

UNIVERSIDADE DO SAGRADO CORAÇÃO

MARCELO DE PAULA LIMA

**SISTEMA DE MONITORAMENTO CARDÍACO PELA
WEB**

BAURU
2014

MARCELO DE PAULA LIMA

**SISTEMA DE MONITORAMENTO CARDÍACO PELA
WEB**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do Título de Bacharel em Ciência da Computação, sob orientação da Prof. Dr. Elvio Gilberto da Silva.

BAURU
2014

Lima, Marcelo de Paula.

L7327s

Sistema de monitoramento cardíaco pela WEB / Marcelo de Paula Lima. -- 2014.

58f. : il.

Orientador: Prof. Dr. Elvio Gilberto da Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Engenharia de Software. 2. Desenvolvimento de Sistemas. 3. Medicina Assistida. 4. Arduino. I. Silva, Elvio Gilberto da. II. Título.

MARCELO DE PAULA LIMA

SISTEMA DE MONITORAMENTO CARDÍACO PELA WEB

Trabalho de Conclusão de Curso apresentado ao centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do Título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva

Banca Examinadora:

Prof. Dr. Elvio Gilberto da Silva
Orientador

Prof. Esp. Alex Setolin Beirigo
Examinador

Prof. Me. Patrick Pedreira Silva
Examinador

Bauru, 08 de Dezembro de 2014.

Dedico este trabalho aos meus pais e a tudo que eles proporcionaram na minha vida, aos meus irmãos pelo apoio, aos professores me auxiliaram e me incentivaram.

AGRADECIMENTOS

Primeiramente, agradeço a Deus, que me deu coragem, força para sempre me dedicar aos estudos, ao conhecimento adquirido ao longo desses anos que foram fundamentais para que eu concluísse essa etapa da minha vida.

Agradeço também aos meus pais, pelo incentivo e apoio durante toda a minha vida, por acreditarem em mim e nunca deixarem eu desistir.

Aos meus irmãos que sempre estiveram ao meu lado me ajudando nos estudos e nas pesquisas durante ao longo do curso.

Agradeço ao Prof. Dr. Elvio Gilberto da Silva, pela paciência, entusiasmo e motivação que sempre demonstrou não só comigo mas com todos os seus alunos e a todos os demais professores que me ajudaram e compartilharam seus conhecimentos com todos nós durante esses últimos quatro anos.

"Que sua colheita seja abundante e eterna e o sorriso da felicidade e do sucesso enfeite os seus lábios."

(Lauro Trevisan)

RESUMO

Considerando os problemas que podem vir a ocorrer durante o processo de obtenção dos dados referentes ao exame de monitorização ambulatorial da pressão arterial, esse projeto consistiu em desenvolver um sistema de monitoramento da frequência cardíaca e da pressão arterial de um paciente, de modo que esse sistema receba os dados e disponibilize-os para que os usuários tenham acesso aos resultados do exame. A elaboração desse projeto pode ser definido por três partes: envio dos dados referentes a medição da pressão arterial do paciente através de uma placa microcontroladora, desenvolvimento de um Web Service, utilizando a linguagem de programação Java, e o desenvolvimento de um sistema web, também utilizando a linguagem de programação Java, o framework SpringMVC para o desenvolvimento de aplicações em Java e tendo como base os processos e métodos da engenharia de software. Este trabalho apresenta um referencial teórico sobre alguns assuntos que estão relacionadas ao exame de MAPA (Monitorização Ambulatorial da Pressão Arterial), tais como: Ciclo cardíaco, pressão arterial e sintomas e diagnóstico da hipertensão arterial. Além disso, também aborda as metodologias e processos relacionados ao desenvolvimento de software, como por exemplo, análise e requisitos de software, modelagem de software, e padrão de arquitetura de software.

Palavras-chave: Medicina Assistida por Computação Científica, JAVA, Arduino, Engenharia de Software.

ABSTRACT

Considering the problems that can occur during the process of obtaining the data for the examination will monitor ambulatory blood pressure. This project was to develop a monitoring system of cardiac rate and blood pressure of a patient, so that this system receives the data and make them available for users to have access to test results. The development of this project can be defined by three parts: sending the data for measuring blood pressure of the patient through a microcontroller board, development of a Web Service using the Java programming language, and the development of a web system also using the Java programming language, SpringMVC framework for developing applications based on Java and the processes and methods of software engineering. This paper presents a theoretical framework about some issues that are relacionadas to ABPM (Ambulatory Blood Pressure Monitoring), such as cardiac cycle, blood pressure and symptoms and diagnosis of hypertension. It also discusses the methods and processes related to software development, such as analysis and software requirements, software modeling, and software architecture standard.

Keywords: Medicine Assisted by Scientific Computing, JAVA, Arduino, Software Engineering.

LISTA DE ILUSTRAÇÕES

Figura 1 - Classificação dos níveis da pressão arterial.	18
Figura 2 - Cenário atual de desenvolvimento de software.....	24
Figura 3 - Processo de levantamento e análise de requisitos.	25
Figura 4 - Método VORD.....	26
Figura 5 - Desenvolvimento ideal de software.....	28
Figura 6 - Elementos do diagrama de caso de uso.....	31
Figura 7 - Exemplo de classe contendo atributos e métodos de um cliente.....	32
Figura 8 - Um diagrama simples exemplificando a relação entre Model, View e Controller.....	35
Figura 9 - Processo de uma requisição MVC.....	36
Figura 10 - Funcionamento de Web Service com REST.....	38
Figura 11 - Arduino Uno.....	39
Figura 12 - Funcionamento do protótipo.....	42
Figura 13 - Diagrama de caso de uso.....	43
Figura 14 - Diagrama de classes.....	44
Figura 15 - Acesso aos módulos do sistema conforme perfil do usuário.....	44
Figura 16 - Trecho do código utilizado para ler os valores medido no aparelho de medição da PA.....	46
Figura 17 - Resultado das medições no console da IDE do Arduino.....	47
Figura 18 - Classes do sistema utilizando o padrão MVC.....	48
Figura 19 - Tela inicial do sistema.....	50
Figura 20 - Tela inicial do sistema em um dispositivo móvel.....	51
Figura 21 - Características dos pacientes participantes do teste.....	52
Figura 22 - Resultado das medições dos três pacientes (pressão arterial - batimentos cardíacos).....	53
Figura 23 - Tela de visualização prévia dos resultados do exame.....	53
Figura 24 - Figura 25 - Gráficos disponíveis no relatório final do exame.....	54

LISTA DE ABREVIATURAS E SIGLAS

CERN - Centro Europeu de Pesquisas Nucleares
CGI - Comitê Gestor da Internet
DARPA - Defense Advanced Research Projects Agency
GoF - Gang of Four
GSM - Global System for Mobile Communication
GRASP - General Responsibility Assignment Software Patterns
MVC - Model–view–controller
HPSA - High-Speed Downlink Packet Access
HTTP - Hypertext Transfer Protocol
IBGE - Instituto Brasileiro de Geografia e Estatística
IDE - Integrated Development Environment
JSON - JavaScript Object Notation
JVM - Java Virtual Machine
MAPA - Monitorização Ambulatorial da Pressão Arterial
PA - Pressão Arterial
REST - Representational State Transfer
UML - Unified Modeling Language
URL - Uniform Resource Locator
XML - eXtensible Markup Language (Linguagem Extensível de Marcação)
WWW - World Wide Web

SUMÁRIO

1 INTRODUÇÃO	14
1.1.1 Objetivo geral	16
1.1.2 Objetivos específicos	16
2 SISTEMA CARDIOVASCULAR	17
2.1 CICLO CARDÍACO.....	17
2.2 PRESSÃO ARTERIAL.....	18
2.3 HIPERTENSÃO ARTERIAL	19
2.4 SINTOMAS E DIAGNÓSTICO	19
3 INFORMÁTICA APLICADA NA SAÚDE.....	21
4 ANÁLISE DE DESENVOLVIMENTO DE SOFTWARE.....	23
4.1 ANÁLISE DE REQUISITOS	24
4.2 DESENVOLVIMENTO DO PROJETO	27
4.3 METODOLOGIA, PROCESSOS e MÉTODOS DE DESENVOLVIMENTO DE SOFTWARE	27
4.4 MODELAGEM DE SOFTWARE	30
4.4.1 Diagrama de Caso de Uso	30
4.4.2 Diagrama de Classes	31
4.5 DESIGN PATTERNS E O DESENVOLVIMENTO ÁGIL DE SISTEMAS.....	32
4.6 FERRAMENTA PARA DESENVOLVIMENTO EM JAVA WEB.....	33
4.7 PADRÃO DA ARQUITETURA DE SOFTWARE MVC.....	34
4.8 O QUE É UM WEB SERVICE	36
4.9 PLATAFORMA ARDUÍNO.....	39
5 METODOLOGIA.....	41
5.1 LEVANTAMENTO DE DADOS DO PROJETO	42
5.2 CONECTANDO O ARDUÍNO AO WEB SERVICE	45

5.3	DESENVOLVIMENTO DO SISTEMA.....	47
6	RESULTADOS E DISCUSSÃO	50
6.1	DESENVOLVIMENTO DO SISTEMA.....	50
6.2	TESTES COM OS USUÁRIOS	51
7	CONSIDERAÇÕES FINAIS	55
8	TRABALHOS FUTUROS	57

1 INTRODUÇÃO

Atualmente no Brasil a saúde é um dos problemas que mais preocupa os brasileiros. A falta de investimentos em políticas públicas para a prevenção de doenças, falta de médicos, equipamentos e sucateamento dos hospitais acabam colaborando para resultado negativo da saúde pública no Brasil. (ENTENDA, 2011). Segundo levantamento de dados do Instituto Brasileiro de Geografia e Estatística (BRASILEIROS..., 2014) realizado entre 23 de novembro de 2013 a 2 de dezembro de 2013, com 15.414 entrevistados em 727 municípios a área da saúde foi apontada como principal problema do país por 58% dos entrevistados.

Tentando sanar esse problema o governo federal em 2013 lançou o programa Mais Médicos,(Ministério da saúde, 2014), que visa ampliar o acesso ao atendimento médico em regiões do interior do país, como cidades isoladas no interior da Amazônia ou cidades onde existe algum déficit de profissionais e equipamentos para consultas e exames médicos. De modo geral, problemas de saúde requerem acompanhamento e exames médicos, que nem sempre estão disponíveis nesses lugares isolados. Embora os gastos públicos com a saúde tenha aumentado desde 1995, eles continuam menores se considerarmos proporcionalmente ao PIB do que em outros países subdesenvolvidos. (GOVERNO... , 2012).

Para Fleury e Ouverney (2007) acerca do questionamento sobre os problemas do SUS (Sistema Único de Saúde) a resposta sobre o verdadeiro problema está relacionada a uma melhoria da gestão dos recursos, com o governo federal, os estados e os municípios assumindo a responsabilidade sobre a gestão desses recursos e agindo de forma integrada em suas regiões.

Atualmente cerca de 20% a 30% da população brasileira adulta possui pressão alta sendo que 95% dos casos as pessoas possuem alguma predisposição para pressão alta. O consumo de alimentos com gordura, consumo exagerado de sal, obesidade e a falta de atividades físicas são fatores que contribuem para potencializar os fatores de riscos. De um modo geral o número de casos são maiores entre as pessoas do sexo feminino do que entre o sexo masculino. (ENTENDA, 2011).

A MAPA (Monitorização Ambulatorial da Pressão Arterial) é um exame útil tanto no diagnóstico e no acompanhamento da Hipertensão Arterial, que permite o

registro da pressão arterial durante 24 horas, ou mais, enquanto o paciente realiza suas atividades habituais durante o dia. Este exame pode fazer parte do protocolo de investigação diagnóstica da hipertensão arterial.

Para a realização do exame do MAPA o paciente precisa se deslocar até uma unidade saúde, onde é feita uma pré-avaliação e encaminhado para a realização do exame. Dependendo da região onde mora o paciente, o resultado desse exame pode levar até meses para sair, podendo o mesmo ser extraviado no meio do processo.

O gerenciamento efetivo da saúde, muitas vezes necessita o acesso à análise de informações para que se possa chegar as repostas de um determinado problema. A evolução científica do século XX promoveu crescimento exponencial de conhecimentos, especialmente na área médica. A alta velocidade de descobertas científicas gerou uma grande quantidade de dados a serem analisados. É neste ponto que a informática tornou-se uma aliada fundamental da medicina. (DOMENICIS JUNIOR, 2014).

A informática de uma maneira geral permite a digitalização de dados e o acesso a eles em tempo real, permitindo por exemplo que um médico acesse as informações de um paciente independente da distância em que ele se encontra permitindo que um profissional solicite uma segunda opinião a outro profissional facilitando os diagnósticos e evitando o deslocamento de pacientes e médicos desnecessariamente.

Segundo Alves (2014) a informática voltada para a saúde lida com os recursos, dispositivos e métodos necessários para otimizar a aquisição, armazenamento, recuperação e utilização das informações. As ferramentas e dispositivos da informática de saúde incluem não se resumem apenas computadores, mas também orientações clínicas, formal, terminologia médica, e sistemas de informação e comunicação.

Diante desse contexto, esse trabalho visa desenvolver um sistema que auxilie no monitoramento da pressão arterial através da web com o intuito de facilitar o acompanhamento médico em lugares de difícil acesso, possibilitando, assim, um diagnostico mais ágil e eficiente.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolver um protótipo de software que receba os dados enviados por um aparelho de medição de pressão através da web, utilizando a linguagem JAVA para o desenvolvimento do sistema e integrando o software através de um Web Service com uma plataforma de prototipagem eletrônica.

1.1.2 Objetivos específicos

- Levantar dados e especificações necessárias para o desenvolvimento de um sistema web para o cadastro de pacientes;
- Modelar as informações levantadas para o desenvolvimento e sistema do sistema para a consulta das informações;
- Implementar um serviço web para receber os dados enviados pelo monitor de pressão arterial;
- Definir os dados que serão apresentados para os pacientes e para os profissionais da área da saúde;
- Desenvolver de um sistema web de modo que seja de fácil visualização pelo usuário e que seja acessível por diferentes tipos de plataformas como computadores, tablets e smartphones;
- Realizar testes de medições da pressão arterial e envio dos dados para o site.

2 SISTEMA CARDIOVASCULAR

O sistema cardiovascular é responsável pela circulação do sangue através dos vasos e capilares pulmonares e sistêmicos com o propósito de troca de oxigênio, gás carbônico, nutrientes, produtos de degradação e água nos tecidos periféricos e nos pulmões. Ele é composto pelo coração e dois sistemas vasculares responsáveis pelas circulações sistêmica e pulmonar. (PERRY; POTTER, 2005).

O coração, por sua vez, possui os ventrículos direito e esquerdo que funcionam como bombas em série, ejetando sangue através de dois sistemas vasculares. A circulação pulmonar de baixa pressão, onde ocorre a troca gasosa (captação de oxigênio e liberação de gás carbônico pela hemoglobina circulante nas hemácias), e a circulação sistêmica que distribui sangue aos órgãos individuais, suprindo as suas demandas metabólicas. O fluxo e a pressão sanguínea estão sob intenso controle do sistema nervoso autônomo. (RÉA NETO et al., 2004)

2.1 CICLO CARDÍACO

Segundo Réa Neto *et al.* (2004) o ciclo inicia-se no nódulo sinoatrial com uma despolarização que leva à contração do átrio. Durante este tempo o fluxo sanguíneo no interior dos ventrículos é passivo, mas a contração atrial aumenta o seu enchimento em 20 a 30%. A sístole ventricular ocasiona o fechamento das valvas atrioventriculares (1ª bulha cardíaca) sendo que a contração é isométrica até que as pressões intraventriculares tornem-se suficientes para abrir as valvas pulmonar e aórtica, dando início à fase de ejeção.

O volume de sangue ejetado é conhecido como volume de ejeção. Ao final desta fase ocorre o relaxamento ventricular e o fechamento das valvas pulmonar e aórtica (2ª bulha cardíaca). Após o relaxamento isovolumétrico, as pressões ventriculares diminuem mais do que as pressões atriais. Isso leva à abertura das valvas atrioventriculares e ao início do enchimento ventricular diastólico. Todo o ciclo então se repete na sequência de outro impulso a partir do nodo sinoatrialtólico. (Bolfer, 2014).

2.2 PRESSÃO ARTERIAL

De acordo com Potter e Perry (2005) a pressão arterial consiste na força exercida pelo sangue sobre as laterais ou paredes de uma artéria sempre que ocorre um batimento cardíaco. O Sangue se move de uma área de alta pressão para uma área de baixa pressão. Pressão arterial ou sistêmica, a pressão sanguínea no sistema das artérias do corpo é um bom indicador de saúde cardiovascular. A contração do coração força o sangue sob alta pressão para dentro da aorta. O pico de pressão máxima quando ocorre a ejeção é a pressão arterial sistólica. Quando os ventrículos relaxam, o sangue que permaneceu nas artérias exerce uma pressão mínima exercida contra as paredes artérias a todo momento.

A unidade padrão para medição da pressão arterial é milímetros por mercúrio (mmHg). Sendo que ela indica a altura a qual a pressão arterial pode alcançar uma coluna de mercúrio. A pressão arterial registrada com a leitura da sistólica antes da diastólica (por exemplo, 120/80). A diferença entre a pressão sistólica e a diastólica é a pressão de pulso. Para uma pressão arterial de 120/80 a pressão do pulso é de 40. (PERRY; POTTER, 2008). A Figura 1 demonstra os a classificação dos níveis da pressão arterial divididos em categorias e indicando o período em que as avaliações médicas devem ocorrer.

Figura 1 - Classificação dos níveis da pressão arterial.

Categoria	Sistólica	Diastólica	Acompanhamento recomendado
Ótima	< 120 e	< 80	Avaliação em 2 anos
Normal	< 130 e	<85	Avaliação em 2 anos
Normal Alta	130 - 139	85 - 89	Avaliação em 1 anos
Hipertensão			
Estágio 1	140 - 159	90 - 99	Avaliar em 2 meses
Estágio 2	160 - 179	100 - 109	Avaliar em 1 mês
Estágio 3	> 179	> 109	Avaliar em 1 semana

Fonte: Perry e Potter (2008, p. 396).

Nota: Adaptado pelo autor.

Conforme a Figura 1, a tabela demonstra que para se considerar uma pessoa com a pressão arterial normal, a pressão sistólica e diastólica devem ser inferiores a 120 mmHg por 80 mmHg, sendo que a unidade de medida utilizada para a medição da pressão arterial é milímetros por coluna de mercúrio. (PERRY; POTTER, 2008). Para os pacientes que possuem pressão arterial ótima, normal e normal alta, o

acompanhamento recomendado é de dois anos a um ano, já para os pacientes considerados hipertensos, o tempo para a avaliação varia de dois meses a uma semana.

2.3 HIPERTENSÃO ARTERIAL

De acordo com Smith e Timby (2005) o termo hipertensão arterial define a elevação sustentada da PA sistólica de 140 mmHg ou mais, e/ou uma pressão arterial diastólica sustentada de 90 mmHg.

Uma pessoa é considerada com hipertensão arterial quando existe uma anormalidade cardíaca e resulta em pressão elevada. A Hipertensão arterial pode ser dividida em duas categorias: hipertensão arterial essencial e hipertensão arterial secundária. (PERRY; POTTER, 2008).

2.4 SINTOMAS E DIAGNÓSTICO

Para Smith e Timby (2005) pacientes com hipertensão arterial, podem ser assintomáticos. O início da hipertensão arterial, considerado “o assassino silencioso”, costuma ser gradual. Pessoas com histórico familiar, obesos, fumantes com estilo de vida sedentários fazem parte de um grupo de alto risco de vir a desenvolver essa patologia.

A medida de pressão arterial é pré-requisito necessário para a verificação de pacientes com suspeita de elevação da pressão arterial ou com hipertensão arterial estabelecida, a qual assegura o diagnóstico e o manuseio corretos da condição de pressão arterial elevada. Além disso ele faz parte de uma etapa essencial para determinar apropriadamente a necessidade para o início de um tratamento para a doença assim como sua eficácia, tanto quanto para estimar o risco do desenvolvimento de complicações relacionadas à hipertensão. A abordagem usual para a medida da pressão arterial na prática diária está fundamentada em leituras convencionais auscultatórias obtidas no ambiente médico. (SILVA; ORTEGA; MION JUNIOR, 2008).

O achado mais obvio durante o exame físico é a elevação constante de uma ou de ambas as mensurações da PA. O pulso pode ser forte devido a contração ventricular. É possível que os pacientes apresentem sobrepeso. Eles podem exibir

face avermelhada em decorrência da congestão dos vasos sanguíneos superficiais. O edema periférico pode estar. Um exame oftalmológico talvez revele alterações vasculares nos olhos, hemorragias retinianas ou edema do nervo óptico, denominado papiledema. (PERRY; POTTER, 2005).

O uso da Monitorização Ambulatorial da Pressão Arterial para estabelecer os valores de normalidade das médias de pressão arterial e de outros parâmetros é um dos principais instrumentos para diagnóstico. Como uma de suas principais vantagens podemos citar que a MAPA é uma técnica não-invasiva que permite a monitorização da pressão arterial durante o sono. (SMITH; TIMBY, 2005).

O descenso da pressão arterial durante o sono ou repouso, uma característica natural do sistema cardíaco pode ser assim calculado: $(\text{média da pressão de vigília} - \text{média da pressão do sono}) \times 100 \div \text{média da pressão de vigília}$. Assim, de acordo com esse cálculo, os indivíduos podem ser classificados com descenso presente, atenuado, ausente ou acentuado, quando a redução da pressão entre os períodos de vigília e sono for $\geq 10\%$, $< 10\%$, $\leq 0\%$ e $\geq 20\%$, respectivamente. (KIKUYA, M; OHKUBO, T; ASAYAMA, K; et al, 2005).

Assim, podemos concluir que o uso da MAPA com o registro ao longo de 24 horas possibilita o melhor entendimento do comportamento da pressão arterial, assim como de outras variáveis que podem ser analisada, permitindo assim conhecer o impacto desses parâmetros sobre a estratificação do risco cardiovascular, sugerindo suas indicações para algumas situações específicas. (SILVA; ORTEGA; MION JUNIOR, 2008).

3 INFORMÁTICA APLICADA NA SAÚDE

A informática quando aplicada na saúde tem obtido excelentes resultados, como uma de suas vantagens podemos citar as técnicas não invasivas para a produção de imagem, como a ultrassonografia e a tomografia, que tem alterado sensivelmente o processo de diagnóstico médico. O surgimento de novos equipamentos digitais para a monitorização de pacientes, como analisadores automáticos de eletrocardiogramas, fluxos sanguíneos e gasosos, tem oferecido informações vitais que auxiliam o médico, quer no tratamento eficaz do paciente, quer no apoio à pesquisa. Na prática a informática ajuda na coleta de registro e na análise de dados, indo a assistência individual à saúde coletiva. (UNIFIESP, 2014).

Com o constante aumento e a facilidade de acesso a aparelhos eletrônicos como celulares, tablets, computadores e smartphones, tornou-se mais fácil o acesso as informações fazendo com que o uso restrito desses equipamentos ao ambiente de trabalho se tornasse comum ao ambiente doméstico. Essa facilidade permitiu a realização de pesquisas relacionadas à integração de dispositivos (fixos e móveis) de uma forma dinâmica, fazendo com que equipamentos eletrônicos que antes eram independentes entre si passassem a se comunicarem recebendo e enviando informações. (PAZ et al., 2012)

Nesse contexto da computação ubíqua onde podemos utilizar recursos da computação móvel independente do local e do momento, Paz et al. (2012) completa que a utilização de novas tecnologias torna o conhecimento na área da saúde mais acessível e facilita o compartilhamento de informações médicas.

Segundo Sabbatini (1998) o desenvolvimento da Informática em Saúde brasileira passou por uma grande transformação a partir de 1983, com a criação de novos grupos especificamente dedicados a esta área de pesquisa e ensino. O primeiro curso de informática voltado para alunos e pós-graduandos de medicina no Rio Grande do Sul, coordenada pela Dra. Mariza Klück Stumpf surgiu em 1982.

Os sistemas especialistas, são programas de computador capazes de analisar dados de uma maneira que, na medicina eles auxiliam na tomada de decisões como por exemplo a análise dos dados hemodinâmicos obtidos por um cateter de Swan-Ganz é muito mais fácil de resolver por um software do que o de determinar a conduta clínica a ser seguida com um paciente com choque cardiogênico. Por isso podemos dizer que os dados necessários para tomar a

decisão podem ser descritos objetivamente com o auxílio dos sistemas especialistas. (WIDMAN, 1998).

Em resumo podemos dizer que a Computação Ubíqua fornece ferramentas inovadoras que podem contribuir com a área da medicina, com por exemplo através da aplicação da computação de forma onipresente (inserida de forma invisível no ambiente) possibilita uma melhora na interação entre os diversos equipamentos médicos, informações em bancos de dados de diversos modelos e os usuários (profissionais da saúde) no ambiente hospitalar. (PAZ et al., 2012). Um sistema especialista voltado para a medicina deve então oferecer tanto serviços básicos, por exemplo banco de dados médicos e tramitação de textos dentre outros, como tecnologias modernas tais como acompanhamento e monitoração à distância com processamento e transmissão de sinais biológicos e imagens médicas. (MEDICINA ASSISTIDA POR COMPUTAÇÃO CIENTÍFICA, 2014).

4 ANÁLISE DE DESENVOLVIMENTO DE SOFTWARE

A análise de desenvolvimento de sistema tem como objetivo encontrar as necessidades do usuário, possibilitando que o programa a ser desenvolvido pelo programador possa atender plenamente as necessidades do mesmo. O desenvolvimento de um software computacional deve possuir embasamento matemático, de lógica de programação e algoritmos. Para facilitar esse processo o programador precisa conhecer modelos e especificações de software a fim de transformá-los em produto de software. (OKUYAMA; MILETTO; NICOLAO,2014).

O desenvolvimento de um software é uma atividade complexa, não existe uma única solução para cada cenário de desenvolvimento. Lidamos o tempo todo com pessoas, o que torna o sucesso do projeto bastante relacionado à competência da equipe e à forma como trabalham, e, para dificultar ainda mais, muitas vezes não fazemos uso de um processo bem definido para apoiar as atividades do projeto. (SPÍNOLA, 2014).

Segundo Hirama (2012) os processos de software são importantes, pois estabelecem para os membros de uma equipe de projeto uma diretriz das etapas a serem seguidas para se alcançar os objetos de um software. O gerenciamento adequado é fundamental para atingir resultados como restrições de custo e prazo, neste sentido o gerenciamento de requisitos trata dos requisitos e das mudanças de requisitos ao longo de todo o ciclo de desenvolvimento.

O custo das atividades de um projeto está diretamente ligado as atividades e a análise dos requisitos, embora o custo da análise de requisito não represente um valor significativo no projeto é nesta etapa em que podem ser encontrada boa parte dos erros de um projeto. A Figura 2 demonstra os custos envolvido conforme a etapa do desenvolvimento do software.

Figura 2 - Cenário atual de desenvolvimento de software.

	% do Custo de Desenvolvimento	% dos erros introduzidos	% dos erros encontrados	Custo relativo da correção
Análise de Requisitos	5	55	18	1
Projeto	25	30	10	1- 1,5
Codificação e teste de unidade	50			
Teste	10	10	50	1- 5
Validação e Documentação	10			
Manutenção		5	22	10 -100

Fonte: Spínola (2014).

Nota: Adaptado pelo autor.

Conforme pode ser observado na Figura 2, a etapa de maior custo durante a etapa de desenvolvimento é a codificação e teste, seguida pela etapa de projeto. Conforme citado por HIRAMA (2012), essas etapas são fundamentais para o desenvolvimento do software, sendo que quanto melhor a qualidade do projeto, menor será o custo das etapas seguintes.

A engenharia de Software envolve processos, métodos, técnicas e ferramentas que auxiliam na produção de softwares cada vez mais complexos e de com maior produtividade tendo um papel fundamental no apoio ao desenvolvimento de software. (HIRAMA, 2012).

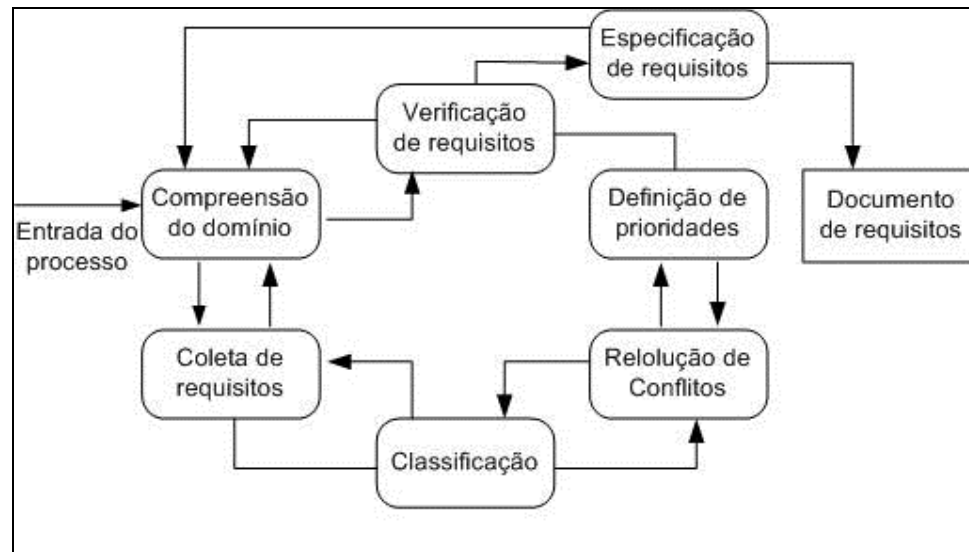
4.1 ANÁLISE DE REQUISITOS

Durante o processo de desenvolvimento de um software, uma das principais chaves para o sucesso é o medida em que ele atende os objetivos e requisitos sob os quais ele foi construído. Uma das principais funções da engenharia de requisito de software é o processo para descobrir e identificar as necessidades e objetivos de um software facilitando o processo de documenta-las, garantido assim que as informações possam ser facilmente interpretadas na etapa de analise. (PRESSMAN, 2011).

Segundo Sommerville (2003) no início de toda a atividade de desenvolvimento de software devemos fazer o levantamento de requisitos, sendo

que o levantamento e análise de requisitos é um processo iterativo, com uma contínua validação de uma atividade para outra. A Figura 3 abaixo exemplifica o processo de levantamento de requisitos:

Figura 3 - Processo de levantamento e análise de requisitos.



Fonte: Sommerville (2003).

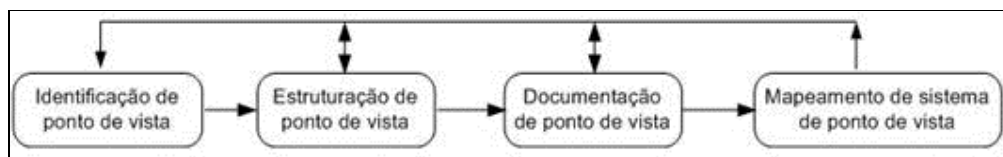
Conforme pode-se observar na Figura 3 a partir da compreensão do domínio inicia-se a coleta de requisitos e verificação dos requisitos, conforme os requisitos vão sendo coletados e verificados eles alternam-se entre a etapa atual e compreensão do domínio, de modo que possam ser classificados sem que haja conflitos e que seja possível ser definido suas prioridades, afim que no final dos projetos seja possível gerar um documento com os requisitos.

Para Pressman (2011) como resultado da análise de requisitos podemos obter as especificações das características operacionais do software, contendo pro exemplo os elementos que iram compor a interface do software, restrições que o software deve possuir e como cada componente do software irá se comportar. Para o autor estudo de como o sistema irá se comportar facilita o seu desenvolvimento, uma vez, que podemos saber previamente como e o que deve ser desenvolvido facilitando assim a comunicação entre o cliente e o usuário, possibilitando assim uma discussão com o usuário sobre modificações e implementações que ser efetuadas durante a etapa de desenvolvimento.

Entre os métodos utilizados no levantamento de requisitos destaca-se o

método VORD (viewpoint-oriented requirements definition - definição de requisitos orientada a ponto de vista), nesse método é feito primeiramente uma análise com o intuito de identificar os serviços e as entidades que interagem no sistema, em seguida os pontos de vistas são agrupados e relacionados, depois é elaborado a documentação do ponto vista e por é feito um mapeamento de sistema, de acordo com contidas nos pontos de vista. (SOMMERVILLE, 2003). A Figura 4 demonstra como as etapas do VORD relacionam-se.

Figura 4 - Método VORD



Fonte: Sommerville (2003).

Conforme a Figura 4, no método VORD deve-se inicialmente identificar o ponto de vista, ou os pontos de vista, a partir desse ponto de vista é possível identificar a estruturação dele, e criar a documentação dele. No final do processo será gerado uma mapeamento das entidades que iram compor os sistema e como eles iram interagir. É importante resaltar que o método VORN é ciclo, sendo que cada etapa interage entre si, fazendo que as etapas possam ser modificadas, conforme haja necessidade ou atualização em uma etapa anterior. (SOMMERVILLE, 2003).

Entre as atividades que compõe a análise de requisitos podemos destacar o levantamento de requisitos, que consiste na tarefa de entrevistas os usuários e clientes afim de obter as informações para o desenvolvimento do sistema. O registro dos requisitos levantados e analisados, que tem como função verificar se os requisitos atendem as necessidades e podem ser documentados, como a elaboração de diagramas de caso de uso, diagramas de classes e etc. E por fim pode-se destacar especificação de requisitos e o modelo de analise que proporcionam a equipe de desenvolvimento do projeto e aos clientes e usuários os meios para que se avaliar a qualidade do software enquanto o mesmo é desenvolvido. (PRESSMAN, 2011).

4.2 DESENVOLVIMENTO DO PROJETO

O projeto de software pode ser descrito como uma representação ou modelo de software, que tem como função indicar como deve ser desenvolvido determinado sistema, fornecendo detalhes sobre a arquitetura de software, estruturas de dados obtidas e interfaces fundamentais que serão implementadas no sistema. Dentro da atividade de projeto software existe um conjunto de princípios, práticas e conceitos que devem ser aplicadas ao desenvolvimento de sistema. (PRESSMAN, 2011).

De acordo com Sommerville (2003), o projeto de software contém os dados que fazem parte do sistema, como a interface, e outros componentes, algumas vezes podendo conter os trechos do código que será utilizado na descrição da estrutura do software que será implementado. O projeto não tem como função detalhar completamente o sistema, mas sim abstrair as informações principais dele, por isso podem ser criados vários modelos com diferentes níveis de abstração diferentes e a cada novo nível criado, geralmente, detecta-se problemas nos m anteriores. Sempre que é criado um novo modelo, ele passa a ser mais detalhado, de forma com que os modelos subsequentes sempre passam a ter um menor número de abstração.

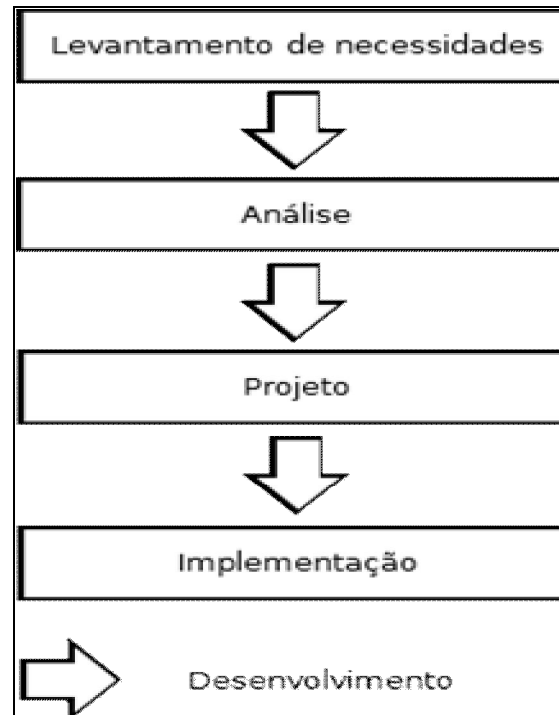
Para Pressman (2011) a representação do um modelo gera solidez e comodidade para o desenvolvimento. Para ele o projeto de software muda continuamente á medida que novos métodos surgem e existe uma melhor análise e entendimento do modelo. A importância do projeto de software pode ser definida como a etapa que fornece representações de software que podem ser avaliadas em termo de qualidade, sendo que o projeto é maneira em que podemos traduzir precisamente os requisitos dos interessados em produto ou software finalizado.

4.3 METODOLOGIA, PROCESSOS e MÉTODOS DE DESENVOLVIMENTO DE SOFTWARE

O ciclo de vida de um software está relacionado a todas as atividades que estão envolvidas e que causaram mudanças no software fazendo com que ele se torne mais complexo e funcional. Um software no geral possui um ciclo de vida e na medida em que as necessidades das pessoas se modificam o software deve se modificar na mesma proporção de maneira que o sistema esteja sempre atualizado. (RODRIGUES, 2008).

Segundo Schach (2007) o software começa a ser desenvolvido a partir das necessidades do cliente, e em seguida é realizada a etapa da análise. Assim que a análise estiver concluída é realizado o projeto e em seguida a implementação do produto de software completo. Na Figura 5 podemos ver as etapas envolvidas no desenvolvimento de software

Figura 5 - Desenvolvimento ideal de software.



Fonte: Schach (2007).

Considerando a Figura 5, onde o modelo apresentando no ciclo de vida ideal de um software, as fases interagem de modo em que se necessário mudanças no software o ciclo passo ser revisto por uma fase anterior. (RODRIGUES, 2008).

Durante a etapa de planejamento definimos as atividades e a para qual membro da equipe será definido uma determinada tarefa, podemos definir isso como processo de software que significa determinar de forma precisa e detalhada quem faz o que, quando e como. As atividades definidas para o processo de software devem possuir prazos e métricas para que se possa avaliar como elas estão sendo realizadas. O processo, ao contrário do método que é algo mais teórico deve determinar ações e práticas a serem realizadas pela equipe para que possa ser corretamente gerenciado durante a sua execução. (LEITE, 2007).

Rodrigues (2008) faz uma ressalva sobre a importância dos processos de software, segundo ele os processos foram muito importante para a evolução da engenharia de software pois permitiu a sistematização do ambiente de desenvolvimento de software permitindo um maior controle nas atividades do projeto. Além disso o processo de software permite que ele seja organizado de maneira lógica as atividades.

Para Leite (2007) um processo é organizado em atividades e essas atividades são de responsabilidade de um membro da equipe (trabalhador). Um processo deve estabelecer uma série de marcos que representa um ponto final de uma atividade de processo. Todas as atividades devem gerar um artefato de saída, que possa ser verificado, e podem requisitar um artefato de entrada. Esse artefato de entrada pode ser definido com um documento, código ou modelo produzido por uma atividade. Uma entrega (liberação) é um artefato entregue ao cliente.

A principal metodologia para a orientação a objeto é o processo unificado que tem como um dos componentes a linguagem de modelagem unificada (UML – *Unified Modeling Language*). O processo unificado é uma metodologia adaptável que pode ser modificada para o produto de software específico a ser desenvolvido, porém alguns de seus recursos não devem ser utilizados em softwares de pequena e média escalas. (SCHACH, 2007).

UML é uma linguagem que permite a criação de vários modelos e tem como principal função permitir a representação dos objetos de software e a comunicação entre eles. Ela nos permite utilizar uma arquitetura que permite a construção mais ágil do software a medida em que o projeto avança. (LOBO, 2009).

De acordo com Rodrigues (2008) os métodos de desenvolvimento de software são regras ou atividades em fases ordenadas, que tem por finalidade definir e organizar o ambiente de desenvolvimento. Esses métodos podem ou não possuir fases e geralmente estão comprometidos com a funcionalidade. O desenvolvimento ágil de software possui métodos que buscam atender mudanças repentinas no projeto de maneira ágil e dinâmica, sempre ligados a um maior controle do ambiente de desenvolvimento com mais rapidez.

4.4 MODELAGEM DE SOFTWARE

Segundo Pressman (2011) uma a atribuição de modelos na engenharia de software ajudam a compressão do que será construído. Na engenharia de software pode-se trabalhar com duas classes de modelos, os modelos de requisitos e os modelos de projeto. Os modelos de requisitos ajudam a descrever o comportamento do software enquanto os modelos de projeto ajuda a representar as características do software que são importantes para que os desenvolvedores vão utilizar para construí-los, como arquitetura, interface, e componentes.

A modelagem de software tem como um se seus objetivos ajudar no mapeamento prévio e na definição do comportamento de determinados componentes do sistema, A modelagem permite um refinamento nos detalhes do sistema, utilizando para isso diversos tipos de diagramas, como diagramas de Classes, diagramas de caso.(SILVA, 2013).

A descrição das interações do sistema é essencial para o entendimento do comportamento de um software. Na modelagem baseada em cenários a criação de cenários na forma de casos ajudam a criar modelos de modo que a análise e o projeto sejam interpretados com maior facilidade. Um diagrama de caso consiste em algumas funções ou atividades que podem ser realizadas por um atos específico. (PRESSMAN, 2011).

4.4.1 Diagrama de Caso de Uso

O diagrama de caso de uso documenta o que o sistema faz do ponto de vista do usuário. Pode-se dizer que ele descreva as principais interações do usuário com o sistema e como as funcionalidades que ele possui se relacionam entre sim. Para a elaboração do diagrama de caso de uso, não nos aprofundamos nos detalhes técnicos do sistema. Este artefato é comumente derivado da especificação de requisitos, que por sua vez não faz parte da UML.(RIBEIRO, 2013).

Ribeiro (2013), lista os quatros componentes básicos do diagrama de caso uso:

- Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.
- Ator: Usuário do sistema, ou melhor, um tipo de usuário.

- Use Case: É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)
- Comunicação: è o que liga um ator com um caso de uso

De acordo com Pressman (2011) toda modelagem possui uma limitação, e o caso de uso não é uma exceção, porém, ele faz uma importante ressalva, a modelagem baseada em cenários é apropriada para grande parte das situações existentes, se desenvolvido corretamente o caso de uso pode gerar grandes benefícios como ferramenta de modelagem. A Figura 6 demonstra algumas definições das figuras do diagrama:

Figura 6 - Elementos do diagrama de caso de uso



Fonte: Ribeiro (2013).

Conforme pode-se ver na Figura 6, todo diagrama de caso de uso deve possuir um ator, que executa uma ação, um caso de uso, que corresponde a uma função determinada que pode ser acessada pelo ator a comunicação, que estabelece o relacionamento que o ator terá com o caso de uso.

4.4.2 Diagrama de Classes

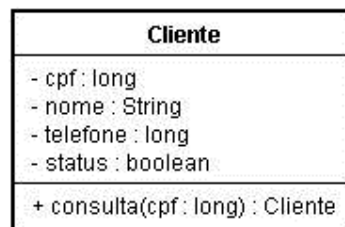
Segundo Silva (2013) o diagrama de classes pode ser considerado como um dos diagramas mais importante e o mais utilizado da UML. Ele tem como principal função permitir a visualização das classes e seus métodos correspondentes, bem como seus atributos e seus relacionamentos. Além disso o diagrama de classes informa como as classes transmitem e dividem informações entre si, apresentando uma visão estática de como as classes estão organizadas, preocupando-se apenas em definir a estrutura lógica das mesmas.

O diagrama de classes baseia-se em como representar os objetos e eles serão manipulados pelo sistema. Os elementos desse modelo baseado em classes

são: classes e objetos, atributos, operações, modelos CRC (classe-responsabilidade-colaborador), diagramas de colaboração e pacotes. (PRESSMAN, 2011).

Para Silva (2013) embora os métodos sejam declarados no diagrama de classes, e esses métodos identificam quais os parâmetros são recebidos por eles e quais parâmetros são devolvidos, um diagrama de classes não deve se preocupar em definir as etapas que estes métodos deverão percorrer quando forem chamados. Ainda de acordo com autor um diagrama de classes deve ser representado por um retângulo com até três divisões, como na Figura 5.

Figura 7 - Exemplo de classe contendo atributos e métodos de um cliente.



Fonte: Silva (2013).

Como pode-se observar na Figura 7, cada classe deve conter na primeira divisão o nome, na segunda divisão os atributos que ela possui e na última parte os métodos, sendo que os métodos devem apresentar apenas os métodos essenciais que a classe deve conter, uma vez que muitos dos métodos serão definidos na etapa de desenvolvimento.

Conforme Pressman (2011), as ferramentas de modelagem de dados facilitam o desenvolvimento do projeto, fornecendo a capacidade de representar objetos de dados, suas características e seus relacionamentos. Essas ferramentas de um modo geral são usadas na elaboração do projeto e ajudam na construção do banco de dados e na criação das classes de objetos.

4.5 DESIGN PATTERNS E O DESENVOLVIMENTO ÁGIL DE SISTEMAS

No desenvolvimento de um sistema, espera-se alguns requisitos que sejam garantidos, como por exemplo: desempenho, robustez, compreensão, facilidade na reutilização, modificação e uso. (PALMEIRA, 2013b).

O *design patterns*, são padrões que tem como objetivo tornar o processo de desenvolvimento mais ágil através de padrões e procedimentos de arquiteturas de projetos. Ele é utilizado em diversas áreas como engenharia e arquitetura.

Segundo Schissato e Pereira (2012) o objetivo dos padrões de projeto, são tornar componentes reutilizáveis que facilitam a padronização, que permita agilidade para as soluções de problemas recorrentes no desenvolvimento do sistema.

Ao longo dos anos os diversos princípios comuns de design se tornaram boas práticas e ajudaram a constituir um design de software de fácil manutenção, entre esses princípios ou padrões de projetos conhecidos na engenharia de software podemos destacar o GoF (*Gang of Four*) e os padrões GRASP (*General Responsibility Assignment Software Patterns*).

Já os padrões GoF tem com objetivo solucionar os problemas comuns que podem vir a surgir durante o desenvolvimento de software orientados a objetos. Esses padrões podem ser distribuídos em três grupos: padrões de criação; padrões estruturais e padrões comportamentais.

Em resumo podemos dizer que com o aumento da complexidade dos sistemas desenvolvidos, a utilização de padrões visando facilitar a programação orientada a objeto tornaram-se de grande relevância para a manipulação dos dados e a apresentação das aplicações.

4.6 FERRAMENTA PARA DESENVOLVIMENTO EM JAVA WEB

Criado em 1991 na Sun Microsystems por James Gosling o java tinha inicialmente a linguagem que se chamava “Oak”, posteriormente em 1995 ela foi renomeada para Java. Inicialmente a Java era uma linguagem de programação voltada para dispositivos portáteis ou para eletroeletrônicos, independente do tipo de processador utilizado.

Uma das principais vantagens é o fato dele ser multiplataforma. Hoje a maioria das linguagens sofrem na transferência de plataforma quando o sistema desenvolvido tem que migrar para outra plataforma, pois quando compilado um programa a ação do compilador é transformar o arquivo-fonte em código de máquina. (PALMEIRA, 2013a).

4.7 PADRÃO DA ARQUITETURA DE SOFTWARE MVC

Segundo Sommerville (2003) o padrão MVC (*Model-View-Controller*) surgiu em 1979 com Trygve Reenskaug no Centro de Pesquisa da Xerox (Xerox PARC) em Palo Alto, Califórnia. Ele surgiu com o intuito criar um modelo computacional que pudesse ser utilizado pelos programadores. Uma aplicação geralmente é formada por um conjunto de códigos que trabalham no processamento de dados, de regras de negócio e de apresentação ou interface. Quando esses códigos estão misturados vários problemas podem surgir. Pensando em uma equipe com vários programas a junção desse código pode fazer com que vários problemas possam vir a surgir entre esses componentes, provocando dificuldade de manutenção, impossibilidade de reúso, repetição de código no caso de inclusão de novas apresentações e mesmo o código de acesso a dados pode sofrer os mesmos problemas.

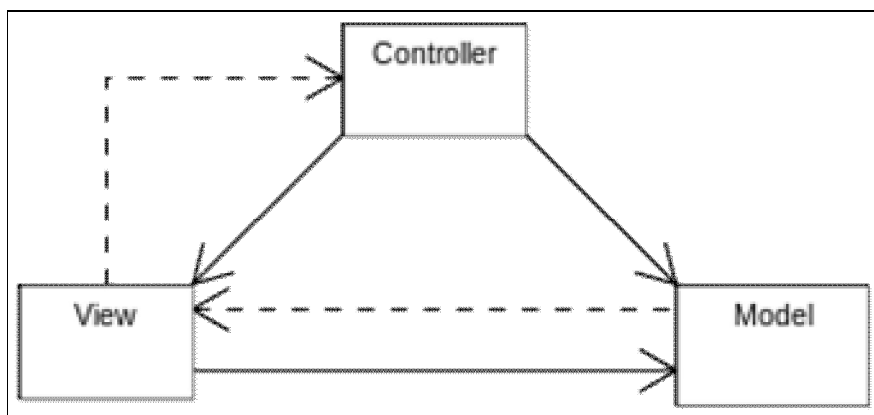
Para Weissmann (2013) A divisão entre o modelo de dados (*Model*) e a visualização (*View*), por exemplo, permite ao desenvolvedor, reutilizar um mesmo objeto de modelo em diversas visualizações diferentes. Desta forma, alterações feitas no layout não afetam a manipulação de dados, assim como qualquer modificação feita em outra camada não irá afetar outras camadas.

A camada de visualização contém a definição da interface com do usuário como por exemplo, campos serão organizados e distribuídos na tela. Pode ser uma página JSP, por exemplo. Esta camada de visualização, estabelece quem é o responsável somente pela visualização, não exercendo nenhum tipo de controle no fluxo de execução da aplicação. (WEISSMANN, 2013).

A camada de controle da aplicação, é onde temos a manipulação dos dados inseridos pelo usuário, sua interpretação e a execução das tarefas correspondentes. A camada de controle é formada por um conjunto de objetos que recebem informações da camada de visualização e enviam as ações responsáveis por atualizar a camada de dados de acordo com a ação do usuário. (WEISSMANN, 2013).

A separação da aplicação nesses três aspectos traz uma série de vantagens ao desenvolvedor, conforme a Figura 8 a seguir.

Figura 8 - Um diagrama simples exemplificando a relação entre Model, View e Controller.



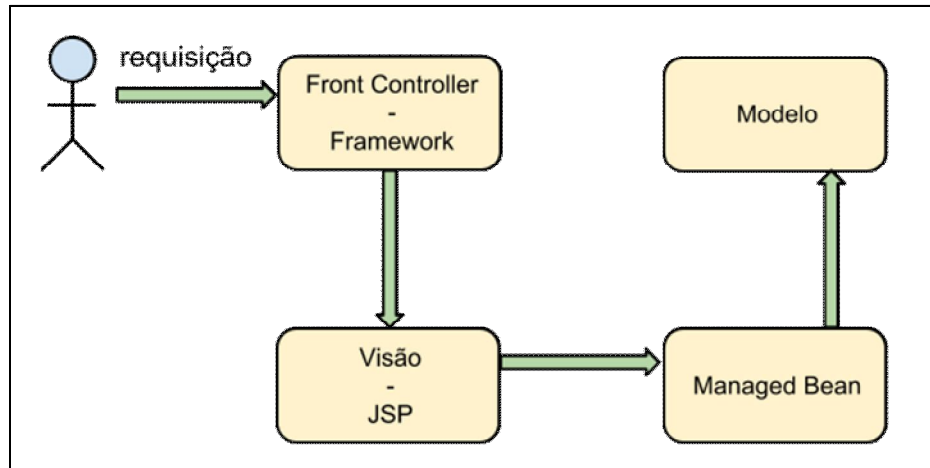
Fonte: Lamin (2009).

A Figura 8 demonstra a relação entre as camadas que representam o modelo MVC. A camada de controle, é responsável por gerenciar o acesso a camada de visualização, e a camada de modelo, responsável pelo controle de transações com o banco de dados.

Conforme Almeida (2012) o Spring MVC Framework é um framework de aplicação de código-fonte aberto popular que tem como objetivo facilitar o desenvolvimento do Java EE. Na essência ele não foi criado para o desenvolvimento web, ele consiste em um contêiner que visa fornecer serviços para uma aplicação, como por exemplo o gerenciamento de objetos, interfaces de usuário, transações e persistência da Web. Uma parte do Spring Framework é o Spring Web MVC, um framework MVC extensível para criação de aplicações web.

O fluxo de processamento, como ilustra a Figura 9, começa com a uma requisição efetuada pelo usuário, conforme a figura 6, a partir desse momento é acionada a camada de controle que tem a função de direcionar o fluxo de acordo com a requisição. A camada de controle aciona a camada de visão que por sua vez pode ou não fazer o uso da camada de Modelo e de todas as regras de negócio. Nesse ponto, temos as regras de visualização implementadas, como por exemplo, exibir as informações em vermelho, dada alguma característica.(ALMEIDA, 2012).

Figura 9 - Processo de uma requisição MVC.



Fonte: Almeida (2012)

A Figura 9 ilustra o processo de uma requisição da seguinte maneira: Um usuário faz uma requisição de uma página, essa requisição vai para a camada de controle, que é responsável por carregar a página que o usuário deseja acessar. Caso essa página que o usuário deseja acessar contenha alguma coleção de dados, acessado as camadas inferiores, como a camada de modelo para carregar as informações nos objetos que serão exibidos na página.

4.8 O QUE É UM WEB SERVICE

Pode-se definir serviços como partes independentes de uma aplicação que tem como uma de suas características principal a autonomia em relação a outros serviços. Cada serviço é responsável por seu próprio domínio, o que tipicamente significa limitar seu alcance para uma função de negócio específico. Um Web Service é alguma lógica de negócio acessível usando padrões da Internet.(SAUDATE, 2014).

Devido à independência que esses serviços o *web service* possui funcionalidades ligadas fracamente entre si, essa característica permite que a lógica de programação possa ser diferente ou que não obedeça uma determinada plataforma ou tecnologia. (ERL, 2014).

XML Web service é um dos tipos de serviço mais utilizado e aceito atualmente. O XML Web service possui dois requisitos fundamentais:

- comunica-se via protocolos internet (normalmente HTTP);
- Envia e recebe dados formatados como documentos XML.

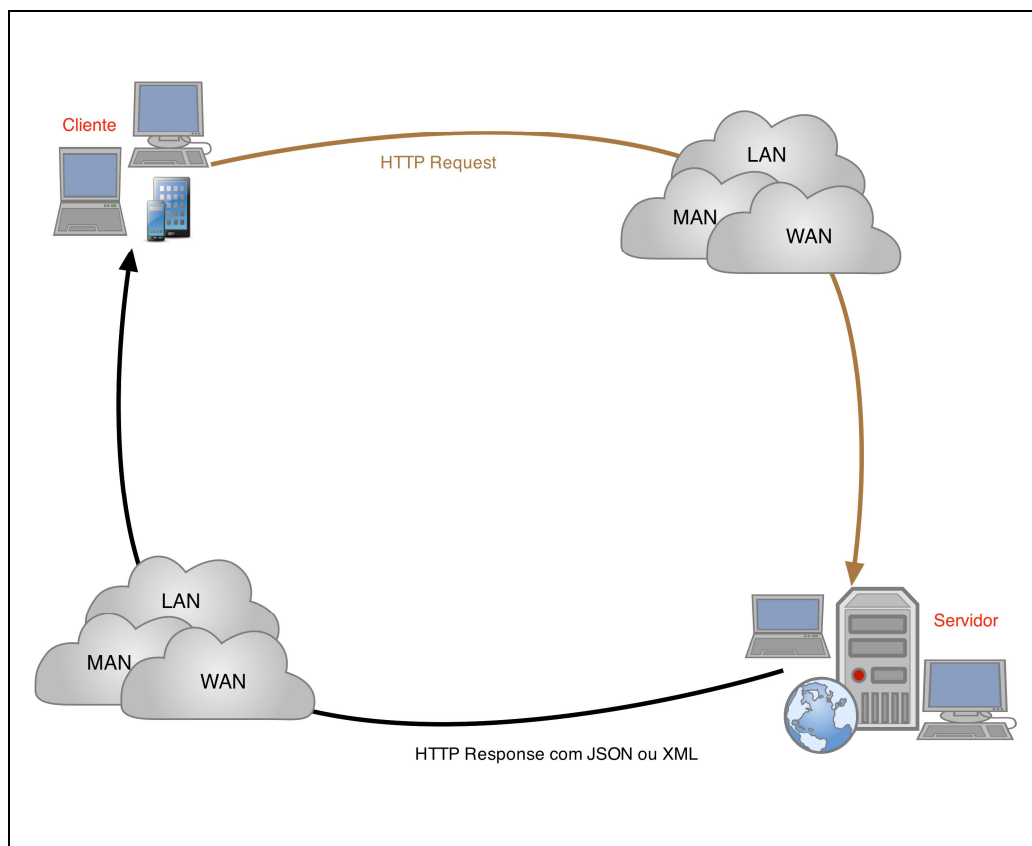
Segundo Erl (2014) a ampla aceitação do *web service* resultou no surgimento de um conjunto de tecnologias suplementares que se tornaram um padrão de fato. Um *Web Service* deve ser capaz de agir como o solicitante e o provedor de um serviço, isto quer dizer, o cliente iniciador do pedido pode ser um *web service* também, onde ele faz a requisição de uma informação e pode receber também um retorno.

Web Services tem como principal atribuição a integração de serviços, permitindo que ferramentas distintas, desenvolvidas em diferentes linguagem conversem entre si, tornando assim, possível o desenvolvimento de soluções desacopladas que facilitam sua integração com sistemas externos. (SANTOS,2013)

Santos (2013) lembra que o conceito de serviços em uma aplicação existe faz algum tempo. Serviços, assim como componentes, são considerados blocos de construção independentes, os quais coletivamente representam um ambiente de aplicação. Cada serviço possui suas características únicas que lhes permitem participar como parte de uma arquitetura orientada a serviços, por exemplo diferentes sistemas desenvolvidos em diferentes poderão ter diferentes serviços cada um com sua particularidade, mas que conversão entre si através de um padrão para troca de informações.

Para Saudete (2013) o estudos a respeito dos *web services* está relacionado ao estudo sobre os serviços REST (*Representational State Transfer*). O REST, por exemplo, pode ser utilizado para realizar a listagem de pacientes de um sistema, através de um navegador ou de algum dispositivo conectado a internet que seja capaz de requisitar e receber uma lista de dados. A Figura 10 demonstra o funcionamento de um *Web Service* utilizando REST, onde pode-se notar que sempre que é feita uma requisição e essa requisição retorna uma quantidade de dados.

Figura 10 - Funcionamento de Web Service com REST.



Fonte: Santos (2013)

Conforme demonstrado na Figura 10, um cliente faz uma requisição para um servidor através da internet, através dos dados enviados na requisição o servidor processa uma resposta e retorna uma coleção de dados para cliente que fez a requisição. Tanto os dados da requisição quanto os dados da resposta, podem ser enviados no formato XML ou JSON (JavaScript Object Notation).

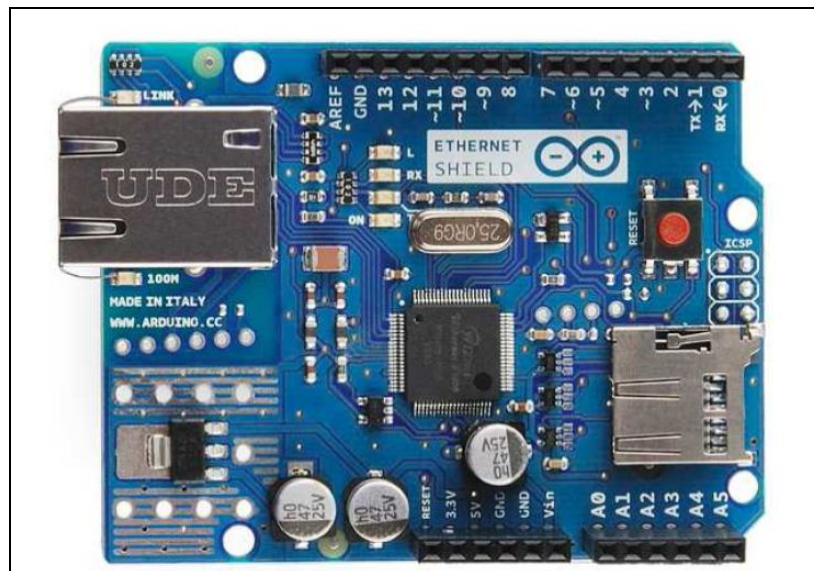
O REST consiste em modelo para o desenvolvimento de web services que foi proposto na tese de doutorado de Roy Fielding. Este, por sua vez, é coautor de um dos protocolos mais utilizados no mundo, o HTTP (*HyperText Transfer Protocol*). Devido a isso o protocolo REST segue as boas práticas para o uso do protocolo HTTP fazendo com ele possua chamadas de recursos de um sistema bem definidas. SAUDATE (2013).

4.9 PLATAFORMA ARDUÍNO

O Arduino consiste, por definição, em um microcontrolador de placa única e um conjunto de software para programá-lo. O hardware é composto de um processador Atmel AVR, um cristal oscilador e um regulador linear de 5 volts. A placa do arduino contem pinos de entrada e saída de dados de maneira com que esses pinos possam se conectar com outras placas e circuitos a fim de adicionar novas funcionalidas a ele. (GOMES; TAVARES, 2013)

O *Arduíno* é uma das plataformas para protótipos eletrônicos que possui um hardware e software flexíveis e fácil de utilização. Ele é composto por um micro controlador de placa única que pode ser programado através de um conjunto de software. (SOARES, 2013). a figura 11 demonstra a placa básica do Aduino.

Figura 11 - Arduino Uno.



Fonte: Arduino (2011).

O ambiente Arduíno foi desenvolvido para ser fácil para iniciantes que não possuem experiência com desenvolvimento de software ou eletrônica, além a placa microcontroladora, ele possui um software que permite que seja programadas funções, utilizando a linguagem C++.(GOMES; TAVARES, 2013).

O Arduíno IDE (*Integrated Development Environment*) consiste em uma aplicação desenvolvida na linguagem JAVA que pode ser utilizada em diversas plataformas. Nessa IDE aplicação multiplataforma escrita, inclui um editor de código

com recursos de realce de sintaxe, parênteses correspondentes e indentação automática, sendo que ele é capaz de compilar e carregar programas no placa do microcontroladora com muita facilidade. (KASUO et al., 2014).

Segundo Gomes e Tavares (2013), atualmente a plataforma Arduino esta difundida por todo o mundo, sendo que esta é utilizada por vários tipos de pessoas. que buscam utilizar uma plataforma de protótipos eletrônicos de código aberto (open source) baseada em hardware a software flexíveis e fácil de usar. É destinado a artistas, designers, hobbyists, e qualquer pessoa interessada em criar objetos ou ambientes interativos.

Conforme Soares (2013) o Arduino pode ser utilizado para desenvolver objetos interativos, tendo entradas para uma uma grande variedade de sensores variedade ou interruptores. Além disso ele pode ser acoplado a placas adicionais, que podem ser montadas à mão ou compradas pré-montadas permitindo controle e envio de informações via web de diversos tipos de dispositivos eletrônicos como motores e outras saídas físicas. Os projetos em Arduino podem se comunicar com um software rodando em um computador e a sua IDE é de código aberto e pode ser baixado gratuitamente.

5 METODOLOGIA

O projeto consistiu no desenvolvimento de um sistema web, capaz de receber os dados de um dispositivo eletrônico, a fim de organizar essas informações de modo a facilitar o monitoramento da pressão arterial de pacientes que se encontram em regiões afastadas dos grandes centros e de difícil acesso ao acompanhamento médico. A realização desse trabalho se baseou como pressuposto inicial a fundamentação teórica obtida através da revisão da literatura.

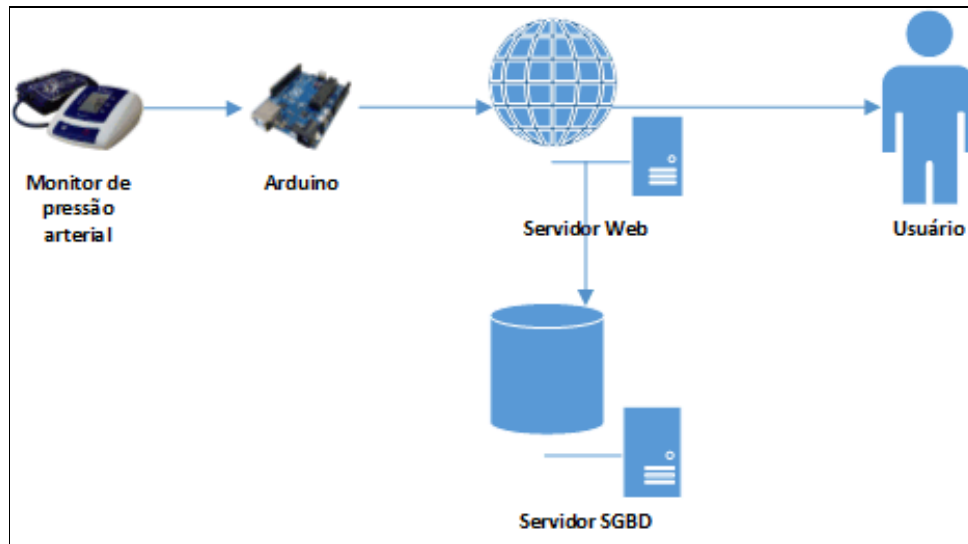
A elaboração desse projeto pode ser definido por três partes: envio dos dados referentes a medição da pressão arterial do paciente através de uma placa microcontroladora, desenvolvimento de um *Web Service* para receber as informações enviadas pelo dispositivo eletrônico e o desenvolvimento de um web site para o acompanhamento dos dados referentes as medições dos pacientes.

Para a obtenção dos dados referentes a pressão arterial do paciente, foi utilizado um monitor de pressão arterial de braço para a leitura da pressão arterial sistólica, diastólica e de pulso acoplado a um Arduíno UNO com uma placa GSM SIM900 para enviar os dados através de uma de dados 3G ou 2G.

Utilizando a linguagem Java para desenvolvimento de aplicações baseadas nessa linguagem, foi desenvolvido o *Web Service* e o site para a consulta de informações dos pacientes. Além disso foi utilizado o tomcat como servidor de aplicação web e o postgresSQL como banco de dados. Visando facilitar o desenvolvimento da aplicação web utilizaremos o modelo MVC (*Model-view-controller*), através do framework Spring, que consiste em um *framework* para o desenvolvimento de aplicações para Java.

As informações referentes aos pacientes foram enviadas para o Arduíno, que estava ligado ao medidor de pressão, o Arduíno envia os dados para um *Web Service* utilizando a tecnologia 3G. Após essas receber as informações o *Web Service* faz os tratamentos necessários e grava elas em um banco de dados para que elas sejam acessadas posteriormente, através de um sistema web, como podemos ver na Figura 12.

Figura 12 - Funcionamento do protótipo



Fonte: Elaborada pelo autor.

Conforme pode ser observado na Figura 12, é realizada a medição da pressão arterial, os dados da medição são enviados para o Arduino, que envia através da tecnologia 3G os dados para o *Web Service*, que após receber os dados grava no banco de dados. Após isso os dados ficam disponíveis para o usuário acessá-los através do sistema.

Nos tópicos subsequentes desse capítulo será abordado mais detalhadamente as tecnologias utilizadas para a implementação do projeto.

5.1 LEVANTAMENTO DE DADOS DO PROJETO

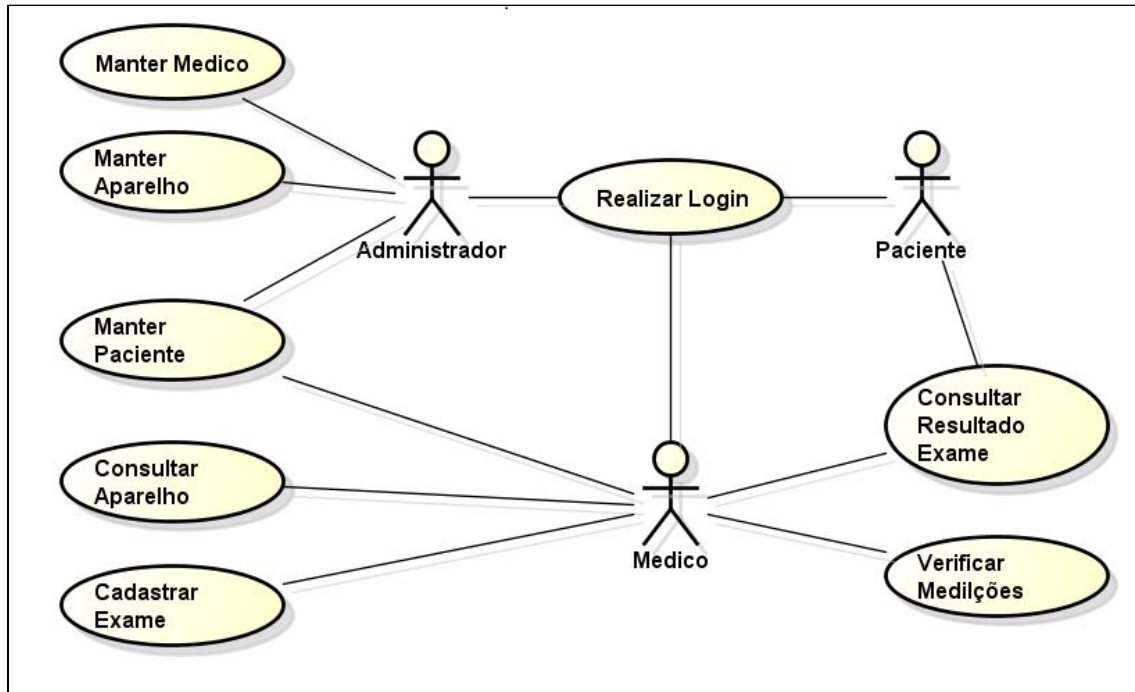
De uma maneira geral o desenvolvimento de software é uma tarefa muito complicada e complexa, nesse contexto a engenharia de software tem como função estabelecer um modelo para facilitar a construção do sistema. O sistema tem como um dos seus objetivos facilitar o acesso das informações organizando-as em gráficos e tabelas, de modo que facilite a leitura e interpretação dos dados.

Assim, como citado no tópico 4.4 sobre a modelagem do software, o mapeamento prévio das interações e a definição do comportamento de determinados componentes do sistema, é fundamental para o entendimento do software como um todo, permitindo o refinamento dos detalhes sistema.

A Figura 13 demonstra como ficou mapeado os componentes do sistema, onde pode se ver como o software interage com três atores, sendo que cada um

pode executar algumas ações em comuns e algumas ações exclusivas.

Figura 13 - Diagrama de caso de uso.



Fonte: Elaborada pelo autor.

Conforme pode ser observado na Figura 13, o sistema possui três atores diferentes, sendo eles, administrador, médico, paciente. O administrador é responsável por realizar os cadastros no sistema, já o médico pode cadastrar um exame, ou consultar os exames dos pacientes relacionados a ele, enquanto o paciente pode consultar os exames que ele realizou.

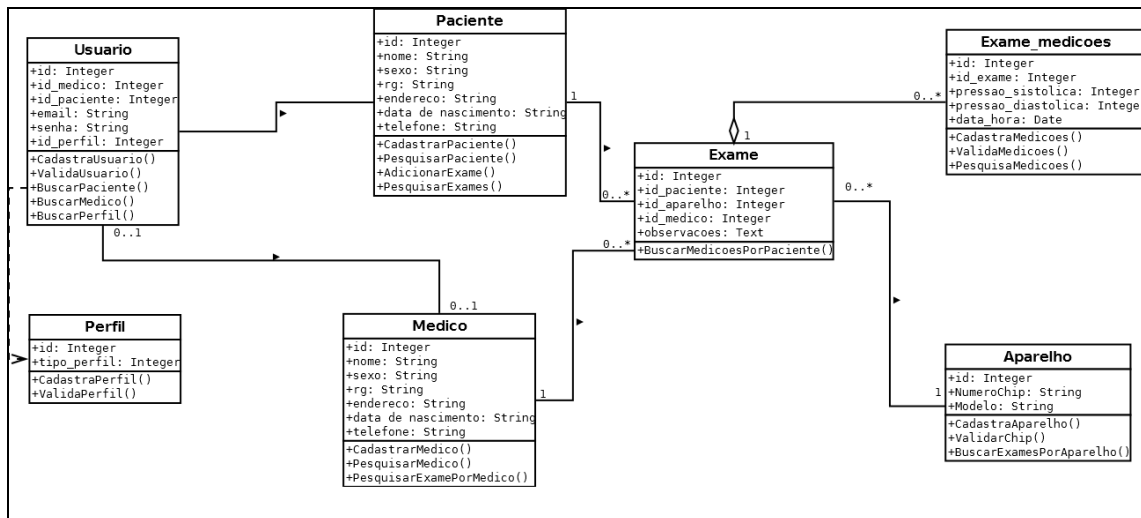
Após a elaboração do diagrama de caso, foi elaborado o diagrama de classes, como mencionado anteriormente no tópico 4.4, que se refere a modelagem de software, toda modelagem possui uma limitação, e o caso de uso não é uma exceção. Procurando aprimorar o desenvolvimento do software foi elaborado também o diagrama de classes. Como citado anteriormente, o diagrama procurar demonstrar os objetos, métodos, atributos e operações que irão compor o sistema.

A criação da diagrama de classes, se baseou nas aplicações que teriam a acesso a manipulação dos dados e nos diferentes usuários que utilizaram o sistema, como, médicos, pacientes, secretarias e etc.

O sistema busca armazenar as informações obtidas das medições de pressão arterial, afim de gerar os dados necessários para a realização do MAPA. Foi

elaborado o diagrama de classes, contendo sete classes, sendo elas: usuário, paciente, perfil, médico, exame, exame_medicoes e aparelho. A Figura 14 demonstra as classes, os atributos e os métodos do sistema.

Figura 14 - Diagrama de classes.



Fonte: Elaborada pelo autor.

Com base no diagrama de classes, demonstrado na Figura 14 pode-se concluir que cada usuário deverá ser associado a um determinado perfil. Os perfis de acesso definidos de acordo com a análise do diagrama de classe e do diagrama de caso de uso foram: Médico, Paciente, Administrador. Cada perfil possui acessos a determinados módulos dentro do sistema, por exemplo, para o perfil do médico ele pode cadastrar exames e consultar exames e pacientes, o perfil do paciente poderá apenas consultar os exames dele. Conforme demonstrado na Figura 15.

Figura 15 - Acesso aos módulos do sistema conforme perfil do usuário.

	Modulo Paciente	Módulo Médico	Modulo Exames
Médico	Consulta	-	Cadastro e Consulta
Paciente	-	-	Consulta
Administrador	Cadastro e Consulta	Cadastro e Consulta	Cadastro e Consulta

Fonte: Elaborada pelo autor.

Outra classe indispensável para o sistema é a exame, ela é responsável por definir como o software deve relacionar os exames cadastrados. Cada exame deve

possuir um paciente, um médico e um aparelho relacionado, sendo que para cada exame deve existir uma ou mais medições. Como citado no tópico 2.4 sobre sintomas e diagnóstico da hipertensão arterial, o exame de MAPA consiste em observar as medições da pressão arterial que são feitas durante um intervalo de 24 horas. Para isso o *Web Service* recebe os dados do Arduino e acessa os métodos da classe *exame_medicoes*, para efetuar as gravações.

5.2 CONECTANDO O ARDUÍNO AO WEB SERVICE

Como mencionado no capítulo 4.9 sobre a plataforma Arduino, ele possibilita a integração de diferentes tipos de plataforma, de maneira simples e interativa, sem a necessidade de que o usuário tenha profundo conhecimento em eletrônica ou programação.

Para a realização do projeto, foi utilizado o Arduino Uno que possui uma memória de 32 Kb para a gravação de programas desenvolvidos utilizando a linguagem C++, com voltagem operacional de 5V e 16 MHz de *clock*, com o auxílio de um *GSM Shield*, módulo para conexão a internet que pode ser adicionado ao Arduino. Ele foi acoplado a um aparelho de medição de pressão arterial, da marca G-Tech, modelo LA250.

A IDE do Arduino possui diversas bibliotecas prontas que facilitam o desenvolvimento e utilização dos recursos do microcontrolador. Durante o projeto foram utilizadas as bibliotecas *ethernet*, para comunicação com a internet, a biblioteca *GSM* para permitir a conexão com a rede 3G e a biblioteca *CapacitiveSensor*, utilizado para ler os valores do sensor do aparelho de pressão.

Foi elaborado um programa utilizando a linguagem C++ que a cada intervalo de 30 minutos acionava o aparelho de pressão para que ele fizesse a medição, após a medição terminar os dados são enviados para o *Web Service*. A Figura 16 demonstra um trecho do código utilizado para ler os valores no aparelho de medição de pressão arterial

Figura 16 - Trecho do código utilizado para ler os valores medido no aparelho de medição da PA.

```

sketch_nov23a §
goto nextBit;
}
//*****
void writeData () {
  Serial.println("Convertendo.....");
  //we only need the data that is "written" , in particular the h
  if(datalog[0]=='W'){
    //sometimes two lines are extracted at once, here the data e
    if(logptr > 10){
      calculateValues(datalog[6], datalog[7]);
      bytesOfData++;
      calculateValues(datalog[15], datalog[16]);
      bytesOfData++;
    } else {
      //when only one line (= "important data byte") is extract
      calculateValues(datalog[6], datalog[7]);
      bytesOfData++;
    }
  }
  measuring_failed = false; //if that comes after
  //device reads from eeprom when you start measuring, so reset
  if(datalog[0]=='R'){
    bytesOfData = 0;
  }else if(bytesOfData > 9){
    bytesOfData = 0;
    writeToSerial();
  }/*else if(bytesOfData >= 8){ //this means that a measurement
    wdt_enable(WDTO_4S); //WATCHDOG!! if no data i
    measuring_failed = true;
  }*/
  //clear datalog[] and logptr

```

Fonte: Elaborada pelo autor.

Na Figura 17, pode-se ver o através do console da IDE do Arduíno a leitura dos dados no momento em que o ele recebe os dados. Sempre que ocorre um erro na medição da pressão arterial é apresentado uma mensagem de erro no console da IDE do Arduíno e a medição é reiniciada. Para o envio das informações através do 3G ocorre o mesmo processo, é feito uma conexão com a Internet, caso ocorra algum erro na conexão é efetuado novamente uma tentativa de envio dos dados.

Figura 17 - Resultado das medições no console da IDE do Arduíno.

```

COM1
Medindo
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
heart rate: 100
pressure: 103/77
----- CONCLUÍDO COM SUCESSO ----- Iniciando
Medindo
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
Convertendo.....
heart rate: 107
pressure: 91/72
----- CONCLUÍDO COM SUCESSO -----
 Autoscroll

```

Fonte: Elaborada pelo autor.

Conforme pode-se observar na Figura 17, os dados dos sensores do aparelho de medição de pressão são lidos pelo Arduíno que fazem a conversão dos mesmos. Após a conversão os dados, como a pressão sistólica e a pressão diastólica, são exibidos no console do Arduíno e enviados para o servidor do sistema web através da internet.

5.3 DESENVOLVIMENTO DO SISTEMA

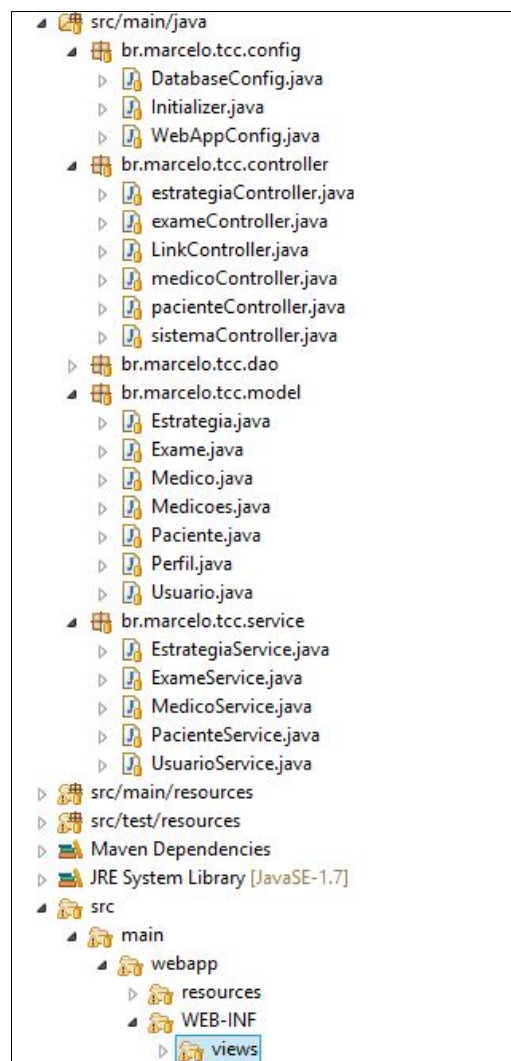
O sistema foi desenvolvido com base no padrão MVC, conforme citado no tópico 4.7 sobre a arquitetura de software, esse padrão visa dividir o projeto em três camadas, sendo elas modelo, visualização e controle. A adoção desse modelo visou facilitar e tornar mais dinâmico esse processo, uma vez que o padrão MVC permite a reutilização dos objetos em diversas visualizações diferentes.

Para a criação do banco de dados foi utilizado o PGADMIN, através de sua interface intuitiva foi possível criar os esquemas, tabelas e campos que foram

utilizados no sistema.

A aplicação web possui uma camada de controle é a camada que redireciona o fluxo de dados do site, todas as requisições de acesso feitas pelo usuário passam primeiramente por essa camada, conforme a requisição ele aciona as camadas mais inferiores como a camada de visualização ou a camada de modelo. A Figura 19 demonstra a estrutura das camadas do sistema.

Figura 18 - Classes do sistema utilizando o padrão MVC.



Fonte: Elaborada pelo autor.

Conforme podemos observar na Figura 19, a estrutura da aplicação, que foi criada utilizando o framework Spring, pode ser representado por cinco pacotes: br.marcelo.tcc.config, br.marcelo.tcc.controller, br.marcelo.tcc.dao, br.marcelo.tcc.model e br.marcelo.tcc.service.

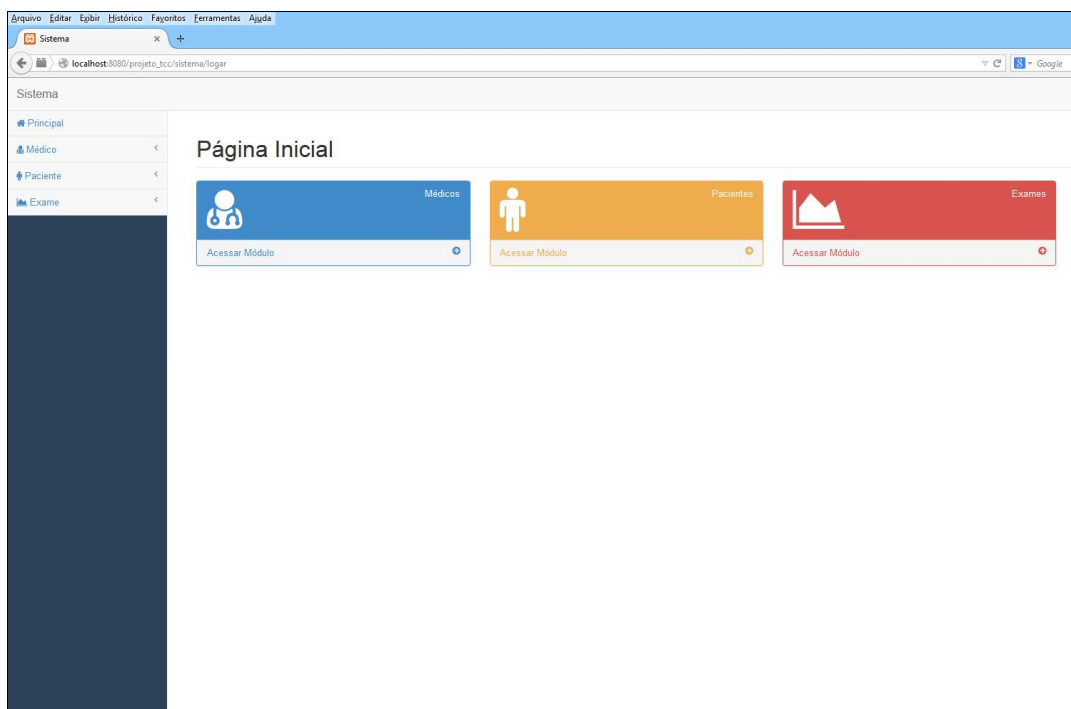
Os pacotes `br.marcelo.tcc.controller` pertencem a camada de controle sendo responsável pelas requisições do sistema. O pacote `br.marcelo.tcc.model` pertence a camada de modelo, sendo responsável por carregar os objetos que será exibido nas páginas.

6 RESULTADOS E DISCUSSÃO

6.1 DESENVOLVIMENTO DO SISTEMA

Os níveis de acessos foram divididos em módulos, conforme mencionado no capítulo 5.1 sobre levantamento de dados do usuário. Os módulos divididos em médico, paciente e exames podem ser acessados conforme os privilégios disponíveis para o perfil do usuário. A Figura 20 demonstra a página inicial do usuário administrador, onde ele possui acesso irrestrito para todos os módulos.

Figura 19 - Tela inicial do sistema



Fonte: Elaborada pelo autor.

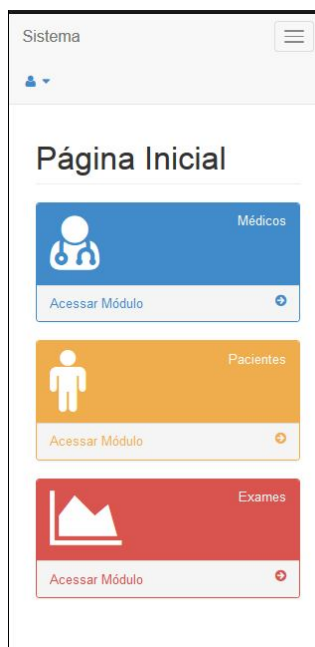
Conforme pode-se observar na Figura 20, ela corresponde á tela de acesso inicial pelo perfil médico. O acesso aos módulos do sistema podem ser feitos pelo menu auxiliar mo lado esquerdo da tela, ou pelos ícones do módulo correspondente.

No módulo Médico o usuário terá acesso ao cadastro, pesquisa e edição de dados dos médicos e dos aparelhos de medição. O módulo paciente dá acesso ao cadastro e pesquisa de pacientes. O módulo exames dá acesso ao cadastro e pesquisa de exames, sendo que os exames podem ser pesquisados, por pacientes

ou médicos.

A elaboração do sistema também levou em consideração o acesso a partir de plataformas móveis, como celulares e tablets, para isso foi feito o uso da biblioteca *Bootstrap*. O *Bootstrap* é uma coleção de elementos e funções em HTML, CSS e *Javascript*, ao desenvolver o site, pode-se escolher os elementos que são disponibilizados pela biblioteca. Figura 20 demonstra como a biblioteca adapta a página inicial do quando o acesso é feito através de um dispositivo móvel.

Figura 20 - Tela inicial do sistema em um dispositivo móvel



Fonte: Elaborada pelo autor.

Conforme podemos observar na Figura 20, onde mostra a tela inicial do sistema quando acessado de um dispositivo móvel com configuração de tela de 360x640 pixels, o sistema se adapta de modo que o menu lateral esquerdo, se torna um menu *dropdown* que pode ser acessado na parte superior da tela.

6.2 TESTES COM OS USUÁRIOS

Os testes foram realizados com pessoas que com idade, sexo e perfil diferentes, afim de obter a maior variação de dados possíveis. Os testes foram realizados em ambiente doméstico, durante um intervalo de três horas. Após criar o

programa para acionar, receber e enviar os dados para o sistema o código foi compilado e gravado no Arduino, sua configuração para a realização dos testes foram para realizar as medições a cada trinta minutos, caso houve algum erro na medição era efetuada uma nova mediação após dois minutos da ultima medição. Em caso de erros no envio dos dados era feito uma nova tentativa de conexão com a internet após dois minutos.

A Figura 21 demonstra a característica dos pacientes que participaram dos testes.

Figura 21 - Características dos pacientes participantes do teste.

	Idade	Sexo	Possui Hipertensão Arterial
Paciente A	68	Masculino	Sim
Paciente B	60	Feminino	Sim
Paciente C	26	Masculino	Não

Fonte: Elaborada pelo autor.

Para cada paciente que participou do teste foram efetuadas 6 medições, sendo que todas as medições obtiveram êxito ao enviar os dados para o sistema. Entre os dados que foram obtidos estão a pressão arterial sistólica que como mencionado no capítulo 2.2, corresponde a pressão arterial verificada quando ocorre a contração cardíaca para a impulsão sanguínea às partes corporais extracardíacas, a pressão arterial diastólica que é verificada quando o coração está em repouso, além dos batimentos cardíacos. A unidade de medição da pressão arterial é a milímetros por coluna de mercúrio (mmHg), enquanto os batimentos cardíacos a sua unidade de medição é batimentos por minuto.

Conforme demonstrado na Tabela 22, pode-se observar os resultados obtidos na medição da pressão dos três pacientes.

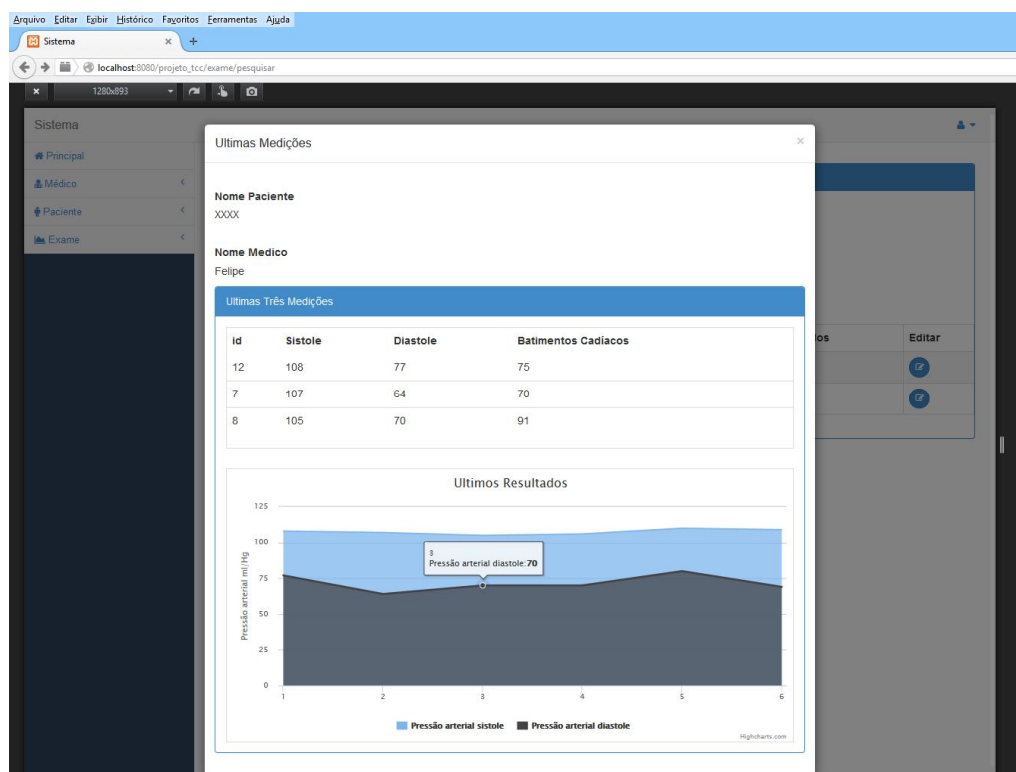
Figura 22 - Resultado das medições dos três pacientes (pressão arterial - batimentos cardíacos).

	1ª Medição	2ª Medição	3ª Medição	4ª Medição	5ª Medição	6ª Medição
Paciente A	102/73 - 72	110/81 - 90	107/76 - 77	108/80 - 69	121/78 - 95	131/80 - 77
Paciente B	107/64 - 70	105/70 - 91	106/70 - 98	110/80 - 68	109/69 - 71	108/77 - 75
Paciente C	131/81 - 74	135/80 - 81	129/77 - 68	125/75 - 77	130/80 - 101	121/69 - 90

Fonte: Elaborada pelo autor.

A fim de facilitar a visualização dos dados antes da finalização do exame, o sistema permite que os últimos três resultados sejam visualizados sem a necessidade de gerar um relatório contendo todas as informações do usuário. Na tela de visualização prévia dos resultados o usuário encontra gráficos e tabelas dos dados do paciente selecionado, essa funcionalidade está disponível para o perfil de médicos e administradores. A Figura 16 demonstra a tela de visualização prévia dos exames.

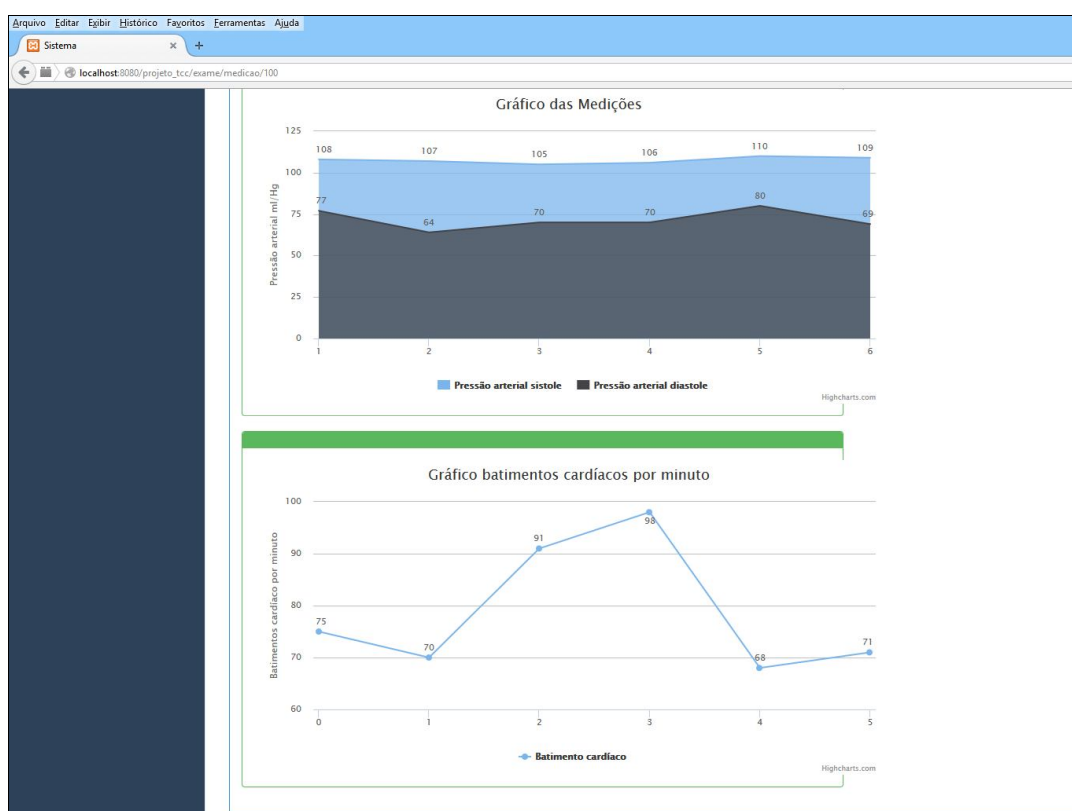
Figura 23 - Tela de visualização prévia dos resultados do exame



Fonte: Elaborada pelo autor.

Após a finalização do exame todos os usuários que possuem acesso ao sistema, podem acessar e visualizar as tabelas e gráficos gerados no decorrer dos exames. Para os usuários que possuem o perfil de médico é possível pesquisar todos os exames dos seus pacientes, e para os usuários que possuem perfil de paciente é possível visualizar todo o histórico de exames. Na Figura 17, pode-se observar todos os dados obtidos no relatório final do exame, como dados da ficha do paciente e os gráficos de batimentos cardíacos e pressão arterial.

Figura 24 - Figura 25 - Gráficos disponíveis no relatório final do exame.



Fonte: Elaborada pelo autor.

Os dados obtidos nas medições, mais alguns exames complementares como citado no tópico 2.4, onde é descrito os passos para o diagnóstico da hipertensão arterial, como por exemplo, a medição da PA durante o exame físico e a análise da ficha do paciente, posteriormente irão auxiliar o médico a chegar ao diagnóstico adequado.

7 CONSIDERAÇÕES FINAIS

Tendo em vista o contexto apresentado, esse projeto se baseou no métodos utilizados atualmente para a realização de exames e a dificuldade em obter os dados, levando em consideração os erros que podem vir ocorrer durante a realização do exame de MAPA, visando criar uma ferramenta para obtenção e visualização dos dados obtidos, permitindo que eles sejam visualizados em qualquer lugar e em qualquer momento, mesmo que o exame ainda esteja em andamento.

O projeto foi desenvolvido com intuito de receber as informações enviadas, permitindo que as mesmas fossem acessadas em tempo real, além de contar com uma interface amigável e que fosse possível se acessada por dispositivos móveis, como celulares e tablets, dando mais flexibilidade para a visualização das informações e evitando a impressão de várias páginas com conteúdo os resultados dos exames obtidos. As

Durante o seu desenvolvimento, algumas dificuldades foram observadas, como por exemplo, a obtenção dos dados referentes as medições da pressão arterial. Inicialmente o modelo utilizado do aparelho, não fornecia o valor da medição formatado, sendo necessário ser feito o tratamento dos dados antes de gravar os mesmo no banco de dados. Esse problema foi resolvido posteriormente com a utilização de outra marca de aparelho para medir a PA.

Outro problema observado durante o seu desenvolvimento se deu no recebimento dos dados, por serem enviados através da internet, utilizando a conexão 3G, foi necessário a utilização de um servidor com IP fixo para que os dados fossem enviados, uma vez em que o projeto foi desenvolvido em ambiente local. Para realização dos testes foi utilizado um servidor empresarial cedido durante alguns dias.

Através do desenvolvimento desse projeto podemos observar a importância da engenharia de software e quanto ela contribuí para o desenvolvimento eficaz de um sistema, seja através de processos, como análise de requisitos ou através dos métodos apresentados para modelagem de software. Também podemos destacar a utilização de ferramentas como o framework Spring que possui uma grande quantidade de métodos prontos que tornaram o desenvolvimento mais rápido e eficaz. Outro ponto que pode ser observado é a integração entre diferentes dispositivos, através da plataforma Arduino, permitindo assim o desenvolvimento de

um projeto interativo e facilitando o envio e leitura dos dados que compõe o exame monitorização ambulatorial da pressão arterial.

8 TRABALHOS FUTUROS

Como sugestões de trabalhos futuros pode-se utilizar uma placa de memória em conjunto com o Arduíno para a gravação dos dados referentes as medições dos pacientes, assim quando houver problemas com a conexão de internet, será possível armazenar os dados para que sejam descarregados posteriormente.

Para o aprimoramento do software pode-se citar a elaboração de relatórios que permitam a exportação em formatos distintos, como por exemplo em pdf. A elaboração de outras funcionalidades, como o envio de e-mails quando algum exame estiver concluído ou quando os valores derem alterados, além de gráficos para realizar comparações com exames anteriores acrescentariam uma melhoria ao software.

O desenvolvimento de um aplicativo próprio para plataforma móveis também pode ser levado em consideração, embora o sistema seja desenvolvido para plataformas móveis também, um aplicativo especialmente projetado para essas plataformas podem a vir facilitar o acesso as informações. E por fim, a utilização de um case onde possa ser inserido o equipamento, uma vez em que a utilização do equipamento necessário para a medição mostrou-se desconfortável para os usuários.

REFERÊNCIAS

ALVES, D. Evolução da Informática na Medicina. **Informática na Medicina**, [2014?]. Disponível em: <<https://sites.google.com/site/informaticanamedicina/home>>. Acesso em: 25 maio 2014.

ALMEIDA, A. Entenda os MVCs e os frameworks Action e Component Based. . **Caelum**, 2012. Disponível em: <<http://blog.caelum.com.br/entenda-os-mvcs-e-os-frameworks-action-e-component-based/>>. Acesso em: 10 maio 2014.

BOLFER, L. **O coração como uma bomba**. Disponível em: <<http://www.geocities.ws/equipecv/fisiologia/corbomba.html>>. Acesso em: 06 jun. 2014.

BRASIL. Ministério da Saúde. Mais médicos. **Portal da Saúde**, 2014. Disponível em: <<http://portalsaude.saude.gov.br/index.php/cidadao/acoes-e-programas/mais-medicos>>. Acesso em: 05 jun. 2014.

BRASILEIROS apontam saúde como principal problema do país. **G1**, 2014. Disponível em: <<http://g1.globo.com/economia/noticia/2014/02/saude-e-o-principal-problema-do-brasil-diz-pesquisa.html>>. Acesso em: 12 abr. 2014.

COSTA, C. J. **Desenvolvimento para Web**. Lisboa: Lusocredito, 2007.

COSTA, D. G. **JAVA em rede**: programação distribuída na internet. Rio de Janeiro: Brasport, 2008.

DOMENICIS JUNIOR, O. **Informática na medicina**. 2014. Disponível em: <<http://www.cura.com.br/artigosmedicos-informatica.html>>. Acesso em: 05 maio 2014

ENTENDA o que é pressão arterial, os fatores de risco e as recomendações. Bem estar, 2011. Disponível em: <<http://g1.globo.com/bemestar/noticia/2011/03/entenda-o-que-e-pressao-arterial-os-fatores-de-risco-e-recomendacoes.html>>. Acesso em: 03 set. 2011.

ERL, T. Introdução às **tecnologias web services: soa, soap, wsdl e uddi - parte1 Webmobile**. São Paulo, [s.n.], 2014. 217 p. Apostila.

FLEURY, S.; OUVENEY, A.M. **Gestão de redes: A estratégia de regionalização da política de saúde**. Rio de Janeiro: Editora FGV; 2007. 204 p.

GOMES, E. V. L.; TAVARES, L. A. **Uma solução com Arduino para controlar e monitorar processos industriais**. 2013. 6 f. Tese (Doutorado) - Curso de Pós-graduação em Engenharia de Sistemas Eletroeletrônicos, Automação e Controle Industrial, Instituto Nacional de Telecomunicações, Porto Alegre, 2013.

GOVERNO mais que dobra gasto social por cidadão de 1995 para 2010. **G1**, 2012. Disponível em: <<http://g1.globo.com/economia/noticia/2012/09/governo-mais-que-dobra-gasto-social-por-cidadao-de-1995-para-2010.html>>. Acesso em: 01 jun. 2014.

HIRAMA, K. **Engenharia de Software**: qualidade e produtividade com tecnologia. 9. ed. Porto Alegre: Elsevier, 2012.

KASUO, D. et al. **Projetos com Arduino**: 2010. Disponível em: <[http://www3.ufpa.br/marcelo/Home Page/Docs-APSH/Sem_APSH_1-2010/Projetos_com_Arduino.pdf](http://www3.ufpa.br/marcelo/Home%20Page/Docs-APSH/Sem_APSH_1-2010/Projetos_com_Arduino.pdf)>. Acesso em: 01 maio 2014.

KIKUYA, M; OHKUBO, T; ASAYAMA, K; et al. **Ambulatory blood pressure and 10-year risk of cardiovascular and noncardiovascular mortality: the OHASAMA study**. Hypertension.2005.

LAMIM, J. MVC - **O padrão de arquitetura de software**. Disponível em: <http://www.oficinadanet.com.br/artigo/1687/mvc_e_padrao_de_arquitetura_de_software>. Acesso em: 05 maio 2014.

LEITE, J. C. Processo de Software. **Engenharia de software**, 2007. Disponível em: <<http://engenhariadesoftware.blogspot.com.br/2007/03/processo-de-software.html>>. Acesso em: 20 mar. 2014.

LOBO, E.. **Guia prático de engenharia de software**. São Paulo: Digeratti Books, 2009.

OKUYAMA, F. Y.; MILETTO, E. M.; NICALAO, M. **Desenvolvimento de Software I: Conceitos Básicos - Série Tekne**. 1. ed. Porto Alegre: Bookman, 2014.z

PALMEIRA, T. V. V. Java: história e principais conceitos. **Devmedia**, 2013a. Disponível em:<<http://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>>. Acesso em: 20 maio 2014.

_____. Entendendo os conceitos dos Padrões de Projetos em Java. **Devmedia**, 2013b. Disponível em: <<http://www.devmedia.com.br/entendendo-os-conceitos-dos-padroes-de-projetos-em-java/29083>>. Acesso em: 20 maio 2014.

PAZ, L. F. et al. **Acesso móvel as informações de saúde do paciente utilizando computação ubíqua**, In: Congresso Brasileiro de Informática em Saúde, 11., 2012, Curitiba. Anais... . Curitiba: SBIS, 2012. v. 1, p. 120 - 126.

PIERIN, A. M. **Hipertensão arterial**. São Paulo: Manole, 2004.

POTTER, A. P.; PERRY, N. A. **Fundamentos de enfermagem**. 6. ed. Rio de Janeiro: Elsevier, 2005.

PRESSMAN, R. S.. **Engenharia de Software**: Uma Abordagem Profissional. 7. ed. São Paulo: Mcgraw-hill, 2011. 780 p.

RÉA NETO, Alvaro et al. **MONITORIZACAO EM UTI**. Revinter: Rio de Janeiro, 2004.

RIBEIRO, L. **O que é UML e Diagramas de Caso de Uso**: Introdução Prática à UML. 2013. Disponível em: <<http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408#ixzz3Jl41NvXh>>. Acesso em: 10 maio 2014.

RIVERA, Mario Cesar Massaro. **MVC**. 2012. Disponível em: <<http://www.devmedia.com.br/mvc/12202>>. Acesso em: 20 maio 2014.

RODRIGUES, E. J. L. **Curso de engenharia de software**: métodos e processos para garantir a qualidade no desenvolvimento de softwares. São Paulo: Digeratti Books, 2008.

SABBATINI, R. M. História da Informática em Saúde no Brasil. **Informática Médica**, Campinas, v. 5, n. 1, p.1-200, jul. 1998. Disponível em: <<http://www.informaticamedica.org.br/informaticamedica/n0105/sabbatini.htm>>. Acesso em: 20 maio 2014.

SANTOS, A. C. **Web Services em aplicações Android e iOS**. Disponível em: <<http://www.devmedia.com.br/web-services-em-aplicacoes-android-e-ios/28901>>. Acesso em: 05 maio 2014.

SILVA, P. C. B.. **Utilizando UML**: Diagrama de Classes. 2013. Disponível em: <<http://www.devmedia.com.br/artigo-sql-magazine-63-utilizando-uml-diagrama-de-classes/12251>>. Acesso em: 06 jun. 2014.

SAUDATE, A. **REST**: Construa API's inteligentes de maneira simples. 1. ed. São Paulo: Casa do Código, 2013.

SCHACH, S. R. **Engenharia de Software**: os paradigmas clássicos orientados a objetos. 7. ed. São Paula: Macgraw-hill, 2007.

SCHISSATO, J.; PEREIRA, R. O que são Design Patterns? **Agência digita Princiweb**, 2012. Disponível em: <<http://www.princiweb.com.br/blog/programacao/design-patterns/o-que-sao-design-patterns.html>>. Acesso em: 20 maio 2014.

SILVA, G. V.; ORTEGA, K. C.; MION JUNIOR, D. **Papel da MAPA e da MRPA na avaliação de pacientes com hipertensão de difícil controle**. Revista Brasileira de Hipertensão, São Paulo, v. 15, n. 1, p.17-20, jun. 2008.

SPÍNOLA, R. O. Engenharia de Software: Introdução à Engenharia de Requisitos Leia mais em: Artigo Engenharia de Software - Introdução à Engenharia de Requisitos. **Devmedia**, 2014. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034>>. Acesso em: 10 maio 2014.

SMITH, N. E.; TIMBY, B. K. **Enfermagem médico-cirurgico**. 8. ed. São Paulo: Manole, 2005.

SOARES, K. O que é um Arduino e o que pode ser feito com ele? **Techtudo**, 2013. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2013/10/o-que-e-um-arduino-e-o-que-pode-ser-feito-com-ele.html>>. Acesso em: 06 jun. 2014.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. São Paulo: Pearson Addison Wesley, 2003.

UNIFIESP. **Histórico Detalhado**. Disponível em: <<http://www.unifesp.br/dis/historico/historico-detalhado>>. Acesso em: 10 maio 2014.
WIDMAN, Lawrence E.. Sistemas Especialistas em Medicina. **Informática Médica**, Campinas, v. 5, n. 1, p.1-200, jul. 1998.

UNIVERSIDADE FEDERAL DE SÃO PAULO. Histórico Detalhado. **UNIFIESP**, c2008. Disponível em: <<http://www.unifesp.br/dis/historico/historico-detalhado>>. Acesso em: 10 maio 2014.

WIDMAN, L. E.. Sistemas Especialistas em Medicina. **Informática Médica**, Campinas, v. 5, n. 1, p.1-200, jul. 1998. Disponível em: <<http://www.informaticamedica.org.br/informaticamedica/n0105/widman.htm>>. Acesso em: 10 maio 2014.

WEISSMANN, H. L. **Vire o jogo com Spring Framework**. 1. ed. São Paulo: Casa do Código, 2013.