

**MIDDLEWARE PARA MONITORAMENTO E ENVIO DE ALERTAS
APLICADO AO AMBIENTE HOSPITALAR**

Bruno Gomes de Araújo

Doutorando do Departamento de Engenharia de Computação e Automação – Universidade Federal do Rio Grande do Norte (UFRN). E-Mail: brunogomes@dca.ufrn.br.

Ricardo Alessandro de Medeiros Valentim

Professor do Departamento de Engenharia Biomédica – Universidade Federal do Rio Grande do Norte (UFRN). E-Mail: ricardo.valentim@ufrnet.br.

Ana Maria Guimarães Guerreiro

Professora do Departamento de Engenharia Biomédica – Universidade Federal do Rio Grande do Norte (UFRN). E-Mail: anamaria@dca.ufrn.br.

Cicília Raquel Maia Leite

Doutoranda do Programa de Pós-graduação em Engenharia Elétrica e de Computação da Universidade Federal do Rio Grande do Norte (UFRN). E-Mail: cicilia.leite@uern.br

João Marcos Teixeira Lacerda

Mestrando do Departamento de Engenharia de Computação e Automação – Universidade Federal do Rio Grande do Norte (UFRN). E-Mail: jonymac@deb.ufrn.br.

Denner Oliveira de Araújo Cardoso Navarro

Aluno do curso de Ciências e Tecnologia da Universidade Federal do Rio Grande do Norte (UFRN). E-Mail: dennernavarro@hotmail.com.

RESUMO

Devido a grande necessidade de gerenciamento, controle e monitoramento das informações a área da automação hospitalar tem sido alvo de muitas pesquisas através das constantes evoluções tecnológicas. No entanto, muitos processos hospitalares encontram-se de forma manual tanto no setor público como privado. Desta forma, o objetivo deste trabalho é o desenvolvimento de uma solução para o processo de monitoramento dos dados provindos de pacientes internados em uma Unidade de Terapia Intensiva (UTI). Assim, será definida uma arquitetura de um middleware para computação móvel aplicado ao ambiente hospitalar, realizando atividades de monitoramento e envio de alertas para dispositivos móveis da equipe médica.

Palavras-chave: Middleware, Automação Hospitalar, Monitoramento, Computação Móvel, Sinais Vitais.

MIDDLEWARE FOR MONITORING AND SENDING ALERTS APPLIED TO THE HOSPITAL ENVIRONMENT

ABSTRACT

Due to the need for management, control and monitoring of information, the hospital automation has been the subject of extensive research by means of the constant technological developments. However, many hospital processes are done manually in both the public and private sectors. Thus, the objective of this study is to develop a solution to the data monitoring process coming from patients hospitalized in an Intensive Care Unit (ICU). It will set an architecture of a middleware for mobile computing applied to the hospital, performing activities of monitoring and sending alerts to mobile devices of the medical team.

Keywords: Hospital Automation, Monitoring, Mobile Computing, Vital Signs.

MIDDLEWARE PARA MONITORAMENTO E ENVIO DE ALERTAS APLICADO AO AMBIENTE HOSPITALAR

INTRODUÇÃO

Em decorrência da globalização e da própria exigência do mercado de trabalho, as empresas e desenvolvedores estão buscando soluções para a execução de suas tarefas em um menor tempo possível, e, aliado a isto, ter um menor gasto na aquisição de equipamentos e desenvolvimento de sistemas.

O desenvolvimento de sistemas está cada vez mais complexo, devido ao grande volume de informações envolvidas e operações a serem realizadas. Com isso, há uma necessidade de maior controle e gerenciamento sobre esses dados, tornando-se uma tarefa crítica, necessitando sempre de um acompanhamento por parte de um especialista de onde quer que ele esteja.

O processamento de fluxo contínuo de dados está surgindo como uma área de pesquisa em expansão e está voltada para o processamento de informações produzidas por dispositivos que geram grandes volumes de dados em alta velocidade e com tempo de vida útil limitado. Por exemplo, as informações geradas por sensores que coletam os sinais vitais de pacientes, são sequências contínuas e ilimitadas de dados.

Tradicionalmente, tais informações requerem dispositivos/equipamentos e software/aplicações especiais para monitorá-las, que processam e reagem à entrada contínua de diversas origens.

Nesse sentido, várias pesquisas têm sido desenvolvidas e geralmente tem abordado vários dos problemas que são pertinentes aos processos que podem ser automatizados no ambiente hospitalar (2). Desta forma, requisitos emergentes surgem na área médica como forma de automatizar os processos encontrados no ambiente hospitalar, como por exemplo, o desenvolvimento de sistemas de monitoramento de pacientes (7), (8) e (9).

A automação desses processos é facilitada ainda com um novo paradigma que está surgindo e possibilitando o acesso à informação em qualquer lugar e a qualquer momento,

a Computação Móvel (5). Várias são as possibilidades de desenvolvimento de aplicações para os dispositivos que possibilitam mobilidade na execução das tarefas (3).

Entretanto, as diversidades na conectividade de rede, capacidade de plataforma e disponibilidade dos recursos podem afetar significativamente ao desempenho dessas aplicações, além de que estas aplicações não estão preparadas para oferecer apoio adequado para abordar os aspectos dinâmicos dos sistemas móveis. Modernas aplicações distribuídas necessitam de um *middleware* que seja capaz de se adaptar as mudanças do ambiente e que sejam compatíveis com o nível exigido de qualidade do serviço.

Visando isto, o presente trabalho tem como objetivo demonstrar um modelo de *middleware* para a computação móvel, utilizado como estudo de caso no ambiente hospitalar para gerenciamento de informações críticas e envio de mensagens.

MIDDLEWARE PARA MONITORAMENTO E ENVIO DE ALERTAS

Os pacientes internados em Unidade de Terapia Intensiva (UTI) necessitam de acompanhamento contínuo dos seus sinais vitais, visando a detecção antecipada de situações de risco, permitindo a intervenção em tempo hábil pelos profissionais de saúde. Assim, ressalta-se a importância da equipe médica receber informações sobre os sinais vitais de um paciente e um pré-diagnóstico sempre que for detectada uma possível situação de risco.

Geralmente esses processos ainda são utilizados de forma manual nos hospitais, os quais acabam tornando ineficiente o monitoramento de pacientes, pois podem gerar atrasos, prejudicando o atendimento de pacientes em situação crítica.

O mecanismo de detecção da anomalia e comunicação com a equipe médica, por exemplo, são precários, uma vez que funcionam de forma manual, e também por utilizarem métodos não eficientes para resolução do caso, como o alerta sonoro, que muitas vezes não é escutado pela equipe, e a comunicação com um profissional responsável, que algumas vezes resulta num processo demorado por não conseguirem localizá-lo, e também devido a própria equipe localizada no ambiente de monitoramento.

Devido à necessidade de um sistema robusto capaz de realizar um controle eficaz do grande volume de informações envolvidas no monitoramento e comunicação, foi criado um *middleware*.

O *middleware* pode ser definido como uma camada intermediária entre o sistema operacional e as aplicações distribuídas possibilitando a comunicação através de um conjunto de interfaces de programação e protocolos padronizados, conforme ilustrado na Figura 1.

O conceito de *middleware* foi criado em meio a ferramentas que buscavam soluções para implementação de uma interface uniforme de acesso aos recursos do sistema, e que contemplassem uma perspectiva multiplataforma (10).

O *middleware* desenvolvido tem funcionalidades relacionadas com o monitoramento de informações, através de consultas em tempo-real a banco de dados, e também capaz de enviar alertas em diversas situações críticas definidas pelo usuário.

Ele irá atuar no processamento das informações obtidas a partir de dispositivos ligados ao paciente. No caso de haver uma alteração na análise dos dados, automaticamente alertas serão preparados e enviados para os dispositivos cadastrados do usuário responsável que estiver no local naquele momento.

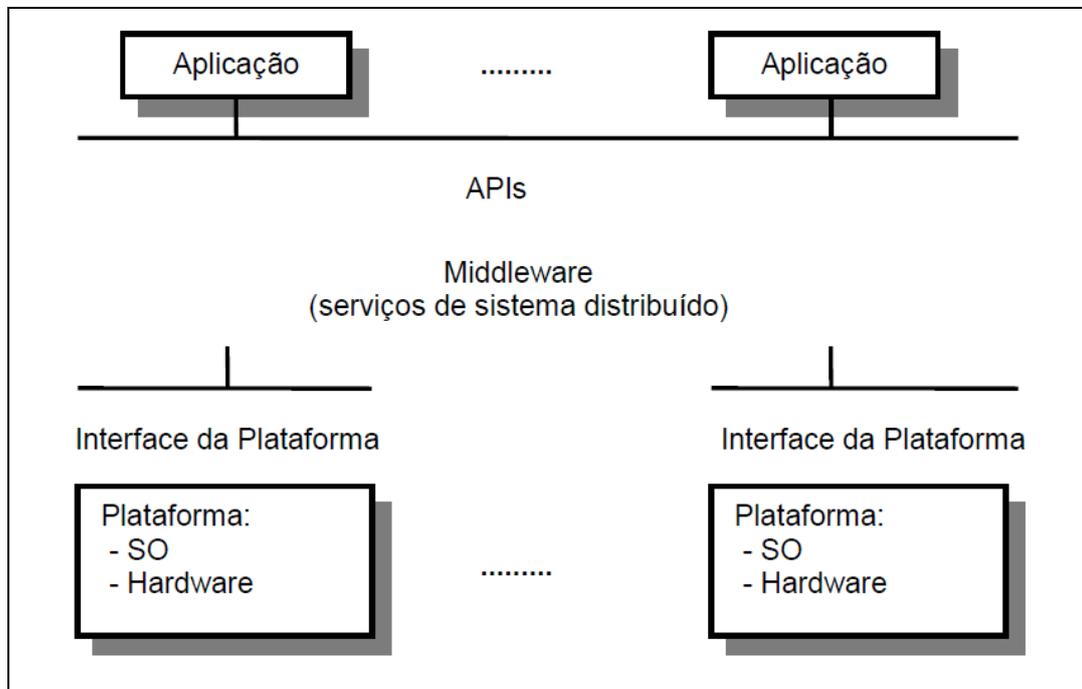


Figura 1 - Representação do Middleware. Fonte: Bernstein (1996)

SERVIÇOS DO MIDDLEWARE

Diante do contexto, os serviços oferecidos pelo *middleware* são: Acesso à Banco de Dados em Tempo Real; Configuração Automática de Banco de Dados; Enviar E-mail; Enviar SMS; Enviar mensagem via *socket*.

Os serviços relacionados ao monitoramento dos dados são apresentados pela Figura 2.

Devido à necessidade do monitoramento das informações, é necessário um mecanismo eficiente de acesso aos dados. A melhor forma de realizar essa operação é o acesso em tempo-real, possibilitando uma maior segurança e robustez a leitura das informações.

O sistema ainda oferece acessos aos mais diversos tipos de banco de dados existentes, oferecendo um serviço de configuração automática do banco de dados, onde o usuário deverá somente passar as informações do banco (servidor, banco, porta, *driver* e modelo da url), e ele se conectará automaticamente, conforme Figura 3.

Caso haja a necessidade de comunicação devido a alguma detecção de estado crítico das informações, existem os serviços assíncronos de envio de alertas, oferecidos pelo *Middleware*, como mostra a Figura 4.

O envio assíncrono consiste em um mecanismo no qual o emissor e receptor são entidades independentes, não sincronizados. Funciona através do envio de *Short Message Service* (SMS), nome dado ao serviço de mensagens curtas, de até 140 caracteres, para telefones

celulares, e e-mail, respectivamente, de um usuário cadastrado. Este mecanismo foi criado para situações em que a equipe médica, mais especificamente o médico plantonista, estiver fora da área de cobertura da rede sem fio do hospital ou mesmo distante do computador, para que ele possa ser alertado através da rede GRPS (*General Packet Radio Service*) do seu dispositivo móvel, através do recebimento de SMS.

A rede GPRS corresponde a um sistema de comunicação que permite o envio e recepção de informações através de uma rede telefônica móvel, a *Global System for Mobile Communications* (GSM), que oferece como vantagens a disponibilidade imediata e uma ampla área de cobertura (11), (12), (13).

Para que seja possível o envio através de SMS e e-mail, é necessária a configuração prévia no sistema. Para o SMS, é necessário realizar a configuração do modem GPRS que irá enviar as mensagens. Para isso, devem ser cadastradas informações como porta, velocidade, marca e modelo do modem, conforme ilustrado na Figura 5.

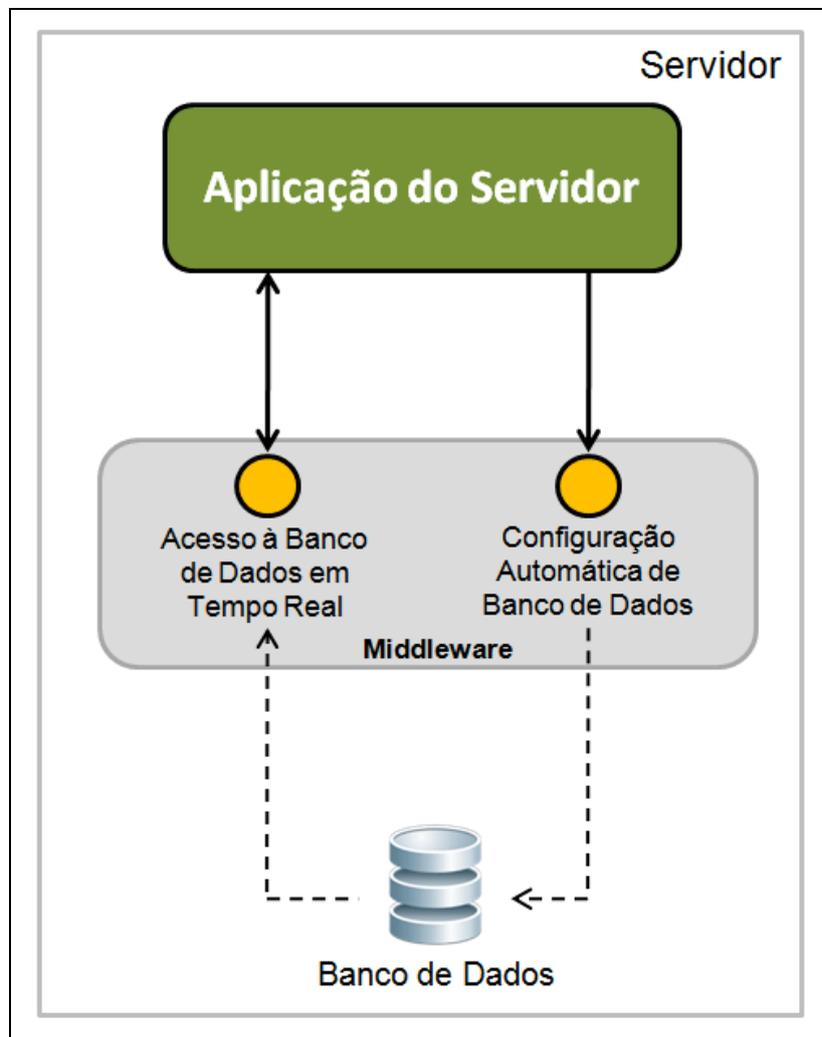


Figura 2 - Serviços Relacionados com o Monitoramento

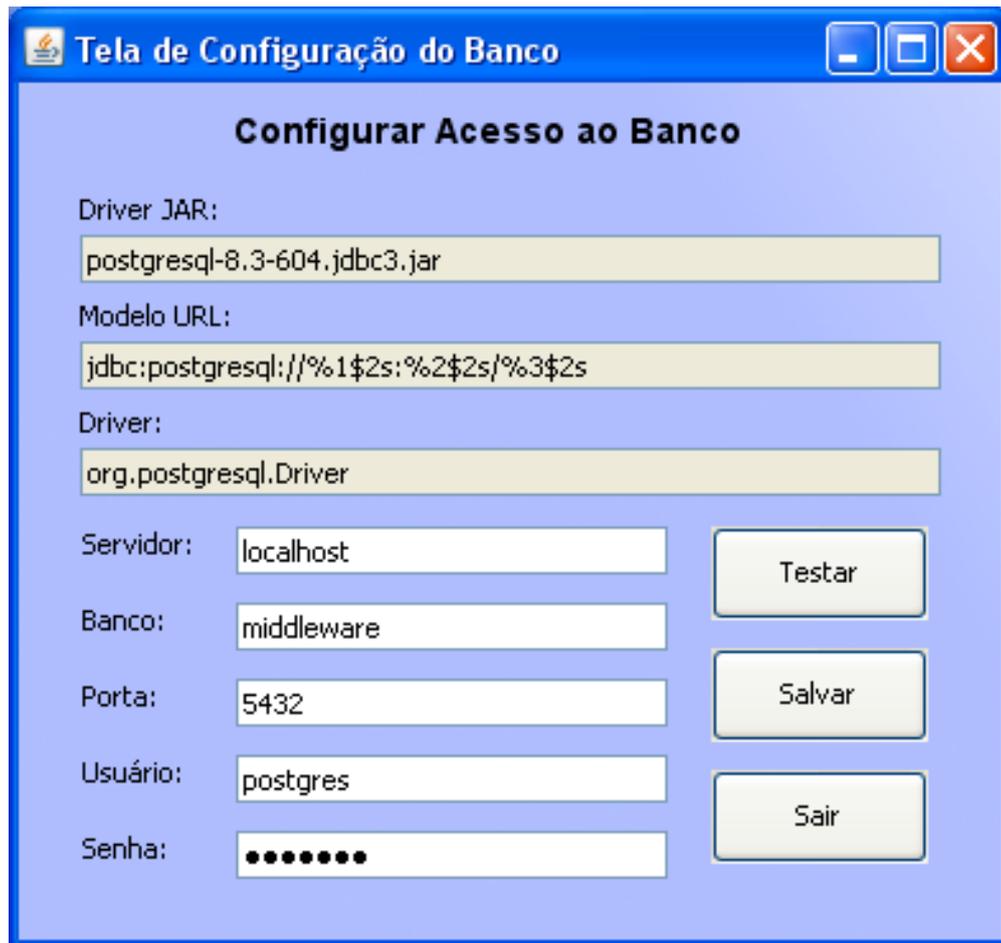


Figura 3 - Configuração Automática do Banco de Dados

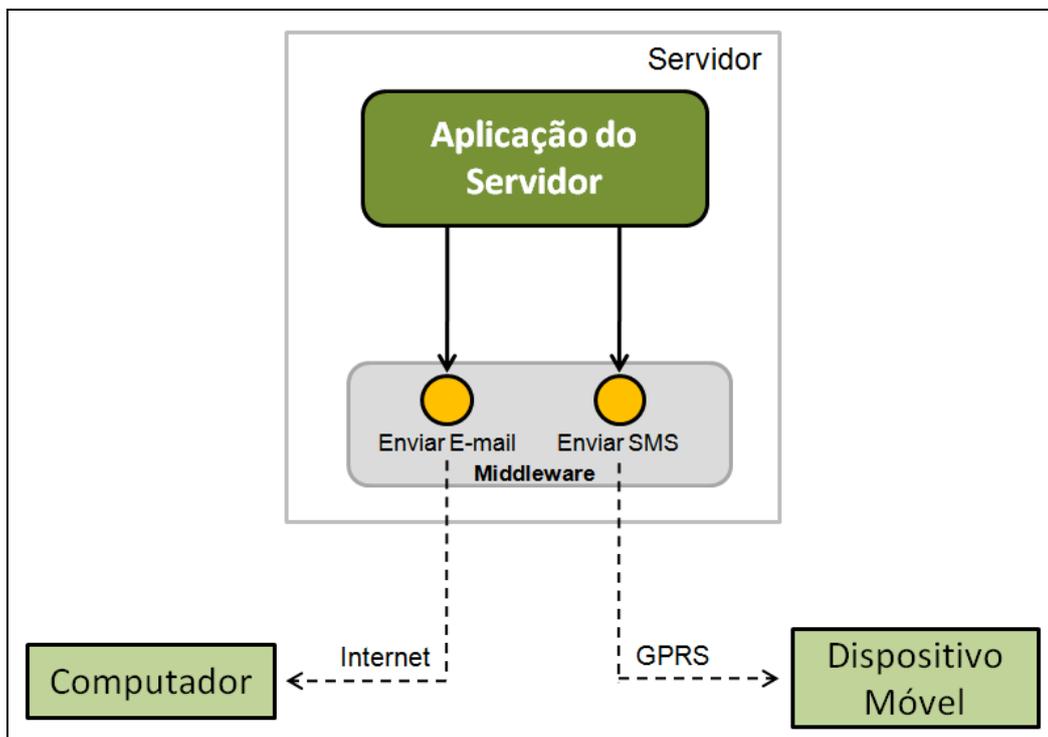


Figura 4 - Serviços relacionados com o Envio de Alertas



Figura 5 - Configuração do Modem

Para o envio através de e-mail, também é necessária a configuração do servidor de envio do e-mail (SMTP, e-mail, usuário e senha), ilustrado na Figura 6.



Figura 6 - Configuração de E-mail

O middleware também oferece o mecanismo de envio síncrono, no qual o remetente e o destinatário estão sincronizados, sendo esta a forma de envio mais segura, pois a

comunicação acontece de forma bidirecional, e há um controle sobre todas as ações tomadas durante a comunicação e sobre as mensagens enviadas.

Para implementação desse mecanismo utilizou-se Sockets, que segundo (11), consiste numa interface local, controlada pelo sistema operacional, localizada entre a camada de aplicação e a de trans-porte do modelo de comunicação TCP/IP, que permite processos distribuídos se comunicarem atra-vés da troca de mensagens, utilizando assim, a forma de comunicação cliente-servidor.

Para utilizar Sockets, é necessário escolher o protocolo de transporte que irá realizar a comunicação entre os processos. Este pode ser o *Transmission Control Protocol* (TCP) ou *User Datagram Protocol* (UDP).

O UDP consiste em um protocolo não confiável, pois ele não garante a entrega dos dados enviados pela rede. Já o TCP garante que os dados trafegados pela rede chegaram ao seu destino, através de mecanismos de retransmissão de mensagens e temporiza-dor. Devido a isto, o TCP foi escolhido para realizar a troca de informações e envio de alertas entre a central e o dispositivo móvel do plantonista. A Figura 7 apresenta como o dispositivo móvel se conecta com a central através dos *Sockets*.

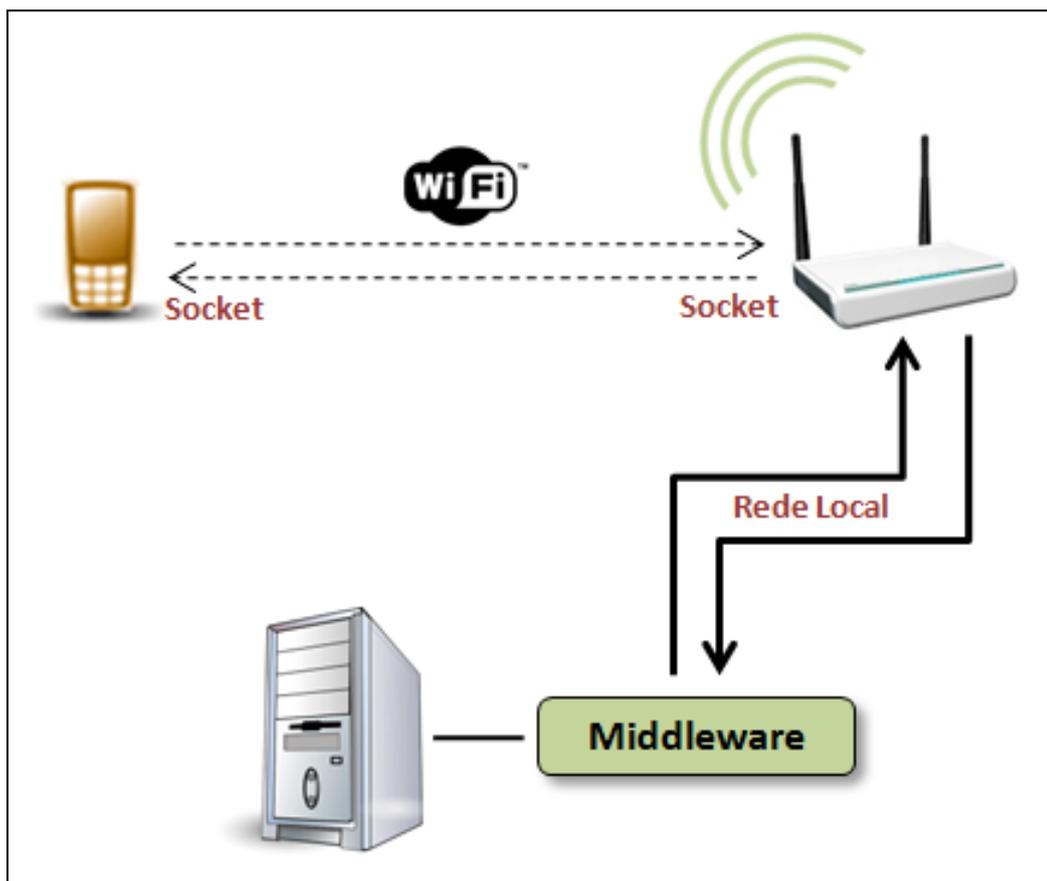


Figura 7 - Comunicação através dos *Sockets*

ESPECIFICAÇÃO DO MIDDLEWARE

O *middleware* foi desenvolvido na linguagem Java utilizando a biblioteca padrão JDK 1.6.0-14, no ambiente de desenvolvimento Eclipse 3.4 (4). De forma complementar, foram

usadas algumas API's e frameworks. O acesso à banco de dados em tempo-real foi desenvolvido utilizando características temporais da linguagem de consulta para tempo-real (LC-BDTR), especificado em (6). A configuração automática do banco de dados se deu utilizando a tecnologia JPA (Java *Persistence* API), e o envio dos alertas através do *Framework* SMSLib, em que é necessário o computador estar conectado à um modem GSM para o envio da SMS. O envio do e-mail basta que o computador esteja ligado a internet.

FUNCIONAMENTO DO MIDDLEWARE

O primeiro passo para o funcionamento se refere ao cadastramento das informações, no qual será realizado o cadastro do médico e todas as possíveis formas de envio de alertas para o mesmo, como e-mail, número de celular, entre outros, que são para testar os serviços do *middleware* através da aplicação.

Em paralelo, acontece a aquisição de dados vindos de pacientes internados em leitos, os quais são processados e armazenados em um banco de dados. Após o cadastramento, qualquer alteração da normalidade dos dados armazenados sobre o paciente, será enviado alertas para o médico plantonista através de e-mail, SMS, ou mesmo via sockets. Uma consulta periódica é configurada no momento em que o paciente entra na UTI, conforme ilustrado na Figura 8.

The screenshot shows a Windows-style application window titled "Tela Principal". Inside, there is a section titled "Consulta em Tempo Real".

- Plantonista:** A text input field containing "Bruno Gomes".
- Consulta SQL:** A text area containing the following SQL query:

```
select numero from batimento
inner join paciente on(batimento.paciente = pac:
where paciente.nome='Joao Paulo' and
```
- Iniciar:** Two input fields for date and time. The date is "13/07/2009" and the time is "09:50:22".
- Intervalo entre consultas:** An input field containing "00:00:10".
- Mensagem de Alerta:** A text area containing "Paciente Joao com batimentos fora d".
- Buttons:** Two buttons labeled "Iniciar" and "Sair".

Figura 8 - Consulta em Tempo Real

A tela de monitoramento pode ser melhor visualizada na Figura 9, na qual é apresentado os estados do paciente. Ela é atualizada em tempo-real.

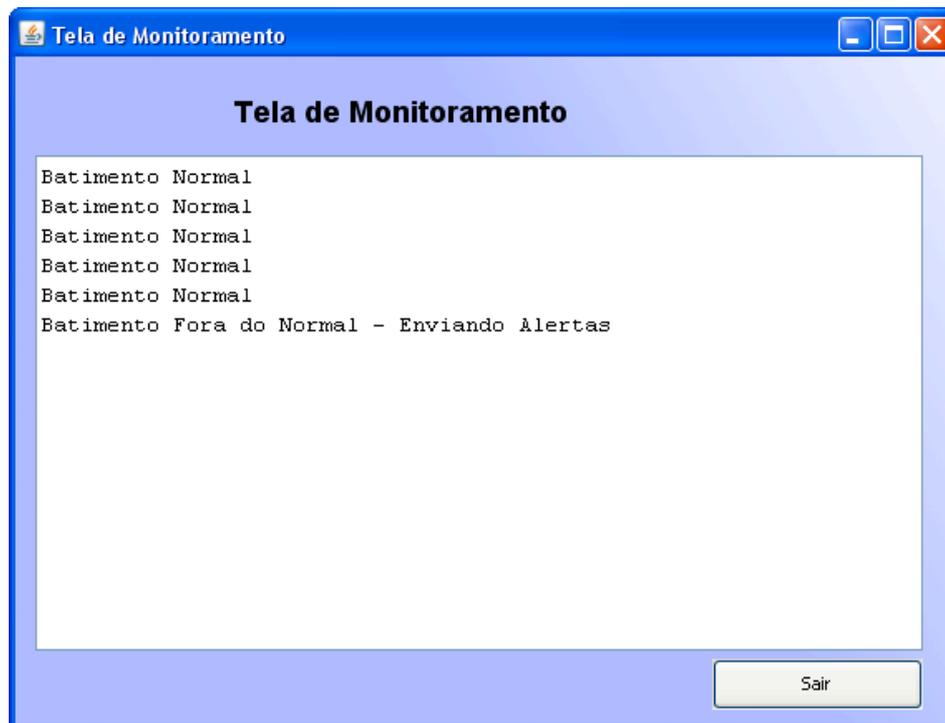


Figura 9. Monitoramento em Tempo-Real

Diante do exposto, a idéia principal do *middleware* foi consolidada de forma satisfatória pelo fato dos serviços propostos terem sido validados com sucesso através do estudo de caso proposto. Destaca-se ainda, que esse *middleware* pode ser utilizado em diferentes cenários, desde monitoramento doméstico ao monitoramento de aplicações mais críticas, apresentando um grande potencial de aplicação.

TESTES E RESULTADOS EXPERIMENTAIS

Com o objetivo de medir o tempo total de transmissão ocorrido durante o envio de mensagens do sistema, foi utilizado o RTT (*Round Trip Time*), que corresponde ao tempo gasto de ida e volta de uma mensagem trafegada pela rede (11). Sendo assim, foi aplicada a Equação (1) para realizar as medições pretendidas.

$$T_G = \frac{RTT}{2} \quad \text{equação (1)}$$

Onde, T_g corresponde ao tempo gasto para que uma mensagem vá da origem ao destino. No teste, foram feitos 1000 envios e recebimentos de mensagens de alertas, cujo objetivo era obter a média do tempo gasto.

O primeiro teste foi realizado em uma rede local, utilizando um roteador de 54 Mbps, com o servidor ligado ao roteador através da rede cabeada. O segundo teste foi realizado

utilizando uma rede de 150 Mbps com diversos computadores ligados e gerando tráfego. Nesta rede foi conectado também o servidor através da rede sem fio. Nos dois testes o dispositivo móvel se conecta através da rede sem fio. Os resultados dos testes são apresentados na Tabela 1.

Tabela 1 – Desempenho dos testes realizados

Redes	Tempo Médio de Envio (Milissegundos)
1	8,194
2	8,460

De acordo com os resultados, o tempo de transmissão das mensagens de alerta se deu na ordem de milissegundos, utilizando duas redes com capacidades e tráfegos diferentes. Devido a isso, a aplicação atende às necessidades exigidas pelo processo de monitoramento e envio de alertas, já que apresenta um tempo curto, que possibilita uma rápida tomada de decisão em caso de urgências.

CONSIDERAÇÕES FINAIS

Pelos estudos desenvolvidos citados anteriormente e diante da grande evolução computacional, pode-se observar que uma área que merece atenção é a área de saúde por tratar de vida e de aplicações críticas. Assim, existindo uma grande necessidade da automatização dos processos hospitalares, visto que a maioria dos hospitais ainda encontra-se realizando os procedimentos de forma manual dificultando o controle dos dados e gerenciamento das informações, podendo levar a erros graves em relação aos pacientes.

Com o desenvolvimento do trabalho foi possível a aquisição, processamento e envio de informações a respeito do monitoramento dos pacientes, que muito auxiliará o diagnóstico de pacientes internados em uma UTI. Com isso a equipe médica poderá realizar o acompanhamento dos pacientes de forma remota.

Pode-se ainda destacar como trabalhos futuros a implementação de um preditor baseado em técnicas de redes neurais e a expansão do *middleware* para que ele possa ser sensível-contexto e orientado-serviço.

AGRADECIMENTOS

Este trabalho contou com o apoio do Laboratório de Automação Hospitalar e Bioengenharia (LAHB) do Departamento de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte e do Laboratório de Inovação Tecnológica em Saúde do Hospital Universitário Onofre Lopes da Universidade Federal do Rio Grande do Norte.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Bernstein, P. (1996) “Middleware: A Model for Distributed System Services.” *Communications of the ACM*, 39:2, 86–98.

2. Brooks, J.; Brooks, L. (1998) Automation in the medical field. *Engineering in Medicine and Biology Magazine*, IEEE Volume 17, Issue 4, July-Aug. Page(s):76, 81.
3. Coulouris, George; Kindberg, Tim; Dollimore, Jean. (2007), *Sistemas Distribuídos: Conceitos e Projeto*. Quarta Edição, Bookman, Porto Alegre.
4. Eclipse, Manual. Disponível em: <http://www.eclipse.org>. Acesso em: 12 de dezembro de 2009.
5. Figueiredo, Carlos M. S.; Nakamura, Eduardo. (2003) *Computação móvel: Novas oportunidades e novos desafios*. Belo Horizonte: Universidade Federal de Minas Gerais: 28p.
6. Leite, C. R. M. (2005) *Linguagem de Consulta para Aplicações em Tempo-Real [dissertação]*, Campina Grande: Universidade Federal de Campina Grande.
7. Murakami, Alexandre; Gutierrez, M. A.; Lage, Silvia Helena Gelas; Rebelo, Marina de Fátima de Sá; Ramires, José Antonio Franchini. (2006) A Continuous Glucose Monitoring System in Critical. *IEEE Computers in Cardiology*, v. 32, p. 10-14.
8. Várady, P., Benyo, Z. Benyo, B. (2002) An open architecture patient monitoring system using standard technologies. *IEEE Transactions on Information Technologies in Bio-medicine*, Vol. 6, No. 1, pp.95-98.
9. Varshney, U. (2006) Patient monitoring using infrastructure - oriented wireless LANs. *International Journal of Electronic Healthcare*, Volume 2, Number 2 / 149-163.
10. Stallings, William. (2005) *Redes e Sistemas de Comunicação de Dados*. Elsevier, Rio de Janeiro.
11. Kurose, James F. e Ross, Keith W. (2006) *Redes de Computadores e a Internet. Uma Abordagem Top-Down 3ª Edição*. São Paulo, Pearson.
12. Comtech M2M. (2010) GPRS tutorial - what is GPRS?. Disponível em: <http://www.comtechm2m.com/m2m-technology/gprs-tutorial.htm>. Acesso em: 15 fev 2010.
13. Bettstetter, Christian, Vögel, Hans-Jörg, e Eberspächer, Jörg. (2010) *GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface*. Disponível em: <http://www.comsoc.org/livepubs/surveys/public/3q99issue/bettstetter.html>. Acesso em: 15 fev 2010.