

Uma Plataforma de Middleware para Integração de Dispositivos e Desenvolvimento de Aplicações em *e-health*

Pedro Maia¹, Augusto Baffa², Everton Cavalcante^{1,3},
Flavia C. Delicato⁴, Thais Batista¹, Paulo F. Pires⁴

¹DIMAp, Universidade Federal do Rio Grande do Norte – Natal, Brasil

²DI, Pontifícia Universidade Católica do Rio de Janeiro – Rio de Janeiro, Brasil

³IRISA-UMR CNRS/Université de Bretagne-Sud – Vannes, França

⁴DCC/PPGI, Universidade Federal do Rio de Janeiro – Rio de Janeiro, Brasil

{pedropetrovitch, augbaffa, evertonranielly,
fdelicato, thaisbatista, paulo.f.pires}@gmail.com

Resumo. Na visão da Internet das Coisas (IoT), qualquer objeto físico ou virtual pode ser identificado, endereçável, controlado e monitorado via Internet. No cenário de *e-health*, tais objetos inteligentes podem ser sensores corporais que permitem um monitoramento contínuo de sinais vitais de pacientes. Diante da inerente heterogeneidade desses dispositivos, é necessário garantir interoperabilidade entre eles e facilitar o desenvolvimento de aplicações que façam uso dos seus dados. Nesse contexto, apresentamos a EcoHealth (Ecosistema de Dispositivos de Health Care), uma plataforma de middleware que integra sensores corporais heterogêneos para permitir o monitoramento remoto de pacientes e melhoria de diagnósticos médicos. Neste artigo são apresentadas a arquitetura da EcoHealth e uma avaliação de seu desempenho considerando uma aplicação de *e-health* desenvolvida como prova de conceito.

Abstract. In the Internet of Things (IoT) vision, any physical or virtual object can be identified, addressable, controlled, and monitored via Internet. In the *e-health* scenario, these smart objects can be body sensors allowing the continuous monitoring of vital signs of patients. Due to the inherent heterogeneity of such devices, it is necessary to enable interoperability among them and to facilitate the development of applications that make use of the provided data. In this context, this paper introduces EcoHealth (Ecosystem of Health Care Devices), a middleware platform that integrates heterogeneous body sensors for enabling the remote monitoring of patients and improving medical diagnosis. In this paper, we present the EcoHealth architecture and a performance evaluation with an *e-health* application developed as proof of concept.

1. Introdução

E-health pode ser definida como a prática de *health care* realizada com o auxílio de dispositivos eletrônicos e tecnologias de informação e comunicação (TICs), incluindo registros médicos e prescrições eletrônicas e monitoramento remoto de pacientes [Boric-Lubecke et al. 2014]. No contexto de *e-health*, pequenos sensores sem fio são colocados no corpo humano ou sob a roupa a fim de aumentar o conforto e a mobilidade do paciente que os utiliza, o qual é monitorado independentemente de sua localização e sem impacto em suas atividades cotidianas [Yuce 2013]. Com isso, é possível aumentar

a qualidade de serviços médicos oferecidos e ao mesmo tempo reduzir o seu custo, além de permitir o desenvolvimento de inúmeras aplicações, como, por exemplo, o monitoramento remoto, prevenção e tratamento de pacientes com doenças crônicas em ambientes de vida assistida (AAL – *Ambient Assisted Living*).

O contexto de *e-health* é apenas um dos domínios de aplicação que podem beneficiar-se da adoção do paradigma de Internet das Coisas (IoT – *Internet of Things*), como discutem alguns estudos recentes [Bui and Zorzi 2011, Bazzani et al. 2012, Sebestyen et al. 2014]. Na visão da IoT, qualquer dispositivo pode ser identificado, endereçado, controlado e monitorado via Internet [Atzori et al. 2010]. Nessa perspectiva, esses dispositivos inteligentes tornar-se-iam aptos a se comunicarem entre si e com outros recursos físicos e virtuais disponíveis na Internet, provendo assim diversas funcionalidades e informações acerca do ambiente no qual estão implantados. Dessa forma, em *e-health*, sensores corporais seriam objetos inteligentes capazes de proporcionar o monitoramento remoto de sinais vitais de pacientes a um baixo custo e em tempo real. Com isso, situações de emergência poderiam ser detectadas e adequadamente tratadas de maneira mais rápida, além de possibilitar diagnósticos médicos mais elaborados com base nas informações providas pelos dispositivos. Mais ainda, considerando que tais dispositivos podem comunicar-se uns com os outros, um dado sensor poderia fazer uso de informações providas por outro sensor para aumentar a qualidade das informações a serem providas a aplicações e usuários e a precisão de processos de tomada de decisão.

Apesar do imenso potencial trazido pela adoção da IoT em *e-health*, considerar sensores corporais como objetos inteligentes traz alguns desafios. O principal deles diz respeito à interoperabilidade entre tais objetos, visto que seria necessário lidar de forma idealmente transparente com uma miríade de dispositivos heterogêneos de vários fabricantes, cada um provendo uma interface de comunicação diferente e criando assim barreiras operacionais para o seu uso de forma sistêmica [Bui and Zorzi 2011]. Além disso, é necessário considerar uma grande quantidade de dados críticos a serem continuamente transmitidos pela rede, levantando questões de segurança e privacidade.

Para endereçar essas questões, plataformas de *middleware* têm surgido como uma solução promissora principalmente para prover interoperabilidade e gerenciar a crescente variedade de dispositivos associados a usuários e aplicações que fazem uso dos dados por eles providos [Bandyopadhyay et al. 2011]. Em *e-health*, uma plataforma de *middleware* tornaria possível integrar essa diversidade de sensores heterogêneos bem como disponibilizar os dados coletados via Internet. Além disso, através de interfaces de alto nível, seria possível acessar dispositivos de maneira transparente e permitir o desenvolvimento de uma gama de aplicações, reunindo e transformando informações provenientes dos dispositivos e provendo informações de valor agregado a usuários.

Nesse contexto, a **EcoHealth – Ecosistema Web de Dispositivos de Health Care** foi desenvolvida como uma plataforma de *middleware* para conectar médicos e pacientes utilizando sensores corporais acoplados e atuadores. O seu objetivo principal é integrar informações obtidas a partir de tais sensores heterogêneos para fins de monitoramento, processamento, visualização e armazenamento desses dados, bem como de notificação e atuação referentes a condições atuais de pacientes e seus sinais vitais. A EcoHealth caracteriza-se como uma customização, para o domínio de *e-health*, da EcoDiF (Ecosistema Web de Dispositivos Físicos) [Pires et al. 2014], uma plataforma Web genérica para IoT cujos principais objetivos são integrar dispositivos físicos heterogê-

neos e prover funcionalidades tais como visualização, processamento e armazenamento de dados oriundos desses dispositivos. A EcoHealth estende a EcoDiF em termos: (i) da implementação de um novo paradigma de comunicação que permite que a plataforma envie comandos de atuação para as plataformas de *hardware* que dão suporte aos sensores e atuadores; (ii) de um novo modelo relacional e dos elementos de código necessários para gerencia-lo, compreendendo aspectos tais como registros de pacientes, serviços de emergência, médicos, sensores e atuadores, e; (iii) de *drivers* específicos que dão suporte a sensores e atuadores relacionados ao domínio de *e-health*.

O projeto e a implementação da EcoHealth são essencialmente baseados em princípios REST (*REpresentational State Transfer*) [Fielding 2000, Webber et al. 2010] e em padrões e tecnologias Web atuais, e.g., HTTP e URIs. Uma visão geral da EcoHealth e sua arquitetura foram inicialmente introduzidas em um trabalho anterior [Maia et al. 2014]. Este artigo, por sua vez, apresenta uma versão mais avançada da arquitetura da EcoHealth e uma aplicação desenvolvida a título de prova de conceito para sua validação. Além disso, são mostrados resultados de experimentos computacionais realizados sobre um protótipo da plataforma a fim de avaliar o seu desempenho.

O resto deste artigo está organizado como segue. A Seção 2 apresenta a EcoHealth e sua arquitetura. A aplicação de *e-health* desenvolvida e os experimentos realizados são apresentados nas Seções 3 e 4. A Seção 5 discute trabalhos relacionados. A Seção 6 contém considerações finais e aponta direções para trabalhos futuros.

2. EcoHealth: Arquitetura

A EcoHealth é organizada em vários módulos que compõem sua arquitetura lógica, conforme ilustrado na Figura 1.

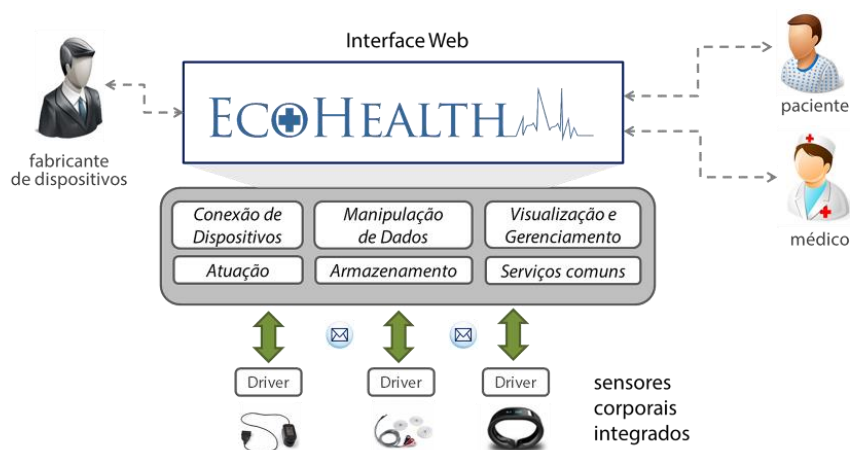


Figura 1. Arquitetura da EcoHealth.

O **Módulo de Conexão de Dispositivos** visa integrar dispositivos físicos (sensores corporais) à plataforma, utilizando, para isso, *drivers* customizados para cada tipo específico de dispositivo. Esses *drivers* desempenham um papel importante no tocante à integração de dispositivos de diferentes fabricantes uma vez que a heterogeneidade de tais dispositivos é abstraída para usuários e aplicações. Na EcoHealth há dois tipos de *drivers*: *ativos* e *passivos*. *Drivers* ativos obtêm informações dos dispositivos através da realização de requisições periódicas mesmo quando os valores das informações providas permanecem inalterados, enquanto *drivers* passivos aguardam por notificações pro-

venientes dos dispositivos em caso de modificações nas informações sensoriadas. Os *drivers* da EcoHealth são construídos fazendo uso de princípios REST e de padrões e protocolos Web, como HTTP e URIs. Assim, o protocolo HTTP não é usado apenas para transportar dados, mas como mecanismo padrão para interagir com os dispositivos integrados à plataforma por meio das operações definidas no protocolo (GET, POST, PUT e DELETE) e eliminando problemas de compatibilidade entre diferentes fabricantes e entre protocolos e formatos de dados [Pautasso et al. 2008]. A EcoHealth permite que a ela sejam associados diversos tipos de sensores corporais que podem ser acoplados às plataformas de *hardware* mais comumente utilizadas, tais como o Arduino¹. Além disso, a plataforma dispõe de *drivers* para suporte a sensores como sensor de temperatura corporal, fluxo de ar, eletrocardiograma, pulso e oxigênio no sangue, posicionamento de paciente (com base em acelerômetro), e resposta galvânica da pele (suor).

Após obter dados (*feeds*) provenientes dos dispositivos integrados, os *drivers* estruturam-nos usando EEML (*Extended Environments Markup Language*)², uma linguagem baseada em XML para descrição de dados obtidos em um contexto específico – neste caso, sinais vitais corporais. Em seguida, tais dados estruturados são enviados para a EcoHealth por meio de requisições HTTP PUT, a fim de que sejam registrados na plataforma pelo **Módulo de Manipulação de Dados**.

No contexto de *e-health*, permitir uma comunicação direcionada aos dispositivos físicos integrados é importante principalmente em casos nos quais atuadores são utilizados para emitir sons de alerta em situações de emergência ou para administrar ações terapêuticas, por exemplo [Chen et al. 2011]. Todavia, a EcoDiF, precursor da EcoHealth, não dispõe de um canal de comunicação no sentido plataforma–dispositivos, limitando-se apenas ao envio de dados a partir dos dispositivos (por meio dos *drivers*) para registra-los na plataforma. Dessa forma, a EcoHealth adiciona essa capacidade por meio do **Módulo de Atuação**, permitindo o envio de comandos da plataforma diretamente para os dispositivos a fim de realizar alterações em suas configurações e/ou fazer com que eles desempenhem tarefas específicas. Nessa nova perspectiva, a EcoHealth (que armazena os endereços dos dispositivos ou de *gateways*, para o caso de dispositivos sem capacidades de suporte às pilhas de comunicação) realiza requisições HTTP PUT para enviar comandos de atuação para os dispositivos, comandos esses que devem ser cadastrados no momento em que os dispositivos são conectados à plataforma. Além disso, diferentemente dos *drivers* disponíveis na EcoDiF, os *drivers* de integração da EcoHealth necessitam executar um servidor Web capaz de responder tanto a requisições HTTP GET, realizadas para obter *feeds* a partir dos dispositivos, quanto a requisições HTTP PUT, realizadas para enviar comandos de atuação para os dispositivos. Ao receber uma requisição HTTP PUT, o *driver* faz o processamento desta e envia efetivamente o comando para o atuador. Se nenhum erro ocorrer, o *driver* envia uma requisição HTTP em resposta à plataforma para que ela saiba a atuação funcionou de fato. É importante destacar que os comandos de atuação que podem ser realizados dependem das capacidades de atuação providas pelos respectivos dispositivos.

Em termos práticos, a atuação é realizada por meio requisições HTTP PUT via uma URI no formato `http://<ip_dispositivo>/<atuador>`. Dessa forma, por

¹ Arduino: <http://arduino.cc/>

² Extended Environments Markup Language (EEML): <http://www.eeml.org/>

exemplo, a URI `http://192.168.0.1/light` poderia ser utilizada para controlar as luzes de um ambiente. O controle pode ser ainda mais preciso dependendo dos tipos de atuadores disponíveis e da implementação do *driver*. Por exemplo, poder-se-ia ter uma URI `http://192.168.0.1/light/1` para indicar o envio de mensagens para a luz 1, ou ainda a URI `http://192.168.0.1/light/emergency`, para indicar o envio de mensagens somente para as luzes de emergência. A ideia é prover pelo menos uma URI para cada atuador disponível na plataforma de *hardware* de modo que se tal plataforma controla um dispositivo de som, a URI `http://192.168.0.1/sound` deveria ser provida pelo driver, possibilitando o seu uso pela EcoHealth. Além da URI, é necessário especificar no *payload* da requisição HTTP os parâmetros que indicam o tipo de ação a ser executada sobre o atuador especificada na URI. Por exemplo, o *payload* para ligar a luz especificada na URI `http://192.168.0.1/light` poderia ser codificado no formato JSON³ como `{"mode": "on"}`. Naturalmente, atuadores mais complexos podem requerer uma lista de parâmetros maior para funcionarem corretamente. Ainda como exemplo, o atuador da URI `http://192.168.0.1/sound` pode requerer que sejam especificados o modo de funcionamento (rádio, USB, Bluetooth, etc.), a faixa a ser escolhida (dependendo do modo) e o volume.

O **Módulo de Visualização e Gerenciamento** provê uma interface Web que permite aos usuários (médicos, pacientes e administradores de sistema) gerenciar informações sobre pacientes, sensores corporais acoplados a eles, registros médicos, notificações, informações sobre serviços de emergência, capacidades de atuação dos dispositivos, etc. Usando essa interface, médicos podem visualizar informações atualizadas dos pacientes coletadas por sensores corporais, assim como informações históricas armazenadas na plataforma. Além disso, os médicos podem criar *triggers*, que são mecanismos de notificação baseados em eventos que os informarão acerca de condições críticas nos sinais corporais que estão sendo medidos (*feeds*). Por exemplo, considerando um *feed* associado com a pressão arterial de um dado paciente, um médico pode criar um *trigger* para notifica-lo sempre que a medida da pressão arterial for superior a 140/90 mmHg. De maneira adicional, também é possível associar *triggers* a capacidades de atuação dos dispositivos. Dessa forma, um *trigger* pode ser configurado para que um alerta sonoro seja emitido no dispositivo sempre que o nível de glicose do paciente ultrapassar um limite estabelecido. Assim, dependendo das capacidades de atuação disponíveis nos dispositivos, seria possível configurar diversos pares *triggers*/atuação que seriam capazes de prover maior qualidade de vida para pacientes de risco.

O **Módulo de Armazenamento** engloba um banco de dados relacional e um sistema de arquivos para armazenar todas as informações usados na EcoHealth, e.g., registros médicos, informações históricas provenientes dos sensores corporais, notificações, informações relacionadas aos pacientes e informações sobre serviços de emergência. É importante ressaltar que esse módulo pode fazer uso de uma infraestrutura de nuvem a fim de armazenar dados relacionais e arquivos, conferindo robustez, confiabilidade, segurança, disponibilidade e escalabilidade à plataforma.

Por fim, o **Módulo de Serviços Comuns** consiste em serviços de infraestrutura providos pela plataforma, como segurança (autenticação, confidencialidade, e integridade), suporte a transações, notificações, etc. Considerando a natureza crítica dos dados

³ JavaScript Object Notation: <http://www.json.org/>

armazenados na plataforma, o esquema de autorização da plataforma garante que somente o médico autorizado pode acessar os registros de um determinado paciente e os dados providos pelos seus sensores corporais. É também essencial criptografar as informações transmitidas pelos sensores a fim de garantir confidencialidade e integridade dos dados.

Dadas as características de seus módulos, a EcoHealth prevê três perfis de usuários, a saber: (i) **fabricantes de dispositivos**, que desenvolvem *drivers* de dispositivos a fim de torna-los compatíveis com a API da EcoHealth; (ii) **médicos**, que continuamente monitoram os sinais vitais dos pacientes por meio da EcoHealth e usam as informações disponíveis na plataforma para prover melhores diagnósticos e resposta a emergências, e; (iii) **pacientes**, que proveem informações para a plataforma por meio de sensores corporais acoplados a eles.

3. Aplicação: Monitoramento Remoto de Pacientes

A fim de validar a EcoHealth, foi desenvolvida uma aplicação de *e-health* que tem como objetivo monitorar um paciente e indicar sua situação atual com base nos dados coletados por sensores nele acoplados. Para realizar medições de sinais vitais, foram utilizados três tipos de sensores, a saber: (i) um sensor de respiração, através de uma máscara fixada no nariz do paciente; (ii) um sensor de temperatura, fixado no braço do paciente, e; (iii) um sensor de eletrocardiograma (ECG) com eletrodos fixados no peito do paciente. Os dados coletados pelos sensores são enviados para a EcoHealth, que permite acompanhar as condições atuais do paciente com base em tais dados e também enviar notificações para o respectivo médico em caso de detecção de anormalidades nas medidas aferidas. A Tabela 1 apresenta os tipos de dados coletados pelos sensores corporais implantados no paciente e as condições tidas como normais e anormais para cada um.

Tabela 1. Tipos de dados coletados pelos sensores corporais e respectivas condições normais e anormais.

Sensor corporal	Dados sensoreados	Condições normais	Condição abaixo da normal	Condição acima da normal
Respiração	Ciclos de inalação/exalação em 60 segundos	Frequência eupneica (normal)	Frequência bradipneica	Frequência taquipneica (curta e acelerada)
Temperatura	Temperatura corporal	Normotermia (36-37° C)	Hipertermia (> 37° C)	Hipotermia (< 35,5° C)
ECG	Batimentos cardíacos	60-100 batimentos por minuto (bpm)	Taquicardia (> 100 bpm)	Bradicardia (< 60 bpm)

Os dados provenientes dos sensores fornecem importantes informações acerca das condições de saúde atuais do paciente. Por exemplo, o monitoramento da temperatura corporal é útil para detectar infecções e doenças, de modo que tal monitoramento constante permite melhor administrar o medicamento adequado e assim controlar o seu aumento. Por outro lado, outros sensores podem ser utilizados para confirmar alguns sintomas detectados pela temperatura ou frequência respiratória. Por exemplo, através do uso do sensor de ECG e da verificação da frequência de batimentos cardíacos, é possível identificar uma variedade de problemas de saúde que podem afetar um paciente, confirmando a sua condição clínica. A inerente heterogeneidade dos sensores que po-

dem ser usados dificulta a utilização sistêmica desses dados que, uma vez combinados, permitiriam um diagnóstico mais preciso por parte dos médicos. Nesse cenário, a EcoHealth pode ser utilizada para integrar os sensores corporais e monitorar variáveis biométricas em tempo de execução, bem como para gerar mensagens de alerta em caso de condições anormais. Para a aplicação em questão, os sensores de temperatura, respiração e ECG foram conectados a um *shield* Cooking Hacks e-Health⁴ acoplado a uma placa Arduino Uno⁵, combinação que permite a integração de uma variedade de sensores biométricos. Os batimentos cardíacos, a temperatura corporal e a frequência respiratória do indivíduo monitorado foram registrados na EcoHealth como *feeds*, de modo que um *driver* ativo desenvolvido para o Arduino Uno (i) coleta os dados medidos pelos sensores, (ii) estrutura-os no formato EEML, e (iii) os envia via requisições HTTP PUT para armazená-los na plataforma.

A Figura 2 mostra um trecho de representação EEML referente ao feed associado ao sensor de temperatura acoplado ao paciente. Na linguagem EEML, *feeds* são descritos através do elemento `environment` (linha 2), que representa os dados do ambiente que está sendo monitorado. O elemento `data` estrutura as informações aferidas pelo sensor e é composto por um conjunto de medições individuais (`datapoints`, linha 5), que informam o valor aferido pelo sensor em um determinado instante de tempo. A representação EEML exibida na Figura 2 informa que o valor de temperatura mais recente aferido pelo sensor (elemento `current_value`, linha 4) foi 36° C, o mesmo valor que foi coletado em medições anteriores (elementos `value`, linhas 6 a 10).

```
1: <eeml xmlns="http://www.eeml.org/xsd/0.5.1" version="0.5.1">
2:   <environment>
3:     <data id="61">
4:       <current_value at="2014-11-11T17:09:24.000-02:00">36</current_value>
5:       <datapoints>
6:         <value at="2014-11-11T16:44:28.000-02:00">36</value>
7:         <value at="2014-11-11T16:44:29.000-02:00">36</value>
8:         <value at="2014-11-11T16:49:59.000-02:00">36</value>
9:         <value at="2014-11-11T16:50:02.000-02:00">36</value>
10:        <value at="2014-11-11T16:50:03.000-02:00">36</value>
11:        ...
12:      </datapoints>
13:      <unit symbol="°C" type="Grau Celsius"/>
14:    </data>
15:  </environment>
16: </eeml>
```

Figura 2. Trecho de representação EEML referente a dados do sensor de temperatura.

A Figura 3 apresenta um diagrama de sequência que ilustra as operações desde a coleta de dados provenientes de dispositivos ao registro destes na EcoHealth. Como descrito anteriormente, o *driver* coleta os dados mensurados pelos sensores, estrutura-os em EEML, e os envia à EcoHealth, operações respectivamente representadas pelas mensagens `getDataFromSensor`, `generateEEML` e `sendEEML`. Uma vez recebidos na EcoHealth, o objeto `EEMLResource`, integrante do Módulo de Manipulação de Dispositivos

⁴ e-Health Sensor Platform v2.0 for Arduino and Raspberry Pi: <http://goo.gl/tdRDzr>

⁵ Arduino Uno: <http://arduino.cc/en/Main/arduinoBoardUno>

tivos, atualiza os dados referentes ao *environment* especificado (mensagem *updateEEML*), primeiramente validando a especificação EEML recebida (mensagem *validateEEML*) e em seguida atualizando os *datapoints* (*updateDatapoints*). Finalmente, o *EEMLResource* faz a verificação dos *triggers* previamente cadastrados que estão relacionados ao *feed* em questão (mensagem *checkTriggers*). Caso as condições dos *triggers* sejam avaliadas como verdadeiras, as notificações necessárias são enviadas (mensagem *sendNotifications*).

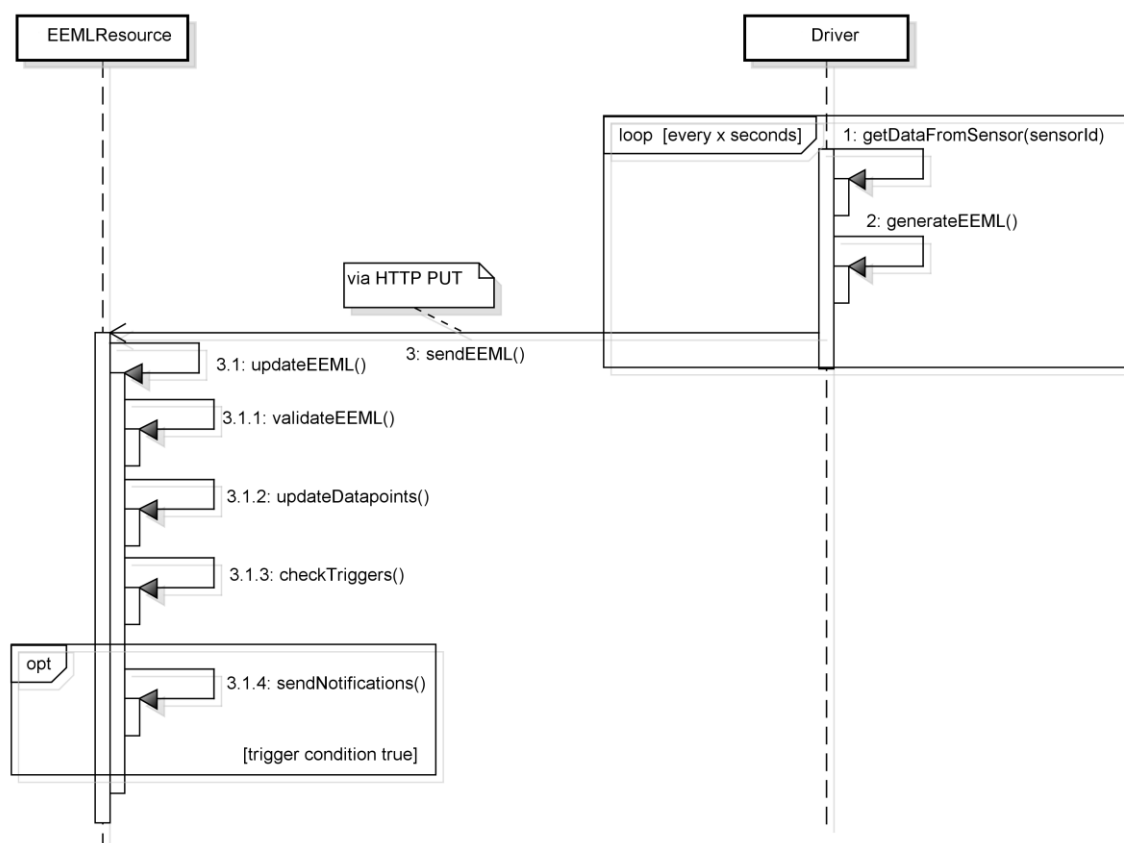


Figura 3. Diagrama de seqüência de operações desde a coleta de dados provenientes de dispositivos ao registro destes na EcoHealth.

Através da EcoHealth, é possível processar os dados obtidos pelos sensores e apresenta-los a médicos e outros profissionais de saúde através de séries históricas. Por exemplo, as variações de temperatura e respiração podem ser verificadas através dos gráficos exibidos e auxiliar na detecção de variações ao longo do dia e da noite. As Figuras 4 e 5 ilustram gráficos exibidos na interface Web da EcoHealth mostrando algumas medições coletadas pelos sensores de respiração e de temperatura, respectivamente. Além disso, como anteriormente mencionado, outra funcionalidade provida pela plataforma é disparar mensagens de alerta para médicos quando as variações nos dados representam condições críticas. Dessa forma, através da EcoHealth, poder-se-ia criar um *trigger* para o *feed* associado à temperatura a fim de enviar notificações caso esta ultrapasse 37° C ou esteja abaixo de 35,5° C, que representam condições anormais de temperatura corporal (cf. Tabela 1).

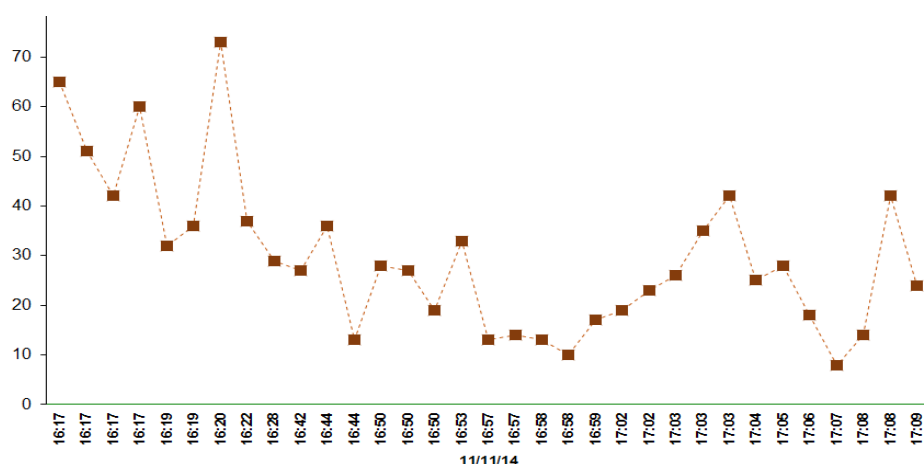


Figura 4. Histórico de monitoramento da respiração do paciente.

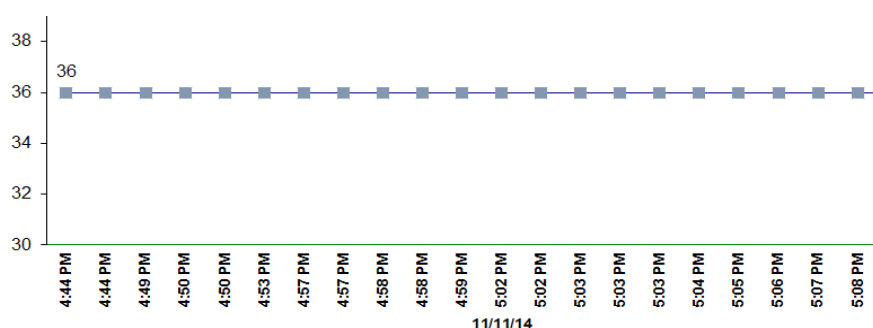


Figura 5. Histórico de monitoramento da temperatura corporal do paciente.

4. Avaliação

Em um cenário de *e-health*, pacientes são geralmente monitorados de maneira contínua, fazendo com que os dados coletados pelos sensores necessitem ser enviados com frequência à EcoHealth. Tendo isso em vista, esta seção apresenta uma avaliação de desempenho da plataforma com relação a sua capacidade de permitir uma atualização constante de *feeds* referentes aos dados provenientes de dispositivos integrados em um curto espaço de tempo. As Seções 4.1 e 4.2 detalham respectivamente o ambiente utilizado para a realização dos experimentos computacionais e a metodologia empregada, enquanto a Seção 4.3 discute os resultados obtidos.

4.1. Especificação do Ambiente de Teste

O ambiente utilizado nos experimentos computacionais realizados consistiu de um servidor Web no qual a EcoHealth foi implantada e de um computador simulando 300 dispositivos de diversos pacientes que enviam dados de temperatura corporal, frequência respiratória e batimentos cardíacos à plataforma. Os dispositivos simulados, implementados usando a linguagem Java e o componente Apache HTTP Components⁶, enviam as informações para a plataforma via requisições HTTP PUT por meio de um *driver* ativo, análogo ao descrito na Seção 3. As configurações utilizadas nos experimentos foram as ilustradas na Figura 6. É importante destacar que a plataforma, durante a avaliação, re-

⁶ HTTP Components: <http://hc.apache.org/>

cebia apenas as requisições enviadas pelo computador simulador de dispositivos, não sendo alvo, portanto, de requisições externas que poderiam interferir nos resultados a serem observados.

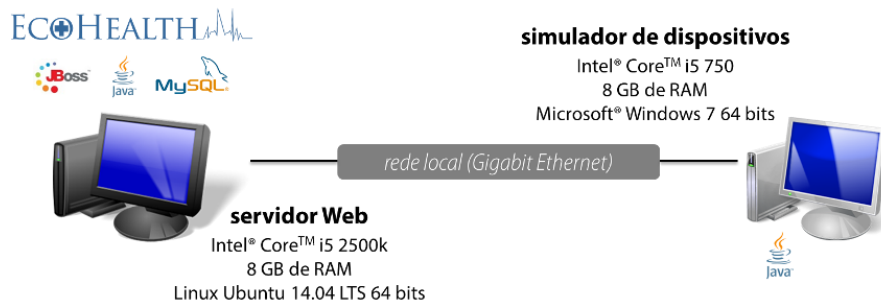


Figura 6. Ambiente utilizado nos experimentos computacionais realizados.

4.2. Metodologia

Após a implantação da EcoHealth no servidor Web, foram cadastrados 300 dispositivos e 300 *feeds*, sendo portanto cada *feed* associado a um dispositivo registrado na plataforma. É importante ressaltar que, na avaliação realizada, não se fez distinção acerca de que tipo específico de dado a ser enviado (i.e., se referente a temperatura corporal, frequência respiratória ou batimentos cardíacos), visto que a semântica dos *feeds* não promoveria alteração nos resultados e sim apenas no tamanho das mensagens enviadas.

Os experimentos realizados consistiram de cinco baterias de testes para avaliar o desempenho da plataforma. Em cada uma das baterias, 300 requisições (uma para cada *feeds*) foram enviadas 50 vezes, perfazendo assim um total de 15 mil requisições. Além disso, variou-se a frequência de envio de requisições de atualização (HTTP PUT) dos valores do *feeds* em 200, 500, 1000, 2000 e 5000 milissegundos. Como métrica, foi aferido o tempo despendido para processar cada requisição. É essencial que esses valores sejam mantidos em patamares considerados adequados para que a plataforma possa atender a todas as requisições em tempo satisfatório, de modo que mensagens de notificação possam ser enviadas em tempo hábil.

4.3. Resultados Obtidos

A Tabela 2 mostra os valores mínimo, médio e máximo, bem como o desvio padrão para o tempo de resposta em cada bateria de teste realizada, considerando diferentes intervalos entre requisições de atualização dos *feeds*. Pelos resultados obtidos, é possível perceber um decréscimo no tempo despendido para processamento das requisições à medida que o intervalo entre requisições aumenta.

Conforme mostra a Tabela 2, a partir do intervalo de 2000 ms, o servidor empregado é capaz de, em média, realizar o processamento de requisições consecutivas de um mesmo *feed* com tempo de resposta inferior ao intervalo entre requisições, de modo que a primeira requisição é completamente processada antes que a seguinte seja enviada para a plataforma. Essa é uma propriedade importante pois, em caso de emergência, a plataforma processará rapidamente a requisição e enviará as notificações necessárias aos interessados o mais rápido possível. Ainda analisando a bateria de testes de intervalo de 2000 ms, é possível notar que, no pior caso (tempo máximo), não é possível garantir a propriedade anteriormente mencionada uma vez que o tempo

necessário para processar uma requisição foi de 3960 ms, i.e., superior ao intervalo entre requisições. Com isso, nos casos em que o tempo de processamento de requisição é superior ao intervalo entre requisições, a plataforma ainda está ocupada processando a requisição corrente quando uma nova requisição (com dados mais atuais) é recebida. Essa requisição será então tratada com atraso e eventuais notificações geradas com base na análise de tal requisição seriam enviadas também com atraso, o que poderia eventualmente colocar em risco a vida de pacientes. Essa mesma análise pode ser estendida aos demais intervalos entre requisições.

Tabela 2. Valores mínimo, médio e máximo e desvio padrão para o tempo de resposta às requisições (em milissegundos).

Intervalo entre requisições (ms)	Mínimo	Médio	Máximo	Desvio padrão
200	51	2899,88	5487	438,76
500	54	2554,89	4862	458,29
1000	37	1991,26	4550	469,91
2000	35	1216,90	3960	651,75
5000	31	305,89	2347	354,36

O tempo mínimo de processamento de requisição mostrou-se consideravelmente baixo para todas as baterias de testes, o que mostra que, em condições ideais, o processamento da requisição pode ser bastante rápido. O fato de todas as baterias de testes apresentarem um tempo mínimo semelhante é devido à ociosidade que ocorre próximo ao fim de cada bateria de teste, i.e., o número de requisições a serem processadas vai chegando ao fim. Além disso, o valor consideravelmente elevado para o desvio padrão mostra que existe uma diferença significativa no tempo de processamento entre requisições distintas em virtude das variações de carga no servidor no decorrer das baterias de teste.

Por fim, é importante ressaltar que os resultados obtidos e, conseqüentemente, as discussões apresentadas, estão diretamente relacionados ao ambiente de teste empregado. Em um *hardware* de melhor qualidade, a EcoHealth poderia ser capaz de processar com qualidade um intervalo menor entre requisições e/ou um maior número de *feeds* sendo atualizados constantemente.

5. Trabalhos Relacionados

Trabalhos recentes na literatura têm discutido o potencial do paradigma de IoT para o aperfeiçoamento de aplicações *e-health* bem como os desafios e oportunidades de pesquisa nesse contexto [Bui and Zorzi 2011, Boric-Lubecke et al. 2014, Sebestyen et al. 2014]. Como anteriormente mencionado, cenários de *e-health* são tipicamente caracterizados por uma alta heterogeneidade com relação aos dispositivos a serem integrados, de modo que plataformas de *middleware* para IoT seriam uma solução promissora para permitir interoperabilidade entre tais dispositivos e aplicações [Bazzani et al. 2012]. Nessa perspectiva, esta seção apresenta brevemente algumas plataformas existentes baseadas em IoT para *e-health* que se constituem como alternativas à EcoHealth.

CardioNet [Sebestyen et al. 2014] é uma plataforma Web dedicada ao tratamento de pacientes com doenças cardiovasculares utilizando o *shield* Cooking Hacks e-Health Sensor em conjunção com a plataforma Arduino Uno (de maneira similar ao cenário da aplicação apresentada na Seção 3 deste artigo) para medição de parâmetros

tais como batimentos cardíacos, pressão sanguínea, temperatura e nível de oxigenação. Além do monitoramento remoto de pacientes, o portal Web da CardioNet oferece suporte a consultas *on-line* e administração de atividades de hospitais. Todavia, essa plataforma é restrita a dispositivos e funcionalidades relacionadas a dados cardiológicos.

REACTION [Ahlsén et al. 2012] é uma plataforma de *middleware* orientada a serviços que possui por objetivo principal facilitar o desenvolvimento de aplicações em *e-health*, em particular voltadas para pacientes com diabetes. A fim de endereçar a heterogeneidade existente em termos de dispositivos, protocolos, etc., a abordagem empregada pela REACTION é de utilizar serviços Web que podem ser desenvolvidos e implantados na plataforma. Tais serviços podem ser desenvolvidos de forma independente utilizando o Open Source Medical Device Connectivity Kit (DCK), um ambiente de desenvolvimento baseado em XML e serviços Web que permite que desenvolvedores construam serviços Web de maneira compatível com a API provida pela REACTION, a fim de permitir a sua implantação na plataforma. Uma vez disponíveis, tais serviços podem ser orquestrados a fim de compor diferentes aplicações, especificadas em alto nível utilizando *workflows* que especificam a ordem de execução desses serviços.

μ WoTOP (*micro Web of Things Open Platform*) [Corredor et al. 2014] é uma plataforma de *middleware* que objetiva facilitar a integração de sensores heterogêneos. Os principais elementos dessa plataforma são: (i) sensores biométricos (e.g., monitores cardíacos, acelerômetros, termômetros corporais, etc.) integrados à plataforma via *adapters*, similares aos *drivers* existentes na EcoHealth; (ii) *gateways* responsáveis por coletar dados dos sensores integrados e transmitir notificações urgentes aos interessados em caso de eventos críticos, e; (iii) aplicações clientes. Além de ser baseado em padrões Web bem estabelecidos, a μ WoTOP provê um ambiente para permitir reuso e compartilhamento de recursos disponíveis na plataforma, de modo que aplicações clientes podem fazer uso de dados de sensores por meio de uma interface RESTful.

Todas essas propostas apresentadas na literatura possuem diversos objetivos em comum com a EcoHealth, principalmente com relação à integração de sensores corporais heterogêneos e à provisão, em alto nível, de informações fornecidas por tais sensores a médicos e/ou aplicações. Além disso, é possível observar certa tendência no tocante ao uso de infraestruturas de nuvem visando promover escalabilidade, disponibilidade, desempenho e utilização de recursos sob demanda. Todavia, um aspecto importante que distingue a EcoHealth de tais plataformas diz respeito à existência de mecanismos genéricos que permitam o desenvolvimento de diferentes aplicações médicas que façam uso dos dados disponíveis na plataforma. No caso da μ WoTOP e da EcoHealth, tal desenvolvimento é facilitado por meio da provisão de uma interface RESTful que permite acessar os dados providos pelos dispositivos de forma transparente e ainda compatível com o paradigma de WoT. Finalmente, dentre as plataformas mencionadas nesta seção, apenas a μ WoTOP e a EcoHealth oferecem mecanismos de notificação com base nas medidas aferidas pelos sensores corporais integrados.

6. Conclusão

Este artigo apresentou a EcoHealth, uma plataforma de *middleware* para IoT baseada na Web para integrar médicos, pacientes e sensores corporais heterogêneos por eles utilizados. Os principais objetivos da EcoHealth são: (i) monitorar, através dos sensores corporais, diversas variáveis referentes à saúde do ambiente; (ii) possibilitar diagnósti-

cos através do controle, visualização, processamento e armazenamento de dados em tempo real, e; (iii) possibilitar a atuação em plataformas de *hardware* a fim de prover auxílio emergencial para pacientes em risco.

O projeto da EcoHealth é baseado em várias tecnologias Web bem estabelecidas (HTTP, REST e EEMML) com o intuito de padronizar e simplificar o desenvolvimento de aplicações no contexto de IoT, minimizando assim problemas de compatibilidade e de interoperabilidade entre fabricantes, protocolos proprietários e formatos de dados. Neste artigo, foi apresentada a arquitetura lógica do EcoHealth e uma aplicação no domínio de *e-health* no qual a plataforma pode ser útil. Por fim, também foi descrita uma avaliação quantitativa do desempenho da plataforma, que foi submetida a uma elevada quantidade de requisições de atualizações de *feeds* em um curto espaço de tempo. A avaliação mostrou que a plataforma pode suportar uma quantidade considerável de dispositivos físicos (e.g., sensores corporais) enviando dados com frequências adequadas para o monitoramento de sinais vitais.

Em trabalhos futuros, a EcoHealth será validada por meio de estudos de caso em diversos cenários reais nos quais pacientes e médicos estarão usando a plataforma. Além disso, pretende-se prover suporte ao desenvolvimento de diferentes aplicações sobre a EcoHealth a fim de permitir que elas tenham acesso às informações disponíveis na plataforma. Tais aplicações podem ser desenvolvidas como *Web mashups* [Guinard and Trifa 2009], uma tecnologia que tem sido considerada adequada para o paradigma de IoT e que consiste de aplicações *ad-hoc* criadas pela composição de diferentes tipos de informações provenientes de diferentes fontes.

Agradecimentos

Esse trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pelo Instituto Nacional de Ciência e Tecnologia para Engenharia de Software (INES)⁷. Flavia C. Delicato é parcialmente financiada pelo CNPq através dos processos 310661/2012-9, 457783/2014-1 e 307378/2014-4. Thais Batista é parcialmente financiada pelo CNPq através dos processos 573964/2008-4 (INES) e 308725/2013-1. Flavia C. Delicato, Thais Batista e Paulo F. Pires são bolsistas de produtividade do CNPq.

Referências

- Ahlsén, M. et al. (2012) “Service-oriented middleware architecture for mobile personal health monitoring”. In: Nikita, K. S. et al., eds. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 83. Germany: Springer-Verlag Berlin Heidelberg. pp. 305–312.
- Atzori, L. et al. (2010) “The Internet of Things: A survey”, *Computer Networks*, vol. 54, no. 15, pp. 2787–2805.
- Bandyopadhyay, S. et al. (2011) “Role of middleware for the Internet of Things: A study”, *International Journal of Computer Science & Engineering Survey*, vol. 2, no. 3, pp. 94–105.

⁷ <http://www.ines.org.br/>

- Bazzani, M. et al. (2012) “Enabling the IoT paradigm in e-health solutions through the VIRTUS middleware”, Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, IEEE Computer Society, USA, pp. 1954–1959.
- Boric-Lubecke, O et al. (2014) “E-healthcare: Remote monitoring, privacy, and security”, Proceedings of the 2014 IEEE MTT-S International Microwave Symposium, USA, pp. 1–3.
- Bui, N., Zorzi, M. (2011) “Health care applications: A solution based on the Internet of Things”, Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, ACM, USA.
- Chen, I. M. et al. (2011) “Personalized biomedical devices & systems for healthcare applications”, Frontiers of Mechanical Engineering, vol. 6, no. 1, pp. 3–12.
- Corredor, I. et al. (2014) “A lightweight Web of Things open platform to facilitate context data management and personalized healthcare services creation”, International Journal of Environmental Research and Public Health, vol. 11, pp. 4676–4713.
- Delicato, F. C. et al. (2013) *Middleware solutions for the Internet of Things*, Springer London, United Kingdom.
- Duquennoy, S. et al. (2009) “The Web of Things: Interconnecting devices with high usability and performance”, Proceedings of the 2009 International Conference on Embedded Software and Systems, IEEE, USA, pp. 323–330.
- Fielding, R. (2000) *Architectural styles and the design of network-based software architectures*. PhD Thesis, University of California-Irvine, USA.
- Guinard, D., Trifa, V. (2009) “Towards the Web of Things: Web mashups for embedded devices”, Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web.
- Maia, P. et al. (2014) “A Web platform for interconnecting body sensors and improving health care”, *Procedia Computer Science*, vol. 40, pp. 135–142.
- Pautasso, C. et al. (2008) “RESTful Web services vs. “big” Web services: Making the right architectural decision”, Proceedings of the 17th International Conference on World Wide Web, ACM, USA, pp. 805–814.
- Pires, P. F. et al. (2014) “A platform for integrating physical devices in the Internet of Things”, Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing, IEEE, USA, pp. 234–241.
- Sebestyen, G. et al. (2014) “eHealth solutions in the context of Internet of Things”, Proceedings of the 2014 International Conference on Automation, Quality and Testing, Robotics, IEEE, Romania, pp. 1–6.
- Webber, J. et al. (2010) *REST in practice: Hypermedia and systems architecture*. USA: O’Reilly Media.
- Yuce, M. R. (2013) “Recent wireless body sensors: Design and implementation”, Proceedings of the 2013 IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications, IEEE, USA, pp. 1–3.