

Modelo arquitetural para geração de alertas aplicado ao monitoramento de pacientes em ambiente hospitalar

Bruno Gomes de Araújo*, Ricardo Alessandro de Medeiros Valentim, João Marcos Teixeira Lacerda, Diego Rodrigues de Carvalho, Marcel da Câmara Ribeiro Dantas, José Diniz Júnior

Resumo O aumento nas demandas por gerenciamento, controle e monitoramento das informações na área da automação hospitalar tem promovido um maior volume de pesquisas que são indutoras do processo de inovação tecnológica na área da saúde. Neste contexto, um aspecto considerado importante na automatização do monitoramento de pacientes consiste na eficiência em detectar e informar em tempo hábil as anomalias encontradas nos sinais vitais dos pacientes. O procedimento de notificar as ocorrências dessas anomalias à equipe médica pode ser implementado por meio da geração e envio de alertas (sonoros ou visuais). Verificando a relevância desse tipo de demanda no ambiente hospitalar, o presente artigo descreve uma arquitetura que tem como fundamento a geração e o envio de alertas, cujos dados são advindos de pacientes internados em Unidades de Terapia Intensiva (UTI). A premissa foi, portanto, otimizar o processo de comunicação das anomalias detectadas de modo que a equipe médica responsável seja notificada de tais eventos de maneira mais eficiente. A arquitetura de comunicação, definida para o ambiente hospitalar, baseou-se em estudos realizados na UTI do Hospital Universitário Onofre Lopes (HUOL). Tais estudos possibilitaram uma análise de requisitos que permitiu definir um gerador de alertas personalizados, e o envio desses para dispositivos móveis das equipes médicas. O processo de envio dos alertas foi baseado em um algoritmo de escalonamento de tempo real, fazendo uso de um middleware e de computação móvel e distribuída, sendo esses os aspectos inovadores dessa arquitetura.

Palavras-chave Automação hospitalar, Monitoramento de pacientes, Sinais vitais, Middleware, Computação móvel, Computação ubíqua.

Architectural model for generating alerts applied to the monitoring of patients in a hospital environment

Abstract *The increase in demand for the management, control and monitoring of information in hospitals has promoted a greater volume of research that induces the process of technological innovation in healthcare. In this context, an important aspect to consider in the automation of patient monitoring is the efficiency to detect and report anomalies in patients' vital signs in a timely manner. The procedure for notifying the medical staff of these anomalies can be implemented by generating and sending alerts (either audible or visual). Noting the relevance of this demand in the hospital environment, this paper describes an architecture based on the generation and transmission of alerts, whose data are coming from patients hospitalized in intensive care units (ICU). The premise was therefore to optimize the procedure for reporting deficiencies so that the medical staff in charge is notified of such events more efficiently. The communication architecture in hospitals, used in this paper, was based on studies conducted at the ICU of the University Hospital Onofre Lopes (HUOL). These studies allowed an analysis of requirements that lead to the definition of a generator of custom alerts, and the sending of these alerts to mobile devices kept by medical staff. The process of sending those alerts was based on a real time scheduling algorithm making use of a middleware and both mobile and distributed computing, which are the innovative aspects of this architecture.*

Keywords *Hospital automation, Patient monitoring, Vital signs, Middleware, Mobile computing, Ubiquitous computing.*

Extended Abstract

Introduction

Hospital procedures, such as monitoring of hospitalized patients in intensive care units (ICU) are being constantly computerized and automated, and this process gains even more attention with the appearance of new information technologies. Nevertheless, there is no mechanism to automate the communication between biomedical equipment, used for monitoring patients, and the medical staff in a hospital environment. One of the solutions used is the sending of alerts. The time to generate an alert and sending it to some medical staff (an alert latency) must occur according to the time constraints imposed by the process of patient monitoring. Compliance with these restrictions enables a more appropriate decision making in relation to the procedures recommended in the protocols of medical care. Aiming at this, we developed an architecture for generating and sending alerts from the process of patient monitoring, based on a real-time scheduling system using mobile, ubiquitous and context-sensitive computing.

Material and Methods

For the construction of the architecture, we used the following technologies: Java Programming Language, Web service, Sockets, communication protocols type TCP (Transmission Control Protocol) and IEEE 802.11, and RFID (Radio Frequency Identification) (Figure 1). Its execution flow begins when it receives, through a Web Service installed on the server, a new alert abnormality (anomaly) detected in a patient being monitored by another system. Information about the anomaly detected, which are previously registered in the system (Figure 3), are passed to the anomaly scheduler which, using a list(stack alerts), performs the sorting of anomalies according to their time constraints, in order to ensure the temporal correctness of the mechanism for sending alerts, as represented in Figure 4. At the same time, the architecture generates the alert message that is sent to the physician's mobile device. To control the transmission, the architecture provides a device scheduler, whose purpose is to avoid the overhead of sending alerts to the medical staff, performing the sorting of the devices that have higher priority to receive the next alert, in accordance with the amount of messages received until the time of shipping. Shortly after, the message is sent to the device with the highest priority through the communication protocol TCP using the model of client-server communication. Upon receiving the message, the mobile device triggers a ringing and vibratory alert for 10 seconds

(Figure 5). There are two other modules that are responsible for submitting reports on the general situation of monitoring conducted at the moment that a professional arrives at the hospital (Figure 6). To verify the feasibility of the architecture, a number of tests were performed, in which the processing times and sending times were measured. The processing time corresponds to the average time the architecture takes to receive a new abnormality and to prepare a new warning message. On the other hand, the sending time consists of the average time that it takes to send a warning message from the architecture to the physician's mobile device. Two tests were run, the first consisting of a sequential sending of 1000 anomalies to the architecture, where the average processing time and average sending time of each anomaly was stored in a data base. The second test used the JMeter tool to check the communication efficiency between the architecture and the mobile device, sending 1000 warning messages in the same format defined for the architecture and storing the total time of communication.

Results

An overview of the developed architecture can be seen in Figure 7, which displays all the deployed services. The general purpose of the performed tests for the architecture was to measure the working time and the efficiency of the system, validating its usability in a real environment. After the execution of the architecture, the data stored in the data base was analyzed and extracted to generate the graphics, providing a general view of all obtained times. The first test originated the graphics in Figure 8 and 9. Figure 8 represents the processing time of the architecture, with an average time of 35,026 milliseconds. Sending times can be found in the Figure 9, with an average time of 3,833 milliseconds. The test performed by JMeter produced an average time of 183 milliseconds of communication between the architecture and the mobile device, as shown in Figure 10. The information obtained by running JMeter is presented in Table 1.

Discussion and Conclusion

The tests performed on the system usability indicate the possibility of using the architecture in a critical environment, such as ICU. The total time for processing and message sending were on the order of milliseconds, showing a good match with the requirements described in the literature. This allows agility in the work of medical staff to act over any abnormality detected during the monitoring of patients.

Introdução

Com o crescimento da computação móvel e com o surgimento de várias tecnologias e de novos paradigmas de comunicação, a automatização de sistemas tornou-se uma tendência. Esse reflexo já pode ser observado na automação hospitalar, por exemplo, com definição da escala de médicos plantonistas que pode ser totalmente automatizada. Apesar da grande difusão dessa nova tendência, alguns processos utilizados na área hospitalar ainda são realizados de forma manual (Nof, 2009), o que permite um campo vasto de estudo. Com isso, várias pesquisas estão sendo desenvolvidas no intuito de automatizar, de alguma forma, tais processos (Brooks e Brooks, 1998). Como exemplo, podemos citar o desenvolvimento de sistemas para o monitoramento de pacientes apresentados por Murakami *et al.* (2006), Várady *et al.* (2002) e Varshney (2006).

O monitoramento de pacientes internados em Unidades de Terapia Intensiva (UTI) é considerado uma tarefa crítica, pois está relacionado diretamente com a saúde e eficiência no tratamento dos mesmos (Leite *et al.*, 2011). Assim, é importante que se busque o desenvolvimento de mecanismos eficientes para monitoramento de pacientes, com o objetivo de notificar a ocorrência de anormalidades em tempo hábil pela equipe médica responsável, permitindo, deste modo, que seja possível realizar tomadas de decisões mais adequadas.

A notificação dessas anormalidades detectadas durante o processo de monitoração de pacientes permite, portanto, uma diversidade de soluções, dentre as quais o envio de alertas, conforme descrito neste trabalho. Trata-se de um modelo simples e conhecido entre os profissionais de saúde, já habituados aos alertas sonoros emitidos, por exemplo, por monitores multiparâmetros das UTIs.

O tempo para gerar um alerta e enviá-lo até um profissional responsável (latência de um alerta) deve ser adequado às restrições temporais impostas pelo processo de monitoramento de pacientes. De acordo com Rausch e Segal (2009), a latência de uma mensagem de alerta para pacientes monitorados, a depender da gravidade, pode ser classificado em três faixas temporais:

- Tempo real – a mensagem deve ser entregue em menos de 10 segundos;
- Periódico – a mensagem deve ser entregue em menos de 1 minuto; e
- Sob demanda – a mensagem deve ser entregue em menos de 10 minutos.

Isso implica, portanto, que os sistemas de envio de alertas que cumpram tais metas temporais, em função

do tipo de serviço disponibilizado (classificação), possibilitam uma tomada de decisão mais adequada em relação aos procedimentos preconizados nos protocolos de atendimento médico.

É comum ainda encontrar processos de comunicação, para monitoração de pacientes, que ainda são realizados de forma manual nos hospitais, fator que torna menos eficiente o monitoramento, pois podem gerar atrasos que favorecem a depreciação da qualidade do serviço, prejudicando o atendimento de pacientes em situação crítica. Além disso, alguns sistemas utilizam alertas sonoros sem o devido direcionamento a um responsável, podendo, muitas vezes, não serem ouvidos pela equipe ou causar desconforto sonoro em todo ambiente da UTI.

Existem alguns trabalhos descritos na literatura, focados no monitoramento de pacientes e envio de alertas. Kogure *et al.* (2005) escrevem o uso de dispositivos móveis com a possibilidade de acompanhamento dos sinais vitais dos pacientes, diretamente por aparelhos móveis via conexão 3G (3rd Generation Mobile Telecommunications). As informações são apresentadas através de gráficos e textos na tela do dispositivo, permitindo que o médico possa buscar informações sobre qualquer paciente. Neste caso, a informação é somente obtida quando o médico deseja, ou seja, não chega de forma automatizada. Esse modelo de monitoramento é mais eficiente quando aplicado para os casos de acompanhamento em *homecare*, pois permite que o médico possa obter os dados dos pacientes sem se deslocar até suas residências.

O trabalho de Murakami *et al.* (2006) também utiliza dispositivo móvel para o acompanhamento em tempo real dos níveis de glicose em pacientes internados em UTI. Neste caso, foi utilizado um *PocketPC* para o acompanhamento, que baixa as informações a cada 5 minutos e as exibe ao usuário. O trabalho contribui de forma bem específica somente para o problema de monitoramento dos níveis de glicemia, não permitindo nenhum outro tipo de monitoramento.

Na mesma linha, Zhang *et al.* (2007) descrevem um sistema baseado em dispositivos móveis para o monitoramento de pacientes, com aplicação desenvolvida na linguagem de programação *Java* e que se comunica por meio de uma rede de comunicação 3G, possibilitando ao médico acompanhar os dados dos sinais vitais de um paciente na tela do dispositivo. Esse trabalho é mais abrangente que os demais apresentados, pois permite monitorar de forma pró-ativa os pacientes e trata de um conjunto de sinais vitais. Todavia, não leva em consideração o problema de escalonamento de mensagens de alertas e dos dispositivos que estão

mais aptos a receber essas mensagens. Esses aspectos apresentam duas implicações: 1) aumento na latência dos alertas, pois todos os sinais vitais têm a mesma prioridade; 2) modelo um para um no envio dos alertas, pois somente um médico pode receber mensagens de alerta de um determinado paciente, ou seja, não há escalonamento de dispositivos, o que pode ser um ponto de gargalo do sistema quando submetido a cenários de maior carga.

Neste contexto, verifica-se que houve avanços em relação ao monitoramento de pacientes. Contudo, percebem-se ainda lacunas que demandam soluções efetivas para a melhoria da qualidade desse tipo de serviço e, consequentemente, do atendimento aos pacientes. Desta forma, este trabalho descreve o projeto e a implementação de uma arquitetura baseada nos conceitos da computação distribuída e no uso de dispositivos móveis, cujo objetivo é estabelecer um mecanismo de envio de alertas para pacientes monitorados, considerando aquelas lacunas encontradas. Para tanto, utiliza-se do envio de alertas, provenientes do monitoramento de quaisquer tipos de sinais vitais dos pacientes de forma pró-ativa, aplicando conceitos de escalonamento baseados em níveis de prioridades para os alertas e na disponibilidade dos profissionais da equipe médica. Tais elementos compõem as principais inovações implementadas nesta arquitetura.

Este artigo também apresenta um *middleware* que faz parte da arquitetura e contribui para superar as questões pertinentes a heterogeneidade das plataformas encontradas nos hospitais, tanto de sistemas de software quanto de sistemas operacionais e plataformas de hardware. Esse artefato da arquitetura permite, portanto, sua integração com qualquer sistema encontrado em ambientes hospitalares.

Materiais e Métodos

Tecnologias utilizadas na construção da arquitetura

Para construção da arquitetura proposta neste artigo, foram utilizadas as seguintes tecnologias: linguagem de programação *Java*, *webservice*, *sockets*, protocolos de comunicação do tipo TCP (*Transmission Control Protocol*) e IEEE 802.11, e RFID (*Radio-Frequency Identification*).

A linguagem de programação *Java* foi utilizada por permitir a portabilidade dos códigos, ser multiplataforma e por possuir uma API (*Application Programming Interface*) específica para o desenvolvimento de aplicações distribuídas. Esta API foi utilizada para o desenvolvimento de um *middleware*

que disponibiliza dois envios assíncronos de dados (Bernstein, 1996), um via GPRS (*General Packet Radio Service*), possibilitando o envio de mensagens SMS (*Short Message Service*) para telefones celulares, e outro via internet, que permite o envio de e-mails com maiores informações sobre os monitoramentos, conforme representado na Figura 1.

A aplicação de *webservice* na arquitetura teve como finalidade prover a interface de comunicação com qualquer outro sistema que esteja realizando o monitoramento de pacientes, possibilitando a comunicação de forma transparente e independente da tecnologia, garantindo a interoperabilidade com outros sistemas por meio do protocolo SOAP (*Simple Object Access Protocol*) (W3C, 2010).

O uso de *sockets* de comunicação do tipo TCP, configurou-se como um dos aspectos importantes da arquitetura, garantindo um serviço de transporte confiável e com comunicação estável (Kurose e Ross, 2006), premissa indispensável para a entrega dos alertas.

O padrão IEEE 802.11 (Institute..., 2012) foi aplicado a fim de permitir a mobilidade da arquitetura, permitindo que os dispositivos móveis se conectem ao servidor por meio da rede sem fios do hospital (Figura 2) e também por ser um padrão amplamente utilizado e com alto poder de interoperabilidade.

O uso da tecnologia RFID na arquitetura permitiu implementar os conceitos de computação ubíqua e sensível a contexto, identificando automaticamente os usuários (equipe de saúde) e tornando-a imperceptível e adaptável a esses (Baldauf *et al.*, 2007; Bardram *et al.*, 2006; Blanks *et al.*, 2007).

Visão da arquitetura: descrição e fluxo de atividades

A arquitetura foi desenvolvida de forma modularizada e seu funcionamento é determinado por um conjunto

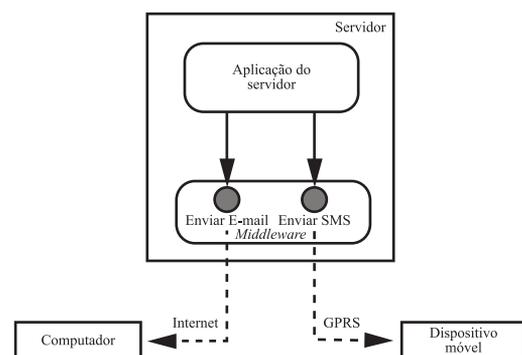


Figura 1. Serviços do Middleware para o envio de relatórios.
Figure 1. Middleware services for dispatching reports.

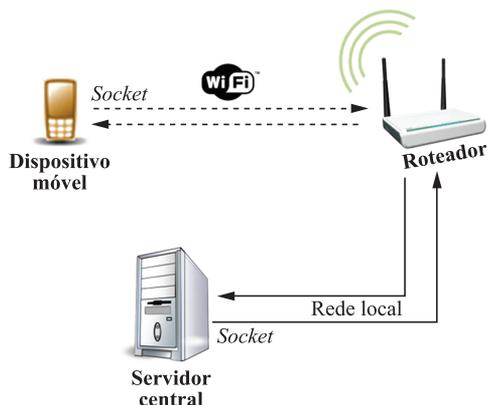


Figura 2. Comunicação do sistema com um dispositivo móvel.
Figure 2. Communication between the system and a mobile device.

de atividades que seguem um determinado fluxo de execução. Esse fluxo de atividades possibilita visualizá-la de forma geral, bem como, compreender a sua integração com o ambiente hospitalar.

A primeira ação, que dispara o uso da arquitetura, consiste no recebimento de alguma anormalidade (denominada “anomalia”) detectada em um paciente e enviada pela rede do hospital. A arquitetura disponibiliza uma interface para o recebimento de anomalias detectadas através de qualquer outro sistema desenvolvido, que esteja operando no hospital. Durante o processo de monitoramento de dados, basta que este outro aplicativo conheça a interface de recebimento da arquitetura para enviar uma mensagem contendo as informações necessárias sobre a anormalidade detectada. Os dados da anomalia são cadastrados por meio de uma tela de configuração, apresentada na Figura 3.

A arquitetura contém uma tabela de referência, implementada em um banco de dados, que armazena todas as anomalias monitoradas. Cada anomalia é composta pela seguinte estrutura de dados:

- Nome único: utilizado para identificar a anomalia;
- Valores máximo e mínimo: limites utilizados para caracterizar o estado normal ou anormal de um paciente durante o monitoramento. Esses dados são opcionais;
- Restrição temporal (*deadline*): limite máximo de tempo no qual o paciente deve ser atendido após a detecção da anomalia. Esta informação é importante para o Escalonador de Anomalias, que irá ordená-las de acordo com esse atributo e deve ser informada pela equipe médica.

As anomalias são repassadas automaticamente ao Escalonador de Anomalias, que representa a

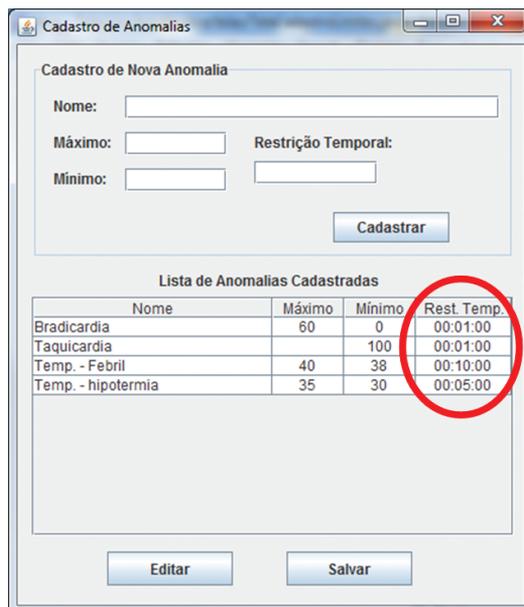


Figura 3. Cadastro de anomalias na arquitetura.
Figure 3. Registry of abnormalities in the architecture.

segunda ação no fluxo de atividades. Esse escalonador mantém uma lista (pilha de alertas) com todas as anomalias recebidas, servindo de base para garantir a correte temporal do mecanismo de envio de alertas e provendo um serviço que visa garantir que o atendimento das mesmas ocorra nos prazos impostos pelas restrições temporais (correte temporal). A ordem de armazenamento das anomalias na lista depende de suas respectivas restrições (prioridades). Inicialmente, as anomalias recebidas recebem o status de ‘Bloqueadas’.

O Escalonador de Anomalias atua então sobre aquela lista, atribuindo para a anomalia de menor restrição temporal (*deadline*) a maior prioridade, colocando-a, assim, no topo da pilha de alertas (ordenação decrescente das anomalias com base nos seus *deadlines*). Para que esse mecanismo seja executado corretamente, sempre que uma nova anomalia é recebida o escalonador reordena a lista. Para tanto, toma como referência o instante (hora, minutos, segundos e milissegundos) que a anomalia foi recebida pela arquitetura, iniciando um evento de contagem, para possibilitar o cálculo do tempo restante para atingir o prazo final definido, passando do status de ‘Bloqueada’ para ‘Executando’. Se a anomalia tiver sido atendida pelo médico antes do seu *deadline*, ela será armazenada no banco de dados com o status de ‘Atendida’, caso contrário, ‘Não Atendida’, como mostra a Figura 4.

Para a construção do escalonador foram utilizadas políticas de sistemas em tempo-real para gerenciar o

controle das anomalias. O algoritmo de escalonamento aplicado foi baseado no *Earliest Deadline First* (EDF) (Moghaddas e Hamidzadeh, 1999).

Em paralelo à operação do Escalonador de Anomalias, a arquitetura prepara as mensagens de alertas a serem enviadas aos dispositivos da equipe médica responsável. Para o controle do envio dessas mensagens aos dispositivos móveis, foi construído o Escalonador de Dispositivos, cujo propósito é evitar a sobrecarga no envio de alertas para a equipe médica, impedindo, por exemplo, que um membro da equipe receba um maior número de alertas em relação aos demais.

Esta funcionalidade da arquitetura permite o balanceamento do envio de alertas entre os dispositivos disponíveis para o monitoramento de pacientes. Assim, é disponibilizada uma lista de dispositivos que estão ligados e disponíveis para receber novos alertas. Esta unidade contém ainda informações sobre os dispositivos cadastrados na rede hospitalar que estão aptos a receber mensagens de alertas, os médicos cadastrados e quantos alertas já foram enviados. O processo de ordenação na arquitetura ocorre sempre que um novo dispositivo cadastrado entra ou sai da rede de comunicação do hospital ou quando ocorre a notificação de que um dispositivo recebeu um novo alerta. O celular (dispositivo móvel) que tiver recebido menos alertas é o que terá a maior prioridade (topo da lista) para receber uma nova notificação.

Logo após a ordenação dos dispositivos, a mensagem de alerta é enviada. Este envio consiste em um dos mecanismos mais importantes, já que é responsável por controlar o envio e recebimento de

mensagens entre os usuários e a arquitetura (interface de comunicação). O modelo de comunicação aplicado foi o cliente-servidor, pois se ajusta melhor à forma empregada na arquitetura, uma vez que permite a troca de informações bidirecionais entre o servidor e o dispositivo móvel, baseado em requisição e resposta. O funcionamento deste componente da arquitetura depende da instalação de uma aplicação no dispositivo móvel para receber as mensagens de alertas, que, ao ser iniciada, tentará se conectar com a Central. Se a conexão for estabelecida, numa tela de espera será aberta até que os alertas sejam gerados e recebidos no dispositivo, como mostra a Figura 5.

Ao receber uma mensagem, o celular apresenta-a na tela para chamar a atenção do responsável, disparando um alerta sonoro e vibratório durante 10 segundos.

Atuando de forma concomitante na arquitetura, existem mais dois componentes: Detecção Automática e Envio dos Relatórios. O propósito desses artefatos da arquitetura é detectar, de forma ubíqua, os usuários que irão interagir com o sistema no instante em que chegam ao hospital (Figura 6) e, logo em seguida, enviar um relatório com os detalhes dos monitoramentos que estão sendo realizados naquele instante, como data e hora de entrada dos pacientes internados, diagnósticos e lista dos alertas gerados e enviados até o momento do recebimento dos relatórios.

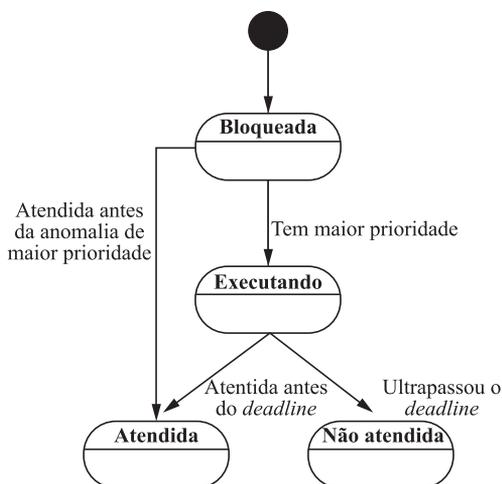


Figura 4. Diagrama de estados do Escalonador das Anomalias.
 Figure 4. State diagram for the Anomalies Scheduler.



Figura 5. Tela da aplicação com os alertas recebidos.
 Figure 5. Screen of the application with the alerts received.

Experimentos

Para verificar a viabilidade da utilização da arquitetura (validação), foram realizados dois testes por meio do envio sequencial de vários alertas. O ambiente de testes consiste em: um computador *Core 2 Duo*, 4GB de memória, Windows7® 32 bits; um celular *Nokia* modelo N95; e um roteador com capacidade de 54 Mbps.

O primeiro teste do sistema consiste no envio de 1000 anomalias para a arquitetura, com o propósito de observar os tempos médios de recebimento de uma nova anomalia e preparação de uma nova mensagem

de alerta (tempo de processamento da arquitetura), assim como a latência para que uma mensagem seja enviada até o celular (tempo de envio). A mensagem é enviada no seguinte formato:

- “Paciente %1\$2s, do leito %2\$2s, está com %3\$2s”.

Os textos %1\$2s, %2\$2s e %3\$2s são substituídos respectivamente pelo nome do paciente monitorado (10 caracteres), identificador do leito (2 caracteres) e qual anomalia detectada nos sinais vitais (10 caracteres). No total, a mensagem corresponde a um total de 53 bytes. O envio foi feito por meio do protocolo TCP,

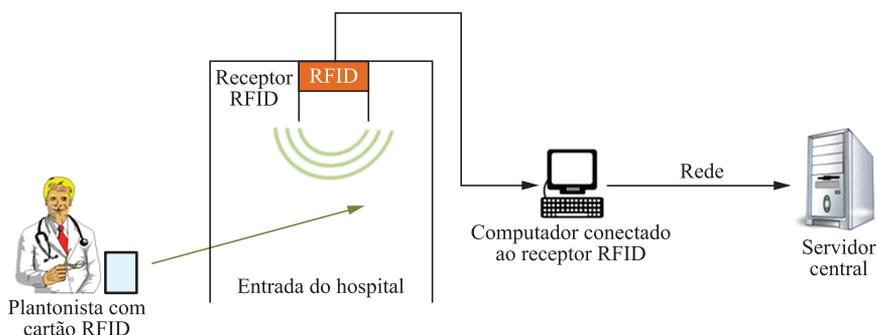


Figura 6. Módulo de detecção automática de usuários.
Figure 6. Module of automatic users detection.

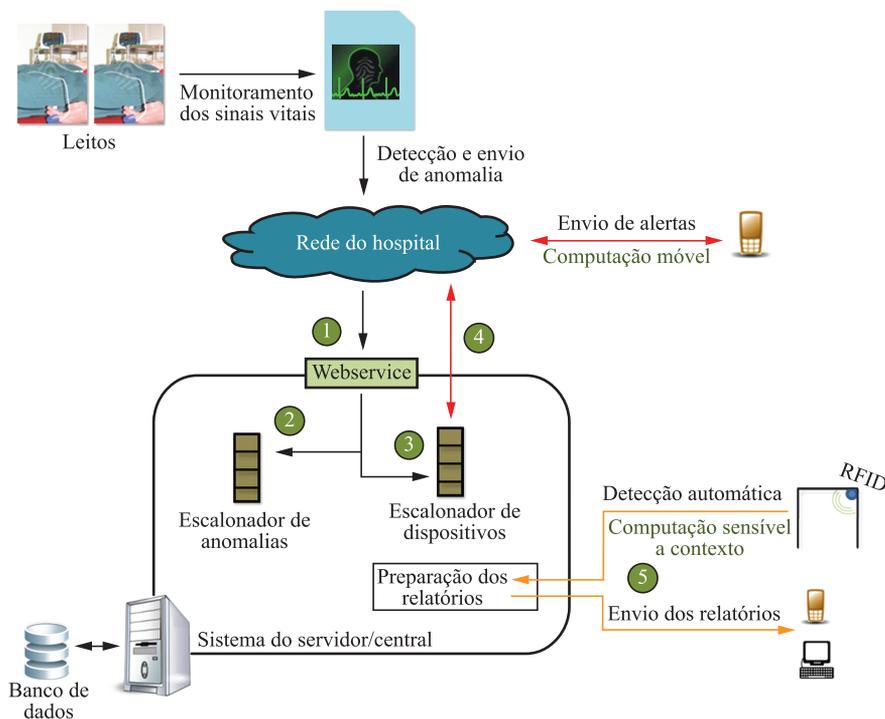


Figura 7. Representação do Modelo Arquitetural de Comunicação.
Figure 7. Representation of the proposed Architectural Model of Communication.

utilizando para isso a implementação em *Socket*, com uma taxa de envio de 1000 anomalias sequencialmente.

O segundo teste foi realizado utilizando a ferramenta *JMeter*, que é desenvolvida em Java e permite a realização de testes funcionais de desempenho em diferentes tipos de sistemas e servidores, como *Web*, *SOAP*, Banco de Dados via *JDBC* (Java Database Connectivity), entre outros (Apache..., 2011). O objetivo deste teste foi analisar a comunicação entre a arquitetura e os dispositivos móveis, uma vez que a ferramenta apresenta resultados mais detalhados e precisos sobre o teste (Garousi, 2008). Foram enviadas 1000 mensagens de alertas para o dispositivo móvel do médico em um intervalo total de 500 segundos, com o mesmo formato da mensagem gerada pela arquitetura. Posteriormente, o tempo total de cada comunicação foi armazenado.

Resultados

A Figura 7 mostra uma visão geral da arquitetura e seus elementos. Nela estão reunidos todos os serviços descritos nas seções anteriores, desde o recebimento de uma nova anomalia, passando pelo processamento das informações, até seu envio.

Os resultados do primeiro teste são apresentados nas Figuras 8 e 9.

A Figura 8 representa todos os testes realizados (eixo x), numerados de 1 a 1000, e o tempo de processamento (eixo y) de cada envio, em milissegundos. Este teste apresentou um tempo médio de processamento igual a 35,026.

A Figura 9 representa os envios realizados (eixo x), numerados de 1 a 1000, e o tempo de envio (eixo y) de cada um deles. O resultado apresentou uma média de 3,833 milissegundos para todos os envios realizados.

O resultado do segundo teste é apresentado na Figura 10, considerando o tempo de todas as comunicações realizadas através do programa *JMeter* (Apache..., 2011).

A média de todos os envios foi de 183 milissegundos. Além disso, o experimento permitiu aferir outros dados referentes à comunicação aferidos pelo *JMeter* (Apache..., 2011), como se observa na Tabela 1. Nela estão contidas informações como:

- O tempo médio de todas as comunicações (Média): 183 milissegundos;
- A comunicação que ocorreu no menor tempo (Mín.): 96 milissegundos;
- A comunicação mais demorada (Máx): 1171 milissegundos;
- O Desvio padrão: 70,49 milissegundos;
- Quantas comunicações não foram realizadas (% de Erro): 0%;

- A Vazão, representando o número de comunicações realizadas por segundo (Vazão): 2,0/seg; e
- Quantos kB (kilobytes) foram trafegados por segundo durante toda a comunicação (KB/s): 0,01 kB.

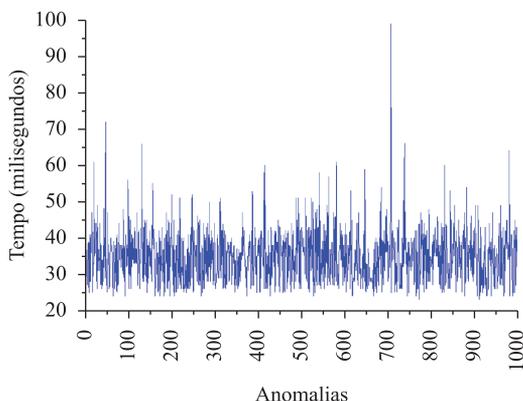


Figura 8. Gráfico com os resultados do teste de processamento.
Figure 8. Graph with results of the test of processing

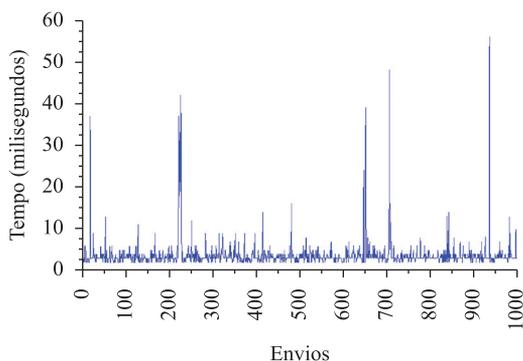


Figura 9. Gráfico com os resultados do teste de envio de mensagens de alerta.
Figure 9. Graph with results of the test of dispatching alert messages.

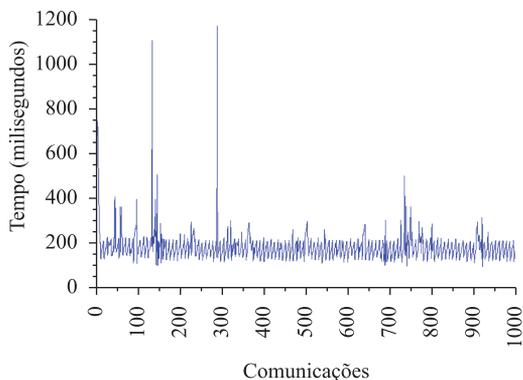


Figura 10. Gráfico com os resultados do teste de envio feito com o *JMeter* (Apache..., 2011).
Figure 10. Graph with results of the test of transmission measured by the *JMeter* (Apache..., 2011).

Tabela 1. Resultados do relatório de sumário do JMeter (Apache..., 2011).

Table 1. Results of JMeter summary report (Apache..., 2011).

Amostras	Média (ms)	Mín. (ms)	Máx. (ms)	Desvio padrão (ms)	% de Erro	Vazão	kB/s
1000	183	95	1171	70,49	0,00	2,0/seg	0,01

Discussão

De acordo com os resultados dos testes, os tempos de transmissão das mensagens de alerta e de processamento da arquitetura se deram na ordem de milissegundos, além de não apresentarem erros durante as comunicações. Com base nesses resultados, e de acordo com os tempos definidos por Rausch e Segal (2009), podemos observar que aplicação atende às necessidades exigidas pelo processo de monitoramento de pacientes e envio de alertas, pois apresenta uma taxa de amostragem alta em relação ao processo de monitoramento de pacientes, possibilitando, portanto, uma rápida tomada de decisão em caso de urgências.

Conclusão

A arquitetura apresenta um mecanismo de geração e envio de alertas eficiente. Tal afirmação pode ser verificada pelos resultados dos experimentos de validação realizados, apresentando valores adequados para a comunicação e garantindo que os médicos possam acompanhar o monitoramento e sejam alertados de qualquer alteração nos sinais vitais do paciente em um tempo hábil, conforme descrito por Rausch e Segal (2009). Tais características contribuem, principalmente, para o rápido diagnóstico de pacientes internados em uma UTI e possibilita seu acompanhamento eficiente por parte da equipe médica.

Pode-se destacar como trabalhos futuros a implementação de um mecanismo tolerante a falhas para a arquitetura, uma vez que a mesma é executada em um ambiente que utiliza dados críticos. Da mesma forma, é importante o estabelecimento de critérios de segurança para as informações utilizadas pela arquitetura, garantindo a privacidade e sigilo dos dados dos pacientes. A estrutura flexível desta arquitetura permite ainda que a mesma seja modificada para que possa ser utilizada não apenas em UTIs, mas também em outros ambientes de um hospital, como quartos, enfermarias, etc.

Agradecimentos

Ao Laboratório de Inovação Tecnológica em Saúde (LAIS) do Hospital Universitário Onofre Lopes (HUOL) da Universidade Federal do Rio Grande do Norte (UFRN), que ofereceu a infraestrutura para o desenvolvimento da pesquisa.

Referências

- Apache Software Foundation. JMeter. [Internet] [cited 2011 mai 20]. Available from: <http://jakarta.apache.org/jmeter/>
- Baldauf M, Dustar S, Rosenberg F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*. 2007; 2(4):263-77.
- Bardram JE, Hansen TR, Mogensen M, Soegaard M. Experiences from real-world deployment of context-aware technologies in a hospital environment. *UBICOMP 2006: Ubiquitous Computing, Lecture Notes in Computer Sciences*. 2006; 4206:369-86. http://dx.doi.org/10.1007/11853565_22
- Bernstein PA. Middleware: a model for distributed system services. *Communications of the ACM*. 1996; 39(2):86-98. <http://dx.doi.org/10.1145/230798.230809>
- Blanks J, Hanny D, Thompson Les G, Pachano MA. RFID applied. Canada: Wiley, John & Sons; 2007. <http://dx.doi.org/10.1002/9780470168226>
- Brooks J, Brooks L. Automation in the medical field. *Engineering in Medicine and Biology, IEEE*. 1998; 17(4):76-81.
- Garousi V. Traffic-aware stress testing of distributed real-time systems based on UML models in the presence of time uncertainty. In: *ICST 2008: Proceedings of the 1th International Conference on Software Testing, Verification, and Validation*; 2008 Apr. 9-11; Lillehammer, Norway. IEEE; 2008. p. 92-101.
- Institute of Electrical and Electronics Engineers - IEEE. The working group for WLAN standards. IEEE 802.11TM Wireless Local Area Networks [Internet]. [cited 2012 Fev 05]. Available from: <http://www.ieee802.org/11/>
- Kogure Y, Matsuoka H, Kinouchi Y Akutagawa M. The development of a remote patient monitoring system using Java-enabled mobile phones. In: *IEEE-EMBS 2005: Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society*; 2005 Sept. 1-4; Shanghai, China. IEEE; 2005. p. 2157-60.
- Kurose JF, Ross KW. *Redes de computadores e a internet: Uma abordagem top-down*. 3th ed. São Paulo: Pearson; 2006.
- Leite CRM, Sizio GRA, Dória Neto AD, Valentim RAM, Guerreiro AMG. A fuzzy model for processing and monitoring vital signs in ICU patients. *BioMedical Engineering Online*. 2011; 10:68. <http://dx.doi.org/10.1186/1475-925X-10-68>
- Moghaddas M, Hamidzadeh B. Batching earliest deadline first scheduling. In: *WORDS 1999: Proceedings of the 5th International Workshop on Object-Oriented Real-Time Dependable Systems*; 1999 Nov. 18-20; Monterey, USA. Monterey; 1999. p. 29-34.

Murakami A, Gutierrez MA, Lage SHG, Rebelo MFS, Guiraldelli RHG, Ramires JAF. A continuous glucose monitoring system in critical cardiac patients in the intensive care unit. In: IEEE Computers in Cardiology: Proceedings of the IEEE Computers in Cardiology; 2006 Sept. 17-20; Valencia, Espanha. IEEE; 2006. p. 233-6.

Nof SY. Springer handbook of automation. Springer; 2009. <http://dx.doi.org/10.1007/978-3-540-78831-7>

Rausch T, Segal B. System and method for patient care [Internet]. 2009 [citado em 2011 out 10]. Available from: <http://www.freepatentsonline.com/y2009/0012812.html>

Várady P, Benyo Z, Benyo B. An open architecture patient monitoring system using standard technologies. IEEE Transactions on Information Technologies in

Bio-medicine. 2002; 6(1):95-8. PMID:11936602. <http://dx.doi.org/10.1109/4233.992168>

Varshney U. Patient monitoring using infrastructure-oriented wireless LANs. International Journal of Electronic Healthcare. 2006; 2(2):149-63.

W3C. Web services architecture [Internet]. [citado em 2011 mai 10]. Available from: <http://www.w3.org/TR/ws-arch/>

Zhang P, Kogure Y, Matsuoka H, Akutagawa M, Kinouchi Y, Shang O. A remote patient monitoring system using a Java-enabled 3G mobile phone. In: EMBS 2007: Proceedings of the 29th Annual International Conference of the IEEE - Engineering in Medicine and Biology Society; 2007 Aug.22-26; Lyon, França. IEEE; 2007. p. 3713-6.

Autores

Bruno Gomes de Araújo*, Ricardo Alessandro de Medeiros Valentim, João Marcos Teixeira Lacerda, Diego Rodrigues de Carvalho, Marcel da Câmara Ribeiro Dantas, José Diniz Júnior

Programa de Pós-Graduação em Engenharia Elétrica e de Computação – PPGEEC, Laboratório de Inovação Tecnológica em Saúde, Hospital Universitário Onofre Lopes – HUOL, Universidade Federal do Rio Grande do Norte – UFRN, Av. Nilo Peçanha, 620, Petrópolis, CEP 59012-300, Natal, RN, Brasil.