



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

PROGRAMA PARA DISPOSITIVOS MÓVEIS PARA COMUNICAÇÃO E CONTROLE DE UM ELETROCARDIÓGRAFO PORTÁTIL

Edvaldo Bezerra Pereira Júnior

Fortaleza – Ceará
Dezembro de 2010



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

PROGRAMA PARA DISPOSITIVOS MÓVEIS PARA COMUNICAÇÃO E CONTROLE DE UM ELETROCARDIÓGRAFO PORTÁTIL

Autor

Edvaldo Bezerra Pereira Júnior

Orientador

Prof. Dr. Paulo César Cortez

Co-orientador

Msc. Luiz Alves de Lima Neto

Proposta de Monografia apresentada à
Coordenação do Curso de Graduação em
Engenharia de Teleinformática da
Universidade Federal do Ceará como parte
dos requisitos para obtenção do título de
Engenheiro de Teleinformática.

Fortaleza – Ceará

2010

EDVALDO BEZERRA PEREIRA JÚNIOR

**PROGRAMA PARA DISPOSITIVOS MÓVEIS PARA COMUNICAÇÃO E
CONTROLE DE UM ELETROCARDIÓGRAFO PORTÁTIL**

Este trabalho foi julgado adequado para a obtenção do título de Engenheiro de Teleinformática e aprovado em sua forma final pelo Departamento de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

Edvaldo Bezerra Pereira Júnior

Banca Examinadora:

Dr. Paulo César Cortez

Orientador

DETI – UFC

Msc. Alexandre Moreira de Moraes

DETI – UFC

Msc. Jarbas Aryel Nunes da Silveira

DETI – UFC

Fortaleza, 16 de dezembro de 2010

Resumo

Uma boa forma de evitar a perda de vidas humanas decorrentes de doenças cardiovasculares é através do diagnóstico precoce das mesmas. Esse diagnóstico é, em muitos casos, realizado através de exames que detectam alterações no ritmo cardíaco. Um desses exames é o eletrocardiograma, que registra a variação dos potenciais elétricos gerados pela atividade elétrica do coração por meio de um equipamento denominado eletrocardiógrafo. Entretanto, em países como o Brasil, esse tipo de prevenção encontra dificuldades pela carência de profissionais capacitados, em localidades longínquas e de difícil acesso, em geral, em condições economicamente desfavoráveis. A comunicação móvel sofreu um grande avanço nas últimas décadas, particularmente a telefonia celular, popularizando o uso do aparelho celular na maioria dos países. Dessa forma, comunidades antes isoladas passaram a possuir cobertura da rede celular. Fazendo uso desta infraestrutura, os equipamentos médicos portáteis surgem como uma possível solução de problemas como os já citados anteriormente, disponibilizando os dados coletados para centros de excelência, onde médicos especialistas podem avaliar e diagnosticar possíveis patologias, de forma remota. Nesse contexto, este trabalho tem como objetivo desenvolver um programa para dispositivos móveis, capaz de controlar, via *Bluetooth*, um sistema de aquisição de biopotenciais cardíacos e de orientar o correto posicionamento dos eletrodos deste sistema, exibir o sinal em tempo de aquisição, obtido via *Bluetooth*, e permitir o armazenamento ou o envio dos exames para uma central localizada em um hospital ou consultório médico através da interface GPRS ou 3G do dispositivo móvel. Os resultados obtidos através de testes feitos por especialistas indicam que a abordagem proposta é capaz de prover eficiência e praticidade no diagnóstico precoce de doenças cardiovasculares. Dessa forma, conclui-se que os *softwares* médicos para dispositivos móveis se mostram como uma solução simples e de baixo custo na construção de instrumentação médica.

Palavras-chave: Bluetooth, celular, eletrocardiograma, potenciais elétricos, tecnologia.

Abstract

A good way to avoid losses of human lives due to cardiovascular diseases is through early diagnosis of them. This diagnosis is, in many cases, made through tests that detect changes in the heart rhythm. One of these tests is the electrocardiogram which records the variation of electrical potentials generated by heart electrical activity through a device called electrocardiograph. However, in countries such as Brazil, this type of prevention is experiencing difficulties related to the lack of trained professionals to perform their functions in distant and inaccessible places, generally in bad economic conditions. The mobile communication, particularly mobile phone, has undergone great advances in the last decades what allowed the popularization of mobile phone in most countries. This way, isolated communities started to receive cellular network coverage. Making use of this infrastructure, the portable medical equipments appear as one possible solutions for problems such as those problems cited above. These instruments are responsible to make available collected information to centers of excellence, where specialists could consider and diagnose possible pathologies on the remote way. In this context, this work has the aim of developing a program for mobile devices that could be able to control, via Bluetooth, an acquisition system of heart biopotentials and advice on finding the correct positioning of the electrodes of this system; display the signal on acquisition time, obtained via Bluetooth, and allow the storage or transition of the examination for a central located in a hospital or doctor's office through the interface GPRS or 3G in the mobile device. The results obtained through tests made by specialists indicate that the proposed approach is capable of providing efficiency and practicality in the early diagnosis of cardiovascular diseases. Thus, it is possible to conclude that medical softwares for mobile devices show us how they could be a simple solution with low costs in construction of medical instrumentation.

Keywords: Bluetooth, cellular phone, electrocardiogram, electrical potentials, technology.

Dedicatória

Dedico este trabalho aos meus pais Edvaldo e Magna, meu irmão Rodrigo e minha namorada Lígia pelo amor e apoio incondicional.

Agradecimentos

A Deus, acima de tudo, por todas as conquistas.

Aos meus pais Edvaldo Bezerra e Tereza Magna, por terem me dado condições de estudar e crescer profissionalmente e socialmente, além do amor e apoio por toda a vida.

Ao meu irmão Rodrigo pelo apoio sempre.

A minha namorada Lígia, que está sempre ao meu lado em qualquer situação.

Ao meu orientador, Prof. Dr. Paulo César Cortez, pela orientação e ajuda, fundamentais para a realização deste trabalho.

Aos amigos da faculdade, por sempre me apoiarem nos momentos de insegurança, em especial: Daniel Bruno, Davyd Bandeira, João Gabriel, Mariana Bento, Raul Holanda.

Aos companheiros que conheci no GIQ (Grupo Informação Quântica), GREat (Grupo de Redes e Engenharia de *Softwares* e Sistemas) e Atlântico, Prof. Dr. Rubens Viana Ramos, João batista, Marco Antonio, Fábio Alencar, Davi Sena, Daniel Barbosa, José Cláudio, João Luzeilton, Paulo Benício, Prof. Dr. Rossana Maria, Marcio Maia, Rute Nogueira, Clayson Sandro, Luíz Correia, Luiz Alves, Marcos Aurélio, Adão Justo, Raphael Amorim, Pedro Henrique, David Mota, Alex Vasconcellos, pelo conhecimento e momentos compartilhados.

“A vitória pertence ao mais perseverante”.

Napoleão Bonaparte

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Lista de Siglas.....	xii
1 INTRODUÇÃO	1
1.1 Motivação.....	2
1.2 Objetivos	4
1.1.1 Objetivo geral.....	4
1.1.2 Objetivos específicos	4
1.3 Organização do trabalho.....	4
1.4 Produção Científica	5
2 NOÇÕES BÁSICAS DE ELETROCARDIOGRAFIA.....	6
2.1 Fisiologia do Coração.....	6
2.2 Biopotenciais Cardíacos	8
2.2.1 Potenciais de Ação	8
2.3 Excitação Rítmica do Coração	10
2.3.1 O Nodo Sinusal	11
2.3.2 Vias Internodais	11
2.3.3 Nodo Atrioventricular	12
2.3.4 Feixe AV	12
2.3.5 Sistema de Purkinje.....	12
2.4 Ciclo Cardíaco.....	14
2.5 Eletrocardiograma	14
2.6 Registro Eletrocardiográfico	16
3 JAVA MICRO EDITION	17
3.1 Visão Geral.....	17
3.1.1 Configurações	18
3.1.2 Perfis	20
3.1.3 APIs Opcionais.....	20
3.1.4 Máquinas Virtuais Java	20

3.2	Perfil MIDP	21
3.3	MIDlets.....	23
3.4	MIDlets Suite	25
3.5	JAR.....	25
3.6	JAD.....	25
3.7	APIs Opcionais.....	26
3.8	API Bluetooth.....	27
3.8.1	Troca de mensagens entre dispositivos Bluetooth	27
3.8.2	JSR-82	28
3.9	JSR-75 (File Connection API)	29
3.10	Frameworks	29
3.10.1	LWUIT	30
4	METODOLOGIA DE IMPLEMENTAÇÃO	32
4.1	Comparativo entre as linguagens de desenvolvimento móvel	32
4.2	Visão geral do sistema.....	33
4.3	Arquitetura do sistema.....	33
4.4	Descrição do hardware do protótipo	34
4.5	Descrição do firmware do protótipo.....	35
4.6	Descrição do <i>software</i>	36
4.7	Processo de <i>software</i>	37
4.7.1	Definição de requisitos.....	38
4.7.2	Projeto de sistema e de software	40
4.7.3	Implementação e testes de unidade	43
4.7.4	Integração e teste de sistema	44
4.7.5	Operação e manutenção	45
4.8	Análise dos casos de uso	45
4.8.1	Realizar Exame	46
4.8.2	Cadastrar Paciente	47
4.8.3	Configurar aquisição	48
4.8.4	Visualizar exame	48
4.8.5	Transmitir Exame.....	49
5	RESULTADOS, CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS	50
5.1	Resultados	50

5.2	Conclusões.....	53
5.3	Contribuições.....	54
5.4	Trabalhos Futuros.....	55
Apêndice A		56
Apêndice B		58
Referências Bibliográficas.....		62

Lista de Figuras

Figura 2.1: visão geral do coração.....	7
Figura 2.2: natureza interligada do músculo cardíaco (GUYTON, 1997).....	8
Figura 2.3: potencial de ação geral na célula cardíaca e seus principais canais iônicos (GUYTON, 1997).	9
Figura 2.4: distribuição das fibras que controlam as contrações cardíacas (GUYTON; HALL, 2006)....	10
Figura 2.5: sistema de condução do coração.	13
Figura 2.6: principais componentes do ECG (CARLOS et al., 2003).	15
Figura 2.7: papel de registro eletrocardiográfico e tipos de calibração, respectivamente.	16
Figura 3.1: hierarquia dos elementos da plataforma Java ME.....	18
Figura 3.2: edições Java e suas respectivas máquinas virtuais.	21
Figura 3.3: ciclo de vida do MIDlet e transições entre os estados.....	24
Figura 3.4: seqüência de operações para comunicação Bluetooth.	28
Figura 3.5: código que retorna a versão da JSR-75 suportada pelo dispositivo.	29
Figura 3.6: exemplos de temas suportados pelo LWUIT.....	31
Figura 3.7: comparação entre protótipo e implementação da tela de cadastro de paciente, respectivamente.	Erro! Indicador não definido.
Figura 4.1: comunicação Bluetooth entre o sistema móvel e o eletrocardiógrafo portátil.	34
Figura 4.2: diagrama de blocos dos protótipos desenvolvidos por LIMA NETO (2010).....	34
Figura 4.3: fluxograma básico do firmware (LIMA NETO, 2010).....	35
Figura 4.4: descrição geral do modelo de processo: modelo em cascata.	38
Figura 4.5: diagrama de Caso de Uso do sistema.	40
Figura 4.6: sistema dividido em subsistemas.....	41
Figura 4.7: protótipo de três telas feitos à mão livre: menu principal, cadastro de pacientes e exame realizado com sucesso, respectivamente.	41
Figura 4.8: organograma simplificado das telas da aplicação.....	42
Figura 4.9: visão geral de implementação.	43
Figura 4.10: processo de depuração de erros (SOMMERVILLE, 2007).....	44
Figura 4.11: tela inicial do sistema e menu principal, respectivamente.....	45
Figura 4.12: telas de seleção de derivação e posicionamento dos eletrodos.	46
Figura 4.13: telas de sucesso e falha no teste dos eletrodos, respectivamente.	46
Figura 4.14: visualização do exame no celular.....	47
Figura 4.15: telas de cadastro de pacientes, anamnese e exame.....	48
Figura 4.16: tela de escolha de frequência de amostragem.....	48
Figura 4.17: telas de busca de paciente, paciente encontrado e paciente não encontrado, respectivamente.	49
Figura 5.1: instalador do sistema proposto.	50
Figura 5.2: biopotencial digitalizado e processado pelo front-end da aplicação.....	51
Figura 5.3: mesmo exame sendo visualizado no emulador e no celular.	52
Figura 5.4: sinal visualizado no PC.....	52

Lista de Tabelas

Tabela 4.1: comparação entre plataformas e linguagens de programação suportadas por dispositivos móveis disponíveis no mercado.....	32
Tabela 4.2: tabela de comandos do firmware (LIMA NETO,2010).....	36
Tabela 4.3: exemplos de Requisitos funcionais e não funcionais do sistema proposto.....	39

Lista de Siglas

ABNT	Associação Brasileira de Normas e Técnicas
ANATEL	Agência Nacional de Telecomunicações
ANVISA	Agência Nacional de Vigilância Sanitária
API	<i>Application Programming Interface</i>
AV	Atrioventricular
AVC	Acidete Vascular Cerebral
CDC	<i>Connected Device Configuration</i>
CLDC	<i>Connected, Limited Device Configuration</i>
CPF	Cadastro de Pessoas Físicas
ECG	Eletrocardiograma
GPRS	<i>General Packet Radio Service</i>
IDE	<i>Integrated Development Enviropment</i>
Java ME	<i>Java Micro Edition</i>
JAD	<i>Java Application Descriptor</i>
JAR	<i>Java Archive</i>
L2CAP	<i>Logical Link Control and Adaptation Protocol</i>
LWUIT	<i>Lightweight UI Toolkit</i>
MIDP	<i>Mobile Information Device Profile</i>
OBEX	<i>Object EXchange</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>
SA	Sinoatrial
SPP	<i>Serial Port Profile</i>
SUS	Sistema Único de Saúde

UTI Unidade de Tratamento Intensivo

WPAN *Wireless Personal Area Networks*

CAPÍTULO 1

INTRODUÇÃO

As doenças cardiovasculares são uma das principais causas de morte em todo o mundo, segundo a OMS (2007), sendo responsável por 17,5 milhões de óbitos a cada ano. Para se dimensionar o impacto em todo o mundo, 7,6 milhões de pessoas morrem de ataque cardíaco e 5,7 milhões morrem de AVC a cada ano. Segundo a AHA (2009), somente nos EUA, 35% de todas as mortes (cerca de uma em cada três) são por doenças cardiovasculares. Na Europa tais doenças representam o equivalente a 48% das mortes. Já no Brasil, esse universo chega a 47,8% aparecendo, assim, em 9º lugar na lista dos países com maior incidência de morte por complicações cardiovasculares, segundo dados do DATASUS (2007).

Com a observação de tais dados, é fácil entender a motivação dos pesquisadores em compreender o funcionamento do coração e registrar sua atividade. A primeira forma desse registro, há mais de cem anos, foi através do eletrocardiograma de repouso. Trabalho pioneiro do holandês Willem Einthoven (1901), colocando eletrodos (placas de metal) na superfície do corpo, ligadas por fios elétricos a um galvanômetro que captava os potenciais elétricos do coração. Nestes mais de 100 anos muitas foram as transformações, sofisticações e aplicações do eletrocardiograma: nos equipamentos de ergometria para o teste de esforço, na monitorização ambulatorial (Holter) e nos sistemas de avaliação das arritmias cardíacas (defeitos do ritmo).

Como todos os métodos, o eletrocardiograma teve o seu apogeu nos primeiros 50 anos com todo o desenvolvimento tecnológico dos métodos correlacionados. No momento em que o mundo do diagnóstico clínico-cardiológico achava que o exame não sobreviveria, a informática, a eletrofisiologia e a biologia molecular cresceram, se desenvolveram e o eletrocardiograma ganhou mais informações e mais agilidade na execução e na transmissão dos traçados.

Às suas reconhecidas propriedades para diagnosticar os casos agudos, como o infarto do miocárdio e as arritmias cardíacas, foram agregadas novas observações e padrões eletrocardiográficos, valorizando mais ainda o método centenário. Desta maneira a utilização dos exames diagnósticos baseados no eletrocardiograma ganhou várias novas frentes, como o desenvolvimento dos estudos da variabilidade da frequência cardíaca, do eletrocardiograma de alta resolução, do mapeamento de superfície, da dispersão da repolarização, entre outros (PASTORE, 2010).

Ao mesmo tempo cresceram as doenças cardiovasculares, como a insuficiência cardíaca, as arritmias cardíacas e as mortes súbitas por doença do coração, exigindo diagnósticos mais rápidos, precisos e objetivos. De novo, as evidências clínicas mostraram que um eletrocardiograma realizado precocemente pode ser um alerta para as doenças raras e pouco diagnosticadas, mas que podem ser fatais: Síndrome de Brugada, o intervalo QT longo e curto, a Miocardiopatia Hipertrófica e a Cardiomiopatia Arritmogênica do Ventrículo Direito (PASTORE, 2010).

Com o advento dos computadores digitais e a miniaturização dos componentes eletrônicos, a Eletrocardiografia passou por uma nova revolução. Os equipamentos estão se tornando cada vez menores e mais versáteis. Em consequência disso, os diagnósticos estão se tornando cada vez mais precoces, o que acarreta na diminuição dos óbitos. Isso colabora para que haja mais pesquisa e desenvolvimento de eletrocardiógrafos que ultrapassem as fronteiras dos hospitais e passem a ser utilizados em residências ou de forma portátil, integrado com os telefones e PDAs.

1.1 Motivação

As doenças cardiovasculares aparecem em primeiro lugar entre as causas de morte no Brasil e representam quase um terço dos óbitos totais e 65% do total de mortes na faixa etária de 30 a 69 anos de idade, atingindo a população adulta em plena fase produtiva (DATASUS, 2007). No Sistema Único de Saúde (SUS), essas doenças foram responsáveis, em 2002, por mais de 1,2 milhão de internações, representaram 10,3% do total de internações e 17% dos gastos financeiros (ARAÚJO; FERRAZ, 2005).

Segundo o DATASUS (2007), no ano de 2002, ocorreram 1.216.394 internações decorrentes de doenças do aparelho circulatório, representando 10,3% do total das internações

no Sistema Único de Saúde (SUS). Em relação ao valor financeiro, a parcela das internações em cardiologia clínica e cirúrgica correspondeu a 17% do total, superando todos os outros grupos de especialidades isoladamente.

Outro problema bastante comum no Brasil é a grande desigualdade na distribuição geográfica dos médicos. Segundo PÓVOA (2004), estes, em geral, estão nos estados e nas regiões com maior desenvolvimento econômico. Enquanto o Nordeste possui 28% da população do Brasil e conta com apenas 16% do total de médicos, o Sudeste possui aproximadamente 42% da população brasileira e quase 60% dos médicos. Esta desigualdade torna-se mais evidente quando se olha a distribuição entre os Estados. O Rio de Janeiro, por exemplo, conta com 8% da população do Brasil e concentra 20% dos médicos.

Diante desses quadros, qualquer iniciativa que leve a um diagnóstico precoce das doenças cardiovasculares, à aderência ao tratamento e à adoção de medidas preventivas, além da atuação de profissionais da saúde, tais como enfermeiros e agentes de saúde que estejam dispostos a prestar seus serviços em localidades mais desfavorecidas em relação à existência de médicos é economicamente justificada.

Paralelamente, a explosão de oferta de mobilidade está abrindo portas para uma nova era, a era do "*all mobile*", na qual os dispositivos móveis (celulares, PDAs, etc) passam a ter um papel central na vida das pessoas e, principalmente, das corporações. Esse cenário é reforçado, como em um círculo virtuoso, pelo elevado número de serviços e aplicações multimídia que estão surgindo no mercado, unindo streaming de áudio e vídeo, acesso à Internet em banda larga, serviços de localização, jogos 3D multiusuários, além das mais diversas aplicações (CPQD, 2008).

Neste contexto, surgem os equipamentos médicos portáteis, de simples manipulação por enfermeiros e agentes de saúde, capazes de disponibilizar os dados coletados para centros de excelência, onde médicos especialistas podem avaliar e diagnosticar possíveis patologias, à distância.

O reconhecimento dos padrões eletrocardiográficos pelos clínicos, cardiologistas e afins é uma ferramenta muito útil nos consultórios, emergências e UTIs.

No presente trabalho apresenta-se uma proposta que objetiva auxiliar na solução dos problemas citados, através do desenvolvimento de um sistema de *software* a ser integrado ao protótipo de eletrocardiógrafo portátil de LIMA NETO (2010). Este sistema controla o protótipo via Bluetooth, além de permitir o cadastro de pacientes, visualização,

armazenamento e respectivo envio dos exames através da interface GPRS ou 3G do dispositivo móvel.

Dentre as contribuições desse trabalho destacam-se a possibilidade de operação do equipamento por agentes de saúde, facilidade de integração do celular ou PDA que possua interface Bluetooth com o eletrocardiógrafo, além da alta portabilidade do sistema e custo baixo.

1.2 Objetivos

1.1.1 Objetivo geral

Objetiva-se através deste trabalho desenvolver um programa para dispositivos móveis que se integre a um eletrocardiógrafo portátil, permitindo seu controle e o recebimento dos dados dos exames coletados pelo aparelho através da interface Bluetooth.

1.1.2 Objetivos específicos

Durante o desenvolvimento deste trabalho, outros objetivos devem ser atingidos: permitir o cadastro de paciente, armazenamento e envio dos exames a partir da interface GPRS do celular ou PDA; criar interfaces amigáveis com o usuário que viabilizem a operação por enfermeiros e/ou agentes de saúde; diminuir os custos do equipamento para aumentar seu uso, principalmente na rede pública; e aumentar a portabilidade e integração com novas tecnologias para que o diagnóstico se torne cada vez mais rápido, diminuindo, assim, o número de óbitos e gastos do governo.

1.3 Organização do trabalho

O presente trabalho está dividido em cinco Capítulos. Apresenta-se no Capítulo 2 os princípios básicos de Eletrocardiografia, fundamental para o entendimento das seções posteriores. Já no Capítulo 3 descreve-se a plataforma Java *Micro Edition*, usada na aplicação, além de sua arquitetura e tecnologias. No Capítulo 4 desenvolve-se a metodologia de implementação e descreve toda a arquitetura do sistema de aquisição de biopotenciais

cardíacos proposto neste trabalho, incluindo hardware, firmware e *software*. Por fim, apresentam-se no Capítulo 5 os resultados produzidos e conclusões deste trabalho, além das contribuições e dos trabalhos futuros que podem ser realizados a partir deste.

1.4 Produção Científica

No período de desenvolvimento do projeto deste trabalho foi produzido o seguinte artigo científico publicado em congresso nacional:

LIMA NETO, L. A.; CORTEZ, P. C.; **PEREIRA JUNIOR, E. B.**; CUNHA, M. A. P. Sistema Móvel de Aquisição de Eletrocardiograma com Comunicação Bluetooth E Transmissão por GPRS ou 3G. In: Proceedings of CBIS, 2010, Porto de Galinhas – PE. XVII Congresso Brasileiro de Informática em Saúde, 2010.

LIMA NETO, L. A.; CORTEZ, P. C.; **PEREIRA JUNIOR, E. B.**; CUNHA, M. A. P. Protótipo de Eletrocardiógrafo Portátil com Uma Derivação e Comunicação Com Celular Via Bluetooth. In: Proceedings of CBEB, 2010, Tiradentes - MG. XXII Congresso Brasileiro de Engenharia Biomédica, 2010.

CAPÍTULO 2

NOÇÕES BÁSICAS DE ELETROCARDIOGRAFIA

O eletrocardiograma é uma ferramenta imprescindível na investigação de queixas comuns de pacientes em prontos-socorros e consultórios. Fazer com que esse instrumento se torne cada vez mais integrado com as novas tecnologias é essencial para que se torne mais acessível a todos. Objetiva-se através desse capítulo oferecer um breve embasamento teórico indispensável à compreensão das grandezas estudadas nos capítulos posteriores a cerca da eletrocardiografia.

2.1 Fisiologia do Coração

O coração é uma bomba de sangue que tem como função principal impulsioná-lo de forma que o mesmo atinja a todos os pontos do corpo, transportando consigo oxigênio e nutrientes (GUYTON; HALL, 2006).

O coração é constituído por duas bombas distintas: o coração direito, que bombeia o sangue pelos pulmões, e o coração esquerdo, que bombeia o sangue pelos órgãos periféricos. Cada um desses corações distintos, por sua vez, é uma bomba pulsátil de duas câmaras composta de um átrio e um ventrículo. O átrio funciona principalmente como reservatório de sangue e como via de entrada para o ventrículo, mas também bombeia fracamente para ajudar a levar o sangue até o ventrículo. O ventrículo, por sua vez, é a principal fonte da força que impulsiona o sangue pela circulação pulmonar (ventrículo direito) ou pela periférica (ventrículo esquerdo) (GUYTON, 1988).

Outra característica importante: o coração possui também quatro válvulas distintas, que facilitam o fluxo de sangue para frente, impedindo seu refluxo. Duas válvulas, as atrioventriculares, localizadas no coração, permitem um fluxo unidirecional do sangue do átrio direito para o ventrículo direito (válvula tricúspide) e do átrio esquerdo para o ventrículo

esquerdo (válvula mitral ou bicúspide). As válvulas semilunares (válvula pulmonar, coração direito e válvula aórtica, coração esquerdo) estão localizadas na parede arterial imediatamente por fora do coração e a função é impedir o retrocesso do sangue para dentro do coração entre uma contração e outra (GUYTON, 1997).

O sangue que é conduzido pelas grandes veias, ao atingir o átrio direito, é forçado pela contração atrial a passar pela válvula tricúspide enchendo o ventrículo direito. Este por sua vez, bombeia o sangue através da válvula pulmonar para a artéria pulmonar, daí para os pulmões e, finalmente, pelas veias pulmonares para o átrio esquerdo. Já a contração do átrio esquerdo força a passagem de sangue para o ventrículo esquerdo através da válvula mitral. Do ventrículo esquerdo, o sangue passa pela válvula aórtica, atinge a aorta e, por essa artéria, percorre toda a circulação sistêmica. Uma visão geral do coração é exibida na Figura 2.1 (GUYTON, 1988; LIMA NETO, 2010).

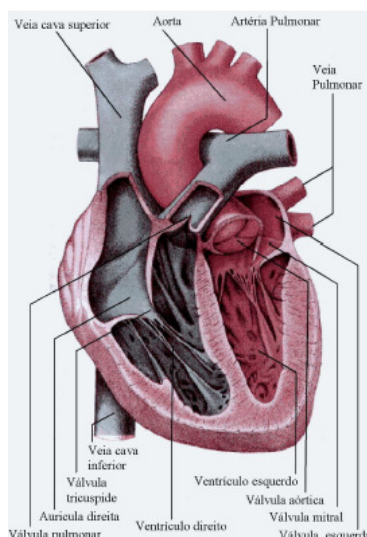


Figura 2.1: visão geral do coração.

Em muitos aspectos, o músculo cardíaco é bastante semelhante ao músculo esquelético, mas possui duas características que o tornam muito adaptado à ação bombeadora do coração. Primeira, as fibras do músculo cardíaco são interconectadas entre si, de modo que um potencial de ação com origem em qualquer ponto da massa muscular pode se propagar por toda a sua extensão e fazer com que toda a massa contraia a um só tempo. Isso permite que o músculo cardíaco da parede de cada uma das câmaras contraia ao mesmo tempo e, portanto, empurre o sangue para adiante. Segunda, o potencial de ação do músculo cardíaco dura por cerca de três décimos de segundo, o que é dez ou mais vezes maior que a do potencial de ação da maioria dos músculos esqueléticos. Portanto, a duração da contração do músculo cardíaco

também é da ordem de três décimos de segundos, que representa o tempo necessário para que o sangue flua dos ventrículos para as artérias (GUYTON, 1988).

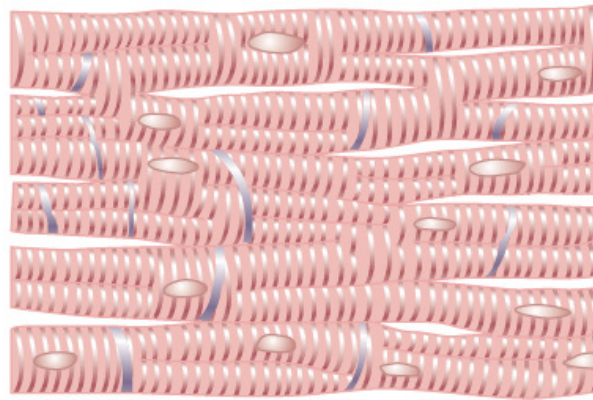


Figura 2.2: natureza interligada do músculo cardíaco (GUYTON, 1997).

O coração é constituído por dois sincícios distintos: o sincício atrial, que constitui as paredes dos dois átrios, e o sincício ventricular, que constitui as paredes dos dois ventrículos. Os átrios são separados dos ventrículos por um tecido fibroso que circunda as aberturas valvulares entre os átrios e os ventrículos. Normalmente, os potenciais de ação só podem ser conduzidos do sincício atrial para o ventricular por meio do sistema de condução especializado, o feixe A-V. Essa divisão da massa muscular do coração em dois sincícios funcionais distintos possibilita que os átrios se contraiam um pouco antes da contração ventricular, o que é importante para a eficácia do bombeamento cardíaco (GUYTON, 1988).

2.2 Biopotenciais Cardíacos

Existem potenciais elétricos através das membranas de praticamente todas as células do corpo humano. Além disso, algumas células são excitáveis, ou seja, capazes de gerar rapidamente variações nos impulsos eletroquímicos em suas membranas, destacando-se os fenômenos elétricos nas células cardíacas (GUYTON, 1988; LIMA NETO, 2010).

2.2.1 Potenciais de Ação

A concentração de íons no interior de uma célula é diferente da concentração no seu exterior, o que propicia a geração de uma diferença de potencial denominada “potencial de membrana” (GUYTON, 1997). Uma célula possui vários íons no seu interior, tais como Sódio

(Na^+), Potássio (K^+) e Cálcio (Ca^{2+}). No seu estado normal, as concentrações de (K^+) são maiores no interior da célula, enquanto as concentrações de (Na^+) e (Ca^{2+}) são maiores no exterior, produzindo uma diferença de potencial que, medida do interior de uma membrana de repouso, é de aproximadamente -90mV (WEBSTER, 1998; Siqueira, 2005).

A célula pode ser considerada fora da situação de repouso quando sofre uma despolarização superior ao seu limiar de ação, ou seja, ocorre uma variação brusca do potencial de repouso negativo para um potencial positivo, que termina com um retorno igualmente rápido ao potencial negativo. Essa cadeia de variações de potencial é conhecida como “potencial de ação”. (GUYTON, 1997).

Segundo Berne (2000), as etapas sucessivas do potencial de ação podem ser esquematizadas como apresentado abaixo, de acordo com a Figura 2.3.

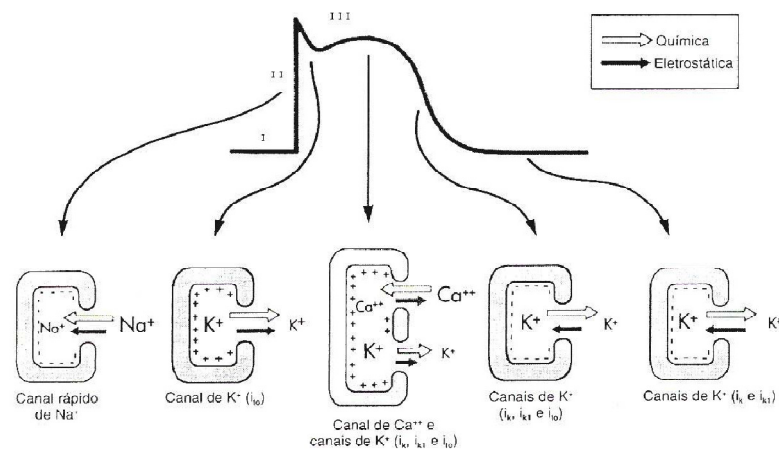


Figura 2.3: potencial de ação geral na célula cardíaca e seus principais canais iônicos (GUYTON, 1997).

A etapa de repouso corresponde à situação estática, associada ao potencial de repouso da membrana (-90mV). Na etapa de despolarização a membrana rapidamente torna-se altamente permeável aos íons sódio, permitindo o fluxo intenso de íons carregados positivamente para dentro da fibra muscular, o que eleva rapidamente o potencial na direção da positividade, ocorre então, a despolarização. Esse processo segue até atingir o limiar de ação, gerando o potencial de ação. Já na etapa de repolarização, milésimos de segundos após a membrana ter ficado permeável aos íons sódio, os canais de sódio começam a se fechar e os canais de potássio se abrem, ainda mais. A rápida difusão de K^+ para o exterior restabelece o potencial negativo da membrana.

2.3 Excitação Rítmica do Coração

O coração é provido de um sistema especializado para a geração de impulsos rítmicos, para causar a contração rítmica do músculo cardíaco, e para a condução rápida desses impulsos por todo o coração. Quando esse sistema funciona normalmente, os átrios se contraem cerca de um sexto de segundo antes da contração ventricular, o que possibilita maior enchimento dos ventrículos antes que eles bombeiem o sangue pelos pulmões e pela circulação periférica. Outra importância especial do sistema é que ele possibilita que todas as partes dos ventrículos se contraíam simultaneamente, o que é essencial para a geração efetiva de pressão nas câmaras ventriculares.

Apresenta-se na Figura 2.4 o sistema especializado de excitação e condução do coração que controla as contrações cardíacas. É mostrado nessa figura: (1) nodo sinusal (também denominada nodo sinoatrial ou nodo SA), no qual é gerado o impulso rítmico normal e vias internodais que conduzem o impulso do nodo sinusal para o nodo A-V; (2) nodo A-V (também conhecido como nodo atrioventricular), no qual o impulso dos átrios sofre retardo antes de passar para os ventrículos; (3) feixe A-V, que conduz o impulso dos átrios para os ventrículos e (4) feixes esquerdo e direito das fibras de purkinje, que conduzem o impulso cardíaco a todas as partes dos ventrículos. (GUYTON, 1997).

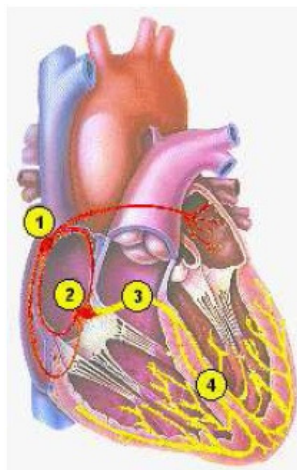


Figura 2.4: distribuição das fibras que controlam as contrações cardíacas (GUYTON; HALL, 2006).

2.3.1 O Nodo Sinusal

Muitas fibras cardíacas têm a capacidade de auto-excitação, um processo que pode ocasionar contrações rítmicas automáticas. Isto é particularmente verdadeiro para as fibras do sistema especializado de condução do coração. A parte desse sistema que apresenta o maior grau de auto-excitação são as fibras do nodo sinusal. Este está localizado na parede superior lateral do átrio direito, imediatamente abaixo e lateral à abertura da veia cava superior. O nodo sinusal controla normalmente a frequência de batimento de todo o coração.

As membranas do nodo SA são muito permeáveis ao Sódio. Assim uma grande quantidade de íons de Sódio penetra o interior da fibra, produzindo um desvio contínuo no potencial de repouso da membrana para um valor mais positivo. Quando o potencial da membrana atinge um nível crítico, chamado de valor “limiar”, é produzido um potencial de ação, o que ocorre abruptamente. Ao término desse potencial de ação, a membrana torna-se temporariamente menos permeável aos íons de Sódio, mas, ao mesmo tempo, mais permeável aos íons de Potássio. A saída desses íons transportando cargas positivas para o exterior, faz com que o potencial da membrana atinja o seu valor mais negativo, caracterizando o estado de hiperpolarização devido a perda excessiva de cargas positivas. Essa condição persiste por uma fração de segundos, e logo depois desaparece, visto que as permeabilidades aos íons de Sódio e Potássio retornam aos seus valores normais, quando a maior permeabilidade ao Sódio da membrana faz com que ocorra outro potencial de ação. Esse processo perdura sem interrupção por toda a vida, levando a excitação rítmica das fibras do nodo AV (GUYTON, 1988; LIMA NETO, 2010).

2.3.2 Vias Internodais

As extremidades das fibras do nodo sinusal fundem-se às fibras musculares atriais circundantes e os potenciais de ação que se originam no nodo sinusal dirigem-se para adiante, por meio dessas fibras. Desse modo, o potencial de ação se propaga por toda a massa muscular atrial, e acaba por chegar também ao nodo AV após cerca de 0,04 segundos da partida do impulso do nodo AS (GUYTON, 1997).

2.3.3 *Nodo Atrioventricular*

O sistema de condução é organizado de tal forma que o impulso cardíaco não passa dos átrios para os ventrículos de modo demasiado rápido; isso dá tempo para os átrios lançarem seu conteúdo nos ventrículos antes que se inicie a contração ventricular. São principalmente o nodo AV e suas fibras de condução associadas que retardam essa transmissão do impulso cardíaco dos átrios para os ventrículos. A causa da condução extremamente lenta tanto nas fibras de transição como nas fibras nodais é, em parte, que seu tamanho é consideravelmente menor do que o das fibras musculares atriais normais. Entretanto, a maior parte da condução lenta é provavelmente causada por dois fatores totalmente diferentes. Em primeiro lugar, todas essas fibras têm potenciais de membrana em repouso que são muito menos negativos que o potencial de repouso normal do restante do músculo cardíaco. Segundo, há grande resistência à condução de íons excitatórios de uma fibra para outra. Assim, havendo tanto baixa voltagem impulsionando os íons como grande resistência ao seu movimento, é fácil ver-se por que cada fibra sucessiva demora a ser excitada. O impulso, ao chegar ao nodo AV, demora 0,12 segundos para seguir em frente e atingir o Feixe AV (GUYTON; HALL, 2006).

2.3.4 *Feixe AV*

Uma característica especial do feixe A-V é a incapacidade dos potenciais de ação, exceto em estados anormais, fazerem o trajeto retrógrado no feixe dos ventrículos para os átrios. Isso impede a reentrada de impulsos cardíacos dos ventrículos para os átrios por essa via, possibilitando apenas a condução dos átrios para os ventrículos (GUYTON, 1997).

2.3.5 *Sistema de Purkinje*

As *fibras de Purkinje* saem do nodo A-V para os ventrículos pelo feixe A-V. Elas têm características funcionais bem opostas às das fibras do nodo A-V; são fibras muito grandes, maiores ainda que as fibras musculares normais dos ventrículos, e transmitem potenciais de ação com velocidade cerca de 5 vezes maior que a verificada no músculo cardíaco habitual e 150 vezes maior que a medida em algumas fibras de transição. Isso possibilita a transmissão quase que imediata do impulso cardíaco por todo o sistema ventricular.

O feixe A-V se divide nos ramos esquerdo e direito. Cada ramo desce até o ápice do ventrículo, dividindo-se em ramos menores que circundam cada câmara ventricular e voltam

em direção à base do coração. Do momento em que o impulso cardíaco chega aos ramos até atingir as terminações das fibras de Purkinje, o tempo total transcorrido é de apenas 0,03 s; assim sendo, uma vez tendo entrado no sistema de Purkinje, o impulso cardíaco dissemina-se quase que imediatamente para toda a superfície endocárdica do músculo ventricular.

Com a chegada dos impulsos no sincício ventricular, rapidamente e com uma grande força, ocorre a contração de todas as suas fibras. A contração das câmaras ventriculares reduz acentuadamente o volume das mesmas, o que faz com que um considerável volume de sangue seja ejetado, do ventrículo direito para a artéria pulmonar e, do ventrículo esquerdo para a artéria aorta. (GUYTON; HALL, 2006; NETO, 2010).

A Figura 2.5 mostra o sistema especializado excitatório do coração, bem como suas estruturas responsáveis e/ou participantes. E, também, os potenciais de ação transmembrana dos nodos sinusal e átrio ventricular, além da atividade elétrica registrada por outras áreas do sistema de condução e músculos atrial e ventricular (GUYTON, 1997).

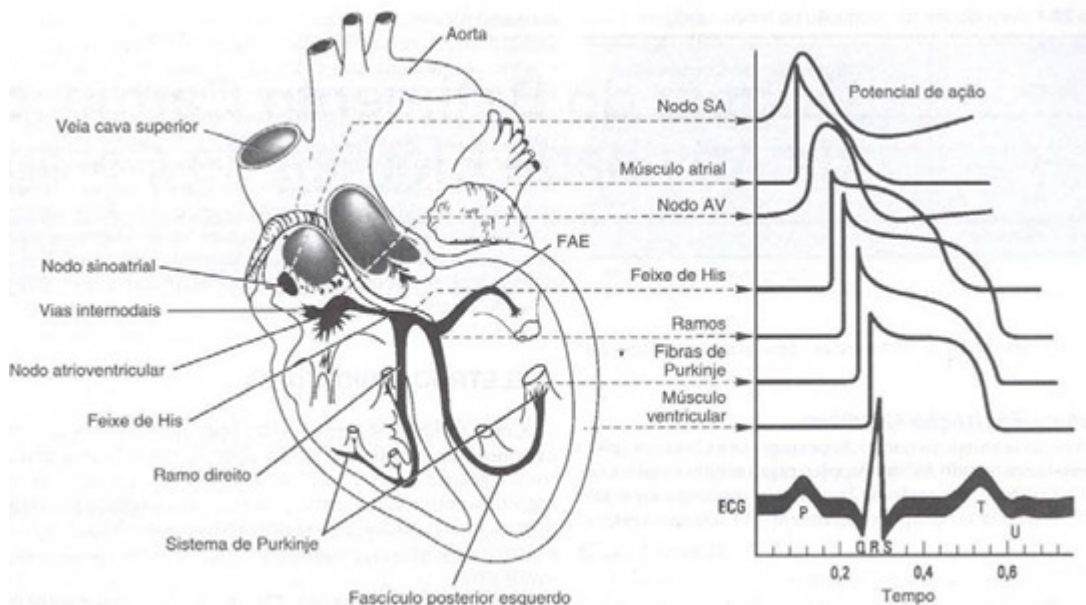


Figura 2.5: sistema de condução do coração.

2.4 Ciclo Cardíaco

No nodo sinusal (fibra excitatória) é originado um potencial de ação que inicia todo o ciclo cardíaco. Este potencial propaga-se através dos átrios até encontrar os feixes Átrio-Ventricular (A-V). Enquanto o potencial propaga-se pelos átrios, o átrio esquerdo recebe sangue oxigenado dos pulmões e passa o fluxo para o ventrículo esquerdo. Do mesmo modo o átrio direito recebe sangue venoso do sistema circulatório e passa o fluxo para o ventrículo direito. Neste momento a válvula A-V está aberta, a válvula aórtica e a válvula pulmonar estão fechadas. No feixe A-V ocorre um atraso na propagação do potencial para que os átrios possam contrair-se antes dos ventrículos e assim preenchê-los em sua forma máxima. Após a contração dos átrios as válvulas A-V fecham-se, o potencial segue propagando-se através dos ventrículos, fazendo com que a pressão dos ventrículos aumente e proporcionando a abertura das válvulas aórtica e pulmonar.

O ciclo cardíaco é composto por dois eventos distintos: a diástole e a sístole. A diástole é quando ocorre o relaxamento do músculo, fazendo com que as câmaras cardíacas se encham com o volume de sangue. A sístole é a expulsão do sangue das câmaras cardíacas, ocorrendo devido à contração ou atividade do coração. Existem dois tipos de sístole, a atrial e a ventricular. Cada uma é precedida por uma diástole (GUYTON, 1997; AGUIAR, 2006).

2.5 Eletrocardiograma

Eletrocardiograma (ECG) é o registro dos fenômenos elétricos que se originam durante a atividade cardíaca, auxiliar valioso no diagnóstico de grande número de cardiopatias e outras condições patológicas (ex. distúrbios hidroeletrólíticos), ressaltando-se que algumas anomalias cardíacas não alteram o ECG. Esse registro é realizado através de um aparelho denominado eletrocardiógrafo. O eletrocardiógrafo nada mais é do que um galvanômetro (aparelho que mede a diferença de potencial entre dois pontos) que mede pequenas intensidades de corrente que recolhe a partir de dois eletrodos (pequenas placas de metal conectadas a um fio condutor) dispostos em determinados pontos do corpo humano.

O eletrocardiograma normal (Figura 2.6) é composto pela onda P, pelo "complexo QRS" e pela onda T. O complexo QRS é com frequência constituído por três ondas distintas, a onda Q, a onda R e a onda S. A onda P é causada por potenciais elétricos gerados quando os

átrios se despolarizam antes da contração. O complexo QRS é causado por potenciais gerados quando os ventrículos se despolarizam antes da contração, ou seja, quando a onda de despolarização se difunde pelos ventrículos. Tanto a onda P quanto os componentes do complexo QRS são, portanto, ondas de despolarização. A onda T é causada por potenciais gerados enquanto os ventrículos se recuperam do estado de despolarização. Esse processo no músculo ventricular ocorre 0,25 a 0,35 s após a despolarização, sendo esta onda conhecida como onda de repolarização (GUYTON, 1997).

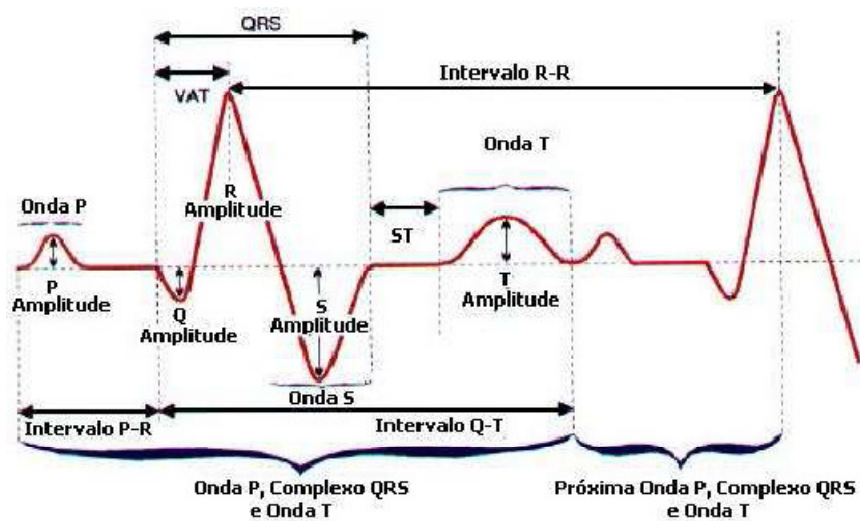


Figura 2.6: principais componentes do ECG (CARLOS et al., 2003).

Segundo a Sociedade Brasileira de Cardiologia (2003), a sensibilidade e a especificidade do ECG são maiores para o diagnóstico das arritmias e distúrbios de condução, do que para as alterações estruturais ou metabólicas, embora exista um grande número de anormalidades fisiopatológicas e estruturais que pode ser reconhecido pelo ECG. Esse exame pode ser útil para conhecer:

- a orientação anatômica do coração;
- o tamanho relativo das diversas câmaras cardíacas;
- uma variedade de alterações do ritmo e condução;
- a extensão, localização e progressão de lesões isquêmicas do miocárdio;
- os efeitos de alterações de concentrações de eletrólitos; e
- a influência de determinados fármacos (p. e. os digitálicos).

A morfologia do ECG depende do gerador de sinal elétrico, do meio condutor e da distribuição e localização dos eletrodos de registro sobre a superfície do corpo (CARLOS et al., 2003).

2.6 Registro Eletrocardiográfico

O papel de registro eletrocardiográfico, ilustrado na Figura 2.7, é quadriculado e dividido em quadrados pequenos de 1mm. Cada grupo de cinco quadradinhos na horizontal e na vertical compreendem um quadrado maior, que é delimitado por uma linha mais grossa. No eixo horizontal, marca-se o tempo. Como o registro é realizado em uma velocidade de 25mm/seg, cada quadradinho equivale a 0,04 segundos. Portanto, cinco quadradinhos (um quadrado maior) equivalem a 0,20s. No eixo vertical, marca-se a voltagem (OLIVEIRA, 2007).

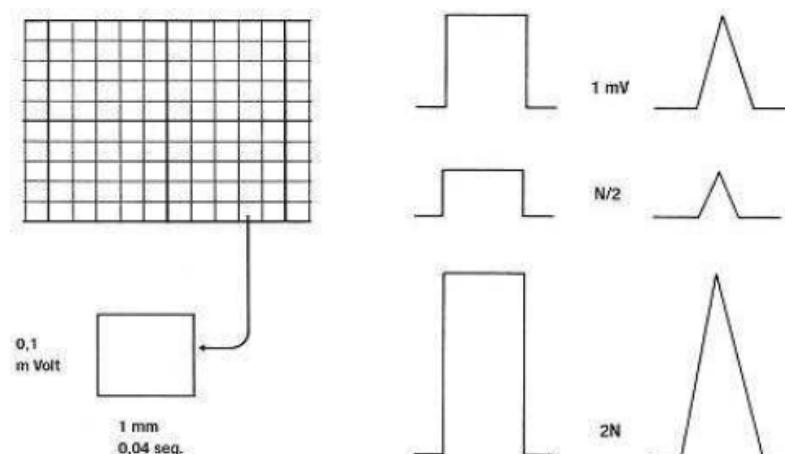


Figura 2.7: papel de registro eletrocardiográfico e tipos de calibração, respectivamente.

Cada quadradinho equivale a 0,1mVolt. Portanto 10 quadradinhos equivalem a 1mVolt. A calibração usual do aparelho deve ser feita para que cada 10mm correspondam a 1mVolt. Mas, por vezes, temos complexos ventriculares de grande amplitude, que não caberiam no papel de registro. Nesta circunstância deveremos fazer o registro N=2, isto é, cada 1mVolt passará a corresponder 5mm. Alguns aparelhos permitem fazer o registro 2N, isto é, fazem o traçado dobrar de tamanho. Cada 1mVolt valerá 20mm. (indicado quando os complexos são muito pequenos). Tal fato pode ser observado na Figura 2.7.

CAPÍTULO 3

JAVA MICRO EDITION

Neste Capítulo são apresentadas as principais características da plataforma Java *Micro Edition* (Java ME), além de sua arquitetura e tecnologias.

3.1 Visão Geral

Segundo a ORACLE (2010), o Java ME é uma coleção de tecnologias e especificações que criam uma plataforma que se ajusta aos requisitos de dispositivos móveis tais como produtos de consumo, dispositivos embarcados e dispositivos móveis avançados.

Java ME é uma plataforma Java direcionada a dispositivos com capacidade de memória, processamento e tela restritos. Esta plataforma é construída com o objetivo de fornecer um ambiente de execução Java capaz de lidar com as características particulares de dispositivos portáteis.

Para utilizar esta plataforma em dispositivos móveis distintos, o Java ME se baseia em três elementos, especificados pela comunidade JCP (*Java Community Process*), que define todos requisitos da plataforma Java, incluindo a especificação de APIs. Os três elementos citados são: configurações, perfis e pacotes opcionais, os quais funcionam sobre uma máquina virtual Java, por sua vez associada a um sistema operacional. Dessa forma, pode-se representar a hierarquia dos elementos nas camadas apresentadas na Figura 3.1 (ZANFOLIM; FERNANDES, 2009).

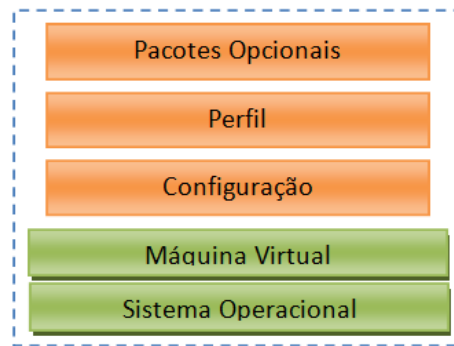


Figura 3.1: hierarquia dos elementos da plataforma Java ME.

3.1.1 Configurações

Para suportar a ampla gama de produtos na plataforma Java ME, a Sun Microsystems (empresa que desenvolveu a plataforma Java) produziu a Configuração. Uma Configuração define uma plataforma Java para uma ampla gama de dispositivos e está intimamente ligada com a JVM (*Java Virtual Machine*). Uma Configuração consiste na especificação de características da linguagem Java e o núcleo de bibliotecas necessárias para o funcionamento da JVM. De acordo com a Sun, “a tecnologia Java ME tem dois centros de projeto – coisas que você segura na mão e coisas que você conecta na parede”. A seguir é mostrado um exemplo das características típicas dos dispositivos dentro das duas configurações atualmente definidos: *Connected Device Configuration* (CDC) e *Connected, Limited Device Configuration* (CLDC).

A *Connected Device Configuration* (CDC) especifica o ambiente Java para dispositivos com maior capacidade de memória e processamento, como TV digital, dispositivos sem fio de alto nível e sistemas automotivos. Em seguida é mostrado um exemplo das características dessa configuração:

- 512 KB (mínimo) de memória para aplicações Java;
- 256 KB (mínimo) para alocação de memória em tempo de execução;
- Conectividade de rede, possivelmente persistente e largura de banda alta.

Já a *Connected, Limited Device Configuration* (CLDC) especifica o ambiente Java para dispositivos com capacidade restrita, como telefones celulares, PDAs e smartphones. Em seguida é mostrado um exemplo das características dessa configuração:

- 128 KB de memória para aplicações Java;
- 32 KB para alocação de memória em tempo de execução.
- interface gráfica restrita;
- baixa potência, tipicamente alimentada por bateria; e
- conectividade de rede, possivelmente wireless, com largura de banda baixa e acesso intermitente.

A CLDC está disponível nas versões 1.0 (JSR-30) e 1.1 (JSR-139) (ORACLE, 2010). Segundo a Sun (2007), as classes da versão 1.0 estão restritas a apenas quatro pacotes:

- *java.io* - tratamento de entrada e saída de dados usando *streams* (abstração);
- *java.lang* - classes básicas da linguagem Java;
- *java.util* - classes de utilidades genéricas (estruturas de dados e manipulação de dados);
- *java.microedition.io* - exclusivamente da plataforma Java ME, incluindo as classes de conexão.

A CLDC 1.1 é uma configuração que engloba os pacotes da versão 1.0 e suporta as seguintes características:

- *Ponto flutuante* - possibilita operações com variáveis do tipo float/double;
- *ClassLoading* - classe abstrata responsável por carregar outras classes;
- *Garbage Collector* - coletor de lixo dos objetos;
- *Finalize()* - com esse método é possível liberar recursos e executar outras operações de limpeza antes que um objeto seja recuperado por coleta de lixo (*garbage collector*);
- *ThreadGroups* - possibilita que os processos sejam executados simultaneamente, facilitando a organização de *threads* em grupos. A *multiThreading* suporta múltiplas linhas de execução através das funções *start()*, *interrupt()*, *pause()*, *resume()* e *stop()* para o controle de *threads*.

3.1.2 Perfis

Para lidar com a ampla gama de capacidades dos dispositivos móveis e fornecer uma maior flexibilidade em mudanças de tecnologia, a Sun lançou o conceito de perfil para a plataforma Java ME. Um perfil é a extensão de uma configuração que fornece as bibliotecas para um desenvolvedor escrever aplicações para um determinado tipo de dispositivo. Por exemplo, o *Mobile Information Device Profile* (MIDP) define APIs para componentes de interface do usuário, movimentação de entrada e de eventos, armazenamento persistente, rede e cronômetros, considerando-se as limitações de memória e tela de dispositivos móveis (MUCHOW, 2001). Devido ao grau de importância desse perfil para o estudo do Java ME, o MIDP será estudado em uma seção separada.

O perfil associado a CLDC é o MIDP (*Mobile Information Device Profile*). Os perfis associados a CDC são: FP (*Foundation Profile*), PP (*Personal Profile*), PBP (*Personal Basis Profile*), RMIP (*Remote Method Invocation Profile*) e GP (*Game Profile*) (JOHNSON, 2007).

3.1.3 APIs Opcionais

Já os pacotes opcionais definem funcionalidades adicionais específicas que podem ser incluídos em uma determinada configuração (ou perfil). O conjunto dessas três subdivisões é chamado de pilha (LI; KNUDSEN, 2005). Devido ao seu grau de importância, essas APIs serão discutidas detalhadamente mais adiante neste trabalho.

3.1.4 Máquinas Virtuais Java

O motor por trás de qualquer aplicação Java é a JVM. Uma vez compilado o código fonte Java em um arquivo *class*, e opcionalmente incluído em um arquivo JAR (*Java Archive*), a JVM traduz os arquivos *class* (mais especificamente, o *byte code* no arquivo *class*) em código de máquina para a plataforma de execução da JVM. A JVM é também responsável por provê segurança, alocar e liberar memória e gerenciar threads de execução. Para a CDC, a máquina virtual tem a mesma especificação do Java SE. Já para a CLDC, a Sun desenvolveu o que é referido como uma implementação de referência de uma máquina virtual, conhecida como *Kilo Virtual Machine*, ou simplesmente KVM. Essa máquina virtual foi

projetada para lidar com as considerações especiais de dispositivos com recursos limitados. Na Figura 3.2 são mostradas várias edições Java e suas respectivas máquinas virtuais (MUCHOW, 2001).

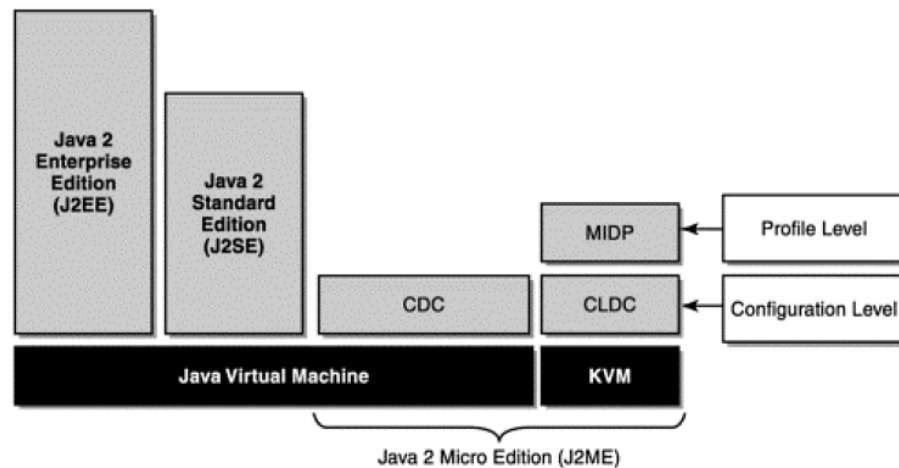


Figura 3.2: edições Java e suas respectivas máquinas virtuais.

3.2 Perfil MIDP

O MIDP - *Mobile Information Device Profile* é um perfil suportado pela configuração CLDC, em que juntos providenciam um ambiente padrão de execução Java para os mais populares dispositivos móveis, como os celulares e PDAs. Este perfil provê um conjunto de bibliotecas e classes que fornecem suporte ao desenvolvimento de aplicações que se referem a diferentes aspectos, como sistema de armazenamento persistente, interface com o usuário, transações seguras, gerência de sons, entre outros. As especificações deste perfil são definidas pela JCP, definindo uma plataforma para desenvolvimento seguro e dinâmico. Atualmente o MIDP possui as versões 1.0 (JSR-37), 2.0 (JSR-118), 2.1 (JSR-118) e 3.0 (JSR-271) (ZANFOLIM; FERNANDES, 2009).

A versão 1.0 do MIDP trabalha integrada com a configuração CLDC 1.0 ou 1.1. Não tem nenhuma API ativa para renderização, não oferece suporte para acesso direto aos pixels de imagens, não tem suporte para *full screen/full canvas* sem uma API proprietária e também não possui suporte direto para áudio. Inclui APIs para o ciclo de vida de aplicações, conectividade de redes HTTP, interface com o usuário e armazenamento persistente. O único

protocolo de rede que o MIDP 1.0 suporta é o HTTP. Os pacotes suportados pela versão 1.0 são:

- *javax.microedition.io* - fornece suporte ao *framework GenericConnection* da configuração CLDC;
- *javax.microedition.lcdui* - API que providencia um conjunto de características para a implementação de interfaces com o usuário;
- *javax.microedition.rms* - providencia um mecanismo de persistência de dados para MIDlets;
- *javax.microedition.midlet* - o pacote MIDlet define aplicações MIDP e interações entre as aplicações e o ambiente no qual a aplicação é executada.

A versão 2.0 é compatível com a versão 1.0 e adiciona novas melhorias como suporte à conexão segura (HTTPS), biblioteca de multimídia, formulário de entrada de dados aprimorada, sensível melhoria na API de suporte a jogos, conceito de aplicações confiáveis (*trusted*) e não confiáveis (*untrusted*). Além de HTTPS, o MIDP 2.0 suporta HTTP, datagramas, *sockets* e SMS (*Short Message Service*). Nesta versão, foi introduzido um conjunto de APIs que é um subconjunto da *Mobile Media API* (MMAPI). Dentre as mudanças nos formulários de entrada, destacam-se a nova aparência do *Form* e a classe *Item*, na qual agora podem ser especificados leiautes horizontais, verticais, novas linhas antes ou depois de itens, entre outros. Novos itens e tipos também foram criados como o item *Spacer*, que serve pra colocar espaçamentos entre os demais itens disponíveis, tipo *pop up* adicionado ao item *choiceGroup*.

No MIDP 2.0 também foram estendidos os comandos de manuseio, introduzida a classe *CustomItem*, que permite ao programador criar seus próprios itens e o suporte a jogos ganhou uma melhoria com o suporte a camadas na tela. Uma camada pode conter um fundo, outra mostrar objetos, uma terceira camada poderia mostrar efeitos especiais ou qualquer outra coisa. Outra mudança é o surgimento da classe *GameCanvas*, uma subclasse de *Canvas* - classe de baixo nível que proporciona o desenvolvimento de jogos (SUN, 2006).

Os pacotes adicionados desta versão são:

- *javax.microedition.lcdui.game* - providencia classes para o desenvolvimento de conteúdo rico para jogos em dispositivos sem fio;

- *javax.microedition.media* - compatível com as especificações da API *Mobile Media*(JSR-135);
- *javax.microedition.media.control* - define o tipo específico, *control*, que pode ser usado como um jogador;
- *javax.microedition.pki* - autentica informações para conexões seguras, através de certificados.

Segundo LI; KNUDSEN (2005), os recursos mínimos de *hardware* do perfil MIDP 2.0 são:

- 130KB de memória não volátil para persistência de dados e bibliotecas da CLDC;
- 32KB de memória volátil para a execução do Java;
- conectividade com algum tipo de rede sem fio;
- interface gráfica;
- uma tela de pelo menos 96 pixels de largura por 54 de altura.

A versão 2.1 do MIDP reforça a especificação MIDP 2.0, tornando a diretiva *leiaute LCDUI* obrigatória, os pacotes *javax.microedition.io.SocketConnection* e *javax.microedition.io.HTTPConnection* não são mais opcionais, entre outros requisitos para aprimorar a versão 2.0.

A versão 3.0 traz como principais características o melhor suporte para dispositivos com telas maiores, suporte a MIDlets para desenhar em telas secundárias, armazenamento RMS seguro e acesso remoto a bancos RMS (JCP, 2009). Esta versão é compatível com o CLDC 1.0, mas o recomendado é o 1.1 e possui compatibilidade com as outras versões do MIDP. O MIDP 3.0 apresenta suporte a IP versão 6, possibilitando fazer parte de um endereço IPv6 (LUZ, 2009).

3.3 MIDlets

Uma MIDlet é uma aplicação Java projetada para ser executada em um dispositivo móvel. Mais especificamente, uma MIDlet tem como seu núcleo as classes da CLDC e MIDP (MUCHOW, 2001).

Todo dispositivo móvel tem um gerenciador de aplicativos (AM - *Application Manager*) que controla os aplicativos a serem instalados, onde e como devem ser armazenados e executados. A comunicação do gerenciador com a MIDlet acontece pela classe MIDlet do pacote *javax.microedition.midlet.MIDlet*. As MIDlets devem herdar esta classe MIDlet que contém métodos que inicializam, resumem, interrompem a execução e destroem MIDlets.

Uma aplicação é iniciada quando o AM executa o método *startApp()*, colocando a aplicação no modo ativo. Enquanto estiver executando, a aplicação pode ser pausada pela AM através do método *pauseApp()*. Isto pode ocorrer quando uma chamada for recebida por exemplo ou o próprio usuário pode pausar a aplicação. E quando a aplicação é encerrada passa para o estado destruído, através do método *destroyApp()*, que limpa todos os recursos para fechar a aplicação (JOHNSON, 2007).

Na figura 3.3 é mostrado o ciclo de vida do MIDlet e as transições entre os estados (LI; KNUDSEN, 2005).

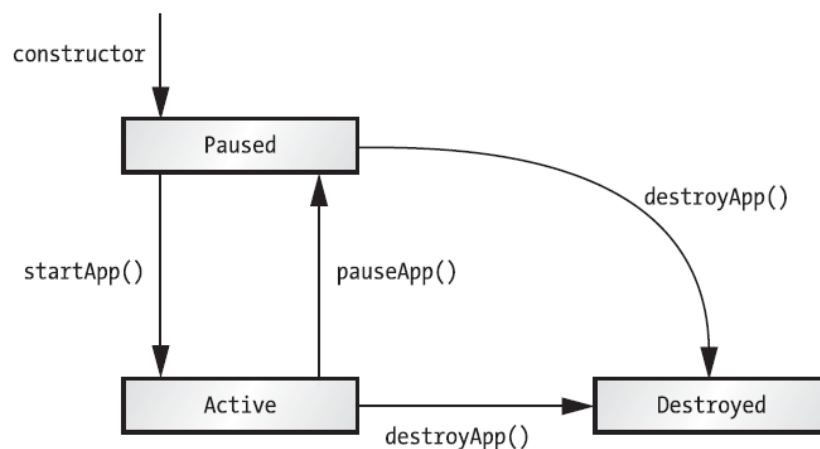


Figura 3.3: ciclo de vida do MIDlet e transições entre os estados.

Segundo MUCHOW (2001), esses três métodos configuram a comunicação que parte do gerenciador de aplicativos para a MIDlet. Além desses métodos, existem também outros três que configuram a comunicação que parte do MIDlet para o gerenciador de aplicativos:

- *destroyApp()* - limpa todos os recursos para fechar a aplicação;
- *notifyDestroy()* - avisa ao gerenciador que pode desligar a MIDlet;
- *notifyPaused()* - envia o pedido de pausa para o gerenciador caso a MIDlet queira pausar;

- *resumeRequest()* - avisa ao gerenciador que a MIDlet pode tornar-se ativa novamente.

3.4 MIDlets Suite

MIDlet Suite é um pacote com um ou mais MIDlets juntas usando um *Java Archive* (JAR) (MUCHOW, 2001). É composto por classes Java, recursos e um arquivo de manifesto, que estão situados dentro do arquivo JAR, além de um arquivo de descrição, chamado *Java Application Descriptor* (JAD). Dentre os recursos estão imagens, dados da aplicação, entre outros. No arquivo de manifesto, cuja extensão é *.mf*, está a descrição do JAR. Já o arquivo JAD descreve os detalhes da aplicação, repetindo muitos dos dados que estão no arquivo de manifesto, porém este arquivo está fora do JAR e pode ser acessado antes de se instalar o arquivo JAR no aplicativo (JOHNSON, 2007).

3.5 JAR

A aplicação final produzida consiste, geralmente, em muitos arquivos. Além de classes Java, outros arquivos tais como imagens e dados da aplicação, conhecidos como recursos, são bastante comuns. Sendo assim, todas essas informações são empacotadas juntas em uma única entidade conhecida como arquivo cuja extensão é *.jar*. Em adicional às classes e arquivos de recursos, um JAR contém também um arquivo conhecido como manifesto. Este arquivo descreve o conteúdo do JAR (MUCHOW, 2001). O arquivo JAR provê muitos benefícios, como: segurança, através de assinaturas digitais; compressão; empacotamento de extensões, juntando vários tipos de extensões diferentes em um único JAR e a portabilidade (ORACLE, 2010).

3.6 JAD

Além de um arquivo JAR, pode ser encontrado um arquivo JAD disponível como parte da MIDlet para fornecer informações sobre a mesma dentro do JAR. O raciocínio por trás da inclusão de um arquivo JAD é fornecer informação para o gerenciador de aplicativos

sobre o conteúdo de um JAR. Com essas informações, podem ser tomadas decisões, como por exemplo, se uma MIDlet é adequada para executar no dispositivo. Uma outra razão para se ter um arquivo JAD é fornecer um meio de passar parâmetros a uma MIDlet sem ter que fazer modificações no arquivo JAR (MUCHOW, 2001). O arquivo JAD possui a seguinte estrutura:

MIDlet-Name: nome da Suite MIDlet;

MIDlet-Version: versão da MIDlet;

MIDlet-Vendor: desenvolvedor da MIDlet;

MIDlet-Icon: especifica o ícone da tela inicial da aplicação;

MIDlet-Description: descrição da aplicação;

MIDlet-info-URL: endereço para um arquivo de informações (JAR); e

MIDlet-DATA-Size: tamanho dos dados.

3.7 APIs Opcionais

Além das classes disponibilizadas obrigatoriamente nas especificações do perfil MIDP, vários outros pacotes podem estar disponíveis para o programador. São as chamadas APIs opcionais, disponibilizadas através de JSRs (*Java Specification Requests*). Essas APIs estão disponíveis apenas em alguns aparelhos, dependendo exclusivamente das capacidades do mesmo e da disponibilidade do fabricante em fornecê-las. Algumas das APIs opcionais disponíveis para o perfil MIDP são citadas por JOHNSON, (2007), tais como:

- *Java APIs for Bluetooth* (JSR-82) - permite o desenvolvimento de aplicações para acesso à rede Bluetooth;
- *Location API* (JSR-179) - permite o desenvolvimento de aplicativos baseados em localização física (LBS);
- *Mobile Game API* (JSR-178) - voltada para criação de jogos;
- *Mobile Media API* (JSR-135) - permite a reprodução de mídia;
- *Security and Trust Services API for J2ME* (JSR-177) - usada para adicionar segurança às aplicações;

- *Wireless Messaging* (JSR-120 e JSR-205) - permite a troca de mensagens SMS.

Entre outras APIs opcionais, importantes para auxiliar no desenvolvimento de aplicações, tem-se:

- *File Connection API* (JSR-75) - permite acesso e armazenamento em arquivos e acesso a programas nativos do dispositivo, como agenda, calendário, entre outros;
- *Mobile Sensor API* (JSR-256) - permite a recuperação de dados de sensores internos ou externos ao aparelho celular;
- *MECHART* - uma API de terceiro, ainda não catalogada pela JCP, que permite construir gráficos de maneira simples e rápida, sem a necessidade de programar diretamente na classe Canvas.

Até o começo do ano de 2010, existiam 927 JSRs na plataforma Java e, para a plataforma Java ME, 83 JSRs foram especificadas. Dentre estas, as JSR-82 e JSR-75 são estudadas, pois, além de ser duas das mais usadas pelos desenvolvedores e possuírem maior reconhecimento, foram também utilizadas na aplicação proposta neste trabalho.

3.8 API Bluetooth

Bluetooth é uma tecnologia de rede sem fio com o intuito de criação de redes temporárias de no máximo oito participantes (WPANs – *Wireless Personal Area Networks*) capaz de transmitir dados e voz. Seu alcance pode chegar a 100 metros e seus dispositivos não precisam estar em linha de visada (de frente um para o outro), pois utiliza radiofrequência para a comunicação entre os elementos que compõe a rede. A faixa utilizada é a ISM (*Industrial, Scientific, Medical*), que opera a frequência de 2,45 GHz. Com o intuito de prover comunicação rápida e barata entre dispositivos móveis, *Bluetooth* pode ser usado também para conectar PC's (WebMobile Magazine, 2008).

3.8.1 Troca de mensagens entre dispositivos Bluetooth

Para ser possível a troca de mensagens entre dispositivos com tecnologia Bluetooth, é necessária uma sequência pré-definida de operações, apresentadas na Figura 3.4.



Figura 3.4: sequência de operações para comunicação Bluetooth.

- *Inquiry*: processo em que um dispositivo procura por outros dispositivos que estejam próximos a ele com o Bluetooth ativado;
- *Paging*: operação que permite a sincronização de alguns dados necessários para a comunicação;
- Estabelecimento de Link: o link é estabelecido para que o fluxo de dados possa ser efetuado;
- Procura por serviços: essa operação permite verificar quais serviços estão disponíveis em um determinado dispositivo;
- L2CAP: com as informações obtidas pela operação de procura de serviço, essa operação permite a criação de um canal de comunicação entre dois dispositivos (WebMobile Magazine, 2008).

3.8.2 JSR-82

Para padronizar a utilização da tecnologia Bluetooth em dispositivos móveis, foi desenvolvida a JSR-82 (*Java Specification Request*). Esse pacote adicional permite o desenvolvimento de aplicações que se comuniquem através de uma comunicação Bluetooth. Seus alvos primordiais são dispositivos que implementem a configuração CLDC (*Connected Limited Device Configuration*) e o perfil MIDP (*Mobile Information Device Profile*). A JSR-82 é dividida em dois pacotes principais: `javax.bluetooth` e `javax.obex`. O primeiro apresenta classes genéricas e representa o núcleo da API e o segundo oferece transferência de objetos com o protocolo OBEX (Object EXchange), sendo opcional e não implementado na maioria dos dispositivos (WebMobile Magazine, 2008).

3.9 JSR-75 (File Connection API)

Normalmente, quando se cria uma aplicação para Java ME, não se tem acesso ao sistema de arquivos do dispositivo. A *FileConnection* API (JSR-75) foi especificada e desenvolvida como um módulo que, se o celular suportar, permite que as aplicações Java ME trabalhem diretamente com os arquivos e diretórios. O primeiro passo ao se usar a JSR-75 é verificar se é suportada no dispositivo dado (NOKIA, 2004). Para isso, pode-se usar o código da Figura 3.5.

```
String v = System.getProperty("microedition.io.file.FileConnection.version");
if( v != null ){
    System.out.println(v);
} else {
    //FCOP não encontrado
}
```

Figura 3.5: código que retorna a versão da JSR-75 suportada pelo dispositivo.

A JSR-75 especifica dois pacotes obrigatórios para a plataforma Java ME. O primeiro, PIM, é um pacote que permite acessar dados de PIM do aparelho, como agenda e listas de tarefa que existem na maioria dos dispositivos móveis. O segundo, FCOP, é um pacote opcional de conexão de arquivos que permite o acesso de diversas formas de dados, tais como imagens, sons, vídeos, texto, entre outros, sobre o sistema de arquivos do dispositivo móvel, incluindo até dispositivos de armazenamento removível (ORACLE, 2004).

A *FileConnection* estende *StreamConnection* pela adição de métodos de manipulação de arquivos e diretórios. Para a abertura de um arquivo é usada a URL no formato: *file://host/path* (ORACLE, 2004).

3.10 Frameworks

Através dos *frameworks* é possível desenvolver aplicações dispondo de várias funcionalidades extras, além de otimizar o desenvolvimento. Entre os *frameworks* mais utilizados para otimização no desenvolvimento de aplicações para dispositivos móveis estão:

- *Floggy* - framework para persistência de dados;
- *J2MicroDB* - fornece um banco de dados relacional limitado para dispositivos móveis;

- *Perst Lite* - banco de dados orientado a objetos para aplicações embarcadas;
- *JMESQL* - banco de dados relacional escrito em Java;
- *KXML* - realiza o parser de arquivos XML;
- *LWUIT* - possibilita o desenvolvimento de interfaces ricas (RIA) e robustas;
- *Java2ME Polish* - framework para personalizar a UI (User Interface) do MIDlet sem alterar o código fonte;
- *Diamont Powder* - acelera a criação de aplicações para coleta de dados;
- *Fire (Flexible Interface Rendering Engine)* - framework para o desenvolvimento de interfaces mais atraentes que as compreendidas pelo MIDP;
- *Marge* - facilita o desenvolvimento de aplicações em Java que fazem uso de Bluetooth.

3.10.1 LWUIT

O LWUIT (*lightweight User Interface Toolkit*) é um framework para criação de interfaces gráficas ricas, modernas, como animações, controles de interface com o usuário e 3D em dispositivos móveis, que suportam a tecnologia Java ME. Baseada no *Swing* do Java SE (*Java Standard Edition*), o LWUIT diminui bastante a complexidade e dá suporte a construção de interfaces gráficas ricas e atraentes para dispositivos móveis (ORACLE, 2009).

Com o LWUIT não há mais necessidade de se desenhar telas em *canvas* para construir interfaces mais amigáveis. Isso é uma grande evolução visto que a complexidade para tal tarefa é alta. O framework oferece melhorias a componentes gráficos de alto nível já existentes no Java ME, tais como *List*, *Form*, *Alert*, entre outros. Ainda oferece várias outras vantagens como suporte a *touch Screen*, diversas fontes, animações, transições de telas animadas e temas variados, que podem ser incluídos pelos próprios usuários.

Além das características citadas acima, o LWUIT também inclui: a utilização de *layouts*, que organizam a disposição dos componentes na tela, além de oferecer melhor portabilidade; integração 3D (se o dispositivo possuir bibliotecas 3D); caixas de diálogo; utilização de abas (como no Java SE) e internacionalização. O LWUIT executa no perfil MIDP 2.0 e está sob a licença GPL v2.0 com a *Classpath Exception*, significando que pode

ser utilizado em aplicações de códigos não abertos, desde que não haja modificação nas classes do framework (LIMA NETO, 2010). Na Figura 3.6 são mostrados exemplos de temas usados pelo LWUIT.



Figura 3.6: exemplos de temas suportados pelo LWUIT.

CAPÍTULO 4

METODOLOGIA DE IMPLEMENTAÇÃO

Neste Capítulo são apresentadas a metodologia de implementação e a arquitetura de um programa para aquisição de biopotenciais cardíacos para dispositivos móveis que permite o recebimento dos sinais de ECG de um eletrocardiógrafo portátil através do *Bluetooth* do celular ou PDA, além da visualização do exame, armazenamento e posterior envio do mesmo. Além disso, são apresentados, também, hardware e *firmware* utilizados para a validação do referido sistema.

4.1 Comparativo entre as linguagens de desenvolvimento móvel

A consolidação dos dados publicados em um relatório produzido pela INTELIMED pode ser observada na Tabela 4.1. Nessa Tabela, é mostrado um comparativo entre as principais plataformas de dispositivos móveis disponíveis no mercado, quanto ao desenvolvimento de *software* e às linguagens de programação suportadas.

	Symbian	Android	Maemo	palmOS	webOS	Windows	BbOS	iPhoneOS
C/C++	X	X	X	X	-	X	-	Obj-c
Java	X	Custom	X	X	Javascript	X	X	-
.NET	-	-	-	-	-	X	-	-
Basic	X	-	-	-	-	X	-	-
Python	X	-	X	-	-	X	-	-

Tabela 4.1: comparação entre plataformas e linguagens de programação suportadas por dispositivos móveis disponíveis no mercado.

Para o desenvolvimento do sistema móvel proposto neste trabalho, é importante o uso uma linguagem que proporcione a maior independência possível da plataforma, logo, que atinja o maior número de dispositivos. Considerando que os aparelhos produzidos pelas

principais fabricantes de PDAs e *smartphones* executam uma das plataformas pesquisadas no referido relatório e, considerando também, a relação presente na Tabela 4.1, a linguagem Java (Java ME para dispositivos móveis) é a que mais se aproxima dos requisitos de portabilidade. Esta linguagem é suportada por quase todas as plataformas (exceto *iPhone OS*), são oferecidas ferramentas gratuitas para o desenvolvimento de aplicações e ainda possui suporte tanto pela ORACLE quanto pela ampla comunidade de desenvolvedores.

4.2 Visão geral do sistema

É proposto um *software* que pode ser instalado em celulares e PDAs que facilita a análise visual dos biopotenciais cardíacos recebidos via Bluetooth de um eletrocardiógrafo portátil, *front-end* da aplicação (LIMA NETO, 2010). Além disso, é possível também cadastrar pacientes, configurar e controlar remotamente o eletrocardiógrafo e enviar posteriormente os dados do ECG através da interface GPRS, nativa do aparelho, a um centro de análise e diagnóstico. O objetivo do programa é simplificar e agilizar a manipulação desse instrumento, tão importante para diagnóstico de doenças cardiovasculares.

Enfermeiros e agentes de saúde em qualquer localidade ao possuírem eletrocardiógrafos portáteis, *front-end* da aplicação, e o *software* instalado em qualquer celular ou PDA devem ser capazes de disponibilizar facilmente os dados coletados para centros de excelência, onde médicos especialistas podem avaliar e diagnosticar possíveis patologias, à distância.

4.3 Arquitetura do sistema

Para o trabalho proposto foi abordado, em conjunto ao sistema em Java ME, o uso de um protótipo de hardware de um eletrocardiógrafo portátil com uma derivação, além de rotinas de *firmware* responsáveis por controlar o hardware, desenvolvidos por LIMA NETO (2010). Dessa forma, o programa proposto neste trabalho foi desenvolvido com a finalidade de controlar o *hardware* e prover uma interface amigável com o usuário. O modelo da comunicação para este trabalho pode ser observado na Figura 4.1.

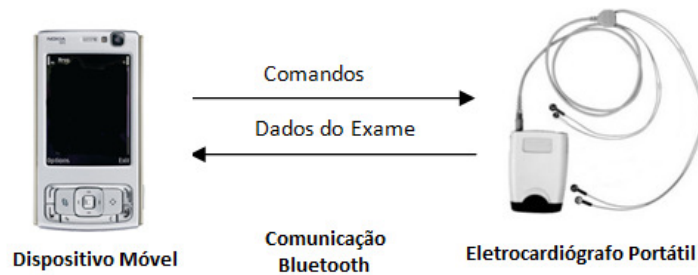


Figura 4.1: comunicação Bluetooth entre o sistema móvel e o eletrocardiógrafo portátil.

4.4 Descrição do hardware do protótipo

Segundo LIMA NETO (2010), a arquitetura de hardware foi planejada e desenvolvida em duas fases, onde, na primeira, foi realizada revisão bibliográfica sobre as funcionalidades de típicos eletrocardiógrafos clínicos, analisadas alternativas de soluções através de simulações dos principais circuitos e foi desenvolvido o primeiro protótipo de hardware contemplando um *front-end* com aquisição, filtragem e amplificação, desenvolvimento das rotinas de firmware para conversão analógico-digital, comunicação serial RS-232 com PC e validação da aquisição dos sinais através do processamento destes, com o algoritmo de segmentação de ECG de MADEIRO et al (2007).

Na segunda fase, foi construído o segundo protótipo de hardware, realizando a correção de erros e implementação de atualizações, com base na análise dos resultados obtidos através do protótipo I, e a adição das funcionalidades de alimentação por bateria, circuito de proteção contra surtos de tensão, detecção de falha/falta de eletrodos e comunicação via Bluetooth. Apresenta-se na Figura 4.2 o diagrama de blocos das funcionalidades implementadas nos protótipos desenvolvidos na fase I e II.

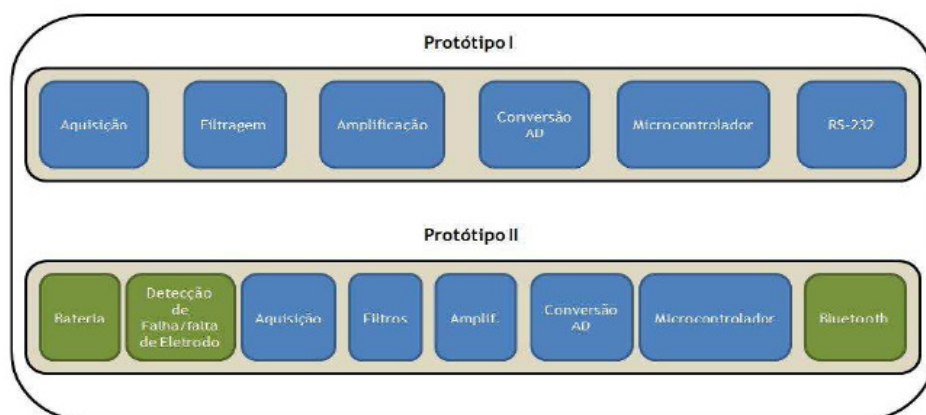


Figura 4.2: diagrama de blocos dos protótipos desenvolvidos por LIMA NETO (2010).

4.5 Descrição do firmware do protótipo

Segundo LIMA NETO (2010), os principais objetivos do firmware desenvolvido são: gerenciar todos os periféricos de *hardware* do protótipo; configurar a taxa de amostragem do exame; digitalizar o sinal e gerenciar a comunicação com o dispositivo móvel para transmissão do exame. Este firmware é totalmente desenvolvido na linguagem C e o fluxograma básico pode ser observado na Figura 4.3.

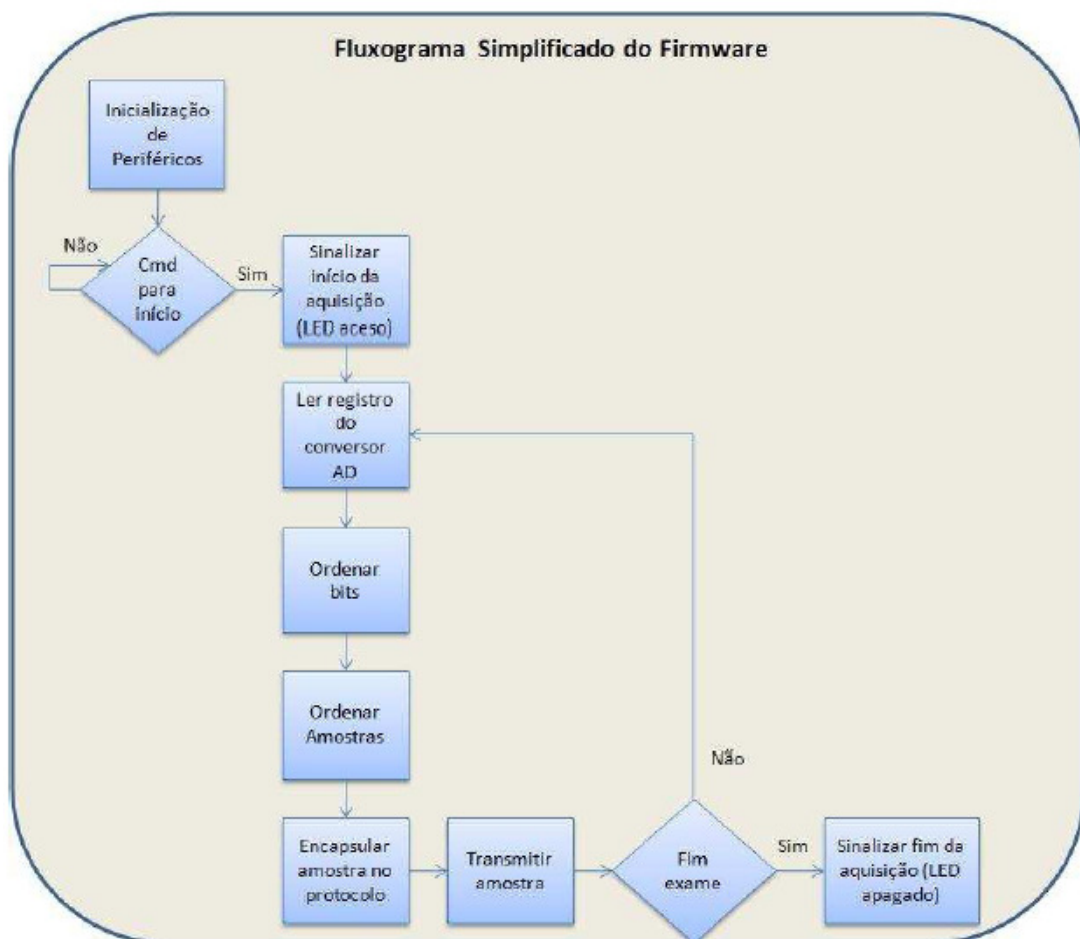


Figura 4.3: fluxograma básico do firmware (LIMA NETO, 2010).

O firmware suporta cinco comandos: **configurar taxa de amostragem** que define a taxa de amostragem utilizada no processo de aquisição do ECG (100Hz, 250Hz, 360Hz ou 1kHz); **testar eletrodos** que verifica se os eletrodos estão posicionados corretamente no paciente, medindo a impedância da interface eletrodo-tecido, enviando uma resposta com o estado da conexão entre os eletrodos e o paciente; **verificar taxa de amostragem** que verifica qual taxa de amostragem está configurada no módulo de ECG; **iniciar exame** que gera

sinalização através de um *led*, iniciando o processo de aquisição do sinal, enviando as amostras coletadas ao celular, através da interface Bluetooth; e **parar exame** que encerra o processo de aquisição do exame.

Os códigos dos comandos e retornos de sucesso e falha são apresentados na Tabela 4.2.

Comandos do <i>Firmware</i>			
Comando	Código (Char)	Retorno Sucesso (Char)	Retorno Falha (Char)
Configurar taxa de amostragem 100Hz 250Hz 360Hz 1kHz	A B C D	O	N
Iniciar Exame	I	## - Início Dados ** - Fim	N
Parar Exame	P	O	N
Testar Eletrodos	T	O	N
Verificar taxa de amostragem	V	A - 100Hz B - 250Hz C - 360Hz D - 1kHz	N

Tabela 4.2: tabela de comandos do firmware (LIMA NETO, 2010).

Os comandos suportados pelo *firmware*, tais como “A”, “B”, “C”, “I” e etc, são implementados no *software* proposto neste trabalho e enviados via *Bluetooth* do celular para o protótipo de hardware dependendo da ação do usuário sobre o sistema.

4.6 Descrição do *software*

Para facilitar o desenvolvimento é utilizado o ambiente de desenvolvimento *NetBeans IDE* (ORACLE, 2009). Trata-se de um ambiente de desenvolvimento integrado livre e de código aberto, que dispõe de todas as ferramentas necessárias para criar aplicações *desktop*, empresarial, *web* e aplicações móveis com a plataforma Java, assim como C/C++, PHP, *JavaScript*, *Groovy* e *Ruby* (NETBEANS, 2010).

Segundo SOMMERVILLE (2007), *software* não é apenas o programa de computador, mas também todos os dados de documentação e configuração associados, necessários para

que o programa opere corretamente. Um sistema de *software* consiste, geralmente, no conjunto de programas separados; arquivos de configuração necessários para o bom funcionamento do sistema; documentação do sistema e do usuário que descrevem a estrutura do sistema e explicam como usar corretamente o sistema, respectivamente; além de *websites*, opcionais, por meio dos quais é possível a obtenção de informações atualizadas sobre o produto.

Os *softwares* que podem ser vendidos para clientes são chamados de produtos de *software*. Existem dois tipos: produtos genéricos e produtos sob encomenda (SOMMERVILLE, 2007). A diferença básica entre estes é que os produtos genéricos são aqueles cuja especificação do *software* é controlada pelo seu desenvolvedor, já os produtos encomendados são aqueles, cuja especificação é normalmente desenvolvida e controlada pelo cliente. Dessa forma, o *software* proposto neste trabalho é classificado como produto genérico.

4.7 Processo de *software*

Segundo SOMMERVILLE (2007), um processo de *software* é um conjunto de atividades, tais como desenvolvimento de *software* propriamente dito, que leva à produção de um produto de *software*.

Já o modelo de processo de *software* é uma representação abstrata de um processo de *software*. O sistema proposto neste trabalho foi desenvolvido segundo um modelo em cascata simplificado, ou seja, nem todas as atividades de alguns estágios foram implementadas no decorrer do projeto. O modelo em cascata foi escolhido por ser usado principalmente quando os requisitos são bem compreendidos e quando há pouca probabilidade de mudanças radicais durante o desenvolvimento do sistema, além de produzir documentação em cada fase e por ser de fácil aderência a outros modelos de processo de *software*. Esse tipo de modelo é dividido em cinco estágios principais e sucessivos.

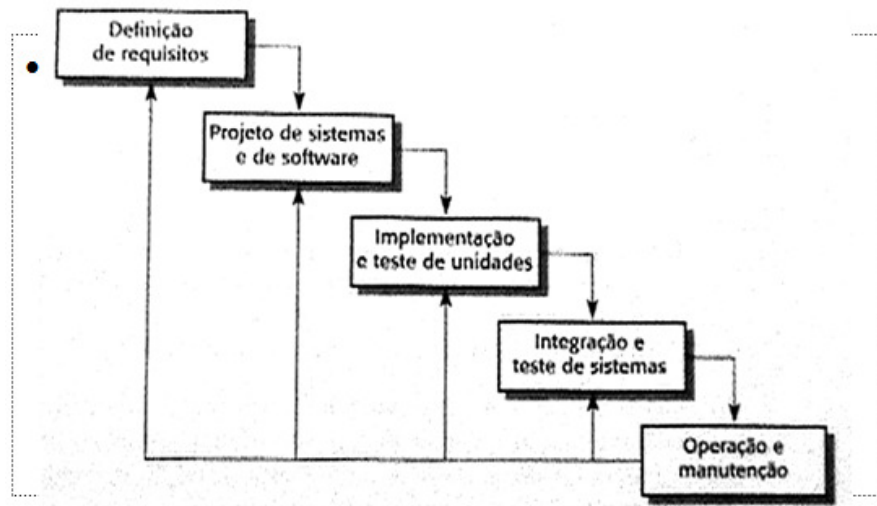


Figura 4.4: descrição geral do modelo de processo: modelo em cascata.

4.7.1 Definição de requisitos

Esse é um estágio particularmente crítico na engenharia de *software*, pois os erros nessa fase conduzem inevitavelmente a problemas posteriores no projeto e na implementação do sistema. No desenvolvimento do sistema proposto, a definição de requisitos foi dividida em quatro fases menores: estudo de viabilidade, elicitação e análise de requisitos, especificação de requisitos e validação de requisitos.

Na fase de estudo de viabilidade foi feita uma avaliação para verificar se o trabalho proposto poderia ser implementado por meio das tecnologias atuais de *software* e hardware. O resultado serviu para a tomada de decisão a cerca da continuação do projeto.

Na fase de elicitação e análise de requisitos foram identificados, dentre as funcionalidades do *software* para *desktop* de um eletrocardiógrafo clínico, quais os principais recursos necessários à aplicação para dispositivos móveis proposta. Neste período foi, também, realizada uma entrevista com o Dr. Roberto Lima Farias, mestre em Cardiologia pela Universidade Federal de São Paulo – Escola paulista de Medicina.

A especificação de requisitos foi a fase durante a qual as informações coletadas na entrevista e na observação de sistemas existentes foram traduzidas em requisitos funcionais, requisitos não funcionais e casos de uso.

Por fim, na fase de validação de requisitos foram verificados se os requisitos estavam de acordo com a realidade. Os erros descobertos foram corrigidos.

Segundo SOMMERVILLE (2007), os requisitos de um sistema de software são frequentemente classificados como funcionais ou não funcionais. Alguns exemplos de requisitos funcionais são: funções que o sistema deve fornecer; como o sistema deve reagir a entradas diferentes; ou o que o sistema não deve fazer. Já como exemplos de requisitos não funcionais têm-se: restrições sobre os serviços ou funções oferecidas pelo sistema, tais como restrições de tempo, padrões, entre outros. Os principais requisitos funcionais e não funcionais do software proposto se encontram na Tabela 4.3.

Requisitos Funcionais	Requisitos Não Funcionais
O sistema deve possibilitar o cadastramento dos pacientes que realizam o exame	O sistema deve ser fácil de ser usado por médicos, enfermeiros e agentes de saúde
O sistema deve controlar um eletrocardiógrafo portátil via <i>Bluetooth</i>	O sistema deve ser desenvolvido em uma linguagem que proporcione a maior independência possível da plataforma
O sistema deve fornecer telas atrativas e intuitivas	O esforço para aprender a utilizar o sistema deve ser mínimo
O sistema deve receber dados dos exames realizados via <i>Bluetooth</i> e plotar o gráfico em tempo real	O esforço necessário para modificar o sistema deve ser mínimo
O usuário deve ser capaz de fazer uma busca pelo nome do paciente para que seja possível a visualização do exame armazenado no dispositivo móvel	O sistema deve ser utilizado para ajudar no diagnóstico precoce de doenças cardiovasculares
O sistema deve ser capaz de armazenar os exames localmente em arquivos de texto	O sistema deve ser de baixo custo
O sistema deve ser capaz de enviar os arquivos armazenados localmente através da interface GPRS ou 3G	O sistema não deve ter letras muito pequenas

Tabela 4.3: exemplos de Requisitos funcionais e não funcionais do sistema proposto.

O diagrama de caso de uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário, tendo o objetivo de auxiliar na comunicação entre os analistas e o cliente. Já o caso de uso é uma unidade significativa de um trabalho. Por exemplo a “autenticação” para um sistema. A Figura 4.5 mostra o diagrama de caso de uso do *software* proposto.

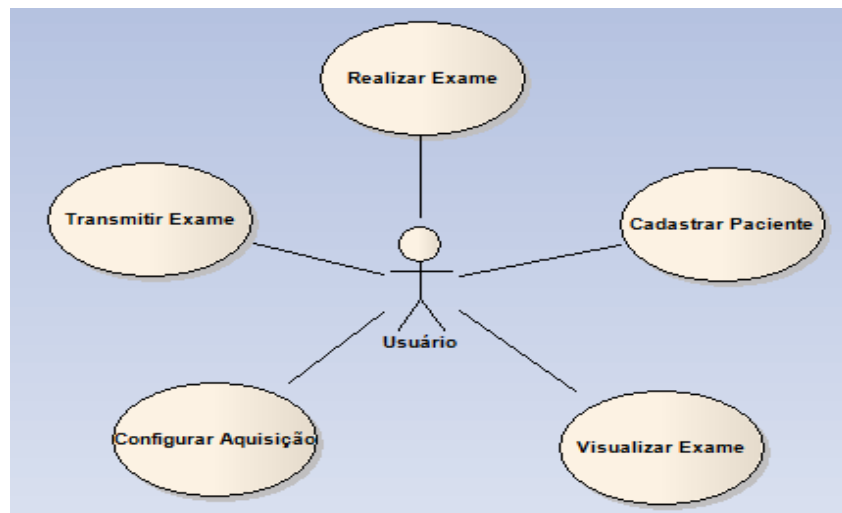


Figura 4.5: diagrama de Caso de Uso do sistema.

4.7.2 Projeto de sistema e de software

Nesta fase, a estrutura do *software* a ser implementado foi descrita. O projeto não foi concluído imediatamente, mas foi desenvolvido iterativamente por meio de diferentes versões. Desse modo, os requisitos foram divididos em subsistemas de *software*, o qual foi estabelecida uma arquitetura geral, como mostra a Figura 4.6.

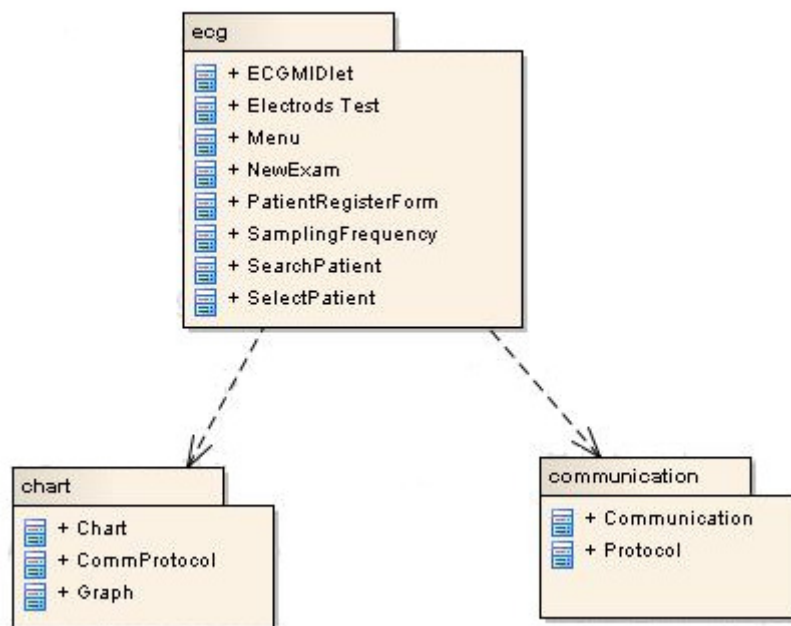


Figura 4.6: sistema dividido em subsistemas.

Os protótipos das principais interfaces do sistema foram feitos para nortear a implementação do *software*, como mostrado na Figura 4.7. Na Figura 4.8 é mostrado um exemplo de um protótipo e sua respectiva implementação usando o framework LWUIT.

Algumas modificações foram feitas para adaptar os protótipos aos componentes gráficos disponibilizados pelo framework escolhido.

Figura 4.7: protótipo de três telas feitos à mão livre: menu principal, cadastro de pacientes e exame realizado com sucesso, respectivamente.



Figura 4.8: comparação entre protótipo e implementação da tela de cadastro de paciente.

Para dar uma visão geral do relacionamento entre essas telas, foi desenvolvido um organograma como mostrado na Figura 4.9.

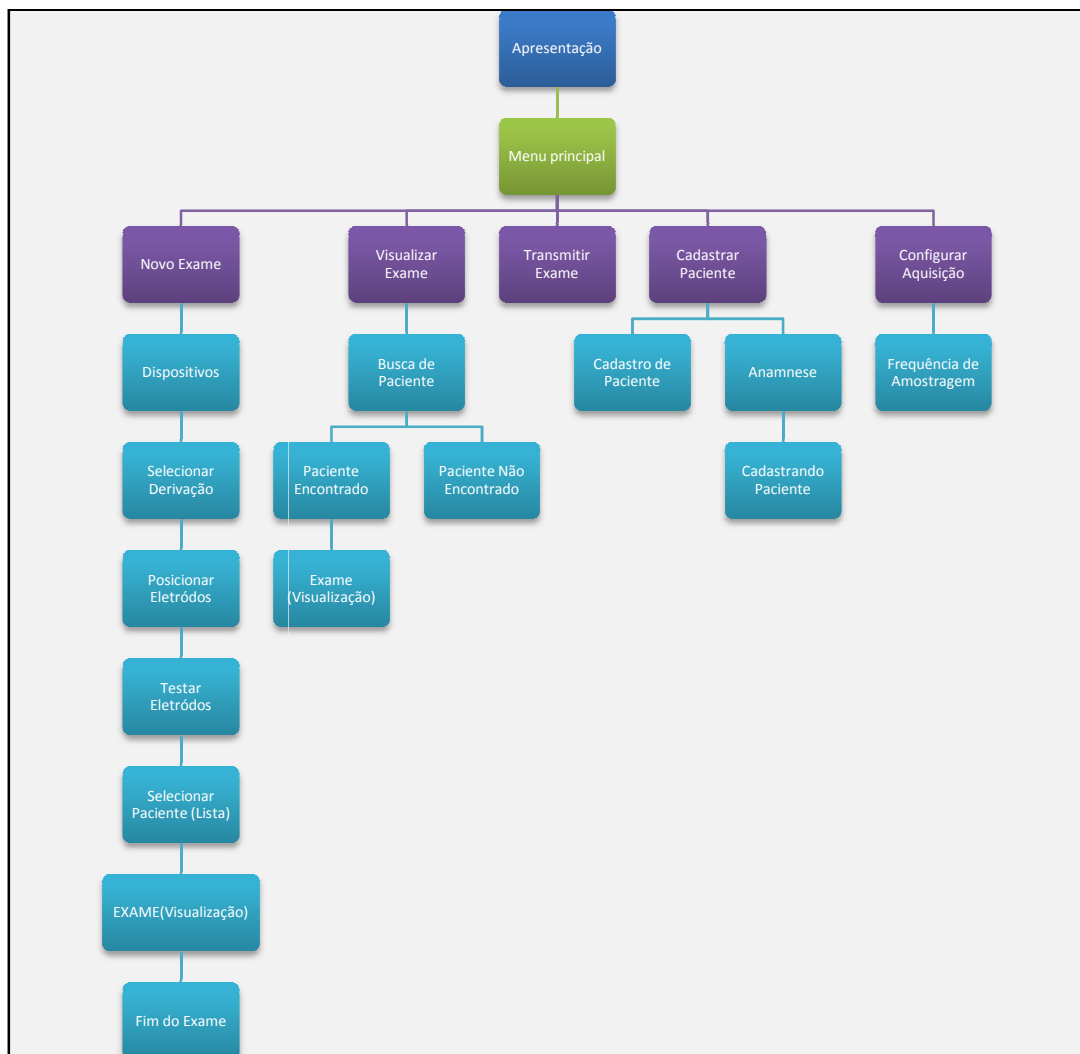


Figura 4.9: organograma simplificado das telas da aplicação.

4.7.3 Implementação e testes de unidade

Segundo SOMMERVILLE (2007), a implementação é o estágio do desenvolvimento no qual as especificações do sistema são convertidas em um sistema executável, envolvendo processos de programação de *software*. Durante este estágio, desenvolveu-se um diagrama de classes afim de se obter uma visão geral da implementação do *software*, conforme mostrado na Figura 4.9.

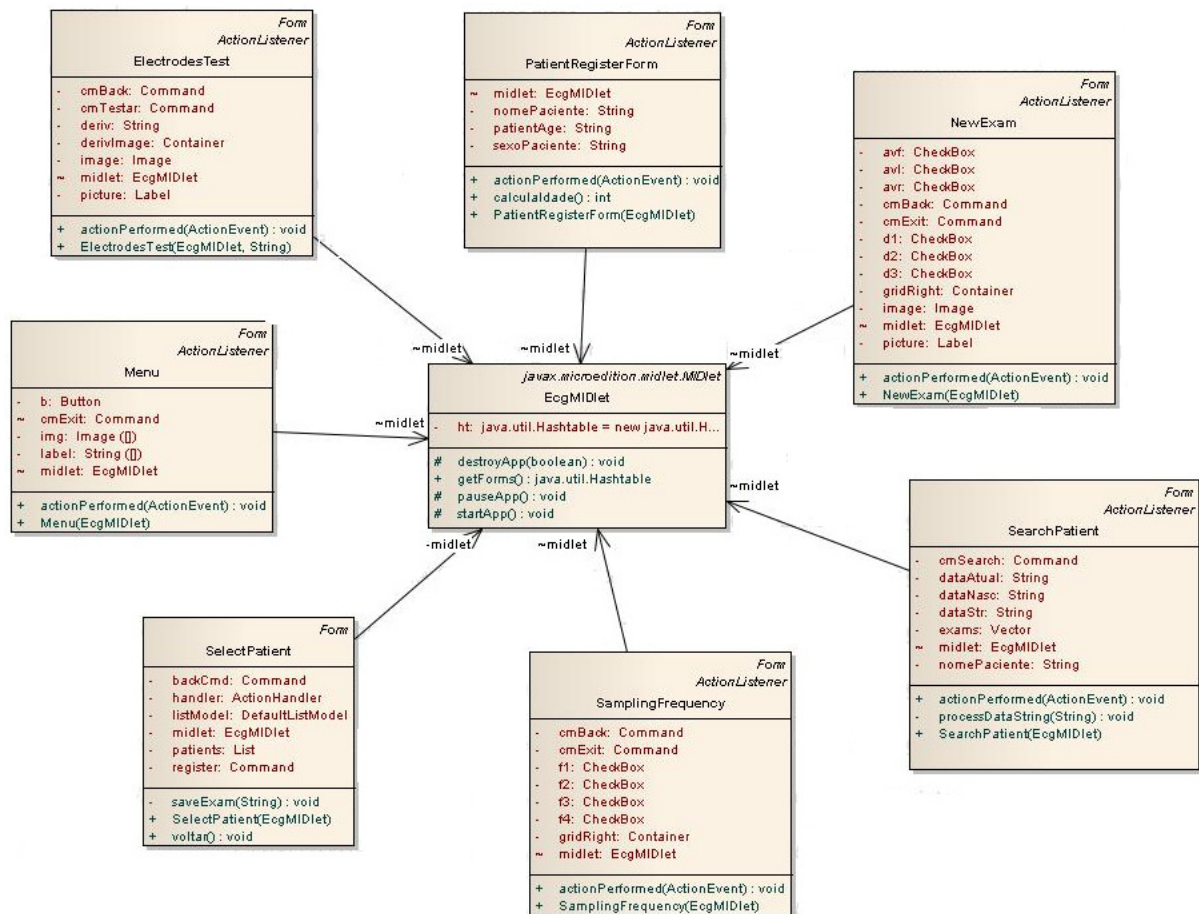


Figura 4.10: visão geral de implementação.

A programação é uma atividade pessoal e não existe um processo geral que seja normalmente seguido (SOMMERVILLE, 2007). No caso do sistema proposto, foi desenvolvida, primeiramente, uma versão do *software* contemplando apenas a implementação das interfaces gráficas e o relacionamento entre as mesmas. Essa versão serviu como base para as atualizações futuras, em relação à usabilidade do sistema. Após testes com diversos

usuários, várias modificações foram feitas sobre esta primeira implementação para que o *software* se tornasse fácil de ser operado por qualquer usuário.

Em seguida, desenvolveu-se uma segunda versão do *software* contemplando, desta vez, a integração da interface gráfica com comunicação via *Bluetooth* e o acesso à árvore de arquivos (acesso a diversas formas de dados sobre o sistema de arquivo do dispositivo móvel).

Além disso, em cada unidade funcional do *software*, foram realizados testes de caixa-preta, que envolveram a verificação da saída dos dados usando entradas de vários tipos, a fim de confirmar se estavam atendendo a sua especificação. Em caso de identificação de erros, a depuração destes erros seguiu as etapas mostradas na Figura 4.10.



Figura 4.11: processo de depuração de erros (SOMMERVILLE, 2007).

4.7.4 Integração e teste de sistema

Ao término das implementações, as unidades individuais dos subprogramas foram integrados e testados como um programa completo. Somente após esse passo foi possível a instalação do *software* em um celular *Nokia N95*, integração com *hardware* e *firmware* citados anteriormente e testes sistêmicos em laboratório, através da aquisição dos biopotenciais cardíacos de dois voluntários, cujas identidades são preservadas.

Segundo LIMA NETO (2010), a coerência e a qualidade dos exames produzidos pelo protótipo de hardware foram analisadas, comparando-se os resultados de desempenho de algoritmos já consolidados de detecção de QRS, propostos por Madeiro et al (2008) e Madeiro et al (2006). Como referência utilizou-se a base de dados QT do MIT-BIH (KÖHLER; HENNING; ORGLMEISTER, 2002), largamente utilizada para fins de validação.

Já a coerência e a qualidade dos exames produzidos pelo *software* foram avaliadas, a partir de vários testes comparativos entre os valores do exame obtido na saída do protótipo de hardware e os valores do exame exibido no dispositivo móvel e salvo em arquivos de texto.

Além disso, foram realizados também vários testes de *stress* no sistema para determinar a sua robustez, consistindo de execuções em tempos superiores a 30 minutos.

4.7.5 Operação e manutenção

Segundo SOMMERVILLE (2007), esta etapa é a mais longa do ciclo de vida de um *software*. A manutenção envolve a correção de erros não detectados nos estágios anteriores do ciclo de vida, no aprimoramento da implementação das unidades de sistema e na amplificação dos serviços de sistema à medida que novos requisitos são identificados. No trabalho proposto, esta etapa começou durante os testes sistêmicos em laboratório com a melhoria de algumas partes do *software* e adição de outras necessárias. É importante ressaltar que essas mudanças (manutenção de *software*) implicaram, algumas vezes, em repetição de estágios anteriores do processo. A operação do sistema pode ser entendida melhor através da observação de seus casos de uso.

4.8 Análise dos casos de uso

Ao ser iniciado, o programa mostra uma tela de apresentação e ao ser escolhida a opção “Iniciar”, o usuário é conduzido ao menu principal da aplicação onde são encontradas as principais funções, mostradas no diagrama de caso de uso da Figura 4.5.



Figura 4.12: tela inicial do sistema e menu principal, respectivamente.

4.8.1 Realizar Exame

O primeiro item do menu é responsável por dar início à aquisição dos biopotenciais cardíacos. Primeiramente, ao ser selecionada essa opção, o sistema aciona o *Bluetooth* do celular ou PDA. Em seguida, o usuário deve escolher a derivação a ser utilizada no exame. Na Figura 4.9 podem ser observadas as telas de seleção de derivação e posicionamento dos eletrodos.

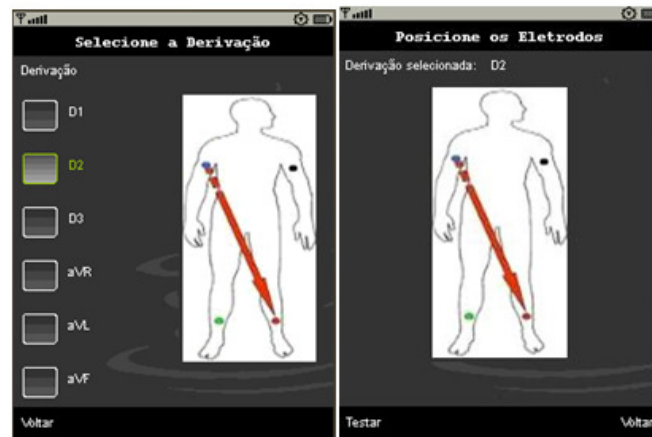


Figura 4.13: telas de seleção de derivação e posicionamento dos eletrodos.

Na tela que orienta o correto posicionamento dos eletrodos, aparece o comando **Testar** que, ao ser acionado, envia um caractere “T”, via *Bluetooth* para o protótipo de hardware através do protocolo RFCOMM. Este protocolo emula um cabo serial e provê uma simples implementação de rede sem fio para comunicação RS-232. O *hardware* testa os eletrodos, medindo a impedância da interface eletrodo-tecido e retorna ao celular, via *Bluetooth*, o *status* do posicionamento. Na Figura 4.14 são mostradas as telas de alerta de acordo com a resposta recebida pelo sistema (sucesso e falha no posicionamento, respectivamente).



Figura 4.14: telas de sucesso e falha no teste dos eletrodos, respectivamente.

Na tela de confirmação de sucesso no teste dos eletrodos, o sistema fornece a opção de iniciar o exame. Ao ser escolhida essa opção, o sistema envia ao protótipo de *hardware* o caractere “I” e espera receber “##” para começar a coleta dos dados do exame e a respectiva escrita desses dados em arquivos de texto. Tanto durante quanto após a realização do exame, é possível visualizar o ECG no celular. Um exemplo de visualização de um exame adquirido pelo protótipo é mostrado na Figura 4.11.



Figura 4.15: visualização do exame no celular.

4.8.2 *Cadastrar Paciente*

O item **Cadastrar Paciente** permite que o usuário crie documentos contendo o nome do paciente, data de nascimento, sexo, idade e data de realização do exame. Além disso, ao clicar no botão **Anamnese**, o profissional da saúde pode guardar as informações obtidas na entrevista que fez com seu paciente. Ao término do cadastro do paciente, os dados são salvos em arquivos de texto. Na Figura 4.16 são observadas as telas de cadastro de paciente, anamnese e sucesso no armazenamento dos dados, respectivamente.



Figura 4.16: telas de cadastro de pacientes, anamnese e exame.

4.8.3 Configurar aquisição

Através do menu **Configurar Aquisição** é possível definir qual das diversas taxas de amostragem utilizadas na realização do exame deverá ser usada. A plataforma de *hardware*, para este tipo de taxa, suporta quatro alternativas: 100Hz, 250Hz, 360Hz e 1kHz.

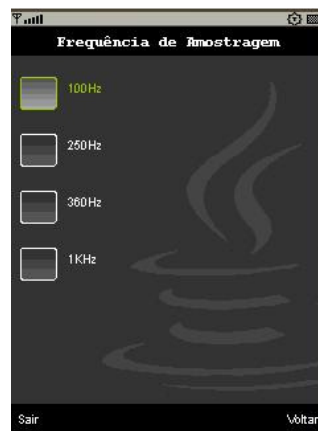


Figura 4.17: tela de escolha de frequência de amostragem.

4.8.4 Visualizar exame

O sistema permite que o usuário visualize os exames realizados e armazenados no próprio dispositivo móvel. Para isso, basta que seja feita uma busca pelo exame do paciente informando nome, data de nascimento e idade do paciente, além da data de realização do exame, como é mostrada na Figura 4.18. Se existir algum registro do paciente procurado, o sistema mostrará uma lista contendo todos os exames cadastrados com aqueles dados

fornecidos na busca. Por outro lado, se não houver registro de paciente com aqueles dados fornecidos na busca, o sistema mostra um alerta de falha na busca.



Figura 4.18: telas de busca de paciente, paciente encontrado e paciente não encontrado, respectivamente.

4.8.5 Transmitir Exame

Nesse caso de uso, o sistema emite um alerta que notifica o usuário sobre o local onde os exames são salvos no sistema de arquivos do dispositivo usado e orienta que deve ser usado o cliente de *e-mail* pessoal do usuário para envio dos exames armazenados.

CAPÍTULO 5

RESULTADOS, CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS

Neste Capítulo são apresentados os resultados obtidos durante o período de desenvolvimento deste trabalho, além das conclusões e trabalhos futuros que podem ser alcançados a partir deste.

5.1 Resultados

Durante o desenvolvimento deste trabalho, foi produzido um sistema móvel de fácil integração com o eletrocardiógrafo portátil de LIMA NETO (2010). Esse programa pode ser instalado em qualquer dispositivo móvel que tenha suporte a plataforma Java e a API Bluetooth. O sistema possui uma interface homem-computador simples, de fácil operação, tornando-se viável o seu uso por parte de médicos, enfermeiros ou agentes de saúde. Observa-se na Figura 5.1 o *software* instalado em um celular *NOKIA 5610 Express Music*.



Figura 5.1: instalador do sistema proposto.

O sistema desenvolvido controla o eletrocardiógrafo através de comandos enviados e recebidos via *Bluetooth*, seguindo o padrão da Tabela 4.2. As telas de teste de eletrodos e configuração da frequência de amostragem podem ser vistas na Figura 4.13 e Figura 4.16, respectivamente. Outros resultados obtidos foram a implementação da função de cadastro de pacientes e armazenamento dos cadastros e exames em arquivos de texto salvos localmente no dispositivo, como apresentado na Figura 4.12.

Segundo LIMA NETO (2010) o sinal na saída do eletrocardiógrafo é obtido conforme mostrado na Figura 5.2. Em seu trabalho, durante os testes, o referido autor teve o auxílio de dois voluntários, na função de pacientes, nos quais o equipamento foi conectado a cada paciente, através de três eletrodos, permitindo a formação de uma das derivações: D_1 , D_2 ou D_3 . Uma vez digitalizados, estes exames foram enviados ao celular. A Figura 5.2 mostra o sinal no osciloscópio obtido na saída do eletrocardiógrafo.



Figura 5.2: biopotencial digitalizado e processado pelo *front-end* da aplicação.

No celular o sinal é exibido, de acordo com a Figura 5.3, a qual mostra o exame no emulador do WTK 2.5.2 (Wireless Toolkit for CLDC) e no celular NOKIA 5610 *Express Music*.



Figura 5.3: mesmo exame sendo visualizado no emulador e no celular, respectivamente.

Ao ser enviado ao PC, o exame pode ser observado fazendo uso dos algoritmos propostos por Madeiro (2007), como é mostrado na Figura 5.4.

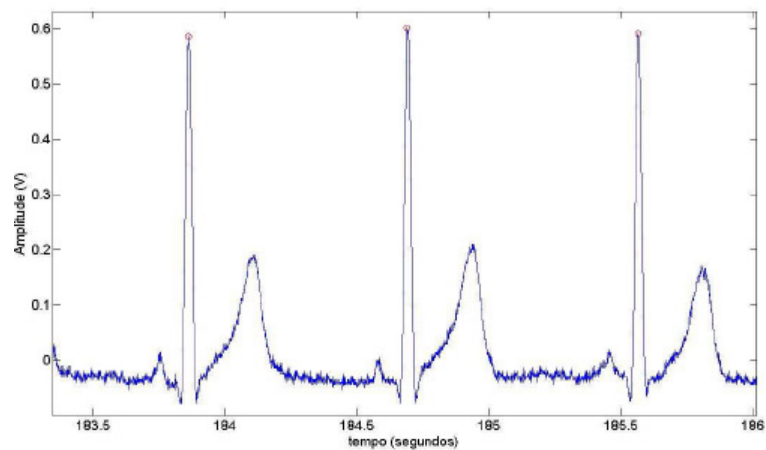


Figura 5.4: sinal visualizado no PC.

A partir da comparação entre os valores dos dados plotados na tela do dispositivo móvel, armazenados em arquivos de texto e a base de dados gerada pelo protótipo de *hardware* de LIMA NETO (2010) que é enviada via *Bluetooth* é possível afirmar que o

sistema apresenta valores coerentes com taxa de erros na transmissão praticamente nula. Esses resultados só foram possíveis devido os serviços de controle de erros oferecidos pela camada *Logical Link Control Adaptation Protocol (L2CAP)* que faz parte da pilha de protocolos do *Bluetooth*. A detecção de erros na comunicação usa o método *Cyclic Redundancy Check (CRC)* que gera um valor expresso em poucos *bits* em função do pacote de dados. O valor gerado é anexado à informação a transmitir e verificada após a recepção. A correção é feita por meio do método *Automatic Repeat Request (ARQ)*. Esse método retransmite o pacote até que um *ACK (acknowledgement)* seja transmitido pelo destino (BLUETOOTH SIG, 2007).

5.2 Conclusões

O foco deste trabalho foi o desenvolvimento de um programa para dispositivos móveis capaz de se comunicar via *Bluetooth* e controlar um eletrocardiógrafo portátil. Essa abordagem pode permitir uma maior mobilidade na obtenção de exames em comunidades carentes de profissionais especializados. Por ser de baixo custo e fácil manipulação este sistema pode ser usado principalmente por profissionais do Sistema Único de Saúde (SUS). Porém, o desenvolvimento desse tipo de *software* não é uma tarefa fácil. Por lidar com vidas humanas, o desenvolvedor precisa garantir a precisão da informação transmitida ao médico para que seja assegurada a segurança do paciente.

Foi mostrado que é possível exibir os sinais recebidos com poucos recursos computacionais, utilizando recursos gráficos do framework LWUIT. Além disso, a usabilidade do sistema ficou bem evidenciada com o uso desse framework que foi empregado principalmente para o desenvolvimento de uma interface gráfica mais moderna, atrativa, robusta e fácil de operar, com componentes comuns em aplicativos *Desktop* como exemplo: botões, *checkbox*, *dialogs*, entre outros.

Com o uso da JSR-75 conseguiu-se acessar, no sistema de arquivos e diretórios do dispositivo, os arquivos de texto onde eram salvos os dados de cadastro e exames. Esses arquivos permitiram que fosse possível economizar recursos de memória do dispositivo móvel. Além disso, o uso da tecnologia *Bluetooth* foi fundamental para a comunicação sem fio confiável entre o *hardware* e o sistema móvel, requisito importante para este tipo de aplicação. Através da JSR-82 foi possível criar uma rede pessoal sem fio de curta distância onde o dispositivo móvel e o eletrocardiógrafo se conectam e se comunicam.

O uso da plataforma Java ME promoveu portabilidade à aplicação, já que, atualmente, grande parte dos dispositivos móveis apresenta suporte a aplicativos Java, além das inúmeras ferramentas gratuitas disponíveis na internet para desenvolvimento e manutenção de aplicações usando essa linguagem. Porém, em alguns momentos foi preciso um esforço maior para implementar funções triviais do Java SE que não estão disponíveis no Java ME, pois essa plataforma para dispositivos móveis é, em algumas partes, limitada devido aos requisitos de memória e processamento dos aparelhos aos quais ela se destina.

O programa também foi instalado e testado em outros dispositivos, tais como alguns celulares da Sony Ericsson, LG e Samsung. Os resultados foram os mesmos obtidos com os celulares da Nokia citados no decorrer deste trabalho. Já a capacidade de armazenamento dos exames depende apenas do sistema de arquivos utilizado no dispositivo. Como exemplo, pode-se citar o FAT32 que suporta tamanho de arquivos até 4GB.

De acordo com o Dr. Roberto Lima Farias, o sistema é apresentado como uma ferramenta útil; simples; moderna; e de baixo custo, sendo, por esse motivo, bem mais acessível do que vários outros equipamentos disponíveis no mercado. No caso dos planos de saúde, este equipamento pode diminuir a dificuldade dos convênios em pagar esse tipo de exame. Já no caso dos hospitais públicos, este equipamento pode contribuir para que esse tipo de exame seja massivo, permitindo um amplo acesso da população através do programa de saúde da família.

Para que seja possível o uso do equipamento em hospitais, faz-se necessário atender às normas brasileiras compulsórias de segurança para equipamentos eletromédicos, bem como a homologação do equipamento junto à Agência Nacional de Vigilância Sanitária, a ANVISA.

5.3 Contribuições

Este trabalho trouxe para a comunidade científica e para os profissionais da saúde algumas importantes contribuições. A primeira delas é o desenvolvimento de um sistema que contempla a interface homem-computador de um eletrocardiógrafo portátil, que pode ser instalado em qualquer celular com suporte a aplicações Java e que possua a API *Bluetooth*, proporcionando uma maior portabilidade e conseqüente diminuição dos custos. Outra contribuição é a disponibilização do código fonte do sistema para estudos e melhorias por parte da comunidade acadêmica.

5.4 Trabalhos Futuros

Sugere-se que, em trabalhos futuros, a função de envio de exame usando a interface GPRS seja implementada de forma que não seja preciso sair do sistema para efetuar esse envio. Além disso, pode-se propor a opção de envio automático do exame realizado para o email do médico pré-cadastrado, juntamente com uma mensagem de texto para o celular deste. Outras sugestões seriam: desenvolvimento de uma nova versão do *software* utilizando a plataforma *Android*, a qual permite que códigos sejam escritos em linguagem Java para controlar os dispositivos móveis via bibliotecas desenvolvidas pela Google; integração com o projeto Registro de Saúde Inteligente Baseado na Web e *Software* Livre (VALENTE, 2010), para a edição do exame com adição de comentários e registros de diagnósticos; e adequação às normas brasileiras de segurança para equipamentos eletromédicos.

Segundo sugestão do Dr. Roberto Lima Farias, pode-se implementar uma versão do sistema para uso próprio dos pacientes que contemple apenas o registro das arritmias que não duram tempo suficiente para o paciente chegar à clínica ou hospital e realizar o registro.

Apêndice A

ENTREVISTA

Entrevistado: Dr. Roberto Lima Farias. Mestre em Cardiologia pela Universidade Federal de São Paulo – Escola paulista de Medicina.

Pergunta 1: O sistema é fácil de usar?

Resposta: “Sim. O sistema se mostrou bastante simples e prático, apesar de haver a necessidade de um breve treinamento antes de usá-lo pela primeira vez”.

Pergunta 2: As cores do sistema estão apropriadas para um *software* médico?

Resposta: “Sim, pois o mais importante, além das cores atraentes, é que cada componente e cada instrução esteja bem visível na tela”.

Pergunta 3: O tamanho das letras está apropriado?

Resposta: “As letras poderiam ser um pouco maiores, a ponto de ficar mais legível sem perder o leiaute simples das telas”.

Pergunta 4: Em quê esse *software* ajuda com relação ao diagnóstico precoce de doenças cardiovasculares?

Resposta: “A partir do momento em que o custo desse sistema se torna bem mais baixo que dos aparelhos encontrados no mercado, mais pessoas poderão se beneficiar desse tipo de exame, aumentando, assim, o número de pessoas tratadas mais cedo e a conseqüente diminuição da taxa de mortalidade relacionada com esses tipos de doenças”.

Pergunta 5: Em quê esse *software* atrapalha com relação ao diagnóstico precoce de doenças cardiovasculares?

Resposta: “Não vejo por que esse sistema atrapalharia no diagnóstico. Talvez, por não fazer parte de um dispositivo dedicado, pode-se acabar cometendo erros mais facilmente”.

Pergunta 6: Qual foi a maior dificuldade encontrada durante o uso da aplicação?

Resposta: “O cadastro dos pacientes feito através das teclas do celular”.

Pergunta 7: O software deve gerar algum tipo de relatório?

Resposta: “Não, deve apenas disponibilizar a possibilidade de imprimir o exame. Como o papel quadriculado é caro, para deixar o uso dessa funcionalidade mais barato, não precisa imprimir nesse tipo de papel, basta imprimir o exame junto com o fundo quadriculado no papel A4”.

Pergunta 8: É possível efetuar o diagnóstico direto no celular ou é imprescindível o envio do exame para o PC para tal fim?

Resposta: “Para uma análise bem detalhada é importante que o exame seja enviado para o PC. Tomando como exemplo a detecção da arritmia, o celular é muito útil para detecção e registro desta, após isso é necessária uma análise mais criteriosa, com mais detalhe no PC”.

Pergunta 9: Quais as melhorias que podem ser indicadas à aplicação?

Resposta: “Na anamnese poderia haver um questionário eletrônico com as perguntas mais rotineiras em consultórios com apenas duas opções de resposta: sim e não, para evitar que o usuário do sistema tenha que ficar digitando muitas informações no teclado do celular. No cadastro de pacientes, além da data de nascimento e data do exame, é importante acrescentar as informações de hora, minuto e segundo da realização do exame. Na realização do exame, é importante que se diminua cada vez mais a interferência do ruído sobre o sinal. Outra melhoria importante seria permitir que os exames sejam realizados contemplando pelo menos duas derivações, combinando D1 ou aVL com D2 ou D3 ou aVF. Dessa forma, é possível usar dois vetores com direção e sentido diferentes”.

Pergunta 10: Você usaria este sistema no seu consultório?

Resposta: “Sim, pois diminuiria consideravelmente meus custos, além de me proporcionar maior praticidade na realização dos exames”.

Pergunta 11: Qual a sua opinião sobre o sistema?

Resposta: “O sistema é apresentado como uma ferramenta útil; simples; moderna; e de baixo custo, sendo, por esse motivo, bem mais acessível que várias outras ferramentas médicas disponíveis no mercado. No caso dos planos de saúde, essa ferramenta pode diminuir a dificuldade dos convênios em pagar esse tipo de exame. Já no caso dos hospitais públicos, essa ferramenta pode contribuir para que esse tipo de exame seja bem mais utilizado”.

Apêndice B

DERIVAÇÕES ELETROCARDIOGRÁFICAS

Na superfície do corpo existem diferenças de potencial, consequentes aos fenômenos elétricos gerados durante a excitação cardíaca. Estas diferenças podem ser medidas e registradas tendo-se uma noção satisfatória do tipo e da intensidade das forças elétricas do coração. Para isto são utilizados galvanômetros de tipo particular que constituem as unidades fundamentais dos eletrocardiógrafos. Nesse sentido os pontos do corpo a serem explorados são ligados ao aparelho de registro por meio de fios condutores (eletrodos). Dessa forma, obtém-se as chamadas derivações que podem ser definidas de acordo com a posição dos eletrodos. Ela é dita bipolar quando os dois eletrodos se encontram a mesma distância do ponto de vista elétrico do coração. Derivações unipolares são aquelas em que o traçado obtido se deve às variações de potencial recolhidas por um dos eletrodos dito explorador. O outro fica relativamente muito mais distante do coração, situado num ponto de potencial zero denominado eletrodo indiferente. Eletrodo explorador e indiferente são também utilizados para as derivações bipolares, sendo o primeiro o eletrodo positivo e o segundo o eletrodo negativo. Por convenção, registram-se curvas ditas positivas quando um dos eletrodos admitido como explorador está orientado para as áreas que se comportam como positivas em relação às que se encontram voltadas para o outro eletrodo (eletrodo indiferente) (OLIVEIRA, 2007).

As derivações eletrocardiográficas são divididas em dois grupos, no plano frontal (Padrão - D1,D2,D3 e Unipolares - aVR, aVL,aVF) e no plano horizontal (Precordiais - V1, V2,... , V6).

As derivações no plano frontal são:

- a) D1: diferença de potencial entre braços esquerdo e direito;
- b) D2: diferença de potencial entre pernas e braço direito;
- c) D3: diferença de potencial entre pernas e braço esquerdo.

As derivações no plano horizontal são:

- a) V1: Quarto espaço intercostal, bordo direito externo;

- b) V2: Quarto EI, bordo esquerdo externo;
- c) V3: entre V2 e V3;
- d) V4: Quinta EI, linha hemiclavicular;
- e) V5: Quinta EI, linha axilar anterior;
- f) V6: Quinta EI, linha axilar média.

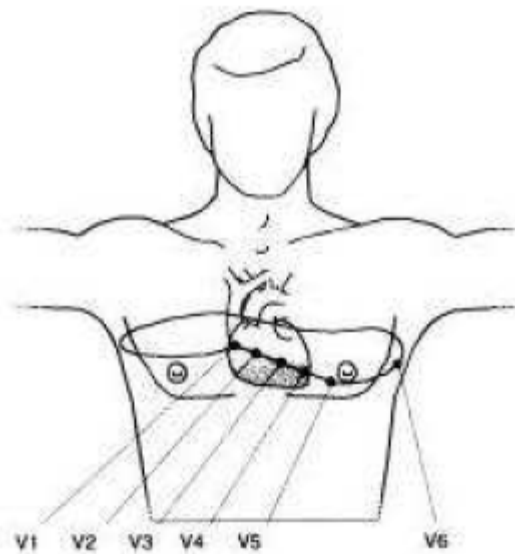


Figura B.1: derivações Precordiais (WINTER; WEBSTER, 1983).

Na eletrocardiografia prática, aceita-se como verdadeiras as seguintes proposições:

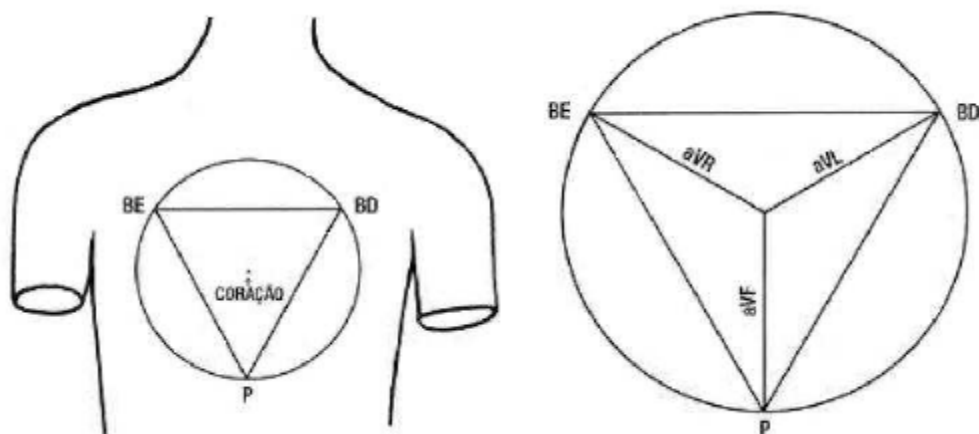


Figura B.2: triângulo de Einthoven.

- a) o tronco humano é uma esfera condutora homogênea, de tamanho infinito;

b) o coração está no centro, ocupando o tamanho de um ponto.

Assim, o conceito de Triângulo de Einthoven pressupõe que a união das extremidades (braços e perna) forma um triângulo equilátero, e que o coração considerado um ponto, ocupe o centro do triângulo e da esfera (tronco).

Desta forma, pode-se aplicar algumas leis físicas de fluxo elétrico e assim obtem-se:

a) aVR, aVL e aVF são potenciais absolutos do BD, BE e perna em relação ao coração, que está no centro do triângulo;

b) traçando paralelas em relação às linhas do triângulo já desenhado. Inicialmente em D1, depois em D2 e a seguir em D3. Repetindo o mesmo procedimento em aVR, aVL e aVF, respectivamente, como é visto no esquema da Figura B.3.

Como em D1, o braço esquerdo é o lado positivo, colocamos no desenho o lado esquerdo em D1 grifado (positivo) e o lado direito em linhas pontilhadas. E assim respectivamente em relação a D2 (perna é positiva) e em D3 (perna positiva). Quanto às derivações unipolares (aVR, aVL e aVF) fica óbvio que cada uma deve ter, respectivamente, BD, BE e perna positivos. Agora, colocando os ângulos no desenho, tem-se por convenção a Figura B.4 (OLIVEIRA, 2007).

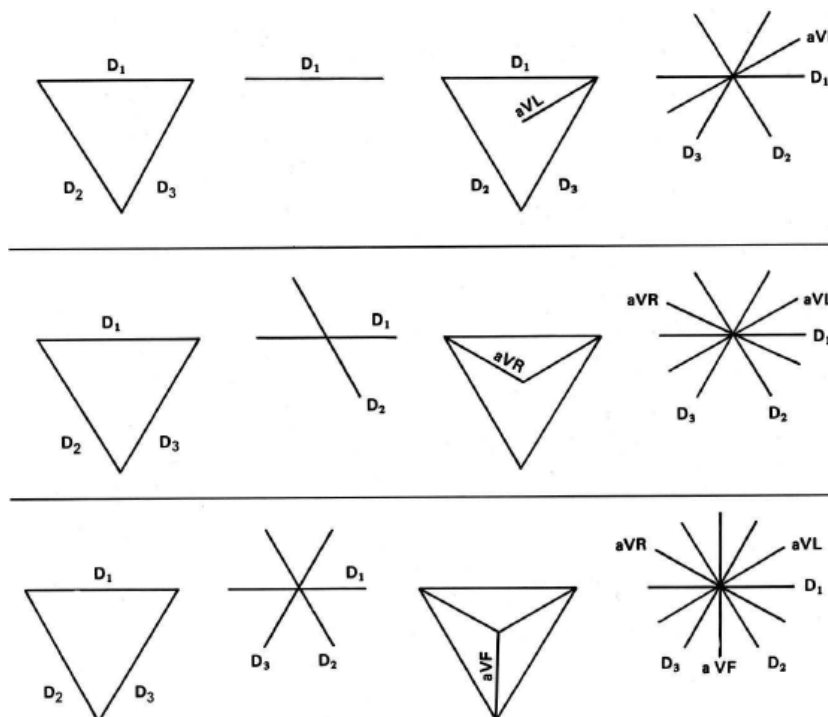


Figura B.3: esquema de potenciais absolutos.

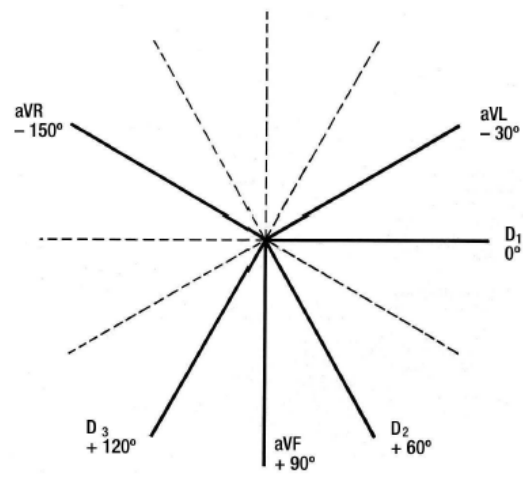


Figura B.4: convenção de derivações.

Referências Bibliográficas

AGUIAR, R. O. *Classificação Não-Supervisionada de Sinais de Eletrocardiograma*. Universidade Federal do Espírito Santo, Vitória, 2006.

AHA. American Heart Association. 2009. *Heart Disease and Stroke Statistics*. Disponível on-line em junho de 2010 no endereço <http://www.heart.org/HEARTORG/>.

ARAUJO, D. V. FERRAZ M. B. *Impacto econômico do tratamento da cardiopatia isquêmica crônica no Brasil: o desafio da incorporação de novas tecnologias cardiovasculares*. São Paulo, 2005.

BERNE, R. M. e LEVY, M. N. et al. : *Fisiologia Médica*. 4 ed. Rio de Janeiro: Guanabara Koogan, 2000.

BLUETOOTH SIG (2007). Specification of the Bluetooth System. Disponível on-line em junho de 2010 no endereço <http://www.Bluetooth.com>.

CPQD, *A revolução da comunicação móvel*. 2008. Disponível on-line em julho de 2010 no endereço <http://www.cpqd.com.br/highlights/2365-a-revolucao-da-comunicacao-movel.html>.

DATASUS, Banco de dados do Sistema Único de Saúde. 2007. ***Indicadores de Mortalidade. Ministério da Saúde. Taxa de mortalidade específica por doenças no aparelho circulatório. Taxa de Doença isquêmica do coração***. Disponível on-line em junho de 2010 no endereço <http://tabnet.datasus.gov.br/cgi/idb2009/matriz.htm>.

GUYTON, A. C. *Fisiologia Humana*. 6. ed. Rio de Janeiro. Guanabara Koogan, 1988.

GUYTON, A. C.; HALL, J. E. *Textbook of Medical Physiology*. 11 ed. Elsevier Saunders, 2006.

GUYTON, A. C. *Tratado de fisiologia Médica*. 9 ed. Rio de Janeiro: Guanabara Koogan, 1997.

INTELIMED (2010). *Suporte remoto a Diagnóstico Médico utilizando Tecnologias Inteligentes*. Disponível on-line em julho de 2010 no endereço <http://www.intelimed.com.br/empresa.php>.

INSTITUTO DE CIÊNCIAS BIOLÓGICAS. Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2006. Disponível on-line em junho de 2010 no endereço <http://www.icb.ufmg.br/>.

JCP (2009). *Jsr 271: Mobile information device profile 3*. Disponível on-line em junho de 2010 no endereço <http://jcp.org/en/jsr/detail?id=271>.

- JOHNSON, M. T. (2007). *Java para Dispositivos Móveis – Desenvolvendo Aplicações com J2ME*. Novatec, 1th edition.
- KÖHLER, B. et al. *The principles of software QRS detection*. IEEE Eng. Med. Biol. Mag, v. 21, p. 42-57, 2002.
- LI, S.; KNUDSEN, J. (2005). *Beginnig J2ME From Novice to Professional. Build Java-based applications and games for mobile and wireless devices like cell phones, PDAs, and more-start at the beginning, and learn the real-world techniques*. 3td edition. Apress, 2005002E.
- LIMA NETO, L. A. (2010). *Eletrocardiógrafo Portátil com uma Derivação e Comunicação Bluetooth*. Dissertação (Mestrado). Universidade Federal do Ceará, Fortaleza, 2010.
- LUZ, M. (2009). MIDP 3.0: *O que vem por ai. A nova especificação do principal perfil do Java ME*. Revista Java Magazine, edição 44.
- MUCHOW, J. W. Core J2ME Technology & MIDP. *The Sun Microsystems*. Press Java Series, December 2001.
- NETO, A. M. (2008). *Lwuit: Swing para Java ME*. Revista Java Magazine, edição 60.
- NOKIA (2008). *JSR 75 File Connection API*. Disponível on-line em Julho de 2010 no endereço [http://wiki.forum.nokia.com/index.php/JSR 75 File connection API](http://wiki.forum.nokia.com/index.php/JSR_75_File_connection_API).
- OLIVEIRA, F. I. (2007). *Transformada de Hilbert sobre bases de Wavelets. Detecção de Complexos QRS*. Dissertação (Mestrado). Universidade Federal do Ceará, Fortaleza, 2007.
- OMS, Organização Mundial de Saúde. 2007. *Cardiovascular Diseases*. Disponível on-lin em junho de 2010 no endereço <http://www.who.int/mediacentre/factsheets/fs317/en/index.html>.
- ORACLE (2004). *Getting Started with the FileConnection APIs*. Disponível on-line em julho de 2010 no endereço <http://developers.sun.com/mobility/apis/articles/fileconnection/>.
- ORACLE (2010). *Lesson: Packaging programs in jar files*. Disponível on-line em julho de 2010 no endereço <http://download.oracle.com/javase/tutorial/deployment/jar/>.
- PASTORE, C. A. (2010). *O eletrocardiograma continua salvando vidas*. Disponível on-line em agosto de 2010 no endereço <http://blogs.estadao.com.br/dicas-de-saude>.
- PÓVOA, F. M. (2005). *Um jogo J2ME/MIDP multi-usuário usando bluetooth. Web Mobile*, v.1 n. 4, p. 45-5.
- PÓVOA, L. M. C. *Distribuição Dois ensaios econômicos sobre a distribuição geográfica dos médicos*. Dissertação (Mestrado). Centro de Desenvolvimento e Planejamento Regional. Faculdade de Ciências Econômicas. Universidade Federal de Minas Gerais, Belo Horizonte, 2004.

SIQUEIRA, R. da S. *Um Método de Compreensão de Sinais de Eletrocardiograma (ECG) Baseado na Forma dos Batimentos Cardíacos*. Dissertação (Mestrado) Universidade Federal do Ceará, Fortaleza, 2005.

SOMMERVILLE, I. *Engenharia de Software*. 8 ed. São Paulo: Pearson Addison-Wesley, 2007.

SUN (2007). *A survey of java me today (Update)*. Disponível on-line em julho de 2010 no endereço <<http://developers.sun.com/mobility/getstart/articles/survey/>>.

SUN (2006). *Mid profile*. Disponível on-line em julho de 2010 no endereço <http://java.sun.com/javame/reference/apis/jsr118/>.

VALENTE, I. R. S. *Registro Eletrônico de Saúde Inteligente Baseado na Web e Software Livre*. Dissertação (Mestrado) – Universidade Federal do Ceará, Fortaleza, 2010.

WEBSTER, J. G. *Medical Instrumental Application and Design*. 3 ed. John Wiley Sons, Inc, 1998.

ZANFOLIM, L. C.; FERNANDES R. C. *Desenvolvendo um Estudo de Caso Utilizando a Plataforma Java ME*. Monografia – Universidade Estadual do Mato Grosso do Sul, Dourados, 2009.