



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA**

**MARCELO ARAÚJO LIMA**

**SISTEMA REMOTO PARA ANÁLISE AUTOMÁTICA DE ECGS NOS PADRÕES HL7  
AECG E DICOM-ECG**

**FORTALEZA**

**2017**

MARCELO ARAÚJO LIMA

SISTEMA REMOTO PARA ANÁLISE AUTOMÁTICA DE ECGS NOS PADRÕES HL7  
AECG E DICOM-ECG

Dissertação apresentada ao Curso de Programa de Pós-Graduação em Engenharia de Teleinformática do Departamento de Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Paulo César Cortez

Co-Orientador: Dr. José Marques Soares

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- L699s Lima, Marcelo Araujo.  
Sistema remoto para análise automática de ECGs nos padrões HL7 AECG e DICOM-ECG / Marcelo Araujo Lima. – 2017.  
78 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2017.  
Orientação: Prof. Dr. Paulo César Cortez.  
Coorientação: Prof. Dr. José Marques Soares.
1. Eletrocardiograma. 2. Web services. 3. Middleware. 4. Sistemas de apoio ao Diagnóstico. I. Título.  
CDD 621.38
-

MARCELO ARAÚJO LIMA

SISTEMA REMOTO PARA ANÁLISE AUTOMÁTICA DE ECGS NOS PADRÕES HL7  
AECG E DICOM-ECG

Dissertação apresentada ao Curso de Programa de Pós-Graduação em Engenharia de Teleinformática do Departamento de Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. Paulo César Cortez (Orientador)  
Universidade Federal do Ceará (UFC)

---

Dr. José Marques Soares (Co-Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Danielo Gonçalves Gomes  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Giovanni Cordeiro Barroso  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. João Paulo do Vale Madeiro  
Universidade da Integração Internacional da  
Lusofonia Afro-Brasileira (UNILAB)

---

Prof. Dr. Augusto José Venâncio Neto  
Universidade Federal do Rio Grande do Norte  
(UFRN)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

## AGRADECIMENTOS

Aos meus amados familiares Francisco (pai), Ângela (mãe) pelo amor, apoio, paciência e dedicação investidos durante toda a minha vida

À minha esposa Roberta e filha Bárbara pelo carinho, amor, compreensão e incentivo para que eu pudesse concluir esta importante etapa da minha vida

Aos Professores Dr. Paulo César Cortez e Dr. João Paulo, pela paciência, oportunidades, transmissão de conhecimento e pelo tempo dedicado ao longo deste trabalho.

Aos meus amigos do laboratório LESC, que muito me ajudaram nesta caminhada. Um agradecimento especial aos amigos Jonas Rodrigues, Rômulo Lopes, Ronaldo Milfont, João Marcelo, Rodrigo Fernandes Freitas, Francisco Anselmo (Franco), Pedro Neto, Rômulo Lopes, Edson Cavalcante, Jilseph Lopes da Silva, Francisco Vanilson, Caio Ribeiro, Prof. Ricardo Jardel e Prof. Jarbas Ariel, Prof. Igor, Prof. Mario Wilson.

A Deus, por toda a força prestada e por fim, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro dado através da bolsa de Mestrado.

“Concedei-nos, Senhor, a Serenidade necessária para aceitar as coisas que não podemos modificar, Coragem para modificar aquelas que podemos, e Sabedoria para distinguir umas das outras.”

(Reinhold Niebuhr)



## RESUMO

De acordo com a Organização Mundial da Saúde (OMS), cerca de 17,5 milhões de pessoas morrem todos os anos vítimas de doenças cardiovasculares. O diagnóstico e tratamento precoce às pessoas com doenças cardiovasculares é fundamental para reduzir este número elevado de óbitos. Devido à importância da análise de Eletrocardiograma (ECG), que representa um dos principais exames para detecção destas doenças cardiovasculares, na presente dissertação é proposto um sistema remoto para análise automática de Eletrocardiograma (ECG)s baseado em *Web services* que pode auxiliar substancialmente o médico especialista, a partir de informações extraídas de ECGs, na emissão de laudos. Neste sentido, propõe-se um *Middleware* para abstrair as complexidades e facilitar a construção de ferramentas que permitam a análise automática de ECGs. Na arquitetura proposta, um servidor disponibiliza os algoritmos computacionais para identificação das principais ondas presentes nos ECGs tais como complexo QRS, ondas P e T, e suporte aos padrões abertos de exames HL7 aECG e DICOM-ECG. Para validar o *Middleware* proposto e contribuir com a comunidade médica, em especial do Hospital Universitário Walter Cantídio, desenvolve-se uma aplicação *Web* para geração de laudos de ECGs acessível remotamente. Testes foram realizados com a solução proposta, utilizando-se a base de dados MIT-BIH Arrhythmia Database (MITDB) para simular os pacientes do sistema e produzir os resultados apresentados nesta dissertação. A solução proposta atendeu aos requisitos estabelecidos e representa uma solução viável, financeiramente e tecnicamente.

**Palavras-chave:** Eletrocardiograma, *Web services*, *Middleware*, Sistemas de apoio ao Diagnóstico .

## ABSTRACT

According to the World Health Organization, about 17.5 million people die each year from cardiovascular disease. The early diagnosis and treatment is crucial to reduce this high number of deaths from cardiovascular diseases. Due to the importance of the electrocardiogram (ECG) analysis, which represents one of the main tests for the detection of these cardiovascular diseases, in the present dissertation is proposed a remote system for automatic ECGs analysis based on webservices that can substantially assist the doctor, using the extracted ECGs information, with the purpose to generate reports. Hence, a Middleware is proposed to abstract the complexities and to facilitate the construction of tools that allow the ECGs automatic analysis. The proposed architecture consists on a server which provides the computational algorithms to identify the main waves present in ECGs such as QRS complex, P and T waves, and support the open standards of HL7 aECG and DICOM-ECG. In order to validate the proposed Middleware and contribute to the medical community, especially the Walter Cantídio University Hospital, a Web application for remotely accessible ECG report was developed. The proposed solution was tested using the database MITDB to simulate the patients of the system and produce the results presented in this dissertation. The proposed solution met the established requirements which makes the proposed solution viable financially and technically.

**Keywords:** Electrocardiograms, Web services, Support system diagnosis, Free software.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Representação teórica dos sinais de um ECG. . . . .	22
Figura 2 – Morfologias do complexo QRS . . . . .	23
Figura 3 – Complexo QRS destacado no ECG. . . . .	23
Figura 4 – Vetor P representativo da ativação global dos átrios. . . . .	24
Figura 5 – Onda P destacada. . . . .	25
Figura 6 – Onda P destacada . . . . .	26
Figura 7 – Organização dos componentes do Middleware e modelo de arquitetura para a construção de aplicações . . . . .	33
Figura 8 – Casos de Uso implementados pelo <i>Middleware</i> . . . . .	36
Figura 9 – Diagrama de Classes do <i>Middleware</i> . . . . .	37
Figura 10 – Diagrama de atividades de requisições ao <i>Middleware</i> . . . . .	38
Figura 11 – Arquitetura da solução . . . . .	39
Figura 12 – Diagrama de Casos de Uso da solução . . . . .	44
Figura 13 – Diagrama de Classes da solução . . . . .	45
Figura 14 – Organização cliente-servidor da aplicação . . . . .	46
Figura 15 – Arquitetura da aplicação de análise de ECGs . . . . .	47
Figura 16 – Diagrama de Sequência para Autenticação . . . . .	48
Figura 17 – Página de login do sistema . . . . .	50
Figura 18 – Página inicial do médico listando os pacientes . . . . .	51
Figura 19 – Página inicial do médico mostrando os exames dos pacientes . . . . .	51
Figura 20 – Página inicial do paciente . . . . .	52
Figura 21 – Página de visualização de exames . . . . .	53
Figura 22 – Área da página de exame destacando a seleção de canais . . . . .	54
Figura 23 – Exibição de dois canais simultaneamente . . . . .	55
Figura 24 – Área da página do exame destacando a opção de selecionar marcação . . . . .	56
Figura 25 – Médico adicionando uma anotação . . . . .	56
Figura 26 – Página com gráfico de intervalos RR . . . . .	57
Figura 27 – Página para editar as marcações de um exame . . . . .	58
Figura 28 – Adicionando uma marcação . . . . .	58
Figura 29 – Gráfico: Tempo médio de respostas x Duração do exame . . . . .	59
Figura 30 – Gráfico: Tempo de resposta x Total de clientes . . . . .	60

Figura 31 – Gráfico: Tempo de resposta x Duração do exame . . . . .	61
Figura 32 – Gráfico: Tempo de resposta x Número de clientes . . . . .	61
Figura 33 – Exemplo de sistema distribuído . . . . .	69
Figura 34 – Exemplo de comunicação via <i>Simple Object Access Protocol</i> (SOAP) entre cliente e servidor . . . . .	74
Figura 35 – Interação entre cliente e servidor . . . . .	75
Figura 36 – Cliente-Servidor 2 níveis . . . . .	75
Figura 37 – Cliente-Servidor 3 níveis . . . . .	76
Figura 38 – Cliente-Servidor n níveis . . . . .	76

## LISTA DE TABELAS

Tabela 2 – Vantagens e desvantagens nos formatos: aECG e DICOM ECG . . . . .	18
Tabela 3 – Requisitos Funcionais e Não-Funcionais. . . . .	35
Tabela 4 – Requisitos Funcionais e Não-Funcionais . . . . .	43
Tabela 5 – Tecnologias de software utilizadas no <i>Middleware</i> e Aplicação Web . . . . .	46
Tabela 6 – Formas de transparência (TANENBAUM, 2007) . . . . .	71
Tabela 7 – Serviços relacionados a <i>Download/Upload</i> de exames . . . . .	77
Tabela 8 – Serviços para <i>Upload</i> e <i>Download</i> de exames . . . . .	78

## LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicity, Consistency, Isolation, Durability
ACR	<i>American College of Radiology</i>
aECG	<i>Annotated ECG</i>
ANSI	<i>American National Standards Institute</i>
ASP	<i>Active Server Page</i>
CSS	<i>Cascading Style Sheets</i>
CWT	<i>Continuous Wavelet Transform</i>
DDD	<i>Domain Driven Design</i>
DICOM	<i>Digital Imaging and Communications in Medicine</i>
ECG	Eletrocardiograma
FDA	<i>Food and Drug Administration</i>
FWT	<i>Fast Wavelet Transform</i>
HL7	<i>Health Level 7</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hyper Text Transport Protocol)</i>
HTTPS	<i>Hyper Text Transport Protocol Secure)</i>
IDL	<i>Interface Definition Language</i>
JSON	<i>JavaScript Object Notation</i>
MITDB	MIT-BIH Arrhythmia Database
MVC	<i>Model-View-Controller</i>
NEMA	<i>National Electrical Manufactureres Association</i>
PACS	<i>Picture Archiving and Communications Systems</i>
REST	<i>Representational State Transfer</i>
SNA	sistema nervoso autônomo
SOAP	<i>Simple Object Access Protocol</i>
SSL	<i>Secure Socket Layer</i>
UML	<i>Unified Modeling Language</i>
VFC	variabilidade da frequência cardíaca
WSDL	<i>Web Services Description Language</i>
XML	<i>Extensible Markup Language</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Objetivo Geral</b>	<b>19</b>
<b>1.1.1</b>	<i>Objetivo Específico</i>	<b>19</b>
<b>1.2</b>	<b>Produção Científica</b>	<b>19</b>
<b>1.3</b>	<b>Organização da Dissertação</b>	<b>19</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS</b>	<b>20</b>
<b>2.1</b>	<b>Eletrocardiograma</b>	<b>20</b>
<b>2.1.1</b>	<i>Complexo QRS</i>	<b>22</b>
<b>2.1.2</b>	<i>Onda P</i>	<b>24</b>
<b>2.1.3</b>	<i>Onda T</i>	<b>25</b>
<b>2.1.4</b>	<i>Formatos de Exames</i>	<b>25</b>
<b>2.1.4.1</b>	<i>DICOM-ECG</i>	<b>26</b>
<b>2.1.4.2</b>	<i>HL7 (aECG)</i>	<b>27</b>
<b>2.2</b>	<b>Trabalhos relacionados</b>	<b>28</b>
<b>2.3</b>	<b>Resumo do Capítulo</b>	<b>32</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>33</b>
<b>3.1</b>	<b>Projeto do <i>Middleware</i></b>	<b>34</b>
<b>3.1.1</b>	<i>Requisitos</i>	<b>34</b>
<b>3.1.2</b>	<i>Modelagem do sistema</i>	<b>35</b>
<b>3.1.3</b>	<i>Arquitetura</i>	<b>37</b>
<b>3.1.4</b>	<i>Implementação</i>	<b>38</b>
<b>3.1.4.1</b>	<i>Algoritmo para identificação do complexo QRS, ondas P e T</i>	<b>39</b>
<b>3.1.4.2</b>	<i>Integração com formato de exame HL7 (aECG)</i>	<b>40</b>
<b>3.1.4.3</b>	<i>Integração com formato de exame DICOM-ECG</i>	<b>41</b>
<b>3.2</b>	<b>Projeto do aplicativo para geração de laudos remotos</b>	<b>41</b>
<b>3.2.1</b>	<i>Requisitos</i>	<b>41</b>
<b>3.2.2</b>	<i>Modelagem do sistema</i>	<b>42</b>
<b>3.2.3</b>	<i>Arquitetura da Aplicação</i>	<b>42</b>
<b>3.3</b>	<b>Autenticação, autorização e segurança</b>	<b>44</b>
<b>3.4</b>	<b>Componentes de software utilizados</b>	<b>46</b>

3.5	Equipamentos utilizados . . . . .	46
3.6	Verificação e validação . . . . .	47
3.7	Resumo do capítulo . . . . .	48
4	<b>RESULTADOS . . . . .</b>	49
4.1	Visão geral . . . . .	49
4.1.1	<i>Perfis de usuários . . . . .</i>	49
4.2	Página de login . . . . .	50
4.2.1	<i>Página inicial do médico . . . . .</i>	50
4.2.2	<i>Página inicial do paciente . . . . .</i>	51
4.3	Página do exame de ECG . . . . .	52
4.3.1	<i>Visualização do traçado do exame . . . . .</i>	53
4.3.2	<i>Marcações . . . . .</i>	53
4.3.3	<i>Anotações . . . . .</i>	54
4.3.4	<i>Laudos . . . . .</i>	55
4.3.5	<i>Gráfico da série de intervalos RR . . . . .</i>	56
4.3.6	<i>Cadastro de marcações . . . . .</i>	57
4.4	Avaliação de desempenho do servidor . . . . .	58
4.4.1	<i>Serviço getQRSComplex . . . . .</i>	58
4.4.2	<i>Serviço uploadFile . . . . .</i>	60
4.5	Discussão dos resultados . . . . .	61
4.6	Considerações finais . . . . .	62
5	<b>CONTRIBUIÇÕES, CONCLUSÕES E TRABALHOS FUTUROS . . .</b>	63
5.1	Contribuições . . . . .	63
5.1.1	<i>Suporte a DICOM-ECG e aECG . . . . .</i>	63
5.1.2	<i>Detecção do Complexo QRS, ondas P e T . . . . .</i>	63
5.1.3	<i>Sistema Web para geração de laudos de ECGs online . . . . .</i>	64
5.2	Conclusões . . . . .	64
5.3	Trabalhos futuros . . . . .	65
	<b>REFERÊNCIAS . . . . .</b>	66
	<b>APÊNDICES . . . . .</b>	68
	<b>APÊNDICE A – SISTEMAS DISTRIBUÍDOS . . . . .</b>	68
A.1	Acesso a recursos . . . . .	69



<b>A.2</b>	<b>Sistemas abertos</b> . . . . .	70
<b>A.2.1</b>	<b>Transparência</b> . . . . .	70
<b>A.3</b>	<b>Web services</b> . . . . .	71
<b>A.3.1</b>	<b>REST</b> . . . . .	72
<b>A.3.2</b>	<b>SOAP</b> . . . . .	73
<b>A.4</b>	<b>Arquitetura Cliente-Servidor</b> . . . . .	74
	<b>APÊNDICE B – WEB SERVICES DISPONIBILIZADOS</b> . . . . .	77

## 1 INTRODUÇÃO

Segundo a Organização Mundial da Saúde (OMS), as doenças cardiovasculares representam uma das principais causas de morte no mundo. Para o diagnóstico destas doenças, o ECG é um dos principais exames não invasivos empregados pelos médicos. Este exame registra as atividades elétricas do coração, a partir da coleta de sinais elétricos na superfície do corpo. A partir deste exame é possível definir o número de batimentos por minuto (frequência cardíaca), além do ritmo do coração, identificando problemas como arritmias (alteração do ritmo normal), distúrbios de condução elétrica e eventuais sobrecargas das cavidades cardíacas. O registro do ECG permite um auxílio ao diagnóstico de diversas doenças cardíacas, tais como: arritmias, doenças congênitas e constatação de áreas infartadas.

Conforme afirmado há muitos anos por HOLTER (1961), a detecção de ondas típicas do eletrocardiograma (ECG), como as ondas P, Q, R, S e T, são importantes para ajudar na análise de ritmos do batimento cardíaco e, conseqüentemente, na identificação de possíveis anormalidades do coração.

A utilização de algoritmos computacionais, aplicando-se técnicas clássicas e modernas de processamento digital de sinais para segmentação e extração automática de parâmetros do sinal de ECG, auxiliam na identificação das ondas P, T e complexo QRS que são essenciais na análise de ECGs (MADEIRO, 2013). Porém, muitos desses algoritmos são desenvolvidos em linguagens de programação acadêmicas, por exemplo Matlab, tornando difícil sua utilização em ambientes profissionais. Transformar esses trabalhos, com objetivos apenas acadêmicos, para uma ferramenta real de auxílio ao diagnóstico por profissionais, fornecendo uma interface remota para sua utilização por outros sistemas, é uma das grandes motivações desta dissertação.

Um outro problema presente em soluções de análises de ECGs é a existência de diversos formatos de dados suportados pelos eletrocardiógrafos modernos, sendo em sua grande maioria proprietários. Devido a esta diversidade de padrões, pode-se assumir a não existência de um padrão definido, pelo menos para a maioria de marcas de equipamentos existentes no Brasil. Dessa forma, a interoperabilidade entre sistemas de análise de ECGs torna-se comprometida, sendo indispensável a aquisição de soluções de *hardware* e *software* fornecidos pela mesma empresa, aumentando o custo de implantação e manutenção. Isso cria dependência com os produtos fornecidos pela mesma empresa. Por isso, faz-se necessário a escolha de equipamentos que utilizem padrões abertos para extração dos sinais de ECGs. Dentre os padrões abertos disponíveis, são escolhidos para este trabalho, por serem os mais empregados: DICOM-WS 30

(conhecido como DICOM-ECG) e *Health Level 7 (HL7) Annotated ECG (aECG)*.

O padrão DICOM foi criado originalmente para armazenar e transmitir imagens de radiografias. Ao longo dos anos, este padrão tem sido ampliado para representar uma maior quantidade de modalidades de exames. Sua escolha, apesar de não estar presente numa grande quantidade de eletrocardiógrafos, deve-se especialmente ao apelo de adotar um formato único para todos os tipos de exames e fornecer suporte à análise de ECGs em Registros Eletrônicos de Saúde (RES) ou ainda em Registros Eletrônicos do Paciente (REP), os quais façam uso desse formato (VALENTE I. R. S., 2010).

Outro padrão de dados de registro médico usado para armazenar e recuperar dados de ECG de um paciente é o HL7 aECG. Assim como em outros formatos, o HL7 é baseado em *Extensible Markup Language (XML)* e representa um padrão aceito pelo *American National Standards Institute (ANSI)*. As principais vantagens e desvantagens presentes nesses formatos se encontram na Tabela 2 (BOND, 2011).

Tabela 2 – Vantagens e desvantagens nos formatos: aECG e DICOM ECG

<b>Formato</b>	<b>Vantagens</b>	<b>Desvantagens</b>
DICOM-ECG	Suporta vários tipos de exames; amplo suporte.	Complexo; depende de software para ler.
HL7 aECG	Possui ampla comunidade; padrão amplamente adotado; fácil leitura (XML).	Tamanho grande de arquivos.

Fonte – (BOND, 2011)

Esses padrões abertos possibilitam a comunicação entre os diversos equipamentos e demais sistemas ou ferramentas computacionais presentes nos hospitais ou clínicas médicas.

Nesta dissertação é proposta a integração desses algoritmos computacionais e o fornecimento de leitura/escrita nos formatos DICOM-ECG e HL7 aECG, como forma de prover interoperabilidade em soluções de análise de ECGs. Para resolver o problema de heterogeneidade, problema clássico presente em sistemas distribuídos, opta-se pela utilização de uma camada de software para fornecer as abstrações de programação, hardwares e sistemas operacionais (conceito de *Middleware*), fornecendo *Web services* para disponibilizar remotamente os recursos necessários na análise de ECGs. Também é fornecido um sistema Web para geração de laudos de ECGs remotamente, o qual permitirá ao Hospital Universitário Walter Cantídio uma maior eficiência de tempo para geração de seus laudos. Isto principalmente pela análise automática de ECGs que pode aumentar a produtividade e oferecer maior precisão nos diagnósticos com base

neste tipo de exame.

## **1.1 Objetivo Geral**

O objetivo desta dissertação é desenvolver uma ferramenta computacional capaz de analisar dados automaticamente de ECGs remotamente, a partir da utilização de tecnologias multiplataforma e de baixo custo.

### **1.1.1 Objetivo Específico**

Há um objetivo específico que vale destacar, o início da criação de uma ferramenta de auxílio ao diagnóstico para ser implantada no Hospital Walter Cantídio da UFC, com capacidade de gerar laudos digitais.

## **1.2 Produção Científica**

Como resultado dos métodos estudados e implementados, gerou-se até o momento o seguinte trabalho científico:

- I. Lima, M. A. Desenvolvimento de um sistema distribuído para análise de eletrocardiograma através de *web services*. XXV Congresso Brasileiro de Engenharia Biomédica, Foz do Iguaçu, p. 2061-2064. Out. 2016.

## **1.3 Organização da Dissertação**

O restante desta dissertação está organizada da seguinte forma: no capítulo 2 é apresentado uma breve contextualização de eletrocardiogramas, importante para a leitura deste trabalho e apresenta-se a revisão bibliográfica realizada a respeito dos principais pontos presentes na solução proposta. No Capítulo 3 descrevem-se as metodologias, apresentando as implementações propostas do *Middleware* e da aplicação Web para geração de laudos de ECGs. Os resultados do trabalho, mostrados através das páginas Web desenvolvidas e de uma análise de desempenho, são apresentados no Capítulo 4. Por fim, o Capítulo 5 elenca as conclusões deste trabalho, apresentando as principais contribuições e possíveis trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste Capítulo são apresentados os principais elementos conceituais teóricos desta dissertação que facilitam sobremaneira sua compreensão, bem como os trabalhos relacionados mais importantes encontrados até o momento.

### 2.1 Eletrocardiograma

Os primeiros registros dos sinais do coração datam por volta da metade do século XIX. Físicos, fisiologistas e médicos foram autores da descoberta dos eletrodos, item indispensável para verificar a diferença de potencial elétrico no corpo humano e produzir os sinais registrados de um eletrocardiograma até os dias atuais. A partir de testes em pequenos animais, foi possível averiguar a contratura muscular na presença de corrente elétrica, trabalho este feito por Kollicker e Muller. Posteriormente, em 1887, Waller, por meio do eletrômetro capilar – instrumento desenvolvido por Gabriel Lippmann – conseguiu medir o nível de corrente elétrica do corpo em relação ao coração (GINEFRA, 2007).

Já em 1901, Willem Einthoven conseguiu registrar com eficiência a corrente elétrica gerada pelo coração e a sua propagação pelo corpo humano. Para isso, o pesquisador utilizou pares de eletrodos bipolares para captar a variação da tensão e um galvanômetro de corda para os registros. Os eletrodos nada mais eram do que potes com uma solução salina concentrada, onde o paciente mergulhava as mãos e o pé esquerdo. Esta solução servia como condutor de eletricidade entre a pele e o metal do pote. As duas mãos formavam um par de eletrodo, a mão esquerda e o pé esquerdo formavam o outro par, e o último par era formado pelo braço direito e o pé esquerdo do paciente (GINEFRA, 2007).

Em conseqüente, Einthoven, fez algumas correções matemáticas no sinal registrado por Lippmann e introduziu o conceito dos pontos PQRST no sinal registrado. Em 1902, Einthoven consegue registrar o eletrocardiograma de um paciente no hospital universitário a uma distância superior a um quilômetro a partir de seu laboratório usando um cabo telefônico. Este evento, também histórico, Einthoven o chamou de "telecardiograma".

Em 1906, Einthoven publicou um trabalho que mostrava a compreensão sobre o funcionamento do coração e a variedade de anomalias. Após alguns anos, ele desenvolveu a corda de galvanômetro, a qual o permitiu descrever detalhes eletrocardiográficos de doenças acometidas pelo coração. Em 1924, seus anos de dedicação e estudos o consagraram com o

prêmio Nobel de Fisiologia ou Medicina. Desta época em diante, vários trabalhos relacionados com o estudo de eletrocardiogramas tornaram-se foco de estudos.

Atualmente, pode-se dizer que o equipamento responsável por registrar o exame eletrocardiograma é denominado eletrocardiógrafo. Este equipamento passou por várias mudanças desde sua primeira versão onde se tinha uma agulha ou estilete que desenhava um gráfico sobre um rolo de papel em movimento. O papel usado no exame do ECG é milimetrado e termossensível e que, normalmente, desloca-SE a uma velocidade constante de 25 mm/s. Dentro deste contexto, quando o estilete que registra o sinal do ECG não oscila, há um potencial zero, ou seja, é traçada uma linha reta chamada linha isoeétrica ou linha base. Quando há oscilação desta linha é porque a atividade elétrica está sendo captada e fazendo com que o estilete movimente-se para cima ou para baixo. O papel também é marcado a cada 5 mm tanto na horizontal quanto na vertical. Isto é usado para facilitar a visualização das características do exame (AZEVEDO, 1999).

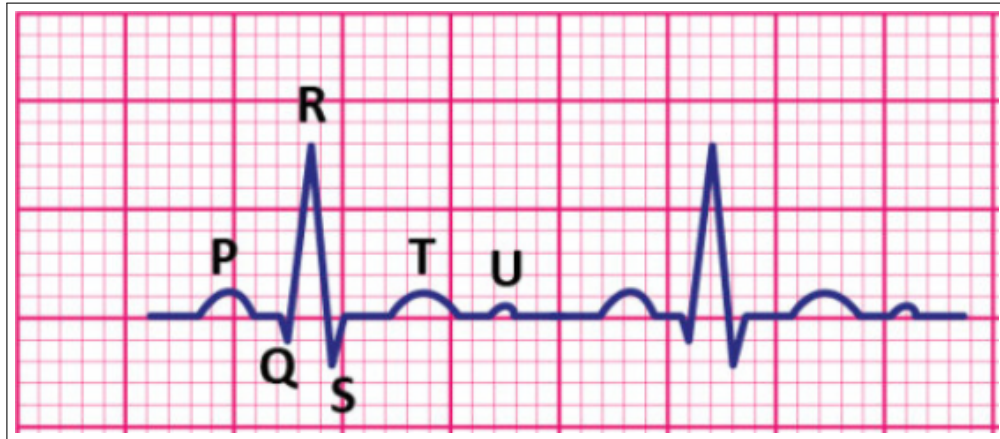
Os eletrocardiógrafos, atualmente, são encontrados com características digitais, dotados ou não de impressoras de alta resolução, saída para outros dispositivos, dentre outras características. Por fim, sabe-se que o eletrocardiograma é um exame no qual o paciente fica em repouso e, geralmente, coloca-se no paciente, em torno de doze elétrodos, nos membros inferiores, superiores, tórax e peito. Assim, é captada a atividade elétrica do coração e enviado ao equipamento que repassa este sinal, no formato de um gráfico impresso em um papel, o qual, posteriormente, serve para análise das sequências dos batimentos que pode caracterizar um tipo de doença (CURY *et al.*, 2011).

O sistema ECG padrão é constituído por 12 derivações que capturam a atividade elétrica do coração em 12 posições diferentes (6 derivações dos membros e 6 derivações pré-cordiais). Cada eletrodo captura a atividade elétrica do coração de um ângulo diferente e monitora porções específicas do coração (CATALANO, 2002).

A sequência de eventos repetitivos de contração (despolarização) e relaxamento (repolarização) dos músculos dos átrios e ventrículos, durante o ciclo cardíaco, gera eventos característicos. Tais eventos são identificados no ECG e denominados ondas P, Q, R, S e T, como pode ser visto na Figura 1. Considerando-se a forma de análise manual com medidas padronizadas de amplitude, em que 1 mm (quadrado menor) equivale a 0,1 mV (eixo Y), e tempo, em que 1 mm (quadrado menor) equivale a 0,04 s (eixo X), tais medidas padronizadas permitem a extração de componentes temporais, por exemplo duração do complexo QRS, e dessa forma

identificar de forma clara possíveis patologias.

Figura 1 – Representação teórica dos sinais de um ECG.



Fonte – Autoria própria.

Nas seções seguintes são descritos os componentes importantes no estudo e análise de ECGs: complexo QRS, ondas P e T.

### 2.1.1 Complexo QRS

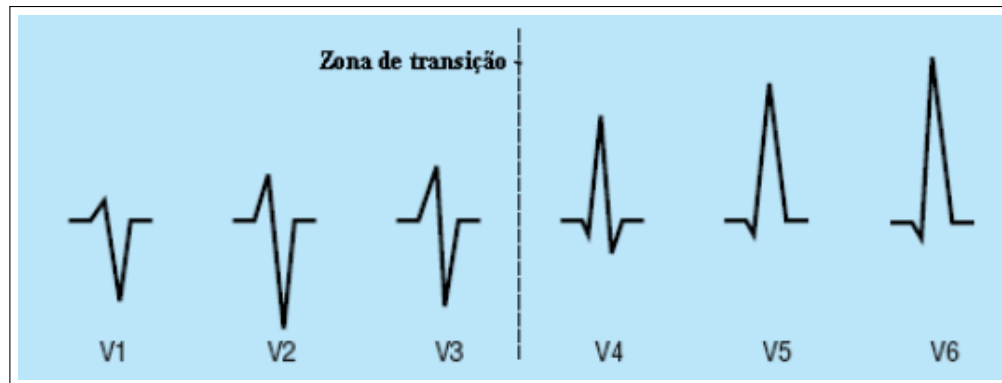
O complexo QRS compreende a contração (despolarização) dos ventrículos, contendo as seguintes características: duração, morfologia, amplitude e polaridade.

A duração do complexo QRS representa o período de tempo total em que ocorre a despolarização dos ventrículos, desde o início da onda Q até o final da onda S. Da mesma maneira que para a onda P, a duração do QRS tende a ser tanto maior quanto mais idoso é o indivíduo e quanto menor é a frequência cardíaca. Entretanto, a duração do QRS aumenta com a idade, mesmo sem modificação da frequência cardíaca, com os incrementos da superfície corporal, ou mais diretamente com os aumentos do tamanho do coração. A duração é maior em atletas e nos indivíduos que apresentam certo retardo na porção final do complexo QRS. Normalmente o complexo QRS se inscreve de 0,05 a 0,10 s com uma duração média de 0,07 s e um desvio padrão de 0,016 s. A duração de 0,11 s em adultos ou 0,09 s em crianças é um dado suspeito e aparece muito raramente em indivíduos normais, sendo mais frequentemente relacionada aos transtornos da condução intraventricular do estímulo ou a crescimento ventricular (OLIVEIRA *et al.*, 2009).

Quanto a morfologia, nas mais diversas derivações o complexo QRS apresenta-se de forma variável. A projeção, em uma dada derivação, dos três vetores principais do fenômeno de

ativação ventricular, produz deflexões de tamanho e polaridade variáveis, segundo as orientações espaciais da linha da derivação e de cada um dos vetores. A posição anatômica do coração influi no registro do complexo QRS normal (MORRIS *et al.*, 2009). Uma representação das morfologias do complexo QRS pode ser vista na Figura 2.

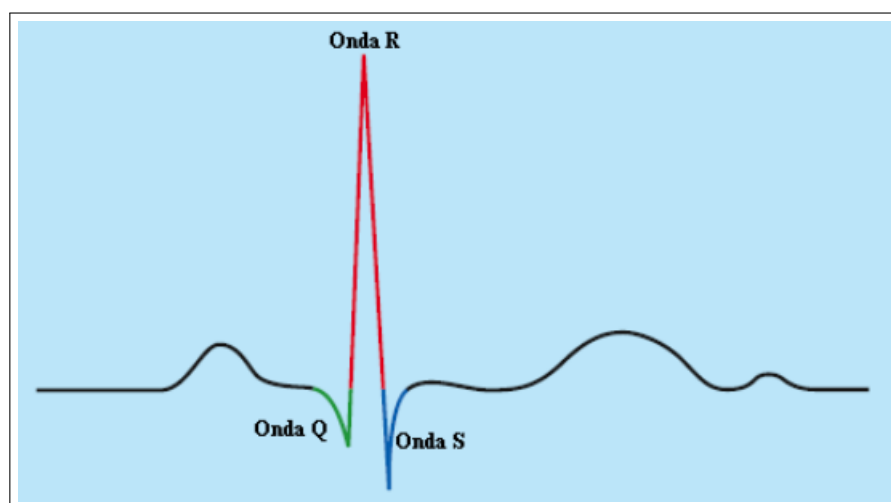
Figura 2 – Morfologias do complexo QRS



Fonte – (MORRIS *et al.*, 2009)

No que tange a amplitude, a tensão do complexo QRS varia muito e depende de condições cardíacas e extra cardíacas, como por exemplo, do meio condutor em torno do coração. Nas crianças de parede torácica delgada, a tensão é relativamente ampla. Já a onda R que possui, em geral, grande amplitude (2 a 3 mV) pode ser registrada normalmente em D2, sendo que a hipertrofia ventricular acarreta, habitualmente, aumento da tensão em D1 e D3 (OLIVEIRA *et al.*, 2009). A Figura 3 representa o complexo QRS marcado num gráfico ECG ideal.

Figura 3 – Complexo QRS destacado no ECG.



Fonte – (MORRIS *et al.*, 2009)



### 2.1.2 Onda P

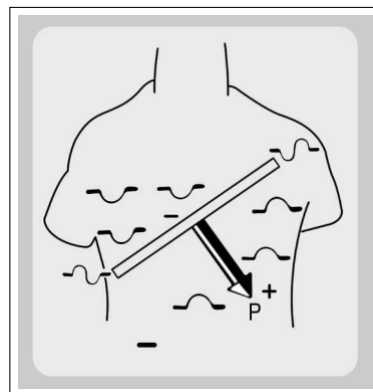
A onda P representa a despolarização atrial e sua duração caracteriza o tempo de excitação total dos átrios e varia com a idade do indivíduo e com a frequência cardíaca. A duração da onda P tende a ser maior quanto maior for a idade, variando de 0,06 a 0,09s nas crianças e 0,08 a 0,11s nos adultos e é tanto menor quanto maior é a frequência cardíaca (LENGYEL, 1974).

A onda P normal é arredondada, monofásica e de inscrição lenta. Nos indivíduos de pouca idade e nos aumentos de frequência cardíaca, pode-se observar as ondas P pontiagudas, mas com tensão ainda dentro dos limites da normalidade. Nos recém-nascidos a onda P, frequentemente pontiaguda, tem como característica normal ser assimétrica, com o ramo ascendente mais lento do que o descendente (OLIVEIRA *et al.*, 2009).

A tensão máxima da onda P em indivíduos normais é considerada entre 0,25 a 0,30 mV, medida em D2. Dependendo da orientação do vetor de ativação atrial, podemos ter onda P isoelétrica em uma dada derivação. Em casos de taquicardia a tensão de P aumenta, não ultrapassando, contudo, o limite máximo normal (OLIVEIRA *et al.*, 2009; LENGYEL, 1974)

Quanto à polaridade, as ondas P na superfície corporal podem ser positivas, negativas ou bifásicas, dependendo da orientação do vetor que representa a ativação global dos átrios. Uma ilustração da orientação do vetor e as ondas na superfície corpórea, pode ser vista na Figura 4.

Figura 4 – Vetor P representativo da ativação global dos átrios.

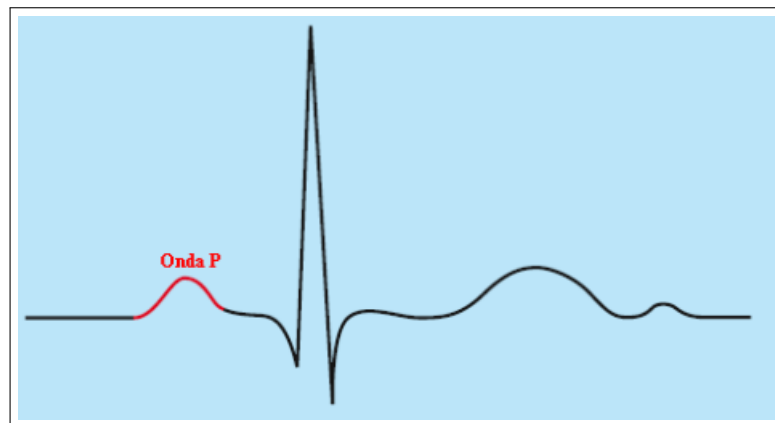


Fonte – (OLIVEIRA *et al.*, 2009)

A orientação resultante da ativação atrial normal aponta para baixo e para a esquerda, com pequena ou nenhuma inclinação para frente ou para trás. Por esta razão, as ondas P são normalmente negativas nas regiões superiores e direitas do tórax, tanto na fase anterior como na

posterior, e positivas nas regiões inferiores esquerdas. Entre estas duas zonas de positividade e negatividade, existe uma estreita faixa onde se registram ondas P bifásicas demarcando o plano zero, que é perpendicular à orientação espacial do vetor resultante P. Geralmente este plano passa ao nível de derivação precordial V1, onde são registradas, em geral, ondas P bifásicas. A Figura 5 mostra a onda P destacada.

Figura 5 – Onda P destacada.



Fonte – (MORRIS *et al.*, 2009)

### 2.1.3 Onda T

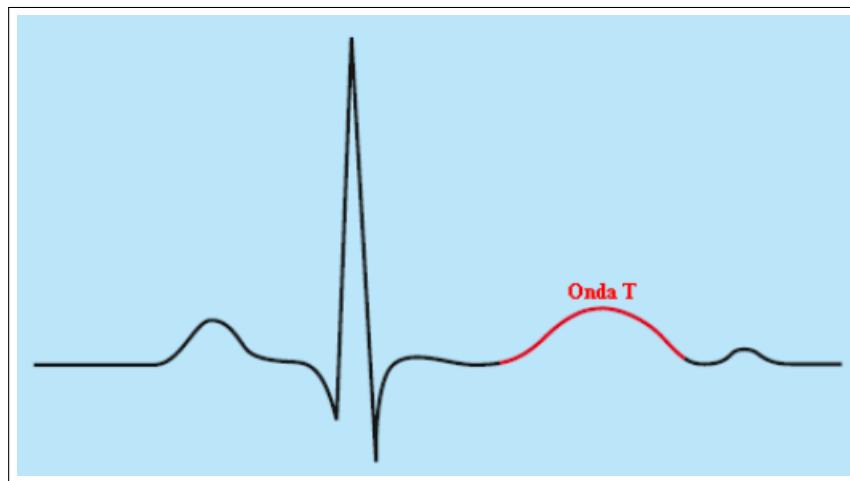
A onda T representa, no eletrocardiograma, a repolarização ventricular. Geralmente o registro é de uma deflexão arredondada e lenta, habitualmente com polaridade igual a do complexo QRS. Normalmente, a onda T é assimétrica, com ramo ascendente lento e descendente com maior inclinação (OLIVEIRA *et al.*, 2009).

Essa onda possui polaridade semelhante a do complexo QRS, sendo positiva nas três derivações clássicas (derivações bipolares D1, D2 e D3), como também nas derivações unipolares aVF e aVL, sendo sempre negativa em aVR (OLIVEIRA *et al.*, 2009). Na Figura 6 se observa a onda T destacada num sinal de ECG.

### 2.1.4 Formatos de Exames

A grande maioria dos eletrocardiógrafos possuem formatos proprietários para armazenar os exames ECGs, embora diversos fabricantes já estão aderindo aos formatos abertos. Este fato tem criado dificuldades para os pesquisadores no estudo e análise destes exames, bem como na criação de bases de dados efetivas para suas pesquisas, tornando-as mais confiáveis

Figura 6 – Onda P destacada



Fonte – (MORRIS *et al.*, 2009)

por avaliar uma maior quantidade de exames. Além disso, facilitaria a geração de protocolos médicos que auxiliassem nos diagnósticos de algumas doenças cardíacas.

A interoperabilidade de ECGs entre sistemas heterogêneos tem sido simplificada por não somente uma, mas por uma quantidade razoável de formatos abertos de exames (especificação bem documentada). Através da adoção desses padrões abertos é possível que um mesmo exame possa ser visto/analísado por várias instituições de saúde, sem que necessariamente possuam os mesmos equipamentos e *softwares*. Desta forma, cria-se um ambiente heterogêneo para análise de ECGs.

Os principais padrões abertos para os exames ECGs são: *Digital Imaging and Communications in Medicine* (DICOM) (formato binário) e HL7 (formato XML). Ambos os padrões são considerados modernos e amplamente utilizados pela indústria. O acréscimo de outros formatos abertos estão previstos para futuras atualizações do sistema. As seções seguintes apresentam os formatos aECG e DICOM-WS 30 (conhecido por DICOM-ECG) (BOND, 2011).

#### 2.1.4.1 DICOM-ECG

Os últimos 50 anos presenciaram o surgimento de uma série de novas e complexas modalidades de diagnóstico por imagem introduzidas pelos departamentos radiológicos ao redor do mundo. Tomografia computadorizada e ressonância magnética, são exemplos dessas modalidades que surgiram. Conseqüentemente, identificou-se a necessidade de um formato padrão de imagem digital para armazenamento físico, permitindo sua transmissão e manipulação por vários dispositivos heterogêneos (impressoras, *scanners* e servidores) e *Picture Archiving*

*and Communications Systems (PACS)*. Como resultado, a cooperação entre a *National Electrical Manufacturers Association (NEMA)* e a *American College of Radiology (ACR)* foi formada para desenvolver o padrão ACR-NEMA em 1985. Este formato passou a ser conhecido como DICOM versão três em 1993 e finalmente tornou-se um padrão Europeu em 1995 (BOND, 2011).

Apesar do padrão DICOM, originalmente, ter sido pensado para armazenamento e transmissão de imagens radiográficas, sua utilização tem crescido tanto que passou a suportar todos os tipos de exames. Como resultado, NEMA tem ampliado o formato DICOM desenvolvendo e publicando documentos suplementares. No ano 2000, foi introduzido o documento complementar DICOM-WS 30 que adicionava ao formato DICOM o suporte às formas de ondas presentes nos exames de ECGs.

Com o acréscimo de vários tipos de exames ao formato DICOM, foi necessário sua especificação ser dividida em grupos de trabalho. O grupo de trabalho 1, conhecido como *Cardiac and Vascular Information*, ficou responsável por desenvolver e manter este complemento. O grupo é composto por representantes de várias empresas, ligadas a área médica e que fabricam equipamentos e software, entre estas empresas estão Siemens, Philips e Ge Healthcare (NEMA, 2017).

#### 2.1.4.2 HL7 (aECG)

Um outro padrão também aberto foi especificado e desenvolvido em 2001, denominado de HL7 (aECG). Em 19 de Novembro desse ano, a USA *Food and Drug Administration (FDA)* organizou uma reunião pública intitulada de "Electronic Interchange Standard for Digital ECG and Similar Data", na qual apresentou os planos da agência para a criação de um formato digital padrão que deveria ser utilizado em vez dos formatos proprietários dos fabricantes de electrocardiógrafos. Como suporte a sua estrutura de dados foi proposto um formato de arquivo XML (MAIA, 2008).

Em 2002, frente à necessidade de especificar como esses dados devem ser codificados, a agência reuniu vários consultores de forma a avaliar os formatos existentes e a definir requisitos para o novo formato. O grupo concluiu que os padrões de ECG existentes estavam essencialmente orientados à codificação das formas de onda recolhidas pelos electrocardiógrafos e a informação demográfica do paciente, bem como apresentou ainda um conjunto de indicativos que mostravam que não só era necessário ter as formas de onda para visualização, como também de ver como

essas formas de onda devem ser usadas na avaliação da segurança cardíaca. Assim, era necessário criar um novo padrão que não apenas com as informações das formas de ondas além de um conjunto de anotações, contendo informações relevantes ao estudo do coração. São exemplos de anotações: batimento, onda, ritmo, e ruído, etc. .

Uma das grandes vantagens desse formato é ter sido implementado em XML, tornando o arquivo do exame humanamente legível. Desta forma, exames no formato aECG podem ser abertos e modificados através de meros editores de texto, sem a necessidade de editores binários para estas ações. Como esse padrão utiliza XML, pode-se validar um arquivo do exame através do *XML Schema*, evitando assim que arquivos inválidos sejam lidos ou transmitidos (BOND, 2011).

## 2.2 Trabalhos relacionados

Nesta seção apresenta-se uma revisão de literatura de algumas soluções propostas para aplicações de ECG, fundamentando assim a pesquisa desta dissertação. Para um melhor entendimento do leitor, esta seção está dividido nos seguintes itens: (i) formatos de exames; (ii) algoritmos para identificação dos pontos fiduciais em um exame, tais como complexo QRS, ondas P e T, (iii) softwares para análise de ECGs e (IV) segurança da informação médica.

A integração dos sistemas de informação nos mais diversos centros de saúde vem ganhando importância nos últimos anos. Em especial, nas organizações de prestação de cuidados com a saúde, existem várias ferramentas com diferentes finalidades que devem/necessitam ser integradas, aos sistemas existentes em hospitais/clínicas médicas, para melhor atender ao propósito médico. De forma ainda mais específica, os exames de ECG tem sido alvo de atenção destes sistemas uma vez que, dado a heterogeneidade de formatos e padrões nestes exames, há uma maior dificuldade na leitura destes por diferentes sistemas.

Diante deste contexto, é necessário então identificar as similaridades e diferenças entre as formatações e padronizações utilizadas em exames de ECG, para melhor projetar como armazenar os dados de forma que permitam uma melhor interoperabilidade entre os variados sistemas.

No trabalho desenvolvido por Bond (2011) é realizado um estudo com 9 formatos de ECG, são eles: SCP-ECG, DICOM-ECG, HL7 aECG, ecgML, MFER, Philips XML, XML-ECG, mECGml e ecgAware. Faz-se uma comparação entre cada um deles, permitindo assim identificar suas vantagens e desvantagens.

Neste trabalho, o autor conclui que a grande quantidade de formatos abertos de ECG dificulta a interação entre os sistemas e acaba por elevar a complexidade de cada um deles. Além disto, é especulado que a representação utilizando a padronização do tipo XML ultrapasse a do tipo binária como método de implementação preferido utilizado na codificação de dados de ECG. Justifica-se isto devido ao impacto causado no setor de saúde desde a chegada dos primeiros eletrocardiógrafos digitais utilizando arquivos XML para codificar os dados de exames. Adicionalmente, o autor identifica que, devido à existência de várias modalidades de exames ECG e cada um com suas respectivas particularidades a serem armazenadas, constitui-se um impedimento a existência de um formato que atenda a todas as necessidades de seus clientes. Desta forma, para que um padrão possa ser utilizado nas mais diversas modalidades de exames é desejável prover o suporte a cada uma delas.

Segundo Kumar *et al.* (2013) o padrão DICOM-ECG é o único que busca suportar todas as modalidades de diagnósticos por imagem, tais como Tomografia Computadorizada, Ressonância Magnética e também ECG que são algumas das modalidades suportadas por este formato. Isto reduz a necessidade de utilizar vários formatos para armazenar dados médicos. Nesse sentido, o DICOM-ECG representa o padrão aberto mais promissor dentre os analisados ao longo do trabalho citado.

De acordo com Ettinger *et al.* (2008) para existir interoperabilidade de exames ECGs entre centros médicos é necessário o suporte aos padrões abertos: SCP-ECG, HL7 aECG e DICOM-ECG. Em seu trabalho o autor fornece um *Toolkit* que permite converter exames de ECG para os três formatos supracitados anteriormente. Outro ponto a destacar foram as constatações que o formato SCP-ECG ocupa menos espaço para armazenamento e menor tempo para transmissão pela rede, representando uma boa opção como padrão de exame ECG. Como ponto negativo, o *Toolkit* fornecido pelo autor foi feito em linguagem C, sendo portanto, dependente de plataforma e de sistema operacional proprietário, dificultado sua utilização em sistemas baseados em soluções livres.

A partir dos dados armazenados é possível transmitir informações entre sistemas computacionais promovendo interoperabilidade entre componentes de software, possibilitando, por exemplo, que soluções distribuídas contribuam colaborativamente na análise de exames. Por ser o exame mais utilizado no diagnóstico de doenças cardíacas e monitoramento de pacientes, sua análise, especialmente a detecção dos pontos fiduciais (complexo QRS, onda P e T) que correspondem a diferentes fases de contração cardíaca, ajudam a identificar doenças

cardiovasculares (LEUTHEUSER *et al.*, 2016).

Dado que esta dissertação visa contribuir como uma ferramenta *online* para análise de ECGs, fez-se necessário buscar na literatura algoritmos consagrados para a identificação de parâmetros nestes exames, ou seja, algoritmos para realização da detecção e identificação automáticas destes parâmetros. Em seu trabalho, Leutheuser *et al.* (2016) é feita uma revisão de três algoritmos para detecção automática desses pontos fiduciais, disponibilizando uma versão melhorada para cada um desses algoritmos. Para validar a implementação dos algoritmos revisados e de cada novo proposto, os autores utilizam as bases de dados do MITDB e Laguna *et al.* (1997), ao final da avaliação destes algoritmos, seleciona-se aquele com taxa de erro aceitável em relação ao baixo tempo de processamento necessário à identificação desses pontos.

Já no trabalho Madeiro (2013), o autor utilizou as transformadas de *Wavelet* e de *Hilbert* para detecção multicanal e identificação desses pontos fiduciais. Os resultados experimentais mostram que o algoritmo desenvolvido possui uma taxa de erro baixa ( $\pm 0,2\%$ ). Para validar o algoritmo o autor utilizou a base de dados MITDB.

As estratégias utilizadas por Madeiro (2013) e Leutheuser *et al.* (2016) levaram a relações de assertividade no que tange a detecção dos pontos fiduciais a um melhor resultado no trabalho de Madeiro (2013), todavia a um custo de desempenho de 2 segundos para cada 30 minutos de exame de ECG. Já em Leutheuser *et al.* (2016), percebe-se a relação inversa, tem-se um melhor desempenho a um custo de uma assertividade menor na detecção das marcações.

Há muitas pesquisas feitas sobre o tema da telemedicina desde os anos 90 (ARMSTRONG; HASTON, 1997). Muitas delas propõem sistemas baseados em tecnologias Web para prover soluções de tempo real, permitindo o armazenamento de informações e sua distribuição através da Internet. Com esse modelo é possível criar aplicações independentes de sistema operacionais ou até mesmo da arquitetura do processador. Isto faz com que o *browser* e internet sejam os elementos imprescindíveis para utilização dessas aplicações.

O trabalho realizado por Kam *et al.* (2014) propõe um software baseado em Web chamado de *VirtualDave System*. A solução é composta de dois dispositivos médicos adquiridos comercialmente, o *CP 150 Resting ECG* e o *PC-based Resting ECG*, e um software Web desenvolvido pelo autor para visualização desses dados.

Na sua aplicação Web, o autor opta por utilizar *HyperText Markup Language* (HTML)5, como componente estático e *Active Server Page* (ASP) combinado com C# para fornecer os aspectos dinâmicos necessários em aplicações Web. Quanto ao formato dos exames

utilizados deu suporte apenas ao fornecido pelo hardware adquirido, portanto proprietário. Assim, caso um novo equipamento seja utilizado e siga um formato diferente, uma nova camada de software faz-se necessário para garantir a heterogeneidade desta solução. Nesta aplicação, *VirtualDave System*, os profissionais de saúde e pacientes possuem acesso aos exames. Na perspectiva do paciente, esta funcionalidade permite aos pacientes visualizar e disponibilizar seus exames para médicos fora do hospital onde foi atendido, mantendo um histórico dos exames realizados.

De forma semelhante, Kumar *et al.* (2013) desenvolve uma aplicação cliente-servidor com 3 camadas utilizando as tecnologias HTML5, *Javascript* e *Cascading Style Sheets* (CSS) na camada de apresentação. Para a camada de lógica utiliza-se o *Ruby on Rails*, o qual emprega a linguagem de *script Ruby*. Como diferencial de seu trabalho, o autor destaca o fato de ter utilizado tecnologias livres, apresentando portanto, uma solução de baixo custo. Um outro ponto importante para destacar em relação a Kam *et al.* (2014), é o fato do eletrocardiógrafo utilizado usar o formato de armazenamento aberto Philips XML, simplificando a decodificação dos dados para que outras soluções de software possam trabalhar colaborativamente.

O advento da comunicação entre sistemas computacionais permite que provedores de serviços de saúde possam monitorar remotamente parâmetros vitais de pacientes e produzir uma grande quantidade de dados que precisam ser armazenados e gerenciados para uso futuro. Em sistemas de saúde, a privacidade dos dados manipulados deve ser assegurada. Portanto, faz-se necessário a utilização de técnicas confiáveis de autenticação, autorização e de transporte dos dados. Nesse sentido, os autores Emerson *et al.* (2015) propõem em seu trabalho a utilização do protocolo aberto OAuth 2.0 (especificação RFC6749), combinado com a autenticação do usuário para permitir o acesso aos recursos. Já para os autores Solapurkar (2016), a utilização do OAuth 2.0 apresenta vulnerabilidade ao precisar armazenar localmente o *refresh token*, o qual é necessário para a criação de um *token* de acesso. Assim, em seu trabalho propõe a utilização em conjunto dos protocolos OAuth 2.0 e *JSON Web Token* (JWT), não sendo mais necessário o armazenamento dos *refresh tokens*. Quanto ao tráfego de informações, ambos os trabalhos adotam o certificado SSL para assegurar que os dados transmitidos sejam criptografados.

A partir da análise dos trabalhos relacionados, acima citados, destacam-se no presente trabalho o fornecimento de recursos computacionais para extração de informações que auxiliam profissionais de saúde na geração dos laudos de ECG. Outro ponto a destacar, é o suporte aos dois principais padrões abertos de exames (DICOM-ECG e HL7 aECG), permitindo a integração



com sistemas que trabalhem com estes formatos. Diferentemente do *VirtualDave System*, que apenas fornece a autenticação como método de segurança da informação, o presente trabalho utiliza o protocolo OAuth 2.0 (autorização e autenticação) e o protocolo SSL (transporte de dados) fornecendo uma solução bem mais segura e em conformidade com trabalhos identificados na comunidade científica (EMERSON *et al.*, 2015; SOLAPURKAR, 2016).

### **2.3 Resumo do Capítulo**

Neste Capítulo foram apresentados alguns conceitos necessários para a compreensão do desenvolvimento desta Dissertação. Inicialmente, foram mostrados os conceitos básicos do exame de ECG. Ao final deste Capítulo mostra-se uma breve explicação a respeito dos formatos de exame. Os conceitos básicos de sistemas distribuídos, em especial, arquitetura cliente/servidor e *Web services*, estão disponíveis no Apêndice A.

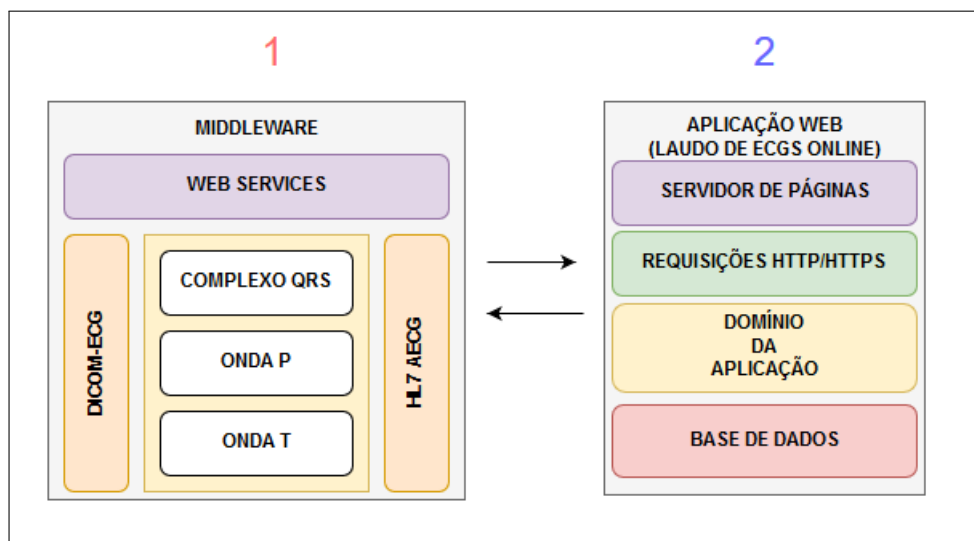
A partir dos conceitos teóricos e dos trabalhos relacionados ao desta dissertação, devem ser descritas as metodologias utilizadas durante o desenvolvimento (*Middleware*) e da aplicação WEB, ambos desenvolvidos para auxiliarem profissionais da saúde no diagnóstico de patologias presentes nos exames de ECG.

### 3 METODOLOGIA

Neste Capítulo são apresentados os componentes computacionais utilizados para o desenvolvimento desta dissertação, a saber: os projetos do *Middleware* e do *software* para geração de laudos de ECGs *online*, compondo em conjunto um sistema para análise de ECGs.

Como pode ser visto na Figura 7, o *Middleware* (item 1) representa, no presente trabalho, a camada de serviços que abstraem a implementação da solução. Nele são encapsuladas todas as complexidades presentes na extração de informações dos ECGs e suporte aos formatos padrões de exames. Para fornecer esses recursos remotamente a outras aplicações fez-se uso de *Web services*. Por outro lado, a aplicação para geração de laudos (item 2) concentra o domínio da aplicação, isto é, toda a lógica de negócios presente na aplicação desenvolvida com o uso do *middleware*. Na Figura 7, apesar de se destacar a comunicação desta aplicação com o *Middleware*, é possível utilizar outros sistemas para fornecer o mesmo conjunto de recursos ou até mesmo outros para atender o domínio da aplicação desenvolvida.

Figura 7 – Organização dos componentes do Middleware e modelo de arquitetura para a construção de aplicações



Fonte – Autoria própria.

Nas próximas seções, descrevem-se cada um desses projetos, apresentando detalhadamente suas construções.

### 3.1 Projeto do *Middleware*

Na revisão do estado da arte da literatura, no que tange aos algoritmos para extração de informações de ECGs, e no estudo dos formatos de exames existentes no mercado, observa-se a inexistência de bibliotecas ou *Web services* que forneçam esta solução integrada. Os componentes de *software* encontrados, na maioria das vezes, ou estão incompletos ou, para utilizá-los, são necessárias diversas adaptações. Devido a este problema, no início do desenvolvimento do sistema foi implementado um servidor de aplicações para concentrar um conjunto de serviços, fornecendo uma camada bem definida de entradas e saídas garantindo assim uma transparência para os sistemas solicitantes.

Para aumentar a interoperabilidade, permitir heterogeneidade e fornecer uma maior transparência para os mais diversos sistemas existentes, opta-se por utilizar *Web services* como forma de exportar as funcionalidades cobertas pelos requisitos desta solução. Os principais conceitos de sistemas distribuídos empregados neste trabalho são apresentados no Apêndice A.

#### 3.1.1 *Requisitos*

A etapa de levantamento e análise de requisitos tem como objetivo principal definir o grupo de requisitos funcionais e não funcionais do software a ser desenvolvido.

Com base nos conceitos de requisitos e nas análises das expectativas a serem atendidas com a solução final, identificam-se as funcionalidades desejadas, dado um intervalo de sinais:

- identificar os complexos QRS;
- identificar as ondas P;
- identificar as ondas T;
- calcular os batimentos cardíacos;
- calcular a duração média do complexo QRS;
- calcular a duração média das ondas P; e
- calcular a duração média das ondas T.

Os itens elencados foram identificados a partir de análise crítica dos pontos fortes e fracos presentes nas aplicações citadas no Capítulo 2. A Tabela 3 representa os requisitos funcionais e não funcionais extraídos do levantamento inicial de requisitos. Para diferenciar entre os tipos de requisitos, no presente trabalho, adotou-se a nomenclatura: RF\_ (*Requisito*

*Funcional*) e RNF\_ (*Requisito Não-Funcional*). Para identificar a prioridade, utilizou-se os termos: BAIXA, MÉDIA e ALTA. Um requisito considerado de BAIXA prioridade é aquele que não sendo implementado não representa impacto à solução final (opcional). Para requisitos considerados MÉDIO são recursos obrigatórios apenas na versão final. Já para itens com prioridade ALTA desde o protótipo inicial até a versão final devem contemplá-los.

Tabela 3 – Requisitos Funcionais e Não-Funcionais.

Requisitos	Descrição	Prioridade
RF_0001	Serviço para identificar os complexos QRS	ALTA
RF_0002	Serviço para identificar as ondas P	MÉDIA
RF_0003	Serviço para identificar as ondas T	MÉDIA
RF_0004	Serviço para calcular os batimentos cardíacos	ALTA
RF_0005	Serviço para plotagem do gráfico de intervalos RR	ALTA
RF_0006	Serviço para cálculo da duração média do complexo QRS	ALTA
RF_0007	Serviço para cálculo da duração média da onda T	MÉDIA
RF_0008	Serviço para cálculo da duração média da onda P	MÉDIA
RF_0009	Fornecer cada um dos serviços por intervalo de sinais para reduzir o tempo de processamento	ALTA
RNF_0010	Garantir que somente usuários cadastrados acessam os serviços	ALTA
RNF_0011	Gravar <i>Logs</i> de todas as atividades	BAIXA
RNF_0012	Toda informação de entrada e saída seja feita em JSON	ALTA
RNF_0013	Serviço para calcular os batimentos cardíacos	ALTA
RNF_0014	Utilizar servidor de aplicações gratuito e robusto	ALTA

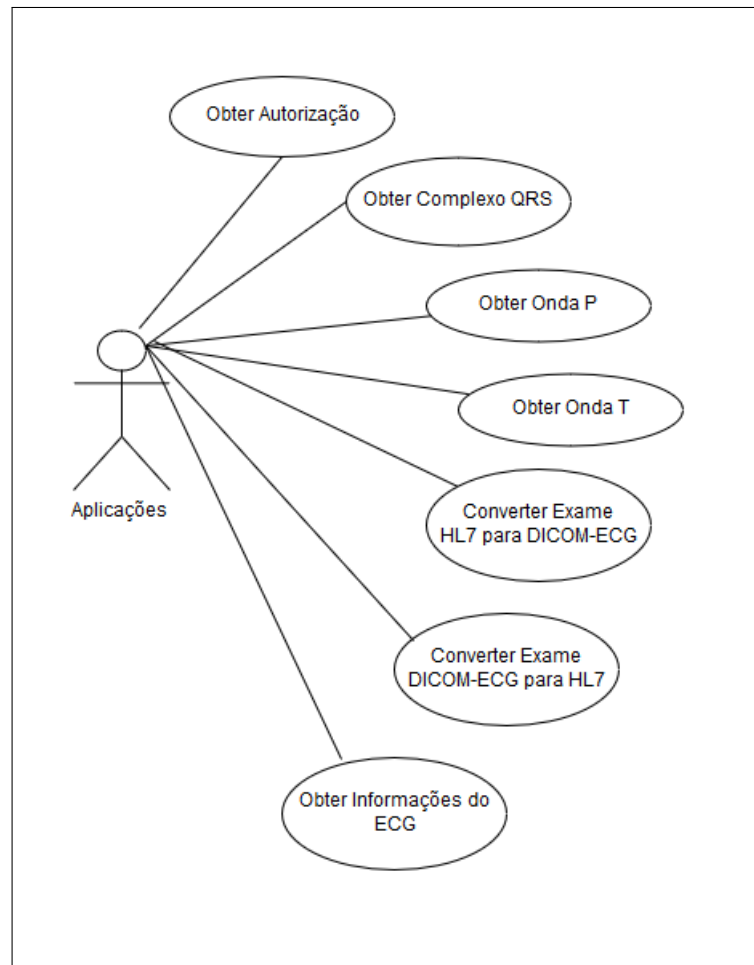
Fonte – Autoria própria.

### 3.1.2 Modelagem do sistema

Após avaliar os requisitos do sistema, foram identificados os principais casos de uso da solução. Estes artefatos, conforme mostrado na Figura 8, apresentam uma visão funcional de uma solução a ser desenvolvida com o uso do middleware.

Com os casos de uso identificados, iniciou-se a etapa de modelagem das classes do sistema. Uma particularidade desta solução, é o fato de ela conter apenas como classes de negócio os dados de autorização do sistema, os algoritmos para extração de informações dos exames e as classes que envolvem a conversão entre os formatos dos exames suportados. As demais classes são apenas objetos utilizados para serialização e deserialização durante a troca de informações entre os sistemas comunicantes, não sendo, portanto, levados em consideração no diagrama apresentado na Figura 9. Para representar os serviços disponibilizados utilizou-se, nesse

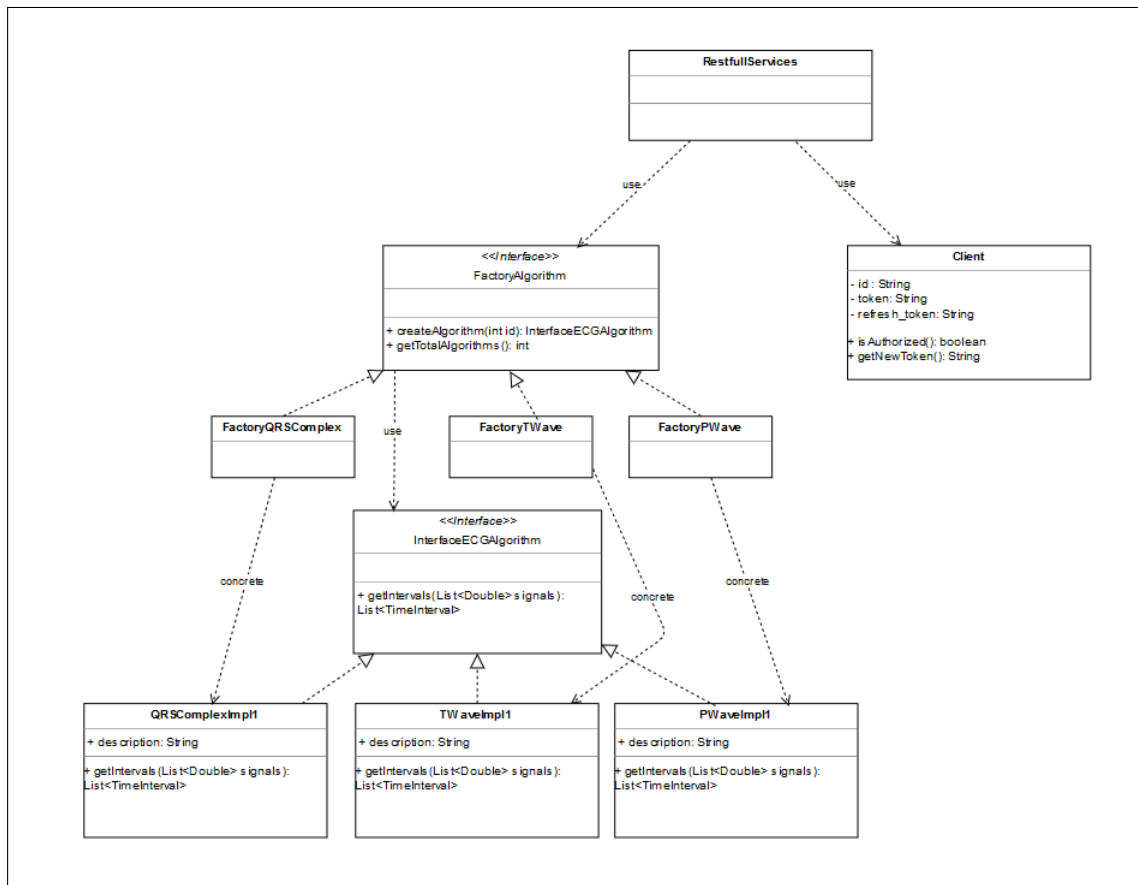
Figura 8 – Casos de Uso implementados pelo *Middleware*



Fonte – Autoria própria.

diagrama de classes, o componente *RestfullServices*. Este componente representa, no modelo, as classes controladoras (*Controller*), as quais recebem as requisições, processa-nas e produz em resposta aos clientes participantes. Todos os serviços disponibilizados pelo *Middleware* podem ser vistos no Apêndice B.

Para representar o processo de troca de mensagem entre os clientes (aplicações externas) e o *Middleware*, faz-se uso do diagrama de atividades, conforme visto na Figura 10. Para enviar uma requisição válida, é necessário, inicialmente, obter um *TOKEN* de acesso. A partir de então, o cliente pode enviar requisições, como descritas no Apêndice B, para o *Middleware* e obter as informações desejadas. Os detalhes do processo de autenticação e obtenção do *TOKEN* são apresentados na seção 3.3.

Figura 9 – Diagrama de Classes do *Middleware*

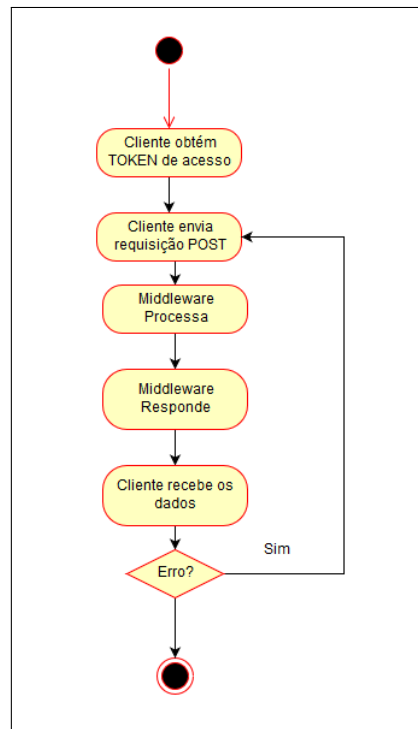
Fonte – Autoria própria.

### 3.1.3 Arquitetura

A Figura 11 representa um esboço da arquitetura desenvolvida. Sua concepção tem como inspiração o estilo arquitetural de N-camadas e, na proposta apresentada, toda a lógica de realização inerente aos serviços expostos pelo middleware fica concentrada no servidor, deixando para os clientes apenas a exibição dos dados processados. Internamente ao servidor, utilizou-se o padrão *Model-View-Controller* (MVC) para receber as requisições *Hyper Text Transport Protocol* (HTTP)/*Hyper Text Transport Protocol Secure* (HTTPS) (*Controller*), processá-las (*Model* e *Controller*) e retornar uma resposta. Apesar de utilizar o padrão de projeto MVC, deve ficar claro para o leitor que a camada de visualização (*View*) não possui relevância neste sistema, pois, não apresentam-se os dados processados diretamente para os clientes do sistema.

Como optou-se pela utilização de *Web services* para fornecer recursos a outros sistemas (*Sistemas Computacionais*), a arquitetura proposta apresenta um mecanismo de comunicação com a Internet que permite que qualquer software externo possa atuar como um cliente

Figura 10 – Diagrama de atividades de requisições ao *Middleware*



Fonte – Autoria própria.

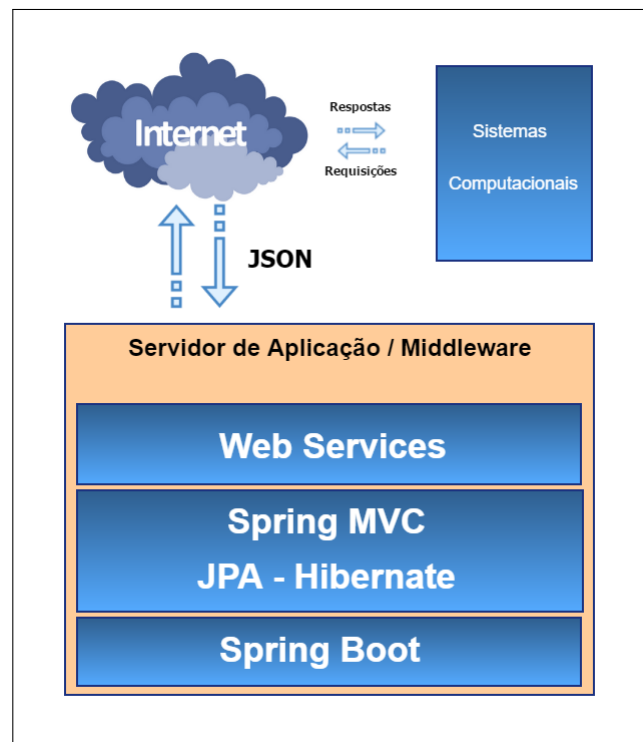
para os recursos disponibilizados pelo *Middleware*. Dessa forma, aplicações clientes podem concentrar seus esforços no desenvolvimento de outros artefatos presentes em sua solução. Todas as informações trocadas entre os clientes e o servidor são feitas através do formato JSON, o qual representa um padrão amplamente utilizado em aplicações Web, especialmente por utilizar menos caracteres que outros formatos consagrados (por exemplo XML), reduzindo o volume de dados transmitido pela Internet.

### 3.1.4 Implementação

Após a fase de levantamento de requisitos e da modelagem do sistema, faz-se necessário um estudo de viabilidade técnica para determinar quais tecnologias usar e quais descartar. Nesse momento do projeto um breve protótipo foi feito para identificar os principais recursos tecnológicos existentes e quais utilizar na solução final.

Durante esta etapa identificaram-se as seguintes palavras-chave: segurança, estabilidade, heterogeneidade e produtividade. Como o *Middleware* funciona *online* e permite que vários clientes solicitem seus serviços, é necessário prover um certo grau de segurança. A estabilidade vem em seguida, pois a solução é pensada para atender desde um cliente tentando visualizar

Figura 11 – Arquitetura da solução



Fonte – Autoria própria.

seus exames quanto um *Holter* enviando os vários sinais que o compõem em tempo real, sendo portanto, pensado para atuar online sem interrupções. A heterogeneidade é um requisito importante em sistemas distribuídos, pois permite que diversas soluções sejam construídas nas mais diversas tecnologias. Por esse motivo, optou-se por fornecer serviços através de *Web services*. Não sendo menos importante que os demais, a produtividade faz-se necessário para atender aos mais diversos requisitos num curto intervalo de tempo.

#### 3.1.4.1 Algoritmo para identificação do complexo QRS, ondas P e T

A detecção do complexo QRS é o passo inicial para uma aplicação envolvendo análise de ECG. Ao longo dos anos, muitos algoritmos foram desenvolvidos com o intuito de detectar corretamente este importante elemento. Porém, para um mesmo conjunto de sinais de teste, é possível identificar uma grande divergência nos resultados obtidos. Desta forma, no presente trabalho utilizam-se algoritmos desenvolvidos pela comunidade científica que tenham obtido uma taxa de acerto superior a 99% em testes com a base de dados MITDB (MOODY; MARK, 1990). Vários algoritmos foram testados e optou-se, para uma versão inicial, por utilizar um algoritmo de tempo real com melhor taxa de acerto na detecção do complexo QRS



(MADEIRO, 2013). Para utilização deste algoritmo no servidor foi necessário portar os códigos, fornecidos pelo autor supracitado anteriormente, da linguagem Matlab para Java.

Para a detecção das ondas P e T utilizou-se um algoritmo fornecido em linguagem de programação C/C++ para detecção do intervalo Q-T (CHESNOKOV Y. C.; NERUKH, 2006). Internamente, o algoritmo desenvolvido consiste da filtragem contínua do sinal de ECG através de *Continuous Wavelet Transform* (CWT) e *Fast Wavelet Transform* (FWT). Como os códigos fontes distribuídos pelo autor são em linguagem C++, optou-se por utilizar a tecnologia *JNA* (*JAVA Native API*) que nativamente realiza a chamada da rotina desejada, não sendo preciso reimplementar os algoritmos para a linguagem alvo (JAVA).

#### 3.1.4.2 Integração com formato de exame HL7 (aECG)

Na solução proposta, o formato HL7 é utilizado para importar e exportar os exames contidos na base de dados. O padrão HL7 trabalha com duas versões distintas: família 2.x (2.3, 2.3.1, 2.4, 2.5. e 2.6) e versão 3. A estrutura das mensagens da versão 2.x pode ser representada em dois formatos distintos:

- (i) Baseado em caracteres ASCII: reservando caracteres especiais para separar um campo de outro em uma mensagem. Apesar de não ser um formato binário, sua leitura ainda era pouco legível o que tornava difícil sua interpretação na ausência de um software.
- (ii) XML: A partir da versão 2.3.1 é possível utilizar XML para representar as mensagens, porém as *tags* não apresentam descrições significativas, o que ainda a torna ilegível.

A versão 3, que vem sendo desenvolvida gradativamente a partir de 1997, proporciona uma nova abordagem: modelagem orientada a objetos incorporando princípios da *Unified Modeling Language* (UML). Além disso, foi constatado que para que o padrão HL7 tenha uma implementação mais legível e reutilizável do que nas versões anteriores, é necessário o emprego de outras tecnologias como, por exemplo, XML.

Para a presente solução adota-se apenas a versão 3 do padrão HL7, uma vez que pressupõe-se que novas soluções busquem utilizar a versão mais recente para suporte ao HL7. Optou-se por utilizar uma biblioteca de terceiro e adaptá-la aos requisitos deste trabalho. Para tanto, foi utilizada uma solução livre que realiza leitura e escrita no formato aECG (HOPKINS, 2015).

### 3.1.4.3 Integração com formato de exame DICOM-ECG

O DICOM-WS 30, conhecido popularmente como DICOM-ECG, é um formato binário largamente adotado pela indústria mundial (BOND, 2011). Várias empresas oferecem em suas soluções suporte ao padrão DICOM-ECG como forma de aumentar a interoperabilidade entre as mais diversas soluções existentes. Dessa forma, exames realizados por produtos que exportem seus dados nesse formato poderão ser consumidos por produtos desenvolvidos por diferentes empresas, garantindo, assim, uma universalidade no acesso aos dados de exames de ECG.

Assim como no caso do HL7, foi utilizada uma biblioteca de terceiros para leitura e escrita nesse formato. Para tanto utilizou-se a biblioteca fornecida pela PixelMed (PIXELMED, 2016). Ela é um *toolkit* DICOM *stand-alone* que implementa leitura e escrita de exames no formato DICOM-ECG e suporta os vários objetos que compõem o padrão.

## 3.2 Projeto do aplicativo para geração de laudos remotos

A fim de validar o *Middleware* construído, foi desenvolvida uma aplicação para acessar remotamente os *Web services* contendo os recursos de análise de ECGs fornecidos pelo *Middleware*. Para definir as funcionalidades desejadas da aplicação avaliou-se um software atualmente utilizado pelo hospital, buscando entender seus pontos fortes e fracos. Nessa pesquisa, compreendeu-se que a principal necessidade encontrada era disponibilizar a geração dos laudos de ECGs online, possibilitando aos médicos do hospital realizarem essa análise de qualquer lugar tendo como dependência o uso da Internet. Um outro ponto não menos importante, era que o sistema pudesse ser utilizado pelos mais diversos dispositivos, por exemplo: *Tablets*, *Smartphones* e computadores. De forma a atender todas as necessidades, optou-se por desenvolver uma aplicação de interface responsiva, a qual pode ser utilizada pelos mais diversos dispositivos. As próximas seções apresentam os passos necessários à construção da solução.

### 3.2.1 Requisitos

Após avaliar o software, atualmente utilizado pelo Hospital Universitário Walter Cantídio, e realizar entrevistas com os *Stakeholders* do projeto, foram identificadas algumas funcionalidades importantes para o sistema proposto, tais como:

- Executar nos mais variados ambientes;

- Permitir autenticação;
- Possuir perfis de acesso;
- Adicionar e visualizar marcações do traçado do ECG;
- Adicionar e visualizar anotações do traçado do ECG;
- Gerar laudos dos exames;
- Apresentar informações úteis do exame;
- Ser baixo custo;
- Permitir vários formatos de exame;
- Apresentar um bom desempenho.

A Tabela 4 representa os requisitos funcionais e não funcionais extraídos desse levantamento inicial de requisitos. Novamente, faz-se uso da nomenclatura adotada na Seção 3.1.1 para apresentar os requisitos da solução.

### **3.2.2 Modelagem do sistema**

A partir dos requisitos levantados, desenvolvem-se os casos de uso, conforme visto na Figura 12. Diferentemente do *Middleware*, descrito em seções anteriores, o qual não possuía interação direta com os usuários, a presente solução é desenvolvida para ser utilizada diretamente por médicos e pacientes dentro de um contexto de aplicações médicas. Neste sentido, fez-se necessário adicionar à aplicação questões relacionadas à usabilidade.

Após finalizar a modelagem dos casos de uso, iniciam-se os diagrama de classe do sistema. Em sua implementação seguiu-se o modelo *Domain Driven Design* (DDD), o qual a maior parte do esforço é dedicado ao domínio da aplicação. Na Figura 13, tem-se uma representação das classes na visão do domínio da aplicação. Destaca-se da figura anterior, a classe "*Exam*". Para sua concepção foi realizado um estudo dos padrões de exames ECG (DICOM-ECG, aECG, SCP entre outros) e das principais informações que representassem uma abstração fiel a esta entidade.

### **3.2.3 Arquitetura da Aplicação**

Para explicar a organização utilizada, utilizaremos o termo arquitetura de duas divisões físicas, o qual possui os conceitos de máquinas clientes e máquinas servidoras (TANENBAUM, 2007). Sendo assim, dividiu-se a aplicação numa extremidade frontal gráfica, que se comunica com o resto da aplicação residente no servidor, por meio de um protocolo específico

Tabela 4 – Requisitos Funcionais e Não-Funcionais

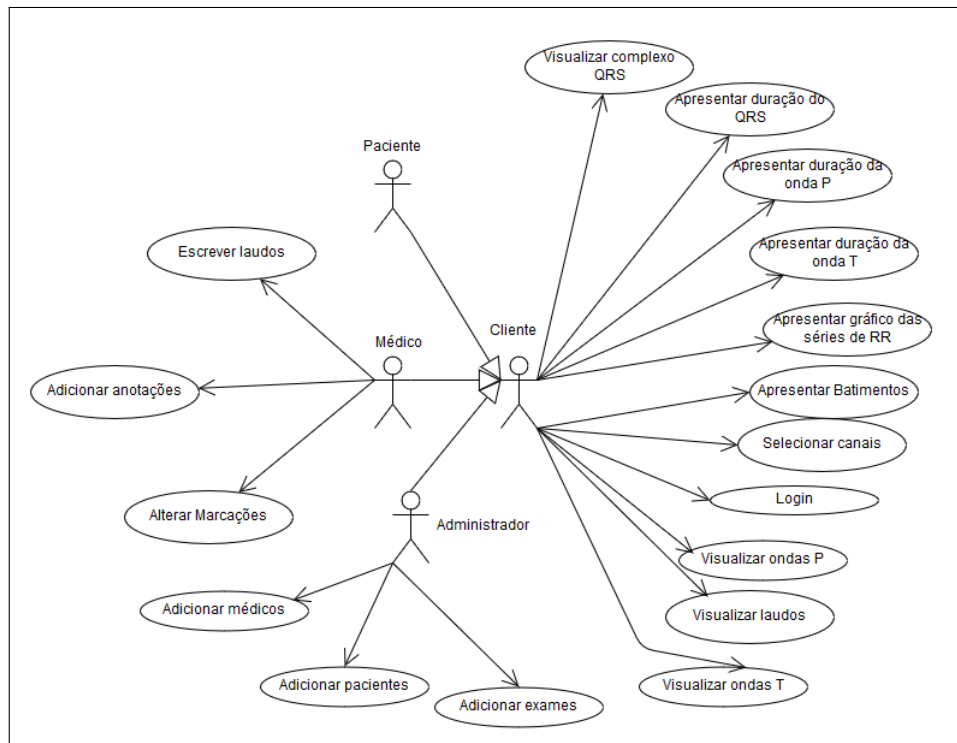
<b>Requisitos</b>	<b>Descrição</b>	<b>Prioridade</b>
RNF_0001	Funcionar nos mais diversos ambientes (heterogeneidade)	ALTA
RF_0002	Disponibilizar uma página de login para acesso a aplicação	ALTA
RF_0003	Disponibilizar 3 perfis de usuários: administrador, médicos e pacientes	ALTA
RF_0004	Permitir a visualização das ondas P identificadas num canal	ALTA
RF_0005	Permitir a visualização das ondas T identificadas num canal	ALTA
RF_0006	Permitir a visualização dos Complexos QRS identificadas num canal	ALTA
RF_0007	Permitir a visualizar de 1 ou mais canais simultaneamente de um exame	ALTA
RF_0008	Apresentar para os clientes a duração do complexo QRS por canal	ALTA
RF_0009	Apresentar para os clientes a duração da onda P por canal	MÉDIA
RF_0010	Apresentar para os clientes a duração da onda T por canal	ALTA
RF_0011	Apresentar para os clientes o total de batimentos por canal	ALTA
RF_0012	Apresentar para os médicos gráfico da série dos intervalos de RR por canal	MÉDIA
RNF_0013	Utilizar tecnologias gratuitas	ALTA
RF_0014	Permitir ao perfil médico realizar anotações no traçado de um canal	ALTA
RF_0015	Permitir ao perfil médico alterar marcações feitas dos Complexos QRS de um canal	MÉDIA
RF_0016	Permitir ao perfil médico alterar marcações feitas das ondas P um canal	MÉDIA
RF_0017	Permitir ao perfil médico alterar marcações feitas das ondas T um canal	ALTA
RF_0018	Permitir aos clientes importarem exames no formato DICOM-ECG	MÉDIA
RF_0019	Permitir aos clientes importarem exames no formato aECG	MÉDIA
RF_0020	Permitir aos clientes exportar exames nos formato aECG e DICOM-ECG	MÉDIA
RF_0021	Permitir ao perfil administrador criar: médicos e pacientes	MÉDIA
RNF_0022	Apresentar um bom desempenho	MÉDIA

Fonte – Autoria própria

da aplicação (*Web services*). A Figura 14 representa o modelo adotado para a solução, o qual, toda a aplicação reside no servidor ficando para o cliente, através de um *browser* de internet, apenas a visualização e interação com as páginas Web fornecidas pelo servidor.

Como é possível observar na Figura 15, a aplicação foi desenvolvida utilizando o modelo MVC. Sendo portanto, um padrão de projeto (*design pattern*), também conhecido como arquitetura, frequentemente utilizado para aplicações que necessitam manter múltiplas visões de um mesmo conjunto de dados. Este padrão define três camadas bem definidas: modelo (*model*), controladora (*controller*) e visão (*view*). Diferentemente do *Middleware* descrito em seções anteriores, na presente aplicação todas as camadas do padrão MVC são utilizadas. Portanto,

Figura 12 – Diagrama de Casos de Uso da solução



Fonte – Autoria própria

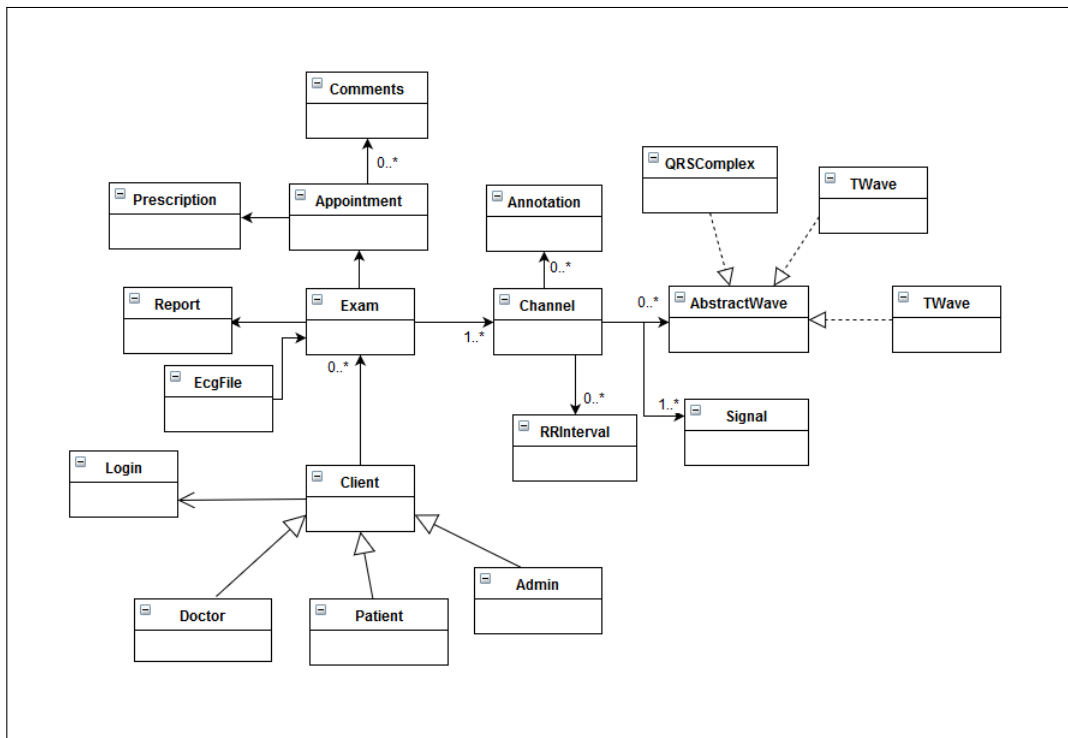
a camada controladora recebe requisições da camada de visualização (*Javascript*), caso seja necessário envia requisições ao *Middleware* ou se comunica com o modelo (banco de dados)

Além da estrutura multicamadas, a arquitetura da presente aplicação proposta apresenta um mecanismo de comunicação com a Internet e com sistemas auxiliares. O sistema auxiliar utilizado pela aplicação é o *Middleware* desenvolvido que fornece uma API aberta, permitindo uma fácil integração e utilização de seus recursos para análise de ECGs. Por fim, na arquitetura também é destacada a presença de um componente *Sistemas de arquivos*, utilizado especialmente para armazenamento dos exames enviados. Como pode ser visto na Figura 15, estes dados podem ser armazenados localmente no servidor, utilizando uma estrutura de pastas, ou externamente através de um servidor de armazenamento de arquivos remotos.

### 3.3 Autenticação, autorização e segurança

Neste trabalho optou-se pela implementação simplificada do protocolo OAuth 2.0 para garantir autenticação e autorização aos recursos. Neste cenário, o *Middleware* representa os papéis de servidor de autorização e de recursos, enquanto a aplicação de laudos remotos é responsável por identificar o usuário em seu sistema e redirecionar os dados de login para o

Figura 13 – Diagrama de Classes da solução



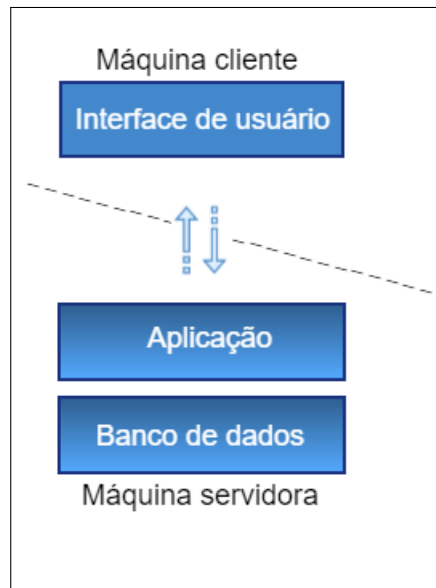
Fonte – Autoria própria

*Middleware*. Esse fluxo de autenticação e autorização é apresentado na Figura 16.

Como pode ser observado, a etapa "1" representa o cliente realizando o *login* no sistema. Os dados são enviados ao servidor da aplicação que busca no banco de dados o campo *login*. Caso o usuário exista, este servidor executa uma requisição HTTPS (*Web service*) do tipo POST ao *Middleware*, etapa "2". No *Middleware* é feita a autenticação final do usuário e caso as credenciais enviadas sejam válidas é retornado, na etapa "3", dois valores: *refresh token* e *token*. A partir desse momento, todas as requisições feitas ao *Middleware* deverão apresentar o *token* no corpo da mensagem. Vale salientar que o *token* possui um tempo máximo de uso de 10 minutos (parâmetro configurável). Portanto, a cada 10 minutos faz-se necessário atualizar este valor, neste momento o servidor da aplicação envia uma solicitação de atualização de *token* e passa para isso o *refresh token*, enviado na primeira autenticação.

Apesar de possuir as desvantagens do custo para aquisição, manutenção dos certificados e perda de desempenho, aplicações médicas possuem como um de seus requisitos mais importantes a segurança. Portanto, para implantar esta solução faz-se necessário a aquisição da certificação *Secure Socket Layer (SSL)* para garantir que todas as informações trocadas entre o cliente e o servidor sejam seguras.

Figura 14 – Organização cliente-servidor da aplicação



Fonte – Adaptado de (TANENBAUM, 2007)

### 3.4 Componentes de software utilizados

A tabela 5 apresenta as tecnologias utilizadas para o desenvolvimento do *Middleware* e da aplicação para análise remota de ECGs.

Tabela 5 – Tecnologias de software utilizadas no *Middleware* e Aplicação Web

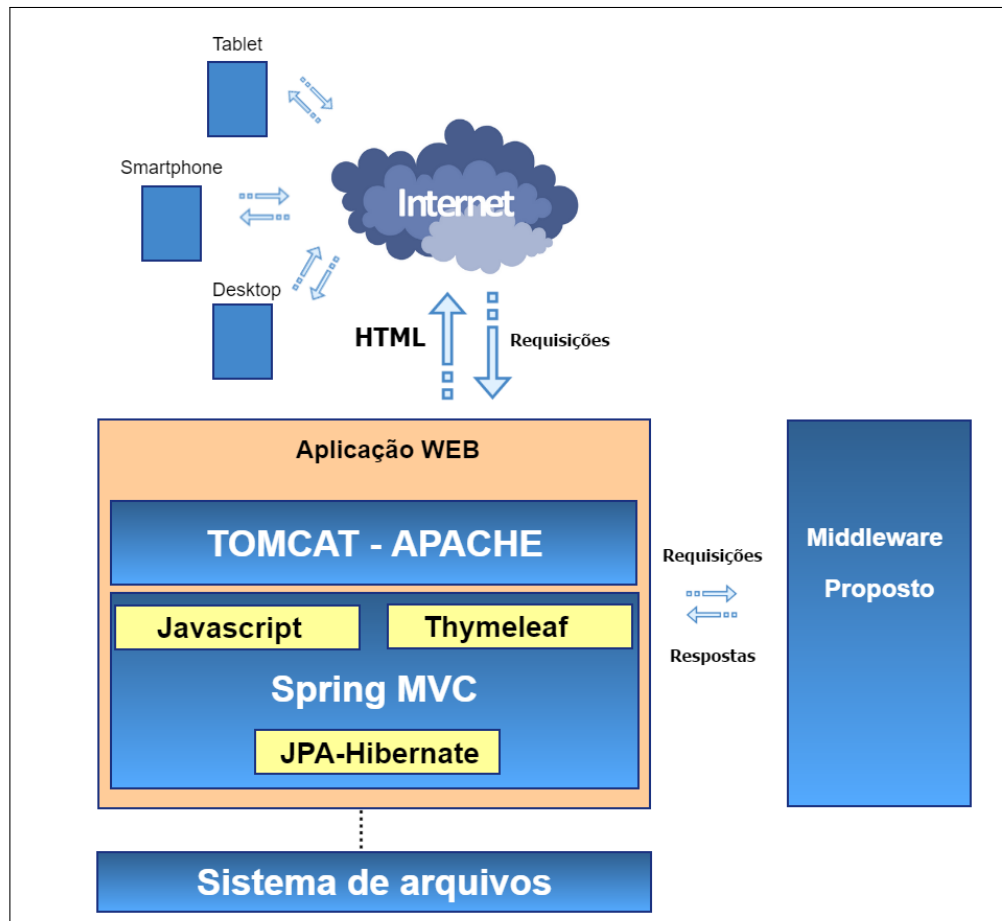
<b>Componente</b>	<b><i>Middleware</i></b>	<b>Aplicação Web</b>
Sistema Operacional	Ubuntu 16.04 Server	Ubuntu 16.04
Framework Web	Spring MVC	Spring MVC
Template Web	-	Thymeleaf
Framework para Persistência de dados	Hibernate / JPA	Hibernate / JPA
Servidor de Aplicação	Spring Boot	TOMCAT
Banco de Dados	MySQL	MySQL
Linguagem de Programação	JAVA	JAVA

Fonte – Autoria própria

### 3.5 Equipamentos utilizados

Para realização desta dissertação utilizou-se um computador pessoal para o desenvolvimento dos principais componentes de software citados ao longo deste capítulo. O computador utilizado possui as seguintes características: processador Intel Core I7 de 3,40GHz, Memória

Figura 15 – Arquitetura da aplicação de análise de ECGs



Fonte – Autoria própria

RAM 32 GB e sistema operacional Ubuntu 16.04 LTS.

Devido a ausência de um servidor com alto desempenho e robustez, optou-se por utilizar um ambiente virtualizado com configurações modestas, similar às fornecidas por sites de hospedagem. Para tanto, utilizou-se o computador com as seguintes configurações: processador com 2 cores, 8 GB de RAM, placa de rede 10/100, disco de 100 GB.

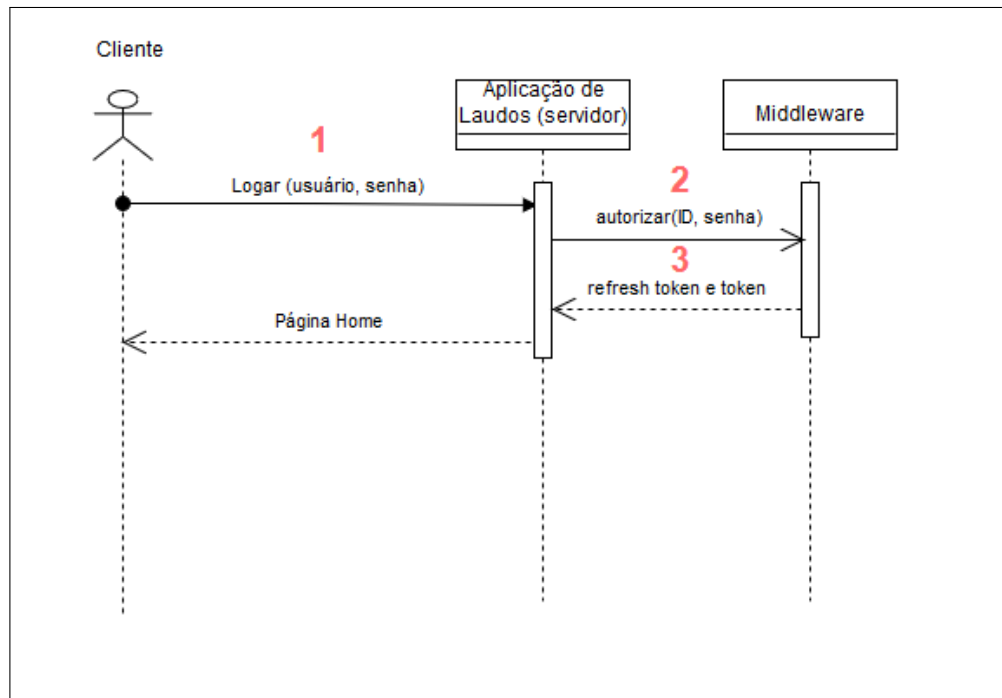
### 3.6 Verificação e validação

Ao final de cada etapa do desenvolvimento, era realizada uma validação das funcionalidades implementadas a fim de garantir a conformidade da solução com os requisitos pré-estabelecidos. Esta tarefa de avaliação foi desempenhada pelo orientador e co-orientador deste trabalho.

Devido a restrição de tempo, optou-se por validar o desempenho da solução proposta apenas através de teste de carga, em especial nos dois serviços que mais consomem recursos,



Figura 16 – Diagrama de Sequência para Autenticação



Fonte – Autoria própria

deixando para trabalhos futuros as demais forma de validação.

### 3.7 Resumo do capítulo

Neste Capítulo foram apresentadas as tecnologias utilizadas na implementação do *Middleware* e da aplicação WEB responsável pela análise de ECGs online. Inicialmente, foi apresentado o *Middleware* utilizado para fornecer serviços de análise de eletrocardiogramas para outras aplicações, além de permitir a leitura dos sinais nos formatos de exames aECG e DICOM-ECG. Também são apresentadas as motivações pela utilização de *Web services* como o mecanismo de disponibilização de funcionalidades. Na segunda parte deste capítulo, apresentou-se os recursos e tecnologias presentes na aplicação WEB desenvolvida para auxiliar profissionais de saúde na análise de ECGs. Durante todo o ciclo de desenvolvimento foi utilizado a linguagem Java e seus *Frameworks* para garantir a construção de soluções robustas, livres e multiplataforma.

## 4 RESULTADOS

### 4.1 Visão geral

Neste capítulo são apresentados os resultados obtidos do sistema Web proposto. Para isto, a abordagem utilizada consiste da captura de um conjunto de imagens obtidas durante o funcionamento do software cliente, com o objetivo de demonstrar seus componentes e também todas as funcionalidades que são consideradas inovações em relação a sistemas similares disponíveis.

Inicialmente, apresenta-se a página de login do sistema, na qual os usuários informam seus dados de identificação para terem acesso ao sistema. Em seguida, são apresentados os perfis de usuários utilizados na aplicação, que foram determinados durante a etapa e levantamento de requisitos. Posteriormente, são apresentadas as principais páginas Web do sistema, exemplificando sua utilização.

Ao final deste capítulo, são realizadas medições nos principais pontos de tráfego de dados no servidor, avaliando assim o desempenho da solução proposta.

#### 4.1.1 Perfis de usuários

Durante a etapa de levantamento de requisitos foi determinado a utilização de perfis de usuários para melhor dispor os dados e funcionalidades. Para a versão inicial do projeto determinou-se a necessidade de pelo menos três perfis: administrador, médico e paciente.

Ao perfil *administrador* determinou-se a responsabilidade de gerenciar cadastros (médicos, pacientes e exames). Visualizar, adicionar, atualizar e remover são ações disponíveis em cada um dos cadastros que podem ser realizados pelo perfil de administrador do sistema. Inicialmente, foi pensado um controle completo sobre a aplicação permitindo-se inclusive alterar as marcações médicas. Durante a revisão dos requisitos optou-se por remover tal poder do administrador, uma vez que o mesmo não possui habilitações específicas necessárias para manipular dados médicos.

Já o perfil *paciente* foi criado para permitir aos clientes a visualização online de seus exames. O paciente pode visualizar, através da internet, os laudos e informações extraídas dos traçados do eletrocardiograma, conteúdo este já revisados por médicos devidamente autorizados.

O último perfil identificado nos requisitos foi o de "médico". Ao médico ficou determinado a possibilidade de alterar as marcações (complexo QRS e ondas P e T), realizar anotações no próprio traçado do exame, além da edição do laudo do exame.

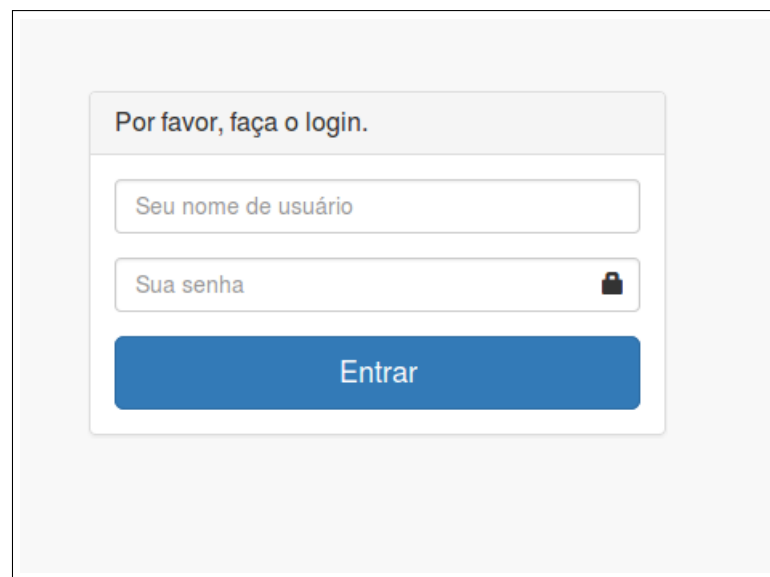
Um ponto a ser observado é que cada cliente da solução pode possuir mais de um perfil. Desta forma, podemos ter um médico com perfis de "médico" e "administrador", possibilitando ao médico não somente manipular exames mas, além disso, manipular dados cadastrais (somente permitido ao perfil "administrador").

## 4.2 Página de login

O módulo de autenticação e autorização desenvolvido para o sistema proposto é baseado em permissões e usuários. Ao longo do trabalho foram definidos os perfis de usuário: administrador, médico e paciente. A cada um dos perfis foi atribuído a utilização de um conjunto de funcionalidades, como descritas na seção anterior.

Para esta versão optou-se por utilizar apenas dois campos: login e senha, como pode ser visto na Figura 17.

Figura 17 – Página de login do sistema



Por favor, faça o login.

Seu nome de usuário

Sua senha

Entrar

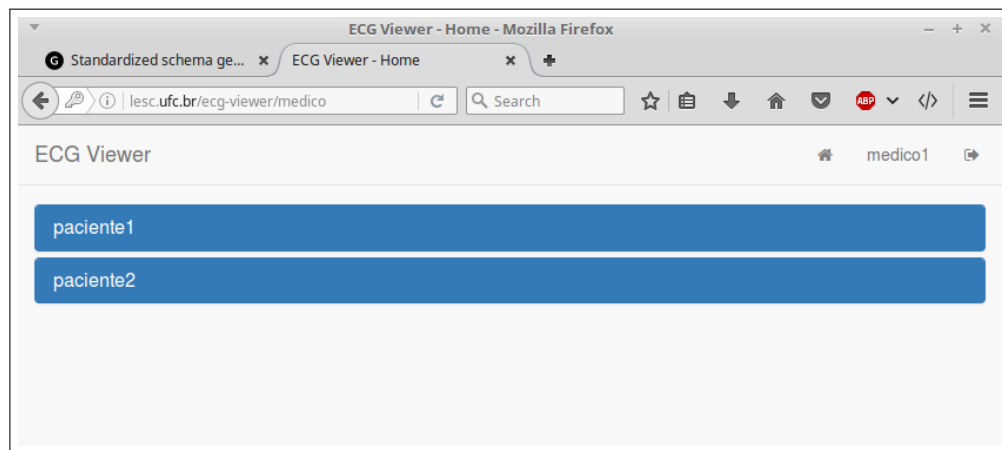
Fonte – Autoria própria

### 4.2.1 Página inicial do médico

Após realizar o login no sistema, usuários do perfil *médico* são redirecionados para sua página principal. Nesta página é possível identificar todos os pacientes associados, isto é, os pacientes os quais ele possui permissão para visualizar. Para cada paciente, é possível identificar todos os seus exames de ECGs realizados ao longo do tempo, com o objetivo de

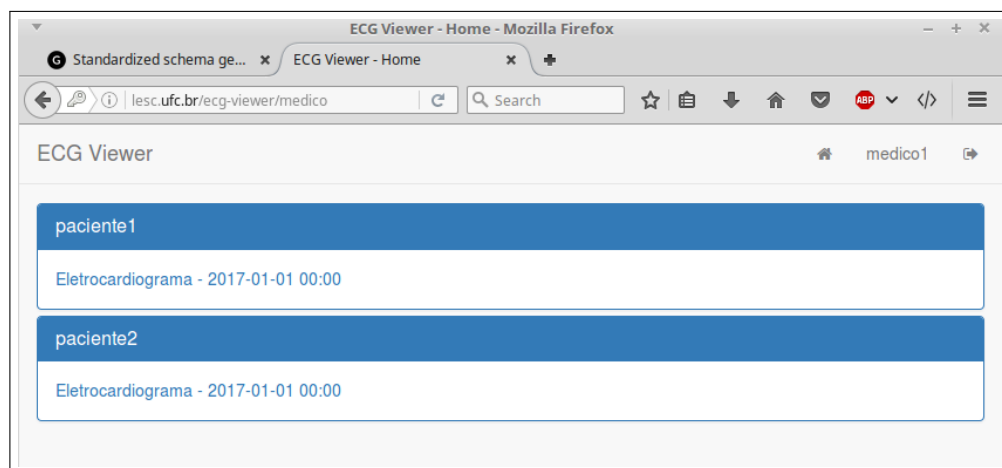
manter um histórico desse paciente durante todo o ciclo de vida da aplicação e do atendimento do paciente pelo respectivo médico. As Figuras 18 e 19 apresentam a listagem dos pacientes e seus respectivos exames de ECGs de médico. Para exemplificar os pacientes de um médico foram criados as entidades: *paciente1* e *paciente2*, os quais foram construídos a partir da base de dados do MITDB (caso de estudo 100 e 101).

Figura 18 – Página inicial do médico listando os pacientes



Fonte – Autoria própria

Figura 19 – Página inicial do médico mostrando os exames dos pacientes



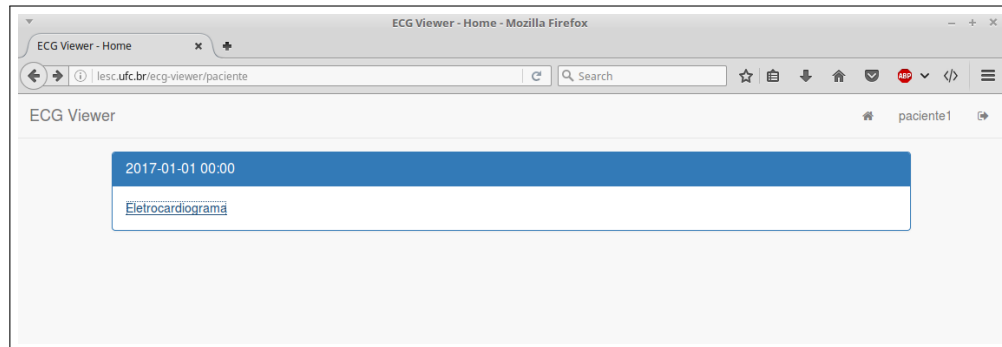
Fonte – Autoria própria

#### 4.2.2 *Página inicial do paciente*

Após realizar o *login* no sistema, usuários do perfil *paciente* são redirecionados para sua página principal. Nesta página é possível identificar todos os exames realizados. Cada

um dos exames mostrados foram acrescentados pelo próprio paciente ou por funcionário do consultório com perfil de administrador. A Figura 20 apresenta a tela com um exame importado pelo sistema. Novamente, utilizou-se o *paciente1* criado anteriormente e que representa o caso de estudo *100* da base de dados MITDB.

Figura 20 – Página inicial do paciente



Fonte – Autoria própria

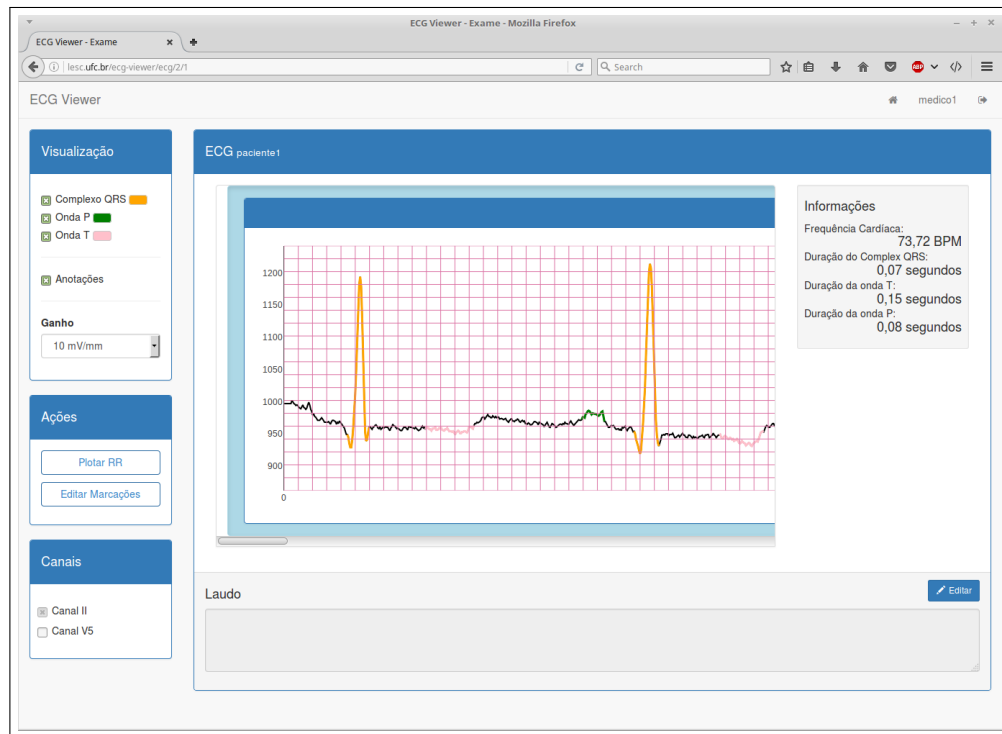
### 4.3 Página do exame de ECG

As páginas de visualização do ECG representam, dentro da solução, a principal contribuição do presente trabalho. Na fase de requisitos avaliaram-se funcionalidades importantes para análise de eletrocardiogramas e que contribuíssem de forma efetiva no auxílio aos médicos na tomada de decisão. Podemos elencar a lista dos principais recursos disponíveis:

- Traçado do ECG por canal.
- Marcações do complexo QRS por canal.
- Marcações das ondas P por canal.
- Marcações das ondas T por canal.
- Realizar anotações no traçado do exame.
- Escrita de laudos.
- Gráfico da série de intervalos RR.

A Figura 21 apresenta a tela de visualização de exames. Nas seções seguintes descreve-se cada um desses itens fornecendo uma melhor compreensão das funcionalidades implementadas.

Figura 21 – Página de visualização de exames



Fonte – Autoria própria

#### 4.3.1 Visualização do traçado do exame

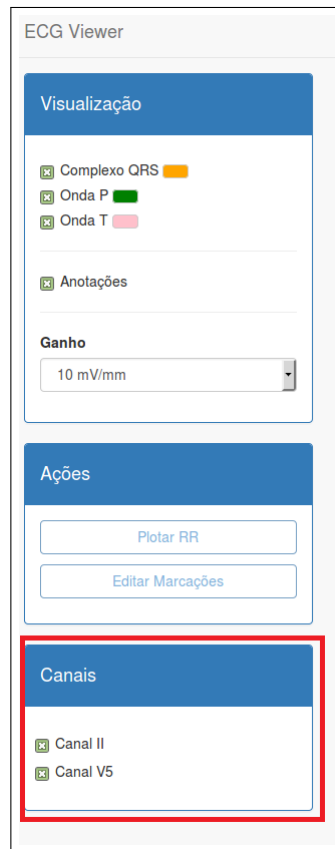
O requisito fundamental de uma aplicação de análise de ECGs é apresentar ao médico o traçado dos sinais para cada um dos canais presentes no exame. Com esta informação o médico é capaz de identificar, por exemplo, as seguintes patologias: irregularidades no ritmo cardíaco (arritmia), artérias bloqueadas, infarto, entre outros problemas citados na literatura (PASTORE *et al.*, 2016). Como pode ser visto na Figura 22, é possível ao médico e paciente visualizarem apenas os canais desejados. Dessa forma pode-se destacar um canal (exibindo-o separadamente) ou visualizar todos os canais de interesse simultaneamente. A Figura 23 apresenta o traçado do exame do paciente 100 (cem) do MITDB, destacando a visualização de seus dois canais.

Adicionou-se uma barra de progresso ao final do último canal em exibição para permitir que todos os canais mostrem juntos seus sinais num mesmo dado instante de tempo, permitindo assim ao médico avaliar todo o traçado pelas várias derivações fornecidas pelo exame.

#### 4.3.2 Marcações

Através das marcações no traçado de ECGs é possível extrair um conjunto de informações para auxiliar médicos no diagnósticos de seus pacientes. Complexo QRS, ondas

Figura 22 – Área da página de exame destacando a seleção de canais



Fonte – Autoria própria

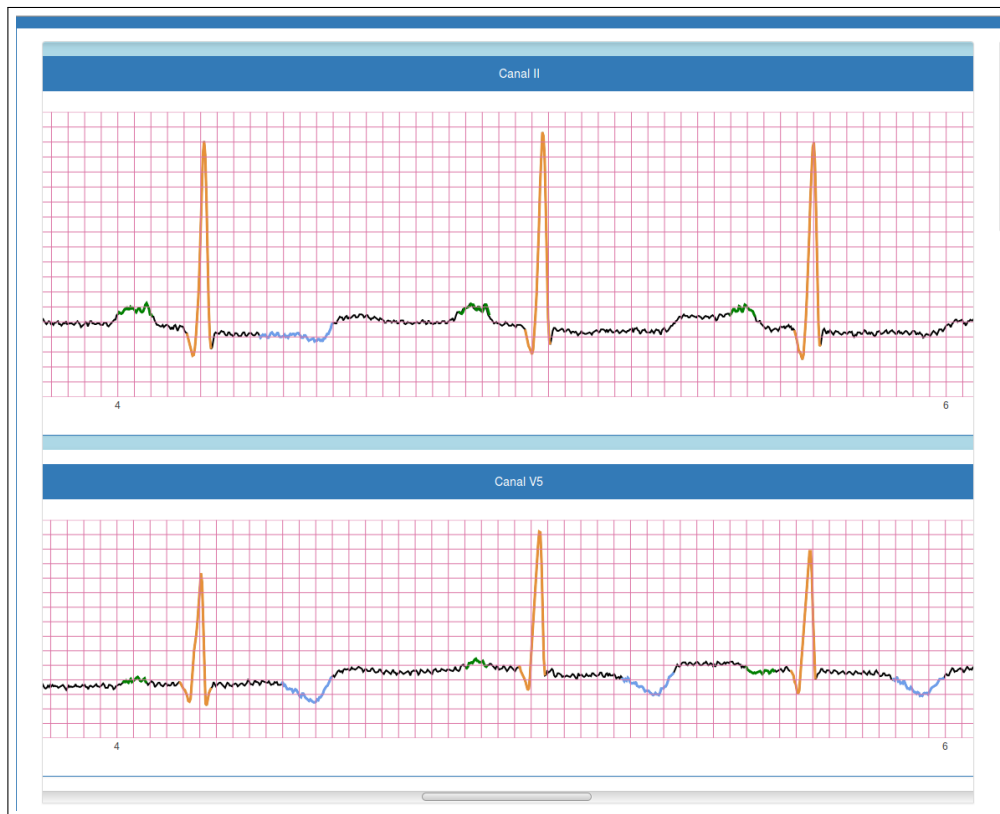
P e T, são exemplos de marcações mostrados pelo presente trabalho. Durante a visualização do eletrocardiograma, o cliente pode selecionar quais marcações ver juntamente com o traçado dos sinais. A Figura 21 mostra um canal do exame exibindo, além do traçado do sinal, suas marcações de complexo QRS e ondas P e T.

Ao médico ou paciente é permitido selecionar quais marcações deseja ver num dado momento. Como pode ser visto na Figura 24, foi adicionada uma palheta de cores para cada uma das opções de marcação, permitindo ao cliente uma melhor visualização de acordo com seu gosto.

### 4.3.3 Anotações

Por meio de anotações no traçado do eletrocardiograma, o médico consegue destacar pontos importantes ao longo do exame. Exames do *MIT-BIH*, HL7 e DICOM possuem em suas especificações campos contendo as anotações. Identificou-se, durante o desenvolvimento, haver divergência em como essa informação é armazenada entre esses padrões.

Figura 23 – Exibição de dois canais simultaneamente



Fonte – Autoria própria

Neste trabalho optamos por não seguir nenhum dos modelos e ficou a cargo do médico escrever, sobre um determinado ponto do traçado, suas observações. A Figura 25 mostra a ação de adicionar uma anotação no sistema. Para realizar esta adição é necessário apenas selecionar um ponto na representação temporal do sinal e escrever o comentário. Após clicar em *Salvar Alterações*, as anotações serão enviadas ao servidor.

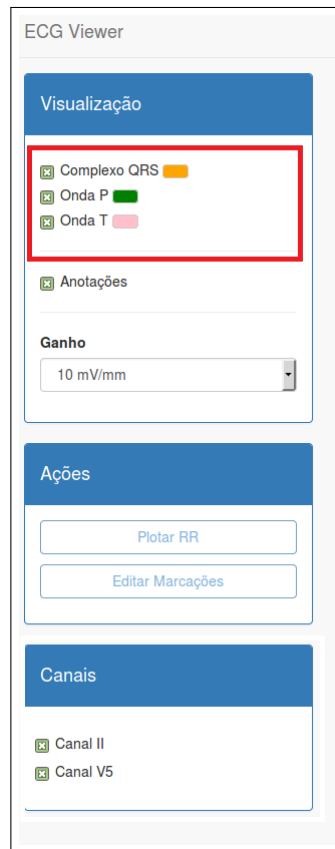
#### 4.3.4 Laudo

No canto inferior da página de exame foi adicionada uma área para edição do laudo médico. Como pode ser visto na Figura 21, tem-se uma caixa de texto na qual o médico deverá preencher o laudo do exame. Esta página segue as seguintes regras de acordo com o perfil:

- Médico: pode visualizar, modificar e imprimir laudo.
- Paciente: pode visualizar e imprimir laudo.
- Administrador: pode visualizar e imprimir laudo.

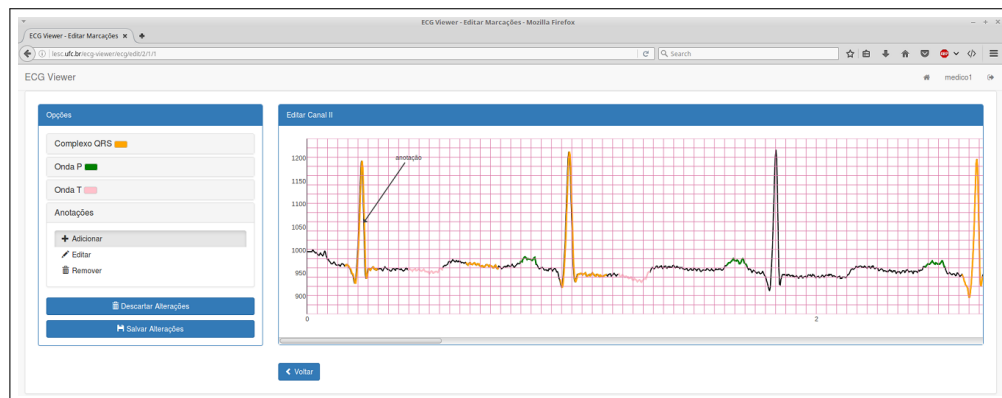


Figura 24 – Área da página do exame destacando a opção de selecionar marcação



Fonte – Autoria própria

Figura 25 – Médico adicionando uma anotação



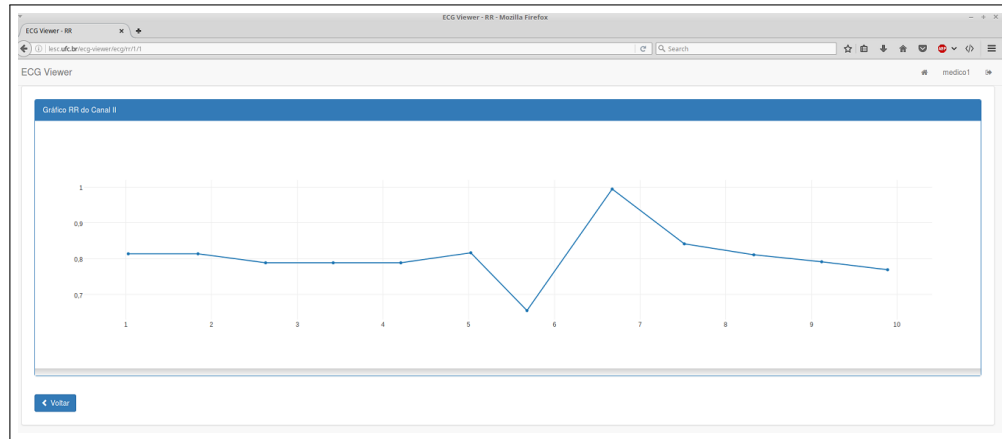
Fonte – Autoria própria

#### 4.3.5 Gráfico da série de intervalos RR

A análise da variabilidade da frequência cardíaca (VFC), por ser uma técnica não invasiva, constitui um método valioso para a avaliação da função do sistema nervoso autônomo (SNA) em muitas condições clínicas (ACHARYA *et al.*, 2006). Portanto, monitorar a VFC permite avaliar se um paciente apresenta mau funcionamento fisiológico do coração.

A Figura 26 ilustra o gráfico dos intervalos de RR para um exame da base de dados utilizada, MITDB, referente ao paciente de N<sup>o</sup> 100 (cem).

Figura 26 – Página com gráfico de intervalos RR



Fonte – Autoria própria

#### 4.3.6 Cadastro de marcações

Sempre que um novo exame é adicionado na base de dados, a aplicação WEB preenche automaticamente todas as marcações referentes ao Complexo QRS, onda P e T. Porém, caso algum canal esteja com bastante ruído, a análise automática pode perder alguma informação e causar uma falha na identificação de uma ou mais marcações. Para solucionar o problema na detecção automática desses artefatos, optou-se por permitir a médicos a edição das marcações, como pode ser visto no menu do lado esquerdo da Figura 27. Para persistir no banco de dados as informações modificadas o médico deve clicar no botão *Salvar Alterações*. A Figura 28 exemplifica a ação de adicionar uma marcação de complexo QRS num dado canal do exame.

A importância da correta identificação dessas marcações está no fato de os algoritmos que calculam os batimentos, duração do QRS, duração da onda P, e as demais informações exibidas na página de exame, utilizarem as marcações nos seus cálculos. Portanto, quanto mais próximo do correto forem as marcações, maior será a precisão das informações repassadas ao médico.

Figura 27 – Página para editar as marcações de um exame



Fonte – Autoria própria

Figura 28 – Adicionando uma marcação



Fonte – Autoria própria

#### 4.4 Avaliação de desempenho do servidor

Para esta avaliação foram selecionados os serviços: *getQRSComplex* e *uploadFile*, sendo cada um deles descritos em detalhes nas próximas seções.

##### 4.4.1 Serviço *getQRSComplex*

O serviço *getQRSComplex* recebe uma requisição *POST* dos clientes contendo um conjunto de sinais e retorna uma lista contendo todos os intervalos de complexo QRS identificados ao longo dos sinais recebidos (mais detalhes são apresentados no Apêndice B). O tempo de processamento é diretamente proporcional à quantidade de sinais enviados para

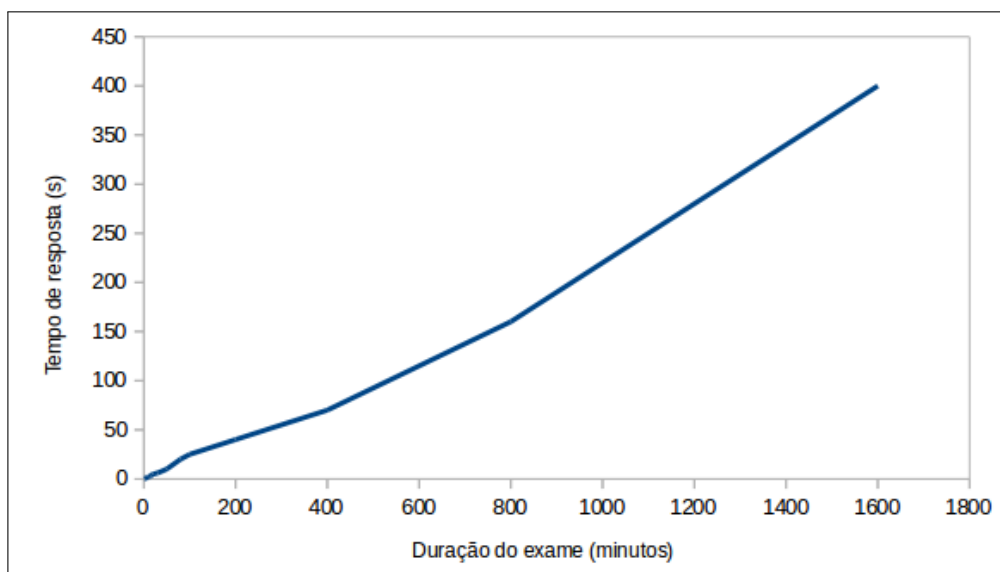
extração dos complexos QRS presentes nele. Dessa forma, em exames de curta duração, este serviço retornará poucos intervalos representando um baixo impacto no desempenho na solução, porém, em exames de longa duração este serviço representará um ponto crítico dentro da solução.

As Figuras 29 e 30 apresentam o tempo de processamento de uma requisição em duas situações distintas, primeira variando o tamanho dos exames e a segunda aumentando o número de clientes simultâneos.

Observa-se que o tempo de resposta do serviço aumenta proporcionalmente à duração do exame. Analisando os *logs* do servidor, identificou-se que tanto o tempo de processamento do algoritmo, que identifica o complexo QRS, quanto o tráfego dos sinais processados foram os responsáveis pela degradação do tempo do serviço.

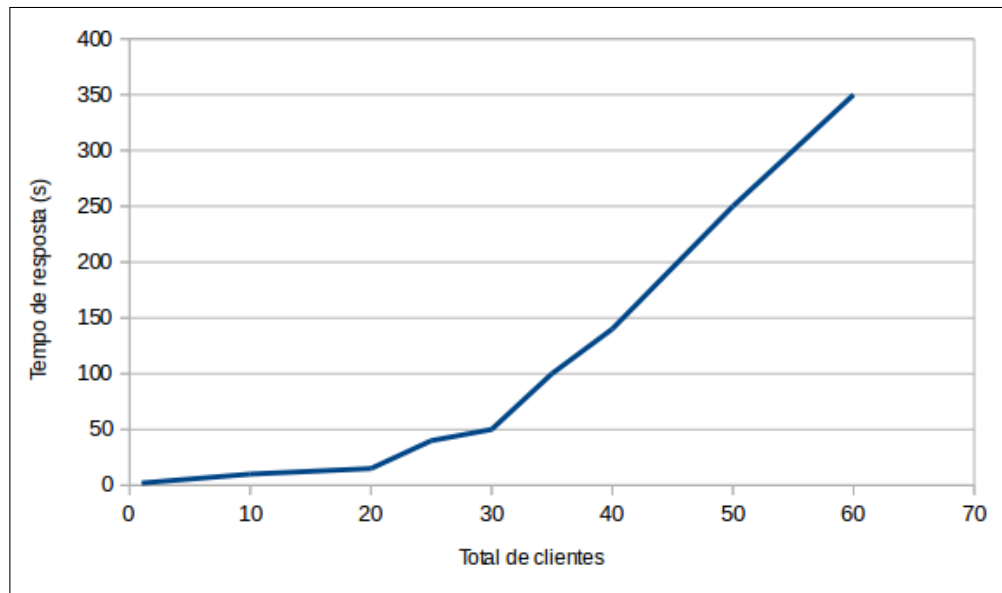
Na Figura 30, para destacar o efeito do aumento do número de clientes no desempenho do sistema, fixou-se a duração do exame em 6 minutos. Como pode ser observado, o aumento do número de clientes atendidos simultaneamente fez com que o tempo de resposta do serviço fosse prejudicado. A partir de 30 clientes observa-se um aumento maior no tempo de resposta. Esta degradação deve-se ao fato de quanto mais clientes em atendimento mais *threads* estarão em execução no servidor, consumindo os recursos compartilhados: memória, processador e disco, e prejudicando no tempo de resposta do serviço. Dessa forma, para manter um bom desempenho faz-se necessário um servidor com *hardware* superior ao utilizado.

Figura 29 – Gráfico: Tempo médio de respostas x Duração do exame



Fonte – Autoria própria

Figura 30 – Gráfico: Tempo de resposta x Total de clientes



Fonte – Autoria própria

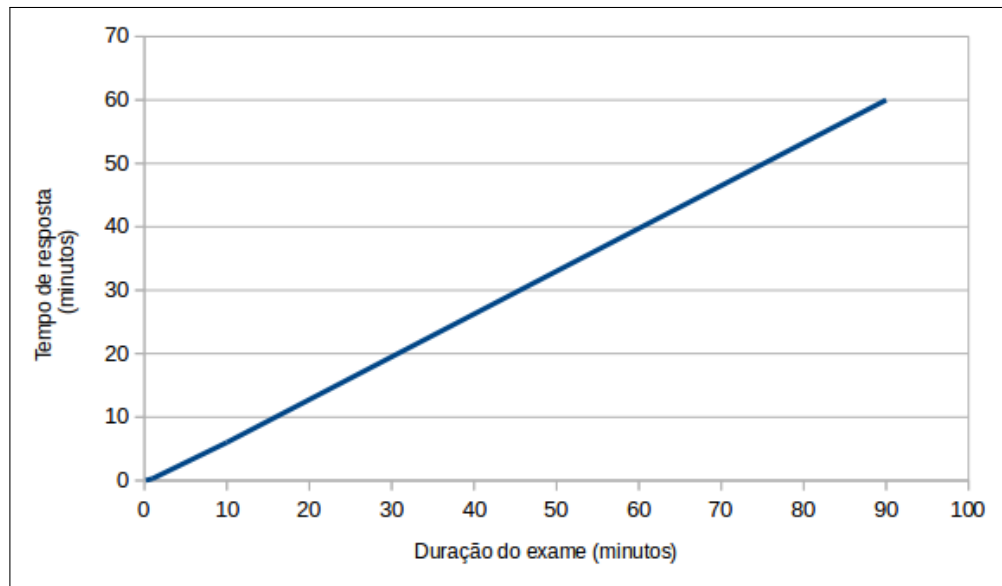
#### 4.4.2 Serviço *uploadFile*

Por meio deste serviço, aplicações clientes armazenam no servidor os exames dos pacientes. O exame é armazenado no sistema de arquivos e seus dados extraídos e persistidos no banco de dados. Em exames de longa duração este serviço poderá representar o maior consumo de recursos pela aplicação, uma vez que possui uma carga elevada de tráfego de informações (*upload* do exame) e alto consumo dos recursos compartilhados (memória, processador e disco) no servidor.

Como pode ser visto na Figura 31, aumentar a duração do exame representa um retardo na resposta do serviço. Analisando os *logs* de execução do servidor, identificou-se que o principal fator que contribuiu para a degradação do desempenho deste serviço foi o tempo de persistência dos sinais do exame no banco de dados.

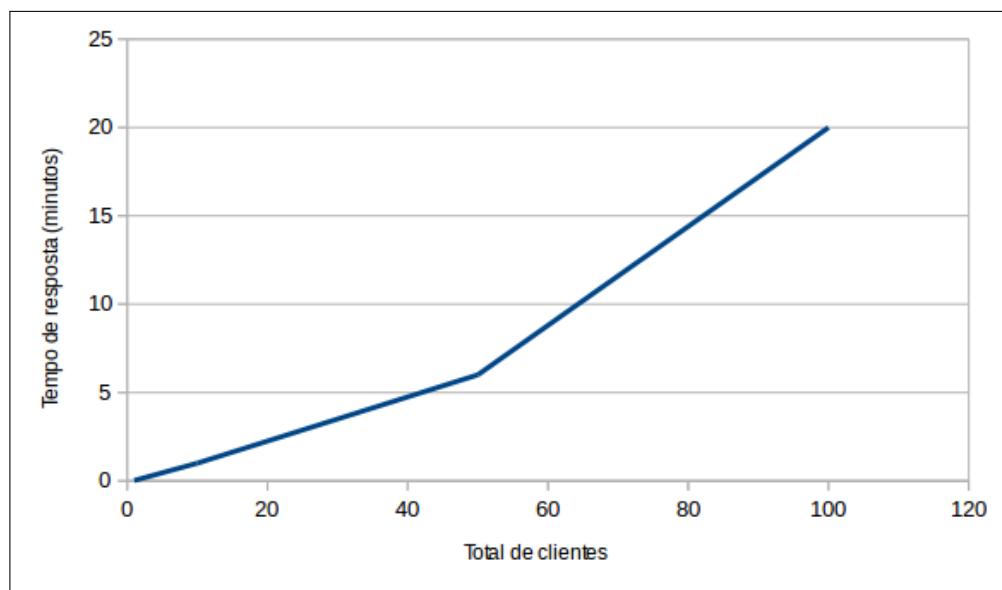
Na Figura 32, como feito na seção anterior, para destacar o efeito do aumento do número de clientes no desempenho do sistema, fixou-se a duração do exame em 10 segundos. Como pode ser observado, o aumento dos clientes atendidos simultaneamente fez com que o tempo de resposta do serviço fosse aumentado. Novamente, o principal motivo para esta degradação foi a limitação do *hardware*.

Figura 31 – Gráfico: Tempo de resposta x Duração do exame



Fonte – Autoria própria

Figura 32 – Gráfico: Tempo de resposta x Número de clientes



Fonte – Autoria própria

#### 4.5 Discussão dos resultados

Como pôde ser observado na Seção 4.4, os serviços que realizam *upload* de exames, *uploadFile*, apresentaram desempenho não satisfatório. Na tentativa de identificar o principal responsável pelo problema, isolou-se *hardware* e *software*. Com o avanço da pesquisa, foi possível comprovar que havia, na presente solução, uma falha tecnológica devida à abordagem de mapear os exames para um banco de dados relacional. As recomendações Atomicity, Consistency,

Isolation, Durability (ACID) para bancos de dados relacionais demandam um certo tempo até registrar o dado no sistema de arquivos. Exames de ECG, por menor que sejam, possuem um elevado número de sinais que o compõem temporalmente. Dessa forma, entre o tempo de extração dos sinais de um determinado exame até sua inserção no banco de dados demandou-se um tempo elevado. A fim de solucionar o problema, realizou-se uma prova de conceito utilizando uma abordagem de armazenar os dados tanto num banco de dados relacional quanto em um banco não relacional (NoSQL). No banco NoSQL, foram armazenados os dados relevantes aos exames e que possuem elevada carga de informações. Portanto, armazenam-se neste banco os sinais dos exames e seus principais componentes identificados (complexo QRS, ondas P e T) e mantém-se as demais informações armazenadas em um banco de dados relacional. Com esta mudança de abordagem, obteve-se uma melhora significativa no desempenho da aplicação. Com o uso exclusivo de banco de dados Relacional, o MITDB de 30 minutos demora em torno de 22 minutos para ser inserido no sistema. Usando um banco NoSQL, chega-se a um tempo máximo de 2 minutos, representando ganho considerável de tempo.

#### **4.6 Considerações finais**

Por fim, as funcionalidades apresentadas permitem conhecer os resultados obtidos no desenvolvimento da aplicação de análise de ECGs proposta, de forma que os objetivos especificados possam ser verificados e validados. Todas as funcionalidades desenvolvidas têm como objetivo principal facilitar o trabalho diário dos profissionais de saúde, tornando-se uma ferramenta útil na análise de eletrocardiogramas. Entretanto, o desenvolvimento do *software* de análise de ECGs faz parte de um processo longo e, como tal, necessita de tempo e de uma equipe de profissionais envolvidos. Por esta razão, algumas funcionalidades não estão disponíveis na primeira versão. Estas funcionalidades poderão ser melhor exploradas em trabalhos futuros, sendo exemplificadas no Capítulo a seguir, juntamente com as conclusões e contribuições deste trabalho.

## 5 CONTRIBUIÇÕES, CONCLUSÕES E TRABALHOS FUTUROS

### 5.1 Contribuições

Durante todo o ciclo de desenvolvimento da solução proposta, o principal objetivo que direcionou e motivou nas escolhas tomadas foi a de disponibilizar remotamente os recursos que contribuam no auxílio ao diagnóstico de doenças cardíacas, em especial as detectáveis via ECG. As contribuições obtidas são sintetizadas nas próximas subseções.

#### 5.1.1 Suporte a DICOM-ECG e aECG

Uma das maiores dificuldades encontradas durante o desenvolvimento do *Middleware* é o fato de muitos eletrocardiógrafos seguirem diferentes formatos para apresentação dos dados, sendo necessário pelos objetivos específicos para atender a esse conjunto de exames. Além disto, existem casos em que sua codificação não é proprietária, impossibilitando frequentemente integrá-lo à ferramenta.

Para garantir a interoperabilidade entre os diversos sistemas existentes é necessário conhecer os padrões de exames utilizados/disponibilizados pelos fabricantes de equipamentos de ECG. Após um vasto estudo, optou-se por disponibilizar através de um servidor a leitura e escrita dos formatos de exames: DICOM-ECG e aECG. Um ponto importante a destacar é a ausência de *Web services* disponibilizados que realizam esta atividade de interpretação dos ECGs. A arquitetura foi elaborada de forma a facilitar a adição de novos formatos, independentemente de serem públicos ou particulares.

#### 5.1.2 Detecção do Complexo QRS, ondas P e T

A detecção do complexo QRS, ondas P e T, ao longo dos sinais de um eletrocardiograma, é um passo importante na extração de informações úteis presentes neste exame. A partir destes dados pode-se obter, por exemplo: total de batimentos, duração do QRS (relacionado ao bloqueio completo do ramo direito) duração da onda P (relacionado à doença arterial coronariana) e duração da onda T (relacionado à sobrecarga atrial).

Este trabalho contribui neste aspecto a partir da disponibilização da detecção dessas marcações através de *Web services*, servindo assim como um *plugin* para aplicações de análise de ECGs. Até o momento em que este trabalho foi redigido, percebeu-se a ausência de serviços



*online* fornecendo a identificação destas marcações.

### **5.1.3 Sistema Web para geração de laudos de ECGs online**

Ao final deste trabalho foi disponibilizado um sistema para análise de eletrocardiogramas, permitindo que médicos e pacientes cadastrados visualizem exames de ECGs, bem como diversas informações extraídas a partir da interpretação dos dados coletados. Todos os recursos disponibilizados estão em conformidade com os requisitos levantados na fase inicial do projeto, como podem ser vistos no Capítulo 3. Destacam-se os seguintes recursos:

- I. Importação de exames nos formatos DICOM-ECG e aECG;
- II. Exportação de exames nos formatos DICOM-ECG e aECG;
- III. Visualização de exames;
- IV. Cadastro e visualização das marcações complexo QRS, ondas P e T, para um dado canal de um exame de ECG;
- V. Cadastramento das anotações ao longo dos canais de um dado exame;
- VI. Atualização e impressão de laudos dos exames;
- VII. Pode ser executado nos mais diversos dispositivos existentes, desde que este possua um *browser*.

## **5.2 Conclusões**

Neste trabalho foi apresentado um *Middleware* utilizado para fornecer *Web services* e abstrair as complexidades existentes na análise de exames ECGs. A fim de validar, o *Middleware* proposto, e contribuir com o Hospital Universitário Walter Cantídio, fornece-se também, uma aplicação web para geração de laudos remotos. Todo o desenvolvimento foi realizado utilizando tecnologias distribuídas gratuitamente e utilizou-se linguagens de programação multiplataforma.

Durante o desenvolvimento desta dissertação, foi possível identificar que grande parte dos eletrocardiógrafos seguem padrões proprietários, o que dificulta a existência de um nicho de desenvolvimento voltado para *softwares* de análise de eletrocardiogramas. No intuito de garantir o mínimo de interoperabilidade com sistemas existentes, foi disponibilizado suporte aos formatos DICOM-ECG e aECG, que são os formatos públicos mais utilizados nos eletrocardiógrafos.

A partir da análise de desempenho realizada no Capítulo 4, foi possível identificar a necessidade de um *hardware* mais moderno e da necessidade da utilização de um banco de dados

não relacional (NoSQL) para armazenar os dados dos sinais de ECGs, os quais representam o maior volume de dados do sistema. Adicionalmente, foram validadas as tecnologias sugeridas e a interação entre as partes comunicantes, garantindo que futuros sistemas baseados no presente trabalho apresentem um baixo risco de implantação.

Conclui-se também que as soluções desenvolvidas atendem aos requisitos propostos, podendo em conjunto serem utilizados como alternativa a sistemas proprietárias disponíveis atualmente no mercado.

### **5.3 Trabalhos futuros**

Baseado nos resultados obtidos, novos trabalhos e pesquisas relacionados soluções para análise de eletrocardiograma podem ser desenvolvidos. Adicionalmente, os resultados demonstram a capacidade das técnicas utilizadas, assim como a viabilidade do uso de tecnologias de baixo custo. Partindo desse pressuposto, alguns trabalhos futuros podem ser elencados:

- Adição de novos algoritmos para identificação do complexo QRS, ondas P e T, proporcionando assim a utilização de um algoritmo com melhor resultado dependendo das características do exame;
- Utilização combinada de soluções de banco de dados relacional e NoSQL para melhorar o desempenho da solução;
- Adição de algoritmos para auxílio ao diagnóstico a partir de exames de ECGs;
- Integração da solução com eletrocardiógrafos de baixo custo fornecidos no mercado;
- Adição da funcionalidade Holter (monitoração do ritmo cardíaco durante o exercício das funções normais do indivíduo em um período de 24 a 48 horas) online para acompanhamento de médicos e pacientes;
- Implantação da solução numa clínica focada em análise de eletrocardiogramas.

## REFERÊNCIAS

- ACHARYA, U. R.; JOSEPH, K. P.; KANNATHAL, N.; LIM, C. M.; SURI, J. S. Heart rate variability: a review. **Medical and biological engineering and computing**, Springer, v. 44, n. 12, p. 1031–1051, 2006.
- ARMSTRONG, I.; HASTON, W. Medical decision support for remote general practitioners using telemedicine. **Journal of Telemedicine and Telecare**, SAGE Publications Sage UK: London, England, v. 3, n. 1, p. 27–34, 1997.
- AZEVEDO, D. Iniciação a eletrocardiografia. **ArtMed**, 1999.
- BOND, R. R. A review of ecg storage formats. **International journal of medical informatics** **80.10**, IEEE, p. 681–697, 2011.
- CATALANO, J. T. **Guide to ECG analysis**. [S.l.]: Lippincott Williams & Wilkins, 2002.
- CHESNOKOV Y. C.; NERUKH, D. G. R. C. Individually adaptable automatic qt detector. **Computers in Cardiology**, IEEE, 2006.
- COULOURIS, G. F.; DOLLIMORE, J.; KINDBERG, T. **Distributed systems: concepts and design**. [S.l.]: pearson education, 2005.
- CURY, L. K. P.; SIQUEIRA, E. B.; GOMES, T. S. Um software para a análise de eletrocardiograma (ecg). 2011.
- EMERSON, S.; CHOI, Y.-K.; HWANG, D.-Y.; KIM, K.-S.; KIM, K.-H. An oauth based authentication mechanism for iot networks. In: IEEE. **Information and Communication Technology Convergence (ICTC), 2015 International Conference on**. [S.l.], 2015. p. 1072–1074.
- ETTINGER, M. van; LIPTON, J.; WIJS, M. de; PUTTEN, N. van der; NELWAN, S. An open source ecg toolkit with dicom. In: IEEE. **Computers in Cardiology, 2008**. [S.l.], 2008. p. 441–444.
- FILHO, O. F. F. **Serviços semânticos: uma abordagem restful**. Tese (Doutorado) — Universidade de São Paulo, 2011.
- GINEFRA, P. A evolução do eletrodo no registro dos potenciais elétricos cardíacos: um pouco de história. **Rev SOCERJ**, v. 20, n. 3, p. 248–250, 2007.
- HOLTER, N. J. New method for heart studies. *Science*, p. 1214–1220, 1961.
- HOPKINS, J. **hl7aECG**. 2015. Acesso em: 01 de Janeiro 2017. Disponível em: <<https://github.com/CardioVascular-Research-Group/hl7aECG>>.
- KAM, K. L.; RAMANE, V.; OOI, C. Y.; HAU, Y. W. Development of platform-independent web-based telecardiology application for pilot case study. In: IEEE. **Biomedical Engineering and Sciences (IECBES), 2014 IEEE Conference on**. [S.l.], 2014. p. 391–396.
- KUMAR, M. A.; SRINIVASAN, A.; BUSSA, N. Html5 powered web application for telecardiology: A case study using ecgs. In: IEEE. **Point-of-Care Healthcare Technologies (PHT), 2013 IEEE**. [S.l.], 2013. p. 156–159.

- LAGUNA, P.; MARK, R. G.; GOLDBERG, A.; MOODY, G. B. A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ecg. In: IEEE. **Computers in Cardiology 1997**. [S.l.], 1997. p. 673–676.
- LECHETA, R. R. **Web Services RESTful: Aprenda a criar web services restful em java na nuvem do google**. [S.l.]: Novatec Editora, 2015.
- LENGYEL, L. **Electrocardiografia clinica**. [S.l.]: SARVIER, 1974.
- LEUTHEUSER, H.; GRADL, S.; ANNEKEN, L.; ARNOLD, M.; LANG, N.; ACHENBACH, S.; ESKOFIER, B. M. Instantaneous p-and t-wave detection: Assessment of three ecg fiducial points detection algorithms. In: IEEE. **Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on**. [S.l.], 2016. p. 329–334.
- MADEIRO, J. P. V. **Detecção e segmentação automática de batimentos cardíacos do Eletrocardiograma por modelagem matemática e combinação das transformadas Wavelet e Hilbert**. Tese (Doutorado) — Tese de Doutorado, Programa de Pós-Graduação em Engenharia de Teleinformática, DETI/UFC., 2013.
- MAIA, C. E. M. **Central de tele-ECG baseada em WEB**. Dissertação (Mestrado) — Universidade de Aveiro, 2008.
- MOODY, G. B.; MARK, R. G. The mit-bih arrhythmia database on cd-rom and software for use with it. In: IEEE. **Computers in Cardiology 1990, Proceedings**. [S.l.], 1990. p. 185–188.
- MORRIS, F.; BRADY, W. J.; CAMM, A. J. **ABC of clinical electrocardiography**. [S.l.]: John Wiley & Sons, 2009. v. 93.
- NEMA. **National Electrical Manufacturers Association**. 2017. Acesso em: 10 de Janeiro de 2017. Disponível em: <<https://www.nema.org>>.
- OLIVEIRA, A. T. *et al.* Detecção do complexo qrs em sinais cardíacos utilizando fpga. Campinas, SP, 2009.
- PASTORE, C.; PINHO, C.; GERMINIANI, H.; SAMESIMA, N.; MANO, R. Diretrizes da sociedade brasileira de cardiologia sobre análise e emissão de laudos eletrocardiográficos. **Arquivos Brasileiros de Cardiologia**, SciELO Brasil, v. 106, n. 4, p. 1–19, 2016.
- PIXELMED. **PixelMed Publishing**. 2016. Acesso em: 01 de Janeiro 2017. Disponível em: <<http://www.pixelmed.com/dicomtoolkit.html>>.
- SOLAPURKAR, P. Building secure healthcare services using oauth 2.0 and json web token in iot cloud scenario. In: IEEE. **Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on**. [S.l.], 2016. p. 99–104.
- TANENBAUM, M. V. S. A. S. **Sistemas Distribuídos: Princípios e paradigmas**. [S.l.]: Pearson Prentice Hall, 2007.
- VALENTE I. R. S., P. C. C. J. M. S. **Registro Eletrônico de Saúde Baseado em Software Livre e Integrado a Sistemas de Apoio ao Diagnóstico**. 2010.
- W3C. **Soap specifications**. 2007. Acesso em: 10 de Janeiro 2017. Disponível em: <<https://www.w3.org/TR/soap/>>.

## APÊNDICE A – SISTEMAS DISTRIBUÍDOS

O desenvolvimento tecnológico dos microprocessadores na década de 80, em conjunto com o alto grau de sofisticação que os protocolos e as redes de comunicação em geral atingiram, foram os dois adventos principais que culminaram no surgimento de uma nova tecnologia, a dos Sistemas Distribuídos.

Segundo Coulouris *et al.* (2005) um sistema distribuído pode ser definido como sendo aquele no qual os componentes de hardware ou software, localizados em computadores interligados em uma rede, se comunicam e coordenam suas ações apenas enviando mensagens entre si. Essa simples definição cobre todo o conjunto de sistemas computacionais, nos quais computadores ou outros dispositivos de comunicação, ligados em rede, podem ser distribuídos de maneira útil, compartilhando recursos e informações.

De forma similar em Tanenbaum (2007) são definidos sistemas distribuídos como um conjunto de computadores independentes que se apresenta a seus usuários com um sistema único e coerente. Segundo o autor essa definição tem vários aspectos importantes. O primeiro é que um sistema distribuído consiste em componentes (isto é, computadores) autônomos. Um segundo aspecto é que os usuários, sejam pessoas ou programas, acham que estão tratando com um único sistema.

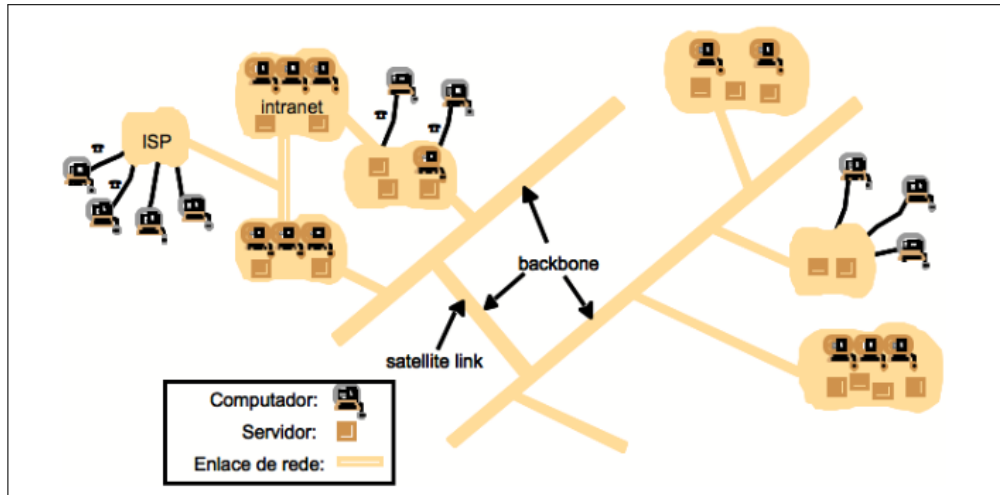
Como estabelecer essa colaboração entre os componentes é o cerne do desenvolvimento de sistemas distribuídos. É importante observar que nenhuma premissa é adotada em relação ao tipo de computadores. Em princípio, até mesmo dentro de um único sistema, eles poderiam variar desde computadores centrais (mainframes) de alto desempenho até pequenos nós em redes de sensores. Dá mesma maneira, nenhuma premissa é adotada quanto ao modo como os computadores são interconectados.

A Internet moderna é um conjunto de redes de computadores interligadas, com uma variedade de tipos que aumenta cada vez mais e que agora inclui, por exemplo, uma grande diversidade de tecnologias de comunicação sem fio, como WiFi, WiMAX, Bluetooth e redes de telefonia móvel. O resultado é que a interligação em redes se tornou um recurso pervasivo, e os dispositivos podem ser conectados a qualquer momento e em qualquer lugar.

A Figura 33 ilustra uma parte típica da Internet. Os programas em execução conectados a ela interagem trocando mensagens através de um meio de comunicação comum. O projeto e a construção dos mecanismos de comunicação da Internet (os protocolos Internet) são uma realização técnica importante, permitindo que um programa em execução em qualquer lugar

envie mensagens para programas em qualquer outro lugar abstraído a grande variedade de tecnologias previamente mencionadas.

Figura 33 – Exemplo de sistema distribuído



Fonte – (COULOURIS *et al.*, 2005)

O fato de ser possível, tecnologicamente falando, montar sistemas distribuídos não quer dizer necessariamente que seja viável sua implantação. Na literatura são discutidas quatro metas importantes que devem ser cumpridas no desenvolvimento de um sistema distribuído. Um sistema distribuído deve oferecer fácil acesso aos seus recursos; deve ocultar razoavelmente bem o fato de que os recursos estão distribuídos por uma rede; deve ser aberto e deve poder ser expandido (TANENBAUM, 2007).

### A.1 Acesso a recursos

Os sistemas distribuídos têm como principal meta facilitar aos usuários, programadores e aplicações o acesso aos recursos (de hardware e software) remotos e o compartilhamentos destes, de maneira segura, controlada e eficiente. Existem algumas razões plausíveis para que as organizações queiram compartilhar recursos. Entre elas pode-se citar a principal: diminuição dos custos.

Conectar usuários e recursos também facilita a colaboração e a troca de informações, o que é claramente ilustrado pelo sucesso da Internet com seus protocolos simples para trocar arquivos, correio, documentos, áudio e vídeo.

Com o aumento da conectividade aumenta o compartilhamento de informações e recursos, sendo necessário cuidados extra com a segurança. Na maioria das vezes, os sistemas

oferecem pouca proteção contra intrusão na comunicação. Senhas e outras informações sensíveis muitas vezes são enviadas como texto comum, isto é, não são nem criptografadas.

## A.2 Sistemas abertos

Uma meta que deve ser atingida na construção de sistemas distribuídos é a abertura. Um sistema distribuído aberto é um sistema que oferece serviços de acordo com regras padronizadas que descrevem sua sintaxe e semântica (TANENBAUM, 2007). Diz-se ainda, que um sistema é aberto quando ele pode ser estendido e reimplementado de várias maneiras (COULOURIS *et al.*, 2005).

No caso de sistemas distribuídos, em geral os serviços são especificados por meio de interfaces, que costumam ser descritas em uma *Interface Definition Language* (IDL). Estas definições explicitam os nomes das funções, os tipos de parâmetros, os valores de retorno, as possíveis exceções e assim por diante.

De acordo com Tanenbaum (2007) interoperabilidade caracteriza até que ponto duas implementações sistemas ou componentes de fornecedores distintos devem coexistir e trabalhar em conjunto, com base na mera confiança mútua nos serviços de cada um, especificados por um padrão comum.

Segundo Coulouris *et al.* (2005) resumidamente temos:

- Os sistemas abertos são caracterizados pelo fato de suas principais interfaces serem publicadas.
- Os sistemas distribuídos abertos são baseados na estipulação de um mecanismo de comunicação uniforme e em interfaces publicadas para acesso aos recursos compartilhados.
- Os sistemas distribuídos abertos podem ser construídos a partir de hardware e software heterogêneo, possivelmente de diferentes fornecedores. Para que um sistema funcione corretamente, a compatibilidade com o padrão publicado deve ser cuidadosamente testada e verificada.

### A.2.1 Transparência

Em seu trabalho Coulouris *et al.* (2005) define transparência como a ocultação, para um usuário final ou para um programador de aplicativos, da separação dos componentes em um sistema distribuído, de modo que o sistema seja percebido como um todo, em vez de uma

Tabela 6 – Formas de transparência (TANENBAUM, 2007)

<b>Transparência</b>	<b>Descrição</b>
Acesso	Oculto diferenças na representação de dados e no modo de acesso a um recurso
Localização	Oculto o lugar em que um recurso está localizado
Migração	Oculto que um recurso pode ser movido para outra localização
Relocação	Oculto que recurso pode ser movido para uma outra localização enquanto em uso
Replicação	Oculto que um recurso é duplicado
Concorrência	Oculto que um recurso pode ser compartilhado por diversos usuários concorrentemente.
Falha	Oculto a falha e a recuperação

coleção de componentes independentes.

Para Tanenbaum (2007) um sistema distribuído é dito transparente se ele é capaz de se apresentar a usuários e aplicações como se fosse apenas um único sistema de computador.

A tabela 6 apresenta os principais tipos de transparência apontados pela literatura.

Como exemplo clássico de transparência podemos citar o Netflix, um sistema distribuído que fornece interface com usuário na Web e cujos os dados estão geograficamente distribuídos. Para cada um dos usuários é transparente o acesso aos dados e filmes, uma vez que mesmo a base de dados e os filmes estando distribuídos em vários servidores no mundo, a aplicação não deixa que o usuário perceba este detalhe.

### **A.3 Web services**

A área de Tecnologias da Informação e Comunicação é extremamente dinâmica, inovações acontecem cotidianamente, novos softwares são escritos nas mais diversas linguagens, em diferentes plataformas e para os mais diversos fins. Esta diversidade de soluções promovem desafios e para viabilizar determinado negócio, algumas destas tecnologias distintas devem interagir para atingir um objetivo comum. É neste cenário que os *Web services* têm um grande destaque.

O emprego de serviços web teve um grande impulso com o surgimento da WEB 2.0, fase na qual o conteúdo deixou de ser meramente estático para tornar-se dinâmico, dando início as aplicações web mais robustas, permitindo que os usuários passassem de expectadores a agentes ativos na construção de conteúdo. Porém, simplesmente gerar conteúdo não faz sentido se o mesmo não for compartilhado, e é justamente no compartilhamento de conteúdo entre



aplicações web que o emprego de serviços web popularizou-se (FILHO, 2011).

Como o nome sugere, um serviço web é um tipo de aplicação Web, ou seja, uma aplicação normalmente entregue sobre o protocolo HTTP. Um serviço web é, portanto, uma aplicação distribuída, cujos componentes podem ser implementados e executados em dispositivos distintos como por exemplo computadores, *smartphones* e outros dispositivos.

De acordo com Lecheta (2015) *Web services* são utilizados como forma de integração e comunicação de sistemas, de modo que um sistema possa realizar uma chamada para um serviço de outro sistema a fim de obter informações. Estas chamadas podem enviar e receber informações em diversos formatos, sendo que atualmente os mais populares são XML e *JavaScript Object Notation* (JSON).

Partindo para uma abordagem tipicamente SOAP de *Web services*, conforme definido pelo W3C (2007) um *Web service* é um sistema de software designado para suportar a interação de máquina-a-máquina de forma interoperável sobre uma rede. Tendo uma interface descrita em um formato processável por máquinas (especificamente *Web Services Description Language* (WSDL)). Outros sistemas interagem com o serviço web da maneira indicada por sua descrição usando mensagem SOAP, tipicamente empregando HTTP com serialização em XML em conjunto com outros padrões Web.

De uma forma mais detalhada um Serviço Web pode ser entendido como uma interface de rede posicionada entre o código da aplicação e o usuário deste código, atuando como uma camada de abstração, possibilitando o acesso a qualquer aplicação que conheça esta interface.

Em geral, os serviços web podem ser divididos basicamente em dois grupos: os serviços baseados em SOAP e os *Representational State Transfer* (REST). Apesar da separação, a distinção entre esses grupos não é nítida, pois, como será explanado nas sessões subsequentes, um serviço baseado em SOAP entregue sobre HTTP é um caso especial dos serviços REST.

### **A.3.1 REST**

Um sistema RESTful pode ser implementado em qualquer arquitetura de rede disponível de forma que, não se faz necessário a criação de novas tecnologias ou protocolos de rede. núcleo da abordagem REST consiste na percepção de que, apesar termo transporte em seu nome, o HTTP consiste em uma API e não um simples protocolo de transporte. Um sistema no estilo REST é denominado RESTful e possui as seguintes características:

- Deve ser um sistema cliente-servidor.
- Tem que ser independente de estado (*stateless*), ou seja, cada requisição deverá ser independente das outras.
- A infraestrutura de rede deve suportar caches em diferentes níveis.
- Cada recurso deve ter um endereço exclusivo e um ponto de acesso válido.
- Deve ser concebido em camadas e deve suportar escalabilidade.

Segundo Lecheta (2015) o REST é uma técnica de desenvolvimento de *Web service* fortemente baseada nos métodos do protocolo HTTP:

- *GET*: lê um recurso.
- *POST*: cria um recurso a partir dos dados requisitados.
- *PUT*: atualiza um recurso a partir dos dados requisitados.
- *DELETE*: remove um recurso.

Em termos gerais, um cliente RESTful faz uma solicitação a um serviço (*POST*, *GET*, *DELETE* ou *PUT*), como por exemplo, listar todos os clientes. Se este pedido for bem sucedido, uma representação do recurso será enviado do servidor que hospeda o serviço até o cliente solicitante.

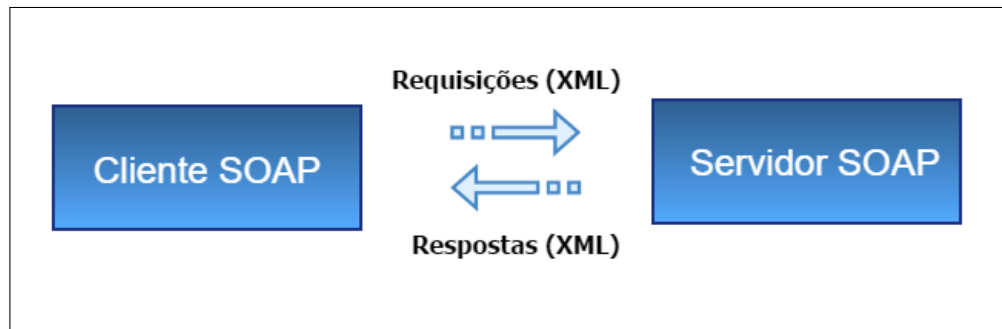
### A.3.2 SOAP

O SOAP é um protocolo de comunicação de *Web services* descritos por um WSDL. Na prática, o SOAP é um grande arquivo XML trafegando entre os sistemas para fazer a comunicação (LECHETA, 2015).

Um típico exemplo é apresentado na Figura 34, o qual um cliente envia requisições ao servidor e recebe de volta respostas. Cada uma dessas solicitações e respostas que trafegam pela rede são feitas com mensagens do tipo SOAP, isto é, usando o XML como estrutura.

Como o SOAP é um grande XML, ele começou a perder espaço para *Web services* RESTful, que apresentam um sintaxe mais limpa e podem enviar e receber informações em formatos mais leves, por exemplo JSON. Especialmente no mundo mobile, o qual a telemedicina se insere, os aplicativos precisam economizar recursos ao trafegar informações, e essa é a principal justificativa para o SOAP ter perdido espaço para REST.

Figura 34 – Exemplo de comunicação via SOAP entre cliente e servidor



Fonte – Autoria própria

#### A.4 Arquitetura Cliente-Servidor

Apesar da falta de consenso sobre muitas questões de sistemas distribuídos, há uma delas com a qual muitos pesquisadores e praticantes concordam: pensar em termos de clientes que requisitam serviços de servidores simplifica entender e gerenciar a complexidade de sistemas distribuídos (TANENBAUM, 2007).

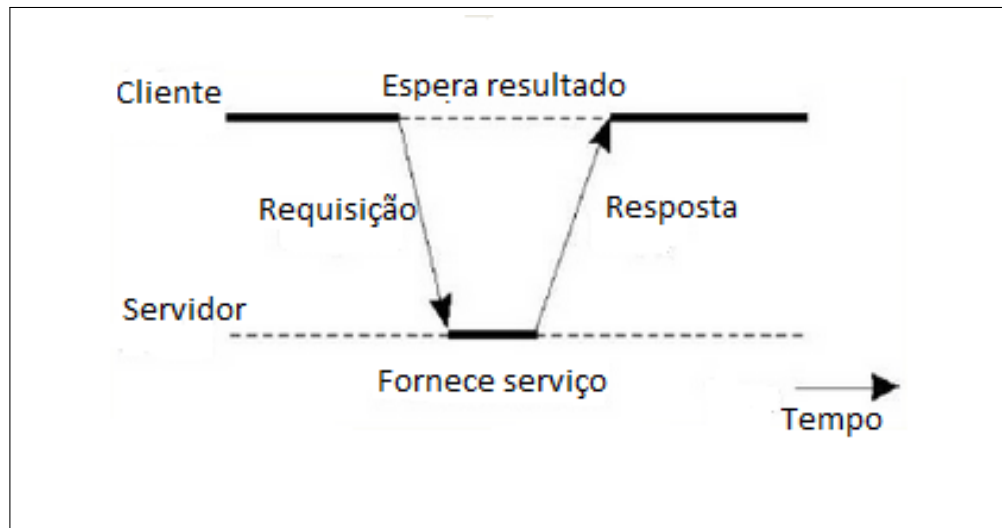
Segundo Coulouris *et al.* (2005) a arquitetura cliente-servidor é a mais citada quando se discute os sistemas distribuídos, sendo historicamente a mais importante e nos dias de hoje continua amplamente utilizada.

Nesse modelo, um servidor é um processo que implementa um serviço específico, por outro lado, o cliente é um processo que requisita um serviço de um servidor enviando-lhe uma requisição e, na sequência, esperando pela resposta do servidor, como pode ser visto na Figura 35.

Uma forma mais simples de visualizar esta arquitetura é através de níveis. sendo os mais comuns:

- 2 níveis: caracteriza os sistemas clientes/servidores pelos quais o cliente pede um recurso e o servidor responde diretamente ao pedido, utilizando seus próprios recursos. Isto significa que o servidor não requer outro aplicativo para realizar parte do serviço.
- 3 níveis: nesta arquitetura, existe um nível intermediário, o que significa que ela está compartilhada entre um cliente, ou seja, o computador que solicita recursos, equipado com um interface de usuário (geralmente um navegador) encarregado da apresentação, e o servidor de aplicativo (também chamado de software intermediário), cuja tarefa é proporcionar os recursos solicitados, mas que recorre a um outro servidor. O servidor de dados, que fornece os dados necessários do servidor do aplicativo.

Figura 35 – Interação entre cliente e servidor

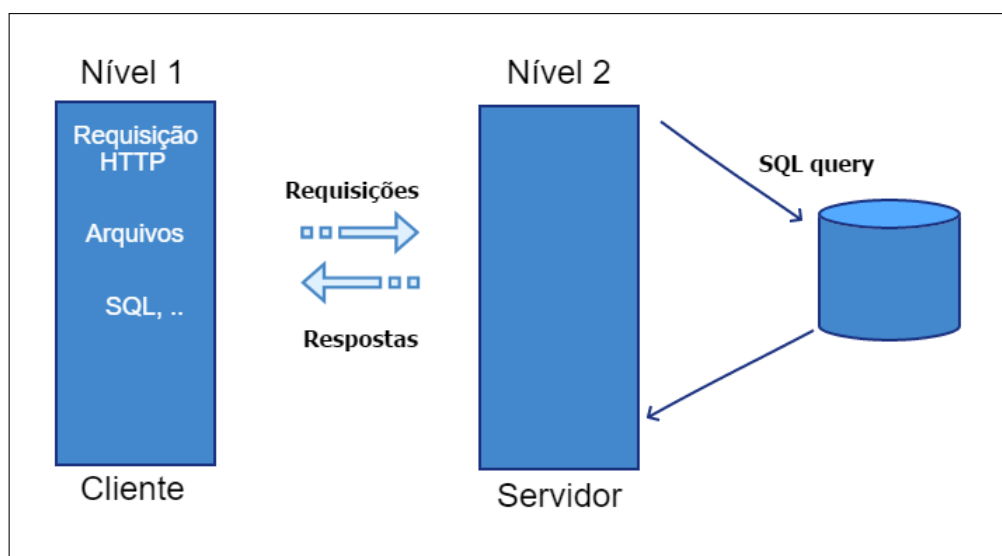


Fonte – (TANENBAUM, 2007)

- Multiníveis: na arquitetura em 3 níveis, cada servidor (níveis 2 e 3) efetua uma tarefa (um serviço) específica. Assim sendo, um servidor pode utilizar os serviços de um ou vários servidores para propor o seu próprio serviço. Conseqüentemente, a arquitetura em três níveis é potencialmente uma arquitetura de n (vários) níveis.

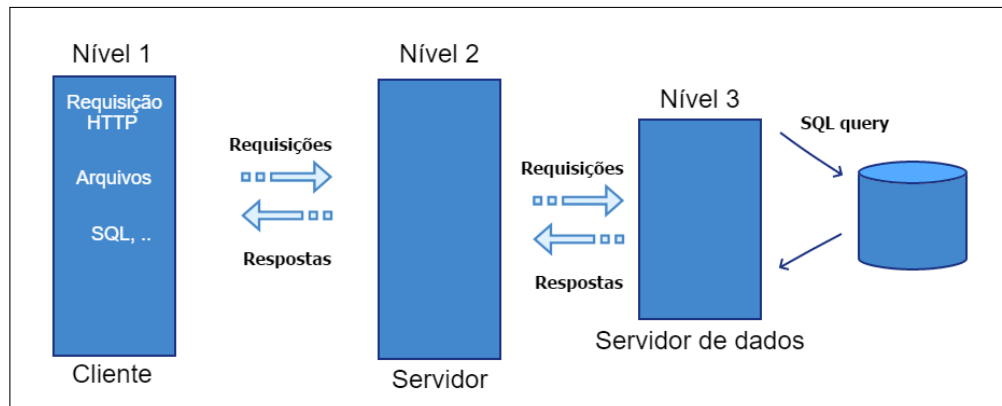
As Figuras 36, 37 e 38 apresentam a arquitetura cliente-servidor e sua representação em níveis.

Figura 36 – Cliente-Servidor 2 níveis



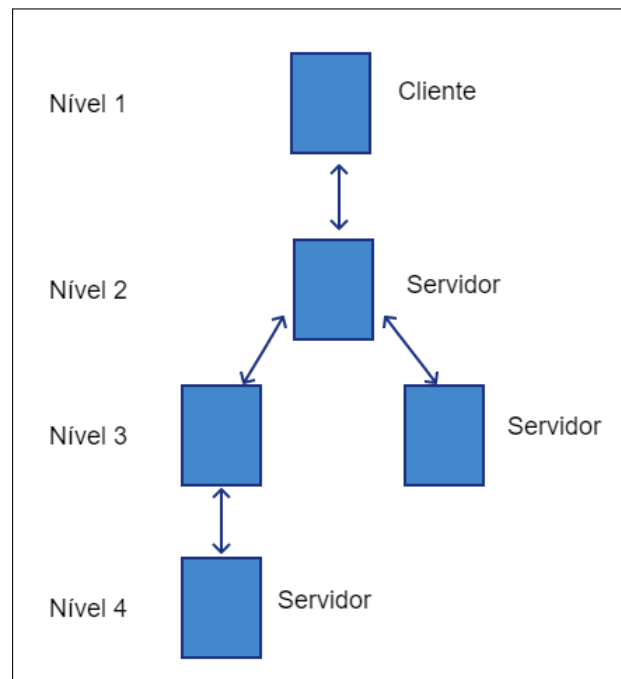
Fonte – Adaptada de (TANENBAUM, 2007)

Figura 37 – Cliente-Servidor 3 níveis



Fonte – Adaptada de (TANENBAUM, 2007)

Figura 38 – Cliente-Servidor n níveis



Fonte – Adaptada de (TANENBAUM, 2007)

## APÊNDICE B – WEB SERVICES DISPONIBILIZADOS

Este capítulo apresenta os principais *Web services* desenvolvidos ao longo do trabalho e serve como guia tanto para utilização (abertura do sistema) quanto referência aos recursos disponibilizados. Como toda a comunicação com o *Middleware* precisa de um *token* passado no corpo da mensagem utilizou-se como método de transporte o POST. As Tabelas 7 e 8, apresentam os principais serviços fornecidos pelo *Middleware*.

Tabela 7 – Serviços relacionados a *Download/Upload* de exames

<b>Descrição</b>	Serviço para enviar exame nos padrões DICOM-ECG e HL7 aECG
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/uploadFile">http://lesc.ufc.br/ecgweb/ecg/uploadFile</a>
<b>Método</b>	POST
<b>Request Parameter</b>	token="valor", file="MultipartFile"
<b>Resposta</b>	{"ecgFileKey":"xxxx"}
<b>Descrição</b>	Serviço para Obter exame em formato aECG
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/getHL7Exam">http://lesc.ufc.br/ecgweb/ecg/getHL7Exam</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token"="XXX", "ecgFileKey":"XXX"}
<b>Resposta</b>	{"link":"xxxx"}
<b>Descrição</b>	Serviço para Obter exame em formato DICOM-ECG
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/getDicomExam">http://lesc.ufc.br/ecgweb/ecg/getDicomExam</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token"="XXX", "ecgFileKey":"XXX"}
<b>Resposta</b>	{"link":"xxxx"}

Fonte – Autoria própria.

Tabela 8 – Serviços para *Upload* e *Download* de exames

<b>Descrição</b>	Serviço de autenticação e autorização
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/login">http://lesc.ufc.br/ecgweb/login</a>
<b>Método</b>	POST
<b>Request Body</b>	{"ID":"XXX", "password":"XXX"}
<b>Resposta</b>	{"token":"UUID", "refreshToken":"UUID"}
<b>Descrição</b>	Serviço para obter o Complexo QRS
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/channel/getQrsComplex">http://lesc.ufc.br/ecgweb/ecg/channel/getQrsComplex</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token":"XXX", "algorithmId":"1 2 3 ...", "signals":[{"idx":"XXX", "value":"Double XX"}, ...]}
<b>Resposta</b>	[{"firstIdx":"XX", "lastIdx":"XX", "peakIdx":"XX"}, ...]
<b>Descrição</b>	Serviço para obter as Ondas T
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/channel/getTWave">http://lesc.ufc.br/ecgweb/ecg/channel/getTWave</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token":"XXX", "algorithmId":"1 2 3 ...", "signals":[{"idx":"XXX", "value":"Double XX"}, ...]}
<b>Resposta</b>	[{"firstIdx":"XX", "lastIdx":"XX", "peakIdx":"XX"}, ...]
<b>Descrição</b>	Serviço para obter as Ondas P
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/channel/getPWave">http://lesc.ufc.br/ecgweb/ecg/channel/getPWave</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token":"XXX", "algorithmId":"1 2 3 ...", "signals":[{"idx":"XXX", "value":"Double XX"}, ...]}
<b>Resposta</b>	[{"firstIdx":"XX", "lastIdx":"XX", "peakIdx":"XX"}, ...]
<b>Descrição</b>	Serviço para plotar o gráfico das séries de RR
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/channel/getPlotRR">http://lesc.ufc.br/ecgweb/ecg/channel/getPlotRR</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token":"XXX", "signals":[{"idx":"XXX", "value":"Double XX"}, ...]}
<b>Resposta</b>	[{"xUnity":"MILLISECOND SECOND MINUTE HOUR", "yUnity":"MILLISECOND SECOND MINUTE HOUR", "plotRR":[{"x":"double value", "y":"double value"}, ...]
<b>Descrição</b>	Serviço para extrair informações gerais do ECG
<b>Link</b>	<a href="http://lesc.ufc.br/ecgweb/ecg/getEcgInformation">http://lesc.ufc.br/ecgweb/ecg/getEcgInformation</a>
<b>Método</b>	POST
<b>Request Body</b>	{"token":"XXX", "channelId":"XXX"}
<b>Resposta</b>	{       "heartRate": 73.72,       "heartRateUnity": "BPM",       "qrsDuration": 0.073,       "qrsDurationUnity": "SECOND",       "tWaveDuration": 0.127,       "tWaveDurationUnity": "SECOND",       "pWaveDuration": 0.05959,       "pWaveDurationUnity": "SECOND"     }