



UMA ARQUITETURA DE SOFTWARE PARA O GERENCIAMENTO DE UM AMBIENTE UBÍQUO E PERVASIVO NA ÁREA DA SAÚDE

Wagner da Silva Silveira¹
Carlos Oberdan Rolim²

Resumo: O objetivo deste trabalho consiste no desenvolvimento de uma arquitetura de software voltada para o gerenciamento de um ambiente ubíquo e pervasivo na área da saúde, bem como o desenvolvimento de um protótipo que faça uso do mesmo com fins de validação e obtenção de resultados. Essa arquitetura tem como principal objetivo facilitar e agilizar o desenvolvimento de outras aplicações com objetivos mais específicos, tal como exemplo, o monitoramento instantâneo de sinais vitais de pacientes. Tal objetivo torna-se possível tendo em vista que o avanço tecnológico permitiu a miniaturização e potencialização do poder comunicacional e computacional de dispositivos móveis. A união desses dispositivos com sensores capazes de capturar informações do ambiente e sinais vitais de pacientes, torna-se possível o desenvolvimento de soluções para conhecidos problemas existentes na área da saúde.

Palavras-chave: Medicina ubíqua. Ubicomp. Ambientes inteligentes.

1 INTRODUÇÃO

Segundo o que Mark Weiser (1952 - 1999) formalizou em seu consagrado artigo (WEISER, 1991, p. 94-104), a computação ubíqua e pervasiva, também conhecida como ubicomp, é o terceiro grande paradigma computacional. Onde ambientes impregnados de sensores e dispositivos computacionais, muitos deles móveis, trabalham cooperativamente, muitas vezes de forma invisível para os usuários. Eliminando assim a tradicional forma de interação direta entre homem e máquina, fazendo com que os usuários não percebam que estão interagindo com sistemas computacionais.

Desenvolver aplicações para tornar a computação ubíqua e pervasiva realidade, não é uma tarefa trivial para desenvolvedores de sistemas. Existem diversos fatores complexos que devem ser levados em consideração para desenvolvimento de tais aplicações. Esses fatores vão desde o gerenciamento da comunicação entre os dispositivos presentes no ambiente, a segurança informação comunicada, a escalabilidade e expressividade para que seja possível

¹ Graduando em Ciência da Computação - URI - Universidade Regional Integrada do Alto Uruguai e das Missões - wagner@santoangelo.uri.br.

² Doutorando em Ciência da Computação - UFRGS - Universidade Federal do Rio Grande do Sul - carlos.oberdan@inf.ufrgs.br.



que no ambiente exista uma grande quantidade de dispositivos, até a extensibilidade que permite a agregação de novas funcionalidades.

Segundo o IBGE, o número de idosos no planeta jamais foi tão grande em toda a história. Em 2050, a expectativa de vida nos países desenvolvidos será de 87,5 anos para os homens e 92,5 para as mulheres (contra 70,6 e 78,4 anos em 1998). Já nos países em desenvolvimento, será de 82 anos para homens e 86 para mulheres, ou seja, 21 anos a mais do que hoje, que é de 62,1 e 65,2. (IBGE, 2010, p.3).

Diante dessas informações, o número de idosos tende a aumentar em escala mundial. A maior parte desses idosos vive com a família, a qual, na maioria das vezes, também tem uma idade avançada; outros, entretanto, acabam vivendo sozinhos em virtude do abandono familiar. Por outro lado, existe pessoas com idade menos avançada mas que necessitam de cuidados médicos no mesmo grau de importância.

Havendo à necessidade de solucionar os diversos problemas existentes na área da saúde que este trabalho foi elaborado. Onde principal objetivo foi desenvolver uma arquitetura de software que facilite o desenvolvimento de outras aplicações para a área da saúde, e que sejam caracterizadas com todos os benefícios proporcionados pela ubicomp. Permitindo que pacientes possam ser melhor monitorados, que possíveis acidentes sejam previstos e evitados, e que a distância entre pacientes e profissionais da saúde não seja um empecilho.

O artigo está estruturado da seguinte forma: a seção 2 apresenta a arquitetura de software proposta; na seção 3 são demonstrados os protótipos para validação da arquitetura; as conclusões e trabalhos futuros são apresentados na seção 4.

2 ARQUITETURA DE SOFTWARE

Aplicações pervasivas precisam de um middleware para servirem de interface entre os muitos dispositivos diferentes (desktops, PDAs, notebooks, sensores, etc.) e as aplicações do usuário final (SAHA; MUKHER JEE, 2003, p. 26). O middleware é uma camada de software que fica entre o sistema operacional e as aplicações. O principal objetivo do middleware é facilitar o desenvolvimento de novas aplicações, abstraindo a complexidade do ambiente e gerenciando a comunicação entre estes dispositivos presentes no ambiente.

Neste trabalho foi desenvolvido um middleware, identificado como *uriMed*. O middleware *uriMed* foi desenvolvido utilizando a linguagem de programação Java e o sistema



gerenciador de banco de dados MySQL. Para a comunicação entre o Dispatcher e as aplicações foi utilizado Web Services. O *uriMed* é um middleware baseado em eventos para sistema distribuídos de larga escala, que utiliza o modelo de comunicação publish/subscribe (publicar/assinar) para comunicar entre os dispositivos presentes no ambiente.

2.1 O modelo publish/subscribe

O modelo de comunicação publish/subscribe é um modelo assíncrono, o que permite o desacoplamento dos diversos dispositivos presentes no ambiente, fazendo não existir a necessidade de alguns dispositivos saber da existência de outros.

Para disseminar as informações provenientes dos eventos existe um gerenciador, o sistema publish/subscribe, e dois tipos de clientes, os produtores de informações (*event publishers*) e os consumidores de informações (*event subscribers*). Os consumidores de informações fornecem uma descrição com o tipo de evento do qual querem ser notificados, e então conforme os produtores de informações publicam os eventos, estes são repassados pelo sistema publish/subscribe aos consumidores adequados. (SOUZA, 2008, p.24).

Esse modelo permite ao middleware *uriMed* a descentralização dos dispositivos existentes no ambiente, bem como possibilidade da utilização do sistema em larga escala. Provendo um meio robusto de comunicação entre um grande número de usuários.

2.1 Middleware *uriMed*

O middleware *uriMed* é formado de componentes de softwares e hardwares apresentados na Figura 1. Onde os sensores são componentes responsáveis por coletar dados do ambiente no qual estão incluídos, fornecendo parâmetros para possíveis atuadores. Neste projeto foi simulado dois tipos de sensores. Um sensor para capturar os batimentos cardíacos e outro para capturar a temperatura corporal dos pacientes.

Monitor é um dispositivo utilizado para demonstrar a aplicação de recebimento de informações providas das aplicações enquadradas nas regras pelo gerente de assinaturas. Neste projeto foi utilizado um smartphone Motorola Milestone 1 A853, equipado com um sistema Android 2.2, para executar uma aplicação que apresenta as informações de pacientes para o responsável (profissional de saúde).

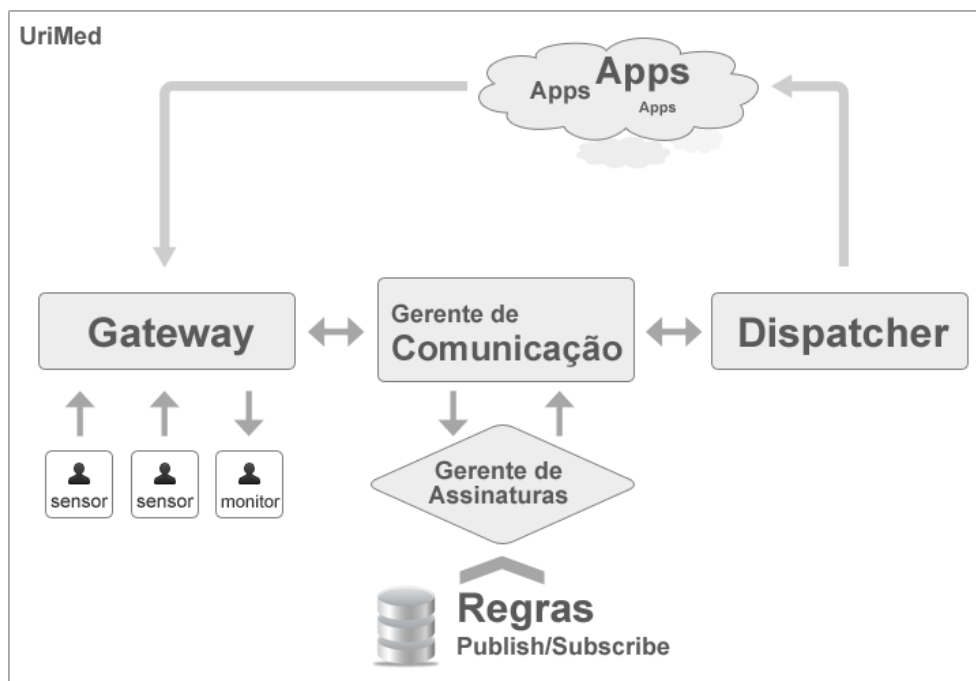


Figura 1: Visão abstrada do middleware uriMed

O Gateway é formado por um conjunto de hardwares responsáveis por fazer a ponte de ligação, para permitir a comunicação física entre aplicações, sensores e o middleware. O gateway recebe os dados dos sensores ou aplicações (via web services) e repassa para o Gerente de Comunicação, para o tratamento dos dados recebidos. Neste projeto foi utilizado um roteador wireless D-Link DIR-300 e um XBee Explorer USB da SparkFun. O Gateway fornece ao Gerente de Comunicação os dados serializados com a estrutura apresentada na tabela 1.

ID:28
DATETIME:2012-03-03 12:52
DATA:38.8
IP:192.168.0.101

Tabela 1: Estrutura dos dados enviados pelo Gateway

O gerente de comunicação faz o interfaceamento entre o Gateway e as aplicações. Utiliza o Gerente de Assinaturas para descobrir as aplicações que tem interesse de receber um determinado dado provido de um sensor, ou até mesmo de outra aplicação. Após descobrir os assinantes, compila as informações num formato XML (eXtensible Markup Language) para então ser repassadas ao Dispatcher. A estrutura do XML é apresentada na tabela 2:



```
<?xml version="1.0" encoding="utf-8"?>
<sensor>
  <id>28</id>
  <description>Sensor de Batimentos Cardíacos</description>
  <datetime>2012-03-03 12:52</datetime>
  <data>38.8</data>
  <ip>192.168.0.101</ip>
</sensor>
```

Tabela 2: Estrutura dos dados enviados pelo Gateway

O Dispatcher é invocado conforme o número de aplicações assinantes encontradas pelo Gerente de Assinaturas.

O Gerente de Assinaturas utiliza o sistema gerenciador de banco de dados MySQL para persistir as regras publish/subscribe, bem como para descobrir os assinantes de determinadas publicações. Basicamente existe para determinar os interessados em receber informações de um “produtor”. Seria o contrato de assinatura. As tabelas do banco de dados são descritas no ER da Figura 2.

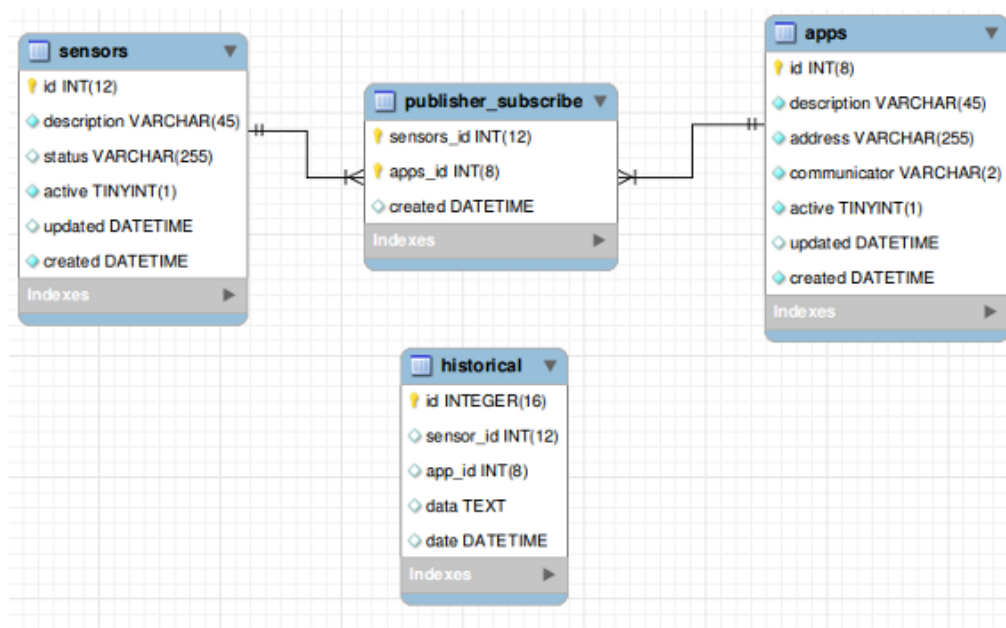


Figura 2: ER simplificado do Gerente de Assinaturas

A estrutura apresentada permite ao Gerenciador de Assinantes descobrir as aplicações interessadas em determinadas informações (sensores) muito facilmente. Bastando apenas, por exemplo, executar o seguinte SQL: `SELECT * FROM publisher_subscribe WHERE sensors_id = 28` (no software essa estrutura é mapeada por um framework *object-relational mapping*), para descobrir os interessados em determinado evento.



O Dispatcher é o responsável por comunicar com as aplicações resultantes das regras aplicadas pelo Gerente de Assinaturas. Essa comunicação é realizada via web service com as aplicações. O Dispatcher também tem a capacidade de comunicar com aplicações executadas em servidores remotos (por exemplo, em cidades diferentes da cidade do paciente), utilizando a Internet e o protocolo HTTP ou HTTPS para encapsular o SOAP e transmitir as informações formatadas com XML via Web services.

3 PROTÓTIPOS

Para validação da arquitetura foi desenvolvido o software para o gerenciamento de pacientes em um determinado local. O software foi desenvolvido para ser executado em Web Browsers. Foi desenvolvido utilizando a linguagem de programação PHP, o sistema Sistema de Gerenciamento de Banco de Dados MySQL e sua interface gráfica utiliza HTML, CSS e Javascript. O software foi executado num servidor Linux Ubuntu 10.04, com apache 2.2.22 e o MySQL 5.5.24.

Código	Foto	Paciente	Frequência Cardíaca	Temperatura Corporal	
32		PACIENTE 02	72 bpm	36.2 °C	Acompanhar
35		PACIENTE 05	118 bpm	41.5 °C	Acompanhar
36		PACIENTE 06	70 bpm	36.8 °C	Acompanhar
41		PACIENTE 11	82 bpm	37.1 °C	Acompanhar

URI - Universidade Regional Integrada do Alto Uruguai e das Missões

Figura 3: Tela principal. Apresenta informações dos pacientes.

A Figura 3 apresenta uma das telas do protótipo desenvolvido para a validação da arquitetura do middleware *uriMed*.



3.1 Resultados

Com o desenvolvimento de tais protótipos foi possível perceber claramente a facilidade que a arquitetura de software proporciona aos desenvolvedores de sistemas. Onde é possível criar aplicações descentralizadas, mas integradas com o restante do ambiente.

4 CONCLUSÃO

O middleware desenvolvido neste trabalho é capaz de prover uma infraestrutura para facilitar o desenvolvimento, manutenção e execução de aplicações voltadas para o gerenciamento do ambiente pervasivo na saúde.

Em relação ao resultado produzido, foi possível superar alguns desafios identificados na área da ubicomp, mais especificamente em medicina ubíqua. Com o middleware proposto será possível alcançar o desenvolvimento de aplicações mais consistentes e aptas a facilitar as atividades realizadas pelos profissionais da saúde, ou permitir uma gestão melhor por parte dos familiares dos pacientes.

Como trabalhos futuros ainda existem grandes necessidades de incremento da arquitetura apresentada neste trabalho. A primeira dela é de fazer uso de ontologias para descoberta de recursos presentes no ambiente ubíquo e pervasivo. Desenvolvimento de algoritmos que garantam entrega de dados com alto grau de relevância.

A SOFTWARE ARCHITECTURE FOR THE MANAGEMENT OF A UBIQUITOUS AND PERVASIVE ENVIRONMENTS IN HEALTH

Abstract: The objective of this work is to develop a software architecture aimed at the management of a pervasive and ubiquitous environment on health, and the development of a prototype which makes use of the same for the purpose of validating the results. This architecture has as main objective to facilitate and expedite the development of other applications with more specific objectives, such as the instant monitoring vital signs of patients. This goal is made possible with a view that technological advancement has enabled the miniaturization and enhancement of communication and computing power of mobile devices. The union of these devices with sensors capable of capturing information from the environment and vital signs of patients, it becomes possible to develop solutions to problems existing in the known health..

Keywords: Ubiquitous medicine. Ubicomp. Smart environments.



REFERÊNCIAS

INSTITUTO BRASILEIRO DE GEOGRAFIA ESTATÍSTICA. **Idosos no mundo**. Rio de Janeiro: IBGE, 2010a. Disponível em:
<http://ibge.gov.br/ibgeteen/datas/idoso/idoso_no_mundo.html>. Acessado em: 08 jan. 2012.

SAHA, D.; MUKHERJEE, A. Pervasive computing: a paradigm for the 21st century. **IEEE Computer**, v.36, n.3, p.26, 2003;

SOUZA, RODRIGO. Middlewares e Redes de Sensores: aspectos conceituais e arquiteturais. **Mobile & Pervasive Computing Group**, p. 24, jul. 2008.

WEISER, MARK. The Computer for the 21st Century. **Scientific American**, p. 94-104, set. 1991.