

DISSERTAÇÃO
apresentada a UTFPR
para obtenção do título de

MESTRE EM CIÊNCIAS

por

PAULO DÊNIS GARCEZ DA LUZ

**SISTEMA DE MONITORAMENTO EM TEMPO REAL DE
PACIENTES EM AMBIENTE HOSPITALAR UTILIZANDO
TRANSPORTE DE INFORMAÇÕES BIOLÓGICAS
ATRAVÉS DE REDES AD HOC**

Banca Examinadora:

Presidente e Orientador:

PROF. DR. PAULO JOSÉ ABATTI

UTFPR

Examinadores:

PROF. DR. GERSON LINCK BICHINHO

PUC-PR

PROF. DR. FÁBIO KURT SCHNEIDER

UTFPR

PROF. DR. LUIS NACAMURA JÚNIOR

UTFPR

Curitiba, novembro 2008.

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

PAULO DÊNIS GARCEZ DA LUZ

**SISTEMA DE MONITORAMENTO EM TEMPO REAL DE PACIENTES EM AMBIENTE
HOSPITALAR UTILIZANDO TRANSPORTE DE INFORMAÇÕES BIOLÓGICAS
ATRAVÉS DE REDES AD HOC**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de “Mestre em Ciências” – Área de Concentração: Engenharia Biomédica.

Orientador: Prof. Dr. Paulo José Abatti

Co-Orientador: Prof. Dr. Luis Nacamura Júnior

Curitiba

2008

AGRADECIMENTOS

A realização deste trabalho exigiu mais que minha vontade e empenho pessoal, sendo de fundamental importância o envolvimento de pessoas e entidades, a quem devo meus sinceros agradecimentos.

Ao professor Dr. Paulo José Abatti pela orientação incondicional e apoio oferecido no decorrer deste trabalho de pesquisa.

Ao professor Dr. Luis Nacamura Júnior pela compreensão e apoio desde o início do trabalho.

Aos colegas de laboratório por todos os momentos vividos ao longo da convivência cotidiana.

À UTFPR por disponibilizar sua estrutura e professores para a capacitação profissional de pesquisa científica, uma vez que esta instituição, além de me proporcionar a diplomação de mestre em ciências, já me concedeu o grau de engenheiro em eletrônica, antes mesmo de se tornar uma universidade.

Ao meu pai, Gualton Garcez da Luz, que me incentivou e ajudou nos momentos mais difíceis desta jornada.

Paulo Dênis Garcez da Luz
Setembro de 2008

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE TABELAS	xi
LISTA DE ABREVIATURAS E SIGLAS.....	xiii
RESUMO	xv
ABSTRACT	xvii
INTRODUÇÃO.....	1
1.1 ENGENHARIA BIOMÉDICA	1
1.2 MOTIVAÇÕES	2
1.3 OBJETIVOS	3
1.4 ESTRUTURA DA DISSERTAÇÃO.....	3
AMBIENTE HOSPITALAR E REDES DE SENSORES	5
2.1 INTRODUÇÃO	5
2.2 DESCRIÇÃO SUSCINTA DO HOSPITAL.....	6
2.2.1 PRONTO-SOCORRO – AMBULATÓRIO.....	8
2.2.2 CENTRO CIRÚRGICO	9
2.2.3 UNIDADE DE TERAPIA INTENSIVA (UTI)	9
2.2.4 CENTRO DE DIAGNÓSTICO.....	10
2.2.5 ENFERMARIA	10
2.2.6 OUTRAS ALAS HOSPITALARES	11
2.3 CORPO CLÍNICO	11
2.4 FLUXO DE INFORMAÇÕES NO HOSPITAL.....	11
2.5 INTERESSE HUMANO NA TEMPERATURA	12
2.5.1 TEMPERATURA NO ORGANISMO HUMANO.....	13
2.6 REDES SEM FIO.....	14

2.6.1 EXTENSÃO GEOGRÁFICA DAS REDES SEM FIO	15
2.6.2 FUNCIONAMENTO DAS WLANS	16
2.6.3 TECNOLOGIAS SEM FIO.....	17
2.6.4 REDES <i>AD HOC</i>	18
2.6.5 UTILIZAÇÃO DE REDES <i>AD HOC</i>	19
2.6.6 INFRA-ESTRUTURA DE REDES <i>AD HOC</i>	19
2.6.7 DESVANTAGENS DA UTILIZAÇÃO DE REDES <i>AD HOC</i>	20
2.7 CONCLUSÃO.....	21
DESCRIÇÃO DO SISTEMA.....	23
3.1 INTRODUÇÃO	23
3.2 APRESENTAÇÃO GERAL DO SISTEMA	24
3.2.1 NÓS MONITORES DA REDE <i>AD HOC</i>	26
3.2.2 CAMPO SENSOR.....	28
3.2.3 <i>GATEWAY / SINK</i>	28
3.2.4 SERVIDOR DE DADOS	29
3.3 FLUXO DAS INFORMAÇÕES.....	30
3.4 ESTRUTURA DO BANCO DE DADOS.....	33
3.5 CONFIGURANDO UMA REDE <i>AD HOC</i> NO <i>WINDOWS XP</i>	35
3.6 CONCLUSÃO.....	37
DESCRIÇÃO DOS TRANSDUTORES UTILIZADOS NO SISTEMA	38
4.1 INTRODUÇÃO	38
4.2 SENSOR DE TEMPERATURA.....	38
4.3 <i>HARDWARE</i> ASSOCIADO AO TRANSDUTOR DE TEMPERATURA.....	42
4.4 ADMINISTRAÇÃO DE SUBSTÂNCIAS INTRA-VENOSAS	44
4.5 CIRCUITO CONTA GOTAS.....	45

4.6 CONCLUSÃO.....	49
PROGRAMAS DO SISTEMA	50
5.1 INTRODUÇÃO	50
5.2 APLICATIVO DO HA.....	50
5.3 APLICATIVO DO HPT	51
5.4 APLICATIVOS NO SERVIDOR	53
5.5 CONCLUSÃO.....	62
DISCUSSÃO E CONCLUSÕES	63
6.1 ANÁLISE DOS RESULTADOS	63
6.2 CONCLUSÕES GERAIS	64
6.3 TRABALHOS FUTUROS	65
6.4 RESULTADOS OBTIDOS	66
REFERÊNCIAS BIBLIOGRÁFICAS	67
APÊNDICE 1 – ESQUEMÁTICO DO HA	73
APÊNDICE 2 – PROGRAMA DO HA	75
APÊNDICE 3 – PROGRAMA DO HPT	85
APÊNDICE 4 – PROGRAMAS DO SERVIDOR.....	95

LISTA DE FIGURAS

Figura 1: a) Termoscópio de Galileu. b) Esquema de termoscópio.	13
Figura 2: Rede WLAN típica.	17
Figura 3: Distribuição dos elementos do sistema.	25
Figura 4: Caminho do tráfego das informações e seus protocolos.	27
Figura 5: a) nós A e B separados sem comunicação e b) nós A e B se comunicando.	28
Figura 6: Protocolo de envio do nó sensor para o <i>hardware</i> , 4 bytes ASCII.	31
Figura 7: Protocolo de envio do <i>hardware</i> para o nó sensor, 10 bytes ASCII.	31
Figura 8: Caminho do tráfego das informações e seus protocolos em relação ao observador.	33
Figura 9: Trecho de código em XML.	33
Figura 10: Descrição da tabelas de dados.	34
Figura 11: Tela do <i>Add Profile</i>	36
Figura 12: Tela principal.	36
Figura 13: Faixas estimadas de temperaturas para pessoas normais. Na direita encontram-se os limites referentes a medidas feitas no reto, enquanto na esquerda estão as faixas de temperatura medidas oralmente.	39
Figura 14: Circuito completo do transdutor de temperatura.	43
Figura 15: a) Exemplo de um cateter e b) Exemplo de um equipo.	44
Figura 16: Primeiro protótipo construído para testes de aquisição da gota.	46
Figura 17: Resposta do circuito quando uma gota passa no sistema.	47
Figura 18: Circuito emissor e receptor de luz.	48
Figura 19: Conversor analógico-digital discreto para controle da luminosidade.	48
Figura 20: Comparador onde a tensão de referência é de 2 V para detecção da passagem da gota e de 3 V para calibração da tensão do LED.	49

Figura 21: Resposta em XML dos dados do paciente.....	52
Figura 22: Aplicativo do nó sensor.	53
Figura 23: Página index.html e orientador.html em seu frame.	56
Figura 24: Página de inclusão de novos pacientes no sistema.	58
Figura 25: Página de alteração de dados de um paciente.	59
Figura 26: Página de consulta ao histórico de um paciente.	60
Figura 27: Página de consulta ao para monitoramento em tempo real.	61
Figura 28: Página de monitoramento em tempo real.	62

LISTA DE TABELAS

Tabela 1: Comparativo entre algumas tecnologias <i>wireless</i>	18
Tabela 2: Comparativo entre transdutores de temperatura comerciais. Os gráficos mostrados passam uma idéia do comportamento da resposta típica de cada dispositivo (Adaptado de Agilent Technologies, 2000).	41
Tabela 3: Tensão por bits original	43

LISTA DE ABREVIATURAS E SIGLAS

AJAX	- <i>Asynchronous Javascript And XML</i>
ANVISA	- Agência Nacional de Vigilância Sanitária
API	- <i>Application Programming Interface</i>
APIDOM	- <i>API Document Object Model</i>
bps	- bits por segundo
CDMA	- <i>Code Division Multiple Access</i>
CSS	- <i>Cascading Style Sheets</i>
FDA	- <i>Food and Drug Administration</i>
GPRS	- <i>General Packet Radio Service</i>
GSM	- <i>Groupe Special Mobile</i>
HA	- <i>Hardware</i> de aquisição
HPT	- <i>Hardware</i> de Processamento e Transmissão
HTML	- <i>HyperText Markup Language</i>
HTTP	- <i>Hypertext Transfer Protocol</i>
IEEE	- <i>Institute of Electrical and Electronics Engineers</i>
LAN	- <i>Local Area Network</i>
LCD	- <i>Liquid Crystal Display</i>
LED	- Diodo Emissor de Luz
MAC	- <i>Media Access Control</i>
MAN	- <i>Metropolitan Area Network</i>
Mobile-Fi	- <i>Mobile Fidelity</i>
OMS	- Organização Mundial de Saúde
PC	- <i>Personal Computer</i>
PHP	- <i>Hypertext Preprocessor</i>

RTDs	- Resistance Temperature Detectors
SARP	- <i>Surgical Assistant Robot for Prostatectomies</i>
SQL	- <i>Structured Query Language</i>
TCP/IP	- <i>Transfer Control Protocol / Internet Protocol</i>
TDMA	- <i>Time Division Multiple Access</i>
URL	- <i>Uniform Resource Locator</i>
USB	- <i>Universal Serial Bus</i>
UTI	- Unidade de Terapia Intensiva
UWB	- <i>Ultrawideband IEEE 802.15.3</i>
WAN	- <i>World Area Network</i>
Wi-Fi	- <i>Wireless Fidelity</i>
WiMax	- <i>Worldwide Interoperability for Microwave Access</i>
WLAN	- <i>Wireless Local Area Network</i>
WMAN	- <i>Wireless Metropolitan Area Network</i>
WPAN	- <i>Wireless Personal Area Network</i>
WWAN	- <i>Wireless Wide Area Network</i>
XML	- <i>eXtensible Markup Language</i>
XSLT	- <i>eXtensible Stylesheet Language</i>

RESUMO

O presente trabalho apresenta o desenvolvimento de um sistema capaz de coletar parâmetros fisiológicos de um paciente, em tempo real, dentro de um ambiente hospitalar e enviá-los a um servidor na *internet*, preparado para responder a consultas que porventura venham a ser realizadas pelos médicos, enfermeiros ou quem de direito, respondendo os dados que estão sendo monitorados, assim como seus históricos.

O sistema foi subdividido em partes distintas para seu desenvolvimento. Primeiramente, foram desenvolvidos os transdutores necessários para realizar os testes de funcionamento do sistema. Foi então configurada uma rede sem fio *ad hoc*¹ para que os dados fossem transmitidos via rede sem fio até um *gateway*, onde seriam enviados ao servidor para seu armazenamento. Nesta etapa, foram desenvolvidos dois programas: um para coletar os dados do paciente e transmiti-los ao nó sensor e outro no nó sensor para transmiti-los ao servidor. Por fim, foram construídos vários programas no servidor para armazenar e exibir estes dados.

Após os testes realizados foi constatado que o sistema realizou todas as tarefas que foram propostas, apresentando os dados de um paciente a qualquer pessoa que queira acessar o sistema através da *internet*.

¹ A partir deste ponto todas as referências no texto sobre “redes sem fio *ad hoc*” serão simplificadas para “redes *ad hoc*”.

ABSTRACT

This work shows a development of a system capable of collecting physiological data from a patient, in real time, within a hospital and send them to an internet server prepared to answer a request that could be done by physicians, nurses or anyone with the right to do so, answering the items that have been tracked, as well as their historical data.

The system was divided in distinct parts for his development. First of all, transducers were developed to realize tests to verify if the system was working. After that, an ad hoc net was set up to transmit data, via wireless, to a gateway, where it would be sent to a server for storage. Then, two programs were developed: one, to collect data from the patient and deliver to the sensor node and another, inside the sensor node, to transmit them to the server. Eventually, many programs were constructed to store data and display them.

After the realization of the tests it was seen the system made all the proposed tasks, showing data from a patient to anyone who would wish to go into the system via internet.

CAPÍTULO 1

INTRODUÇÃO

1.1 ENGENHARIA BIOMÉDICA

Como em quase todos os campos da tecnologia, a aplicação da engenharia na medicina foi profundamente afetada pelos avanços das novas tecnologias de informação aliadas a evolução crescente da microeletrônica. Especificamente, a engenharia biomédica evoluiu com o desenvolvimento da instrumentação, sensores, próteses, dispositivos terapêuticos, e com a aplicação da teoria de sistemas e da física nos sistemas fisiológicos, incluindo o uso de modelos matemáticos (GESELOWITZ & GESELOWITZ, 1999).

As grandes mudanças na medicina ocorreram no final do século XIX e início do século XX, com o rápido desenvolvimento das ciências aplicadas, tais como química, física, engenharia, microbiologia, fisiologia, entre outras. Este processo de evolução foi caracterizado pelo intenso intercâmbio entre disciplinas, que possibilitou que a pesquisa médica tirasse proveito da tecnologia existente e desenvolvesse técnicas para diagnóstico e tratamento de doenças. Por exemplo, em 1903, o fisiologista holandês Willem Einthoven construiu o primeiro eletrocardiógrafo prático para registrar a atividade elétrica do coração (BRONZINO, 2000).

A aplicação de descobertas científicas para a análise do processo biológico deu origem a uma nova era na medicina, introduzindo técnicas e dispositivos para medição de parâmetros biológicos. Estudos realizados por Luigi Galvani, em 1780, sobre a eletricidade nos animais deram origem à eletrofisiologia. Em 1888, Augustus Desiré Waller mostrou que, com um eletrômetro capilar, era possível registrar tensões elétricas na superfície do coração. O modelo matemático proposto por William Thomson (depois por Lord Kelvin), em 1855, para o cabo telegráfico subatlântico foi modificado na virada do século XIX para descrever o mecanismo de condução ao longo de uma fibra nervosa. Em 1895, Wilhelm Röntgen inventou a aquisição de imagens por raio-X, representando um grande impacto na medicina. No ano seguinte as empresas Siemens e General Electric iniciaram a fabricação dos primeiros equipamentos de raio-X para uso médico. Estes são alguns dos exemplos da colaboração científica entre a física / engenharia e a medicina (NEBEKER, 2002).

Uma das ferramentas mais importantes para a pesquisa biomédica desenvolvida no século XX foi o microscópio eletrônico. Esse equipamento de análise foi inventado no começo dos anos 30 por dois grupos de pesquisa alemães, um deles pertencentes à universidade de Berlim e dirigido por Hans Hermann Knoll e Ernest Ruska, sendo que o segundo grupo estava situado na Siemens-Schuckert e dirigido por Reinhold Rüdenberg (NEBEKER, 2002).

A robótica teve suas primeiras aplicações na engenharia biomédica no início da década de 90. Em cirurgias, principalmente neurológicas, existem robôs que auxiliam os cirurgiões ou mesmo efetuam a cirurgia de forma autônoma, mediante programação prévia. Em 25 de março de 1991 no hospital de Shaftesbury, em Londres, o robô cirúrgico SARP (*Surgical Assistant Robot for Prostatectomies*), controlado por computador, desempenhou uma operação cirúrgica pela primeira vez. Em 1993 o FDA (*Food and Drug Administration*) autorizou testes clínicos com robôs em procedimentos cirúrgicos para implante artificial de bacia (NEBEKER, 2002).

Nas últimas três décadas a engenharia biomédica se consolidou como uma área de conhecimento abrangente e indispensável para a evolução contínua da medicina e da pesquisa neste ramo. Atualmente, a engenharia biomédica é definida como o uso de princípios e técnicas de engenharia para resolver problemas nas áreas de biologia e medicina, provendo meios para melhorar o diagnóstico e o tratamento médico (NEBEKER, 2002).

1.2 MOTIVAÇÕES

Dentre as diversas possibilidades, das buscas pelo desenvolvimento de sistemas de monitoração de parâmetros que são considerados essenciais para a determinação do quadro clínico do paciente destacam-se sistemas que sejam seguros, confiáveis e que sejam viáveis financeiramente.

Neste sentido, este trabalho visa o desenvolvimento de um sistema que extrapola a engenharia biomédica usual, pois além de sensores de parâmetros fisiológicos, o sistema utiliza a instrumentação biomédica clássica aliando sua integração com redes monitoras *ad hoc* e *internet*, visando assim uma ampliação e extensão da leitura local dos parâmetros fisiológicos, podendo estes serem visualizados remotamente e em tempo real.

1.3 OBJETIVOS

O objetivo geral deste trabalho é o desenvolvimento de uma rede de sensores sem fio (*wireless*), indicadora de parâmetros fisiológicos ou patológicos de tal forma que estes possam ser disponibilizados na rede mundial de computadores *internet* servindo de apoio à decisão do corpo clínico dos hospitais mesmo que o médico envolvido encontre-se à distância.

O sistema deverá ser capaz de permitir consultas em tempo real ou ao histórico do paciente, mantendo um monitoramento constante para que o corpo clínico possa sempre que necessário acessar o estado atual do paciente, podendo também ser implementado um sistema de alarmes para sinalizar situações críticas.

Desta forma, o trabalho buscará auxiliar a decisão medicina na monitoração da prescrição e diagnóstico, aliando tecnologia e confiabilidade, além de, agregar maior conforto e segurança ao paciente, diminuindo o tempo de permanência no leito hospitalar.

Impactos futuros do projeto:

- 1) A possibilidade de o paciente ser acompanhado remotamente no hospital ou em casa (*gateway* residencial).
- 2) Confiabilidade dos dados obtidos através dos sensores anexados aos pacientes.
- 3) Diminuição do tempo de permanência do paciente em observação nas UTI'S (Unidade de Terapia Intensiva).
- 4) Diminuição do tempo de permanência do paciente no hospital.
- 5) O estabelecimento do processo contínuo de monitoramento dos pacientes (tempo real).
- 6) Melhorar a integração de recursos humanos especializados (equipe médica) com o paciente.

1.4 ESTRUTURA DA DISSERTAÇÃO

O trabalho descrito é apresentado em seis capítulos e quatro apêndices. O segundo capítulo, sob o título **ambiente hospitalar e redes de sensores**, aborda a estrutura física do ambiente hospitalar, onde o sistema irá ser implantado, ressaltando alguns pontos da estrutura hospitalar que justificam uma melhoria na assistência prestada.

Aparentemente a estrutura de um hospital é propícia para a instalação de uma rede de sensores. O hospital serve como um campo sensor onde os pacientes são os nós da rede. A rede monitora *ad hoc* será utilizada como infra-estrutura da rede de sensores; esta estrutura é necessária devido à locomoção dos pacientes dentro do ambiente hospitalar.

O terceiro capítulo, sob o título **descrição do sistema**, aborda todos os pontos referentes ao funcionamento do sistema, detalhando todos os aspectos que são necessários para que a informação monitorada em um paciente possa “locomover-se” dentro das diferentes partes do sistema, passando pela rede monitora *ad hoc*, atravessando o *gateway*, mudando de infra-estrutura para que a partir deste ponto passe à *internet* e chegue ao servidor, onde fica a base de dados do sistema. No mesmo servidor existem programas que disponibilizam as informações coletadas de duas formas: em tempo real ou em forma de histórico.

O quarto capítulo, sob o título **descrição dos sensores**, aborda a concepção dos dois sensores que foram utilizados para a aquisição dos dados que serão monitorados no paciente. Eles são a temperatura e o controle de administração de substâncias intravenosas no paciente, através da contagem de gotas dentro do equipo do paciente. Todos os detalhes elétricos e eletrônicos relevantes são apresentados, como também suas respectivas soluções para o funcionamento dos sensores.

O quinto capítulo, sob o título **programas do sistema**, aborda todas as necessidades em termos de programas que precisaram ser escritos para conectar todos os sub-sistemas e gerar um fluxo contínuo de informações desde de sua obtenção até o seu armazenamento no servidor de dados. Apresenta também as interfaces gráficas de alguns programas além de suas funcionalidades.

O sexto capítulo, sob o título **discussão e conclusões**, apresenta as conclusões obtidas a partir da construção e dos teste do sistema, pondera sobre seus diversos aspectos, sugere modificações futuras e discute novas idéias e implementações.

CAPÍTULO 2

AMBIENTE HOSPITALAR E REDES DE SENSORES

2.1 INTRODUÇÃO

A complexidade de um ambiente hospitalar se estende desde a administração de sua infra-estrutura até a administração do quadro de funcionários. É uma estrutura que requer um enorme esforço para se manter funcionando, com um intuito muito nobre, zelar por vidas humanas (SILVA, 1997).

Durante sua história, os hospitais transformaram-se de locais destinados ao abrigo e atendimento essencialmente espiritual de pobres, velhos e desabrigados a instituições de cura, formação e transmissão de saber (FOUCAULT, 1996).

Com as transformações ocorridas ao longo de sua trajetória, o hospital contemporâneo não é apenas uma instituição que evoluiu, mas uma nova instituição que assumiu outras características, ainda que resguardando algumas que as precederam. Ocorreram mudanças em suas características, objetivos e finalidades, sua administração, seus indivíduos, instrumentos e processos de trabalho do seu cotidiano (RIBEIRO, 1993).

Os avanços científicos e tecnológicos conjuntamente a diversidade e complexidade dos processos de trabalho existentes em um hospital, traduzem-se em um ambiente potencialmente de risco, de diversos tipos, que repercutem sobre os indivíduos, trabalhadores, pacientes e seus familiares, e que pode também, em determinadas situações, vir a afetar o meio ambiente (SILVA, 1997).

As atividades dos profissionais do setor saúde, em especial, nas unidades de internação, envolvem um contato permanente com o sofrimento humano (dor e morte) e exposição aos riscos ocupacionais de natureza física, química, biológica, ergonômica e de acidente existentes nos processos e organização do trabalho (SILVA, 1997).

As precárias condições de instalações e equipamentos, deficiência quantitativa e qualitativa de pessoal e material nos hospitais do Brasil, resultam em sobrecarga física, cognitiva e psíquica com graves repercussões sobre a saúde (BULHÕES, 1994).

Um hospital necessita de muitos recursos e uma estrutura funcional para realizar suas tarefas com o máximo de êxito e o mínimo de risco possível. Examinando essa estrutura

pode-se notar que existem inúmeras lacunas a serem aprimoradas dentro da estrutura hospitalar.

2.2 DESCRIÇÃO SUSCINTA DO HOSPITAL

O hospital é conceituado como instituição prestadora de serviços pelo Ministério da Saúde (BRASIL, 1978, p. 27), sendo "parte integrante de uma organização médica e social, cuja função básica consiste em proporcionar à população assistência médica integral, curativa e preventiva, sob quaisquer regimes de atendimento, inclusive o domiciliar, constituindo-se também em centro de educação, capacitação de recursos humanos e de pesquisas em saúde, bem como de encaminhamento de pacientes, cabendo-lhe supervisionar e orientar os estabelecimentos de saúde a ele vinculados tecnicamente". Este conceito é fundamentado no conceito formulado pela Comissão de Peritos em Assistência Médica da OMS (Organização Mundial de Saúde), em reunião realizada em Genebra, no período de 18 a 23 de junho de 1956.

Uma vez conceituado, pode-se afirmar que, nos dias de hoje, é inadmissível aceitar a visão simplista do hospital como sendo o local para diagnóstico e tratamento da doença. Entende-se que o hospital deve ser encarado como uma instituição dotada de estrutura física, equipamentos e organização adequados à recepção de clientes em regime de internação, para o seu tratamento, a fim de devolvê-los à comunidade em condições satisfatórias de saúde (MIRSHAWKA, 1994).

No Brasil, o órgão regulamentador e fiscalizador ANVISA (Agência Nacional de Vigilância Sanitária) define que uma empresa na área de saúde pode ser considerada um hospital se possuir uma estrutura que envolva os setores listados a seguir.

- **Departamento Médico.**

- Corpo necessário de funcionários para desempenhar todas as necessidades referentes ao cotidiano do ambiente hospitalar.

- **Departamento de suporte Técnico.**

- Central de esterilização.
- Central de manipulação intravenosa.
- Farmácia.
- Nutrição.

- Serviços de arquivo médico.
- **Departamento de Enfermagem.**
 - Alas de internamento
 - Ambulatório
 - Centro cirúrgico.
 - Unidade de terapia intensiva.
- **Departamento de suporte operacional.**
 - Hotelaria.
 - Atendimento.
- **Departamento de serviço auxiliar de diagnóstico terapêutico.**
 - Análises clínicas.
 - Anatomia patológica.
 - Banco de sangue.
 - Diagnóstico por imagem.
 - Fisioterapia.
 - Fonoaudiologia.
 - Psicologia clínica.
 - Quimioterapia.
 - Radioterapia.
- **Departamento de faturamento**
 - Convênios.
 - SUS (Sistema único de saúde).
- **Coordenação de divisão técnica.**
- **Coordenação de divisão de administração.**
- **Coordenação de divisão de corpo clínico.**
- **Coordenação de divisão de enfermagem.**
- **Comissão de nutrição enteral.**
- **Comissão de registro de câncer.**
- **Comissão de humanização.**
- **Comissão de farmácia terapêutica.**
- **Comissão de controle de infecção hospitalar.**
- **Comissão de revisão de prontuários.**
- **Comissão de revisão de óbitos.**
- **Comissão de ética.**

- **Comissão de gerenciamento de resíduos de serviços de saúde.**
- **Núcleo da qualidade.**
- **Núcleo da auditoria de contas.**
- **Núcleo das informações executivas.**

O hospital é uma empresa que funciona vinte e quatro horas por dia durante os trezentos e sessenta e cinco dias do ano, possuindo todos os problemas de uma grande empresa, acrescido da função básica da restauração da saúde e da prevenção de doenças. Como se pode constatar, de todas as empresas modernas, nenhuma parece ser mais complexa do que o hospital e, como não poderia ser diferente, exige planejamento e gestão eficiente para o seu bom funcionamento. O sucesso do hospital está baseado numa tríade: bom planejamento comunitário, bom projeto de construção e boa administração. A falta de um item dessa combinação, ou a má gerência de algum deles resulta em um hospital inadequado à boa recuperação dos pacientes. Na rotina dos hospitais, na área da prática assistencial ou na área da administração, seja no ensino ou na pesquisa, existe uma forte convivência com problemas de diferentes origens e diferentes efeitos colaterais. Um bom ambiente hospitalar requer sempre atenção especial para tais efeitos, na busca contínua de suas soluções. Estudos desenvolvidos em hospitais mostram uma preocupação cada vez maior com a assistência prestada ao cliente, com as novas formas de abordagem no cuidado e no tratamento, bem como, com o avanço tecnológico para o diagnóstico e terapia (SILVA, 1997).

Do ponto de vista operacional o hospital é dividido em departamentos ou alas.

2.2.1 PRONTO-SOCORRO – AMBULATÓRIO

Uma definição popular de pronto-socorro diz que aquele é o lugar em que o paciente pode ir sem marcar hora. (NUNES, 2003)

O pronto-socorro é a ala do hospital que está preparada para o tratamento imediato e provisório para os casos de urgência ou emergência a qualquer hora do dia ou da noite, dado em casos de acidentes ou enfermidades imprevistas. Geralmente um socorro prévio é prestado no local do acidente, até que se possa colocar o paciente a cargo de um médico para o tratamento, este primeiro contato com o hospital se dá através do ambulatório ou pronto-socorro.

2.2.2 CENTRO CIRÚRGICO

Centro Cirúrgico é um lugar especial dentro do hospital, convenientemente preparado segundo um conjunto de requisitos que o tornam apto à prática da cirurgia. O centro cirúrgico é uma ala do hospital onde se realizam intervenções micro-cirúrgicas ou cirúrgicas, visando atender a resolução de intercorrências, por meio da ação de uma equipe integrada (BENDIN, RIBEIRO, BARRETO 2004). Nele são realizadas técnicas estéreis para garantir a segurança do cliente quanto ao controle de infecção.

Neste local o acesso ao público é limitado, ficando restrito a circulação dos profissionais que lá atuam. Para efeito de controle asséptico, o Centro Cirúrgico divide-se em: **área irrestrita** onde profissionais podem circular livremente usando roupas próprias (secretaria, vestiário e corredor de entrada), **área semi-restrita** aquela que permite a circulação limitada de pessoal de modo a não intervir nas rotinas de controle e manutenção da assepsia e **área restrita** onde só é possível circular usando roupa própria do centro cirúrgico, inclusive máscaras e gorros conforme normas da unidade e onde as técnicas assépticas devem ser utilizadas de maneira rigorosa, a fim de diminuir os riscos de infecção.

2.2.3 UNIDADE DE TERAPIA INTENSIVA (UTI)

Esta ala do hospital é dotada de sistemas de monitoração contínua e é utilizada somente para casos considerados críticos onde, geralmente, há risco de morte do paciente. Ficam na UTI os pacientes que se submetem a procedimentos cirúrgicos de alta complexidade ou vítimas de situações críticas de emergência. Ainda é função da UTI amenizar o sofrimento, tais como sintomas de dor e falta de ar independente do prognóstico. Essas unidades nasceram da necessidade de oferecer um suporte de vida avançado para pacientes com problemas agudos. Em resumo, a UTI é uma unidade hospitalar para o tratamento clínico, tanto pré-operatório quanto pós-operatório, de pacientes que necessitam de vigilância e monitoração ininterrupta por uma equipe devidamente treinada. Equipamentos de alta tecnologia são utilizados no acompanhamento de pacientes com diferentes doenças, oferecendo um tratamento especial de acordo com as necessidades de cada um.

2.2.4 CENTRO DE DIAGNÓSTICO

Esta ala do hospital é onde se encontram a maioria de aparelhos de diagnósticos do hospital. Dotada de diversos sistemas para a execução de exames diversos, pode gerar a maioria de exames que um médico possa requisitar com rapidez e comodidade para o paciente. Esta ala surgiu da necessidade de agilidade perante aos exames, pois caso os exames sejam feitos em laboratórios externos, pode ocorrer demora no diagnóstico, prejudicando o tratamento do paciente.

2.2.5 ENFERMARIA

Esta ala é responsável pelo acompanhamento dos pacientes após alguma intervenção médica que por isso necessita de cuidados especiais. O pessoal da enfermaria é responsável pelo monitoramento de todos os pacientes que estão internados nos leitos do hospital. Faz parte do trabalho da equipe de enfermagem o acompanhamento de todos os sinais vitais do paciente, sendo eles a pressão, o batimento cardíaco, a temperatura, administração de substâncias intravenosas e ainda, quando necessário, ajudar os pacientes que necessitam de assistência locomotiva ou que por ventura sofram algum mal súbito e peçam auxílio.

Através do prontuário, o médico indica ao corpo de enfermagem as necessidades do paciente, descrevendo o que o enfermeiro necessita fazer, especificando inclusive o horário destas atividades.

O corpo de enfermagem também é responsável por comunicar ao médico o estado do paciente, sendo que qualquer anomalia no tratamento deve ser reportada com urgência, a fim de que ele possa intervir pronta e adequadamente.

Ressalta-se que em geral esta ala é extremamente dependente dos relatos do corpo de enfermagem, e ainda (em geral) possui poucos recursos tecnológicos a sua disposição.

2.2.6 OUTRAS ALAS HOSPITALARES

Existem muitas outras alas dentro de um hospital, como por exemplo a ala de hemoterapia, nutrição entre outras. A configuração de um hospital não é fixa, cada hospital pode possuir alas distintas. Ressalta-se que independentemente disto todos têm como prerrogativa zelar por vidas humanas e executando os devidos procedimentos clínicos.

2.3 CORPO CLÍNICO

Para manter toda esta estrutura operacional em termos de recursos humanos é necessário um grande corpo clínico, composto por profissionais das mais diferentes especialidades médicas.

Sua função é de atender ao paciente de tal forma que (dentro das disponibilidades técnicas) o mesmo tenha sua saúde restabelecida. Para isto, muitas vezes, é necessário que este paciente seja internado (algumas vezes na UTI) para que possa usufruir das facilidades de um acompanhamento mais amigável, incluindo o uso de equipamentos especializados. Por exemplo, um paciente pode, após um procedimento cirúrgico de emergência, passar 24 horas na UTI, período depois do qual fica internado por cinco dias em um quarto. Observe que neste último período, o paciente, além de ser regularmente monitorado pelo corpo de enfermagem e ter frequentes visitas médicas, o mesmo pode ser deslocado, por exemplo, ao centro de raio-X por exemplo para ser radiografado.

2.4 FLUXO DE INFORMAÇÕES NO HOSPITAL

A informação sobre um paciente em internamento deveria chegar ao médico com rapidez e com alto grau de confiabilidade. Porém, devido ao método utilizado em muitos hospitais que se resume ao prontuário do paciente, que é atualizado pelos enfermeiros, estes dados acabando sendo repassados aos médicos com um atraso considerável quando comparados a um sistema eletrônico de monitoramento. Neste processo, particularmente na troca de turnos dos enfermeiros, aumenta a possibilidade de algumas leituras acabarem não

sendo realizadas, gerando assim um mecanismo de fluxo de informações com ligeiras falhas (SILVA, 1999).

As informações que comumente são monitoradas pelo corpo clínico do hospital quando um paciente encontra-se internado são: a temperatura, o batimento cardíaco, a pressão e o controle da administração de substâncias intravenosas.

2.5 INTERESSE HUMANO NA TEMPERATURA

Os registros históricos existentes situam a primeira tentativa de estabelecer uma escala de temperaturas entre os anos 130 d.C. e 200 d.C., em que Galeno, médico grego cujos ensinamentos constituíram a base da prática clínica até ao século XVII, sugeriu que as sensações de quente e frio fossem medidas com base numa escala com quatro divisões numeradas acima e abaixo de um ponto neutro. Tal escala termométrica atribuiu a temperatura de 4° de calor para a água em ebulição, a temperatura de 4° de frio ao gelo e a temperatura neutra à mistura de quantidades iguais daquelas duas substâncias.

Cerca de 1300 anos mais tarde, o médico Harme de Berna criou uma escala de temperaturas baseadas nas latitudes terrestres, atribuindo 4° graus de frio aos pólos e 4° graus de calor ao equador. De acordo com essa escala, seria possível estabelecer a mistura ideal de drogas a ministrar em pacientes com enfermidades, conforme o local da terra em que o paciente se encontrasse sendo, a dose nas regiões mais frias (WANG, FELTON, 1983).

Galileu Galilei é geralmente tido como o inventor do primeiro termômetro, denominado termoscópio, sendo o pioneiro em quantificar a temperatura através de um aparelho. Em 1592, suspendendo sobre um reservatório aberto, cheio de álcool colorido, um tubo estreito de vidro, com sua extremidade em forma de esfera, verificou que o ar dentro da esfera quando aquecido expandia-se e borbulhava através do álcool. Esfriando a esfera, o líquido penetrava no interior do tubo. As flutuações da temperatura da esfera podiam assim ser observadas, anotando a posição do líquido dentro do tubo. Embora atuasse com funcionamento simples para medição de temperatura, o equipamento sofria efeitos da altitude em relação ao nível do mar (BENEDICT, 1984).

Na Figura 1a é apresentada uma réplica genérica de um termoscópio de Galileu, e na figura 1b, tem-se o diagrama esquemático simplificado de seu funcionamento, onde a coluna de álcool é movimentada para cima ou para baixo conforme a temperatura se altere na porção superior do dispositivo.

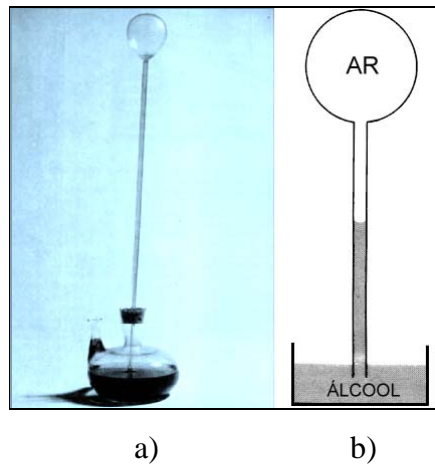


Figura 1: a) Termoscópio de Galileu. b) Esquema de termoscópio.

Um avanço na padronização da temperatura foi estabelecido por Gabriel Fahrenheit aproximadamente em 1714, quando estabeleceu uma escala para medir a temperatura que iniciava em 0° e chegava a 212° , sendo que este limite superior teria correspondência térmica equivalente a temperatura de ebulição da água. Em meados de 1742, Anders Celsius, sugeriu uma escala própria que foi adaptada posteriormente para uma escala proporcional ao aumento de temperatura (RING, 1998). Em torno de 1865, Carl Wunderlich, começou a estudar a correlação da temperatura corpórea com enfermidades que causavam sua variação nos enfermos (RING, 1998). A partir desse ponto foram surgindo métodos mais sofisticados para a avaliação de temperatura.

2.5.1 TEMPERATURA NO ORGANISMO HUMANO

Em geral, a sobrevivência de formas de vida está restrita a faixas de temperatura, quanto mais complexo o organismo mais dependente da temperatura ele será e, conseqüentemente, menor será a faixa de temperatura tolerável para que o organismo se mantenha em condições adequadas de funcionamento. Enquanto que, para organismos mais simples a tolerância a variação de temperatura aumenta, pois existe uma quantidade muito menor de sistemas para manter o organismo em pleno funcionamento.

Existem dois tipos de temperatura corpórea. A primeira é denominada temperatura basal, referente ao interior do corpo, que se mantém praticamente constante, sofrendo pequenas variações ao longo do dia. A temperatura dos tecidos subcutâneos e por fim da pele, constitui o segundo tipo, denominado temperatura superficial, a qual pode variar acentuadamente devido às alterações da temperatura do ambiente. Não existe um valor exato para a temperatura corporal em condições normais, uma vez que mensurações realizadas em diversas pessoas mostraram leves variações da temperatura basal. Contudo, pode-se assumir uma faixa de temperatura considerada normal, que vai de aproximadamente 36°C a um pouco mais de 37,2 °C. Em geral, a temperatura normal média, quando medida oralmente, é considerada sendo 37 °C e em torno de 37,5 °C para medidas efetuadas no reto (GUYTON, 1986).

Quando organismos homeotérmicos apresentam diferença em sua temperatura corporal padrão, tem-se como reflexo imediato a suspeita de alguma infecção bacteriana ou de alguma virose espalhando-se pelo organismo. A alteração da temperatura corpórea é tida como um importante parâmetro para diagnósticos clínicos e para acompanhamento de casos que requerem atenção médica, como por exemplo, quadros pós-operatórios (BERTOTTI, 2005).

2.6 REDES SEM FIO

“As pessoas se movem. As redes não! Mais do que qualquer outra coisa, estas duas declarações podem explicar a explosão das redes *wireless*” (GAST, 2002).

A palavra *wireless*, na sua tradução literal, significa “sem cabos” e representa um tipo de rede de computadores que, ao invés de utilizar cabos metálicos para sua comunicação, faz uso das ondas eletromagnéticas – em especial as ondas de rádio. A norma que regulamenta este tipo de conexão é fornecida pelo IEEE (*Institute of Electrical and Electronics Engineers*) sob o número 802.11.

Em sua obra “802.11 *wireless networks: the definitive guide*”, Matthew S. Gast (2002) especifica quatro critérios que resultam na grande aceitação e conseqüentemente na expansão das redes sem fio, são eles:

- mobilidade - os usuários tendem a se mover enquanto os dados ficam, tipicamente, armazenados centralmente, sendo que a possibilidade dos usuários

acessarem dados enquanto se movem pode ser traduzida como ganho de produtividade;

- facilidade e rapidez de instalação - em muitos lugares a instalação das tradicionais redes de computadores é bastante complicada, por exemplo, velhas construções são um problema freqüente, uma vez que o processo de passar a fiação das redes por suas estruturas, invariavelmente, representa um desafio e mesmo nas estruturas modernas as instalações de cabos podem ser demasiada caras e demandantes de tempo significativo;
- flexibilidade - as redes sem fio permitem que os usuários se movimentem com tamanha flexibilidade que, em poucos instantes, estes podem transformar suas estações de trabalho em nós de uma sala de reuniões, em pontos de acesso de teleconferência ou, até mesmo, nas tradicionais “estruturas-cubículo” tão usadas nos escritórios;
- custo - em alguns casos as conexões sem fio estão se mostrando mais baratas do que as redes tradicionais, por exemplo, uma rede que tenha por objetivo interligar dois prédios, onde inicialmente, os custos para a instalação da rede sem fio seriam mais dispendiosos do que aqueles destinados à instalações de redes convencionais, “com o passar do tempo, os links sem fio ponto-a-ponto custariam mais barato do que alugar capacidades das companhias telefônicas”.

2.6.1 EXTENSÃO GEOGRÁFICA DAS REDES SEM FIO

Quando se trata das redes sem fio, altera-se esta nomenclatura e suas representações que são usadas para uma rede convencional, haja vista a grande flexibilidade das redes sem fio que ignoram as tradicionais concepções que se valem de figuras arquitetônicas LAN (*Local Area Network*), MAN (*Metropolitan Area Network*), WAN (*World Area Network*), etc.). Segundo Campos (2006, p.15), é possível destacar a existência de quatro grandes grupos:

- WPAN (*Wireless Personal Area Network*) - concentra as tecnologias *wireless* de pequeno alcance (entre 10 e 100 metros) e de baixa taxa de transmissão, configurando-se como um padrão de redes que interligam dispositivos pessoais ou redes de sensores sem fio, sendo definida pela norma 802.15 do IEEE;

- WLAN (*Wireless Local Area Network*) - designam a interligação de redes locais com alcance entre 100 e 300 metros, configurando-se como um padrão implementado como extensão ou alternativa para as redes com cabeamento convencional (par metálico ou fibra óptica);
- WMAN (*Wireless Metropolitan Area Network*) - este grupo concentra as tecnologias que tratam dos acessos de banda larga para redes em áreas metropolitanas, sendo seu alcance médio de 6 km;
- WWAN (*Wireless Wide Area Network*) - aqui as tecnologias estão voltadas para redes de longa distância em telecomunicações e atendem aos serviços de voz e alguns serviços de dados.

2.6.2 FUNCIONAMENTO DAS WLANS

De acordo com o que será necessário para o desenvolvimento do sistema, neste trabalho foi utilizado o conceito de redes WLAN's.

Primeiramente, as WLAN's constituem a comunicação de dados entre os pontos da rede utilizando portadoras de rádio ou comunicação infravermelha nos quais os dados são modulados.

É importante ressaltar que múltiplas portadoras de rádio podem coexistir num mesmo meio, sem que isto resulte em interferência de uma sobre a outra. Para captar o sinal e extrair os dados, um receptor será sintonizado em uma frequência específica, rejeitando as outras portadoras que apresentem frequências diferentes.

Uma configuração WLAN típica é mostrada na figura 2. Nela o dispositivo *transceptor* (transmissor/receptor), também chamado de acesso (*access point*), será conectado a uma rede local *ethernet* convencional (com cabos). Os *access points* não executam simplesmente a comunicação com a rede convencional, mas também intermedeiam o tráfego de dados com os *access points* vizinhos, utilizando uma configuração de células com *roamings* semelhantes a um sistema de telefonia celular (SILVA, 1998).

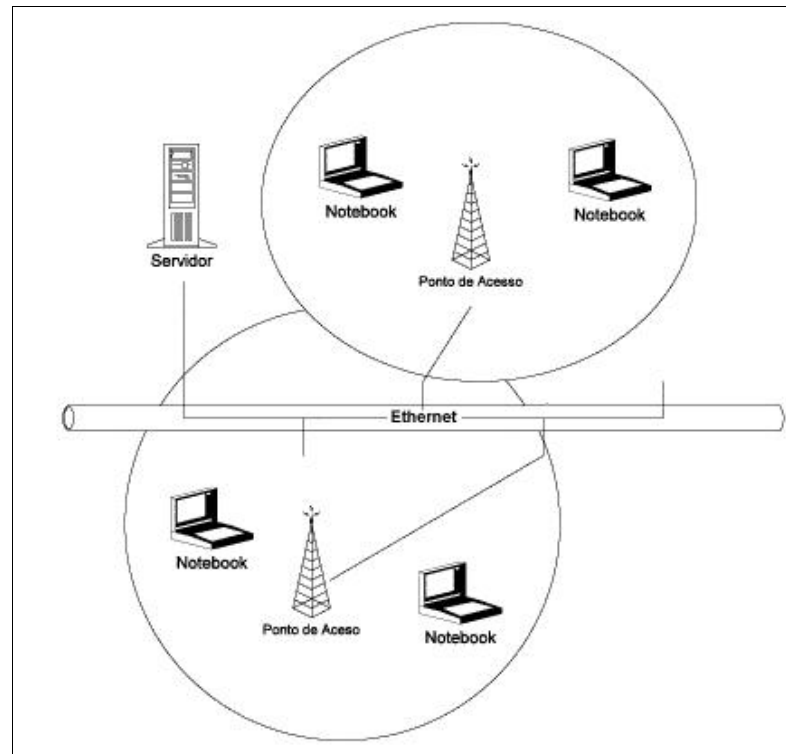


Figura 2: Rede WLAN típica.

2.6.3 TECNOLOGIAS SEM FIO

Várias tecnologias vêm sendo desenvolvidas para realizar as conexões nas redes sem fio. Elas diferem em vários aspectos, desde a distância que conseguem cobrir até a taxa de dados que conseguem transmitir. Nas redes WPAN a tecnologia mais usada é o *Bluetooth* que atende à norma IEEE 802.15.1, porém as tecnologias *ZigBee* (IEEE 802.15.4) e UWB (*ultrawideband* IEEE 802.15.3) também têm sido utilizadas. Nas WLAN's as atenções estão todas voltadas para o padrão *Wi-fi* (*wireless fidelity*), regulamentado pela IEEE 802.11b. As WWAN's contam com os sistemas usados pela telefonia celular: o TDMA (*Time Division Multiple Access* 9,6 Kbps) e o CDMA (*Code Division Multiple Access* 14,4 Kbps). Contudo as WWAN's estão migrando para tecnologias mais modernas como o GSM/GPRS (*Groupe Special Mobile/General Packet Radio Service*). Nas redes WMAN uma recente tecnologia como o nome de *WiMax* (*Worldwide Interoperability for Microwave Access* IEEE 802.16) alcança a taxa de transmissão de 75 Mbps cobrindo um raio de 50 km. Há, no entanto, outros padrões, tais como o *Mobile-Fi* (*Mobile Fidelity* IEEE 802.20). A tabela 1 apresenta de forma comparativa algumas tecnologias sem fio (CAMPOS, 2006).

Tabela 1: Comparativo entre algumas tecnologias sem fio.

	HOMERF	BLUETOOTH (802.15.1)	ZIGBEE (802.15.4)	WI-FI (802.11B)
APLICAÇÕES	REDES RESIDENCIAS	CONECTIVIDADE PESSOAL	RSSS	REDES COMPUTADORES
CAMADA FÍSICA	FHSS	FHSS	DSSS	DSSS
DISTÂNCIA	50 m	10 m	100 m	100 m
FREQUÊNCIA	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz
TOPOLOGIA	PONTO A PONTO	PONTO - MULTIPONTO	PONTO - MULTIPONTO	PONTO - MULTIPONTO
MAX. POTÊNCIA TRANSMITIDA	100 mW	(1-100) Mw	(1-100) Mw	100 mW
TAXA DE DADOS	2 Mbps	1 Mbps	250 Kbps	11 Mbps
TIPOS DE DADOS	VOZ E DADOS	VOZ E DADOS	DADOS	DADOS
TIPOS DE REDES	WLAN	WPAN	CR-WPAN	WLAN
CUSTO	US\$ 100 POR COMPUTADOR	US\$ 5 POR MÓDULO	US\$ 3 POR MÓDULO	US\$ 100 POR MÓDULO

2.6.4 REDES *AD HOC*

As redes *ad hoc* são sistemas telemáticos que não possuem infra-estrutura. Uma característica fundamental destas redes é o fato de que os nós que a compõem são capazes de se comunicar uns com os outros funcionando, eles mesmos, como roteadores (YI, NALDURG e KRAVETS, 2001).

Alguns nós desta rede podem ser contatados por outro nó dispensando a necessidade de outros intermediários para realizar o “roteamento” dos pacotes transmitidos. Estes nós são conhecidos como “nós vizinhos” ao nó transmissor. Exemplificando, se o nó A é capaz de se comunicar com os nós C, D e F sem a necessidade de utilizar nós intermediários, neste caso os nós C, D e F são “nós vizinhos” do nó A. No entanto, cabe lembrar que cada um dos nós membros de uma rede *ad hoc* é capaz de realizar o roteamento de pacotes. Assim sendo, se este nó A desejar transmitir dados para um outro nó que não for um de seus vizinhos, uma rota entre esses dois nós deverá ser viabilizada para que esta comunicação se torne possível. Os responsáveis pela viabilização destas rotas são os protocolos de “roteamento”, e é necessário que cada um dos nós de uma rede *ad hoc* execute pelo menos um algoritmo de “roteamento” (ALBUQUERQUE, 2003).

Tratando-se de uma rede sem fio, é preciso considerar que o ar é o meio físico compartilhado pelos seus nós. Portanto, protocolos de acesso ao meio devem ser utilizados para controlar eventuais disputas pelo acesso ao meio físico. A camada MAC (*Media Access Control*), que permite o compartilhamento do meio físico, e a camada de Rede, onde são executados os protocolos de “roteamento”, devem ser transparentes para as aplicações. A escolha dos protocolos mais adequados para estas camadas terá uma influência determinante para o nível do serviço conseguido na rede (ALBUQUERQUE, 2003).

2.6.5 UTILIZAÇÃO DE REDES *AD HOC*

Tipicamente, a utilização das redes *ad hoc* está associada a cenários onde existe a necessidade de se instalar rapidamente uma rede. Normalmente isto ocorre em cenários onde não há uma infra-estrutura de rede previamente instalada. Andrade e Colli (2003) fornecem exemplos de situações onde há demanda de instalação de redes desta espécie:

- a) coordenação de equipes de resgate em situações de desastre como terremotos, furacões e inundações;
- b) troca de informações em campos de batalha; e
- c) compartilhamento de informações em reuniões ou aulas.

2.6.6 INFRA-ESTRUTURA DE REDES *AD HOC*

Efetuando a comparação entre as redes fixas, as redes sem fio infra-estruturadas e as redes *ad hoc*, é possível identificar sensíveis vantagens que o uso desta topologia oferece. Andrade e Colli (2003) listam as seguintes vantagens:

- a) instalação rápida - as redes *ad hoc* podem ser estabelecidas dinamicamente em locais onde não haja previamente uma infra-estrutura de rede instalada;

- b) tolerância a falhas - as permanentes adaptações e reconfigurações de rotas em redes *ad hoc*, permitem que as perdas de conectividade entre os nós possam ser facilmente resolvidas através do estabelecimento de uma nova rota;
- c) conectividade - dois nós móveis podem se comunicar diretamente (sem intermediários); e
- d) mobilidade - esta é uma vantagem primordial em relação às redes fixas.

2.6.7 DESVANTAGENS DA UTILIZAÇÃO DE REDES *AD HOC*

Contudo, apesar das vantagens explícitas desta tipologia de rede, é preciso ponderar seus aspectos negativos. Vários autores têm se debruçado sobre este tema e apresentam uma série de desvantagens das redes *ad hoc*. Para Andrade e Colli (2003) estas desvantagens são as seguintes:

- a) roteamento - a mobilidade dos nós e o caráter dinâmico desta topologia contribuem para o aumento da complexidade dos algoritmos de “roteamento”, tornando esta tarefa um dos principais desafios enfrentados na implementação das redes *ad hoc*;
- b) localização - outra dificuldade enfrentada nas redes *ad hoc* é a determinação da localização de um nó, pois além do endereço da máquina não ter relação com a posição atual do nó, também não existem informações geográficas que auxiliem na determinação de seu posicionamento;
- c) taxa de erros - a taxa de erros associada a enlaces sem-fio é mais elevada.

Em outro trabalho, os autores Gracioli e Nunes (2006) apresentam uma listagem alternativa de questões problemáticas oriundas da utilização das redes *ad hoc*:

- a) problema de mobilidade - nas redes sem fio o algoritmo de detecção de erros deve considerar que os nós estão sempre em movimento, pois se em um momento eles podem ser considerados “nós vizinhos”, em outro eles podem estar fora do alcance de transmissão, assim sendo, o serviço de detecção de erros não deve suspeitar de um nó simplesmente porque ele saiu, momentaneamente, da vizinhança;

- b) problema na qualidade da comunicação sem fio - esse problema se refere à maior vulnerabilidade, à perda de mensagens e ao desempenho da rede *ad hoc* (largura de banda menor, atraso na comunicação, latência, etc.), pois o detector de erros deve ser capaz de trabalhar sem violar as propriedades de abrangência e exatidão, tidas como desejáveis para um detector;
- c) problemas de bateria e consumo de energia - os nós são alimentados por baterias, conseqüentemente possuem uma séria limitação de energia, e isto leva à necessidade de reformular o modelo de troca de informações e dados da rede, otimizando-o para um consumo mínimo de energia, o que maximiza o tempo de operação de um nó.

2.7 CONCLUSÃO

Neste capítulo foi apresentada de forma sucinta a estrutura organizacional e física de um hospital, foram apresentados os principais parâmetros que a equipe de enfermagem monitora rotineiramente junto ao paciente e foram apresentadas as características principais dos diversos tipos de rede sem fio de computadores.

Pelas características do presente trabalho optou-se por monitorar apenas dois parâmetros: a temperatura do paciente e o consumo intravenoso de soro e/ou medicamento.

Além disto, confrontando-se a estrutura de um hospital observou-se que a utilização das redes com características sem fio seria adequada, em especial a rede *ad hoc* é indicada em ambientes onde a estrutura física pode eventualmente se movimentar, ou seja, onde existem pontos de acesso que estão se movendo dentro da infra-estrutura da rede. Este modo apresenta como uma das principais vantagens a possibilidade de movimentação dos nós sem comprometer sua comunicação. Não obstante, uma das principais desvantagens do modo *ad hoc* está na redução da banda, o que, em alguns casos, inviabiliza sua utilização. Contudo, no presente trabalho esta desvantagem acabou não sendo prejudicial, pois a quantidade de dados que trafega na rede é pequena. Ainda assim, estas informações trafegam na rede somente quando é detectada uma alteração em relação à última leitura, o que pode demorar vários segundos.

Ao aliar-se a observação das características físicas e operacionais do ambiente hospitalar com a possibilidade tecnológica de considerar os pacientes como os nós sensores de

uma rede, é possível concluir que torna-se interessante, para este projeto, utilizar as características funcionais do modo *ad hoc* de redes sem fio.

CAPÍTULO 3

DESCRIÇÃO DO SISTEMA

3.1 INTRODUÇÃO

Os avanços provenientes das novas tecnologias de informação, principalmente na área de redes sem fio, a integração e a miniaturização da microeletrônica, os sensores e a *internet* permitem que sejam desenvolvidos sistemas de monitoração de pacientes, com uma maior confiabilidade e agilidade entre trocas de informações, de uma maneira geral trazendo benefícios para o cuidado da saúde.

Existem diferenciados sistemas desenvolvidos para aplicações na área de saúde. Alguns têm foco na prevenção de doenças, ou seja, fazer um diagnóstico preventivo. Outros sistemas focalizam na detecção de uma doença baseados em parâmetros e sintomas geralmente encontrados em casos de suas manifestações. O presente trabalho vai agir em outro setor, pois pretende ser um sistema de auxílio aos profissionais de saúde, melhorando a agilidade (velocidade) de acesso as informações sobre o estado do paciente, resultando em uma maior confiabilidade dos dados e conseqüentemente um fácil acesso a eles.

Com as novas tecnologias os sistemas estão comparativamente compactos, mais confiáveis e mais baratos. Esta tendência está abrindo a oportunidade para que o setor de sistemas que cuidam da saúde se torne um ótimo nicho para novas aplicações. Com o objetivo de monitorar os dados de pacientes com precisão e de modo contínuo e, por fim, também conseguir transmiti-los para que se tornem acessíveis em toda a extensão do hospital, ou mesmo fora dele, foi inicialmente concebida uma estrutura pautada em conceitos que podem ser considerados comparativamente simples. Obviamente para isto é necessário possuir um *hardware* que faça a aquisição de determinados dados do paciente. Além disto, para que o mesmo possa se locomover por todo o ambiente hospitalar sem que o sistema perca sua capacidade de monitoração, é necessário que seja compacto e com capacidade de transmissão sem fio das informações. Finalmente para que os profissionais de saúde possam ter acesso a todas estas informações, em tempo real e a qualquer lugar, seja dentro ou fora do hospital, optou-se pelo uso da estrutura fornecida pela *internet*.

3.2 APRESENTAÇÃO GERAL DO SISTEMA

O presente trabalho está pautado na união de duas frentes do conhecimento, a engenharia biomédica e a tecnologia de rede sem fio. A idéia é permitir o monitoramento de pacientes dentro de um hospital, de modo que seus dados vitais possam ser disponibilizados em qualquer parte do hospital e também na *internet*. Assim o médico responsável, pode saber a qualquer instante e em qualquer lugar, o valor dos parâmetros que estejam sendo monitorados. Além disto, o sistema pode ser utilizado nas enfermarias do hospital para o gerenciamento do paciente, automatizando as tarefas.

Na área de engenharia biomédica o trabalho consiste em desenvolver ou adaptar os sensores que desempenharão as leituras dos parâmetros relevantes no cotidiano do ambiente hospitalar. Desta forma para demonstrar a viabilidade do sistema, selecionaram-se inicialmente como parâmetros a serem monitorados a temperatura e o acompanhamento da administração de soro (conta gotas).

Como discutido anteriormente por sua confiabilidade e custo, optou-se pelo uso de rede *ad hoc*. Nestas redes os nós são unidades basicamente compostas por um *hardware* que tem acesso aos sensores e a um sistema operacional para sua gerência.

Assim cada nó de rede *ad hoc* pode transmitir seus pacotes e rotear informações de outros nós, escolhendo rotas e diminuindo o congestionamento da rede, a fim de propiciar sempre a melhor opção de caminho para a informação chegar ao destino final, neste caso o servidor que armazena os dados coletados.

O nó monitor é composto de um *hardware* de aquisição (composto basicamente pelos transdutores e respectivos circuitos de acondicionamento de sinal), um *hardware* de processamento e transmissão podendo ser um PC (*Personal Computer*), um *handheld* ou até mesmo um *pocket PC*, todos equipados com equipamentos 802.11, para funcionarem em modo de rede *ad hoc*). A figura 3 mostra esquematicamente o sistema desenvolvido.

Observa-se que a rede monitora *ad hoc* serve como infra-estrutura para que os dados cheguem ao *gateway*. Portanto, a utilização desta rede não necessita acréscimo de módulos extras ao sistema operacional.

No nó monitor existe um aplicativo desenvolvido em C++ utilizando a API (*Application Programming Interface*) do sistema operacional presente (*Windows, Windows Mobile, PalmOS ou Linux*). Este aplicativo utiliza a porta de comunicação 80, onde a troca de informações é baseada no protocolo HTTP (*Hypertext Transfer Protocol*). Porém os dados enviados são codificados em XML (*extensible markup language*), para que o programa possa

ser o mais genérico possível e para que também possa ser colocado em forma de pacotes que são facilmente entendidos pelos componentes da APIDOM (*API Document object model*). Assim é necessário que todos os aplicativos do sistema entendam e processem XML.

O aplicativo do *gateway* de rede recebe então um pacote XML através da porta 80, e envia os dados para uma base de dados *MySql* localizada em um servidor remoto.

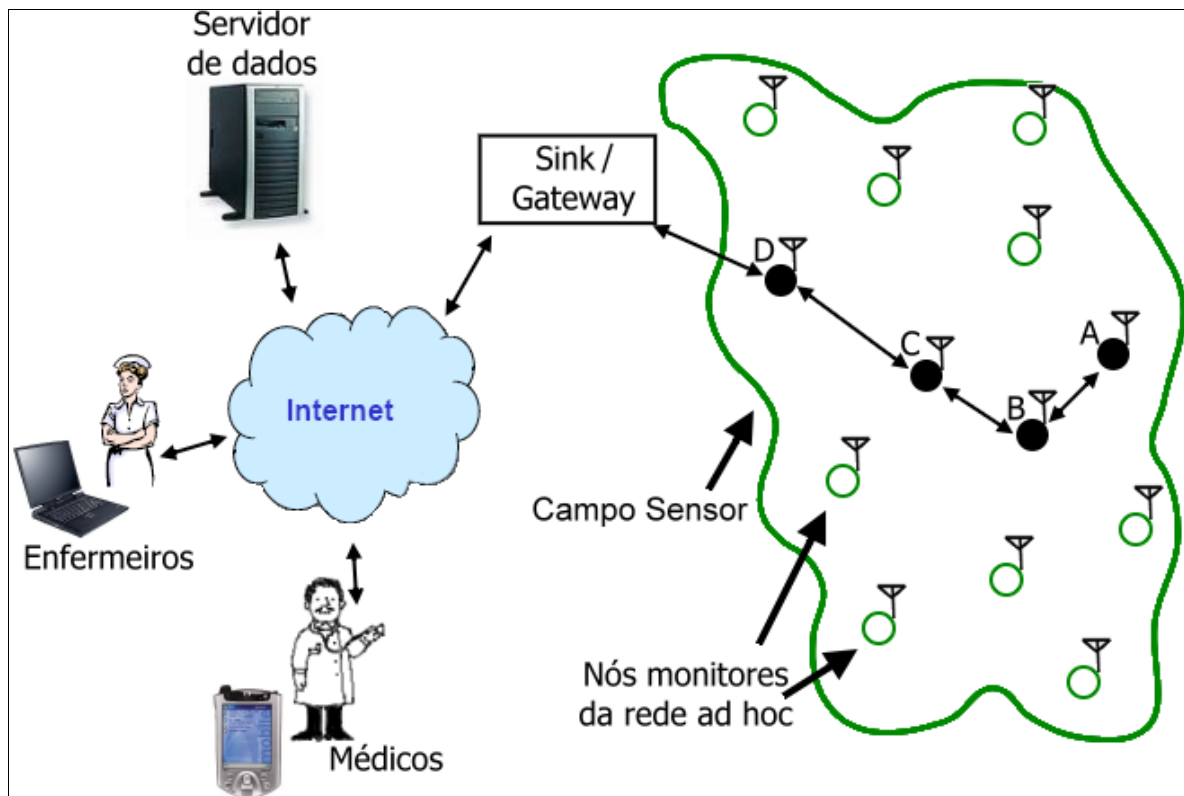


Figura 3: Distribuição dos elementos do sistema.

Assim que um usuário acessa o sistema ele tem os serviços básicos sobre o banco de dados, inclusão, exclusão, alteração e consulta. Mas o objetivo principal de todo o processo é o monitoramento em tempo real do paciente. Para esta tarefa foi necessária a construção de um aplicativo que consiga ficar sempre requisitando ao servidor de dados a última informação sobre o paciente. Através da utilização do AJAX (*Asynchronous Javascript And XML*), que na verdade não é uma linguagem de programação e sim a união de várias linguagens (HTML (*HyperText Markup Language*), Javascript, CSS (*Cascading Style Sheets*), XML, XSLT (*extensible stylesheet language*) transformation, APIDOM, PHP (*Hypertext Preprocessor*), SQL (*Structured Query Language*)) com o intuito de proporcionar o melhor balanceamento entre o programa local e o programa remoto, foi construído um aplicativo específico.

Para que os dados do pacientes estejam sendo mostrados para o usuário em tempo real, visto que o atraso entre a aquisição e a visualização são apenas os tempos de propagação dos dados através da rede *ad hoc* até chegar no *gateway* e o tempo de gravação na base de dados, existe também um aplicativo que está no servidor esperando por qualquer tipo de requisição de um usuário (inclusão na base de dados, exclusão, alteração e consulta).

Essa solução é muito dependente dos programas que estão rodando em todo o sistema. Na figura 4 são apresentados os possíveis caminhos que os dados podem trafegar pelo sistema e seus protocolos. Esta é uma solução que visa atender os conceitos de (1) deixar o processamento da aplicação do usuário no *browser* e o processamento dos dados no servidor, (2) transformar a aplicação *web* em um processo contínuo sem recarregar as páginas e (3) empregar padrões que possam ser utilizados por outras linguagens de programação. Com isso (1) o tamanho do documento HTML é sempre o menor possível, diminuindo assim o tempo de transmissão, (2) sempre os dados mais atualizados do banco de dados são mostrados, pois o aplicativo no cliente sempre trabalha com a última atualização do banco de dados e (3) não há a necessidade de recarregamentos contínuos da tela de monitoração, pois a mesma está continuamente exibindo o último dado que está na base de dados.

3.2.1 NÓS MONITORES DA REDE *AD HOC*

Um sensor, tipicamente, consiste de cinco componentes: detector de *hardware*, memória, bateria, processador embutido e *transceiver*. Estes componentes físicos dos sensores requerem cuidados especiais, tais como o monitoramento contínuo para a verificação de seu funcionamento. Para uma perfeita harmonia e correto funcionamento, o sistema operacional deve cuidar de todos os detalhes envolvidos com o envio das mensagens pela rede, aliviando a carga de processamento do *hardware* local.

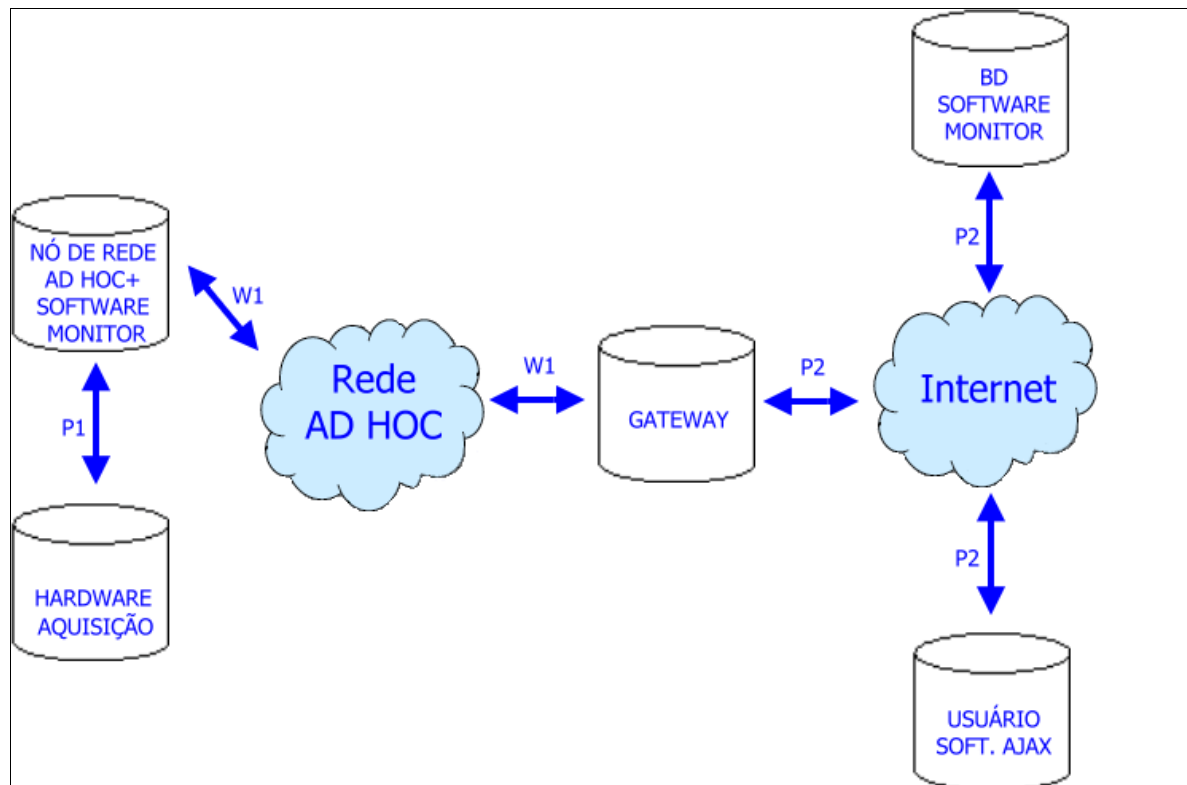


Figura 4: Caminho do tráfego das informações e seus protocolos.

O sensor é o dispositivo que executa a monitoração de um fenômeno qualquer e gera relatórios de medidas (podendo ser através de comunicação sem fio, USB (Universal Serial Bus) ou RS232). No presente sistema, cada nó se comunica com o equipamento que possui o *hardware* em conformidade com o padrão 802.11 através de comunicação RS232. Cada vez que um novo dado se encontra presente no nó, este é disseminado na estrutura *ad hoc* até chegar ao *gateway*. Os dispositivos de detecção, geralmente, têm características físicas e teóricas diferentes. Muitos modelos de complexidade variada podem ser construídos baseados na necessidade da aplicação e características do ambiente em monitoramento, em questão o hospital. Na maioria dos modelos de dispositivos sensores a habilidade de detecção diminui com o aumento da distância do sensor ao fenômeno e melhora com o aumento do tempo que o sensor fica exposto para coletar informações (BROOKS, 2003).

Após o recebimento do dado atualizado do paciente, um aplicativo presente no nó efetua a comunicação com servidor de dados para que o mesmo seja incluído na base de dados.

3.2.2 CAMPO SENSOR

O campo sensor é definido como sendo a região espacial onde estão dispostos os nós sensores. Estes podem estar dispostos em forma aleatória, pois a estrutura de funcionamento da rede em modo *ad hoc*, consegue gerenciar os nós dentro do alcance dos *tranceivers* e configurá-los para fazerem parte da rede. Genericamente cada nó tem uma área de cobertura em forma de círculo e a rede é possível quando as áreas dos nós são sobrepostas, gerando assim comunicação entre os nós na topologia *ad hoc*.

Numa rede monitora *ad hoc* típica, os sensores individuais coletam amostras de valores locais (medidas) e disseminam esta informação direto ao observador ou quando necessário, para outros nós de rede para que os dados sejam roteados até o observador.

Uma demonstração de sobreposição de dois nós pode ser visualizada na figura 5, onde os dados podem ser roteados entre nós adjacentes. Para gerar um campo sensor é só generalizar o que está na figura 5 para uma quantidade maior de nós.

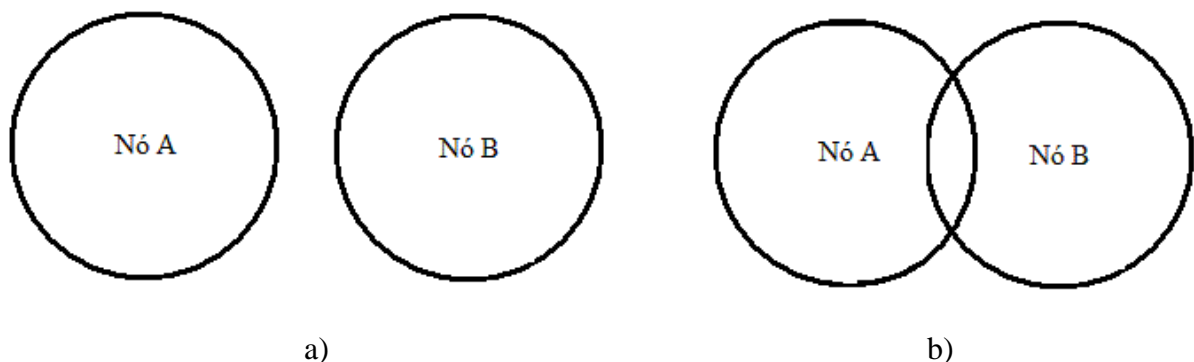


Figura 5: a) nós A e B separados sem comunicação e b) nós A e B se comunicando.

3.2.3 GATEWAY/SINK

Um *gateway* ou *sink* é uma máquina que é destinada a interligar duas redes com tecnologias diferentes. No sistema desenvolvido (ver figura 2), o *gateway* desenvolve o papel de conversão de tecnologia de rede *ad hoc*, para o padrão 100T-*ethernet*, onde se encontra uma porta para *internet*. No sistema é utilizado um microcomputador padrão *Intel 386*, com conexão para os tipos de redes: *ad hoc* e 100T-*ethernet*. Os dados na verdade já possuem um destino certo, que é a base de dados do sistema que se encontra no servidor.

3.2.4 SERVIDOR DE DADOS

O servidor é fundamental para o funcionamento do sistema, pois é nele que está armazenada a base de dados. Além disto, é nele que estão sendo executados os aplicativos, que são necessários para gerenciar, mostrar e armazenar os dados que estão sendo monitorados pela rede monitora *ad hoc*. No presente sistema existe uma máquina compatível com a tecnologia *Intel 386*, rodando sistema operacional *Linux*, capaz de executar o *Webserver Apache* e a base de dados *MySql*. Os dois aplicativos são de domínio público e estão presentes em vários servidores espalhados pelo mundo. No sistema desenvolvido estes aplicativos são necessários, pois as trocas de informações são feitas através do protocolo HTTP, e programas escritos para serem compatíveis com a tecnologia AJAX precisam rodar no servidor através da interpretação de programas escritos com a linguagem PHP, para poder devolver as requisições feitas pelos usuários do sistema.

3.2.5 USUÁRIOS

E por fim existem os usuários do sistema. Estes podem consultar o estado atual do paciente em tempo real, ou realizar consultas de algum período em que o paciente estava sendo monitorado. Também são capazes de executar tarefas simples de configuração, busca, inserção de pacientes, alteração de base de dados.

Uma grande ajuda nesse sentido foi elaborada através da linguagem *JavaScript*. É uma linguagem de alto nível, utilizada muitas vezes de forma errônea e limitada para criar *pop-ups*, imagens rotativas e outros artifícios, às vezes incômodos para os usuários. Deste uso é que vem a má fama do *JavaScript*, mas se essa linguagem for bem empregada, tornará a experiência do usuário com a *web* enriquecedora (OLIVIEIRO, 2001).

É preciso deixar bem claro que *JavaScript* não tem nada a ver com *Java*, pois *JavaScript* nem descende dela, visto que inicialmente chamava-se *LiveScript* e na última hora foi renomeada pela *NetScape* como *JavaScript*.

O que distingue o *JavaScript* é sua capacidade de interação plena com a página HTML, permitindo que novos pedaços de código HTML ou mesmo *JavaScript* sejam adicionados dinamicamente, além de possuir a maioria das estruturas típicas de uma linguagem moderna. É possível ainda utilizar a orientação a objetos em *JavaScript*, apesar de

a linguagem não contemplar todos os elementos da O.O. (herança, propriedades e métodos privados ou protegidos, por exemplo).

Um dos pilares do AJAX é o CSS, uma ferramenta poderosa para modificações do estilo de uma página *web*, e o que é mais interessante, as modificações podem acontecer de forma dinâmica, proporcionando melhor interação com o usuário. Com o CSS fica fácil modificar praticamente qualquer informação em uma página, desde a cor de fundo de uma célula até a fonte, borda, posição do texto, imagem de fundo e muitas outras propriedades dos elementos de uma página *web* (SOARES, 2006).

A utilização de CSS com AJAX é fundamental, pois possibilita alertar o usuário sobre cada uma das etapas do processo, desde mensagens informando que o processo foi iniciado (“aguarde - processando”, por exemplo) até formas complexas e criativas para chamar a atenção sobre novas informações em determinadas partes da página *web*.

Outra parte fundamental na estrutura AJAX é o XML, e aliado a APIDOM, fica relativamente fácil manipular documentos neste formato, tornando possível gerenciar uma página *web* como um documento XML, ou seja, como uma coleção de nós (de uma árvore binária), elementos básicos desta linguagem.

Para isso existem dois objetos nesta APIDOM, o primeiro é o *XMLDocument*, através do qual se pode criar um objeto XML e manipulá-lo livremente, incluindo nós, atributos, etc. O segundo é fundamental para o AJAX é o *XMLHttpRequest*, o qual não é um padrão *DOM*, mas já está implementado nos mais importantes *browsers* disponíveis no mercado. Isso traz como desvantagem o fato de que não existe uma forma padrão de criar esse objeto, mas é conseguido facilmente com poucas linhas de *JavaScript*. O mais importante é sua característica de realizar interações assíncronas com o servidor *web* (apesar de ser possível realizar conexões não assíncronas).

3.3 FLUXO DAS INFORMAÇÕES

O fluxo das informações segue uma estrutura física e flui pelo sistema utilizando protocolos diferentes em cada parte do mesmo (ver figura 4).

O protocolo P1 é o protocolo utilizado para comunicar a aquisição feita pelo *hardware* de aquisição e transportá-la ao *hardware* de processamento do nó monitor de rede *ad hoc*, onde se encontra um aplicativo que entende o protocolo P1 e o converte no protocolo W1. Este navega pela rede monitora *ad hoc* podendo ser roteado pelos nós adjacentes até chegar

no *gateway*. O protocolo P1 é executado utilizando a camada física de comunicação padrão RS232 com velocidade de 19200 bps (bits por segundo).

O quadro de dados do protocolo P1 está descrito na figura 6 e 7. Os dados podem ser requisitados pelo *hardware* de processamento e transmissão junto ao *hardware* de aquisição ou podem ser enviados automaticamente por este quando existe uma variação nos valores dos parâmetros que estão sendo monitorados.

O protocolo P1 utiliza 4 *bytes* para requisição de informação junto ao *hardware* de aquisição. Dois tipos de informação podem ser requisitadas, (1) o número de série do *hardware*, onde os *bytes* “#” e “*” são os marcadores de começo e fim do quadro e os *bytes* “H” e “W” são a identificação da palavra *hardware* e (2) a última aquisição feita pelo *hardware*, onde os *bytes* “D” e “T” são a identificação de *data* (dados).

Quando o *hardware* responde a uma requisição ou simplesmente envia uma alteração dos parâmetros para o nó sensor, nesta comunicação são utilizados 10 *bytes* para que sejam passados os parâmetros atuais de monitoramento.

O protocolo P1 em resposta transmite o número do *hardware* que será utilizado como identificador de cada nó sensor. Os *bytes* HW0 e HW1 são aqueles que correspondem ao número do *hardware* no formato *unsigned short int*, o que possibilita o sistema possuir um máximo de 65535 unidades de nós no sistema. Observa-se que caracteres “U”, “A”, “T” e “X” foram acrescentados para completar o quadro de 10 *bytes* de envio. Já no caso da transmissão, os dados que estão sendo monitorados, os *bytes* G0,G1,G2 e G3 são de um número inteiro no formato *unsigned long* que correspondem no presente trabalho a contagem de gotas do paciente, enquanto os *bytes* T0 e T1 são os que correspondem a temperatura do paciente. Observa-se que os dados de temperatura estão no formato *unsigned short int*. Desta forma este dado requer um tratamento matemático para os ajustes das casas decimais da leitura em questão (ver item 4.x.x).

4 bytes

1)	#	H	W	*
2)	#	D	T	*

Figura 6: Protocolo de envio do nó sensor para o *hardware*, 4 bytes ASCII.

10 bytes

1)	#	H	W	HW0	HW1	U	A	I	X	*
2)	#	D	T	G0	G1	G2	G3	T0	T1	*

Figura 7: Protocolo de envio do *hardware* para o nó sensor, 10 bytes ASCII.

No aplicativo que roda no nó sensor é executada a conversão dos dados transmitindo-os via protocolo HTTP em cima do TCP/IP (*Transfer Control Protocol / Internet Protocol*) até o servidor de dados, passando por duas topologias de rede diferentes.

O pacote primeiramente é transportado utilizando a estrutura de rede sem fio configurada no modo *ad hoc*, protocolo W1. Quando o pacote cruza o *gateway* ele não muda de protocolo de transporte, só muda de topologia passando a 100T – *ethernet*, identificado pelo protocolo P2. Em P2 e W1 tem-se o protocolo TCP/IP responsável pelas comunicações feitas entre o *gateway*, servidor e observador. O transporte dos dados coletados se dá na forma de uma requisição HTML padrão. O nó requisita um *socket* na porta 80 do servidor de dados e, após se conectar ao servidor, envia um pedido requisitando um endereço URL (*Uniform Resource Locator*). No mesmo pedido o nó transmite os dados do paciente. Neste endereço está um aplicativo que é responsável por extrair os dados do pedido e transferi-los para a base de dados. Assim os dados do paciente são armazenados e estão prontos para serem consultados.

O observador quando requisita algum dado para o servidor, recebe uma resposta HTML + XML em cima do protocolo TCP/IP, para se comunicar com o observador. Após isto, localmente no *browser*, é iniciada a ferramenta de busca, configuração e visualização dos dados para o observador. O fluxo de dados e protocolos utilizados para que as informações trafeguem entre os diversos programas existentes no sistema, necessita de todos os protocolos e comunicações exemplificadas na figura 8.

Os componentes existentes no *browser* APIDOM, conseguem distinguir os dados somente se estes estiverem em descrição XML, caso contrário, eles são apenas um conglomerado de *bytes* enviados do servidor ao *browser*, sem significado algum.

Após estas considerações e ainda ressaltando que originalmente o *browser* responde apenas a comandos executados pelo observador, é necessário que alguns procedimentos sejam executados em linguagem *JavaScript*. É nesta parte da programação que se consegue então retirar a necessidade de intervenção do usuário junto ao *browser*, para que seja recarregada uma página, e por último, é utilizada a linguagem *XLS* para melhorar a visualização dos dados recebidos em XML. Na figura 9 um trecho de código XML.

Os detalhes deste processo estão descritos no capítulo 5 onde estão descritos todos os passos do transporte dos dados e seus respectivos aplicativos.

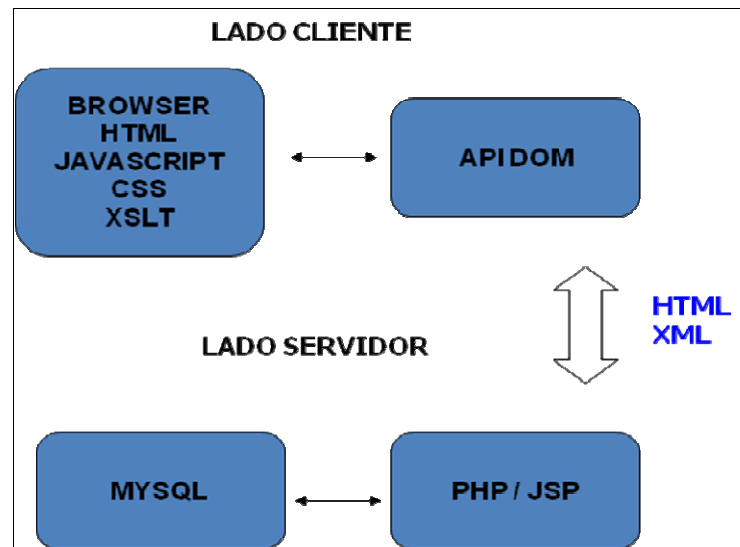


Figura 8: Caminho do tráfego das informações e seus protocolos em relação ao observador.

```

- <estadoatual id="1">
  - <paciente>
    <hospital>Sao Lucas</hospital>
    <medico>Joao M.</medico>
    <temperatura>26.1</temperatura>
    <gotas>189</gotas>
    <volumet>9.31</volumet>
    <temporestante>1458</temporestante>
  </paciente>
</estadoatual>

```

Figura 9: Trecho de código em XML.

3.4 ESTRUTURA DO BANCO DE DADOS

A linguagem SQL é uma das linguagens mais utilizadas para consulta, inclusão, exclusão e alteração de dados em bases de dados relacionais e objeto-relacionais. Dentre os grandes gerenciadores de banco de dados que se utilizam desta linguagem pode-se citar o *Oracle*, o *Microsoft SQL Server*, o *Sybase*, o *Postgree SQL*, o *MySql* entre outros. A linguagem SQL é um conjunto de comandos de manipulação de banco de dados utilizado para criar e manter a estrutura desse banco de dados, além de incluir, excluir, modificar e pesquisar informações nas tabelas dele (OLIVEIRA, 2002).

A linguagem SQL não é uma linguagem de programação autônoma, podendo ser chamada de “sublinguagem”. Quando se escrevem aplicações para banco de dados, é

necessário utilizar uma linguagem de programação tradicional (C, Java, Pascal, Cobol etc.) e embutir comandos SQL para manipular os dados (OLIVEIRA, 2002).

Em um modelo relacional, apenas um tipo de estrutura de dados fica em evidência: a tabela. Novas tabelas são criadas com a junção ou combinação de outras tabelas. Utilizando apenas um comando SQL é possível pesquisar dados em diversas tabelas ou atualizar e excluir diversas linhas de tabelas.

No sistema de monitoramento proposto, não se pretende construir uma base de dados sólida e robusta, pois para isso seriam necessários vários estudos de casos do processo como um todo. Assim foram propostas apenas as tabelas e dados estritamente necessários para se ter um monitoramento em tempo real e o armazenamento dos dados para futura consulta pelo sistema.

A análise mínima dos dados gerou as tabelas de dados que podem ser visualizadas na figura 10.

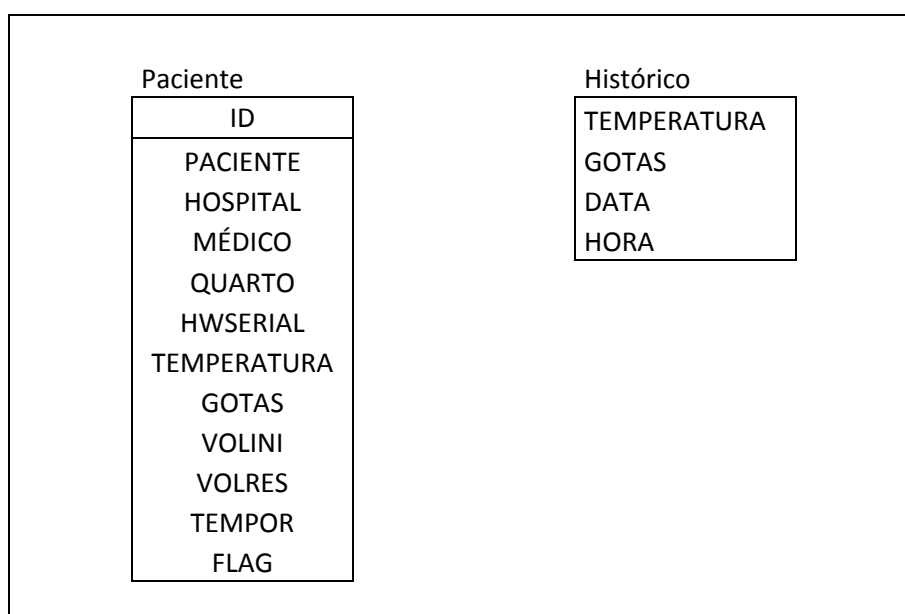


Figura 10: Descrição da tabelas de dados.

Com estes dados foi possível realizar as consultas básicas, incluir dados, alterar dados e fazer o monitoramento em tempo real do paciente. A tabela paciente contém o item ID como chave primária e é classificado como único para cada registro, e sendo incrementado a cada paciente que for incluído no sistema. Os demais dados são: nome do paciente (paciente), nome do hospital (hospital), médico responsável pelo tratamento (médico), número serial do *hardware* (hwserial), temperatura atual do paciente (temperatura), número de gotas já

administradas no paciente (gotas), volume inicial a ser administrado no paciente (volini), a quantidade de volume restante a ser administrado no paciente (volres), tempo restante para término do volume inicial que sendo administrado no paciente (tempor) e o estado atual do paciente (*flag*), que indica se o paciente está ou não sendo monitorado em tempo real. A segunda tabela é o histórico do paciente, é formada por: temperatura do paciente (temperatura), quantidade de gotas (gotas), a data da última alteração (data) e a hora da última atualização (hora).

O funcionamento do sistema em relação aos dados é simples. O aplicativo que está monitorando o *hardware* que está presente no nó sensor, quando envia os dados para o servidor, grava ao mesmo tempo na tabela de dados do paciente e na tabela de histórico o último dado recebido. Para cada paciente existe uma tabela de histórico que está vincula ao paciente através do campo ID da tabela histórico. Assim para o aplicativo leitor de dados no observador dentro do *browser*, se o *flag* da tabela paciente estiver habilitado, a última leitura dos dados do paciente encontra-se na própria tabela paciente, se não, está disponível somente o histórico do paciente, que se encontra na tabela do histórico.

3.5 CONFIGURANDO UMA REDE AD HOC NO WINDOWS XP

É possível, através dos *drivers* controladores de alguns *hardwares* USB sem fio, implementar uma rede *ad hoc* gerenciada pelo sistema operacional *Windows XP*. Abaixo, apresenta-se a rotina de configuração do equipamento *TP-LINK TL-WN321G* para o desempenho desta função.

1º. Passo: Instalar o *driver* do dispositivo sem fio.

2º. Passo: Através do aplicativo instalado, é possível alterar as configurações padrões deste dispositivo. Acessando a opção *Add Profile*, observam-se alguns campos. O campo *SSID* é destinado para o recebimento do “nome da rede”. O campo *network type* destina-se à seleção do modo através do qual a rede irá funcionar (estruturado ou *ad hoc*). No campo *ad hoc wireless mode* seleciona-se o padrão de comunicação IEEE desejado. Na figura 11, exemplifica-se uma rede intitulada “APODEME”, operando sob o modo *ad hoc* com o padrão 802.11b.

3º. Passo: Acessando a janela principal o campo *wireless mode* já estará configurado no padrão 802.11b. É preciso, agora, estabelecer uma taxa de transmissão (*Tx Rate*) fixa para que a rede não proceda a mudanças de valores durante a comunicação. Na figura 12 a *Tx Rate*

foi fixada em 1Mbps, a menor taxa de transmissão possível, mas que possibilita um alcance máximo da rede.

Para o correto, e desejado, funcionamento do modo *ad hoc*, é necessário que estes procedimentos sejam efetuados em todos os nós da rede.

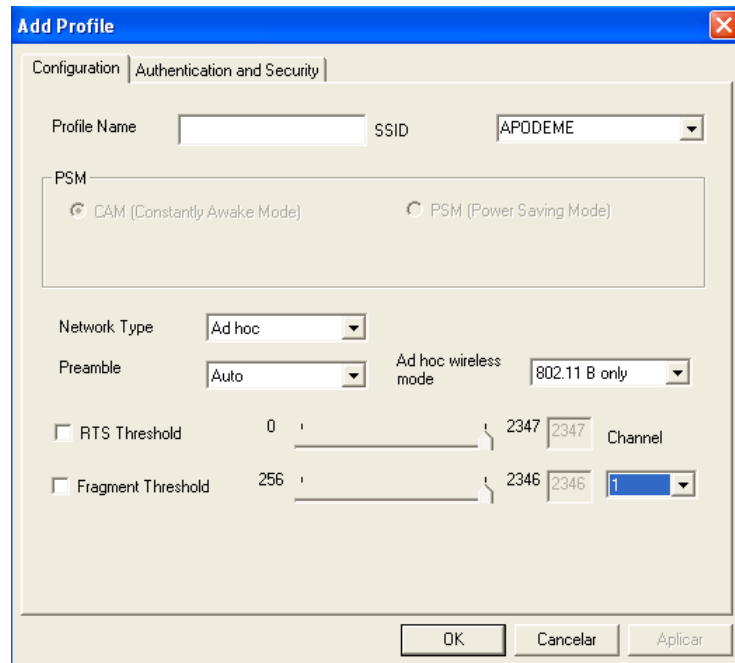


Figura 11: Tela do *Add Profile*.

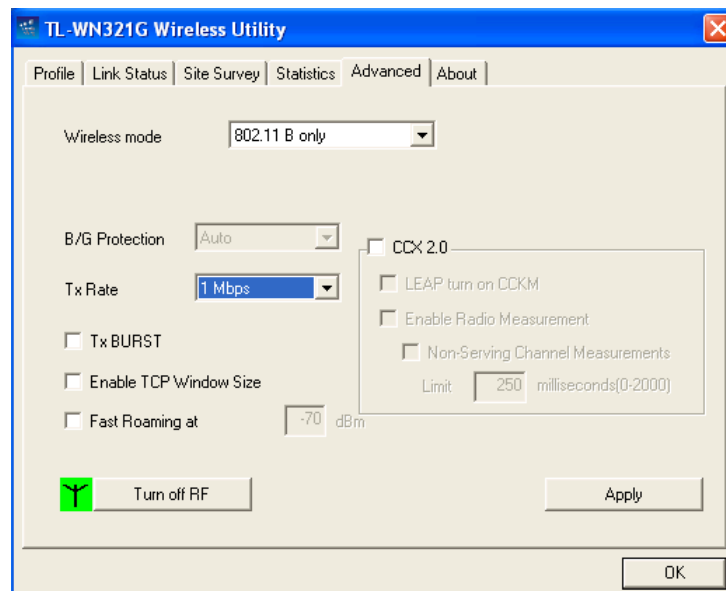


Figura 12: Tela principal.

3.6 CONCLUSÃO

A estrutura proposta é uma das muitas possibilidades existentes para conseguir o monitoramento de dados em tempo real, tendo sido desenvolvida, com base em decisões e análises feitas de forma a se obter um sistema de monitoramento em tempo real com confiabilidade adequada.

Vários detalhes da estrutura básica foram apresentados, e cada um se torna essencial para que o objetivo do sistema seja alcançado, sendo cada elemento do sistema descrito com base em sua funcionalidade. Nos próximos capítulos serão apresentadas suas implementações.

CAPÍTULO 4

DESCRIÇÃO DOS TRANSDUTORES UTILIZADOS NO SISTEMA

4.1 INTRODUÇÃO

Dentre os parâmetros mais comuns presentes no cotidiano da enfermagem estão a medição de temperatura e a administração de soluções intravenosas (Silva, 1997).

Neste capítulo serão apresentados os *hardwares* desenvolvidos especificamente para avaliar a temperatura do paciente e a taxa de administração de soluções intravenosas.

4.2 SENSOR DE TEMPERATURA

Tomando como base as faixas de temperatura que o organismo humano costuma apresentar em situações do cotidiano, expostas na figura 14, tem-se uma melhor interpretação de alguns fenômenos e reações do organismo a variações de temperatura.

Aliado a estas informações o sensor de temperatura que precisará ser confeccionado deve atender as necessidades do corpo médico-hospitalar. A maioria dos termômetros que são utilizados nos hospitais têm as seguintes características:

- 1) Precisão de 0,1°.
- 2) Escala na faixa de 25° a 50°.
- 3) Tempo de resposta na faixa de 40 a 60 segundos.

Desta forma, um sensor de temperatura para ser utilizado no lugar de um termômetro de mercúrio, comumente encontrado no ambiente hospitalar, deve atender no mínimo a estas três especificações.

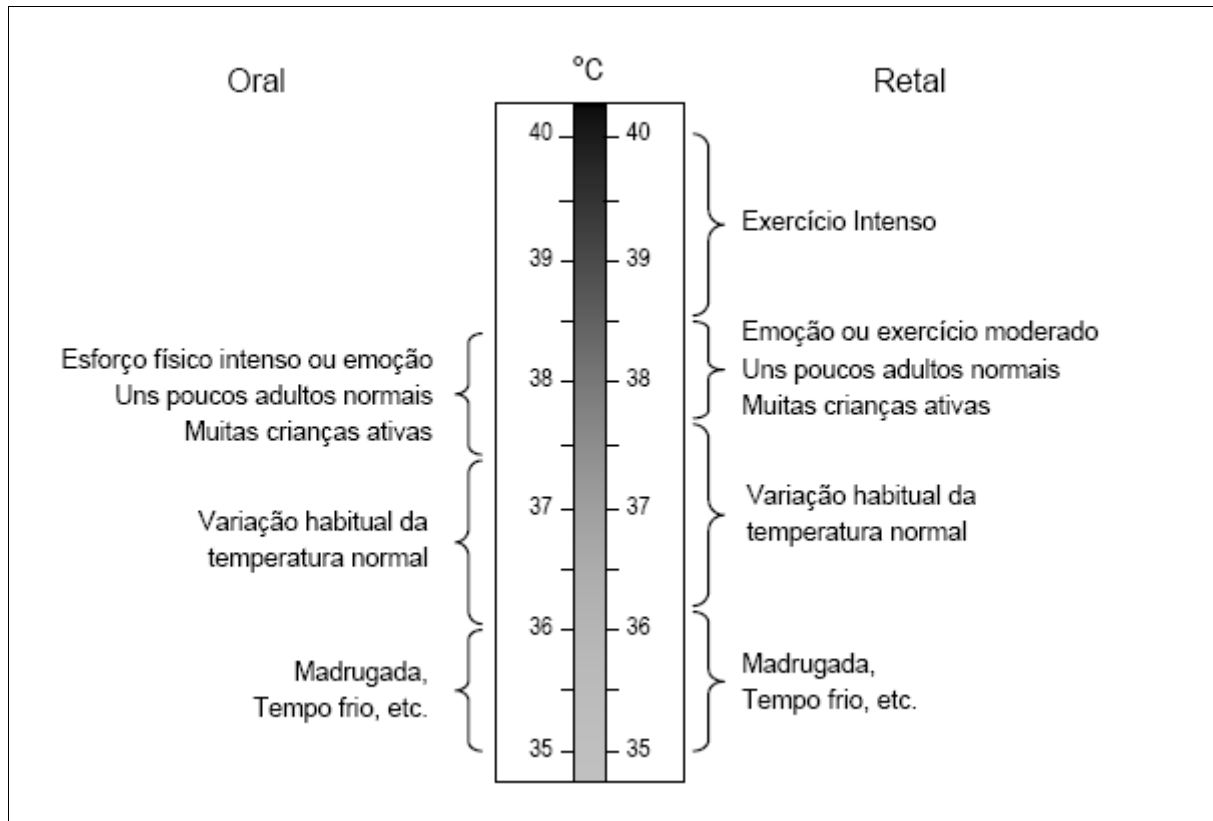


Figura 13: Faixas estimadas de temperaturas para pessoas normais. Na direita encontram-se os limites referentes a medidas feitas no reto, enquanto na esquerda estão as faixas de temperatura medidas oralmente.

Processos físicos muitas vezes são influenciados ou dependem da temperatura, podendo assim ser usados como meio para a construção de transdutores térmicos. Provavelmente, a expansão térmica é o princípio mais simples de medição de temperatura, sendo usado nos termômetros de mercúrio e em uma grande variedade de instrumentos de medição e controle (COBBOLD, 1974).

Para a transdução elétrica da temperatura, diferentes métodos e dispositivos podem ser empregados, os quais incluem transdutores termoresistivos, termoelétricos, ópticos e piezoelétricos.

A dependência da resistência elétrica com a temperatura nos metais foi evidenciada por Humphry Davy, em 1821, sendo que um termômetro baseado na resistência da platina foi desenvolvido por William Siemens em 1871. A partir de então, alguns metais passaram a ser usados para detectar temperatura. As vantagens dos transdutores termoresistivos se resumem na simplicidade de conexão dos mesmos com circuitos elétricos. Estes transdutores podem ser classificados em três grupos: RTDs (Resistance Temperature Detectors), detectores com junção p-n e termistores (FRADEN, 1996).

O efeito termoelétrico foi descoberto por Seebeck, em 1821, o qual verificou a existência de uma tensão na junção de dois metais diferentes. O primeiro fenômeno associado a este efeito, descoberto por Peltier, se deve apenas à tensão que surge pelo contato de dois metais e a temperatura da junção, a qual influencia muito pouco a tensão resultante. O segundo fenômeno foi estudado por Thomson mais tarde (Lord Kelvin) e corresponde à força eletromotriz que aparece ao longo do gradiente térmico aplicado em um condutor, sendo que a tensão produzida é proporcional ao quadrado da diferença entre a temperatura de duas junções (WEBSTER, 1998). O dispositivo que concretiza estes efeitos é denominado termoacoplador ou termopar.

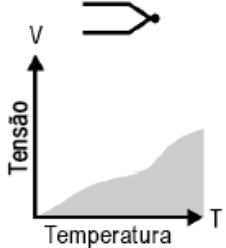
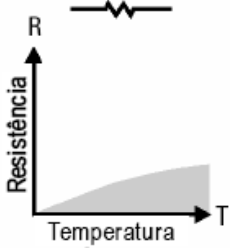
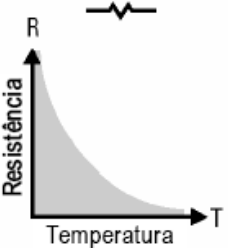
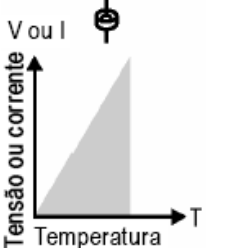
A junção p-n de um diodo ou de um transistor bipolar apresenta uma forte dependência térmica. Quando a junção é conectada a uma fonte constante de corrente, a tensão resultante reflete a medida da temperatura desta junção. Uma característica interessante destes transdutores é o elevado grau de linearidade, facilitando significativamente sua calibração. Normalmente, estes dispositivos se apresentam comercialmente como sensores de temperatura em que a junção p-n encontra-se encapsulada com circuitos de tratamento de sinal analógico e/ou digital, facilitando assim sua conexão com diversos circuitos e com outros dispositivos. (FRADEN, 1996).

Os detectores ópticos utilizam métodos indiretos para avaliação da temperatura, tais como fluorescência, interferência entre ondas e absorção espectral. Tais transdutores são mais complexos e indicados para medidas da temperatura em regiões onde existem campos magnéticos, campos elétricos ou campos eletromagnéticos intensos, que geram interferências em circuitos elétricos (FRADEN, 1996).

Os termistores são elementos semicondutores que sofrem variações acentuadas de sua resistência em função de alterações térmicas, a única desvantagem é que sua resposta não é linear. Seu tamanho reduzido e boa sensibilidade permitem que este dispositivo seja um dos preferidos para aplicações biomédicas (COBBOLD, 1974).

A Tabela 2 mostra um comparativo entre os dispositivos comercialmente utilizados para medir temperatura, dentre os quais, os termoacopladores e termistores compreendem os dispositivos freqüentemente utilizados em aplicações biomédicas (CROMWELL, WEIBELL & PFEIFFER, 1980).

Tabela 2: Comparativo entre transdutores de temperatura comerciais. Os gráficos mostrados passam uma idéia do comportamento da resposta típica de cada dispositivo (Adaptado de Agilent Technologies, 2000).

Termoacoplador	RTD	Termistor	Sensor Integrado
			
VANTAGENS			
Não precisa ser alimentado	Mais Estável	Ampla faixa de saída	Mais Linear
Simples			
Robusto			
Apresenta-se em vários formatos	Mais Preciso	Resposta rápida	Maior faixa de saída
Baixo custo			
Atua em uma ampla faixa de temperaturas	Mais linear que o termoacoplador	Utiliza apenas dois fios	Baixo custo
DESVANTAGENS			
Resposta não-linear	Elevado custo	Resposta não-linear	Atua em temperaturas inferiores a 250 °C
Baixa tensão de saída	Resposta lenta	Faixa de temperatura limitada	Necessita ser alimentado
Requer uma referência	Requer uma fonte de corrente	Estrutura frágil	Resposta lenta
Menos estável	Pequena alteração na resistência	Requer uma fonte de corrente	Apresenta auto-aquecimento
Menos sensível	Utiliza 4 fios para medição	Apresenta auto-aquecimento	Configurações limitadas

Cada sensor possui suas vantagens e desvantagens, porém os critérios necessários em um ambiente hospitalar, comparando-se estes sensores com um termômetro de mercúrio utilizado em hospitais, são facilmente alcançados utilizando-se qualquer um dos sensores apresentados.

4.3 *HARDWARE* ASSOCIADO AO TRANSDUTOR DE TEMPERATURA

Para a construção do circuito transdutor de temperatura optou-se pelo dispositivo **LM35**, que é de fácil aquisição e fornece um nível de tensão proporcional à temperatura de forma direta na sua saída (NATIONAL, 2000). Este dispositivo tem como características principais segundo seu manual, a linearidade, com sensibilidade de 10,0 mV / °C; uma faixa de leitura que vai de -55°C a 150°C; tensão de alimentação que pode variar de 4 a 30 volts e corrente de funcionamento de 60 µA.

Ajustes na saída padrão deste dispositivo foram necessárias, pois no *hardware* do sistema foi utilizado um conversor analógico-digital de 8 bits e tensão de alimentação 5 V. Isto irá fornecer uma precisão inicial (PI) de:

$$PI = \frac{5V}{2^8 \text{ bits}} = \frac{5}{256} = 0,01953125 \text{ volts / bit.}$$

Pegando-se este valor de precisão inicial e colocando seus múltiplos em uma tabela (tabela 3), foi verificado que com esta precisão inicial não seria possível representar todos os valores da primeira casa decimal após a vírgula, pois tem-se um valor de leitura praticamente múltiplo de 0,02 contra os 10 mV / °C. Conseqüentemente a conversão da tensão em temperatura ficaria prejudicada na primeira casa decimal, pois o sensor utilizado converte linearmente a temperatura em tensão na sua saída. Esta tensão de saída tem escala ajustada de tal forma que 0 V corresponde a 0 °C. Para um melhor entendimento do problema pegou-se como exemplo duas leituras do sensor, 20°C que convertido em tensão na saída do componente fica 0,2 V e 20,1 °C que convertido em tensão na saída do componente fica 0,201 V. Note que estas leituras apresentam erro na hora da conversão, pois tomando-se o valor de $10 \times (PI) = 0,1953125V$ e $11 \times (PI) = 0,21484375V$ verifica-se que as duas medidas estão entre estes valores. Para que os valores significativos das leituras entrem na faixa de conversão, basta que se multiplique o valor da tensão de leitura por 100. Assim, a primeira casa decimal foi deslocada duas vezes, resultando uma conversão sem erro até a primeira casa após a vírgula. Porém, uma multiplicação por 100, através de um amplificador, fica inviável, pois a tensão máxima fornecida pelo *hardware* é de 5 V. Assim esta multiplicação foi feita em duas etapas. A primeira é feita por um amplificador operacional configurado no modo não inversor com ganho 10 (o modo não inversor torna-se necessário tendo em vista que a unidade remota do sistema é alimentada por baterias). O resultado desta amplificação limita a máxima

leitura de temperatura que fica em 48,8 °C, mas permanece condizente com a utilização em pacientes. Se fosse necessário efetuar leituras de temperaturas maiores que este valor, a tensão de alimentação deveria ser maior que os 5 V utilizados.

Desta maneira a adição deste amplificador resolve parcialmente o problema, pois ainda falta uma segunda multiplicação por 10. No presente trabalho optou-se por fazer esta segunda multiplicação via programação. Contudo este artifício diminui a exatidão da leitura uma vez que os valores aqui são obtidos a partir da multiplicação por 10 de um valor que já foi digitalizado. Assim para se minimizar este problema decidiu-se fazer a média aritmética das 4 últimas leituras de temperatura.

Tabela 3: Tensão por bits original

Valor Decimal	Bits	Tensão
1	000	0 V
2	001	0,01953125 V
3	010	0,0390625 V
4	011	0,05859375 V
5	100	0,078125 V
6	101	0,09765625 V
7	110	0,1171875 V

O circuito completo do transdutor de temperatura está descrito na figura 15. A linha ADCD0 é a linha que será conectada ao microcontrolador.

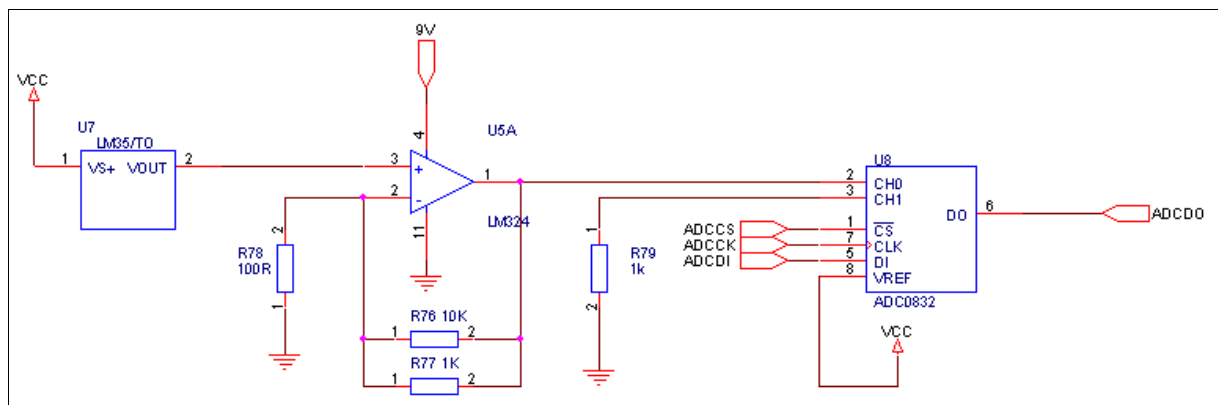


Figura 14: Circuito completo do transdutor de temperatura.

4.4 ADMINISTRAÇÃO DE SUBSTÂNCIAS INTRA-VENOSAS

Em muitos casos os pacientes encontram-se fracos e debilitados e precisam repor várias substâncias no organismo. Estas precisam entrar na circulação sanguínea na velocidade e quantidades adequadas para poder restabelecer sua falta de maneira eficaz melhorando assim a recuperação do paciente. Por outro lado existem pacientes que apresentam problemas no sistema digestivo, portanto necessitam de uma alimentação diferenciada da tradicional, sendo esta através da introdução de substâncias diretamente no sistema circulatório (PETERLINI, 2006).

Em ambos os casos o acesso ao sistema circulatório é feito através de um cateter (um tubo que é inserido em um vaso sanguíneo e que serve como acesso ao vaso em questão). Através dele é possível fazer a administração de várias substâncias diretamente no sistema circulatório. A figura 15 mostra um exemplo de cateter, comparando seu tamanho com uma agulha de injeção.

O controle do fluxo das soluções é feito através de outro equipamento chamado equipo. Este é dotado de um regulador que possibilita o controle do volume da substância que está sendo administrada. Após o equipo ser regulado, através da contagem das gotas por minuto, o enfermeiro passa a verificar somente se a substância se esgotou no frasco e se há necessidade de reajustar a velocidade do volume injetado caso a regulagem tenha sofrido uma alteração. Na figura 15 um exemplo de equipo.

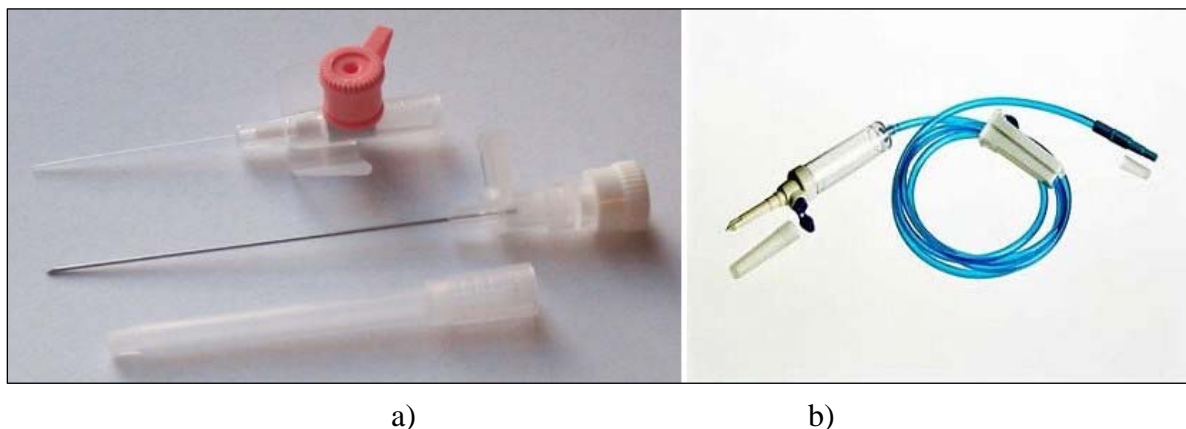


Figura 15: a) Exemplo de um cateter e b) Exemplo de um equipo.

4.5 CIRCUITO CONTA GOTAS

O circuito conta gotas foi desenvolvido com o intuito de melhorar o controle da administração de substâncias intravenosas e gerar alarmes de aviso quanto a qualquer anomalia que pode ocorrer neste processo.

A idéia foi construir um sensor que monitorasse o equipo do paciente, contabilizando a quantidade de gotas administradas. Sabendo-se o volume de cada gota (todas as gotas tem volume aproximadamente igual a $4,9261 \times 10^{-2}$ ml), pode-se computar o volume de solução intravenosa administrada ao paciente. Outrossim, sabendo-se também que as gotas são aplicadas de forma regular com intervalo de tempo entre elas pré-ajustados, pode-se detectar qualquer variação neste ritmo inclusive sua interrupção, que ocorre tipicamente quando a mangueira do equipo dobra.

O princípio utilizado neste sensor foi baseado na idéia de refração que a luz sofre ao atravessar de um meio para outro que possuem diferentes índices de refração (ar – substância intravenosa). Assim a gota no equipo poderia ser considerada como um prisma dispersando a luz, uma vez que esta ao atravessar a gota sofre refração.

Os elementos básicos utilizados para conseguir captar essa refração são o LED (Diodo Emissor de Luz) capaz de emitir luz e o foto-transistor capaz de detectar variações nesta luz emitida quando da presença de uma gota de solução intravenosa no caminho.

Evidentemente, um arranjo para montar esse detector deve ser construído para que o mesmo possa ser encaixado no equipo e também para que o foto-sensor fique protegido de luzes externas que estão presentes no ambiente.

A primeira construção feita para sensor foi o isolamento do foto-transistor em um tubo opaco, para que a única entrada de luz fosse a entrada do tubo. Assim luzes oblíquas a direção de entrada do tubo não afetariam o sensor. A superfície interna do tubo deve ser capaz de atenuar o máximo possível as luzes oblíquas, para tanto foi pintada a superfície interna do tubo com tinta preta e a parte traseira do tubo foi selada com vedante preto visando atingir o mesmo propósito.

O arranjo inicial construído para testes foi o modelo que se encontra na figura 16.

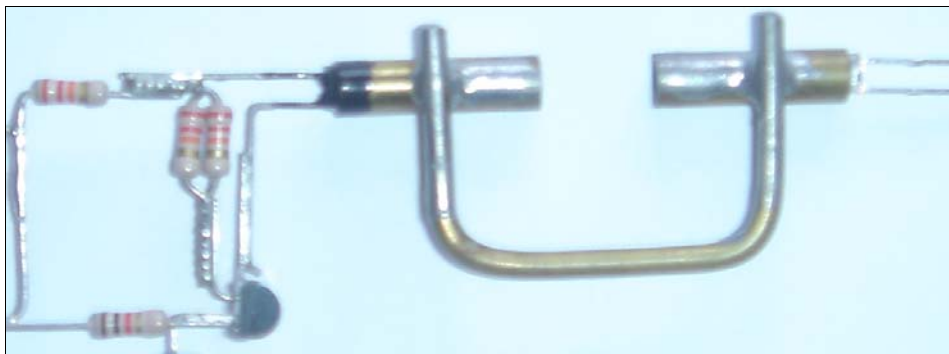


Figura 16: Primeiro protótipo construído para testes de aquisição da gota.

O mesmo processo foi repetido para a fonte luminosa, onde um LED foi fixado dentro de um tubo, para que posteriormente o arranjo pudesse ser fixado ao equipo.

Após o isolamento de ambas as partes (emissor – receptor), foram iniciados os testes com LEDs diferentes. Isso teve que ser feito porque o arranjo composto por um foto-transistor infravermelho e um LED infravermelho formaram um par com alto grau de acoplamento entre emissor e receptor, gerando assim a saturação do foto-transistor mesmo com intensidades extremamente baixas de emissão de luz. Desta forma o transistor por estar operando na região de saturação, mesmo com a gota fazendo o papel do prisma dispersor de luz, não conseguia tirar o foto-transistor da região de saturação, colocando ele na região ativa de funcionamento.

Após estas constatações também fica claro que será necessário calibrar tanto o emissor de luz quanto o receptor, pois a resposta do sistema depende tanto da quantidade de luz emitida quanto da corrente no receptor (proporcional a quantidade de luz recebida). Ajustando estes fatores foi possível identificar a presença da gota cristalina no sistema.

Este modelo foi utilizado para testar LEDs diferentes e fundamentar a idéia que seria possível a identificação de uma gota atravessando o sistema. Após a fixação mecânica e alinhamento dos tubos foi construído um circuito simples utilizando no foto-transistor uma configuração *Darlington* para amplificar a corrente do foto-transistor e o segundo tubo foi utilizado para substituir os LEDs e fazer os testes necessários. Foram testados todos os LEDs comuns, com as cores verde, amarelo, vermelho, azul e branco. A melhor resposta obtida foi com a configuração que utilizava como fonte de iluminação o LED branco, pois este tem uma emissão de infravermelho linear, ou seja, diretamente proporcional a corrente aplicada no LED. Para os outros LEDs, a resposta não era linear sendo que alguns nem possuíam a emissão de infravermelho.

A figura 17 ilustra a variação da tensão de saída do fototransistor durante a passagem de uma gota de soro.

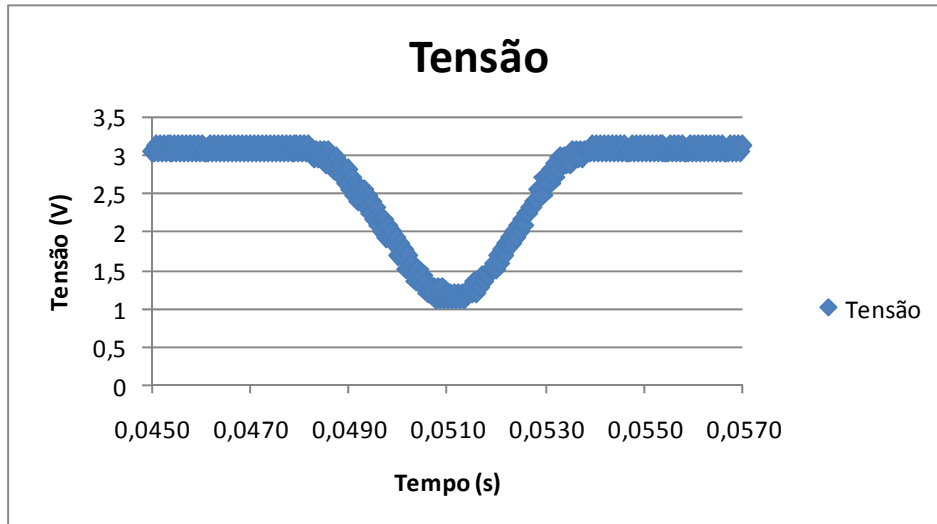


Figura 17: Resposta do circuito quando uma gota passa no sistema.

Após esta análise inicial, que resultara nos ajustes de *hardware* comentados, foi necessária a elaboração de um circuito que possibilite uma maior flexibilidade de uso junto ao equipo. Isto se fez necessário, pois foi verificado que existem vários modelos de equipo, sendo a principal diferença entre os mesmos o material que o compõe, que em alguns casos é plástico transparente, em outros o é acrílico com coloração em leves tons de azul.

Por isto, para cada tipo de equipo foi necessário ajustar a intensidade da fonte de luz para ser possível se detectar a presença da gota. Isto foi conseguido através de algumas considerações sobre o circuito como um todo.

Primeiro a construção dos circuitos levou em consideração a utilização de baterias comuns (pilhas). A tensão de 3V foi adotada como referência positiva no circuito (por isso na figura 17 a tensão na saída do circuito do foto-transistor é 3V). A figura 18 mostra os circuitos emissor e receptor de luz.

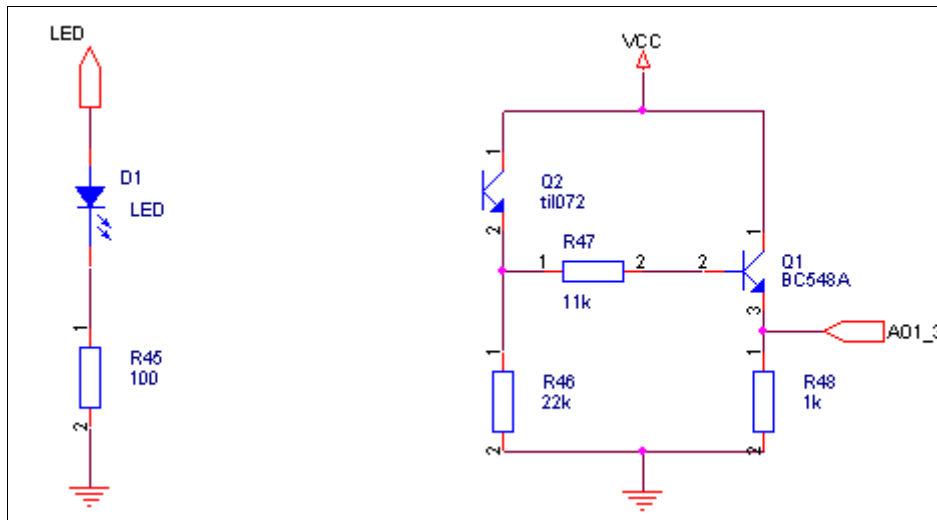


Figura 18: Circuito emissor e receptor de luz.

Segundo para se atingir a condição padrão para a detecção da gota pelo sistema, seria necessário modificar a intensidade da fonte luminosa LED, através da alteração da sua corrente ou a sensibilidade do receptor, esta podendo ser mais alta, ou mais baixa.

Como no *hardware* onde se encontram os sensores há um microcontrolador, optou-se por controlar a corrente no LED com um conversor analógico digital, com sua saída alimentando a fonte luminosa (figura 19).

Observa-se que neste foi utilizado um conversor discreto. A razão para isto foi a alimentação do sistema, que impossibilita a introdução de circuitos integrados de conversão que em muitos casos precisam de alimentação simétrica para funcionar.

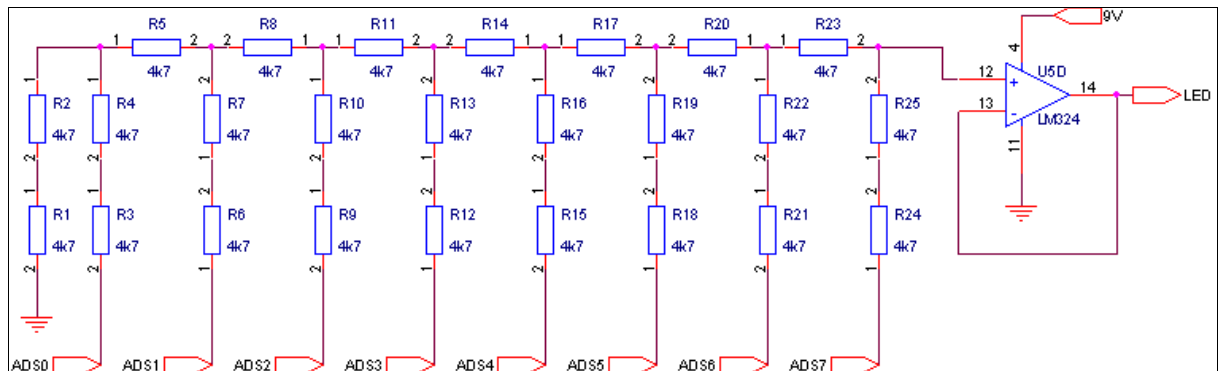


Figura 19: Conversor analógico-digital discreto para controle da luminosidade.

Desta forma através do microcontrolador devidamente programado o sistema automaticamente ajusta a intensidade da luz emitida para que o mesmo possa detectar a presença de gotas de solução intravenosa.

Finalmente houve ainda a necessidade de um circuito que converta o sinal analógico indicativo da presença da gota em um sinal digital. Além disto, no processo de ajuste

automático da intensidade da luz emitida pelo LED se faz necessário informar o microcontrolador quando o sistema está com a tensão adequada no detector luminoso. Para isto foi utilizado um amplificador operacional configurado como comparador com saída digital. Assim foi possível sinalizar ao microcontrolador quando na recepção luminosa a tensão atingiu 3V, nível estabelecido por causa da tensão de alimentação utilizada. Também foi utilizado comparadores para sinalizar ao microcontrolador a passagem da gota, este sinalizador foi configurado em 2V, funcionando como um ceifador de nível inferiores a 2V. O circuito comparador utilizado pode ser vistos na figura 20.

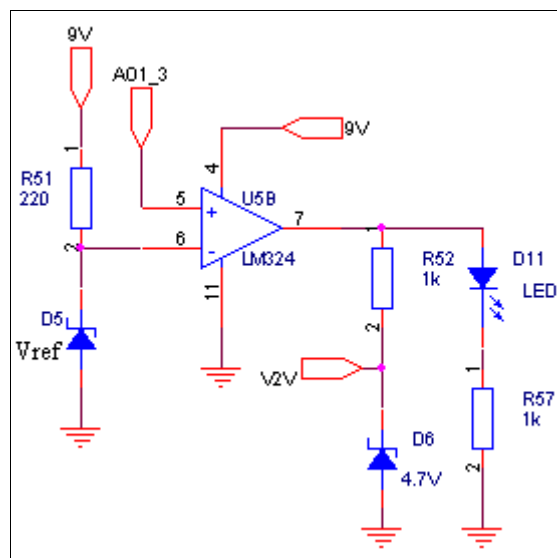


Figura 20: Comparador onde a tensão de referência é de 2 V para detecção da passagem da gota e de 3 V para calibração da tensão do LED.

4.6 CONCLUSÃO

Visando a validação do sistema construiu-se um circuito de monitoração de temperatura e outro para a contagem de gotas. Os circuitos conseguem monitorar estes parâmetros do paciente, sendo que foi colocado ao cargo do microcontrolador a transferência dos mesmos ao nó para sua transmissão para o servidor de dados.

CAPÍTULO 5

PROGRAMAS DO SISTEMA

5.1 INTRODUÇÃO

A interconexão das partes do sistema se dá através de vários aplicativos. Para tal desenvolvimento foram utilizadas várias linguagens de programação, pois cada parte do sistema constitui um ambiente próprio e possui suas peculiaridades.

No HA (*hardware* de aquisição) foi utilizada a linguagem “C”. Esta linguagem foi escolhida por ser uma linguagem de alto nível e apresentar maior agilidade no desenvolvimento do *firmware*. O compilador Keil foi utilizado para gerar os códigos compatíveis com o microcontrolador utilizado no *hardware* do sistema.

No HPT (*hardware* de processamento e transmissão) foi desenvolvido um programa em linguagem “C++”, utilizando sistema operacional *Windows*, sendo o compilador o *Borland builder*, ambos escolhidos por apresentarem muitas facilidades na hora do desenvolvimento do programa.

No servidor optou-se por utilizar o sistema operacional *Linux* e dentre as várias linguagens de programação disponíveis selecionou-se para a base de dados. O *MySQL*, para as páginas o HTML e o AJAX, sendo que os programas intermediários foram desenvolvidos em PHP.

5.2 APLICATIVO DO HA

Dentre as tarefas que o HA deve desempenhar está a leitura dos dados do paciente (temperatura e número de gotas da solução intravenosa), fazer um ajuste nos dados recebidos via correções matemáticas e comunicar-se com o programa do HPT. Para desempenhar esta função, o programa fica constantemente lendo os dados do paciente e ao ser detectada uma alteração em algum dado, seja uma variação de temperatura ou uma nova gota no equipo, este a comunica ao programa do HPT e volta a ficar lendo os dados do paciente. Note que ele só irá se comunicar novamente com o programa do HPT quando detectar outra alteração nos dados do paciente ou caso receba uma requisição de dados do próprio programa do HPT.

Para conseguir executar estas tarefas o programa fica em modo *polling*, não se utilizando de nenhuma interrupção de *hardware* para a verificação dos dados do paciente. Caso o programa do HPT queira executar uma requisição de dados, esta é recebida via interrupção, porém a resposta fica a encargo da temporização feita dentro do *polling* do programa, para que não haja perda de nenhuma informação. Além disto, existe ainda uma interface com um *display* LCD (*Liquid Crystal Display*), conseqüentemente existe uma rotina de atualização dos dados que serão mostrados respeitando também a temporização do *polling*. Cada HA possui um número serial que será vinculado ao paciente na base de dados.

Como mencionado anteriormente, o programa armazena 4 leituras e executa a média delas para considerar a temperatura atual do paciente. Após executar a média multiplica o valor por 10 para deslocar a vírgula da leitura mais uma casa para direita. Por último transforma a leitura em um valor com significado para o usuário, ou seja, em graus *Celsius*, sendo que para isto multiplica a leitura por 5 e divide por 256, ou seja:

$$Temperatura = Leitura \cdot \frac{5(\text{Volts}) \cdot 10(\text{deslocamento_da_vírgula})}{256(8_bits_do_conversor_AD) \cdot 4(\text{média_das_leituras})}.$$

O *polling* do programa foi programado para que sejam executadas todas estas tarefas sem perder nenhum dado e sem perder muito tempo entre as atualizações necessárias no sistema.

Os arquivos do programa do HA são o “*display.h*”, o “*display.c*” e “*main.c*”. Detalhes e códigos fontes estão mostrados no apêndice 2.

5.3 APLICATIVO DO HPT

O aplicativo do HPT tem o papel fundamental de propagar pela rede monitora *ad hoc* os dados recebidos do monitoramento do paciente até os mesmos atingirem o servidor de dados. Além desta tarefa este programa tem que cuidar de alguns cálculos para gerar as estimativas temporais de término do volume intravenoso do paciente.

Ao iniciar, este programa se comunica com o HA para obter o seu código, através do protocolo apresentado na descrição do sistema. De posse deste código o programa entra em contato com o servidor através de uma requisição HTML e requisita os dados do paciente. Em síntese este programa através do *localiza_hardware_ajax.php*, obtém os dados do paciente

após uma consulta no banco de dados. Observa-se que esta informação é enviada ao servidor de dados para o HPT utilizando a linguagem XML sobre o protocolo HTTP. Os dados são recebidos no formato apresentado na figura 21.

```
<?xml version="1.0" ?>
- <hardware>
- <hwserial>
  <id>5</id>
  <nome>Paulo Jose Abatti</nome>
  <volini>65</volini>
</hwserial>
</hardware>
```

Figura 21: Resposta em XML dos dados do paciente.

Estes dados são extraídos do formato XML e apresentados na tela do programa do HPT. A figura 22 mostra uma imagem típica da tela do aplicativo do nó sensor.

Para que o programa consiga executar conexões na porta 80, utilizada pelo protocolo HTML, é necessário o uso de dois componentes existentes no compilador *Borland C++ Builder 6.0*.

O primeiro componente utilizado foi o “*NMHTTP*”, que possui a função de requisitar os dados do paciente na carga do programa. O segundo componente utilizado foi o “*CppWebBrowser*”, que deve enviar os dados coletados do HPT para o servidor de dados. O envio dos dados para o servidor é efetuado no formato padrão de requisições em HTML, com passagem de parâmetros na requisição. No servidor encontra-se um programa exclusivo para receber estes dados com o nome de *inserir_dados.php*. Por exemplo, os dados são passados no formato “*inserir_dados.php ? id=5 & t=25.7 & g=57 & d=28/05/2008 & h=17:53:05:30 & va=2.81 & tr=95.7*”, no qual a variável **id** é o número do paciente no banco de dados, **t** é o temperatura do paciente, **g** é o número de gotas, **d** é a data desta aquisição, **h** é a hora desta aquisição, **va** é o volume atual de substância intravenosa já administrada e **tr** é o tempo restante para acabar o volume de substância intravenosa.

Além destes componentes foram utilizados dois componentes “*TIMER*”, para controlar o intervalo de tempo entre as gotas que atravessam o equipo. A comunicação entre o aplicativo e o *hardware* foi desenvolvida em API *Windows* utilizando *threads*.

Gateway / Nó Sensor

Hardware: [RX]

23 01 00 00 00 00 38 02 0E 2A
 23 01 00 00 00 00 38 02 0D 2A
 23 01 00 00 00 00 38 02 0E 2A
 23 01 00 00 00 00 39 02 0E 2A
 23 01 00 00 00 00 39 02 0F 2A
 23 01 00 00 00 00 39 02 0E 2A

Internet: [HEADER]

HTTP/1.0 200 OK
 Date: Wed, 28 May 2008
 20:52:00 GMT
 Server: Apache/1.3.39 (Unix)
 mod_python/2.7.11
 Python/2.4.3

Internet: [BODY]

```
<?xml version="1.0"?><
<hardware><hwserial><id>5
</id><nome>Paulo Jose
Abatti</nome><volini>65</vo
lini></hwserial></hardware><
```

Paciente:
 Paulo Jose Abatti

ID: 5 **Volume Inicial:** 65

Volume Atual: 2.81 **Tempo Restante:** 95,7395858764648

Timer Gota: 91 **Hardware:** 1

Gotas: 57 **Temperatura:** 25.7

1 byte String 1 ☒ Ativar Gateway

String 2 HW-Internet

DT-Internet

Proxy: 192.168.000.001 **Porta:** 3128 ☐ Usar Proxy

Url:
 http://lasd.hturbo.com/inserir_dados.php?id=5&t=25.7&g=57&d=28/05/2008&h=17:53:05:30&va=2.81&tr=95

Figura 22: Aplicativo do nó sensor.

Assim o aplicativo do HPT é capaz de executar todas as funções necessárias para a transferência de informações com o servidor de dados e o *hardware* monitor. Maiores detalhes e código deste programa podem ser encontrados no apêndice 3.

5.4 APLICATIVOS NO SERVIDOR

A *web* evoluiu rapidamente desde sua criação, começando com páginas estáticas em modo puramente de texto até chegar aos dias atuais com linguagens de alto nível para processamento no servidor, tais como: PHP, ASP, *Java*, *Perl* e recursos junto ao usuário (*browser*), como *JavaScript*, CSS, APIDOM, e XML (SOARES, 2006).

Para tornar o convívio entre aplicações e usuário mais harmônico e intuitivo surgiu o aplicativo AJAX, *JavaScript* e XML assíncrono. Especificamente neste trabalho optou-se pelo uso do AJAX. De qualquer forma, o desenvolvedor, de forma genérica, precisa ter em mente três princípios fundamentais:

- a aplicação deve ser processada pelo *browser* e os dados, pelo servidor;
- a aplicação deve ser um processo contínuo e coerente e
- utilizar padrões e melhores práticas para desenvolvimento, tornando a aplicação harmônica e plena de recursos.

Ressalta-se que esta escolha deveu-se em boa parte a não necessidade de recarregar os *sites* sempre que uma nova informação for requisitada ou enviada automaticamente.

Os aplicativos do sistema estão divididos conforme suas tarefas:

- a) Para criar e inserir dados nas tabelas dentro da base de dados, foi escrito o programa “*criatab_biota.php*” em linguagem PHP (que executa comandos SQL).
- b) O programa utilizado como interface de consulta ao banco de dados do servidor foi escrito também em linguagem PHP (*localiza_hardware_ajax.php*). Os dados consultados são os dados de um paciente que está sendo monitorado. Este programa recebe a identificação do *hardware* e faz uma busca no banco de dados *MySql* pelo paciente que está vinculado com o serial do *hardware* recebido como parâmetro. Também é utilizada a sintaxe SQL para realizar esta consulta. Como retorno este programa apresenta, caso exista o cadastro no banco de dados, uma página *web* com os dados da consulta, ou um erro indicando que tal serial não está na base de dados. O formato de retorno está no padrão XML.
- c) Igualmente escrito em linguagem PHP (*inseri_dados.php*), este programa é utilizado como interface de inserção dos dados que estão sendo monitorados de um paciente. Ele recebe os dados da última leitura do estado do paciente e os coloca no banco de dados do sistema. É utilizada a sintaxe SQL para realizar esta inserção. Este programa não possui retorno, ele simplesmente se conecta a base de dados e faz a inserção do último dado. Assim, caso exista algum erro no processo, este dado simplesmente não é inserido na base de dados.

- d) O arquivo Ajax.css contém todos os estilos e cores das fontes e classe de objetos utilizados em todas as páginas *web* do projeto.
- e) O programa Ajax.js contém o objeto criado para fazer requisições de páginas para o servidor de dados de forma assíncrona. É neste programa que se utiliza o componente *httprequest*, que é o responsável por fazer requisições de páginas dinamicamente sem a necessidade de executar um recarregamento da página atual.
- f) O programa altera.js contém o código para requisitar dados de um determinado paciente ou mesmo para alterá-los. Este programa cria um objeto AJAX e se utiliza do componente *xmldoc* da APIDOM. O objeto AJAX se encarrega das requisições de páginas enquanto que o objeto *xmldoc* é o responsável por decodificar os elementos XML devolvidos em resposta a consulta feita.
- g) O programa listarid.js contém o código para requisitar dados de um determinado paciente e mostrá-los para o usuário permitindo que este consiga fazer uma consulta ao histórico do paciente. Este programa cria um objeto AJAX para comunicação e se utiliza do objeto *xmldoc* para decodificar os elementos XML devolvidos em resposta a consulta feita.
- h) O programa monitora.js contém o código para se requisitar dados de um grupo de pacientes (mesmo que unitário) de maneira que, após selecionar um paciente, uma nova janela do navegador seja aberta e nela apareçam os dados monitorados em tempo real. Este programa cria um objeto AJAX para comunicação e se utiliza do objeto *xmldoc* para decodificar os elementos XML devolvidos em resposta a consulta feita.
- i) A página *index.html* é a página inicial do projeto onde se encontram as opções de inserção de pacientes, alteração de dados, consulta de histórico e o acesso ao monitoramento em tempo real do paciente. Esta página pode ser visualizada na figura 23.
- j) A página *orientador.html* é um *frame* utilizado para simplesmente mostrar os nomes dos responsáveis pelo trabalho. Este *frame* de página principal também pode ser visualizado na figura 23.
- k) A página *codigof.html* é um *link* na página inicial para um *pop-up* que contém todo o código fonte utilizado no servidor *web*.

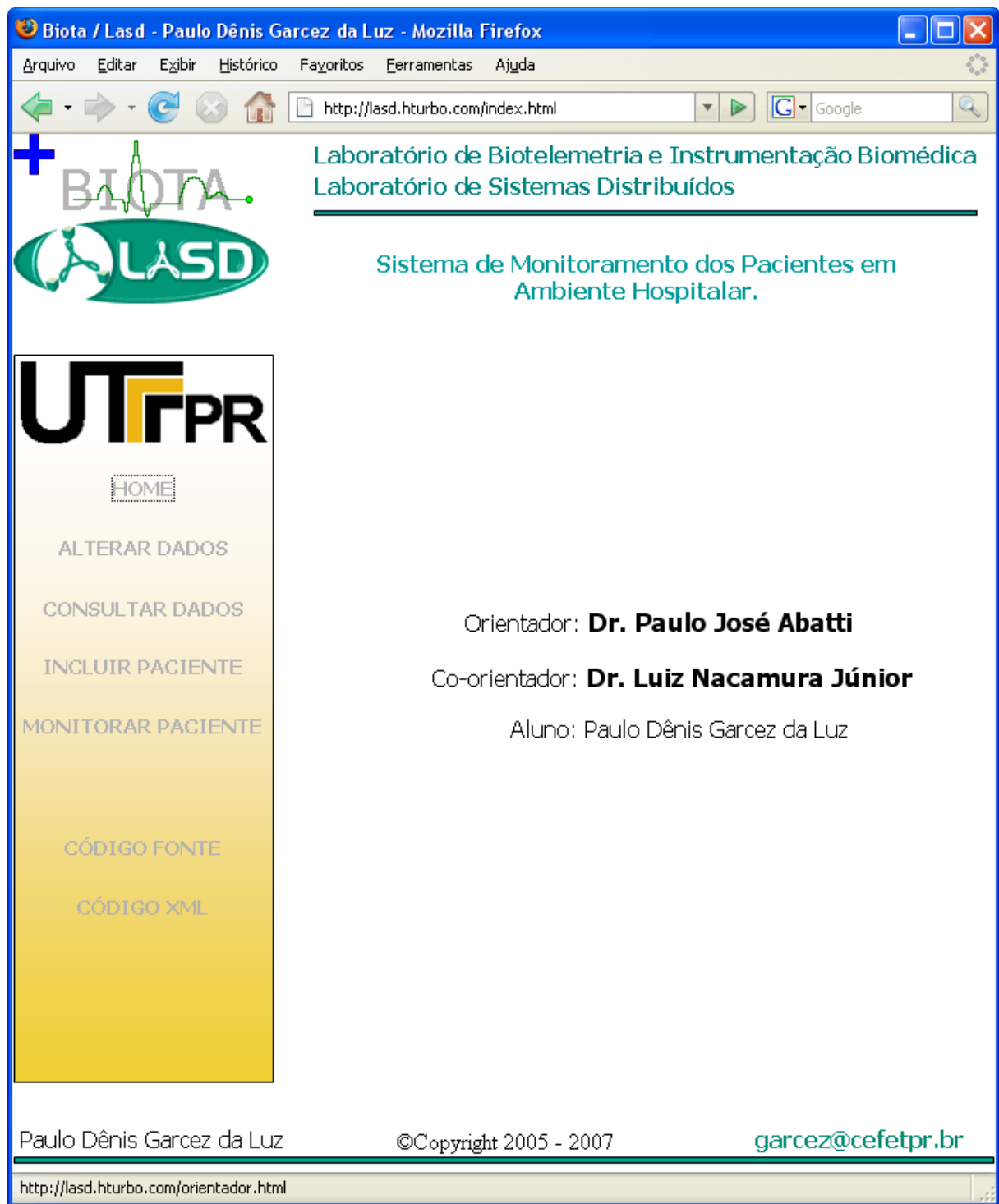


Figura 23: Página index.html e orientador.html em seu frame.

- 1) A página xmldosc.html é um *link* na página inicial que abre no *frame* uma lista de novos *links* e estes sendo acessados mostram na tela exemplos de códigos em XML utilizados nas comunicações entre o servidor e o componente *xmldoc*.

- m) O programa `alterapaciente.php` altera os dados de um paciente que já se encontra presente na base de dados. Recebe os dados via parâmetros de uma requisição padrão HTML e utiliza comandos SQL para manipular a alteração dos mesmos na base de dados do servidor.
- n) O programa `incpac.php` inclui um novo paciente na base de dados do sistema. Recebe os dados via parâmetros de uma requisição padrão HTML e utiliza comandos SQL para manipular a inclusão dos mesmos na base de dados do servidor.
- o) O programa `lista_pacientes_desc_ajax.php` faz uma busca na base de dados do servidor a procura de pacientes que satisfazem aos parâmetros da consulta, que são passados via uma requisição padrão HTML, utilizando comandos SQL, e retorna uma lista referente a esta consulta no padrão XML. Esta lista é retornada ao programa requisitante para que o mesmo possa decodificar os dados e os exiba corretamente na tela do usuário.
- p) O programa `monitorar_pacientes.php` faz uma busca na base de dados à procura do último dado recebido pelo servidor do monitoramento em tempo real do paciente. O programa utiliza comandos SQL para buscar os dados e retorna uma lista com o resultado no padrão XML.
- q) A página `incluir.php` é uma página que pega os dados digitados pelo usuário e envia ao servidor para que seja feita a inclusão de um novo paciente no sistema utilizando o programa (`incpac.php`). A figura 24 mostra uma imagem desta página.
- r) A página `alterar.php` é uma página que pega os dados digitados pelo usuário e envia ao servidor para que seja feita uma alteração dos dados de um paciente já existente na base de dados do sistema utilizando o programa (`altera.js`). A figura 25 apresenta uma imagem desta página.

Biota / Lasd - Paulo Dênis Garcez da Luz - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://lasd.hturbo.com/index.html

Google

BIOTA

Lasd

Laboratório de Biotelemetria e Instrumentação Biomédica
Laboratório de Sistemas Distribuídos

Sistema de Monitoramento dos Pacientes em
Ambiente Hospitalar.

UTFPR

HOME

ALTERAR DADOS

CONSULTAR DADOS

INCLUIR PACIENTE

MONITORAR PACIENTE

CÓDIGO FONTE

CÓDIGO XML

Inclusão de Pacientes

Nome:

Médico:

Hospital:

Quarto:

Paulo Dênis Garcez da Luz ©Copyright 2005 - 2007 garcez@cefetpr.br

Concluído

Figura 24: Página de inclusão de novos pacientes no sistema.

Biota / Lasd - Paulo Dênis Garcez da Luz - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://lasd.hturbo.com/index.html

Google

BIOTA

Lasd

Laboratório de Biotelemetria e Instrumentação Biomédica
Laboratório de Sistemas Distribuídos

Sistema de Monitoramento dos Pacientes em
Ambiente Hospitalar.

UTPR

HOME

ALTERAR DADOS

CONSULTAR DADOS

INCLUIR PACIENTE

MONITORAR PACIENTE

CÓDIGO FONTE

CÓDIGO XML

Alterar Dados do Paciente !!!

Máximo número de pacientes na busca:

Considerar Monitoramento offline na Busca: ☒

Procurar em qualquer parte do Nome do paciente: ☐

Paciente	Volume Ini.	Hardware	Quarto	*Alterar*
Paulo Denis Garcez da Luz	<input type="text" value="1000"/>	<input type="text" value="0"/>	<input type="text" value="32"/>	Alterar
Paulo Jose Abatti	<input type="text" value="150"/>	<input type="text" value="1"/>	<input type="text" value="36"/>	Alterar

Resposta:
Paciente: Paulo Jose Abatti, I1=150, I2=1, I3=36

Paulo Dênis Garcez da Luz

©Copyright 2005 - 2007

garcez@cefetpr.br

Concluído

Figura 25: Página de alteração de dados de um paciente.

- s) A página listarid.php é uma página que coleta os dados digitados pelo usuário e envia ao servidor para que seja feita uma busca no histórico do paciente utilizando o programa (listarid.js). A figura 26 mostra uma tela típica.

Biota / Lasd - Paulo Dênis Garcez da Luz - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://lasd.hturbo.com/index.html

Google

BIOTA

Lasd

Laboratório de Biotelemetria e Instrumentação Biomédica
Laboratório de Sistemas Distribuídos

Sistema de Monitoramento dos Pacientes em Ambiente Hospitalar.

UTPR

HOME

ALTERAR DADOS

CONSULTAR DADOS

INCLUIR PACIENTE

MONITORAR PACIENTE

CÓDIGO FONTE

CÓDIGO XML

Consulta ID !!!

Máximo número de pacientes na busca:

Considerar Monitoramento offline na Busca: ☒

Procurar em qualquer parte do Nome do paciente: ☐

Paciente	Médico	Hospital	Quarto	*ID*
Paulo Denis Garcez da Luz	Joao M.	Sao Lucas	32	1
Paulo Jose Abatti	Luiz J.	Vita	36	5

Consulta Leituras !!!

ID: Data: Hora:

Data	Hora	Temperatura	Gotas
28/05/2008	17:53:05:30	25.7	57
28/05/2008	17:53:12:92	25.6	57

Paulo Dênis Garcez da Luz ©Copyright 2005 - 2007 garcez@cefetpr.br

Concluído

Figura 26: Página de consulta ao histórico de um paciente.

- t) A página monitorar.php é uma página que pega os dados digitados pelo usuário e envia ao servidor para que seja aberta uma nova janela contendo os dados do paciente que está sendo monitorado em tempo real. Na figura 27 está mostrada uma imagem desta página.

Biota / Lasd - Paulo Dênis Garcez da Luz - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://lasd.hturbo.com/index.html

Google

BIOTA

Laboratório de Biotelemetria e Instrumentação Biomédica
Laboratório de Sistemas Distribuídos

Sistema de Monitoramento dos Pacientes em Ambiente Hospitalar.

UTPR

HOME

ALTERAR DADOS

CONSULTAR DADOS

INCLUIR PACIENTE

MONITORAR PACIENTE

CÓDIGO FONTE

CÓDIGO XML

Monitorar Paciente !!!

Máximo número de pacientes na busca:

Considerar Monitoramento offline na Busca: ☒

Procurar em qualquer parte do Nome do paciente: ☐

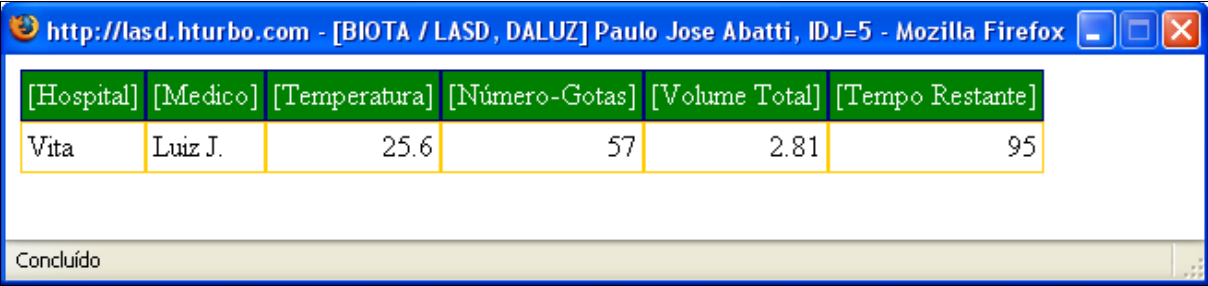
Paciente	Médico	Hospital	Quarto	*Monitorar*
Paulo Denis Garcez da Luz	Joao M.	Sao Lucas	32	Monitorar
Paulo Jose Abatti	Luiz J.	Vita	36	Monitorar

Paulo Dênis Garcez da Luz ©Copyright 2005 - 2007 garcez@cefetpr.br

Concluído

Figura 27: Página de consulta ao para monitoramento em tempo real.

- u) A página janelam.php é uma página que fica mostrando os dados do paciente que está sendo monitorado em tempo real. A figura 28 apresenta uma tela típica desta página.



[Hospital]	[Medico]	[Temperatura]	[Número-Gotas]	[Volume Total]	[Tempo Restante]
Vita	Luiz J.	25.6	57	2.81	95

Concluído

Figura 28: Página de monitoramento em tempo real.

5.5 CONCLUSÃO

Com a construção de todos esses programas e páginas, foi obtido um conjunto de funções mínimas necessárias para controlar o processo de monitoramento em tempo real do paciente. Conjuntamente com a inclusão, alteração e consultas aos históricos dos pacientes.

Este sistema não visou ser uma solução completa em termos de funcionalidades para o sistema, nem procurou ser um sistema com segurança, simplesmente teve por objetivo exemplificar o funcionamento do sistema de acordo com o proposto, comprovando a sua funcionalidade.

A linguagem AJAX, neste caso, foi imprescindível para transformar o comportamento padrão do *browser*, acrescentando a atualização dos dados na página sem a interação do usuário. Isto torna possível a página receber os últimos dados do paciente de uma forma contínua.

CAPÍTULO 6

DISCUSSÃO E CONCLUSÕES

6.1 ANÁLISE DOS RESULTADOS

Como o sistema foi dividido em três partes: *hardware* (HA e HPT – compondo o nó sensor), rede monitora *ad hoc* e aplicativos (programas de execução local – *browser* – e de execução remota – servidor), existiram alguns pontos relevantes na sua construção. Para se chegar à concepção final do sistema, foram escolhidas algumas tecnologias, como a rede *ad hoc*, programas em AJAX, estruturas simplificadas de dados, entre outras escolhas menores em termos de concepção, mas importantes no contexto geral.

Estas escolhas foram trabalhadas, acredita-se, da forma adequada para se obter como resultado final, o sistema idealizado inicialmente, que tem como foco principal o monitoramento de dados fisiológicos de pacientes em tempo real e acessível, localmente, no ambiente hospitalar e mesmo fora dele, através de acessos feitos pela *internet*.

Mais especificamente, na construção dos sensores, dois parâmetros foram escolhidos para serem observados pelo sistema: a temperatura e a contagem de gotas do equipo.

Na construção do sensor de temperatura não existiram grandes dificuldades, porém teve que existir uma correção matemática da leitura para que fosse obtido um valor com exatidão de uma casa decimal. Em um sistema real existe a necessidade de aumentar a resolução da conversão analógica-digital para a obtenção direta do valor sem precisar de uma manobra matemática para a geração de uma leitura correta.

O outro parâmetro lido pelo sistema, a contagem de gotas do equipo, gerou um exaustivo processo construtivo e vários testes para se conseguir uma alteração no acoplamento óptico-elétrico da construção imaginada inicialmente. Neste subsistema dois pontos foram essenciais: a escolha do LED utilizado e o circuito que calibra a intensidade de luz emitida para os diferentes tipos de equipos existentes e utilizados no cotidiano do ambiente hospitalar. Existe um ajuste no circuito receptor de luz, mas esse ajuste, na verdade, é para calibrar o circuito após a escolha do LED.

A escolha da rede *ad hoc* como infra-estrutura da rede de sensores se deu por ser a opção mais simples e prática para a construção do sistema e execução de testes. Embora a estrutura original da rede de sensores seja baseada em *hardware* e aplicativos abertos

utilizando o sistema operacional intitulado de *Tinyos*, que possivelmente seria mais adequada à presente proposta, a mesma se mostrou de difícil acesso pois depende de importação de muitos de seus componentes. Fica então uma ressalva, que esta seria uma melhor opção para o sistema a utilização desta estrutura de redes de sensores. De qualquer forma a rede *ad hoc* utilizada no sistema foi configurada com certa facilidade, porém depende muito da placa sem fio utilizada no nó sensor, pois muitos modelos não programam em seu *driver* de dispositivo uma camada de programas referente ao modo *ad hoc*.

A parte mais trabalhosa foi a construção de todos os programas necessários para colocar o sistema em funcionamento. O primeiro deles é o programa que está no microcontrolador e é responsável pelo gerenciamento e leitura dos dados do paciente. Seguindo a linha de comunicação dos programas, em seguida está o programa do HPT, responsável por transmitir os dados pela rede *ad hoc*, com destino no servidor de dados. Por fim um conjunto de programas e páginas foram desenvolvidos com tecnologia AJAX, para a obtenção dos dados em tempo real remotamente em qualquer lugar onde esteja disponível uma conexão com a *internet*. Ressalta-se, contudo, que uma programação em linguagem Java poderia ter em princípio um desempenho diferenciado nos recursos finais das páginas, agregando um acabamento mais “profissional” às mesmas.

6.2 CONCLUSÕES GERAIS

O sistema apesar de possuir diversos pontos críticos foi desenvolvido e atingiu seu objetivo principal que era demonstrar a possibilidade de efetuar o monitoramento de dados fisiológicos, em tempo real, de um paciente dentro de um ambiente hospitalar (de fato os experimentos foram realizados em laboratório simulando as condições do hospital). Apesar de existirem várias abordagens diferenciadas que gerariam o mesmo resultado, pode-se dizer que a estrutura proposta apresentou um funcionamento satisfatório.

O processo de monitoramento contínuo foi atingido, podendo ser visualizado através do histórico do paciente, que pode ser requisitado por um usuário do sistema. A confiabilidade dos dados sofreu um aprimoramento, diminuindo os erros mais comuns de leitura e controle manuais comuns no cotidiano hospitalar. Uma otimização nos recursos humanos especializados (equipe médica), provavelmente pode ser adquirida com a implantação do sistema em um ambiente hospitalar. O sistema pode agregar mais sensores, com isso, poderia ser utilizado na diminuição do tempo de um paciente dentro da UTI, pois

quando o mesmo necessitar estar sob observação, poderia ser monitorado com o sistema e poderia ser deslocado para um quarto ou enfermaria, sem perder o aspecto de monitoramento contínuo. Assim o tempo total de um internamento poderia ser diminuído, pois com uma eficiência maior no controle dos dados, o paciente seria assistido com maior precisão pela equipe hospitalar. Em alguns casos o sistema poderia acompanhar o paciente até a sua casa, pois para funcionar, bastaria a criação de um *gateway* residencial, agregando um monitoramento após a saída do paciente do hospital.

6.3 TRABALHOS FUTUROS

Neste sistema existem vários aspectos que podem ser modificados, podendo gerar em alguns casos um sistema mais robusto.

Algumas propostas imediatas surgem automaticamente com o entendimento do sistema e o conhecimento de algumas tecnologias. São elas;

- o desenvolvimento de uma maior quantidade de sensores,
- a substituição da rede *ad hoc* por uma rede de sensores utilizando *Tinyos* e
- a construção dos programas em linguagem Java.

Para um sistema real, dentro de um hospital, detalhes relevantes devem ser observados. Por exemplo, recomenda-se o uso de um conversor A/D de no mínimo 12 bits para medir a temperatura com 0,1 °C de resolução.

Além disto, um sistema de gerenciamento de energia (pois trata-se de um sistema portátil alimentado por baterias) deve estar presente no circuito de leitura de gotas no equipo, entre outras providências a serem tomadas para gerar um sistema mais robusto e capaz de realizar tarefas mais complexas e ao mesmo tempo gerenciar uma quantidade maior de dados.

6.4 RESULTADOS OBTIDOS

Este trabalho gerou um artigo publicado no CBEB 2006 (Congresso Brasileiro de Engenharia Biomédica) intitulado: APLICAÇÃO DE REDES DE SENSORES EM AMBIENTES HOSPITALARES.

O trabalho foi contemplado na chamada pública da FINEP 2007 sob o título: APODEME – SISTEMA DE APOIO À DECISÃO MÉDICA USANDO TÉCNICAS WIRELESS E INTERNET.

REFERÊNCIAS BIBLIOGRÁFICAS

AGILENT TECHNOLOGIES, Pratical Temperature Measurements. Manual Técnico. Aplicacion Note 290, 2000.

ALBUQUERQUE, L. R. *Segurança em Redes ad Hoc*. Programa de Engenharia de Sistema e Computação – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.

ANDRADE, M. B.; COLLI, R. *Redes ad hoc*. In: Comunicações Digitais e Tópicos Relacionados, vol1, Departamento de Engenharia – Instituto de Educação Superior de Brasília, agosto de 2003.

BEDIN, E.; RIBEIRO, L. B. M.; BARRETO, R. A. S. S. Humanização da assistência de enfermagem em centro cirúrgico. *Revista Eletrônica de Enfermagem*, v. 06, n. 03, p. 400-409, 2004.

BENEDICT, R.P. *Fundamentals of Temperature, Pressure na Flow Measurements*. Third Edition. New York: John Wiley & Sons, Inc., 1984.

BERTOTTI, F. L. **Desenvolvimento de um medidor de fase para um sistema de biotelemetria passiva**. 2005 149 f. Dissertação (Mestrado em Ciências) - Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2005.

BRASIL. Ministério da Saúde - Secretaria Nacional de Ações Básicas de Saúde. Divisão Nacional de Organização de Serviços de Saúde. Centro de Documentação. **Normas e Padrões de Construções e Instalações de Serviços de Saúde**. Brasília, 1978.

BRONZINO, J. D. *The Biomedical Engineering Handbook*. Second Edition. New York: IEEE Press, 2000.

BROOKS, R. R. “Reactive sensor networks”, Applied Research Laboratory, Pennsylvania State University, disponível em (2003) <http://strange.arl.psu.edu/RSN>.

BULHÕES, I., 1994. Riscos do Trabalho de Enfermagem. Rio de Janeiro: Carioca.

CAMPOS, C. F. P. S. *Estudo e especificação de um sistema de instrumentação para unidades de elevação de petróleo utilizando tecnologia sem fio*. Dissertação (Programa de pós-graduação em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal, 2006.

CHRUSCIAK, D. *Curso Técnico Em Informática Com Ênfase Em Suporte E Manutenção*. Centro Estadual De Educação Profissional Pedro Boaretto Neto, Curitiba, 2008.

COBBOLD, R.S.C. *Transducers for Biomedical Measurements – Principles and Applications*. New York: John Wiley & Sons, Inc., 1974.

CROMWELL, L.; WEIBELL, F.J.; PFEIFFER, E. A. *Biomedical Instrumentation and Measurements*. Second Edition. New Jersey: Prentice-Hall Inc., cap.1, p. 1-343, 1980.

DENIS, E. *Avaliação da tecnologia bluetooth para transmissão de vídeo em tempo real com qualidade de serviço(qos)*, Dissertação de mestrado, Escola Politécnica da Universidade de São Paulo, 2004.

FOUCAULT, M., 1996. *Microfísica do poder*. Rio de Janeiro: Graal.

FRADEN, J. *Handbook of Modern Sensors – Physics, Designs and Applications*. Second Edition. New York Springer-Verlag New York, Inc., cap.16, p. 458-493, 1996.

GAST, M.S. *802.11 wireless networks: the definitive guide*. Sebastopol: O'Reilly, 2002.

GESELOWITZ, D. B.; GESELOWITZ, M. N. The Bioelectrical Century: Bioelectrical Engineering and the “Inside Story” of the Electrical Century. In: **Proceedings of the IEEE**. v. 87, n. 10, p. 1842-1846, October, 1999.

GRACIOLI, G.; NUNES, R. C. Detectores de Defeitos para Redes Wireless Ad Hoc. In: IX Escola Regional de Redes de Computadores, 2006, Passo Fundo - RS. IX Escola Regional de Redes de Computadores, 2006.

GUYTON, A.C. Tratado de Fisiologia Médica. 6ª ed. Rio de Janeiro: Guanabara Koogan, cap.72, p.765-864, 1986.

MENDES, R. **Medicina do Trabalho e doenças ocupacionais**. São Paulo : Savier, 1980.

MIRSHAWKA, V. **Hospital - Fui bem atendido - a vez do Brasil**. São Paulo : Makron Books, 1994.

MOUNTCASTLE, V. B. Fisiologia Médica. 13ª ed. Rio de Janeiro: Guanabara Koogan, cap. 56, p.1306-1342, 1978.

MOURA, A. R. L.; MISKULIN, R. G. S.. *Curso de Introdução a redes de computadores*.

LAPPEMEC, UNICAMP. Disponível em:

<http://www.cempem.fae.unicamp.br/lapemmec/cursos/fe190/hpalunos/turatti/arquivos/introducao_a_redes.pdf>. Acesso em: 29 abr. 2008.

MURHAMMER, M. W.; *et al*; **TCP/IP Tutorial e Técnico**. São Paulo: Makron Books do Brasil, 2000.

NEBEKER, F. Golden Accomplishments in Biomedical Engineering. **IEEE History Center Publication**, Piscataway, NJ, U.S.A., 2002.

OLIVEIRA, C. H. P. SQL: *Curso Prático*. São Paulo: Novatec, 2002.

OLIVIEIRO, C. A. J. Faça um Site: *JavaScript orientado por projeto*. São Paulo: Erica, 2001.

PETERLINI, M.A.S., CHAUD M. N., PEDREIRA, M.L.G. **Incompatibilidade entre fármacos e materiais de cateteres e acessórios da terapia intravenosa.** Online Brazilian Journal of Nursing, Vol 5, No 3, 2006.

QUARTO, C. C. *Redes de Computadores e Internet: aspectos gerais.* Centro Federal de Educação Tecnológica do Maranhão, São Luiz, 2006.

RIBEIRO, H. P., 1993. O Hospital: história e crise. São Paulo: Cortez.

RING, E.F.J. Progress in the Measurement of Human Body Temperature – From the Early Thermoscope to Modern Infrared Imaging, Using Temperature as an Indication of Health or Disease. **IEEE Engineering in Medicine and Biology**, v.17, n.4, p.19-24, August, 1998.

SCHIMITT, A.; *Roteamento.* Curso de Roteamento. Universidade Federal do Rio Grande do Sul, Porto Alegre, agosto de 1996.

SILVA, A. J. S. *As tecnologias de rede wireless.* In: Boletim bimestral sobre tecnologia de redes, vol. 2, n. 5, Rede Nacional de Ensino e Pesquisa, maio de 1998.

SILVA, M. A. Introdução ao Ambiente de Trabalho. In: REIBNITZ, K. S., PRADO, M. L. (Coord.). **Fundamentando o exercício profissional do auxiliar de enfermagem.** 2^a ed., Florianópolis : NFR/SPB, CCS-UFSC, 1997.

SILVA, M. A. **Concepção ergonômica dos locais e dos espaços de trabalho de uma unidade de emergência hospitalar.** 1999. Dissertação (Mestrado em Ciências) - Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1999.

SOARES, L. F. G.; LEMOS, G.; COLCHER, S.; *Redes de Computadores: das LANs, MANs e WAN às Redes ATM.* Ed. Campus, Rio de Janeiro, 1995.

SOARES, W. AJAX: *Asynchronous JavaScript and XML Guia Prático*. São Paulo: Erica, 2006.

TANENBAUM, A. S. **Redes de Computadores**. Rio de Janeiro: Campus, 1997.

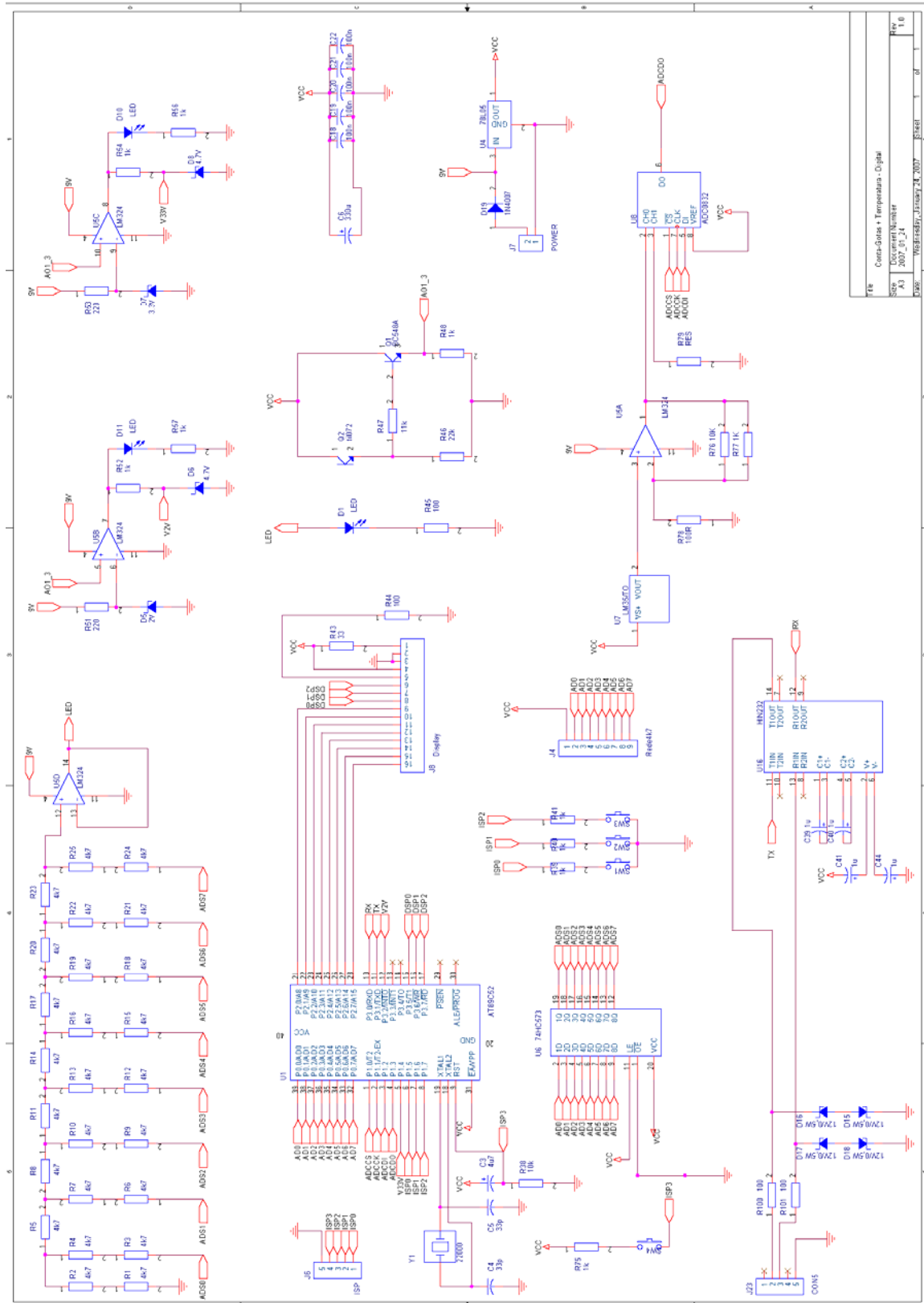
WANG, J.Y.; FELTON, C.M.M. *Instruments for physical environmental measurements*. 2nd ed. Dubuque: Kendall/Hunt Publishing Company, 1983.

WEBSTER, J.G. Medical Instrumentation – Application and Design. Third Edition. New York John Wiley: & Sons, Inc., cap.2, p.61-85, 1998.

YI, S.; NALDURG, P.; KRAVETS, R. *Security-Aware Ad-Hoc Routing for Wireless Networks*. Agosto, 2001.

APÊNDICE 1

ESQUEMÁTICO DO HA



APÊNDICE 2

PROGRAMA DO HA

DISPLAY.C

```
#include <REGX52.H>
#include <string.h>
#include <intrins.h>

#define DISPLAY_RS P1_4
#define DISPLAY_RW P1_3
#define DISPLAY_E P1_2

extern void delay (int tempo);

void inicialcd (void);
void escrevestrlcd (char posicao, char string[]);
void escrevelcd (bit rs, bit rw, char dado);

void inicialcd (void)
{
    escrevelcd(0, 0, 0x38); //Códigos de inicializacao do display
    escrevelcd(0, 0, 0x38);
    escrevelcd(0, 0, 0x06);
    escrevelcd(0, 0, 0x0C);
    escrevelcd(0, 0, 0x01);
}

void escrevestrlcd (char posicao, char string[20])
{
    int b;
    posicao = posicao | 0x80; //Ajusta a posicao do display
    escrevelcd (0, 0, posicao);
    DISPLAY_RS = 1; //dados
    DISPLAY_RW = 0; //escrita
    delay(10);
    for(b=0; b<strlen(string); b++)
    {
        P2 = string[b];
        delay(10);
        DISPLAY_E = 1;    //habilitado
        delay(10);
        DISPLAY_E = 0;
        delay(10);
    }
}

void escrevelcd (bit rs, bit rw, char dado)
```

```

{
    DISPLAY_RS = rs; //instrução
    DISPLAY_RW = rw; //escrita
    P2 = dado;
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    DISPLAY_E = 1;    //habilitado
    DISPLAY_E = 0;
    delay(10);
}

```

HA.C

```

#include <REGX52.h>
#include <string.h>
#include <stdio.h>
#include <display.h>
#include <stdlib.h>
#include <intrins.h>

#define DI P3_3    //pin 13
#define DD P3_4    //pin 14
#define CLK P3_5   //pin 15
#define CS P3_6    //pin 16
#define GTS P3_7   //pin 17
#define NV P3_2    //pin 17

#define HW0 0x01   //Hardware ID
#define HW1 0x00   //Hardware ID

void delay (int tempo);
void pulso_adc(void);
unsigned char ler_adc(unsigned char ch);
void stx(unsigned char temp);
void serialcfg (void);

bit tec1=0;
bit compg=1;
bit flag_gc1=1;
bit flag_rx=1;
bit flag_gc2=1;
unsigned long int gotas=0;
unsigned char *ponteiro,j;
unsigned char display[16];
unsigned char adc0=0,adc1=0,tecc=0,vg[4]={0,0,0,0},rx[4];
unsigned short temp;

```



```

unsigned int contador=0;
float grausc=0;

void srx() interrupt 4 using 0
{
  rx[0]=rx[1];
  rx[1]=rx[2];
  rx[2]=rx[3];
  rx[3]=SBUF;
  if (rx[0]=='#' && rx[3]=='*') flag_rx=1;
  RI=0;
}

void main (void)
{
  inicialcd();
  serialcfg();
  CS=1;
  CLK=0;
  P0=0x00;
  j=0;
  //   escrevestrlcd(0x80,"0123456789012345");
  //   escrevestrlcd(0xC0,"9876543210012345");
  while (compg)
  {
    j++;
    temp=(unsigned char)j;
    P0=temp;
    //   escrevelcd(0, 0, 0x01);
    sprintf(display,"AUTO=[%d]",(int)temp);
    escrevestrlcd(0x80,display);
    //   sprintf(display,"AUTO=[%d]",(int)j);
    //   escrevestrlcd(0xC0,display);
    delay(200);
    P0=temp;
    if (NV) compg=0;
  }
  gotas=0;
  escrevelcd(0, 0, 0x01); //limpar display
  sprintf(display,"G[%lu]",gotas);
  escrevestrlcd(0x80,display);
  sprintf(display,"T[%2.1f]",grausc);
  escrevestrlcd(0xC0,display);
  j=0;
  while (1)
  {
    contador++;
    //Conta Gotas
    if (compg)
    {

```

```

        if (GTS) compg=0;
    }
else
{
    if (GTS==0)
    {
        compg=1;
        gotas++;
        sprintf(display,"G[%lu]",gotas);
        escrevestrlcd(0x80,display);
    }
}
//Recepção Serial
if (flag_rx)
{
    flag_rx=0;
    //Envia HW serial
    if (rx[1]=='H' && rx[2]=='W')
    {
        display[0]='#';
        display[1]='H';
        display[2]='W';
        display[3]=HW0;
        display[4]=HW1;
        display[5]='U';
        display[6]='A';
        display[7]='T';
        display[8]='X';
        display[9]='*';
        for (j=0;j<10;j++) stx(display[j]);
    }
    //Envia Dados
    if (rx[1]=='D' && rx[2]=='T')
    {
        ponteiro=(unsigned char *)&gotas;
        display[0]='#';
        display[1]='D';
        display[2]='T';
        display[3]=(unsigned char)*ponteiro;
        display[4]=(unsigned char)*(ponteiro+1);
        display[5]=(unsigned char)*(ponteiro+2);
        display[6]=(unsigned char)*(ponteiro+3);
        ponteiro=(unsigned char *)&temp;
        display[7]=(unsigned char)*ponteiro;
        display[8]=(unsigned char)*(ponteiro+1);
        display[9]='*';
        for (j=0;j<10;j++) stx(display[j]);
    }
}
//Cálculo da temperatura

```

```

if (flag_gc2)
{
    if (contador>10000)
    {
        flag_gc2=0;
        flag_gc1=1;
        grausc=(float)temp*(float)12.5/(float)256;
    }
}
//Conta Gotas
if (compg)
{
    if (GTS) compg=0;
}
else
{
    if (GTS==0)
    {
        compg=1;
        gotas++;
        sprintf(display,"G[%lu]",gotas);
        escrevestrlcd(0x80,display);
    }
}
//Imprime no Display: temperatura
if (flag_gc1)
{
    if (contador>20000)
    {
        flag_gc1=0;
        sprintf(display,"T[%2.1f]",grausc);
        escrevestrlcd(0xC0,display);
    }
}
//Conta Gotas
if (compg)
{
    if (GTS) compg=0;
}
else
{
    if (GTS==0)
    {
        compg=1;
        gotas++;
        sprintf(display,"G[%lu]",gotas);
        escrevestrlcd(0x80,display);
    }
}
//Leitura do adc0832

```

```

if (contador>30000)
{
//          adc0=ler_adc(0xC0);
//          adc1=ler_adc(0xE0);
//          sprintf(display,"%bu",[%bu]",adc0,adc1);
//          grausc=(float)adc1*(float)50/(float)256;
    adc1=ler_adc(0xE0);
//          for (j=0;j<3;j++) vg[j]=vg[j+1];
    vg[0]=vg[1];
    vg[1]=vg[2];
    vg[2]=vg[3];
    vg[3]=adc1;
    temp=(vg[0]+vg[1]+vg[2]+vg[3]);
    contador=0;
    flag_gc2=1;
}
//Conta Gotas
if (compg)
{
    if (GTS) compg=0;
}
else
{
    if (GTS==0)
    {
        compg=1;
        gotas++;
        sprintf(display,"G[%lu]",gotas);
        escrevestrlcd(0x80,display);
    }
}
//Botao 1
if (tec1)
{
    if (P1_1) tecc++;
    else tecc=0;
    if (tecc>50)
    {
        tec1=0;
        tecc=0;
    }
}
else
{
    if (P1_1==0)
    {
        tec1=1;
        gotas++;
        sprintf(display,"G[%lu]",gotas);
        escrevestrlcd(0x80,display);
    }
}

```

```

    }
}

}

unsigned char ler_adc(unsigned char ch)
{
    CS=0;                                //Habilita conversao
    _nop_();
    _nop_();
    _nop_();
    CLK=0;                                //coloca o clock no estado inicial
    _nop_();
    _nop_();
    _nop_();
    ACC=ch;                                //carrega o canal

    //Escolha de canal
    DI=ACC_7;
    pulso_adc();                          //pulso de clock
    DI=ACC_6;
    pulso_adc();                          //pulso de clock
    DI=ACC_5;
    pulso_adc();                          //pulso de clock
    pulso_adc();                          //pulso de clock
    pulso_adc();                          //pulso de clock

    //prepara para receber a leitura do canal
    ACC=0;
    ACC_7=DD;
    pulso_adc();                          //pulso de clock
    ACC_6=DD;
    pulso_adc();                          //pulso de clock
    ACC_5=DD;
    pulso_adc();                          //pulso de clock
    ACC_4=DD;
    pulso_adc();                          //pulso de clock
    ACC_3=DD;
    pulso_adc();                          //pulso de clock
    ACC_2=DD;
    pulso_adc();                          //pulso de clock
    ACC_1=DD;
    pulso_adc();                          //pulso de clock
    ACC_0=DD;
    pulso_adc();                          //pulso de clock
    CS=1;
    return(ACC);
}

void pulso_adc(void)
{

```

```

CLK=1;
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
CLK=0;
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
}

```

```

void delay (int tempo)
{
    int a;
    int b;
    for (a=0 ; a<tempo ; a++)
        for (b = 0; b<3; b++);
}

```

```

void serialcfg (void)
{
    //T2CON=0X34;
    SCON=0x50;
    TCLK=1;
    RCLK=1;
    C_T2=0;
    RCAP2H=0XFF;
    //RCAP2L=0XB8;    //9600bps
    RCAP2L=0XDC;    //19200bps
    TR2=1;
    EA=1;
    ES=1;
}

```

```

void stx(unsigned char temp)
{
    SBUF=temp;
    while(!TI);
    TI=0;
}

```

```

/*
unsigned char ler_adc(unsigned char ch)
{
    CS=0;
    _nop_();
    _nop_();
}

```

```

_nop_();
CLK=0;
_nop_();
_nop_();
_nop_();
ACC=ch;
B=3;
#pragma asm
LOOPA:
RLC A
MOV DI, C          //envia bit para DI
#pragma endasm
pulso_adc();       //pulso de clock
#pragma asm
DJNZ B, LOOPA
#pragma endasm
pulso_adc();       //pulso de clock
ACC=0;
CY=0;
#pragma asm
MOV B, #8          //8 bits a receber
LOOP2:
MOV C, DD          //recebe bit de DO
RLC A
#pragma endasm
pulso_adc();       //pulso de clock
#pragma asm
DJNZ B, LOOP2
SETB CS            //desabilita o ADC0832
#pragma endasm
CS=1;
return(ACC);
}
*/

```


APÊNDICE 3

PROGRAMA DO HPT

GATEWAY.CPP

```
//-----
#include <vcl.h>
#pragma hdrstop
USEFORM("Unit1.cpp", Form1);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    return 0;
}
//-----
```

UNIT1.CPP

```
//-----
#include <vcl\vcl.h>
#include <stdio.h>
#include <dos.h>
#pragma hdrstop

#include "Unit1.h"

// programa multitarefa incluir a tarefa 02
#include "Unit2.h"

// variáveis globais
HANDLE hComm = NULL;
TRead *ReadThread;
COMMTIMEOUTS ctmoNew = {0}, ctmoOld;
OVERLAPPED serialoverlap;
unsigned long int serialevent=EV_TXEMPTY;
char Serial[10];
unsigned short int Hardware, Temperatura;
```

```

unsigned long int Gotas;
unsigned int Tmedio,Timer;
char Body[10000];

//-----
#pragma link "SHDocVw_OCX"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    // fim do multitarefa.
    ReadThread->Terminate();
    // se a porta está aberta, fechar.
    if (hComm)
    {
        Sleep(250); // espera para fechar a tarefa 02
        SetCommTimeouts(hComm, &ctmoOld);
        CloseHandle(hComm);
    }
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    TObject *b;
    DCB dcbCommPort;
    // abrindo uma porta com.
    hComm=CreateFile("COM1",
        GENERIC_READ | GENERIC_WRITE,
        0,
        0,
        OPEN_EXISTING,
        0,
        0);
    // Se a porta não foi aberta fim.
    if (hComm == INVALID_HANDLE_VALUE) Application->Terminate();
    // configurando os timeouts.
    GetCommTimeouts(hComm,&ctmoOld);
    ctmoNew.ReadTotalTimeoutConstant=100;
    ctmoNew.ReadTotalTimeoutMultiplier=0;
    ctmoNew.WriteTotalTimeoutMultiplier=0;
    ctmoNew.WriteTotalTimeoutConstant=0;
}

```

```

SetCommTimeouts(hComm,&ctmoNew);
// configurando baud rate, paridade, tamanho da palavra e stop bit.
dcbCommPort.DCBlength=sizeof(DCB);
GetCommState(hComm,&dcbCommPort);
BuildCommDCB("19200,N,8,1",&dcbCommPort);
dcbCommPort.fDtrControl=DTR_CONTROL_DISABLE;
SetCommState(hComm,&dcbCommPort);
// ativando o multi-tarefa. se o argumento é falso, suspende o mtsk.
ReadThread = new TRead(false);
Button3Click(b);
CppWebBrowser1->Visible=false;
CppWebBrowser1->Visible=false;
}
//-----

```

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
TransmitCommChar(hComm,'P');
//utilize quando conectado para sincronizacao
//SetCommMask(hComm,EV_TXEMPTY);
//WaitCommEvent(hComm,&serialevent,&serialoverlap);
}
//-----

```

```

void __fastcall TForm1::Button2Click(TObject *Sender)
{
unsigned long out=0;
char *temp="#HW*";
WriteFile(hComm,&temp[0],strlen(temp),&out,NULL);
}
//-----

```

```

void __fastcall TForm1::Button3Click(TObject *Sender)
{
unsigned long out=0;
char *temp="#DT*";
WriteFile(hComm,&temp[0],strlen(temp),&out,NULL);
}
//-----

```

```

void __fastcall TForm1::Button4Click(TObject *Sender)
{
char temp[150];
sprintf(temp,"http://lasd.hturbo.com/localiza_hardware_ajax.php?hwserial=%d",Hardware);
Url->Text=temp;
if (CheckBox1->Checked)
{
NMHTTP1->Proxy = Proxy->Text;
}
}

```

```

    NMHTTP1->ProxyPort = StrToInt(Porta->Text);
}
NMHTTP1->TimeOut = 0;
NMHTTP1->InputFileMode = false;
NMHTTP1->OutputFileMode = false;
NMHTTP1->ReportLevel = Status_Basic;
if (Url->Text.Length()>0) NMHTTP1->Get(Url->Text);
Memo2->Text = NMHTTP1->Header;
Memo3->Text = NMHTTP1->Body;
strcpy(Body,NMHTTP1->Body.c_str());

char nivel1[1000],*temp2,item[3][200],flag=1;
char *acha1="<id>";
char *acha2="<nome>";
char *acha3="<volini>";
int i,j,k,l;

//<id>=item[0]
j=strlen(acha1);
flag=1;
if (strlen(Body)>0)
{
    strcpy(nivel1,Body);
    temp2=strstr(nivel1,acha1);
    if (temp2!=NULL) strcpy(nivel1,temp2);
    else flag=0;
}
i=0;
l=0;//id
k=strlen(nivel1);
while ((i+j)<k&&flag)
{
    while ((nivel1[i+j]!='<')&&((i+j)<k))
    {
        item[l][i]=nivel1[i+j];
        i++;
    }
    item[l][i]=0;
    flag=0;
}

//<nome>=item[1]
j=strlen(acha2);
flag=1;
if (strlen(Body)>0)
{
    strcpy(nivel1,Body);
    temp2=strstr(nivel1,acha2);
    if (temp2!=NULL) strcpy(nivel1,temp2);
    else flag=0;
}

```

```

    }
    i=0;
    l=1;//nome
    k=strlen(nivel1);
    while ((i+j)<k&&flag)
    {
        while ((nivel1[i+j]!='<')&&((i+j)<k))
        {
            item[l][i]=nivel1[i+j];
            i++;
        }
        item[l][i]=0;
        flag=0;
    }

    //<volini>=item[2]
    j=strlen(acha3);
    flag=1;
    if (strlen(Body)>0)
    {
        strcpy(nivel1,Body);
        temp2=strstr(nivel1,acha3);
        if (temp2!=NULL) strcpy(nivel1,temp2);
        else flag=0;
    }
    i=0;
    l=2;//volini
    k=strlen(nivel1);
    while ((i+j)<k&&flag)
    {
        while ((nivel1[i+j]!='<')&&((i+j)<k))
        {
            item[l][i]=nivel1[i+j];
            i++;
        }
        item[l][i]=0;
        flag=0;
    }
    //Colocar itens no Form1 ...
    Edit1->Text=item[1];//Nome
    Edit2->Text=item[0];//Id
    Edit3->Text=item[2];//Volini
    }
    //-----

void __fastcall TForm1::Button5Click(TObject *Sender)
{
    AnsiString Body2;
    char temp[250],temp2[150],temp3[150];

```

```

int id=0;
float t=(float)Temperatura*(float)12.5/(float)256;
float va=(float)50*(float)Gotas/(float)1015;
float ter=(Tmedio*(Edit3->Text.ToInt()-va)*1015)/60000;//tempo restante
struct time ti;
struct date dt;
getdate(&dt);
sprintf(temp2,"%02d/%02d/%04d",dt.da_day,dt.da_mon,dt.da_year);
gettime(&ti);
sprintf(temp3,"%02d:%02d:%02d:%02d",ti.ti_hour,ti.ti_min,ti.ti_sec,ti.ti_hund);

//NMHTTP1->Abort();

Edit5->Text=ter;
if (Edit2->Text.Length()>0) id=Edit2->Text.ToInt();
if (id!=0)
{

sprintf(temp,"http://lasd.hturbo.com/inserir_dados.php?id=%d&t=%.1f&g=%d&d=%s&h=%s&va=%.2f&tr=%d",id,t,Gotas,temp2,temp3,va,(int)ter);
    Url->Text=temp;
    if (CheckBox1->Checked)
    {
        NMHTTP1->Proxy = Proxy->Text;
        NMHTTP1->ProxyPort = StrToInt(Porta->Text);
    }
    NMHTTP1->TimeOut = 0;
    NMHTTP1->InputFileMode = false;
    NMHTTP1->OutputFileMode = false;
    NMHTTP1->ReportLevel = Status_Debug;
    if (Url->Text.Length()>0)
    {
//        NMHTTP1->Get(temp);
        CppWebBrowser1->Navigate(WideString(Url->Text));
    }
    Memo2->Text = NMHTTP1->Header;
    Memo3->Text = NMHTTP1->Body;
//    strcpy(Body,NMHTTP1->Body.c_str());
}
}
//-----

void __fastcall TForm1::CheckBox2Click(TObject *Sender)
{
    TObject *b;
    Button4Click(b);
}
//-----

```

```

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
Memo1->Clear();
}
//-----

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
Timer++;
}
//-----

```

UNIT2.CPP

```

//-----
#include <vcl\vcl.h>
#include <stdio.h>
#pragma hdrstop

// incluir a multitarefa, unidade 01
#include "Unit1.h"
#include "Unit2.h"

extern HANDLE hComm;
extern char Serial[10];
extern unsigned short int Hardware,Temperatura;
extern unsigned long int Gotas;
extern unsigned int Tmedio,Timer;
unsigned char InBuff[2];
unsigned long int Lg;
char Lt[10];

__fastcall TRead::TRead(bool CreateSuspended)
: TThread(CreateSuspended)
{
}
//-----

void __fastcall TRead::DisplayIt(void)
{
int i;
unsigned char *p;
float t;
char temp[10];

sprintf(temp,"%0.2X ",InBuff[0]);
Form1->Memo1->SetSelTextBuf(temp);
//Form1->Memo1->SetSelTextBuf(InBuff[0]);

```

```

for (i=0;i<9;i++)
{
    Serial[i]=Serial[i+1];
}
Serial[9]=InBuff[0];
if (Serial[0]=='#'&&Serial[9]=='*')
{
    if (Serial[1]=='H'&&Serial[2]=='W')
    {
        p=(unsigned char *)&Hardware;
        *p=Serial[3];
        *(p+1)=Serial[4];
        Form1->HW->Text=Hardware;
    }
    else
    {
        p=(unsigned char *)&Hardware;
        *p=Serial[1];
        *(p+1)=Serial[2];
        Form1->HW->Text=Hardware;
        p=(unsigned char *)&Gotas;
        *p=Serial[6];
        *(p+1)=Serial[5];
        *(p+2)=Serial[4];
        *(p+3)=Serial[3];
        Form1->GT->Text=Gotas;
        p=(unsigned char *)&Temperatura;
        *p=Serial[8];
        *(p+1)=Serial[7];
        t=(float)Temperatura*(float)12.5/(float)256;
        sprintf(temp,"%0.1f",t);
        Form1->TP->Text=temp;
    }
}
if (Form1->CheckBox2->Checked)
{
    TObject *b;
    t=(float)Temperatura*(float)12.5/(float)256;
    sprintf(temp,"%0.1f",t);
    if ((Lg!=Gotas)||((strcmp(temp,Lt)!=0)))
    {
        if (Lg!=Gotas)
        {
            Form1->Timer2->Enabled=false;
            Tmedio=Timer;
            Timer=0;
            Form1->Timer2->Enabled=true;
            Form1->Edit6->Text=Tmedio;
        }
        Lg=Gotas;
    }
}

```



```

        strcpy(Lt,temp);
        if (Form1->Edit2->Text.Length()>0)
        {
            Form1->Button5Click(b);
        }
    }
    t=(float)50*(float)Gotas/(float)1015;
    sprintf(temp,"%.2f",t);
    Form1->Edit4->Text=temp;
}
//Sleep(0);
}
//-----

void __fastcall TRead::Execute()
{
    DWORD dwBytesRead;
    FreeOnTerminate=true;
    //Sleep(0);
    while (1)
    {
        if (Terminated) return;
        ReadFile(hComm,InBuff,1,&dwBytesRead,NULL);
        InBuff[1]=0;
        if (dwBytesRead)
        {
            InBuff[dwBytesRead]=0;
            Synchronize(DisplayIt);
        }
        Sleep(0);
    }
}
//-----

```


APÊNDICE 4

PROGRAMAS DO SERVIDOR

CRIATAB_BIOTA.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remotelhost
/**
$dbNome = "hcom_ajax";
$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
**/

$db = mysql_connect("localhost", $dbuser, $dbpass) or die("Falha na conexão :
".mysql_error());
echo ("Conectado ao Mysql !!!<br>");

$elimina = "DROP TABLE $tbNome";

$criacao = "CREATE TABLE $tbNome ( "
    ." id INT NOT NULL AUTO_INCREMENT,"
    ." paciente TEXT,"
    ." hospital TEXT,"
    ." medico TEXT,"
    ." quarto INT,"
    ." hwserial INT,"
    ." temperatura DOUBLE,"
    ." gotas INT,"
    ." volini DOUBLE,"
    ." volres DOUBLE,"
    ." tempor CHAR(10),"
    ." flag INT,"
    ." PRIMARY KEY(id)";

echo (" $criacao<br>");

//$indice = "CREATE UNIQUE INDEX indCli ON $tbNome()";
$insert = "INSERT INTO $tbNome VALUES ('Paulo Denis Garcez da Luz','Sao
Lucas','Joao M.','32','11','37.5','15','1000','0','10:00','1'),"

```

```

        .("','Leticia Tiboni','HC','Sergio
D.','33','10','37.5','160','1000','0','10:00','1'),"
        .("','Max Schrappe','Vita','Carlos
Q.','34','9','37.5','750','1000','0','10:00','1'),"
        .("','Guilherme Souza e Silva','Cotolengo','Murilo
J.','35','8','37.5','15','900','0','10:00','1'),"
        .("','Paulo Jose Abatti','Vita','Luiz
J.','36','7','37.5','15','1500','0','10:00','1'),"
        .("','Luiz Nacamura Junior','HC','Jose
M.','37','6','37.5','15','800','0','10:00','1'),"
        .("','Fabio Luiz Bertotti','HC','Jose
M.','38','5','37.5','15','750','0','10:00','1'),"
        .("','Rogerio Guimaraes Ribeiro','Sao Lucas','Ricardo
N.','39','4','37.5','15','2000','0','10:00','1'),"
        .("','Luma de Oliveira','Vita','Paulo
L.','40','3','37.5','15','2500','0','10:00','1'),"
        .("','Ricardo Jose Leal','HC','Murilo
K.','41','2','37.5','15','350','0','10:00','0'),"
        .("','Milton Brero dos Campos','Cotolengo','Joaquim
M.','42','1','37.5','15','3000','0','10:00','0')";

```

```

$resDrop = mysql_db_query($dbName, $elimina);
$resCria = mysql_db_query($dbName, $criacao);
// $resIndx = mysql_db_query($dbName, $indice);
$resInse = mysql_db_query($dbName, $insert);

```

```

if ($resDrop > 0) echo "Tabela $tbNome eliminada.<br>";
else echo "Tabela $tbNome não existe.<br>";

```

```

if ($resCria > 0) echo "Tabela $tbNome criada.<br>";
else echo "Tabela $tbNome não pode ser criada.<br>";

```

```

//if ($resIndx > 0) echo "Índice da Tabela $tbNome criado.<br>";
//else echo "Índice da Tabela $tbNome não pode ser criado.<br>";

```

```

mysql_close();
?>

```

LOCALIZA_HARDWARE_AJAX.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remmotelhost

```

```

///  

$dbNome = "hcom_ajax";  

$tbNome = "pacientes";  

$dbpass = "biota";  

$dbuser = "hcom_lasd";  

///  

$_con=new mysqli("localhost",$dbuser,$dbpass,$dbNome);  

if (!$_con)  

{  

    echo "Não foi possível conectar ao MySQL. Erro #" .mysqli_connect_errno() . " : " .  

    mysqli_connect_error();  

    exit;  

}  

// Verificar se a Cesta Existe  

if ($_GET["hwserial"]==0)  

{  

    echo "ERRO 008 !!!";  

}  

else  

{  

    $_hwserial=$_GET["hwserial"];  

}  

$_docxml=new DOMDocument('1.0');  

$_resp=$_docxml->createElement('hardware');  

if (!empty($_GET["hwserial"]))  

{  

    $_sql="SELECT * FROM $tbNome WHERE hwserial = $_hwserial";  

    $_res=$_con->query($_sql);  

    if ($_res==FALSE)  

    {  

        $_err=$_docxml->createElement("erro");  

        $_tex=$_docxml->createTextNode("Erro na Busca do Paciente");  

        $_err->appendChild($_tex);  

        $_docxml->appendChild($_err);  

    }  

    else  

    {  

        while ($_row = $_res->fetch_assoc())  

        {  

            $_prod=$_docxml->createElement("hwserial");  

            $_cod=$_docxml->createElement("id");  

            $_id=$_docxml->createTextNode($_row["id"]);  

            $_cod->appendChild($_id);  

            $_prod->appendChild($_cod);  

            $_cod=$_docxml->createElement("nome");  

            $_id=$_docxml->createTextNode($_row["paciente"]);  

            $_cod->appendChild($_id);

```

```

        $_prod->appendChild($_cod);

        $_cod=$_docxml->createElement("volini");
        $_id=$_docxml->createTextNode($_row["volini"]);
        $_cod->appendChild($_id);
        $_prod->appendChild($_cod);

        $_resp->appendChild($_prod);
    }
}
}
$_docxml->appendChild($_resp);
header('Content-type: application/xml');
echo $_docxml->saveXML();
?>

```

INSERI_DADOS.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remotelhost
/**
$dbNome = "hcom_ajax";
$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
**/

$db = mysql_connect("localhost", $dbuser, $dbpass) or die("Falha na conexão :
".mysql_error());

if (empty($_GET["id"]))
{
    echo ("Erro ==> GET ID !!!<br>");
    exit(0);
}
$id=$_GET["id"];

if (empty($_GET["t"]))
{
    echo ("Erro ==> GET Temperatura !!!<br>");
    exit(0);
}

```

```

$temperatura=$_GET["t"];

/*
if (empty($_GET["g"]))
{
    echo ("Erro ==> GET Gotas !!!<br>");
    exit(0);
}
*/
$gotas=$_GET["g"];

if (empty($_GET["d"]))
{
    echo ("Erro ==> GET Data !!!<br>");
    exit(0);
}
$data=$_GET["d"];

if (empty($_GET["h"]))
{
    echo ("Erro ==> GET Hora !!!<br>");
    exit(0);
}
$horas=$_GET["h"];

if (empty($_GET["tr"]))
{
    echo ("Erro ==> GET Tempo Restante !!!<br>");
    exit(0);
}
$tempor=$_GET["tr"];

if (empty($_GET["va"]))
{
    echo ("Erro ==> GET Volume Atual !!!<br>");
    exit(0);
}
$tempor=$_GET["va"];

$criacao = "CREATE TABLE p$Sid ( "
    ." temperatura DOUBLE,"
    ." gotas INT,"
    ." data TEXT,"
    ." hora TEXT"
    .")";

$insert = "INSERT INTO p$Sid VALUES ('$t','$g','$d','$h')";

```

```

$altera = "update $tbNome set tempor='$tr', volres='$va', gotas = '$g', temperatura = '$t'
where id = $id";
//echo (" $id $temperatura $gotas $data $hora<br>");
//echo (" $criacao<br>");
//echo (" $insert<br>");
//echo (" $altera<br>");
$resCria = mysql_db_query($dbNome, $criacao);
$resInse = mysql_db_query($dbNome, $insert);
$resalte = mysql_db_query($dbNome, $altera);
mysql_close();
?>

```

AJAX.CSS

```

.base {
    font-weight: bold;
    color: black;
    height: 25px;
}
.aguarde {
    background-color: #dadada;
    color: Navy;
    text-decoration: blink;
}
.aviso5 {
    font-size: 14px;
    color: Navy;
    background-color: yellow;
    padding: 5px;
}
.aviso4 {
    font-size: 14px;
    color: Navy;
    padding: 5px;
    background-color: #ffff5b;
}
.aviso3 {
    font-size: 14px;
    color: Navy;
    background-color: #ffff95;
    padding: 5px;
}
.aviso2 {
    font-size: 14px;
    color: Navy;
    background-color: #ffffbf;
    padding: 5px;
}

```



```
.aviso1 {
    font-size: 14px;
    color: Navy;
    background-color: #ffffd2;
    padding: 5px;
}
.aviso0 {
    font-size: 14px;
    font-weight: bold;
    color: Navy;
    background-color: white;
    padding: 5px;
}
.produtoDesc {
    border: 1px solid Navy;
    font-size: 12px;
    color: white;
    background-color: Navy;
    padding: 2px;
}
.produto {
    border: 1px solid Navy;
    font-size: 12px;
    color: #3300cc;
    padding: 2px;
}
.f0 {
    background-color: #ffff00;
}
.f1 {
    background-color: #ffff33;
}
.f2 {
    background-color: #ffff66;
}
.f3 {
    background-color: #ffff99;
}
.f4 {
    background-color: #ffffcc;
}
.f5 {
    background-color: #ffffff;
}
.tabela {
    border: none;
    border-collapse: collapse;
}
.linha {
    color: black;
```

```

        border: 1px Solid Navy;
        padding: 3px;
    }
    .cabecalho {
        border: 1px solid Navy;
        background-color: Green;
        color: white;
        padding: 3px;
    }
    .monitorar {
        color: black;
        padding: 3px;
        text-decoration: none;
    }
    .monitorar:Hover {
        color: Navy;
        background-color: #00cccc;
        padding: 3px;
    }
    .tabelat {
        background-color: #00CC00;
        padding: 3px;
    }
    .tabelat:Hover {
        background-color: #CC0000;
        padding: 3px;
    }
    .cab_cesta {
        background-color: #ffcc00;
        color: Navy;
        font-weight: bold;
        padding: 3px;
        border: 1px Solid #ffcc00;
        border-collapse: collapse;
    }
    .linha_cesta {
        color: black;
        border: 1px Solid #ffcc00;
        padding: 3px;
    }
    .excluir {
        color: black;
        padding: 3px;
        text-decoration: none;
    }
    .excluir:Hover {
        color: red;
        padding: 3px;
        font-weight: bold;
        text-decoration: none;
    }

```

```

}
.produtoDescSoap {
    border: 1px solid #808080;
    font-size: 12px;
    color: white;
    background-color: #9d9d9d;
    padding: 2px;
}
.produtoSoap {
    border: 1px solid #808080;
    font-size: 12px;
    color: #990033;
    padding: 2px;
}

```

AJAX.JS

// Objeto AJAX para comunicação Assincrona com um servidor de aplicações WEB

```

function AJAX(url,metodo,params,processa,modo)
{
    this.url = url;
    this.metodo = (metodo) ? metodo : 'GET';
    this.params = (metodo=='GET') ? null : params;
    this.processaresultado = processa;
    this.Header = new Array();
    this.modos = (modo) ? modo : 'T';
    if (this.modos!='T'&&this.modos!='X')
    {
        this.modos = 'T';
    }
    this.conectar();
}

```

```

AJAX.prototype = {

```

```

    addHeader: function(h,v)
    {
        this.Header[h] = v;
    },

```

```

    delHeader: function(h)
    {
        delete(this.Header[h]);
    },

```

```

    setHeader: function()
    {
        if(this.httprequest==null) { return;}
    }
}

```

```

        for(h in this.Header)
        {
            this.httprequest.setRequestHeader(h,this.Header[h]);
        }
    },

conectar: function()
{
    if(this.url==undefined||this.url=="")
    {
        return;
    }
    this.httprequest = null;
    if (window.XMLHttpRequest)
    { // Mozilla, Safari,...
        this.httprequest = new XMLHttpRequest();
    }
    else if (window.ActiveXObject)
    { // IE
        try {
            this.httprequest = new ActiveXObject("Msxml.XMLHTTP");
        }
        catch (e)
        {
            try {
                this.httprequest = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e) {}
        }
    }
    if(this.httprequest!=null&&this.httprequest!=undefined)
    {
        var obj = this;
        this.httprequest.onreadystatechange = function()
        {
            obj.processaretorno.call(obj);
        }
        if(this.metodo==undefined||this.metodo=="")
        {
            this.metodo = 'GET';
        }
        this.httprequest.open(this.metodo,this.url, true);
        this.setHeader();
        this.httprequest.send(this.params);
    }
},

processaretorno: function()
{
    if(this.httprequest.readyState==4)

```

```

        {
            if(this.httprequest.status==200)
            {
                var resp=(this.modos=="T") ? this.httprequest.responseText :
this.httprequest.responseXML;
                if(this.processaresultado!=null)
                {
                    this.processaresultado(resp);
                }
            }
            else
            {
                document.write(resp);
            }
        }
        else
        {
            this.processaerro();
        }
    }
},

processaerro: function()
{
    alert(this.httprequest.status + '-' + this.httprequest.statusText + ':->' + this.url);
}
}

```

ALTERA.JS

```

var time = null;
var TEMPO_PADRAO = 100;
var TAMANHOPADRAO = 10;
var TAMANHOMAXIMO = 100;
var ultimabusca = "";
var cont = 0;

var num_cesta = 0;
var qde_produto = new Array();
var vlr_produto = new Array();
var total_cesta = 0;

function mostratexto(vo)
{
    var agora = new Date();
    if (time==null)
    {
        time = agora;
    }
    var dif = (agora-time);

```

```

cont=0;
monitoraTempo();
time=agora;
}

function monitoraTempo()
{
if (time!=null&&el.value!=ultimabusca)
{
    var agora = new Date();
    var dif = (agora-time);
    if (dif>=TEMPO_PADRAO)
    {
        // Processar a requisição
        selecionarPacientes();
        time=agora; // para não chamar novamente
        cont = 99;
        ultimabusca = el.value;
    }
}
cont++;
if (cont==1)
{
    setTimeout('monitoraTempo()',TEMPO_PADRAO);
}
else
{
    cont = 0;
}
}

function selecionarPacientes()
{
var desc=el.value;
var ajax=new AJAX();
var params='?nomep='+desc;
//var params='?descricao='+desc;
if (document.BUSCA.offline.checked)
{
    params=params + '&offline=N';
}
if (document.BUSCA.modos.checked)
{
    params=params + '&modo=T';
}
params=params + '&linhas=' + document.BUSCA.tamanho.value;
ajax.url='altera_pacientes_desc_ajax.php' + params;
ajax.modos='X';
ajax.processaresultado=function(xmlDoc)
{

```

```

Array('Paciente','Volume Ini.','Hardware','Quarto','*Alterar*');
s=document.getElementById("pacientes");
nos=xmldoc.getElementsByTagName('paciente');
var xy=0;

tab=document.createElement("table");

tbo=document.createElement("tbody");

// cabeçalho da tabela
document.createElement("tr");
{
td=document.createElement("td");
td.className='cabecalho';
tx=document.createTextNode(titulo[t]);
}
tbo.appendChild(tr);
if (xmldoc.hasChildNodes() && nos.length > 0)
{
for (var i=0; i<nos.length; i++)
{
var no=nos[i];
var tr=document.createElement("tr");

var td=document.createElement("td");
var
tx=document.createTextNode(no.childNodes[0].firstChild.nodeValue);
xy=no.childNodes[4].firstChild.nodeValue;
qde_nomes[xy]=no.childNodes[0].firstChild.nodeValue;
td.className='linha';
td.appendChild(tx);
tr.appendChild(td);
//Input 1
var td = document.createElement("td");
td.align = 'center';
qde_produto=new Array();
qde_nomes=new Array();
var titulo=new
var
var
s.innerHTML='';
s.style.display='none';
var
tab.className='tabela';
var
tab.appendChild(tbo);
var tr =
for (t in titulo)
var
var
var
td.appendChild(tx);
tr.appendChild(td);

```

```

document.createElement("input");
no.childNodes[4].firstChild.nodeValue;

no.childNodes[1].firstChild.nodeValue;

qde_produto[no.childNodes[4].firstChild.nodeValue] =
no.childNodes[1].firstChild.nodeValue;

    function(e) {

        qde_produto[this.name] = this.value;

    }

td.appendChild(qd);

'linha';

tr.appendChild(td);

//Input 2
var td = document.createElement("td");

document.createElement("input");
    xy=no.childNodes[4].firstChild.nodeValue;
    xy++;
    qd.name = xy

no.childNodes[2].firstChild.nodeValue;

= no.childNodes[2].firstChild.nodeValue;

    function(e) {

        qde_produto[this.name] = this.value;

    }

td.appendChild(qd);

'linha';

tr.appendChild(td);

//Input 3
var td = document.createElement("td");

```

var qd =
qd.name =
qd.size = 5;
qd.value =

qd.onkeyup =

td.className =

td.align = 'center';
var qd =

qd.size = 5;
qd.value =

qde_produto[xy]
qd.onkeyup =

td.className =

td.align = 'center';


```

document.createElement("input");
    xy=no.childNodes[4].firstChild.nodeValue;
    xy++;
    xy++;
    qd.name = xy

no.childNodes[3].firstChild.nodeValue;
= no.childNodes[3].firstChild.nodeValue;

    function(e) {

        qde_produto[this.name] = this.value;

    }

td.appendChild(qd);

'linha';

tr.appendChild(td);

td=document.createElement("td");

a=document.createElement("a");
    a.href='javascript:
Monitora('+no.childNodes[4].firstChild.nodeValue+')';

a.className='monitorar';

tx=document.createTextNode("Alterar");

a.appendChild(tx);

td.className='linha';

td.appendChild(a);

tr.appendChild(td);

tbo.appendChild(tr);

    }
    s.appendChild(tab);
    s.style.display='inline';
}
}
ajax.conectar();

```

var qd =

qd.size = 5;

qd.value =

qde_produto[xy]

qd.onkeyup =

td.className =

var

var

var

```

}

function verificaTamanho(v)
{
if (v.value<=0)
{
v.value=TAMANHOPADRAO;
}
else if (v.value>TAMANHOMAXIMO)
{
v.value=TAMANHOMAXIMO;
}
}

function Monitora(paciente)
{
// Envia os dados para o servidor
//http://10.1.1.82/Biota/alterapaciente.php?id=11&quarto=101&volini=1000&hwserial=50
var ajax = new AJAX();
var params = '?id='+paciente;
params = params + '&volini='+qde_produto[paciente];
params = params + '&hwserial='+qde_produto[paciente+1];
params = params + '&quarto='+qde_produto[paciente+2];
ajax.url = 'alterapaciente.php' + params;
ajax.modulo = 'X';
ajax.produto = paciente;
ajax.quantidade = 1;
ajax.processarResultado = function(xmlDoc) { }
var texto='Paciente: '+qde_nomes[paciente]+' I1='+qde_produto[paciente]+' I2='+qde_produto[paciente+1]+' I3='+qde_produto[paciente+2];
var e = document.getElementById('resposta');
var n = e.childNodes;
n[0].nodeValue = texto;
ajax.conectar();
/*
for (var i=0;i<30;i++)
{
document.write('<br>'+qde_produto['+i+']='+qde_nomes[i]);
}
*/
//var e=document.getElementById('resposta');
//var no=document.createElement('DIV');
//var t=document.createTextNode(texto);
//e.appendChild(no);
//no.appendChild(t);
/*
var tr = document.createElement("tr");
tr.setAttribute("id","resposta");
var td = document.createElement("td");
var tx = document.createTextNode(texto);

```

```

td.className = 'linha_cesta';
td.appendChild(tx);
tr.appendChild(td);
*/
/*
document.write('Paciente='+paciente+',Input1='+qde_produto[paciente]+'Input2='+qde_produto[paciente+1]+'Input3='+qde_produto[paciente+2]);
for (var i=0;i<30;i++)
{
    document.write('<br>'+qde_produto['+i+']='+qde_produto[i]);
}
var ajax = new AJAX();
var params = '?cesta=' + num_cesta;
params = params + '&produto='+produto;
params = params + '&quantidade='+qde_produto[produto];
ajax.url = 'cesta_compras_ajax.php' + params;
ajax.moda = 'X';
ajax.produto = produto;
ajax.quantidade = qde_produto[produto];
*/
}

```

LISTARID.JS

```

var time = null;
var TEMPO_PADRAO = 100;
var TAMANHOPADRAO = 10;
var TAMANHOMAXIMO = 100;
var ultimabusca = "";
var cont = 0;

```

```

var num_cesta = 0;
var qde_produto = new Array();
var vlr_produto = new Array();
var total_cesta = 0;

```

```

function mostratexto(vo)
{
    var agora = new Date();
    if (time==null)
    {
        time = agora;
    }
    var dif = (agora-time);
    cont=0;
    monitoraTempo();
    time=agora;
}

```

```

function monitoraTempo()
{
if (time!=null&&el.value!=ultimabusca)
{
    var agora = new Date();
    var dif = (agora-time);
    if (dif>=TEMPO_PADRAO)
    {
        // Processar a requisição
        selecionarPacientes();
        time=agora; // para não chamar novamente
        cont = 99;
        ultimabusca = el.value;
    }
}
cont++;
if (cont==1)
{
    setTimeout('monitoraTempo()',TEMPO_PADRAO);
}
else
{
    cont = 0;
}
}

function selecionarPacientes()
{
var desc=el.value;
var ajax=new AJAX();
var params='?nomep='+desc;
//var params='?descricao='+desc;
if (document.BUSCA.offline.checked)
{
    params=params + '&offline=N';
}
if (document.BUSCA.modos.checked)
{
    params=params + '&modo=T';
}
params=params + '&linhas=' + document.BUSCA.tamanho.value;
ajax.url='lista_pacientes_desc_ajax.php' + params;
ajax.modos='X';
ajax.processaresultado=function(xml)
{
    qde_produto=new Array();
    qde_nomes=new Array();
    var titulo=new
    Array('Paciente','Médico','Hospital','Quarto','*ID*');

```

```

s=document.getElementById("pacientes");
nos=xmldoc.getElementsByTagName('paciente');

tab=document.createElement("table");

tbo=document.createElement("tbody");
        // cabeçalho da tabela
document.createElement("tr");
        {
td=document.createElement("td");
td.className='cabecalho';
tx=document.createTextNode(titulo[t]);
        }
tbo.appendChild(tr);
if (xmldoc.hasChildNodes()&&nos.length>0)
{
    for (var i=0;i<nos.length;i++)
    {
        var no=nos[i];
        var tr=document.createElement("tr");
        for (var ii=0;ii<no.childNodes.length-1;ii++)
        {
            var td=document.createElement("td");
            var
tx=document.createTextNode(no.childNodes[ii].firstChild.nodeValue);
td.className='linha';
td.appendChild(tx);
        }
tr.appendChild(td);
        var
td=document.createElement("td");
        var
tx=document.createTextNode(no.childNodes[4].firstChild.nodeValue);
td.className='linha';

```



```

monitoraTempo();
time=agora;
}

function monitoraTempo()
{
if (time!=null&&el.value!=ultimabusca)
{
    var agora = new Date();
    var dif = (agora-time);
    if (dif>=TEMPO_PADRAO)
    {
        // Processar a requisição
        selecionarPacientes();
        time=agora; // para não chamar novamente
        cont = 99;
        ultimabusca = el.value;
    }
}
cont++;
if (cont==1)
{
    setTimeout('monitoraTempo()',TEMPO_PADRAO);
}
else
{
    cont = 0;
}
}

function monitoraPacienteTempo(paciente)
{
    Monitorapac(paciente);
    var url='monitoraPacienteTempo('+paciente+')';
    setTimeout(url,750);
}

function selecionarPacientes()
{
    var desc=el.value;
    var ajax=new AJAX();
    var params='?nomep='+desc;
    //var params='?descricao='+desc;
    if (document.BUSCA.offline.checked)
    {
        params=params + '&offline=N';
    }
    if (document.BUSCA.modos.checked)
    {
        params=params + '&modo=T';
    }
}

```

```

    }
    params=params + '&linhas=' + document.BUSCA.tamanho.value;
    ajax.url='lista_pacientes_desc_ajax.php' + params;
    ajax.modos='X';
    ajax.processaresultado=function(xmlDoc)
    {
        qde_produto=new Array();
        qde_nomes=new Array();
        var titulo=new
        Array('Paciente','Médico','Hospital','Quarto','*Monitorar*');
        var
        s=document.getElementById("pacientes");
        var
        nos=xmlDoc.getElementsByTagName('paciente');
        s.innerHTML="";
        s.style.display='none';
        var
        tab=document.createElement("table");
        tab.className='tabela';
        var
        tbo=document.createElement("tbody");
        tab.appendChild(tbo);
        // cabeçalho da tabela
        var tr =
        document.createElement("tr");
        for (t in titulo)
        {
            var
            td=document.createElement("td");
            td.className='cabecalho';
            var
            tx=document.createTextNode(titulo[t]);
            td.appendChild(tx);
            tr.appendChild(td);
        }
        tbo.appendChild(tr);
        if (xmlDoc.hasChildNodes()&&nos.length>0)
        {
            for (var i=0;i<nos.length;i++)
            {
                var no=nos[i];
                var tr=document.createElement("tr");
                for (var ii=0;ii<no.childNodes.length-1;ii++)
                {
                    var td=document.createElement("td");
                    var
                    tx=document.createTextNode(no.childNodes[ii].firstChild.nodeValue);
                    td.className='linha';

```



```

td.appendChild(tx);
                                }
                                var
td=document.createElement("td");
                                var
a=document.createElement("a");
                                var xy=no.childNodes[4].firstChild.nodeValue;
                                qde_nomes[xy]=no.childNodes[0].firstChild.nodeValue;
                                a.href='javascript:
Monitora(' + no.childNodes[4].firstChild.nodeValue + ');';

a.className='monitorar';
                                var
tx=document.createTextNode("Monitorar");

a.appendChild(tx);

td.className='linha';

td.appendChild(a);

tr.appendChild(td);

tbo.appendChild(tr);
                                }
                                s.appendChild(tab);
                                s.style.display='inline';
                                }
                                }
ajax.conectar();
}

function verificaTamanho(v)
{
if (v.value<=0)
{
v.value=TAMANHOPADRAO;
}
else if (v.value>TAMANHOMAXIMO)
{
v.value=TAMANHOMAXIMO;
}
}

function Monitora(paciente)
{
var params='?paciente='+paciente;
params = params +'&nome='+qde_nomes[paciente];

```

```

var url='janelam.php'+params;
window.open(url,paciente,'toolbar=no,location=no,status=no,scrollbar=no,resizable=yes,width
=650,height=90');
//document.write(url);
}

function Monitorapac(paciente)
{
// Envia os dados para o servidor
var ajax=new AJAX();
var params='?paciente=' + paciente;
ajax.url='monitorar_pacientes_ajax.php' + params;
ajax.modoo='X';
ajax.produto=paciente;
ajax.quantidade=qde_produto[paciente];
ajax.processaresultado=function(xmlDoc)
{
no=xmlDoc.getElementsByTagName('paciente')[0];
tbo=document.getElementById("monitorar");
item=document.getElementById(ajax.produto);
//document.write('text'+ajax.url);
//document.write(no.childNodes[0].firstChild.nodeValue);
{
tbo.removeChild(item);
}
tr=document.createElement("tr");
ii=0;ii<no.childNodes.length;ii++)
{
var td=document.createElement("td");
tx=document.createTextNode(no.childNodes[ii].firstChild.nodeValue);
td.className='linha_cesta';
{
td.align='right';
}
td.appendChild(tx);
/*
var td=document.createElement("td");

```

```

a=document.createElement("a");

ajax.produto + ');';

document.createTextNode(" Excluir ");
    a.appendChild(tx);

*/

/*

vlr=no.childNodes[4].firstChild.nodeValue;

*/

    }
ajax.conectar();
}

```

INDEX.HTML

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Biota / Lasd - Paulo D&ecirc;nis Garcez da Luz</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
.menuTopo {
    font-family: Tahoma;
    font-size: 14px;
    color: #B2B2B2;
}
.linkTopo:link {
    text-decoration:none;
    font-family: Tahoma;;
    font-size: 14x;
    color: #B2B2B2;
}
.linkTopo:active {
    text-decoration:none;

```

```

var

a.href='javascript: excluir(' +

a.className='excluir';
var tx =

td.className='linha_cesta';
td.appendChild(a);
tr.appendChild(td);

tbo.appendChild(tr);
// guarda o valor total deste item

var

var exp=\/.\/g;
vlr=vlr.replace(exp,"");
var exp=\/.\/g;
vlr=vlr.replace(exp,".");
vlr=parseFloat(vlr);
vlr_produto[ajax.produto]=vlr;

```

```

        font-family: Tahoma;;
        font-size: 14px;
        color: #B2B2B2;
    }
    .linkTopo:visited {
        text-decoration:none;
        font-family: Tahoma;;
        font-size: 14px;
        color: #B2B2B2;
    }
    .linkTopo:hover {
        text-decoration:none;
        font-family: Tahoma;;
        font-size: 14px;
        color: #644C22;
    }
</style>
</head>

<body leftmargin="0" topmargin="3" marginwidth="0" marginheight="0">

<table width="693" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#FFFFFF">
    <!--DWLayoutTable-->
    <tr>
        <td colspan="2" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
            <!--DWLayoutTable-->
            <tr>
                <td width="186" height="60" valign="top"><a
href="http://www.cpgei.cefetpr.br/%7Ebiota/"></a></td>
                <td width="17">&nbsp;</td>
                <td colspan="3" rowspan="2" valign="top"></td>
            </tr>
            <tr>
                <td rowspan="2" valign="top"><a href="http://www.lasd.cefetpr.br/"></a></td>
                <td height="6"></td>
            </tr>
            <tr>
                <td height="54"></td>
                <td width="33">&nbsp;</td>
                <td width="409" valign="top"><div align="center">
                    <p>
                        <strong><font color="#009999" size="3" face="Tahoma">Sistema de
                        Monitoramento dos Pacientes em Ambiente Hospitalar.</font></strong></p>
                    </div></td>
                <td width="48">&nbsp;</td>
            </tr>

```

```

</table>
</td>
</tr>
<tr>
<td width="184">&nbsp;</td>
<td width="509" rowspan="2" valign="top"><table width="100%" border="0"
cellpadding="0" cellspacing="0">
<!--DWLayoutTable-->
<tr>
<td width="509" height="540" valign="top"><iframe src="orientador.html"
name="iframe" width="100%" marginwidth="7" height="100%" marginheight="18"
align="left" scrolling="auto" frameborder="0"></iframe>      &nbsp;</td>
</tr>
</table>
</td>
</tr>
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="0" cellspacing="0"
background="imagens/imagem3.png">
<!--DWLayoutTable-->
<tr>
<td width="184" height="86">&nbsp;</td>
</tr>
<tr>
<td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
<!--DWLayoutTable-->
<tr>
<td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="orientador.html" target="iframe"
class="linkTopo"><strong>HOME</strong></a></div>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td height="23">&nbsp;</td>
</tr>
<tr>
<td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
<!--DWLayoutTable-->
<tr>
<td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="alterar.php" target="iframe" class="linkTopo"><strong>ALTERAR
DADOS</strong></a></div>
</td>
</tr>
</table>

```

```

        </td>
    </tr>
    <tr>
        <td height="24">&nbsp;</td>
    </tr>
    <tr>
        <td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
            <!--DWLayoutTable-->
            <tr>
                <td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="listarid.php" target="iframe"
class="linkTopo"><strong>CONSULTAR
                DADOS</strong></a></div>
            </td>
        </tr>
        </table>
    </td>
</tr>
    <tr>
        <td height="22">&nbsp;</td>
    </tr>
    <tr>
        <td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
            <!--DWLayoutTable-->
            <tr>
                <td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="incluir.php" target="iframe" class="linkTopo"><strong>INCLUIR
                PACIENTE</strong></a></div>
            </td>
        </tr>
        </table>
    </td>
</tr>
    <tr>
        <td height="23">&nbsp;</td>
    </tr>
    <tr>
        <td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
            <!--DWLayoutTable-->
            <tr>
                <td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="monitorar.php" target="iframe"
class="linkTopo"><strong>MONITORAR
                PACIENTE</strong></a></div>
            </td>
        </tr>
        </table>
    </td>

```

```

        </td>
    </tr>
    <tr>
        <td height="67">&nbsp;</td>
    </tr>
    <tr>
        <td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
            <!--DWLayoutTable-->
            <tr>
                <td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="codigof.html" target="_blank"
class="linkTopo"><strong>C&Oacute;DIGO
                FONTE</strong></a></div>
                </td>
            </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td height="23">&nbsp;</td>
    </tr>
    <tr>
        <td height="19" valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
            <!--DWLayoutTable-->
            <tr>
                <td width="184" height="19" valign="top" class="menuTopo"><div
align="center"><a href="xmldocs.html" target="iframe"
class="linkTopo"><strong>C&Oacute;DIGO
                XML</strong></a></div>
                </td>
            </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td height="114">&nbsp;</td>
    </tr>
</table>
</td>
</tr>
<tr>
    <td height="36" colspan="2" valign="top"><table width="100%" border="0"
cellpadding="0" cellspacing="0" background="imagens/imagen2.png">
        <!--DWLayoutTable-->
        <tr>
            <td width="270" height="7"></td>
            <td width="174"></td>

```

```

        <td width="249"></td>
    </tr>
    <tr>
        <td height="19"></td>
        <td valign="top">&copy;Copyright 2005 - 2007</td>
        <td></td>
    </tr>
    <tr>
        <td height="10"></td>
        <td></td>
        <td></td>
    </tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

ORIENTADOR.HTML

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Orientadores</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font face="Tahoma"></font><font
face="Tahoma">Orientador:<font size="4"><strong> Dr.
Paulo Jos&eacute; Abatti </strong></font></font></p>
<p><font face="Tahoma">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
face="Tahoma"></font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font

```



```

face="Tahoma"></font>&nbsp;&nbsp;&nbsp;<font face="Tahoma"></font>&nbsp;<font
face="Tahoma"></font>Co-orientador: <strong><font size="4">Dr. Luiz Nacamura
J&uacute;nior</font></strong></font></p>
</body>
</html>

```

ALTERAPACIENTE.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remotelhost
/*
$dbNome = "hcom_ajax";
$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
*/

if (empty($_GET["id"]))
{
    echo ("Erro ==> GET ID !!!<br>");
    exit(0);
}
$_id=$_GET["id"];

if (empty($_GET["quarto"]))
{
    echo ("Erro ==> GET Temperatura !!!<br>");
    exit(0);
}
$_quarto=$_GET["quarto"];

if (empty($_GET["hwserial"]))
{
    echo ("Erro ==> GET Gotas !!!<br>");
    exit(0);
}
$_hwserial=$_GET["hwserial"];

if (empty($_GET["volini"]))
{
    echo ("Erro ==> GET Gotas !!!<br>");
    exit(0);
}

```

```

    }
    $_volini=$_GET["volini"];

    $_docxml=new DOMDocument('1.0','utf-16');
    $_resp=$_docxml->createElement('allterar');
    $_att=$_docxml->createAttribute("id");
    $_xid=$_docxml->createTextNode($_GET["id"]);
    $_att->appendChild($_xid);
    $_resp->appendChild($_att);

    $res1 = mysql_connect("localhost", $dbuser, $dbpass) or die("Falha na conexão :
    ".mysql_error());

    if ($res1)
    {
        //altera hw antigo para zero
        $altera = "update $tbNome set hwserial = 0 where hwserial = '$_hwserial'";
        $res1 = mysql_db_query($dbNome, $altera);
        //altera hw para atual
        $altera = "update $tbNome set volini = '$_volini', hwserial = '$_hwserial', quarto =
        '$_quarto' where id = $_id";
        $res2 = mysql_db_query($dbNome, $altera);
        if ($res2)
        {
            $_prod=$_docxml->createElement("paciente");

            $_cod=$_docxml->createElement("resposta");
            $_id=$_docxml->createTextNode("alterado");
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_resp->appendChild($_prod);
            //      echo ("Paciente Altarado !!!");
        }
        else
        {
            $_prod=$_docxml->createElement("paciente");

            $_cod=$_docxml->createElement("resposta");
            $_id=$_docxml->createTextNode("erro");
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_resp->appendChild($_prod);
            //      echo("Erro de inclusão ".mysql_error()."\n");
        }
    }
    else
    {
        echo("Erro de conexão ".mysql_error()."\n");
    }

```

```

    }
    mysql_close($res1);
    $_docxml->appendChild($_resp);
    header('Content-type: application/xml');
    echo $_docxml->saveXML();
?>

```

INCPAC.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remotelhost
///*
$dbNome = "hcom_ajax";
$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
//*/

$res1 = mysql_connect("localhost", $dbuser, $dbpass) or die("Falha na conexão :
".mysql_error());

if ($res1 && $ednome!="" && $edhospital!="" && $edmedico!="" && $edquarto!="")
{
    $insert = "INSERT INTO $tbNome VALUES
(',$ednome',$edhospital',$edmedico',$edquarto',' ',' ',' ',' ')";
    $res2 = mysql_db_query($dbNome, $insert);
    if ($res2)
    {
        echo("Novo Paciente Adicionado !!!");
    }
    else
    {
        echo("Erro de inclusão !!!".mysql_error()."\n");
    }
}
else
{
    echo("Erro de conexão !!!".mysql_error()."\n");
}
mysql_close($res1);
?>

```

LISTA_PACIENTES_DESC.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remotelhost
/**
$dbNome = "hcom_ajax";
$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
**/

$_con=new mysqli("localhost",$dbuser,$dbpass,$dbNome);
if (!$_con)
{
    echo "Não foi possível conectar ao MySQL. Erro #" .
    mysqli_connect_errno() . " : " . mysqli_connect_error();
    exit;
}
$_docxml=new DOMDocument('1.0','utf-16');
$_resp=$_docxml->createElement('categoria');
$_att=$_docxml->createAttribute("id");
$_id=$_docxml->createTextNode($_GET["categoria"]);
$_att->appendChild($_id);
$_resp->appendChild($_att);
define('LIMITEPADRAO',10);
define('LIMITEMAXIMO',100);
if (!empty($_GET["nomep"]))
{
    $_flag = ($_GET["offline"]!="N" ? " AND flag > 0 " : "");
    $_limite = (int) (!empty($_GET["linhas"]) ? $_GET["linhas"] : LIMITEPADRAO);
    if ($_limite<0)
    {
        $_limite = LIMITEPADRAO;
    }
    elseif ($_limite>LIMITEMAXIMO)
    {
        $_limite = LIMITEMAXIMO;
    }
    $_descricao = "" . ($_GET["modo"]!="T" ? "" : "%") . $_GET["nomep"] . "%";

```

```

    $_sql="SELECT * FROM $tbNome WHERE paciente like {$_descricao} {$_flag}
ORDER BY paciente LIMIT {$_limite}";
    $_res=$_con->query($_sql);
    if ($_res===FALSE)
    {
        $_err=$_docxml->createElement("ERRO");
        $_tex=$_docxml->createTextNode("Erro na Busca dos Pacientes");
        $_err->appendChild($_tex);
        $_docxml->appendChild($_err);
    }
    else
    {
        while ($_row = $_res->fetch_assoc())
        {
            $_prod=$_docxml->createElement("paciente");

            $_cod=$_docxml->createElement("nomep");
            $_id=$_docxml->createTextNode($_row["paciente"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_desc=$_docxml->createElement("medico");
            $_d=$_docxml->createTextNode($_row["medico"]);
            $_desc->appendChild($_d);
            $_prod->appendChild($_desc);

            $_cod=$_docxml->createElement("hospital");
            $_id=$_docxml->createTextNode($_row["hospital"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_cod=$_docxml->createElement("quarto");
            $_id=$_docxml->createTextNode($_row["quarto"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_cod=$_docxml->createElement("idpaciente");
            $_id=$_docxml->createTextNode($_row["id"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_resp->appendChild($_prod);
        }
    }
    $_docxml->appendChild($_resp);
    header('Content-type: application/xml');
    echo $_docxml->saveXML();
?>

```

MONITORAR_PACIENTES_AJAX.PHP

```

<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remotelhost
/**
$dbNome = "hcom_ajax";
$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
**/

$_con=new mysqli("localhost",$dbuser,$dbpass,$dbNome);
if (!$_con)
{
    echo "Não foi possível conectar ao MySQL. Erro #" .mysqli_connect_errno() . " : " .
    mysqli_connect_error();
    exit;
}
// Verificar se a Cesta Existe
if ($_GET["paciente"]==0)
{
    echo "ERRO 007 !!!";
}
else
{
    $_id_paciente=$_GET["paciente"];
}
$_docxml=new DOMDocument('1.0');
$_resp=$_docxml->createElement('estadoatual');
$_att=$_docxml->createAttribute("id");
$_id=$_docxml->createTextNode($_id_paciente);
$_att->appendChild($_id);
$_resp->appendChild($_att);
if (!empty($_GET["paciente"]))
{
    $_sql="SELECT * FROM $tbNome WHERE id = $_id_paciente";
    $_res=$_con->query($_sql);
    if ($_res===FALSE)
    {
        $_err=$_docxml->createElement("ERRO");
    }
}

```

```

        $_tex=$_docxml->createTextNode("Erro na Busca do Paciente");
        $_err->appendChild($_tex);
        $_docxml->appendChild($_err);
    }
else
    {
        while ($_row = $_res->fetch_assoc())
        {
            $_prod=$_docxml->createElement("paciente");

            //      $_cod=$_docxml->createElement("nomepac");
            //      $_id=$_docxml->createTextNode($_row["paciente"]);
            //      $_cod->appendChild($_id);
            //      $_prod->appendChild($_cod);

            $_cod=$_docxml->createElement("hospital");
            $_id=$_docxml->createTextNode($_row["hospital"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_desc=$_docxml->createElement("medico");
            $_d=$_docxml->createTextNode($_row["medico"]);
            $_desc->appendChild($_d);
            $_prod->appendChild($_desc);

            $_cod=$_docxml->createElement("temperatura");
            $_id=$_docxml->createTextNode($_row["temperatura"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_cod=$_docxml->createElement("gotas");
            $_id=$_docxml->createTextNode($_row["gotas"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_cod=$_docxml->createElement("volumet");
            $_id=$_docxml->createTextNode($_row["volres"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_cod=$_docxml->createElement("temporestante");
            $_id=$_docxml->createTextNode($_row["tempor"]);
            $_cod->appendChild($_id);
            $_prod->appendChild($_cod);

            $_resp->appendChild($_prod);
        }
    }
}
$_docxml->appendChild($_resp);

```

```
header('Content-type: application/xml');
echo $_docxml->saveXML();
?>
```

INCLUIR.PHP

```
<html>
<head>
  <title>Incluir Pacientes !!!</title>
</head>
<form name="incpac" action="incpac.php" method="post">
<h1>Inclusão de Pacientes</h1>
<p><b>Nome:</b><br><input type="text" name="ednome" size=60 maxlength=100></p>
<p><b>Médico:</b><br><input type="text" name="edmedico" size=60
maxlength=100></p>
<p><b>Hospital:</b><br><input type="text" name="edhospital" size=60
maxlength=100></p>
<p><b>Quarto:</b><input type="text" name="edquarto" size=10
maxlength=10>
<input type="submit" value=" Incluir "></p>
</form>
</html>
```

ALTERAR.PHP

```
<html>
<head>
  <title>--- BIOTA / LASD * Paulo D. G. da Luz ---</title>
  <link rel="stylesheet" href="ajax.css" type="text/css">
  <script language="JavaScript" src="ajax.js"></script>
  <script language="JavaScript" src="altera.js"></script>
</head>
<body>
<form name="BUSCA">
  <table border=0 cellpadding=2>
    <tr>
      <h1>Alterar Dados do Paciente !!!</h1>
      <td>Máximo número de pacientes na busca: </td>
      <td><input name="tamanho" style="border: 1px Solid Navy;" size=3
maxlength=3
      onChange="verificaTamanho(this);"></td>
    </tr>
    <tr>
      <td>Considerar Monitoramento offline na Busca:</td>
      <td><input name="offline" type="checkbox" checked="true"></td>
    </tr>
    <tr>
      <td>Procurar em qualquer parte do Nome do paciente:</td>
      <td><input name="modo" type="checkbox"></td>
    </tr>
  </table>
<table border=0 cellpadding=2>
```



```

        <tr>
            <td colspan=2>
                <input name="texto" style="border: 1px Solid Navy;" size="50"
autocomplete="off" onKeyUp="mostratexto(this.value);"><bR>
                <div id="pacientes">
                </div>
            </td>
        </tr>
    </table>
    <br>
    <!-- Monitoramento -->
    <table border=0 cellspacing=0 cellpadding=0>
        <tbody>
            <tr>
                <td><b>Resposta:</b></td>
            </tr>
            <tr>
                <td><div id="resposta"> </div></td>
            </tr>
        <!--
            <tr>
                <th class="cabecalho">[Paciente]</th>
                <th class="cabecalho">[Hospital]</th>
                <th class="cabecalho">[Temperatura]</th>
                <th class="cabecalho">[Número-Gotas]</th>
                <th class="cabecalho">[Volume Total]</th>
                <th class="cabecalho">[Tempo Restante]</th>
                <th class="cabecalho">[Teste]</th>
            </tr>
        -->
    </tbody>
</table>
<table border=0 cellspacing=0 cellpadding=0 >
    <tr>
        <td id="resumo" height="35" valign="bottom"></td>
    </tr>
</table>
</form>
<script>
    document.BUSCA.tamanho.value=TAMANHOPADRAO;
    el=document.BUSCA.texto;
    el.focus();
</script>
</body>
</html>

```

LISTARID.PHP

```
<html>
```

```

<head>
    <title>--- BIOTA / LASD * Paulo D. G. da Luz ---</title>
    <link rel="stylesheet" href="ajax.css" type="text/css">
    <script language="JavaScript" src="ajax.js"></script>
    <script language="JavaScript" src="listarid.js"></script>
</head>
<body>
<form name="BUSCA">
    <table border=0 cellpadding=2>
        <tr>
            <h1>Consulta ID !!!</h1>
            <td>Máximo número de pacientes na busca: </td>
            <td><input name="tamanho" style="border: 1px Solid Navy;" size=3
maxlength=3
                                onChange="verificaTamanho(this);"></td>
        </tr>
        <tr>
            <td>Considerar Monitoramento offline na Busca:</td>
            <td><input name="offline" type="checkbox" checked="true"></td>
        </tr>
        <tr>
            <td>Procurar em qualquer parte do Nome do paciente:</td>
            <td><input name="modo" type="checkbox"></td>
        </tr>
    </table>
    <table border=0 cellpadding=2>
        <tr>
            <td colspan=2>
                <input name="texto" style="border: 1px Solid Navy;" size="50"
autocomplete="off" onKeyUp="mostratexto(this.value);"><bR>
                <div id="pacientes">
                </div>
            </td>
        </tr>
    </table>
    <br>
    <!-- Monitoramento -->
    <table border=0 cellspacing=0 cellpadding=0>
        <tbody>
<?php
//localhost
/*
$dbNome = "bd_ajax";
$tbNome = "pacientes";
$dbpass = "camelo77";
$dbuser = "root";
*/
//remmotelhost
///*
$dbNome = "hcom_ajax";

```

```

$tbNome = "pacientes";
$dbpass = "biota";
$dbuser = "hcom_lasd";
/**/
/*
if (empty($_GET["id"]))
{
    echo ("Erro ==> GET ID !!!<br>");
    exit(0);
}
$id=$_GET["id"];
*/
if ($data==""&&$hora=="")
{
    echo ("<form method=\"POST\">");
    echo ("<h1>Consulta Leituras !!!</h1>");
    echo ("<b>ID:</b><input type=\"text\" name=\"id\" size=5 maxlength=15>");
    echo ("<b>Data:</b><input type=\"text\" name=\"data\" size=15 maxlength=15>");
    echo ("<b>Hora:</b><input type=\"text\" name=\"hora\" size=15 maxlength=15>");
    echo ("<input type=submit value=\"Consulta\">");
    echo ("</form>");
}
else
{
    echo ("<form method=\"POST\">");
    echo ("<h1>Consulta Leituras !!!</h1>");
    echo ("<b>ID:</b><input type=\"text\" name=\"id\" size=5 maxlength=15 value=\"$id\">");
    echo ("<b>Data:</b><input type=\"text\" name=\"data\" size=15 maxlength=15
value=\"$data\">");
    echo ("<b>Hora:</b><input type=\"text\" name=\"hora\" size=15 maxlength=15
value=\"$hora\">");
    echo ("<input type=submit value=\"Consulta\">");
    echo ("</form>");

    $id=11;
    $_con=new mysqli("localhost",$dbuser,$dbpass,$dbNome);
    if (!$_con)
    {
        echo "Não foi possível conectar ao MySQL. Erro #".mysqli_connect_errno() . " : " .
mysqli_connect_error();
        exit;
    }
    $tbNome='p'.$id;
    $_sql="SELECT * FROM $tbNome where data='$data' and hora>='$hora'";
    $_res=$_con->query($_sql);
    if ($_res!=FALSE)
    {
        $flag=1;
        while ($_row=$_res->fetch_assoc())
        {

```

```

if ($flag==1)
{
    echo("<table>");
    echo("<tr>");
    echo("<th class=\"cabecalho\">Data</th>");
    echo("<th class=\"cabecalho\">Hora</th>");
    echo("<th class=\"cabecalho\">Temperatura</th>");
    echo("<th class=\"cabecalho\">Gotas</th>");
    echo("</tr>");
    $flag=2;
}
$d=$_row["data"];
$h=$_row["hora"];
$t=$_row["temperatura"];
$g=$_row["gotas"];
echo("<tr>");
echo("<td class=\"linha\" align=\"center\">$d</td>");
echo("<td class=\"linha\" align=\"center\">$h</td>");
echo("<td class=\"linha\" align=\"center\">$t</td>");
echo("<td class=\"linha\" align=\"center\">$g</td>");
echo("</tr>");
}
if ($flag==2) echo("</table>");
else echo("<br><br>Sem dados para esta data !!!");
}
}
?>
<!--

<tr>
    <td colspan=6>Pacientes Monitorados:</td>
</tr>
<tr>
    <th class="cabecalho">[Paciente]</th>
    <th class="cabecalho">[Hospital]</th>
    <th class="cabecalho">[Temperatura]</th>
    <th class="cabecalho">[Número-Gotas]</th>
    <th class="cabecalho">[Volume Total]</th>
    <th class="cabecalho">[Tempo Restante]</th>
    <th class="cabecalho">[Teste]</th>
</tr>

-->

</tbody>
</table>
<table border=0 cellpadding=0 cellspacing=0 >
    <tr>
        <td id="resumo" height="35" valign="bottom"></td>
    </tr>
</table>
</form>
<script>

```

```

        document.BUSCA.tamanho.value=TAMANHOPADRAO;
        el=document.BUSCA.texto;
        el.focus();
    </script>
</body>
</html>

```

MONITORAR.PHP

```

<html>
<head>
    <title>--- BIOTA / LASD * Paulo D. G. da Luz ---</title>
    <link rel="stylesheet" href="ajax.css" type="text/css">
    <script language="JavaScript" src="ajax.js"></script>
    <script language="JavaScript" src="monitora.js"></script>
</head>
<body>
<form name="BUSCA">
    <table border=0 cellpadding=2>
        <tr>
            <h1>Monitorar Paciente !!!</h1>
            <td>Máximo número de pacientes na busca: </td>
            <td><input name="tamanho" style="border: 1px Solid Navy;" size=3
maxlength=3
                                onChange="verificaTamanho(this);"></td>
        </tr>
        <tr>
            <td>Considerar Monitoramento offline na Busca:</td>
            <td><input name="offline" type="checkbox" checked="true"></td>
        </tr>
        <tr>
            <td>Procurar em qualquer parte do Nome do paciente:</td>
            <td><input name="modo" type="checkbox"></td>
        </tr>
    </table>
    <table border=0 cellpadding=2>
        <tr>
            <td colspan=2>
                <input name="texto" style="border: 1px Solid Navy;" size="50"
autocomplete="off" onKeyUp="mostratexto(this.value);"><bR>
                <div id="pacientes">
                </div>
            </td>
        </tr>
    </table>
    <br>
    <!-- Monitoramento -->
    <table border=0 cellspacing=0 cellpadding=0>
        <tbody>

```

```

<!--
        <tr>
            <td colspan=6>Pacientes Monitorados:</td>
        </tr>
        <tr>
            <th class="cabecalho">[Paciente]</th>
            <th class="cabecalho">[Hospital]</th>
            <th class="cabecalho">[Temperatura]</th>
            <th class="cabecalho">[Número-Gotas]</th>
            <th class="cabecalho">[Volume Total]</th>
            <th class="cabecalho">[Tempo Restante]</th>
            <th class="cabecalho">[Teste]</th>
        </tr>
-->
        </tbody>
    </table>
    <table border=0 cellpadding=0 cellspacing=0 >
        <tr>
            <td id="resumo" height="35" valign="bottom"></td>
        </tr>
    </table>
</form>
<script>
    document.BUSCA.tamanho.value=TAMANHOPADRAO;
    el=document.BUSCA.texto;
    el.focus();
</script>
</body>
</html>

```

JANELAM.PHP

```

<html>
<head>
<?php
$_id=$_GET["paciente"];
$_nome=$_GET["nome"];
echo("<title>[BIOTA / LASD, DALUZ] $_nome, IDJ=$_id</title>");
?>
    <link rel="stylesheet" href="ajax.css" type="text/css">
    <script language="JavaScript" src="ajax.js"></script>
    <script language="JavaScript" src="monitora.js"></script>
</head>
<?php
$_id=$_GET["paciente"];
echo("<body onLoad=\"monitoraPacienteTempo($_id);\">");
?>
    <table border=0 cellpadding=0 cellspacing=0>
        <tbody id="monitorar">

```

```
<!--      <tr>
          <td class="cabecalho">[Paciente]</td> -->
          <td class="cabecalho">[Hospital]</td>
          <td class="cabecalho">[Medico]</td>
          <td class="cabecalho">[Temperatura]</td>
          <td class="cabecalho">[Número-Gotas]</td>
          <td class="cabecalho">[Volume Total]</td>
          <td class="cabecalho">[Tempo Restante]</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)