

WHMS4 - Um Sistema Integrado de Monitoramento Remoto de Saúde

Thiago de Souza Rios
GSORT - IFBA
Salvador, Bahia
Email: thiagorios@ifba.edu.br

Romildo Martins da Silva Bezerra
GSORT - IFBA
Salvador, Bahia
Email: romildo@ifba.edu.br

Resumo—Daily monitoring of vital signs using wearable devices is gaining more research and applications every day. This paper describes a medical monitoring system using a group of sensors coupled at the body of a patient. The system developed in this work aims to help health care professionals to monitor patients and also provide users with a treatment without interference in their routine.

Keywords—Sistemas embarcados, computação ubíqua, computação vestível para monitoramento da saúde (wearable health monitoring), biofeedback, Internet das coisas

I. INTRODUÇÃO

A Computação vem proporcionando ao longo dos anos inúmeros benefícios para a sociedade. Os avanços na construção de *hardware* e *software* trouxeram melhorias para diversas áreas do conhecimento humano. Vivemos numa época em que os dispositivos computacionais ganham cada vez mais importância no nosso cotidiano, influenciando nossa forma de realizar tarefas, adquirir informação e interagir socialmente.

Os objetos que nos cercam no dia-a-dia como carros, TV's e telefones estão sendo equipados com sensores e atuadores, deixando de ser simples objetos para transformarem-se em sistemas inteligentes, cada vez mais úteis e eficientes [1].

A relação entre a utilização de dispositivos inteligentes com os usuários de forma espontânea e natural é fundamentada pela Computação Ubíqua, teoria idealizada por Mark Weiser em meados dos anos 80. As grandes empresas de tecnologia do mundo já perceberam essa tendência e a cada ano, desenvolvem novos produtos baseados em conceitos ubíquos.

A Medicina é uma área beneficiada por esse cenário, pois novas aplicações para monitoramento médico que utilizam dispositivos embarcados surgem a cada dia [2]. Os recursos tecnológicos atuais podem ser utilizados por exemplo, no monitoramento remoto de pacientes sob tratamento médico [3][4] ou em constante avaliação física para fins esportivos [5].

Em alguns casos, são desenvolvidos sistemas capazes de prevenir doenças, avaliando continuamente o quadro de saúde de pacientes, detectando possíveis sintomas [2].

A comunidade científica comumente utiliza os termos *Electronic Health (eHealth)*, *Mobile Health (mHealth)* ou saúde móvel) e *Wearable Health Monitoring* (computação vestível para monitoramento da saúde) para designar a prática de

cuidados médicos suportados por processos eletrônicos e de comunicação.

A *mHealth* tem como foco desenvolver produtos que possuam poder computacional embutido. Esses dispositivos são utilizados para acompanhamento e avaliação do estado de saúde dos usuários, através do uso de sensores que captam dados biométricos como temperatura, batimentos cardíacos e respiração.

O objetivo no desenvolvimento desses produtos é proporcionar uma interação natural entre o usuário e o dispositivo, sem interferir nas suas atividades, sendo o mais invisível possível. O monitoramento contínuo realizado por esses sistemas é necessário para casos em que o tratamento médico exige verificação constante dos sinais vitais do paciente.

Os sistemas de monitoramento de saúde possibilitam também o acompanhamento remoto quando a presença física não for possível. Isso se adequa a realidade brasileira uma vez que existem cidades remotas com sérios problemas sociais. Já em grandes centros urbanos, a administração da logística para tais pacientes e o grande tráfego urbano podem comprometer a qualidade no atendimento.

Contudo, ainda existem alguns desafios para construção de sistemas baseados em *mHealth*:

- 1) Lidar com a imprecisão e heterogeneidade dos sensores e a confiabilidade dos dados;
- 2) Oferecer uma solução de baixo custo e adaptável à necessidade do usuário;
- 3) Oferecer um sistema de monitoramento integrado (equipe médica - atendimento de emergência - usuário) capaz de prover uma solução flexível e com boa interface gráfica.

A imprecisão dos sensores pode ser atenuada com o uso de técnicas de tolerância à falhas. Para o gerenciamento de energia são detectados momentos de ociosidade do sistema, limitando o seu funcionamento [4]. Para lidar com a heterogeneidade dos sensores, podem ser definidos padrões que generalizem o funcionamento e utilização dos sensores, que contemplem as diferentes saídas desses componentes.

A oferta de soluções de baixo custo é fundamental para viabilizar a utilização em larga escala deste tipo de serviço por pacientes sem recursos financeiros ou planos de saúde. Os dispositivos para monitoramento existentes costumam ter custo elevado e pouca flexibilidade para adaptação e customização dos sensores para situações diferentes.

Este trabalho propõe um sistema de monitoramento de sinais vitais de pacientes denominado WHMS4 (*Wearable Health Monitoring Simple Secure and Scalable System*), mediante a utilização de um conjunto de sensores acoplados à roupa de um paciente em tratamento, integrados em um microcontrolador visando o acompanhamento integrado dos pacientes, em especial, de idosos em asilos.

O objetivo do WHMS4 é fornecer informações do estado de saúde de pacientes idosos através da *Internet*, mediante a utilização de um dispositivo computacional vestível, customizável e de baixo custo. Através do monitoramento *online*, profissionais da área de saúde poderão acompanhar o estado dos pacientes em tratamento de qualquer lugar, nos casos em que a presença física não seja possível.

Esta monografia está organizada da seguinte forma: na Seção II são apresentados os conceitos básicos para entendimento do trabalho. A Seção III detalha os trabalhos relacionados. A Seção IV descreve os requisitos exigidos para construção do projeto. A Seção V apresenta uma visão geral sobre o projeto, incluindo funcionamento e arquitetura. A Seção VI detalha os requisitos especificados para a construção do projeto. A Seção VII descreve as etapas de desenvolvimento. A Seção VIII apresenta os resultados dos testes e validação do projeto. Na Seção IX é feita a conclusão do trabalho.

II. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta conceitos de Computação Ubíqua e Computação Vestível, discutindo suas características, desafios, requisitos e históricos de pesquisa. Serão apresentadas em particular as aplicações para monitoramento médico usando dispositivos vestíveis (*Wearable Health Monitoring*).

A. Computação Ubíqua

Em meados dos anos 80, Mark Weiser, cientista-chefe dos laboratórios da Xerox Parc, juntamente com sua equipe de pesquisadores, idealizou um futuro onde os computadores fariam cada vez mais parte do nosso cotidiano, embarcados em todos os objetos de forma transparente e onipresente [6].

Em um de seus artigos, Weiser introduziu o conceito da Computação Invisível (ou "*Calm Technology*"), projetando um futuro onde os computadores seriam utilizados sem a percepção das pessoas, ou seja, o computador transformaria-se em uma tecnologia ubíqua.

A característica principal dessas tecnologias é que elas são utilizadas sem uma preocupação maior com o seu manuseio ou como foram desenvolvidas. Por exemplo, a energia elétrica e o motor de um carro são tecnologias que estão impregnadas nas nossas vidas e que são utilizadas sem percebermos a complexidade que as envolve. Desta forma, o usuário terá um foco maior na tarefa e não em como realizá-la através da ferramenta.

Nesse contexto, a Computação passaria de um paradigma centralizado (onde cada pessoa possui um único computador pessoal) para um cenário altamente distribuído, onde os computadores estariam espalhados pelo ambiente e integrados aos objetos do dia a dia.

Cada pessoa possuiria vários dispositivos e estes, por sua vez, seriam responsáveis por realizar tarefas de forma

transparente, comunicando-se entre si para oferecer serviços aos usuários da forma mais natural possível [7]. A teoria idealizada por Weiser foi denominada de Computação Ubíqua.

A Computação Ubíqua é uma área de pesquisa que envolve a utilização de aplicações ou dispositivos que relacionam-se com o usuário de forma espontânea, pois não necessitam de uma interferência direta para seu controle. Geralmente, os sistemas ubíquos não possuem interfaces comuns para usuários (como *mouse*, teclados e janelas de aplicativos), porque utilizam formas mais naturais de interação como reconhecimento de gestos e voz. Os sistemas ubíquos possuem a capacidade de entender o contexto em que estão envolvidos para antecipar ou realizar ações sem que o usuário precise atuar de forma consciente.

Os avanços contínuos nas tecnologias de comunicação e semicondutores desde a década de 80 permitiram a criação de novos produtos, como microprocessadores que requerem menos espaço e consomem menos energia, protocolos de comunicação sem fio (*Bluetooth*[8], *Wi-Fi*[9], *ZigBee*[10]), dispositivos portáteis (*smartphones*, *tablets*) e sensores cada vez menores [11].

Graças a esses avanços e a redução do tamanho e preço do *hardware*, alguns conceitos imaginados para a Computação Ubíqua ainda na década de 80 podem ser realizados hoje [12]. Nesse cenário e com o advento da Computação Móvel, o computador tradicional começa a perder espaço para versões menores e de uso mais específico, como os microcontroladores. Esses dispositivos possuem uma maior capacidade de interação com o ambiente, possibilitando todo tipo de aplicação [13].

B. Sensibilidade ao Contexto

Um sistema ubíquo deve ser capaz de perceber o que está ao seu redor, por meio da análise de informações de eventos ocorridos em um ambiente, denominadas de contexto. Através dessas informações, é possível melhorar a interação homem-computador e produzir serviços mais úteis para os usuários [14].

A integração com o mundo físico em um sistema ubíquo é realizada pelo processamento de dados coletados a partir de um conjunto de sensores e de sistemas especializados em reconhecimento de contexto do ambiente.

Segundo [14], um contexto pode ser um lugar, a identidade de pessoas e objetos próximos ao sistema ou situações que ocorram no ambiente, sendo que há três aspectos importantes a serem considerados: onde o usuário está, quem ele é e quais recursos estão disponíveis.

Para [15], contexto é qualquer informação relacionada à aspectos de circunstâncias físicas que caracterizam a situação de uma entidade. Uma entidade é uma pessoa, um lugar ou objeto que é considerado relevante para a interação entre usuários e aplicações.

Essas informações incluem valores relativamente simples como localização, hora, temperatura, a identidade de um usuário operando um dispositivo, a presença ou estado de um objeto.

Um sistema é considerado *sensível ao contexto* se for utilizado para fornecer informações relevantes ou serviços

dependentes de informações de contexto [14]. Existem três categorias de recursos que uma aplicação sensível ao contexto pode suportar:

- Apresentação de informações e serviços ao usuário;
- Execução automática de um serviço para um usuário;
- Marcação de contexto à informação para apoiar a recuperação posterior.

Para que as informações de contexto do ambiente possam ser capturadas, são usados sensores de diferentes tipos e funções. Esses dados são utilizados para uma análise mais precisa da situação do ambiente e são em sua maioria, de natureza analógica.

Posteriormente, essas informações são convertidas para um formato digital e transmitidas à uma base computacional com maior poder de processamento, onde geralmente são efetuadas as deduções lógicas sobre os fatos.

C. Computação Vestível

A Computação Vestível é uma subárea da Computação Ubíqua e caracteriza-se pela utilização de computadores portáteis integrados em roupas ou até mesmo no corpo do usuário, sem causar desconforto [16]. A Computação Vestível pode ser definida como o estudo ou prática de construir ou usar dispositivos computacionais e sensores em miniatura sobre o corpo [17].

Os computadores vestíveis podem ser usados sobre as roupas ou também podem ser a própria roupa. Os dispositivos atuam com o objetivo de capturar dados de contexto do usuário para analisar sua condição física ou do ambiente que o envolve.

Idealmente, um computador vestível deve possuir algumas características com o objetivo de ser um assistente constante do usuário, percebendo o ambiente ao seu redor e entendendo como ele se relaciona com o mundo. Algumas dessas características são [18]:

- Persistir e prover constante acesso para serviços de informação;
- Adaptar as modalidades de interação baseadas no contexto do usuário;
- Prover interação mediada ou aumentada com o ambiente do usuário.

Em um nível mais físico e relacionado às características de *hardware*, os dispositivos vestíveis devem possuir como atributos [13]:

- Manuseio (*hands-free*);
- Sensores como câmeras, microfones, acelerômetros e GPS;
- Proatividade, pois devem fornecer informação mesmo não sendo utilizados diretamente;

D. Monitoramento da Saúde com Dispositivos Vestíveis

A população mundial está ao mesmo tempo crescendo e envelhecendo. Como consequência, existe uma maior preocupação das instituições de saúde mundiais com doenças relacionadas ao envelhecimento. Para essa faixa da população, há uma necessidade constante de cuidados especiais.

O problema é que os tratamentos domésticos de saúde (*homecare*) atuais estão cada vez mais caros [19]. Além dos idosos, existem outras parcelas da população que precisam desse tipo de tratamento, como recém operados, deficientes físicos ou pessoas que estão em processo de reabilitação.

Para resolver essas questões, surgiram formas inovadoras de tratamento doméstico (*home care*), baseadas no uso da tecnologia da informação, como os sistemas de monitoramento da saúde com dispositivos vestíveis (*Wearable health-monitoring systems - WHMS*) [4][19].

Segundo [19], os WHMS representam a nova geração de sistemas para cuidados de saúde, fornecendo em tempo real monitorização discreta de parâmetros fisiológicos dos pacientes através da implantação de vários sensores no corpo ou até mesmo dentro dele.

Esses sistemas caracterizam-se pela utilização de dispositivos eletrônicos móveis em objetos do cotidiano do usuário. Diferente de sistemas móveis convencionais, eles podem ser operados sem nenhuma ou com pouco interferência nas atividades do usuário [3].

E. Panorama da Saúde para Idosos no Brasil

O grande desafio de um sistema de saúde, público ou privado, é garantir acesso dos cidadãos ao atendimento de qualidade e eficácia. Em países com um vasto território e com grandes desigualdades sociais ainda é necessário ofertar um serviço reduzindo as desigualdades regionais, promovendo melhor equidade social.

Nesse contexto, torna-se mais difícil atender algumas classes de usuários do sistema público de saúde do Brasil como: idosos, portadores de necessidades especiais e pessoas recém-operadas ou que precisem de monitoramento contínuo.

No caso dos idosos, existe uma preocupação especial pois a população brasileira vem aumentando sua expectativa de vida a cada censo [20]. Isso torna o atendimento aos idosos um problema recente e não devidamente tratado. Faltam geriatras e casas de repouso (asilos) para os idosos terem um atendimento digno, uma vez que estes contribuíram com impostos ao longo da vida.

Um estudo do IPEA (Instituto de Pesquisa Econômica Aplicada) mostrou que as instituições públicas que abrigam os idosos não estão preparadas para o crescimento populacional desta categoria. Em 2010, o Brasil contava com 20 milhões de idosos, com projeções para 30 milhões de idosos em 2025 [20][21].

Por outro lado, atualmente o Brasil tem em torno de 5.500 instituições (em 5.570 cidades no Brasil), sendo apenas 238 delas públicas, sendo a maioria de origem filantrópica. Em geral prestam um bom atendimento social, mas falta acompanhamento médico contínuo nestes locais.

III. TRABALHOS RELACIONADOS

Existem diversos trabalhos na literatura relacionados a *Wearable Health Monitoring* que tratam desde a construção do *hardware* necessário, como também do desenvolvimento de *software/middleware* capazes de gerenciar recursos para o monitoramento dos pacientes.

Um dos trabalhos mais relevantes da área é *AMON: A Wearable Medical Computer for High Risk Patients* [3]. Na sua apresentação, é mostrado um dispositivo para monitoramento médico em forma de pulseira, equipado com sensores de pressão, temperatura e respiração. O sistema combina monitoramento médico complexo e possibilita análise de dados de forma confiável. Como ponto forte, a solução é totalmente integrada, mas em contrapartida sua arquitetura é fechada, não oferecendo flexibilidade para adaptação.

O projeto *Smart Clothes* [4] apresenta uma plataforma de monitoramento de saúde com uma arquitetura bem definida na utilização de sensores, com o objetivo de acompanhamento contínuo e de longo prazo. Utiliza cinco tipos de sensores diferentes para obtenção dos sinais vitais do usuário: (i) Eletrocardiograma (ECG); (ii) Temperatura; (iii) Acelerômetro; (iv) Eletromiografia (EMG); (v) Respiração.

Além dos sensores, a plataforma é composta por uma aplicação *web* e por uma combinação de *hardware* e dispositivos móveis (*smartphone*). A aplicação fornece diagnóstico com análise de situações de emergência.

Em [22] é demonstrado o projeto *SQUID*, uma camisa inteligente utilizada para monitorar exercícios de pessoas em reabilitação após sofrerem derrames. A plataforma utiliza eletrodos de ECG e motores de vibração em conjunto para indicar os níveis de atividade cardíaca.

Em [19] é descrito um *framework* para sistemas de monitoramento de saúde, chamado *Prognosis*. O projeto tem a proposta de fornecer um modelo de diagnóstico de possíveis doenças, usando algoritmos de inteligência artificial como lógica *fuzzy* e modelagem dos resultados usando redes de Petri.

Existem soluções que apresentam uma rede de sensores acoplada ao corpo, onde o conjunto dos sinais enviados são usados para monitorar pacientes em reabilitação [23][24]. O sistema realiza a análise dos sinais dos sensores, fornecendo orientação e *feedback* para o usuário através da geração de alertas com base no seu estado. As informações registradas são transferidas para um servidor na Internet, integrado à uma base de dados de registros médicos.

Em [25] é apresentada uma camisa inteligente que pode medir sinais de aceleração para vigilância contínua da saúde, utilizando sensores e tecidos condutores (como eletrodos) para obter sinais vitais do corpo. Os dados dos sensores de ECG e de atividades físicas são transmitidos em uma rede sem fio para uma estação base.

Esses dados são armazenados em um servidor para consulta e monitoramento remoto. Um diferencial deste projeto é a redução física dos dispositivos e baixo consumo de energia para reduzir o tamanho da bateria.

Em [26] é proposto um sistema de monitoramento físico e psicológico usando sensores acoplados ao corpo, com o intuito

de auxiliar pessoas que precisam de atendimento domiciliar, deficientes, idosos e pessoas que estão sendo evacuadas de locais onde aconteceram desastres. Além dos sensores para coletar sinais vitais como temperatura, pressão e batimentos cardíacos, o sistema utiliza GPS para localizar os pacientes e sensores de eletroencefalograma para detectar o estado psicológico do usuário.

O projeto *WeCare* consiste no desenvolvimento de um *hardware* específico com o objetivo de monitorar pacientes com doenças cardiovasculares [27]. O *WeCare* possui uma abordagem de coleta de sinais vitais tolerante à erros, utilizando um conjunto maior de eletrodos de ECG para garantir o acompanhamento contínuo do paciente.

Em [28] é apresentado um sistema para análise de sinais vitais usando acelerômetros em conjunto com sensores de ECG. O projeto utiliza um algoritmo RVM (*Relevance Vector Machine*)¹ para diminuir a complexidade computacional do monitoramento. A combinação dos sensores garante precisão na obtenção de informações, pois somente os dados de um tipo de sensor são insuficientes para definir o estado de um paciente.

Em [29] foi desenvolvido um dispositivo para monitoramento de sinais vitais em forma de cinto, utilizando sensores de ECG, acelerômetros e sensores de temperatura. A captura e envio dos dados funciona da mesma forma que outros projetos correlatos. A aquisição de dados é dividida em módulos, um para cada tipo de leitura de sinal vital.

O trabalho apresentado em [30] propõe um sistema de monitoramento de saúde utilizando a plataforma *Arduino* [31]. São utilizados sensores de temperatura, respiração e de transpiração da pele como entrada para os sinais vitais do corpo. Assim como em outros projetos, o objetivo do sistema apresentado é auxiliar pacientes idosos ou em estado de risco.

Em [32] é apresentado um sistema de bio-monitoramento, usado para aquisição de parâmetros fisiológicos como ECG, EMG e respiração durante as atividades diárias do usuário. O funcionamento do sistema é baseado na utilização de unidades eletrônicas bio sensíveis e de eletrodos, conectados através de materiais condutivos flexíveis (*e-textiles*).

Essas unidades são incorporadas em uma camisa de material especial, que garante a contração muscular e aumento da circulação sanguínea. A escolha por esse material foi feita para aumentar o contato dos eletrodos com a pele do usuário, maximizando o efeito elétrico. O sistema é dividido em quatro partes funcionais: unidades bio-sensíveis, meio de transmissão (fios de costura condutores), unidade eletrônica para processamento dos sinais e *software* para análise dos dados.

A elaboração da camisa com eletrodos e a escolha dos materiais para o monitoramento é o ponto forte do trabalho apresentado. Porém, a visualização dos dados monitorados é *offline*, através de uma aplicação *desktop*, tornando a plataforma dependente de uma estrutura inflexível para as exigências de mobilidade atuais.

¹RVM é uma técnica de aprendizado de máquina que usa inferências de redes Bayesianas para obter soluções de classificação probabilística.

Trabalho	Ano	Autor	Resumo	Comparativo
<i>AMON: A Wearable Medical Computer for High Risk Patients</i>	2004	Urs Anliker, Jamie A. Ward, Paul Lukowicz, et. al.	Apresenta um dispositivo de monitoramento em forma de pulseira, integrando sensores capazes de coletar múltiplos sinais vitais de pacientes sob cuidados médicos	O dispositivo embarcado será acoplado em uma camisa, com sensores coletando sinais vitais do paciente
<i>Smart-Clothes - Prototyping of a Health Monitoring Platform</i>	2013	Yung-Cheng Ma, Yi-Ping Chao, and Tzung-You Tsai	O projeto consiste em uma roupa inteligente com vários sensores embarcados com o objetivo de coletar os sinais vitais do usuário.	Além do dispositivo de monitoramento, o projeto terá aplicações responsáveis pela indicação do quadro de saúde do paciente, mediante informações de status fornecidas pelos médicos
<i>Squid: Sensorized shirt with smartphone interface for exercise monitoring and home rehabilitation</i>	2013	A. Farjadian, M. Sivak, and C. Mavroidis	Camisa inteligente para monitorar exercícios de pessoas que sofreram derrames. Utiliza sensores e motores de vibração para detectar os níveis cardíacos dos pacientes	Aplicação com um enfoque específico para monitoramento de pacientes com problemas cardiovasculares, enquanto que o WHMS4 possui abrangência dependendo dos sensores e parâmetros utilizados
<i>Prognosis - A Wearable Health-Monitoring System for People at Risk: Methodology and Modeling</i>	2010	Alexandros Panteopoulos	Apresenta um <i>framework</i> para sistemas de monitoramento de sinais vitais, fornecendo um modelo de diagnósticos de doenças.	A solução proposta trata-se de um sistema distribuído, envolvendo um sistema embarcado, uma aplicação móvel e uma aplicação <i>web</i>
<i>A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation</i>	2005	E. Jovanov, A. Milenkovic, C. Otto, and P. de Groen	É demonstrada a utilização de uma rede de sensores acoplada ao corpo para monitoramento de sinais vitais.	Os sensores serão acoplados à roupa, podendo ser removidos individualmente e não espalhados pelo corpo, o que poderia dificultar a utilização e implantação dos sistema
<i>Mobicon: Mobile context monitoring platform: Incorporating context-awareness to smartphonecentric personal sensor networks</i>	2012	Y. Lee, Y. Ju, C. Min, J. Yu, and J. Song	O <i>Mobicon</i> é um <i>middleware</i> para monitoramento de sinais vitais, que fornece uma API para integração dos dados dos sensores com qualquer aplicação	A abordagem do <i>Mobicon</i> é diferente, pois a aplicação é quem requisita os dados dos sensores e estes não são enviados de forma contínua. No WHMS4, a aplicação espera o recebimento dos dados
<i>Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring</i>	2009	Y.-D. Lee and W.-Y. Chung	Neste projeto foi construído um <i>chip</i> específico para monitoramento de saúde, utilizando sensores de ECG e eletrodos para captar sinais vitais do corpo	O projeto tem enfoque maior no <i>hardware</i> . O principal objetivo do WHMS4 é fornecer um sistema final para indicação do estado de saúde do paciente. Além disso, não foi demonstrada a utilização de outro tipo de sensor no projeto
<i>Wearable Physical and Psychological Health Monitoring System</i>	2013	Kohei Arai	Sistema de monitoramento físico e psicológico através de sensores acoplados ao corpo.	Esse projeto tem como enfoque auxiliar pessoas que estão em áreas de desastres. O WHMS4 permite o monitoramento personalizado de outros tratamentos
<i>WE-CARE: An Intelligent Mobile Telecardiology System to Enable mHealth Applications</i>	2013	Anpeng Huang, Chao Chen, Student Kaigui Bian, et. al.	Desenvolvimento de um plataforma de computação vestível para monitorar pacientes com doenças cardiovasculares	O sistema é específico para monitoramento de problemas cardiovasculares, enquanto que o WHMS4 não possui um uso especializado
<i>Human daily activity recognition by fusing accelerometer and multi-lead ecg data</i>	2013	R. Jia and B. Liu	Sistema de monitoramento de sinais vitais utilizando sensores de ECG e acelerômetro. Combina os dados dos sensores para análise do estado de saúde do paciente	O WHMS4 não fornece análise do quadro de saúde, somente a indicação dos sinais vitais
<i>A wearable health monitoring system with multi-parameters</i>	2013	J. Liu, F. Xie, Y. Zhou	Projeto é constituído por um cinto para monitoramento de sinais vitais, utilizando sensores de temperatura, ECG e acelerômetros	O WHMS4 não tem como finalidade a construção de <i>hardware</i> específico
<i>Application of arduino based platform for wearable health monitoring system</i>	2013	A. Sugathan, G. Roy, G. Kirthyvijay, and J. Thomson	Sistema de monitoramento de saúde utilizando a plataforma <i>Arduino</i> . São utilizados sensores de temperatura, respiração e resposta dermo-elétrica da pele como entrada para os sinais vitais do corpo	Possui um enfoque maior no <i>hardware</i> do que o WHMS4
<i>Bio-monitoring system with conductive textile electrodes integrated into t-shirt</i>	2014	M. Hanic, L. Sladek, F. Horinek, M. Jagelka, et. al.	Sistema de bio-monitoramento de sinais vitais baseado na utilização de <i>e-textiles</i> , eletrodos e unidades bio sensíveis	Este projeto possui um trabalho mais aprimorado na construção do <i>hardware</i> , através da escolha e utilização de materiais específicos para a monitorização

Tabela I. QUADRO COMPARATIVO DOS TRABALHOS RELACIONADOS

O WHMS4 tem como diferencial o desenvolvimento da aplicação final para coleta de dados e acompanhamento dos pacientes. Mesmo que seja utilizado um *hardware* semelhante ao que foi apresentado nos trabalhos desta seção, o sistema será construído para aceitar a utilização de outros dispositivos. A Tabela I apresenta um quadro comparativo dos trabalhos relacionados com WHMS4 de forma a evidenciar as diferenças dos projetos.

IV. WHMS4 - REQUISITOS NÃO-FUNCAIONAIS E ASPECTOS DE COMPUTAÇÃO UBÍQUA

O WHSM4 tem como objetivo principal, inicialmente, monitorar o estado de saúde de idosos em algum tipo de tratamento, fornecendo dados de sinais vitais através da *Internet* mediante a utilização de um dispositivo embarcado em uma roupa.

O escopo deste trabalho não é garantir o monitoramento *online* dos usuários, porque pacientes em estado crítico devem estar sob os cuidados de um centro médico em leitos de atendimento intensivo ou semi-intensivo. O WHMS4 também não realizará uma análise mais profunda de problemas ou sintomas clínicos do paciente.

O projeto não tem como objetivo desenvolver *hardware* específico para a realizar a captação dos sinais vitais, utilizando técnicas modernas de microeletrônica. O sistema embarcado será desenvolvido com plataformas que disponibilizam funcionalidades básicas para implementação de um protótipo.

Sendo assim, o projeto tem um enfoque maior no sistema de apoio ao monitoramento dos pacientes, através da identificação dos sinais vitais e dados de localização.

Os objetivos específicos do projeto proposto são:

- Fornecer uma interface simples e intuitiva, com informações de identificação, localização, sinais vitais e estatísticas de monitoramento dos usuários;
- Possibilitar a modularização dos sensores de forma que possam ser substituídos conforme a necessidade, sendo todas as versões da roupa suportadas pelo sistema de monitoramento;
- Desenvolver uma aplicação móvel para *smartphone*, de forma a integrar os dados do microcontrolador à um servidor remoto;

Para alcançar esses objetivos listados, alguns requisitos devem ser atendidos.

A. Requisitos Não-Funcionais

No contexto social de países emergentes, um requisito importante é o investimento necessário (mas não o principal) para que a solução apresentada seja utilizada pela população com poucos recursos financeiros. Para isso, o produto final deve possuir baixo custo com relação a soluções existentes no mercado.

O WHMS4 deve possuir flexibilidade para adaptação e integração com outros sistemas (por exemplo, sistemas de Centros Médicos ou Hospitais), fornecendo uma interface para troca e utilização das informações. A flexibilidade deve prever versões de *hardware* e capacidades (do inglês *capabilities*) de diferentes dispositivos permitindo a interoperabilidade entre eles.

O WHMS4 deve ser confiável, pois trata-se de uma aplicação que lida com informações vitais do usuário. A confiança no funcionamento (em inglês *dependability*) é a habilidade para entregar um serviço que pode justificadamente ser confiável [33].

Além disso, a comunicação de dados deve prover criptografia e utilizar um canal seguro. O sistema deve ofertar visões de acesso, reduzindo o escopo da aplicação de acordo com o perfil do utilizador, evitando eventuais erros de operação.

O sistema deve prover a escalabilidade, permitindo sua ampliação e mantendo a operação estável. A escalabilidade é uma característica desejável em qualquer sistema ou infraestrutura que manipule uma porção crescente de trabalho de forma uniforme, mantendo a qualidade do serviço mesmo com o aumento de sua demanda [34]. A Figura 1 apresenta os requisitos funcionais descritos nesta seção.

B. Aspectos de Computação Ubíqua

Soluções ubíquas são amplamente baseadas na integração de dispositivos físicos sem interface com o usuário. Para atingir esse integração, os sistemas ubíquos devem atender alguns requisitos, listados na Figura 2.

Um aspecto fundamental de sistemas ubíquos é a Heterogeneidade. Essa característica relaciona-se com a capacidade de integração do sistema com diferentes arquiteturas de *hardware* e *software*, provenientes de fabricantes diferentes.

Outro aspecto da Computação Ubíqua é a Invisibilidade. A definição formal expressa por Weiser consiste no desaparecimento completo da tecnologia de computação em função da



Figura 1. Requisitos Funcionais



Figura 2. Aspectos Ubicomp

consciência do usuário [6][7]. Na prática, uma aproximação razoável deste ideal é a mínima percepção do usuário.

A Sensibilidade ao Contexto é outra característica relevante. Sistemas ubíquos devem ser capazes de entender o mundo físico ao seu redor, através da utilização de estruturas (ontologias ou bases de conhecimento) que descrevam o ambiente em que ele está sendo utilizado e de sensores que extraíam informações de contexto.

Por fim, a Localização consiste na identificação de posicionamento do usuário dentro do sistema. Um alerta emitido no

contexto da computação vestível pode ter criticidade diferente em função da localidade informada.

V. WHMS4 - VISÃO GERAL

Esta seção apresenta uma visão geral do WHMS4, mostrando características, definições, arquitetura e funcionamento geral da aplicação.

A. Arquitetura e Funcionamento

O WHMS4 é composto de três subssistemas com características distintas, que integrados são capazes de prover os objetivos especificados anteriormente que denominaremos de camadas: Camada Embarcada, Camada Móvel e Camada Web (Figura 3).

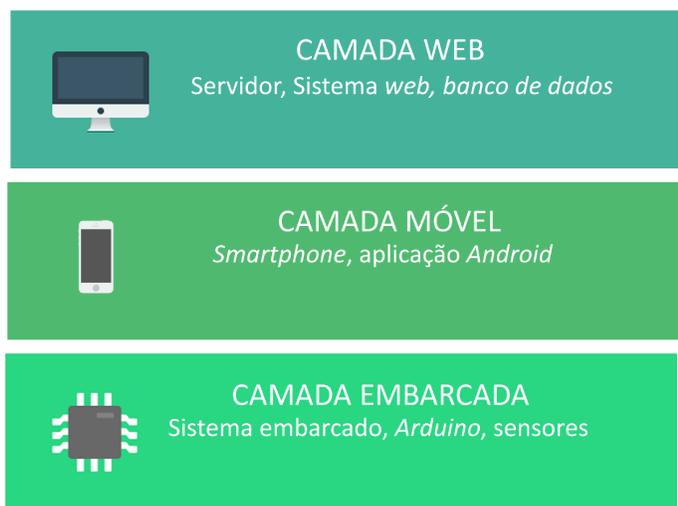


Figura 3. Camadas do WHMS4

As camadas interligam-se por diferentes interfaces de rede para fornecer aos usuários um sistema único, conforme exibido na Figura 4. Utilizando essa divisão, as funcionalidades e características de cada camada são:

- **Camada Física (ou Embarcada)** - Corresponde ao sistema embarcado na roupa (*Wearable Computing*) que coleta as informações vitais dos pacientes através de sensores distribuídos em sua extensão. Visando reduzir o índice de erros na coleta dos dados aplicamos uma coleta com múltiplos sensores, descartando os que estiverem fora do desvio padrão. Podem ser utilizadas roupas desenvolvidas por empresas desde que tenham suporte a *Bluetooth* ou 802.11 e exportem dados em modo texto;
- **Camada Móvel** - Camada composta por uma aplicação para dispositivos móveis. Nessa camada é realizado o envio dos dados para a Camada Web, utilizando uma conexão *Wi-Fi* ou 3G/4G. Essa aplicação atua como intermediária entre a Camada Física e a Camada Web, permitindo redução de custo do *hardware* ao eliminar a necessidade de uma placa *Wi-Fi* para o Arduino, maximizando a utilização da bateria. Nesta camada são implementados conversores de dados para

que a adaptabilidade da solução seja possível com produtos comerciais;

- **Camada Web** - Essa camada apresenta o sistema final para os usuários. É composta por uma aplicação *web*, responsável pela visualização e gerenciamento de todas as informações e logística operacional do modelo. Os dados de monitoramento são exibidos nessa aplicação, onde o estado de saúde do paciente poderá ser avaliado por profissionais da área médica.

VI. WHMS4 - ESPECIFICAÇÕES DO SISTEMA

Nesta seção são apresentadas as especificações do WHMS4. As camadas já apresentadas nas seções anteriores possuem modelos diferentes, mas que ao final, deverão servir para explicar o funcionamento do sistema como um todo.

Será usada uma abordagem *top-down*, onde as camadas são descritas por ordem de complexidade. Inicialmente, será apresentado o modelo da camada *web*, por tratar-se do sistema final para o usuário e por ser o elemento onde a maioria das informações são manipuladas e exibidas.

Nesta abordagem assume-se que as camadas que estão no topo da ordenação esperam dados num formato especificado das camadas imediatamente inferiores.

Portanto, é possível desenvolver o modelo de uma camada sem a preocupação de como os elementos de outras camadas funcionam. Essa característica torna o sistema interdependente, pois as camadas são desenvolvidas para lidar com interfaces distintas.

A. Estrutura do Software Web

O sistema *web* representa o sistema como um todo para o usuário final e tem como funcionalidade permitir o gerenciamento dos elementos do projeto. Nele é possível analisar as informações de sinais vitais dos pacientes cadastrados, mediante a visualização em telas e gráficos.

Os usuários do WHMS4 possuem perfis associados entre médico, auxiliar/enfermeiro e administrador. O cadastro dos elementos como sensores, roupas, tratamentos e alertas é feito por esse grupo de usuários. A Figura 5 apresenta um diagrama de caso de uso para esses perfis.

O médico é responsável por definir os tratamentos que serão aplicados aos pacientes, cadastrando os parâmetros que serão utilizados para comparação com os valores lidos pelo sistema embarcado. Os pacientes são usuários indiretos, porque ao utilizar o sistema embarcado enviam dados para o sistema *web*. Cada roupa possui um tipo diferente, personalizada conforme o tratamento ao qual o paciente está associado.

Um tratamento define um tipo de roupa com recursos e capacidades específicas, através da utilização de sensores diferentes entre os modelos. Os dados recebidos no sistema *web* contém a identificação do paciente, sua localização e valores de leitura dos sensores. Em intervalos de tempo especificados, esses valores são comparados com os requisitos cadastrados no tratamento, como forma de avaliar o *status* do paciente.

Os profissionais de saúde que estão prestando monitoramento podem emitir alertas quando algum paciente atingir um

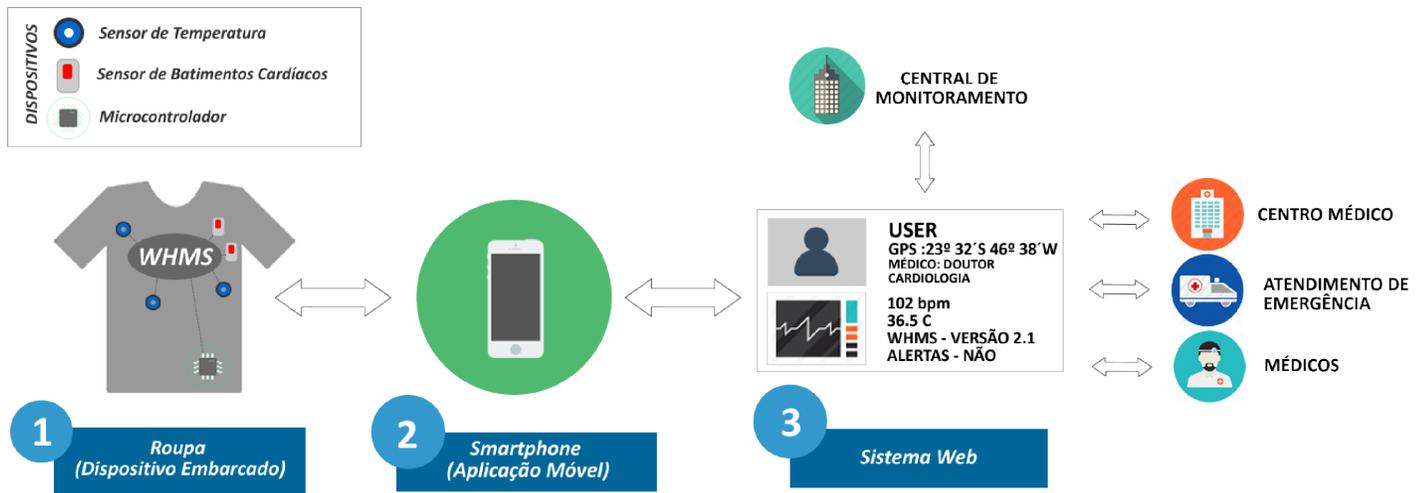


Figura 4. Visão Geral da Arquitetura

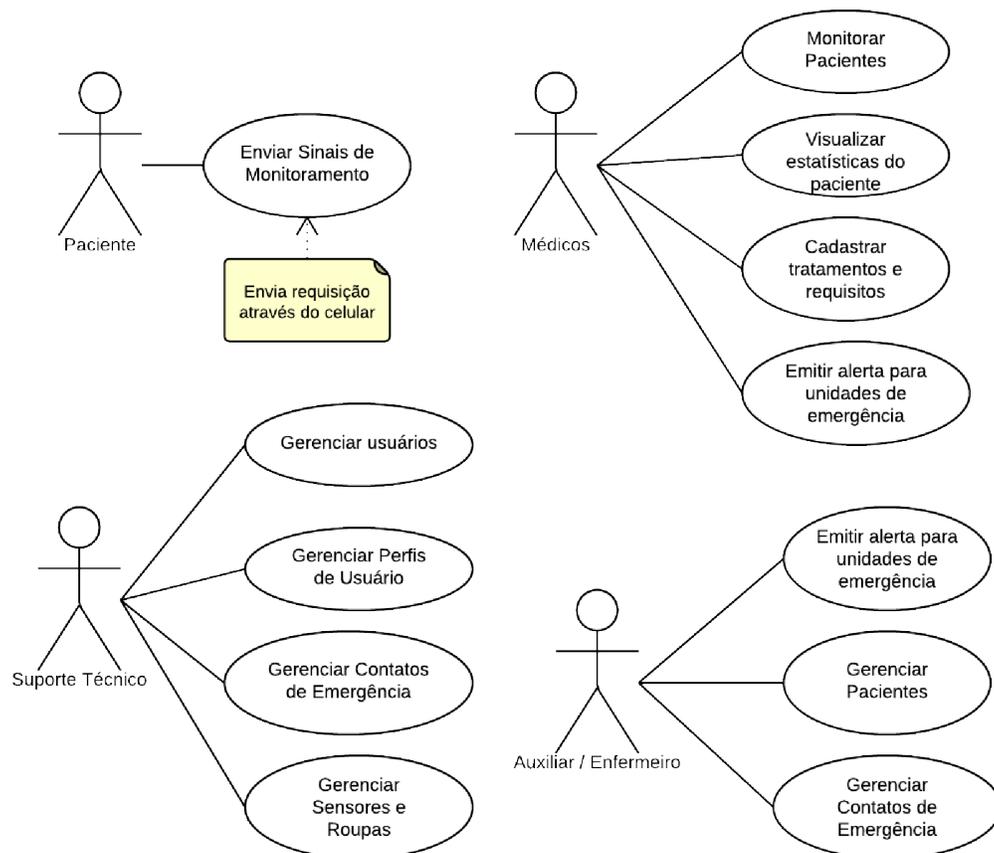


Figura 5. Diagrama de Caso de Uso

nível de urgência. Devido à sua maior complexidade, o sistema *web* é dividido em módulos principais, contendo classes para funcionalidades específicas. As classes do sistema podem ser vistas no diagrama exibido na Figura 6. Os módulos utilizados no sistema são:

- **Módulo Base** - Módulo responsável por gerenciar as informações básicas para funcionamento do sistema, como pacientes, usuários e tratamentos, correspondentes a um prontuário médico móvel;

- **Módulo de Aquisição de Dados** - Módulo responsável por receber e analisar as informações dos dispositivos recebidas pelo servidor, realizando os tratamentos e validações necessárias para poder aumentar a adaptabilidade do modelo apresentado;
- **Módulo de Conversão de Dados** - Módulo que permite a reutilização dos dados por outras plataformas de *software*, através da disponibilização de interfaces utilizando formato de dados padronizados como *XML (eXtensible Markup Language)* e *JSON (Javascript Object Notation)*;
- **Módulo de Tratamento e Requisitos** - Módulo responsável por gerenciar as informações de tratamento inseridas por profissionais de saúde. A análise do quadro de saúde dos pacientes é realizada por esse módulo, mediante a comparação dos valores especificados para um determinado tratamento com os valores recebidos no servidor. O histórico do paciente pode ser analisado pelo médico especialista que o assiste;
- **Módulos de Alertas** - O módulo de alertas serve para cadastrar avisos para unidades de emergência em casos críticos. Os alertas para os centros médicos podem ser feitos por telefone, *e-mail*, SMS, *WhatsApp* ou mensagem de voz. Isso permitirá maior agilidade no atendimento e uma análise prévia do prontuário do paciente pela equipe de emergência que se deslocará para o local. Caso esteja integrado com um sistema de gerenciamento de trânsito, os médicos podem receber a rota indicada;

Baseado nas regras e definições dos módulos mencionados, foram elaborados diagramas de classe e entidade relacionamento para o sistema. Os diagramas podem ser vistos nas Figuras 6 e 7. O diagrama mostrado na Figura 6 apresenta as classes separadas por módulos (ver legenda).

Inicialmente, essas serão as classes que o sistema utilizará para seu funcionamento, mas novos módulos ou recursos poderão ser adicionados conforme necessidade. O diagrama na Figura 7 apresenta a modelagem para o banco de dados da aplicação. As tabelas são baseadas nas definições de classes apresentadas no diagrama da Figura 6, sendo que algumas foram criadas para dar suporte ao modelo de entidade relacionamento.

Algumas classes utilizam mais de uma tabela internamente, como forma de granularizar a informação e evitar a redundância dos dados. Nesse sentido, os dados comuns de usuários, pacientes e médicos foram agrupados em uma entidade denominada *Pessoa*, de forma a evitar que a estrutura das tabelas fosse repetida.

O diagrama de classe oculta algumas das tabelas mostradas no diagrama entidade relacionamento, pois trata de informações específicas para a normalização da base. Por exemplo, no diagrama de classes não existe um objeto para a tabela *TipoPessoa*, pois entende-se como um atributo da classe *Pessoa*. A Figura 8 exhibe o Diagrama de Sequência do principal fluxo da camada *web*, descrevendo as etapas para recebimento e armazenamento dos sinais vitais do paciente.

B. Estrutura do Software Móvel

A aplicação móvel do WHMS4 funciona como um elemento intermediário entre o sistema embarcado e o sistema *web*, enviando dados entre esses componentes utilizando recursos nativos do *smartphone*, como *WiFi*, *Bluetooth* e GPS.

O funcionamento básico da aplicação deve permitir a descoberta e pareamento do *smartphone* com o dispositivo embarcado através do uso do *Bluetooth*. Após o pareamento, a aplicação recebe continuamente os dados do *Arduino* e em seguida os envia para o sistema *web* através da interface *WiFi*.

A divisão da aplicação é feita em poucas classes, pois reutiliza bibliotecas nativas do *Android* para gerenciamento do *Bluetooth*, *WiFi* e envio de dados via HTTP. A classe principal deverá possuir além dos métodos padrão para aplicações *Android*, métodos de verificação dos recursos utilizados.

Essas funções habilitam automaticamente os recursos quando o usuário inicia o aplicativo, facilitando sua utilização. A aplicação contém funções que gerenciam o envio dos dados para a Camada *Web* em um endereço especificado, permitindo que o usuário escolha o dispositivo que irá conectar-se ao *smartphone* para o recebimento dos dados.

O aplicativo deverá armazenar as configurações do endereço do servidor para onde os dados serão enviados e também a identificação do paciente que está sendo monitorado. Para organização do código da aplicação, as funções responsáveis por listar e detectar os dispositivos prontos para pareamento usando o *Bluetooth* serão associados à uma classe específica.

A Figura 9 demonstra o diagrama de classes baseado nas especificações para a aplicação móvel do WHMS4. A aplicação é dividida em quatro classes (*Main*, *DeviceList*, *ConnectedThread* e *Settings*), sendo que a classe *Activity* é uma classe genérica da plataforma *Android*, da qual todas as demais classes tem herança.

A classe principal (*Main*) tem somente métodos que fazem chamada direta para esses métodos ou a métodos adaptadores dos recursos utilizados. Por padrão, a placa *Bluetooth* utilizada em conjunto com o *Arduino* possui uma identificação padrão para pareamento.

Esta classe é responsável por iniciar a aplicação e todo o seu ciclo de vida (executar, parar, pausar e destruir). Esse ciclo é controlado por funções nativas do *Android* (*onCreate*, *onDestroy*, *onStop*, *onBackPressed*) e algumas devem ser implementadas por padrão.

Nessa classe residem os métodos para iniciar os adaptadores do *Bluetooth* e da antena *WiFi* (*checkBluetoothState* e *checkWiFiState*, respectivamente). A função *connectDevice* recebe os dados da lista de dispositivos pareados (gerada pela classe *DeviceList*) e realiza a conexão com o dispositivo escolhido pelo usuário (através da função *connectToDevice*).

A função *connectToDevice* é responsável pela criação do canal de comunicação com o sistema embarcado, iniciando a tarefa paralela para recepção dos dados (utilizando a classe *ConnectedThread*). Por fim, a função *postHttp* encaminha os dados recebidos do sistema embarcado para o sistema *web* do WHMS4.

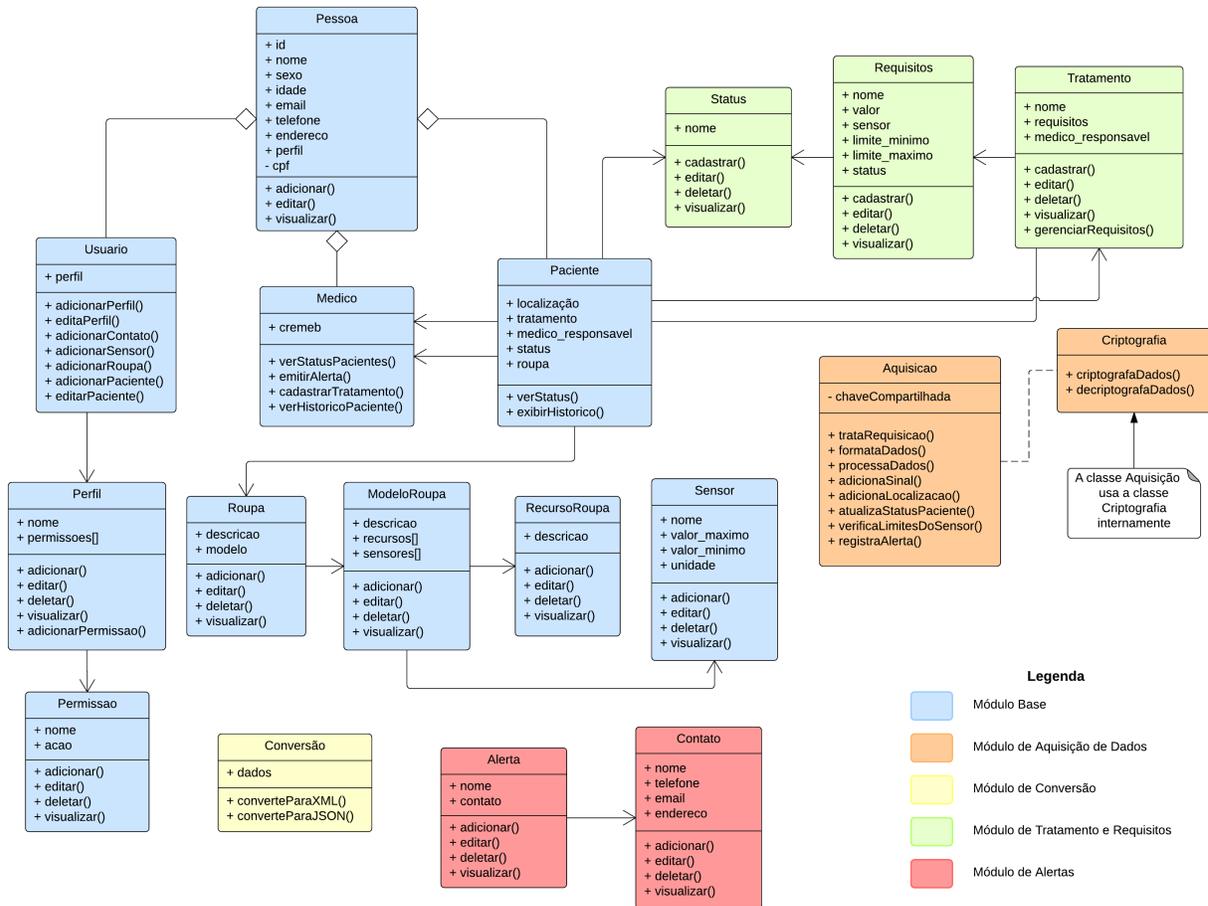


Figura 6. Diagrama de Classes

A classe *DeviceList* foi criada para descobrir e listar os dispositivos com *Bluetooth* ativado para conexão. Nessa classe estão os métodos que utilizam o adaptador instanciado na classe principal para descobrir novos dispositivos e pará-los com o *smartphone*.

A classe *ConnectedThread* é específica para o gerenciamento do canal de comunicação entre o sistema embarcado e a aplicação *Android*. Essa classe implementa uma *thread*, para que a tarefa de receber dados do sistema embarcado seja realizada de forma assíncrona, em paralelo com as demais funções do aplicativo.

A classe *Settings* disponibiliza uma tela onde o usuário poderá gravar preferências da aplicação, como endereço do servidor remoto para envio dos dados e identificação do paciente. Assim que as mudanças são confirmadas, as novas configurações são salvas e a aplicação é reiniciada.

C. Estrutura de Hardware e Software Embarcado

O sistema embarcado do WHMS4 envolve a construção e montagem de *hardware* para leitura de sinais vitais do

paciente. Para realizar a leitura desses sinais, será utilizado um microcontrolador da plataforma *Arduino*, conectado a um grupo de sensores e a um módulo *Bluetooth*.

Esse módulo é responsável pela comunicação do sistema embarcado na roupa com a aplicação móvel no *smartphone*. O sistema embarcado envia continuamente os dados lidos dos sensores para a aplicação móvel. Os componentes básicos para o desenvolvimento desse sistema são:

- Um microcontrolador (placa do *Arduino*);
- Sensores;
- Fonte de Alimentação;
- Módulo *Bluetooth*;

Na computação, falhas em *software* são comuns. No caso de sistemas embarcados, as falhas podem ocasionar seu mau funcionamento a longo prazo, criando desconfiança por parte do usuário quanto ao dispositivo utilizado. Segundo [35], o grau de confiança de um sistema é medido pela crença por parte do usuário de que seu funcionamento corresponderá

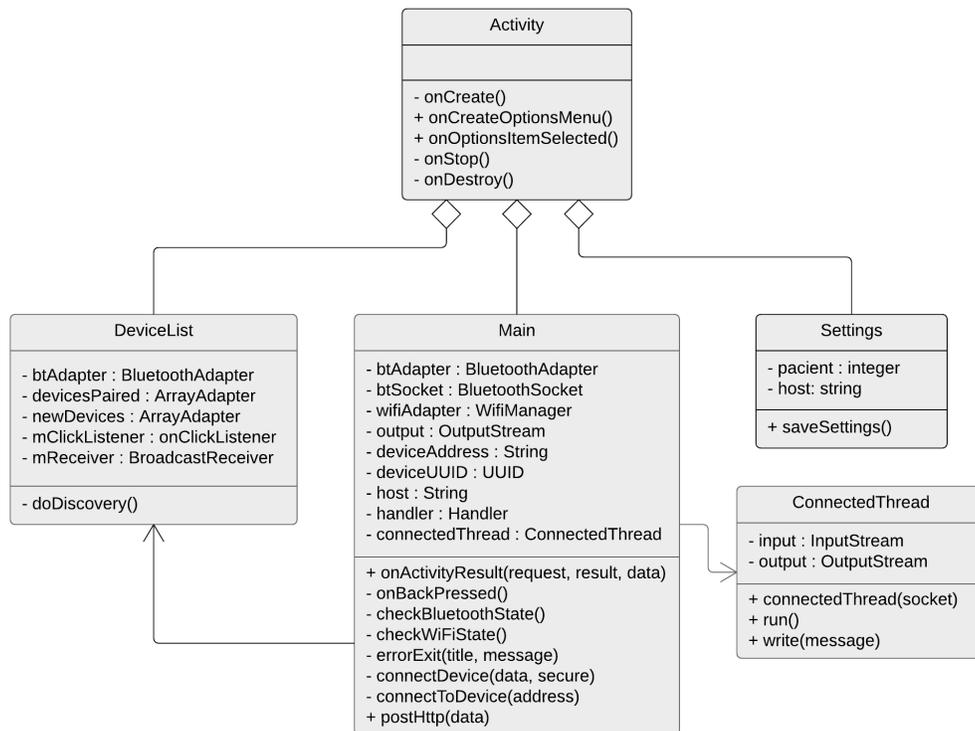


Figura 9. Diagrama de Classes da Aplicação Móvel

conforme suas expectativas, não ocasionando falhas durante sua utilização normal. A partir desse conceito, há quatro graus de confiança, conforme listados a seguir:

- **Disponibilidade:** probabilidade do sistema estar sempre pronto para execução, fornecendo serviços úteis a qualquer instante;
- **Confiabilidade:** característica do sistema em fornecer serviços de forma correta, em um dado período de tempo, conforme esperado pelo usuário;
- **Segurança:** um sistema não deve causar danos ao usuário ou ambiente;
- **Proteção:** resistência contra acessos acidentais ou intencionais.

As propriedades listadas podem ser decompostas em propriedades mais específicas, como por exemplo, a *confiabilidade* pode incluir *correção* (garantia de que os serviços estejam conforme especificados) e *precisão* (garantia de que a informação seja liberada em um certo nível de detalhe) [35].

Além das quatro dimensões, podem ser adicionados outros requisitos de confiança dependendo do sistema, como tolerância a erros e capacidade de recuperação. Para o desenvolvimento de um sistema embarcado, alguns requisitos referentes ao *hardware* escolhido devem ser levados em consideração.

Sistemas embarcados normalmente tem rígidas restrições sobre funcionalidade e implementação. Eles devem garantir a operação em tempo real e reagir a estímulos externos [36]. Além disso, devem satisfazer os requisitos de confiabilidade e segurança. Alguns dos requisitos para o *hardware* do WHMS4 são:

- **Processamento correto dos sinais:** O processamento e reação aos estímulos devem ser realizados de forma que não prejudique a operação sendo feita em um dado momento. A computação correta da operação (*correctude*) depende do tempo em que o processamento dos estímulos foram entregues;
- **Limites de tamanho e peso:** O sistema proposto deve ser utilizado constantemente pelo usuário. Por isso, os limites de tamanho e peso devem ser considerados como os menores possíveis, de forma a proporcionar uma utilização mais confortável para o usuário;
- **Baixo consumo de energia e dissipação de calor:** Os dispositivos escolhidos deverão consumir pouca energia e dissipar pouco calor. O consumo de energia é uma questão importante, pois demanda gerenciamento das fontes de energia utilizadas, através de mecanismos que detectem períodos de inatividade do sistema para economizar energia;

- *Baixo custo:* O *hardware* escolhido deverá possuir baixo custo, visto que o sistema proposto tem como objetivo atingir diversas classes da população que necessitem de um monitoramento médico através de um sistema embarcado.

1) *Especificação do hardware utilizado:* O microcontrolador escolhido para o desenvolvimento deste trabalho foi o *Arduino Pro Micro*, por apresentar o melhor custo benefício entre os modelos estudados [31]. A utilização de um microcontrolador pequeno é justificada com o intuito de oferecer mais conforto na utilização da camisa.

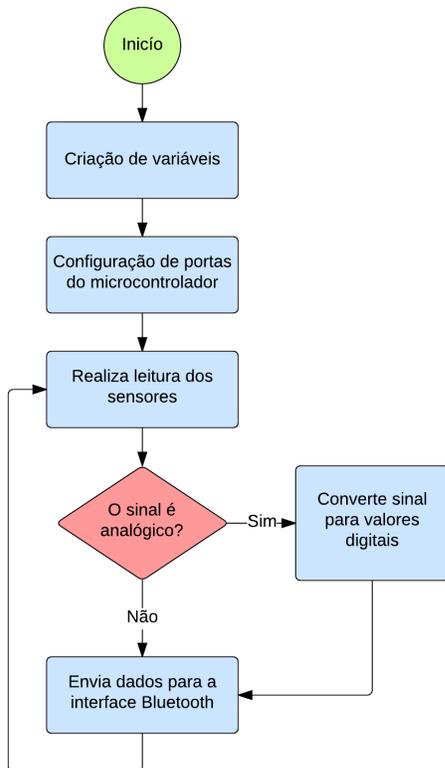


Figura 10. Fluxograma do Sistema Embarcado

Esse modelo é um dos menores da plataforma e possui uma conexão USB integrada, evitando o uso de placas externas para gravação do programa no microcontrolador. Essa característica compensa o custo mais alto (em média \$10) [37] em comparação com outros modelos semelhantes da plataforma.

Mesmo que fosse utilizado um modelo de menor custo, seria necessário a compra de uma placa externa, acarretando um custo maior no orçamento e perda de praticidade no desenvolvimento do projeto.

O *Arduino Pro Micro* possui a quantidade suficiente de pinos para interligação com os sensores (14 digitais e 8 analógicos), de forma que poderão ser utilizados sensores em redundância para a análise dos sinais. A plataforma escolhida permite a substituição dos componentes interligados de forma personalizada, oferecendo flexibilidade na construção do sistema.

O modelo escolhido possui um processador interno de 16MHz e suas dimensões são de 3,30 x 1,77 cm [37]. O tempo de leitura necessário para cada um dos sensores ainda precisa ser avaliado para indicar se o microprocessador atende aos requisitos de operação do sistema embarcado. Geralmente, esse valor é suficiente de acordo com outros projetos que utilizam o mesmo dispositivo [30].

O *software* desenvolvido para o sistema embarcado do WHMS4 foi programado para efetuar a leitura de todos os sensores a ele conectados, verificando continuamente os sinais gerados. Após a leitura dos sensores, os dados obtidos passam por etapa de conversão, dependendo do tipo de dado obtido. Como exemplo, o sinal obtido no sensor de temperatura é analógico, devendo então ser convertido para valores digitais, mais próximo do entendimento humano. A Figura 10 exhibe um fluxograma de funcionamento do programa que deverá ser gravado no *Arduino*.

O programa representado no fluxograma segue basicamente a estrutura padrão dos programas do *Arduino*: possui uma função de configuração dos pinos de entrada e saída (*setup*) e uma função para controle e manipulação do que deve ser lido e acionado pelo microcontrolador (*loop*). Na função *loop*, a leitura de sensores é feita de forma contínua, até o *Arduino* ser desligado. O acionamento é representado pela escrita dos valores lidos na interface *Bluetooth*.

Como mencionado anteriormente, o desenvolvimento do *hardware* não é o enfoque do projeto. Por esse motivo, alguns dos requisitos listados não foram cumpridos na construção da camada embarcada. Os requisitos específicos e de confiabilidade da camada embarcada do WHMS4 listados são necessários para o desenvolvimento completo do sistema, mas somente podem ser atingidos com um nível de produção que envolve recursos tecnológicos específicos para construção de *hardware*.

VII. WHMS4 - DESENVOLVIMENTO

Esta seção descreve as etapas de desenvolvimento dos sistemas que compõem o WHMS4. Após a fase de modelagem e análise dos requisitos para cada camada, o *software/hardware* foi construído baseando-se no que foi apresentado nos diagramas e fluxogramas.

As ferramentas e metodologias escolhidas para o desenvolvimento do projeto levaram em consideração algumas das decisões tomadas na etapa de modelagem (como a escolha do *Arduino* e da plataforma *Android*).

A. Desenvolvimento do Sistema Embarcado

Com a utilização do *Arduino* na Camada Física, o desenvolvimento do sistema embarcado foi facilitado, devido à abstração da complexidade da montagem de *hardware* oferecida pela plataforma. Para programação do microcontrolador, foi utilizada a IDE² padrão do *Arduino*, disponibilizada através do site oficial da plataforma, sem custo adicional [31].

Para montagem do sistema embarcado, foi necessário planejar o esquema de ligação dos sensores e demais componentes

²Do inglês "Integrated Development Environment" (Ambiente Integrado de Desenvolvimento)

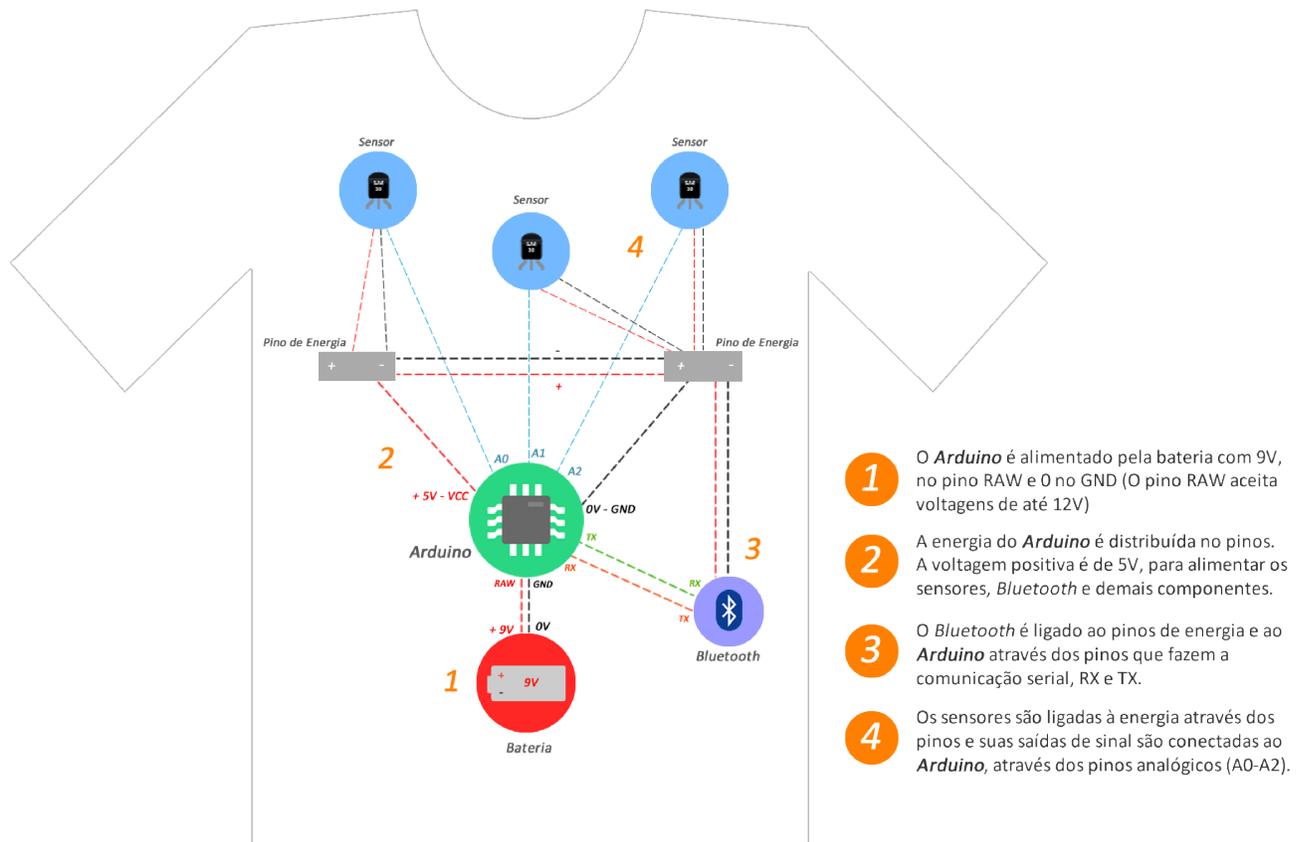


Figura 11. Esquema de ligação dos componentes na camisa

ao **Arduino**, antes de efetuar as ligações. A Figura 11 ilustra a ligação dos componentes na camisa, seguindo a especificação da modelagem.

Seguindo o esquema elétrico, foram adquiridos os demais componentes para construção do sistema embarcado como resistores, placas PCB, sensores, pinos para conexão e fios. Na Figura 12 temos o **Arduino Pro Micro** ainda sem pinos para conexão, em sua configuração padrão.

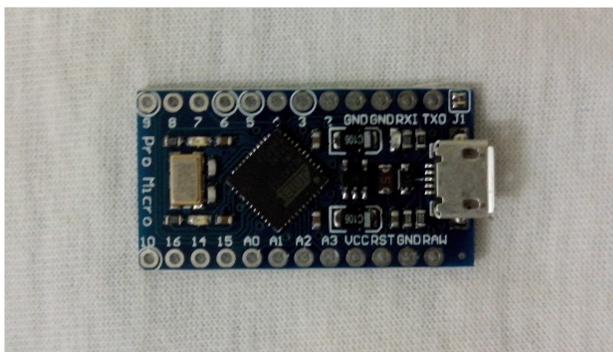


Figura 12. Arduino Pro Micro

Para que fosse possível utilizar o **Arduino** em conjunto com o **Bluetooth**, foi utilizada uma bateria de 9V. O **Arduino Pro Micro** possui uma entrada para voltagens mais altas que o

padrão (5V), que permite que sejam usadas outras fontes de energia. Na Figura 11, é possível visualizar como a energia foi distribuída pelos componentes.

Para disponibilizar uma voltagem de 5V para os componentes, foi utilizada a saída VCC do **Arduino**, ligando fios às barras de pinos (conforme Figura 11). Os sensores e o **Bluetooth** foram ligados à essas barras para funcionar. Por fim, foi feita a ligação das portas seriais da placa **Bluetooth** e dos sensores com o **Arduino**.

B. Desenvolvimento da Aplicação Móvel

Para o desenvolvimento da aplicação móvel foi escolhida a plataforma **Android**, por ser a mais utilizada no mercado e possibilitar o desenvolvimento de aplicativos sem custo adicional, licenciamento ou compra de ferramentas.

A plataforma permite a criação de aplicações móveis utilizando a linguagem Java. Existem diversas ferramentas de uso gratuito para criação desses aplicativos (Eclipse, NetBeans), mas por recomendação do site oficial para desenvolvedores da plataforma³, foi utilizado o **Android Studio**.

Com base na modelagem, as classes do diagrama apresentados na Figura 9 foram reproduzidas na IDE. A tela principal da aplicação é demonstrada na Figura 13. Para cada classe

³<http://developer.android.com/index.html>



Figura 13. Tela principal da aplicação móvel

especificada para controlar recursos dentro da aplicação, foi criada uma *Activity*, classe que representa uma tela com seus respectivos componentes, em que os usuários podem interagir para realizar algo.

Cada *Activity* possui uma janela onde é renderizada sua interface (incluindo botões, caixa de texto, etc). As *Activities* possuem ciclo de vida, composto por estados da tela durante a execução da aplicação.

Para leitura dos dados da interface *Bluetooth*, foi criada a classe *ConectedThread*, responsável por receber continuamente os dados após a conexão ser realizada. Os dados de localização são obtidos através da utilização da biblioteca *Location* do *GooglePlay Services*.

Com esses recursos configurados, os dados de sinais vitais e localização são enviados continuamente (de acordo com o intervalo de envio do *Arduino*) para o servidor *web*, onde será feita uma análise mais apurada dos dados.

C. Desenvolvimento do Sistema Web e do Banco de Dados

O sistema *web* é o principal componente do WHMS4, por se tratar do elemento apresentado ao usuário final (médicos, enfermeiros e auxiliares).

A camada *web* compreende não só a aplicação, mas também a base de dados responsável por armazenar os valores dos sinais vitais enviados pelas camadas inferiores até o servidor. O número de funcionalidades nessa camada é bem maior do que nas outras, o que demanda um maior planejamento da aplicação.

Para produzir a aplicação, foi escolhida a linguagem PHP devido à sua simplicidade e difusão no mercado de desenvol-

vimento *web*, utilizando o *framework*⁴ *CakePHP* que segue o padrão de projeto MVC⁵.

O uso de um *framework* implica na utilização de um padrão de projeto (no caso o MVC) no processo de desenvolvimento, o que acaba por facilitar a modelagem do banco de dados e código da aplicação.

Como o foco do projeto não é o desenvolvimento desses módulos, a escolha desse *framework* considera a abstração na criação dessas funcionalidades, em detrimento do desenvolvimento de módulos que envolvam efetivamente os requisitos definidos nas seções de especificação.

A ferramenta escolhida para a criação do banco de dados foi o *MySQL*, por padrão configurado no *CakePHP*. O *MySQL* é muito utilizado em aplicações que não necessitam de um maior desempenho no acesso à dados.

Com a possibilidade de crescimento do sistema, pode-se utilizar outra ferramenta para banco de dados, pois o *CakePHP* permite a adaptação de diferentes bases de dados. O uso de um *framework* baseado no padrão MVC possibilita a independência do modelo de dados com relação ao SGBD (Sistema de Gerenciamento de Banco de Dados) usado. Assim, a possibilidade de migração para outra plataforma é facilitada, pois a definição do modelo não está somente no banco, mas também representada na aplicação.

Nas figuras a seguir são exibidas algumas telas da aplicação *web*. A Figura 14 exhibe a tela inicial (*dashboard*), onde os usuários podem ver informações gerais sobre os pacientes monitorados. A Figura 15 apresenta a página de visualização de dados de um determinado paciente. Nesta tela os usuários podem ver com detalhes o estado atual do paciente, suas informações de sinais vitais e localização.

VIII. TESTES E VALIDAÇÃO

O objetivo da validação é confirmar a possibilidade de utilização da aplicação conforme os requisitos definidos. Mesmo com os teste funcionais para cada camada, é necessário realizar testes de utilização reais como forma de simular percepções dos usuários sobre o projeto.

Portanto, além dos testes funcionais, a aplicação também deve passar por testes de desempenho. Para realizar a validação, podemos dividir os testes em dois grupos: testes funcionais e testes de desempenho.

A. Testes Funcionais

Os testes funcionais (ou testes unitários) envolvem a utilização de módulos da aplicação como forma de avaliar funções, consultas e classes, no intuito de garantir que funcionem de acordo com a especificação. Geralmente são escritos pelo próprio desenvolvedor da aplicação ou por uma equipe especializada em testes.

⁴Um *framework* é um conjunto de códigos comuns em diversas aplicações provendo uma funcionalidade genérica. Geralmente, o *framework* impõe regras para o fluxo das aplicações.

⁵Da sigla em inglês *Model View Control*. Denota um padrão de projeto que divide as aplicações em camadas distintas, a saber: uma camada de visualização (*view*), uma camada de acesso e controle de dados (*model*) e uma camada de controle de fluxo da aplicação (*controller*.)

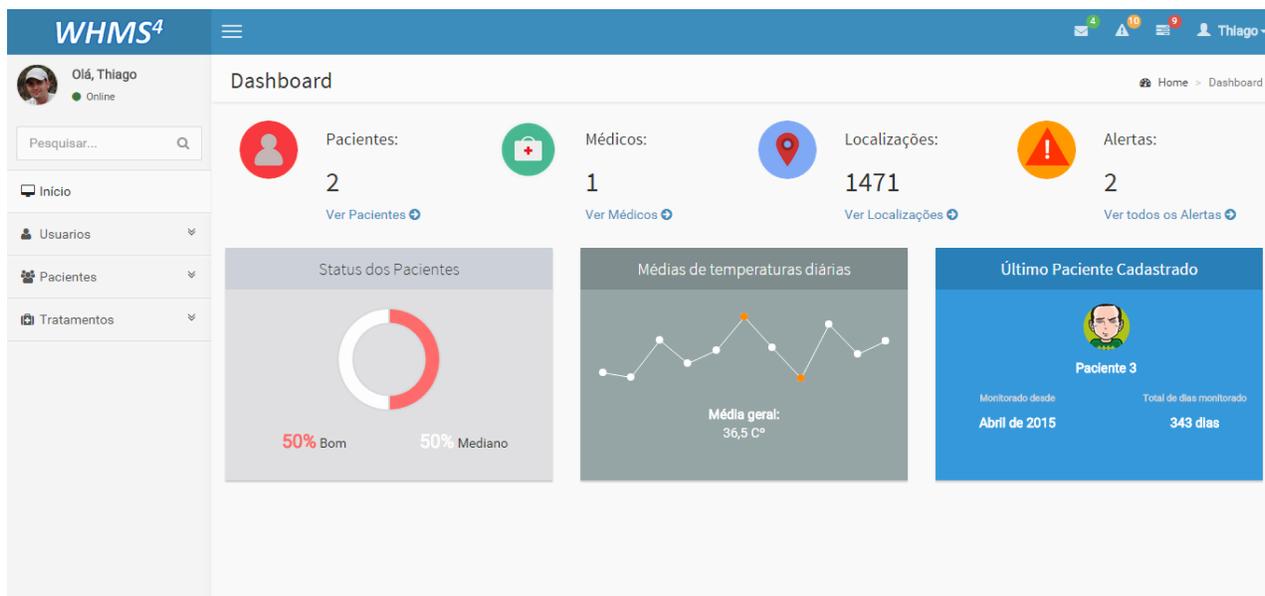


Figura 14. Tela inicial do sistema web

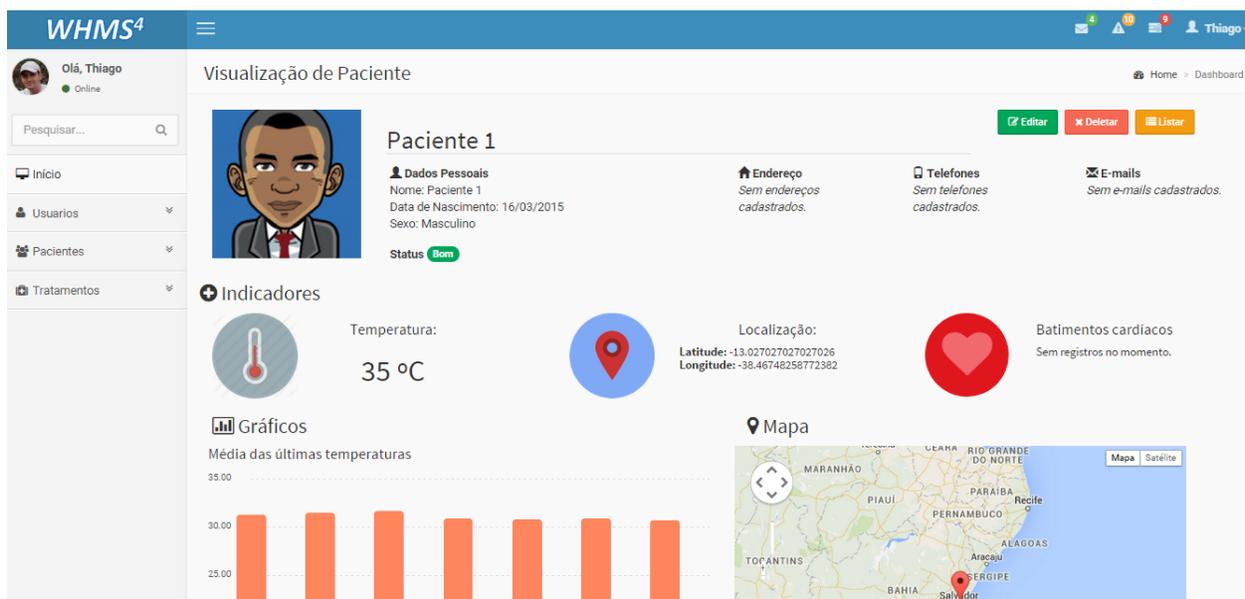


Figura 15. Tela de visualização de dados dos pacientes

No caso do WHMS4, com a utilização do *CakePHP*, o módulo de testes é incluído por padrão, necessitando somente da criação dos casos de teste para os elementos da aplicação. O *CakePHP* utiliza o *PHPUnit*⁶, um *framework* para testes de aplicações desenvolvidas em PHP.

Para validar as funcionalidades da aplicação são criados arquivos de caso de teste, responsáveis por entender a estrutura do arquivo sendo testado verificando suas entradas e saídas através de asserções (*assertions*) nos valores testados.

B. Testes de Desempenho

Os testes de desempenho são definidos pela utilização de técnicas e ferramentas para determinar a responsividade, confiança e/ou escalabilidade de um sistema sob uma carga de trabalho.

Geralmente, esses testes envolvem a análise de tempos de resposta, taxas de transferências (*throughput*) em determinados momentos e a utilização de recursos que sirvam para determinar a performance da aplicação [38].

Os testes de desempenho são realizados para:

- Avaliar a disponibilidade do ambiente de produção;
- Comparar características de desempenho de vários

⁶<https://phpunit.de/>

sistemas ou configurações de um sistema;

- Encontrar a fonte de problemas de desempenho;
- Ajustar o suporte do sistema;
- Encontrar os níveis de taxas de transferências (*throughput*).

Os testes de desempenho são divididos em alguns tipos, comumente utilizados para aplicações *web*:

- **Testes de carga** (*load tests*): esse teste verifica o comportamento da aplicação sob condições normais e em picos de carga. Permite medir tempos de respostas, taxas de transferência e níveis de utilização de recursos para determinar o ponto de falha da aplicação;
- **Testes de capacidade ou escalabilidade** (*capacity or scalability tests*): teste utilizado para determinar quantos usuários ou transações um dado sistema irá suportar, continuando a atender os objetivos de performance, ou seja, serve para testar a escalabilidade do sistema. Este teste é usado para indicar a inclusão de recursos adicionais caso o número de usuários ou volume de dados aumente;
- **Testes de stress** (*stress tests*): serve para determinar ou validar o comportamento da aplicação quando condicionado à condições não esperadas ou extremas, como em picos de carga altos ou falha de algum recurso do sistema. Este teste é utilizado principalmente para revelar defeitos na aplicação que só são descobertos em situações atípicas;

Para realizar os testes relacionados, deverá ser utilizado algum mecanismo que simule as requisições HTTP de forma automática, usando possíveis cenários de utilização da aplicação. Entre as diversas opções de ferramentas, optou-se por utilizar o *JMeter*⁷, por apresentar mais vantagens de acordo com o escopo do projeto.

O *JMeter* é uma ferramenta de uso livre, desenvolvida e mantida pela *Apache Foundation*, feita em Java. O objetivo do *JMeter* é realizar testes de desempenho para medir performance e analisar o comportamento das aplicações nas situações de carga de uso.

Os testes incluem parâmetros que são utilizados para indicar a situação da aplicação sob funcionamento normal ou em carga excessiva. Nos testes, são verificados vazão de dados por segundo (*throughput*), tempo de resposta das requisições, número de usuários virtuais utilizando a ferramenta e tempo de duração dos testes.

Para todos os cenários de teste foi utilizado um servidor na nuvem (*cloud hosting*), com a configuração mais básica disponível: processador *Intel Pentium Dual-Core*, frequência de 2.0GHz, 1GB de memória RAM, sistema operacional *Linux*, rodando PHP através do *Lampp*, utilizando *Apache* como *web server* e banco de dados *MySQL*.

Os testes apresentados tem como foco a funcionalidade de armazenamento e análise de sinais vitais. Assume-se que o tempo de resposta aceitável para esta operação é de 10

segundos, pois esse é o intervalo de tempo estipulado em que o sinal chega ao servidor.

Para simular essa funcionalidade, foi utilizada uma função do módulo de *Aquisição* que consome dados de sinais vitais e localização no formato XML aceitos pela aplicação (o mesmo formato de dados enviado pela Camada Móvel).

Os testes também incluem cenários que além do recebimento de sinais, irão processar as demais atividades do sistema como visualização de pacientes, requisitos e tratamentos.

Teste de Carga

Neste teste, utilizaremos um grupo virtual de 20 usuários, sendo iniciados em intervalos de 0,1 segundos, utilizando a funcionalidade de envio de sinais, durante o tempo de 3 minutos. O limite de tempo de resposta é de 10 segundos.

De acordo com o gráfico de Tempo de Resposta (Figura 16), podemos ver que para um grupo de 20 usuários, o sistema funcionou como esperado, pois não ultrapassou o limite de processamento da requisição, mantendo o processamento das requisições entre 1 e 2ms.

Teste de Escalabilidade

Para testar a escalabilidade do sistema, devemos aumentar a carga de processamento concorrente, verificando em que ponto a aplicação começa a deteriorar seus serviços. Esse teste é importante para identificar possíveis gargalos da aplicação, com o objetivo de melhorar as funções que precisam de mais performance.

O cenário para este teste utilizará 50 usuários, iniciados a cada 0,1 segundos, acessando as mesmas funcionalidades do teste de carga, durante o tempo de 3 minutos. Os resultados do teste podem ser vistos na Figura 17.

C. Resultados dos testes

Pelos gráficos apresentados nota-se que com o aumento do número usuários concorrentes o tempo de resposta foi acrescido. Contudo, esse mesmo tempo foi mantido para todos usuários ao longo da duração dos testes. Isso indica que o sistema não sofreu uma grande degradação em seu desempenho, porque manteve um padrão no tempo de processamento das requisições. A tabela II apresenta a comparação dos dados dos dois testes. A estrutura utilizada ajustou-se automaticamente para atender uma quantidade grande de usuários concorrentes de forma homogênea.

Os resultados mostraram-se satisfatórios, pois mesmo utilizando um servidor com configuração básica, que possui recursos limitados, foi possível ter noção do funcionamento do projeto caso seja necessário o aumento de capacidade do servidor da aplicação.

Teste	Qtd. de Usuários	Inicialização	Requisições	Throughput (Vazão)
Carga	20	0,1s	1251	648,155 req/min
Escalabilidade	50	0,1s	3321	623,919 req/min

Tabela II. RESULTADOS DOS TESTES DE DESEMPENHO

⁷<http://jmeter.apache.org/>

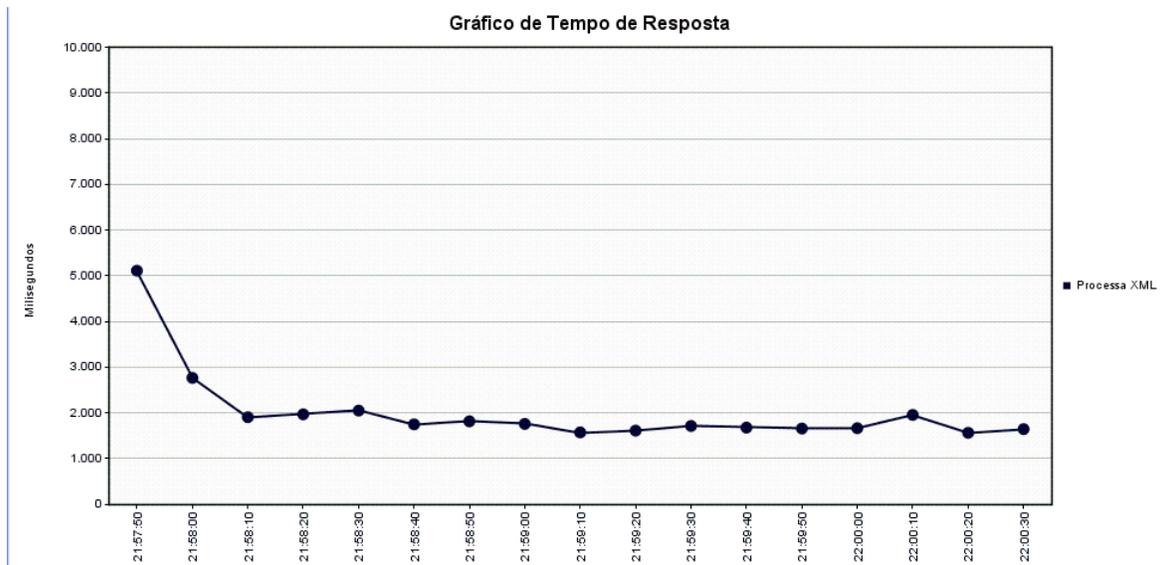


Figura 16. Gráfico de Tempo de Resposta para Teste de Carga

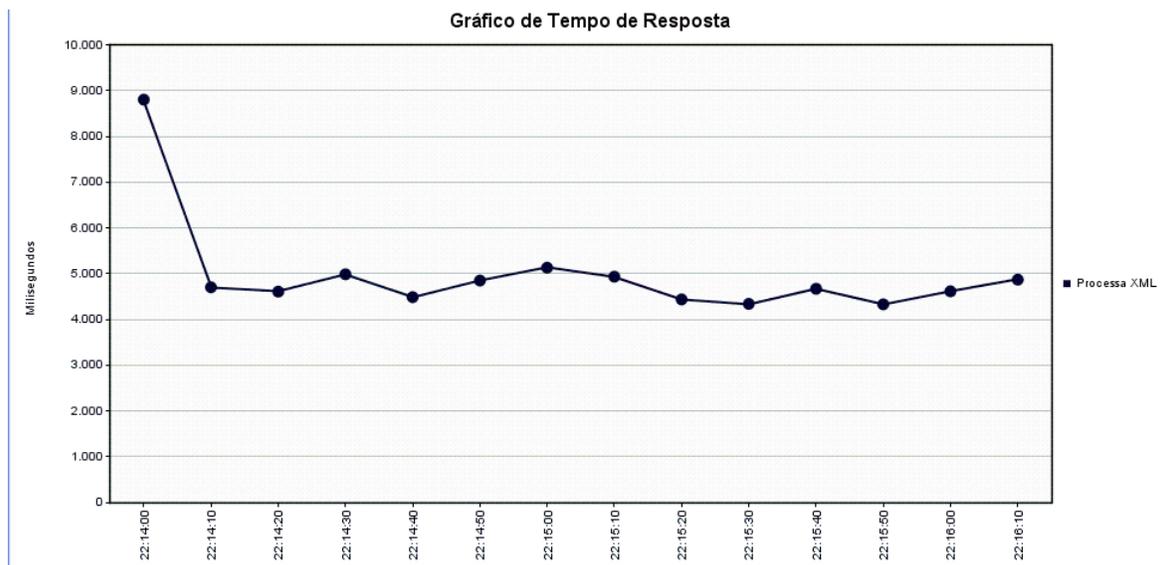


Figura 17. Gráfico de Tempo de Resposta para Teste de Escalabilidade

IX. CONCLUSÃO

Os recursos computacionais atuais podem beneficiar a sociedade a obter uma melhor qualidade de vida. Para atingir esse objetivo, pesquisas na área de Computação Ubíqua tem como meta proporcionar uma interação mais natural entre o homem e a máquina, de forma que o uso da tecnologia seja cada vez mais frequente no nosso cotidiano, de forma benéfica.

A utilização de técnicas e conceitos ubíquos para monitoramento de saúde vem sendo cada vez mais disseminada pelo mundo, oferecendo produtos que tornam as atividades de monitoramento menos intrusivas para os usuários.

Através do monitoramento intensivo, é possível prestar um serviço que será usado sem maiores intervenções no cotidiano dos usuários. O sistema auxiliará os médicos e demais profissionais da área de saúde a gerenciar o tempo de serviço prestado

aos seus pacientes, quando não houver uma necessidade maior de acompanhamento médico presencial.

O WHMS4 em sua versão atual consegue cumprir a maioria dos requisitos listados. O produto final tem baixo custo, por utilizar tecnologias que são de uso livre (PHP, *Android*) em conjunto com recursos que são acessíveis financeiramente, se comparados com tecnologias ou produtos semelhantes (plataforma *Arduino*).

O projeto também é flexível, devido a possibilidade de configuração dos parâmetros dos tratamentos. A confiabilidade é um requisito que não é atendido em sua totalidade, pois não foram realizados teste em pacientes para afirmar que o seu estado de saúde é fielmente representado através das informações geradas pelo sistema. Por fim, o WHMS4 atende ao requisito de escalabilidade, de acordo com os testes realizados na seção anterior.

Com relação aos aspectos ubíquos, somente a invisibilidade é minimamente cumprida, pois necessita que a plataforma de sensoriamento seja produzida de forma a incorporar-se à camisa sem que o paciente tenha consciência de que deve manusear um dispositivo eletrônico.

Para um futuro aperfeiçoamento do projeto e uma possível disponibilização como produto, o sistema embarcado deveria ser desenvolvido sem a utilização de plataformas de prototipagem. O ideal é que fosse construído um *hardware* específico, usando um *chip* que pudesse transmitir diretamente os dados via *WiFi*, com fonte de energia interna e portas de conexão para entrada/saída de componentes.

No tecido usado para construir a roupa, deveriam ser usados fio de costura ou tintas condutoras, como barramentos ligando os sensores espalhados na camisa ao *chip* de leitura dos sinais. Os sensores também podem ser padronizados para possuir um mesmo número de saídas padronizadas, de forma que pudessem ser utilizados em qualquer parte.

O projeto poderia ser estendido para utilização em outros objetos que pudessem ser acoplados ao corpo, como pulseiras, relógios ou jóias. Esse componentes poderiam comunicar-se com o microncontrolador, da mesma forma que os sensores normais.

Outra possibilidade de extensão do projeto é a utilização para outros tipos de monitoramento, não somente para idosos. O WHMS4 atualmente pode ser adaptado para monitorar qualquer tipo de pessoa, devido à generalização da forma do registro de acompanhamento. O sistema *web* poderá permitir diferentes customizações do *layout* e funcionalidades, como forma de atingir diferentes públicos.

Este trabalho demonstrou que é possível desenvolver projetos que sejam baseados em conceitos da Computação Ubíqua. A tecnologia atual ainda não apresenta a visão de futuro de acordo com as teorias de Weiser e outros pesquisadores, contudo, alguns elementos já estão sendo integrados rapidamente à nossas vidas.

X. TRABALHOS PUBLICADOS

A. Capítulos de Livros

BEZERRA, R. M. S. ; RIOS, T. S. . WHMS4 - Um Sistema de Monitoramento Remoto de Pacientes de Forma Integrada, Flexível, Escalável e com Baixo Custo. In:(Org.). XX CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN. Editora: Universidad Nacional de La Matanza. ISBN:978-987-3806-05-6.

B. Artigos Completos Publicados em Anais de Eventos

BEZERRA, R. M. S. ; RIOS, T. S. . WHMS4: Um Modelo Integrado para Monitoramento Remoto de Saúde. In: CISTI'2015 - 10a Conferência Ibérica de Sistemas e Tecnologias de Informação, 2015, Aveiro - Portugal. Anais da 10a Conferência 2015.

BEZERRA, R. M. S. ; RIOS, T. S. . WHMS4 - Um Sistema de Monitoramento Remoto de Pacientes de Forma Integrada, Flexível, Escalável e com Baixo Custo. In: XX CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN, 2014, Buenos Aires. XX CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN, 2014.

REFERÊNCIAS

- [1] A. Pentland, "Smart rooms, smart clothes," vol. 2, pp. 949–953 vol.2, Aug 1998.
- [2] D. Roggen, B. Arrich, and G. Tröster, "Life style management using wearable computer," 2006.
- [3] U. Anliker, J. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic, "Amon: a wearable multiparameter medical monitoring and alert system," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 4, pp. 415–427, Dec 2004.
- [4] Y.-C. Ma, Y.-P. Chao, and T.-Y. Tsai, "Smart-clothes - prototyping of a health monitoring platform," pp. 60–63, Sept 2013.
- [5] E. Chi, G. Borriello, G. Hunt, and N. Davies, "Guest editors' introduction: Pervasive computing in sports technologies," *Pervasive Computing, IEEE*, vol. 4, no. 3, pp. 22–25, July 2005.
- [6] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, Jul. 1999. [Online]. Available: <http://doi.acm.org/10.1145/329124.329126>
- [7] M. Weiser and J. S. Brown, "The coming age of calm technology," 1996.
- [8] Bluetooth, <http://www.bluetooth.com/>, Maio 2015.
- [9] Z. Alliance, "Wifi - ieee standards association - ieee 802.11: Wireless lans," <http://www.zigbee.org/>, Maio 2015.
- [10] ZigBee, <http://www.zigbee.org/>, Maio 2015.
- [11] K.-W. Park, B.-C. Kim, J.-W. Yoo, K.-H. Park, and K. Wohn, "Enabling sportive games with a do-it-yourself wearable computing platform," pp. 1–7, Dec 2010.
- [12] T. Martin and D. Siewiorek, "Wearable computers," *Potentials, IEEE*, vol. 13, no. 3, pp. 36–38, Aug 1994.
- [13] B. Rhodes, "The wearable remembrance agent: a system for augmented memory," in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, Oct 1997, pp. 123–128.
- [14] A. K. Dey and G. D. Abowd, "Towards a better understanding of context," vol. 22, 1999.
- [15] B. Schilit, N. Adams, and R. WALT, "Context-aware computing applications," pp. 85–90, 1994.
- [16] J. Krumm, Ed., *Ubiquitous Computing Fundamentals*. Boca Raton, FL: Chapman and Hall/CRC, 2010.
- [17] S. Mann, "Wearable computing," 2013. [Online]. Available: http://www.interaction-design.org/encyclopedia/wearable_computing.html
- [18] T. Starner, "The challenges of wearable computing: Part 1," *Micro, IEEE*, vol. 21, no. 4, pp. 44–52, Jul 2001.
- [19] A. Pantelopoulou and N. Bourbakis, "Prognosis - a wearable health-monitoring system for people at risk: Methodology and modeling," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 3, pp. 613–621, May 2010.
- [20] I. de Pesquisa Econômica Aplicada (IPEA), "Condições de funcionamento e infraestrutura das instituições de longa permanência para idosos no brasil," no. 93, p. 17, May 2011.
- [21] —, "71% dos municípios não têm instituições para idosos," http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=8574, May 2011.
- [22] A. Farjadian, M. Sivak, and C. Mavroidis, "Squid: Sensorized shirt with smartphone interface for exercise monitoring and home rehabilitation," pp. 1–6, June 2013.
- [23] E. Jovanov, A. Milenkovic, C. Otto, and P. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 1, p. 6, 2005. [Online]. Available: <http://www.jneuroengrehab.com/content/2/1/6>
- [24] Y. Lee, Y. Ju, C. Min, J. Yu, and J. Song, "Mobicon: Mobile context monitoring platform: Incorporating context-awareness to smartphone-centric personal sensor networks," pp. 109–111, June 2012.
- [25] Y.-D. Lee and W.-Y. Chung, "Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring," *Sensors and Actuators B: Chemical*, vol.

- 140, no. 2, pp. 390 – 395, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925400509003724>
- [26] K. Arai, “Wearable physical and psychological health monitoring system,” pp. 133–138, Oct 2013.
- [27] A. Huang, C. Chen, K. Bian, X. Duan, M. Chen, H. Gao, C. Meng, Q. Zheng, Y. Zhang, B. Jiao, and L. Xie, “We-care: An intelligent mobile telecardiology system to enable mhealth applications,” *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, no. 2, pp. 693–702, March 2014.
- [28] R. Jia and B. Liu, “Human daily activity recognition by fusing accelerometer and multi-lead ecg data,” pp. 1–4, Aug 2013.
- [29] J. Liu, F. Xie, Y. Zhou, Q. Zou, and J. Wu, “A wearable health monitoring system with multi-parameters,” pp. 332–336, Dec 2013.
- [30] A. Sugathan, G. Roy, G. Kirthyvijay, and J. Thomson, “Application of arduino based platform for wearable health monitoring system,” pp. 1–5, Dec 2013.
- [31] Arduino, “Arduino home page,” <http://www.arduino.cc>, May 2014.
- [32] M. Hanic, L. Sladek, F. Horinek, M. Jagelka, M. Donoval, M. Daricek, and D. Donoval, “Bio-monitoring system with conductive textile electrodes integrated into t-shirt,” pp. 1–4, April 2014.
- [33] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, “Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines,” *Computer Network*, vol. 54, no. 8, pp. 1245–1265, Jun. 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2010.03.005>
- [34] A. Burns, “Scheduling hard real-time systems: a review,” *Software Engineering Journal*, vol. 6, no. 3, pp. 116–128, May 1991.
- [35] I. Sommerville, *Engenharia de Software*, 8th ed. Addison-Wesley, 2007.
- [36] P. Koopman, “Embedded system design issues (the rest of the story),” in *Proceedings of the 1996 International Conference on Computer Design, VLSI in Computers and Processors*, ser. ICCD ’96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 310–. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645464.653663>
- [37] S. Eletronics, <http://www.sparkfun.com>, Maio 2014.
- [38] J. Meier, C. Farre, P. Bansode, S. Barber, and D. Rea, *Performance Testing Guidance for Web Applications: Patterns & Practices*. Redmond, WA, USA: Microsoft Press, 2007.