

RICARDO EMANUEL VAZ VARGAS

**SISTEMA DE MONITORAMENTO REMOTO *ONLINE* DE SINAIS  
CARDÍACOS VIA REDE *GPRS***

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na área de concentração em Automação.

Orientador: Prof. Dr. Celso José Munaro

Co-orientador: Prof. Dr. Rodrigo Varejão Andreão

VITÓRIA  
2008

Dados Internacionais de Catalogação-na-publicação (CIP)  
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

---

V297s Vargas, Ricardo Emanuel Vaz, 1981-  
Sistema de monitoramento remoto online de sinais cardíacos  
via rede GPRS / Ricardo Emanuel Vaz Vargas. – 2008.  
114 f. : il.

Orientador: Celso José Munaro.  
Co-Orientador: Rodrigo Varejão Andreão.  
Dissertação (mestrado) – Universidade Federal do Espírito  
Santo, Centro Tecnológico.

1. Eletrocardiografia. 2. Sistemas de comunicação sem fio. I.  
Munaro, Celso José. II. Andreão, Rodrigo Varejão. III. Universidade  
Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

---

RICARDO EMANUEL VAZ VARGAS

**SISTEMA DE MONITORAMENTO REMOTO *ONLINE* DE SINAIS  
CARDÍACOS VIA REDE *GPRS***

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisição parcial para a obtenção do Grau de Mestre em Engenharia Elétrica - Automação.

Aprovada em 16 de junho de 2008.

COMISSÃO EXAMINADORA

---

**Prof. Dr. Celso José Munaro**  
Universidade Federal do Espírito Santo  
Orientador

---

**Prof. Dr. Rodrigo Varejão Andreão**  
Universidade Federal do Espírito Santo  
Co-orientador

---

**Prof. Dr. Anilton Salles Garcia**  
Universidade Federal do Espírito Santo

---

**Prof. Dr. Raimes Moraes**  
Universidade Federal de Santa Catarina

## AGRADECIMENTOS

Primeiramente aos meus pais, por terem me oferecido todo suporte necessário até então.

À minha namorada, pela paciência, pelos conselhos, pelo carinho e, principalmente, por entender os diversos momentos em que estive ausente.

Ao meu orientador, professor Celso Munaro, por toda sua atenção, orientação e dedicação dispensadas nesses últimos 2 anos e 5 meses.

Ao meu co-orientador, professor Rodrigo Varejão, principalmente pela fundamental orientação no campo da Telemedicina.

Ao professor Anilton Salles, pelas essenciais discussões relacionadas a telecomunicações.

Ao professor Paulo Amaral, pelas importantes consultorias (gratuitas) prestadas em relação à comunicação serial e à “bufferização” de dados.

Ao professor Hans Schneebeli, pelas importantes consultorias (gratuitas) prestadas em relação ao *hardware* utilizado nesta dissertação.

Ao professor Evandro Salles, por ter disponibilizado seu kit de desenvolvimento (*FPGA*), além do código-fonte utilizado na avaliação de desempenho do dispositivo móvel.

Ao professor Moisés Ribeiro, pela importante consultoria (gratuita) prestada em relação à implementação do Sistema Servidor.

Aos amigos, e mestres, Rodrigo Bonfá e Ádrian Drago, pelas importantes discussões técnicas sobre os aspectos positivos e negativos das diversas soluções tecnológicas possíveis.

Ao colega Ighor Fegali, por ter disponibilizado os resultados do seu trabalho de iniciação científica: esquema do banco de dados e interface *Web*.

Aos colegas de laboratório Emilio Bumachar e José Geraldo, pelo apoio técnico durante alguns momentos de dificuldade com o *hardware* e com a implementação do *firmware*.

Ao Adriano José e ao Bruno Zanetti, pelo apoio quase que diário durante esta empreitada.

Ao CNPq, pela bolsa de estudos.

Obrigado!

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>13</b>
1.1	MOTIVAÇÃO .....	14
1.2	JUSTIFICATIVAS .....	15
1.3	OBJETIVOS .....	16
1.4	ORGANIZAÇÃO DO DOCUMENTO .....	16
<b>2</b>	<b>ELETROCARDIOGRAMA .....</b>	<b>18</b>
2.1	A ANATOMIA DO CORAÇÃO.....	19
2.2	SISTEMA DE CONDUÇÃO ELÉTRICA DO CORAÇÃO .....	20
2.3	PRINCIPAIS TÉCNICAS DE REGISTRO.....	21
2.4	CARACTERÍSTICAS ELÉTRICAS DO SINAL CARDÍACO.....	24
2.5	CONCLUSÕES .....	25
<b>3</b>	<b>SISTEMA DE MONITORAMENTO PROPOSTO .....</b>	<b>26</b>
3.1	SISTEMA CLIENTE .....	27
3.1.1	Dispositivo Móvel.....	27
3.1.2	<i>Firmware</i> .....	34
3.2	SISTEMA SERVIDOR.....	48
3.2.1	<i>Hardware</i> do Sistema Servidor.....	48
3.2.2	<i>Software</i> do Sistema Servidor .....	49
3.3	CONCLUSÕES .....	53
<b>4</b>	<b>PROTOCOLO DE MONITORAMENTO REMOTO (PMR) .....</b>	<b>55</b>
4.1	MÉTODOS DO PMR .....	59
4.1.1	Métodos de Requisição .....	60
4.1.2	Métodos de Resposta .....	67
4.1.3	Métodos de Notificação .....	71
4.2	MÁQUINAS DE ESTADOS DO PMR.....	76
4.2.1	Máquina de Estados do PMR Cliente .....	76
4.2.2	Máquina de Estados do PMR Servidor .....	79
4.3	CONCLUSÕES .....	81
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS.....</b>	<b>82</b>
5.1	QUALIDADE DO SISTEMA DE MONITORAMENTO .....	83
5.1.1	Tempo Médio de Conexão .....	83
5.1.2	Atraso Médio de Amostra .....	87
5.2	QUALIDADE DA CONEXÃO <i>TCP</i> .....	96
5.2.1	Tempo Médio de Resposta.....	96
5.2.2	Taxa Média de Retransmissão .....	99
5.2.3	Taxa Média de Erro.....	101
5.3	QUALIDADE DO DISPOSITIVO MÓVEL .....	102
5.3.1	Taxa Média de Perda de Amostras.....	102

5.3.2	Exatidão do Relógio.....	105
5.3.3	Exatidão do Período de Amostragem.....	106
5.4	CONCLUSÕES .....	107
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>109</b>
6.1	TRABALHOS FUTUROS .....	110
<b>7</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>112</b>

## LISTA DE FIGURAS

Figura 1: eletrocardiógrafo montado por Willem Eighoven para estudo de eletrocardiogramas [12].	18
Figura 2: (a) anatomia do coração; (b) fluxo sanguíneo no sistema circulatório (adaptado do site <a href="http://www.unb.br">http://www.unb.br</a> ).	19
Figura 3: sistema de condução elétrica do coração [13].	20
Figura 4: relação entre as ondas do ECG e a atividade elétrica do coração [13].	21
Figura 5: sistema de derivações bipolares (adaptado do site <a href="http://www.medizinkabel.com">http://www.medizinkabel.com</a> ).	22
Figura 6: sistema de derivações unipolares dos membros (adaptado do site <a href="http://www.medizinkabel.com">http://www.medizinkabel.com</a> ).	23
Figura 7: sistema de derivações unipolares precordiais (adaptado do site <a href="http://www.medizinkabel.com">http://www.medizinkabel.com</a> ).	23
Figura 8: banda de frequência utilizadas em ECG [14].	24
Figura 9: visão geral do sistema de monitoramento proposto.	26
Figura 10: fotografia do protótipo desenvolvido.	28
Figura 11: fotografia do simulador de sinais vitais utilizado.	29
Figura 12: <i>Data Modem G24</i> [15].	30
Figura 13: fotografia da parte interna do <i>Data Modem G24</i> .	30
Figura 14: módulos de <i>hardware</i> de um dispositivo móvel.	30
Figura 15: diagrama de blocos da <i>MCU</i> HCS08GT60 [19].	31
Figura 16: Modem <i>GPRS</i> modelo G24 da Motorola [15].	33
Figura 17: representação da atuação do Modem <i>GPRS</i> como <i>gateway</i> de comunicação.	33
Figura 18: diagrama de contexto do <i>firmware</i> .	35
Figura 19: diagrama zero do <i>firmware</i> .	36
Figura 20: fluxograma do processo: Monitorar Botão de Pânico.	37
Figura 21: fluxograma do tratador de interrupção de conversão A/D.	39
Figura 22: acondicionamento de amostras, e seus canais, em uma PARTE.	41
Figura 23: <i>buffer</i> circular de PARTES.	42
Figura 24: fluxograma do tratador da interrupção de tempo que mantém o relógio interno à <i>MCU</i> .	44
Figura 25: fluxograma do tratador da interrupção de tempo que estabelece a frequência de amostragem.	45
Figura 26: arquitetura do <i>firmware</i> .	46
Figura 27: diagrama de contexto do <i>software</i> do Sistema Servidor.	49
Figura 28: sequência de criações de conexões e <i>threads</i> entre um Sistema Servidor e dois Sistemas Clientes.	51
Figura 29: esquema de banco de dados utilizado.	52
Figura 30: interface <i>Web</i> utilizada no projeto [20].	53
Figura 31: formato geral de uma mensagem PMR.	59
Figura 32: formato da mensagem PMR do tipo REQCNX.	62
Figura 33: formato da mensagem PMR do tipo REQRCNF.	66
Figura 34: formato da mensagem PMR do tipo RESCNXS.	68
Figura 35: formato das mensagens PMR do tipo RESCNXF e RESRCNFF.	70
Figura 36: formato da mensagem PMR do tipo RESRCNFS.	71
Figura 37: formato da mensagem PMR do tipo NOTBLCAMO.	73

Figura 38: formato da mensagem PMR do tipo NOTREQBLCAMO. ....	73
Figura 39: formato da mensagem PMR do tipo NOTALRMAC. ....	74
Figura 40: formato da mensagem PMR do tipo NOTALRMAT. ....	75
Figura 41: máquina de estados do PMR cliente. ....	78
Figura 42: máquina de estados do PMR servidor. ....	80
Figura 43: tempo médio de conexão por período do dia. ....	85
Figura 44: gráfico obtido com a ferramenta <i>multcompare</i> para o tempo médio de conexão por período do dia (grupo 2 selecionado aleatoriamente). ....	85
Figura 45: tempo médio de conexão por dia. ....	86
Figura 46: gráfico obtido com a ferramenta <i>multcompare</i> para o tempo médio de conexão por dia (grupo 3 selecionado aleatoriamente). ....	86
Figura 47: esquema de medição do atraso de amostra. ....	89
Figura 48: atraso médio de amostra em função do dia da semana e do tamanho do BLOCO. ....	92
Figura 49: gráfico obtido com a ferramenta <i>multcompare</i> para o atraso médio de amostra por dia. (grupo 2 selecionado aleatoriamente) ....	93
Figura 50: atraso médio de amostra em função do período do dia e do tamanho do BLOCO. ....	94
Figura 51: gráfico obtido com a ferramenta <i>multcompare</i> para o atraso médio de amostra por período do dia (grupo 1 selecionado aleatoriamente). ....	95
Figura 52: gráfico obtido com a ferramenta <i>multcompare</i> para o atraso médio de amostra – todos os dias e períodos (grupo 1 selecionado aleatoriamente). ....	96
Figura 53: tempo médio de resposta em função do dia da semana. ....	97
Figura 54: tempo médio de resposta em função do período do dia. ....	97
Figura 55: gráfico obtido com a ferramenta <i>multcompare</i> para o tempo médio de resposta por dia da semana (grupo 6 selecionado aleatoriamente). ....	98
Figura 56: gráfico obtido com a ferramenta <i>multcompare</i> para o tempo médio de resposta por período do dia (grupo 2 selecionado aleatoriamente). ....	98
Figura 57: taxa média de retransmissão em função do dia da semana. ....	100
Figura 58: taxa média de retransmissão em função do período do dia. ....	100
Figura 59: gráfico obtido com a ferramenta <i>multcompare</i> para a taxa média de retransmissão por dia da semana (grupo 1 selecionado aleatoriamente). ....	101
Figura 60: gráfico obtido com a ferramenta <i>multcompare</i> para a taxa média de retransmissão por período do dia (grupo 2 selecionado aleatoriamente). ....	101
Figura 61: taxa média de perda de amostras em função do dia da semana. ....	103
Figura 62: taxa média de perda de amostras em função do período do dia. ....	104
Figura 63: gráfico obtido com a ferramenta <i>multcompare</i> para a taxa média de perda de amostras por dia da semana (grupo 7 selecionado aleatoriamente). ....	104
Figura 64: gráfico obtido com a ferramenta <i>multcompare</i> para a taxa média de perda de amostras por período do dia (grupo 2 selecionado aleatoriamente). ....	105



## LISTA DE TABELAS

Tabela 1: codificação do campo MET.....	63
Tabela 2: codificação do campo VER.....	63
Tabela 3: codificação do campo TIP.....	63
Tabela 4: codificação do campo SIN.....	63
Tabela 5: codificação do campo AMO.....	63
Tabela 6: codificação do campo CAN.....	63
Tabela 7: codificação do campo BIT.....	63
Tabela 8: codificação do campo UNI.....	63
Tabela 9: codificação do campo MEN.....	63
Tabela 10: codificação do campo MAI.....	63
Tabela 11: codificação do campo FRE.....	64
Tabela 12: codificação do campo IDSIN.....	66
Tabela 13: codificação do campo IDPAR.....	66
Tabela 14: codificação do campo PAR.....	66
Tabela 15: codificação do campo REL.....	68
Tabela 16: codificação do campo CAM.....	70
Tabela 17: codificação do campo IDCAM.....	70
Tabela 18: cronograma dos experimentos de medição de tempo de conexão.....	84
Tabela 19: relações entre tamanho de BLOCO, tamanho da mensagem e número de amostras transmitidas em cada experimento de medição de atraso de amostra.....	87
Tabela 20: cronograma de realização dos experimentos de medição de atraso de amostra.....	88
Tabela 21: atrasos médios de amostras, desvios-padrão e coeficientes de variação entre resultados diários.....	93
Tabela 22: atrasos médios de amostras, desvios-padrão e coeficientes de variação entre resultados de cada período do dia.....	94

## RESUMO

Doenças cardiovasculares são consideradas atualmente a principal causa de morte e de lesões físicas limitantes e definitivas, que ocorrem em muitas das vezes em idade precoce. O exame de ECG é considerado, pelos especialistas, o padrão ouro para diagnóstico desse tipo de doença. Isso porque tal exame é barato, não-invasivo e fundamental para diagnóstico de diversas arritmias cardíacas. Portanto, o monitoramento de ECG pode significar a diferença entre o falecimento e a vida de muitos pacientes. Esta dissertação tem como objetivo o desenvolvimento de uma solução de monitoramento remoto, contínuo e *online*, baseada na tecnologia *GPRS*, de sinal de ECG. Essa tecnologia de comunicação se mostra adequada à solução e apresenta um ótimo custo-benefício, pois: mantém conexões sempre ativas, baseia-se em tarifação por volume de dados transferidos e provê grande área de alcance. A solução apresentada é composta por dispositivos móveis, mantidos por pacientes, que amostram 3 derivações de ECG e as transmitem, via rede *GPRS*, a um servidor instalado em um instituição de saúde. Para a realização desse tipo de transmissão, os dispositivos móveis filtram, amplificam e digitalizam os sinais amostrados. O servidor recebe as transmissões de diversos dispositivos móveis e persiste os sinais de ECG recebidos em um banco de dados, centralizando, portanto, informações médicas de diversos pacientes. Uma interface *Web* pode disponibilizar tais informações na Internet, possibilitando, assim, avaliações por especialistas a partir de qualquer ponto de acesso do planeta à grande rede. Tendo em vista a inexistência de um protocolo de comunicação aberto, padrão e adequado às necessidades do sistema proposto, um protocolo de camada de aplicação denominado PMR foi projetado e implementado. O PMR se mostrou versátil e útil a aplicações que transmitem informações de sinais analógicos diversos, não somente relacionados à Telemedicina. Um protótipo, contemplando *hardware* e *software*, foi desenvolvido para uma avaliação de desempenho da solução. Foram monitoradas diversas variáveis capazes de mensurar a qualidade da proposta apresentada. Os resultados obtidos evidenciam a viabilidade da solução.

## **ABSTRACT**

Nowadays heart diseases are considered the main cause of death and limiting and final injuries that happens in several times at early age. The ECG examination is considered by the specialists the default procedure to diagnose this type of disease. It is cheap, non-invasive, and necessary to diagnose many arrhythmias of the heart. Therefore, the ECG monitoring may mean the difference between death and life of several patients. The goal of this thesis was to develop a solution of remote, continuous and online ECG monitoring based on GPRS technology. This technology has been used in this kind of system and it presents excellent features: connections are always kept active; the rate is based on the volume of transferred data; and its covered area is large. The presented solution is composed of mobile devices, kept by patients, that capture 3 ECG leads and send them, through a GPRS network, to a server installed in a health institution. Before the transmission, a mobile device filters, amplifies and converts to digital the signals that have been captured. A server receives the transmissions from many mobile devices and save all ECG information in a data base. A Web interface can provide this information on the Internet. As any open communication protocol, suitable to the proposed solution, has not been found an application protocol called PMR was developed also. It has shown itself flexible and useful to applications that need to transfer information of analog signals in general, not just signals related to telemedicine. A prototype including hardware and software was developed to allow an assessment of the performance of the presented solution. Several parameters that can measure the quality of the proposal were monitored and the obtained results show that the solution is feasible.

# 1 INTRODUÇÃO

O principal propósito da Telemedicina é melhorar a qualidade dos serviços oferecidos na área de saúde, assim como minimizar seus custos, por meio de uma adequada transmissão de informações médicas entre pacientes e profissionais da saúde [1].

Nos últimos anos, têm-se realizado grandes investimentos em projetos de Telemedicina em diversos países do globo. As principais aplicações que tem sido desenvolvidas relacionam-se com:

- concentração de informações médicas de diversos pacientes em um único repositório [2];
- transmissão prévia de sinais vitais entre ambulâncias e hospitais [2];
- monitoramento de atletas (F1, natação, alpinismo, etc.) [2];
- monitoramento de pacientes de alto risco [3]; e
- hospitalização domiciliar de pacientes de baixo e médio risco, utilizando-se sistemas *TeleHomeCare* [4].

Diversas tecnologias de telecomunicações vêm sendo consideradas por essa gama de aplicações destinadas a monitoramento de sinais vitais diversos. As tecnologias que implementam comunicação sem fio são as mais buscadas, pois, na grande maioria dos casos, sua utilização propicia consideráveis benefícios a pacientes e instituições de saúde.

As tecnologias que mais se destacam atualmente na literatura são [1] [2] [3] [4] [5] [6] [7] [8] [9]:

- Internet;
- *Bluetooth* (padrão *IEEE 802.15*);
- *Zigbee* (padrão *IEEE 802.15.4*);
- *Wi-Fi* (padrão *IEEE 802.11*);
- *WAP* (*Wireless Application Protocol*);
- *UWB* (*Ultra Wide Band*);
- *HomeRF*;
- *Hiperlan*;
- *Narrowband*;

- *CDMA (Code Division Multiple Access)*;
- *GSM (Global System for Mobile Communications)*;
- *GPRS (General Packet Radio Service)*; e
- *SMS (Short Message Service)*.

De acordo com [6] e [7], os sistemas de monitoramento de sinais vitais podem ser classificados a partir de dois aspectos principais: área de alcance e modo de operação.

A área de alcance sempre está relacionada à tecnologia de telecomunicações utilizada. Com relação a esse aspecto, tais sistemas podem ser classificados como operados em ambientes internos ou externos. Sistemas operados em ambientes internos geralmente utilizam *links* cabeados ou *links* sem fio de pequeno alcance. Já os operados em ambientes externos utilizam, em sua maioria, *links* sem fio de grande alcance.

Com relação ao modo de operação, há sistemas que registram sinais vitais de forma contínua ou de forma orientada a evento. No segundo caso, os principais eventos utilizados são gerados por botões acionados pelo paciente e/ou algoritmos de detecção de arritmias.

Existem ainda, sistemas que são desenvolvidos para transmitirem informações médicas de forma *online* ou *offline*. Os sistemas *online* transmitem cada informação praticamente no momento em que essa é obtida. Já os sistemas *offline* registram tais informações para serem analisadas posteriormente. Nesse segundo caso, o tempo entre a obtenção das informações e sua análise por um especialista pode chegar a algumas horas ou até dias.

## 1.1 MOTIVAÇÃO

Doenças cardiovasculares constituem, na atualidade, a primeira causa de óbito no escopo mundial. Essa é uma verdade não somente em países em desenvolvimento, como o Brasil, mas também em países considerados desenvolvidos [9].

Nesse contexto, o eletrocardiograma (ECG) tem sido considerado o exame mais indicado, basicamente por ser barato, não-invasivo e fundamental no diagnóstico de diversas arritmias cardíacas. Por conta disso, o monitoramento de sinal de ECG tem sido amplamente empregado em sistemas de Telemedicina [3].

A motivação para este trabalho partiu da idéia de desenvolver um sistema que realize monitoramento de ECG de forma *online*, remota e contínua, que possa trazer reais benefícios a pacientes e instituições de saúde.

Do ponto de vista de um paciente, o aumento de sua qualidade de vida, por poder realizar suas atividades diárias enquanto é monitorado, é um benefício bastante expressivo. Já com relação a instituições de saúde, um dos diversos benefícios que podem ser citados é a capacidade de centralização de informações médicas de diversos pacientes monitorados.

## 1.2 JUSTIFICATIVAS

O principal exame baseado em eletrocardiograma chama-se ECG de repouso. Ele é realizado principalmente em consultórios médicos em poucos minutos. Contudo, sabe-se que arritmias cardíacas podem ocorrer, a qualquer momento, durante atividades diárias do paciente [8]. Portanto, em alguns casos é necessário monitorar o paciente de forma contínua.

Nesses casos, tipicamente utiliza-se um equipamento chamado de *Holter*. Ele realiza o monitoramento de sinais cardíacos de forma contínua e remota, porém *offline*. Com esse equipamento instalado junto ao corpo, o paciente pode retornar as suas atividades diárias normais, não sendo necessária sua permanência no consultório. Durante algumas horas de exame, tal equipamento registra continuamente, em sua memória interna, o sinal cardíaco do paciente. Após o término do exame, o *Holter* deve ser levado de volta ao consultório para que os registros armazenados sejam transferidos para outro equipamento, que auxiliará no diagnóstico do exame. O *Holter* mostra-se eficiente em determinados cenários, entretanto existem casos em que o monitoramento contínuo, remoto e *online* é necessário [8].

Casos típicos de pacientes que necessitam de monitoramento contínuo, remoto e *online* são os que já sofreram alguma desordem cardíaca. Esses pacientes apresentam riscos relativamente altos de morte repentina e muitos deles vivem em suas casas sem nenhum tipo de monitoramento [7]. Observa-se que essa abordagem propicia diversos benefícios, tanto a pacientes quanto a instituições de saúde. Alguns desses benefícios são:

- predições de anomalias cardíacas podem ser feitas analisando-se históricos de sinais de ECG;
- ações de resgate podem ser consideravelmente agilizadas em eventuais situações de emergência; e

- geração de certo sentimento de segurança aos pacientes, ao saberem que estão sendo monitorados de forma *online* e contínua, durante suas atividades diárias, tanto em ambiente externos quanto internos.

### 1.3 OBJETIVOS

Este trabalho tem como objetivo geral o desenvolvimento de uma solução de monitoramento, baseada na tecnologia *GPRS* (*General Packet Radio Service*), focada nos cenários em que o monitoramento remoto, contínuo e *online* mostra-se necessário e/ou vantajoso.

Para o desenvolvimento dessa solução, os seguintes objetivos específicos foram traçados:

- analisar o mercado em busca de um *hardware* microcontrolado e com um modem *GPRS*, que trouxesse o maior valor agregado possível à solução;
- projetar um protocolo de comunicação dedicado a transferências de sinais de ECG e notificações de alarmes entre diversos dispositivos móveis e uma central de monitoramento;
- projetar e implementar um *firmware*, a ser executado pelo *hardware* microcontrolado, dedicado a: amostrar e digitalizar sinal de ECG; gerenciar o Modem *GPRS*; e implementar o protocolo de comunicação projetado (lado do cliente);
- projetar e implementar um *software* servidor, a ser executado em uma máquina localizada em uma central de monitoramento, dedicada a: receber informações de sinais de ECG de diversos pacientes e armazenar tais informações em banco de dados; e
- realizar avaliação de desempenho das principais variáveis que refletem a qualidade da solução desenvolvida.

### 1.4 ORGANIZAÇÃO DO DOCUMENTO

O Capítulo 1 introduz o tema abordado nesta dissertação e apresenta suas motivações, objetivos e justificativas.

O Capítulo 2 apresenta, sucintamente, a fundamentação teórica relacionada ao ECG. A anatomia do coração humano é explorada, seu sistema de condução elétrica é brevemente

descrito, e as características mais relevantes do sinal elétrico capturado na superfície corporal são listadas. As principais técnicas de registro de ECG também são abordadas.

O sistema de monitoramento desenvolvido é apresentado no Capítulo 3. Sua primeira seção é destinada à descrição do Sistema Cliente. Já a sua segunda, e última, seção descreve o Sistema Servidor. Em ambas as descrições são abordados os aspectos tanto de *hardware* quanto de *software*.

No Capítulo 4, é feita uma descrição detalhada do Protocolo de Monitoramento Remoto (PMR). Esse é o protocolo de camada de aplicação, desenvolvido neste trabalho, que estabelece as regras para a transferência de dados entre Sistemas Clientes e Sistemas Servidores.

O Capítulo 5 descreve os diversos experimentos realizados para avaliação de desempenho dos principais aspectos da solução proposta. A metodologia e os resultados de cada experimento são devidamente descritos.

O Capítulo 6 traz as discussões técnicas e científicas relacionadas aos principais assuntos abordados nesta dissertação. Em seguida, são apresentadas algumas possibilidades concretas de trabalhos futuros, que podem ser realizados aproveitando-se das contribuições trazidas por esta dissertação.

Por fim, o Capítulo 7 lista a bibliografia utilizada em ordem alfabética.



## 2 ELETROCARDIOGRAMA

Um eletrocardiograma (ECG) é um registro contínuo de sinais elétricos do coração adquiridos através de eletrodos aplicados em pontos padronizados da superfície do corpo humano [10]. Muito embora um ECG registre sinais elétricos em sua essência, alterações no coração, do ponto de vista mecânico, também são determinantes no resultado de um ECG. Isso porque tais alterações influenciam de forma considerável a propagação dos sinais elétricos no coração [11].

O holandês Einthoven foi o primeiro pesquisador a propor o uso do ECG como método empírico de investigação clínica de seres humanos. A Figura 1 apresenta um eletrocardiógrafo por ele montado.

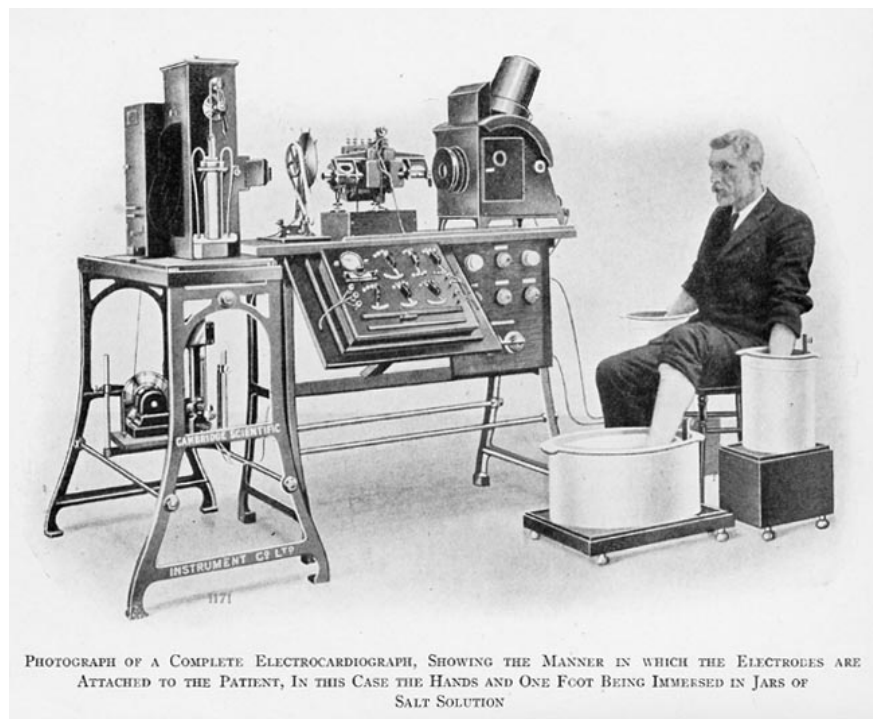


Figura 1: eletrocardiógrafo montado por Willem Einthoven para estudo de eletrocardiogramas [12].

Com o avanço da tecnologia, o ECG passou a ser obtido basicamente por equipamentos eletrônicos compostos por amplificadores transistorizados, filtros analógicos e dispositivos microprocessados.

Durante muitos anos, alterações na morfologia do ECG foram registradas e relacionadas a ocorrências de diversas doenças ligadas ao sistema cardíaco. Como consequência, o eletrocardiograma é na atualidade o recurso diagnóstico mais utilizado na avaliação de distúrbios do ritmo cardíaco [2].

## 2.1 A ANATOMIA DO CORAÇÃO

O coração é um órgão muscular oco em forma de cone composto por quatro câmaras, duas no lado direito e duas no lado esquerdo. Esse órgão localiza-se entre os pulmões e acima do diafragma. Sua função básica é vital ao ser humano: propiciar, por meio de bombeamento sanguíneo, nutrição e oxigenação a todas as células do organismo. A anatomia do coração e uma representação simplificada do fluxo sanguíneo no sistema circulatório humano são mostrados na Figura 2.

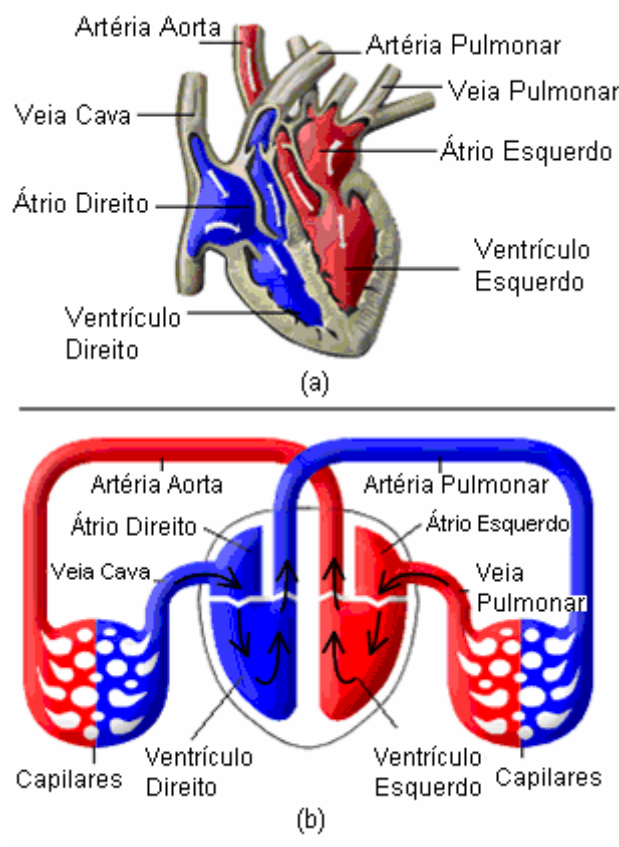


Figura 2: (a) anatomia do coração; (b) fluxo sanguíneo no sistema circulatório (adaptado do site <http://www.unb.br>).

Cada lado do coração é um sistema de bombeamento independente. O fluxo sanguíneo através do sistema de circulação humano realiza basicamente o seguinte percurso [2]:

- o sangue venoso (não-oxigenado) da circulação sistêmica chega ao coração, mais especificamente ao átrio direito, pela veia cava. Esse átrio, ao se contrair, bombeia o sangue até o ventrículo direito;
- o ventrículo direito, por sua vez, se contrai e bombeia a porção de sangue que se encontra em seu interior, pela artéria pulmonar, aos pulmões;

- no pulmão, ocorrem trocas gasosas de forma que o gás carbônico ( $\text{CO}_2$ ) presente no sangue é expelido e oxigênio ( $\text{O}_2$ ) é absorvido pelo sangue;
- o sangue, já oxigenado, retorna ao coração pelo átrio esquerdo e é bombeado ao ventrículo esquerdo; e
- enfim, o ventrículo esquerdo comprime-se, bombeando o sangue oxigenado de volta à circulação sistêmica.

## 2.2 SISTEMA DE CONDUÇÃO ELÉTRICA DO CORAÇÃO

Na Seção 2.1, as contrações dos átrios e ventrículos do coração, responsáveis pelo bombeamento sanguíneo, foram descritas. Em situações normais, tais contrações ocorrem em uma sequência bem ordenada. Isso acontece graças a uma atividade elétrica que também respeita uma sequência bem determinada.

A atividade elétrica do coração surge de potenciais elétricos provenientes de reações eletroquímicas em células conhecidas como excitáveis. Essas células formam o sistema de condução elétrica do coração (Figura 3).

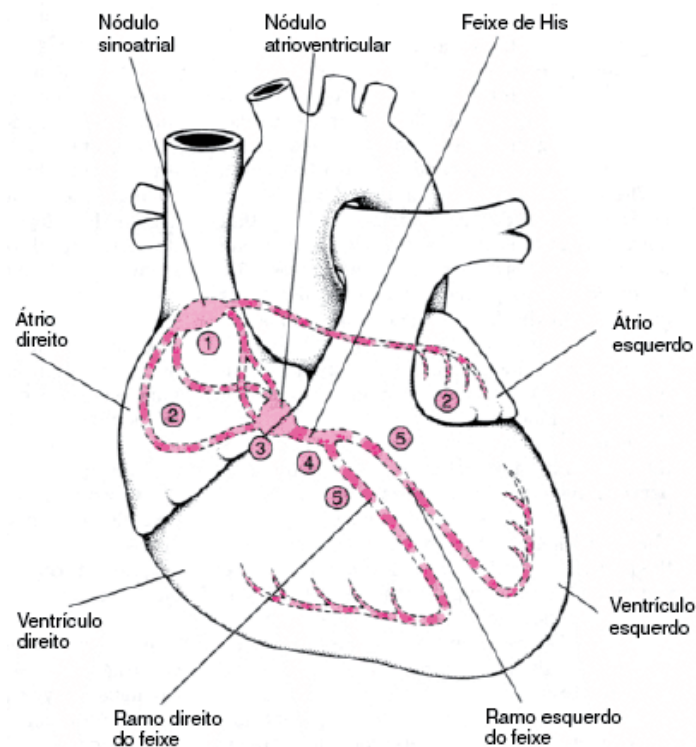


Figura 3: sistema de condução elétrica do coração [13].

Esse sistema controla o ritmo e a frequência dos batimentos cardíacos seguindo periodicamente as seguintes etapas (enumeradas na Figura 3): o nódulo sinoatrial (1) inicia

um impulso elétrico que flui sobre os átrios direito e esquerdo (2), fazendo essas câmaras se contraírem. Quando o impulso elétrico chega ao nódulo atrioventricular (3), ele sofre um ligeiro retardo. Em seguida, o impulso dissemina-se ao longo do feixe de His (4), o qual se divide em ramo direito (para o ventrículo direito) (5), e em ramo esquerdo (para o ventrículo esquerdo) (5). Por fim, o impulso é disseminado sobre os ventrículos, produzindo sua contração [13].

Durante um batimento cardíaco, a atividade elétrica em cada etapa supracitada compõe a forma de onda do sinal de ECG. Cada parte (onda) do ECG é designada por uma letra. A onda P representa a ativação dos átrios. O complexo QRS representa a ativação dos ventrículos. A onda T representa a onda de regularização dos ventrículos. A Figura 4 evidencia essa relação entre as ondas do ECG e a atividade elétrica do coração.

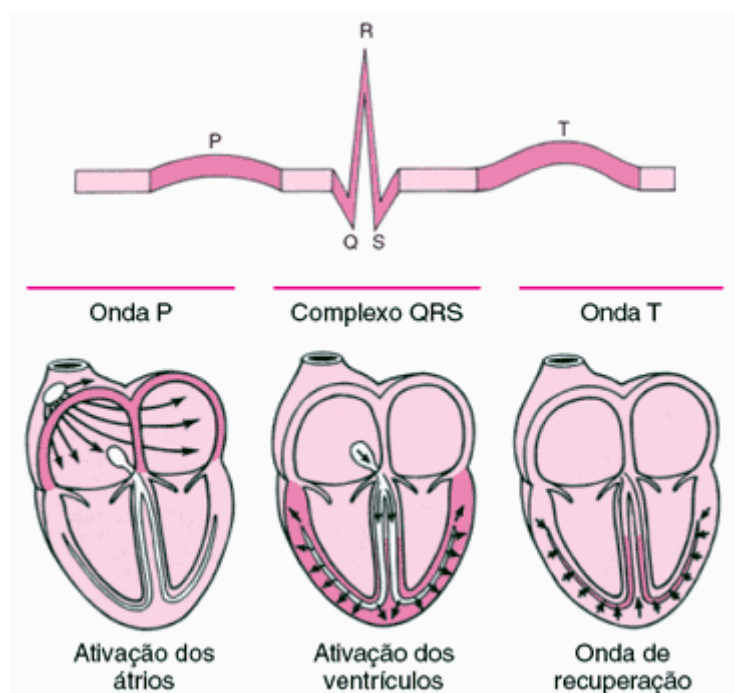


Figura 4: relação entre as ondas do ECG e a atividade elétrica do coração [13].

### 2.3 PRINCIPAIS TÉCNICAS DE REGISTRO

Atualmente, existem três técnicas de registro de eletrocardiograma muito utilizadas. Cada técnica apresenta suas vantagens, o que a faz adequada a determinados exames ou avaliações [11].

Das três técnicas, a mais conhecida é a ECG de repouso, também denominada ECG basal, ECG de superfície, ECG convencional ou, somente, eletrocardiograma. Essa técnica é

utilizada geralmente em consultórios médicos e com o paciente em repouso (geralmente deitado), como o próprio nome da técnica revela. São registradas 12 diferentes formas de onda, captadas em diferentes posições da superfície do corpo. Cada uma dessas formas de onda recebe o nome de derivação. Portanto, essa técnica se baseia no registro de 12 derivações [10].

Dessas 12 derivações, 3 são chamadas de clássicas ou bipolares (I, II e III) que registram a diferenças de potencial entre dois pontos da superfície corporal, conforme indicado na Figura 5 [2].

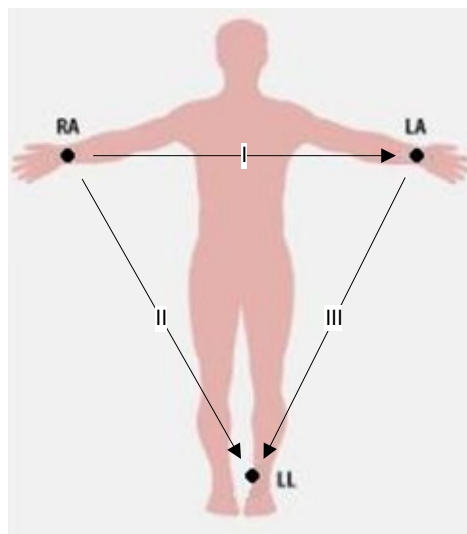


Figura 5: sistema de derivações bipolares (adaptado do site <http://www.medizinkabel.com>).

As outras 9 derivações são ditas unipolares, pois o potencial de cada derivação é medido a partir de um único ponto da superfície corporal, utilizando-se como referência um valor médio dos potenciais captados em LA, RA e LL (Terminal Central de Wilson). Quando um ponto da superfície corporal localizado em um dos membros é utilizado, a derivação é dita unipolar de membro (Figura 6). Quando um ponto da região torácica é explorado, a derivação é chamada de unipolar precordial (Figura 7) [2].

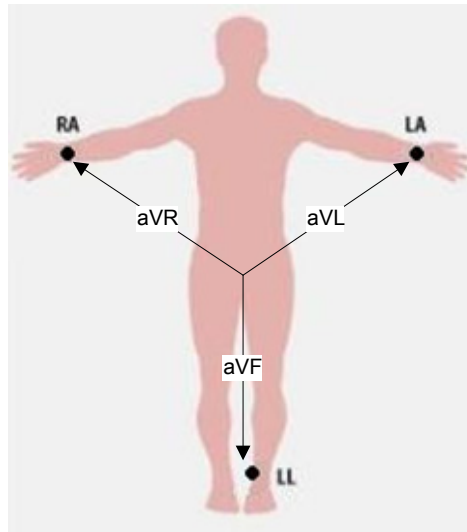


Figura 6: sistema de derivações unipolares dos membros (adaptado do site <http://www.medizinkabel.com>).

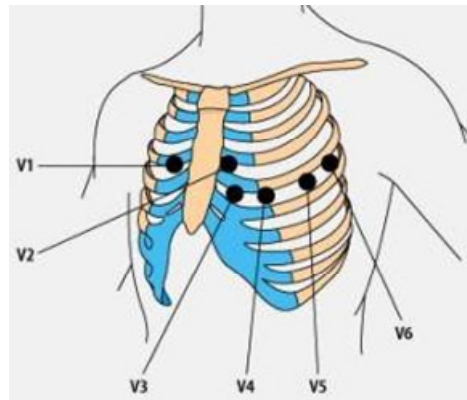


Figura 7: sistema de derivações unipolares precordiais (adaptado do site <http://www.medizinkabel.com>).

Outra técnica de registro utilizada é o vetorcardiograma (VCG). Nessa técnica, são utilizadas 3 derivações ortogonais de forma que um modelo vetorial tridimensional da atividade elétrica do coração é registrado. Ao utilizar essa técnica, muitos equipamentos comerciais permitem que a representação tridimensional da atividade elétrica do coração seja rotacionada, em qualquer sentido ou ângulo, facilitando a visualização de certos aspectos que podem ser camuflados em uma representação planar.

A terceira técnica de registro abordada neste texto é a ambulatorial, também chamada de ECG dinâmico. Essa é a técnica utilizada pelos sistemas do tipo *Holter*, que realizam monitoramento de ECG durante longos períodos, registrando uma, duas ou três das 12 derivações exploradas pelo ECG de repouso. Uma expressiva vantagem obtida com essa técnica é a maior possibilidade de registro de distúrbios cardíacos casuais, já que o monitoramento é realizado durante um período maior, podendo chegar a 72 horas. Uma

limitação dessa técnica é o fato de se monitorar uma menor quantidade de derivações, comparado à técnica de ECG de repouso.

Como o objetivo do sistema proposto é realizar o monitoramento de forma *online*, remota e contínua, essa última técnica, ECG ambulatorial, é adotada como referência. Tal escolha é fundamentada basicamente em dois pilares: (i) essa técnica é adequada ao monitoramento prolongado e (ii) minimiza a quantidade de dados a serem transmitidos e armazenados.

## 2.4 CARACTERÍSTICAS ELÉTRICAS DO SINAL CARDÍACO

A determinação da faixa de frequência dos sinais elétricos (derivações), obtidos a partir da superfície do corpo do paciente, depende do propósito dos registros obtidos durante um exame. São apresentadas na Figura 8, as três faixas de frequência geralmente mencionadas na literatura [14].

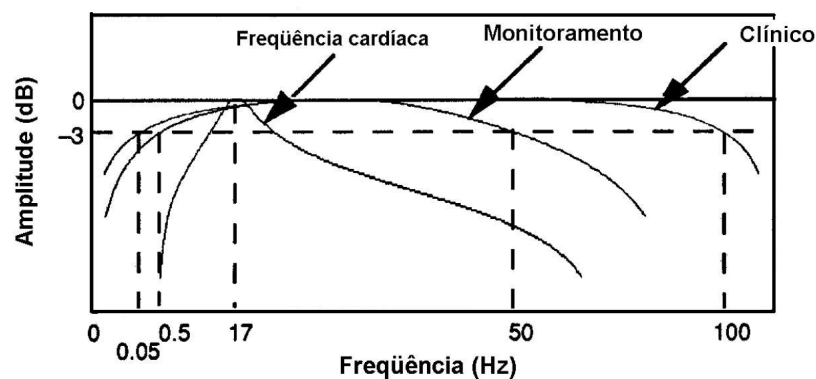


Figura 8: banda de frequência utilizadas em ECG [14].

A técnica de ECG convencional, ou ECG de repouso, adotada em sua maioria pelas clínicas médicas, é aplicada predominantemente utilizando-se equipamentos que implementam a faixa 0,05 a 100 Hz [14].

Já a técnica de ECG ambulatorial, ou ECG dinâmico, utilizada geralmente em monitoramentos prolongados, é associada, na maior parte das vezes, a equipamentos que implementam a faixa de 0,5 a 50 Hz. Nesse caso, utiliza-se uma faixa mais estreita de forma a atenuar ruídos de baixa frequência (menores que 0,5 Hz), causados pela movimentação dos eletrodos, e ruídos de maior frequência, devido a contrações musculares (o paciente geralmente não se encontra em repouso ao se utilizar essa técnica). Outro benefício

proveniente da limitação da faixa em 50 Hz é a eliminação de ruídos advindos da rede elétrica [14].

Cardiotacômetros – equipamentos de medição de frequência cardíaca – implementam filtros passa-faixa mais estreitos e centrados em 17 Hz. Essa faixa maximiza a relação sinal-ruído do complexo QRS, cujas ocorrências são utilizadas no cálculo da frequência cardíaca [2].

Com relação à amplitude do sinal elétrico do coração, o valor tipicamente observado na superfície do tórax é 1 mV. Esse valor pode variar entre indivíduos, dependendo também da posição de fixação do eletrodo, entre 50  $\mu$ V e 5 mV.

## 2.5 CONCLUSÕES

Atualmente, o ECG é tido como o padrão ouro para diagnóstico de diversas doenças do coração. Algumas características que o leva a essa condição são:

- não-invasivo;
- relativamente barato;
- fundamental na detecção de: ritmo, frequência, posição, tamanho das câmaras cardíacas [10];
- permite diagnóstico de: distúrbios da condução (áreas eletricamente inativas), isquemia miocárdica, distúrbios eletrofisiológicos do miocárdio, pericardiopatias, estimulação cardíaca artificial e alterações metabólicas eletrolíticas sistêmicas [10]; e
- auxilia na documentação de influências autonômicas e farmacológicas (tóxicas ou terapêuticas) [10].

Como doenças cardiovasculares constituem na atualidade a primeira causa de óbito no mundo, o monitoramento contínuo de ECG em indivíduos com problemas cardíacos pode ajudar a salvar muitas vidas.



### 3 SISTEMA DE MONITORAMENTO PROPOSTO

Conceitualmente, o sistema de monitoramento proposto foi modelado por meio de dois subsistemas: Sistema Cliente e Sistema Servidor. Ambos são compostos por elementos próprios de *hardware* e de *software*. O primeiro foi projetado para ser utilizado diretamente por pacientes. Já o Sistema Servidor foi concebido para ser instalado em instituições de saúde. A Figura 9 apresenta uma visão geral do sistema.

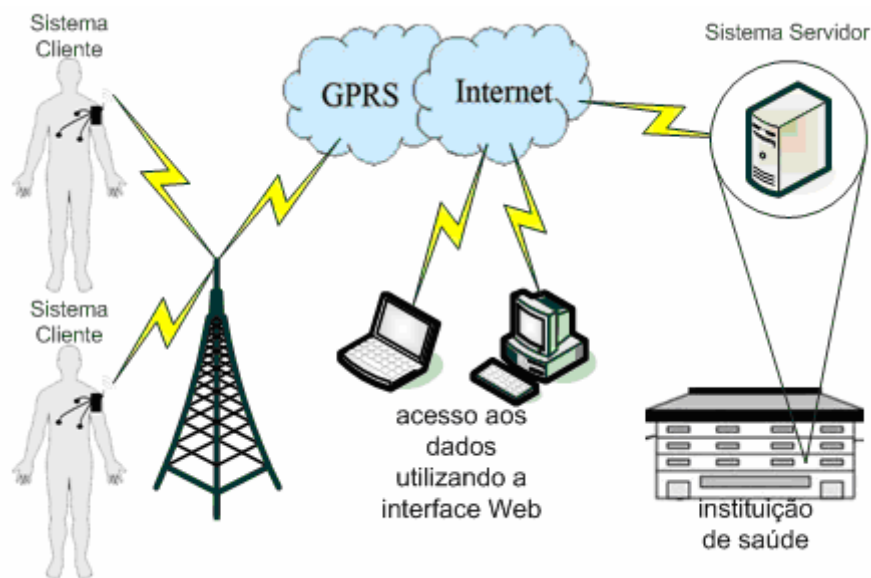


Figura 9: visão geral do sistema de monitoramento proposto.

É interessante ressaltar que o Sistema Servidor foi projetado para servir, simultaneamente, diversos Sistemas Clientes. Ou seja, o sistema de monitoramento foi pensado de forma que um único Sistema Servidor possa monitorar diversos pacientes de forma concorrente.

Foi desenvolvido também um protocolo de camada de aplicação, denominado PMR (Protocolo de Monitoramento Remoto), por meio do qual tais sistemas, Cliente e Servidor, se comunicam. Esse protocolo é detalhadamente descrito no Capítulo 4.

As duas primeiras seções deste capítulo descrevem, de forma geral, os componentes de *software* e de *hardware* envolvidos no sistema de monitoramento proposto. Além disso, são detalhadas as instâncias, de tais componentes, implementadas no protótipo que foi desenvolvido para realizações de experimentos.

### 3.1 SISTEMA CLIENTE

De forma geral, o propósito maior deste sistema é obter os dados monitorados (por exemplo, sinal de ECG e acionamentos de alarme) e transmiti-los a um Sistema Servidor. Ele é formado por componentes de *hardware*, que compõem um dispositivo móvel, e por componentes de *software* que compõem um *firmware*<sup>1</sup>. Tais componentes são descritos a seguir.

#### 3.1.1 Dispositivo Móvel

É proposto neste trabalho um dispositivo móvel, parte do Sistema Cliente, formado por três módulos de *hardware*, a saber: Pré-amplificador, Unidade de Microcontrolador (*MCU*) e Modem *GPRS*. Cada módulo provê funcionalidades específicas e fundamentais para o sistema de monitoramento proposto.

Um sinal cardíaco, na superfície do tórax de um ser humano, possui baixas amplitudes que tipicamente variam de 50 $\mu$ V a 5mV. Para que toda a faixa de conversão análogo-digital (A/D), realizada pela Unidade de Microcontrolador, possa ser aproveitada, maximizando assim sua precisão, esse sinal deve ser amplificado a uma faixa aceitável de trabalho. Essa primeira etapa é realizada pelo módulo Pré-amplificador.

Após ser amplificado e filtrado, um sinal cardíaco deve ser digitalizado e processado antes de ser enviado a um Sistema Servidor. A digitalização e o processamento desse sinal são realizados pela *MCU*. Essa unidade é utilizada também para verificar acionamentos de botões de alarmes. Por ser o único módulo programável presente no dispositivo móvel, sua programação deve contemplar, dentre outras atividades, o gerenciamento do Modem *GPRS*.

O Modem *GPRS* é um módulo de *hardware*, cujo *firmware* é pré-programado pelo fabricante. Sua atuação em um Sistema Cliente se dá como um *gateway* de comunicação<sup>2</sup> entre uma *MCU* e um Sistema Servidor. Toda comunicação digital entre a *MCU* e o Modem *GPRS* se dá por meio de comandos e respostas AT [15]. Já a comunicação entre um Sistema Servidor e um Modem *GPRS* é realizada através de uma pilha de protocolos *TCP/IP*. Os principais recursos disponibilizados por esse módulo a uma *MCU* são [15]: funcionalidades

---

<sup>1</sup> *Firmware* é um *software* que é armazenado em memória não-volátil (*ROM*, *EPROM*, *EEPROM*, *FLASH*, etc.) e que, portanto, tem seu conteúdo preservado mesmo quando a memória deixa de ser energizada.

<sup>2</sup> *Gateway* de comunicação é um equipamento que realiza conversão entre protocolos [28].

da pilha de protocolos *TCP/IP* e memória interna destinada a enfileiramento de dados a serem transmitidos a um Sistema Servidor.

Essas são as características básicas relacionadas ao *hardware* do dispositivo móvel proposto. A partir deste ponto, são descritos os principais aspectos acerca do protótipo desenvolvido para a realização dos experimentos de avaliação de desempenho do sistema de monitoramento projetado.

#### —PROTÓTIPO DESENVOLVIDO

A Figura 10 mostra o protótipo desenvolvido para a realização da fase de avaliação de desempenho do sistema de monitoramento proposto.



Figura 10: fotografia do protótipo desenvolvido.

Por conveniência, durante a fase de desenvolvimento desse protótipo, não se utilizou sinais de ECG reais, obtidos a partir de seres humanos. Foram utilizados sinais gerados por um simulador de sinais vitais. O simulador utilizado foi o *Handy Sim HS-14* (Figura 11) fabricado pela *R&D Mediq Ltda.* Com esse equipamento, é possível simular sinais de ECG normal, bradicardia e taquicardia.



Figura 11: fotografia do simulador de sinais vitais utilizado.

Sabe-se que a fase de desenho e desenvolvimento de *hardware* compromete, geralmente, grande parte do cronograma e do orçamento de projetos dessa natureza. Pensando nisso, foram buscados no mercado, e também em outros projetos científicos, equipamentos ou módulos de *hardware* que pudessem agregar valor ao projeto e, conseqüentemente, agilizar o desenvolvimento do protótipo.

#### — PRÉ-AMPLIFICADOR

Um pré-amplificador desenvolvido e utilizado no Projeto Telecárdio [16] mostrou-se adequado às necessidades do sistema de monitoramento proposto. De forma cordial e colaborativa, os coordenadores desse projeto cederam um de seus módulos sempre que necessário.

Esse módulo é basicamente composto por um circuito puramente analógico responsável por acondicionar o sinal de ECG a ser digitalizado. Seu circuito foi projetado para amplificar o sinal em 1.000 vezes e eliminar artefatos indesejados. Ele também utiliza filtro analógico do tipo passa baixa *Butterworth* de segunda ordem com frequência de corte de 50 Hz para eliminar a componente de frequência de 60 Hz da rede elétrica.

#### — DATA MODEM G24

Após uma ampla análise de mercado, um equipamento denominado *Data Modem G24* (Figura 12 e Figura 13), projetado pela empresa *Informat Technology* [17], foi elencado como o mais adequado ao projeto. Ele apresenta ótimo custo-benefício e atende praticamente todos os requisitos do sistema de monitoramento proposto. Uma única ressalva foi observada durante a

fase de avaliação de desempenho do sistema: capacidade de armazenamento em memória *RAM*. Na Subseção 5.3.1, esse aspecto é discutido e são apresentadas a sua origem e como o problema pode ser contornado.



Figura 12: *Data Modem G24* [15].



Figura 13: fotografia da parte interna do *Data Modem G24*.

Estão agregados nesse equipamento, os seguintes módulos: *MCU* e Modem *GPRS*. A Figura 14 apresenta as interconexões entre os três módulos de *hardware* que compõem o dispositivo móvel do protótipo desenvolvido.

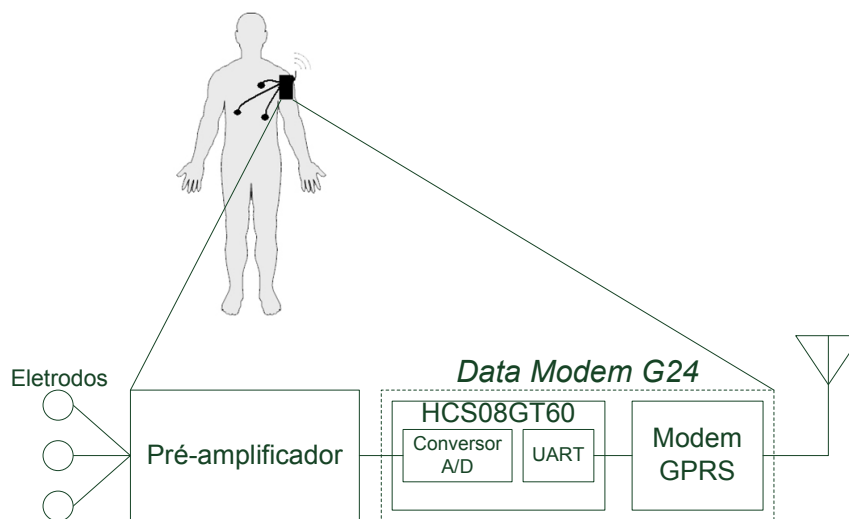
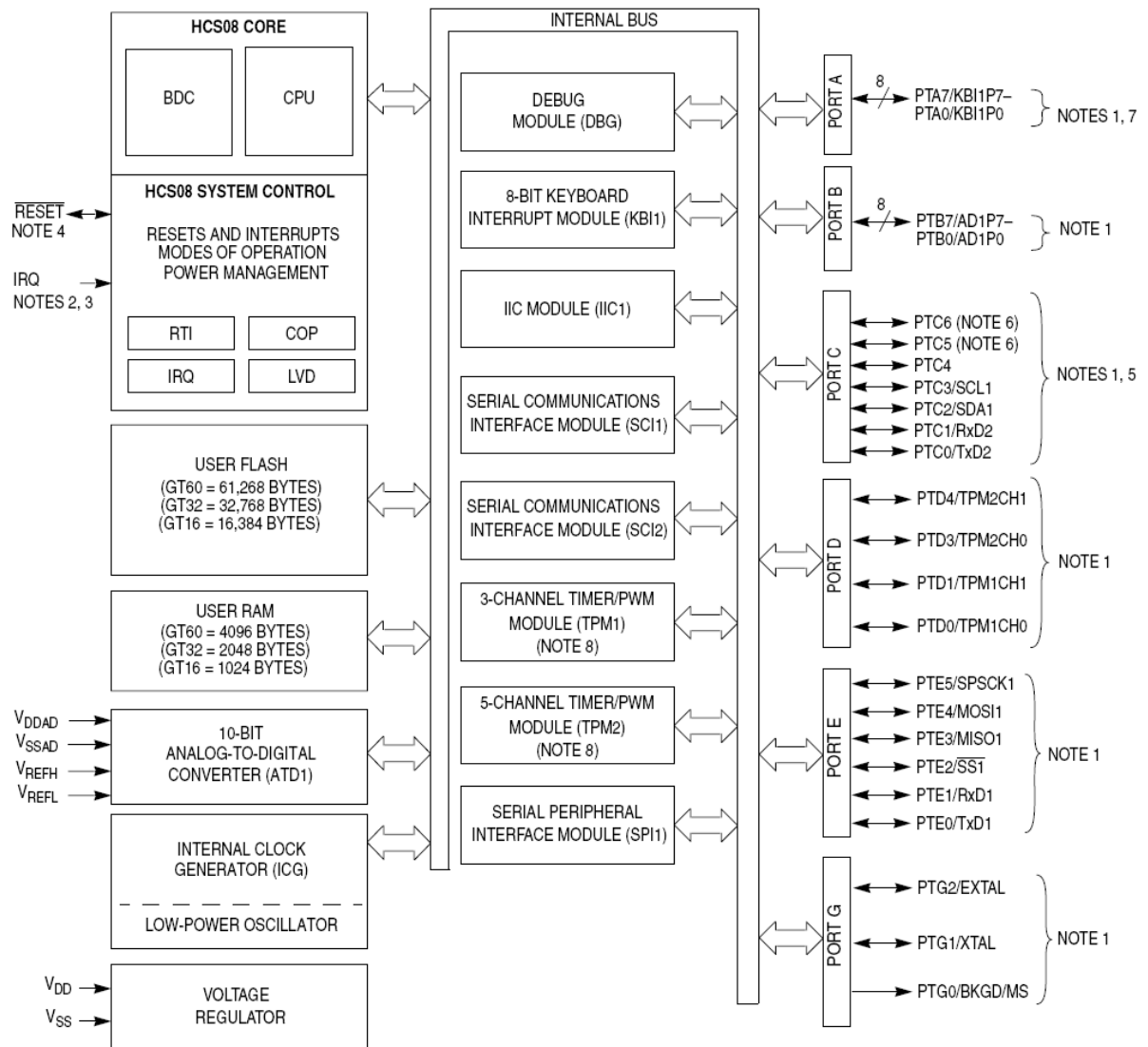


Figura 14: módulos de *hardware* de um dispositivo móvel.

#### — UNIDADE DE MICROCONTROLADOR

A *MCU* presente no equipamento adquirido é a HCS08GT60, fabricada pela *Freescale Semiconductor* [18]. Seu diagrama de blocos é apresentado na Figura 15.



## NOTES:

1. Port pins are software configurable with pullup device if input port.
2. Pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to V<sub>DD</sub>. IRQ should not be driven above V<sub>DD</sub>.
4. Pin contains integrated pullup device.
5. High current drive
6. PTC[6:5] are not available on the 42-pin SDIP package.
7. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown available when KBI enabled (KBIPn = 1).
8. Only two timer channels per TPM are bonded out. All channels are available for use as software compare.

Figura 15: diagrama de blocos da MCU HCS08GT60 [19].

Suas principais características e recursos são [19]:

- baixo custo;
- alto desempenho;
- processador de 8 bits, funcionando em até 40 MHz;
- sistema de depuração em *background*;
- 4 KB de memória *RAM*;
- 60 KB de memória *FLASH*;

- dois temporizadores de 16 *bits* cada;
- um conversor analógico-digital (A/D) de 10 *bits*, funcionando em uma frequência de conversão de 2 MHz; e
- duas interfaces de comunicação serial (*UARTs*).

Cada um desses recursos é utilizado para um propósito específico.

A memória *RAM* é utilizada, pela *MCU*, basicamente para armazenar e manipular as estruturas de dados que variam durante o funcionamento do Sistema Cliente. Em outras palavras, essa memória é dedicada aos segmentos de dados (*data segment*) e pilha (*stack segment*). Dos 4 KB disponíveis, 3.403 *bytes* são realmente utilizados.

Já a memória *FLASH* é utilizada para armazenar o código-fonte do *software* do Sistema Cliente. Dos 60 KB existentes, tal código-fonte consome 5.678 *bytes*. Esse *software* é descrito, em detalhes, na Subseção 3.1.2.

Um de seus temporizadores é utilizado para estabelecer a frequência de amostragem do sinal de ECG monitorado. O outro temporizador mantém um relógio interno ao dispositivo móvel.

Seu conversor A/D é utilizado na conversão dos três canais do sinal de ECG. Esse procedimento é realizado da seguinte forma: (i) cada canal é fisicamente conectado a uma porta de I/O específica (PTB0, PTB1 e PTB2) da *MCU*; (ii) primeiramente, uma das três portas é selecionada para conversão; (iii) em seguida, uma solicitação de conversão é feita; (iv) quando a conversão se completa, outra porta de I/O é selecionada para ser convertida. Esse processo se repete até que o sinal tenha sido amostrado, ou seja, tenham sido realizadas conversões de seus três canais.

Toda comunicação entre o Modem *GPRS* e a *MCU* é realizada por meio de uma de suas *UARTs*, que foi configurada para trabalhar com as seguintes características: 230.400 *bauds/s*, sem controle de fluxo, sem paridade e com formato de 8 *bits*. Faz-se necessário explicar que o controle de fluxo entre o Modem *GPRS* e a *MCU* é de fato realizado, porém não pelas suas respectivas *UARTs*, e sim por pinos de I/O interconectados, específicos para esse propósito. Já a segunda *UART* é fisicamente interligada a um conector DB9, sendo útil, portanto, na comunicação com diversos equipamentos externos, por exemplo, um microcomputador.

### — MODEM *GPRS*

O Modem *GPRS* modelo G24 da Motorola (Figura 16) está também presente no dispositivo móvel do protótipo desenvolvido. Como já mencionado, ele atua como um *gateway* de comunicação entre uma *MCU* e um Sistema Servidor. Por conseguinte, seu principal propósito é intermediar a comunicação entre esses dois equipamentos, que se utilizam de protocolos diferentes para se comunicarem.

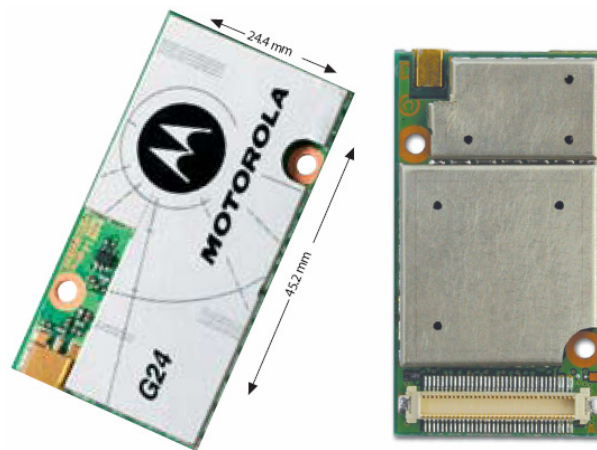


Figura 16: Modem *GPRS* modelo G24 da Motorola [15].

De certa forma, durante toda comunicação no sentido *MCU* - Sistema Servidor, esse *gateway* realiza conversões de comandos AT em segmentos *TCP*. Durante comunicações no sentido inverso, as conversões realizadas são de segmentos *TCP* em respostas AT. Mais detalhes sobre essas comunicações são apresentadas na Figura 17. É importante adiantar que descrições minuciosas sobre o *firmware* e o Núcleo do Sistema Servidor são, convenientemente, realizadas em subseções a seguir.

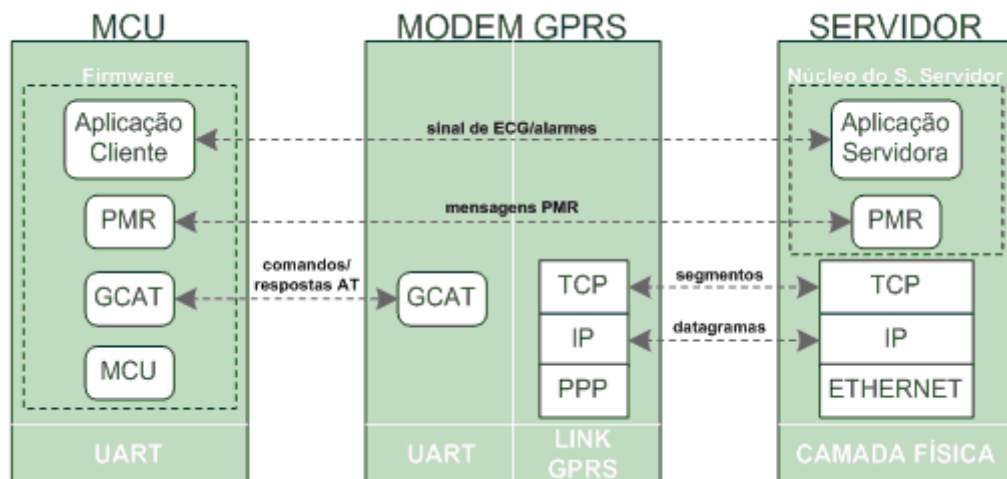


Figura 17: representação da atuação do Modem *GPRS* como *gateway* de comunicação.



As características mais relevantes desse modem são:

- pilha *TCP/IP* interna;
- tamanho físico reduzido;
- peso: 10,1 gramas;
- baixo consumo ( $< 2,5$  mA em modo *sleep*);
- duas interfaces de comunicação serial (*UARTs*);
- *multi-slot* classe 10; e
- *quad band*: 850/900/1.800/1.900 MHz.

### 3.1.2 *Firmware*

Esse é o *software* que foi projetado e implementado para ser executado pela *MCU*, de forma a gerenciar seus recursos e implementar todas as atividades essenciais<sup>3</sup> do Sistema Cliente. Ele ajuda a compor, portanto, o protótipo do sistema de monitoramento proposto.

Durante seu projeto, foram utilizadas ferramentas de análise de sistemas, em especial diagramas de fluxos de dados<sup>4</sup> e fluxogramas, de forma a auxiliar na identificação dos requisitos do sistema. Além disso, seus artefatos criados, ou seja, os próprios diagramas e fluxogramas, agilizaram consideravelmente a fase de implementação do *firmware*, que foi realizada com a linguagem ANSI C. As principais técnicas utilizadas no projeto da arquitetura de *software* foram a modularização e a parametrização. Essas são técnicas consolidadas em engenharia de *software* que propiciam principalmente flexibilidade, clareza e reuso.

Toda codificação do *firmware* foi realizada utilizando-se o *CodeWarrior Development Studio* versão 5.7.0, fornecido pela própria *Freescale Semiconductor*. Seus principais recursos que justificam sua utilização são:

- depuração *in-circuit* e em tempo real de execução;
- prática ferramenta de embarcação de código-fonte em diversos modelos de *MCU*; e
- conveniente ferramenta gráfica intitulada *Device Initialization*, que facilita consideravelmente a configuração inicial de diversos registros da *MCU*.

---

<sup>3</sup> Atividades essenciais são aquelas que devem ser realizadas independentemente da tecnologia utilizada. Essas atividades compõem a essência do sistema modelado [29].

<sup>4</sup> Diagrama de fluxo de dados é uma ferramenta de modelagem que permite imaginar um sistema como uma rede de processos funcionais, interligados por fluxos e depósitos de dados [30].

A descrição desse *software* é feita inicialmente de forma conceitual e com alto grau de abstração. Posteriormente, cada atividade essencial do Sistema Cliente é descrita em detalhes. Em seguida, ainda nesta subseção, é apresentada a arquitetura de *software* sob a qual esse *firmware* foi implementado. Essa forma de apresentação visa facilitar o entendimento do *firmware*.

A Figura 18 exibe o diagrama de fluxo de dados do *firmware* implementado. Por meio desse diagrama, são evidenciadas as entidades externas e como elas interagem com o *firmware*. Cabe ressaltar que as entidades externas são assim consideradas em relação ao *firmware*, e não ao sistema de monitoramento como um todo.

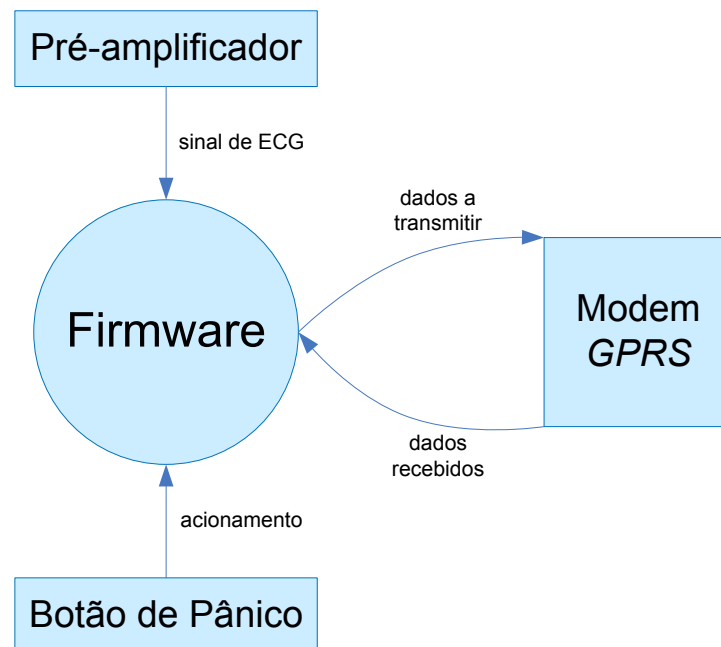
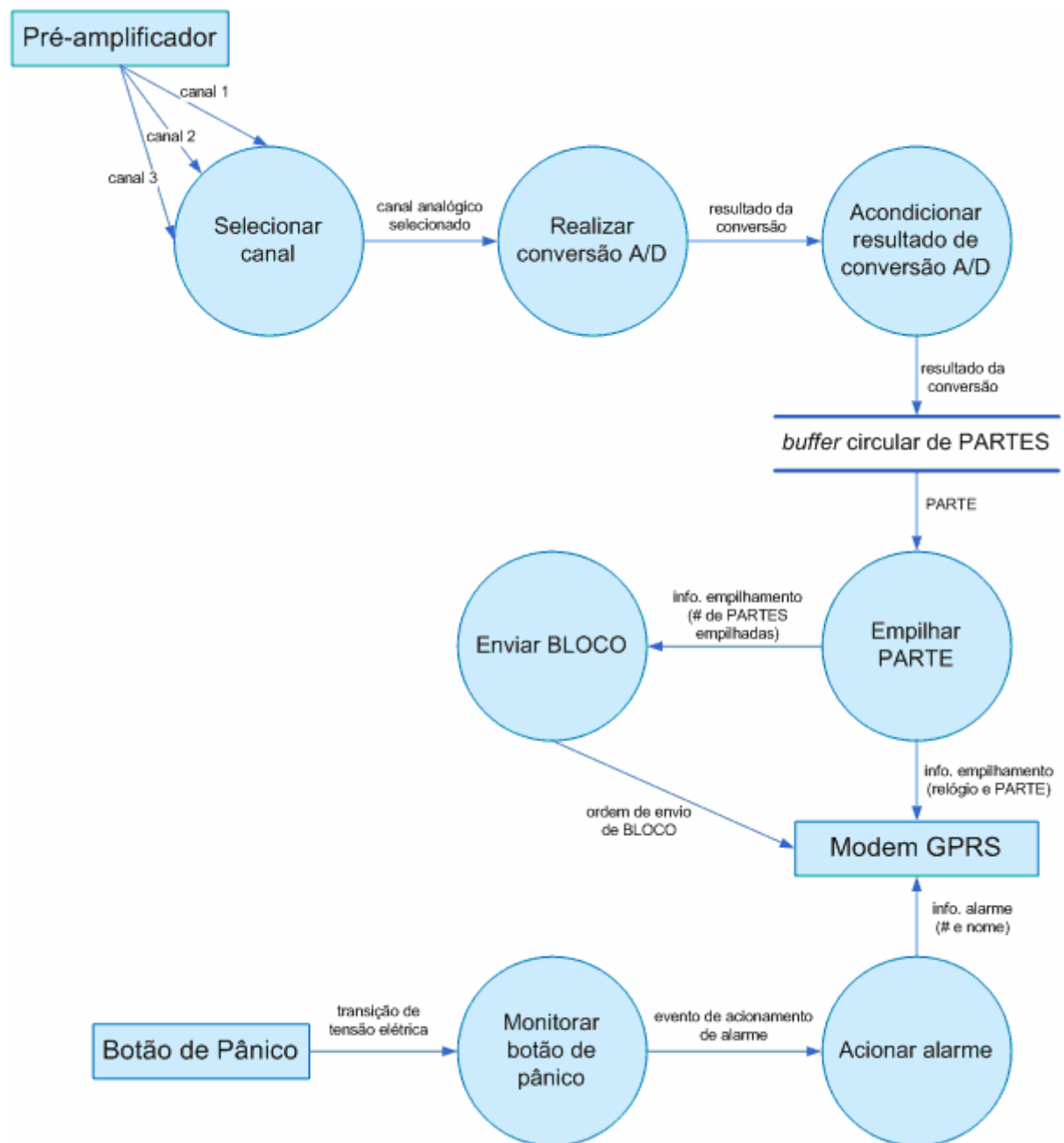


Figura 18: diagrama de contexto do *firmware*.

Ao se enriquecer em detalhes o diagrama de contexto, reduzindo assim o grau de abstração em que o *firmware* é observado, um diagrama de fluxo de dados específico é obtido (diagrama zero). Esse diagrama apresenta, dentre outros aspectos, as atividades essenciais, ou processos, do Sistema Cliente, que são representados graficamente por bolhas. Cada um desses processos é descrito a seguir. São descritos também os processos de controle que ajudam a integrar o Sistema Cliente: manter relógio interno e estabelecer frequência de amostragem. Tais processos de controle não são representados no diagrama zero do *firmware* (Figura 19).

Figura 19: diagrama zero do *firmware*.

### —MONITORAR BOTÃO DE PÂNICO

Monitoramentos dessa natureza podem ser realizados seguindo-se a técnica de *pooling*<sup>5</sup>, que, apesar de simples, em geral revela-se onerosa em termos de eficiência. Alternativamente, quando se busca eficiência e robustez, tais monitoramentos podem ser feitos por meio de

<sup>5</sup> *Pooling* é uma técnica de programação em que se realizam consecutivas verificações, por *software*, de um dado valor, por exemplo uma variável, até que esta seja alterada por um agente externo.

interrupções. Nesse contexto, optou-se em implementar este processo utilizando-se uma das interrupções em modo *input-capture* disponíveis na *MCU*.

O *firmware* foi desenvolvido de forma que um evento externo, produzido pelo botão de pânico do dispositivo móvel demande a execução do tratador associada a essa interrupção. Basicamente, tal rotina foi desenvolvida para reportar o acionamento do botão de pânico à rotina principal do *firmware* e, em seguida, limpar a *flag* de controle responsável pela indicação dessa interrupção. Essa sinalização entre o tratador da interrupção e a rotina principal é realizada por meio de uma *flag* específica (variável de *software*).

O fluxograma da rotina de tratamento da interrupção que implementa este processo é apresentado na Figura 20.

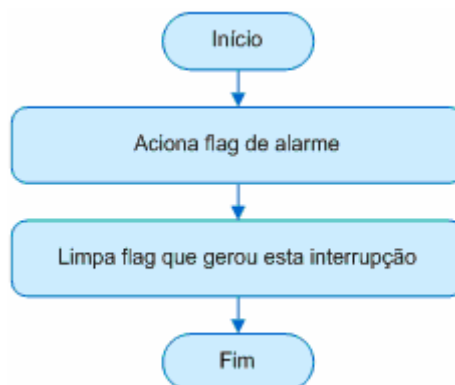


Figura 20: fluxograma do processo: Monitorar Botão de Pânico.

#### —ACIONAR ALARME

Este processo foi implementado por um conjunto de procedimentos que são executados de forma coordenada. De forma geral, a rotina principal do *firmware* encarrega-se de verificar o estado atual da *flag* específica que indica um acionamento do botão de pânico. Caso o estado atual dessa *flag* indique um acionamento de alarme, este processo é inicializado.

Uma vez inicializado este processo, procedimentos são executados para montar a mensagem PMR específica para o caso e transmiti-la a um Sistema Servidor através do Modem *GPRS*. Essa mensagem PMR contém, dentre outras informações, o nome do alarme acionado e um número que servirá como identificação futura para esse alarme.

### —REALIZAR CONVERSÃO A/D

A realização deste processo é feita pelo módulo de conversão A/D presente na *MCU*. É importante perceber que tais conversões são realizadas de forma concomitante a algum eventual processamento que esteja sendo realizado pela *MCU*, sem prejuízos para qualquer parte.

Ao final de cada conversão, uma interrupção é gerada e o tratador apresentado na Figura 21 é inicializado. Esse tratador, executado pela *MCU*, é responsável por:

- selecionar o canal ECG a ser convertido;
- inicializar o processo de acondicionar resultado de conversão A/D; e
- gerenciar *buffer* circular de PARTES.

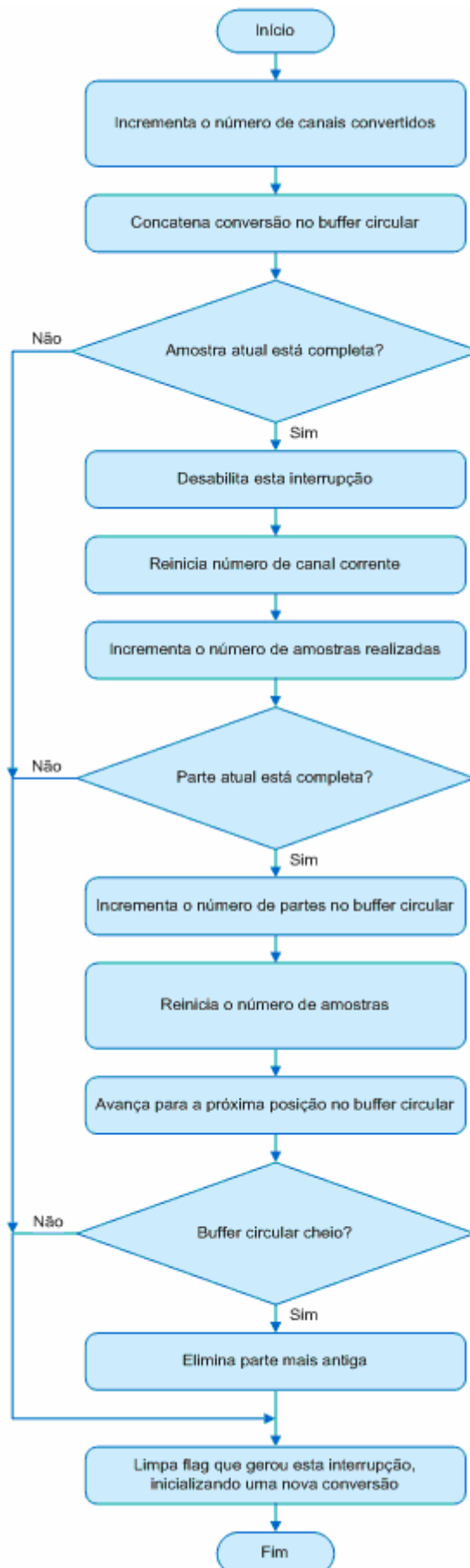


Figura 21: fluxograma do tratador de interrupção de conversão A/D.

### —SELECIONAR CANAL ECG

Como já foi mencionado na Subseção 3.1.1 (Unidade de Microcontrolador), a *MCU* utilizada no protótipo possui apenas um conversor A/D. Portanto, esse conversor deve ser utilizado nas conversões dos três canais que compõem o sinal de ECG. Nesse contexto, selecionar o canal a ser convertido é uma atividade essencial a ser realizada pelo *firmware*. Tal seleção é convenientemente realizada pelo próprio tratador de interrupção de conversão A/D (Figura 21).

Durante a inicialização da *MCU*, antes da ocorrência de qualquer conversão, o primeiro canal do sinal de ECG é selecionado. Após a conclusão da primeira conversão A/D, seu tratador é invocado e, então, o próximo canal é selecionado. Imediatamente a seguir, uma nova conversão é realizada e o terceiro e último canal é selecionado. Novamente, realiza-se a terceira conversão A/D, dando fim à geração de uma amostra de sinal ECG. Durante o tratamento da conversão do terceiro canal de cada amostra, o primeiro canal volta a ser selecionado, reiniciando assim este processo.

Cabe ressaltar que apesar das conversões dos canais dois e três ocorrerem imediatamente após a última conversão realizada, a primeira conversão somente se dá no início de cada período de amostragem. Esse é o mecanismo que garante a periodicidade da amostragem do sinal de ECG. Além disso, é garantido também que as conversões dos três canais sejam realizadas praticamente de forma simultânea.

### —ACONDICIONAR RESULTADO DE CONVERSÃO A/D

Como indicado na Figura 21, este processo é realizado por uma função específica chamada pelo tratador de interrupção de conversão A/D.

Cada conversão A/D gera um resultado expresso por um número de 10 *bits*, já que o conversor utilizado possui tal precisão. Portanto, são necessários 30 *bits* para expressar uma amostra de sinal de ECG composto por três canais. Para que diversas amostras sejam transmitidas a um Sistema Servidor, foram definidas de forma conveniente duas unidades abstratas de comunicação: PARTE e BLOCO. Toda transmissão de dados de sinal de ECG entre uma *MCU* e um Modem *GPRS* é realizada utilizando-se a unidade PARTE, que é composta por vinte amostras, ocupando, portanto, 75 *bytes* (30 *bits*/amostra\*20 amostras\*1/8 *byte/bits*). Já a transmissão de dados de sinal de ECG entre um Modem *GPRS* e um Sistema

Servidor é feita utilizando-se um BLOCO, outra unidade abstrata de comunicação. Um BLOCO de amostras pode ser composto de uma a 18 PARTES, ocupando, assim, entre 75 e 1.350 *bytes*. Esse limite superior é determinado pelo tamanho da memória, disponível no Modem *GPRS*, destinada ao enfileiramento de PARTES, ou seja, confecção de um BLOCO de amostras a serem transmitidas. O tamanho do BLOCO não foi fixado, pois um dos principais objetivos deste trabalho é estudar como esse tamanho influencia no comportamento da transmissão de dados através da rede *GPRS*.

Para que um conjunto de três conversões forme uma amostra de 30 *bits*, vinte amostras formem uma PARTE e diversas PARTES formem um BLOCO, cada resultado de conversão deve ser devidamente condicionado de forma a não permitir *bits* insignificantes entre quaisquer dois resultados. Ou seja, esse acondicionamento deve ser feito de forma que resultados sejam concatenados e salvos continuamente em memória. Esse é justamente o objetivo central deste processo.

A Figura 22 e a Figura 23 demonstram como os canais, amostras e PARTES são acondicionados na memória interna à *MCU*.

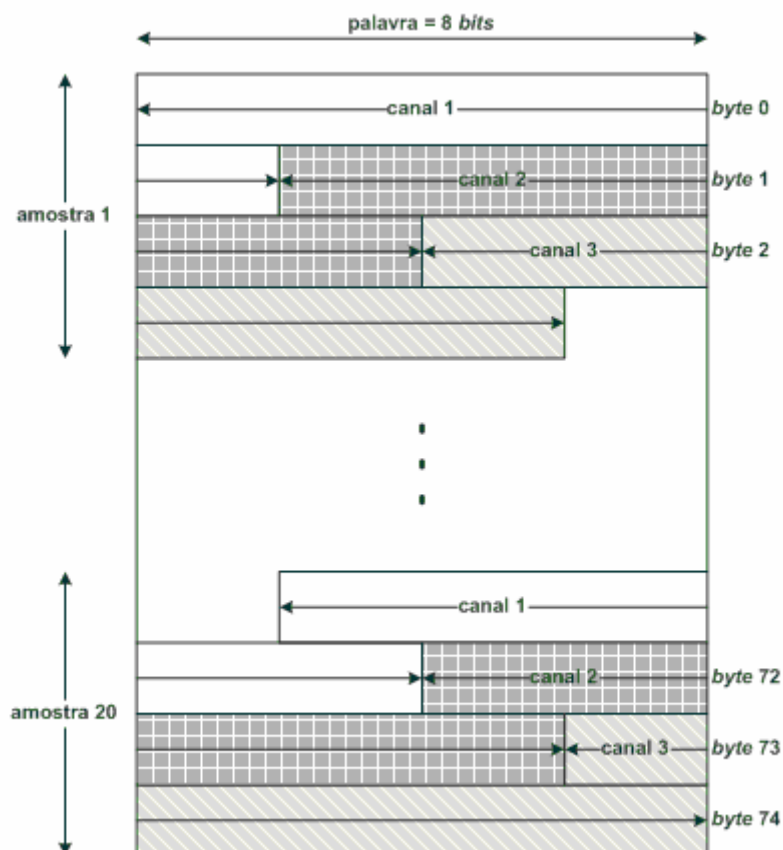


Figura 22: acondicionamento de amostras, e seus canais, em uma PARTE.



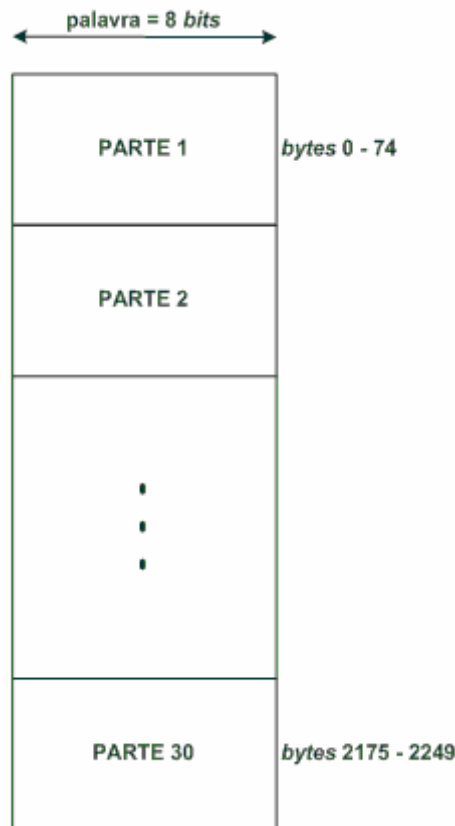


Figura 23: *buffer* circular de PARTES.

Como pode ser observado na Figura 23, o *buffer* circular implementado no *firmware* do protótipo tem capacidade para armazenar até 30 PARTES. Essa limitação é gerada pelo tamanho da memória *RAM* presente na *MCU*. Como foi utilizada a frequência de amostragem de 200 amostras por segundo e uma PARTE é formada por 20 amostras, conclui-se que tal *buffer* permite um período de interrupção na transferência de sinais de ECG de no máximo 10 segundos.

#### —EMPILHAR PARTE

Este processo é realizado por um conjunto de funções que são executadas em uma sequência bem definida. Em linhas gerais, a rotina principal do *firmware* encarrega-se de verificar, de tempos em tempos, se há PARTES a serem transmitidas ao Modem *GPRS*. Caso haja alguma, este processo é inicializado.

Durante a execução deste processo, todas as PARTES disponíveis na memória da *MCU* são enfileiradas, uma a uma, na memória do Modem *GPRS*. Esse enfileiramento é realizado de forma que a PARTE mais antiga presente no *buffer* circular é sempre a primeira PARTE a ser enfileirada.

### —ENVIAR BLOCO

Na medida em que PARTES vão sendo enfileiradas na memória interna ao Modem *GPRS*, um BLOCO de amostras vai sendo formado. Quando um BLOCO está completo, este processo se inicializa.

Basicamente, este processo se encarrega de emitir uma solicitação ao Modem *GPRS* para que o conteúdo de sua memória seja transmitido ao Sistema Servidor. Após a conclusão dessa transmissão, um novo BLOCO de amostras pode ser integrado em sua memória, que se encontra vazia.

### —MANTER RELÓGIO INTERNO

Este é o primeiro processo de controle a ser descrito. Sua execução é fundamental devido à necessidade de informações de relógio ao se transmitir dados de sinais de ECG a um Sistema Servidor. Ou seja, o Sistema Cliente deve informar ao Sistema Servidor em que momento tais dados de sinais de ECG foram obtidos.

Em termos de implementação, a manutenção do relógio interno à *MCU* é realizada por meio de uma de suas interrupções de tempo, configurada para ocorrer a cada um décimo de segundo. O fluxograma do procedimento tratador dessa interrupção é apresentado na Figura 24.

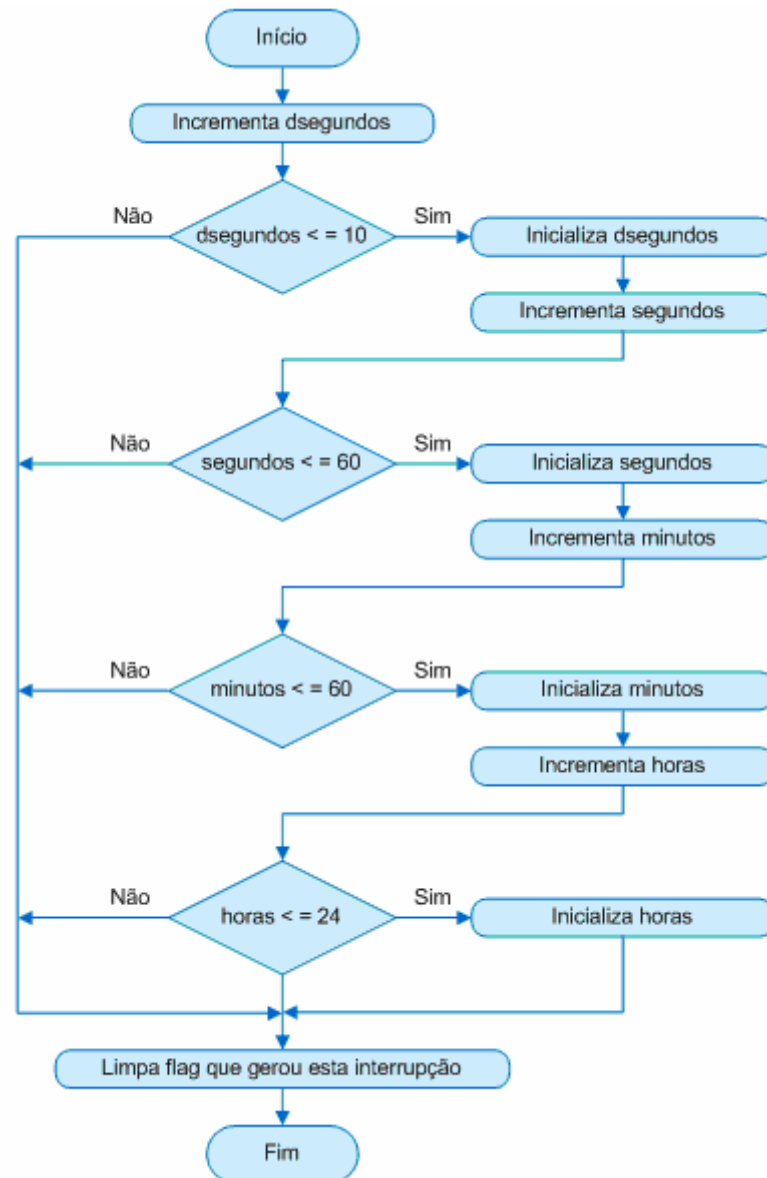


Figura 24: fluxograma do tratador da interrupção de tempo que mantém o relógio interno à *MCU*.

Como se pode notar, o relógio em questão é cíclico em 24 horas e possui informações atualizadas de horas, minutos, segundos e décimos de segundo. Tais características se mostram suficientes para o cumprimento de todas as atividades essenciais do Sistema Cliente.

Uma observação que se faz necessária é o fato deste processo ser responsável apenas pela manutenção do relógio interno, e não pela sua inicialização ou sincronização. Essa tarefa é realizada pelo protocolo de camada de aplicação descrito no Capítulo 4.

#### —ESTABELECEER FREQUÊNCIA DE AMOSTRAGEM

Esta também é uma atividade de controle fundamental para o Sistema Cliente. Sua execução se dá por meio da segunda interrupção de tempo disponível na *MCU*.

A única responsabilidade de seu tratador (Figura 25) é habilitar a interrupção de conversão A/D, de forma que uma seqüência de três conversões seja iniciada, formando, após sua conclusão, uma amostra de sinal ECG. Essa interrupção de tempo foi configurada para que seu tratador seja executado a cada 5 ms. Portanto, equivale dizer que são geradas 200 amostras por segundo.

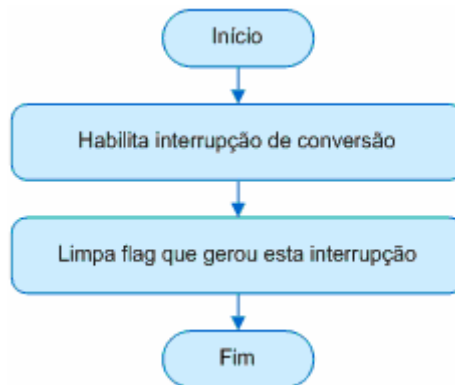


Figura 25: fluxograma do tratador da interrupção de tempo que estabelece a frequência de amostragem.

#### —ARQUITETURA DO *FIRMWARE*

Até este ponto no texto, o *firmware* foi descrito com um enfoque conceitual, de forma a evidenciar suas atividades essenciais. A partir de agora, a arquitetura de *software*<sup>6</sup> proposta neste trabalho, sob a qual o *firmware* deve ser implementado, é explorada.

Sabe-se que o desenvolvimento de sistemas computacionais em módulos propicia diversos benefícios, que podem ser observados tanto durante a fase de desenvolvimento quanto após sua conclusão. Alguns desses benefícios são:

- promove-se um alto grau de reuso de código-fonte. Por exemplo, sistemas similares podem ser desenvolvidos reutilizando-se módulos em comum;
- auxilia-se na geração de código-fonte facilmente adaptável. Novas funcionalidades são, em geral, fáceis de serem implantadas;
- incentiva-se trabalho em equipe. Equipes distintas podem trabalhar em módulos diferentes, inclusive de forma simultânea; e

---

<sup>6</sup> Arquitetura de *software* expressa uma divisão desse *software* em subsistemas, outros componentes de nível mais baixo, as interfaces entre os componentes e as interações através das quais eles realizam as funções do *software* [30].

- facilita-se a depuração de erros, já que na maioria dos casos é possível identificar o módulo que deve ser corrigido.

Nesse contexto, a arquitetura do *firmware* (Figura 26) foi desenhada com quatro módulos de *software* que se comunicam entre si por meio de chamadas de função e compartilhamento de variáveis.

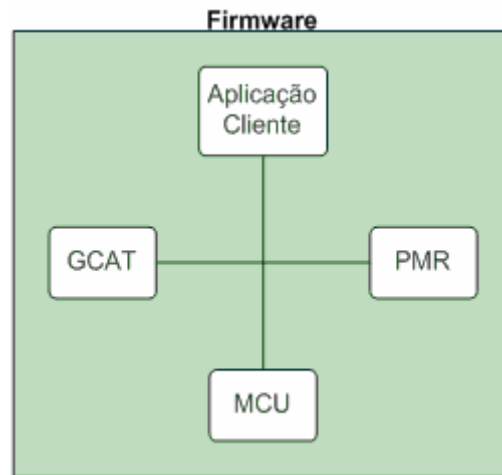


Figura 26: arquitetura do *firmware*.

Cada módulo foi desenvolvido para agrupar um sub-conjunto bem definido das funcionalidades do Sistema Cliente.

O módulo *MCU* recebe tal nome por implementar as funcionalidades altamente influenciadas por aspectos específicos da unidade de microcontrolador presente no dispositivo móvel utilizado. Suas funcionalidades mais relevantes são:

- inicializar a *MCU*;
- inicializar o Modem *GPRS*;
- acondicionar resultados de conversões A/D;
- gerenciar o *buffer* circular de dados;
- transmitir *bytes* ao Modem *GPRS*;
- ler *bytes* transmitidos pelo Modem *GPRS*;
- gerenciar controle de fluxo entre a *MCU* e o Modem *GPRS*;
- manter relógio interno;
- estabelecer frequência de amostragem;
- monitorar botão de pânico; e

- gerenciar LEDs de sinalização.

O módulo GCAT (Gerenciador de Comandos AT) conhece os comandos AT utilizados na gerência das atividades realizadas pelo Modem *GPRS*. De forma geral, quando se deseja solicitar uma determinada atividade ao modem, uma função desse módulo deve ser chamada e esta utilizará comandos AT apropriados para transmitir tal solicitação ao modem. Além disso, esse módulo encarrega-se de interpretar as respostas AT recebidas e tomar as devidas providências. As principais funcionalidades desse módulo são:

- realizar/desligar chamadas *PPP*;
- abrir/fechar conexões *TCP*; e
- prover acesso à pilha *TCP/IP* interna ao Modem *GPRS*.

O terceiro módulo, PMR (Protocolo de Monitoramento Remoto), implementa o protocolo de camada de aplicação encarregado de montar as mensagens<sup>7</sup>, de abertura e fechamento de conexão PMR e as de dados (sinal de ECG e/ou alarmes), a serem transmitidas a um Sistema Servidor. Além disso, esse módulo encarrega-se de interpretar as mensagens PMR recebidas de um Sistema Servidor e tomar as devidas providências. Esse protocolo é descrito em detalhes no Capítulo 4.

Por fim, o módulo Aplicação Cliente utiliza as funções providas pelos três módulos descritos anteriormente para concretizar as funcionalidades do Sistema Cliente. Além disso, esse módulo é responsável por estabelecer as políticas de funcionamento desse sistema. Todas as funcionalidades implementadas pelos três primeiros módulos descritos implementam somente mecanismos, deixando para seus usuários a responsabilidade de definir suas políticas. Por exemplo, o módulo GCAT implementa a função para se estabelecer chamadas *PPP*, que inclusive podem falhar. Em caso de falha, tal função não impõe nenhuma política, como realizar nova tentativa, por exemplo. Ao invés disso, o módulo Aplicação Cliente, usuário de tal função, fica responsável por estabelecer que até cem tentativas de chamadas *PPP* sejam realizadas, caso todas falhem, a reinicialização do dispositivo móvel é providenciada.

---

<sup>7</sup> Mensagens são pacotes de dados da camada de aplicação do modelo de referência *ISO/OSI* [31].

### 3.2 SISTEMA SERVIDOR

Como já mencionado, este sistema foi projetado para ser instalado em instituições de saúde. Suas principais funcionalidades são as seguintes:

- receber dados monitorados (sinais de ECG e acionamentos de alarme de pânico), transmitidos por Sistemas Clientes;
- persistir tais informações em banco de dados, possibilitando análises de históricos de dados de pacientes monitorados;
- prover interface *Web* para tais informações. Esse mecanismo possibilita o monitoramento a partir de qualquer ponto do globo, bastando para tal acesso à Internet; e
- atender requisições de seus usuários, em geral profissionais da saúde. Um exemplo de requisição é quando um profissional de saúde atende um alarme acionado por um paciente. Nesse caso, o Sistema Servidor acata tal atendimento enviando uma mensagem específica ao Sistema Cliente que teve seu alarme acionado.

Este sistema é também formado por componentes de *hardware* e de *software* que são descritos, separadamente, a seguir.

#### 3.2.1 *Hardware* do Sistema Servidor

Em termos de *hardware*, este sistema foi projetado de forma a necessitar apenas de um microcomputador pessoal convencional, referenciado neste texto por Servidor.

Essa simples configuração atende todas as necessidades do Sistema Servidor e ainda traz expressivos benefícios à instituição que o implanta, tais como: baixo custo, comparado a máquinas servidoras de grande porte; alta disponibilidade para reposição de periféricos; fácil manutenção, comparada a de equipamentos mais sofisticados; e possível reutilização de máquinas servidoras pré-existentes (por exemplo, máquinas servidoras de impressão ou máquinas de *backup*).

Especificamente no protótipo desenvolvido, o microcomputador utilizado tinha as seguintes características:

- processador *Pentium 4*, 3.2 GHz;
- 512 MB de *RAM*;

- 40 GB de *HD (hard drive)*;
- interface de rede *Fast Ethernet*;
- sistema operacional *Linux* (distribuição *Ubuntu*); e
- gerenciador de banco de dados *MySQL*.

### 3.2.2 *Software* do Sistema Servidor

Durante a fase de projeto conceitual deste *software*, também foram utilizados diagramas de fluxo de dados. Assim como ocorreu no desenvolvimento do *firmware* do Sistema Cliente, essa ferramenta agilizou consideravelmente o processo de implementação deste *software*.

A descrição deste *software* é feita de forma análoga à descrição do *firmware* do Sistema Cliente. Ou seja, aspectos conceituais são expostos inicialmente e, então, os componentes que integram o *software* do Sistema Servidor são detalhadamente descritos.

Inicia-se, portanto, a descrição conceitual com a apresentação dos relacionamentos entre o *software* do Sistema Servidor e suas entidades externas (Figura 27).

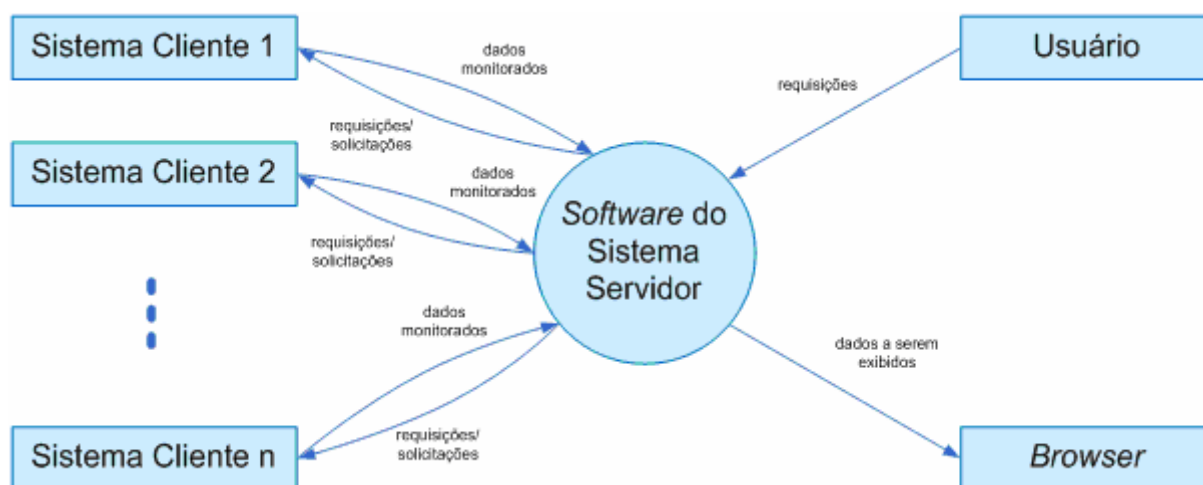


Figura 27: diagrama de contexto do *software* do Sistema Servidor.

O *software* do Sistema Servidor foi pensado e desenvolvido em três partes. A primeira parte, denominada de Núcleo, é responsável por grande parte das atividades essenciais do Sistema Servidor. Um banco de dados, que persiste informações de pacientes monitorados, é a segunda parte. Por fim, uma interface *Web*, que se encarrega de acessar o banco de dados para exigir suas informações em um navegador de Internet, compõe a terceira parte.



## —NÚCLEO DO SISTEMA SERVIDOR

O núcleo foi codificado com a linguagem Java. Os principais aspectos que influenciaram a escolha dessa linguagem foram:

- linguagem livre (gratuita e aberta);
- classes pré-existentes para comunicação *TCP/IP*;
- possibilidade de uso de *threads* para tratar atendimentos simultâneos a Sistemas Clientes; e
- fácil integração com diversos sistemas de gerenciamento de banco de dados.

Seguindo-se a mesma linha de desenvolvimento modular aplicado ao *firmware* do Sistema Cliente, resolveu-se desenvolver este Núcleo em três classes: Aplicação Servidora, PMR e Servidor.

A classe Aplicação Servidora estabelece e gerencia as funcionalidades do Sistema Servidor. Além disso, seus métodos são responsáveis por estabelecer as políticas de funcionamento deste sistema. Uma política que deve ser estabelecida diz respeito às condições sob as quais o Sistema Servidor deve aceitar conexões com Sistemas Clientes. Especificamente com relação ao Núcleo implementado, optou-se em aceitar conexões apenas com Sistemas Clientes que monitoram sinais de ECG compostos por três canais, amostrados a 200 Hz, utilizando-se para tal, conversores de 10 *bits*. Cabe ressaltar que apesar dessas exigências terem sido estabelecidas, o restante do *software* do Sistema Servidor, principalmente o protocolo de camada de aplicação desenvolvido, permite que outras políticas sejam desenvolvidas. Outra política a ser desenvolvida se refere às providências que devem ser tomadas ao se receber, de Sistemas Clientes, notificações de alarme. Uma possibilidade é programar o Núcleo para, ao receber uma notificação de alarme acionado, emitir um sinal sonoro e exibir em uma tela gráfica, o nome do alarme e o horário em que a notificação foi recebida.

A classe PMR implementa o protocolo de camada de aplicação desenvolvido, reproduzindo seus métodos e serviços. Como já mencionado, esse protocolo é descrito no Capítulo 4.

A terceira classe, e não por isso menos importante, que compõe o Núcleo do Sistema Servidor recebeu o nome de Servidor. Isso porque é essa classe que possibilita vários

Sistemas Clientes serem servidos, simultaneamente, por um mesmo Sistema Servidor. Para cada Sistema Cliente que inicia uma conexão com o Sistema Servidor, através de uma porta *TCP* bem conhecida, métodos dessa classe criam *threads* das classes Aplicação Servidora e PMR (Núcleo do Sistema Servidor) para gerenciarem tal conexão. É esse o mecanismo que permite um único Sistema Servidor monitorar, concorrentemente, diversos pacientes.

A Figura 28 apresenta um caso em que tal mecanismo é utilizado para que dois Sistemas Clientes possam estar conectados a um Sistema Servidor.

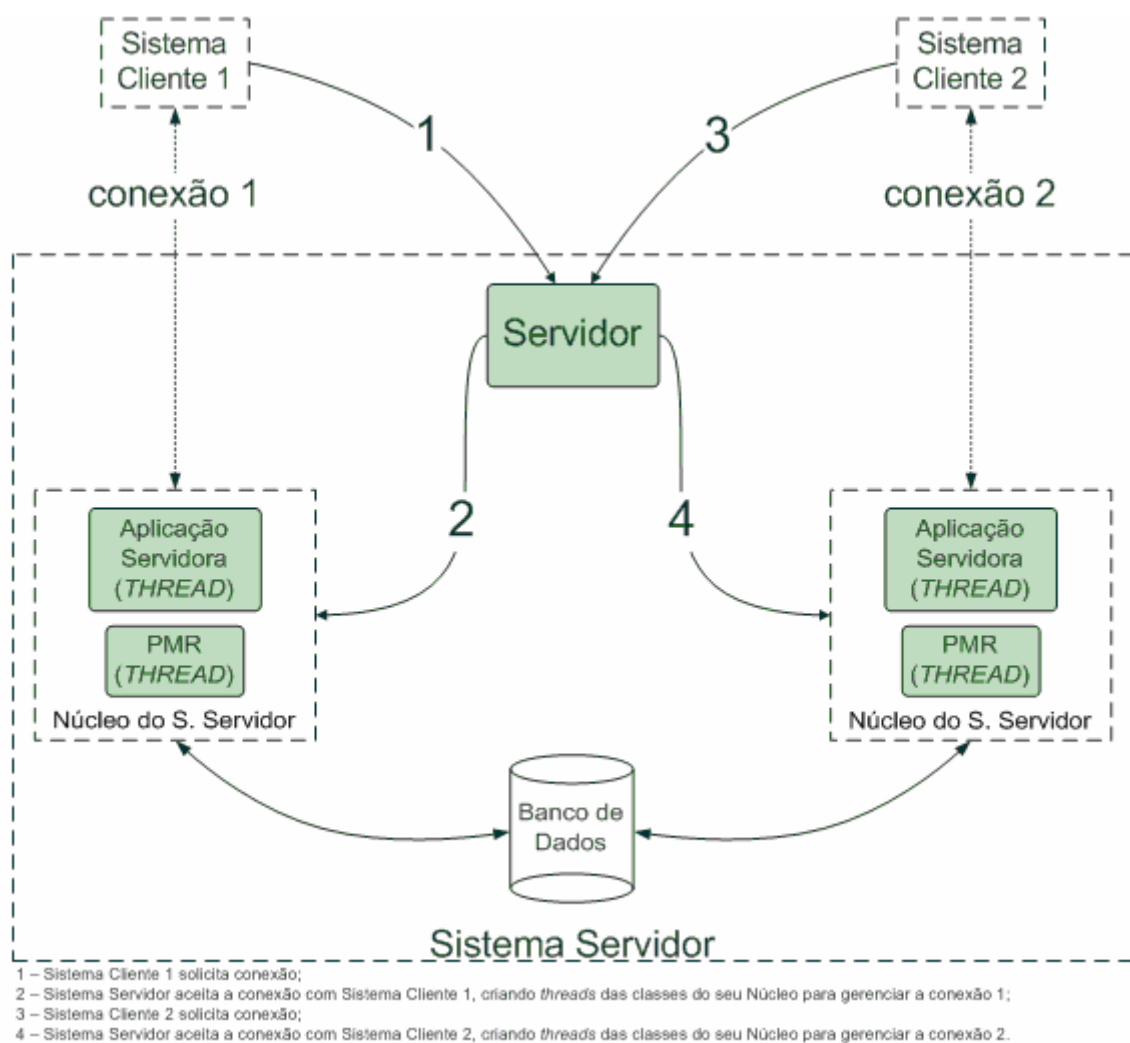


Figura 28: sequência de criações de conexões e *threads* entre um Sistema Servidor e dois Sistemas Clientes.

#### —BANCO DE DADOS

Uma discussão interessante sobre utilização de banco de dados e interface *Web* em sistemas de monitoramento é feita em [20]. Dentre as diversas vantagens apresentadas, a que mais se destaca no caso de monitoramento de sinais vitais é o fato do banco de dados possibilitar a

realização de análises de históricos de pacientes, tanto de forma *online* quanto *offline*. Em muitos casos, tais análises são importantes na determinação de diagnósticos e/ou tratamentos.

Para o efetivo armazenamento das informações monitoradas, utilizou-se parte do esquema de banco de dados desenvolvido em [20]. A parte utilizada do esquema (Figura 29) é composta por duas tabelas: *cardom\_ECG* e *cardom\_ECG\_dados*. A primeira registra características gerais sobre os sinais de ECG monitorados. Já a segunda tabela armazena fundamentalmente os valores dos canais que compõem os históricos dos sinais de ECG monitorados.

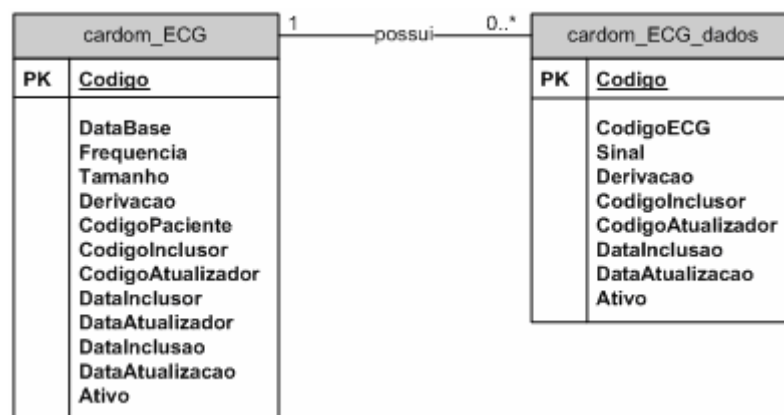


Figura 29: esquema de banco de dados utilizado.

O gerenciador de banco de dados escolhido para a materialização do esquema de banco de dados utilizado foi o *MySQL*. Suas principais características são as seguintes:

- multi-usuário;
- multi-tarefa;
- robustez;
- eficiência;
- flexibilidade para armazenar grande quantidade de dados; e
- utilização da *SQL (Structured Query Language)*.

#### —INTERFACE *WEB*

Também faz parte da solução proposta, uma interface *Web* que, em suma, permite que informações de sinais de ECG de pacientes monitorados possam ser acessadas a qualquer momento e a partir de qualquer ponto do planeta, bastando para tal acesso à Internet.

Ao se desenvolver o protótipo da solução proposta, utilizou-se a interface *Web* desenvolvida em [20]. Ela foi implementada com as tecnologias: *HTML* (*Hyper Text Markup Language*) e *PHP* (*Hypertext Preprocessor*).

A *HTML* é uma linguagem de marcação de *hypertexto* amplamente utilizada e padrão na *WWW* (*World Wide Web*). O *PHP* é uma linguagem baseada em *scripts* amplamente difundida, cujo código é livre [21]. Ela é particularmente apropriada para desenvolver aplicativos baseados na *Web*, pois oferece suporte a diversos gerenciadores de banco de dados e pode ser facilmente incorporada ao *HTML* [20] e [21]. A Figura 30 apresenta um sinal de ECG automaticamente desenhado na interface *Web* utilizada.

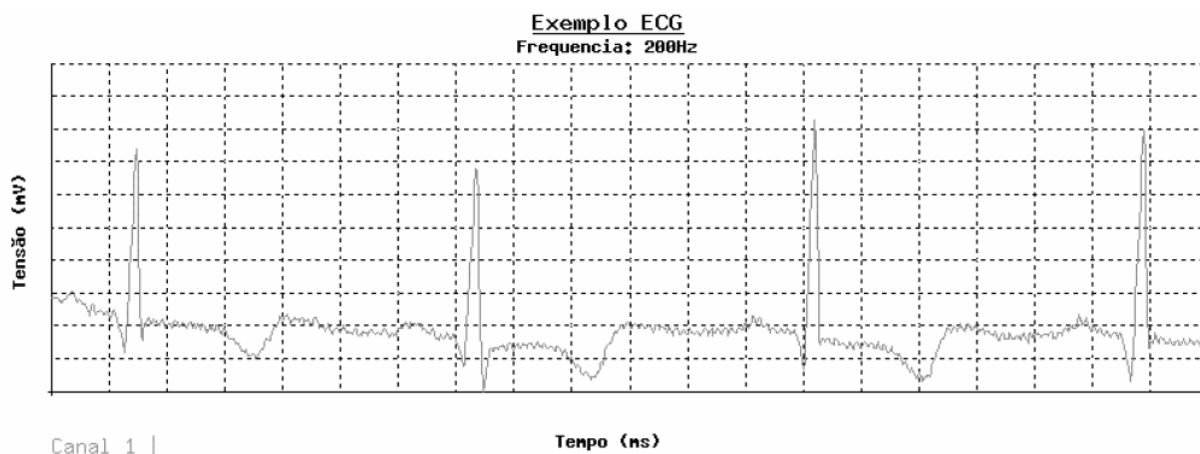


Figura 30: interface *Web* utilizada no projeto [20].

### 3.3 CONCLUSÕES

O projeto do sistema de monitoramento proposto foi realizado com sucesso. Além disso, um protótipo desse sistema foi desenvolvido, possibilitando assim uma melhor avaliação de desempenho da solução (Capítulo 5).

Durante o desenho, o desenvolvimento e a documentação do protótipo, notaram-se grandes benefícios provenientes da utilização de conceitos e ferramentas estabelecidas pela engenharia de *software*, tais como: modularização, parametrização, diagrama de fluxos de dados, diagrama de blocos e fluxogramas.

Para o desenvolvimento do Sistema Cliente em particular, a aquisição do *Data Modem G24* mostrou-se muito vantajosa. Contudo, observou-se durante a avaliação de desempenho que para aplicação explorada (monitoramento contínuo de ECG) a capacidade de

armazenamento de sua memória *RAM* necessita ser expandida. O motivo para essa conclusão é devidamente descrita na Subseção 5.3.1.

Ainda com relação ao *hardware* do Sistema Cliente, observou-se que o Módulo Pré-amplificador utilizado agregou valor à solução proposta e o fato de não ter sido necessário desenvolvê-lo possibilitou a canalização de mais energia ao desenvolvimento dos módulos de *software* do sistema de monitoramento.

Com relação ao Sistema Servidor, a configuração relativamente simples e padrão da máquina servidora torna a solução bastante atraente do ponto de vista de instituições de saúde. Suas outras importantes características que saltam aos olhos são:

- capacidade de monitoramento simultâneo de diversos pacientes;
- capacidade de monitoramento de diversos sinais vitais para cada paciente;
- persistência dos dados monitorados em sistema gerenciador de banco de dados; e
- interface *Web* que permite a visualização *online* de dados monitorados a partir de qualquer ponto de acesso do planeta à Internet.

## 4 PROTOCOLO DE MONITORAMENTO REMOTO (PMR)

Neste capítulo, é apresentado o protocolo de camada de aplicação, projetado nesta dissertação, utilizado pelo sistema de monitoramento de sinais cardíacos proposto. Esse protocolo é nomeado de PMR (Protocolo de Monitoramento Remoto).

Tendo em vista a grande energia necessária para se projetar um protocolo de comunicação, uma importante pergunta vem à tona: por que não utilizar protocolos padronizados? Neste trabalho em específico, o principal pilar da resposta para essa pergunta é o fato de não ter sido encontrado nenhum protocolo de camada de aplicação aberto e adequado ao tipo de comunicação que o sistema de monitoramento desenvolvido realiza.

É feita em [22] uma excelente discussão sobre protocolos de camada de aplicação utilizados atualmente na Telemedicina. Segundo seus autores, as soluções proprietárias são predominantes, mas há também algumas iniciativas de padronização, incluindo: *DICOM*, *HL7* [23], *IEEE* [24] e *CEN* [25]. Apesar dessas soluções serem abertas, elas não se mostraram adequadas ao projeto. Um dos fatores em desconformidade é o fato delas serem orientadas a objeto. Portanto, as informações são transmitidas de forma encapsulada em objetos, o que proporciona considerável *overhead*<sup>8</sup> de comunicação.

Em determinados cenários, mesmo que o *overhead* gerado seja considerável, ele pode se mostrar não-crítico para a comunicação das informações. Contudo, particularmente nesta dissertação, devido ao volume de dados gerado pela frequência de amostragem de 200 amostras por segundo e a limitação de banda existente na rede *GPRS*, chegou-se à conclusão de que tal *overhead* deveria ser evitado.

Além disso, os principais protocolos abertos encontrados foram desenvolvidos originalmente para transmissão de imagens, e não de sinais com formato de onda (*waveform*), que é o caso do ECG.

Esses fatores evidenciaram a necessidade do desenvolvimento de um protocolo de camada de aplicação que, juntamente com outros protocolos de camadas inferiores, atendessem

---

<sup>8</sup> *Overhead* refere-se a qualquer processamento ou armazenamento em excesso, seja de tempo de computação, de memória, de largura de banda ou qualquer outro recurso requerido para ser utilizado ou gasto durante execução de uma determinada tarefa.

os requisitos do projeto no que diz respeito à comunicação de dados. Tais requisitos são listados a seguir:

- baixo *overhead*;
- sem perda de dados; e
- com entrega ordenada.

Baixo *overhead* é exigido para possibilitar que os dados gerados pelo processo de amostragem possam ser transmitidos a um Sistema Servidor, a partir de um dispositivo móvel que contém uma memória finita e através de uma rede *GPRS* com banda limitada. A comunicação deve se dar sem perda de dados, pois dados perdidos significam menos informações, sobre o sinal monitorado, que podem ser valiosas em determinados cenários, principalmente quando se está lidando diretamente com vidas humanas. Exige-se que a entrega dos dados monitorados a um Sistema Servidor seja feita de forma ordenada, devido à relação existente entre tais dados, que são obtidos a partir de um sinal analógico, e o tempo decorrido. Em outras palavras, equivale dizer que um Sistema Servidor deve receber as amostras de um sinal monitorado na mesma sequência em que essas amostras foram obtidas. Caso contrário, a integridade desse sinal estaria completamente comprometida.

Durante a fase inicial de desenho deste protocolo, percebeu-se que o aumento da complexidade em se desenvolver um protocolo que pudesse ser utilizado para monitorar sinais analógicos diversos, inclusive sinais de ECG, não seria expressivo. Esse grau de generalização proporciona uma série de benefícios. Um dos que merece destaque é a possibilidade de um dispositivo móvel monitorar diversos sinais (ECG, temperatura corporal, pressão arterial e outros), utilizando para tal somente um protocolo de camada de aplicação, o que minimizaria a complexidade de seu *firmware*. Outro benefício que se pode observar é o fato deste protocolo não estar necessariamente vinculado a monitoramento de sinais vitais. Ou seja, sua utilização também se mostra interessante em outras áreas que necessitam de transferência de sinais analógicos, por exemplo, na instrumentação industrial.

Tendo em vista os requisitos do projeto relacionados à comunicação de dados e as opções de escolha ao se utilizar a pilha de protocolo *TCP/IP*, algumas decisões de projeto foram tomadas.

A primeira decisão assumida foi a escolha do protocolo *TCP* (*Transmission Control Protocol*) em detrimento do *UDP* (*User Datagram Protocol*). Ambos são protocolos de camada de transporte, sendo responsáveis, portanto, basicamente por promover transferência de dados entre a máquina de origem e a máquina de destino, independentemente das redes físicas em uso [26]. Em suma, essa decisão foi tomada baseando-se nos tipos de serviços oferecidos por cada um desses dois protocolos.

O *UDP* oferece um serviço de transporte com as seguintes características: não-orientado à conexão, sem controle de fluxo, sem controle de erro e sem retransmissão após a recepção de um segmento incorreto. Uma área na qual o *UDP* é especialmente útil é a de situação cliente-servidor, em que o cliente envia frequentemente pequenas solicitações ao servidor e, então, espera uma pequena resposta de volta. Se a solicitação ou a resposta se perder, o cliente simplesmente chegará ao *timeout*<sup>9</sup> e tentará de novo [26].

Já o *TCP* foi projetado para oferecer um fluxo confiável de *bytes*, fim a fim, em uma rede não-confiável. Ele oferece um serviço de transporte com as seguintes características: orientado à conexão, *full-duplex*<sup>10</sup>, ponto a ponto<sup>11</sup>, com controle de fluxo, com controle de erro e com retransmissão após a recepção de um segmento incorreto. Uma área na qual a utilização do *TCP* se mostra interessante é a de cliente-servidor, em que os segmentos devem ser garantidamente entregues, de forma ordenada e sem erros, ao servidor. Esse é justamente o cenário desejado para o sistema proposto.

Faz-se necessário observar, contudo, que essa primeira decisão poderia ter sido diferente, ou seja, o *UDP* poderia ter sido escolhido no lugar do *TCP*. Nesse caso, porém, teria sido obrigatório implementar no protocolo de camada de aplicação todos os mecanismos necessários ao sistema proposto que não se encontram presentes no *UDP*. Portanto, observa-se que essa decisão diminui consideravelmente a complexidade do protocolo de camada de aplicação que foi projetado.

Com o uso do *TCP* já determinado, foram definidas as diretrizes para o projeto do protocolo a ser proposto. São elas:

---

<sup>9</sup> *Timeout* refere-se a um período contínuo de tempo em que um determinado evento não ocorre [26].

<sup>10</sup> *Full-duplex* é um tipo de comunicação em que o tráfego pode ser feito em ambos os sentidos ao mesmo tempo [26].

<sup>11</sup> Ponto a ponto significa que cada conexão possui exatamente dois pontos terminais [26].



- simplificar o quanto for possível;
- orientação a *bit*;
- orientação a conexão;
- comunicação ponto a ponto; e
- *statefull*.

Optou-se pela orientação a *bit* basicamente devido a dois fatores: flexibilidade na escolha de tamanhos de campos e minimização de *overhead*. Em muitos casos, não se faz necessário alocar um número de *bits* múltiplo de oito (tamanho de um *byte*). Por exemplo, há um campo MET, que ainda será explanado, que exige somente 11 diferentes valores. Caso fosse necessário alocar um número inteiro de *bytes* para tal campo, 1 *byte* seria suficiente, mas os 4 *bits* mais significativos sempre seriam desperdiçados, o que significa um *overhead* de 50%. Ao se considerar o somatório dos *bits* que seriam desperdiçados em cada campo da mensagem, devido a uma possível orientação a *byte*, muito possivelmente o *overhead* resultante seria alto relativamente às necessidades do sistema.

Todo monitoramento de sinais cardíacos, inclusive outros sinais analógicos, tem suas próprias características, algumas delas são: frequência de amostragem do sinal monitorado, número de canais que compõem o sinal e precisão do conversor A/D utilizado na digitalização do sinal. A orientação a conexão é necessária justamente para que os sistemas envolvidos no monitoramento, Cliente e Servidor, possam negociar, durante a fase de abertura de conexão, as características desejadas para tal monitoramento.

A diretriz de comunicação ponto a ponto foi fundamentalmente definida por simplificar consideravelmente a complexidade do PMR, mantendo o cumprimento dos requisitos do projeto. Como o *TCP*, protocolo utilizado na camada imediatamente abaixo do PMR, utiliza comunicação ponto a ponto, nenhum mecanismo de endereçamento foi necessário ser implementado dentro do PMR. Por outro lado, caso a comunicação multi-ponto fosse definida para essa diretriz, algum tipo de endereçamento no nível de PMR seria necessário, o que aumentaria desnecessariamente a complexidade deste protocolo.

Considerando-se todo o período em que uma conexão PMR encontra-se aberta, observou-se que seria interessante este protocolo padronizar comportamentos diferentes sob certas circunstâncias. Nesse sentido, para que o PMR possa identificar as diferentes circunstâncias em que ele pode se encontrar, uma variável de estado é utilizada em cada

entidade deste protocolo. Ou seja, o PMR implementado em uma ponta da conexão possui sua própria variável de estado e o PMR implementado na outra ponta possui outra variável de estado, sendo que essas duas variáveis são independentes entre si. Nesse contexto, como cada entidade de protocolo armazena seu estado atual, por meio de variável de estado, este protocolo também é considerado *statefull*. Duas máquinas de estados, ligeiramente diferentes entre si, fazem parte da especificação do PMR. Quando essas máquinas de estados são devidamente respeitadas, dados monitorados são transmitidos e recebidos de forma correta. Ambas as máquinas de estados mencionadas são descritas na Seção 4.2.

#### 4.1 MÉTODOS DO PMR

Toda mensagem PMR é identificada por um dos onze métodos existentes no PMR. O formato geral desse tipo de mensagem é apresentado na Figura 31.



Figura 31: formato geral de uma mensagem PMR.

Cada método, além de definir o tipo da mensagem, define também o tamanho e a semântica de cada campo que compõe a mensagem. Ao longo desta seção, todos os métodos existentes no PMR são descritos.

Em termos de propósito, tais métodos são divididos em três classes: os que transmitem requisições, os que transmitem respostas e os que transmitem notificações. Foram desenvolvidos dois métodos de requisição, quatro de resposta e cinco de notificação. Os métodos de requisição são utilizados na abertura e na reconfiguração da conexão, exigindo sempre uma resposta da outra ponta. Os métodos de resposta são utilizados para responder, com sucesso ou falha, os métodos de requisição. Os métodos da última classe são utilizados para notificar um conjunto de dados, um evento ou uma solicitação, não sendo necessária nenhuma resposta da outra ponta.

Alguns destes métodos foram projetados para serem enviados somente por Sistemas Clientes, não fazendo sentido tais métodos serem enviados por Sistemas Servidores. Outros métodos foram projetados para a situação inversa. Há também métodos que podem ser enviados por ambos os sistemas. Ao longo da descrição de cada método, feita a seguir, essa característica é especificada.

Uma observação geral aos métodos que se cabe ressaltar é o fato de suas mensagens serem, em geral, de tamanhos variáveis. Nesses casos, o tamanho final da mensagem é dependente de alguns valores de seus campos. Além de serem variáveis, o tamanho final de cada mensagem deve ser, obrigatoriamente, múltiplo de 8 *bits* (tamanho de 1 *byte*), já que não é possível transmitir frações de *byte* via *TCP*. Portanto, mesmo que o PMR seja orientado a *bit*, o fato do *TCP* ser orientado a *byte* tem que ser respeitado. Essa adequação é feita da seguinte forma: caso o número de *bits* necessário para transmitir todo o conteúdo de uma determinada mensagem não seja múltiplo de 8, a técnica de *bit stuffing* é utilizada, ou seja, são adicionados *bits* sem valor semântico ao final da mensagem até que essa tenha seu tamanho expresso em um número inteiro de *bytes*.

#### 4.1.1 Métodos de Requisição

Como já foi mencionado no início desta seção, a classe de requisições é composta por dois métodos, a saber: REQCNX (requisição de conexão) e REQRCNF (requisição de reconfiguração de parâmetros de monitoramento).

O REQCNX foi projetado para ser utilizado somente por Sistemas Clientes. Ele deve ser utilizado quando se deseja abrir uma conexão PMR com um Sistema Servidor. Seus campos transmitem informações sobre os parâmetros desejados para o monitoramento a ser realizado e sobre o próprio Sistema Cliente que está enviando a mensagem. Quando essa mensagem é recebida pelo Sistema Servidor, ele tem condições de avaliar se suporta o tipo de monitoramento requisitado. Após essa avaliação, o Sistema Servidor deve transmitir de volta uma resposta, informando se a abertura de conexão PMR requisitada foi aceita ou recusada.

Os campos que compõem uma mensagem identificada pelo método REQCNX são:

- MET (4 *bits*): identificador do método igual a REQCNX;
- IMEI (60 *bits*): identificador do Modem *GPRS* (único no mundo);
- VER (4 *bits*): versão utilizada do PMR; e
- SIN (8 *bits*): número de sinais que se deseja monitorar.

O campo IMEI<sup>12</sup> é composto por 60 *bits* devido ao fato do próprio Modem *GPRS* possuir um identificador único (configurado no momento de sua fabricação) desse tamanho, que pode ser acessado via *software*. A principal motivação para a transmissão dessa informação é possibilitar o cadastramento de quais Sistemas Clientes utilizam quais identificadores. Esse tipo de cadastramento permite, por exemplo, que um Sistema Servidor identifique quais Sistemas Clientes devem ser aceitos ao solicitarem abertura de conexão. Cabe salientar também que esse identificador pode ser utilizado, em esquema do banco de dados, para identificar univocamente os históricos dos sinais monitorados por cada Sistema Cliente cadastrado.

Os principais propósitos que se almeja ao considerar um campo VER na mensagem PMR de abertura de conexão são: possibilitar que o PMR evolua com o passar do tempo e permitir que diferentes versões desse protocolo possam ser utilizadas por sistemas distintos. Nesse contexto, um Sistema Cliente escolhe a versão do PMR que deseja utilizar e o Sistema Servidor reporta, durante a abertura da conexão, se suporta ou não tal versão.

Já o campo SIN foi pensado para informar a um Sistema Servidor o número de sinais a serem monitorados, concorrentemente, por um mesmo Sistema Cliente. Seu tamanho foi escolhido de forma a permitir que cada dispositivo móvel monitore até 256 ( $2^8$  *bits*) sinais ao mesmo tempo.

Para cada sinal que se deseja monitorar, são adicionados na mensagem PMR campos que representam suas características de monitoramento, são eles:

- TIP (5 *bits*): tipo de sinal;
- UNI (5 *bits*): unidade do sinal;
- FRE (8 *bits*): frequência de amostragem do sinal utilizada;
- AMO (12 *bits*): número de amostras que compõem um BLOCO;
- CAN (2 *bits*): número de canais utilizado (que compõem uma amostra);
- BIT (4 *bits*): número de *bits* que compõem a amostra de um canal;
- MEN (10 *bits*): menor valor absoluto do sinal; e
- MAI (10 *bits*): maior valor absoluto do sinal.

---

<sup>12</sup> IMEI é um acrônimo para *International Mobile Station Equipment Identity*, cuja especificação foi realizada pelo GSM 03.03.

Como já mencionado, o PMR foi projetado fundamentalmente para ser utilizado em diferentes formas de monitoramento de sinais analógicos diversos. Ou seja, sua utilização não se restringe a um determinado monitoramento (com características fixas) de um único tipo de sinal analógico. Sendo assim, durante a abertura de conexão entre um Sistema Cliente e um Sistema Servidor, é necessário o primeiro informar ao segundo: (i) as características dos sinais analógicos que se pretende monitorar, (ii) as características sob as quais tais sinais devem ser monitorados. Conhecendo essas características, o Sistema Servidor envolvido tem condições de manipular corretamente as mensagens PMR que transmitem informações de sinais monitorados. O principal momento onde esse tipo de manipulação é determinante ocorre ao se extrair valores de canais a partir de amostras, que são encapsuladas em BLOCOS. Caso o Sistema Servidor não conhecesse tais características, essa extração não poderia ser realizada de forma adequada.

A disposição de todos os campos que compõem uma mensagem PMR do tipo REQCNX é exibida na Figura 32.

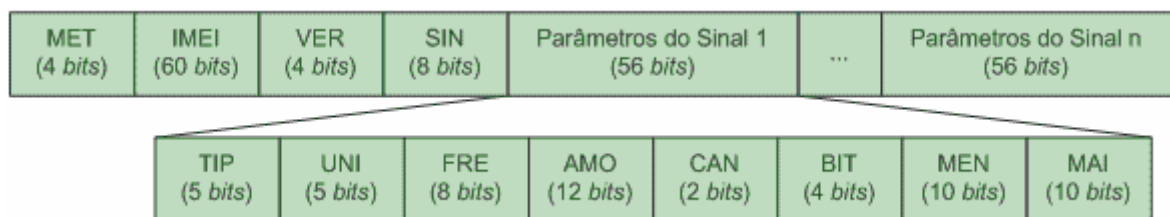


Figura 32: formato da mensagem PMR do tipo REQCNX.

No PMR, além dos métodos e seus campos serem definidos, cada valor possível para cada campo também é padronizado. Essa padronização é feita por meio de codificação, onde é atrelado um significado a cada código de cada campo do PMR.

Os códigos, e seus significados, dos campos que compõem uma mensagem identificada pelo método REQCNX são apresentados da Tabela 1 a Tabela 11.

Tabela 1: codificação do campo MET.

Método	Código (4 bits)
REQCNX	0
REQRCNF	1
RESCNXS	2
RESCNXF	3
RESRCNFS	4
RESRCNFF	5
NOTBLCAMO	6
NOTREQBLCAMO	7
NOTALRMAC	8
NOTALRMAT	9
NOTDCNX	10
-	11 a 15

Tabela 2: codificação do campo VER.

Versão	Código (4 bits)
1.0	0
-	1 a 15

Tabela 3: codificação do campo TIP.

Tipo de Sinal	Código (5 bits)
Nível	0
Velocidade	1
Pressão	2
Tensão	3
Corrente	4
Temperatura	5
-	6 a 31

Tabela 4: codificação do campo SIN.

Nº de Sinais	Código (8 bits)
1	0
1 a 255	SIN - 1

Tabela 5: codificação do campo AMO.

Nº de Amostras	Código (12 bits)
1	0
2 a 4.095	AMO - 1
4.096	4.095

Tabela 6: codificação do campo CAN.

Nº de Canais	Código (2 bits)
1	0
2	1
3	2
4	3

Tabela 7: codificação do campo BIT.

Nº de Bits	Código (4 bits)
1	0
2 a 15	BIT - 1
16	15

Tabela 9: codificação do campo MEN.

Valor Mínimo	Código (10 bits)
0	0
1 a 1.022	MEN
1.023	1.023

Tabela 8: codificação do campo UNI.

Unidade de Sinal	Código (5 bits)
mm	0
cm	1
m	2
km	3
mm/s	4
cm/s	5
m/s	6
km/s	7
atm	8
pascal	9
µV	10
mV	11
V	12
kV	13
µA	14
mA	15
A	16
kA	17
°C	18
°F	19
-	20 a 31

Tabela 10: codificação do campo MAI.

Valor Máximo	Código (10 bits)
0	0
1 a 1.022	MAI
1.023	1.023

Tabela 11: codificação do campo FRE.

Frequência (Hz)	Código (8 bits)	Frequência (Hz)	Código (8 bits)
500	0	210	29
490	1	200	30
480	2	190	31
470	3	180	32
460	4	170	33
450	5	160	34
440	6	150	35
430	7	140	36
420	8	130	37
410	9	120	38
400	10	110	39
390	11	100	40
380	12	90	41
370	13	80	42
360	14	70	43
350	15	60	44
340	16	50	45
330	17	40	46
320	18	30	47
310	19	20	48
300	20	10	49
290	21	9	50
280	22	8	51
270	23	7	52
260	24	6	53
250	25	5	54
240	26	4	55
230	27	3	56
220	28	2	57
Frequência		Código (8 bits)	
1, 2 ... 10 (s <sup>-1</sup> )		58 a 67	
15, 20 ... 60 (s <sup>-1</sup> )		68 a 77	
2, 3 ... 10 (m <sup>-1</sup> )		78 a 86	
15, 20 ... 60 (m <sup>-1</sup> )		87 a 96	
70, 80 (m <sup>-1</sup> ) ... 24 (h <sup>-1</sup> )		97 a 234	
2, 3 ... 7 (dia <sup>-1</sup> )		235 a 240	
-		241 a 255	

O segundo, e último, método de requisição é o REQRCNF. Ele foi projetado para ser utilizado tanto por Sistemas Clientes quanto por Sistemas Servidores. Seu uso deve se dar em situações em que se deseja reconfigurar, junto a outra ponta da conexão, parâmetros de monitoramento de um determinado sinal que já esteja sendo monitorado. Quando uma mensagem desse tipo é recebida por uma das pontas, seu sistema tem condições de avaliar se suporta o novo tipo de monitoramento requisitado. Após essa avaliação, esse sistema deve transmitir de volta uma resposta informando que a reconfiguração requisitada foi aceita ou

recusada. Caso a reconfiguração seja aceita, o monitoramento passa a ser realizado de acordo com os novos parâmetros. Caso contrário, as configurações anteriores continuam sendo utilizadas.

Um exemplo de cenário no qual esse método é útil é o seguinte: imagine um monitoramento de um sinal analógico qualquer sendo realizado com uma frequência de amostragem de 1 amostra por hora; considere também que o Sistema Servidor utilizado tenha acabado de receber o último valor desse sinal e que esse valor seja considerado suspeito. Nesse contexto, uma alternativa que facilitaria o diagnóstico de eventuais problemas nesse sinal monitorado seria aumentar sua frequência de amostragem. Fazendo isso, seria possível, por exemplo, identificar eventuais oscilações que possam estar ocorrendo no sinal monitorado.

Os campos que compõem uma mensagem identificada pelo método REQRCNF são listados a seguir:

- MET (4 *bits*): identificador do método igual a REQRCNF;
- IDSIN (8 *bits*): identificador do sinal que se deseja reconfigurar; e
- PAR (3 *bits*): número de parâmetros do sinal em questão que se deseja reconfigurar.

O campo IDSIN foi projetado de forma a identificar qualquer sinal dentre os que estejam sendo monitorados por um dispositivo móvel. Para isso, bastou estabelecer seu tamanho igual ao tamanho do campo SIN, que inclusive já foi descrito nesta seção. Essa identificação é feita na mesma ordem em que os sinais monitorados foram especificados na mensagem PMR utilizada na abertura da conexão. Considere, por exemplo, um suposto caso em que dois sinais estejam sendo monitorados, temperatura corporal e ECG, e que as características do sinal de ECG tenham sido adicionadas por último na mensagem PMR do tipo REQCNX. Nesse caso, o sinal de temperatura é identificado como primeiro e o ECG como segundo.

O campo PAR, por sua vez, foi especificado de forma a possibilitar a reconfiguração, de uma só vez, de todos os oito parâmetros de um determinado sinal analógico que esteja sendo monitorado.



Para cada parâmetro que se deseja alterar, são adicionados na mensagem PMR dois novos campos, um que identifica o parâmetro a ser alterado e outro que especifica seu novo valor desejado. Os nomes e tamanhos desses campos são:

- IDPAR (3 *bits*): identificador do parâmetro que se deseja reconfigurar; e
- VAL (12 *bits*): novo valor desejado para o parâmetro.

É importante salientar que os identificadores de parâmetros foram estabelecidos de acordo com a ordem em que tais parâmetros são dispostos na mensagem PMR do tipo REQCNX. Ou seja, na ordem: TIP, UNI, FRE, AMO, CAN, BIT, MEN e MAI.

O seqüenciamento dos campos que compõem uma mensagem PMR do tipo REQRCNF é mostrado na Figura 33.

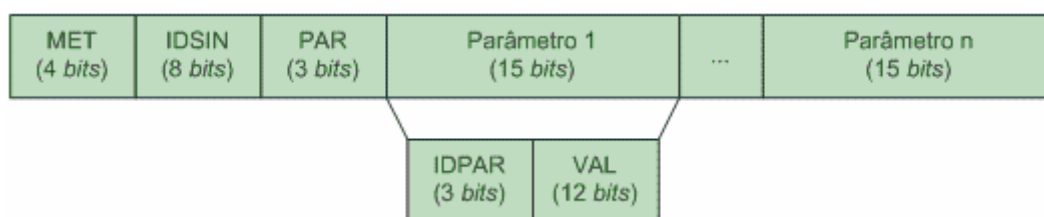


Figura 33: formato da mensagem PMR do tipo REQRCNF.

Os códigos dos campos que compõem o corpo da mensagem identificada pelo método REQRCNF e seus respectivos significados são apresentados da Tabela 12 a Tabela 14.

Tabela 12: codificação do campo IDSIN.

Id. do Sinal	Código (8 <i>bits</i> )
1	0
2 a 255	IDSIN - 1
256	255

Tabela 13: codificação do campo IDPAR.

Id. do Parâmetro	Código (3 <i>bits</i> )
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7

Tabela 14: codificação do campo PAR.

Nº do Parâmetro	Código (3 <i>bits</i> )
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7

#### 4.1.2 Métodos de Resposta

Os métodos pertencentes a esta classe foram desenhados para serem utilizados em situações em que uma ponta da conexão recebe uma mensagem do tipo requisição. Nessa situação, a ponta em questão deve, após analisar a requisição recebida, utilizar um dos métodos desta classe para transmitir à ponta requisitante sua resposta.

Uma observação importante acerca desse tipo de mensagem é o fato de não ser exigido da ponta que a recebe uma nova mensagem de resposta. Ou seja, O PMR foi projetado de forma que não se responde uma mensagem do tipo resposta.

Os dois primeiros métodos desta classe, a serem descritos a seguir, foram projetados para responderem requisições de conexão PMR. Como requisições desse tipo somente são recebidas, coerentemente, por Sistemas Servidores, esses dois métodos são, portanto, de uso exclusivo desses sistemas.

O primeiro deles, RESCNXS (resposta de conexão com sucesso), deve ser utilizado nos casos em que um Sistema Servidor decide que uma determinada conexão requisitada pode ser realizada e o será. Sua composição se dá por dois campos, são eles:

- MET (4 *bits*): identificador do método igual a RESCNXS; e
- REL (20 *bits*): relógio atual do Servidor.

Como se pode perceber, a única informação útil transmitida por meio desse tipo de mensagem é o horário atual do Servidor. Essa informação é necessária devido ao fato do dispositivo móvel proposto não possuir relógio próprio. Dessa forma, durante a abertura da conexão entre os Sistemas Cliente e Servidor, é realizado um sincronismo de relógio para que o sistema de monitoramento como um todo possa identificar e registrar em que momentos foram realizadas as amostragens dos sinais monitorados. A descrição de como o Sistema Cliente mantém o horário informado pelo Sistema Servidor é feita na Subseção 3.1.2 (Figura 24).

A Figura 34 apresenta como os campos MET e REL são dispostos em uma mensagem do tipo RESCNXS e a Tabela 15 especifica os códigos padronizados do campo REL. Nessa tabela, as siglas H, M, S e DS denotam, respectivamente, horas, minutos, segundos e décimos de segundo.

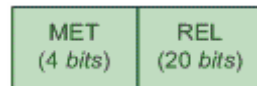


Figura 34: formato da mensagem PMR do tipo RESCNXS.

Tabela 15: codificação do campo REL.

Relógio	Código (20 bits)
0:0:0:0	0
H:M:S:DS	$H*36000 + M*600 + S*10 + DS$
24:0:0:0	864.000

O segundo método utilizado para responder requisições de conexão recebeu o nome de RESCNXF (resposta de conexão com falha) e deve ser utilizado por um Sistema Servidor quando esse decide que a conexão solicitada não deve ser estabelecida. Esse tipo de mensagem possui uma estrutura mais elaborada e permite ao Sistema Servidor transmitir, ao Sistema Cliente que requisitou a conexão, as informações de quais parâmetros de quais campos da mensagem do tipo REQCNX não foram aceitos. Dessa forma, o Sistema Cliente toma ciência de quais parâmetros ele deve alterar, se conveniente for, para que haja alguma chance de sucesso em uma nova tentativa de abertura de conexão com tal Sistema Servidor.

Um exemplo em que esse mecanismo se mostra interessante é descrito a seguir. Imagine que um Sistema Cliente requisi uma conexão com um Sistema Servidor para que um determinado sinal possa ser monitorado a uma frequência de amostragem de 1 Hz. Imagine também que esse Sistema Servidor responda essa requisição com uma mensagem do tipo RESCNXF, que carrega consigo a informação de que o parâmetro FRE desejado para esse sinal não pôde ser aceito. Dessa forma, o Sistema Cliente toma conhecimento de que a única característica de monitoramento não-suportada por aquele Sistema Servidor foi o código 58 para o parâmetro FRE (vide Tabela 11). Em seguida, se o Sistema Cliente desejar, ele pode alterar a frequência de amostragem do sinal a ser monitorado para  $(2\text{ s})^{-1}$ , antes de uma nova tentativa de abertura de conexão.

Os campos que compõem uma mensagem identificada pelo método RESCNXF são:

- MET (4 bits): identificador do método igual a RESCNXF; e
- CAM (9 bits): número de campos inaceitáveis.

Para cada campo não-suportado, são adicionados na mensagem PMR duas novas informações, uma que identifica tal campo e outra que apresenta quantos são seus parâmetros não-suportados. Essas informações são transmitidas por meio dos seguintes campos:

- IDCAM (9 *bits*): identificador do campo não-suportado; e
- PAR (3 *bits*): número total de parâmetros inaceitáveis (não-suportados).

O campo CAM foi convenientemente projetado de forma a possibilitar a indicação simultânea de não-aceitação dos campos MET, IMEI, VER e SIN, além dos 256 sinais que podem estar sendo monitorados de forma concorrente. É justamente nessa ordem que seus identificadores (IDCAM) são especificados.

É importante perceber que os quatro primeiros campos (MET, IMEI, VER e SIN) não possuem nenhum parâmetro (vide Figura 32). Portanto, nos casos em que o IDCAM é especificado de forma a representar algum desses campos, o valor do campo PAR deve ser ignorado. Nesses casos, conclui-se que o próprio valor do campo identificado por IDCAM não é suportado pelo Sistema Servidor. Por exemplo, imagine uma situação em que um Sistema Servidor tenha recebido uma mensagem de requisição de conexão, no entanto, ele tenha verificado que tal conexão não pode ser estabelecida devido a duas características desejadas não serem suportadas: a versão do PMR e o número de canais por amostra do primeiro sinal. Nesse caso, o Sistema Servidor deve retornar, à ponta requisitante, uma mensagem do tipo RESCNXF com a seguinte semântica:

- MET com o código do método RESCNXF;
- CAM informando que o total de campos não-suportados foi dois;
- IDCAM com o código referente ao campo VER;
- PAR com valor qualquer, pois ele deve ser ignorado;
- IDCAM com o código referente ao primeiro sinal; e
- PAR com o código associado ao parâmetro CAN.

Por outro lado, caso o campo IDCAM identifique um campo diferente dos quatro primeiros, o código especificado em PAR deve ser analisado. Como é possível prever, para cada parâmetro não-suportado, é adicionado na mensagem PMR um código que identifica que parâmetro é esse. Imagine que o número de parâmetros não-suportados (inaceitáveis) seja

três. Nesse caso, são adicionados três identificadores de parâmetros que apontam quais exatamente não são suportados pelo Sistema Servidor. O formato desse código é o seguinte:

- IDPAR (3 *bits*): identificador do parâmetro não-suportado.

O formato geral de uma mensagem PMR identificada pelo método RESCNXF pode ser visto na Figura 35.

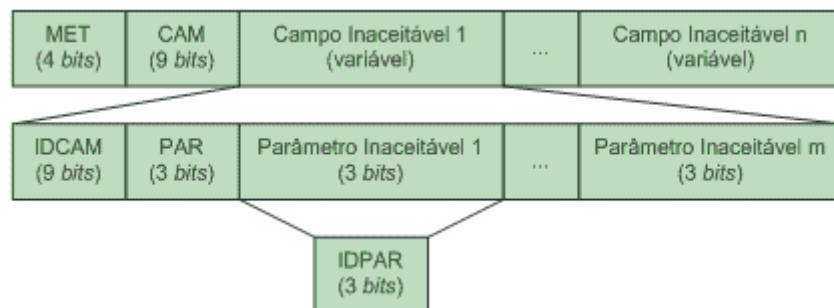


Figura 35: formato das mensagens PMR do tipo RESCNXF e RESRCNFF.

Os códigos, e seus significados, dos campos que compõem uma mensagem identificada pelo método RESCNXF, que ainda não foram especificados neste texto, são apresentados da Tabela 16 a Tabela 17.

Tabela 16: codificação do campo CAM.

Nº de Campos	Código (9 <i>bits</i> )
1	0
2 a 512	CAM - 1
512	511

Tabela 17: codificação do campo IDCAM.

Id. do Campo	Código (9 <i>bits</i> )
1	0
2 a 512	IDCAM - 1
512	511

A seguir, são descritos os dois últimos métodos de resposta. Esses métodos foram desenhados para responderem requisições de reconfiguração de parâmetros de monitoramento. Como requisições desse tipo podem ser enviadas e recebidas por ambos os sistemas, Cliente e Servidor, esses dois métodos também podem ser utilizados por ambos os sistemas.

O primeiro deles, RESRCNFS, deve ser utilizado quando a ponta requisitada aceita realizar a reconfiguração especificada na mensagem PMR do tipo REQRCNF que deu início ao processo de reconfiguração. O formato da mensagem PMR identificada por esse tipo de resposta (Figura 36) é fundamentalmente simples, pois a única informação que se deseja

transmitir à ponta requisitante é que a sua requisição de reconfiguração de parâmetros foi aceita. Nesse sentido, é necessário apenas um campo:

- MET (4 *bits*): identificador do método igual a RESRCNFS.



Figura 36: formato da mensagem PMR do tipo RESRCNFS.

O último método desta classe recebeu o nome de RESRCNFF. Ele também deve ser utilizado pela ponta requisitada a reconfigurar parâmetros de monitoramento, porém, com o objetivo de comunicar à ponta requisitante que tal reconfiguração não pôde ser realizada.

Além dessa informação, a ponta requisitada também informa, ao utilizar esse método, à ponta requisitante, quais parâmetros de quais campos presentes na mensagem PMR do tipo REQRCNF que deu início ao processo de reconfiguração não são suportados. Esse tipo de resposta é o mesmo utilizado pelo método RESCNXF. Portanto, apesar dos métodos RESRCNFF e RESCNXF serem de fato métodos distintos e com propósitos diferentes, eles compartilham o mesmo formato de mensagem PMR (Figura 35).

#### 4.1.3 Métodos de Notificação

Esta é a terceira e última classe de métodos do PMR. Ela é composta por cinco métodos que são utilizados para transmissão de notificações diversas. Nesta classe, há métodos que devem ser utilizados somente por Sistemas Clientes, outros que somente devem ser utilizados por Sistemas Servidores e ainda os que podem ser utilizados por ambos os sistemas. É importante salientar que o propósito geral dos métodos desta classe é transmitir, como o próprio nome da classe já diz, uma notificação. A ponta notificada, portanto, recebe esse tipo de mensagem e não tem nenhum tipo de obrigação para com a ponta notificante.

O primeiro método de notificação a ser descrito é o NOTBLCAMO (notificação de BLOCO de amostras). Sua utilização deve se dar somente por Sistemas Clientes que desejam transmitir, a um Sistema Servidor, um BLOCO de amostras de um determinado sinal monitorado. Os campos que definem uma mensagem PMR desse tipo são os seguintes:

- MET (4 *bits*): identificador do método igual a NOTBLCAMO;

- REL (20 *bits*): instante no qual a primeira amostra, do BLOCO a ser transmitido, foi realizada;
- IDSIN (8 *bits*): identificador do sinal monitorado associado ao BLOCO a ser transmitido; e
- BLOCO (tamanho dependente dos parâmetros AMO, CAN e BIT): BLOCO de amostras que se deseja notificar.

Como já foi explicado na Subseção 3.1.2, um BLOCO é uma entidade abstrata de dados composta por certo número de amostras, que por sua vez são formadas por certo número de canais, que são expressos em certa quantidade de *bits*. Esse tratamento generalista, que também é realizado pelo PMR, traz dois grandes benefícios ao sistema de monitoramento remoto proposto, a saber: diversos tipos de sinais analógicos podem ser monitorados através deste protocolo; e diversos BLOCOS com características distintas, muitas vezes ditadas por restrições de *hardware* de um determinado dispositivo móvel, podem ser transmitidos através deste protocolo.

Para que esse mecanismo de parametrização possa ser bem entendido, dois exemplos são detalhados a seguir.

Imagine um primeiro tipo de monitoramento remoto que considera um sinal analógico de pressão arterial humana. Partindo do princípio que o Sistema Cliente utiliza um conversor A/D que forneça seus resultados em 8 *bits* e que se transmita BLOCOS compostos por 1.200 amostras. Como esse tipo de sinal exige somente um canal, seriam necessários 1.200 *bytes* ( $1.200 \text{ amostras} * 1 \text{ canal por amostras} * 8 \text{ bits por canal} = 9.600 \text{ bits} = 1.200 \text{ bytes}$ ) para armazenar um único BLOCO.

Outro exemplo poderia ser um monitoramento de sinal de ECG composto por três canais. Se um conversor A/D de 10 *bits* fosse utilizado e um BLOCO fosse sempre formatado por 400 amostras, o tamanho de tal BLOCO seria de 1.500 *bytes* ( $400 \text{ amostras} * 3 \text{ canais por amostra} * 10 \text{ bits por canal} = 12.000 \text{ bits} = 1.500 \text{ bytes}$ ).

A Figura 37 exhibe o arranjo entre os campos que compõem uma mensagem PMR identificada por esse primeiro método de notificação.

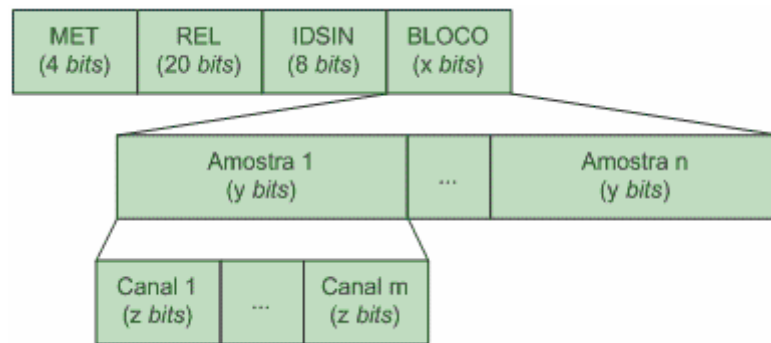


Figura 37: formato da mensagem PMR do tipo NOTBLCAMO.

O próximo método a ser descrito é o NOTREQBLCAMO (notificação de requisição de BLOCO de amostras). Esse é um dos métodos de utilização exclusiva dos Sistemas Servidores. Sua utilização deve se dar em momentos em que um Sistema Servidor deseja obter amostras atualizadas de um determinado sinal que já esteja sendo monitorado. Cabe salientar que o envio de uma mensagem PMR desse tipo ao Sistema Cliente envolvido no monitoramento não o obriga a transmitir de volta amostras atualizadas do sinal monitorado. Isso porque esse método é essencialmente um método de notificação, embora seu nome contenha as iniciais da palavra requisição.

Para que se possa ter uma idéia da utilidade desse método, pense no cenário descrito a seguir. Um sistema de monitoramento remoto realizando um monitoramento preventivo de um determinado sinal analógico a cada 30 minutos. Por algum motivo qualquer, 5 minutos após o envio da última amostra realizada, os operadores desse sistema são levados a acreditar que há algo de errado com o sinal monitorado. Para que esses operadores não se sintam obrigados a esperar 25 minutos para que uma nova amostra seja notificada, correndo-se riscos associados à anormalidade desse sinal, o método NOTREQBLCAMO pode ser utilizado.

A formatação de uma mensagem PMR identificada por esse método (Figura 38) é feita a partir de somente dois campos, a saber:

- MET (4 bits): identificador do método igual a NOTREQBLCAMO; e
- IDSIN (8 bits): identificador do sinal monitorado associado ao BLOCO de amostras que está sendo requisitado.



Figura 38: formato da mensagem PMR do tipo NOTREQBLCAMO.



Além de prover facilidades ao monitoramento de sinais analógicos diversos, o PMR também possibilita acionamentos e atendimentos de alarmes, que são transmitidos, de uma ponta da conexão à outra, utilizando-se um dos dois métodos descritos a seguir.

O primeiro é o NOTALRMAC (notificação de alarme acionado). Como seu próprio nome revela, sua utilização deve ser feita por Sistemas Clientes quando um de seus alarmes for acionado. Do ponto de vista deste protocolo, não importa o mecanismo de detecção de alarme utilizado, pois essa é uma responsabilidade do *firmware*. O que realmente importa é que a detecção foi feita e que o método NOTALRMAC pode ser utilizado para notificar tal acionamento ao Sistema Servidor envolvido no monitoramento.

Um exemplo de como um acionamento de alarme pode ser detectado é via inclusão de um procedimento de checagem de sinal no próprio *firmware* do Sistema Cliente. Esse procedimento pode ser configurado para detectar um valor de sinal abaixo de um limiar inferior ou acima de um limiar superior, previamente configurados. Esse mecanismo é, portanto, automático e executado pelo próprio dispositivo móvel.

Outra possibilidade é passar para o usuário do dispositivo móvel a responsabilidade de detectar alarmes. Um caso em que essa possibilidade é utilizada de fato é em monitoramentos de pacientes que podem acionar alarmes em situações de desconforto ou emergência. Perceba que nessas situações, os pacientes têm que detectar o estado de desconforto ou emergência para, então, acionar o alarme a ser transmitido ao Sistema Servidor.

A seguir são listados os campos utilizados na composição de uma mensagem PMR do tipo NOTALRMAC:

- MET (4 *bits*): identificador do método igual a NOTALRMAC;
- IDALRM (4 *bits*): número desejado a ser associado ao alarme acionado; e
- NALRM (160 *bits* = 20 caracteres): descritor desejado para o alarme acionado.

A disposição desses campos em uma mensagem PMR de acionamento de alarme é feita de acordo com a Figura 39.



Figura 39: formato da mensagem PMR do tipo NOTALRMAC.

O segundo método relacionado a alarmes é o NOTALRMAT (notificação de alarme atendido). Sua utilização deve ser feita somente por Sistemas Servidores, já que foram considerados no sistema de monitoramento proposto apenas acionamentos de alarmes realizados somente por Sistemas Clientes.

Cabe salientar que não é de responsabilidade do PMR garantir que um alarme que esteja sendo atendido por um Sistema Servidor tenha sido realmente acionado por um Sistema Cliente. Este protocolo foi desenvolvido para criar os mecanismos de transmissão de acionamentos e atendimentos de alarmes e não para determinar certa política de gerenciamento desse tipo de transmissão.

Ao seguir essa linha, estabeleceu-se que cada aplicação, seja ela Cliente ou Servidora, é livre para definir sua própria política de transmissão de mensagens relacionadas a alarme. Por exemplo, uma determinada Aplicação Cliente, projetada para ser o mais simples possível, poderia ser implementada de forma que nenhum atendimento de alarme seja aguardado. Já outra Aplicação Cliente poderia ser projetada para deixar aceso um dos LEDs presentes no dispositivo móvel envolvido. Nesse caso, seria necessário aguardar recepções de atendimentos de alarmes.

Por outro lado, caso uma determinada política tivesse sido implementada como parte do PMR, todos os seus usuários seriam obrigados a respeitar tal política, o que seria um ponto de inflexibilidade considerável.

Os campos que compõem esse tipo de mensagem PMR são:

- MET (4 *bits*): identificador do método igual a NOTALRMAT; e
- IDALRM (4 *bits*): número associado ao alarme atendido.

O formato de uma mensagem PMR cujo objetivo é transmitir uma notificação de alarme atendido pode ser visualizado na Figura 40.

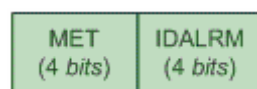


Figura 40: formato da mensagem PMR do tipo NOTALRMAT.

O último método do PMR a ser descrito, o que não o faz menos importante, é o NOTDCNX (notificação de desconexão). Uma mensagem desse tipo deve ser enviada, por

qualquer uma das pontas, antes de um encerramento voluntário de uma conexão PMR. Nesse sentido, é possível perceber que o PMR foi desenhado para que iniciativas de desconexões possam ser tomadas tanto por Sistemas Clientes quanto por Sistemas Servidores.

Como o propósito desse método é bastante simples, apenas notificar que uma ponta está se desconectando da outra, uma mensagem PMR identificada por ele também o será. Uma mensagem desse tipo é formada por apenas um campo, justamente o que a identifica:

- MET (4 *bits*): identificador do método igual a NOTDCNX.

## 4.2 MÁQUINAS DE ESTADOS DO PMR

No início deste capítulo, a diretriz de projeto que definiu como necessária a utilização de estados de conexão foi devidamente justificada. Foi explanado também que cada entidade PMR, ou seja, cada ponta da conexão PMR, deve manter de forma independente sua própria variável de estado.

Durante todo o período em que uma conexão PMR é mantida aberta, tais variáveis de estado são utilizadas para ditarem o comportamento de cada entidade PMR envolvida. Além de definir métodos, e seus respectivos formatos, que podem ser utilizados para transferir informações de uma ponta da conexão à outra, este protocolo de comunicação também especifica seus possíveis estados e comportamentos. Isso é realizado por meio de duas máquinas de estados, uma para cada tipo de entidade PMR.

É válido ressaltar que uma determinada implementação do PMR deve respeitar, estritamente, a máquina de estados correspondente ao seu tipo (cliente ou servidor). Em outras palavras, essa implementação deve realizar somente o que é especificado pela máquina de estados correspondente. Caso contrário, sua interação com outras entidades PMR estará comprometida.

### 4.2.1 Máquina de Estados do PMR Cliente

A Figura 41 exibe a máquina de estados que deve ser respeitada pelas entidades PMR utilizadas em Sistemas Clientes. Nessa figura, as setas contínuas representam mudanças de estados ocasionadas por ações da Aplicação Cliente. Já as setas tracejadas indicam mudanças de estados ocorridas por recepções de mensagens PMR enviadas por Sistemas Servidores (a outra ponta da conexão). Para cada mudança de estado, são especificados nesta máquina de

estados o evento responsável pela alteração de estado e a ação que a entidade PMR envolvida deve realizar. É importante perceber que as marcações a e b nesta máquina de estados são utilizadas para indicar ligações entre duas setas, uma que sai de um estado e outra entra em outro estado. Esse artifício propiciou a obtenção de uma máquina de estados menos “poluída” e de interpretação mais fácil.

Para exemplificar o que acaba de ser dito, considere que o estado atual de uma determinada entidade cliente seja DESCONNECTADO. Como se pode verificar na máquina de estados, a única mudança de estado possível nessa circunstância ocorre no caso da Aplicação Cliente gerar um evento de solicitação de conexão. Quando isso acontece, essa entidade deve, além de alterar o estado atual para CONFIGURANDO SERVIDOR, enviar uma mensagem PMR do tipo REQCNX ao PMR servidor.

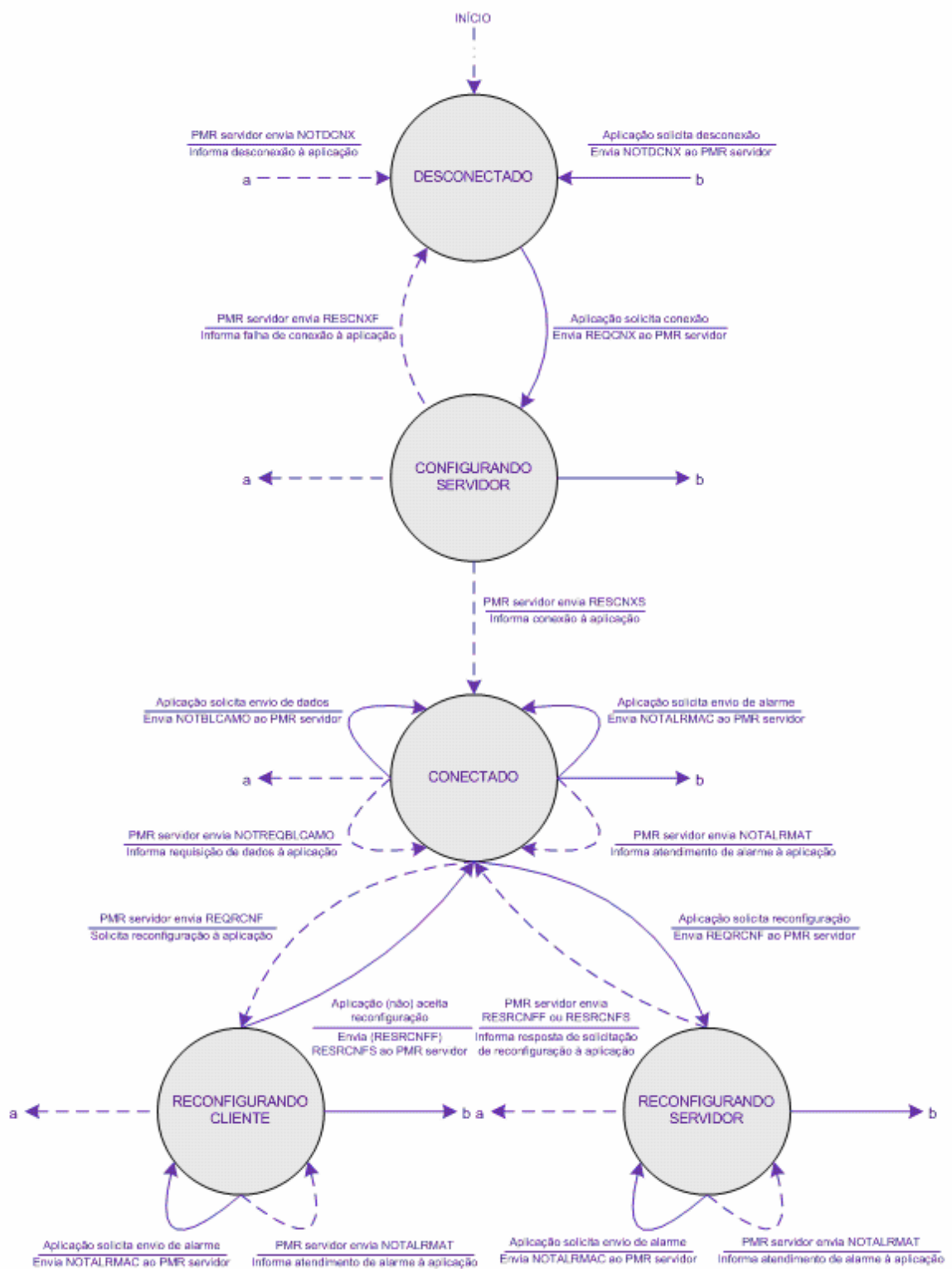


Figura 41: máquina de estados do PMR cliente.

#### 4.2.2 Máquina de Estados do PMR Servidor

A máquina de estados que deve ser utilizada, exclusivamente, pelas entidades PMR executadas por Sistemas Servidores é apresentada na Figura 42. Em um primeiro momento, seu funcionamento pode parecer idêntico ao da máquina de estados do PMR cliente. Todavia, ao confrontá-las, é possível perceber algumas particularidades importantes. De qualquer forma, essas máquinas de estados possuem certo grau de similaridade, o que é natural e comum entre os protocolos *statefull*.

Por conta dessa similaridade, a leitura desta máquina de estados é feita de forma ligeiramente diferente à da máquina de estados do PMR cliente. As setas contínuas representam modificações de estados decorrentes de ações da Aplicação Servidora envolvida. Por outro lado, as setas tracejadas indicam alterações de estados ocasionadas por recepções de mensagens PMR enviadas por Sistemas Clientes (a outra ponta da conexão). As marcações c e d nesta máquina de estados também são utilizadas para indicar ligações entre duas setas, uma que sai de um estado e outra entra em outro estado.

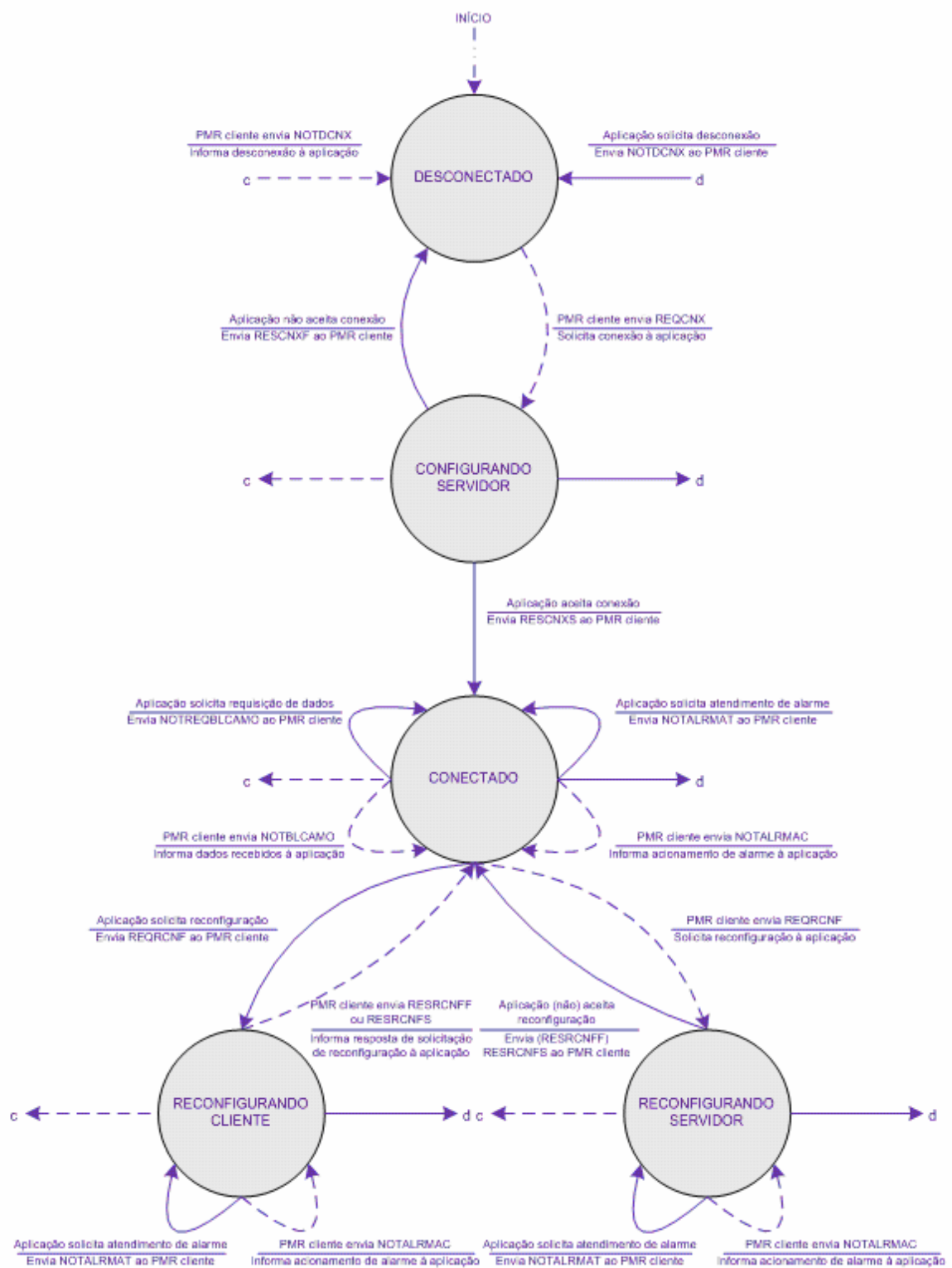


Figura 42: máquina de estados do PMR servidor.

### 4.3 CONCLUSÕES

O desafio de desenvolver um protocolo aberto e adequado às necessidades da solução proposta foi concluído com êxito. A principal justificativa para esse desenvolvimento foi o fato de não ter sido encontrado nenhum protocolo aberto com características similares o bastante.

Para o desenvolvimento do PMR, foram estabelecidos os seguintes requisitos: mínimo *overhead*, sem perda de dados, entrega ordenada de dados e capacidade de transmissão de diversos sinais analógicos (inclusive sinais vitais). Todos esses requisitos foram alcançados com as seguintes escolhas de projeto:

- utilização do *TCP* como protocolo de transporte;
- orientação a *bit* na formação dos campos das mensagens PMR;
- orientação a conexão;
- comunicação ponto a ponto entre Sistema Cliente e Sistema Servidor; e
- utilização da característica *statefull*.

Além do PMR ter sido projetado neste trabalho, grande parte da sua especificação também foi implementada tanto no Sistema Cliente (com a linguagem C) quanto no Sistema Servidor (com a linguagem Java) que compõem o protótipo desenvolvido. Tais implementações possibilitaram uma avaliação de desempenho da solução proposta e, também, evidenciou sua factibilidade. Essa avaliação também faz parte desta dissertação e é descrita no Capítulo 5.

É extremamente pertinente notar que o PMR possui um grande potencial. Isso porque ele se mostra adequado a transmissão de diversos tipos de sinais analógicos, inclusive sinais não relacionados à Telemedicina. Alguns exemplos de aplicações que podem se beneficiar com a utilização do PMR são:

- monitoramento de vazão em gasodutos;
- monitoramento de tensão e/ou corrente em redes elétricas;
- monitoramento de variáveis associadas a clima: temperatura, umidade, velocidade de ventos, pluviosidade, etc.; e
- monitoramento de diversos sinais relacionados à instrumentação industrial: nível, velocidade, pressão, tensão, corrente, temperatura, etc.



## 5 EXPERIMENTOS E RESULTADOS

Em projetos como este, se mostra fundamental analisar o quão bem o sistema proposto funciona e sob quais circunstâncias isso acontece. Assim sendo, realizou-se uma fase de avaliação de desempenho em que foram executados diversos experimentos bem definidos metodologicamente. Cada um destes experimentos é detalhadamente descrito nas seções a seguir.

Na Seção 5.1, os experimentos mais gerais e seus resultados são apresentados. Esses são os experimentos em que todos os componentes do sistema de monitoramento proposto são utilizados. Em seguida, na Seção 5.2 são abordados os experimentos mais relacionados a variáveis de redes de computadores, em especial as envolvidas com o protocolo *TCP*. Por fim, são descritos na Seção 5.3, os experimentos realizados especificamente para avaliar a qualidade do dispositivo móvel que compõe o protótipo desenvolvido. Essa sequência de apresentação foi escolhida de forma a facilitar a análise dos resultados e suas eventuais relações.

Resultados de 6 variáveis, do total de 8, são expressos por meio de seus valores médios calculados. Duas ferramentas tradicionais de análise estatística fornecidas pelo MATLAB foram utilizadas nesta dissertação: *anova1* e *multcompare*. Basicamente, tais ferramentas possibilitam verificar se dois ou mais grupos de valores possuem médias significativamente diferentes em termos estatísticos. Todas as análises estatísticas foram realizadas da seguinte forma:

- realizou-se agrupamentos de valores obtidos para uma determinada variável de interesse. Os critérios utilizados em tais agrupamentos são descritos em cada uma das subseções deste capítulo;
- executou-se o *anova1* para determinados grupos de valores construídos anteriormente;
- analisou-se os resultados dessa primeira ferramenta, mais especificamente o valor “Prob>F”, para determinar se tais grupos têm médias significativamente diferentes. O critério utilizado para isso foi o seguinte: se “Prob>F”  $\leq 0,05$ , foi considerado que há diferença significativa entre as médias de cada grupo. Se “Prob>F”  $> 0,05$ , foi considerado que esse tipo de diferença não existe; e
- a ferramenta *multcompare* foi utilizada para investigar quais grupos possuem médias significativamente diferentes entre si.

A realização dos experimentos que, de alguma forma, exigiram transferência de dados via rede *GPRS*, se deu sob as seguintes circunstâncias:

- utilizando-se a infra-estrutura de telecomunicações da operadora TIM;
- dispositivo móvel localizado no Laboratório de Controle e Instrumentação (LCI), que fica no prédio CT-II da Universidade Federal do Espírito Santo (UFES); e
- Servidor instalado na rede *Fast Ethernet* do LCI. Essa rede é interligada, por meio de um *link Gigabit Ethernet*, ao roteador do Departamento de Engenharia Elétrica, que por sua vez se conecta ao roteador do Núcleo de Processamento de Dados da UFES através de outro *link Gigabit Ethernet*.

## 5.1 QUALIDADE DO SISTEMA DE MONITORAMENTO

Nesta seção, são descritos dois experimentos cujos resultados revelam, de forma bem objetiva, a qualidade do sistema desenvolvido quando em funcionamento. Cabe salientar que é importante o leitor relacionar tais resultados com outros resultados mais específicos que são abordados nas seções seguintes.

### 5.1.1 Tempo Médio de Conexão

Tempo de conexão se refere ao período entre um dispositivo móvel ser ligado e estar apto a transferir amostras e/ou acionamentos de alarme. Esse período, na verdade, é composto pelo somatório dos tempos de: inicialização da *MCU*; inicialização do Modem *GPRS*, que demanda aproximadamente 7 segundos; realização de chamada *PPP*; estabelecimento de conexão *TCP*; e estabelecimento de conexão PMR.

O monitoramento desta variável foi realizado de acordo com a seguinte metodologia:

- foram realizadas ao todo 210 conexões, no decorrer de 7 dias seguidos, nos períodos manhã, tarde e noite;
- o primeiro dia de monitoramento se deu em uma quinta-feira e o último em uma quarta-feira;
- a Tabela 18 mostra o dia e o horário em que cada conexão foi realizada; e
- a obtenção do tempo médio de conexão foi feita a partir da média aritmética simples dos 210 tempos de conexão.

Tabela 18: cronograma dos experimentos de medição de tempo de conexão.

Dia [1, 3, 5 e 7]			Dia [2, 4 e 6]		
Período	Conexões	Horário	Período	Conexões	Horário
Manhã	10	~9h	Manhã	10	~11h
Tarde	10	~14h	Tarde	10	~16h
Noite	10	~19h	Noite	10	~21h

A medição de cada tempo de conexão foi realizada utilizando-se um kit de desenvolvimento composto por, dentre outros dispositivos, uma *FPGA*<sup>13</sup> e 4 dígitos de sete segmentos. Esse *hardware* foi programado para desempenhar a função de cronômetro digital com precisão na ordem de centésimo de segundo, exibindo em seus dígitos uma contagem circular no intervalo [0; 5.999] que representava um período entre 0 e 59,99 segundos (60 segundos de período).

O *firmware* do Sistema Cliente foi executado na *MCU* do dispositivo móvel e um de seus pinos de entrada e saída foi utilizado para sinalizar os eventos de início e fim do período de conexão. A *FPGA* foi programada justamente para medir o tempo entre tais eventos, mostrando o resultado nos 4 dígitos do kit de desenvolvimento. Cada tempo foi devidamente registrado em planilha eletrônica para facilitar os cálculos estatísticos.

Na Figura 43, é apresentado como o período do dia influenciou no tempo médio de conexão. Para a obtenção do tempo médio de conexão durante o período da manhã, foram consideradas todas as conexões (70) estabelecidas nas manhãs dos 7 dias da fase de avaliação de desempenho. Procedimentos análogos foram realizados para os períodos da tarde e da noite. O desvio-padrão (DP) e o coeficiente de variação (CV) dos tempos médios de conexão obtidos em cada período do dia também são apresentados na Figura 43.

<sup>13</sup> *FPGA* é um acrônimo para *Field Programmable Gate Array*, um tipo de dispositivo largamente utilizado para processamento de informações digitais.

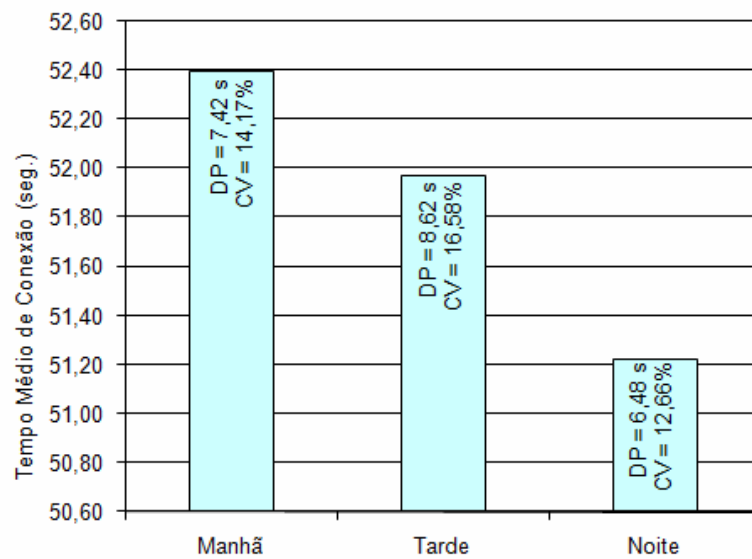


Figura 43: tempo médio de conexão por período do dia.

Ao se analisar os dados da Figura 43 com a ferramenta *anova1*, verificou-se que o período do dia não influenciou estatisticamente no tempo médio de conexão, pois “Prob>F” foi de 0,6469. Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura 44.

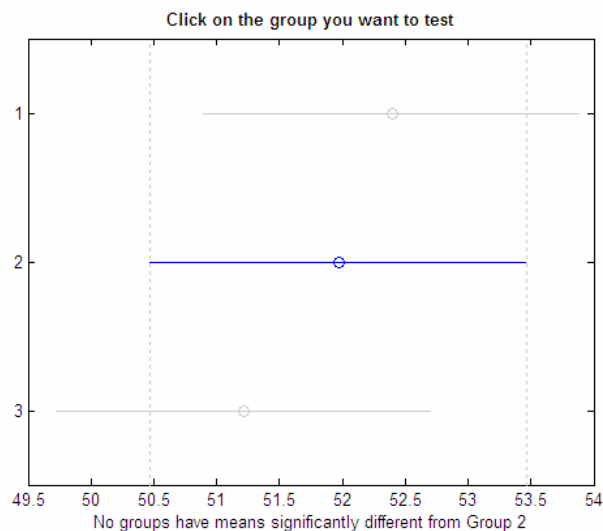


Figura 44: gráfico obtido com a ferramenta *multcompare* para o tempo médio de conexão por período do dia (grupo 2 selecionado aleatoriamente).

Outras análises podem ser feitas a partir do gráfico mostrado na Figura 45, onde são relacionados os tempos médios de conexão e os dias em que esses experimentos foram realizados. Para a obtenção do tempo médio de conexão durante o dia 1, foram consideradas todas as conexões (30) estabelecidas nos três períodos desse dia. Procedimentos análogos

foram realizados para os demais dias. O desvio-padrão (DP) e o coeficiente de variação (CV) dos tempos médios de conexão obtidos em cada dia também são apresentados na Figura 45.

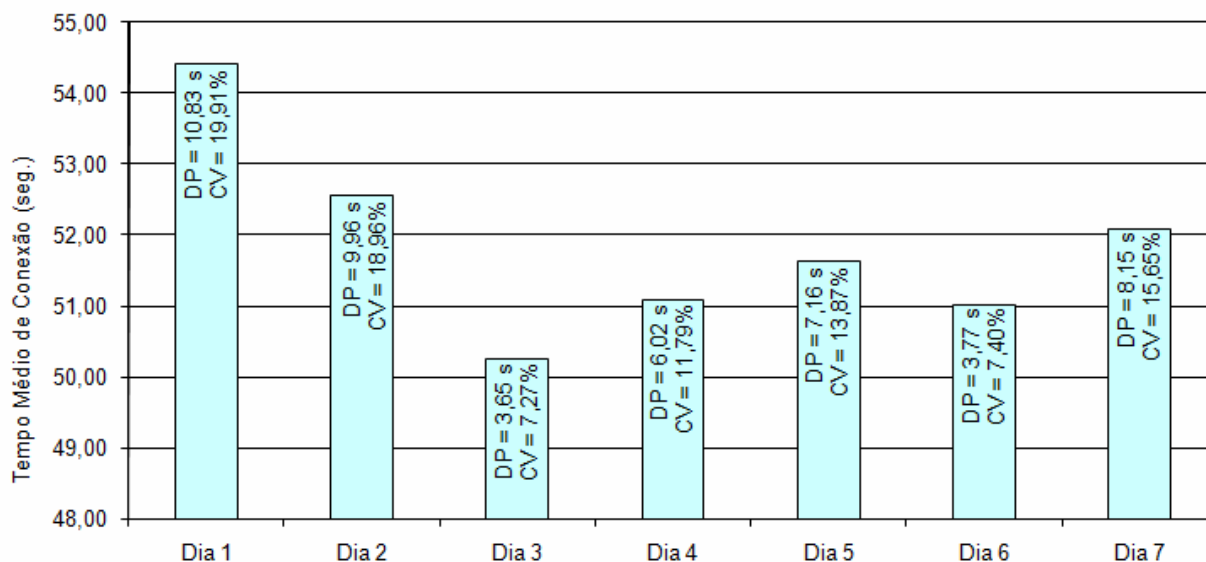


Figura 45: tempo médio de conexão por dia.

Ao se analisar os dados da Figura 45 com as ferramentas *anova1*, verificou-se que o dia da semana não influenciou estatisticamente no tempo médio de conexão, pois “Prob>F” foi de 0,4517. Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura 46.

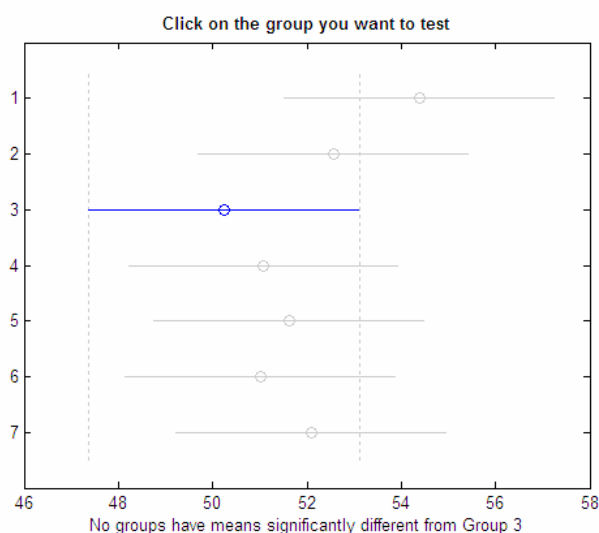


Figura 46: gráfico obtido com a ferramenta *multcompare* para o tempo médio de conexão por dia (grupo 3 selecionado aleatoriamente).

Considerando-se todas as conexões, sem distinção de período ou dia em que foram realizadas, o tempo médio de conexão, seu desvio-padrão e seu coeficiente de variação foram de, respectivamente: 51,86 segundos, 7,54 segundos e 14,53%.

### 5.1.2 Atraso Médio de Amostra

No contexto desta dissertação, a variável atraso de amostra refere-se ao período de tempo entre a produção de uma amostra, realizada por um Sistema Cliente, e sua completa recepção por um Sistema Servidor. Neste ponto, é extremamente importante ficar claro que a variável atraso de amostra não se refere ao tempo de transmissão de um BLOCO, muito embora esse tempo esteja incluído nesta variável.

Observam-se duas componentes no atraso de certa amostra: o tempo entre sua produção e o seu envio; e o tempo entre seu envio e sua recepção. É intuitivo perceber que quanto maior for o número de amostras que compõem um BLOCO, maiores serão essas duas componentes. No intuito de se analisar como esse comportamento se dá de fato, foram realizados experimentos variando-se justamente o número de amostras que compõem um BLOCO, em outras palavras, variando-se o tamanho da mensagem PMR.

O monitoramento dessa variável foi realizado de acordo com a seguinte metodologia:

- a frequência de amostragem do sinal monitorado utilizada foi de 200 Hz;
- ao final de todos os experimentos, foram transferidas exatamente 453.600 amostras, 75.600 para cada tamanho de mensagem PMR considerado (6 ao todo);
- foi obtido um atraso médio associado a cada tamanho de BLOCO; e
- a Tabela 19 apresenta as informações sobre cada tamanho de mensagem considerado. Já a Tabela 20 relaciona os horários e os dias em que foram realizados os experimentos.

Tabela 19: relações entre tamanho de BLOCO, tamanho da mensagem e número de amostras transmitidas em cada experimento de medição de atraso de amostra.

Amostras por BLOCO	Tamanho da Mensagem PMR ( <i>bytes</i> )	Amostras Transmitidas por Experimento
60	229	3.600
120	454	3.600
180	679	3.600
240	904	3.600
300	1129	3.600
360	1354	3.600

O tamanho de uma mensagem PMR cuja função é transmitir dados sobre um sinal monitorado é determinado pela soma entre o tamanho do seu cabeçalho (32 *bits*) e o tamanho do seu corpo (um BLOCO). No caso da terceira linha da Tabela 19, por exemplo, o tamanho

da mensagem PMR é calculado da seguinte forma: 4 *bytes* de cabeçalho mais 675 *bytes* de corpo (180 amostras \* 30 *bits*/amostra).

Tabela 20: cronograma de realização dos experimentos de medição de atraso de amostra.

Dia [1, 3, 5 e 7]			Dia [2, 4 e 6]		
	Amostras por BLOCO	Horário		Amostras por BLOCO	Horário
Manhã	60	~9h	Manhã	60	~11h
	120			120	
	180			180	
	240			240	
	300			300	
	360			360	
Tarde	60	~14h	Tarde	60	~16h
	120			120	
	180			180	
	240			240	
	300			300	
	360			360	
Noite	60	~19h	Noite	60	~21h
	120			120	
	180			180	
	240			240	
	300			300	
	360			360	

Devido ao fato dos dois sistemas, Cliente e Servidor, possuírem relógios dessincronizados antes de uma conexão PMR, a tarefa de medir o atraso de amostra com uma boa precisão não é simples. Portanto, a obtenção desta variável foi realizada seguindo-se o esquema apresentado na Figura 47.

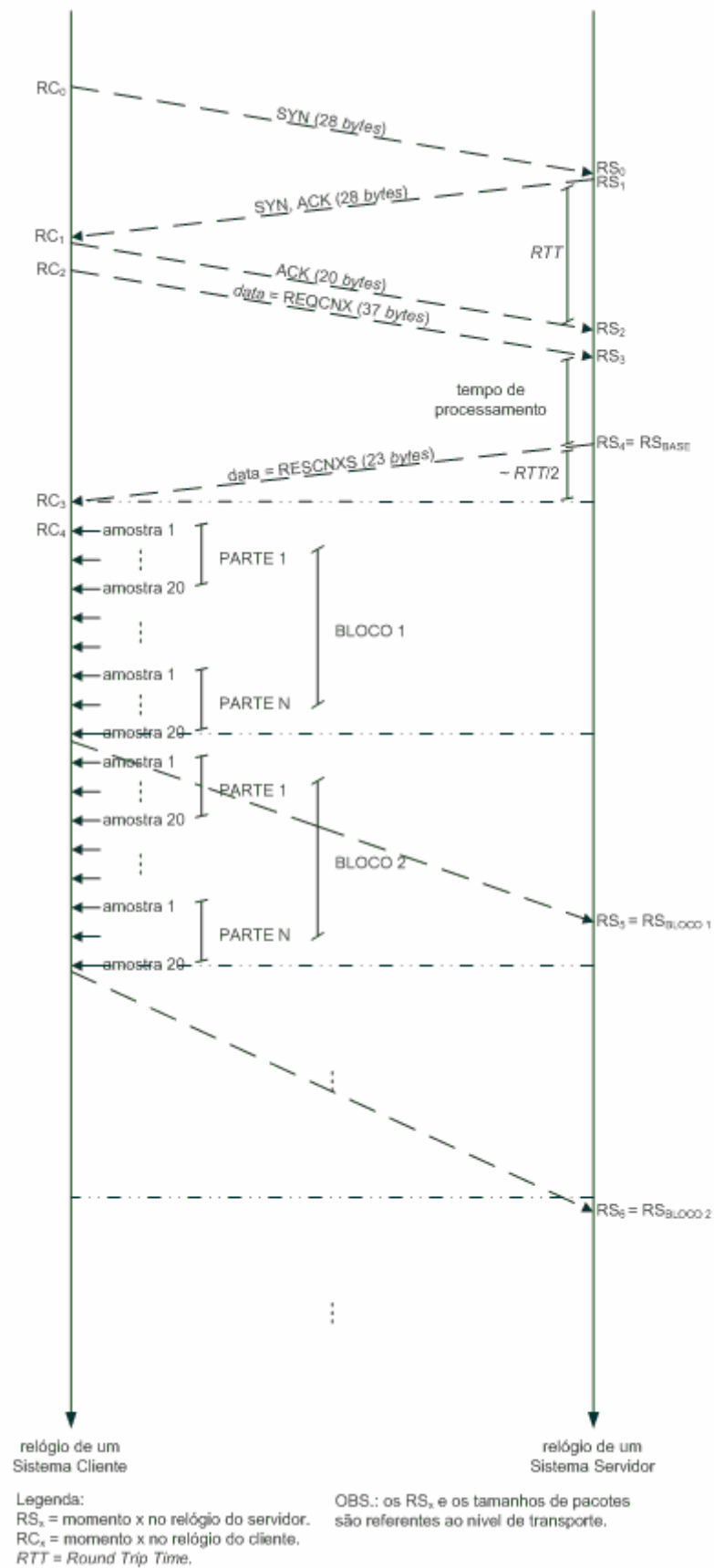


Figura 47: esquema de medição do atraso de amostra.



Para a medição do atraso de cada amostra, optou-se por utilizar somente as informações de relógio do microcomputador Servidor envolvido na comunicação ( $RS_0, RS_1, \dots$ ). Isso porque essas informações são suficientes para tais medições, além de serem mais acessíveis em relação às informações de relógio do dispositivo móvel ( $RC_0, RC_1, \dots$ ).

Todas as informações relacionadas a transmissão de dados foram obtidas por meio do analisador de protocolos *Wireshark*, um *sniffer*<sup>14</sup> de rede com opções avançadas de filtragem e classificação de campos de diversos protocolos abertos [27]. É importante perceber que as informações de relógio e de tamanhos de pacotes apresentados na Figura 47 são referentes à camada de transporte da arquitetura *TCP/IP*.

O esquema de medição apresentado na Figura 47 foi utilizado da forma descrita a seguir. Com as informações de tempo dos segmentos utilizados na abertura de conexão *TCP* (*three way handshake*), obtém-se uma ótima estimativa do  $RTT$ <sup>15</sup> da rede. Filtrando-se os segmentos *TCP* recebidos/enviados, com relação aos seus tamanhos de carga-útil, é possível obter o momento em que o Sistema Servidor envia a mensagem PMR que confirma a abertura de conexão PMR com o Sistema Cliente ( $RS_{BASE}$ ). É possível também obter os momentos em que as mensagens contendo as amostras do sinal monitorado ( $RS_{BLOCO\ X}$ ) são recebidas pelo Sistema Servidor. Inspeccionando-se o esquema de medição, não é difícil concluir que os atrasos da primeira e da segunda amostra do BLOCO 1 são obtidos por meio das Equações (1) e (2).

$$atraso\ da\ amostra_1 = RS_{BLOCO\ 1} - RS_{BASE} - \frac{RTT}{2} - 1P, \quad P = \text{período de amostragem.} \quad (1)$$

$$atraso\ da\ amostra_2 = RS_{BLOCO\ 1} - RS_{BASE} - \frac{RTT}{2} - 2P \quad (2)$$

Ao se observar as Equações (1) e (2), é possível encontrar uma equação geral para o atraso das amostras que compõem o BLOCO 1. Essa é a Equação (3).

Aplicando-se a média aritmética simples nos atrasos de todas as amostras do BLOCO 1, obtém-se a Equação (4).

<sup>14</sup> *Sniffer* é uma classe de ferramentas de *software* e/ou *hardware* capazes de interceptar e registrar tráfego de uma rede de computadores [31].

$$\text{atraso da amostra}_i = RS_{BLOCO\ 1} - (RS_{BASE} + \frac{RTT}{2} + iP), \quad \forall i \text{ tal que } 1 \leq i \leq 20N \quad (3)$$

$$\underbrace{\hspace{1.5cm}}_{\text{momento da recepção da amostra } i.} \quad \underbrace{\hspace{1.5cm}}_{\text{momento da geração da amostra } i.}$$

$$\text{atraso médio das amostras do BLOCO 1} = (RS_{BLOCO\ 1} - RS_{BASE} - \frac{RTT}{2}) - (1 + 20N)P/2 \quad (4)$$

$$\underbrace{\hspace{1.5cm}}_{\text{tempo de geração e transmissão do BLOCO 1}} \quad \underbrace{\hspace{1.5cm}}_{\text{tempo médio de geração das amostras do BLOCO 1}}$$

Se houvesse interesse em conhecer apenas o atraso médio das amostras transmitidas em um único BLOCO, bastaria a Equação (4) ser aplicada. Contudo, há de se convir que esse espaço amostral não se mostra suficiente para que uma boa precisão na média dos atrasos de amostras seja obtida. Nesse sentido, durante cada experimento, diversos BLOCOS foram transmitidos. (vide Tabela 19). Para que a média dos atrasos de todas as amostras transmitidas/recebidas durante certo tempo possa ser obtida, é necessário, portanto, aplicar novamente a média aritmética nos atrasos médios de amostras encontrados para cada BLOCO transmitido/recebido. A Equação (5) apresenta justamente essa média, que representa o atraso médio de amostra que interessa.

$$\text{atraso médio das amostras de } M \text{ BLOCOS} = \frac{\sum_{j=1}^M RS_{BLOCO\ j}}{M} - RS_{BASE} - \frac{RTT}{2} - (1 + 20N)P/2 \quad (5)$$

É importante manter em vista que a Equação (5) somente pode ser utilizada quando os tamanhos dos M BLOCOS são idênticos. Como já foi dito, uma das investigações realizadas neste trabalho foi a de como o atraso de amostra é influenciado pelo tamanho escolhido para o

<sup>15</sup> A RFC 1983 [32] define o *RTT* (*Round Trip Time*) como sendo uma aproximação do *delay* de rede. Em [26], o *RTT* é apresentado como a melhor estimativa para o tempo de percurso de ida e volta até o destino em questão.

BLOCO de amostras. Para a realização de tal investigação, faz-se importante esclarecer que a Equação (5) foi utilizada em separado para cada tamanho de BLOCO considerado (vide Tabela 19).

Sabe-se que redes do tipo *GPRS* comportam-se de forma diferente em determinados dias e/ou períodos do dia. Isso se deve, basicamente, à variação da utilização da rede por cada um de seus usuários. Ou seja, como cada usuário pode utilizar a rede de forma mais intensa em diferentes horários ou dias, é natural que todo o tráfego gerado não seja constante com o tempo. Nesse sentido, mostra-se muito interessante investigar o comportamento do atraso de amostra em função do dia da semana e também do período do dia.

Essa investigação foi realizada agrupando-se os atrasos médios de amostras transmitidas em BLOCOS de mesmo tamanho. Realizou-se tal agrupamento para cada dia da fase de avaliação de desempenho e o comportamento apresentado na Figura 48 foi obtido.

Como já foi mencionado na Subseção 5.1.1, a fase de avaliação de desempenho foi realizada em 7 dias seguidos, sendo que o primeiro se deu em uma quinta-feira e o último em uma quarta-feira.

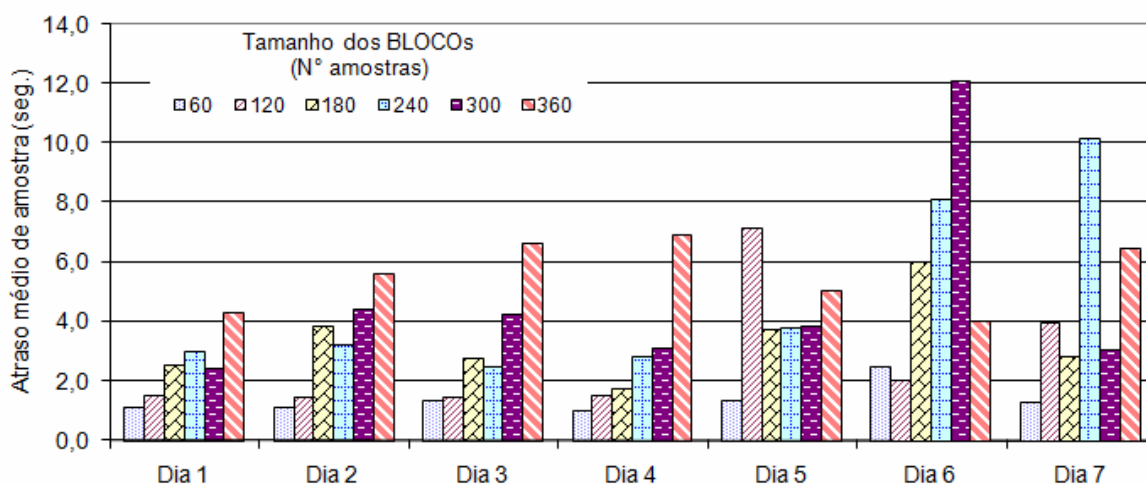


Figura 48: atraso médio de amostra em função do dia da semana e do tamanho do BLOCO.

Para a obtenção do atraso médio associado a amostras transmitidas em BLOCOS compostos por 60 amostras durante o dia 1, foram considerados todos os atrasos de amostras transmitidas nesse tipo de BLOCO durante os três períodos desse dia. Procedimentos análogos foram realizados para os outros dias e tipos de BLOCOS.

O desvio-padrão (DP) e o coeficiente de variação (CV) associados a cada tamanho de BLOCO considerado entre os atrasos médios de amostras transmitidas em cada dia da fase de avaliação de desempenho são relacionados na Tabela 21.

Tabela 21: atrasos médios de amostras, desvios-padrão e coeficientes de variação entre resultados diários.

Tamanho dos BLOCOS (Nº amostras)	Atraso Médio de Amostra (s)	DP do Atraso Médio de Amostra (s)	CV do Atraso Médio de Amostra
60	1,38	0,51	36,69%
120	2,71	2,15	79,36%
180	3,35	1,38	41,25%
240	4,78	3,04	63,60%
300	4,74	3,31	69,86%
360	5,57	1,16	20,84%

Ao se analisar os dados da Figura 48 com a ferramenta *anova1*, verificou-se que o dia da semana influenciou significativamente no atraso médio de amostra, pois “Prob>F” foi de  $1,0703 \cdot 10^{-5}$ . Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura 49. Nessa análise, o grupo 1 é constituído por atrasos médios de amostras transmitidas em BLOCOS de todos os tamanhos transmitidos durante os três períodos do dia 1. Critérios similares são utilizados para os grupos de 2 a 7.

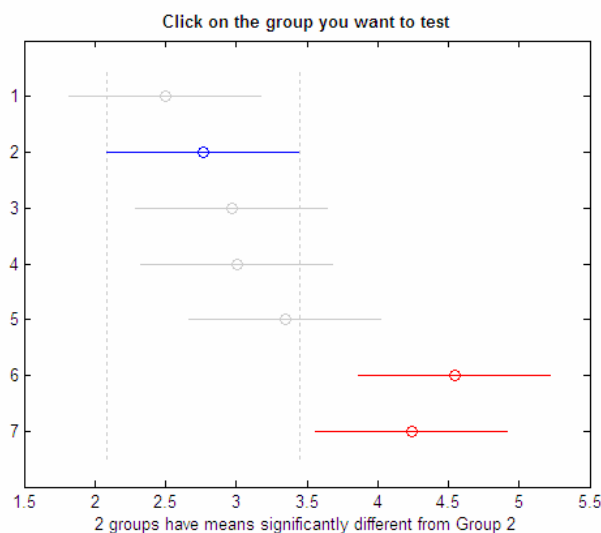


Figura 49: gráfico obtido com a ferramenta *multcompare* para o atraso médio de amostra por dia. (grupo 2 selecionado aleatoriamente)

O comportamento do atraso médio de amostra em função do período do dia foi obtido de forma análoga à anterior, no entanto, agrupando-se os atrasos médios de amostras

transmitidas em BLOCOS de mesmo tamanho para cada período do dia. Esse segundo comportamento pode ser visto na Figura 50.

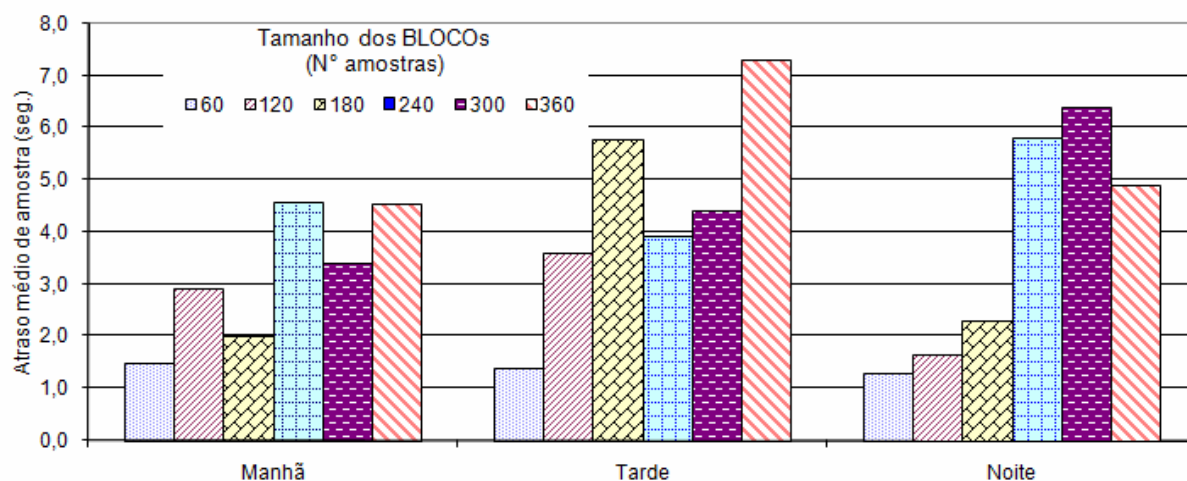


Figura 50: atraso médio de amostra em função do período do dia e do tamanho do BLOCO.

Para a obtenção do atraso médio associado a amostras transmitidas em BLOCOS compostos por 300 amostras durante o período 1, foram considerados todos os atrasos de amostras transmitidas nesse tipo de BLOCO durante o período 1 dos 7 dias da fase de avaliação de desempenho. Procedimentos análogos foram realizados para os outros períodos e tipos de BLOCO.

O desvio-padrão (DP) e o coeficiente de variação (CV) associados a cada tamanho de BLOCO considerado entre os atrasos médios de amostras transmitidas em cada período do dia são relacionados na Tabela 22.

Tabela 22: atrasos médios de amostras, desvios-padrão e coeficientes de variação entre resultados de cada período do dia.

Tamanho dos BLOCOS (N° amostras)	Atraso Médio de Amostra (s)	DP do Atraso Médio de Amostra (s)	CV do Atraso Médio de Amostra
60	1,38	0,11	7,67%
120	2,71	0,99	36,51%
180	3,35	2,10	62,85%
240	4,78	0,96	20,10%
300	4,74	1,52	32,03%
360	5,57	1,51	27,12%

Ao se analisar os dados da Figura 50 com a ferramenta *anova1*, verificou-se que o dia da semana influenciou significativamente no atraso médio de amostra, pois “Prob>F” foi de 0,0384. Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura

51. Nessa análise, o grupo 1 é constituído por atrasos médios de amostras transmitidas em BLOCOS de todos os tamanhos transmitidos durante o período da manhã dos 7 dias. Critérios similares são utilizados para os grupos 2 e 3.

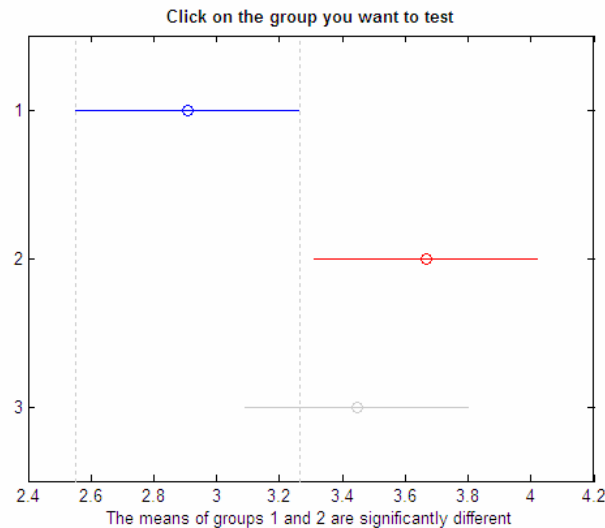


Figura 51: gráfico obtido com a ferramenta *multcompare* para o atraso médio de amostra por período do dia (grupo 1 selecionado aleatoriamente).

Analisou-se também o comportamento isolado do atraso médio de amostra em função do tamanho de BLOCO. Ou seja, foram gerados 6 grupos de atrasos médios de amostras considerando-se todos dias da semana e todos os períodos do dia em que essas foram transmitidas. Em outras palavras, o grupo 1 é composto por atrasos médios de amostras transmitidas em BLOCOS de todos os tamanhos, nos 3 períodos dos 7 dias da semana. Critérios similares foram utilizados na geração dos grupos de 2 a 6. Os resultados (Figura 52) obtidos com as ferramentas *anova1* e *multcompare* evidenciam que o tamanho do BLOCO influenciou significativamente no atraso médio de amostra, pois “Prob>F” foi de 0,00.

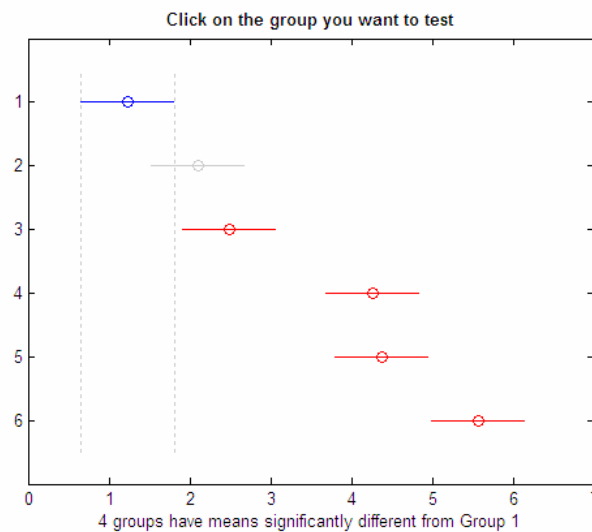


Figura 52: gráfico obtido com a ferramenta *multcompare* para o atraso médio de amostra – todos os dias e períodos (grupo 1 selecionado aleatoriamente).

## 5.2 QUALIDADE DA CONEXÃO TCP

Nesta seção, são descritos os experimentos desenvolvidos especificamente para a avaliação da conexão *TCP* sobre um enlace *GPRS* (em parte da rota). São considerados nessa avaliação três dos mais importantes parâmetros relacionados a esse protocolo, a saber: tempo de resposta, taxa de retransmissão e taxa de erro.

Novamente, deve-se ressaltar a importância da relação entre esses parâmetros aqui descritos e os que já foram abordados. De uma forma bastante simplista, vale dizer, por exemplo, que quanto maior for o tempo médio de resposta da rede, maiores serão o tempo médio de conexão e o atraso médio de amostra.

### 5.2.1 Tempo Médio de Resposta

A obtenção deste tempo médio se deu a partir dos dados fornecidos pelo analisador de protocolos de rede *Wireshark*. Ele nos permite filtrar informações e exportar dados para outros aplicativos, por exemplo o Excel da *Microsoft*, o que facilita e muito a manipulação de tais dados.

O monitoramento desta variável foi realizado conjuntamente com o monitoramento da variável atraso de amostra. Em cada abertura de conexão *TCP*, é realizado um procedimento conhecido como *three way handshake*. Entre o envio do segundo segmento, feito pelo *TCP* servidor, e a recepção deste pelo *TCP* cliente há meio *RTT* (*Round Trip Time*). Entre o envio do terceiro segmento, feito pelo *TCP* cliente, e a recepção deste pelo *TCP* servidor há mais

meio *RTT*. Para cada experimento dos 126 realizados (6 tipos de BLOCO \* 3 vezes ao dia \* 7 dias), um valor de *RTT* foi devidamente registrado. O tempo médio de resposta foi calculado considerando a média aritmética simples entre os 126 *RTTs* registrados. É possível perceber, portanto, que o tamanho do BLOCO nada interfere no tempo de resposta da rede, exatamente como se espera.

Assim como no caso da variável atraso de amostra, foi realizada também uma investigação de como o tempo de resposta se comporta em diferentes períodos do dia e em diferentes dias da semana. Ambos os agrupamentos realizados, por dia (Figura 53) e por período do dia (Figura 54), seguiram a mesma linha de raciocínio utilizada para a variável atraso de amostra.

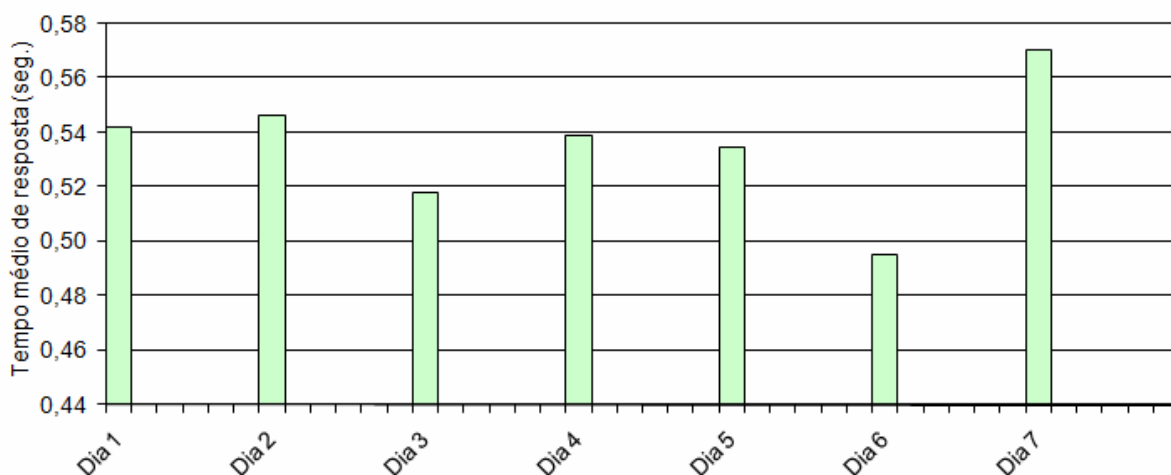


Figura 53: tempo médio de resposta em função do dia da semana.

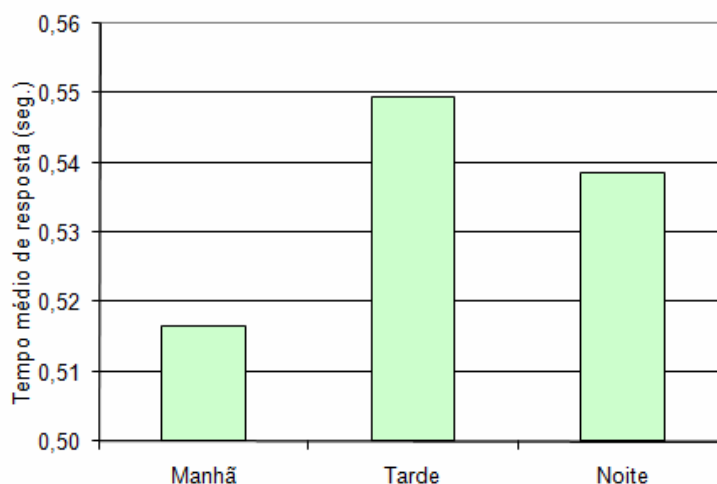


Figura 54: tempo médio de resposta em função do período do dia.

Ao se analisar os dados da Figura 53 e da Figura 54 com a ferramenta *anova1*, verificou-se que o dia da semana e o período do dia não influenciaram significativamente no



tempo médio de resposta, pois os respectivos “Prob>F” foram de 0,7835 e de 0,5320. Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura 55 e na Figura 56, respectivamente.

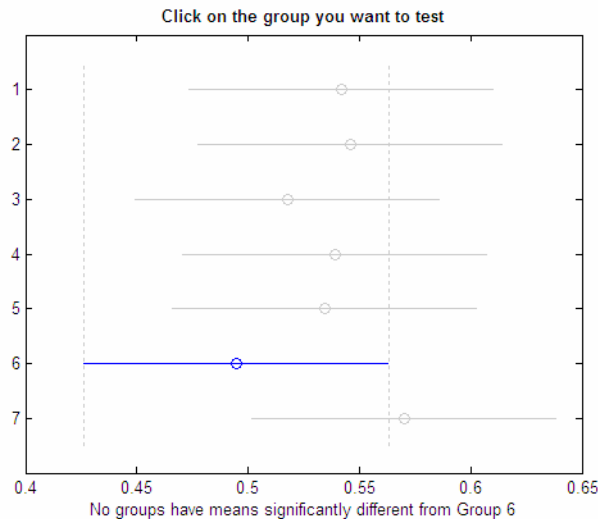


Figura 55: gráfico obtido com a ferramenta *multcompare* para o tempo médio de resposta por dia da semana (grupo 6 selecionado aleatoriamente).

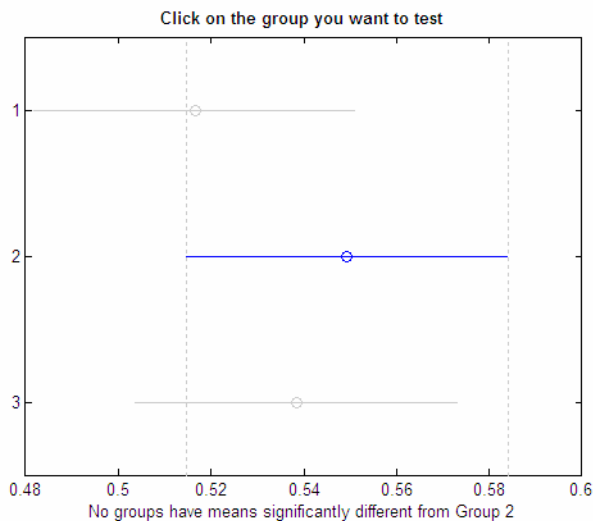


Figura 56: gráfico obtido com a ferramenta *multcompare* para o tempo médio de resposta por período do dia (grupo 2 selecionado aleatoriamente).

Considerando-se todos os *RTTs* registrados, indiferentemente do período ou dia em que as aberturas de conexão *TCP* foram realizadas, o tempo médio de resposta foi de 0,53 segundos. O desvio-padrão e o coeficiente de variação associados a essa média foram de 0,14 segundos e 25,56%.

### 5.2.2 Taxa Média de Retransmissão

A consideração desta variável também se mostra interessante, pois sua análise permite estudar o seu grau de relação com a variável atraso de amostra. De forma simplificada, vale notar que quanto maior o número de retransmissões *TCP*, maior será o atraso de amostra.

Novamente, o analisador *Wireshark* mostrou-se extremamente útil para a coleta dos dados relevantes a este experimento.

Assim como para a variável tempo de resposta, o monitoramento da taxa de retransmissão foi também realizado de forma concorrente ao monitoramento da variável atraso de amostra. Sua análise foi feita obtendo-se o número total de segmentos *TCP* com carga útil (amostras) enviados pelo Sistema Cliente e o número de segmentos *TCP* com carga útil retransmitidos pelo Sistema Cliente, por terem sido previamente perdidos na rede de comunicação. A taxa média de retransmissão foi considerada como a divisão entre o segundo e o primeiro resultado. Uma observação importante é o fato dos segmentos sem carga útil não terem sido considerados no cálculo desta variável.

Em termos de espaço amostral, para cada experimento dos 126 realizados (6 tipos de BLOCO \* 3 vezes ao dia \* 7 dias) certo número de segmentos *TCP* com carga útil foram transmitidos a partir do cliente, onde cada segmento *TCP* transportava um BLOCO de amostras. Observando-se a Tabela 19, é fácil concluir que esse número é variável em relação ao tamanho do BLOCO utilizado, visto que o total de amostras transmitidas se mantém constante. Enquanto para o BLOCO de tamanho igual a 60 amostras o número de segmentos transmitidos por experimento é 60 ( $60 \times 60 = 3.600$  amostras por experimento), para o BLOCO de tamanho igual a 360 amostras o número de segmentos cai para 10 ( $360 \times 10 = 3.600$  amostras por experimento). Somando-se os BLOCOS transmitidos em um período de um dia obtém-se o número 147 ( $10 + 12 + 15 + 20 + 30 + 60$ ). Multiplicando-se esse valor pelos 3 períodos do dia e pelos 7 dias em que esses experimentos foram realizados, conclui-se que foram transmitidos ao todo 3.087 BLOCOS de amostras.

Uma investigação de como a taxa média de retransmissão comporta-se em diferentes períodos do dia e em diferentes dias da semana também foi realizada. Tais agrupamentos, apresentados na Figura 57 e na Figura 58, foram feitos exatamente da mesma forma aplicada no caso da variável tempo de resposta.

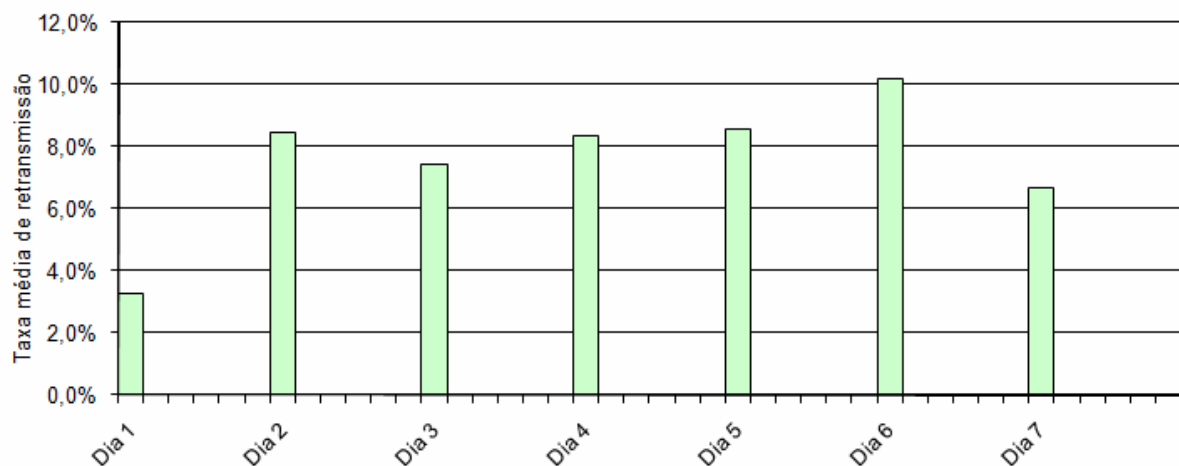


Figura 57: taxa média de retransmissão em função do dia da semana.

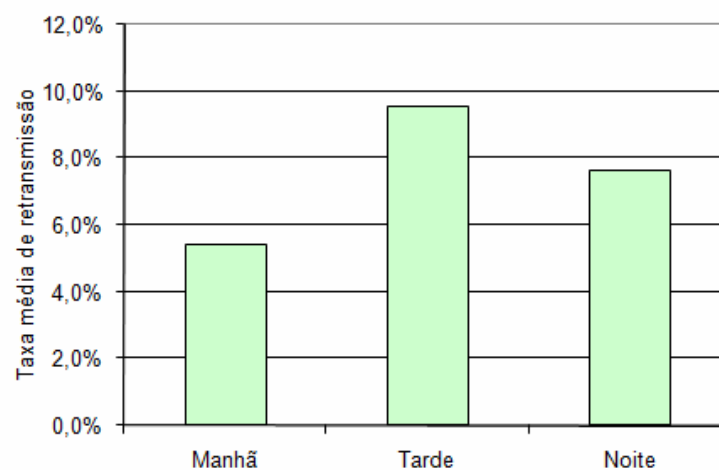


Figura 58: taxa média de retransmissão em função do período do dia.

Ao se analisar os dados da Figura 57 e da Figura 58 com a ferramenta *anova1*, verificou-se que o dia da semana não influenciou significativamente na taxa média de retransmissão, mas o período do dia realizou essa influência, pois os respectivos “Prob>F” foram de 0,1256 e de 0,0308. Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura 59 e na Figura 60, respectivamente.

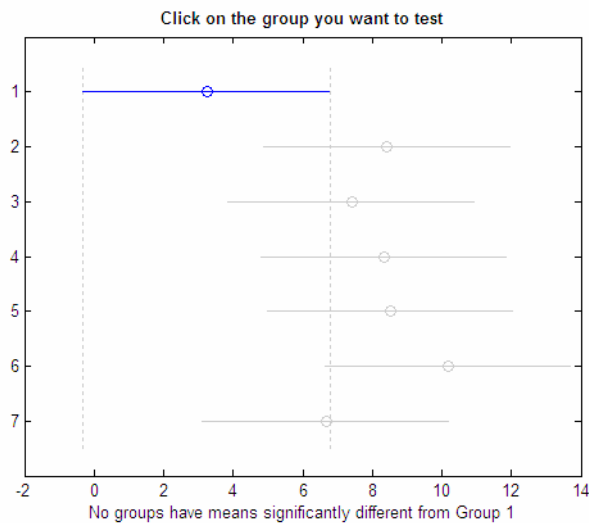


Figura 59: gráfico obtido com a ferramenta *multcompare* para a taxa média de retransmissão por dia da semana (grupo 1 selecionado aleatoriamente).

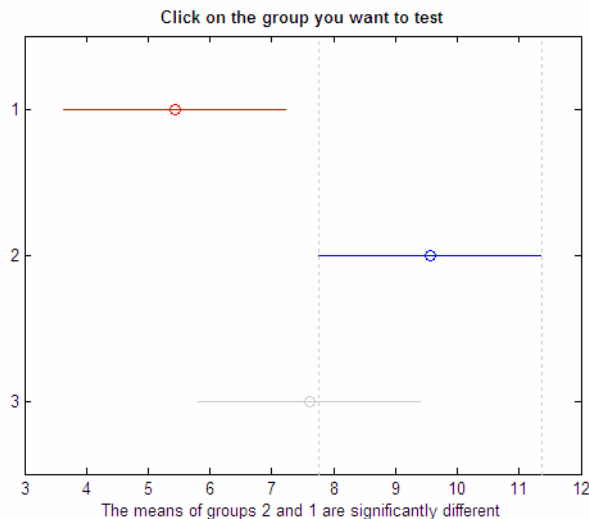


Figura 60: gráfico obtido com a ferramenta *multcompare* para a taxa média de retransmissão por período do dia (grupo 2 selecionado aleatoriamente).

Considerando-se todos os 3.087 segmentos *TCP* supracitados, indiferentemente do período ou dia em que foram transmitidos, a taxa média de retransmissão foi de 7,54%. O desvio-padrão e o coeficiente de variação associados a essa média foram de 7,22% e 95,71%.

### 5.2.3 Taxa Média de Erro

O interesse por esta variável também se deve ao fato dela se relacionar com a variável atraso de amostra. De forma geral, quanto maior a taxa de erro, maior será a taxa de retransmissão e, por consequência, maior será o atraso médio de amostra.

Seu monitoramento foi realizado seguindo-se a mesma metodologia aplicada ao monitoramento da taxa de retransmissão. Ao serem analisados no analisador *Wireshark* os

3.087 segmentos *TCP* diretamente responsáveis pela transmissão de amostras, foi facilmente verificado que nenhum deles foi corrompido durante a transmissão. Portanto, a taxa média de erro obtida, em nível de transporte, foi de 0%.

### 5.3 QUALIDADE DO DISPOSITIVO MÓVEL

Nesta seção, são descritos os experimentos realizados para se verificar particularmente a qualidade do dispositivo móvel desenvolvido. As principais variáveis estreitamente relacionadas ao dispositivo móvel foram consideradas.

#### 5.3.1 Taxa Média de Perda de Amostras

Como a frequência de monitoramento escolhida é relativamente alta (200 Hz) e o tamanho do *buffer* circular de PARTes implementado no dispositivo móvel é relativamente pequeno, há chances de ocorrerem perdas de amostras durante interrupções temporárias de fluxo de dados entre um Sistema Cliente e um Sistema Servidor.

Como já foi detalhado na Subseção 3.1.2, o tempo limite entre um bloqueio e um subsequente desbloqueio de fluxo de dados para que nenhuma amostra seja perdida é de 10 segundos. Essa limitação deve-se ao fato da memória *RAM* disponível na *MCU* utilizada ser relativamente pequena. Portanto, contornar essa limitação é relativamente simples, basta (1) considerar outra *MCU*, cuja memória *RAM* seja maior, ou (2) incorporar ao *hardware* do dispositivo móvel um módulo externo de memória.

Nos casos em que o fluxo se encontra bloqueado por mais de 10 segundos, as amostragens continuam sendo realizadas e as amostras mais antigas são sobrescritas (ou perdidas). Essa política foi escolhida de forma a privilegiar as informações mais recentes em caso de perda de amostras.

O monitoramento desta variável foi realizado de acordo com a metodologia descrita a seguir. Durante os experimentos realizados para o estudo da variável atraso de amostra, não se transmitiu resultados de conversões A/D, e sim valores de uma variável que ficou sendo alterada de forma circular no intervalo [0; 255]. Ou seja, a primeira amostra transmitida continha sempre os três canais iguais a 0, a segunda amostra continha os canais iguais a 1 e assim por diante. Dessa forma, é possível saber, ao certo, o valor de cada canal de cada

amostra transmitida. Com esse conhecimento, é possível detectar, durante a recepção de amostras, se alguma delas foi perdida.

A análise desta variável foi realizada contando-se o número total de amostras transmitidas (vide Tabela 19) e o número de amostras perdidas. Considerou-se como a taxa média de perda de amostras a divisão entre segundo e o primeiro resultado.

Para que a obtenção do número de amostras perdidas fosse confiável, foram utilizados dois mecanismos redundantes entre si. O primeiro foi a verificação dos valores dos canais das amostras recebidas e registradas em banco de dados. Cada descontinuidade dentro do intervalo [0; 255] representava perda de amostras. Adicionalmente, utilizou-se uma variável de depuração interna ao *firmware* que registrava o número de amostras sobrescritas durante eventuais interrupções de fluxo. Todos os resultados de número de amostras perdidas obtidos por ambos os mecanismos foram convergentes entre si.

Os mesmos agrupamentos já explicados, por dia da semana e por período do dia, foram aplicados também a esta variável. Seus resultados são apresentados na Figura 61 e na Figura 62.

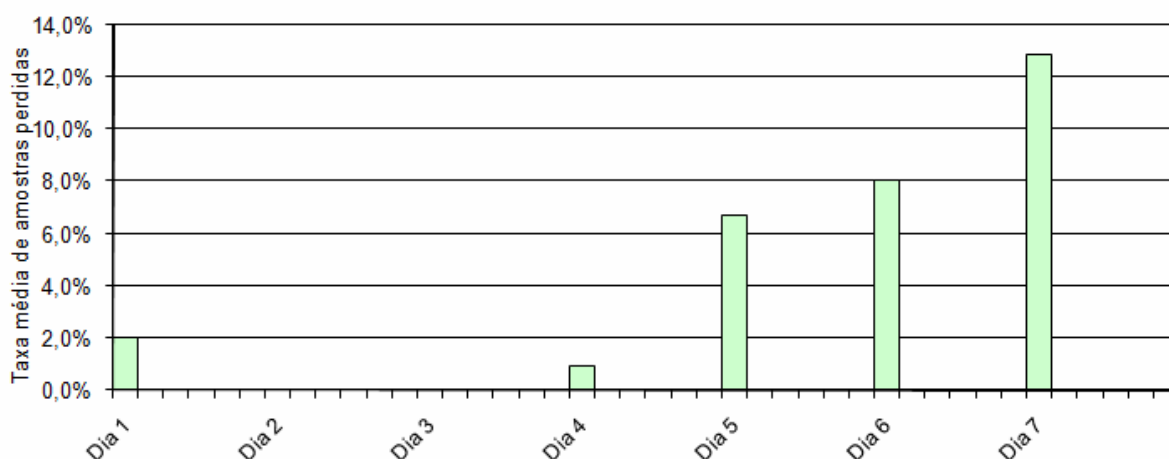


Figura 61: taxa média de perda de amostras em função do dia da semana.



Figura 62: taxa média de perda de amostras em função do período do dia.

Ao se analisar os dados da Figura 61 e da Figura 62 com a ferramenta *anova1*, verificou-se que o dia da semana e o período do dia não influenciaram significativamente na taxa média de perda de amostra, pois os respectivos “Prob>F” foram de 0,0683 e de 0,4327. Os resultados produzidos pela ferramenta *multcompare* são apresentados na Figura 57 e na Figura 58, respectivamente.

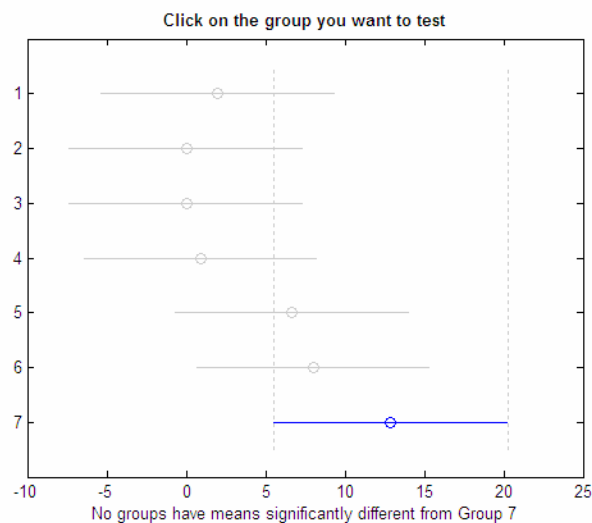


Figura 63: gráfico obtido com a ferramenta *multcompare* para a taxa média de perda de amostras por dia da semana (grupo 7 selecionado aleatoriamente).

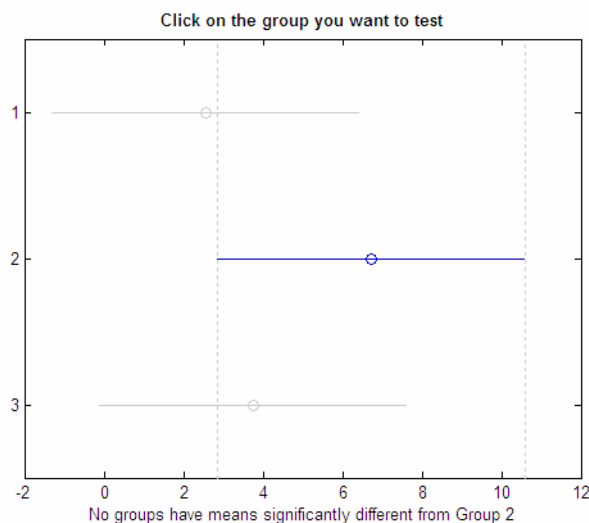


Figura 64: gráfico obtido com a ferramenta *multcompare* para a taxa média de perda de amostras por período do dia (grupo 2 selecionado aleatoriamente).

Considerando-se todas as amostras transmitidas, indiferentemente do dia ou período em que foram transmitidas, a taxa média de perda de amostras foi de 4,36%. O desvio-padrão e o coeficiente de variação associados a essa média foram de 15,11% e 346,91%.

### 5.3.2 Exatidão do Relógio

Como já explicado no Capítulo 4, o PMR foi desenvolvido de forma a transferir amostras de um determinado sinal analógico juntamente como uma informação (relógio) que permite identificar os momentos em que tais amostras foram obtidas. Nesse sentido, o dispositivo móvel deve ter a habilidade de manter um relógio, que é ajustado inicialmente na fase de estabelecimento de conexão PMR, em funcionamento.

Torna-se importante, então, medir a exatidão do relógio interno ao dispositivo móvel desenvolvido, pois por menor que seja o erro entre esse relógio e um “relógio ideal”, ele sempre existirá. Em casos críticos, pode ocorrer do relógio interno estar ou muito atrasado ou muito adiantado após certo período de funcionamento. Nesse caso, a confiabilidade dos tempos associados às amostras estaria comprometida.

A manutenção do relógio interno ao dispositivo móvel desenvolvido foi realizada por meio de uma interrupção de tempo dedicada exclusivamente a essa tarefa. Seu tratador atualiza quatro variáveis que expressam: décimos de segundo, segundos, minutos e horas. Ou seja, implementou-se um relógio interno com informações até a ordem de décimos de segundo, pois essa é a precisão máxima que o PMR permite.



A seguinte metodologia foi adotada neste experimento:

- utilizou-se como cronômetro o kit de desenvolvimento descrito na Subseção 5.1.1. Como seus 4 dígitos de sete segmentos não provêm uma boa exatidão relativa para curtos períodos de tempos (ordem de centésimo de segundos), realizou-se um experimento com tempo suficientemente alto, de forma que 1 centésimo de segundo passa ser desprezível;
- o procedimento que mantém o relógio interno ao dispositivo móvel foi ligeiramente alterado para sinalizar à *FPGA* de tal kit eventos de início e fim de contagem; e
- esses eventos foram programados para ocorrerem com um espaçamento de duas horas.

Assumindo-se o relógio do kit como ideal, calculou-se o erro do relógio do dispositivo móvel por meio do erro relativo entre o período de duas horas (relógio impreciso) e o período registrado pela *FPGA* (relógio ideal) entre o início e fim de contagem, que indiretamente foi apresentado pelos 4 dígitos do kit. Após a realização desse cálculo, concluiu-se que o erro do relógio foi de  $(-6,53 \pm 1,38) * 10^{-5}$ . Portanto, são necessários mais de 15.000 segundos (mais de 4 horas) de funcionamento ininterrupto para que o relógio interno ao dispositivo móvel se atrase em 1 segundo.

Neste ponto, é importante lembrar que a conexão PMR pode ser desfeita e, então, restabelecida para que um novo relógio possa ser obtido a partir do Sistema Servidor. Esse mecanismo se mostra interessante principalmente em aplicações que demandam monitoramentos prolongados.

### 5.3.3 Exatidão do Período de Amostragem

A medição desta exatidão mostra-se interessante, basicamente, pelas mesmas razões apresentadas na Subseção 5.3.2. Em casos críticos, um determinado sinal monitorado seria apresentado com distorções no lado do Sistema Servidor, pois este sempre leva em consideração a frequência de amostragem informada pelo Sistema Cliente e não a frequência de amostragem que carrega consigo certo erro. Nesse caso, a qualidade e a confiabilidade dos dados da curva estariam comprometidas.

A frequência de amostragem foi implementada por meio de uma interrupção de tempo exclusivamente dedicada a essa tarefa. Seu tratador sinaliza a necessidade de realização de uma nova amostragem a cada início de período de amostragem.

A metodologia adotada neste experimento segue os mesmos princípios da metodologia do experimento 5.3.2. O tratador aqui mencionado foi programado para sinalizar à *FPGA* eventos de início e fim de contagem. O tempo utilizado entre esses eventos foi de 1.000.000 períodos de amostragem, ou seja, 5.000 segundos.

Assumindo-se o relógio do kit como ideal, calculou-se o erro da frequência de amostragem por meio do erro relativo entre o período de 5.000 segundos (relógio impreciso) e o período registrado pela *FPGA* (relógio ideal) entre o início e fim de contagem, que indiretamente foi apresentado pelos 4 dígitos do kit. Após a realização desse cálculo, concluiu-se que o erro do período de amostragem foi de  $(1,18 \pm 0,20) * 10^{-3}$ .

## 5.4 CONCLUSÕES

Tendo em vista o número de variáveis monitoradas e a qualidade metodológica em que essa fase foi realizada, conclui-se que a solução proposta foi de fato experimentada.

As principais conclusões que se pode chegar ao analisar os resultados apresentados em cada subseção deste capítulo são:

- o tempo de conexão se mostrou relativamente comportado e independente do dia e do período do dia em que foi estabelecida. Sua média geral foi de 51,86 segundos;
- o atraso de amostra mostrou-se dependente do dia, do período do dia e também do tamanho de BLOCO utilizado. De forma geral, observa-se que esse atraso cresce com o aumento do tamanho do BLOCO. Ao considerar variação somente no parâmetro tamanho de BLOCO, as médias gerais obtidas estão dentro do intervalo [1,23; 5,57] segundos;
- o tempo médio de reposta também mostrou-se relativamente bem comportado e independente do dia e do período do dia em que foi transmitida. Sua média geral foi de 0,53 segundos;
- a taxa de retransmissão mostrou-se dependente do período do dia e independente do dia da semana. Uma relação interessante entre essa variável e o atraso de amostra, levando-se em consideração o período do dia, se apresenta forte. Isso porque é possível verificar que os comportamentos dessas duas variáveis mostram-se bastante

parecidos, de forma que fica evidente a influência da taxa média de retransmissão no atraso médio de amostra. Sua média geral foi de 7,54%;

- a taxa de erro observada foi nula, graças ao mecanismo de retransmissão do *TCP*;
- a taxa de perda de amostras revelou-se não-comportada, em função tanto do dia da semana quanto do período do dia. Porém ela se mostrou independente do dia e do período do dia. É importante ressaltar que a causa desse comportamento indesejado pode ser extinta de forma relativamente fácil: basta aumentar a capacidade de memória *RAM* da *MCU*; e
- as precisões de relógio e de período de amostragem se mostram satisfatórias. Essas precisões foram da ordem de  $10^{-5}$  e  $10^{-3}$ , respectivamente.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Ao final deste trabalho, o seu objetivo geral foi alcançado com sucesso. Ou seja, foi desenvolvida e implementada uma solução de monitoramento baseada na tecnologia *GPRS* e apropriada para os cenários em que o monitoramento remoto, contínuo e *online* mostra-se necessário e/ou vantajoso.

Em termos de *hardware* do dispositivo móvel, após uma vasta pesquisa de mercado, um equipamento, denominado *Data Modem G24*, composto por uma unidade de microcontrolador e um Modem *GPRS* foi encontrado e utilizado na solução de monitoramento desenvolvida. Sua aquisição apresentou um custo-benefício muito bom, já que a energia poupada ao não implementar esse tipo de equipamento pôde ser canalizada para outras atividades fundamentais relacionadas ao trabalho.

Já em termos de *software*, um *firmware* executado por um dispositivo móvel e um sistema executado por um Servidor foram projetados e implementados de forma que todos os seus requisitos foram cumpridos. Durante o projeto desses sistemas, a utilização de ferramentas consolidadas de Engenharia de *Software* se mostrou bastante benéfica e apropriada.

Durante a fase de desenho da solução proposta, observou-se a inexistência de um protocolo de comunicação aberto, padrão e adequado ao tipo de transmissão de dados que esse tipo de sistema realiza. A solução encontrada para essa dificuldade foi desenvolver um protocolo de camada de aplicação adequado à transmissão de informações de sinais analógicos, que é o caso do sinal de ECG. Esse protocolo, nomeado PMR, foi implementado e se mostrou flexível e eficiente, inclusive na transmissão de sinais analógicos não relacionados à Telemedicina.

Com a solução proposta já implementada, foi realizada uma fase de avaliação de desempenho das principais variáveis capazes de refletir sua qualidade. Os resultados obtidos mostram que a solução apresentada é viável e sua implantação pode trazer reais benefícios a pacientes e a instituições de saúde.

## 6.1 TRABALHOS FUTUROS

Diversas são as possibilidades de trabalho a partir do legado gerado por esta dissertação. Muitas dessas possibilidades lançam novos desafios interessantes de investigação científica, outras se caracterizam pelo forte cunho de implementação.

Como sugestão, pode-se citar os seguintes trabalhos:

- relacionado ao *hardware* do dispositivo móvel:
  - reprojeter a placa principal de forma a: adicionar botões de pânico e liga/desliga; englobar o módulo Pré-amplificador; e integrar um módulo de memória *RAM* externa à *MCU*.
- relacionados ao *firmware* do dispositivo móvel:
  - analisar agregação de um módulo de detecção de anomalia cardíaca, por *software*, de forma a possibilitar monitoramento orientado a evento; e
  - reprojeter o código-fonte de forma a usufruir dos recursos disponíveis para minimização de consumo energético, tanto da *MCU* quanto do Modem *GPRS*. Uma avaliação do ganho também poderia ser feita.
- relacionado ao Sistema Servidor:
  - finalizar a implementação do PMR de forma a contemplar todo o seu potencial.
- relacionados ao PMR:
  - avaliar a utilização do protocolo de comunicação *SSL* (*Secure Sockets Layer*), sobre o *TCP*, de forma a estabelecer conexões seguras entre Sistemas Clientes e Sistemas Servidores. Resumidamente, essa tecnologia provê: compactação, criptografia, integridade de dados, autenticação e comunicação secreta [26];
  - avaliar a utilização de alternativas ao *TCP*, por exemplo: *RTP* (*Real-Time Transport Protocol*) e *RTCP* (*RTP Control Protocol*); e
  - estudar os benefícios de inserção de mecanismo de ressincronismo de relógio de Sistemas Clientes a partir do relógio de Sistemas Servidores, durante uma conexão.
- relacionados ao sistema de monitoramento proposto como um todo:
  - avaliar os benefícios ao se utilizar outras tecnologias para comunicação alternativa de monitoramento, tais como *e-mail* e *SMS* (*Short Message Service*); e

- modelar e implementar esquema de banco de dados e interface *Web* apropriados à solução.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Istepanian, R.S.H. Woodward, B. Richards, C.I. (2001). Advances in Telemedicine Using Mobile Communications, *Proceeding of the 23rd Annual EMBS International Conference*.
- [2] Martincoski, D.H. (2003). Sistema para Telemetria de Eletrocardiograma Utilizando Tecnologia Bluetooth, *Dissertação (Mestrado em Eng. Elétrica), Universidade Federal de Santa Catarina*.
- [3] Anliker, U. Ward, et al. (2004). AMON: A Wearable Multiparameter Medical Monitoring and Alert System, *IEEE Transactions on Information Technology in Biomedicine, Vol. 8, No. 4*.
- [4] Fensli, R. et. al (2005). A Wearable ECG-recording System for Continuous Arrhythmia Monitoring in a Wireless Tele-Home-Care Situation, *Proceedings of the 18th IEEE Symposium on Computer-Based Medical System, 1063-7125/05*.
- [5] Bonho, S. et al. (2006). Monitoração de Sinais de ECG pela Internet em Tempo Real, *X Congresso Brasileiro de Informática em Saúde*.
- [6] Fadlee, M. Rasid, A. Woodward B. (2005). Bluetooth Telemedicine Processor for Multichannel Biomedical Signal Transmission via Mobile Cellular Networks, *IEEE Transactions on Information Technology in Biomedicine, Vol. 9, No. 1*.
- [7] Fensli R. Gunnarson, E. Hejlesen, O. (2004). A Wireless ECG System for Continuous Event Recording and Communication to a Clinical Alarm Station, *Proceedings of the 26th Annual International Conference of the IEEE EMBS*.
- [8] Led, S. Fernández, J. Serrano, L. (2004). Design of a Wearable Device for ECG Continuous Monitoring Using Wireless Technology, *Proceedings of the 26th Annual International Conference of the IEEE EMBS*.
- [9] Araujo, M.C. Contreras, R.C. Rodrigues, M.A.B. (2006). Holter de ECG via Tecnologia GSM, *XX CBEB 2006, 1299-1302*.
- [10] Guimarães, J. I. et al. (2003). Normatização dos Equipamentos e Técnicas para a Realização de Exames de Eletrocardiografia e Eletrocardiografia de Alta Resolução. *Arquivos*

*Brasileiros de Cardiologia*. São Paulo, SP, v. 80, n. 5 (Mai.), p. 572-583. Disponível em: <http://www.scielo.br/>. Acesso em: 02 jun. 2008.

[11] Tompkins, W. J. Biomedical Digital Signal Processing apud Martincoski, D.H. (2003). Sistema para Telemetria de Eletrocardiograma Utilizando Tecnologia Bluetooth, *Dissertação (Mestrado em Eng. Elétrica)*, Universidade Federal de Santa Catarina.

[12] Disponível em: <http://www.historiadelamedicina.org/einthoven.html>. Acesso em: 01 jun 2008.

[13] Berkow, R. Manual Merck de Informação Médica – Saúde para a Família. Disponível em: <http://www.msd-brazil.com>. Acesso em 02 jun. 2008.

[14] Paim, F. C. A. (2005). Desenvolvimento de Um Sistema de Telemetria para Aquisição de Sinais Fisiológicos com Aplicação em Programas de Reabilitação Cardíaca, *Dissertação (Mestrado em Eng. Elétrica)*, Universidade Federal de Santa Catarina.

[15] Motorola, Motorola G24 Developer's Guide – Module Hardware Description, Datasheet (2007). Disponível em: [http://developer.motorola.com/docstools/developerguides/G24\\_HardwareDescription\\_Developer\\_Guide.pdf](http://developer.motorola.com/docstools/developerguides/G24_HardwareDescription_Developer_Guide.pdf). Acesso em: 10 abri. 2008.

[16] Andreão, R.V. et al. (2006). TeleCardio: Telecardiologia a Serviço de Pacientes Hospitalizados em Domicílio, *X Congresso Brasileiro de Informática em Saúde*.

[17] Disponível em: <http://www.informatetechnology.com.br>. Acesso em: 01 fev 2008.

[18] Disponível em: <http://www.freescale.com/>. Acesso em 01 fev 2008.

[19] Freescale Semiconductor, HCS08 Microcontrollers, Datasheet (2004). Disponível em: [www.freescale.com/files/microcontrollers/doc/data\\_sheet/MC9S08GB60.pdf](http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08GB60.pdf). Acesso em: 10 abri. 2008.

[20] Feghali, I. Andreão R.V. Segatto, M.V. (2006). Abordagem na Web para o Telemonitoramento do Eletrocardiograma de Pacientes Domésticos, *VI Workshop de Informática Médica*.

[21] Disponível em: <http://www.php.net/>. Acesso em: 01 fev 2008.



- [22] Onken, M. Riesmeier, J. Eichelberg, M. (2006). Standardized exchange of medical signals using DICOM waveforms, *Proceedings EuroPACS*.
- [23] Health Level 7: “Application Protocol for Electronic Data Exchange in Healthcare Environments“, Version 2.5, Standard, (2003). Disponível em: <http://www.hl7.org>. Acesso em: 01 mar 2008.
- [24] *Institute of Electrical and Electronics Engineers (IEEE)*: IEEE 1073 Point of Care Medical Device Communication Standards, Family of standards, (2006). Disponível em: <http://www.ieee1073.org>. Acesso em: 01 mar 2008.
- [25] *European Committee for Standardization (TC 251)*: European Standardization of Health Informatics.
- [26] Tanenbaum, A. S. Redes de Computadores (2003). Tradução da Quarta Edição, Editora Campus.
- [27] Disponível em: <http://www.wireshark.org/>. Acesso em 01 fev 2008.
- [28] Moraes, A. F. Redes de Computadores: fundamentos, 5ª Edição, Editora Érica.
- [29] McMenamim, S. Palmer, J. Análise Essencial de Sistemas (1991). Ed. McGraw- Hill.
- [30] Filho, W. P. P. Engenharia de Software: Fundamentos, métodos e padrões (2005). 2ª Edição, Editora LTC.
- [31] Kurose, J.F. Ross, K.W. Redes de computadores e a Internet, Uma abordagem top-down, 3ª Edição, Editora Pearson.
- [32] Malkin, G. S. RFC 1983. Disponível em: <http://rfc.net/rfc1983.html>. Acesso em: 20 mar. 2008.