Ciência de Contexto na IoT: Uma Contribuição à Pesquisa Agropecuária

Patrícia Davet[1], Huberto Kaiser Filho[1], Leonardo João[1], Lucas Xavier[1], Rodrigo Souza[2], João Lopes[2], Ana Marilza Pernas[1], Nelsi Warken[3], Adenauer Yamin[1]

[1]Universidade Federal de Pelotas (UFPEL) – Pelotas, RS – Brazil

[2]Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre, RS – Brazil

[3]Embrapa Clima Temperado – Pelotas, RS – Brazil

{ptdavet,hkaiser,ldrsjoao,lmdsxavier,marilza,adenauer}@inf.ufpel.edu.br,
{rssouza,jlblopes}@inf.ufrgs.br, nelsi.warken@cpact.embrapa.br

Abstract. *The Context-Awareness has been gaining attention in the scenario of the IoT due to increasing need for autonomous behavior both from the devices such as of the applications. This paper presents the PlenUS project and software architecture for Context-Awareness giving you supports it, called EXEHDA-IoT. This architecture was designed to promove the acquisition, the storage and the processing of the context information, so distributed, independent of the applications in an autonomic approach based on rules. To evaluate the functionalities of the EXEHDA-IoT, a case study at Embrapa Temperate Climate is presented, having been obtained promising results.*

Resumo. *A Ciência de Contexto vem ganhando destaque no cenário da IoT devido a crescente necessidade de comportamento autônomo por parte tanto dos dispositivos como das aplicações. Este artigo apresenta o projeto plenUS e a arquitetura de software para Ciência de Contexto que lhe dá suporte, denominada EXEHDA-IoT. Esta arquitetura foi concebida para prover a aquisição, o armazenamento e o processamento das informações de contexto, de forma distribuída, independente das aplicações, em uma perspectiva autonômica baseada em regras. Para avaliar as funcionalidades do EXEHDA-IoT, um estudo de caso junto a Embrapa Clima Temperado é apresentado, tendo sido obtidos resultados promissores.*

1. Introdução

O projeto plenUS (*plentiful Ubiquitous Systems*) tem como premissa o emprego do middleware EXEHDA com o objetivo de disponibilizar Sistemas Ubíquos, que explorem relações pró-ativas entre usuários, softwares e equipamentos, auxiliando nas pesquisas em desenvolvimento na Embrapa Clima Temperado. O mesmo vem sendo desenvolvido por um consórcio de pesquisa envolvendo o LUPS (*Laboratory of Ubiquitous and Parallel Systems*) da UFPEL e a Embrapa Clima Temperado.

De modo mais específico o projeto plenUS tem como foco atender as demandas referentes ao provimento de serviços computacionais cientes de contexto, permitindo um registro histórico dos estados contextuais em que se encontram os equipamentos dos seguintes laboratórios da Embrapa: (i) Laboratório de Qualidade do Leite (Lableite), (ii)

Laboratório de Reprodução Animal e (iii) Laboratório de Análise de Sementes Oficial (LASO), como também outras necessidades de sensoriamento em ambientes administrativos como o do Núcleo de Tecnologia da Informação (NTI), e de uma Sala de Testes do plenUS.

Com a implementação do sistema de qualidade nestes laboratórios aumentaram as exigências no controle dos equipamentos onde são realizadas as análises, visando precisão e confiabilidade nos resultados. Deste modo, tornam-se necessários registros das leituras de grandezas físicas como temperatura, umidade e luminosidade dos equipamentos envolvidos (germinadores, câmaras frias, salas de trabalho, etc.), bem como uma atuação automatizada em função das informações sensoriadas, sempre que for o caso.

A infraestrutura computacional proporcionada pela Internet das Coisas (*Internet of Things* - IoT) permite que estes equipamentos possam trocar informações entre si e/ou com usuários via Internet, permitindo o desenvolvimento de aplicações com maior valor agregado. Isto é possível devido à incorporação de dispositivos computacionais a estes equipamentos, unicamente identificados, e com capacidade de conexão à Internet, sensoriamento, armazenamento e processamento [Pires et al. 2015].

Por sua vez as aplicações IoT, enquanto ubíquas e assim dotadas de comportamento autônomo, necessitam ter ciência das informações contextuais que lhe interessam. Para o desenvolvimento de aplicações cientes de contexto no cenário da IoT, há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais adquiridas por sensores, a partir de um grande número de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, armazenamento, e realização de inferências para seu uso em procedimentos de atuação [Perera et al. 2013].

Neste sentido, este trabalho tem como objetivo a concepção de uma arquitetura de software comprometida com a premissa de fornecer mecanismos para o desenvolvimento de aplicações cientes de contexto na IoT, denominada EXEHDA-IoT. Esta arquitetura é baseada no CoIoT (Context+IoT) e sua avaliação aconteceu junto ao projeto plenUS. O CoIoT especializa o middleware EXEHDA (*Execution Environment for Highly Distributed Applications*) no tratamento das questões de Ciência de Contexto na IoT [Souza et al. 2015].

O artigo está organizado da seguinte forma: a Seção 2 descreve a modelagem e principais funcionalidades da arquitetura de software proposta. A Seção 3 apresenta a avaliação da arquitetura pelo projeto plenUS. A Seção 4 discute os trabalhos relacionados, e, por fim, a Seção 5 apresenta as considerações finais.

2. EXEHDA-IoT: Arquitetura e Funcionalidades

O EXEHDA é um middleware baseado em serviços que visa criar e gerenciar um ambiente ubíquo, bem como promover a execução de aplicações sobre esse ambiente. No EXEHDA, as condições de contexto são pró-ativamente monitoradas e o suporte à execução deve permitir que tanto a aplicação como ele próprio utilizem essas informações na gerência da adaptação de seus aspectos funcionais e não-funcionais. O mecanismo de adaptação proposto para o EXEHDA emprega uma estratégia colaborativa entre aplicação e ambiente de execução, através da qual é facultado ao programador individualizar

políticas de adaptação para reger o comportamento de cada um dos componentes que constituem o software da aplicação [Lopes et al. 2014].

O fato do EXEHDA ter uma origem enquanto middleware para UbiComp, faz com que o tratamento da Ciência de Contexto seja inerente à sua arquitetura. Para que o EXEHDA continue provendo Ciência de Contexto tendo em vista os desafios do cenário da IoT, onde a aquisição das informações de contexto ocorre a partir de um grande número de dispositivos heterogêneos, distribuídos dinamicamente e com capacidade restrita, está em desenvolvimento o CoIoT [Souza et al. 2015].

Considerando isso, o EXEHDA-IoT foi modelado empregando os seguintes pressupostos do CoIoT:

- *automatização*: implica que um middleware para IoT deve suportar a coleta de informações, processamento e inferência contextual de forma autônoma;
- *inteligência*: dispositivos IoT devem estar capacitados a operar, com algum grau de adaptação as variações contextuais (variações nas infraestruturas de rede, aplicações e usuários, por exemplo);
- *dinamicidade*: um dispositivo pode se mover de um lugar a outro, necessitando que o middleware esteja apto para reconhecer esta mudança e consequentemente realizar os ajustes necessários;
- *zero configuração*: para simplificar os procedimentos de integração e/ou remoção de dispositivos, recursos *plug and play* devem estar disponíveis, reduzindo os esforços de gerência por parte dos usuários e facilitando o crescimento descentralizado que é inerente nas infraestruturas da IoT.

O ambiente ubíquo disponibilizado pelo EXEHDA-IoT é organizado em células de execução, nas quais os dispositivos computacionais são distribuídos (vide Figura 1). Cada célula é constituída dos seguintes componentes: (i) EXEHDAbase, o elemento central da célula, sendo responsável por todos serviços básicos e constituindo referência para os demais elementos; (ii) o EXEHDAnodo que corresponde aos dispositivos computacionais responsáveis pela execução das aplicações; (iii) o EXEHDAnodo móvel, um subcaso do anterior, que corresponde aos dispositivos tipicamente móveis que podem se deslocar entre as células do ambiente ubíquo, como *notebooks*, *tablets* ou *smartphones*, (iv) o EXEHDAborda, responsável por fazer a interoperação entre os serviços do middleware e os diversos tipos de *gateways*; e (V) o EXEHDAgateway, que consiste no elemento responsável por setorizar pontos de coleta e/ou atuação distribuídos, disponíveis no meio físico, realizando a interação destes com os outros componentes do middleware.

Para provimento de Ciência de Contexto na IoT, o EXEHDA-IoT se vale de dois tipos principais de servidores: Servidor de Borda e Servidor de Contexto (vide Figura 1). O Servidor de Borda é instanciado em um equipamento do tipo EXEHDAborda e se destina a gerenciar a interação com o meio físico através de *gateways*, bem como a execução de regras de contingência e armazenamento temporário das informações contextuais coletadas, em caso de falha de comunicação. O Servidor de Contexto, por sua vez, é alocado no EXEHDAbase e atua no armazenamento e no processamento das informações contextuais, integrando informações históricos e aquelas provenientes de diferentes Servidores de Borda distribuídos no ambiente ubíquo.

Na proposta do EXEHDA-IoT a premissa é que os sensores e/ou atuadores sejam incorporados ao Servidor de Borda através de *gateways*. Os *gateways* são utilizados

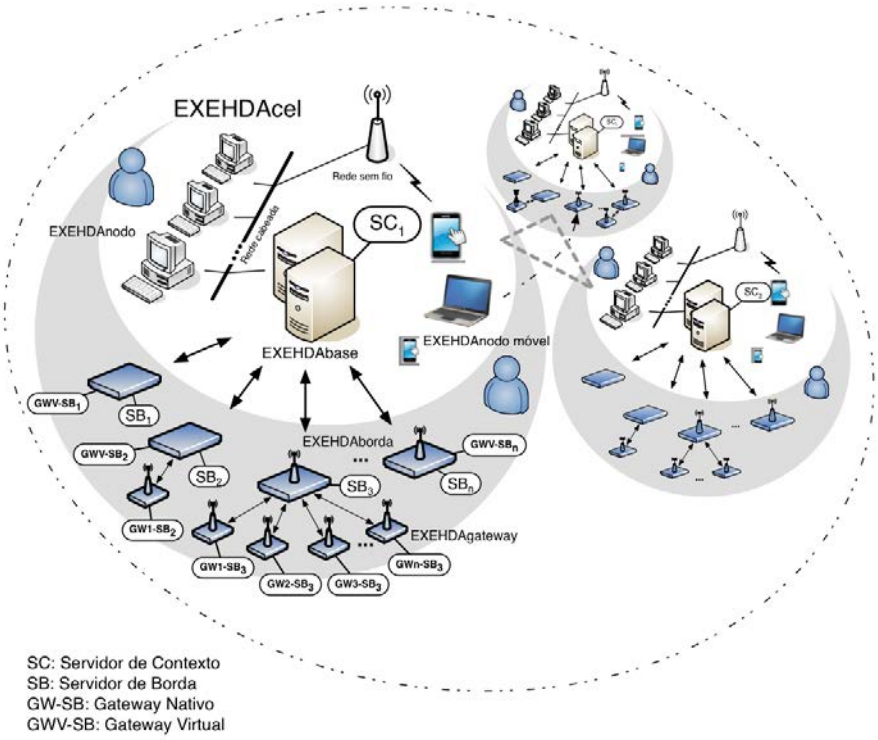

Figura 1. Organização Celular do Ambiente Ubíquo Gerenciado pelo EXEHDA-IoT

então, para tratar a heterogeneidade, tanto de hardware como de software, inerentes a dispositivos de sensoriamento e/ou atuação, realizando a conversão de protocolos e o gerenciamento dos dispositivos, além de fornecer a estes capacidade de comunicação via Internet.

Com o intuito de garantir a interoperabilidade com as tecnologias de mercado, e também potencializar a distribuição das iniciativas de coleta e/ou atuação, foram previstos três tipos distintos de *gateways*:

- Gateways Proprietários (GW-PR), que tem funcionalidades e protocolos heterogêneos variando de acordo com seus fabricantes;
- Gateways Nativos (GW-SB), cujas funcionalidades operam de maneira integrada à arquitetura do EXEHDA, sendo instanciados em equipamentos do tipo EXEHDAgateway, e;
- Gateway Virtual (GWV-SB), localizado no mesmo hardware em que executa o Servidor de Borda, constituindo-se em uma virtualização do Gateway Nativo, dispensando quando possível a necessidade de um hardware específico para o *gateway*.

O suporte arquitetural aos Gateways Proprietários depende dos recursos oferecidos pelas suas APIs, sendo garantida, a priori, apenas a leitura dos valores aquisitados e o envio de comandos de atuação, quando suportados. Por outro lado, os Gateways Nativos e Gateways Virtuais, além de oferecer acesso direto aos seus recursos (atuação e sensoriamento), possuem mecanismos de sensoriamento gerenciados através de eventos utilizando estratégias de *trigger*.

Considerando a dinamicidade e escalabilidade do cenário da IoT, o EXEHDA-IoT possibilita a identificação e descoberta automática dos dispositivos de sensoriamento e/ou atuação, a partir dos *gateways* integrados a arquitetura, facilitando a gerência dos dispositivos, quando da sua inclusão e/ou remoção. Para tal, foi utilizada a estratégia de comunicação *plug and play* do protocolo UPnP (*Universal Plug and Play*) [UPnP 2015].

A arquitetura de software do EXEHDA-IoT é baseada no CoIoT (vide Figura 2) e provê comunicação: (i) entre os *gateways* e o Servidor de Borda; (ii) entre os Servidores de Borda e o Servidor de Contexto; (iii) entre os Servidores de Contexto localizados em diferentes células do ambiente ubíquo; e (iv) com outros serviços do *middleware* ou aplicações. As linhas tracejadas na representação dos *gateways* indicam que algum dos tipos de *gateways* podem não estar presentes na arquitetura, a qual possibilita utilizar somente *gateways* físicos (Nativos ou Proprietários), somente Gateway Virtual, ou ambos.

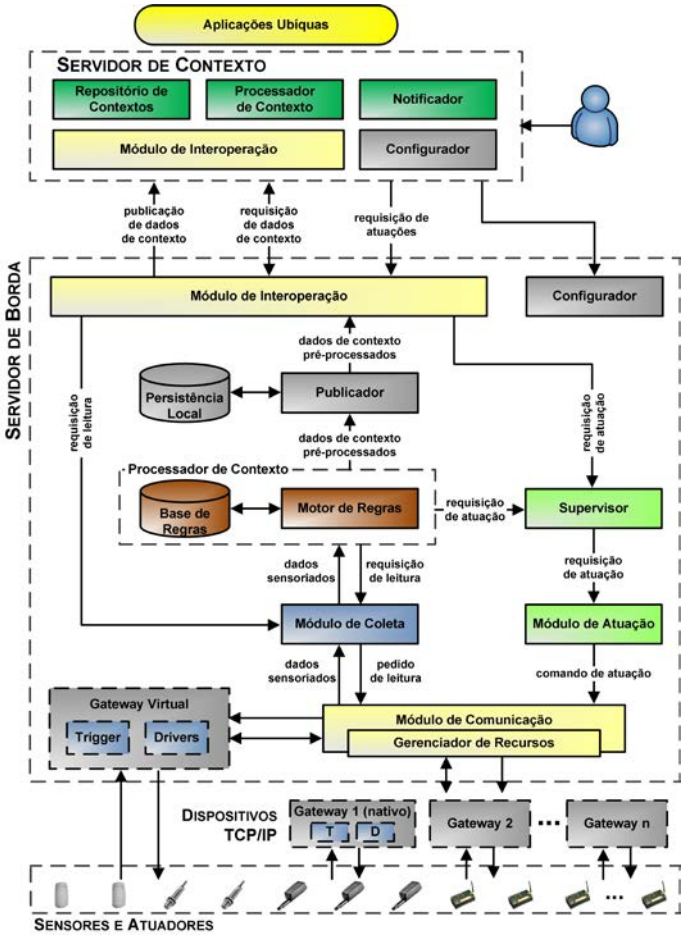

Figura 2. Arquitetura EXEHDA-IoT

2.1. Servidor de Contexto

O Servidor de Contexto no EXEHDA-IoT tem suas funcionalidades organizadas através dos seguintes módulos arquiteturais.

Repositório de Contextos: armazena dados de configuração da arquitetura, e os valores sensoriados publicados pelos Servidores de Borda existentes no ambiente ubíquo.

Processador de Contexto: realiza tarefas de manipulação e dedução das informações contextuais. Baseia-se em regras do tipo ECA (Evento-Condição-Ação), de forma a tratar eventos gerados a partir de mudanças nos estados dos contextos de interesse, que podem acarretar no disparo de ações pertinentes em função do estado contextual.

Notificador: tem a função de gerar notificações necessárias para a operação do EXEHDA-IoT. As notificações são consequências do processamento das informações contextuais coletadas. Este módulo utiliza uma estratégia de notificação baseada no modelo *publish/subscribe*, em que recebe subscrições de todos os serviços e/ou aplicações que requerem notificações a respeito das mudanças nos estados contextuais. Também é função do Notificador enviar comandos de atuação aos Servidores de Borda quando necessário, bem como a realização de notificação pela aplicação Web de novo gateway descoberto, o qual só será utilizado mediante autenticação pelo usuário interessado.

Módulo de Interoperação: considerando o perfil inerentemente distribuído das aplicações alvo do EXEHDA-IoT, o Módulo de Interoperação foi concebido para gerenciar a troca de informações com os Servidores de Borda, bem como os demais serviços do middleware em execução em outros equipamentos, como também as aplicações, oferecendo para isso uma interface padronizada e bem definida. A concepção deste módulo teve como referência o estilo arquitetural REST, sendo utilizado o protocolo HTTP e suas operações (GET, PUT, POST, DELETE) no suporte às trocas de informações.

Configurador: aglutina todos os procedimentos de configuração necessários para o funcionamento do EXEHDA-IoT. Através deste módulo é disponibilizada uma aplicação através da qual é possível gerenciar dispositivos *gateways* e regras de processamento de contexto, bem como realizar todo tipo de cadastro necessário ao funcionamento da aplicação IoT ciente de contexto.

2.2. Servidor de Borda

No EXEHDA-IoT o Servidor de Borda tem suas funcionalidades sistematizadas através dos seguintes módulos arquiteturais.

Módulo de Interoperação: é destinado a gerenciar a comunicação com o Servidor de Contexto, tendo também como referência o estilo arquitetural REST. Com isso, todas as funcionalidades associadas aos *gateways*, sensores e atuadores são mapeadas na forma de recursos e acessadas via URIs.

Configurador: no Servidor de Borda o módulo Configurador é responsável pelos ajustes necessários para o seu funcionamento. Entre as funcionalidades oferecidas tem-se: definição do Servidor de Contexto para a publicação das informações sensoriadas, bem como a inserção, atualização, ativação e desativação dos *gateways* e seus serviços por meio do protocolo de comunicação UPnP.

Publicador: é responsável por coordenar o principal fluxo de informações entre os Servidores de Borda e o Servidor de Contexto, promovendo a publicação de todos as informações contextuais coletadas e garantindo uma **Persistência Local** das mesmas nos períodos que a publicação ficar inviabilizada. As transações ocorrem sobre o protocolo HTTPS (*HyperText Transfer Protocol Secure*), o que garante que as informações serão transferidas criptografadas, evitando acessos indevidos aos mesmos.

Processador de Contexto: é responsável por tratar regras de contingência, com a

intenção de evitar que os dispositivos envolvidos atinjam estados críticos, o que faz com que este módulo adquira elevada importância quando a comunicação com o Servidor de Contexto for interrompida por problemas na infraestrutura de rede.

Supervisor: possui como objetivo aglutinar os comandos de atuação. Uma vez recebidos os parâmetros para controle da atuação, o módulo Supervisor avalia eventuais conflitos entre as requisições de atuação oriundas de diferentes fontes, para após enviar o comando de atuação ao Módulo de Atuação.

Módulo de Coleta: tem a função de direcionar os pedidos de coleta ao Módulo de Comunicação que envia aos respectivos *gateways*, sob demanda tanto do Motor de Regras, como do Servidor de Contexto ou das aplicações a qualquer momento. O módulo faz a recepção assíncrona das solicitações e repassa o ID do sensor a ser lido pelo gateway responsável. Este módulo também tem a função de direcionar aos demais elementos da arquitetura as informações sensoriadas enviadas pelos *gateways*, produzidas a partir de eventos do ambiente.

Módulo de Atuação: possui um funcionamento análogo ao Módulo de Coleta. Ele recebe do Supervisor comandos com o ID do atuador e os correspondentes padrões de operação (tempo de duração, potência de ativação, etc.), e os envia ao Módulo de Comunicação a fim de se serem interpretados e direcionados ao gateway responsável pelo efetivo tratamento.

Gateway Virtual: realiza o interfaceamento entre sensores e/ou atuadores (dispositivos sem conexão Internet) ligados diretamente ao Servidor de Borda. Possui dois módulos básicos: **Drivers** e **Trigger**. Os Drivers são responsáveis pelo encapsulamento individualizado de aspectos tecnológicos, tanto de sensores como atuadores, o que evita que as suas diferenças operacionais se projetem nos demais módulos da arquitetura. O Trigger é um módulo que tem a finalidade de gerenciar a leitura dos sensores através de eventos. Consiste em um tipo simplificado de regra de produção, que dá suporte a um conjunto limitado de funções para o tratamento de condições e ações.

Módulo de Comunicação: tem a função de prover aos demais módulos da arquitetura um acesso padronizado aos recursos disponibilizados pelos *gateways* no que diz respeito ao tratamento de sensores e/ou atuadores. Gerencia as comunicações com os Gateways Nativos e Gateways Virtuais através do protocolo UPnP, que se utiliza de princípios REST como forma de prover interoperabilidade entre dispositivos cliente e servidor. Por outro lado, o acesso aos serviços dos Gateways Proprietários é realizado através das APIs disponibilizados pelos seus fabricantes, que em muitos casos baseiam-se em protocolos típicos da IoT, como CoAP[1] e MQTT[2].

Gerenciador de Recursos: foi concebido com o objetivo de administrar procedimentos, como a entrada e saída de operação de dispositivos, bem como à reposição por avaria ou necessidade de adicionar outro ponto de monitoramento. É neste módulo que é instanciado o Ponto de Controle UPnP, que possui como tarefa descobrir os dispositivos UPnP (Gateways Nativo e Virtual) presentes na rede, armazenando as informações sobre os dispositivos e seus respectivos serviços descobertos (sensores e/ou atuadores) em um diretório, para eventual controle destes.

[1] http://coap.technology/
[2] http://mqtt.org/

3. EXEHDA-IoT: Avaliação no Domínio Agropecuário

As funcionalidades propostas para o EXEHDA-IoT, foram avaliadas através de estudo de caso no domínio agropecuário relacionado ao projeto plenUS. Os ambientes priorizados pelo projeto, para recebimento de infraestrutura para ciência do contexto em que se encontram seus equipamentos, bem como os tipos de grandezas mensuradas e a disposição dos equipamentos utilizados podem ser visualizados na Figura 3.

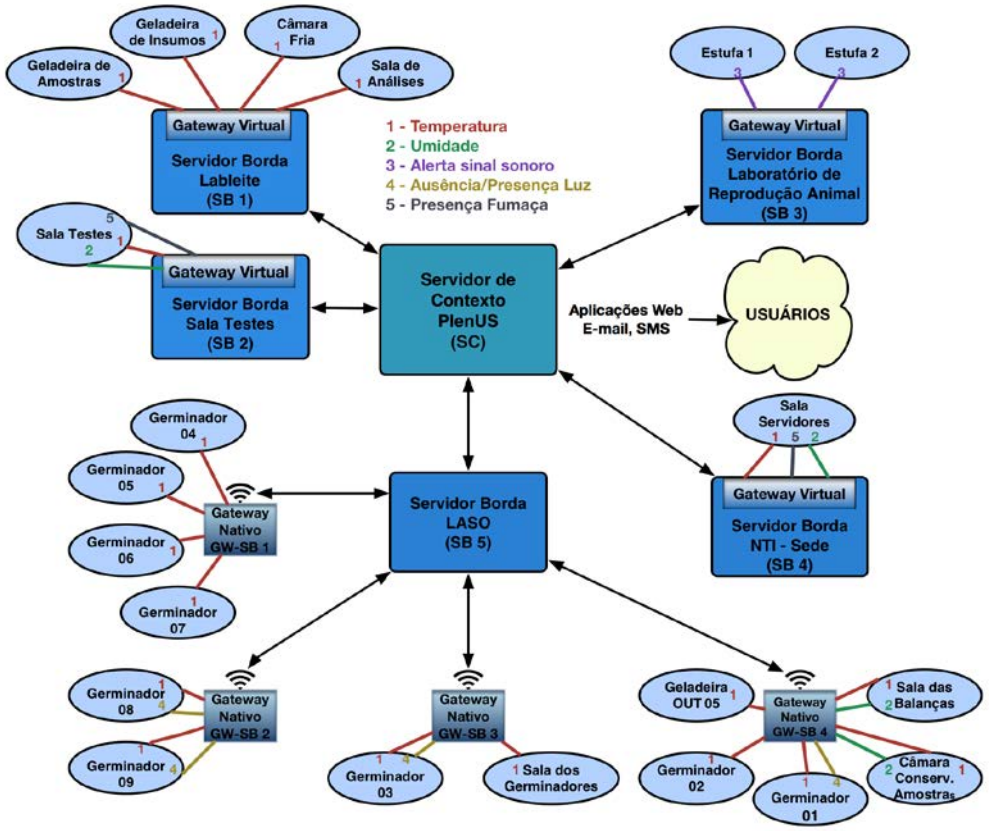

Figura 3. Estudo de Caso - Projeto plenUS

Considerando a estrutura física do estudo de caso, um Servidor de Contexto, cinco Servidores de Borda e quatro Gateways Nativos foram instalados para atender as demandas da Embrapa como um todo (vide Figura 3), constituindo uma célula de execução do ambiente ubíquo.

O Servidor de Contexto e o Servidor de Borda 4 (SB 4) foram posicionados no NTI, localizado na unidade Sede da Embrapa Clima Temperado, enquanto os outros Servidores de Borda foram posicionados na unidade Estação Experimental Terras Baixas (ETB), distante 21Km.

O hardware utilizado para o Servidor de Contexto foi um desktop com processador Intel Pentium E2160-1.8GHz de dois núcleos, com 3Gb de memória RAM e com o Sistema Operacional Ubuntu Server. Já os Servidores de Borda foram prototipados sobre o Sistema Operacional Raspbian, sendo usado como hardware a Raspberry PI Modelo B+ [3] (vide Figura 4a).

[3]http://www.raspberrypi.org

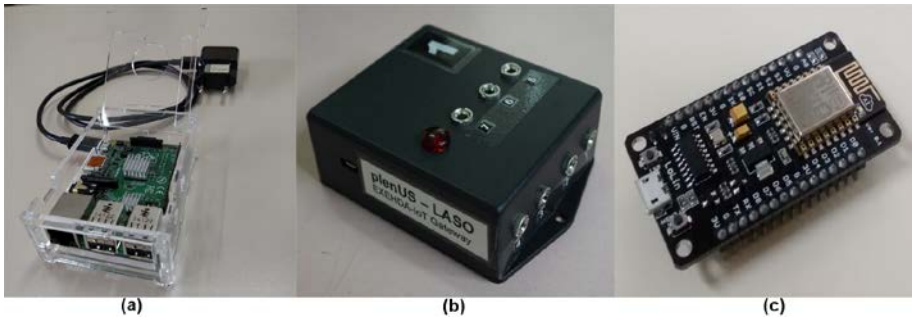

Figura 4. Hardwares: (a) Raspberry PI; (b) Gateway Nativo; (c) ESP8266-12

Enquanto aos *gateways* previstos pelo EXEHDA-IoT e utilizados para o estudo de caso, estes são de duas naturezas: **Gateways Virtuais**, internos ao Servidor de Borda (SB), e **Gateways Nativos** com conexão wi-fi com o SB. Os Gateways Nativos foram utilizados no caso onde as demandas de sensoriamento introduziram a sua necessidade, seja pela disposição física dos equipamentos e/ou salas ou pela quantidade significativa de sensores.

Para a escolha do hardware utilizado na concepção do Gateway Nativo (vide Figura 4b) foram considerados aspectos como: baixo consumo energético, possibilidade de operação em regime sem fio (wi-fi) e emprego de plataforma de software disseminada. Após revisão de literatura a escolha recaiu sobre o processador ESP8266-12 [Kurniawan 2015] (vide Figura 4c).

A operacionalização do cenário propiciado pelo estudo de caso é realizada através de regras distribuídas do tipo ECA (Evento-Condição-Ação) entre os Servidores de Borda e Contexto. Uma contextualização da forma de processamento das regras tipo ECA no cenário de avaliação é apresentado na Figura 5 por meio de uma notação BPMN [Group 2015].

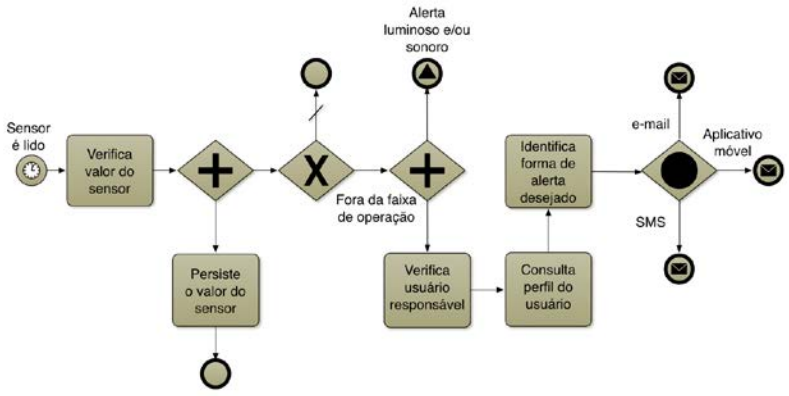

Figura 5. Processamento de Regra Tipo ECA no Cenário de Avaliação (Notação BPMN)

O cenário consiste em monitorar uma determinada grandeza e produzir registro histórico. Enquanto a grandeza é coletada e armazenada, a mesma tem seu valor avaliado tendo em vista as faixas de operação para cada dado de contexto coletado. Alertas são enviados aos usuários envolvidos, quando os valores estiverem fora da faixa de operação

especificada para o experimento.

A interação do usuário com o cenário é disponibilizada por meio de uma aplicação desenvolvida para navegadores web, que contempla tarefas de gerenciamento dos dispositivos de sensoriamento e/ou atuação, com suporte a descoberta automática, o que agregou funcionalidades de cadastro e/ou remoção automática dos dispositivos *gateways*, bem como de seus sensores quando de sua disponibilidade. Além disso, a aplicação também permite a visualização do registro histórico das informações contextuais, tanto de forma gráfica como textual.

O relatório gráfico (vide Figura 6) oferecido pelo sistema permite a visualização simultânea das informações de vários sensores. A seleção dos sensores é feita a partir de um menu com suporte a múltipla seleção. Também é disponibilizado um recurso de inspeção que permite a comparação dos valores em um determinado instante do tempo.

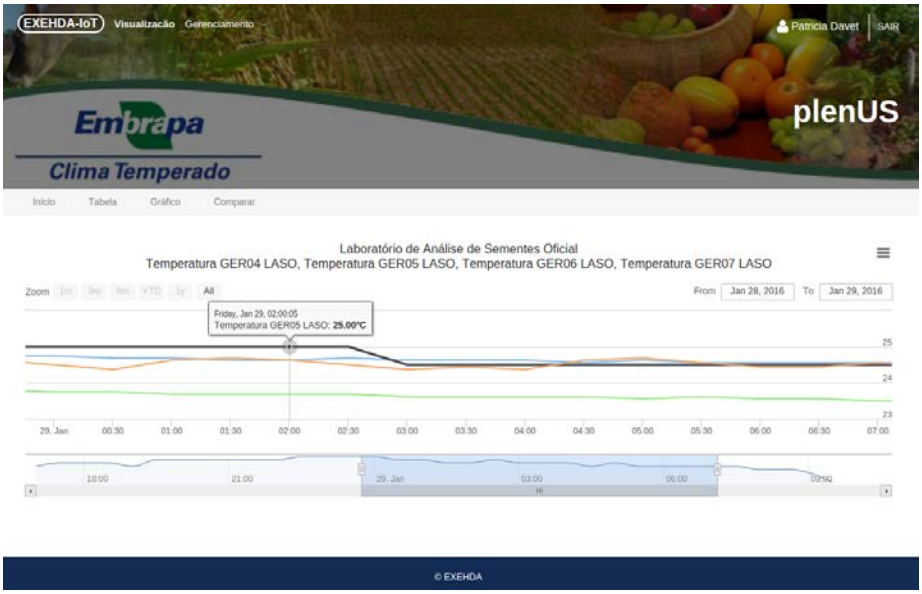

Figura 6. Projeto plenUS - Relatório gráfico

Para atender o fato da rotina de trabalho dos laboratoristas implicar em uma mobilidade nos diversos recintos do laboratório, a aplicação Web desenvolvida possui características responsivas, o que veio agregar maior funcionalidade ao seu uso, pois possibilitou que as configurações e testes dos dispositivos de sensoriamento e/ou atuação pudessem ser realizadas no local de instalação por meio de *smartphones*, devido ao seu *layout* se adaptar ao dispositivo computacional utilizado.

4. Trabalhos Relacionados

Uma análise dos trabalhos relacionados EcoDif [Pires et al. 2014], S^3OIA [Vega-Barbas et al. 2012], LinkSmart [LinkSmart 2015] e Carriots [Carriots 2015] em relação à satisfação dos requisitos IoT considerados neste trabalho é apresentada na Tabela 1, onde o símbolo "X" denota que o requisito é completamente atendido, o símbolo "O" indica que o requisito é parcialmente atendido, e o símbolo "-" indica que o requisito não é atendido.

Tabela 1. Análise dos Trabalhos Relacionados

Plataformas de middleware	Interoperabilidade	Descoberta e Gerenciamento de Dispositivos	Interfaces de Alto Nível	Ciência de Contexto	Escalabilidade
EcoDiF	X	O	X	O	X
S^{3}OIA	X	X	-	X	-
LinkSmart	X	O	-	X	-
Carriots	X	O	X	O	X
EXEHDA-IoT	X	X	X	X	X

Registre-se que o único requisito endereçado de uma maneira ou outra por todos os trabalhos relacionados, foi o de interoperabilidade. Isto é devido ao fato deste ser um requisito primordial e que deve ser imperativamente endereçado por qualquer plataforma de middleware para IoT.

Dentre os trabalhos analisados, o S^{3}OIA destaca-se por ser o único a apresentar uma estratégia de descoberta de dispositivos automática, *plug and play* e de autoconfiguração para o gerenciamento de sensores e/ou atuadores. Os outros atendem ao critério de descoberta e gerenciamento de dispositivos de forma parcial, pressupondo que os dispositivos e seus recursos requeridos estejam disponíveis por meio de um cadastro prévio. No EXEHDA-IoT, por outro lado, se utiliza o UPnP como estratégia de autoconfiguração e descoberta, tendo o gateway como elemento básico do mecanismo de descoberta e a partir dele, os sensores e atuadores são facilmente descobertos e gerenciados. Além disso, a abordagem utilizada considera todo o potencial de protocolos de uso consolidado.

A ciência de contexto é contemplada por todos os trabalhos relacionados, porém o suporte oferecido, como no caso do EcoDif e Carriots mostra-se limitado. Já o EXEHDA-IoT fornece uma organização distribuída para coleta e processamento de contexto através de regras, como também para atuação em ambientes com características heterogêneas e escaláveis. Além disso, por ter sido desenvolvido a partir de um middleware para UbiComp, o EXEHDA, também provê suporte a outros serviços ubíquos disponibilizados pelo middleware, como adaptação, comunicação desacoplada, migração de código, etc.

Por fim, outra abordagem adotada pelo EXEHDA-IoT, que trouxe benefícios principalmente no que tange à otimização do tráfego de rede e ao consumo de energia, é a proposição do uso de *triggers* para gerenciar o fluxo de informações transmitidas. A abordagem de *triggers* foi concebida para permitir a personalização da coleta das informações através de eventos considerando as características de variabilidade física de cada grandeza monitorada, o que proporciona minimizar o fluxo das informações entre os *gateways* e o Servidor de Borda.

Além disso, a abordagem distribuída do processamento do contexto entre os Servidores de Borda e Contexto proposta pelo EXEHDA-IoT também contribui com a minimização do fluxo de informações contextuais em relação as abordagens centralizadas, empregadas pelos trabalhos relacionados.

5. Considerações Finais

O principal diferencial do EXEHDA-IoT em relação aos trabalhos relacionados diz respeito a possibilidade de gerenciar a aquisição, armazenamento e processamento das

informações de contexto, de forma distribuída, independente das aplicações, em uma perspectiva autonômica baseada em regras.

Os resultados obtidos pelo estudo de caso, no domínio agropecuário atenderam as demandas dos usuários, além de fornecer um *feedback* para avanços futuros, estimulando a continuidade das pesquisas na área.

Como trabalhos futuros, os seguintes aspectos foram priorizados: (i) explorar estudos de caso em que as regras de processamento contextual utilizem outros mecanismos de inferência de mais alto nível; e (ii) implantação de um framework para gerenciamento do motor de regras, tanto no Servidor de Borda como no Servidor de Contexto.

Referências

Carriots (2015). Carriots iot application platform. Disponível em :<https://www.carriots.com/>. Acesso em julho de 2015.

Group, O. M. (2015). Business Process Model and Notation. Disponível em :<http://www.bpmn.org>. Acesso em maio de 2015.

Kurniawan, A. (2015). *Sparkfun ESP8266 Thing Development Workshop*. PE PRESS, Canada.

LinkSmart (2015). Middleware for networked embedded systems. Disponível em :<https://linksmart.eu/redmine/projects/linksmart-opensource/wiki>. Acesso em julho de 2015.

Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *Journal of Universal Computer Science*, 20(9):1327–1351.

Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2013). Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1):414–454.

Pires, P. F., Cavalcante, E., Barros, T., Delicato, F. C., Batista, T., and Costa, B. (2014). A platform for integrating physical devices in the internet of things. *Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing*, pages 234–241.

Pires, P. F., Delicato, F. C., Batista, T., Barros, T., Cavalcante, E., and Pitanga, M. (2015). Plataformas para a internet das coisas. *Livro Texto de Minicursos - SBRC 2015*.

Souza, R. S., Lopes, J. L. B., Davet, P. T., Garcia, C. G., Gadotti, G. I., Yamin, A. C., and Geyer, C. F. R. (2015). Context awareness in ubicomp: an iot oriented distributed architecture. *Proceedings of 2015 IEEE International Conference on Electronics, Circuits, and Systems*.

UPnP, F. (2015). Fórum upnp. Disponível em :<http://upnp.org//>. Acesso em julho de 2015.

Vega-Barbas, M., Casado-Mansilla, D., Valero, M., López-de Ipina, D., Bravo, J., Flórez, F., et al. (2012). Smart spaces and smart objects interoperability architecture (s3oia). *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 725–730.