeitos da aquisição dos sinais de ECG foram utilizados alguns indivíduos do sexo feminino e do sexo masculino com idades compreendidas entre os 18 e os 60 anos de idade.

6.1.1 Contextualização

O paciente encontra-se no seu domicílio com o sistema de aquisição de biosinais BITalino ou similar ligado ao seu corpo por via dos 3 eléctrodos que realizam a interface entre o sensor de ECG e o próprio paciente.

No *smartphone* do paciente, encontra-se instalada uma aplicação que regista e monitoriza em tempo real os dados de ECG previamente adquiridos e enviados por *Bluetooth* para a aplicação móvel. Nesta aplicação móvel, o paciente numa primeira fase efectua um registo

onde coloca o seu nome completo, o seu número de utente, o nome da Instituição que o está a seguir, o nome de utilizador e a *password* que pretende definir para efectuar o *login* na aplicação.

Cada um destes dados é transmitido ao servidor através do método *POST* do próprio protocolo *https*. Este servidor possui uma aplicação com um *servlet* que gere os pedidos de *https* da aplicação cliente comunicando com a base de dados através da classe *PreparedStatement* da *API JDBC* da *Oracle*. A estação de visualização comunica com o servidor através de um *URL* que interage com a *servlet* do servidor por meio do método *GET* do protocolo *Https*.

Quando se efectua o *login* na aplicação móvel, são transmitidos para o servidor o nome de utilizador e a sua *password* devidamente encriptada (como a encriptação *AES* ou chaves assimétricas), encaminhados para a *servlet* existente no servidor, existindo no servidor um ficheiro *PHP* que verifica os dados obtidos pelo método *GET* no servidor com os dados contidos na base de dados.

6.1.2 Módulo de Aquisição

O Módulo de Aquisição seleccionado para a realização deste projecto foi a plataforma de aquisição de biosinais *BITalino* devido à sua modularidade, baixo custo e possibilidade de integrar outro tipo de sensores.

As principais características desta placa encontram-se mencionadas na Tabela 4.

Esta placa é composta por um módulo de processamento constituído pelo microprocessador e por um *ADC* (conversor analógico-digital) de aproximações sucessivas, um módulo de gestão da energia e um módulo de transmissão. O módulo de gestão de energia possui uma ligação para a bateria e inclui um regulador de tensão de 3.3 V. Este regulador de tensão faz com que a tensão à entrada do microprocessador assuma valores abaixo dos 3.3 V. O módulo de transmissão é constituído por um módulo *Bluetooth* de classe II que permite um alcance até 10 m.

Os sensores ligam-se à placa através de conexões do tipo *RJ22* em cada uma das portas analógicas e/ ou digitais.

Este *kit* de desenvolvimento permite o envio das leituras dos sensores por *Bluetooth* para um dispositivo móvel ou para um *PC* através de uma *stream* de dados.

Os dados dos sensores após a etapa/estágio de aquisição são enviados em pacotes de 8 *bits* através do módulo *Bluetooth* incluído no *BITalino*, sendo processados ao nível do *smartphone* e enviados para um servidor central através das tecnologias *3G(UMTS)*, *GPRS* e eventualmente *WiFi*.

Os dados primeiramente são divididos em pacotes de dados de 8 *bits* de acordo com o esquema de empacotamento de dados que se apresenta na Tabela 7.

		Bits							
		7	6	5	4	3	2	1	0
Bytes	7	S	S	S	S	CRC	CRC	CRC	CRC
	6	D0	D1	D2	D3	A0	A0	A0	A0
	5	A0	A0	A0	A0	A0	A0	A1	A1
	4	A1	A1	A1	A1	A1	A1	A1	A1
	3	A2	A2	A2	A2	A2	A2	A2	A2
	2	A2	A2	A3	A3	A3	A3	A3	A3
	1	A3	A3	A3	A3	A4	A4	A4	A4
	0	A4	A4	A5	A5	A5	A5	A5	A5

Tabela 7: Estrutura dos pacotes de dados do *BITalino*.

Na Figura 7, *S* corresponde a um identificador da sequência, *A* representa um canal analógico e *D* um canal digital.

Fazendo a leitura a partir do *Byte* 0:

- O primeiro *Byte* é constituído pelos 6 *bits* do canal analógico A5 e por 2 *bits* do canal analógico A4.
- O segundo *Byte* é formado pelos outros 4 *bits* do canal analógico A4 e por 4 *bits* do canal analógico A3.
- O terceiro *Byte* é composto pelos restantes 6 *bits* do canal analógico A3 e por 2 *bits* do canal analógico A2.
- O quarto *Byte* é constituído pelos restantes 8 *bits* do canal analógico A2.
- O quinto *Byte* corresponde aos 8 *bits* do canal analógico A1.
- O sexto *Byte* é formado pelos outros 2 *bits* do canal analógico A1 e por 8 *bits* do canal analógico A0.
- O sétimo *Byte* é constituído pelos restantes 4 *bits* do canal analógico A0 e por 4 *bits* relativos aos 4 canais digitais D0, D1, D2 e D3.
- O oitavo e último *Byte* inclui 4 *bits* do CRC e 4 *bits* que correspondem a um identificador de cada sequência.

Posteriormente, estes pacotes de dados são enviados via *Bluetooth* para o módulo de processamento onde é realizado um pré-processamento dos dados enviados.

Em termos práticos, o módulo de processamento encontra-se integrado na aplicação móvel a desenvolver sob a forma de um serviço.

6.1.2.1 API do BITalino

O *BITalino* disponibiliza aos seus utilizadores um vasto conjunto de *Application Programming Interfaces (API)* para desenvolvimento de aplicações em linguagens de programação como o *Python, Java, Lab-View, c#* e *Matlab*.

A *API* seleccionada para desenvolvimento da solução proposta foi a *API* desenvolvida em *Java*.

Esta *API* possui 6 classes: *BITalinoDevice, BITalinoErrorTypes, BITalinoException, BITalinoFrame, BITalinoFrameDecoder* e *BITalinoSocket*.

A primeira classe, *BITalinoDevice*, representa um dispositivo *BITalino* e fornece métodos para interagir com ele.

Esta classe possui apenas um construtor que admite como argumentos a frequência de amostragem e o conjunto dos canais analógicos passíveis de leitura.

Este construtor começa por validar a frequência de amostragem, que pode assumir apenas os valores 1, 10, 100, ou 1000 Hz. Este último valor corresponde ao valor por defeito. Este construtor verifica se o número de canais e os respectivos identificadores foram correctamente especificados. No caso de os canais analógicos não serem válidos é criada uma excepção do tipo *BITalinoException*.

Por fim, determina o número total de *Bytes* a serem lidos com base no número de canais analógicos de 10 *bits* e de 6 *bits* sujeitos a aquisição.

Esta classe fornece métodos para abrir as *Streams* de dados relacionadas com a conexão *Bluetooth*. Permite também definir a frequência de amostragem no dispositivo *Bluetooth* (método *open(final InputStream is, final OutputStream os)*). Através desta classe é possível efectuar a leitura dos canais analógicos predefinidos (método *start()*), parar a aquisição no *BITalino* (método *stop()*), encerrar a conexão *Bluetooth* e libertar alguns recursos disponíveis (método *close()*). Esta classe possibilita também a consulta da versão do dispositivo (método *version()*) e a leitura dos dados a partir do *Bluetooth* (*read(final int numberOfSamples)*).

A classe *BITalinoErrorTypes*, enumera os vários tipos de erros definidos para o *BITalino* criando para o efeito um *HashMap*, em que se faz corresponder a descrição de um determinado erro ao código de erro correspondente. Esta classe possui um construtor que permite instanciar um elemento desta classe a partir dos argumentos valor e descrição do erro.

Pertencem a esta classe os métodos *getValue()*, *getDescription()* e *fromValue(final int value)*.

O método *getValue()* retorna o código do erro instanciado. *getDescription()* permite obter a descrição do mesmo erro. O método *fromVa-*

lue(final int value) permite obter a descrição do erro a partir do código do erro introduzido como argumento.

A classe *BITalinoException* possui um único construtor que permite instanciar uma exceção no *BITalino* através de um argumento do tipo *BITalinoErrorTypes*.

A partir do método *getCode()* é possível retornar o código associado à exceção. A classe *BITalinoSocket* fornece um construtor que permite instanciar elementos desta classe introduzindo como argumentos uma *DataInputStream* e uma *OutputStream*.

Os argumentos passados a este construtor não podem ser nulos senão é gerada uma mensagem de erro.

Esta classe possui os métodos de leitura e escrita *read(final int[] analogChannels, final int totalBytes, final int numberOfSamples)* e o método *write(final int data)*. O primeiro método lê os dados a partir do *Bluetooth* tendo por base o conjunto de canais analógicos, o número de *Bytes* total e o número de amostras. O segundo método permite escrever dados para a *Stream* de dados do *Bluetooth*. Possui os métodos para obtenção das *streams* de dados *getInputStream()* que devolve a *InputStream* e *getOutputStream()* que retorna a *OutputStream*. Possui também o método *close()* que permite encerrar a conexão e libertar alguns recursos disponíveis,

A classe *BITalinoFrame*, uma vez instanciado um *BITalinoFrame*, possui um conjunto de vários *getters* e *setters* e implementa os métodos *equals(Object obj)* e *toString()*.

Os *getters* são constituídos por um método que retorna o *Cyclic Redundancy Check (CRC)* (*getCRC()*), um método que retorna a sequência (*getSequence()*) e um método que retorna o valor de um canal analógico ou digital numa dada posição (*getAnalog(final int pos)* e *getDigital(final int pos)*, respectivamente).

Os *setters* definem um conjunto de métodos para atribuir um valor ao *CRC* (*setCRC(int CRC)*), para definir a sequência (*setSequence(int seq)*) e para atribuir um determinado valor a um canal analógico ou digital que se encontre numa dada posição (*setAnalog(final int pos, final int value)* e *setDigital(final int pos, final int value)*, respectivamente). O método *equals(Object obj)* verifica se 2 objectos do tipo *BITalinoFrame* são iguais. O método *toString()* permite converter o *CRC*, a sequência e o conjunto de canais analógicos e digitais numa *String*.

Esta *API* não permite definir frequências para canais independentes, sendo possível configurar as frequências de amostragem dentro da mesma apenas para uma frequência de 1, 10, 100 ou 1000 Hz.

6.1.3 Módulo de Processamento

Posteriormente à recepção dos dados recolhidos a partir dos sensores fisiológicos, o *smartphone* através de uma aplicação móvel efectua um pré-processamento dos dados para filtrar o ruído que se encontra

presente no sinal, realiza a detecção dos complexos *QRS* e determina a frequência cardíaca por contagem do número de complexos *QRS*. Este pré-processamento é utilizado juntamente com uma heurística para determinar algumas arritmias mais frequentes entre pacientes com doenças cardiovasculares. Algumas das arritmias mais frequentes neste tipo de pacientes são a Bradicardia, a Taquicardia, a Fibrilhação Auricular (*AV*) e o *Flutter* Atrial (*AF*). Este módulo é constituído pelos submódulos de Remoção da Flutuação de Linha de Base, Remoção da Interferência de Rede Eléctrica, Remoção do Ruído de *EMG*, *Smoothing*, Detecção da onda *P*, Complexo *QRS*, onda *T* e determinação dos Intervalos *PR* e *ST*, Detecção do Ritmo Cardíaco e Detecção de Anomalias Cardíacas.

6.1.4 *Módulo de compressão de dados*

Depois de os dados serem analisados pela aplicação, os sinais são comprimidos e armazenados num cartão *SD*.

O *smartphone* possui recursos limitados, pelo que se torna necessário comprimir esses mesmos dados.

Este processo de compactação do sinal original permite reduzir a banda necessária para transmitir o sinal e o consumo de energia do telemóvel durante esse mesmo período.

6.1.5 *Módulo de transmissão de dados*

A aplicação móvel, depois de analisar os dados, procede à transmissão desses dados para um servidor central. Nesse servidor, a informação dos vários pacientes que são monitorizados no domicílio é mantida sob a forma de registos numa base de dados *SQL* ou *mySQL*.

Este módulo comunica com o servidor central através dos protocolos de transmissão *GSM/GPRS* ou *UMTS (3G)*.

6.1.6 *Módulo de log*

No caso da detecção de alguma anomalia cardíaca, é efectuado um registo no *log* do sistema com os vários parâmetros do sinal de *ECG* juntamente com uma *timestamp*.

Os parâmetros de *ECG* a incluir no *log* da aplicação incluem o intervalo *PR*, o intervalo *QT*, o intervalo *ST* e a amplitude das ondas *P*, *R* e *S*, assim como as durações da onda *P*, *T* e do complexo *QRS*.

A *timestamp* possui a data no formato *dd-mm-yyyy* e a hora no formato *HH:MM:SS*.

6.1.7 Servidor Central

No servidor central é alocada uma base de dados SQL ou MySQL com o registo de potenciais eventos cardíacos por paciente.

A base de dados contém em cada tuplo o número de utente, parâmetros importantes da onda de ECG e um possível diagnóstico efectuado pela aplicação. Os parâmetros de ECG a incluir são o intervalo PR, a amplitude da onda R e as durações da onda P e do complexo QRS. Cada tuplo da base de dados contém também o sinal comprimido, um *score* que tem a ver com a existência de valores atípicos para esses parâmetros e o número da equipa médica que foi atribuída ao paciente.

A base de dados pode conter uma tabela que relaciona o número da equipa médica ao número de telemóvel do responsável por essa equipa.

O servidor central envia a informação de cada paciente a partir da WLAN (IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n ou IEEE 802.11ac) para a estação de visualização, onde se encontra um grupo de médicos para analisar e corrigir a informação oriunda da aplicação no que diz respeito aos diagnósticos dos vários pacientes.

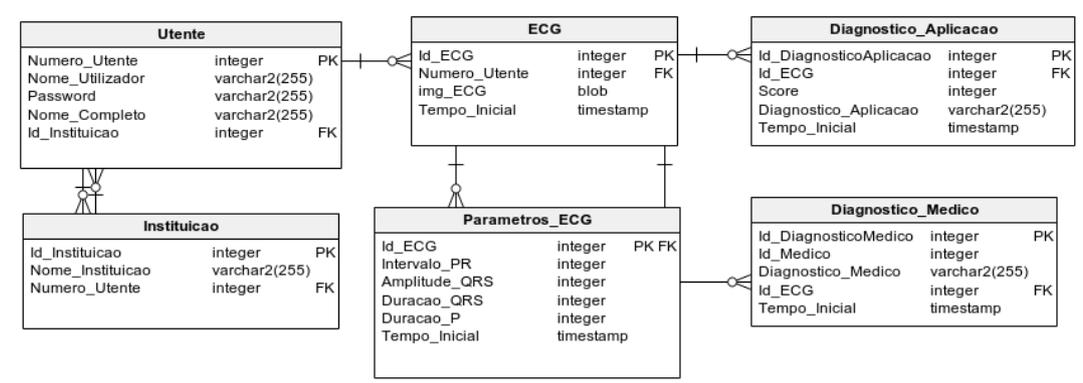


Figura 47: Esquema Relacional da Base de Dados.

A Figura 47 representa o esquema relacional da base de dados a alojar no servidor. Este esquema relacional inclui uma relação do utente, da sua instituição de saúde, do sinal de ECG e dos seus parâmetros. Inclui também uma relação com os dados do diagnóstico da aplicação e outra com os dados do diagnóstico efectuado pelo médico. Na relação do utente consta o seu número de utente, o nome de utilizador, a password, o nome completo e um número de identificação da instituição. O número de utente e o número de identificação da instituição são do tipo inteiro, ao passo que o nome de utilizador, a password e o seu nome completo são do tipo *varchar2*. Esta relação surge associada à relação instituição através das chaves estrangeiras. A relação Instituição possui um identificador da instituição, o nome

da mesma e o número de Utente. A relação *ECG* associa-se à relação Utente através da chave estrangeira e inclui o identificador do sinal de *ECG*, o número de utente, uma imagem do sinal de *ECG* para ser analisada pela equipa médica, e o tempo inicial da aquisição desse tipo de sinal. A relação *Parametros_ECG* surge associada à relação *ECG* a partir da chave estrangeira e inclui a duração do intervalo *PR*, a amplitude *QRS*, a duração do complexo *QRS*, a duração da onda *P* e o tempo inicial de aquisição deste tipo de sinais. A relação *ECG* por sua vez relaciona-se com as relações *Diagnostico_Aplicacao* e *Diagnostico_Medico* que correspondem ao diagnóstico da aplicação e ao diagnóstico do médico.

6.1.8 Estação de Visualização

A partir da Estação de Visualização é possível um médico ou um grupo de médicos visualizar os dados de vários pacientes.

Esta inclui uma aplicação *Web* que realiza uma primeira triagem de acordo com o *score* atribuído a cada paciente e transmitido pelo servidor central, sendo possível aos profissionais de saúde tratarem primeiro dos casos mais urgentes.

A informação é extraída da base de dados a partir da Linguagem de Programação *JSON* ou *JavaScript*, sendo a informação descomprimida e gerado um gráfico com os dados do sinal do paciente.

Os médicos ou profissionais de saúde devem poder seleccionar os pacientes que pretendem seguir e os parâmetros que consideram mais importantes para que a triagem seja adequada a cada profissional de saúde.

O médico devidamente identificado perante a aplicação *Web* efectua o seu próprio diagnóstico que pode ser coincidente ou não com o diagnóstico efectuado primeiramente pela aplicação móvel.

Este último diagnóstico é enviado para a aplicação móvel, sendo o *Log* do sistema actualizado de acordo com o parecer do médico e com a coincidência ou não do diagnóstico efectuado pela aplicação com o diagnóstico efectuado pelo médico.

Em qualquer tipo de situação desde condições não críticas a urgentes os pacientes encontram-se organizados no sistema por um sistema de cores que compreende as cores amarelo e vermelho (código de *Manchester*).

Nos casos mais urgentes, o servidor central para além de emitir *beeps* na estação de visualização também envia uma notificação sob a forma de mensagem para o telemóvel, smartphone ou PDA do responsável pela equipa médica atribuída a um determinado paciente. Podem ser também enviadas notificações sob a forma de mensagens de texto com a geolocalização do paciente no caso de ser estritamente necessário.

Nesta secção foi definida a arquitectura de um sistema de monitorização de *ECG* de pacientes em mobilidade.

Foi utilizado o dispositivo de biosinais *BITalino* como módulo de aquisição e fez-se a comunicação desse dispositivo de aquisição com o dispositivo móvel através da *API* disponibilizada.

Os dados obtidos a partir do canal de *ECG* foram utilizados na geração de gráficos na aplicação móvel através da biblioteca *achartengine*.

6.2 PLATAFORMAS DE TESTE E DE DADOS BIOMÉTRICOS

Nesta secção é efectuada uma breve descrição dos *Datasets* utilizados, assim como uma comparação dos seus formatos com um formato de ficheiros conhecido (o formato *EDF*), juntamente com uma caracterização da *API* do *BITalino* e uma descrição das plataformas de teste (*MATLAB*) e desenvolvimento (*Android*) da aplicação móvel.

6.2.1 Plataformas de teste

Para realizar o teste e a validação do sistema foram utilizadas duas plataformas de teste: o *MATLAB* e o sistema de desenvolvimento *Android*.

O *MATLAB* é um ambiente multi-paradigma de computação numérica e uma linguagem de programação que permite a manipulação de matrizes e inclui a possibilidade de gerar gráficos a partir de funções e de dados no geral, implementar algoritmos, criar interfaces de utilizador e realizar a interface com programas escritos noutras linguagens de programação como o *C*, o *C++*, o *Java* e o *Fortran*.

Devido ao facto de possuir várias bibliotecas que integram os comandos necessários ao desenvolvimento de aplicações é fácil usar esta ferramenta para uma prototipagem rápida das mesmas.

O sistema *Android* é um sistema operativo móvel baseado num *kernel Linux* a um nível mais baixo e por uma máquina virtual *Java*, a nível superior.

Actualmente existem múltiplas plataformas de desenvolvimento de aplicações para dispositivos móveis como a plataforma *Android*, o *Apple iOS*, o *Blackberry*, o *Symbian* e a plataforma *Windows (Windows Mobile ou Windows Phone)* [66].

Devido ao facto de a plataforma *Android* ser o sistema mais utilizado no âmbito do desenvolvimento de aplicações móveis, e ser um sistema relativamente versátil, capaz de correr num grande número de plataformas de *hardware*, recorreu-se a esta plataforma para desenvolver a aplicação móvel.

A Linguagem de Programação mais comumente utilizada no ambiente de desenvolvimento *Android* é a Linguagem de Programação *Java*, sendo os programas escritos em *Java* compilados em *bytecode*

para a máquina virtual *Java* e posteriormente traduzidos para *Bytecode Dalvik* e armazenados em *.dex* (Dalvik EXecutable) e *.odex* (Optimized Dalvik EXecutable).

A linguagem *Java* é uma linguagem de programação orientada a objectos, tendo sido criada pela *Sun Microsystems* em 1991.

O *kernel Linux* presente na plataforma *Android* permite que as aplicações desenvolvidas para este tipo de plataformas tenham uma grande liberdade a funcionalidades da mesma.

Os dispositivos integram sensores de *GPS*, acelerómetro e giroscópio e suportam comunicação por *bluetooth*, *GSM*, *3G (UMTS)*, *CDMA* e *Wi-Fi*.

Em comparação com os computadores *desktop*, servidores ou portáteis, os dispositivos móveis apresentam recursos de memória e de armazenamento limitados, baixo poder de processamento e uma baixa resolução, para além de recursos de energia limitados [66].

6.2.2 Plataformas de dados biométricos

Para a realização deste projecto recorreu-se a dois *Datasets*: um orientado para a filtragem do ruído presente no sinal original (*Dataset* proveniente da base de dados *ECG-ID Database* do *PhysioNet* [35]) e outro orientado para a detecção de arritmias (*Dataset* proveniente da base de dados *MIT-BIH Arrhythmia Database*) do *PhysioNet* ([35]).

O primeiro *Dataset* a ser utilizado consistia de um conjunto de 310 registos de *ECG* que foram obtidos de uma amostra de 90 pessoas.

Cada um desses registos inclui:

- O sinal da derivação I do *ECG*, definido para um período de 20 s, amostrado com uma frequência de 500 Hz e com uma resolução de 12 *bits* ao longo de um intervalo nominal de +- 10 mV.
- 10 pulsos anotados (Anotações dos picos das ondas *T* e *R* efectuadas a partir de um detector automático).
- Informação contendo a idade, o género e a data de registo.

Os registos foram obtidos a partir de voluntários (44 homens e 46 mulheres com idades compreendidas entre os 13 e os 75 anos de idade).

Os sinais de *ECG* não filtrados apresentam ruído e contêm componentes de ruído de baixa e de alta frequência.

Cada registo inclui os sinais filtrados e não filtrados:

- Sinal 0: Sinal da derivação I do *ECG* (sinal não filtrado).
- Sinal 1: Sinal da derivação I do *ECG* filtrado (sinal filtrado).

O segundo *Dataset* inclui 48 registos de *ECG* com aproximadamente 30 minutos cada um obtidos a partir de 47 indivíduos (25 homens com idades compreendidas entre 32 e 89 anos e 22 mulheres com idades compreendidas entre 23 e 89 anos) submetidos a estudo pelo laboratório de arritmias do Boston's Beth Israel Hospital (*BIH*) entre 1975 e 1979.

Vinte e três destes registos foram escolhidos aleatoriamente deste conjunto. Os outros 25 registos foram escolhidos deste mesmo conjunto com o objectivo de integrarem uma multitude de fenómenos raros mas relevantes em termos clínicos. As anomalias cardíacas relevantes do ponto de vista clínico correspondem a arritmias ventriculares, juncionais e supraventriculares complexas e anomalias na condução.

Estes registos foram amostrados a uma frequência de 360 Hz com uma resolução de 11 *bits* ao longo de um intervalo de 10 *mV*.

6.2.3 Comparação dos Formatos de Ficheiros dos Datasets com o formato EDF

Um formato de ficheiro consiste de um cabeçalho com metadados sobre o ficheiro e um conjunto de dados binários que definem o conjunto de dados a transmitir.

O *formato 16* e o *formato 212* [67] são os formatos mais utilizados pelas bases de dados do *PhysioNet* [35].

Todos os formatos utilizados pelas bases de dados do *Physionet* possuem um cabeçalho que contém no mínimo uma linha de registo que especifica o nome do registo, o número de segmentos e o número de sinais. Possui também uma linha de especificação do sinal para cada um dos sinais que contém o nome do ficheiro. Este nome de ficheiro contém as amostras e o formato utilizado [67].

O *formato 16* consiste na representação de cada amostra por 16 *bits* (2 *Bytes*) em notação de complemento para 2, efectuando-se o registo dos *bytes* no ficheiro de dados do *Byte* menos significativo para o mais significativo.

Os *bits* mais próximos do *bit* mais significativo (*MSB*) são sujeitos à operação de extensão do sinal a partir do *bit* mais significativo (*MSB*) [67].

O *formato 212* consiste na representação de cada uma das amostras por 12 *bits* (1 *Byte* + 4 *bits*(1 *nibble*)) em notação de complemento para 2, efectuando-se o registo dos *bytes* no ficheiro de dados do *byte* menos significativo para o mais significativo [67].

A primeira amostra é obtida a partir dos 12 *bits* mais próximos em relação ao *bit* menos significativo (*LSB*) dos primeiros 2 *Bytes*, seguindo a lógica *least significant Byte first*.

A segunda amostra é formada a partir dos últimos 4 *bits* dos primeiros 2 *Bytes* (que equivalem aos 4 *bits* mais próximos do *bit* mais significativo *MSB* da amostra de 12 *bits*) e o próximo *byte*.

Este é um processo iterativo que é repetido para cada conjunto sucessivo de 2 amostras [67].

O formato *EDF* é um formato simples e versátil para transmissão e armazenamento de sinais fisiológicos e biológicos multi-canal.

Neste formato de ficheiros, os primeiros 256 *bytes* do cabeçalho especificam o número da versão do formato, a identificação local do paciente e do registo, o número de registos de dados e o número de sinais (*ns*) em cada registo de dados [68].

Para cada sinal seguem-se mais 256 *bytes* no cabeçalho que especificam o tipo de sinal (*EEG*, temperatura corporal, etc.), a calibração da amplitude e o número de amostras em cada registo de dados. A partir da calibração da amplitude e do número de amostras de cada registo é possível deduzir a frequência de amostragem uma vez que se conhece a duração de um registo de dados [68]

Deste modo, o formato permite diferentes ganhos e frequências de amostragem para cada sinal.

O cabeçalho contém $256 + (ns \times 256)$ *bytes* [68].

Neste formato, à semelhança do formato 16 do *PhysioNet* cada amostra é representada por 2 *Bytes* (16 *bits*) em notação de complemento para 2.

6.3 IMPLEMENTAÇÃO DO PROJECTO

6.3.1 Análise MATLAB

Numa primeira fase, recorreu-se à ferramenta *MATLAB* para definir uma estratégia, ou seja, um algoritmo com os passos a seguir para implementação da aplicação móvel no que se refere às etapas de amostragem, filtragem e compressão.

Para tal, pretendia-se utilizar dois *Datasets* do *PhysioNet*: um orientado para a filtragem do ruído presente no sinal original (*Dataset* proveniente da base de dados *ECG-ID Database* do *PhysioNet* [35]) e um orientado para a detecção de arritmias (*Dataset* proveniente da base de dados *MIT-BIH Arrhythmia Database*), tendo sido utilizado apenas o primeiro *DataSet* para filtragem do ruído no sinal de ECG.

O primeiro *DataSet* utilizado foi o *DataSet ECG-ID Database* do *PhysioNet* sobre o qual se efectuaram diversas experimentações com base em alguns algoritmos já existentes e técnicas de *Thresholding*.

A Figura 48 representa um dos sinais presentes no *DataSet ECG-ID Database* do *PhysioNet* ainda sem nenhum estágio de filtragem.

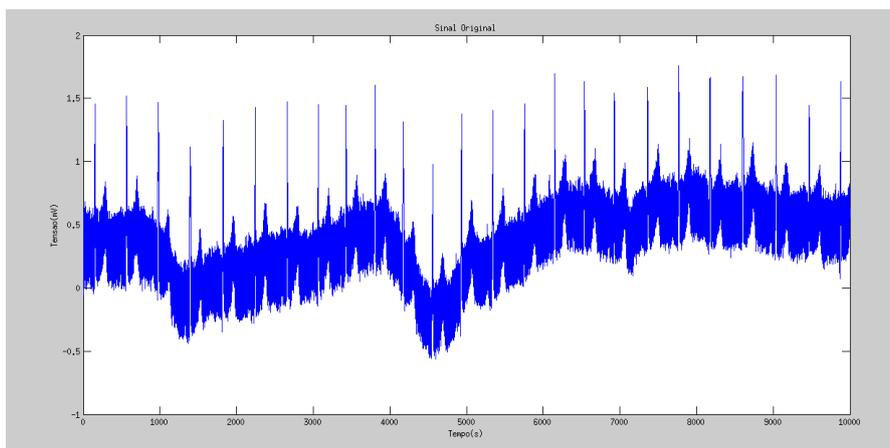


Figura 48: Sinal de ECG original/ não filtrado.

A Figura 49 corresponde a um zoom de 2 períodos do sinal original.

Na estratégia implementada subamostrou-se o sinal original por um factor de 2. Isto permite reduzir a frequência de amostragem e os custos de transmissão de dados em termos de energia associados à aplicação móvel.

Este passo foi implementado com recurso ao comando *downsample()* do *MATLAB* como é apresentado de seguida:

```
n=2;
```

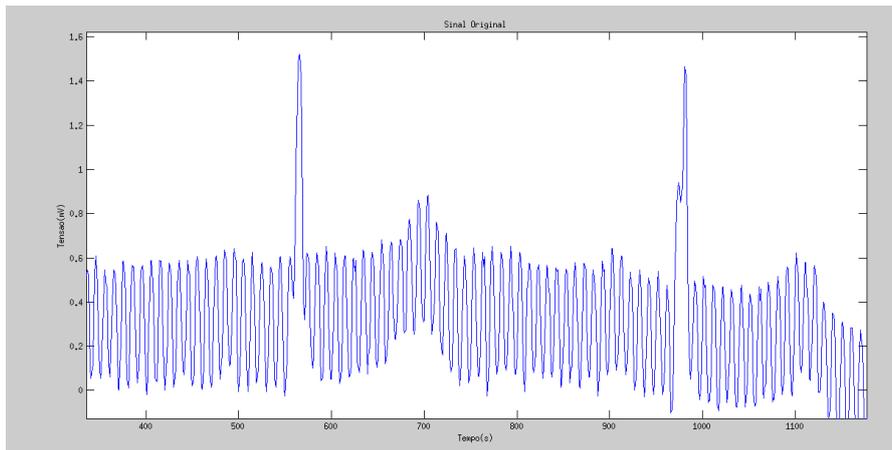


Figura 49: Zoom de 2 períodos do sinal original.

```
t=downsample(t,n);
s=downsample(s,n);
fs=fs/n;
```

O comando do *MATLAB* *downsample* recebe como argumentos um vector ou uma matriz como primeiro argumento e um factor do tipo inteiro como segundo argumento. Este comando inicia com a primeira amostra do sinal que se pretende subamostrar e mantém todas as amostras que sejam múltiplas do factor inteiro que foi previamente passado como argumento. Todas as outras amostras são descartadas do sinal original obtendo-se um sinal subamostrado por um certo factor n .

Na Figura 50 apresenta-se o sinal original subamostrado por um factor de 2.

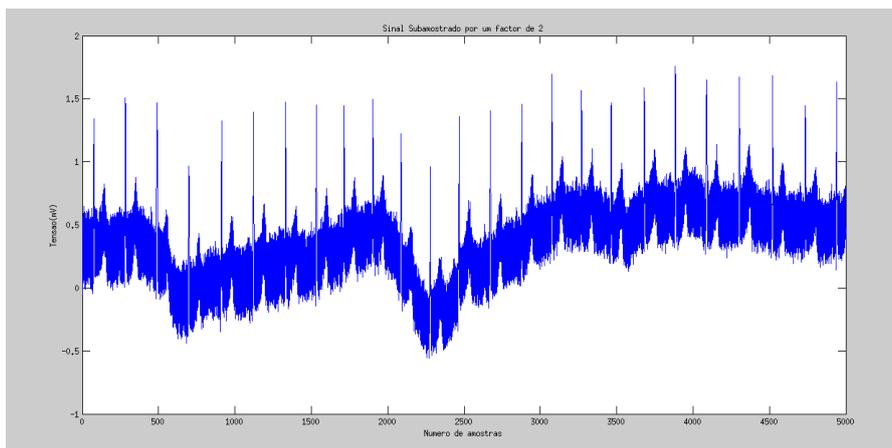


Figura 50: Sinal de ECG subamostrado por um factor de 2.

A Figura 51 corresponde a um Zoom de 2 períodos do sinal subamostrado.

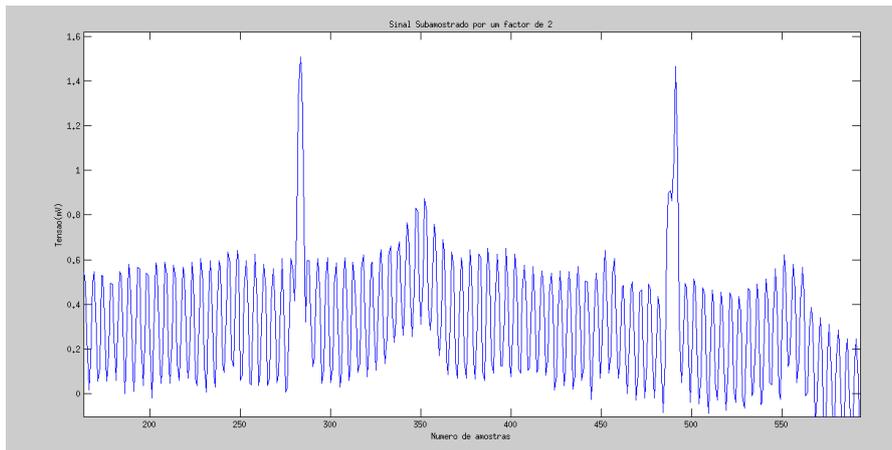


Figura 51: Zoom de 2 períodos do sinal subamostrado.

Para remoção da Flutuação de Linha de Base recorreu-se ao filtro de Hamilton definido pela Equação 48:

$$y(nT) = y(nT - T) - x(nT)/32 + x(nT - 16T) - x(nT - 17T) + x(nT - 32T)/32 \quad (48)$$

e implementado no *MATLAB* através dos zeros do numerador e do denominador e com recurso ao comando *filter*:

```
b=zeros(1,33);
b(1)=-1/32;
b(17)=1;
b(18)=-1;
b(33)=1/32;

a=zeros(1,33);
a(1)=1;
a(2)=-1;

s=filter(b,a,s);
```

Na Figura 52 representa-se o sinal após se ter procedido à remoção da Flutuação da Linha de Base.

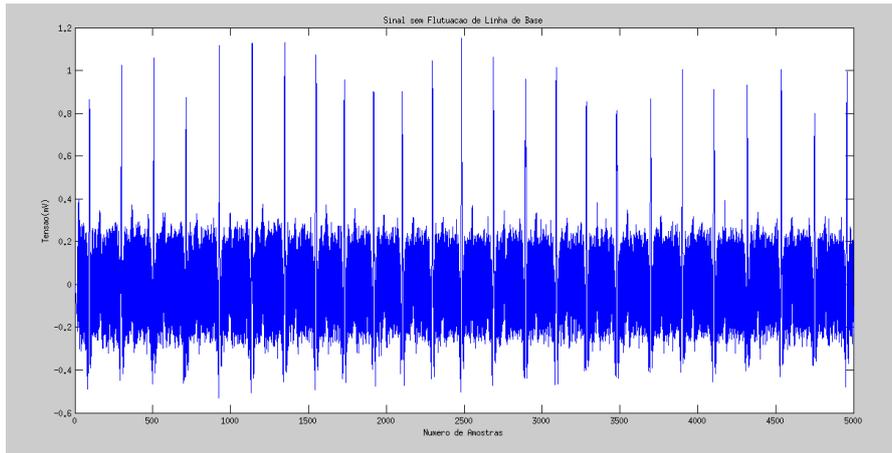


Figura 52: Sinal de ECG sem Flutuação de Linha de Base.

A Figura 53 corresponde a um zoom de 2 períodos do sinal de ECG sem Flutuação de Linha de Base.

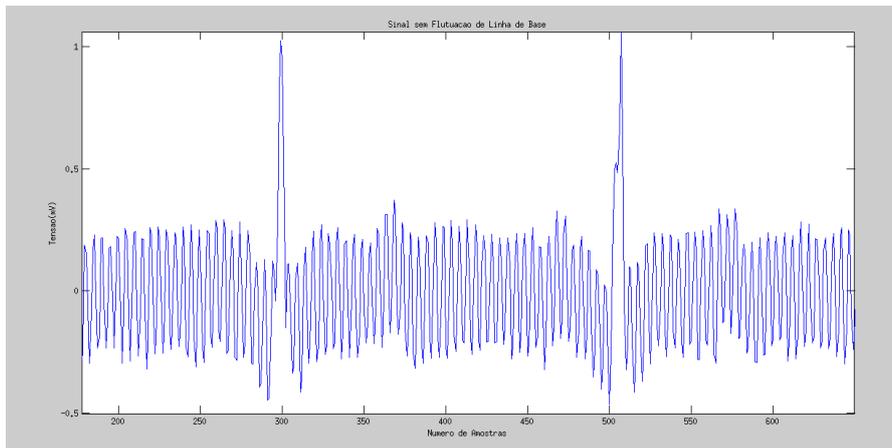


Figura 53: Zoom de 2 períodos de ECG sem Flutuação de Linha de Base.

Para remoção do ruído de *EMG* foi utilizado um filtro passa-baixo de Lynn-Hamilton de acordo com a Equação 49:

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T) \quad (49)$$

e implementado no *MATLAB* da seguinte forma:

```
b=zeros(1,13); % coeficientes do numerador -> bx
b(1)=1;
b(7)=-2;
b(13)=1;
```

```

a=zeros(1,13); % coeficientes do denominador -> -ay
a(1)=1;
a(2)=-2;
a(3)=1;

s=filter(b, a, s)/13;

```

A Figura 54 representa o sinal de *ECG* após os estágios de remoção da flutuação da linha de base e remoção do ruído de *EMG*.

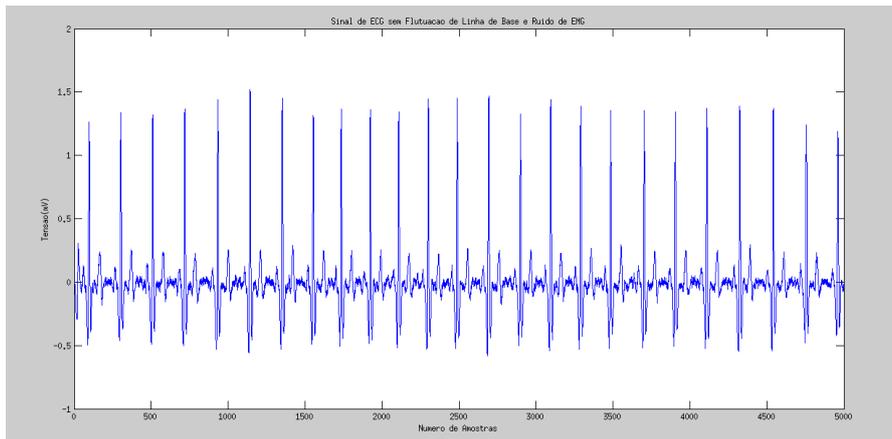


Figura 54: Sinal de *ECG* sem Flutuação de Linha de Base e ruído de *EMG*.

A Figura 55 corresponde a um zoom de 2 períodos do sinal de *ECG* sem Flutuação de Linha de Base e ruído de *EMG*.

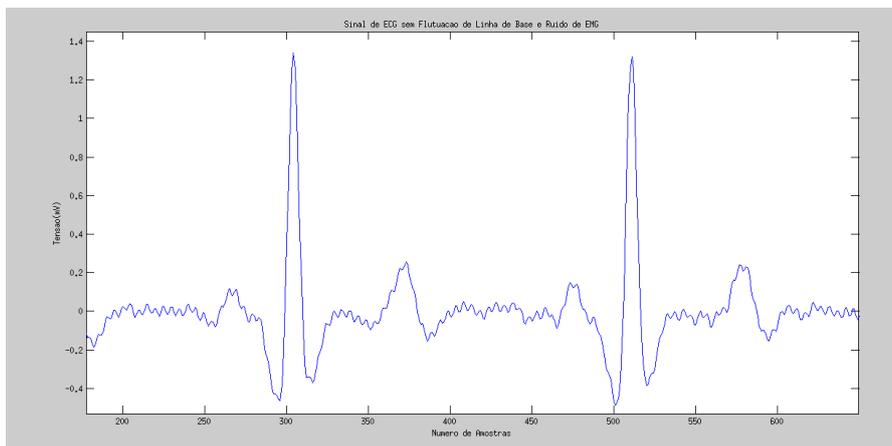


Figura 55: Zoom de 2 períodos do sinal de *ECG* sem Flutuação de Linha de Base e ruído de *EMG*.

Para atenuar a interferência da rede eléctrica aplicou-se um filtro *Notch*:

```
[b,a]=fir1(100,[50*2/fs-0.03 50*2/fs+0.03],'stop');
```

Na figura 56 apresenta-se o sinal após os vários estágios de filtração.

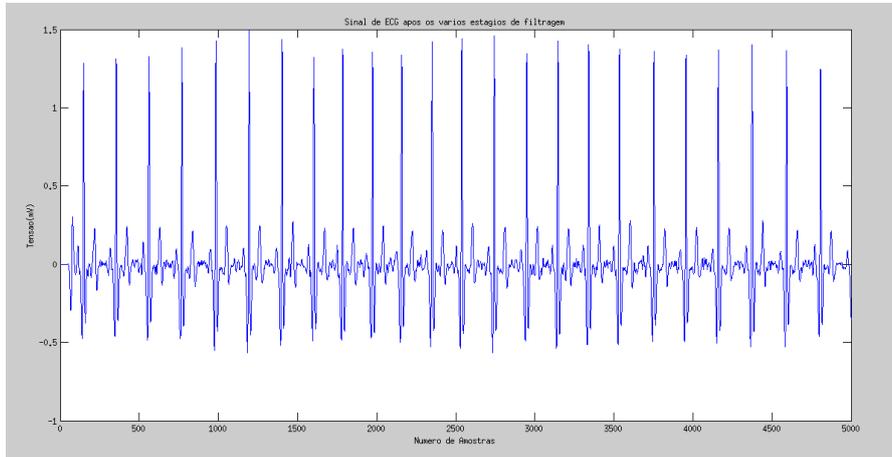


Figura 56: Sinal de ECG após os vários estágios de filtração.

Na figura 57 apresenta-se o *zoom* de 2 períodos do sinal após os vários estágios de filtração.

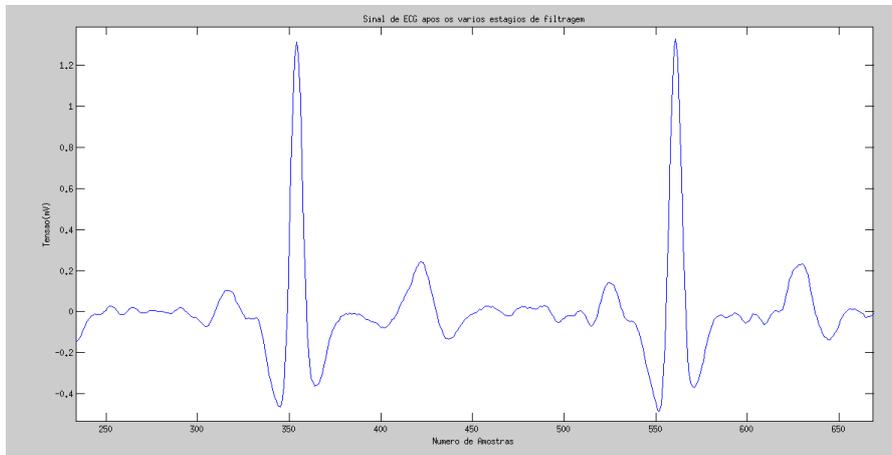


Figura 57: Zoom de 2 períodos do sinal de ECG após os vários estágios de filtração.

Após a aplicação de algumas técnicas de *Thresholding*, surgem delineados os vários tipos de onda no sinal de ECG inicial (ver Figura 58).

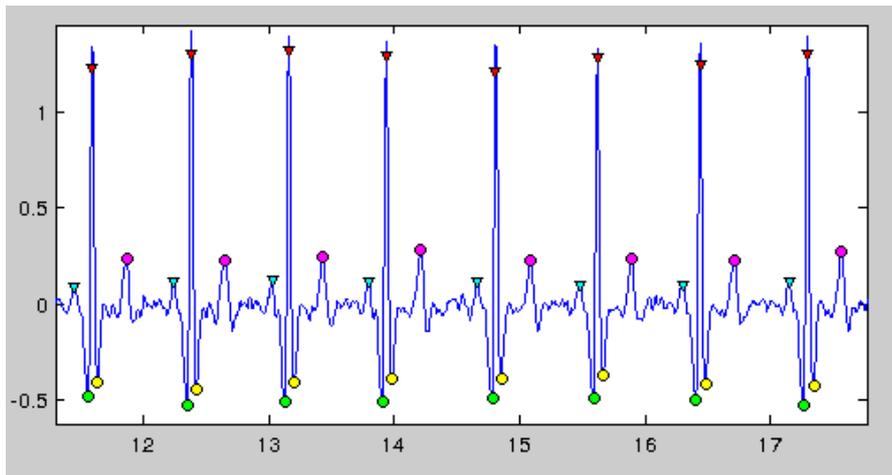


Figura 58: Sinal de ECG filtrado com a delineação das ondas P, Q, R, S e T.

Estas técnicas de Thresholding tiveram por base as primeiras diferenças do sinal de ECG. No caso de a primeira diferença ser positiva e a segunda negativa obtém-se um máximo local do sinal. Quando a primeira diferença é negativa e a segunda é positiva verifica-se um mínimo local do sinal. Comparando os máximos locais do sinal é possível obter o seu máximo global. O máximo global definido para cada período corresponde ao valor da onda R no ECG. Em cada período que corresponde a um intervalo de tempo de cerca de 200 ms, analisa-se a presença de um mínimo (onda S) e de um máximo local (onda T), após o índice de posição da onda R. Verifica-se também a presença de um máximo (onda P) e de um mínimo (onda Q) locais, antes do índice de posição da onda R. Esta análise em termos de extremos da função é efectuada por via das primeiras diferenças tal como referido acima.

6.3.2 Análise em plataforma móvel

Numa segunda fase, migrou-se para a plataforma *Android* e testou-se o algoritmo desenvolvido em *MATLAB* de modo a desenvolver a aplicação móvel para detecção de algumas arritmias cardíacas.

A aplicação móvel constante das Figuras 60, 61, 65, 66 é constituída por uma única classe que estende a classe *Activity* e que invoca os métodos *getAdapter()* e *enableBluetooth()* no interior do método *OnCreate()*.

Estes 2 métodos, o método *getAdapter()* e o método *enableBluetooth()* foram definidos da seguinte forma:

- *getAdapter()*:

```
@TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
public void getAdapter(){

    if(version <= jellybeanMR1){
        btadapter = BluetoothAdapter.getDefaultAdapter();
    }

    else{
        btmanager = (BluetoothManager)
            this.getSystemService(BLUETOOTH_SERVICE);
        btadapter = btmanager.getAdapter();
    }
}
```

No caso de a versão *Android* presente no dispositivo móvel ser inferior à versão *Android 4.2*, é utilizado o método **BluetoothAdapter.getDefaultAdapter()** para obter o adaptador de *Bluetooth*. Caso contrário, o método utilizado para obtenção do adaptador de *Bluetooth* é o método **btmanager.getAdapter()**.

- *enableBluetooth()*:

```
public void enableBluetooth(){

    if(btadapter!=null){

        if(!btadapter.isEnabled()){
            Intent iEnableBluetooth = new
                Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(iEnableBluetooth,
                REQUEST_ENABLE_BT);
        }

        else{
            return;
        }

    }

    else{
        toast("O dispositivo não suporta Bluetooth");
    }
}
```

Se existir adaptador de *Bluetooth* no dispositivo móvel e este não se encontrar activo, é criado um *Intent* que inquirir ao utilizador se este pretende permitir que a aplicação utilize o *Bluetooth* (Figura 64). Este *Intent* é iniciado com o método `startActivityForResult()`, possuindo também um método `onActivityResult()`

que invoca os métodos `pair()`, `connect()` e `acquire()` para além de gerar os gráficos da aplicação.

Os métodos `pair()`, `connect()` e `acquire()` são definidos da seguinte forma:

– **pair()**

```
public void pair(){
    try {
        btdevice = btadapter.getRemoteDevice(remoteDevice);
        toast("A emparelhar dispositivos...");
        Method mpair =
            btdevice.getClass().getMethod("createBond",
                (Class[]) null);
        mpair.invoke(btdevice, (Object[]) null);
        toast("Os dispositivos encontram-se emparelhados");
    } catch (Exception e) {
        toast("Não foi possível emparelhar os dispositivos");
        pair();
    }
}
```

– **connect()**

```
public void connect(){

    try {
        btadapter.cancelDiscovery();
        btsocket =
            btdevice.createRfcommSocketToServiceRecord(uuid);
        btsocket.connect();
        toast("Os dispositivos encontram-se conectados");
    } catch (IOException e) {
        toast("Não foi possível conectar os dispositivos");
        connect();
    }

}
```

- acquire()

```
public void acquire(){

    try {
        BITalinoDevice bitalino = new BITalinoDevice(fs,
            selChannels);
        bitalino.open(btsocket.getInputStream(),
            btsocket.getOutputStream());
        bitalino.start();
        frames = bitalino.read(10000);

    } catch (BITalinoException e) {
        toast("Não foi possível entrar no modo de aquisição");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        toast("Não foi possível obter as streams de dados");
    }

}
```

Figura 59: Método Acquire().

O método *acquire()* instancia um objecto do tipo *BITalinoDevice* com uma certa frequência de amostragem, *fs*, e com os canais dos quais se pretende fazer a aquisição, *selChannels*. Posteriormente, através do método *open()* são abertas as *streams* de dados desse mesmo dispositivo para leitura e escrita. De seguida, o objecto do tipo *BITalinoDevice* é inicializado e através do método *read()* é possível ler um número finito de amostras (10000 no caso da Figura 59).

A aplicação desenvolvida em *Android* recebe os dados do canal de *ECG* da placa de aquisição através do módulo *Bluetooth*. Com a Biblioteca *achartengine* foram gerados gráficos na aplicação *Android*. A aplicação *Android* das Figuras 60, 61, 65, 66 representa em termos gráficos o sinal de *ECG*. Este sinal de *ECG* corresponde ao sinal original sem nenhum estágio de filtragem a nível digital, sendo que o *BITalino* implementa algum tipo de filtragem por *hardware*. Foram desenvolvidas classes de código para subamostrar o sinal original e para realizar filtragem digital do sinal. Também se procurou através de determinadas técnicas de *Thresholding* detectar algumas arritmias no sinal de *ECG*.

As Figuras 60, 61, 62, 63, 64, 65 e 66 representam a configuração geral do dispositivo de aquisição e a Interface da aplicação móvel. A Figura 67 define um diagrama das várias etapas de desenvolvimento da aplicação.

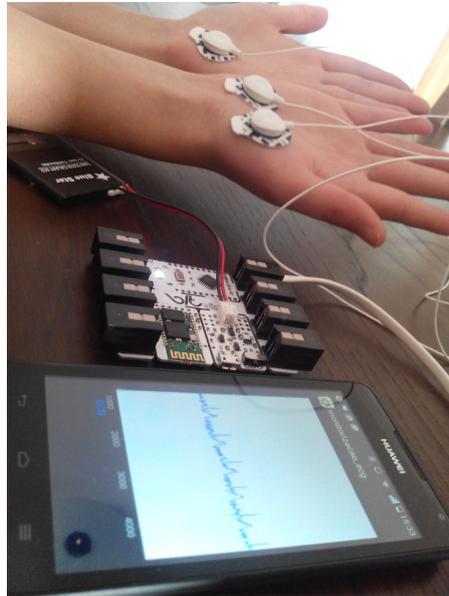


Figura 60: Configuração geral dos eléctrodos.

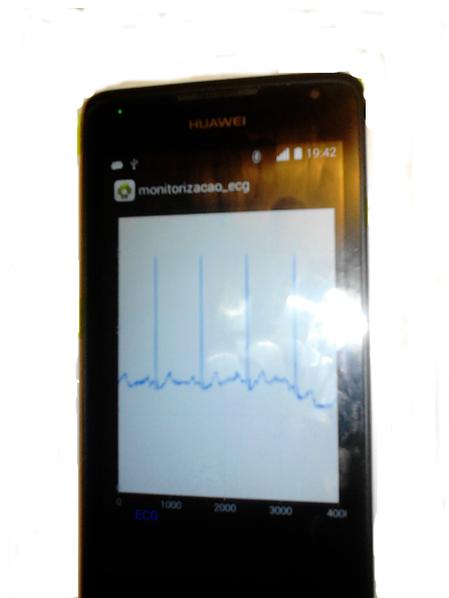


Figura 61: Geração dos gráficos na aplicação *Android*.

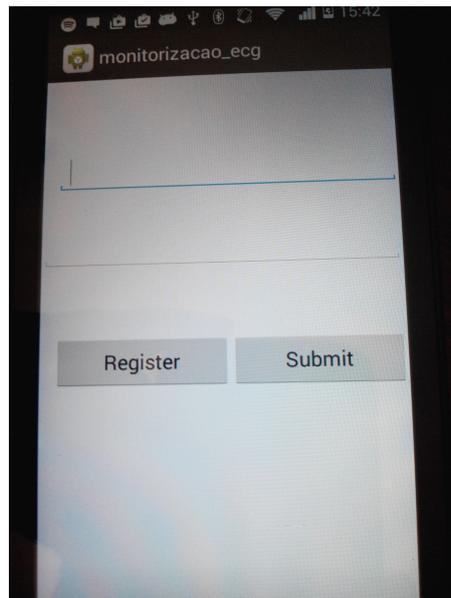


Figura 62: *Activity de Login.*

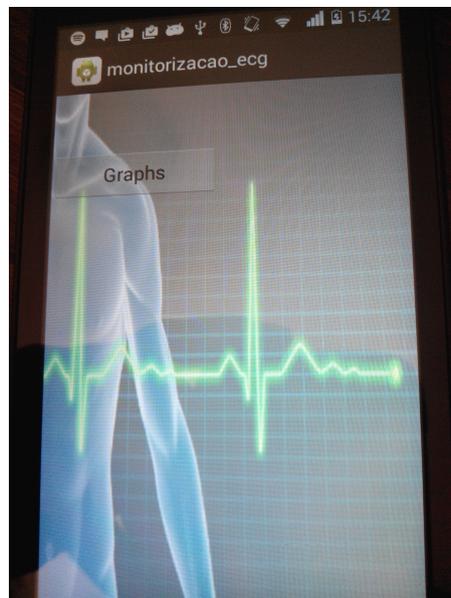


Figura 63: *SplashScreen.*

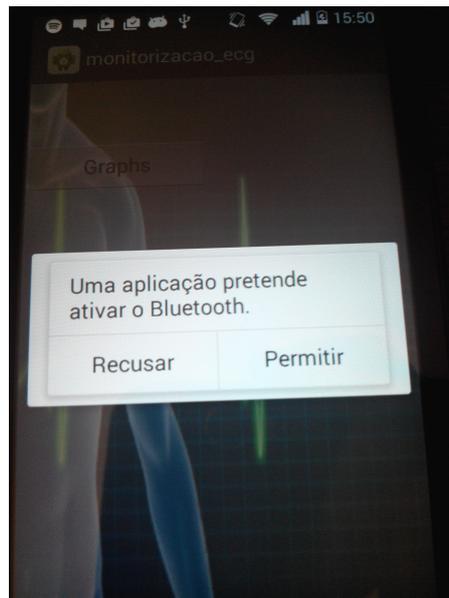


Figura 64: *Intent* para permitir a utilização do *Bluetooth*.

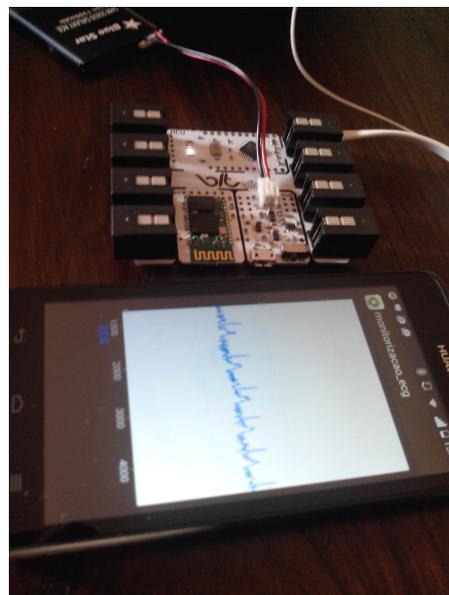


Figura 65: Dispositivo de aquisição e aplicação móvel.

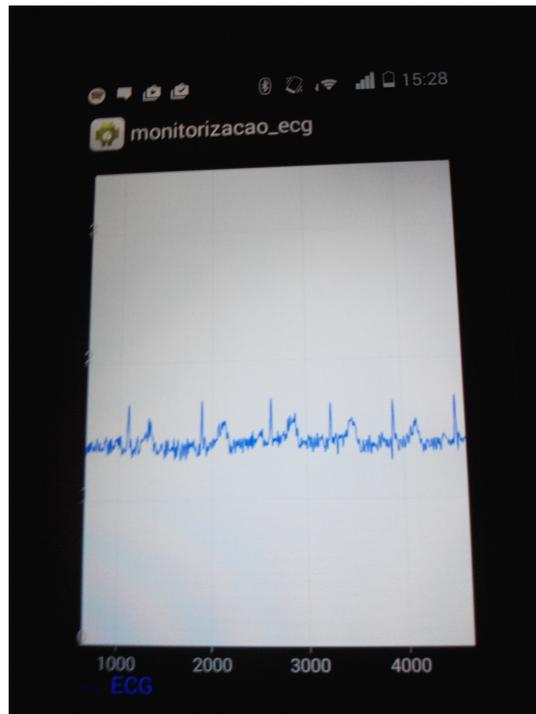


Figura 66: Interface Gráfica da Aplicação desenvolvida em Android.



Figura 67: Diagrama da Aplicação móvel desenvolvida.

O código foi reformulado passando a ser constituído por onze classes de código, sendo que cinco dessas classes estendem também a classe *Activity*. As classes *Comm*, *LoginActivity* (Figura 62) e *SplashScreen* (Figura 63) encontram-se a funcionar correctamente, não tendo sido possível verificar as funcionalidades de todas as classes. As classes foram definidas utilizando a Linguagem de Programação *Java*, que é uma linguagem de Programação Orientada a Objectos. Os *Layouts* associados a cada *activity* foram definidos em *XML*. Foi utilizado o *SDK* do *Android* e o *IDE* eclipse juntamente com o *Plugin ADT* para desenvolvimento de código para a plataforma *Android*.

O dispositivo móvel utilizado foi um Huawei Y530-U00, com um processador Dual Core de 1.2 GHz e 512 MB de RAM. A versão *Android* disponível no dispositivo é a versão 4.3.

Foram utilizadas a *API* do *BITalino* para *Android* e as bibliotecas guava [69] e *achartengine* [70], sendo que esta última permite a geração de gráficos na aplicação móvel.

Fazem parte dessas onze classes de código a classes *Comm*, *DetectAbnormalities*, *DSP*, *Ecg*, *GraphsActivity*, *LogActivity*, *LoginActivity*, *SplashScreen*, *TabLayout*, *ThisBitalino* e *Utils*. Apenas as classes *GraphsActivity*, *LogActivity*, *LoginActivity*, *SplashScreen* e *DetectAbnormalities* estendem a classe *Activity*. De seguida apresenta-se uma breve descrição das várias classes.

Classe Comm

Nesta classe encontra-se definida a parte de comunicações. Esta classe possui um construtor em que são passados como argumentos uma *String* que representa o dispositivo ao qual se pretende conectar a partir do *smartphone* e uma *Activity* que corresponde à *activity* em que é instanciado o elemento da Classe *Comm*. Este construtor estabelece que se essa *String* que é passada como argumento for igual ao primeiro elemento de um array com as *Strings* "bitalino" e "servidor" então será invocado o método *enableBluetooth*. Esta classe possui os métodos de classe *getAdapter()*, *enableBluetooth()* e *pair()*. O método *getAdapter* obtém um adaptador de *Bluetooth* para a versão de *Android* existente no *smartphone*. O método *enableBluetooth()* invoca o método anterior e no caso de existir um Adaptador *Bluetooth*, se o *Bluetooth* não se encontrar activado, cria um *Intent* para a aplicação permitir a utilização do *Bluetooth* e inicia uma *ActivityForResult*.

Classe Ecg

Nesta classe definem-se métodos para obter valores de durações e amplitudes e para determinar o ritmo cardíaco. Esta classe possui 2 *enums*, um com os Intervalos de ECG (*PR*, *QRS*, *ST* e *QT*) e um com as ondas de ECG (*P*, *Q*, *R*, *S* e *T*). Possui os métodos *duration(String string)*, *value(String wave,int pos)*, *heartRate()*, *isWave(String wave)*, *getValues()*, *getTimes()*, *Qvalues()*, *Svalues()*, *Tvalues()*, *Pvalues()* e *Rvalues()*.

Classe DSP

Nesta classe são definidos métodos que viabilizem o processamento digital dos sinais a analisar. Esta classe possui os métodos *downsample(double [] array, double times)*, *filtering(Ecg signal, String filter)* e *compression()*. O método *downsample()*, ao qual são passados com argumentos um *array* do tipo *double* e um factor também do tipo *double*, subamostra o sinal por este mesmo factor. O método *filtering()* possui 2 argumentos, o primeiro corresponde a um sinal de ECG do tipo *Ecg* e o segundo corresponde a um filtro dos que estão contidos no enum *Filters* ("FLB", "EMG" ou "PLI"). Consoante o filtro que seja passado como argumento, é aplicada a função de transferência respectiva ao sinal de ECG. O método *compression()* não se encontra ainda definido talvez tenha sentido implementá-lo nas próximas versões deste projecto.

Classe GraphsActivity

Nesta *Activity* surge o gráfico gerado a partir dos valores do canal de ECG obtidos pela plataforma de aquisição de biosinais utilizada (*BITalino*). Este gráfico é gerado a partir da biblioteca *achartengine* que foi importada para o projecto *Android* no *eclipse*.

Classe LogActivity

Esta classe através de uma *listview* e de um *ArrayAdapter* dispõe um conjunto de *Strings* em que cada *String* é constituída por uma *timestamp* e pelas várias medidas de um sinal de ECG (Ritmo Cardíaco, Duração do Intervalo *PR*, *QRS*, *ST*, valor médio das ondas *P*, *Q*, *R*, *S*, *T*).

Classe LoginActivity

Esta *Activity* constitui a *Activity* de *Login* na Aplicação e encontra-se definida no ficheiro *Manifest.xml* do Projecto como *Main Activity* sendo a *activity* a qual se associa um *Launcher* para a Aplicação.

Classe TabLayout

Esta *Activity* cria um elemento do tipo *TabHost* ao qual são adicionados dois separadores, um para representar os gráficos e outro que serve como *Log* do sistema com os vários parâmetros do *ECG* representado no outro separador. Cada um dos separadores carrega uma *Activity*: no caso do primeiro separador é carregada a *GraphsActivity* que representa em termos gráficos a informação fornecida pelos sensores aquando do estágio de aquisição e que é iniciada no caso de o *Bluetooth* se encontrar activo; no segundo caso, é carregada a *Log-Activity* que realiza o *Log* dos parâmetros dos vários sinais de *ECG* organizados de acordo com uma *timestamp*.

Classe ThisBitalino

Esta classe define um elemento da classe *bitalino* tal como definido na *API* do *BITalino* para Android com um determinado conjunto de parâmetros (*Frequencia de Amostragem*, *Mac Address* e *UUID* do dispositivo) tendo sido seleccionado o canal correspondente ao *ECG* no dispositivo de aquisição através da sua *API*. Esta classe possui um construtor ao qual são passados como argumentos a *Activity* na qual será instanciado um elemento do tipo *ThisBitalino* e um valor de frequência de amostragem do tipo inteiro.

Classe SplashScreen

Esta classe possui um *background* e é onde aparecem as mensagens da aplicação para o utilizador e onde surgem os *Intents* para activação do *Bluetooth*.

Classe Utils

Esta classe define um conjunto de métodos gerais para serem utilizados pelas outras classes. O método *toast()* instancia um objecto do tipo *Toast* com a mensagem que lhe é passada como argumento, colocando-a no centro do ecrã. O método *SamplestoTime* permite a utilização deste mesmo método com diferentes argumentos (um array do tipo *double*, um array do tipo *int*, ou um array do tipo *BitalinoFrame*). Converte o número de amostras num tempo em segundos. O método *TimeToArray()* recebe como argumento um tempo máximo do tipo *double* e cria um array entre 0 e esse valor que foi definido anteriormente. O método *zeros()* cria um array preenchido com zeros.

Classe DetectAbnormalities

Esta classe visa detectar algumas arritmias através de técnicas de *Thresholding*. Possui um construtor que recebe como argumento um sinal do tipo *Ecg*. No caso de o ritmo cardíaco ser inferior a 60 *bpm* é atribuída a *flag* *Bradicardia=true*, se for superior a 100, *Taquicardia* toma o valor *true*. Se não existir onda *P* e a duração do complexo *QRS* for inferior a 0.06 s, ou a duração do complexo for superior a 0.1 s então as *flags* *AF* e *Taquicardia* tomam o valor *true*.

CONCLUSÕES E TRABALHOS FUTUROS

7.1 CONCLUSÕES

No sistema actual a aplicação comunica com o *BITalino* por *Bluetooth*, obtém os dados do sensor de *ECG* e gera gráficos na aplicação através da biblioteca *achartengine* [70].

Inicialmente foi utilizada a plataforma de testes *MATLAB* onde foram testados vários tipos de filtros tais como um filtro passa-alto de Butterworth com frequência de corte aos 0.5 Hz e um filtro de Chebyshev tipo I e um filtro de Chebyshev tipo II, ambos com frequência de corte aos 0.5 Hz e um *ripple* máximo de 3 dB.

Também se experimentou reduzir algumas interferências de sinal com um filtro de Savitzky-Golay, no entanto, este filtro induz alguma distorção no sinal.

Por fim, recorreu-se a alguns filtros do Algoritmo Pan-Tompkins e concluiu-se que estes apresentam um melhor resultado à saída, atenuando a Flutuação de Linha de Base e a Interferência do sinal de *EMG*.

Para remoção da interferência de rede eléctrica de 50 Hz utilizou-se um filtro *notch* também designado de rejeita-banda. Neste processo de desenvolvimento de uma estratégia de filtragem, utilizou-se o *DataSet ECG-ID Database* do *PhysioNet*.

Numa segunda fase migrou-se para a plataforma *Android*. A aplicação desenvolvida recebe os dados do canal de *ECG* do dispositivo de aquisição seleccionado (*BITalino* - versão *plugged*).

A comunicação com esse dispositivo é efectuada através do protocolo de transmissão de dados a curta distância *Bluetooth*, através das classes de código desenvolvidas para a parte de Comunicações. Com a Biblioteca *achartengine* foram gerados gráficos na aplicação *Android*.

A aplicação *Android* das Figuras 60, 61, 65, 66 representa em termos gráficos o sinal de *ECG*.

Este sinal de *ECG* corresponde ao sinal original sem nenhum estágio de filtragem a nível digital, sendo que o *BITalino* implementa algum tipo de filtragem por *hardware*.

Utilizaram-se alguns indivíduos do sexo feminino e masculino com idades compreendidas entre os 18 e os 60 anos de idade, tendo sido colocados os eléctrodos na zona palmar das mãos para poder gerar os gráficos na aplicação móvel a partir dos dados dos sinais de *ECG* de cada um desses indivíduos.

Foram desenvolvidas classes de código para subamostrar o sinal original e para realizar filtragem digital do sinal. Também se procu-

rou através de determinadas técnicas de *Thresholding* detectar algumas arritmias no sinal de *ECG*.

Dos sistemas de aquisição, o mais leve é o *BITalino*, no entanto, o *kit* de desenvolvimento apresenta uma portabilidade reduzida devido ao facto de os fios dos eléctrodos terem um comprimento bastante limitado e devido à fragilidade dos fios na zona da conexão aos eléctrodos. Além disso, possui um cabo *RJ22* que não permite grande liberdade de movimentos e apresenta uma conexão deficiente ao sensor.

A versão do *BITalino* utilizada (versão *Plugged*) não é a mais adequada para o desenvolvimento do projecto, uma vez que o cabo *RJ22* que faz a conexão da placa principal ao sensor é muito longo e pouco flexível o que dificulta a sua utilização por parte de quem o esteja a utilizar.

No conjunto de acessórios do *BITalino* consta um cabo *RJ22* com menor comprimento, no entanto isto não vem com o *BITalino* por defeito e seria necessário adquiri-lo à parte.

A versão *Board* seria talvez mais adequada para o desenvolvimento deste tipo de projecto uma vez que é possível ligar o módulo do sensor de *ECG* directamente à placa principal.

A *API* do *BITalino* em *Java* através do método *read()* permite apenas a leitura de um conjunto finito de amostras.

Como no caso concreto deste projecto se pretendia uma monitorização contínua em tempo real, invocou-se o método *read()* com 5000 amostras, no interior de um ciclo *while* que era executado caso se verificasse a existência de um dispositivo *Bluetooth* com o *Mac Address* do dispositivo *BITalino*, para tentar replicar uma situação de aquisição dos biosinais em tempo real, até que os dispositivos deixassem de estar emparelhados entre si.

Foram sentidas algumas dificuldades na conversão da descrição textual dos algoritmos para pseudo-código, uma vez que, nalguns casos não se encontravam descritos todos os passos dos algoritmos.

A conversão para pseudo-código seria mais fácil a partir de um código noutra linguagem de programação, no entanto não foram encontradas versões destes algoritmos noutras linguagens de programação.

Por uma questão de tempo, em vez de se efectuar a experimentação com o *DataSet MIT-BIH Arrhythmia Database* passou-se directamente para *Android*.

7.2 TRABALHOS FUTUROS

De modo a implementar uma primeira versão da arquitectura proposta sugere-se que seja incluído nos trabalhos futuros a criação de um servidor integrado com uma base de dados com uma aplicação composta por um *servolet* que gere os pedidos de *https* da aplicação cliente e que interage com um ficheiro *PHP* para verificar se os dados

que são transmitidos ao servidor pela aplicação já constam da base de dados.

Para tentar evitar ataques informáticos ao sistema do tipo *man in the middle*, tanto a comunicação entre o dispositivo de aquisição e o smartphone como a comunicação entre a aplicação móvel e o servidor têm de se encontrar devidamente protegidas através de técnicas criptográficas que podem ser implementadas com recurso a chaves assimétricas ou à encriptação do tipo *AES (Advanced Encryption Standard)*.

Por forma a reduzir os custos de transmissão, o sinal de ECG que se encontra representado em termos gráficos pela aplicação móvel deve ser comprimido, encriptado e enviado para o servidor de forma a ser armazenado na base de dados para ser posteriormente visualizado pelos profissionais de saúde.

Por questões de interoperabilidade, os dados devem ser transmitidos sob a forma de pacotes de dados que contêm tanto os parâmetros a analisar, como o sinal de ECG comprimido e devidamente encriptado organizados num formato de uso geral como o formato *EDF* para uma mais fácil integração com os ficheiros da unidade de saúde em causa.

A nível das classes de código desenvolvidas, considera-se que a classe DSP, que define alguns métodos para processamento digital do sinal, deveria ser implementada na Linguagem de Programação C (Programação Imperativa) em vez da Linguagem de Programação Java (Programação Orientada a Objectos) por uma questão de desempenho da aplicação.

Talvez tenha sentido a inclusão de um estudo de acelerometria para tentar de alguma forma reduzir o número de artefactos induzidos no sinal de ECG pela contracção muscular.

Para cada um dos algoritmos enunciados no presente documento, seria útil efectuar uma análise da complexidade de modo a seleccionar os algoritmos mais eficientes.

De modo a aumentar a portabilidade do dispositivo e para que a comunicação com o servidor possa ser efectuada mesmo em situações críticas em que não existem grandes meios de comunicação e não é possível a utilização de *smartphones*, teria sentido modificar o BITalino para que este comunicasse directamente com o servidor por *GSM* que é uma das tecnologias de rede utilizadas pelos dispositivos móveis.

Para tal, em vez do módulo Bluetooth ter-se-ia um módulo GSM cujos terminais Rx e Tx seriam ligados directamente aos terminais Tx e Rx, respectivamente do microcontrolador presente no *BITalino*.

Neste caso, seria apenas criada uma aplicação web para a captura dos dados e uma interface web para ser utilizada pelos profissionais de saúde.

Uma vez que o âmbito de aplicação deste projecto se traduz na monitorização dos sinais vitais dos pacientes em mobilidade, teria sen-

tido que fosse efectuado um teste piloto numa Instituição de Saúde de modo a proceder à sua implementação e poder prestar aos pacientes um melhor serviço.

BIBLIOGRAFIA

- [1] José Guerreiro Ramón González Juanatey. Tutorial de electrocardiograma (apontamentos da universidade de santiago de compostela).
- [2] Liu Puming. Service agreements and facilities for m-health vital sign monitoring. 2006.
- [3] Joseph D. Bronzino. *The Biomedical Engineering Handbook*, volume 1. CRC Press and IEEE Press, 2 edition, 2000.
- [4] Fórum bitalino. Disponível em: <http://forum.bitalino.com> Consultado em: Março de 2014.
- [5] Michaela Lascu and Dan Lascu. Labview event detection using pan-tompkins algorithm. 2007.
- [6] Monitor de holter. Disponível em: <http://www.nhlbi.nih.gov/health/health-topics/topics/holt/while.html> Consultado em: Maio de 2014.
- [7] Ritmo sinusal normal. Disponível em: http://www.madsci.com/manu/ekg_rhy.htm Consultado em: Maio de 2014.
- [8] Jaako Malmivuo and Robert Plonsey. *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press, 1995.
- [9] Ken Grauer. *A 1st Book on ECGs*. KG/EKG Press, 2014.
- [10] Arritmia sinusal. Disponível em: <http://research.vet.upenn.edu/Portals/75/sinusal.gif> Consultado em: Fevereiro de 2014.
- [11] Mrinal Kr. Mandal. *Multimedia Signals and Systems*. Kluwer Academic Publishers, 2003.
- [12] M. L. Ahlstrom and W. J. Tompkins. Digital filters for real-time ecg signal processing using microprocessors. *IEEE Transactions on Biomedical Engineering*, 32:708–713, 1985.
- [13] Azita Dordari. Data reduction algorithm for on-line ecg applications. Master's thesis, Carleton University, Ottawa, Ontario, 2006.
- [14] Bitalino board. Disponível em: <https://codebits.eu/s/blog/1a0a660c785d668a2b28f0b702df2611> Consultado em: Dezembro de 2013.

- [15] Hugo Silva, José Guerreiro, André Lourenço, Ana Fred, and Raúl Martins. Bitalino: A novel hardware framework for physiological computing. 2014.
- [16] e-health sensor platform. Disponível em: <http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> Consultado em: Novembro 2013.
- [17] Sensium. Disponível em: <http://www.techdesignforums.com/practice/technique/sensium-a-1v-micropower-soc-for-vital-sign-monitoring/> Consultado em: Novembro de 2013.
- [18] Mercury: A wearable sensor network platform for high-fidelity motion analysis. Disponível em: <http://fiji.eecs.harvard.edu/Mercury> Consultado em: Fevereiro 2014.
- [19] Carsten W. Mundt, Kevin N. Montgomery, Usen E. Udoh, Valerie N. Barker, Guillaume C. Thonier, Arnaud M. Tellier, Robert D. Dicks, Robert B. Darling, Yvonne D. Cagle, Nathalie A. Cabrol, Stephen J. Ruoss, Judith L. Swain, John W. Hines, and Gregory T. A. Kovacs. A multiparameter wearable physiologic monitoring system for space and terrestrial applications. *IEEE Transactions on Information Technology in Biomedicine*, 9, 2005.
- [20] Adrian Burns, Barry R. Greene, Michael J. McGrath, Terrance J. O'Shea, Benjamin Kuris, Steven M. Ayer, Florin Stroiescu, and Victor Cionca. Shimmer - a wireless sensor platform for noninvasive biomedical research. *IEEE Sensors Journal*, 10, 2010.
- [21] Bioharness. Disponível em: <http://www.takingspace.org/now-streaming-data-from-the-zephyr-bioharness-3/> Consultado em: Março de 2014.
- [22] Urs Anliker, Jamie A. Ward, Paul Lukowickz, Gerhard Tröster, François Dolveck, Michael Baer, Fatou Keita, Eran B. Schenker, Fabrizio Catarsi, Luca Coluccini, Andrea Belardinelli, Dror Shklarski, Menachem Alon, Etienne Hirt, Rolf Schmid, and Milica Vuskovic. Amon: A wearable multiparameter medical monitoring and alert system. *IEEE Transactions on Information Technology in Biomedicine*, 2004.
- [23] Codeblue: Wireless sensors for medical care | harvard sensor networks lab. <http://fiji.eecs.harvard.edu/CodeBlue>. Consultado em: Dezembro de 2013.
- [24] Qadeer Khan and Sarvesh Bang. Energy harvesting for self powered wearable health monitoring system (health, s. francisco). 2009.

- [25] Lifepebble. Disponível em: www.medgadget.com/2009/10/sensium_life_pebble_wireless_vitals_monitor_for_sport_training_cardiac_health_auditing.html Consultado em: Fevereiro de 2014.
- [26] Vital-jacket. Disponível em: <http://paginas.fe.up.pt/~ee02200/wiki/doku.php?id=project:stuff:finalreport:full> Consultado em: Março de 2014.
- [27] Vital-jacket. Disponível em: <http://www.portugalglobal.pt/PT/ComprarPortugal/Paginas/Inovacao.aspx> Consultado em: Abril de 2014.
- [28] Helena Fernandez-Lopez, José Augusto Afonso, José Higinio Correia, and Ricardo Simões. Zigbee-based remote patient monitoring. In *PHealth 2012: Proceedings of the 9th International Conference on Wearable Micro and Nano Technologies for Personalized Health, June 26-28, 2012, Porto, Portugal*, volume 177, page 229. IOS Press, 2012.
- [29] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. Alarm-net: Wireless sensor networks for assisted-living and residential monitoring. Technical report, Department of Computer Science, University of Virginia, 2006.
- [30] Rami J. Oweis and Basim O. Al-Tabbaa. Qrs detection and heart rate variability analysis: A survey. *Biomedical Science and Engineering*, 2014.
- [31] Pedro Gomes. *Sistema para aquisição, monitorização e análise da actividade cardíaca*. PhD thesis, Universidade do Minho, 2010.
- [32] Murray Longmore, Ian B. Wilkinson, Edward H. Davidson, Alexander Foulkes, and Ahmad R. Mafi. *Oxford Handbook of Clinical Medicine*. Oxford University Press, 8 edition, 2010.
- [33] Tabela de códigos ascii. Disponível em: <http://www.asciitable.com/>. Consultado em: Outubro de 2014.
- [34] Barry Mirkin and Mary Beth Weinberger. The demography of population ageing. *Population Bulletin of the United Nations*, 42(43):37–53, 2001.
- [35] Physionet. <http://www.physionet.org/>. Consultado em Abril de 2014.
- [36] Paul Kligfield, Leonard S Gettes, James J Bailey, Rory Childers, Barbara J Deal, E William Hancock, Gerard van Herpen, Jan A

- Kors, Peter Macfarlane, David M Mirvis, et al. Recommendations for the standardization and interpretation of the electrocardiogram part i: The electrocardiogram and its technology a scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology; the american college of cardiology foundation; and the heart rhythm society endorsed by the international society for computerized electrocardiology. *Journal of the American College of Cardiology*, 49(10):1109–1127, 2007.
- [37] Zhongyun Yuan, Jong Hak Kim, and Jun Dong Cho. Adaptive sampling for ecg detection based on compression dictionary. *JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE*, 13(6):608–616, 2013.
- [38] Patrick S. Hamilton and Willis J. Tompkins. Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database. *IEEE Transactions on Biomedical Engineering*, 1986.
- [39] Patrick S. Hamilton. A comparison of adaptive and nonadaptive filters for reduction of power line interference in the ecg. *IEEE Transactions on Biomedical Engineering*, 1996.
- [40] Chavdar Levkov, Georgy Mihov, Ratcho Ivanov, Ivan Daskalov, Ivaylo Christov, and Ivan Dotsinsky. Removal of power-line interference from the ecg: a review of the subtraction procedure. *BioMedical Engineering OnLine*, 2005.
- [41] P. A. Lynn. Online digital filters for biological signals: some fast designs for a small computer. *Med. & Biol. Eng & Comput.*, 1977.
- [42] John P. Abenstein and Willis J. Tompkins. A new data-reduction algorithm for real-time ecg analysis. *IEEE Transactions on Biomedical Engineering*, 29:43–48, 1982.
- [43] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, 1985.
- [44] Shubha Kadambe, Robin Murray, and G. Faye Bourdreaux-Bartels. Wavelet transform-based qrs complex detector. *IEEE Transactions on Biomedical Engineering*, 1999.
- [45] Masahiko Okada. A digital filter for the qrs complex detection. *IEEE Transactions on Biomedical Engineering*, 26, 1979.
- [46] Mohamed Elgendi, Björn Eskofier, Socrates Dokos, and Derek Abbott. Revisiting qrs detection methodologies for portable, wearable, battery-operated, and wireless ecg systems. *PLOS ONE*, 9, 2014.

- [47] David Salomon and Giovanni Motta. *Handbook of Data Compression*. Springer.
- [48] Ana Priscila Alves, Hugo Silva, André Lourenço, and Ana Fred. Bitalino: A biosignal acquisition system based on the arduino. 2012.
- [49] José Guerreiro, Raúl Martins, Hugo Silva, André Lourenço, and Ana Fred. Bitalino: A multimodal platform for physiological computing. In *Proc. of the 10th ICINCO Conf*, 2013.
- [50] Technical report tr-08-05. Technical report, Division of Engineering and Applied Sciences, Harvard University, 2005.
- [51] David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES 2004)*, 2004. Disponível em: <http://www.eecs.harvard.edu/mdw/papers/codeblue-bsno4.pdf>.
- [52] Aart Van Halteren, Richard Bults, Katarzyna Wac, Nicolai Dikovskiy, George Koprinkov, Ing Widya, Dimitri Konstantas, Val Jones, and Rainer Herzog. Wireless body area networks for healthcare: the mobihealth project. *Studies in Health Technology and Informatics*, 108:181–193, 2004.
- [53] Mobihealth. <http://www.mobihealth.com/home/en/home.php>. Consultado em: Dezembro de 2013.
- [54] ACW Wong, D McDonagh, O Omeni, C Nunn, M Hernandez-Silveira, and AJ Burdett. Sensium: An ultra-low-power wireless body sensor network platform: Design & application challenges. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 6576–6579. IEEE, 2009.
- [55] João P Silva Cunha, Bernardo Cunha, António Sousa Pereira, William Xavier, Nuno Ferreira, and Luis Meireles. Vital-jacket®: A wearable wireless vital signs monitor for patients' mobility in cardiology and sports. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, pages 1–2. IEEE, 2010.
- [56] Helena Fernández-López, José A Afonso, José Higinio Correia, and Ricardo Simões. Hm4all: a vital signs monitoring system based in spatially distributed zigbee networks. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, pages 1–4. IEEE, 2010.

- [57] K Montgomery, C Mundt, G Thonier, A Tellier, U Udoh, V Barker, R Ricks, L Giovanrandi, P Davies, Y Cagle, et al. Lifeguard—a personal physiological monitor for extreme environments. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 1, pages 2192–2195. IEEE, 2004.
- [58] Lifeguard. <http://www.nasa.gov/centers/ames/research/technology-onepagers/life-guard.html>. Consultado em: Dezembro de 2013.
- [59] Jason WP Ng, Benny PL Lo, Oliver Wells, Morris Sloman, Nick Peters, Ara Darzi, Chris Toumazou, and Guang-Zhong Yang. Ubiquitous monitoring environment for wearable and implantable sensors (ubimon). In *International Conference on Ubiquitous Computing (Ubicomp)*, 2004.
- [60] Ubimon. <http://www.doc.ic.ac.uk/vip/ubimon/home/index.html>. Consultado em: Dezembro de 2013.
- [61] Young-Dong Lee. *Wireless Vital Signs Monitoring System for Ubiquitous Healthcare with Practical Tests and Reliability Analysis*. PhD thesis, University of Oulu, 2010.
- [62] Atmel: Datasheet do microcontrolador atmega328p. Disponível em: www.atmel.com/Images/8161s.pdf, Outubro 2009.
- [63] Ti: Datasheet do microcontrolador msp430f1611. Disponível em: www.ti.com/lit/ds/symlink/msp430f1611.pdf, Outubro 2002.
- [64] Jennic: Datasheet do módulo jn5139-xxx-m00. Disponível em: www.jennic.com/files/.../JN-DS-JN5139MO-1v6.pdf.
- [65] Microchip: Datasheet do microcontrolador pic16f87x. Disponível em: ww1.microchip.com/downloads/en/DeviceDoc/30292D.pdf.
- [66] Maged N. Kamel Boulos, Steve Wheeler, Carlos Tavares, and Ray Jones. How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from ecalyx. *Biomedical Engineering OnLine*, 2011.
- [67] George B. Moody. *WFDB Applications Guide*. Harvard-MIT Division of Health Sciences and Technology, Março 2014.
- [68] Bob Kemp, Alpo Varri, Agostinho C. Rosa, Kim D. Nielsen, and John Gade. A simple format for exchange of digitized polygraphic recordings. *Electroencephalography and Clinical Neurophysiology*, 1992.
- [69] Biblioteca guava. Disponível em: <https://github.com/google/guava>. Consultado em: Abril de 2014.

[70] Biblioteca achartengine. Disponível em:
<https://code.google.com/p/achartengine/>. Consultado em:
Abril de 2014.