

Bernardo Nunes Gonçalves

*Projeto de um ECG-Wrapper para a  
plataforma Infracore*

Vitória - ES, Brasil

20 de dezembro de 2006

Bernardo Nunes Gonçalves

*Projeto de um ECG-Wrapper para a  
plataforma Infracore*

Monografia apresentada para obtenção do  
Grau de Bacharel em Ciência da Com-  
putação pela Universidade Federal do Es-  
pírito Santo.

Orientador:

José Gonçalves Pereira Filho

Co-Orientadores

Giancarlo Guizzardi

Rodrigo Varejão Andreão

DEPARTAMENTO DE INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

20 de dezembro de 2006

Monografia de Projeto Final de Graduação sob o título “*Projeto de um ECG-Wrapper para a plataforma Infrared*”, defendida por Bernardo Nunes Gonçalves e aprovada em 20 de dezembro de 2006, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

---

Prof. Dr. José Gonçalves Pereira Filho  
Orientador

---

Prof. Dr. Saulo Bortolon  
Universidade Federal do Espírito Santo

---

Prof. Dr. Álvaro Barbosa  
Universidade Federal do Espírito Santo

# *Resumo*

Nos últimos anos, o desenvolvimento acelerado das tecnologias de informação e comunicação vem rompendo barreiras territoriais e tornando a computação cada vez mais presente nas atividades humanas. Esse novo cenário traz consigo a possibilidade de se explorar uma nova gama de serviços e aplicações, em especial as aplicações móveis e sensíveis ao contexto, que exploram o contexto dinâmico de seus usuários através de dispositivos portáteis. O uso de tais aplicações manifesta-se em domínios diversificados. Merece destaque a área de saúde, em que as novas tecnologias têm estimulado o atendimento médico à distância. Dessa forma, a Telecardiologia tem oferecido cenários potenciais envolvendo o monitoramento remoto e sensível ao contexto de pacientes através da transmissão do eletrocardiograma (ECG).

No entanto, a ampla variedade de dispositivos de aquisição de dados clínicos aliada à complexidade de se lidar com seus protocolos de comunicação e formatos de dados específicos têm apontado a necessidade do emprego de softwares capazes de encapsular esses dados a fim de separar as tarefas de aquisição e utilização da informação clínica.

Nesta direção, este trabalho oferece o projeto de um software destinado a tal propósito, através do encapsulamento dos dados de ECG adquiridos dos pacientes que, segundo pesquisas, deve ser feito através de abordagens guiadas pelas demandas dos serviços médicos, ao invés de serem direcionadas pelas tecnologias disponíveis.

# *Abstract*

# *Dedicatória*

# *Agradecimientos*

# Sumário

## Lista de Figuras

## Lista de Tabelas

<b>1</b>	<b>Introdução</b>	p. 17
1.1	Objetivos e Justificativas . . . . .	p. 19
1.2	Metodologia . . . . .	p. 21
1.3	Estrutura do Trabalho . . . . .	p. 22
<b>2</b>	<b>Projetos e Trabalhos Relacionados</b>	p. 23
2.1	<i>Context Toolkit</i> . . . . .	p. 23
2.2	<i>Mobile e-Health Plataform</i> . . . . .	p. 26
2.3	<i>Vital Signs Protocol Format</i> . . . . .	p. 30
2.4	INFRAWARE . . . . .	p. 33
2.5	CARDOM . . . . .	p. 35
2.6	TeleCardio . . . . .	p. 36
2.7	Conclusão do Capítulo . . . . .	p. 38
<b>3</b>	<b>Domínio de Aquisição e Encapsulamento de Contexto</b>	p. 40
3.1	Definindo Contexto . . . . .	p. 40
3.2	Relação entre Domínios e Aplicações . . . . .	p. 41
3.3	Encapsulamento de Contexto . . . . .	p. 43

3.4	Aquisição de Contexto na Infracore . . . . .	p. 46
3.5	Wrappers Contextuais . . . . .	p. 49
3.6	Requisitos e Arquitetura Conceitual do Context-Wrapper . . . . .	p. 51
3.7	Conclusão do Capítulo . . . . .	p. 55
<b>4</b>	<b>Domínio de Representação e Transmissão de ECG</b>	<b>p. 56</b>
4.1	Camadas de Aplicação e Apresentação do Modelo OSI . . . . .	p. 56
4.2	Eletrocardiografia . . . . .	p. 58
4.3	AHA/MIT-BIH . . . . .	p. 61
4.3.1	Registros . . . . .	p. 63
4.3.2	Sinais, Amostras, e Tempo . . . . .	p. 63
4.3.3	Anotações . . . . .	p. 64
4.3.4	Discussão sobre o AHA/MIT-BIH . . . . .	p. 65
4.4	SCP-ECG . . . . .	p. 65
4.4.1	<i>SCP-ECG Overview</i> . . . . .	p. 66
4.4.2	Estrutura Detalhada do Registro SCP-ECG . . . . .	p. 66
4.5	XML . . . . .	p. 68
4.5.1	XML Schema (XSD) . . . . .	p. 69
4.6	<i>FDA XML Data Format (FDADF)</i> . . . . .	p. 70
4.6.1	Sinal . . . . .	p. 72
4.6.2	Anotações . . . . .	p. 73
4.6.3	Visualização de ECG . . . . .	p. 73
4.7	<i>ecgML</i> . . . . .	p. 75
4.7.1	Estrutura <i>ecgML</i> . . . . .	p. 75
4.7.2	Comentários Gerais sobre o <i>ecgML</i> . . . . .	p. 80

4.8	Resumo do Domínio e Especificação de Requisitos . . . . .	p. 80
4.8.1	Resumo do Domínio . . . . .	p. 81
4.8.2	Especificação de Requisitos . . . . .	p. 81
4.9	Conclusão do Capítulo . . . . .	p. 83
<b>5</b>	<b>Especificação de um Formato de Representação de ECG</b>	p. 85
5.1	<i>ecgAware</i> . . . . .	p. 86
5.2	Comentários Gerais e Avaliação do <i>ecgAware</i> . . . . .	p. 107
<b>6</b>	<b>Cenários de Uso e Requisitos do <i>ECG-Wrapper</i></b>	p. 110
6.1	Elementos e Configurações . . . . .	p. 111
6.2	Cenário de Monitoramento Domiciliar . . . . .	p. 113
6.3	Cenário da Unidade móvel de emergência (ambulância) . . . . .	p. 114
6.4	Cenário de Monitoramento em Ambiente Externo . . . . .	p. 115
6.5	Casos de Uso e Requisitos do Sistema . . . . .	p. 116
6.5.1	Atores . . . . .	p. 117
6.5.2	Descrição de Casos de Uso . . . . .	p. 118
6.5.3	Descrição de Requisitos . . . . .	p. 119
6.6	Conclusão do Capítulo . . . . .	p. 120
<b>7</b>	<b>Projeto do <i>ECG-Wrapper</i></b>	p. 121
7.1	Projeto de Alto Nível . . . . .	p. 121
7.1.1	Visão Geral do Sistema . . . . .	p. 121
7.1.2	Arquitetura do <i>ECG-Wrapper</i> . . . . .	p. 122
7.2	Camada de Comunicação Wireless PAN . . . . .	p. 124
7.2.1	<i>ZigBee Overview</i> . . . . .	p. 124

7.3	Camada de Interpretação . . . . .	p. 129
7.4	Camada de Tradução . . . . .	p. 130
7.5	Camada de Comunicação <i>Web Service</i> . . . . .	p. 132
7.5.1	<i>Web Services</i> . . . . .	p. 132
7.5.2	Modelos de entrega da informação contextual à <i>Infraware</i> . . . . .	p. 136
7.6	GUI . . . . .	p. 137
7.7	Conclusão do Capítulo . . . . .	p. 138
<b>8</b>	<b>Implementação de Protótipo</b>	p. 139
8.1	Arquitetura do Protótipo . . . . .	p. 139
8.2	Interfaces e Funcionalidades . . . . .	p. 141
8.2.1	Interface Principal . . . . .	p. 141
8.2.2	Cadastro do Paciente . . . . .	p. 142
8.2.3	Sessão de Monitoramento . . . . .	p. 143
8.2.4	Disponibilização de Estudo de ECG . . . . .	p. 146
8.3	Conclusão do Capítulo . . . . .	p. 147
<b>9</b>	<b>Conclusões e Trabalhos Futuros</b>	p. 149
9.1	Conclusões Gerais . . . . .	p. 149
9.2	Trabalhos Futuros . . . . .	p. 150
	<b>Referências</b>	p. 153
	<b>Apêndice A – Exemplo de <i>XML Document</i> no formato <i>ecgAware</i></b>	p. 159

# *Lista de Figuras*

1	Exemplo de configuração dos componentes do <i>Context Toolkit</i> [1]. . .	p. 26
2	(a) Visão geral da <i>Mobile e-Health</i> , e (b) Distribuição de componentes na sua arquitetura [12]. . . . .	p. 28
3	Arquitetura do software componente da MBU [12]. . . . .	p. 29
4	Processo de codificação da informação [18]. . . . .	p. 32
5	Arquitetura Geral da Plataforma Infracore. . . . .	p. 34
6	Domínios verticais e horizontais [32]. . . . .	p. 42
7	Relações entre domínios e aplicações [34]. . . . .	p. 42
8	Separação das diferentes tarefas, de utilização e de aquisição de contexto.	p. 44
9	Aquisição de dados utilizando <i>GUI Widgets</i> . . . . .	p. 45
10	Diferença entre (a) sistema tradicional e (b) sistema <i>Context-aware</i> . .	p. 46
11	Componente de Acesso e Integração de Dados na Infracore. . . . .	p. 47
12	Níveis de esquemas e suas relações [40]. . . . .	p. 48
13	Transformação de dados obtidos de sensores em informação [41]. . . .	p. 50
14	Arquitetura conceitual proposta para o <i>Context-Wrapper</i> . . . . .	p. 54
15	Camadas de Aplicação e Apresentação no Modelo OSI [18]. . . . .	p. 57
16	Ondas elementares, segmentos, e intervalos de um batimento elementar de ECG [50]. . . . .	p. 59
17	(a) Posições dos eletrodos nas derivações dos membros; (b) As doze derivações padronizadas na eletrocardiografia. . . . .	p. 60
18	Estrutura do registro ECG [4]. . . . .	p. 66

19	Registro ECG [61]. . . . .	p. 66
20	Estrutura da seção de um registro SCP-ECG [61]. . . . .	p. 67
21	Meta-modelo XSD e um modelo <i>XML Document</i> gerado a partir desse XSD. . . . .	p. 70
22	Exemplo de um R-MIM [74]. . . . .	p. 72
23	<i>Screenshot</i> de uma aplicação de visualização de ECG [74]. . . . .	p. 74
24	Diagrama do elemento <i>ECGRecord</i> [70]. . . . .	p. 76
25	Diagrama do elemento <i>PatientDemographics</i> [64]. . . . .	p. 76
26	Diagrama do elemento <i>Record</i> [70]. . . . .	p. 77
27	Diagrama do elemento <i>ClinicalProtocol</i> [64]. . . . .	p. 77
28	Diagrama do elemento <i>RecordingDevice</i> [64]. . . . .	p. 78
29	Diagrama do elemento <i>RecordData</i> [70]. . . . .	p. 78
30	Diagrama do elemento <i>Waveforms</i> [64]. . . . .	p. 79
31	Diagrama do elemento <i>Annotations</i> [64]. . . . .	p. 79
32	Elemento raiz <i>ECGStudy</i> . . . . .	p. 87
33	Elemento <i>PatientInformation</i> . . . . .	p. 88
34	Elemento <i>Demographics</i> . . . . .	p. 89
35	Elemento <i>EPR</i> . . . . .	p. 90
36	Elemento <i>Record</i> . . . . .	p. 91
37	Elemento <i>RecordLeads</i> . . . . .	p. 92
38	Elemento <i>RecordLead</i> . . . . .	p. 93
39	Elemento <i>XValues</i> . . . . .	p. 93
40	Elemento <i>YValues</i> . . . . .	p. 94
41	Elemento <i>IntValue</i> . . . . .	p. 95
42	Elemento <i>BinaryData</i> . . . . .	p. 95

43	Elemento <i>LeadAnnotations</i> . . . . .	p. 96
44	Elemento <i>LeadPointNotation</i> . . . . .	p. 97
45	Elemento <i>LeadWaveNotation</i> . . . . .	p. 98
46	Estrutura comum aos elementos filhos de <i>LeadWaveNotation</i> . . . . .	p. 98
47	Elemento <i>LeadMeasurements</i> . . . . .	p. 99
48	Elemento <i>RecordWaveNotation</i> . . . . .	p. 100
49	Elemento <i>RecordAnnotations</i> . . . . .	p. 100
50	Elemento <i>RecordPointNotation</i> . . . . .	p. 101
51	Elemento <i>RecordMeasurements</i> . . . . .	p. 101
52	Elemento <i>RecordingDevice</i> . . . . .	p. 102
53	Elemento <i>Context</i> . . . . .	p. 103
54	Elemento <i>ClinicalProtocol</i> . . . . .	p. 104
55	Elemento <i>Report</i> . . . . .	p. 105
56	Estrutura hierárquica do <i>ecgAware</i> . . . . .	p. 106
57	Distinção entre aquisição e utilização da informação médica. . . . .	p. 108
58	Ilustração de uma possível configuração da UMS [12]. . . . .	p. 112
59	Monitoramento de ECG em domicílio ou Unidade de Saúde. . . . .	p. 113
60	Ambulância equipada para Telecardiologia [78]. . . . .	p. 114
61	Monitoramento de sinais cardíacos enquanto o paciente realiza atividade física [79]. . . . .	p. 116
62	(a) PDA com holter incorporado. (b) <i>Vital Jacket</i> [80]. . . . .	p. 116
63	Diagrama de Casos de Uso do <i>ECG-Wrapper</i> . . . . .	p. 118
64	Visão geral do sistema. . . . .	p. 122
65	Arquitetura do sistema. . . . .	p. 123
66	Arquitetura <i>ZigBee</i> [85]. . . . .	p. 125

67	Topologia estrela no <i>Bluetooth: piconets</i> e <i>scatternets</i> [87]. . . . .	p.126
68	Topologias <b>(a)</b> <i>star</i> , e <b>(b)</b> <i>mesh</i> , no <i>ZigBee</i> . . . . .	p.128
69	Estrutura do algoritmo de classificação do sinal ECG [22]. . . . .	p.129
70	Fraco acoplamento entre sistemas que interagem através de <i>Web Services</i> .p.	133
71	Arquitetura SOA [90]. . . . .	p.134
72	Pilha de protocolos da arquitetura de <i>Web Service</i> [91]. . . . .	p.134
73	Interação WS consumidor-provedor [94]: <b>(a)</b> direta, <b>(b)</b> através da descoberta de serviços. . . . .	p.136
74	Diagrama de pacotes desse protótipo do <i>ECG-Wrapper</i> . . . . .	p.140
75	Interface principal do protótipo de demonstração. . . . .	p.141
76	Interface para inserção de dados do paciente. . . . .	p.142
77	Interface que permite o início de uma nova sessão. . . . .	p.144
78	Interface de monitoramento de uma sessão. . . . .	p.145
79	Janela de alerta da ocorrência de um evento de urgência. . . . .	p.146
80	Janela de consulta da aplicação cliente. . . . .	p.147

# *Lista de Tabelas*

1	Requisitos do <i>Context-Wrapper</i> . . . . .	p. 53
2	Descrição do conteúdo de cada seção SCP-ECG [61]. . . . .	p. 67
3	Dicionário de Termos do domínio de representação digital de ECG. . . . .	p. 82
4	Requisitos de Representação e Transmissão de ECG. . . . .	p. 83
5	Atributos e elementos filhos de um estudo de ECG. . . . .	p. 87
6	Atributos e elementos filhos de <i>PatientInformation</i> . . . . .	p. 88
7	Elementos que o compõem <i>Demographics</i> . . . . .	p. 89
8	Elementos filhos que compõem <i>EPR</i> . . . . .	p. 90
9	Atributos e elementos filhos de <i>Record</i> . . . . .	p. 92
10	Conteúdo de <i>RecordLeads</i> . . . . .	p. 92
11	Conteúdo de <i>RecordLead</i> . . . . .	p. 93
12	Elementos filhos de <i>XValues</i> . . . . .	p. 94
13	Atributo e opções de elemento filho de <i>YValues</i> . . . . .	p. 94
14	Conteúdo de <i>IntValue</i> . . . . .	p. 95
15	Atributo e elementos filhos de <i>BinaryData</i> . . . . .	p. 96
16	Elemento <i>LeadAnnotations</i> . . . . .	p. 97
17	Elemento <i>LeadPointNotation</i> . . . . .	p. 97
18	Conteúdo de <i>LeadWaveNotation</i> . . . . .	p. 98
19	Sub-elementos de <i>Pwave</i> , <i>QRScomplex</i> , <i>Twave</i> , <i>Uwave</i> , e <i>Other-wave</i> . . . . .	p. 99

20	Atributos e elementos filhos de <i>LeadMeasurements</i> . . . . .	p. 99
21	Atributo e elementos filhos de <i>RecordingDevice</i> . . . . .	p. 102
22	Conteúdo do elemento <i>Context</i> . . . . .	p. 103
23	Elemento <i>ClinicalProtocol</i> . . . . .	p. 104
24	Elemento <i>Report</i> . . . . .	p. 105
25	Descrições dos casos de uso. . . . .	p. 119
26	Requisitos do <i>ECG-Wrapper</i> . . . . .	p. 120

# 1 *Introdução*

Os anos recentes experimentaram um enorme desenvolvimento nas tecnologias de informação e comunicação (TIC). Nesta verdadeira revolução tecnológica, o emprego em larga escala das tecnologias de informática e das telecomunicações ignora fronteiras territoriais e diminuem as diferenças culturais. Esse novo cenário traz consigo a possibilidade de se explorar uma nova gama de aplicações computacionais nas mais diversas atividades humanas e áreas do conhecimento. Um exemplo são as aplicações móveis sensíveis ao contexto (*context-aware mobile applications*) [1], que exploram o contexto dinâmico dos seus usuários provocado pela mobilidade e constante mudança no ambiente, e que fazem uso de novos dispositivos portáteis multifuncionais, como PDAs e telefones celulares.

No entanto, ao mesmo tempo em que as novas tecnologias eliminam fronteiras e fortalecem a chamada comunidade global, elas também aumentam a distância que separa regiões mais pobres e menos favorecidas sem acesso ao conhecimento e às facilidades proporcionadas pelo uso das modernas tecnologias daquelas onde os avanços científicos e tecnológicos estão cada vez mais presentes. Ao não conseguirem acesso ou mesmo acompanhar a velocidade das inovações, essas regiões e, por conseguinte, seus cidadãos, tornam-se cada vez mais dependentes dos grandes centros para a solução de problemas cotidianos. Na área de saúde, por exemplo, sobretudo no Brasil, é comum o deslocamento de um grande número de pacientes aos centros urbanos, em busca de um atendimento médico eficaz, que disponha de exames, profissionais e hospitais especializados.

Essa realidade tem fomentado o uso das novas tecnologias de informática e das telecomunicações em serviços e aplicações de atendimento médico à distância. Neste sentido, a Telemedicina [2] pode ser entendida como a distribuição de serviços de Saúde

e o compartilhamento de informações médicas utilizando as redes de telecomunicações, notadamente as modernas redes de alta velocidade. Ao permitir compartilhar conhecimentos dos grandes centros regionais, nacionais ou mesmo internacionais, esse novo paradigma de atendimento médico possibilita o diagnóstico em tempo real de pacientes localizados em regiões remotas, reduzindo gastos com hospitalização, diminuindo a taxa de ocupação dos leitos hospitalares bem como permitindo o debate médico e decisões sobre diagnósticos médicos com margem de erro reduzida.

Particularmente na Telemedicina, a sub-área da Telecardiologia tem se desenvolvido fundamentalmente com a transmissão do eletrocardiograma (ECG). O ECG é o exame de coração mais empregado em cardiologia, destacando-se por, ao comparado com outros exames de coração, ser rápido, barato e não invasivo. Através do ECG o médico pode diagnosticar uma ampla variedade de doenças do coração, pois a caracterização de cada cardiopatia se manifesta em modificações específicas da forma de onda do sinal [3]. Desde que as doenças do coração, por exemplo, em países europeus, estão entre as principais causas de mortalidade, é fácil notar a importância da eletrocardiografia nessa conjuntura, a tal ponto que estimativas indicam que mais de 100 milhões de ECGs são realizados anualmente só na Europa Ocidental [4]. E em direção a essa demanda, a Telecardiologia pode reduzir significativamente os custos envolvidos no tratamento de cardiopatias.

A Telemedicina tem se beneficiado do crescimento dos serviços e aplicações móveis sensíveis ao contexto. Os últimos avanços das tecnologias móveis e sem fio (ex: *Bluetooth*, *WiFi*, GPRS) e a popularização dos dispositivos móveis (PDAs, celulares, GPS e pequenos dispositivos médicos, como *holters*), possibilitaram o monitoramento remoto de pacientes. Essas aplicações podem ser programadas para utilizar informações contextuais a fim de selecionar e executar dinamicamente as ações que melhor atenderem às necessidades dos seus usuários. Assim, ao invés de tratar a mobilidade dos usuários como um problema, as aplicações *context-aware* exploram a natureza contextual provocada por ela, com a clara intenção de produzir serviços mais flexíveis, adaptáveis, ricos em funcionalidade e centrados no usuário.

Nessa direção, a plataforma de apoio a sistemas sensíveis ao contexto **Infraware** [5] visa prover a infra-estrutura necessária à execução de tal classe de aplicações em domínios variados, e lança mão das características e facilidades das tecnologias e dis-

positivos disponíveis, para se comunicar e prover os serviços dos mais variados para suas aplicações-alvo. Neste sentido, a Telecardiologia surge como um oportuno universo de instanciação da Infracare, na medida em que possui cenários em potencial que demandam monitoramento em tempo real incluindo o desencadeamento automático de ações. Levando isto em consideração, foi criado o projeto **TeleCardio** - *Telecardiologia a Serviço do Paciente em Ambientes Hospitalares e Residenciais* [6], que se aproveita da infra-estrutura oferecida pela plataforma Infracare, no telemonitoramento de pacientes.

Entretanto, não obstante os diversos fatores que favorecem o uso da Telemedicina na prestação de serviços médicos, um contratempo significativo é que a maior parte das aplicações computacionais desse gênero são guiadas pelas novas tecnologias disponíveis, ao invés de serem concebidas a partir das demandas existentes na área de saúde [7]. De fato, este problema não está presente somente nas aplicações da área de saúde; de acordo com [8], ele se manifesta no desenvolvimento de sistemas sensíveis ao contexto de maneira geral, impedindo uma maior difusão dessa classe de aplicações. Portanto, para atender às demandas da área de saúde se aproveitando das novas tecnologias disponíveis, faz-se necessária uma abordagem orientada ao domínio do problema, a fim de capturar a sua essência e criar modelos semanticamente adequados para a representação das informações do domínio.

Dessa forma, no desenvolvimento de sistemas *context-aware*, incluindo os que são concebidos para a área de saúde, é preciso a criação de softwares capazes de lidar com uma ampla variedade de dispositivos de hardware, capturar os dados contextuais por eles providos, e convertê-los para um formato semanticamente coerente com as necessidades humanas de utilização dessas informações. Com isso, softwares dessa natureza abstraem às aplicações usuárias da informação contextual toda a complexidade das múltiplas tecnologias (protocolos de comunicação, codificações em formato binário, etc) envolvidas nos processos de aquisição de contexto. Este trabalho é direcionado para a pesquisa e o desenvolvimento desse tipo de software.

## 1.1 Objetivos e Justificativas

Na linha de toda essa discussão, o foco central deste trabalho consiste no projeto do *ECG-Wrapper*, um software destinado a integrar a plataforma Infracare no mon-

itoramento remoto de pacientes cardíacos, para (i) adquirir os dados de ECG, ainda brutos, oriundos de um dispositivo de aquisição de eletrocardiograma ambulatorial em um formato específico, (ii) processá-los já incorporando uma semântica mais apurada, (iii) encapsulá-los em um formato adequado, e (iv) disponibilizá-los na Web para que eles sejam consumidos.

No entanto, para atender a essas funções relativas ao monitoramento remoto de sinais cardíacos de forma satisfatória, é necessário inicialmente entender questões pertinentes ao problema de aquisição e encapsulamento de contexto de maneira geral. Existe na literatura um número bastante limitado de trabalhos dedicados a compartilhar experiências relacionadas a esse problema. Em geral, a aquisição e encapsulamento de contexto é realizado de forma específica e o conhecimento adquirido nessas experiências mantém-se isolado nas mentes dos desenvolvedores. Levando isso em consideração, sobretudo pelo fato de que a plataforma Infraware é definida de forma genérica para ser instanciada em domínios diversificados do mundo real, este trabalho também se dedica à tarefa de documentar o conhecimento adquirido sobre encapsulamento de contexto, a fim de reduzir o esforço na adaptação da solução que será aqui proposta (referente ao ECG) para outros domínios a serem futuramente tratados pela Infraware.

Afora isso, uma característica marcante deste trabalho é possuir um caráter interdisciplinar que combina as áreas do conhecimento de Computação e de Medicina. Neste sentido, é alcançado um objetivo mais geral do projeto TeleCardio, que envolve a formação intelectual de profissionais de diversas áreas a fim de aplicar inovações tecnológicas na área de saúde, e com isso proporcionar uma melhoria na qualidade de serviços médicos voltados a pacientes crônicos e hospitalizados em domicílio ou em alguma Unidade de Saúde.

Nesse ínterim, é necessário ressaltar que este trabalho faz parte do projeto TeleCardio, mais especificamente da interação entre os trabalhos dos grupos de pesquisa do LPRM/DI/UFES e do LABTEL/DEE/UFES, que integram esse projeto. A partir do desenvolvimento do projeto INFRAWARE [9] no Laboratório de Pesquisas em Redes e Multimídia (LPRM), e do desenvolvimento do projeto CARDOM [10] no Laboratório de Telecomunicações (LABTEL), o projeto TeleCardio representa, sob o ponto de vista de ambos, a aplicação de estudos realizados de maneira genérica. Neste sentido, o objeto deste trabalho incorpora módulos desenvolvidos no LABTEL que serão referenciados e

apresentados ao longo do texto.

## 1.2 Metodologia

Para atingir os objetivos definidos na seção anterior, este trabalho segue uma metodologia dividida nas fases de análise, projeto, e implementação.

Na fase de análise, primeiramente foram realizadas pesquisas a respeito do encapsulamento de contexto, bem como diversas reuniões do grupo de pesquisa de sistemas sensíveis ao contexto do LPRM que incluíram esse assunto. Na investigação científica dessa linha de pesquisa, merece destaque o estudo de alguns trabalhos relacionados, sobretudo de [1] (seção 2.1), e de trabalhos desenvolvidos no LPRM. O Capítulo 2 remete-se a esse assunto.

Em seguida, foi feito um estudo dentro do escopo da Telecardiologia com vista a obter conhecimento a respeito desse domínio, que é necessário para tratar da representação e transmissão de eletrocardiogramas. Esse estudo incluiu a análise de padrões já existentes e amplamente utilizados na área médica, que são destinados ao mesmo propósito. Esta parte da análise é apresentada no Capítulo 3.

Em paralelo a tudo isso, foi estudado o universo de Engenharia de Domínio, de forma que o conhecimento adquirido através de tal estudo permitiu uma abordagem mais rica do tema principal de pesquisa, baseando-se em diferentes perspectivas. Uma breve discussão relacionada a essa questão é feita na primeira seção do Capítulo 3.

Em consequência da fase de análise, foram levantados requisitos pertinentes a: (i) o domínio de encapsulamento de contexto (Capítulo 3), (ii) o domínio de representação e transmissão de eletrocardiogramas (Capítulo 4), e (iii) a combinação desses dois domínios aplicada em cenários reais vislumbrados no projeto TeleCardio (Capítulo 6).

Na fase de projeto, buscou-se identificar as melhores soluções de tecnologia para atender aos requisitos obtidos na fase de análise, e com isso definir a arquitetura do sistema. Para tal propósito, foram realizados estudos de diversas tecnologias, com destaque para XML, *Web Services*, *ZigBee*, e novamente os padrões da área médica já investigados anteriormente com um enfoque mais conceitual. Esse assunto é tratado no Capítulo 7.

Finalmente, na fase de implementação, foi desenvolvido um protótipo do *ECG-Wrapper* que devido a restrições de tempo implementa somente alguns dos requisitos expressos neste trabalho, mas que fornece um ponto de partida para um processo iterativo de aperfeiçoamento, e posterior avaliação do projeto do sistema.

## 1.3 Estrutura do Trabalho

Em consequência da metodologia apresentada na seção anterior, o restante deste trabalho é organizado da seguinte forma:

- Capítulo 2: apresenta um pequeno resumo de trabalhos relacionados que serviram como base para este trabalho, bem como os projetos de pesquisa referentes a este trabalho;
- Capítulo 3: estabelece muitas das bases teóricas deste trabalho, definindo contexto, a relação entre domínios e aplicações e, discutindo a tarefa de aquisição de contexto no universo de sistemas *context-aware*; este capítulo gera uma especificação de requisitos para o encapsulamento de contexto;
- Capítulo 4: apresenta um estudo de padrões e tecnologias da área médica associados a representação e transmissão de ECG, e extrai a partir deles requisitos para essa tarefa;
- Capítulo 5: guiado pelos requisitos obtidos no capítulo anterior, este capítulo apresenta a especificação de um formato de representação de ECG;
- Capítulo 6: fornece uma descrição dos principais cenários de uso do *ECG-Wrapper*, derivando a partir deles requisitos gerais do sistema;
- Capítulo 7: contém o projeto arquitetural do sistema discutindo cada componente que o integra em detalhes;
- Capítulo 8: apresenta um protótipo de demonstração do *ECG-Wrapper*;
- Capítulo 9: contém as conclusões gerais e as possibilidades vislumbradas de trabalhos futuros.

## 2 *Projetos e Trabalhos Relacionados*

O objetivo deste capítulo é apresentar e discutir brevemente projetos e trabalhos relacionados ao objeto desta monografia e usados como referência durante as pesquisas, buscando com isso posicionar o *ECG-Wrapper* diante dos mesmos e da linha de pesquisa de sistemas *context-aware*.

Dessa forma, serão abordados em cada seção deste capítulo (de 2.1 a 2.6), os seguintes projetos e trabalhos:

- *Context Toolkit*
- *Mobile e-Health Platform*
- *Vital Signs Protocol Format*
- INFRAWARE
- CARDOM
- TeleCardio

A seção 2.7 fornece a conclusão do capítulo.

### 2.1 *Context Toolkit*

Produto de trabalhos desenvolvidos no *Georgia Institute of Technology*, sobretudo da tese de doutorado de Dey [1], o *Context Toolkit* é um *framework* conceitual de suporte

à construção, ao desenvolvimento, à organização, e ao funcionamento de aplicações sensíveis ao contexto.

Concebido essencialmente a partir de critérios da Engenharia de Software a fim de alavancar o desenvolvimento de aplicações *context-aware*, o *Context Toolkit* introduz conceitos e abstrações interessantes associados à separação de preocupações e reuso. Além disso, também apresenta estratégias relevantes no desenvolvimento e no apoio à execução de aplicações sensíveis ao contexto.

Desta maneira, foram identificadas algumas questões que são comuns a qualquer aplicação *context-aware*, e por esta razão demandam um tratamento especial e que seja concebido sob um ponto de vista genérico a todas as aplicações. Assim, o *Context Toolkit* atende aos seguintes requisitos: (i) *separação de preocupações*<sup>1</sup>, que distingue as tarefas de aquisição e utilização do contexto; (ii) *interpretação de contexto*, que através da combinação de múltiplas informações contextuais oriundas dos sensores, provê níveis mais altos de abstração dessas informações às aplicações; (iii) *comunicação transparente e distribuída*, que consiste em ambos os projetistas das aplicações e dos sensores não precisarem se preocupar com o fato de que esses elementos via de regra não estarão fisicamente conectados, mas pelo contrário, estarão geograficamente distribuídos; (iv) *constante disponibilidade da informação contextual*, associada ao fato de que aplicações e sensores funcionam separadamente, de tal forma que muitas aplicações podem concorrer no acesso da mesma informação contextual, assim como uma aplicação pode acessar múltiplos sensores concomitantemente; (v) *armazenamento do histórico contextual*, que está diretamente ligado ao requisito anterior, e refere-se à possibilidade das aplicações acessarem uma informação contextual muito após o instante da aquisição da mesma; e (vi) *descoberta de recursos*, que envolve a questão de uma aplicação ser capaz de descobrir um novo provedor de contexto, conhecendo assim o tipo de contexto que ele oferece.

Para atender a esses requisitos, o *Context Toolkit* possui as seguintes categorias de componentes: *Context Widgets*, *Interpreters*, *Aggregators*, *Services* e *Discoverers*; de maneira que, a combinação deles busca satisfazer os requisitos anteriormente descritos no desenvolvimento das aplicações.

---

<sup>1</sup>Este item é de especial atenção no escopo deste trabalho e será aprofundado mais a frente no texto na seção 3.2.

Particularmente, de acordo com o foco deste trabalho, merecem destaque os *Context Widgets*, ou simplesmente *Widgets*, que são componentes de software que provém às aplicações acesso às informações contextuais em seus ambientes de operação. Eles abstraem a complexidade de uso e comunicação dos sensores e fornecem informações em formatos adequados para o entendimento dos dados pelas aplicações. Em resumo, os *Widgets* são blocos de componentes que podem ser organizados e reutilizados de acordo com a necessidade das aplicações, fornecendo um mecanismo de acesso uniforme e encapsulado das informações contextuais.

Com relação aos demais componentes, os *Interpreters* são responsáveis por elevar o nível de abstração de uma informação contextual; por exemplo, uma informação de localização pode ser expressa tanto na forma de latitude e longitude quanto em níveis de abstração mais elevados como uma cidade ou uma rua. Além disso, *Interpreters* podem ser compostos em múltiplas camadas, de acordo com o nível de abstração desejado pelas aplicações. Os *Aggregators* coletam informações de diversas fontes de informações contextuais logicamente relacionadas e as disponibilizam em um repositório único. Os *Services*, por sua vez, são componentes capazes de executar ações de acordo com o interesse e a necessidade das aplicações, de forma que, quando uma combinação de condições for atingida, uma ação pré-definida pode ser tomada a pedido de uma aplicação. E por fim, os *Discoverers* são responsáveis pela manutenção de um registro das informações de todos os componentes instanciados e suas respectivas competências; quando um *Widget*, um *Interpreter*, um *Aggregator* ou um *Service* é instanciado no *framework*, o componente *Discovery* é notificado dessa presença, registrando a existência do novo componente instanciado. A Figura 1 mostra uma possível configuração dos componentes apresentados.

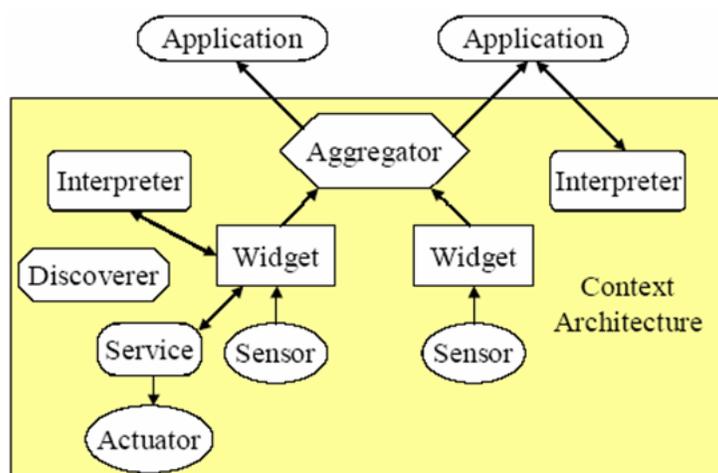


Figura 1: Exemplo de configuração dos componentes do *Context Toolkit* [1].

Cada componente, ao ser instanciado, registra-se no *Discovery*. Isto permite que os componentes instanciados sejam localizados por outros componentes ou mesmo por aplicações interessadas. Os sensores fornecem dados aos *Context Widgets*, que podem armazenar essas informações ou enviá-las aos *Interpreters*. Estes, por sua vez, podem manipular as informações primitivas provenientes dos *Widgets* para gerar novas informações com maior nível de abstração e disponibilizá-las aos demais componentes do toolkit e às aplicações. Um *Aggregator* coleta e armazena informações contextuais obtidas a partir dos *Widgets*. Finalmente, as aplicações podem requisitar aos *Aggregators*, aos *Widgets* ou aos *Interpreters* as informações contextuais de seu interesse.

Alguns dos componentes conceituais do *Context Toolkit* foram desenvolvidos e implementados em [1]. Contudo, as estruturas implementadas foram destinadas a uma classe específica de aplicações, e não fornecem um nível de generalidade suficiente para um conjunto maior de potenciais situações que podem ocorrer em ambientes ubíquos reais.

## 2.2 Mobile e-Health Plataform

Desenvolvida através de uma parceria entre a *University of Twente* e a *Yucat Mobile Business Solutions* [11], a *Mobile e-Health Plataform* [12] foi projetada a partir da *MobiHealth Plataform* finalizada em 2004 no projeto europeu MOBIHEALTH [13], como

infra-estrutura de suporte a diversos cenários nos quais o monitoramento remoto pode melhorar a qualidade no tratamento de saúde dos pacientes utilizando-se de tecnologias atualmente disponíveis. Segundo [12], a principal característica da *Mobile e-Health*, que a diferencia da *MobiHealth*, é a flexibilidade.

O projeto MOBIHEALTH visava a aplicabilidade, aceitabilidade, e usabilidade desse tipo de plataforma. E a avaliação feita em [14], mostrou resultados que comprovaram a relevância da plataforma *MobiHealth* no atendimento a tais requisitos. Os projetos *Awareness* [15], e *HealthService24* [16], apesar de possuírem diferentes objetivos, utilizam resultados do MOBIHEALTH, incluindo código fonte. No entanto, de acordo com a análise feita por Backx, ainda havia na *MobiHealth* uma carência por flexibilidade, que o levou a se dedicar à realização de uma re-engenharia da *MobiHealth*, gerando assim, a *Mobile e-Health*.

Conforme [12], tais plataformas genéricas devem possuir diferentes aspectos de flexibilidade, com destaque para (i) a flexibilidade de implementação, associada a questões como possíveis variações no conjunto de sensores conectados ao paciente, ou o tipo de interface gráfica exibida na aplicação do profissional de saúde, ou as funcionalidades providas pela plataforma em geral; e (ii) a flexibilidade em run-time, relacionada principalmente à mobilidade do usuário, que acarreta em variações na largura de banda da conexão, muitas vezes exigindo mudanças em tempo real restringindo o conjunto de sinais vitais a serem transmitidos.

Neste sentido, será apresenta aqui uma visão geral da *Mobile e-Health Plataform* (Figura 2a), cuja arquitetura (Figura 2b), que é baseada na *JINI Surrogate Architecture* [17], integra os seguintes componentes:

- ***Mobile Base Unit (MBU)***: oferece o serviço de fornecer os dados coletados dos sensores e outras funcionalidades ao *surrogate*, com o qual ela se conecta através de diferentes tecnologias de comunicação sem fio;
- ***Lookup service (LUS)***: usado para descobrir e registrar *e-health services*;
- ***e-Health Service***: serviço oferecido pelo *surrogate*, que é registrado no LUS pela exportação do *e-health proxy*; este *proxy* se comunica com o serviço através de uma interface pré-definida (em vermelho na Figura 2b);

- ***e-Health Proxy***: o *e-health* service exporta o *proxy* para permitir que os clientes se comuniquem com o serviço, e o *proxy* oferece a interface do serviço (em azul na Figura 2b) ao cliente;
- ***Surrogate host***: provê um ambiente de execução para os MBUs *surrogates*, permitindo que eles sejam incorporados à rede JINI;
- ***Healthcare application***: fornece a funcionalidade ao usuário final, enquanto utiliza o *e-health client* para se comunicar indiretamente com a MBU;
- ***E-Health Client***: é parte da *Healthcare application* que pode se comunicar com o *e-Health Service*. Ele recupera o *proxy* através da LUS, e utiliza sua interface para interagir com o serviço.

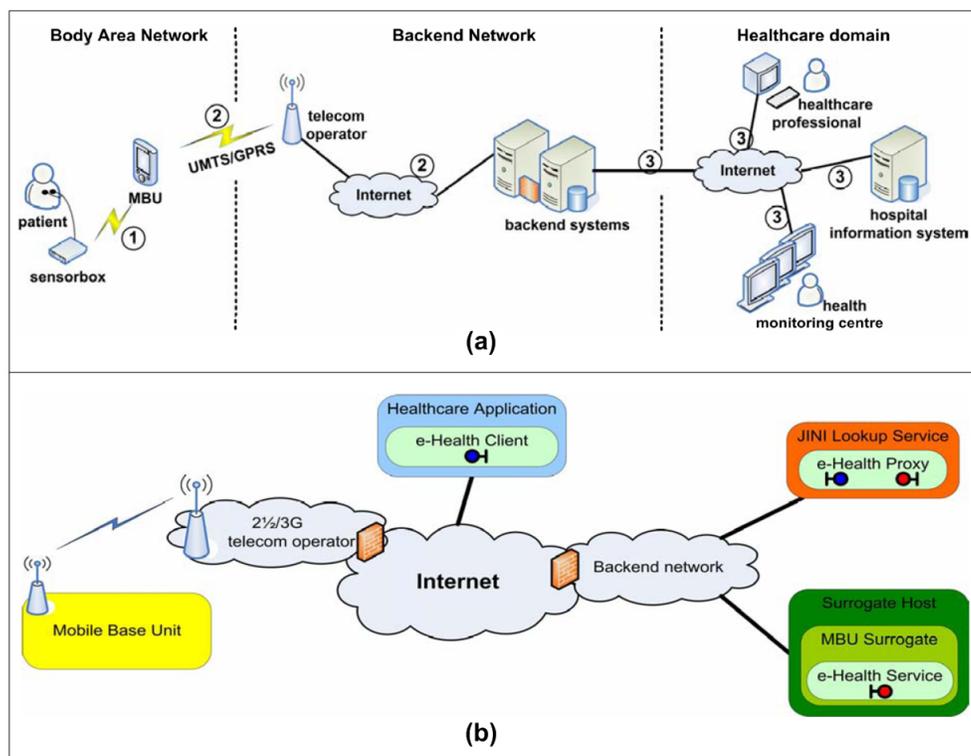


Figura 2: (a) Visão geral da *Mobile e-Health*, e (b) Distribuição de componentes na sua arquitetura [12].

De especial relevância para o foco de projeto do *ECG-Wrapper*, a MBU consiste em (i) um PDA que se comunica com os dispositivos sensores e é capaz de, utilizando uma

conexão 2.5/3G transmitir os dados coletados para a Internet; e (ii) um software<sup>2</sup> que implementa na MBU o procedimento de aquisição e entrega dos dados, cuja arquitetura é mostrada na Figura 3.

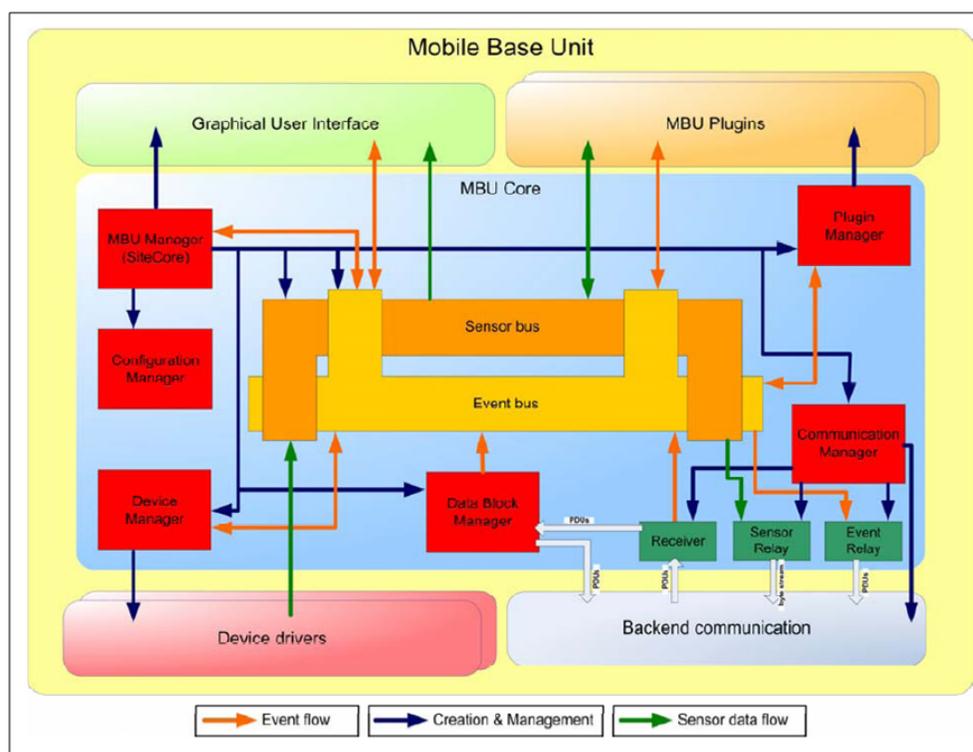


Figura 3: Arquitetura do software componente da MBU [12].

A arquitetura em questão é composta dos seguintes módulos [12]: (i) o *MBU Core*, que consiste nos componentes responsáveis pela funcionalidade básica da MBU (as setas na Figura 3 mostram os fluxos de eventos e de dados); (ii) os *Device Drivers*, que são comprometidos com o agrupamento dos dados coletados das fontes externas; (iii) *Plug-in components*, são elementos opcionais ao funcionamento básico do sistema, mas que oferecem funcionalidade adicional, normalmente necessária nos cenários mais complexos; (iv) *Graphical User Interface*, uma parte do sistema importante, já que é a mais visível ao usuário final; e o (v) *MBU/Surrogate Communication*, que por tratar-se do gargalo de comunicação da plataforma, exige uma transmissão dos dados de forma otimizada.

<sup>2</sup>Na implementação da *MobiHealth* esse software é denominado *banware*, em referência a *Body Area Network* (BAN).

Apesar de o software integrado na MBU e o *ECG-Wrapper* pertencerem a conjunturas diferentes, de fato o estudo de tal arquitetura e da interação de seus componentes oferece contribuições de análise e de projeto a serem aplicadas na conjuntura do *ECG-Wrapper*.

Com relação à *MobiHealth*, o projeto do software da MBU desenvolvido na *Mobile e-Health* representa uma completa re-estruturação. A arquitetura do *Device Driver* adiciona vários estados; sobretudo o mecanismo de *plug-in* e o *framework* de comunicação, fornecem melhorias significativas aumentando a flexibilidade de implementação, que também é privilegiada pela *Finite State Machine* (Máquina de Estados Finitos) do procedimento da MBU, e pela abordagem adotada no projeto da interface com o usuário. O *Communication Manager* e os componentes de entrega dos dados adquiridos verificam, a partir de então, se a largura de banda está sendo usada de maneira eficiente, oferecendo a flexibilidade *run-time* requerida [12].

Não obstante as importantes contribuições oferecidas, conforme ressaltado em [12] o projeto de software da MBU deixa alguns gaps para trabalhos futuros devido a algumas das complexas questões (por exemplo, gerência de conexão) associadas a um sistema distribuído.

## 2.3 Vital Signs Protocol Format

Também desenvolvido na *University of Twente*, o *Vital Signs Protocol Format* [18] consiste em um protocolo que aplica a distinção entre especificações de formatos de dados em diferentes níveis de abstração, especificamente entre a sintaxe abstrata e a sintaxe de transferência de dados, para aplicações distribuídas. A metodologia de projeto utilizada em [18] se aproveita da semântica permitida pela meta-linguagem XML na camada de Aplicação do Modelo OSI, que é independente dos esquemas de codificação aplicados nas camadas inferiores; ao passo que, em virtude da codificação textual do formato XML consumir um tamanho demasiadamente grande, para obter um formato compacto de transferência, são utilizadas as regras de codificação do ASN.1 que geram uma codificação binária eficiente.

Motivado pela dificuldade existente na transferência de informações médicas entre plataformas heterogêneas, o trabalho em questão foi destinado à elaboração de

tal protocolo, exclusivamente para transferência de sinais vitais, buscando suprir essa demanda no âmbito de sistemas *e-health* (*electronic-healthcare*). Para atingir esse objetivo, foi realizado em [18] um estudo sobre a fisiologia dos sinais vitais, bem como uma investigação de padrões de formato de tais sinais.

Os métodos de projeto utilizados para atingir as demandas citadas, são classificados em *(i) design-time*, *(ii) compile-time*, e *(iii) run-time*. O produto de *(i)* é a especificação de uma sintaxe abstrata de dados; ao passo que, o produto em *(ii)* é uma codificação *run-time* usando BER (*Basic Encoding Rule*) ou PER (*Packed Encoding Rule*); e por fim, o produto de *(iii)* é o protocolo em código binário BER/PER [18]. Na Figura 4 tais métodos são mostrados durante todo o processo de codificação da informação.

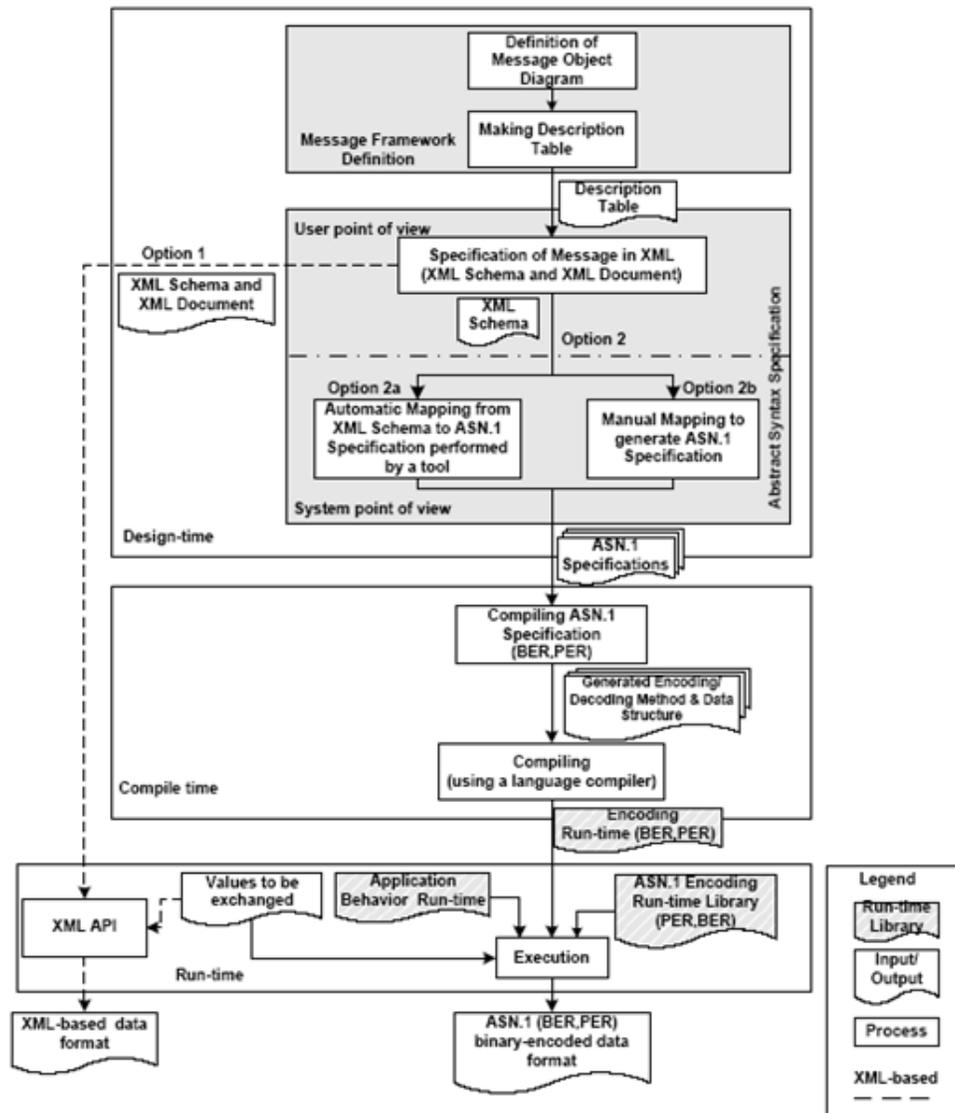


Figura 4: Processo de codificação da informação [18].

Para o *design-time*, é proposto um *message framework* que utiliza uma estrutura hierárquica de mensagem para representação da informação a ser transferida. Esse *framework* foi definido pelo *Health Level 7 (HL7)*, e exceto pela não inclusão de questões associadas a tecnologia como tipos de dados, é semelhante a um DTD e a um *XML Schema*. Em seguida, é especificada a partir do *message framework* a sintaxe abstrata de dados através de um *XML Schema*. Caso seja da preferência do desenvolvedor de aplicações, é possível transmitir a informação já neste formato XML (opção 1 na Figura 4). Por outro lado, se for desejável um formato mais compacto, deve-se escolher

o ASN.1 (opção 2 da Figura 4). Neste caso, existem duas opções: a primeira (opção 2a na Figura 4) corresponde ao mapeamento automático através de uma ferramenta, do *XML Schema* para o ASN.1; e a segunda (opção 2b na Figura 4) é definir a especificação ASN.1 manualmente.

No *compile-time*, cada especificação ASN.1 é usada como entrada pelo compilador que, após verificá-las, gera as estruturas de dados e as funções de codificação e decodificação que utilizam as regras ASN.1, BER ou PER. E então, ambas estruturas e funções serão compiladas para produzir uma codificação run-time em BER/PER.

E finalmente, o método utilizado em *run-time* é destinado a produzir, a partir da sua entrada - valores ASN.1 a serem transmitidos -, o código binário ASN.1 utilizando BER ou PER que contém os dados. O processo de execução envolve o procedimento *run-time* da aplicação (feito pelo desenvolvedor da aplicação), a codificação/decodificação *run-time*, e a biblioteca de codificação ASN.1 (fornecida com o compilador ASN.1).

Em resumo, [18] apresenta uma estratégia interessante para representação e transmissão de sinais vitais buscando privilegiar as demandas dos diferentes pontos de vista associados a, humano e máquina.

## 2.4 INFRAWARE

Financiado pela FAPES (Fundação de Apoio a Pesquisa do Espírito Santo), e conforme já mencionado, o projeto INFRAWARE [9] vem sendo desenvolvido no LPRM/DI/UFES e consiste na construção de uma plataforma de apoio a aplicações *context-aware*, a *Infraware* [5].

A plataforma *Infraware* é um middleware baseado na tecnologia de distribuição *Web Services*, e foi definida visando atender a vários requisitos funcionais presentes em ambientes sensíveis ao contexto, e integrá-los em uma infra-estrutura única, formando assim uma arquitetura flexível e adequada para a concepção de aplicações reais relacionadas a domínios variados.

A *Infraware* foi concebida a partir da arquitetura proposta pelo projeto WASP [19], um projeto holandês desenvolvido conjuntamente pela *University of Twente*, *Telematica Instituut* e *Ericsson*, e que posteriormente teve continuidade no Laboratório de

Pesquisas em Redes e Multimídia (LPRM) do Departamento de Informática da UFES, através de uma parceria informal com o ASNA (*Architecture and Services for Network Applications*) Group, da *Faculty of Electrical Engineering, Mathematics and Computer Science da University of Twente*. Neste sentido, algumas questões pouco exploradas pelo WASP foram tratadas em maiores detalhes pela plataforma Infraware, com destaque para: interpretação de contexto, gerência de subscrição, gerência de serviços, resolução de conflitos entre aplicações e interface com sensores heterogêneos. A Figura 5 ilustra a arquitetura geral proposta para a plataforma Infraware.

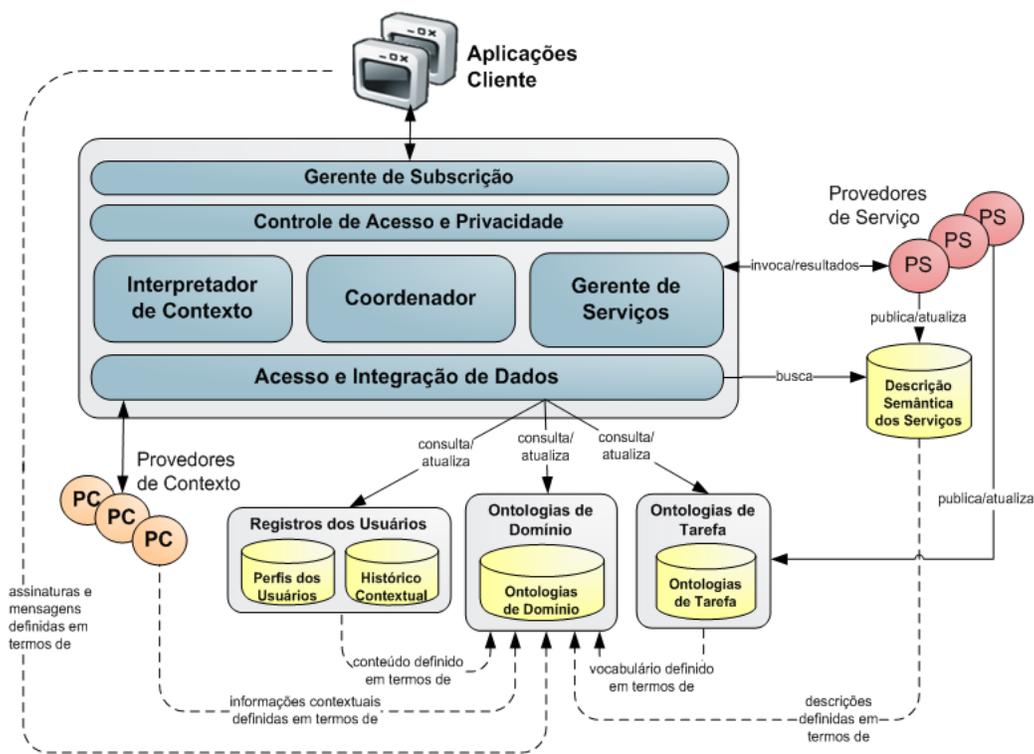


Figura 5: Arquitetura Geral da Plataforma Infraware.

O projeto da plataforma Infraware introduz componentes específicos para atender os seguintes requisitos funcionais: (i) tratamento dos pedidos de subscrição enviados pelas aplicações clientes; (ii) interpretação de contexto; (iii) aquisição e integração de dados contextuais heterogêneos; (iv) gerência integrada de serviços com descrição semântica; (v) resolução de conflitos e coordenação entre aplicações; e (vi) tratamento de questões relacionadas à privacidade e segurança. Além disso, para possibilitar a personalização dos serviços para os usuários da plataforma, são armazenados no repositório de Registros

dos Usuários os perfis contendo informações referentes às preferências dos usuários, juntamente com seus históricos de informações contextuais.

Marcada pelo uso de conceitos da Web Semântica, a Infraware utiliza *Web Services* como tecnologia de distribuição, permitindo que aplicações acessem os serviços oferecidos através de protocolos da Internet e facilitando a inclusão de novos serviços à plataforma por terceiros. Além disso, a plataforma é composta de ontologias que especificam modelos formais extensíveis descrevendo não somente o domínio das aplicações, mas também os serviços. Essa abordagem provê meios de configurar as interações aplicação-plataforma em tempo de execução (*run-time*), e de customizar a plataforma com a adição de novos serviços e entidades ao se estender as ontologias. Dessa maneira, a Infraware caracteriza-se por possuir uma flexibilidade que a torna adequada ao desenvolvimento de uma larga gama de aplicações em cenários reais.

A Infraware em sua versão mais recente no momento da escrita deste trabalho é apresentada de forma mais ampla em [20]. Para o enfoque deste trabalho, merece destaque o requisito de aquisição e integração de dados heterogêneos, mais especificamente a elaboração de uma interface uniforme com os diversos tipos de sensores, que será abordado em detalhes mais a frente neste texto, na seção 3.2.

## 2.5 CARDOM

O projeto CARDOM [10], também financiado pela FAPES, e intitulado Telecardiologia a Serviço do Paciente em Domicílio, foi iniciado em Setembro de 2005 e tem por objetivo a concepção de um sistema de tele-monitoramento ambulatorial da atividade elétrica do coração de pacientes mantidos em domicílio. A originalidade do trabalho consiste na implementação de métodos seguros de acionamento de alarmes e identificação precoce de situações de emergência baseados em um sistema de análise automática do ECG, além do estudo de novas tecnologias de comunicação sem fio e miniaturização aplicadas à sua integração. O CARDOM apóia-se nas competências complementares de pesquisadores do Programa de Pós-Graduação em Engenharia Elétrica da UFES.

Diante do exposto, é possível destacar como principais resultados esperados deste projeto a definição de:

1. Métodos de utilização do sistema de análise automática de ECG em cenários reais;
2. Um mecanismo de comunicação sem fio que melhor se adapte ao tele-monitoramento da atividade cardíaca de pacientes.

Em relação ao primeiro resultado esperado, vale ressaltar que, o sistema de análise automática de ECG foi desenvolvido na tese de doutorado de Andreão [21] e vem sendo adaptado para sua utilização em situações reais [22]. Em [21], o sistema era composto basicamente pela (i) Base de Dados contendo os registros de ECG necessários para as simulações dos algoritmos; por (ii) Algoritmos de Processamento de ECG que utilizam a base de dados para gerar resultados de segmentação e classificação dos sinais de ECG; e finalmente pela (iii) Interface GUI que permite o usuário executar os algoritmos e exibir os resultados.

Neste sentido, o sistema utilizava tal base de dados para fazer as simulações; no entanto, o objetivo do CARDOM é que seja feita a aquisição do eletrocardiograma por um dispositivo portátil, que efetua um monitoramento constante da atividade elétrica do coração do paciente por meio de três eletrodos amostrados a 250 Hz. Além disso, o dispositivo deve armazenar os dados de ECG e enviá-los a uma estação receptora periodicamente. Nessa estação receptora funciona este sistema, que interpreta os dados e gera alarmes caso detecte algum evento elétrico perigoso.

Quanto ao segundo resultado esperado, estudos já realizados como em [23], analisam as técnicas de aquisição e transmissão em tempo real de eletrocardiograma ambulatorial, enquanto outros investigam protocolos proprietários de comunicação sem fio de curta distância que sejam adequados para tal propósito.

Portanto, esses dois objetivos do projeto CARDOM, que estão relacionados, serão integrados na aplicação de tais pesquisas em cenários reais, sobretudo os abordados no projeto TeleCardio.

## 2.6 TeleCardio

O projeto TeleCardio [6] consiste em uma iniciativa dos Programas de Pós-Graduação em Engenharia Elétrica e em Informática do Centro Tecnológico da UFES. O TeleCardio

propõe o desenvolvimento integrado (hardware e software) de um sistema de Telecardiologia voltado para o acompanhamento da atividade elétrica do coração em pacientes crônicos.

Também financiado pela FAPES, e intitulado Telecardiologia a Serviço do Paciente em Ambientes Hospitalares e Residenciais, este projeto possui, com relação ao CARDOM, um enfoque mais prático, uma vez que foi concebido em cima de cenários reais de aplicação das pesquisas. Além disto, no TeleCardio passam a ser contemplados não somente ambiente domiciliares de monitoramento de pacientes, mas também cenários hospitalares onde serviços *context-aware* proporcionam além do monitoramento de pacientes, uma colaboração entre profissionais de saúde. Com relação ao projeto INFRAWARE, o TeleCardio proporciona a validação da plataforma de apoio a sistemas sensíveis ao contexto Infraware, através de uma aplicação móvel piloto *context-aware* de monitoramento de pacientes em ambiente hospitalar com suporte para trabalho cooperativo (*whiteboard*, conferência, etc.) usando um sistema de PEP (Prontuário Eletrônico do Paciente) como base de apoio de informações sobre os pacientes. Ao fim do projeto INFRAWARE, esse protótipo deverá ser testado em alguma unidade hospitalar do Governo Estadual e/ou no Hospital Universitário da UFES [9].

Além disso, o TeleCardio possui como objetivo geral e elemento motivador, a criação do **Centro de Tecnologia em Saúde** (CTS), visando a formação intelectual em uma área que vem exigindo cada vez mais competências para aplicar as inovações tecnológicas na área de saúde. Neste sentido, o TeleCardio busca explorar competências na área médica e tecnológica tendo em vista a melhoria da qualidade dos serviços médicos voltados a pacientes crônicos e hospitalizados em domicílio ou em alguma Unidade de Saúde, pública ou privada [6].

O enfoque do projeto está ligado mais especificamente ao tele-monitoramento da atividade cardíaca de pacientes motivado, por um lado, pelo alto índice de mortalidade relacionada a doenças do coração o que justifica a relevância social do projeto e, por outro, pela experiência acumulada de pesquisadores do CTS na análise automática do eletrocardiograma (ECG) ambulatorial. O ponto culminante do projeto será o desenvolvimento (projeto e implementação) de um protótipo para teste e avaliação da solução tecnológica alcançada. Observa-se que as tecnologias envolvidas podem ser diretamente usadas ou facilmente adaptadas para uso dentro de Unidades Hospitalares e

não somente no domicílio do paciente [6].

Dessa maneira, o TeleCardio se apóia em duas linhas de pesquisa distintas atualmente em andamento nos Departamentos de Engenharia Elétrica (DEL) e de Informática (DI) da UFES, e que se integram para superar limitações de sistemas similares atuais. São elas [6, 24]: *(i)* a utilização de métodos originais de processamento automático do ECG ambulatorial em vista da geração de alarmes em situações de risco para a saúde do paciente, em particular episódios de isquemia silenciosa [25]; e *(ii)* o uso de plataformas de apoio (*middleware*) a aplicações móveis e sensíveis ao contexto dotadas de funcionalidades adequadas para o desenvolvimento e execução de aplicações de tele-monitoramento que explorem o contexto dinâmico dos seus usuários (os pacientes e os profissionais de saúde) [26].

Levando tudo isso em consideração, conforme já mencionado anteriormente este trabalho de conclusão de curso faz parte do projeto TeleCardio [6], de maneira que, o sistema *ECG-Wrapper* se aproveita de resultados do projeto CARDOM, e visa realizar a interface entre os trabalhos dos grupos de pesquisa do LPRM/DI/UFES e do LABTEL/DEE/UFES, que integram esse projeto.

## 2.7 Conclusão do Capítulo

A apresentação dos projetos INFRAWARE, CARDOM, e TeleCardio, destinou-se a oferecer um posicionamento deste trabalho perante tais projetos.

Enquanto isso, o estudo dos trabalhos apresentados neste capítulo contribuiu no desenvolvimento desta monografia, principalmente no que se refere ao encapsulamento de contexto em plataformas de apoio a sistemas *context-aware*. Particularmente, o trabalho descrito na seção 2.3 ofereceu um ponto de partida para a investigação da representação e transmissão de sinais vitais incluindo eletrocardiogramas.

De fato, existem na literatura muitos trabalhos envolvendo *middlewares* de apoio a sistemas sensíveis ao contexto que trazem consigo resultados proveitosos quanto a funcionalidades comuns nesse gênero de sistema, tais como interpretação de contexto, descoberta de serviços, controle de privacidade, etc. Muito embora, com relação às tarefas de aquisição e encapsulamento de contexto, é notória a escassez de resultados

---

que privem pela generalidade. Provavelmente, isto se deve à já mencionada ampla diversidade de tecnologias e de tipos de sensores existentes, que acarretam no desenvolvimento de soluções por demasiado específicas a cada caso em questão, desprezando características comuns que podem ser identificadas. De maneira particular, o trabalho descrito na seção 2.1 constitui uma exceção à essa regra.

Especificamente com relação ao procedimento de aquisição dos dados contextuais junto aos sensores, quase que inexistem informações descrevendo essa tarefa. Isto constituiu um dos maiores desafios desta pesquisa.

## 3 *Domínio de Aquisição e Encapsulamento de Contexto*

Este capítulo estabelece boa parte das bases teóricas do trabalho, apresentando os conceitos fundamentais relacionados ao tema contexto e ao encapsulamento desse tipo de informação.

Inicialmente, na seção 3.1 são apresentadas as definições de contexto e termos afins adotadas neste trabalho; em seguida é apresentada de forma sucinta na seção 3.2 uma classificação de domínios e suas relações com aplicações sob o ponto de vista da computação. Posto isso, na seção 3.3 é introduzido o problema de encapsulamento de contexto em plataformas de apoio a sistemas *context-aware*. Na seção 3.4, é mostrado o posicionamento de tal problema na plataforma Infraware. Diante de tudo isso, na seção 3.5 é definida uma adaptação de conceitos citados nas seções anteriores para atender a requisitos identificados através desse estudo. E por fim, a seção 3.6 realiza a conclusão do capítulo.

### 3.1 Definindo Contexto

Contexto, situação, dados contextuais, e aplicação sensível ao contexto, podem ser definidos de várias maneiras, uma delas, apresentada por Schmidt e Laerhoven, define esses termos como [27]:

- **Situação:** uma situação no mundo real;
- **Dados contextuais:** os dados capturados por sensores para representar uma situação;
- **Contexto:** uma descrição abstrata de uma situação no mundo real;

- **Aplicação sensível ao contexto:** uma aplicação que muda o seu procedimento de acordo com o contexto. De acordo com [28], incorporando à definição de aplicação sensível ao contexto a condição apresentada em [1], de que para ser definida como tal, a aplicação somente deverá mudar seu procedimento se a mudança no contexto tiver relevância para a atividade exercida pelo usuário dessa aplicação, esse conjunto de definições é adotado neste trabalho, de maneira que a palavra contexto e os outros termos apresentados referir-se-ão, ao longo do texto, unicamente às definições aqui apresentadas.

## 3.2 Relação entre Domínios e Aplicações

Um domínio caracteriza-se por uma terminologia comum para descrever os conceitos e os problemas existentes no seu escopo [29]. Mais especificamente, em Ciência da Computação, um domínio pode ser definido como um conjunto de problemas ou funções que as aplicações desse domínio podem resolver [29]. Portanto, o desenvolvimento de um sistema computacional visa atender à demanda de algum tipo de domínio no qual existe um problema ou uma função que pode ser modelado(a) e resolvido(a) através de um computador. Em Engenharia de Domínio, foi criada uma classificação de domínios, que os divide em: *(i)* domínios verticais, e *(ii)* domínios horizontais [30].

Nos domínios verticais, as aplicações são classificadas em relação a alguma área ou assunto do mundo real. Alguns exemplos são sistemas relativos às áreas do conhecimento de física, biologia, economia, etc; ou às áreas de negócio como a administrativa, de finanças, marketing, e outras; ou que atendem a alguma atividade mais específica, por exemplo, reserva de passagens aéreas. Já nos domínios horizontais, os sistemas computacionais são classificados de acordo com a função que exercem dentro do universo da computação. Exemplos desses domínios são as áreas de banco de dados, de redes, bibliotecas GUIs, compiladores, etc [31]. A Figura 6 permite uma visualização dessa classificação.

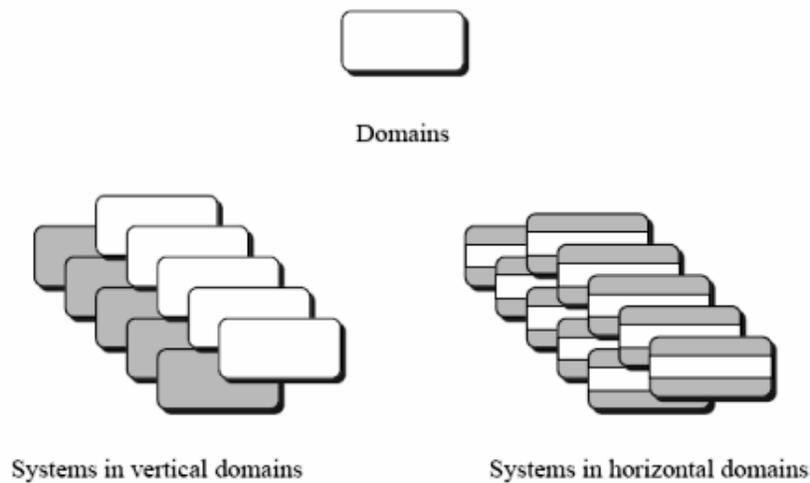


Figura 6: Domínios verticais e horizontais [32].

Um sistema computacional pode simultaneamente pertencer a diversos domínios. A título de exemplo, uma aplicação bancária abrange, no mínimo, os domínios de: práticas bancárias em geral, interfaces com usuários, gerência de *workflow*, gerência de banco de dados, e redes [33]. No entanto, aplicações não necessariamente cobrem domínios inteiros, de maneira que não é incomum somente partes de domínios se apresentarem em uma aplicação [31]. A Figura 7 exhibe essas possibilidades, onde as aplicações estão representadas como retângulos e os domínios como áreas hachuradas.

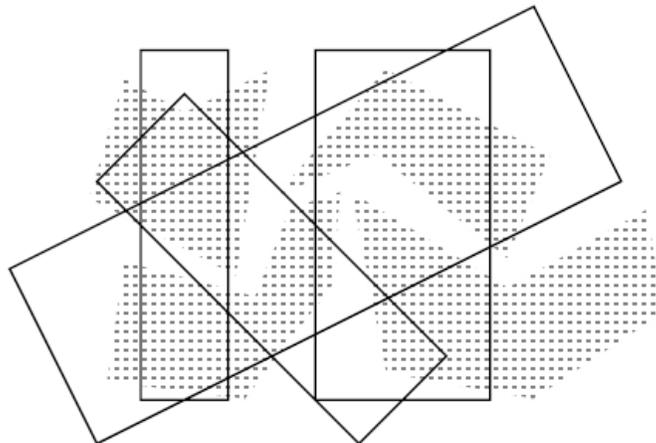


Figura 7: Relações entre domínios e aplicações [34].

No caso do sistema objeto deste trabalho, de dentro do universo da computação é possível distinguir: (i) o domínio horizontal de sistemas sensíveis ao contexto, sobretudo no que se refere ao encapsulamento de contexto, discutido a seguir na seção 3.2; e (ii) o domínio vertical de Telecardiologia, especificamente abordando a representação e transmissão de eletrocardiogramas, discutido no Capítulo 4.

### 3.3 Encapsulamento de Contexto

Em 1998, ao fazer uma análise do domínio de sistemas *context-aware*, Pascoe percebeu que, a ampla diversidade de tipos de contexto a serem explorados e o excesso de tecnologias de sensoriamento, estavam na verdade dificultando uma maior difusão de aplicações sensíveis ao contexto, na medida em que as aplicações, em geral, eram desenvolvidas de forma *ad-hoc* [8].

De fato, conforme ressaltado por Pascoe, criar software que interaja com uma grande variedade de hardware para capturar contexto, convertê-lo para um formato desejado, efetuar análises e comparações em torno dele, e apresentá-lo ao usuário com uma semântica refinada, é uma tarefa difícil e demorada. Por esta razão, é necessário que esse processo seja feito de tal forma a capturar os problemas e soluções comuns, para posteriormente adaptá-los e incorporá-los a cada nova aplicação a ser desenvolvida. Por exemplo, para incorporar às suas funcionalidades o contexto de localização, toda aplicação enfrenta desafios semelhantes, tais como estabelecer uma comunicação com o dispositivo de aquisição, extrair dele a informação de localização, convertê-la de latitude e longitude para um formato mais apropriado, e projetá-la em um mapa [8]. E da mesma forma ocorre com relação a qualquer novo tipo de informação contextual e seus respectivos sensores que uma aplicação pretenda incorporar.

Neste sentido, Dey destaca que uma das principais razões pelas quais aplicações sensíveis ao contexto ainda não se tornaram mais comuns, é a de que não existe uma forma padronizada de adquirir e tratar contexto. Comumente, isso é feito de forma improvisada de acordo com cada dispositivo de hardware a ser utilizado, na medida em que os desenvolvedores das aplicações escolhem a forma mais fácil de implementar, sob a perda da generalidade e do reuso. Basicamente, existem duas formas de lidar com contexto: (i) na abordagem sensor-driven, os drivers dos sensores são dire-

tamente conectados nas aplicações; por outro lado, (ii) na abordagem direcionada a uma separação de preocupações, programas encapsuladores são usados para esconder das aplicações os detalhes dos sensores [35].

Na primeira situação, os desenvolvedores das aplicações são forçados a escrever código para lidar com os detalhes dos sensores, tendo que se adaptar seja qual for o protocolo ditado pelos mesmos. Essa técnica, além de tornar o desenvolvimento de uma aplicação sensível ao contexto muito mais complexo do que precisaria ser, vai de encontro às recomendações da Engenharia de Software, pois não permite a separação entre a semântica mais apurada da aplicação e os detalhes de baixo nível de abstração da aquisição de contexto dos sensores. Pelo contrário, a técnica em questão implica em perda de generalidade, tornando difícil o reuso dos sensores em outras aplicações, bem como o uso simultâneo em várias aplicações [35].

Enquanto isso, não obstante já existirem sistemas que utilizam servidores para lidar com a aquisição de contexto dos sensores utilizando a segunda técnica, inclusive gerenciando a ocorrência de eventos através de mecanismos de consulta e notificação, esses trabalhos se caracterizaram pelo projeto de servidores específicos que não compartilham de uma interface comum, o que força a aplicação a lidar com cada servidor de uma maneira distinta [36]. Portanto, diante do exposto, faz-se necessário na aquisição de dados em plataformas de apoio a aplicações sensíveis ao contexto, uma interface uniforme com os diversos tipos de sensores. Além disso, é necessário distinguir em tais plataformas a utilização de contexto da aquisição de contexto, conforme ilustrado na Figura 8.

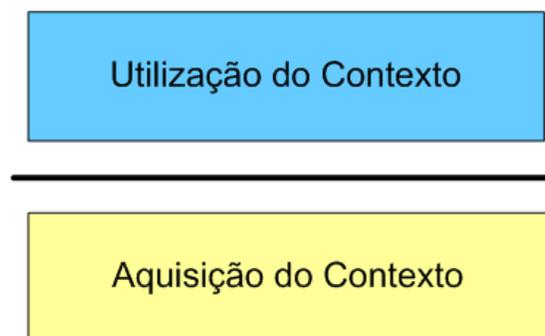


Figura 8: Separação das diferentes tarefas, de utilização e de aquisição de contexto.

Em [35], Dey analisa a tarefa de aquisição de contexto fazendo uma analogia com um problema semelhante pertencente ao domínio horizontal de *Graphical User Interfaces* (GUIs), em que é desejável que a aplicação não precise se preocupar com a forma como os dados são adquiridos, de maneira que, uma troca no dispositivo de aquisição não cause impacto na mesma. Assim, a aplicação pode abstrair a complexidade da aquisição de dados através de um componente, no caso o *GUI Widget*, que recebe e transmite-lhe os dados inseridos pelo usuário. A Figura 9 permite uma visualização desse conceito.

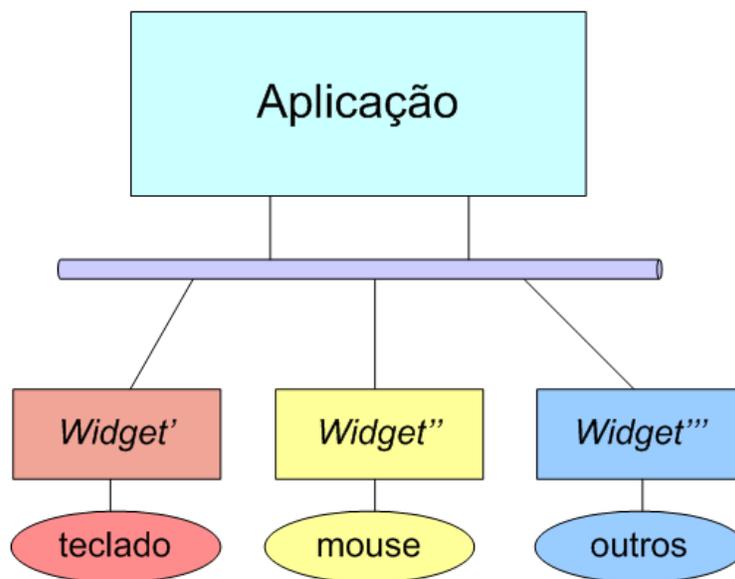


Figura 9: Aquisição de dados utilizando *GUI Widgets*.

Cabe ressaltar aqui a distinção de que em aplicações GUI, a aquisição de dados é toda realizada explicitamente através de interações com o usuário; já em aplicações *context-aware*, a aquisição de dados pode ser realizada de ambas as formas, explícita e implícita, justamente buscando exigir o esforço mínimo do usuário, proporcionando, assim, uma sensibilidade ao seu contexto sem incomodá-lo. Essa distinção é ilustrada na Figura 10.

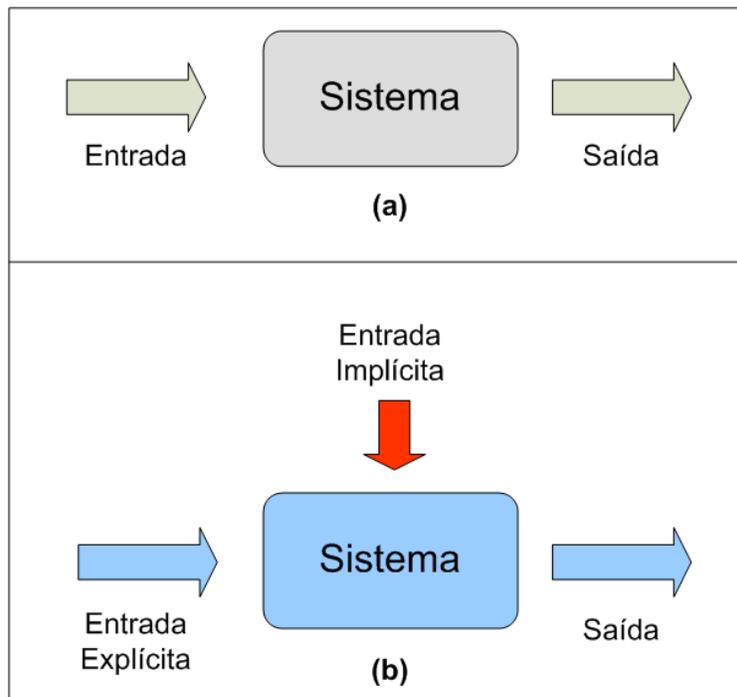


Figura 10: Diferença entre (a) sistema tradicional e (b) sistema *Context-aware*.

Uma outra questão pertinente à aquisição de contexto em tais plataformas, é entregar a informação já minimamente processada, após a realização de um tratamento nos dados, que depende do tipo de informação contextual do domínio em questão. Nos casos em que os dados obtidos dos sensores são contínuos (formas de onda), são necessárias filtragens no sinal a fim de eliminar ruídos. Um exemplo deste caso é a filtragem no sinal de eletrocardiogramas referente ao domínio vertical de eletrocardiografia [22]. Já no caso de dados discretos, como localização, temperatura, etc, muitas vezes é necessária uma calibração, através de funções estatísticas, para obter resultados como média e desvio padrão de várias amostras e evitar que um ou outro erro de coleta afete de forma significativa o significado da informação contextual [27]; e assim, conseqüentemente é feita uma síntese que reduz o volume de dados a ser armazenado e transmitido.

### 3.4 Aquisição de Contexto na Infraware

Levando em consideração os quesitos discutidos na seção anterior, a arquitetura da plataforma Infraware, conforme apresentado na seção 2.4, possui um componente de Acesso e Integração de Dados, responsável pela aquisição de contexto, composto de

dois sub-componentes: o (i) *middleware* de acesso e integração de dados heterogêneos CoDIMS, e o (ii) *Wrapper*, objeto deste trabalho. A Figura 11 fornece uma visualização deste componente, em que as interações entre o CoDIMS e cada instância *Wrapper* são padronizadas de acordo com um formato de mensagem, enquanto que as interações entre cada *Wrapper* com seu respectivo conjunto de sensores (*sensor box*) são específicas de acordo com o domínio vertical instanciado por esses elementos.

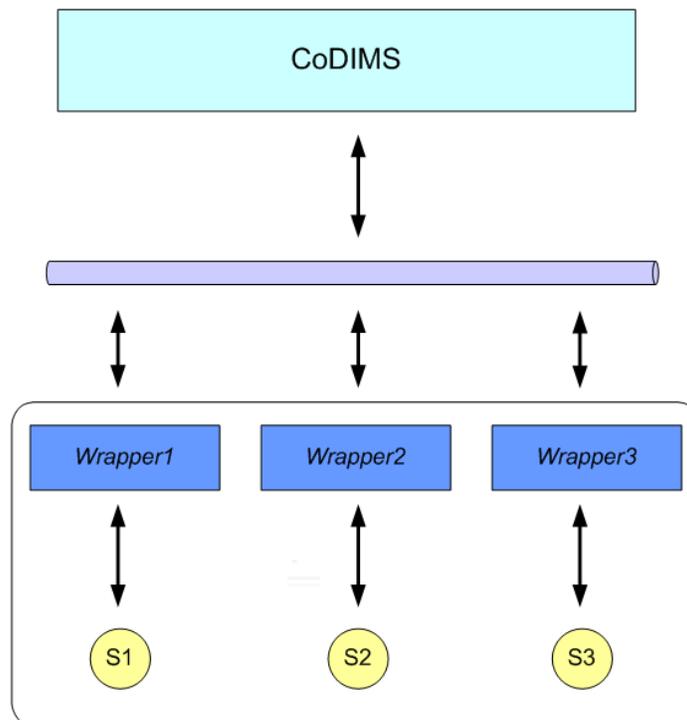


Figura 11: Componente de Acesso e Integração de Dados na Infraware.

Oriundo da área de Banco de Dados, mais especificamente de Integração de Dados Heterogêneos [37, 38], o conceito de *Wrapper* está associado a um encapsulador de fontes de dados heterogêneas. Uma definição encontrada na literatura dessa área conceitua um *Wrapper* como um software integrante de um sistema, que faz a ligação entre este e um sistema externo, para o qual é específico, eliminando o *gap* existente entre os mesmos. Essa ligação é realizada através da conversão de consultas, comandos, e dados, entre os formatos interno e externo [39]. Para melhor ilustrar tudo isso, é feita uma breve discussão sobre sistemas de integração de dados e metadados.

De acordo com [40], um sistema de integração de dados tem como foco prover uma

interface geral para um conjunto de fontes heterogêneas, distribuídas, e autônomas. Esta heterogeneidade está associada com os diferentes formatos de dados, mecanismos de comunicação, e tipos de hardware relativos às fontes, que são distribuídas em diferentes localizações, e possuem autonomia desde que as fontes de dados foram projetadas de maneira independente, e operam sob um controle local.

Nesses sistemas, os metadados exercem um papel fundamental, na medida em que descrevem os dados armazenados pelas fontes, fornecendo as informações necessárias para que sejam possíveis os mapeamentos entre os diferentes formatos de dados. Uma das maneiras de se representar metadados é através de esquemas que, em geral, são classificados de acordo com quatro níveis distintos [40], são eles: *(i)* os esquemas locais, pertencentes às fontes de dados e estruturados conforme a natureza de cada fonte; *(ii)* o esquema global, definido pelo sistema integrador e que fornece uma visão homogênea sobre todas as fontes de dados; *(iii)* os esquemas de exportação, que são decorrentes da conversão dos esquemas locais para o esquema global através de mapeamentos; e *(iv)* os esquemas externos, que fogem ao escopo deste trabalho, pois se tratam de visões interessantes para consultas dos usuários nas aplicações. Na Figura 12 são mostradas as relações entre esses tipos de esquema.

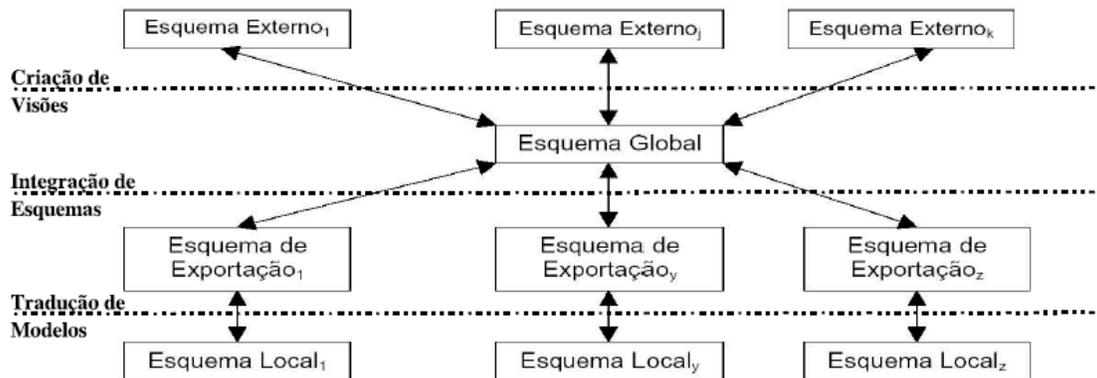


Figura 12: Níveis de esquemas e suas relações [40].

Posto isso, é possível identificar na própria Figura 12 a função dos *Wrappers* na integração de dados: a partir dos esquemas locais oriundos de cada fonte de dados, efetuar a conversão desses esquemas para os esquemas de exportação, compreensíveis pelo sistema integrador, no caso da Infraware, o CoDIMS.

O componente de Acesso e Integração de Dados utilizado na plataforma Infraware é fruto da incorporação e adaptação de um outro trabalho de pesquisa do LPRM/DI/UFES: o CoDIMS (*Configurable Data Integration Middleware System*) [37], um ambiente configurável para integração de dados heterogêneos que tem por objetivo prover transparência de localização, modelo de dados e forma de comunicação.

## 3.5 Wrappers Contextuais

Por sua vez, ao aplicar o conceito de *Wrapper* em sistemas sensíveis ao contexto, é necessário adaptá-lo de forma a contemplar as peculiares questões pertinentes a esse domínio. Neste caso, ao invés de somente traduzir dados em formatos heterogêneos, que por si só já é uma tarefa complexa, o *Wrapper* tem de lidar com uma grande diversidade e quantidade de sensores, padrões, e linguagens de comunicação. Além disso, inclui-se também o aspecto temporal da informação e, ainda em boa parte dos casos, os sensores mantêm-se em movimento, exigindo de seu consumidor de dados (na Infraware, o CoDIMS) uma constante verificação quanto à disponibilidade dos mesmos.

Neste sentido, inspirando-se no *Context Widget* proposto por Dey em [1], mas trazendo as contribuições do conceito de *Wrapper* existente na área de Integração de Dados Heterogêneos, é definido aqui o ***Context-Wrapper***, um software para compor a plataforma Infraware, sendo instanciado ao tipo de informação contextual tratado no domínio de aplicação da plataforma - no caso deste trabalho, monitoramento de ECG -, para oferecer o serviço de (i) adquirir os dados contextuais, ainda brutos, oriundos de um conjunto de sensores em um formato específico, (ii) processá-los já incorporando uma semântica mais apurada, (iii) convertê-los para um formato padrão entendível pela plataforma, e (iv) disponibilizá-los através de um *Web Service*. A Figura 13 ilustra basicamente a função do *Context-Wrapper* que, conforme pode ser visto, transforma os dados brutos obtidos dos sensores em informação compreensível do ponto de vista humano.

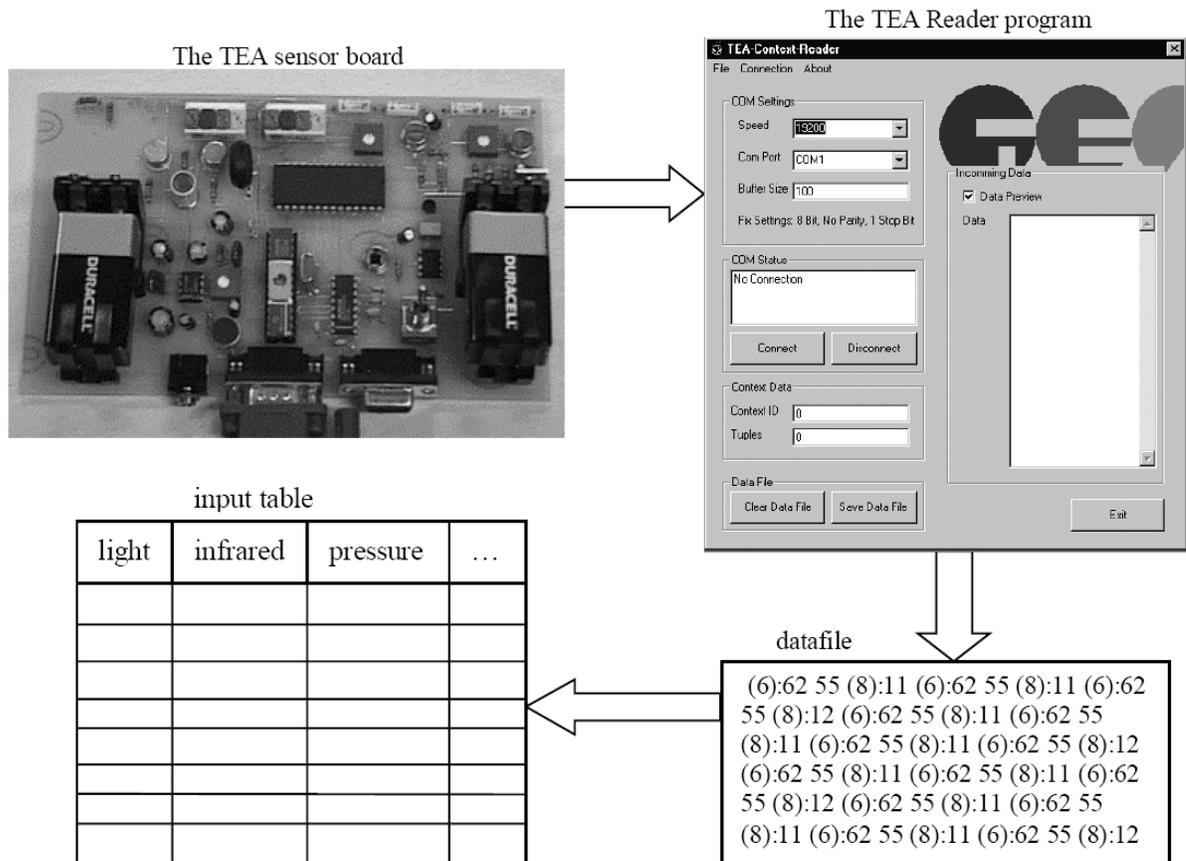


Figura 13: Transformação de dados obtidos de sensores em informação [41].

Portanto, assim como o *Context Widget* [35], o *Context-Wrapper* permite:

- Uma separação das diferentes preocupações de utilizar e adquirir contexto, permitindo que a plataforma Infraware não precise se preocupar com a complexidade dos sensores utilizados de modo que uma mudança nos sensores não cause impacto na mesma, e também que as aplicações sejam executadas de forma independente do componente provedor de contexto;
- Uma abstração da informação contextual adequada à expectativa das aplicações e, além disso, permitindo que as informações possam ser utilizadas por múltiplas aplicações simultaneamente e também que a plataforma só notifique uma determinada aplicação quando for do interesse da mesma;
- A existência de blocos de informação contextual reusáveis, com diferentes níveis de abstração. Isso pode ser ilustrado no seguinte exemplo: do bloco de contexto

obtido através de sensores de movimento instalados no chão juntamente com o bloco de contexto obtido de sensores de som em um ambiente, pode-se inferir, por exemplo, se está ou não havendo uma reunião em uma sala, dentre outras coisas.

Além disso, o software *Context-Wrapper* aqui definido incorpora as seguintes facilidades:

- Através da criação de uma nova instância *Context-Wrapper*, possibilitar a adaptação a um novo tipo de contexto, e conseqüentemente novos tipos de sensores e de aplicações, sem haver a necessidade de grandes mudanças na plataforma Infraware;
- Provê uma representação do tipo de contexto do domínio vertical em questão, *user-driven*, através de um meta-modelo que captura tal informação;
- Realiza um gerenciamento de eventos a serem detectados durante uma análise da informação contextual, que pode desencadear ações na plataforma Infraware e em aplicações sensíveis ao contexto;
- Mantém uma comunicação constante com os sensores, e baseada em demanda com a plataforma;
- Serve como um *pipe* (tubo) entre os provedores e os consumidores de dados, permitindo o desacoplamento dos mesmos e conseqüentemente um uso eficiente da informação contextual.
- Entrega a informação contextual à plataforma Infraware já com uma confiabilidade garantida, após realizar sobre ela filtragens ou outros tipos de correção necessários.

Essas características constituem a direção para a especificação dos requisitos gerais de um *Context-Wrapper*, apresentados na próxima seção.

## 3.6 Requisitos e Arquitetura Conceitual do Context-Wrapper

Para garantir as facilidades mencionadas, é necessário que o *Context-Wrapper*, através de seus sub-componentes, atenda aos seguintes requisitos dispostos na Tabela

1, que são comuns a todas as instâncias do domínio horizontal de encapsulamento de contexto:

Em geral, na literatura são encontradas implementações do requisito R1.1 de diversas maneiras. Em [41] a comunicação é feita diretamente conectando um laptop na placa de sensores, já em [12] é utilizado um *link Bluetooth*. Mais a frente, na seção 4.3 será apresentada a solução de tecnologia melhor avaliada para ser implementada neste trabalho.

Quanto ao requisito R1.2, é feito no *Context-Wrapper* um pré-armazenamento da informação contextual para permitir que os dados não sejam perdidos ao fim da execução do programa. A verdadeira persistência da informação contextual é realizada pelo middleware de Acesso e Integração de dados da plataforma Infraware, o CoDIMS. Com relação à *bufferização*, de fato, é possível a partir disto haver uma separação entre os provedores de contexto (sensores) e os consumidores de contexto (a Infraware, mais precisamente as aplicações) [12], preservando-se assim, largura de banda e capacidade de processamento, na medida em que o contexto só é repassado à plataforma quando necessário. Na própria fase de *bufferização*, deve ocorrer o tratamento da informação quanto à sua precisão, de forma que os dados, sejam contínuos ou discretos, possam ser entregues à plataforma já mais refinados [28].

Na plataforma Infraware, existe uma hierarquia de interpretação. Portanto, nem toda análise de contexto será realizada no componente Interpretador de Contexto [20] citado na seção 2.4. Este é responsável pelas interpretações mais refinadas e de mais alto nível de abstração, que em geral envolvem múltiplos blocos de contexto oriundos de conjuntos de sensores distintos. Dessa forma, conforme o requisito R1.3, fica a cargo do *Context-Wrapper* um processamento do contexto bruto adquirido junto aos sensores a fim de (i) detectar a ocorrência de eventos de interesse às aplicações; e (ii) entregá-lo à plataforma em um formato mais refinado semanticamente.

A criação do formato de representação referente ao requisito R1.4 é necessária para a consolidação de uma abordagem user-driven que permita um diálogo comum entre os especialistas do domínio vertical em questão e os desenvolvedores de aplicações. A partir do meta-modelo gerado, será possível a instanciação de modelos (mensagens) contendo a informação contextual correspondente ao domínio, adquirida junto aos sensores. Neste sentido, toda instância *Context-Wrapper* associada a um domínio vertical

R#	Descrição	Classificação
1.1	Adquirir as informações contextuais providas pelos sensores através de algum mecanismo de comunicação. Este pode ser implementado de diversas maneiras, como simplesmente conectando o dispositivo de computação onde o <i>Context-Wrapper</i> estiver funcionando diretamente nos sensores; mas para privilegiar mobilidade, é desejável que essa comunicação dê-se através de um <i>wireless link</i> , exigindo assim a utilização de um protocolo de comunicação sem fio a fim de possibilitar o máximo possível de mobilidade ao usuário.	Obrigatório
1.2	Para controlar de forma adequada e eficiente o fluxo de dados oriundos do dispositivo portátil, o sistema tem de ter um esquema de <i>bufferização</i> , e de pré-armazenamento em disco, que proporcione o controle desejado sobre a coleta e a entrega dos dados quando eles forem do interesse de alguma aplicação. Junto a esse esquema, é necessária também, a realização de filtragens ou médias, a fim de livrá-los de, respectivamente, ruídos (sinais contínuos) ou incertezas (dados discretos).	Obrigatório
1.3	Processamento dos dados coletados dos sensores a fim de já proporcionar à plataforma interpretação semântica (com nível de granularidade variável para cada caso) desses dados e o desencadeamento de ações relacionadas a eventos detectados através dessa análise.	Opcional
1.4	Criação de uma sintaxe abstrata e de uma sintaxe de transferência de dados, relativas às camadas de Aplicação e Apresentação do modelo OSI, para representar e transmitir a informação contextual adquirida dos sensores (vide seção 3.3). Assim, será gerado um meta-modelo específico para o domínio vertical tratado na instância do <i>Context-Wrapper</i> em questão.	Obrigatório
1.5	Conversão <i>online</i> dos modelos gerados a partir do meta-modelo definido no item anterior para modelos formatados segundo o meta-modelo global utilizado pelo módulo de Acesso e Integração de Dados da plataforma Infraware.	Obrigatório
1.6	O <i>Context-Wrapper</i> deve disponibilizar na Web as informações nele produzidas, através de, inicialmente três modelos de entrega: <i>request-response</i> , <i>time-driven</i> , e <i>event-driven</i> .	Obrigatório
1.7	Dependendo do tipo da informação contextual a ser tratada, ou dos cenários de uso do sistema que constitui uma instância de <i>Context-Wrapper</i> , é desejável haver uma GUI para permitir interações com o usuário que sejam relevantes para o sistema.	Opcional

Tabela 1: Requisitos do *Context-Wrapper*.

demanda um estudo da realidade desse domínio do mundo real, a fim de capturá-la e representá-la isomorficamente no tal meta-modelo. Além disso, será feita, como expresso no requisito R1.5, uma conversão *online* entre esse meta-modelo específico do domínio de contexto tratado em cada instância, para o meta-modelo padrão do módulo de Acesso e Integração de Dados da plataforma Infracore. Por seu turno, a sintaxe de transferência é necessária para, de acordo com as questões de eficiência na troca de dados através da Internet, adaptar a mensagem gerada pelo formato abstrato de forma a privilegiar uma transferência eficiente da informação contextual.

Uma vez que a plataforma Infracore é parcialmente distribuída, seus componentes interagem através de *Web Services*, devidamente catalogados, oferecidos na Internet. Por esta razão, o requisito R1.6 atende a uma questão fundamental para o fluxo de dados através dos componentes da Infracore.

Por último, o requisito R1.7 prevê a elaboração de uma interface com o usuário que possibilite a inserção no sistema, de dados explícitos fornecidos por ele, que sejam relevantes à aplicação *context-aware*. Este requisito é opcional devido a não ser comum a todos os domínios de contexto a necessidade de tais interações com usuários.

Em resumo, é apresentada aqui, conforme a Figura 14, uma proposta de arquitetura conceitual para o *Context-Wrapper*.

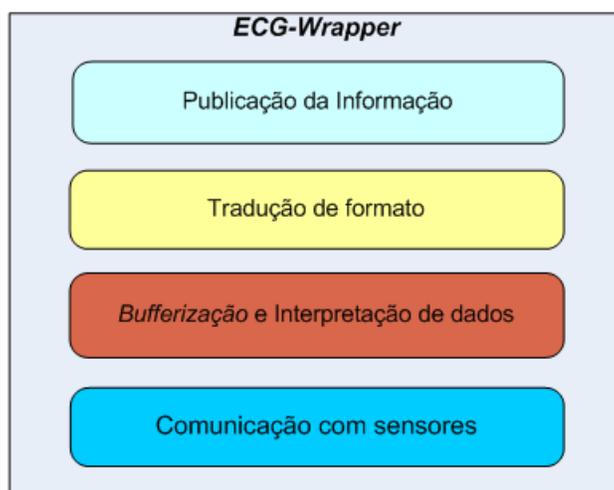


Figura 14: Arquitetura conceitual proposta para o *Context-Wrapper*.

A arquitetura *Context-Wrapper* é dividida em quatro camadas bem definidas, responsáveis por implementar soluções que atendam aos requisitos da Tabela 1. A camada de

*Comunicação com os sensores* atende ao requisito R1.1. Já a camada de *Bufferização e Interpretação de Dados*, é responsável pelos requisitos R1.2 e R1.3. Por sua vez, a camada de *Tradução de formato* deve garantir o cumprimento dos requisitos R1.4 e R1.5. Por fim, a camada de *Publicação da Informação* tem em vista os requisitos R1.6. O requisito R1.7 é opcional.

## 3.7 Conclusão do Capítulo

Neste capítulo foi brevemente apresentada uma classificação de domínios sob o ponto de vista da computação, e essencialmente, foi realizada uma análise da tarefa de aquisição e encapsulamento de contexto em plataformas de apoio a sistemas *context-aware*. Os resultados aqui alcançados estão relacionados à captura de funcionalidades comuns ao problema de aquisição e encapsulamento de contexto. A partir de então, será possível desenvolver um sistema que represente uma instância de um *Context-Wrapper* de maneira mais simples, através do reuso dos conceitos gerais aqui destacados.

No capítulo seguinte será abordado, dentro do domínio vertical de Telecardiologia, o problema da representação e transmissão de eletrocardiogramas. Buscando oferecer um melhor entendimento desse problema, serão discutidos padrões da área médica destinados a tal propósito.

## 4 *Domínio de Representação e Transmissão de ECG*

Este capítulo consiste em um estudo do domínio de representação e transmissão de eletrocardiogramas, que foi necessário para o projeto do *ECG-Wrapper*, particularmente para a concepção de um meta-modelo referente a esse domínio.

Inicialmente, na seção 4.1 são situadas ambas as tarefas de representação e transmissão de dados de acordo com o modelo OSI; em seguida, na seção 4.2 é apresentado um estudo da eletrocardiografia; a partir da seção 4.3 até a 4.7 são apresentados e discutidos padrões referentes ao domínio de representação e transmissão de ECG, e também o uso de XML nesse problema; posto isso, na seção 4.8 é fornecido um resumo desse domínio e apresentada uma especificação de requisitos para um formato de representação e transmissão de ECG; e por fim, a seção 4.9 concluiu o capítulo.

### 4.1 Camadas de Aplicação e Apresentação do Modelo OSI

Para fazer a aquisição, o processamento e a transmissão do sinal ECG de um paciente, é necessário inicialmente, conforme indica o requisito R1.4 da seção 3.2, a definição de (i) uma sintaxe abstrata de representação dos dados, bem como uma (ii) uma sintaxe de transferência desses dados. Tomando como base o modelo OSI, os dois itens descritos acima são funções bem definidas das camadas de mais alto nível de abstração do modelo, que são as de Aplicação, e de Apresentação, respectivamente [42]. É importante lembrar que na camada de Aplicação o foco é uma representação mais apropriada para o entendimento humano, e já na de Apresentação, o foco é preservar o enlace de comunicação buscando comprimir ao máximo a informação transmitida [18].

Essa discussão é ilustrada na Figura 15.

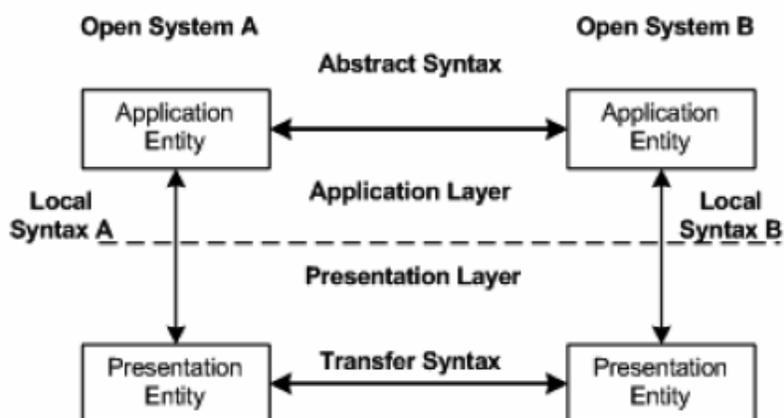


Figura 15: Camadas de Aplicação e Apresentação no Modelo OSI [18].

Para atingir esse objetivo, neste trabalho optou-se por buscar o conhecimento adquirido nos padrões também destinados à representação digital e à transmissão de ECG já existentes, preservando-se assim, interoperabilidade com os mesmos, já que são produtos de amplas discussões entre os especialistas da área médica, e também, são adotados nas implementações dos dispositivos utilizados atualmente.

Sendo assim, é foco deste capítulo a análise dos requisitos de representação e transferência de ECG tomando como critério o reuso de conceitos, nomenclatura, e outros parâmetros que já foram investigados pelo AHA/MIT-BIH, SCP-ECG, FDADF, e *ecgML*. Vale lembrar que, existem outros importantes padrões de transmissão de informação médica em geral, com destaque para o DICOM<sup>1</sup> [44], o CORBAMED [45], o I-Med [46] e o HL7<sup>2</sup> [47], que não foram incluídos no escopo deste trabalho para priorizar os padrões específicos de ECG.

<sup>1</sup>O DICOM já está desenvolvendo um método para representação e transmissão de ECG através de imagem [43].

<sup>2</sup>Este padrão foi utilizado indiretamente, pois constitui a base para o FDADF.

## 4.2 Eletrocardiografia

Eletrocardiografia consiste na técnica de registrar o sinal elétrico gerado pela atividade cardíaca, desenvolvida ao longo da história através de contribuições de diversos pesquisadores, com destaque para o inglês Augustus D. Waller, e o alemão Willem Einthoven [48]. Basicamente, à medida que o impulso cardíaco cursa pelo coração, correntes elétricas se propagam para os tecidos que o cercam, atingindo ligeiramente a superfície do corpo. Dessa forma, a voltagem no coração pode ser medida através de sensores - pares de eletrodos ligados a um eletrocardiógrafo<sup>3</sup> - colocados sobre a pele do paciente em pontos opostos do coração [49] e registrada em papel gráfico ou em alguma mídia analógica ou digital. Esse registro, representado em forma de gráfico, é denominado eletrocardiograma [48].

Assim, o acrônimo ECG (ou EKG) representa ambos os termos, eletrocardiografia e eletrocardiograma, que conforme mencionado no capítulo introdutório, é o exame de coração mais empregado em cardiologia. Através do ECG, é possível diagnosticar diversas doenças do coração, que se caracterizam por modificações peculiares da forma de onda do sinal [3].

Para bombear o sangue para todo o corpo humano, o coração executa uma contração muscular periódica de tal forma que é possível distinguir no sinal ECG um batimento, composto de ondas elementares<sup>4</sup>, que se repete ao longo do tempo [3]. Na Figura 16 tal batimento é exibido com destaque para as ondas elementares associadas. O estudo dessas ondas e de suas amplitudes constitui a base da análise do sinal ECG.

---

<sup>3</sup>Aparelho criado por Einthoven composto basicamente de um galvanômetro [48].

<sup>4</sup>Também conhecidas como formas elementares, essas ondas foram identificadas e denominadas PQRST por Einthoven em 1895 [48].

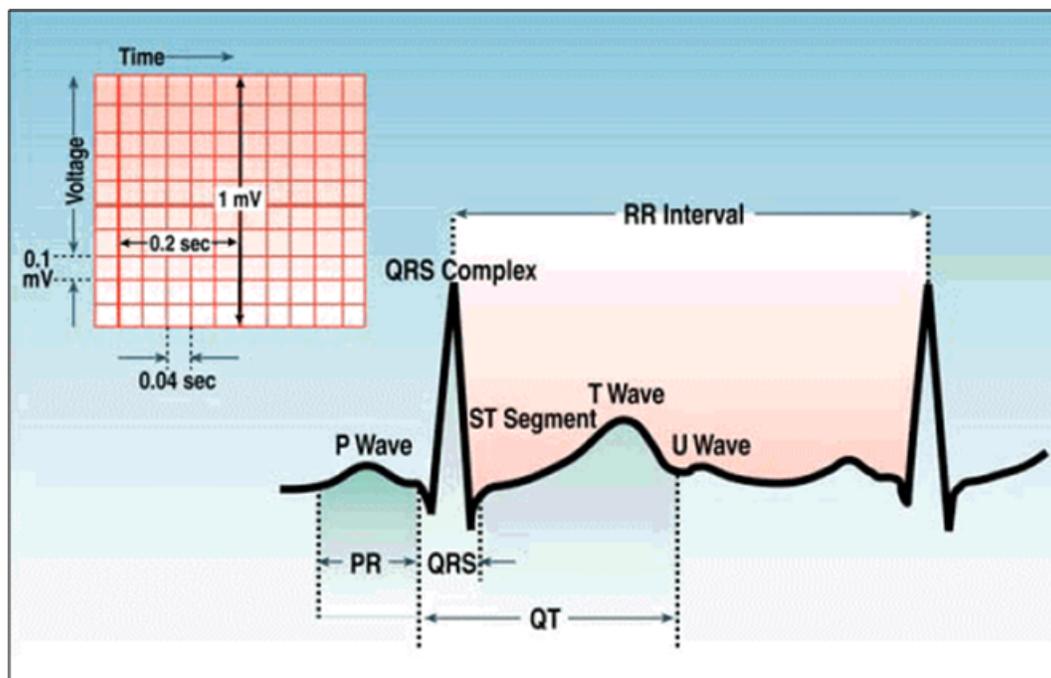


Figura 16: Ondas elementares, segmentos, e intervalos de um batimento elementar de ECG [50].

O eletrocardiograma normal é formado por (i) uma onda P, (ii) um complexo QRS que normalmente (mas nem sempre) aparece sob a forma das três ondas Q, R, e S, e (iii) uma onda T [49]. A onda P, e o complexo QRS, são causados pelos potenciais elétricos gerados, respectivamente, pelos átrios<sup>5</sup> e pelos ventrículos<sup>6</sup>, ao se despolarizarem antes de se contraírem. Sendo assim, tanto a onda P como o complexo QRS são ondas de despolarização. Já a onda T é causada por potenciais gerados à medida que os ventrículos se recuperam do estado de despolarização; esse processo ocorre no músculo ventricular, normalmente de 0,25 a 0,35 segundo após a despolarização, e essa onda é chamada de onda de repolarização [49]. A origem da onda U não é clara, mas provavelmente representa a situação pós-despolarização nos ventrículos [50].

Com relação aos intervalos entre as ondas, PR é o intervalo de tempo entre o ponto de início da despolarização dos átrios (onda P) ao ponto de início da despolarização dos ventrículos (complexo QRS). Já o QRS, representa a duração da despolarização

<sup>5</sup>Duas câmaras cardíacas, direita e esquerda, por onde o sangue chega ao coração vindo do corpo, e pulmões, respectivamente.

<sup>6</sup>Duas câmaras cardíacas, direita e esquerda, de onde o sangue parte do coração para os pulmões, e o corpo, respectivamente.

do músculo ventricular. O intervalo QT corresponde à duração da despolarização e repolarização ventricular. O RR é a duração do ciclo cardíaco ventricular (um indicador da frequência ventricular), e o PP é a duração do ciclo atrial (indicador da frequência atrial) [50].

Conforme já mencionado anteriormente, o sinal ECG é medido através de pares de eletrodos colocados no corpo humano. No entanto, para obter-se um sinal confiável, foi padronizada na eletrocardiografia a realização de doze derivações (*ECG leads*) do sinal, isto é, a mesma atividade cardíaca é medida separadamente por doze diferentes posições dos eletrodos no corpo humano escolhidas de forma a contemplar as diversas partes do coração, melhores vistas em cada uma dessas posições [18]. Dessa forma, é possível obter-se uma medição correta do potencial elétrico no coração através da combinação das diversas derivações. As doze derivações são classificadas em derivações (i) dos membros: I, II, III, aVR, aVL, e aVF; e (ii) do tórax: V1, V2, V3, V4, V5, e V6; e podem ser ilustradas conforme a Figura 17.

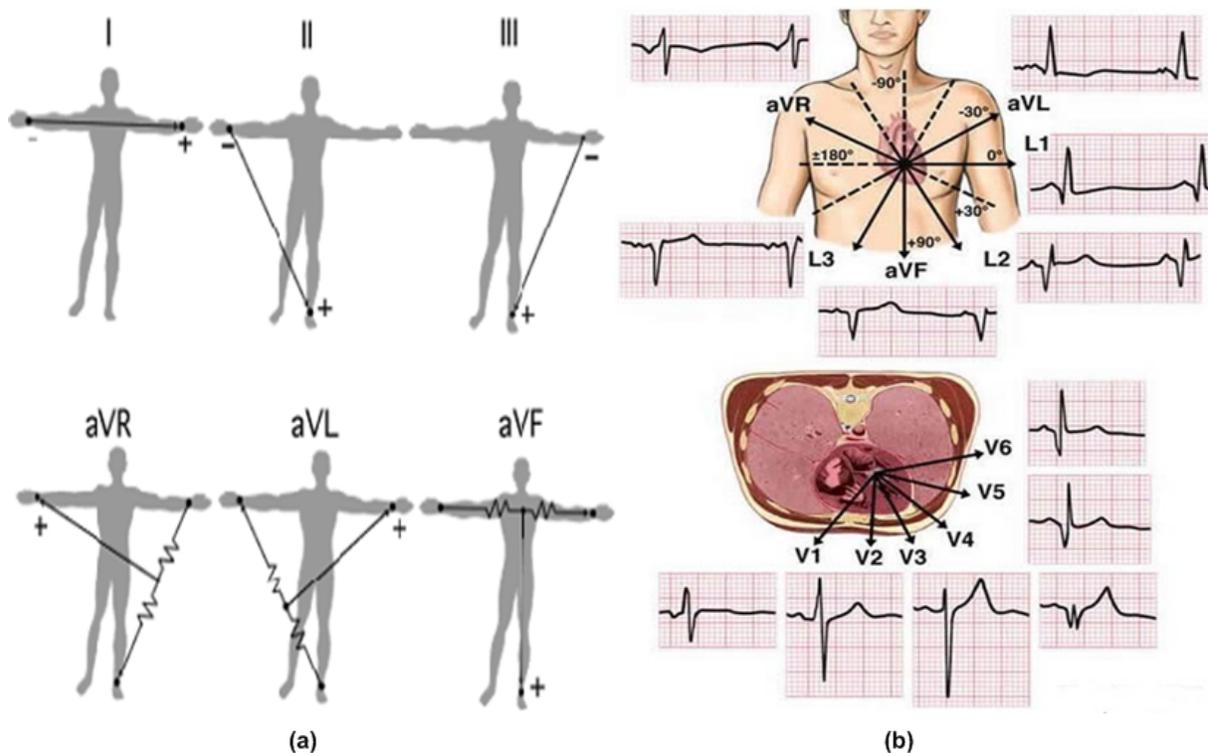


Figura 17: (a) Posições dos eletrodos nas derivações dos membros; (b) As doze derivações padronizadas na eletrocardiografia.

Do ponto de vista da análise do ECG, alguns distúrbios nas ondas elementares são facilmente identificáveis e associados a doenças; entretanto, existem outros que são de difícil percepção, até mesmo para um médico treinado [3]. Conforme [51], estudos demonstraram que de 2 a 8% dos pacientes examinados em prontos-socorros são equivocadamente diagnosticados. Encaminhados às suas casas, um quarto dessas pessoas morrem ou sofrem uma parada cardíaca completa. Sendo assim, é interessante que, além de ter acesso ao ECG, o médico possa contar com o auxílio de sistemas de análise do ECG no diagnóstico correto a respeito de um sinal de ECG, minimizando o erro humano.

No entanto, apesar da maior parte dos aparelhos de medição de ECG ambulatorial já possuir programas de análise do sinal e detecção de isquemia, quando avaliados por um especialista experiente os resultados obtidos são frequentemente considerados incorretos [3]. Além disso, os sistemas existentes não permitem que o especialista médico possa configurá-los no âmbito da classificação de arritmias. E, um outro ponto fraco, é a incapacidade desses sistemas de se adaptarem ao sinal ECG do paciente [3].

Diante dessa realidade, uma nova abordagem markoviana de análise do sinal ECG foi proposta por Andreão [25, 52, 53], e, segundo a avaliação da mesma, os resultados obtidos em termos de precisão e classificação do batimento cardíaco, e detecção de isquemia miocárdica comprovaram a sua consistência [3], confirmando assim, uma identificação precisa e robusta das formas elementares do batimento cardíaco. Essa abordagem será objeto da seção 7.3.

A seguir serão apresentados e discutidos padrões usados na eletrocardiografia, voltados à Telecardiologia, e o uso da tecnologia XML para o mesmo propósito. Questões relacionadas à aquisição do ECG, bem como sua representação visual em uma interface gráfica, também serão tratadas ao longo do texto.

### 4.3 AHA/MIT-BIH

Desde 1975, os laboratórios do BIH (*Boston's Beth Israel Hospital*, hoje *Beth Israel Deaconess Medical Center*) e do MIT (*Massachusetts Institute of Technology*) têm realizado em conjunto pesquisas em torno da análise de exames médicos e assuntos relacionados [54]. Como fruto dessas pesquisas, o primeiro produto a ser disponibilizado

foi o *MIT-BIH Arrhythmia Database* em 1980, um material testado e padronizado para avaliação e detecção de arritmias cardíacas, que tem sido utilizado em diversas pesquisas pelo mundo a cerca da atividade do coração [54]. Também entre o final dos anos 70 e início dos anos 80, a *American Heart Association* (AHA) criava o *AHA Database for Evaluation of Ventricular Arrhythmia Detectors* [55].

Posteriormente, surge o *Research Resource for Complex Physiologic Signals*, um projeto cooperativo criado por pesquisadores do BIH, MIT, *Harvard Medical School*, *Boston University*, e *McGill University*, e patrocinado pelo *National Institutes of Health's National Center for Research Resources* (NIH/NCRR) [56], tendo como objetivo estimular as pesquisas já em andamento e iniciar novas investigações a cerca dos complexos sinais fisiológicos. Esse projeto gerou três componentes ligeiramente relacionados [57]:

- ***PhysioNet*** [54], um web site destinado a dar apoio à comunidade científica biomédica sendo uma fonte de acesso a dados fisiológicos e softwares livres para manipulação desses dados;
- ***PhysioBank***, um banco de dados aberto com mais de 4000 registros de sinais vitais, especialmente ECGs, em sua maioria anotados - marcados com informações de semântica do sinal -, disponível em [54];
- ***PhysioToolkit***, um software de visualização, importação e exportação de dados, análise de sinais e simulações, também disponível em [54].

A partir de então, foi criada a *Waveform Database interface library* (WFDB library), uma biblioteca de funções C que provê um acesso uniforme aos sinais digitais, eventualmente já anotados, armazenados em variados formatos. Essas funções foram, originalmente, projetadas para bancos de dados de eletrocardiogramas, incluindo o MIT DB e o AHA DB. Em Fevereiro de 1990, o conjunto de anotações pré-definidas foi estendido para adaptar-se ao *European ST-T Database* (ESC DB). E assim, a *WFDB library* tem evoluído para atender o desenvolvimento de vários outros bancos de dados, incluindo sinais vitais como pressão sanguínea, respiração, saturação de oxigênio, e eletroencefalograma [58].

Para o foco deste trabalho, é importante ressaltar que a biblioteca de arquivos (*file*

*library*) do MIT-BIH disponibiliza *flat files*<sup>7</sup>, com nomenclatura e outros parâmetros a serem utilizados na formatação de informação médica (especialmente ECG). A seguir serão mostradas as definições dos conceitos pertinentes ao domínio em estudo, extraídas de [58].

### 4.3.1 Registros

Os bancos de dados para os quais foi projetada a *WFDB library* consistem em registros, de tamanho a partir de um megabyte. Até 1990, esses registros geralmente eram obtidos de mídias analógicas (*tapes*), e tinham de ser digitalizados; por essa razão histórica, é possível encontrar-se referências aos registros como *tapes*. Cada registro contém a informação contínua de somente um paciente. Uma típica aplicação acessa somente um registro por vez, e geralmente de forma seqüencial. Entretanto, eventualmente pode ser interessante comparar o conteúdo de diversos registros, ou de conjuntos selecionados de registros.

Registros são identificados por nomes, de tamanhos que variam até vinte caracteres (o limite é MAXRNL, definido em <wfdb/wfdb.h>); no MIT DB por exemplo os identificadores dos registros são de três dígitos numéricos, no AHA DB são de quatro dígitos numéricos, e já no ESC DB são de quatro dígitos numéricos precedidos pela letra 'e'.

Um registro é composto de uma coleção de arquivos, com possibilidade de extensão, que contém o sinal ECG, as anotações, e especificações dos atributos do sinal. Cada arquivo pertencente a um registro normalmente inclui o identificador do registro como parte inicial do seu nome, mas os arquivos não necessitam estarem localizados no mesmo diretório, nem no mesmo disco. Portanto, é possível, por exemplo, criar um arquivo local de anotações correspondentes a um registro lido de um servidor Web, ou de um CD-ROM, e tratar esse arquivo como parte do registro.

### 4.3.2 Sinais, Amostras, e Tempo

O conceito de sinal é comumente associado a uma função do tempo obtida através da observação de uma variável física que representa um fenômeno. No entanto, neste

---

<sup>7</sup>Um *flat file* é um arquivo de códigos, ou macros, que representam informações com significado para um banco de dados, ou sistema operacional [59].

escopo, de forma mais restrita, a definição de sinal é uma seqüência finita de amostras (pontos) inteiras, normalmente obtida digitalizando uma função contínua do tempo observada a uma determinada freqüência de amostras, expressa em Hz (amostras por segundo). Os registros de ECG do MIT DB são amostrados a 360 Hz; já os do AHA DB e do ESC DB a 250 Hz.

O intervalo de tempo entre qualquer par de amostras adjacentes em um dado sinal é um intervalo de amostra, de forma que todos os intervalos de amostra são iguais; o valor inteiro de cada amostra é interpretado como a voltagem. Além disso, uma amostra possui como atributo um número que a identifica, e uma unidade de tempo é exatamente um intervalo de amostra, portanto, o instante de tempo de uma amostra corresponde ao próprio número de amostra que a identifica. Isso faz com que, em diferentes registros, amostras contendo o mesmo número possam ser comparadas.

### 4.3.3 Anotações

Os registros de ECG no MIT DB, são de 30 minutos de duração, completamente anotados em arquivos de anotações, que são organizados sequencialmente, em ordem com relação ao tempo. Esses arquivos tipicamente possuem cerca de 2000 anotações correspondentes aos batimentos elementares, e outras do ritmo e da qualidade do sinal. Por outro lado, os registros do AHA DB são ou de 35 minutos ou de três horas de duração, e somente os 30 minutos finais são anotados. Já os registros do ESC DB possuem duração de duas horas, e também são anotados por completo.

O instante de tempo de cada anotação é exatamente o número da amostra à qual a anotação é associada, de forma que uma anotação corresponde a uma amostra. Comumente existem vários arquivos de anotações associados ao mesmo registro, mas eles são distinguidos por prefixos, da seguinte forma: o nome de anotação *atr* é reservado para identificar arquivos de anotação que rotulam os batimentos elementares. Assim, é possível criar seus próprios prefixos de anotação adotando-se qualquer critério. Além disso, anotações são vistas para o usuário da *WFDB library* como estruturas em C, como campos que podem especificar tempo, tipo de batimento, e diversas outras variáveis definidas pelo usuário. Assim, a *WFDB library* realiza eficientes conversões entre essas estruturas e esquemas de codificação compactos em binário usados para armazenar as anotações em arquivos.

### 4.3.4 Discussão sobre o AHA/MIT-BIH

Apesar das contribuições trazidas pelo AHA/MIT-BIH, que incluem os aspectos de análise do sinal ECG, esse tipo de formato de dados não permite uma boa legibilidade do ponto de vista humano, um requisito desejável para análise dos especialistas do domínio, e também não possui interoperabilidade através da Internet.

## 4.4 SCP-ECG

SCP-ECG (*Standard Communications Protocol for Computer-Assisted Electrocardiography*) é um padrão que especifica o formato e o procedimento de transmissão do registro ECG do dispositivo de aquisição para o *host* onde a mensagem será armazenada e posteriormente recuperada. Inicialmente, de 1989 a 1990, foi feito um levantamento dos métodos de compressão de ECG e uma nova abordagem para a compressão do sinal com garantia de qualidade foi desenvolvida [4]. Nesse período, houve uma cooperação de esforços entre fabricantes e usuários europeus, americanos e japoneses.

Posteriormente, em 1993, o *Comité Européen de Normalization* (CEN) [60] aprovou o SCP como o pré-padrão ENV 1064. Mais tarde, tornou-se recomendação ISO TC215, sendo constantemente atualizado pelos diversos grupos de trabalho WGI, WGII, WGIII, e WGIV - criados visando a sua melhoria. Este texto baseia-se na versão 1.3 (prEN 1064:2002) publicada em 2002 pelo CEN/TC-251 [61].

Também vale ressaltar que, colocar o SCP-ECG em destaque colaborando com outros grupos esforçados em padronização na cardiologia, é um grande objetivo de um outro projeto europeu mais amplo, e mais recente, chamado *OpenECG* [62], criado em 2002 para: (i) promover o uso consistente de padrões de formato e comunicação para ECGs processados por computador, e (ii) guiar o desenvolvimento de padrões similares para stress ECG, *Holter ECG*, e monitoramento em tempo real. Em suma, o *OpenECG* visa a interoperabilidade da eletrocardiografia computadorizada em nível europeu e internacional, estimulando o uso de padrões. Coordenado de forma interdisciplinar, o *OpenECG Consortium* busca atrair membros das autoridades de saúde, cardiologistas, engenheiros, instituições de padronização, fabricantes, e o público em geral [62].

#### 4.4.1 SCP-ECG Overview

A estrutura do registro ECG, definida em função dos requisitos desejáveis para a sua transmissão conforme a Figura 18 e a Figura 19, é dividida em diferentes seções. O conteúdo e o formato detalhados de cada uma dessas seções encontra-se em [61].

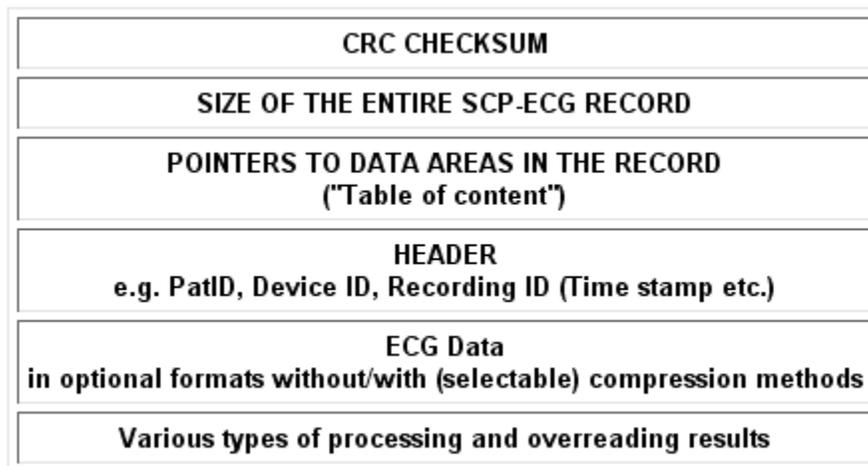


Figura 18: Estrutura do registro ECG [4].

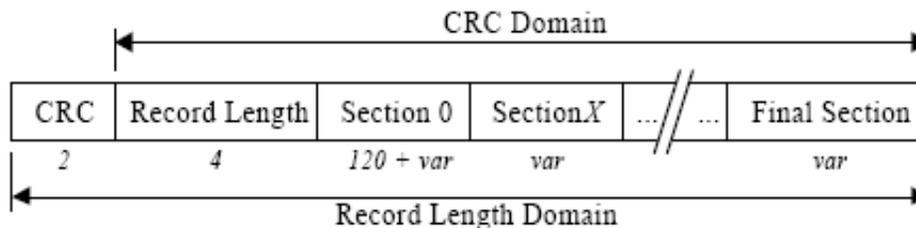


Figura 19: Registro ECG [61].

#### 4.4.2 Estrutura Detalhada do Registro SCP-ECG

A informação descrita de forma global no item anterior é desmembrada nas seções da maneira como é mostrado na Tabela 2.

Conforme pôde ser visto, a estrutura do registro SCP-ECG é auto-indicativa e estruturada de tal forma que é possível entender e distinguir o conteúdo de cada seção facilmente, e assim as aplicações viewers podem selecionar de acordo com as opções ou

Status	Section	Content
Mandatory	-	2 bytes - Checksum - CRC - CCITT over the entire record (excluding this word)
Mandatory	-	4 bytes - (unsigned) size of the entire ecg record
Mandatory	0	Pointers to data areas in the record
Mandatory	1	Header information - Patient data/ECG Acquisition data
Optional	2	Huffman tables used in encoding of ECG data (if used)
Optional	3	ECG lead definition
Optional	4	QRS locations (if reference beats are encoded)
Optional	5	Encoded reference beat data if reference beats are stored
Optional	6	Residual signal after reference beat subtraction if reference beats are stored, otherwise encoded rhythm data
Optional	7	Global measurements
Optional	8	Textual diagnosis from the "interpretive" device
Optional	9	Manufacturer specific diagnostic and over reading data From the "interpretive" device
Optional	10	Lead measurement results
Optional	11	Universal statement codes from the interpretation

Tabela 2: Descrição do conteúdo de cada seção SCP-ECG [61].

o perfil do usuário quais informações serão exibidas, por exemplo, mostrar a onda ECG com ou sem anotações.

Aumentando um pouco mais a granularidade, é possível visualizar na Figura 20 a estrutura básica de cada seção, que é dividida em duas partes: (i) *ID Header*, e (ii) *data part*; o tamanho do *Header* será sempre de 16 bytes, já o da parte de dados é variável.

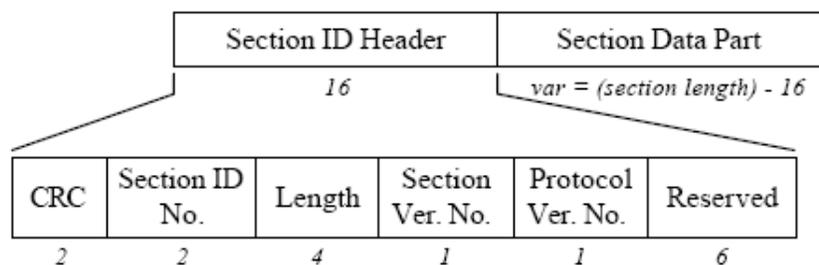


Figura 20: Estrutura da seção de um registro SCP-ECG [61].

O padrão SCP permite algumas opções de armazenamento e formatação do sinal; o ECG pode ser coletado com diferentes frequências de amostra, pode ou não ser com-

primido, e pode ou não conter anotações. Já outras informações, como o número de derivações (vide seção 4.2) e o tempo de duração do registro são deixados em aberto para os fabricantes [4].

Por outro lado, apesar do SCP-ECG atender a quesitos como: possuir alguma flexibilidade, potencial de compressão, e interoperabilidade; seu formato binário é ilegível a um humano, as possibilidades de anotações são limitadas, e além disso, sua implementação parece ser uma tarefa complexa especialmente no que se refere ao modo de compressão. Considerando que atualmente os recursos computacionais são muito mais desenvolvidos do que quando da criação deste padrão, largura de banda e capacidade de memória e armazenamento não são mais tão prioritários como antes [63]. Enquanto isso, outras demandas têm surgido com relevância, destacando-se a necessidade de modelos abertos que sejam independentes de plataforma, de sistema e de aplicação, e também que sejam facilmente interpretáveis por ambos humanos e máquinas [64]. Dessa forma, formatos baseados em XML como o FDADF [65], e o *ecgML* [64] têm rapidamente ganhado espaço na área de Telecardiologia.

## 4.5 XML

Extensible Markup Language (XML) [66] é um simples e flexível formato de texto derivado da SGML (ISO 8879), que apesar de originalmente projetado para solucionar os desafios da publicação eletrônica em larga-escala, também está exercendo um papel cada vez mais importante na troca de uma imensa variedade de dados na Web e em diversos outros domínios [67].

Desenvolvido como um subconjunto da SGML (*Standard Generalized Markup Language*), XML foi criado em 1996 pelo *World Wide Web Consortium* (W3C) [67] com a expectativa de, sem substituí-la, ser uma linguagem similar à HTML, porém extensível, e preocupada com a descrição dos dados [68]. Sendo assim, XML seria um novo padrão para a representação de dados através da Internet que tornaria possível a separação entre o conteúdo dos dados e a forma como são apresentados.

Passando a recomendação W3C em 1998, XML logo tornou-se largamente usado para representação e transmissão de dados pela Internet. Desde então, cada vez mais cresce o número de linguagens específicas de domínio definidas a partir de uma

*Document Type Definition* (DTD) ou, mais frequentemente, de um *XML Schema* (XSD) [69, 68].

Como resultado disso, comitês de diversas organizações da área de saúde como CEN/TC251, HL7, *American Society for Testing and Materials* (ASTM) dentre outras, vem trabalhando na elaboração de recomendações para o uso de XML na Telemedicina [70].

#### 4.5.1 XML Schema (XSD)

A função de ambos XSDs e DTDs é descrever a estrutura de um documento XML, definindo os blocos válidos que podem o compor. No entanto, o *XML Schema* merece destaque com relação ao DTD, pois (i) permite a definição dos tipos dos dados que irão preencher o *XML Document*, (ii) permite a restrição de quantas instâncias de cada elemento poderão compor o *XML Document*, e ainda (iii) é especificado na própria sintaxe XML. Desta forma, o XSD vem substituindo o DTD na função de descrever a estrutura dos documentos XML distribuídos na Web [68].

Neste sentido, um XSD representa um meta-modelo do domínio ao qual ele está associado gerando, assim, modelos que contêm as informações estruturadas de acordo com esse meta-modelo. Um exemplo de um XSD e um *XML Document* bem formado com relação a esse XSD é mostrado na Figura 21.



Figura 21: Meta-modelo XSD e um modelo *XML Document* gerado a partir desse XSD.

## 4.6 FDA XML Data Format (FDADF)

*Food and Drug Administration* (FDA) é uma das mais antigas criada em 1902 - e respeitadas agências americanas de proteção ao consumidor; O CDER (*Center for Drug Evaluation and Research*) da agência, que verifica a segurança e eficiência das drogas disponíveis nos Estados Unidos, tem proposto diversas recomendações para a troca de informações médicas, especialmente ECGs [71].

Uma das atividades da FDA é a coleta e transmissão de dados biológicos, normalmente em forma de onda, adquiridos dos pacientes. Os objetivos principais da agência são facilitar a submissão desses dados, e proporcionar precisão e consistência

nas análises realizadas a partir deles [65]. Depois de analisar os padrões já existentes, a FDA optou por usar XML como tecnologia para essa especificação, baseando-se na 3ª versão do padrão de mensagem HL7 (*Health Level Seven*) [72]; isso, além de atingir a linha estratégica da agência para submissão de dados, também está alinhado com as iniciativas de outras indústrias como o *Clinical Data Interchange Standards Consortiums Submission Data Model* (CDISCs SDM) [65].

Assim, em abril de 2002, foi elaborado o *FDA XML Data Format* (FDADF), que é especificado em [65] com o objetivo de definir uma representação digital do ECG, e garantir que os *stakeholders* (indivíduos e organizações) do projeto tenham o mesmo entendimento sobre ela. O escopo do documento citado abrange os dados de ECG propriamente ditos, e também as informações de submissão relevantes, procurando identificar elementos que casem perfeitamente com os requisitos previamente levantados em [73].

Os elementos XML (*tags*) definidos no FDADF são derivados do *HL7 Refined Message Information Model* (R-MIM), que define todos os componentes de uma mensagem HL7 específica. O R-MIM é composto de um conjunto de elementos que podem ser **ações**<sup>8</sup>, **entidades**, ou **papéis** [74]. São exemplos de ações, um procedimento, ou uma observação médica; e exemplos de entidades são, uma pessoa, uma organização, etc. Cada entidade que participa na execução de uma ação, o faz com um papel específico, por exemplo como paciente, ou como médico. Além disso, uma única mensagem pode conter diversas ações que se relacionam entre si; isso pode ser ilustrado na ação de especificar anotações ECG que se relaciona com a ação de especificar o próprio registro ECG ao qual as anotações correspondem [74]. Na Figura 22 é mostrado um esquema no qual os diferentes conceitos HL7 compõem a estrutura de uma mensagem.

---

<sup>8</sup>Uma ação neste escopo tem o sentido de um registro já feito, sendo feito, ou por fazer.

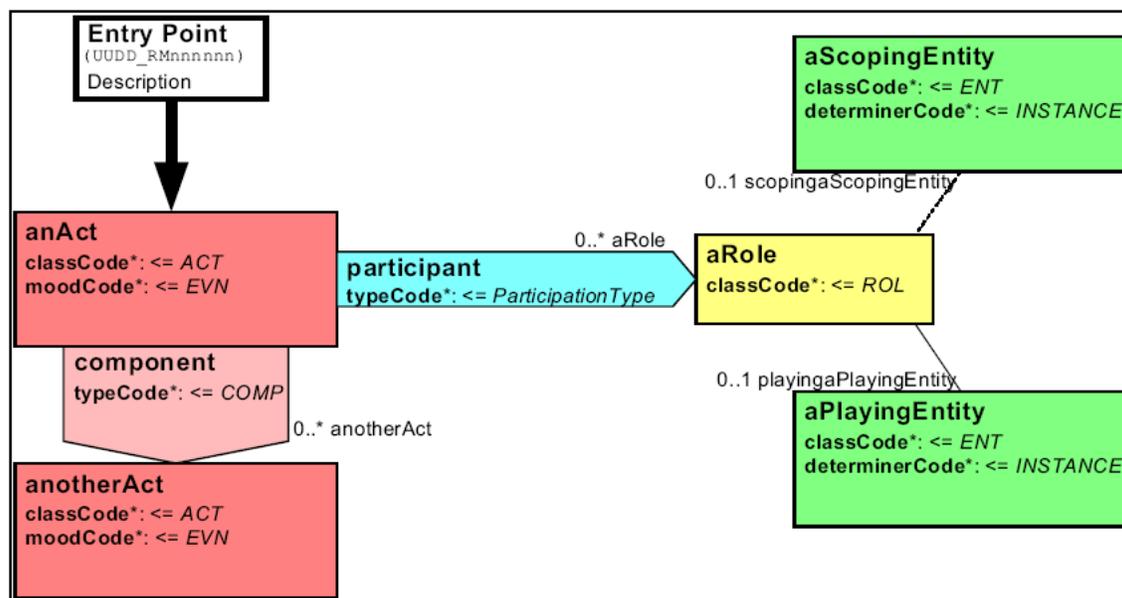


Figura 22: Exemplo de um R-MIM [74].

Cada R-MIM do HL7 é univocamente mapeado em uma estrutura XML que, conforme a Figura 22, possui como ponto de partida o *Entry Point*, correspondendo ao elemento XML raiz (*root element*). Dessa forma, cada elemento HL7 é mapeado em um elemento XML com atributos correspondentes às suas características no R-MIM. A título de exemplo, uma anotação ECG em HL7 torna-se uma *tag* XML com atributos especificando o tipo da anotação (seja uma onda P, ou uma T, etc).

#### 4.6.1 Sinal

O sinal ECG gravado sincronizadamente através dos diversos canais do dispositivo de aquisição é representado no HL7 como uma ação denominada **Série**, que é composta de uma ou mais ações chamadas **Conjunto de Seqüência**. Uma **Seqüência** também é uma ação, que consiste em uma coleção de valores coletados no tempo através de um canal, como uma seqüência de amostras de uma derivação ECG (vide seção 4.2), ou uma seqüência de instantes no tempo. Diversas *seqüências*, quando e somente quando gravadas simultaneamente em cada canal, compõem uma instância de um *Conjunto de Seqüência*. Na terminologia do HL7, *seqüências* pertencentes ao mesmo *Conjunto de Seqüência* são observações correlacionadas. A título de exemplo, um ECG gravado simultaneamente através de 12 canais possui uma instância de *Conjunto de Seqüência*

(todos os 12 canais correlacionados), assim como um ECG com 4 *sequências* de 3 derivações cada, é representado por 4 instâncias de *Conjunto de Sequência* cada uma associada com 3 canais de observações correlacionadas [74].

As relações entre tempo real, uma sessão de registro, quaisquer conjuntos de dados, etc podem ser vistas como um sistema de coordenadas relacionado. Os diversos tipos de sequências de dados podem compartilhar a dimensão *Tempo* relacionada às sessões de registro sincronizando esses dados no *eixo-X* através de uma simples transladação e/ou transformação de escala. Se um desses tipos não está diretamente ligado ao tempo - por exemplo, uma média de batimentos representa uma frequência, e não tempo corrido -, esse tipo apenas possuirá alguma relação com uma determinada sessão de registro, mas não pode ser sincronizado e plotado no gráfico. Por isso, cada tipo de dados possui atributos com relação à sessão de registro da qual esses dados foram obtidos, que são relevantes em uma eventual sincronização para representá-los no gráfico [65].

### 4.6.2 Anotações

As anotações são representadas por uma ação denominada **Conjunto de Anotações**, composta de ações distintas chamadas de **Anotação** geradas por uma mesma entidade (que pode ser uma pessoa ou um dispositivo), representando o papel de autor das anotações. As *anotações* que compõem um *Conjunto de Anotações* de uma entidade podem ser de cinco diferentes escopos<sup>9</sup>: batimento elementar, onda elementar, ritmo, marcapasso, e ruído. É incorporado no padrão HL7, um dicionário de tipos específicos de anotação pertencentes a cada escopo, mas isso não impede o usuário de definir os seus próprios tipos, criando assim, um dicionário personalizado [74].

### 4.6.3 Visualização de ECG

Um outro aspecto relevante do FDADF, é que nele é considerado com especial atenção o quesito de plotagem do gráfico que representa o ECG. Além disto, podem ser encontradas a livre acesso na Internet ferramentas de visualização de ECGs no formato FDA XML. A Figura 23 mostra a GUI da aplicação *XMLFDA Viewer* [75].

---

<sup>9</sup>O escopo é especificado na propriedade *code* da ação *Anotação*.

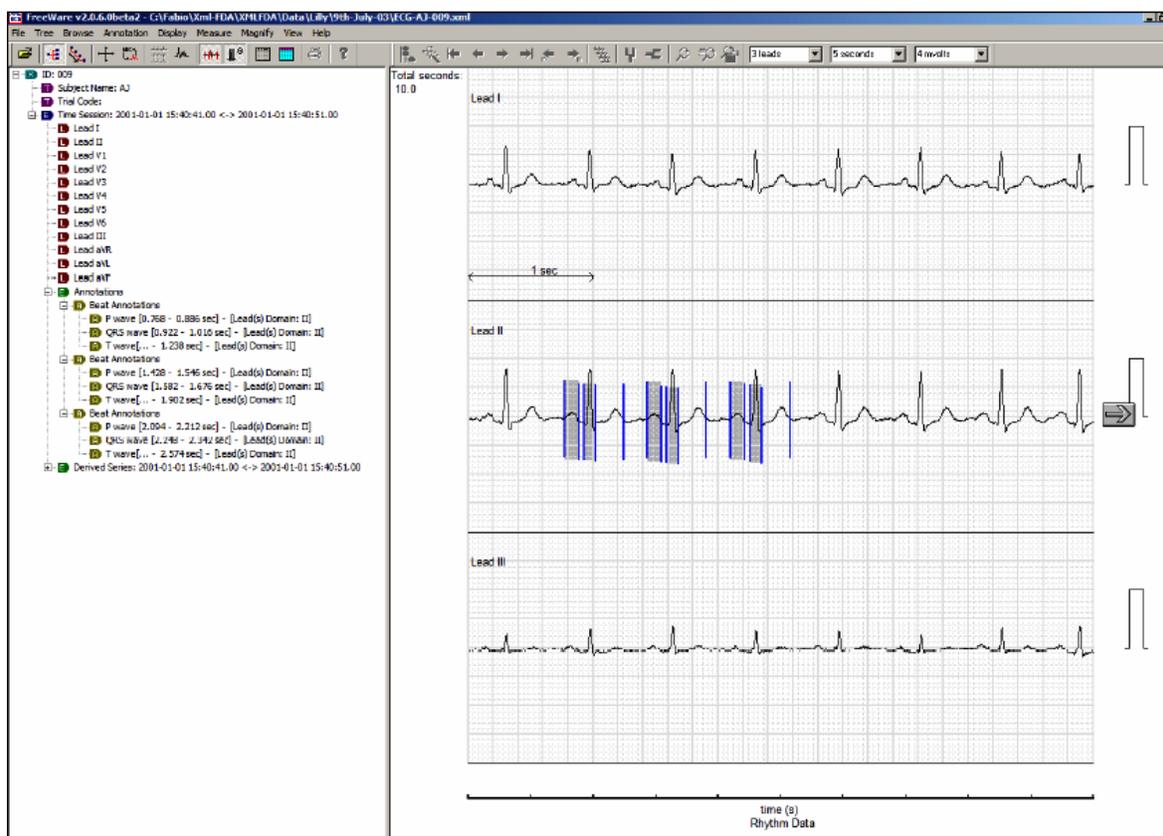


Figura 23: Screenshot de uma aplicação de visualização de ECG [74].

#### 4.6.4 Comentários Gerais sobre o FDADF

O FDA apresenta em [76] toda a análise do domínio de aquisição de ECG, bem como um glossário dos termos utilizados nesse domínio. Já em [65], é especificado um Modelo Entidade-Relacionamento, um XML DTD, e um *XML Schema* para a formatação do ECG.

O FDADF proporcionou um avanço significativo no que se refere à representação de ECG utilizando XML. Entretanto, em concordância com o argumento de [70], esse formato não explorou perfeitamente os benefícios oferecidos por essa tecnologia, pois ele incorporou na sua representação do ECG, elementos (*tags*) relacionados à apresentação, e não somente ao conteúdo. A motivação para a inclusão desses elementos foi a de que eles são úteis na apresentação do ECG em aplicações *ECG Viewers*.

## 4.7 *ecgML*

Diante da necessidade de tornar homogênea e independente de plataforma a representação do sinal ECG adquirido de diferentes dispositivos, bem como manter na mensagem de transmissão informações relevantes como identificação do paciente e do dispositivo, e também interpretações do sinal [70], foi criado o *ecgML*, um *XML Schema* concebido a partir do estudo de diversos padrões de representação de ECG buscando apoiar a análise e a transmissão do ECG entre plataformas heterogêneas [64]. *ecgML* consiste em uma síntese das experiências adquiridas pelos vários grupos de trabalho dedicados ao mesmo problema já citados neste texto, contudo, baseou-se especialmente no FDADF, agregando assim, as grandes contribuições desse padrão, mas trazendo como vantagem uma maior separação entre conteúdo e apresentação.

A seguir, será apresentada a estrutura hierárquica *ecgML* de representação do ECG, especificada em ambos os documentos [64] e [70]. Seguindo a terminologia dos autores, serão exibidos em negrito e itálico, ambos elementos e atributos XML, porém, os elementos começando com letra maiúscula e os atributos com letra minúscula. A cardinalidade e a obrigatoriedade de ambos os componentes, serão expressas nos diagramas ilustrados nas figuras através da seguinte simbologia: o caractere '+' representa um ou vários componentes; '?' significa zero ou um; o símbolo '\*' quer dizer zero ou vários; e quando não houver nenhum símbolo a cardinalidade é de necessariamente um.

### 4.7.1 Estrutura *ecgML*

No *ecgML*, o documento XML, que representa um estudo de ECG de um paciente, começa com o elemento raiz ***ECGRecord***, que é identificado univocamente pelo seu atributo *studyID*. Esse elemento raiz possui como filhos os elementos: ***StudyDate*** e ***StudyTime***, que representam o dia e a hora do último registro de todo o estudo do ECG; ***Diagnosis***, que contém em texto o último diagnóstico do ECG; ***Medical-History***, uma descrição do histórico médico (doenças e diagnósticos) do paciente; e finalmente os principais componentes do estudo, que são, um elemento ***PatientDemographics***, e um ou mais elementos ***Record***. A estrutura do elemento raiz ***ECGRecord*** pode ser visualizada na Figura 24.

Expandindo o nível de granularidade, dois dos elementos filhos de ***ECGRecord*** são

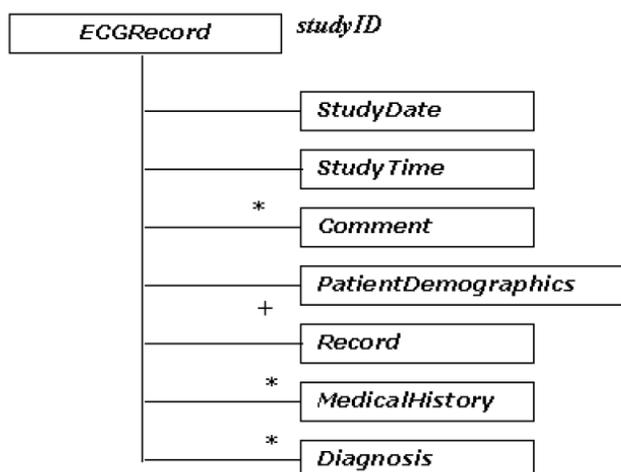


Figura 24: Diagrama do elemento *ECGRecord* [70].

compostos de outros elementos. *PatientDemographics*, conforme mostra a Figura 25, contém informações pessoais e para contato do paciente.

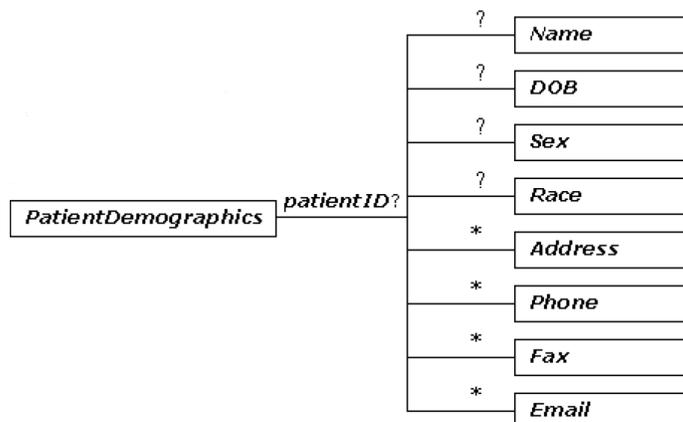


Figura 25: Diagrama do elemento *PatientDemographics* [64].

Como principal componente do elemento raiz, um ou mais elementos do tipo *Record*, como ilustrado na Figura 26, representam cada registro de sinal ECG propriamente dito, ou seja, o conteúdo básico de todo o estudo de ECG, armazenado fisicamente. Este elemento possui como atributos *investigatorID* e *siteID*, usados para identificar a pessoa e a instituição responsáveis por cada registro, e como elementos filhos: *AcquisitionDate* e *AcquisitionTime*, especificando, respectivamente, o dia e o instante nos quais cada registro foi obtido; *ClinicalProtocol*, incluindo um relatório clínico do

paciente (exibido no diagrama da Figura 27); *RecordingDevice*, descrevendo, conforme o diagrama da Figura 28, o dispositivo de aquisição dos dados; e finalmente o seu elemento mais importante e, elemento chave no *ecgML*, *RecordData*.

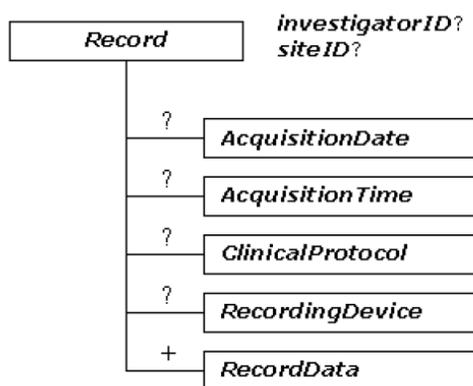


Figura 26: Diagrama do elemento *Record* [70].

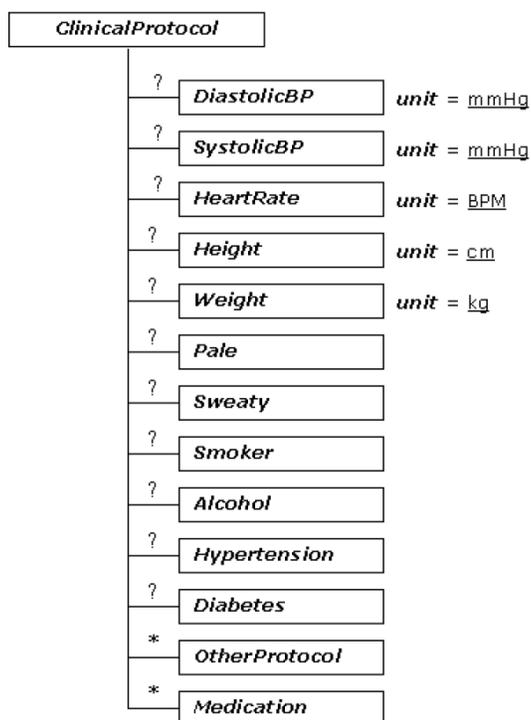


Figura 27: Diagrama do elemento *ClinicalProtocol* [64].

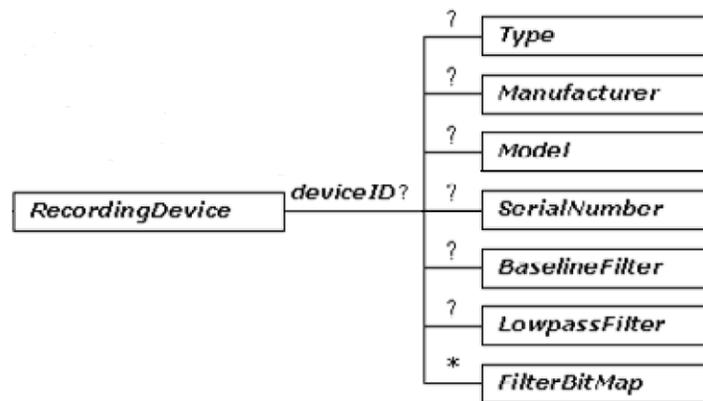


Figura 28: Diagrama do elemento *RecordingDevice* [64].

Existem um ou mais elementos *RecordData* em um arquivo (um estudo de ECG), que correspondem aos sinais coletados em diferentes canais. Portanto, seu elemento filho *Channel*, baseado no [43], identifica a derivação do ECG associada a um canal do dispositivo de aquisição do ECG. Além desse elemento, *RecordData* inclui, conforme a Figura 29, outros três sub-componentes: *Waveforms*, *Annotations*, e *Measurements*, todos associados ao mesmo canal.

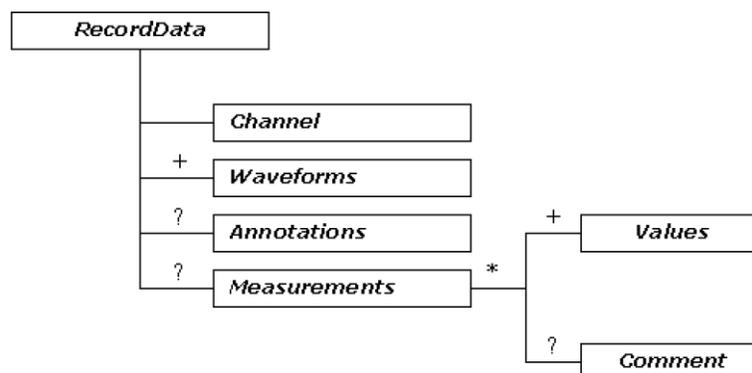


Figura 29: Diagrama do elemento *RecordData* [70].

*Waveforms*, baseado no formato recomendado pelo *PlotGroup* do FDA [65], são representadas por uma série de valores, associados aos eixos X e Y, denominados *XValues* e *YValues*. A Figura 30 exibe a estrutura deste elemento em um diagrama.

O elemento *Annotations* é utilizado tipicamente para marcar no sinal eventos rele-

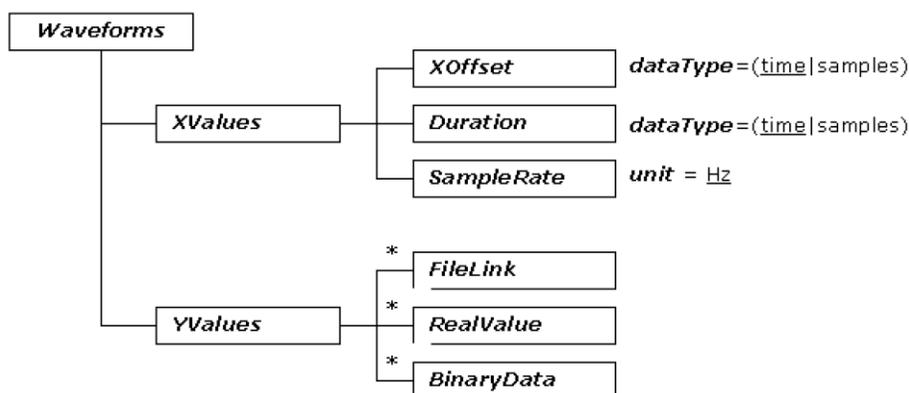


Figura 30: Diagrama do elemento **Waveforms** [64].

vantes. Ele define, como mostrado na Figura 31, pontos ou intervalos no tempo contendo informações de marcação e classificação relativas às ondas elementares pertencentes à derivação obtida no canal, através de uma coleção de elementos **PointNotation** e **WaveNotation**.

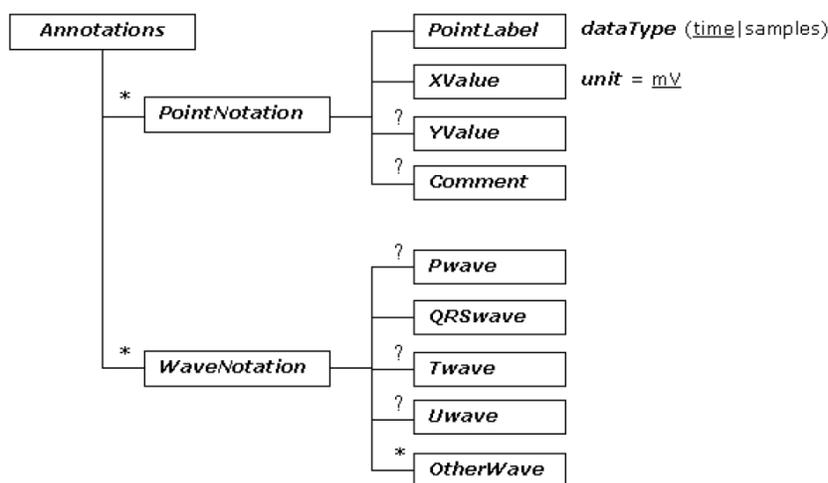


Figura 31: Diagrama do elemento **Annotations** [64].

Por último, o elemento **Measurements** (Figura 29) consiste em medições eventualmente acompanhadas de observações, relativas às ondas elementares identificadas nas anotações. Este elemento contém uma lista de elementos **Values**, de forma que, cada um possui os atributos **label** e **unit**, e o elemento **Comment**.

### 4.7.2 Comentários Gerais sobre o *ecgML*

O *ecgML* é um interessante padrão aberto de representação e transmissão do sinal ECG através da Internet entre diferentes plataformas, e dispositivos de aquisição e visualização de ECG. Além disso, assim como o FDADF, por ser definido em XML, ele possui importantes qualidades. Dentre as quais estão: ser bastante legível ao usuário, auto-descritivo, e aberto a modificações.

No entanto, conforme seus próprios criadores mencionam em [64], existem questões relevantes a serem analisadas como:

- A mensagem *ecgML* não seria excessivamente preenchida com informações desnecessárias?
- Uma compressão como a realizada pelo protocolo SOAP, é suficiente para transmitir esse tipo de mensagem XML?
- *ecgML* é apropriado para aplicações de 24 horas de monitoramento?
- Existiria algum tipo de informação não contemplada neste formato?

Dessa forma, é possível concluir que, assim como os outros padrões apresentados neste trabalho, apesar de possuir alguns aspectos negativos, *ecgML* oferece contribuições significativas no problema da representação e transmissão do sinal ECG, sobretudo através da Internet, que são úteis na elaboração de um novo formato para tal propósito.

## 4.8 Resumo do Domínio e Especificação de Requisitos

Após o estudo dos padrões apresentados neste trabalho, foram capturados de forma geral: conceitos, nomenclatura, parâmetros e outras características do ECG relevantes para a sua modelagem, com vistas ao processo de aquisição, processamento (incluindo a análise automática do ECG), transmissão, e posterior visualização do ECG através de uma GUI na aplicação a qual o médico terá acesso. Será descrito a seguir um resumo desse domínio, e em seguida será apresentada uma especificação de requisitos para um formato de representação e transmissão de ECG.

### 4.8.1 Resumo do Domínio

Um *paciente* monitorado tem o seu *estudo de ECG* coletado pelo *dispositivo de aquisição* em uma ou mais *sessões*; cada sessão gera um *registro de ECG*, composto de uma ou mais *derivações* obtidas simultaneamente cada uma em um *canal* do dispositivo, e por isso, é padronizado o termo canal referir-se a derivação. Todo esse conjunto de dados é relativo ao período de tempo, com início e duração, no qual o sinal (todas as derivações obtidas concomitantemente) foi obtido.

De caráter bidimensional, os valores das *amostras* são, na verdade, coordenadas no *eixo-X* (tempo) e no *eixo-Y* (voltagem) que podem ser plotados em um gráfico, de modo que a sua forma de onda representa visualmente o ECG, permitindo sua interpretação semântica pelo médico. Entretanto, o gráfico *Voltagem vs. Tempo* não é o único conjunto de dados que pode ser obtido. Informações adicionais, como a frequência cardíaca, podem ser derivadas através da realização de *anotações* e *medições* no sinal de ECG; ambas são efetuadas para destacar eventos relevantes. Anotações e medições podem ser de três níveis distintos: de registro, de múltiplas derivações, e especificamente de uma derivação. Mas a diferença entre elas, é que anotações fornecem uma indicação precisa da localização de eventos conhecidos, principalmente dos *pontos de referência* e das *ondas elementares* no *batimento*, e já as medições são voltadas à intensidade e duração desses e de outros eventos. Essas informações normalmente são incluídas nos conjuntos de dados estatísticos do ECG.

Para o caso da obtenção da *frequência cardíaca*, são anotados os picos *das ondas de referência* de ciclos cardíacos consecutivos, conhecidas como *Complexo QRS*, e, em seguida, mede-se os intervalos entre os complexos QRS, chamados de *intervalo RR*. Por fim, o valor médio da frequência cardíaca referente a esses ciclos cardíacos é calculado, como sendo o inverso da média dos intervalos RR. A Tabela 3 provê um dicionário de termos para o domínio.

### 4.8.2 Especificação de Requisitos

Tomando como base os padrões apresentados nas seções anteriores, são especificados e dispostos na Tabela 4, os requisitos para *(i)* a sintaxe de representação, e *(ii)* a sintaxe de transmissão do ECG. Os requisitos definidos aqui foram elaborados sob o uso de

Termo	Significado
ECG, também encontrado como EKG (do alemão Elektrokardiogramm)	Eletrocardiografia, ou eletrocardiograma.
Eletrocardiógrafo	Dispositivo de medição de ECG.
Holter	Eletrocardiógrafo de medição de ECG ambulatorial em períodos de 24 a 48h de duração.
Sinal ECG	Conjunto de valores do tipo par-ordenado Voltagem vs. Tempo que representa o ECG formando uma onda bidimensional. O sinal ECG é composto de uma ou mais derivações.
Derivação (do inglês lead)	Sinal ECG obtido através de uma determinada posição do par de eletrodos colocado no corpo humano de forma a medir a atividade cardíaca sob a visada correspondente a essa posição.
Canal	Corresponde a uma derivação de ECG limitada em uma frequência.
Registro	Sinal ECG representado e armazenado digitalmente que integra uma ou mais derivações obtidas simultaneamente em diferentes canais.
Sessão	Período de tempo no qual foi feita uma coleta de ECG armazenada em um registro.
Estudo de ECG	Um ou mais registros de ECG obtidos cada um em uma sessão de registro.
Batimento	Contração muscular periódica do coração que corresponde a um ciclo no sinal ECG.
Onda elementar, ou forma elementar	Formas de onda que compõem um batimento elementar. São classificadas em: P, Q, R, S, T e U.
Pontos de referência	São os pontos mais relevantes do sinal. Os principais são os que começam ou terminam uma onda elementar, bem como os valores de pico dessas ondas.
Intervalo	O intervalo entre dois pontos de referência.
Segmento	Parte de um intervalo que interliga ondas elementares consecutivas.
Amostra	Um valor do sinal ECG no gráfico, que corresponde ao potencial elétrico medido em um instante de tempo.
Offset do registro	Instante de tempo do início do registro.
Frequência de amostragem	A partir dela e do Offset obtém-se todos os valores do eixo-X do gráfico.
Anotação, ou marcação.	Identificação de um evento relevante no sinal de ECG.
Medição	Medida de intensidade ou duração de um fenômeno relevante no ECG.

Tabela 3: Dicionário de Termos do domínio de representação digital de ECG.

critérios relacionados à conjuntura da plataforma Infracore, e aos objetivos do projeto TeleCardio.

Esses requisitos irão direcionar a elaboração de um formato de representação de ECG que será especificado no Capítulo 5.

R#	Descrição	Natureza
2.1	Sempre que possível, reusar conceitos, nomenclatura, e estruturas de dados dos padrões já existentes.	Reuso e Interoperabilidade
2.2	No caso do modelo de entrega time-driven (ver seção 6.6.2), permitir a transmissão periódica do sinal ECG à plataforma Infracore, com um atraso menor do que o intervalo de um período.	Performance
2.3	O formato de representação deve ter uma estrutura hierárquica tal que seja fiel às relações entre os conceitos pertencentes ao domínio de ECG.	Consistência
2.4	É desejável que seja realizada compressão, podendo esta ser feita no nível de arquivo, ou no de registro.	Performance
2.5	Possuir um identificador único para cada registro de ECG dentro do escopo de uma mensagem.	Consistência
2.6	O formato de representação deve ser user-driven, flexível, e focar somente no conteúdo e não na apresentação da informação.	Portabilidade
2.7	Comparativamente ao ecgML, deve ter um cunho mais prático e objetivo, para que o tamanho do arquivo da mensagem seja menor.	Performance
2.8	Desejável incorporar informações relativas ao contexto do paciente, como suas coordenadas de localização, a atividade realizada durante a sessão de registro do ECG, etc.	Context-Awareness
2.9	Preservar a semântica recebida do sistema de análise do sinal em forma de anotações, medições, laudos, etc, através de elementos destinados a esse propósito.	Compleitude
2.10	Desejável possuir um elemento destinado a uma breve e descomprometida representação do Prontuário Eletrônico do Paciente (PEP).	Compleitude

Tabela 4: Requisitos de Representação e Transmissão de ECG.

## 4.9 Conclusão do Capítulo

Neste capítulo foi apresentado o domínio de representação e transmissão de eletrocardiogramas sobretudo através do estudo de padrões de referência destinado a esse propósito. Em seguida, foi descrito um resumo do domínio e apresentada uma especificação de requisitos para um formato de representação e transmissão de ECG. O conhecimento adquirido e documentado neste capítulo contribuiu para o aprendizado dos conceitos pertencentes a tal domínio, bem como para o entendimento de benefícios e prejuízos a serem avaliados em uma abordagem de representação de ECG.

Os padrões aqui descritos podem ser divididos entre os que são baseados em XML,

e os que não são. Não obstante o AHA/MIT-BIH e o SCP-ECG (padrões que não são baseados em XML) se tratarem de especificações implementadas em muitos dos dispositivos e sistemas legados da área médica, é notório o fortalecimento dos padrões que se baseiam em XML como o FDADF e o ecgML. Estes tiram proveito principalmente das características de interoperabilidade na Internet e legibilidade ao usuário proporcionadas pela meta-linguagem de marcação, XML. Na conjuntura atual, em que recursos computacionais têm se tornado cada vez mais disponíveis, tais características têm se destacado como prioritárias quando comparadas, por exemplo, com eficiência.

No entanto, com o surgimento de um novo paradigma de Computação, o da Computação móvel e ubíqua, sistemas sensíveis ao contexto oferecem novas vantagens, com destaque para a colaboração entre profissionais, entre sistemas, e o desencadeamento de ações a partir da detecção na mudança de contexto do usuário, no caso deste trabalho alguma alteração relevante no ECG de um paciente. Levando isso em consideração, é necessário explorar essa característica diferenciada que pode ser oferecida por plataformas de apoio a sistemas sensíveis ao contexto.

Sendo assim, de acordo com essa questão até mesmo os padrões baseados em XML descritos neste capítulo, já não podem oferecer algumas facilidades potenciais no apoio à decisão médica, e no disparo automático de ações em situações emergenciais. Neste sentido, faz-se necessária uma abordagem que privilegie a **utilização** da informação clínica. No próximo capítulo será apresentado um formato de representação de ECG capaz de explorar as vantagens supracitadas.

## 5 *Especificação de um Formato de Representação de ECG*

Após o estudo no Capítulo 4, dos padrões de representação e transmissão de ECG, e primordialmente a partir dos requisitos da Tabela 4, é definido aqui o *ecgAware*, um formato relativo à camada de Aplicação do Modelo OSI. O *ecgAware* é destinado a encapsular (i) o sinal de ECG obtido do dispositivo portátil, (ii) informações derivadas da análise do sinal de ECG realizada no sistema de análise automática de ECG, e (iii) dados do paciente inseridas no sistema através de uma GUI (seção 7.6), a fim de possibilitar a transmissão dessas informações, sobretudo através da Internet.

Para atender ao requisito R2.6 da Tabela 4, confirmando assim a tendência mencionada no Capítulo 4 que aponta para padrões mais representativos semanticamente, e garantir, conforme os requisitos R1.4 e R1.6 da Tabela 1, a interoperabilidade entre os *Web Services* oferecidos pelos componentes da plataforma Infracore, a metalinguagem de marcação XML foi escolhida como tecnologia de suporte ao *ecgAware*. Dessa maneira, a definição do formato consiste, na verdade, na especificação de uma linguagem específica de domínio na forma de um *XML Schema*.

Por outro lado, o formato em XML não atinge da melhor maneira os requisitos R2.2, R2.4, e R2.7 da Tabela 4, relativos a questões de otimização e compressão no envio das informações através de redes de computadores. Portanto, é desejável, conforme discutido em [18], que o formato XML seja posteriormente convertido para algum formato binário eficiente com relação a esse critério. No entanto, essa demanda não será atendida neste trabalho, e ficará a cargo de trabalhos futuros.

Enquanto isso, o *ecgAware*, como o seu próprio nome sugere, traz como característica diferenciada com relação aos outros formatos, o cumprimento do requisito R2.8, permitindo assim, a exploração do contexto do paciente como mais uma fonte de infor-

mação útil na tomada de decisões.

Afora isso, então, uma vez que a criação de um *XML Schema* na sua forma tradicional textual trata-se de uma tediosa tarefa nos casos que envolvem uma quantidade razoável de tags a serem incluídas, a ferramenta *XML Spy*, da Altova<sup>1</sup>, foi adotada para facilitar a especificação do meta-modelo *ecgAware*, já que permite que essa tarefa seja feita em notação de diagrama. E também, com isso a sua legibilidade pôde ser mantida ao longo de toda a tarefa.

A seguir, na seção 5.1 será apresentada a estrutura hierárquica *ecgAware* de representação do ECG em *XML Schema*, exibindo em negrito e itálico, ambos elementos e atributos XML, porém, os elementos começando com letra maiúscula e os atributos com letra minúscula. A árvore correspondente ao *ecgAware* será percorrida em pré-ordem (*prefix*), isto é, apresentando e expandindo enquanto for possível cada elemento complexo encontrado no caminho.

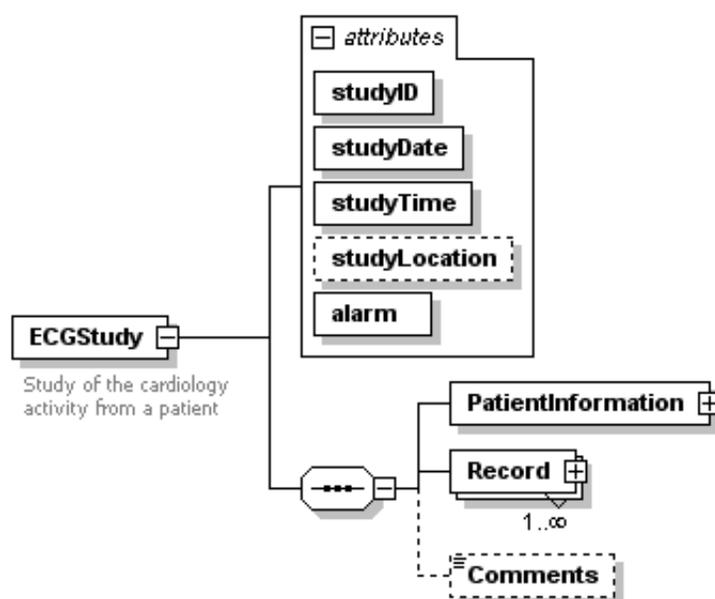
## 5.1 *ecgAware*

A mensagem *ecgAware*, isto é, o documento XML, consiste em um estudo de ECG de um paciente, que tem como elemento raiz ***ECGStudy*** (Figura 32 e Tabela 5), composto dos atributos ***studyID***, que identifica a mensagem, ***studyDate*** e ***studyTime***, que representam respectivamente o dia e a hora do último registro de ECG que compõe a mensagem, ***studyLocation***, que opcionalmente indica a última localização geográfica obtida durante a coleta do último registro de ECG, e ***alarm***, que sinaliza se houve ou não no estudo alguma anomalia digna de urgência; e de três elementos filhos: (i) ***PatientInformation***, encapsulando dados pessoais do paciente e de seu prontuário eletrônico; (ii) de um ou vários elementos ***Record***, que contém o registro de ECG propriamente dito; e também de (iii) um elemento ***Comments***, que permite em texto livre comentários gerais sobre o estudo.

O elemento ***PatientInformation*** possui dois atributos, ***patientID*** e ***record-Number***, necessários para conter o identificador único do paciente e o número de registros de ECG já coletados e armazenados pelo sistema de monitoramento. Além disso, esse elemento é composto dos elementos filhos ***Demographics*** e ***EPR*** (*Electronic*

---

<sup>1</sup><http://www.altova.com>.

Figura 32: Elemento raiz *ECGStudy*.

Nome	Natureza	Tipo	Uso	Descrição
<i>studyID</i>	attribute	int	required	Identificador único de um estudo de ECG de um paciente.
<i>studyDate</i>	attribute	date	required	Dia de aquisição do último registro obtido no estudo.
<i>studyTime</i>	attribute	time	required	Hora de aquisição do último registro obtido no estudo.
<i>studyLocation</i>	attribute	string	optional	Localização geográfica onde foi realizado o último registro obtido no estudo.
<i>alarm</i>	attribute	boolean	required	Indicador para o disparo de ações de urgência devido a alguma anomalia identificada durante o monitoramento.
<i>PatientInformation</i>	children	complex	required	Informações do paciente.
<i>Record</i>	children	complex	required	Registro de ECG.
<i>Comments</i>	children	string	optional	Comentários gerais sobre o estudo.

Tabela 5: Atributos e elementos filhos de um estudo de ECG.

*Patient Record*), responsáveis por encapsular dados pessoais do paciente e informações relativas ao seu Prontuário Eletrônico.

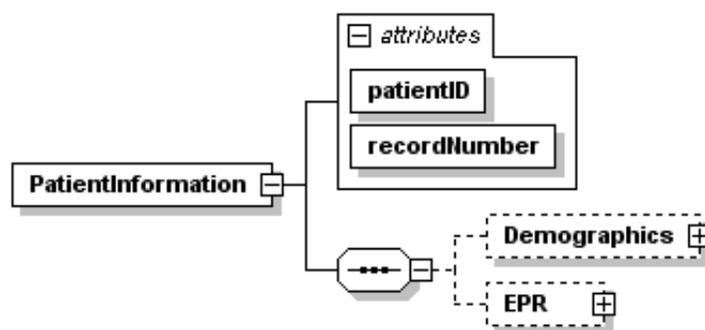
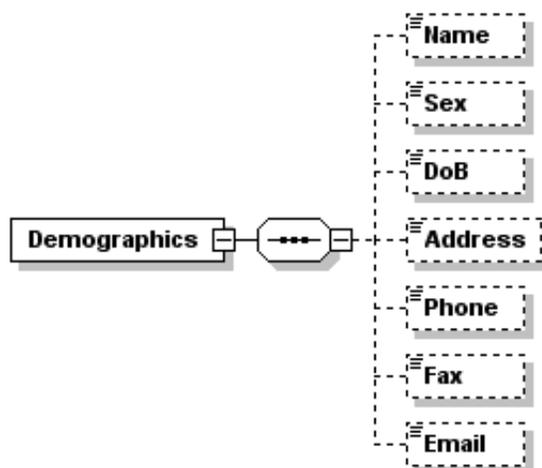


Figura 33: Elemento *PatientInformation*.

Nome	Natureza	Tipo	Uso	Descrição
<i>patientID</i>	attribute	int	required	Identificador único do paciente.
<i>recordNumber</i>	attribute	int	required	Número de registros de ECG já coletados e armazenados do paciente.
<i>Demographics</i>	children	complex	optional	Informações pessoais do paciente.
<i>EPR</i>	children	complex	optional	Abstração de um Prontuário Eletrônico do Paciente.

Tabela 6: Atributos e elementos filhos de *PatientInformation*.

*Demographics* (Figura 34 e Tabela 7), opcionalmente é destinado a integrar dados pessoais do paciente que permitem a sua identificação e contato, quando necessário. Ele possui como elementos filhos opcionais, *Name*, *Sex*, *DoB* (*Date of Birthday*), *Address*, *Phone*, *Fax*, e *Email*.

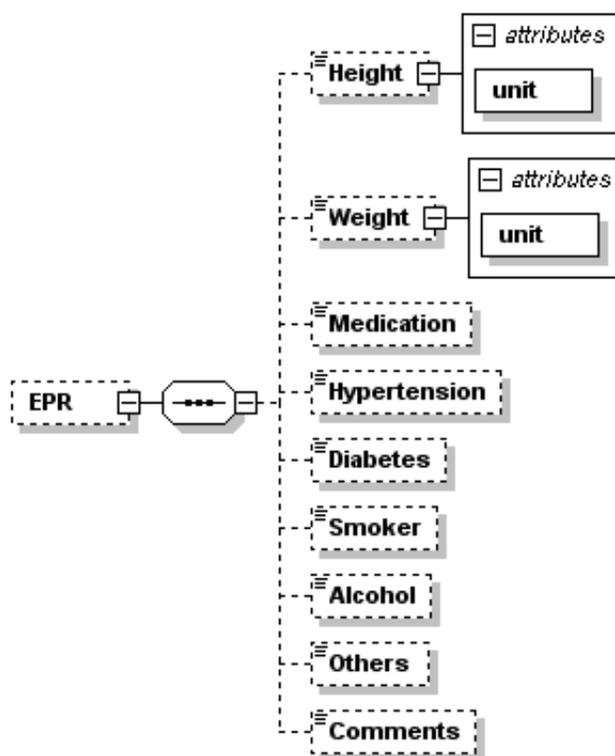
Figura 34: Elemento *Demographics*.

Nome	Natureza	Tipo	Uso	Descrição
<i>Name</i>	children	string	optional	Nome do paciente.
<i>Sex</i>	children	string	optional	Sexo do paciente.
<i>DoB</i>	children	date	optional	Data de nascimento.
<i>Address</i>	children	string	optional	Endereço do paciente.
<i>Phone</i>	children	string	optional	Telefone de contato.
<i>Fax</i>	children	string	optional	Fax.
<i>Email</i>	children	string	optional	e-mail de contato.

Tabela 7: Elementos que o compõem *Demographics*.

*EPR*, também opcional, representa sem grandes comprometimentos<sup>2</sup> uma abstração de Prontuário Eletrônico do Paciente (PEP). Assim, este elemento consiste em um histórico clínico do paciente composto dos elementos opcionais: *Height* e *Weight*, representando respectivamente altura e peso do paciente; *Medication*, para encapsular em texto livre os medicamentos que o paciente estiver usando; *Hypertension*, *Diabetes*, *Smoker*, e *Alcohol*, que indicam se o paciente sofre de hipertensão, ou de diabetes, se é fumante, ou se faz uso de álcool, respectivamente; um elemento *Others*, para permitir a inserção de outros dados clínicos; e por fim, o elemento *Comments*, um campo livre para comentários.

<sup>2</sup>A intenção aqui é somente permitir que, caso seja conveniente, possam ser incluídas algumas informações clínicas do paciente, sem pretender com isso alcançar uma completa representação de um PEP.

Figura 35: Elemento *EPR*.

Nome	Natureza	Tipo	Uso	Descrição
<i>Height</i>	children	float	optional	Altura do paciente.
<i>Weight</i>	children	float	optional	Peso do paciente.
<i>Medication</i>	children	string	optional	Medicação(ções) utilizada(s) frequentemente.
<i>Hypertension</i>	children	string	optional	Indica se o paciente é hipertenso.
<i>Diabetes</i>	children	string	optional	Indica se o paciente sofre de diabete.
<i>Smoker</i>	children	string	optional	Indica se o paciente é fumante.
<i>Alcohol</i>	children	string	optional	Indica se o paciente faz uso constante de álcool.
<i>Others</i>	children	string	optional	Alguma outra observação relevante.
<i>Comments</i>	children	string	optional	Comentários gerais sobre o histórico.

Tabela 8: Elementos filhos que compõem *EPR*.

Já o elemento *Record* (Figura 36 e Tabela 9), o mais importante da mensagem, representa um registro de ECG. Ele possui como atributos o identificador único do registro *recordID*, o sinalizador da ocorrência de um evento de urgência nesse registro *alarm*, e opcionalmente o atributo *investigatorID* identificando o profissional de saúde

responsável pelo monitoramento do paciente; e como seus elementos filhos: uma ou várias *RecordLeads*, uma ou várias *RecordAnnotations*, também uma ou várias *RecordMeasurements*, *RecordingDevice*, *Context*, e opcionalmente o elemento *Report*.

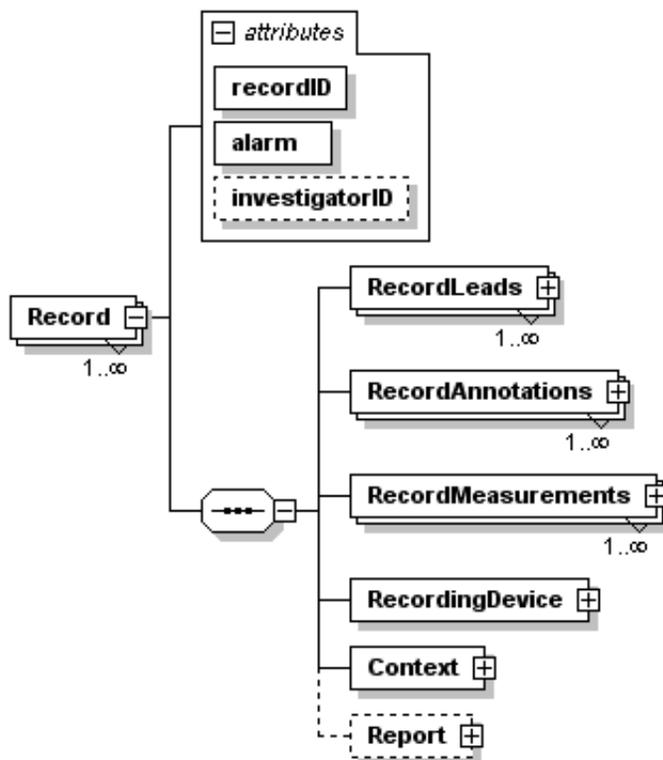
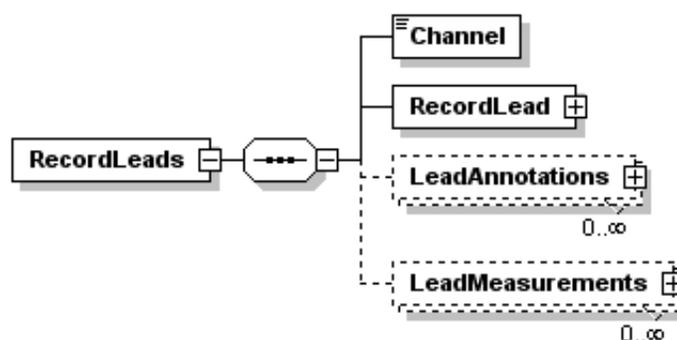


Figura 36: Elemento *Record*.

Expandindo um registro, as instâncias do elemento *RecordLeads* (Figura 37 e Tabela 10) contêm as múltiplas derivações que compõem o registro de ECG. Este elemento integra os elementos filhos *Channel*, que identifica a derivação, *RecordLead*, que contém o sinal de ECG propriamente dito da derivação, e *LeadAnnotations* e *LeadMeasurements*, que correspondem a anotações e medições opcionais relativas à derivação.

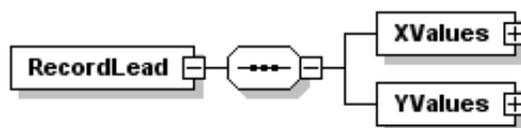
Nome	Natureza	Tipo	Uso	Descrição
<i>recordID</i>	attribute	int	required	Identificador único do registro.
<i>alarm</i>	attribute	boolean	required	Sinalizador da ocorrência de evento urgente no monitoramento do registro em questão.
<i>investigatorID</i>	attribute	string	optional	Identificador do profissional de saúde responsável pelo monitoramento.
<i>RecordLeads</i>	children	complex	required	Derivações obtidas nos diversos canais do dispositivo.
<i>RecordAnnotations</i>	children	complex	optional	Anotações relativas ao registro.
<i>RecordMeasurements</i>	children	complex	optional	Medições relativas ao registro.
<i>RecordingDevice</i>	children	complex	required	Informações relativas ao dispositivo de aquisição de ECG.
<i>Context</i>	children	complex	required	Contexto da aquisição do registro.
<i>Report</i>	children	complex	optional	Laudo relativo ao registro.

Tabela 9: Atributos e elementos filhos de *Record*.Figura 37: Elemento *RecordLeads*.

Nome	Natureza	Tipo	Uso	Descrição
<i>Channel</i>	children	string	required	Indica qual é a derivação (Ex: Lead III).
<i>RecordLead</i>	children	complex	required	Sinal obtido através de uma derivação.
<i>LeadAnnotations</i>	children	complex	optional	Anotações relativas a uma derivação.
<i>LeadMeasurements</i>	children	complex	optional	Medições relativas a uma derivação.

Tabela 10: Conteúdo de *RecordLeads*.

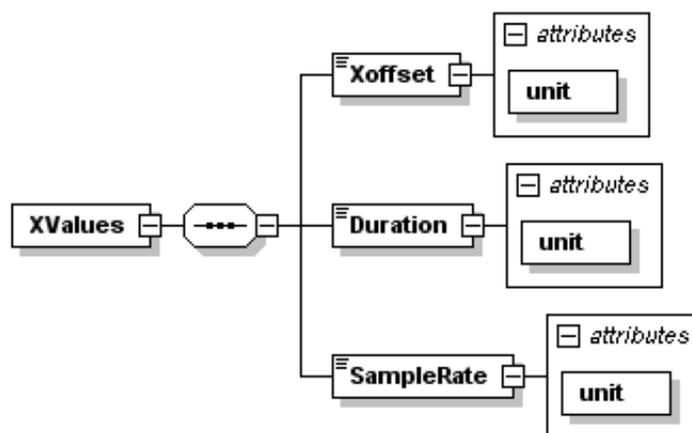
O sinal de ECG obtido de uma derivação é composto de amostras, pares-ordenado de valores dos eixos X e Y, representados no elemento *RecordLead* (Figura 38 e Tabela 11) através dos elementos *XValues* e *YValues*.

Figura 38: Elemento *RecordLead*.

Nome	Natureza	Tipo	Uso	Descrição
<i>XValues</i>	children	complex	required	Valores associados ao eixo X das amostras.
<i>YValues</i>	children	complex	required	Valores associados ao eixo Y das amostras.

Tabela 11: Conteúdo de *RecordLead*.

Entretanto, os valores de tempo associados ao eixo X do gráfico (elemento *XValues*) não precisam ser armazenados no documento XML, pois através do instante inicial (*Xoffset*), da duração do registro (*Duration*), e da frequência de amostragem (*SampleRate*), podem ser obtidos todos os valores do eixo X, uma vez que os pontos são periódicos, isto é, igualmente espaçados. Esses elementos são ilustrados na Figura 39 e na Tabela 12.

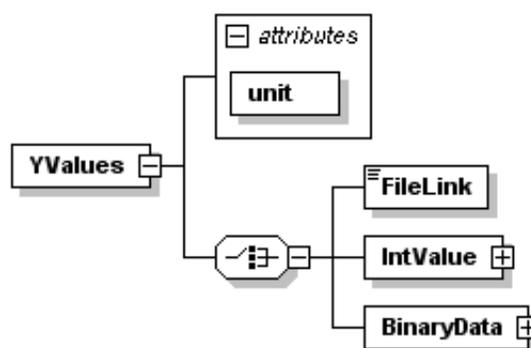
Figura 39: Elemento *XValues*.

Por sua vez, conforme a Figura 40 e a Tabela 13, os valores de amostra do eixo Y (*YValues*) têm de ser armazenados na mensagem, embora existam múltiplas opções para isso. Esses valores, cuja unidade é indicada no atributo *unit*, podem ser armazenados em um arquivo externo cujo *link* seria indicado em *FileLink*; ou em uma

Nome	Natureza	Tipo	Uso	Descrição
<i>Xoffset</i>	children	time	required	Instante de tempo do início da coleta.
<i>Duration</i>	children	time	required	Duração da coleta.
<i>SampleRate</i>	children	int	required	Frequência de amostragem (amostras/segundo).

Tabela 12: Elementos filhos de *XValues*.

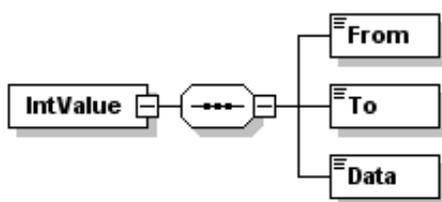
seqüência de inteiros no elemento *IntValue*; ou em uma codificação binária no elemento *BinaryData*. É importante ressaltar que somente uma dessas opções será utilizada.

Figura 40: Elemento *YValues*.

Nome	Natureza	Tipo	Uso	Descrição
<i>unit</i>	attribute	string	required	Unidade dos YValues (normalmente mV).
<i>FileLink</i>	children	children	required	Link para um arquivo externo à mensagem contendo os YValues (normalmente em formato .dat).
<i>IntValue</i>	children	complex	required	Contém os YValues em uma seqüência de inteiros.
<i>BinaryData</i>	children	complex	required	Contém os YValues em uma codificação binária.

Tabela 13: Atributo e opções de elemento filho de *YValues*.

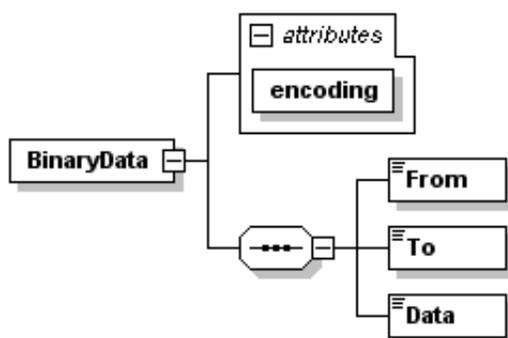
O armazenamento de uma seqüência de inteiros no elemento *YValue* (Figura 41 e Tabela 14) pode ser feito através dos elementos *From*, *To*, e *Data*, consistindo no instante de tempo correspondente ao primeiro valor, o instante de tempo do último valor, e a lista de inteiros com os valores coletados, respectivamente.

Figura 41: Elemento *IntValue*.

Nome	Natureza	Tipo	Uso	Descrição
<i>From</i>	children	time	required	Instante de tempo relativo à primeira amostra.
<i>To</i>	children	time	required	Instante de tempo relativo à última amostra.
<i>Data</i>	children	Sequence of Int	required	Seqüência de inteiros YValues.

Tabela 14: Conteúdo de *IntValue*.

Já o armazenamento em codificação binária (Figura 42 e Tabela 15) pode ser feito através do atributo *encoding* indicando o esquema de codificação utilizado, e dos elementos *From*, *To*, e *Data*, como em *IntValue*.

Figura 42: Elemento *BinaryData*.

Voltando a *RecordLead*, ainda podem existir, opcionalmente, os elementos *LeadAnnotations* e *LeadMeasurements*, que consistem em anotações e medições (vide seção 4.8) específicas de uma derivação, da mesma forma que, o elemento *Record*, com os elementos obrigatórios *RecordAnnotations* e *RecordMeasurements*. Tanto as anotações como as medições, sejam elas específicas de uma derivação ou relativas a um registro, possuem estruturas iguais, com a exceção de que nas anotações de uma derivação,

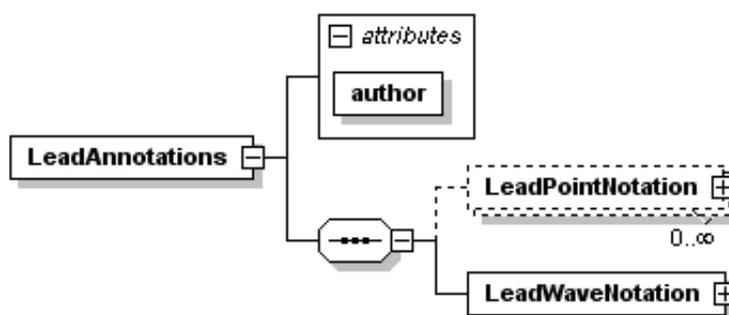
Nome	Natureza	Tipo	Uso	Descrição
<i>encoding</i>	attribute	base64Binary	required	Esquema de codificação binária utilizado.
<i>From</i>	children	time	required	Instante de tempo relativo à primeira amostra.
<i>To</i>	children	time	required	Instante de tempo relativo à última amostra.
<i>Data</i>	children	string	required	Código binário contendo os YValues das amostras.

Tabela 15: Atributo e elementos filhos de *BinaryData*.

alguns elementos obrigatórios nas anotações de um registro passam a ser opcionais. Isto se deve ao fato de que uma determinada onda elementar pode ser melhor observada em uma certa derivação e, por esta razão, é conveniente anotá-la somente nesta derivação e no registro como um todo, evitando anotá-la em outras derivações cuja visada não favorece uma observação precisa.

A diferença de uso entre anotações e medições relativas a uma derivação ou a um registro ocorre em função de que, as anotações e medições realmente fundamentais serem as que contemplam todo o registro, fruto da análise de todas as derivações. Por sua vez, anotações e medições relativas a uma derivação podem eventualmente serem úteis para, após se analisar as do registro, haver a possibilidade do médico saber que uma determinada anotação ou medição é oriunda de uma certa derivação, pois o fenômeno em questão foi melhor observado de acordo com esta.

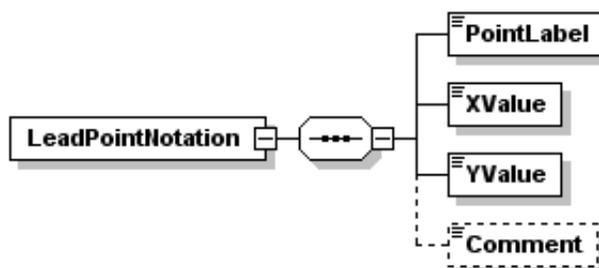
Neste sentido, *LeadAnnotations* (Figura 43 e Tabela 16) é composto de um atributo *author* identificando o autor das anotações, e dos elementos *LeadPointNotation*, e *LeadWaveNotation*.

Figura 43: Elemento *LeadAnnotations*.

Nome	Natureza	Tipo	Uso	Descrição
<i>author</i>	attribute	string	required	Autor das anotações.
<i>LeadPointNotation</i>	children	complex	optional	Anotação relativa a um ponto.
<i>LeadWaveNotation</i>	children	complex	required	Anotação relativa a uma forma elementar.

Tabela 16: Elemento *LeadAnnotations*.

*LeadPointNotation*, conforme a Figura 44 e a Tabela 17, consiste em zero ou várias anotações sobre amostras do sinal, normalmente pontos de referência (vide seção 4.8). Ele possui os elementos *PointLabel*, que indica o rótulo do ponto, *XValue* e *YValue*, que são os valores da amostra no gráfico, e um elemento *Comment*, opcional, para comentário.

Figura 44: Elemento *LeadPointNotation*.

Nome	Natureza	Tipo	Uso	Descrição
<i>PointLabel</i>	children	string	required	Nome do ponto de referência (Ex: P onset).
<i>XValue</i>	children	time	required	Valor da amostra no eixo X.
<i>YValue</i>	children	int	required	Valor da amostra no eixo Y.
<i>Comment</i>	children	string	optional	Comentário sobre a amostra (Ex: normal).

Tabela 17: Elemento *LeadPointNotation*.

E *LeadWaveNotation* (Figura 45 e Tabela 18), refere-se às anotações relativas às ondas elementares de um batimento. São elas: *Pwave*, *QRScomplex*, *Twave*, e *Uwave*; e um ou vários elementos *OtherWave*, para possíveis outras ondas ou formas. Conforme a discussão iniciada anteriormente, somente *QRScomplex* é obrigatório, pois trata-se da única forma elementar que pode ser bem observada de qualquer derivação.

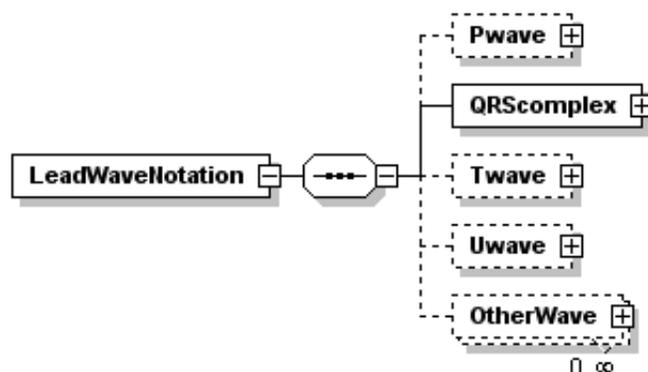


Figura 45: Elemento *LeadWaveNotation*.

Nome	Natureza	Tipo	Uso	Descrição
<i>Pwave</i>	children	complex	optional	Anotações da onda P.
<i>QRScomplex</i>	children	complex	required	Anotações do complexo QRS.
<i>Twave</i>	children	complex	optional	Anotações da onda T.
<i>Uwave</i>	children	complex	optional	Anotações da onda U.
<i>Otherwave</i>	children	complex	optional	Anotações de alguma outra forma.

Tabela 18: Conteúdo de *LeadWaveNotation*.

Em virtude dos elementos dispostos na Tabela 18 possuírem a mesma estrutura, somente será apresentado *Pwave*, conforme a Figura 46. Este e os outros elementos citados são compostos, de acordo com a Tabela 19, dos elementos *Onset*, *Peak*, *Offset*, *Interpretation*, e *Comment*.

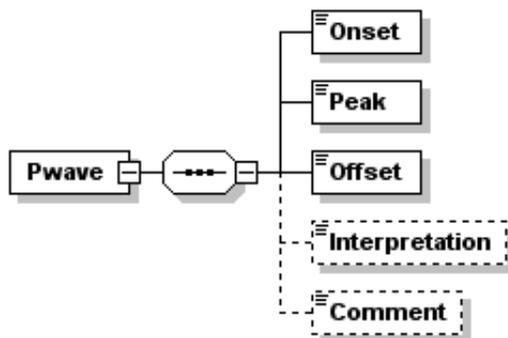


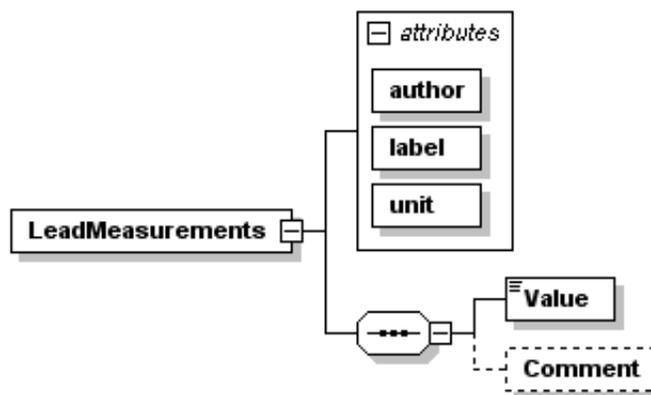
Figura 46: Estrutura comum aos elementos filhos de *LeadWaveNotation*.

E com relação às medições de uma derivação, *LeadMeasurements* (Figura 47 e Tabela 20), é composto dos atributos *author*, *label*, e *unit*, respectivamente indicando

Nome	Natureza	Tipo	Uso	Descrição
<i>Onset</i>	children	time	required	<i>XValue</i> do início da onda.
<i>Peak</i>	children	time	required	<i>XValue</i> do pico da onda.
<i>Offset</i>	children	time	required	<i>XValue</i> do fim da onda.
<i>Interpretation</i>	children	string	optional	Interpretação do estado da onda (Ex: normal).
<i>Comment</i>	children	string	optional	Comentário sobre a onda ou interpretação.

Tabela 19: Sub-elementos de *Pwave*, *QRScomplex*, *Twave*, *Uwave*, e *Otherwave*.

o autor, a descrição, e a unidade da medição; bem como dos elementos *Value*, que contém o valor da medição, e *Comment*, permitindo algum possível a respeito da mesma.

Figura 47: Elemento *LeadMeasurements*.

Nome	Natureza	Tipo	Uso	Descrição
<i>author</i>	attribute	string	required	Autor das medições.
<i>label</i>	attribute	string	required	Descrição da medição (Ex: duração da onda T).
<i>unit</i>	attribute	string	required	Unidade da medição.
<i>Value</i>	children	float	required	Valor da medição.
<i>Comment</i>	children	string	optional	Comentário sobre a medição (Ex: anormal).

Tabela 20: Atributos e elementos filhos de *LeadMeasurements*.

Conforme já discutido, tanto a estrutura de anotações quanto a de medições são iguais tratando-se do escopo de uma derivação ou de um registro, exceto pelo uso diferenciado dos componentes. Portanto, retornando ao elemento *Record* é necessário

destacar, de acordo com a Figura 48, a estrutura das anotações das ondas elementares relativas a um registro que, diferentemente das relativas a uma derivação (Figura 45), é completa, ou seja, todas as formas elementares são obrigatoriamente representadas. Isto é resultado de uma seleção que se aproveita das observações específicas de cada derivação e fornece uma visão global do sinal de ECG.

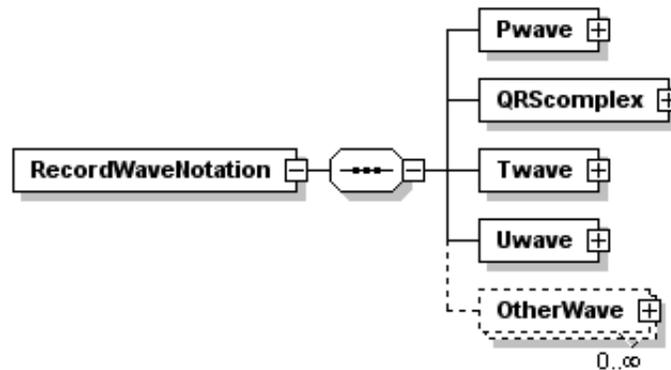


Figura 48: Elemento *RecordWaveNotation*.

As Figuras 49, 50, e 51, ilustram respectivamente os elementos *RecordAnnotations*, *RecordPointNotation*, e *RecordMeasurements*.

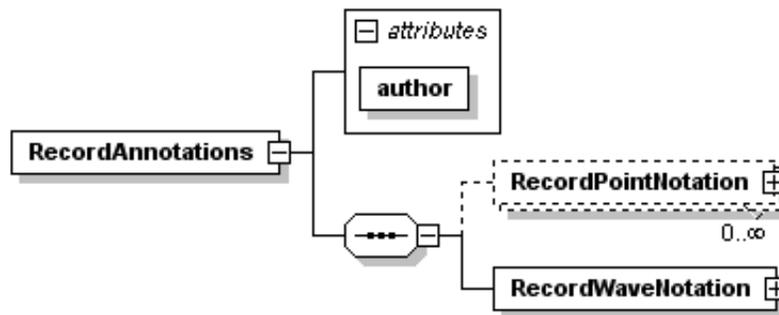


Figura 49: Elemento *RecordAnnotations*.

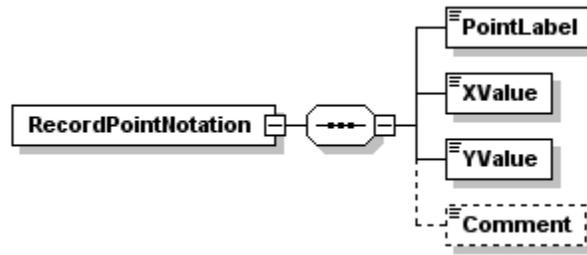


Figura 50: Elemento *RecordPointNotation*.

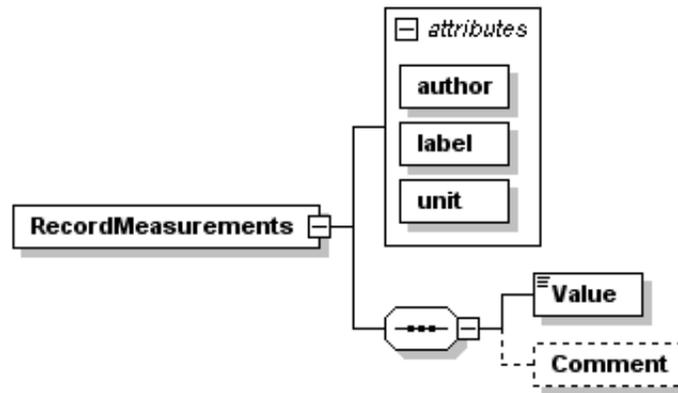
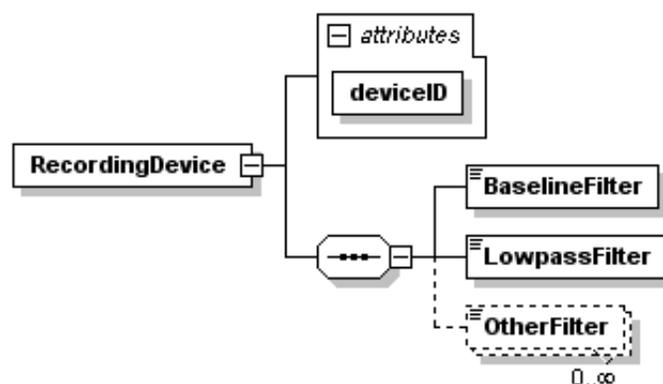


Figura 51: Elemento *RecordMeasurements*.

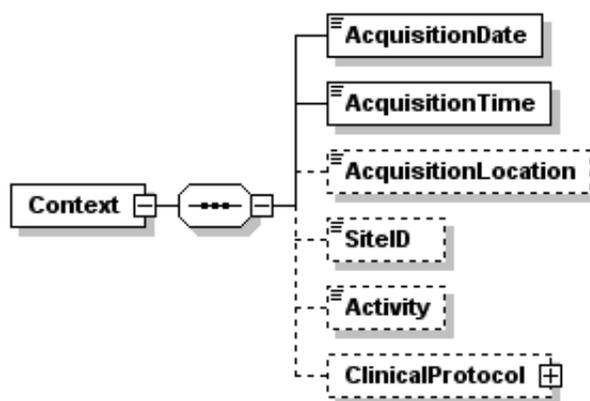
*Record* ainda possui outros elementos filhos; o elemento *RecordingDevice*, de acordo com a Figura 52 e a Tabela 21, é composto de informações a respeito do dispositivo portátil e de sua atuação no sinal. Dentre elas estão o atributo *deviceID*, identificando univocamente o dispositivo, e os elementos filhos *BaselineFilter*, *Low-passFilter*, e um ou vários *OtherFilter*, que consistem em, respectivamente, filtragem para eliminação de ruídos na linha de base do sinal, filtragem para eliminar os componentes de frequência do sinal acima de um valor máximo determinado, e alguma outra filtragem efetuada no sinal.

Figura 52: Elemento *RecordingDevice*.

Nome	Natureza	Tipo	Uso	Descrição
<i>deviceID</i>	attribute	int	required	Identificador único do dispositivo de aquisição, que permite a posteriori acessar informações sobre ele.
<i>BaselineFilter</i>	children	float	required	Filtragem que elimina oscilações (amostras inúteis) na linha de base do sinal.
<i>LowpassFilter</i>	children	float	required	Filtragem que limita o valor máximo de uma amostra pertencente ao sinal, cortando as que o excederem.
<i>OtherFilter</i>	children	string	optional	Alguma outra filtragem possível.

Tabela 21: Atributo e elementos filhos de *RecordingDevice*.

O elemento *Context* representa, conforme a Figura 53 e a Tabela 22, e sob um ponto de vista *context-aware*, o contexto do paciente durante a aquisição do registro, integrando os elementos: *AcquisitionData* e *AcquisitionTime*, que correspondem respectivamente ao dia e hora da aquisição do registro; os elementos opcionais *AcquisitionLocation* e *SiteID*, relativos à última localização geográfica do paciente durante a aquisição desse registro, e a uma descrição abstrata do local onde o registro foi obtido, respectivamente; o elemento *Activity*, também opcional, que contém uma descrição da atividade realizada pelo paciente durante a coleta do registro; e finalmente o elemento composto *ClinicalProtocol*.

Figura 53: Elemento *Context*.

Nome	Natureza	Tipo	Uso	Descrição
<i>AcquisitionDate</i>	children	date	required	Data de aquisição do registro.
<i>AcquisitionTime</i>	children	time	required	Hora de aquisição do registro.
<i>AcquisitionLocation</i>	children	string	optional	Útima localização geográfica obtida durante a coleta do registro.
<i>SiteID</i>	children	string	optional	Identificador do local de realização do monitoramento.
<i>Activity</i>	children	string	optional	Atividade realizada pelo paciente durante a coleta do registro.
<i>ClinicalProtocol</i>	children	complex	optional	Informações clínicas obtidas no momento de aquisição do registro.

Tabela 22: Conteúdo do elemento *Context*.

Pertencente ao elemento *Context*, o elemento opcional *ClinicalProtocol* (Figura 54 e Tabela 23) consiste em informações clínicas do paciente correspondentes ao momento da aquisição de um registro de ECG. São elas: *Medication* (drogas presentes no organismo do paciente), e opcionalmente *DiastolicBP* (pressão baixa), e *SystolicBP* (pressão alta).

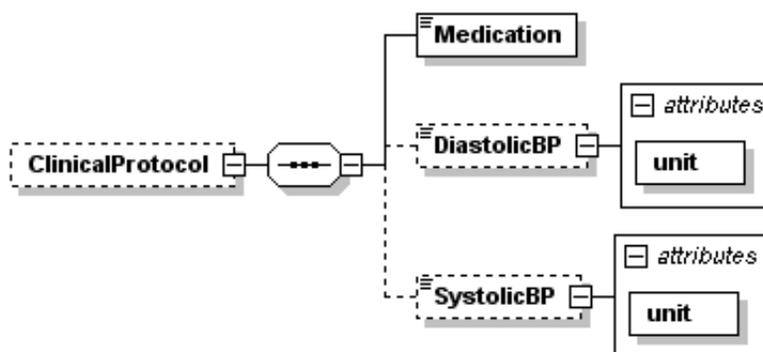
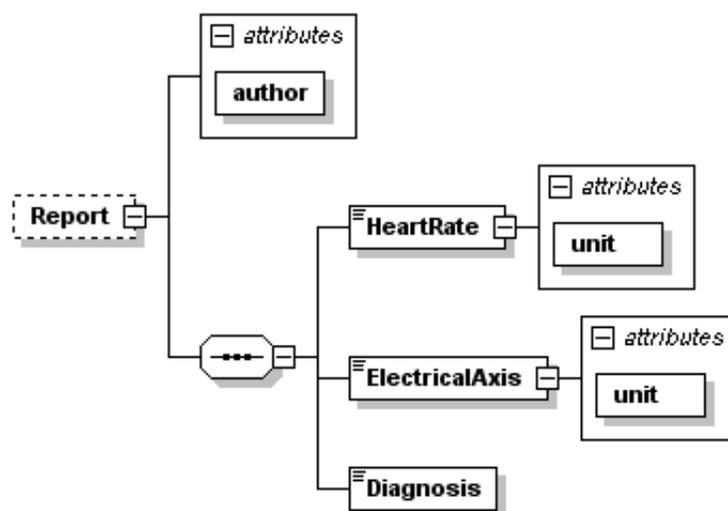


Figura 54: Elemento *ClinicalProtocol*.

Nome	Natureza	Tipo	Uso	Descrição
<i>Medication</i>	children	string	optional	Medicação em uso no momento da coleta do registro.
<i>DiastolicBP</i>	children	int	optional	Valor da pressão baixa do paciente.
<i>SystolicBP</i>	children	int	optional	Valor da pressão alta do paciente.

Tabela 23: Elemento *ClinicalProtocol*.

Por fim, também compõe um registro (*Record*) o elemento opcional *Report* que, conforme a Figura 55 e a Tabela 24, corresponde ao laudo de avaliação desse registro. Ele possui o atributo *author* indicando o médico ou o sistema que gerou o laudo a respeito do registro; e os elementos filhos *HeartRate* (frequência cardíaca), *ElectricalAxis* (ângulo do eixo elétrico do coração), e *Diagnosis* (avaliação propriamente dita).

Figura 55: Elemento *Report*.

Nome	Natureza	Tipo	Uso	Descrição
<i>author</i>	attribute	string	required	Autor do laudo.
<i>HeartRate</i>	children	int	required	Frequência cardíaca derivada do sinal do registro.
<i>ElectricalAxis</i>	children	int	required	Eixo elétrico do coração (Ex: 60 graus).
<i>Diagnosis</i>	children	string	required	Diagnóstico relativo ao registro.

Tabela 24: Elemento *Report*.

A Figura 56 apresenta uma visão global da estrutura hierárquica do *ecgAware*.



## 5.2 Comentários Gerais e Avaliação do *ecgAware*

O formato de mensagem *ecgAware* de fato foi influenciado pelos padrões analisados no Capítulo 4. No entanto, o *ecgAware* é marcado por uma característica diferenciada: explorar o aspecto contextual do sinal de ECG e de suas informações complementares. Isto é possível sobretudo em função de dois fatores:

1. Através de um sistema de análise automática do sinal [21], é feita uma segmentação e classificação do ECG por meio de um sistema a base de regras, capaz de detectar eventos que sinalizam características de forte semântica para a observação de cardiologistas;
2. A partir do processamento realizado no sistema de análise automática, a plataforma *Infraware* realiza um gerenciamento eficiente das informações obtidas, capaz de sinalizar às aplicações médicas sensíveis ao contexto eventos de interesse ao usuário final (profissional da área médica), bem como cruzar dados oriundos de outros sistemas<sup>3</sup> e eventualmente disparar ações necessárias.

Desde que o *ecgAware* é aplicado entre esses dois sistemas justamente para fazer a ligação entre eles, é necessário que ele possua uma representatividade tal capaz de explorar ao máximo as funcionalidades de ambos. Neste sentido, o *ecgAware* consiste em uma abordagem original que permite a importante distinção entre adquirir a informação médica, e utilizá-la de maneira eficiente e sensível ao contexto do paciente (Figura 57). Mais precisamente, o *ecgAware* é de fato, uma interface flexível entre essas duas tarefas distintas que, através do encapsulamento da informação médica, isola do ambiente de trabalho de profissionais de saúde a complexidade relativa à aquisição de sinais biomédicos, repassando-os com semântica refinada.

Uma outra característica deste formato, é que o mesmo priva pela generalidade, a fim de manter possibilidades de uso em cenários futuros que podem envolver a aquisição e transmissão de outros sinais vitais além do ECG. Isto pode ser feito adotando-se a mesma estratégia e se aproveitando da flexibilidade da tecnologia XML, que permite

---

<sup>3</sup>Um possível exemplo é, a partir da sinalização de alarme feita pelo sistema de análise automática, o módulo Interpretador de Contexto da *Infraware* combinar informações, por exemplo, com um sistema de identificação de rotas de trânsito para indicar o caminho mais rápido do domicílio do paciente até um hospital.

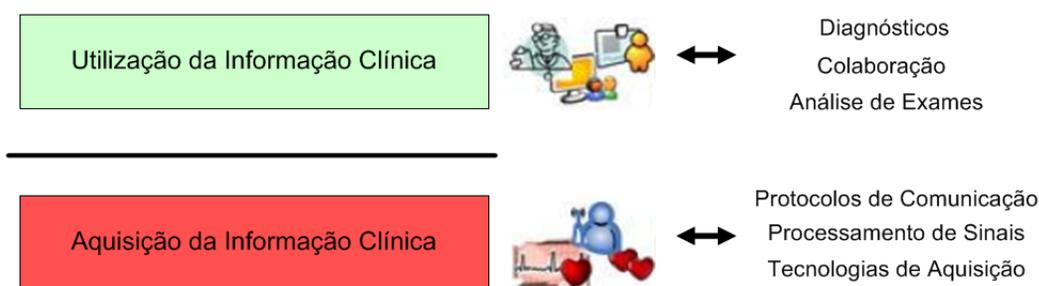


Figura 57: Distinção entre aquisição e utilização da informação médica.

a integração de um *XML Schema* dentro de outro. Dessa forma, é possível utilizar as estruturas genéricas do *ecgAware* necessárias para transmissão de qualquer sinal vital, e incorporar a ele *XML Schemas* definidos a partir de um estudo do domínio de cada sinal vital incluído.

Levando tudo isso em consideração, já no elemento raiz da mensagem existem atributos destinados a conter o meta-conteúdo do estudo de ECG do paciente, para que tão logo as informações desses atributos forem acessadas, seja possível distinguir o grau de prioridade que deve ser dado ao estudo, eventualmente disparando alguma ação na plataforma Infracore que utilizaria dados como o último instante e a última localização do monitoramento do paciente.

Além disso, foi preservada a possibilidade de utilizar informações clínicas históricas do paciente através do EPR (*Electronic Patient Record*), que podem ser úteis especialmente em cenários futuros a serem explorados pela Infracore.

Com relação ao registro de ECG, o elemento mais importante da mensagem, pode-se destacar seu elemento filho *Context*, que fornece o contexto em que foi obtido o registro; e também toda a análise realizada pelo sistema de análise automática de ECG desenvolvido na tese de doutorado de Andreão [21], que detecta e classifica com precisão as formas elementares de cada batimento cardíaco identificando possíveis distúrbios na atividade cardíaca do paciente, e até mesmo emite um laudo relativo ao registro, que é útil ao médico responsável.

Finalmente, no que se refere aos requisitos da Tabela 4, exceto pelos requisitos de performance, todos os outros foram perfeitamente atendidos na especificação *ecgAware*. No Apêndice A é apresentado na forma textual um exemplo de *XML Document* (mod-

elo) estruturado conforme o *ecgAware*.

## 6 *Cenários de Uso e Requisitos do ECG-Wrapper*

Neste capítulo serão especificados os requisitos do *ECG-Wrapper*, que constituem uma instanciação dos requisitos gerais de um *Context-Wrapper* qualquer, descritos na Tabela 1. A estratégia escolhida para tal propósito é conhecida na área de Engenharia de Software como *Scenario-based Design* (SBD), que vem se difundindo cada vez mais na engenharia de sistemas computacionais, principalmente por proporcionar uma rápida comunicação sobre as possibilidades de uso e preocupações entre diferentes *stakeholders* (partes interessadas) [77]. SBD consiste em algumas técnicas, por exemplo, descrição narrativa, que permitem que o uso de um sistema ainda a ser criado seja concretamente descrito no momento inici

al do processo de desenvolvimento [77]. Sendo assim, de acordo com as necessidades dos principais cenários de uso do *ECG-Wrapper*, e levando em conta as restrições de tecnologia concernentes a cada cenário, os requisitos aqui discutidos servirão de base para o projeto do sistema. Para o restante deste trabalho, o termo sistema sem nenhuma identificação específica referir-se-á única e exclusivamente ao programa *ECG-Wrapper*.

Na seção 6.1, são apresentados os outros elementos envolvidos no uso do *ECG-Wrapper*, bem como possibilidades de configuração desses elementos. Partindo para os cenários de uso propriamente ditos, são descritos nas seções 6.2, 6.3, e 6.4, os três principais cenários explorados pelo sistema: *(i)* Monitoramento domiciliar, *(ii)* Unidade móvel de emergência (ambulância), e *(iii)* Monitoramento em ambiente externo, respectivamente, que conforme [6, 24], são adotados no projeto TeleCardio. Após isso, na seção 6.5 são descritos casos de uso e requisitos do sistema, e finalmente na seção 6.6 é concluído o capítulo.

## 6.1 Elementos e Configurações

Os cenários de uso do sistema podem ser classificados quanto a: (i) a localização - ou nível de ubiqüidade - do paciente a ser monitorado pelo sistema, e (ii) a duração do monitoramento da atividade cardíaca do paciente; determinando assim, diferentes configurações dos recursos de monitoramento, que se adequam à forma de aquisição, processamento e transmissão dos dados, bem como à forma de interação do usuário.

No entanto, a todos os cenários de uso é comum a presença de uma entidade aqui definida, a **UMS - Unidade Móvel de Sensoriamento**, composta dos seguintes elementos de hardware:

- **dispositivo portátil**, que integra um holter, um botão de alarme e um transmissor RF; este elemento faz a aquisição dos dados de ECG oriundos do paciente através dos sensores, e os transmite ao sistema através de algum mecanismo de comunicação. Além disso, este dispositivo pode, também, incorporar em hardware um sistema de análise automática do sinal ECG.
- **computador remoto**, que pode ser um desktop, um laptop, um PDA (*Personal Digital Assistant*), ou até mesmo um telefone celular programável; a escolha mais apropriada deste elemento envolve questões como a capacidade de processamento requerida, o tamanho deste dispositivo, e a duração de bateria compatível com o cenário de uso no que se refere ao tipo de ambiente (ex.: interno, externo) e à duração do monitoramento do paciente (monitoramentos de 24h exigem mais recursos).

E de software:

- **sistema de análise de ECG**, é responsável por receber os dados ainda brutos e, através de algoritmos baseados em modelos ocultos de Markov (HMM), realizar um processamento capaz de, segmentar, classificar, e detectar batimentos prematuros e episódios de isquemia no sinal ECG [3]; este elemento, em uma configuração ideal, estaria alocado no próprio dispositivo portátil implementado em hardware, mas em função da complexidade desta alternativa, há uma forte tendência de que este seja incorporado ao *ECG-Wrapper* no computador remoto.

- **sistema *ECG-Wrapper***, objeto deste trabalho, que é executado no computador remoto, sendo responsável por receber os dados de ECG do dispositivo portátil e, dependendo da configuração processá-los fazendo uma análise automática do sinal, formatá-los e entregá-los adequadamente à plataforma Infracore. É importante ressaltar que, a eventual alocação do sistema de análise automática de ECG fora do sistema *ECG-Wrapper* não contradiz o requisito R1.3 da Tabela 1, pois este é um requisito genérico de um *Context-Wrapper*, que pode ser implementado de uma forma alternativa em um caso específico.

A Figura 58 apresenta uma possível configuração da UMS, composta por um PDA com o software *ECG-Wrapper* integrado (à esquerda), e o dispositivo portátil composto de uma *sensor box* com sistema de análise automática integrado e um botão de alarme na parte de cima (à direita acima).



Figura 58: Ilustração de uma possível configuração da UMS [12].

Conforme explícito na apresentação da UMS, esta possui diversas possibilidades de configuração. Os critérios para a escolha adequada de uma configuração dependem do cenário no qual o sistema será aplicado, mais especificamente do grau de mobilidade, do tipo e da duração do monitoramento, e das restrições de tecnologia existentes no cenário.

## 6.2 Cenário de Monitoramento Domiciliar

No monitoramento domiciliar, o paciente mantém-se em seu domicílio ou em alguma Unidade de Saúde equipado com a UMS, para a qual recebe instruções de uso, sendo continuamente monitorado pelo sistema. A Figura 59 oferece uma visualização deste cenário.



Figura 59: Monitoramento de ECG em domicílio ou Unidade de Saúde.

A dinâmica de uso deste cenário consiste em: a partir da aquisição do sinal ECG do paciente, o dispositivo portátil o envia ao computador remoto, no qual é feito um tratamento desse sinal, que posteriormente é enviado já formatado e com semântica apurada, à Central de Monitoramento onde funciona a plataforma Infraware; e então, a qualquer momento o profissional de saúde pode subscrever-se ao sistema e acessar as informações do paciente.

Neste cenário o paciente possui alguma mobilidade, mas por questões de custo, autonomia de bateria, e levando-se em conta que ele encontra-se em um ambiente confinado, o paciente utiliza um holter acoplado a um dispositivo de comunicação sem fio de curto alcance e baixo consumo de energia. O processamento e pré-armazenamento do sinal ficam a cargo do computador remoto, que de acordo com a configuração mais adequada para este cenário, consiste em um desktop ou laptop, devido à maior capaci-

dade de armazenamento, uma vez que se trata de 24 horas ininterruptas de gravação de ECG.

### 6.3 Cenário da Unidade móvel de emergência (ambulância)

Este cenário de uso exige uma maior mobilidade em relação ao domiciliar, e por isso sugere fortemente a utilização de um PDA ou de um telefone celular programável como computador remoto, por ambos proporcionarem tal mobilidade, facilitando assim, a organização interna da ambulância. Na Figura 60 é mostrada uma ambulância equipada com o sistema de monitoramento de ECG, e mais especificamente, com uma Wireless PAN (*Personal Area Network*) como enlace de comunicação sem fio entre o dispositivo portátil e o computador remoto, permitindo assim, ainda mais mobilidade neste cenário.



Figura 60: Ambulância equipada para Telecardiologia [78].

Neste tipo de situação, de cunho emergencial, o paciente é monitorado e os seus sinais cardíacos são transmitidos por algum enlace de comunicação sem fio, que normalmente utiliza a infra-estrutura de comunicação móvel celular, à Central de Monitoramento enquanto a ambulância se dirige à Unidade de Saúde mais próxima habilitada

a atender o caso. De acordo com o estado do paciente, diversas ações podem ser disparadas pelo sistema, como a solicitação à Unidade de Saúde selecionada de serviços médicos apropriados já contactando o plano de saúde do paciente; o envio de mensagens SMS ao médico particular do paciente ou à própria Unidade de Saúde; e também é possível utilizar outros serviços através de uma integração runtime com outros sistemas, como o de obtenção de rotas de trânsitos mais apropriadas até a Unidade de Saúde [6, 24].

## 6.4 Cenário de Monitoramento em Ambiente Externo

O cenário de monitoramento em ambiente outdoor permite, diferentemente dos outros já abordados, o monitoramento do paciente durante uma maior variedade de atividades, especialmente durante a realização de atividades físicas, o que possibilita diagnósticos antecipados de cardiopatias.

Neste caso, conforme exibido na Figura 61, a UMS é acoplada à roupa do paciente de tal forma que não atrapalha consideravelmente sua movimentação; para isso é necessário que o computador remoto seja um PDA (Figura 62a) ou um telefone celular programável.

Já existe na área de Telemedicina o projeto de uma jaqueta denominada *Vital Jacket*, capaz de incorporar sensores de forma a monitorar e transmitir os sinais vitais via algum enlace de comunicação sem fio. A Figura 62b mostra um protótipo desenvolvido pelo grupo de pesquisa do Laboratório de Sistemas de Informação na Área da Saúde (SIAS Lab) do Instituto de Engenharia Eletrônica e Telemática de Aveiro (IEETA) em Portugal.

É possível observar que, em cada cenário citado, existe uma demanda diferente por portabilidade, o que limita recursos de tecnologia como largura de banda e capacidade de processamento; quanto maior for a demanda por portabilidade, em geral, mais limitados serão esses recursos. Por isso essa questão trata-se de uma importante decisão de projeto envolvendo as necessidades de cada cenário e a disponibilidade de recursos computacionais e de infra-estrutura.



Figura 61: Monitoramento de sinais cardíacos enquanto o paciente realiza atividade física [79].

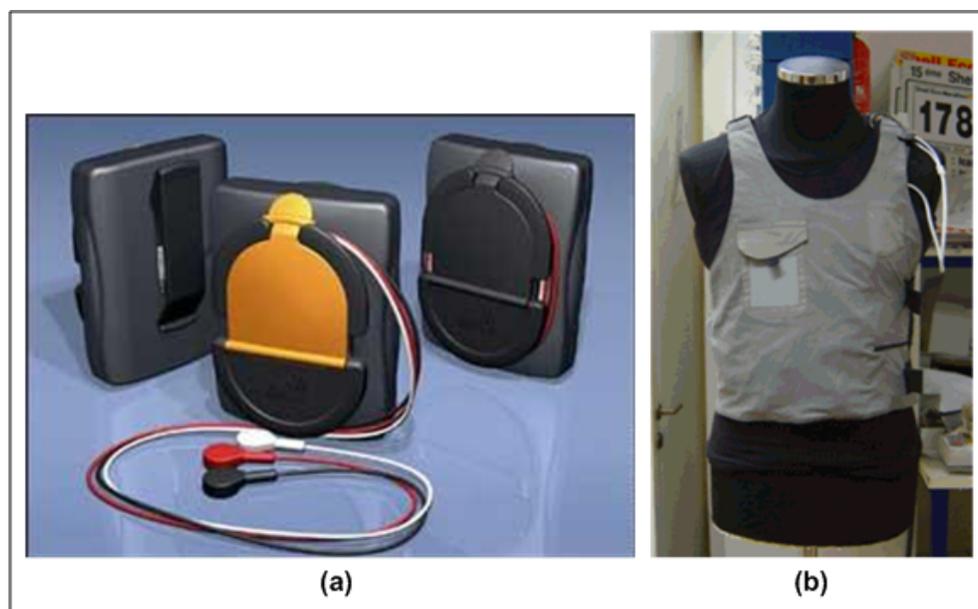


Figura 62: (a) PDA com holter incorporado. (b) *Vital Jacket* [80].

## 6.5 Casos de Uso e Requisitos do Sistema

Conforme [81], a relevância de um sistema computacional é motivada pelas suas interações com atores externos. Neste sentido, um caso de uso (*use case*) de um sistema

é uma interação típica entre este sistema e um ator - que pode ser um usuário humano, um dispositivo, ou outro sistema computacional. Assim, casos de uso capturam funções visíveis aos atores e buscam com isso, atingir uma meta do usuário. Sendo assim, a descrição dos casos de uso serve como guia, tanto para a definição dos requisitos funcionais do sistema, quanto para posterior avaliação desses requisitos.

De posse dos cenários a serem explorados, é possível determinar os casos de uso de interesse no escopo deste texto. Entretanto, é necessário, antes disso, definir os atores do sistema que integram os diferentes cenários de uso e os principais pontos de interação.

### 6.5.1 Atores

Na perspectiva externa ao *ECG-Wrapper*, pode-se identificar quatro atores que lidam com o sistema regularmente. Vale lembrar que, *(i)* esses atores são papéis, portanto não têm de ser uma pessoa específica; e *(ii)* sistemas externos que interagem com o sistema em questão também são denominados atores do sistema.

- **Paciente:** é a pessoa, possivelmente possuidora de alguma cardiopatia, que será monitorada pelo sistema. No momento da instalação da UMS, o paciente recebe algumas instruções de uso para que ele possa executar operações básicas, eventualmente trocar mensagens com o seu médico, bem como acionar o botão de alarme se necessário.
- **Agente de saúde:** é o profissional que lida diretamente com o paciente - dependendo do cenário, um enfermeiro, um médico, ou até mesmo um personal trainer -, seja no momento da instalação da UMS, ou de uma visita domiciliar, ou de um atendimento emergencial com ambulância; este indivíduo deve ter um conhecimento mínimo sobre monitoramento de sinais vitais, deve estar apto a manipular a UMS, e também, inserir dados de identificação do paciente e de Prontuário Eletrônico no sistema.
- **Dispositivo portátil:** este dispositivo interage com o sistema para comunicar os dados adquiridos, e reportar algum evento de alarme acionado pelo paciente.

- **Infraware:** é a plataforma de serviços *context-aware*, para onde é enviado todo o output do *ECG-Wrapper*. Existem três tipos de interação entre a Infraware e o sistema: (i) *request-response*, em que a plataforma requisita uma atualização e o sistema a entrega; (ii) *time-driven*, em que a plataforma informa uma determinada frequência com que deseja receber as atualizações; e (iii) *event-driven*, em que o sistema envia uma atualização com maior prioridade reportando à Infraware a ocorrência de um evento.

### 6.5.2 Descrição de Casos de Uso

Diante dos três cenários discutidos nas seções 6.2, 6.3, e 6.4, é possível abstrair, em alto nível, os casos de uso mostrados no diagrama da Figura 63 e descritos na Tabela 25.

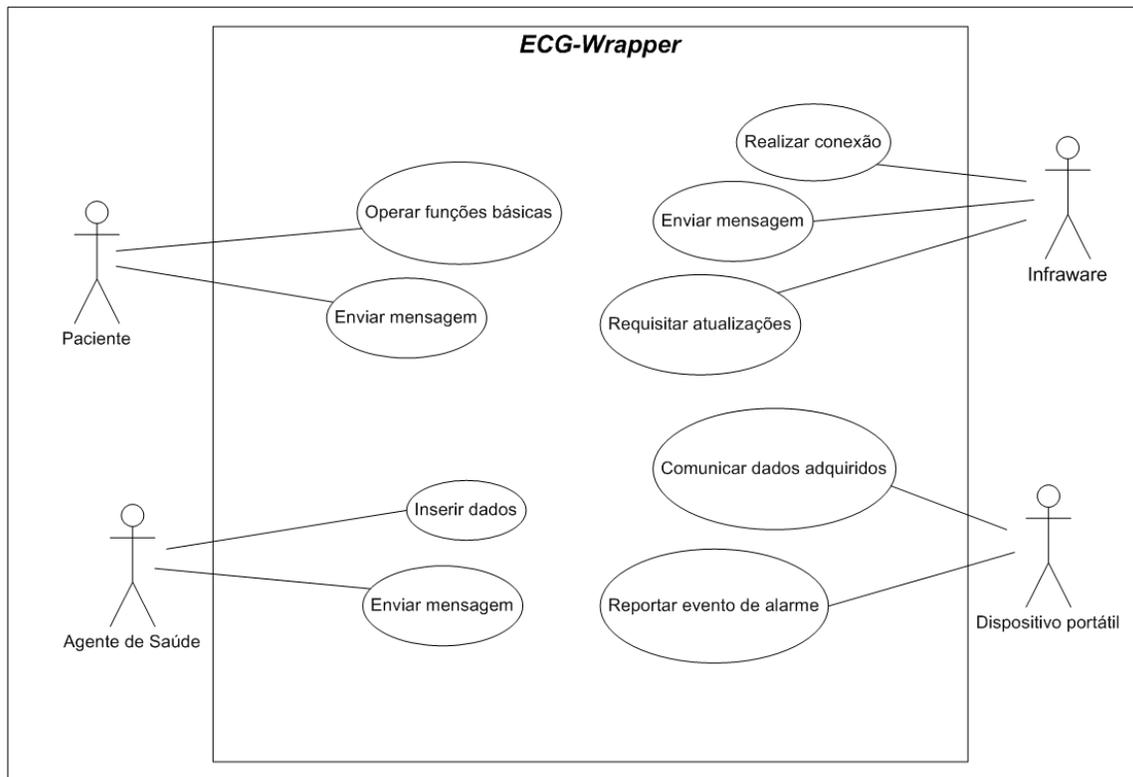


Figura 63: Diagrama de Casos de Uso do *ECG-Wrapper*.

U#	Caso de Uso	Ator	Descrição
1	Operar funções básicas	Paciente	O paciente opera funções básicas como iniciar uma sessão de registro de ECG, ou visualizar uma mensagem recebida.
2	Enviar mensagem	Paciente	O paciente ou acompanhante envia uma breve mensagem ao médico responsável através da GUI.
3	Inserir dados	Agente de Saúde	O agente de saúde, ao ativar o sistema, insere através da GUI dados de identificação do paciente, eventualmente informações de Prontuário Eletrônico, e também dados relativos às sessões de monitoramento (por exemplo, a pressão sanguínea no instante de início de uma sessão).
4	Enviar mensagem	Agente de Saúde	Se necessário, o agente de saúde também pode enviar mensagens ao médico do paciente informando alguma ocorrência relevante.
5	Realizar conexão	Infraware	Uma aplicação cliente da plataforma Infraware solicita uma informação deste provedor de contexto ( <i>ECG-Wrapper</i> ), e então a plataforma tenta abrir uma conexão com o sistema.
6	Requisitar atualização	Infraware	Uma aplicação cliente da plataforma Infraware solicita uma informação deste provedor de contexto ( <i>ECG-Wrapper</i> ), e então a plataforma a repassa ao sistema.
7	Enviar mensagem	Infraware	O médico do paciente, online na Infraware, envia uma mensagem ao seu paciente, que é repassada pelo CoDIMS ao sistema.
8	Comunicar dados adquiridos	Dispositivo Portátil	O dispositivo portátil transmite os dados coletados pelos sensores através de um protocolo de comunicação sem fio.
9	Reportar evento de alarme	Dispositivo Portátil	Ao sentir algum distúrbio grave de saúde, o paciente aciona o botão de alarme no dispositivo portátil; em seguida, imediatamente o dispositivo comunica-se com o sistema sinalizando o evento.

Tabela 25: Descrições dos casos de uso.

### 6.5.3 Descrição de Requisitos

Com base nos casos de uso definidos na Tabela 25, e por um mapeamento direto, são finalmente obtidos os requisitos do sistema, dispostos na Tabela 26.

R#	Descrição	Origem <sup>1</sup>
3.1	O sistema precisa ser capaz de transferir para a plataforma Infracare, em tempo real, e a qualquer outro momento que seja requisitado, os dados de ECG do paciente monitorado.	U5, U6, U8
3.2	É responsabilidade do sistema garantir que, uma eventual anomalia nos batimentos cardíacos do paciente detectada pelo sistema de análise automática seja imediatamente reportada à plataforma Infracare.	U8
3.3	O sistema deve encaminhar imediatamente à plataforma Infracare a devida sinalização quando o paciente acionar o botão de alarme.	U9
3.4	O sistema deve permitir que o agente de saúde, no momento da instalação da UMS, possa inserir dados de identificação do paciente, eventualmente informações de Prontuário Eletrônico do Paciente, bem como dados relativos às sessões de monitoramento.	U3
3.5	O sistema deve possibilitar a ambos, paciente e médico responsável, trocarem mensagens e notificações entre si, assim como ao agente de saúde, particularmente no cenário que envolve a ambulância, possa enviar mensagens de urgência à Unidade de Saúde para a qual se dirige.	U2, U4, U7
3.6	É necessário que o sistema mantenha-se alimentado pelo dispositivo portátil com os dados de ECG, mas saiba lidar com uma possível perda de contato (exceção) com ele.	U8

Tabela 26: Requisitos do *ECG-Wrapper*.

## 6.6 Conclusão do Capítulo

Através de uma perspectiva global do *ECG-Wrapper*, combinando o domínio horizontal de encapsulamento de contexto e a sua aplicação no domínio vertical de Telecardiologia, este capítulo forneceu um ponto de vista geral de como o sistema pode ser usado, gerando assim, requisitos necessários para guiar o projeto do *ECG-Wrapper* de acordo com os seus principais cenários de uso.

No próximo capítulo será apresentado tal projeto, que visa aplicar as soluções computacionais mais bem avaliadas durante a pesquisa, no desenvolvimento desse sistema.

## 7 *Projeto do ECG-Wrapper*

Este capítulo apresenta as soluções computacionais avaliadas como mais adequadas, dentre todas possíveis, para atender as demandas criadas na análise do sistema realizada no capítulo anterior. Assim, o projeto do *ECG-Wrapper*, sob ambas as perspectivas externa e interna, busca atender de forma objetiva aos requisitos expressos nas tabelas 1, 4, e 26, de tal maneira a permitir um mapeamento direto entre esses e as estruturas do sistema. Com isso, será possível ao final deste trabalho uma avaliação apontando quais requisitos foram atendidos.

O capítulo está estruturado da seguinte forma: a seção 7.1 apresenta o projeto do sistema em um alto nível de abstração; a partir da seção 7.2 até a 7.5 são descritas cada camada presente na arquitetura do *ECG-Wrapper*; na seção 7.6 é discutido o projeto da GUI do sistema; e finalmente, na seção 7.7 é feita uma conclusão do capítulo.

### 7.1 Projeto de Alto Nível

Inicialmente, nesta primeira seção relacionada ao projeto do sistema *ECG-Wrapper* propriamente dito, ele é abordado em uma perspectiva externa e de um ponto de vista geral, a fim de fornecer a visão, o fluxo de dados, e a arquitetura do sistema, para posteriormente cada componente ser apresentado mais detalhadamente.

#### 7.1.1 Visão Geral do Sistema

Não obstante terem sido citadas várias opções de configuração da UMS, bem como terem sido apresentados três diferentes cenários de uso do sistema (vide Capítulo 6), devido a limitações de complexidade e tecnologia esta versão do projeto irá focar-se no cenário de monitoramento domiciliar com a seguinte configuração da UMS: (*i*) dispos-

itivo portátil composto de holter, botão de alarme e transmissor RF; (ii) computador remoto instanciado por um laptop; e (iii) sistema de análise automática de ECG integrando o *ECG-Wrapper*, alocado no computador remoto. Esta visão geral do sistema é ilustrada na Figura 64.

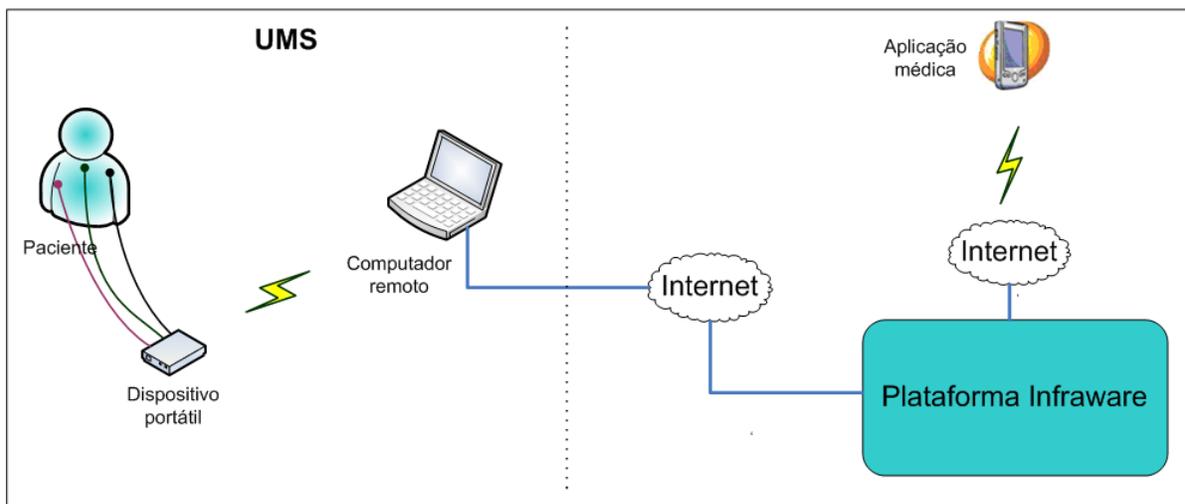


Figura 64: Visão geral do sistema.

O fluxo dos dados dá-se da seguinte forma: o dispositivo portátil coleta o ECG do paciente e o envia através do enlace de comunicação sem fio, de acordo com algum protocolo de comunicação (vide seção 7.2), ao computador remoto. Este, ao receber os dados de ECG, faz uma análise automática no sinal, e um encapsulamento de ambos, ECG e sua interpretação, de acordo com o formato para representação de ECG especificado no Capítulo 5; e então, disponibiliza a mensagem à plataforma Infraware de acordo com os três modelos de entrega da informação contextual (seção 7.5.2).

### 7.1.2 Arquitetura do *ECG-Wrapper*

De acordo com os critérios discutidos em [82], o estilo arquitetônico escolhido neste trabalho como mais adequado para o sistema é o da divisão em camadas. A escolha baseia-se no fato de que, cada componente, possui uma função bem definida correspondente a um determinado nível de abstração, e interage apenas com seus dois componentes adjacentes.

A arquitetura do sistema (Figura 65) é composta, do ponto de vista bottom-up,

das camadas de: (i) Comunicação *Wireless PAN* com o dispositivo portátil; (ii) Interpretação, que consiste no sistema de análise automática de ECG; (iii) Tradução, que formata a informação adequadamente para a Plataforma Infracore, e (iv) Comunicação *Web Service* com a Infracore.

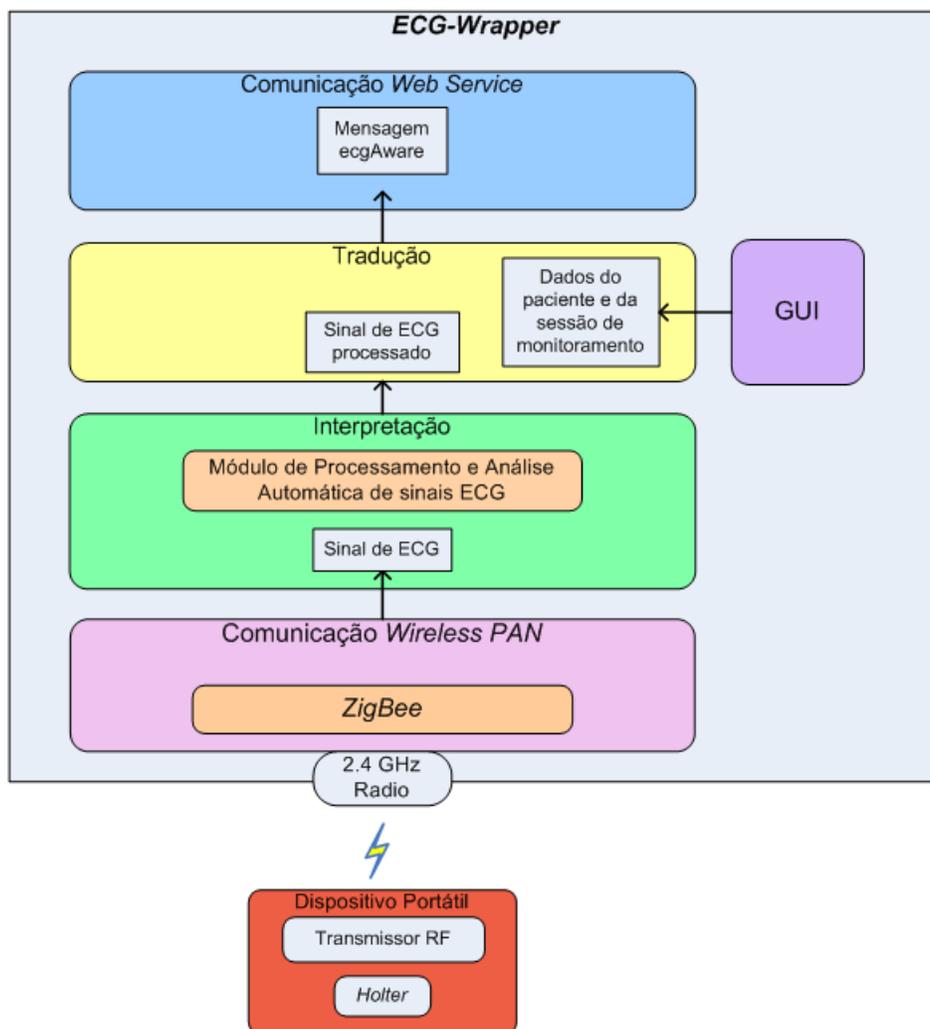


Figura 65: Arquitetura do sistema.

A seguir serão descritas separadamente as quatro camadas, bem como o módulo GUI, remetendo-se aos requisitos que motivam a existência de cada módulo do sistema.

## 7.2 Camada de Comunicação Wireless PAN

De acordo com o requisito R1.1 da Tabela 1, entre os sensores e o *Context-Wrapper*, deve haver algum mecanismo de comunicação, que pode variar diante de cada cenário e configuração da UMS. Por exemplo, se um PDA compuser a UMS instanciando o computador remoto, é possível que o *Holter* seja incorporado a ele de maneira *hardwired*, evitando assim, a necessidade de um *link* de comunicação e reduzindo o consumo de energia.

No entanto, mais especificamente no caso do cenário e configuração adotados neste projeto (vide seção 7.1.1), é necessário que haja entre o *laptop* (computador remoto) onde estará em execução o programa *ECG-Wrapper* e o *Holter* (dispositivo portátil) de aquisição dos sinais cardíacos, um mecanismo de comunicação sem fio, cujo raio da área de abrangência não precisa ser maior do que cerca de 10 metros, ou seja, uma PAN (*Personal Area Network*).

Desta forma, foi adotado para tal propósito o protocolo proprietário de comunicação *Wireless PAN ZigBee*. É importante ressaltar que, esse módulo do sistema faz parte do escopo de trabalho do grupo de pesquisa do projeto TeleCardio que integra o Departamento de Engenharia Elétrica da UFES. Portanto, será apresentado aqui somente uma visão geral do protocolo *ZigBee*.

### 7.2.1 ZigBee Overview

O *ZigBee* é um padrão de comunicação sem fio por radiofrequência entre computadores e dispositivos relacionados<sup>1</sup>, criado para resolver as demandas de monitoramento e controle remotos, e também de aplicações de redes de sensores [83]. Ratificado em Dezembro de 2004, e baseando-se no padrão IEEE 802.15.4, mais especificamente IEEE 802.15.4-2003 *Low Rate WPAN*, *ZigBee* faz uso da ampla especificação desse padrão no que se refere às camadas física e de enlace de radiofrequência, adicionando especificações relativas à camada de rede, a segurança, e à camada de aplicação [84, 85]. A Figura 66 apresenta a arquitetura do *ZigBee*.

Neste sentido, o *ZigBee* é um dos padrões globais de protocolo de comunicação

---

<sup>1</sup>Dentre esses dispositivos pode-se incluir telefones, PDAs, sensores, e controles remotos, localizados em poucos metros de distância um do outro [83].

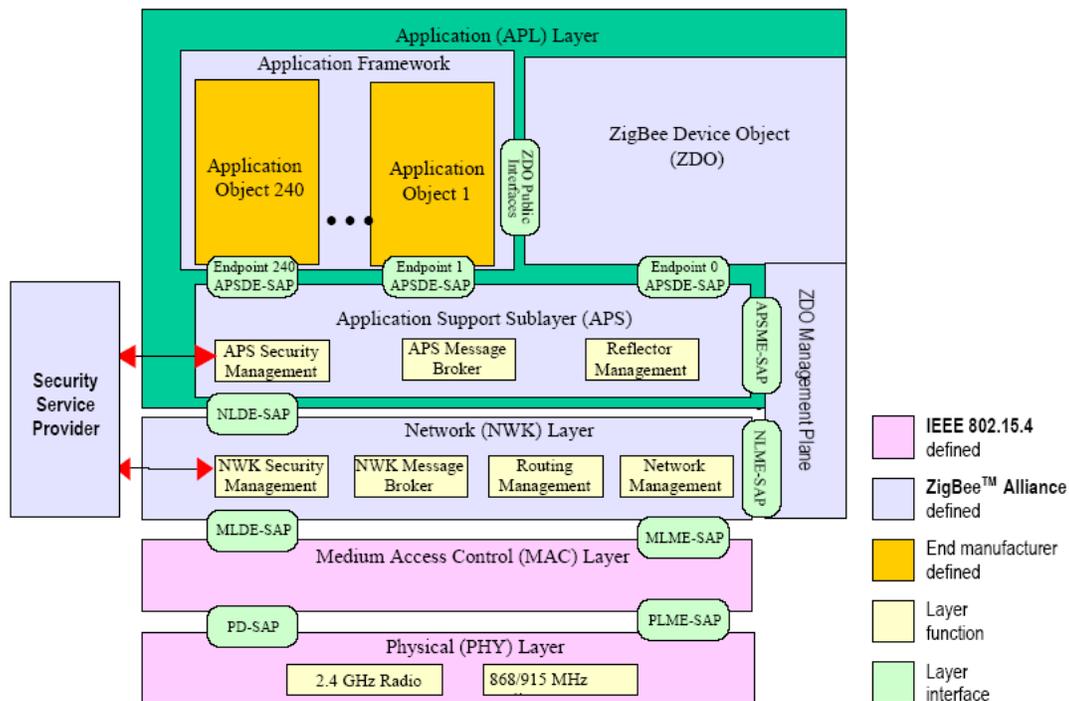


Figura 66: Arquitetura ZigBee [85].

formulados por uma task force do IEEE 802.15 *Working Group*; quarto da série, *WPAN Low Rate/ZigBee* é o mais recente deles, e provê um conjunto de especificações para dispositivos que possuem baixas taxas de transmissão de dados, consomem muito pouca energia e, conseqüentemente, são caracterizados por uma longa duração de bateria. Enquanto que, outros padrões IEEE 802.15 como *Bluetooth* e *IrDA*, são destinados a aplicações de altas taxas de transmissão como de voz, vídeo, e LAN [83].

Dessa forma, é importante distinguir as diferentes demandas de, (i) uma rede sem fio de dados tradicional, e (ii) uma rede de sensores sem fio (RSSF) para controle e sensoriamento.

Na primeira classe de redes, estas são projetadas para interconectar computadores, PDAs, impressoras, *Access Points*, etc, em que grandes quantidades de dados são enviadas em ambas as direções. Ou seja, a ênfase nesse caso é na velocidade de transmissão, e o projeto e a evolução dos padrões IEEE 802.11 é um bom exemplo para comprovar essa busca contínua por incrementar a velocidade de transmissão a fim de permitir rápidos *downloads*, por exemplo, de música e vídeo, para os usuários finais [84].

Entretanto, redes sem fio para aplicações industriais de controle e sensoriamento,

acima de tudo, têm de ser confiáveis, adaptáveis, e escaláveis, já que sensores enviam somente poucos bits de dados por segundo ou por minuto, provendo informações como temperatura, pressão, localização geográfica, etc, em que raramente são necessárias taxas de transmissão de 11 ou 54 Mbps. Dessa forma, as principais características de desempenho em uma rede de sensores, são: (i) *latência*, ou seja, o tempo desde o envio de uma mensagem até o recebimento da mesma; (ii) *precisão* de uma informação obtida pelos sensores; (iii) *tolerância à falhas*, isto é, a capacidade da rede de suportar, por exemplo, a perda de um sensor e adaptar-se à nova topologia; (iv) *escalabilidade*, que é capacidade de suportar muitos usuários consultando informações; e (v) *exposição dos sensores*, uma medida de quão bem colocada está uma determinada rede diante de um objeto ou ambiente observado [86].

Posto isso, também é importante um comparativo com relação às topologias de rede nos padrões IEEE 802.15. No *Bluetooth*, a unidade básica da sua arquitetura é uma *piconet*, que consiste em um nó mestre e até sete escravos ativos, além de até 255 inativos. Os nós escravos podem estar situados em uma distância de até 10 metros do nó mestre, formando uma rede com topologia em estrela, embora possam existir muitas *piconets* no mesmo local, conectadas por um nó ponte formando uma *scatternet* [86], conforme ilustra a Figura 67.

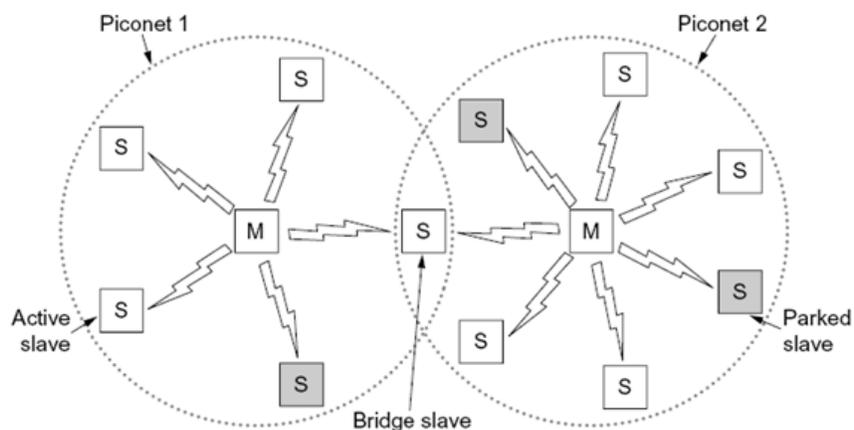


Figura 67: Topologia estrela no *Bluetooth*: *piconets* e *scatternets* [87].

Nesse tipo de topologia *point-to-multipoint*, somente a estação base ou *Access Point*, controla toda a comunicação na rede e, conseqüentemente, todos os sinais convergem para um único nó. Em função disto, a confiabilidade da rede é determinada pela qualidade do link de RF entre o nó raiz e os nós comuns. Em aplicações de controle e

sensoriamento, normalmente é difícil haver uma localização para o nó raiz que atenda perfeitamente cada nó escravo, pois movendo o nó raiz para melhorar a comunicação com um nó, prejudica a comunicação com um outro. E ainda que seja possível alocar múltiplos *Access Points* em um mesmo local, o próprio custo desta opção acaba por depreciá-la [84].

Por sua vez, a camada de rede do *ZigBee* suporta além da topologia em estrela (*star*), a topologia *mesh* [85]. E existem três diferentes tipos de dispositivos *ZigBee*:

- *ZigBee Coordinator (ZC)*: usado como nó *root* da rede, o ZC é o dispositivo de mais recursos, e pode servir de ponte para outras redes. Cada rede é composta de somente um ZC, que é capaz de armazenar informação sobre a rede, inclusive chaves de segurança.
- *ZigBee Router (ZR)*: usados como roteadores, ZRs são nós intermediários que passam os dados para os outros nós da rede.
- *ZigBee End Device (ZED)*: dispositivos mais simples da rede, os ZEDs somente possuem a funcionalidade suficiente para interagir com nós roteadores ou com o coordenador. Eles não conseguem se comunicar com outros ZEDs, demandando assim, pouca memória e, conseqüentemente, são de fabricação de mais baixo custo comparados com ZRs ou ZCs.

Na topologia *star*, a rede é controlada pelo nó ZC, que é responsável por iniciar e manter a participação dos outros nós da rede, os ZEDs, que se comunicam somente com o ZC [85]. Já na topologia *mesh*, o ZC é responsável por iniciar a rede e escolher os parâmetros de chave, mas a rede pode ser estendida através de roteadores, os ZRs. Utilizando uma estratégia hierárquica de roteamento, os ZRs repassam mensagens aos seus nós roteadores vizinhos, ou as entregam em definitivo ao ZED destinatário da mensagem. Assim, seja qual for o seu caminho percorrido, um pacote de dados sempre alcança seu destino final [84]. A Figura 68 ilustra os dois tipos de topologia mencionados.

É possível destacar como vantagens da topologia *mesh*: (i) assim como a Internet, uma rede *mesh* oferece múltiplos caminhos redundantes; (ii) se ocorrer uma falha de um nó, as mensagens são automaticamente roteadas para caminhos alternativos, e o *ZigBee* garante ampla QoS com confirmação de mensagem, checagem de erros, e mudança de

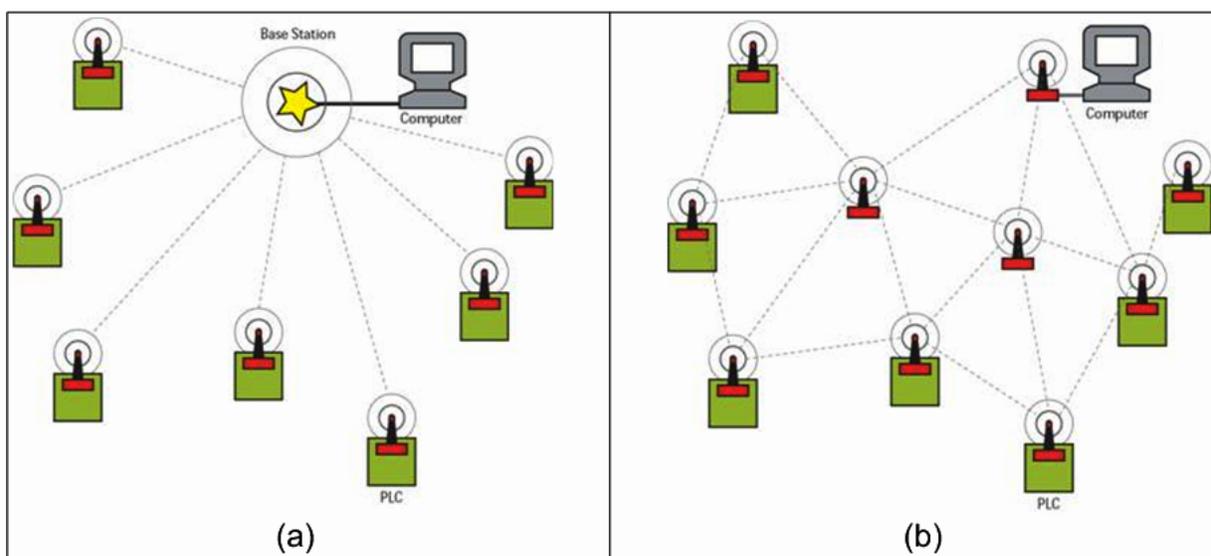


Figura 68: Topologias (a) *star*, e (b) *mesh*, no *ZigBee*.

frequência para evitar interferência; (iii) a distância entre os nós é reduzida, o que aumenta bastante a qualidade do link entre eles, tornando esses links mais confiáveis sem incrementar a energia gasta pelos nós na transmissão dos dados; e por fim, (iv) uma rede *mesh* provê excelente escalabilidade, pois desde que a operação da rede não depende um nó central, adicionar novos nós ou *gateways* não causa grandes impactos, permitindo assim, a presença de milhares de nós na rede [84].

Levando tudo isso em consideração, pode-se concluir que, dentre os padrões IEEE 802.15, *ZigBee* é o mais apropriado para aplicação em redes de sensores sem fio (RSSF). E apesar da transmissão de ECG *Holter-Laptop* a princípio não envolver uma rede de sensores, os critérios relacionados a controle e monitoramento mencionados nesta seção por si só já justificam a escolha do *ZigBee* em detrimento do *Bluetooth*. Além disso, não bastasse a possibilidade de incluir-se um dispositivo de aquisição de localização como GPS, cenários futuros podem envolver além do ECG, a transmissão de outros sinais vitais coletados por outros dispositivos sensores que, possivelmente, serão interconectados em rede.

Assim como o padrão *Wi-Fi*, o *ZigBee* também possui uma associação de empresas que trabalham em conjunto para promovê-lo. A *ZigBee Alliance* [88] visa a fabricação de dispositivos confiáveis, de baixo custo, de baixo consumo de energia, de comunicação sem fio, monitoramento e controle, baseando-se no padrão aberto global IEEE 802.15.4-

2003. Os fabricantes fornecem chips, kits de avaliação, módulos, pilhas de protocolos, e ferramentas de desenvolvimento, oferecendo pacotes completos aos consumidores para aplicação em redes domésticas, comerciais, e industriais.

## 7.3 Camada de Interpretação

A camada de interpretação é decorrente, na perspectiva de um *Context-Wrapper* qualquer, do requisito R1.3; enquanto que, na perspectiva do *ECG-Wrapper*, do requisito R3.2. Este módulo do sistema não será implementado neste trabalho, pois pertence ao escopo do grupo de pesquisa do Departamento de Engenharia Elétrica da UFES no projeto TeleCardio.

O sistema de análise do ECG, conforme citado na seção 6.1, é responsável por receber os dados ainda brutos e, através de algoritmos baseados em Modelos Ocultos de Markov (HMM), realizar um processamento capaz de segmentar, classificar, e detectar batimentos prematuros e episódios de isquemia no sinal ECG [3].

Conforme exibido na Figura 69, os algoritmos utilizados no módulo de Processamento de Sinal ECG estão organizados em uma estrutura composta por três camadas [21].

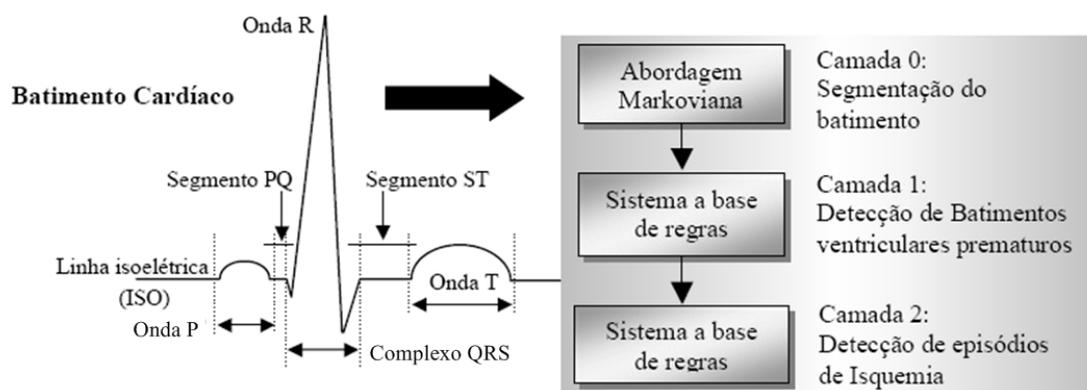


Figura 69: Estrutura do algoritmo de classificação do sinal ECG [22].

A camada 0 diz respeito ao processamento mais básico do sinal ECG, que consiste em identificar em um registro de ECG os eventos elétricos elementares relevantes para a interpretação do traçado do sinal. Primeiramente, o sinal ECG é processado pela transformada *wavelet* Chapéu Mexicano, que visa realçar a informação útil do

sinal em detrimento do ruído. A escolha da função Chapéu Mexicano deve-se ao seu melhor desempenho entre as *wavelets* contínuas no processamento de ECG, além de possuir uma alta resolução temporal e uma morfologia mais próxima do complexo QRS normal [21]. Em seguida, uma abordagem estatística baseada nos modelos ocultos de Markov (HMM) realiza no sinal uma segmentação de acordo com as formas de onda elementares do batimento cardíaco. Essa segmentação procura associar as amostras do sinal processado ao HMM de cada forma elementar para maximizar a verossimilhança [22].

A partir da saída gerada pela camada 0, na camada 1 é feita uma classificação dos batimentos através de um sistema a base de regras heurísticas que busca traços característicos no sinal segmentado seguindo a abordagem do cardiologista. A classificação implementada pelo sistema é a de batimentos ventriculares prematuros (ESV), que são caracterizados por um complexo QRS largo e contração ventricular prematura [22].

E finalmente, na camada 2, o sistema realiza uma análise de longo termo para identificar eventos anormais persistentes. Atualmente, o sistema faz a identificação de episódios de isquemia cardíaca, monitorando o desvio da amplitude do segmento ST ao longo do tempo. Com isso, é possível identificar precocemente os episódios de isquemia que normalmente precedem o infarto do miocárdio. Isso acontece quando o desvio ultrapassa um limiar de 0,1 mV durante um período de no mínimo 30 segundos [22].

Assim, quando o módulo de Processamento de Sinais detecta algum evento anormal, este é notificado ao módulo de Formatação do *ECG-Wrapper* a fim de ser imediatamente informado à Infracare para que as devidas ações sejam desencadeadas. Dessa maneira, os eventos de risco para a saúde do paciente podem ser antecipados e possivelmente as complicações decorrentes dos mesmos evitadas.

## 7.4 Camada de Tradução

Esta camada é responsável por, conforme a discussão sobre metadados e esquemas na seção 3.3 e a partir dos requisitos R1.4 e R1.5 da Tabela 1, *(i)* receber os dados obtidos pelos sensores e encapsulá-los de acordo com o formato específico de representação desses dados; e *(ii)* posteriormente converter esse formato específico do domínio vertical

em questão para o formato global adotado pelo middleware de acesso e integração de dados da plataforma Infraware.

No *ECG-Wrapper*, o formato no qual os dados contextuais são encapsulados é a linguagem de marcação *ecgAware*, especificada no Capítulo 5. Desta forma, neste módulo do sistema os dados são recebidos junto ao módulo de interpretação já processados, e são então, embutidos no modelo (documento XML) gerado a partir do meta-modelo *ecgAware*. Quando a informação é requisitada pela Infraware, esse documento XML é convertido para o modelo entendido pelo CoDIMS e enviado.

Conforme a especificação do *ecgAware*, o encapsulamento que gera o documento XML é feito não só sobre os dados contextuais oriundos dos sensores, mas também sobre as informações de classificação e análise efetuadas no módulo de análise automática. Esse encapsulamento é feito através de um mapeamento de objetos Java e seus atributos para elementos e atributos XML (*Java-XML Binding*). Com relação à conversão exigida pela plataforma Infraware do modelo no formato *ecgAware* para o formato global entendido pelo CoDIMS, esta pode ser feita de maneiras distintas. Primeiramente, vale ressaltar que tal formato global consiste no formato relacional atributo-valor, e esse tipo de conversão XML-Relacional/Relacional-XML foi objeto de [89], outro trabalho desenvolvido no LPRM.

Duas formas de se atender a esse requisito são: (i) antes de serem gravadas as informações em um documento XML no formato *ecgAware*, gravá-las através de um procedimento *Java-XML Binding* semelhante, só que no formato relacional atributo-valor; ou (ii) já a partir do documento XML no formato *ecgAware* efetuar uma conversão automática para o formato relacional atributo-valor através de alguma ferramenta computacional<sup>2</sup>.

Devido a restrições de tempo, não foi possível neste trabalho realizar o projeto da interface do *ECG-Wrapper* com a plataforma Infraware, contemplando questões como a conversão de modelos supracitada, a especificação de um protocolo de comunicação com máquina de estados finitos, etc. Portanto, essas tarefas ficarão a cargo de trabalhos futuros.

---

<sup>2</sup>A própria ferramenta *XML Spy* da Altova utilizada neste trabalho para definir o *XML Schema ecgAware* realiza esse tipo de conversão.

## 7.5 Camada de Comunicação *Web Service*

A camada de comunicação *Web Service* é consequência natural de que o sistema objeto deste trabalho é um sistema distribuído, que tem como responsabilidade básica disponibilizar a sua saída na Web para seu consumidor de dados direto, a Infracore. Portanto, essa camada tem especial relevância no *ECG-Wrapper*, atendendo de forma geral ao requisito R1.6 da Tabela 1, e de maneira específica aos requisitos R3.1, R3.2, e R3.3 da Tabela 26.

Desafortunadamente, e de acordo com a discussão feita no final da seção anterior, o projeto da interface do *ECG-Wrapper* com a plataforma Infracore exigiria um esforço além do escopo deste trabalho na especificação de um protocolo de comunicação com máquina de estados finitos para definir precisamente o comportamento deste módulo de comunicação do sistema ao longo de sua execução. Desta forma, o projeto aqui apresentado visa oferecer uma visão geral de *Web Services*, que foi necessária para a implementação do protótipo apresentado no Capítulo 8, bem como apresentar os modelos de entrega da informação contextual previstos inicialmente para as interações entre um *Context-Wrapper* e a Infracore.

### 7.5.1 *Web Services*

Conforme a definição apresentada em [90], *Web Services* são programas cujo objetivo é permitir a comunicação entre diferentes aplicações através de uma rede, utilizando para isto mensagens e protocolos baseados na linguagem XML e em protocolos da Web tais como HTTP.

*Web Services* (WS) proporcionam um fraco acoplamento entre as aplicações que interagem através desse tipo de serviço, permitindo entre elas uma comunicação que independe de plataformas e linguagens de programação, desde que estas aplicações sigam os protocolos de WS [90]. Assim, o grau de interdependência entre esses sistemas é minimizado, de forma que modificações internas em cada um deles não são percebidas pelo outro. Essa característica é ilustrada na Figura 70, em que alterações nos módulos B e C da composição interna do sistema X (Figura 70a), gerando a nova versão X (Figura 70b), não afetam a sua interação com o sistema Y.

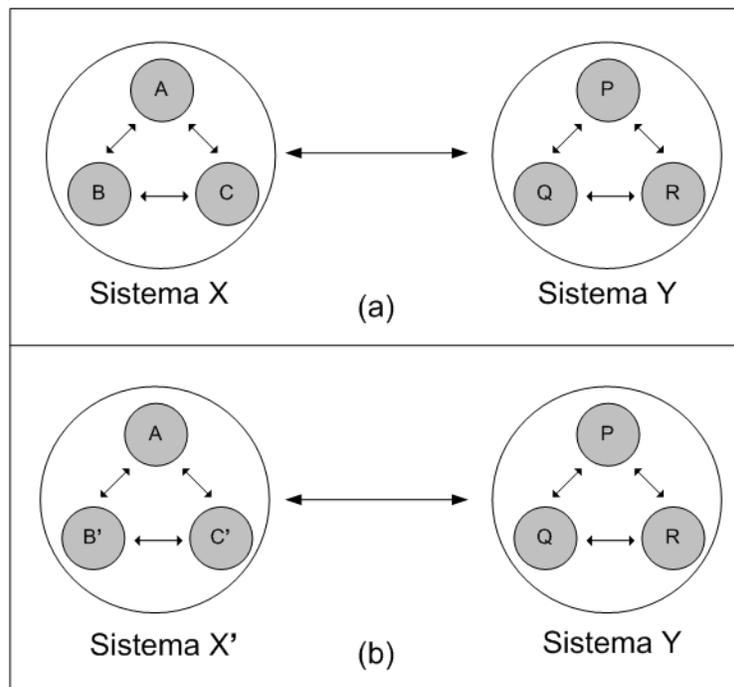


Figura 70: Fraco acoplamento entre sistemas que interagem através de *Web Services*.

Dentre os procedimentos efetuados por *Web Services*, é possível destacar [91]: (i) auto-descrição de suas funcionalidades; (ii) publicação das descrições dos serviços oferecidos; (iii) descoberta de funcionalidades; e (iv) transferência de dados com outros WS.

A dinâmica de funcionamento dos WS baseia-se na arquitetura orientada a serviços (*Service Oriented Architecture - SOA*). A SOA define a maneira que entidades computacionais associadas a três papéis distintos podem interagir; a Figura 71 oferece uma visualização dessas interações.

Conforme ilustrado, os provedores de serviço produzem WS e os oferecem através de publicações nos registros (repositórios centrais de serviços) ou os disponibilizam para acesso direto dos consumidores. Estes efetuam operações de busca para encontrar os serviços desejados, e então os requisitam aos registros ou diretamente aos provedores dos serviços [91].

A arquitetura dos *Web Services* ainda encontra-se em desenvolvimento, mas o núcleo da sua pilha de protocolos pode ser apresentado em camadas de acordo com a Figura 72.

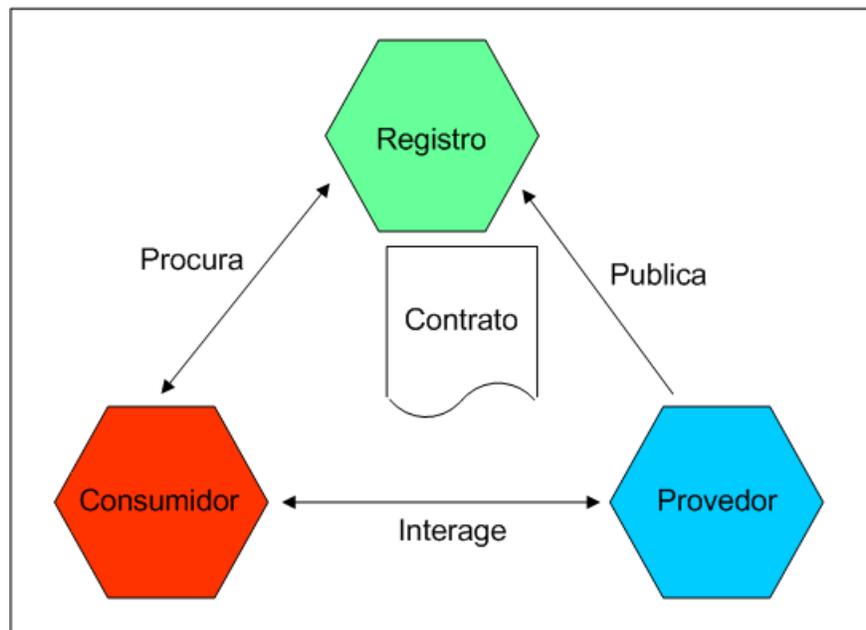


Figura 71: Arquitetura SOA [90].

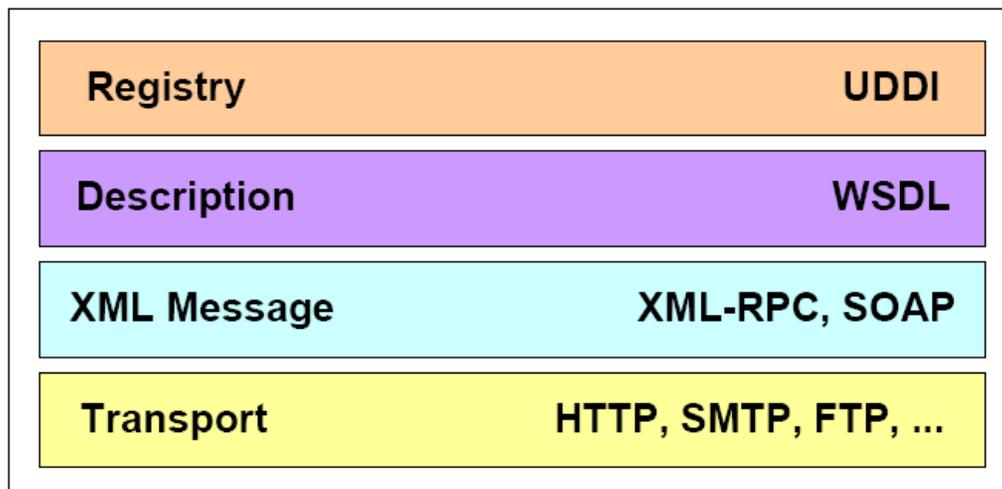


Figura 72: Pilha de protocolos da arquitetura de *Web Service* [91].

A partir de uma descrição *Bottom-up*, tais camadas são responsáveis pelas seguintes funções [91, 90]:

- **Transport**: transferir as mensagens entre as aplicações através de protocolos tradicionais da Internet como *Hypertext Transfer Protocol* (HTTP), *Simple Mail Transfer Protocol* (SMTP), *File Transfer Protocol* (FTP), e outros como *Blocks Extensible Exchange Protocol* (BEEP);

- **XML Messages:** definir a formatação básica das mensagens a partir da linguagem XML, permitindo que elas sejam entendidas pelos participantes da comunicação. Esta camada inclui o *XML Remote Procedure Call* (XML-RPC) e principalmente o *Simple Object Access Protocol* (SOAP).
- **Service Description:** descrever a interface pública de um determinado WS. A linguagem utilizada para isso é a *Web Services Description Language* (WSDL).
- **Service Registry:** centralizar as descrições dos serviços em um repositório central permitindo a publicação e a busca dos serviços. Esta camada é normalmente implementada utilizando *Universal Discovery, Description and Integration* (UDDI) [92].

De especial relevância para o foco deste trabalho, o SOAP possui um mecanismo de otimização da transmissão de uma mensagem formatada de acordo com esse protocolo, através da codificação de certas partes da mensagem, mais especificamente dos elementos que possuem conteúdo do tipo *xs:base64Binary* [93]. Esse mecanismo é importante na implementação do protótipo do Capítulo 8, para compactar principalmente o elemento *YValue* do *ecgAware*, pois esse elemento contém no formato *base64Binary* a maior parte da informação do ECG, e conseqüentemente possui o maior tamanho em bytes.

Com relação à interação entre as entidades provedora e consumidora que efetivam o oferecimento e o uso de um WS, embora ela possa se dar de várias maneiras, em geral essa interação é composta dos seguintes passos, ilustrados na Figura 73a [94]: (1) provedor e consumidor se conhecem, ou pelo menos um conhece o outro; (2) consumidor e provedor de alguma forma firmam acordo quanto a descrição e semântica do serviço que vão conduzir a interação entre os agentes dessas entidades; (3) a descrição e a semântica do serviço são consolidadas pelos agentes do provedor e do consumidor; e por fim, (4) as entidades trocam mensagens.

Conforme o passo 1 da Figura 73b, para o caso da entidade consumidora não conhecer previamente a entidade provedora e o serviço que ela oferece, a entidade consumidora tem de descobrir um provedor candidato e seu serviço. Conforme o [95], descobrir é o ato de localizar uma descrição computável de um serviço inicialmente desconhecido e que atende determinados critérios funcionais.

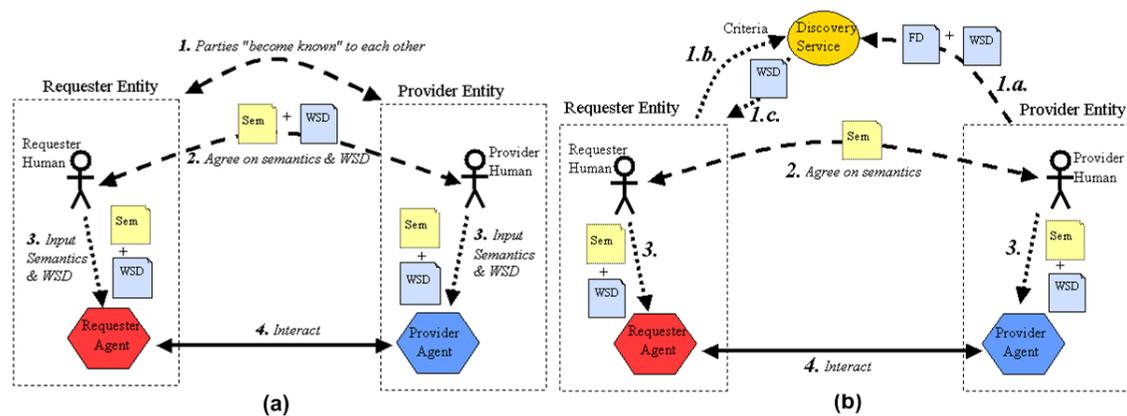


Figura 73: Interação WS consumidor-provedor [94]: (a) direta, (b) através da descoberta de serviços.

### 7.5.2 Modelos de entrega da informação contextual à Infraware

No projeto da Infraware, os três modelos de entrega da informação contextual definidos para possibilitar a comunicação entre os seus módulos componentes são: *request-response*, *time-driven* e *event-driven*. No que se refere às interações entre provedores de contexto (*Context-Wrappers*) e o *middleware* de Acesso e Integração de Dados CoDIMS, esses modelos serão implementados através de um protocolo de comunicação entre esses dois módulos.

No modo *request-response*, o *ECG-Wrapper* receberá do CoDIMS requisições explícitas, originalmente oriundas das aplicações médicas sensíveis ao contexto, e enviará ao mesmo as informações contextuais, encerrando em seguida o serviço. Um exemplo desse tipo de interação através da plataforma é um médico requisitar por meio de uma aplicação o último registro de ECG coletado de seu paciente, ou um estudo com vários registros relativos a um determinado período de tempo, e receber a resposta de sua consulta contendo a informação requisitada.

O modelo *time-driven* define que a informação atualizada seja enviada pelo *ECG-Wrapper* ao CoDIMS em intervalos periódicos de tempo, definidos pela aplicação médica *context-aware* consumidora dessa informação. Um exemplo desse modelo de entrega é o médico requisitar através da aplicação atualizações de ECG do paciente periodicamente em um determinado intervalo de tempo.

Finalmente, no modo *event-driven*, uma informação é enviada pelo *ECG-Wrapper*

ao CoDIMS para o disparo de ações pela plataforma quando há ocorrência de um evento de risco para a saúde do paciente (requisitos R3.2 e R3.3 da Tabela 26). Esse evento pode ser identificado automaticamente pelo módulo de interpretação componente do *ECG-Wrapper* (seção 7.3), ou ser acionado diretamente pelo paciente ou familiar através do botão de alarme integrado no dispositivo portátil de aquisição de ECG.

## 7.6 GUI

O requisito opcional R1.7 descrito na perspectiva geral de encapsulamento de contexto na Tabela 1, torna-se obrigatório no projeto deste sistema em função dos cenários e casos de uso do *ECG-Wrapper*, conforme indicado pelos requisitos R3.4 e R3.5 na Tabela 26. Portanto, o sistema precisa possuir interfaces gráficas que possibilitem a inserção de informações relativas a:

- **Dados pessoais do paciente**, dos quais somente o identificador único do paciente é obrigatório, uma vez que, possivelmente, as outras informações como nome, endereço, etc já estarão armazenadas na plataforma Infracare ou nas aplicações;
- **Informações de PEP**, que também não são obrigatórias devido à possibilidade de já estarem armazenadas fora da UMS, mas que, caso contrário, são relevantes sobretudo para apoiar um diagnóstico;
- **Informações relativas às sessões de monitoramento**, que são fornecidas para informar o contexto do paciente associado a uma determinada sessão de monitoramento, ou seja, o contexto referente ao registro de ECG coletado nessa sessão;
- **Troca de mensagens**, que podem ser entre paciente e médico, ou entre o profissional de saúde envolvido e um hospital ou unidade de saúde.

A interface gráfica projetada neste trabalho foi desenvolvida na linguagem Java utilizando-se o ambiente de desenvolvimento NetBeans 5.5, e será apresentada no capítulo seguinte.

## 7.7 Conclusão do Capítulo

Este capítulo apresentou o projeto do sistema *ECG-Wrapper* de acordo com as soluções computacionais avaliadas como mais adequadas para atender aos requisitos definidos na fase de análise. Para realizar essa tarefa, foi feito um mapeamento direto entre tais requisitos e as estruturas do sistema.

As restrições de tempo, aliadas à complexidade de algumas soluções, sobretudo da especificação do protocolo de comunicação com o CoDIMS, impossibilitaram o desenvolvimento de um projeto que satisfizesse a todos os requisitos levantados nas tabelas 1, 4, e 26. Entretanto, essas carências serão alvo de trabalhos futuros necessários para possibilitar o funcionamento da plataforma Infracore nos cenários reais do projeto Tele-Cardio, descritos no Capítulo 6.

No capítulo seguinte será apresentada a implementação do protótipo do *ECG-Wrapper* desenvolvido neste trabalho.

## 8 *Implementação de Protótipo*

Este capítulo apresenta a implementação de um protótipo de demonstração do *ECG-Wrapper* que, na mesma linha do que foi discutido no Capítulo 7, atende somente a alguns dos requisitos especificados durante a fase de análise. Inclusive com relação aos módulos de Comunicação *Wireless PAN* e de Interpretação, ambos ainda não integram a arquitetura desse protótipo, mas serão posteriormente incorporados ao sistema na fase final do projeto TeleCardio.

Neste sentido, o protótipo aqui apresentado não contempla questões de eficiência, nem mesmo de técnicas de programação. O intuito da implementação desse aplicativo de demonstração é mostrar tão somente a potencialidade de uso real da solução proposta, inclusive ao tratar questões relacionadas ao encapsulamento de contexto.

Dessa forma, o capítulo é composto da seção 8.1, descrevendo a arquitetura desse protótipo; da seção 8.2, apresentando suas interfaces gráficas e discutindo as funcionalidades oferecidas ao usuário; e finalmente, da seção 8.3, que realiza a conclusão do capítulo.

### 8.1 *Arquitetura do Protótipo*

O protótipo em questão foi desenvolvido na plataforma Java SE 6.0 através da IDE NetBeans 5.5. Grande parte das classes que foram definidas representam um mapeamento dos elementos e atributos do *XML Schema ecgAware* para objetos e atributos Java. Por sua vez, a implementação do *Web Service* foi feita sob o uso das facilidades providas pela plataforma Java SE 6.0, e incluiu também a implementação de uma simples aplicação cliente para validar o serviço oferecido pelo *ECG-Wrapper*.

Em que pese o curto período de tempo disponível para a realização desse mapea-

mento, não foi possível o estudo de técnicas avançadas de *XML-Java Binding*. Por esta razão, o mapeamento mencionado foi feito de uma forma simples e intuitiva, suficiente para os propósitos dessa demonstração. Além disso, com relação à disponibilização do *Web Service*, também não foi possível o estudo de mecanismos de otimização do protocolo SOAP, uma vez que este oferece diferentes opções de codificação de mensagem.

O sistema foi dividido em módulos implementados em diferentes pacotes, conforme o diagrama mostrado na Figura 74.

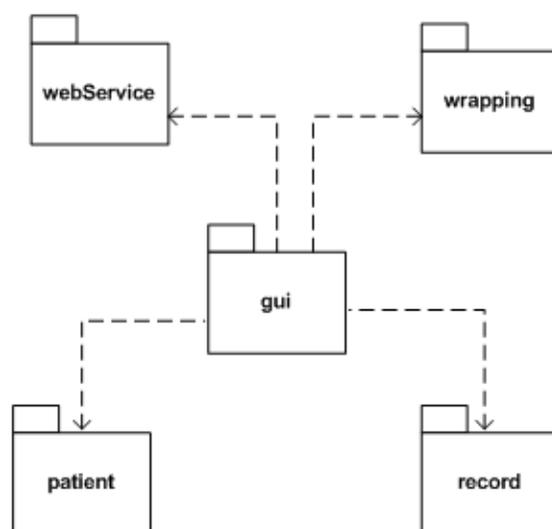


Figura 74: Diagrama de pacotes desse protótipo do *ECG-Wrapper*.

O pacote *patient* contém as classes associadas ao cadastro do paciente; por sua vez, o pacote *record* integra as classes relativas aos registros de ECG propriamente ditos e dados complementares adquiridos de forma simulada; o pacote *wrapping* contém as classes responsáveis por montar os documentos XML a serem entregues quando requisitados pela aplicação cliente; já o pacote *webService* possui a classe que implementa os dois métodos disponíveis na Web que retornam documentos XML, seja contendo o último registro de ECG, ou contendo um estudo composto de vários registros; e por fim, o pacote *gui* contém a classe *javax.swing.JFrame*, que define as interfaces gráficas do sistema e possui o método *main*, o método executável do sistema.

## 8.2 Interfaces e Funcionalidades

O programa possui uma interface principal tão logo exibida no instante da sua execução, e outras interfaces relativas a três grupos básicos de funcionalidades derivadas de alguns dos casos de uso descritos na seção 6.5.2, são eles: (i) cadastro do paciente, (ii) sessão de monitoramento, e (iii) disponibilização de estudo de ECG na Web. Esses conjuntos de funções são descritos a seguir nas seções 8.2.2, 8.2.3, e 8.2.4.

### 8.2.1 Interface Principal

A interface principal do programa, conforme pode ser visto na Figura 75, contempla os grupos de funcionalidades supracitados.



Figura 75: Interface principal do protótipo de demonstração.

No instante inicial de sua execução, o sistema verifica se há cadastro de paciente através da procura pelo arquivo XML denominado “PatientInformation.xml” no diretório corrente do sistema de arquivos. Caso já exista esse arquivo, o sistema carrega

o cadastro na memória em um novo objeto da classe *PatientInformation* do pacote *patient*. Essa operação é implementada na classe *PatXmlReader* também do pacote *patient*, e é feita através do *parser* DOM, disponível nos pacotes *javax.xml.parsers* e *org.w3c.dom*. Um outro detalhe relevante é que nessa recuperação dos dados do paciente é obtido o número de registros de ECG do mesmo já coletados pelo sistema e pré-armazenados em disco também no diretório corrente.

## 8.2.2 Cadastro do Paciente

A partir da interface principal é possível acessar a interface que contém o formulário de preenchimento dos dados do paciente (Figura 76), incluindo informações de Prontuário Eletrônico. Vale lembrar que, com exceção do código identificador do paciente, todas as informações desse formulário são opcionais, uma vez que, possivelmente, elas já estarão armazenadas na Infracore e, neste caso, não é necessário que sejam enviadas.

The screenshot shows a Windows-style application window titled "ECG-Wrapper". The main content area is titled "Informações do Paciente". It contains several form fields and sections:

- Menu**: A small menu bar at the top left.
- Pessoal**: Fields for "Nome:", "Data de Nascimento:", and "Sexo" (with radio buttons for "Masculino" and "Feminino").
- Contato**: Fields for "Telefone:", "Fax:", "E-mail:", and "Endereço" (with a scrollable text area).
- Prontuário Eletrônico**: A section with a checked checkbox "Incluir Prontuário Eletrônico" and a "Código do Paciente:" field containing "0123". Below are fields for "Altura:", "Peso:", "Unidade:", "Unidade:", "Hipertensão" (Sim/Não), "Diabetes" (Sim/Não), "Fumante" (Sim/Não), and "Consumidor de Álcool" (Sim/Não).
- Medicação**: A scrollable text area for entering medication.
- Outros**: A scrollable text area for other information.
- Comentários**: A scrollable text area for comments.
- Buttons**: "OK" and "Cancelar" buttons at the bottom right.
- Legend**: A small legend at the bottom right indicating "\* Campo Obrigatório".

Figura 76: Interface para inserção de dados do paciente.

Através dessa interface, o profissional de saúde que acompanha o paciente no mo-

mento da instalação da UMS pode cadastrar um novo paciente, editar um cadastro já existente, ou mesmo excluir um cadastro e os registros de ECG relativos ao mesmo. Sempre que um cadastro é alterado e validado pela checagem do código do paciente, ele é gravado no arquivo “PatientInformation.xml” no diretório corrente. Cabe ressaltar que não é interessante haver mais de um paciente cadastrado em uma instância do sistema, uma vez que se trata de monitoramento restrito a um indivíduo.

A maior parte das funcionalidades aqui descritas foi implementada em classes pertencentes aos pacotes *patient* e *gui*, através de métodos *getters* e *setters* que recuperam e alteram os dados no objeto *PatientInformation* e em *containers* do pacote *javax.swing*. Com relação à escrita do objeto *PatientInformation* no documento XML “PatientInformation.xml”, esta é implementada na classe *PatXmlWriter* do pacote *wrapping*.

### 8.2.3 Sessão de Monitoramento

O sistema permite que o paciente acompanhado ou não de um profissional de saúde inicie novas sessões de monitoramento de acordo com as instruções de seu médico responsável. Ao clicar no botão “Nova Sessão” da interface principal, o usuário tem acesso à interface mostrada na Figura 77. Através dela é possível a inserção de dados relativos ao contexto do paciente como a atividade a ser realizada durante a sessão, ou uma descrição do local de monitoramento (ex: domicílio do paciente), ou a pressão sanguínea medida no momento de início da sessão e a(s) medicação(ões) utilizada(s) durante a sessão compondo o protocolo clínico; e também inserção do nome do profissional de saúde responsável pelo monitoramento. Além disso, nessa interface pode ser definido o tempo de duração do monitoramento.

The screenshot shows a window titled "ECG-Wrapper" with a "Menu" bar. The main content area is titled "Sessão 001" and is divided into four panels:

- Protocolo Clínico:** Contains a dropdown menu for "Medicação utilizada:" with "Digoxina (digitálico)" selected. Below it are input fields for "Pressão Sistólica:" (110) and "Pressão Diastólica:" (60), each with a "Unidade:" dropdown set to "mmHg".
- Indicadores do Monitoramento:** Contains a time input field for "Tempo de duração da sessão:" (00:10:00) and a frequency input field for "Frequência de Amostragem:" (256 Hz).
- Descrição da Sessão:** Contains a dropdown menu for "Atividade realizada:" with "Repouso" selected and a text input field for "Local de realização:" with "Domicílio do paciente" entered.
- Monitoramento:** Contains a text input field for "Médico Responsável:" with "Dr. Rodrigo Varejão" entered and a large blue button labeled "Iniciar Monitoramento".

Figura 77: Interface que permite o início de uma nova sessão.

O clique no botão “Iniciar Monitoramento” dá início a uma nova sessão que, ao seu final, traz como resultado um registro de ECG que é armazenado em disco. O procedimento de monitoramento é implementado em classes do pacote *record* que recebem dados simulados, e a escrita de um registro em disco no arquivo “Record#.xml” é feita pela classe *RecordXmlWriter* do pacote *wrapping*. A interface exposta na tela durante o monitoramento é mostrada na Figura 78.

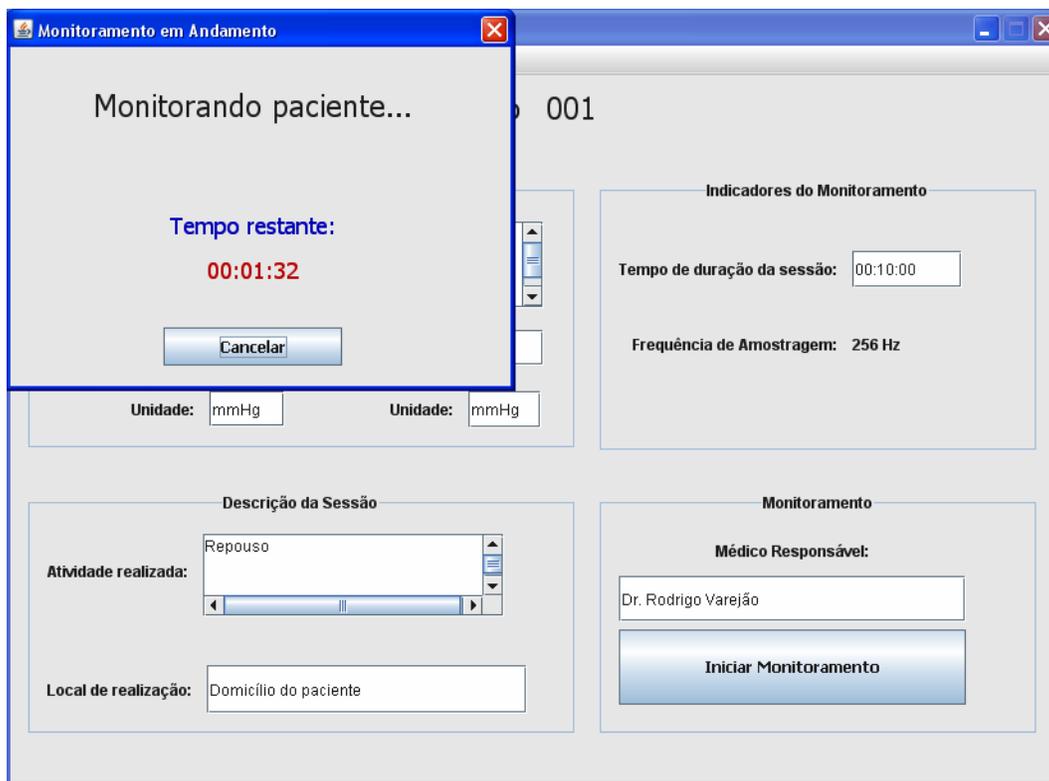


Figura 78: Interface de monitoramento de uma sessão.

Caso haja um evento de emergência durante o monitoramento, o sistema exibe uma janela de alerta (Figura 79) sinalizando a sua ocorrência.

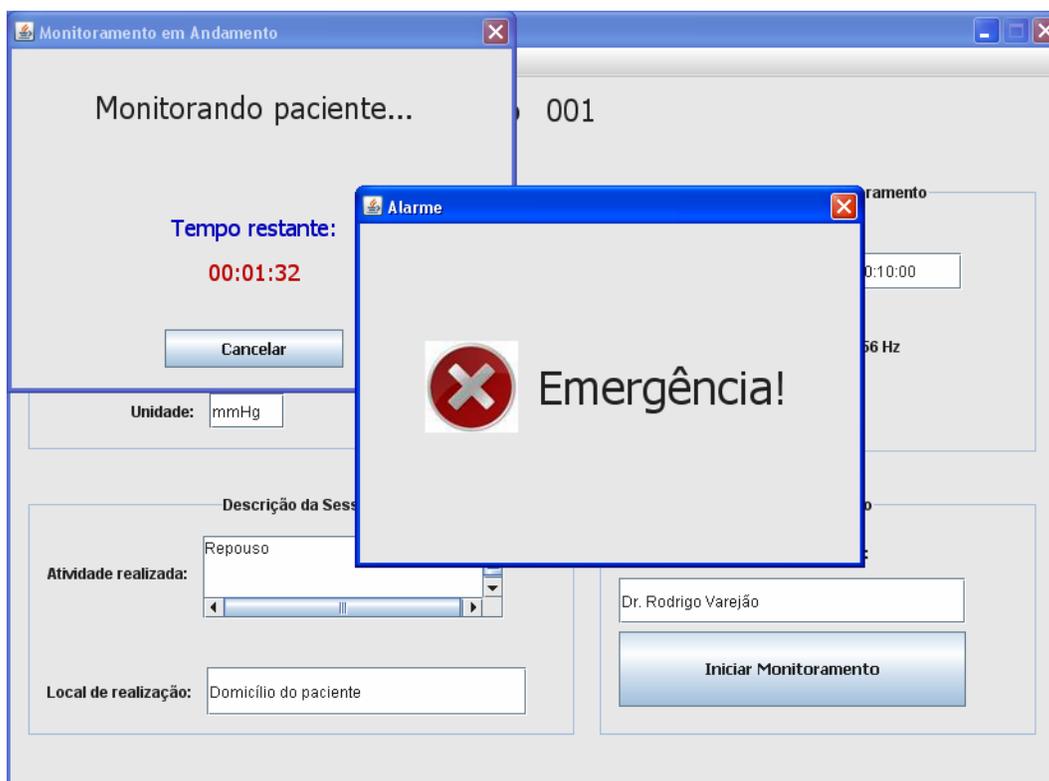


Figura 79: Janela de alerta da ocorrência de um evento de urgência.

#### 8.2.4 Disponibilização de Estudo de ECG

Todo o trabalho realizado pelo *ECG-Wrapper*, por se tratar de um componente de um sistema distribuído, tem como ponto crucial a transmissão da informação nele produzida. Portanto, é fundamental a existência desta funcionalidade, cuja disponibilidade pode ser conferida diretamente na interface principal do sistema (Figura 75).

A disponibilização do serviço oferecido pelo sistema é feita através de dois métodos publicados na Web. O primeiro deles, ao ser chamado remotamente retorna o documento XML “ecgStudy#”<sup>1</sup> contendo o cadastro do paciente (eventualmente só com o seu identificador) e o último registro de ECG adquirido pelo sistema. Já no caso do segundo método, no instante em que este é chamado, uma janela de consulta (Figura 80) é aberta na aplicação cliente para obter do usuário desta o período de tempo (data e hora) de seu interesse ao qual correspondem os registros de ECG a serem enviados no documento XML “ecgStudy#”.

<sup>1</sup>O símbolo # representa o número inteiro identificador do estudo.

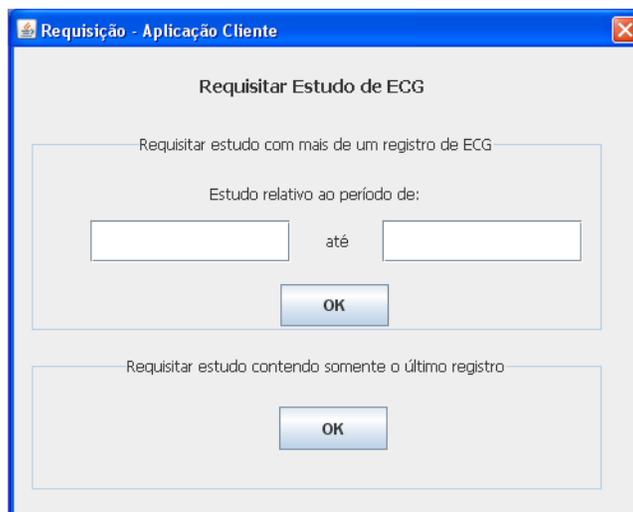


Figura 80: Janela de consulta da aplicação cliente.

O procedimento de montagem do documento XML a ser disponibilizado é implementado no pacote *wrapping*. De acordo com a escolha informada pela aplicação cliente, o sistema faz a leitura de um ou vários arquivos XML que contêm os registros solicitados, através do mesmo procedimento que utiliza o *parser* DOM mencionado anteriormente para a recuperação dos dados do paciente.

## 8.3 Conclusão do Capítulo

De fato, implementar um ambiente completo de aquisição e transmissão de eletrocardiogramas para a plataforma de apoio a sistemas sensíveis ao contexto Infraware não é uma tarefa trivial, pois exige conhecimento dos diversos domínios envolvidos, seja o de encapsulamento de contexto, ou o de representação e transmissão de sinais vitais, ou o de protocolos de comunicação e seus mecanismos de otimização, ou até mesmo de técnicas de programação adequadas à manipulação de documentos XML e disponibilização de *Web Services*.

Neste sentido, após terem sido feitas amplas investigações sobretudo no que se refere à representação de ECG, em função das restrições de tempo, o protótipo aqui apresentado consiste tão somente em uma demonstração que serve de ponto de partida para um processo iterativo de desenvolvimento que, ao seu final, tem por objetivo produzir o sistema *ECG-Wrapper* que será usado nos cenários reais explorados pelo

projeto TeleCardio.

## 9 *Conclusões e Trabalhos Futuros*

Este capítulo sintetiza conclusões gerais a respeito deste trabalho e indica trabalhos futuros a serem desenvolvidos para aperfeiçoar o *ECG-Wrapper*. A seção 9.1 fornece as conclusões gerais, e a seção 9.2 apresenta as expectativas de trabalhos futuros.

### 9.1 *Conclusões Gerais*

Neste trabalho foi desenvolvido o projeto de uma arquitetura para o *ECG-Wrapper*, o módulo de aquisição e encapsulamento de contexto da plataforma Infracore aplicada ao telemonitoramento da atividade cardíaca de pacientes. Essa arquitetura priva pela generalidade e, especialmente, inclui a especificação de um formato de representação de ECG, que pode ser estendido para a representação de outros sinais vitais.

Os esforços empregados na definição da arquitetura do *ECG-Wrapper* envolveram investigações em torno de (i) o domínio de encapsulamento de contexto, (ii) o domínio de representação e transmissão de eletrocardiogramas, (iii) cenários de uso do *ECG-Wrapper*, e de (iv) tecnologias potencialmente úteis na arquitetura. Como resultados da fase de análise foram especificados requisitos referentes aos três primeiros itens supracitados; a fase de projeto produziu o formato de representação de ECG e a arquitetura geral do *ECG-Wrapper*; na fase de implementação, devido a restrições de tempo, foi possível somente a produção de um protótipo de demonstração de funcionamento do sistema a fim de mostrar a potencialidade de uso real da solução proposta neste trabalho.

Ainda recentemente, a maioria das abordagens associadas à aquisição de contexto em plataformas de apoio a aplicações *context-aware* existentes na literatura são deficientes com relação à generalidade, e à representatividade semântica das soluções

propostas. Isto se deve ao fato de que essas abordagens normalmente são guiadas pelas tecnologias disponíveis, ao invés de utilizarem os critérios de engenharia de domínio adotados neste trabalho. Essa carência seguramente tem contribuído para a postergação de uma maior difusão de sistemas sensíveis ao contexto.

Neste sentido, sob o ponto de vista de context-awareness, as principais contribuições da pesquisa desenvolvida neste trabalho envolvem a captura de requisitos importantes no tratamento de tal problema, decisões arquiteturais de projeto, e a escolha de tecnologias como XML, *Web Services*, e *Zigbee*. A partir de agora a tarefa de aquisição de contexto em cenários futuros de uso da Infracore em outros domínios verticais, como os de turismo, *e-learning*, dentre outros, não envolve mais a investigação de como encapsular contexto, uma vez que já foi definida uma arquitetura conceitual para esse propósito. Desta forma, o *Context-Wrapper* pode ser instanciado para domínios verticais somente através do estudo de tais domínios. Além disso, o *Context-Wrapper* proporciona à Infracore aquisição de contexto livre da dependência de tecnologias, e de complexidade transparente.

Ao passo que, do ponto de vista da área de saúde, mais precisamente de telemonitoramento de pacientes cardíacos, este trabalho produziu resultados significativos para o projeto TeleCardio. Uma vez finalizado e testado, este projeto será aplicado em cenários reais de hospitais, unidades de saúde, e domicílios de pacientes cardíacos, para oferecer, inicialmente à comunidade do Estado do Espírito Santo, uma maior qualidade no atendimento médico capaz de diminuir diferenças sociais e culturais existentes, oferecendo às comunidades menos privilegiadas serviços de saúde eficazes.

Futuramente, a extensão do escopo do TeleCardio ao monitoramento de outros sinais vitais, permitirá um serviço mais completo capaz de explorar novos cenários de uso. Nessa direção este trabalho também oferece resultados, tanto de projeto quanto de reuso da metodologia aqui aplicada.

## 9.2 **Trabalhos Futuros**

Em que pese o escopo deste trabalho (Projeto Final de Graduação), tópicos importantes identificados durante as pesquisas não foram investigados em maior profundidade. A adição desses estudos ao trabalho permitiria acrescentar ao sistema especi-

ficações mais bem definidas da solução de tais questões, permitindo a implementação das funcionalidades a elas associadas. Uma lista não exaustiva desses tópicos inclui:

1. Protocolo de comunicação com a Infracore: de acordo com a arquitetura SOA discutida na seção 7.5, a plataforma Infracore exerce o papel de consumidora dos dados produzidos no *ECG-Wrapper*. Portanto, é necessária neste sistema não só a disponibilização de um *Web Service*, mas também a especificação de um protocolo de comunicação que define as interações com a plataforma e as mudanças de estado do sistema;
2. Procedimento de compressão dos dados produzidos no *ECG-Wrapper*: é desejável que a mensagem XML a ser enviada à Infracore seja comprimida através de um esquema de otimização, que demanda a especificação de um formato mais eficiente para a transmissão de dados, privilegiando assim, a perspectiva da camada de Apresentação do Modelo OSI;
3. Conversão de modelos *ecgAware*-Relacional: para entregar os dados no formato relacional ao CoDIMS, é preciso uma conversão do modelo no formato *ecgAware* para o formato relacional. Algumas alternativas para essa tarefa foram apresentadas na seção 7.4.
4. Implementação da interface entre a camada de Tradução e a de Interpretação: essa interface permite que o módulo de *wrapping* - que encapsula os dados no *ecgAware* - receba os dados processados no módulo de análise automática de ECG (vide seção 7.3);
5. Pesquisas em torno de outros sinais vitais a fim de produzir para cada sinal vital estudado um meta-modelo semelhante ao *ecgAware*: a partir disso, será possível ampliar o potencial de monitoramento remoto de pacientes atendendo com mais intensidade demandas da área de saúde;
6. Criação de ontologias de domínio para sinais vitais, capazes de capturar ainda com mais precisão os conceitos e leis associadas a eles;
7. Criação de uma ontologia de domínio para o encapsulamento de contexto: a partir do conhecimento capturado e documentado neste trabalho é possível a

especificação de tal ontologia, que viria a facilitar ainda mais o projeto de *Context-Wrappers* para o tratamento de domínios verticais associados a outras instâncias da plataforma Infracore;

8. Adaptação do *ECG-Wrapper* que permita a execução desse sistema em dispositivos de computação detentores de menos recursos computacionais, como PDAs, *smartphones*, etc.

Dentre os oito tópicos listados acima, pode-se destacar os quatro primeiros da seqüência como sendo de relevância imediata para o projeto TeleCardio.

Dessa forma, o investimento no estudo e na busca da solução de tais questões pode gerar trabalhos científicos acadêmicos de graduação e pós-graduação. Uma vez finalizada a implementação do *ECG-Wrapper*, ele será utilizado no Projeto TeleCardio em cenários reais, podendo assim, juntamente com os outros módulos desenvolvidos em trabalhos relacionados no LPRM e no LABTEL, ser explorado comercialmente por empresas da área de saúde interessadas.

## *Referências*

- [1] DEY, A. K. Providing architectural support for building context-aware applications. 2000. Ph.D. Thesis, Georgia Institute of Technology.
- [2] BASHSHUR, R. L. e. a. Telemedicine/telehealth: An international perspective. *Telemedicine Journal and e-Health*, v. 8, n. 01, 2002.
- [3] AO, R. A.; FILHO, J. *HMMECGGUI*  
*Programa de monitoramento de ECG ambulatorial*. [S.l.], 2006.
- [4] SCP How to Implement - Part I. [S.l.]. Disponível em <http://www.openecg.net>.
- [5] FILHO, J. P.; PESSOA, R.; CALVI, C. e. a. Infraware: Um middleware de suporte a aplicações móveis sensíveis ao contexto. *24 Simpósio Brasileiro de Redes de Computadores, Curitiba-PR*, 2006.
- [6] TELECARDIO. Telecardiologia a Serviço do Paciente em Ambientes Hospitalares e Residenciais. 2005. Projeto DI/DEE/UFES, Financiamento: FAPES.
- [7] ZAJTCHUK, G. Telemedicine: A new dimension in the practice of medicine. *Disease-a-month*, v. 45, n. 6, June 1999.
- [8] PASCOE, J. Adding Generic Contextual Capabilities to Wearable Computers. In: *Proceedings of the 2nd IEEE International Symposium on Wearable Computers (ISWC'98)*, pp. 92-99, Pittsburgh, PA, IEEE. [S.l.: s.n.], 1998.
- [9] INFRAWARE. Infraware - uma plataforma para desenvolvimento de aplicações móveis e context-aware. 2005. Projeto FAPES (Processo 30897041/2005), Departamento de Informática/UFES.
- [10] CARDOM. CARDOM - telecardiologia a serviço do paciente em domicílio. 2005. Projeto FAPES (Processo 30899583/2005), Departamento de Engenharia Elétrica/UFES.
- [11] YUCAT. *Yucat Mobile Business Solutions B.V., Corporate website*. [Http://www.yucat.com](http://www.yucat.com).
- [12] BACKX, N. On the design of a mobile e-health platform - towards deployment flexibility. 2006. Thesis for a Master of Science degree in Telematics from the University of Twente, Enschede, The Netherlands.
- [13] MOBIHEALTH. *MobiHealth, Project website*. [Http://www.mobihealth.org](http://www.mobihealth.org).

- [14] MELANDER-WIKMAN et al. D5.1: Overall evaluation of the mobihealth trials and services;. *MobiHealth project*, v2.6, September 2004.
- [15] AWARENESS. *Freeband Awareness, Project website*. [Http://awareness.freeband.nl](http://awareness.freeband.nl).
- [16] HS24. *HealthService24, Project website*. [Http://www.healthservice24.com](http://www.healthservice24.com).
- [17] SUN/JINI. Jini Technology Surrogate Architecture Specification 1.0. *Sun Microsystems Inc.*, 2003. Disponível em: <http://surrogate.jini.org/>.
- [18] ERFIANTO, B. *Design of a Vital Sign Protocol Format Using XML and ASN.1*. Dissertação (Mestrado) — A thesis submitted to the department of Computer Science of the University of Twente for the partial fulfillment of the requirements of the degree of Master of Science in Telematics, 2004.
- [19] DOCKHORN, P. *Towards a Services Platform for Context-Aware Applications*. Dissertação (Mestrado) — University of Twente, The Netherlands, 2003.
- [20] PESSOA, R. *Infraware: Um Middleware de Suporte a Aplicações Sensíveis ao Contexto*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, 2006.
- [21] ANDREÃO, R. Segmentation de battements ecg par approche markovienne: application à la détection d'ischémies, tese de doutorado. 2004. Dep. EPH, Institut National des Télécommunications, Evry, França, 182p.
- [22] BORGES, C.; ANDREÃO, R.; SEGATTO, M. Processamento de sinais de ecg para geração automática de alarmes. 2006. Proc. Workshop de Informática Médica. Vila Velha, ES.
- [23] BUMACHAR, E.; AO, R. A.; SEGATTO, M. Aquisição e transmissão sem fio de sinais ecg com uso intermitente de transceptor. 2006. Proc. Workshop de Informática Médica, Vila Velha, ES.
- [24] AO, R. A.; FILHO, J. P.; CALVI, C. TeleCardio: Telecardiologia a Serviço de Pacientes Hospitalizados em Domicílio. *X Congresso Brasileiro de Informática em Saúde, Florianópolis, Brasil*, 2006.
- [25] Andreão R.V et al. ST-segment Analysis Using HMM Beat Segmentation: Application to Ischemia Detection. *Computers in Cardiology, Chicago, EUA*, 2004.
- [26] Pessoa R.M. et al. Aplicação de um middleware sensível ao contexto em sistema de telemonitoramento de pacientes cardíacos. 2006. SEMISH (CSBC 2006), Campo Grande (MS).
- [27] SCHMIDT, A.; LAERHOVEN, K. How to build smart appliances. *IEEE Personal Communications*, 6-11, 2001.

- [28] LEHTONEN, J. Designing Context-Aware Systems: Choosing Sensors. *Laboratory of Computational Engineering. P.O. Box 9203, FIN-02015 HUT, FINLAND*, 2004.
- [29] TRACZ, W. Domain-specific software architecture (DSSA) frequently asked questions (FAQ). *Software Engineering Notes*, 1994.
- [30] CZARNECKI, K.; EISENECKER, U. *Generative Programming: Methods, Tools, and Applications*. [S.l.]: Addison-Wesley, 2000.
- [31] HARSU, M. *A Survey on Domain Engineering*. [S.l.], 2002.
- [32] WICKMAN, G.; SOLDERITSCH, J. A systematic software reuse program based on an architecture-centric domain analysis. *Software Technology Conference, Salt Lake City, Utah*, 1994. Disponível em: <http://www.asset.com/stars/lmtds/Papers/ReusePapers.html>.
- [33] CLEMENTS, P.; NORTHROP, L. *Software Product Lines: Practices and Patterns*. [S.l.]: Addison-Wesley, 2002.
- [34] SCHMID, K. Scoping software product lines. *Patrick Donohoe, editor, Software Product Lines, Experience and Research Directions, pages 513:532. Kluwer Academic Publisher*, 2000.
- [35] DEY, A.; ABOWD, G. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction Journal* 16, 24, pp. 97-166, 2001.
- [36] Bauer M. et al. A collaborative wearable system with remote sensing. *Proceedings of the 2nd International Symposium on Wearable Computers (ISWC'98), 10-17. Los Alamitos, CA: IEEE*, 1998.
- [37] BARBOSA, A. C. P. *Middleware para Integração de Dados Heterogêneos baseado em Composição de Frameworks*. Tese (Doutorado) — PUC-Rio, Brasil, In Portuguese, 2001.
- [38] SILVESTRE, L. J.; BIANCARDI, C.; BARBOSA, A. Sistemas para Integração de Dados Heterogêneos. 2004. Disponível em: <<http://codims.lprm.inf.ufes.br/sistIntDados.pdf>>. Acesso em: 10 de Novembro de 2006.
- [39] ZIEGLER, P.; DITTRICH, K. Three decades of data integration - all problems solved? *JACQUART, R. (Ed.). 18th IFIP World Computer Congress (WCC 2004), Volume 12, Building the Information Society. Toulouse, France, August 22-27: Kluwer, 2004. (IFIP International Federation for Information Processing, v. 156), p. 3\_12*, 2004.
- [40] SILVESTRE, L. *Uma Abordagem Baseada em Ontologias para a Gerência de Metadados do CoDIMS*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, 2005.

- [41] LAERHOVEN, v. K. *On-line Adaptive Context Awareness Starting From Low-level Sensors*. Tese (Doutorado) — A thesis submitted in partial fulfillment of the requirements for the degree of Licentiaat in de Informatica, Free University of Brussels, 1999.
- [42] BRISA. *Arquiteturas de Redes de Computadores OSI e TCP/IP*. [S.l.]: Sociedade Brasileira para Interconexão de Sistemas Abertos, 2 Edição Revisada e Ampliada, 1997.
- [43] DICOM\_SUPPL.30. *DICOM Suppl. 30*. Medical.nema.org/Dicom/supps/sup30\_lb.pdf.
- [44] DICOM-WEBPAGE. *NEMA'S OFFICIAL DICOM WEB Page*. [Http://medical.nema.org](http://medical.nema.org).
- [45] CORBAMED. *CORBAméd*. [Www.acl.lanl.gov/OMG](http://www.acl.lanl.gov/OMG).
- [46] I-MED. *I-Med*. [Http://www.hnbe.com/healthweb/imedpub](http://www.hnbe.com/healthweb/imedpub).
- [47] HL7. *HL7*. [Http://www.hl7.org](http://www.hl7.org).
- [48] ECG-LIBRARY. *ECG Library*. <http://www.ecglibrary.com/ecghist.html>.
- [49] GUYTON, A.; HALL, J. *Tratado de Fisiologia Médica*. [S.l.]: 9 edição, Editora Guanabara Koogan, 1997.
- [50] ECG-LEARNING\_CENTER. *ECG Learning Center*. [Http://medlib.med.utah.edu/kw/ecg/ecg\\_outline/Lesson1/index.html](http://medlib.med.utah.edu/kw/ecg/ecg_outline/Lesson1/index.html).
- [51] GAWANDE, A. *Complicações: Dilemas de um cirurgião diante de uma ciência imperfeita*. Rio de Janeiro: Objetiva, 2002.
- [52] Andreão R.V et al. Online Beat Segmentation and Classification Through Hidden Markov Models. *Proceedings IFMBE, vol. 5, João Pessoa, Brasil, 2004*.
- [53] Andreão R.V.; Dorizzi B.; Boudy J. ECG Analysis through hidden Markov models. *IEEE Trans. Biomed. Eng., vol. 53, n. 8, 2006*.
- [54] PHYSIONET. *Physionet*. [Http://www.physionet.org/](http://www.physionet.org/).
- [55] PHYSIONET-PHYSIOBANK. *Physionet Physiobank*. [Http://www.physionet.org/physiobank/other.shtml](http://www.physionet.org/physiobank/other.shtml).
- [56] NIH-NCRR. *NIH/NCRR*. [Http://www.ncrr.nih.gov](http://www.ncrr.nih.gov).
- [57] PHYSIONET\_RESOURCE. *Physionet Resource*. [Http://www.physionet.org/resource.shtml](http://www.physionet.org/resource.shtml).
- [58] MOODY, G. *WFDB Programmer's Guide*. Tenth edition, revised and with additions for wfdb library version 10.4.4. [S.l.], 2006. Disponível em: <http://www.physionet.org/physiotools/wpg/>.

- [59] DICTIONARY.COM. *flat file. Dictionary.com. The Free On-line Dictionary of Computing, Denis Howe*. [Http://dictionary.reference.com/browse/flat file](http://dictionary.reference.com/browse/flat%20file).
- [60] CENTC251. *CENTC251*. [Http://www.centc251.org/](http://www.centc251.org/).
- [61] SCP Document CEN/TC-251 N02-15. [S.l.]. Disponível em: <http://www.centc251.org/>.
- [62] OPENECG. *OpenECG*. [Http://www.openecg.net/](http://www.openecg.net/).
- [63] CLUNIE, A. Extension of an open source DICOM toolkit to support SCP-ECG Waveforms. *2nd OpenECG Workshop 2004, Berlin, Germany*, 2004.
- [64] Wang H. et al. A markup language for electrocardiogram data acquisition and analysis (ecgML). *BMC Medical Informatics and Decision Making 2003, 3:4*, 2003.
- [65] FOOD AND DRUGS ADMINISTRATION. *FDA Specification*. [S.l.]. [FDA\\_XML\\_Data\\_Format\\_Design\\_Specification\\_DRAFT\\_C](http://www.fda.gov/cder/oc/oxml/FDA_XML_Data_Format_Design_Specification_DRAFT_C.pdf).
- [66] XML. *XML*. [Http://xml.org](http://www.xml.org/).
- [67] W3C. *W3C*. [Http://www.w3.org/xml](http://www.w3.org/xml/).
- [68] W3SCHOOLS. *W3Schools*. [Http://www.w3schools.com/xml/xml\\_what.asp](http://www.w3schools.com/xml/xml_what.asp).
- [69] XML-CORE-STANDARDS. *XML Core Standards*. [Http://www.xml.coverpages.org/coreStandards.html](http://www.xml.coverpages.org/coreStandards.html).
- [70] Wang H. et al. ecgML: Tools and Technologies for Multimedia ECG Presentation. *Proceedings of XML Europe Conference, London, IDEAlliance, O'Reilly (London)*, 2003.
- [71] FDA. *FDA*. [Http://www.fda.gov/cder/site/default.htm](http://www.fda.gov/cder/site/default.htm).
- [72] HL7-ECG. *HL7 - ECG*. [Http://www.hl7.org/V3AnnECG/index.htm](http://www.hl7.org/V3AnnECG/index.htm).
- [73] FDADF Requirements Specification. [FDA\\_XML\\_Data\\_Format\\_Requirements\\_Specification\\_DRAFT\\_C](http://www.fda.gov/cder/oc/oxml/FDA_XML_Data_Format_Requirements_Specification_DRAFT_C.pdf).
- [74] BADILINI, F.; ISOLA, L. Freeware ECG Viewer for the XML FDA Format. *2nd OpenECG Workshop 2004, Berlin, Germany*, 2004.
- [75] AMPS-LLC. *AMPS LLC*. [Http://www.amps-llc.com/](http://www.amps-llc.com/).
- [76] FDA-ECG. *FDA - ECG*. [Http://www.fda.gov/cder/regulatory/ersr/ECGdata.htm](http://www.fda.gov/cder/regulatory/ersr/ECGdata.htm).
- [77] ROSSON, M.; CARROLL, J. The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. In: \_\_\_\_\_. [S.l.]: Lawrence Erlbaum Associates, 2002. cap. 53 - Scenario-Based Design, p. pp. 1032–1050.

- [78] TILOS. *TILOS: MEDICINE AND TELE-MEDICINE, "TIM-TEM"1997-1998*. [Http://users.otenet.gr/timtem/html/telpho.html](http://users.otenet.gr/timtem/html/telpho.html).
- [79] CTIT/TWENTE. *CTIT/TWENTE*. [Http://www.ctit.utwente.nl/research/sro/ehealth/](http://www.ctit.utwente.nl/research/sro/ehealth/).
- [80] IEETA. *IEETA - Instituto de Engenharia Eletrônica e Telemática de Aveiro, Portugal*. [Http://www.ieeta.pt/sias/projects\\_Details.php?id=1](http://www.ieeta.pt/sias/projects_Details.php?id=1).
- [81] FALBO, R. *Análise de Sistemas, Notas de Aula*. Digite o texto aqui. 2002.
- [82] GARLAN, D.; SHAW, M. *Advances in Software Engineering and Knowledge Engineering, Series on Software Engineering and Knowledge Engineering*. In: \_\_\_\_\_. [S.l.]: World Scientific Publishing Company, Singapore, 1994. Vol 2, cap. Introduction to Software Architecture, p. pp. 1–39. Digite o texto aqui.
- [83] ZIGBEE-TUTORIAL. *ZigBee Tutorial*. [Http://www.tutorial-reports.com/wireless/zigbee/zigbee-introduction.php](http://www.tutorial-reports.com/wireless/zigbee/zigbee-introduction.php).
- [84] MICROCONTROLLER. *Microcontroller*. [Http://www.microcontroller.com/Embedded.asp?did=14](http://www.microcontroller.com/Embedded.asp?did=14)
- [85] ZIGBEE Specification. [S.l.]. Disponível em: [http://www.zigbee.org/en/spec\\_download/download\\_](http://www.zigbee.org/en/spec_download/download_)
- [86] CALVI, C.; GONCALVES, B.; FILHO, J. *Uma Introdução as Tecnologias de Computação Móvel*. [S.l.], 2006.
- [87] TANENBAUM, A. *Redes de Computadores*. [S.l.: s.n.], 2003. 4a Edição, Ed. Campus.
- [88] ZIGBEE-ALLIANCE. *ZigBee Alliance*. [Http://www.zigbee.org/](http://www.zigbee.org/).
- [89] CÔCO, T. *Implementando Wrappers XML e Relacional para o CoDIMS*. *Universidade Federal do Espírito Santo, Projeto Final de Graduação*, 2005.
- [90] SILVA, F. *Controle de Acesso a Web Services Baseado em um Protocolo de Autenticação Segura*. *Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Uberlândia*, 2004.
- [91] SANTOS, L. O. B. S. *Semantic Services Support for Context-Aware Platforms*. *Dissertação de Mestrado, Programa de Pós-Graduação em Informática, UFES*, 2004.
- [92] UDDI. *UDDI*. [Http://www.uddi.org/specification.html](http://www.uddi.org/specification.html).
- [93] MALHOTRA, A.; BIRON, P. *XML Schema Part 2: Datatypes Second Edition*. *W3C Recommendation*, 2004. Disponível em: [<http://www.w3.org/TR/xmlschema-2/>](http://www.w3.org/TR/xmlschema-2/).
- [94] WS. *WS*. [Http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#id2260892](http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#id2260892).
- [95] WS-GLOSSARY. *WS Glossary*. [Http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/](http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/).

## *APÊNDICE A - Exemplo de XML Document no formato ecgAware*

```
<?xml version="1.0" encoding="UTF-8"?>
<ECGStudy studyDate="2006-08-13" studyTime="14:20:00" studyID="78" alarm="false">
<PatientInformation recordNumber="23" patientID="0">
<Demographics>
<Name>Bernardo Goncalves</Name>
<Sex>Male</Sex>
<DoB>1982-08-13</DoB>
<Phone>2799161812</Phone>
<Email>ber.nunes@gmail.com</Email>
</Demographics>
<EPR>
<Height unit="cm">190.0</Height>
<Weight unit="kg">80.0</Weight>
<Hypertension>true</Hypertension>
<Diabetes>true</Diabetes>
<Smoker>true</Smoker>
<Alcohol>true</Alcohol>
</EPR>
</PatientInformation>
<Record alarm="false" recordID="24" investigatorID="Rodrigo Andreao">
<RecordLeads>
<Channel>Lead III</Channel>
<RecordLead>
<XValues>
```

```
<Xoffset unit="time">14:00:00</Xoffset>
<Duration unit="time">00:20:00</Duration>
<SampleRate>256</SampleRate>
</XValues>
<YValues unit="mV">
<FileLink>http://www.physionet.org/physiobank/database/mitdb/ecg0000001.xml</FileLink>
</YValues>
</RecordLead>
</RecordLeads>
<RecordAnnotations author="ecgwrapper">
<RecordWaveNotation>
<Pwave>
<Onset>587</Onset>
<Peak>604</Peak>
<Offset>628</Offset>
<Interpretation>Normal</Interpretation>
</Pwave>
<QRScomplex>
<Onset>643</Onset>
<Peak>663</Peak>
<Offset>670</Offset>
<Interpretation>Normal</Interpretation>
</QRScomplex>
<Twave>
<Onset>758</Onset>
<Peak>782</Peak>
<Offset>818</Offset>
<Interpretation>Normal</Interpretation>
</Twave>
<Uwave>
<Onset>840</Onset>
<Peak>858</Peak>
<Offset>875</Offset>
<Interpretation>Normal</Interpretation>
```

```
</Uwave>
</RecordWaveNotation>
</RecordAnnotations>
<RecordMeasurements unit="ms" author="ecgwrapper" label="P-duration">
<Value>20</Value>
</RecordMeasurements>
<RecordingDevice deviceID="08">
<BaselineFilter>0.5</BaselineFilter>
<LowpassFilter>120</LowpassFilter>
</RecordingDevice>
<Context>
<AcquisitionDate>2006-08-13</AcquisitionDate>
<AcquisitionTime>14:20:00</AcquisitionTime>
<SiteID>domicilio</SiteID>
<Activity>repouso</Activity>
<ClinicalProtocol>
<Medication>digoxina (digitalico)</Medication>
<DiastolicBP unit="mmHg">60</DiastolicBP>
<SystolicBP unit="mmHg">110</SystolicBP>
</ClinicalProtocol>
</Context>
</Record>
</ECGStudy>
```