

# Instrumentação na IoT: uma Abordagem Autônoma e Distribuída

Lucas M. S. Xavier<sup>\*†</sup>, Huberto Kaiser Filho<sup>\*</sup>, Patrícia T. Davet<sup>\*</sup>, Eduardo S. Abreu<sup>\*</sup>, Wemerson D. Parreira<sup>†</sup> e Adenauer C. Yamin<sup>\*†</sup>

**Resumo**—As facilidades tipicamente providas por infraestruturas da IoT, como acesso remoto e configuração automatizada, podem ser apropriadas para responder às demandas dos sistemas de instrumentação modernos. Este trabalho apresenta a concepção de um instrumento eletrônico desenvolvido de forma integrada a um middleware para computação ubíqua, buscando explorar a convergência tecnológica entre a instrumentação contemporânea e as plataformas para IoT. Especificamente, foram empregadas, no estudo, as capacidades de ciência de contexto do middleware EXEHDA. O instrumento desenvolvido, denominado ubiMeter, teve suas funcionalidades avaliadas em dois cenários de teste distintos, nos quais os resultados obtidos se mostraram promissores.

**Palavras-Chave**—Instrumentação remota, Internet das Coisas, middleware, Instrumentação ubíqua.

**Abstract**—IoT infrastructures typically provide several features that can be useful in the development and deployment of modern instrumentation systems; examples include remote access and automated configuration. In this paper, we explore the convergence between instrumentation and IoT platforms through the development of a remote instrument, whose design prioritizes integration with an ubiquitous computing middleware. Specifically, the instrument, which is called ubiMeter, uses the context awareness capabilities of the EXEHDA middleware. Its features were evaluated in two different case studies, with promising results.

**Keywords**—Remote instrumentation, Internet of Things, middleware, Ubiquitous instrumentation

## I. INTRODUÇÃO

A Internet das Coisas (*Internet of Things* ou IoT), enquanto paradigma para a evolução da Internet nos próximos anos, prevê que cada vez mais os objetos cotidianos deverão ganhar capacidades de processamento, comunicação e sensoriamento. Essa visão abre espaço para que esses objetos sejam incorporados às redes de computadores e passem a interagir de forma autônoma, promovendo uma ampla integração do mundo físico ao digital [1]. Dentro dessa perspectiva, as aplicações se tornam capazes de empregar a infraestrutura da IoT para realizar a computação de forma amplamente distribuída e, tanto quanto possível, provendo interfaces homem-máquina proativas, adquirindo um caráter efetivamente ubíquo. Para a plena implementação desse cenário, contudo, é necessária a superação de uma série de desafios relacionados a aspectos como escalabilidade e interoperabilidade [2].

<sup>\*</sup>Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas-RS, Brasil, E-mails: {lmdsxavier,hkaiser,ptdavet,eabreu}@inf.ufpel.edu.br

<sup>†</sup>Centro Politécnico, Universidade Católica de Pelotas, Pelotas-RS, Brasil, E-mail: {adenauer.yamin, wemerson.parreira}@ucpel.edu.br

Em ambientes IoT, existe a necessidade de coletar dados continuamente gerados por um conjunto heterogêneo de sensores e objetos, bem como de disponibilizar essas informações de forma remota para diversas aplicações e usuários [3]. Fica caracterizada, assim, uma forte convergência entre os requisitos abarcados pelas plataformas IoT e as demandas dos sistemas de instrumentação modernos, que cada vez mais compartilham dos mesmos desafios tecnológicos e de pesquisa. Essas necessidades se verificam tanto no escopo industrial [4], [5] quanto no âmbito de laboratórios para ensino e pesquisa [6].

Assim, a proposta central desse trabalho é o emprego de um *middleware* IoT para o desenvolvimento e prototipação de um instrumento de medição que atenda a algumas das principais premissas da instrumentação ubíqua. A abordagem adotada considera que os instrumentos eletrônicos modernos possuem capacidades computacionais crescentes, dedicadas a funcionalidades como autocalibração e visualização das medições. Essas capacidades podem ser estendidas para disponibilizar uma interface de serviços por meio de protocolos típicos de ambientes IoT [7].

O *middleware* IoT utilizado neste estudo foi o EXEHDA. Este *middleware* provê uma arquitetura de software baseada em serviços que visa criar e gerenciar um ambiente ubíquo, promovendo a execução de aplicações IoT sob este ambiente. Seu foco é permitir que as aplicações adquiram um caráter ciente de contexto, tornando-se reativas e adaptativas em resposta a variações nos seus contextos de interesse. O instrumento desenvolvido foi integrado ao Subsistema de Reconhecimento e Adaptação ao Contexto do EXEHDA.

O restante do artigo está estruturado como segue. A Seção II apresenta uma visão geral da arquitetura do EXEHDA, com ênfase no Subsistema de Reconhecimento e Adaptação ao Contexto. A concepção do instrumento IoT é discutida na Seção III. A Seção IV aborda o desenvolvimento de um protótipo funcional seguindo a arquitetura proposta, bem como os estudos utilizados para avaliação do instrumento. Na Seção V, são resumidos alguns dos principais trabalhos relacionados. Finalmente, a Seção VI traz uma série de considerações finais a respeito do trabalho.

## II. MIDDLEWARE EXEHDA

O EXEHDA é um *middleware* voltado para aplicações ubíquas amplamente distribuídas, móveis e cientes de contexto. Para isso, adota uma organização baseada em serviços dinamicamente carregados por um núcleo mínimo conforme a demanda. As principais funcionalidades providas pelo *middleware* estão divididas nos subsistemas de acesso ubíquo,

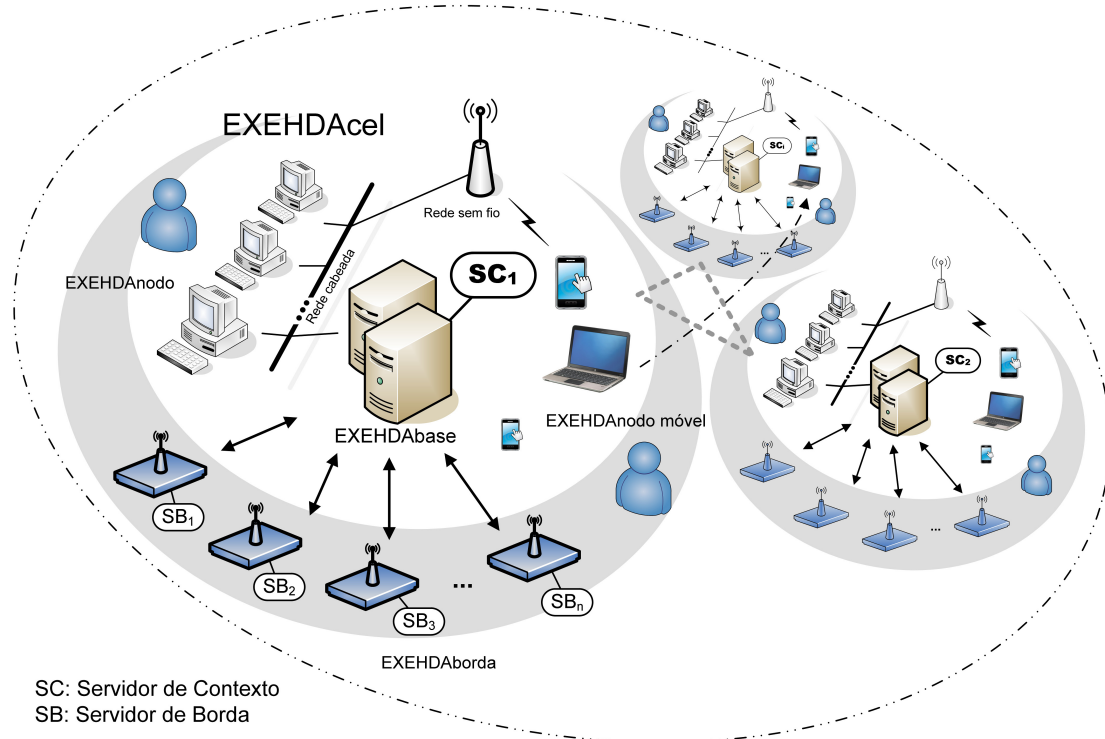


Fig. 1. Organização celular do ambiente ubíquo provido pelo EXEHDA

comunicação, execução distribuída e adaptação e reconhecimento do contexto [8].

O ambiente ubíquo provido pelo EXEHDA é formado por células que agrupam os recursos disponíveis na infraestrutura física. Esses recursos são mapeados para três abstrações básicas [9]: (i) EXEHDAAnodos, que são dispositivos computacionais responsáveis pela execução das aplicações; (ii) EXEHDAbases, que são o elemento central de uma célula, fornecendo serviços básicos e de referência; e (iii) EXEHDAbordas, que são elementos responsáveis por interoperar os serviços do *middleware* com os dispositivos IoT. A organização celular dos recursos, na visão do *middleware*, é mostrada na Figura 1.

Dentre os diversos serviços do EXEHDA, o Subsistema de Reconhecimento e Adaptação ao Contexto (SRAC) é o módulo responsável pela aquisição, pelo armazenamento e pelo processamento de informações contextuais, além do suporte a procedimentos de atuação. Suas responsabilidades estão divididas em dois tipos de servidores [10]. O Servidor de Borda, executado em um nodo tipo EXEHDAborda, se destina a gerenciar a interação com o meio físico através de sensores e atuadores, bem como executar o primeiro estágio do processamento do contexto. O Servidor de Contexto, por sua vez, é instanciado em uma EXEHDAbase e atua no armazenamento e no segundo estágio de processamento das informações contextuais, integrando dados históricos e dados provenientes de diferentes Servidores de Borda.

A arquitetura do SRAC prevê, ainda, o uso de *gateways* como dispositivos dedicados à interoperação entre os Servidores de Borda e nodos sensores e atuadores heterogêneos, potencialmente não programáveis. Os *gateways* agem no sen-

tido de interligar esses dispositivos IoT ao SRAC, utilizando, para isso, protocolos como CoAP, MQTT e HTTP sobre a rede TCP/IP. Esses equipamentos têm hardware e capacidades dedicadas para tratar tecnologias específicas, fazendo conversão de protocolos e gerenciamento de dispositivos que não possam ser tratados diretamente por Servidores de Borda.

### III. UBIMETER: VISÃO GERAL E ARQUITETURA

Em ambientes IoT típicos, há um elevado volume de comunicação entre os diversos objetos inteligentes, que se apresentam em grande número e sob formas variadas, diferindo significativamente em suas capacidades computacionais. Nesse cenário, surgem desafios relacionados a aspectos como interoperabilidade, escalabilidade e complexidade de configuração. Assim, o desenvolvimento de novos dispositivos IoT deve priorizar, como premissas (entre outros itens): (i) a operação autônoma, provendo mecanismos para coleta e processamento automatizados dos dados; (ii) o uso de protocolos bem difundidos e padronizados para comunicação com outros objetos inteligentes; e (iii) a simplicidade de configuração, considerando a dinamicidade das infraestruturas de rede.

A solução caracterizada neste artigo propõe a implementação de um instrumento eletrônico que, além de disponibilizar as medições por uma interface de rede, seja capaz de atender às premissas enumeradas. A arquitetura proposta para o ubiMeter pode ser dividida, em um primeiro nível de abstração, em dois grandes blocos: um hardware de aquisição, que provê uma interface de acesso às medições por meio de canais de comunicação sem fio; e uma camada de software, executada em outro dispositivo, que disponibiliza as informações providas para usuários e aplicações. Uma

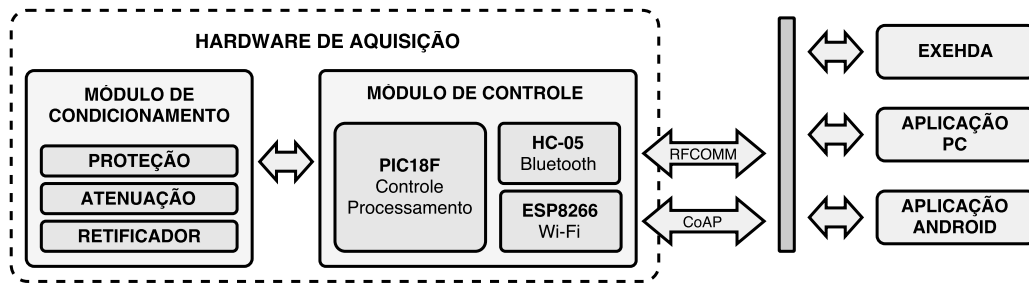


Fig. 2. Visão geral da arquitetura do ubiMeter

visão geral para a arquitetura do instrumento é mostrada na Figura 2.

O software, nessa perspectiva, pode ser representado tanto por um *middleware* IoT quanto por uma aplicação específica que seja capaz de interoperar com o hardware de aquisição. A comunicação se dá por meio de uma interface de serviços, disponibilizada pelo hardware de aquisição e implementada sobre os protocolos de comunicação previstos para o instrumento. Os serviços podem ser agrupados em duas categorias: funcionalidades de configuração do instrumento e procedimentos para aquisição das grandezas medidas.

O hardware de aquisição, por sua vez, pode ser subdividido em dois blocos. O Módulo de Controle compreende um circuito microprocessado que digitaliza as grandezas medidas e implementa a interface com o software de virtualização. O Módulo de Condicionamento, representado por um circuito analógico, realiza a adequação dos sinais de entrada às características elétricas exigidas pelo microcontrolador do Módulo de Controle.

Quando utilizado de forma integrada ao EXEHDA, o instrumento é modelado como um *gateway*. Nesse caso, o ubiMeter disponibiliza uma interface para configuração e leitura das medições empregando o protocolo CoAP. A comunicação entre o instrumento e o *middleware* se dá de forma assíncrona, com requisições realizadas diretamente pelas aplicações ou disparadas pela avaliação de uma ou mais regras pelo processador de contexto do Servidor de Borda.

#### IV. UBIMETER: FUNCIONALIDADES E AVALIAÇÃO

O protótipo desenvolvido para este trabalho foi um voltímetro para medição de tensões contínuas, baseado na premissa de permitir a operação autônoma ao longo de uma faixa de medição ampla. Para isso, foi utilizado um esquema de seleção automática de faixa de operação, no qual o Módulo de Controle regula a atenuação imposta ao sinal de entrada por meio de um conjunto de relés presente no Módulo de Condicionamento. A opção por um medidor de tensão objetiva simplificar a etapa de validação do ubiMeter, sem prejuízo da generalidade da arquitetura.

O Módulo de Controle teve suas funcionalidades implementadas utilizando um microcontrolador PIC18F4550, um módulo Wi-Fi ESP-01 e um módulo Bluetooth HC-05. O PIC, que possui conversor A/D integrado de 10 bits, assume as funcionalidades de aquisição e controle, que são realizadas em um esquema de multitarefa cooperativo. Os outros dois

módulos, por sua vez, realizam a comunicação com o software de virtualização; o módulo Wi-Fi disponibiliza a interface CoAP para coleta de dados; e o módulo Bluetooth permite, além da leitura das medições, a configuração da conectividade Wi-Fi do instrumento. A comunicação Bluetooth é baseada no protocolo RFCOMM, com um protocolo de camada de aplicação próprio.

O Módulo de Condicionamento foi implementado como um circuito analógico linear que opera com entrada diferencial de até 350 V e produz como saída um sinal proporcional na faixa de 0 V a 5 V. Para isso, o circuito possui quatro faixas de operação (com fundo de escala de 2 V, 20 V, 200 V e 400 V), que são selecionadas automaticamente pelo Módulo de Controle por meio de um sistema de *autorange*. O nível de atenuação é controlado por um conjunto de quatro relés do tipo *reed* que alteram a relação entre os resistores de um divisor resistivo, bem como o ganho de um estágio amplificador.

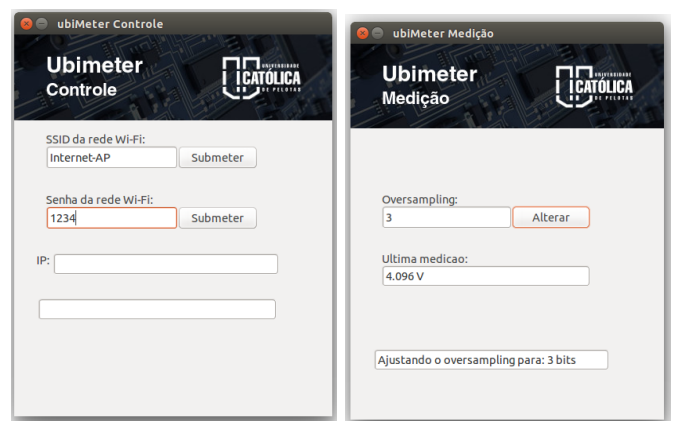


Fig. 3. Aplicações desktop para visualização dos dados e gerenciamento das configurações

Para demonstrar a interoperação entre o software de virtualização e o hardware de aquisição, foram desenvolvidas duas aplicações gráficas, mostradas na Figura 3, para comunicação entre o ubiMeter e um PC. Uma das aplicações utiliza a interface Bluetooth para oferecer funcionalidades básicas de configuração da conectividade do instrumento. A outra, baseada na interface CoAP, permite uma funcionalidade mínima de visualização das medições e configuração das opções de aquisição e pré-processamento dos dados. Ambas as aplicações foram desenvolvidas em Python e testadas em sistemas baseados em Linux.

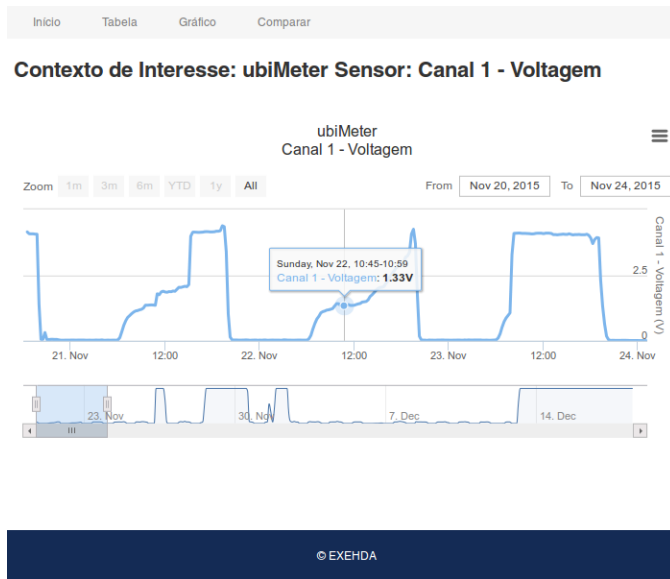


Fig. 4. Aplicação web do EXEHDA para visualização das medições obtidas com o ubiMeter

No caso de instalação do ubiMeter em um ambiente gerenciado pelo EXEHDA, foi utilizada, para visualização dos dados, uma aplicação web baseada na tecnologia de software trabalhada no grupo de pesquisa [11], [12]. A aplicação é mostrada na Figura 4. Entre as principais funcionalidades utilizadas no trabalho, pode-se enumerar a geração de gráficos e tabelas baseados nas medidas realizadas. Outras funcionalidades da aplicação incluem o envio de alertas por SMS ou e-mail, no caso de medições fora dos parâmetros esperados, e a comparação entre valores de múltiplos sensores.

Para avaliação do trabalho desenvolvido, foram concebidos dois cenários de teste; o primeiro, focado na integração do ubiMeter com o *middleware* EXEHDA; e o segundo, voltado para a verificação de requisitos não funcionais ligados à eletrônica do hardware de aquisição.

No primeiro cenário, o instrumento desenvolvido foi instalado em um ambiente ubíquo gerenciado pelo EXEHDA. Ao longo de três dias, foi monitorada a tensão sobre um fotoresistor localizado em um ponto fixo de uma sala. As medições foram periodicamente coletadas por um Servidor de Borda instanciado em uma Raspberry Pi e armazenadas em um Servidor de Contexto instanciado em uma máquina remota desktop. A Figura 4 mostra dados extraídos do período de testes.

No segundo cenário, o protótipo foi submetido a um conjunto de testes para verificar os erros introduzidos pelo hardware de aquisição. Um conjunto de doze tensões de referência, abrangendo duas das quatro faixas de operação do Módulo de Condicionamento, foi escolhido de forma que a maior entrada (para cada faixa) não disparasse uma mudança no algoritmo de *autorange*. As referências foram geradas por uma fonte de tensão contínua ajustável e aferidas com auxílio de um multímetro ET-1002 da Minipa, que possui faixas de operação divididas de forma similar ao protótipo e precisão de  $\pm 0,8\%$  mais  $0,25\%$  do fundo de escala.

Para cada tensão de referência, foram realizadas trinta

TABELA I  
RESULTADOS DOS TESTES PARA A FAIXA DE 20 V

Referência	ubiMeter			
	Média (V)	Erro médio	Erro máximo	Desvio Padrão
3,2 V	3,20	0,01%	1,86%	1,15%
6,4 V	6,42	0,27%	0,95%	0,56%
9,6 V	9,60	0,04%	0,61%	0,40%
12,8 V	12,82	0,14%	0,56%	0,31%
16 V	16,01	0,06%	0,50%	0,23%
19,2 V	19,23	0,14%	0,60%	0,23%

TABELA II  
RESULTADOS DOS TESTES PARA A FAIXA DE 2 V

Referência	ubiMeter			
	Média (V)	Erro médio	Erro máximo	Desvio Padrão
320 mV	0,322	0,49%	16,74%	12,18%
640 mV	0,638	0,39%	8,73%	5,75%
960 mV	0,962	0,19%	5,61%	4,07%
1,28 V	1,278	0,19%	4,21%	2,85%
1,6 V	1,590	0,63%	3,31%	2,17%
1,92 V	1,926	0,30%	3,13%	1,91%

medições com o ubiMeter. O Módulo de Controle foi configurado para realizar uma sobreamostragem de 64 amostras, estendendo a resolução para 13 bits. Nas tabelas I e II são mostradas a média das trinta medições, bem como o erro máximo, o erro médio e o desvio padrão.

Com o protótipo operando na faixa de 20 V, nenhuma das medições apresentou erro superior a  $0,8\%$  da medição mais  $0,25\%$  do fundo de escala (a precisão do próprio multímetro usado como referência). Assim, a precisão obtida foi similar à de um instrumento típico de mercado. Na faixa de 2 V, a inclusão de um estágio amplificador no caminho do sinal acentuou a flutuação dos valores mensurados, amplificando erros introduzidos nos estágios anteriores do Módulo de Condicionamento. Em função disso, as medições apresentaram grande variabilidade, com desvio padrão superior a  $2\%$  e erro absoluto de até 60 mV. O amplificador mencionado é empregado apenas na faixa de 2 V.

## V. TRABALHOS RELACIONADOS

Existe uma ampla literatura relacionada ao desenvolvimento de plataformas voltadas para a aquisição de dados produzidos por redes de sensores sem fio, tanto no escopo da IoT como em ambientes ubíquos [2], [3], [13]. Contudo, a maioria dos trabalhos não enfatiza as necessidades específicas dos cenários de instrumentação e medição tradicionais, dificultando a comparação direta.

A exceção é o trabalho [14], que propõe uma infraestrutura IoT para aplicações de medição e aquisição de dados distribuídas e executadas de forma concorrente. A plataforma visa, ainda, facilitar a integração de diferentes tipos de sensores e protocolos de comunicação. Assim como no presente trabalho, *gateways* são usados para prover interoperabilidade e realizar a interface com dispositivos de baixo poder computacional. Contudo, a arquitetura proposta não é organizada de forma hierárquica, interpondo um *IoT Server* como única abstração entre as redes de sensores e as aplicações.

Entre os outros trabalhos, que não possuem foco específico em instrumentação, é possível pontuar alguns exemplos de plataformas que poderiam ser usadas para aplicações de medição distribuída.

O projeto LinkSmart [15] é um *middleware* para inteligência ambiental com capacidades de ciência de contexto. A plataforma é baseada em uma arquitetura distribuída, orientada a serviços e com componentes estruturados em três camadas: (i) camada de rede, responsável pela comunicação com os dispositivos; (ii) camada de serviço, responsável pelo gerenciamento de eventos, dispositivos e escalonamento de recursos; e (iii) camada semântica. Um de seus diferenciais é a possibilidade de contemplar uma descrição semântica dos dispositivos através do uso de ontologias, o que proporciona interoperabilidade em nível sintático e semântico.

O *middleware* Servilla [16] é voltado para redes de sensores heterogêneas. As aplicações são estruturadas na forma de tarefas independentes de plataforma e executadas em máquinas virtuais nos nodos disponíveis. Códigos dependentes de plataforma são abstraídos como serviços, que são dinamicamente vinculados às tarefas em tempo de execução. A plataforma usa uma linguagem própria para descrição de serviços e tarefas.

Xively [17] é uma solução comercial de PaaS que provê serviços de *middleware* para produtos e aplicações IoT. Além de funcionalidades de armazenamento e processamento de dados, a plataforma permite a descoberta de dispositivos por meio de um diretório de serviços, bem como a implantação de aplicações amplamente escaláveis. Embora vários formatos de dados sejam suportados, os dados de entrada não são homogeneizados, e o suporte a interoperabilidade é parcial [13].

## VI. CONSIDERAÇÕES FINAIS

De forma geral, as tecnologias estratégicas para os sistemas de instrumentação modernos são as mesmas que permitem a materialização do cenário da IoT. Este trabalho busca explorar essa convergência, apropriando-se de uma infraestrutura de IoT para conferir um caráter de ubiquidade a um instrumento de medição remoto. Levando isso em consideração, o desenvolvimento do ubiMeter foi realizado de forma integrada a um *middleware* e contemplando as principais premissas da instrumentação ubíqua.

As funcionalidades do ubiMeter, ao longo de sua avaliação, mostraram resultados promissores nesse sentido, em duas situações distintas: (i) quando do seu uso enquanto instrumento virtual, operado através de uma interface local via Bluetooth; e (ii) quando da sua integração com o EXEHDA, viabilizando o acesso às medições em uma perspectiva distribuída. Pode-se pontuar, por exemplo, a contribuição do *autorange* para a autonomia de operação do instrumento, já que a seleção automática de faixa de medição minimiza a necessidade de intervenção do usuário.

Com o prosseguimento dos esforços de pesquisa, pretende-se incorporar formatos de dados da web semântica aos serviços de medição, instrumentando com metadados as grandezas disponibilizadas. Ainda, entende-se como vantajoso o suporte a protocolos de descoberta de serviços (como o UPnP), o que

representaria um avanço ainda maior no sentido de reduzir a complexidade de configuração do instrumento. Além disso, é necessário qualificar o Módulo de Condicionamento do ubiMeter para aprimorar a precisão na faixa de medição de 2 V. Finalmente, pode ser interessante a realização de estudos de caso de escopo mais amplo, explorando aspectos como escalabilidade e precisão de forma mais aprofundada e com valores de tensão mais elevados.

## AGRADECIMENTOS

Os autores querem registrar seu agradecimento aos órgãos de fomento CAPES, CNPq e FAPERGS pelo apoio aos trabalhos de pesquisa relacionados ao *middleware* EXEHDA.

## REFERÊNCIAS

- [1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of Things," *Internet Computing, IEEE*, vol. 14, no. 1, pp. 44–51, Jan 2010.
- [2] P. F. Pires, F. C. Delicato, T. Batista, T. Barros, E. Cavalcante, and M. Pitanga, "Plataformas para a Internet das Coisas," in *Livro Texto de Minicursos – SBRC 2015*. Vitória: SBC, 2015.
- [3] C. Perera, A. B. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *Communications Surveys & Tutorials, IEEE*, vol. abs/1305.0982, 2013. [Online]. Available: <http://arxiv.org/abs/1305.0982>
- [4] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Final Report of the Industrie 4.0 Working Group*. Forschungunion, 2013.
- [5] M. W. Condry and C. B. Nelson, "Using smart edge IoT devices for safer, rapid response with industry IoT control operations," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 938–946, May 2016.
- [6] C. Salzmann, S. Govaerts, W. Halimi, and D. Gillet, "The Smart Device specification for remote labs," in *REV 2015*. Bangkok: IAOE, Feb 2015, pp. 199–208.
- [7] J. P. Carmo and J. H. Correia, "Wireless instrumentation," in *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement*, J. G. Webster and H. Eren, Eds. CRC press, 2014, ch. 85.
- [8] J. Lopes, R. Souza, A. Pernas, A. Yamin, and C. Geyer, "A distributed architecture for supporting context-aware applications in Ubicomp," in *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, May 2014, pp. 584–590.
- [9] R. S. d. Souza, J. Lopes, G. Gadotti, A. M. Pernas, A. Yamin, and C. Geyer, "Gerenciamento proativo de redes de sensores na Ubicomp," *Revista Brasileira de Computação Aplicada*, vol. 7, no. 3, pp. 53–64, 2015.
- [10] J. Lopes, R. Souza, G. Gadotti, A. Pernas, A. Yamin, and C. Geyer, "An architectural model for situation awareness in ubiquitous computing," *IEEE Latin America Transactions*, vol. 12, no. 6, pp. 1113–1119, 2014.
- [11] P. Davet, H. Kaiser Filho, L. João, L. Xavier, T. Carvalho, A. Fleischmann, A. Yamin, J. Lopes, and R. Souza, "Consciência de contexto na IoT: uma arquitetura distribuída e escalável," in *Proceedings of the V SBESC*. CPS, 2015.
- [12] H. Kaiser Filho, L. João, T. a. R. Carvalho, P. Davet, and A. Yamin, "Uma contribuição ao gerenciamento de recursos de sensoriamento e atuação no middleware EXEHDA direcionado à Internet das Coisas," in *Anais da ERAD/RS 2016*. São Leopoldo: SBC, 2016.
- [13] M. A. Razaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, Feb 2016.
- [14] E. Spanò, S. D. Pascoli, and G. Iannaccone, "Internet-of-Things infrastructure as a platform for distributed measurement applications," in *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*, May 2015, pp. 1927–1932.
- [15] P. Kostelnik, M. Sarnovsk, and K. Furdik, "The semantic middleware for networked embedded systems applied in the Internet of Things and Services domain," *Scalable Computing: Practice and Experience*, vol. 12, no. 3, pp. 307–316, 2011.
- [16] C.-L. Fok, G.-C. Roman, and C. Lu, "Servilla: a flexible service provisioning middleware for heterogeneous sensor networks," *Science of Computer Programming*, vol. 77, no. 6, pp. 663–684, 2012.
- [17] LogMeIn Inc. (2016) Xively. [Online]. Available: <https://xively.com/>