

Desenvolvimento de uma Aplicação para Escritórios Inteligentes Usando a Plataforma FIWARE e o ambiente FIWARE-Lab@RNP

Francielly S. Almeida¹, Gustavo T. Laureano, Leandro A. Freitas², Sand L. Correa¹

¹Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Câmpus Samambaia
74.690-900 – Goiânia – GO – Brasil

²Instituto Federal de Goiás
Núcleo de Estudos aplicados a Redes de computadores e
Sistemas distribuídos (NumbERS)
Avenida Universitária, s/nº, Vale das Goiabeiras
75.400-000 – Inhumas – GO – Brasil

{franciellyalmeida,sand}@inf.ufg.br, gustavo@inf.ufg.br

leandro.freitas@ifg.edu.br

Resumo. Este trabalho apresenta os resultados de um estudo comparativo envolvendo a FIWARE, uma plataforma de desenvolvimento de aplicações de Internet das Coisas (IoT), e o FIWARE-Lab@RNP, um laboratório virtual e remoto para prototipação e experimentação de aplicações IoT baseadas na plataforma FIWARE. O objetivo principal do FIWARE-Lab@RNP é facilitar a criação, configuração e gerenciamento de implantações FIWARE, disponibilizando instâncias da plataforma de forma transparente e sem a necessidade de conhecer especificidades dos principais componentes e serviços da plataforma. O estudo apresenta uma visão prática e tem como objetivo mensurar o quanto um ambiente, com as características apresentadas pelo FIWARE-Lab@RNP, diminui o esforço de desenvolvimento de aplicações IoT baseadas na plataforma FIWARE. Para atingir esse objetivo, desenvolvemos uma aplicação piloto de IoT, no contexto de Escritórios Inteligentes, diretamente na plataforma, como também através dos recursos providos pelo FIWARE-Lab@RNP.

Abstract. This work presents the results of a comparative study involving FIWARE, a platform for the development of Internet of Things (IoT) applications, and the FIWARE-Lab@RNP, a virtual and remote laboratory for prototyping and experimentation of IoT applications based on the FIWARE platform. The main goal of the FIWARE-Lab@RNP is to ease the creation, configuration, and management of FIWARE deployments. This is done by providing instances of the platform in a transparent manner and without knowing the specificities of the main components and services of the platform. The study presents a practical view and aims to measure how much an environment, with the characteristics presented by the FIWARE-Lab@RNP, decreases the effort to develop IoT applications based on the FIWARE platform. To achieve this goal, we developed a pilot IoT application in the context of Smart Offices using both the platform and the resources provided by the FIWARE-Lab@RNP.

1. Introdução

O paradigma de Internet das Coisas (*Internet of Things* - IoT) descreve um mundo onde objetos físicos cotidianos, conectados à Internet, coletam dados de seus arredores com o propósito de monitoramento, análise, otimização e controle [Al-Fuqaha et al. 2015]. Para realizar essa visão, aplicações IoT requerem diferentes tipos de software, hardware e tecnologias de comunicação trabalhando de forma integrada. Para facilitar o desenvolvimento de tais aplicações, um *middleware* ou plataforma IoT é normalmente empregado, visando: i) abstrair as especificidades dos dispositivos; ii) promover interoperabilidade; e iii) facilitar o gerenciamento e a segurança dos dispositivos e dos dados [Singh and Kapoor 2017, Razzaque et al. 2016, da Cruz et al. 2018].

Dentre as diversas plataformas IoT existentes atualmente, a FIWARE [FIWARE 2020], uma plataforma aberta desenvolvida pela Comunidade Europeia, tem sido amplamente utilizada pela indústria e a comunidade científica. A popularidade da FIWARE se deve ao fato de ser uma plataforma aberta, bem como ao seu conjunto extenso de funcionalidades relevantes para o desenvolvimento de aplicações IoT em diferentes domínios de aplicação. Essas funcionalidades são oferecidas através de componentes denominados *Generic Enablers* (GEs), os quais podem ser estendidos e reutilizados para facilitar o desenvolvimento de uma aplicação.

Uma vez selecionados os componentes GEs necessários ao desenvolvimento da aplicação, tais componentes devem ser implantados na infraestrutura do desenvolvedor de forma manual ou através de ferramentas como contêineres. Contudo, dado o catálogo extenso de componentes funcionais atualmente oferecidos pela plataforma e as especificidades de como instalar, configurar e instanciar cada componente, implantar um ambiente FIWARE pode ser uma tarefa complexa, especialmente para usuários com pouca experiência na plataforma.

Para minimizar esse problema, o ecossistema FIWARE disponibiliza alguns laboratórios virtuais, onde instâncias de execução contendo os principais GEs da plataforma são disponibilizados através da Internet. Esses laboratórios, denominados *FIWARE-Labs*, permitem os usuários explorarem a plataforma FIWARE sem a necessidade de implantá-la em suas próprias infraestruturas computacionais. Tais laboratórios virtuais também proveem serviços que escondem especificidades de instalação e configuração dos componentes da plataforma. No Brasil, existe apenas um ambiente *FIWARE-Lab* em implantação, denominado *FIWARE-Lab@RNP*, sendo esse gerenciado e mantido pela Rede Nacional de Ensino e Pesquisa (RNP).

Este trabalho apresenta um estudo comparativo envolvendo a plataforma FIWARE e o ambiente *FIWARE-Lab@RNP*. O estudo tem como objetivo mensurar o quanto um ambiente com as características apresentadas pelo *FIWARE-Lab@RNP* diminui o esforço de desenvolvimento de aplicações IoT baseadas na plataforma FIWARE. Para atingir esse objetivo, utilizamos uma abordagem prática. Desenvolvemos uma aplicação piloto de IoT para Escritórios Inteligentes considerando duas abordagens: i) utilizando manualmente a plataforma; e ii) utilizando os recursos providos pelo *FIWARE-Lab@RNP*. Finalmente, comparamos o esforço de desenvolver o piloto nas duas abordagens propostas. Resultados mostram que ao prover imagens pré-configuradas, a partir das quais os principais componentes da plataforma podem ser instanciados, e fornecer um ambiente gráfico para instanciar, configurar e interagir com as instâncias criadas, o ambiente *FIWARE-Lab@RNP*

alcance seu objetivo.

Este trabalho está organizado da seguinte forma. As Seções 2 e 3 apresentam, respectivamente, uma visão geral da plataforma FIWARE e do ambiente FIWARE-Lab@RNP. A Seção 4 descreve em maiores detalhes os requisitos funcionais da aplicação piloto e como ela foi desenvolvida nas duas abordagens. A Seção 5 apresenta os resultados experimentais obtidos neste trabalho e a Seção 6 a conclusão e os trabalhos futuros.

2. FIWARE

A FIWARE é uma plataforma IoT de código aberto criada pela Comissão Europeia no contexto da Internet do Futuro [Domingue et al. 2011]. A plataforma tem por objetivo oferecer um ambiente em que haja suporte ao desenvolvimento de aplicações em diversas áreas, como cidades inteligentes, logística, fontes de energias renováveis, agricultura de precisão, dentre outras [Trindade et al. 2017]. Dado o seu foco em diferentes domínios de aplicação, a FIWARE apresenta uma arquitetura extensível, composta por diversos elementos denominados *Generic Enablers* (GE) que usam o padrão *Next Generation Service Interface* (NGSI-09/NGSI-10) para se comunicarem. O padrão NSGI foi desenvolvido pela *Open Mobile Alliance* (OMA) e provê uma interface rica para uso em vários serviços web baseados em RESTfull.

Na FIWARE, o desenvolvedor escolhe os GEs que serão usados pela sua aplicação. O GE mais importante e que deve estar obrigatoriamente presente em todo desenvolvimento FIWARE é o *Orion Context Broker*. Esse componente é responsável pelo gerenciamento das informações de contexto, implementando um modelo *Publish/Subscribe* para a disseminação de dados dentro da plataforma. Na FIWARE, entidades de contexto representam os dispositivos IoT, enquanto informações de contexto são dados coletados pelos dispositivos. Entretanto, o *Orion Context Broker* armazena apenas o último valor publicado para uma entidade, sendo necessário o uso de outros GEs para armazenamento histórico dos dados.

Além do *Orion Context Broker*, componentes GEs de armazenamento e de interface com dispositivos são comumente utilizados. Particularmente, para armazenamento, os componentes *STH Comet* e *Cygnus* são comumente usados. O primeiro é responsável pelo armazenamento de informação de contexto de curto prazo, enquanto o segundo é responsável pelo armazenamento de longo prazo.

GEs para interface com dispositivos são denominados *IoT-Agents*. Esses componentes são a interface de entrada e conexão dos mais diversos sensores e atuadores com a plataforma. Eles são responsáveis pela conversão de protocolos específicos, usados nos dispositivos IoT, para chamadas NSGI, usadas pela plataforma (*Orion Context Broker*). Um *IoT-Agent* provê suporte a um protocolo de comunicação específico, como LWM2M, MQTT, HTTP, ou AMQP. Diversos *IoT-Agents* são suportados nativamente pela plataforma. Adicionalmente, a FIWARE provê uma biblioteca para que o desenvolvedor escreva seu próprio *IoT-Agent*. Dessa forma, novos dispositivos ainda não suportados nativamente podem ser utilizados e associados à plataforma. A Figura 1 ilustra uma implantação básica da plataforma FIWARE com um componente *Orion Context Broker* e vários componentes *IoT-Agents*. Além dos GEs descritos acima, a FIWARE disponibiliza componentes para processamento, análise e visualização de informação de contexto, bem como para serviços de segurança.

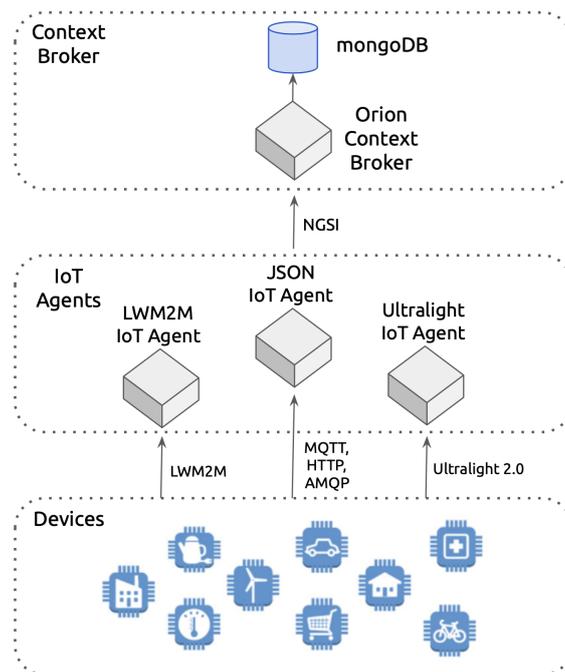


Figura 1. Arquitetura da plataforma FIWARE (extraído de [Araujo et al. 2019]).

Para criar uma aplicação FIWARE, primeiramente, os componentes da plataforma necessários ao desenvolvimento da aplicação devem ser implantados. Em seguida, é necessário criar as entidades de contexto no componente *Orion Context Broker* e configurar os elementos da aplicação nos componentes instalados. Na maioria das vezes, isso inclui criar um serviço que represente a aplicação nos componentes. Em seguida, é necessário registrar os dispositivos IoT na plataforma usando o componente *IoT-Agent*. O registro do dispositivo o associa a uma entidade de contexto na plataforma, de forma que, dados coletados pelo dispositivo serão usados para popular propriedades das entidades de contexto. Finalmente, depois que os elementos de aplicação são configurados na plataforma, o código fonte que executará nos dispositivos IoT pode ser desenvolvido usando informações sobre os elementos da aplicação (e.g., identificador do dispositivo e identificador do serviço associado à aplicação).

3. FIWARE-Lab@RNP

O FIWARE-Lab@RNP [Dantas et al. 2019] é um laboratório virtual para prototipação e experimentação de aplicações IoT baseadas na plataforma FIWARE, sendo o ambiente disponibilizado e gerenciado sobre uma infraestrutura computacional da RNP. O FIWARE-Lab@RNP adota um modelo PaaS (*Platform-as-a-Service*) de computação em nuvem, onde recursos da plataforma FIWARE são disponibilizados através da Internet de forma transparente. Dessa forma, usuários do FIWARE-Lab@RNP podem desenvolver e experimentar aplicações IoT fazendo uso de GEs da FIWARE, sem a necessidade de implantar e operar uma instância própria da plataforma em seus ambientes de trabalho.

De maneira geral, as funcionalidades do ambiente FIWARE-Lab@RNP podem ser agrupadas em duas tarefas principais: i) o gerenciamento de instâncias de componentes GEs; e ii) a configuração dos elementos de uma aplicação nesses componentes.

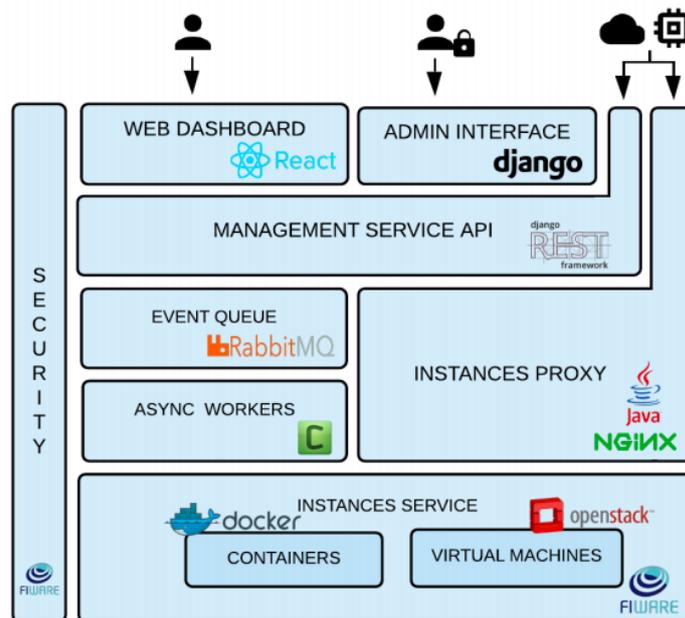


Figura 2. Arquitetura do FIWARE-Lab@RNP (extraído de [Dantas et al. 2019]).

Para prover essas tarefas, o FIWARE-Lab@RNP é estruturado em um conjunto de componentes arquiteturais, como mostrado na Figura 2. O componente *Management Service API* é responsável por processar as requisições recebidas pelo laboratório virtual. Essas requisições são enviadas ao laboratório através de um portal Web (geralmente utilizado pelos usuários finais) ou externamente através de chamadas RESTfull providas pela plataforma.

O componente *Event Queue* implementa um serviço de mensagem assíncrona, de forma que as requisições enviadas ao laboratório virtual possam ser processadas sem a necessidade de bloquear o cliente. Dessa forma, requisições enviadas ao laboratório virtual são armazenadas assincronamente pelo componente *Event Queue*.

O processamento de grande parte das requisições enviadas ao laboratório virtual envolve a criação de instâncias de componentes GEs. Essas instâncias, por sua vez, são armazenadas pelo componente *Instances Service*. Este componente permite a execução de instâncias com base em imagens pré-configuradas de componentes GEs cuja configuração é finalizada após a criação da instância. O FIWARE-Lab@RNP permite a criação de dois tipos de instâncias: locais e globais. Instâncias locais são criadas pelos usuários finais e acessadas apenas por eles. Instâncias locais são providas por tecnologias de virtualização baseadas em contêineres (mais especificamente docker). Instâncias globais, por sua vez, são criadas pelo administrador e compartilhadas por todos os usuários que têm acesso ao laboratório virtual.

Como ambos tipos de instâncias suportados pelo ambiente se baseiam em tecnologias de virtualização, torna-se necessário mapear as portas dos servidores que as executam para as portas internas das instâncias virtualizadas. Isso é feito pelo componente *Instances Proxy*. Por fim, o componente *Web Dashbord* oferece uma interface gráfica para usuários finais e administradores. Através da *dashboard*, o usuário final pode criar,

de forma simples, instâncias de componentes da plataforma FIWARE a partir de imagens pré-configuradas. O usuário também pode definir, de forma simplificada, os elementos da aplicação, ou seja: definição do serviço associado à aplicação, registro dos dispositivos IoT utilizados, criação das entidades de contexto e criação de subscrições sobre informações das entidades de contexto.

4. Aplicação Piloto

Para avaliar o quanto um ambiente como o FIWARE-Lab@RNP pode facilitar o desenvolvimento de aplicações baseadas na plataforma FIWARE, implementamos uma aplicação piloto de duas formas: i) instanciando manualmente os componentes da plataforma; e ii) usando as facilidades do FIWARE-Lab@RNP. Mais especificamente, nossa aplicação piloto envolve o contexto de Escritórios Inteligentes voltada para o controle de utilização de salas de reunião. Os seguintes requisitos foram definidos para esse piloto:

- A aplicação deve interagir com o Sistema de Agendamento de Salas da RNP e recuperar todos os agendamentos para uma dada sala no dia;
- Após obter as informações de agendamento, a aplicação deve iniciar o monitoramento da sala através de sensores de monitoramento;
- Após um intervalo de tempo configurável, se a sala estiver vazia, a aplicação deve enviar um e-mail para o administrador do ambiente de agendamento indicando que a sala está vazia, podendo ser alocada para outra reunião.

Para atender esses requisitos, projetamos um piloto em que nós Raspberry Pi, conectados a sensores ultrassônicos, são espalhados em alguns pontos da sala de reunião. Esses nós se conectam à Internet através de um roteador WiFi. Uma aplicação de monitoramento executa em cada nó Raspberry Pi coletando dados do sensor e enviando os dados coletados para a plataforma FIWARE, a qual executa remotamente. A plataforma FIWARE, por sua vez, fornece mecanismos para a distribuição dos dados coletados de forma segura e acurada. Neste caso específico, o consumidor interessado nas informações de contexto da sala é uma aplicação de notificação, a qual envia ao administrador do ambiente mensagens relacionadas à disponibilidade da sala e o *status* de funcionamento de cada nó instalado no ambiente. A Figura 3 ilustra os principais elementos de hardware e software que compõem esse piloto.

Em termos de componentes de software, implementamos três aplicações para implantar o piloto descrito na Figura 3. A primeira é uma aplicação Relógio, responsável por interagir com o Sistema de Agendamento de Salas da RNP e obter dados de agendamento para a sala no dia corrente. Essas consultas acontecem em intervalos de vinte minutos, para o caso de haver alguma alteração na base de dados do Sistema de Agendamento de Salas da RNP. Adicionalmente, essa aplicação verifica, a cada minuto, se alguma reserva foi iniciada. A segunda é uma aplicação de Monitoramento, responsável por coletar os dados dos sensores ultrassônicos acoplados ao nó de sensoriamento (Raspberry Pi). Essa aplicação também coleta informações sobre o *status* do nó. Por fim, a terceira aplicação é a de Notificação, a qual consome e processa os dados coletados pelos nós de sensoriamento, com a finalidade de identificar presença ou ausência de pessoas no ambiente.

A Figura 4 ilustra a implementação da aplicação piloto usando a plataforma FIWARE sem os recursos do FIWARE-Lab@RNP. Chamaremos tal implementação de versão FIWARE. Além das três aplicações descritas acima, implementamos também três

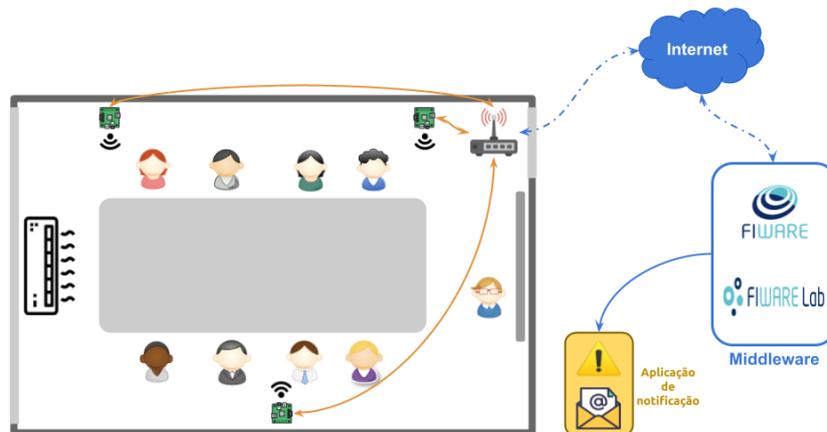


Figura 3. Elementos de hardware e software utilizados no piloto.

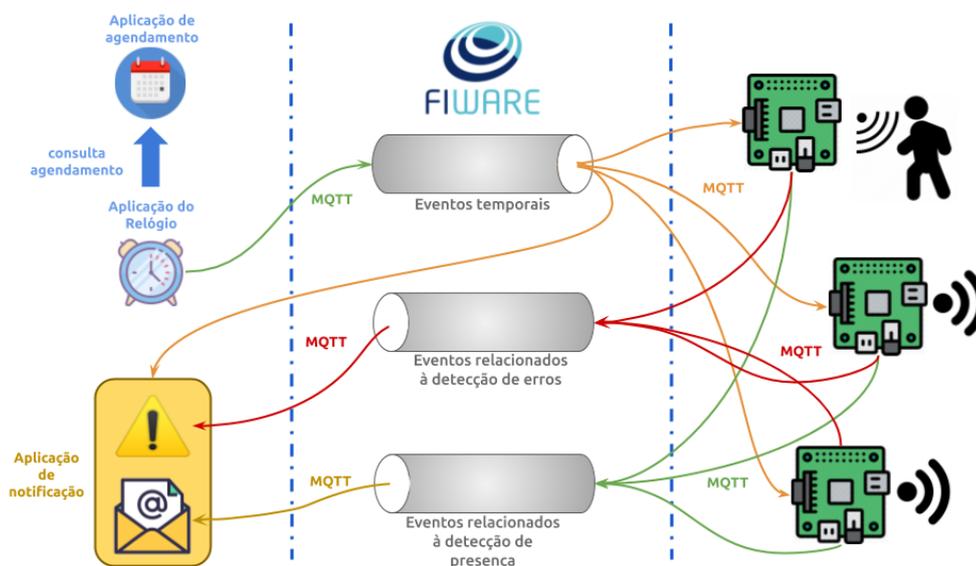
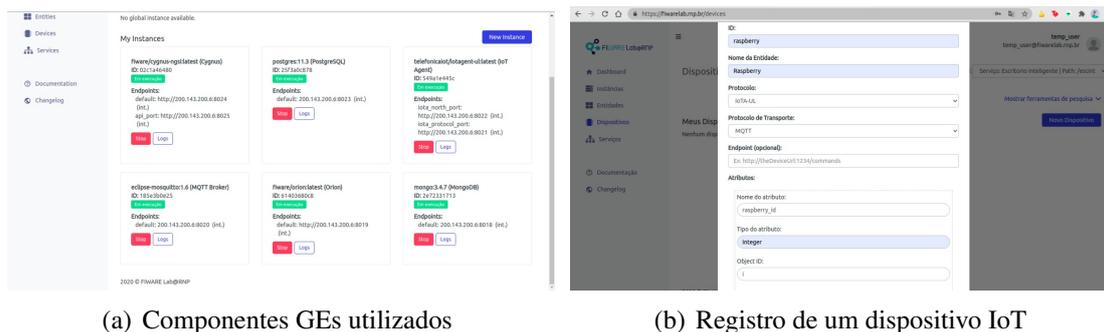


Figura 4. Implementação da aplicação piloto na FIWARE.

tópicos de comunicação dentro da plataforma FIWARE. O primeiro tópico, denominado Eventos Temporais, recebe mensagens (eventos) referentes às reservas da sala. Esses eventos são posteriormente consumidos pelas aplicações de Monitoramento executando nos nós de sensoriamento. O segundo tópico recebe mensagens sobre o *status* dos nós de sensoriamento, as quais são consumidas pela aplicação de Notificação. Finalmente, o terceiro tópico recebe as mensagens enviadas pelos nós de sensoriamento referentes à detecção de pessoas no ambiente. Essas mensagens são posteriormente processadas pela aplicação de Notificação para concluir se há ou não pessoas na sala e se a reunião deve ser considerada finalizada. Além das aplicações e tópicos descritos acima, dois componentes GEs foram usados nesta versão do piloto, a saber, os componentes *Orion Context Broker* e *IoT Agent MQTT*.

De maneira similar, a aplicação piloto implementada a partir dos recursos do FIWARE-Lab@RNP, denominada versão FIWARE-Lab, envolve os mesmos componen-



(a) Componentes GE utilizados

(b) Registro de um dispositivo IoT

Figura 5. Dashboard do ambiente FIWARE-Lab@RNP.

tes de software (aplicação Relógio, de Monitoramento e de Notificação) e os mesmos tópicos de comunicação implementados na versão FIWARE. Entretanto, a implementação na versão FIWARE-Lab envolve três componentes GEs: *Orion Context Broker*, *IoT Agent* MQTT e *Cygnus*. Dessa forma, a versão FIWARE-Lab mantém um histórico dos valores enviados pelos sensores, enquanto a versão FIWARE mantém apenas o último valor.

A Figura 5 ilustra algumas telas da *dashboard* capturadas durante o desenvolvimento da versão FIWARE-Lab. A primeira tela (Figura 5(a)) ilustra os componentes GEs instanciados, enquanto a segunda tela (Figura 5(b)) mostra o registro de um nó de sensoriamento usando a *dashboard*.

5. Avaliação Experimental

Implementamos e validamos a aplicação piloto nas duas versões descritas na Seção 4, ou seja, FIWARE e FIWARE-Lab.

Na versão FIWARE, os componentes FIWARE e as aplicações Relógio e de Notificação foram instanciados em uma máquina virtual com 2 vCPU, memória RAM de 8GB, disco de 20GB e o sistema operacional Ubuntu 18.04. Essa máquina virtual foi implantada na infraestrutura da RNP. A aplicação de Monitoramento foi instanciada em cada nó de sensoriamento. Neste trabalho, usamos dois microcomputador Raspberry Pi 3B+, com interface de rádio WiFi, como nó físico de sensoriamento. Como não foi possível implantar a aplicação em um escritório físico da RNP, devido à situação pandêmica acarretada pelo do Novo Coronavírus (COVID- 19), os nós de sensoriamento foram implantados em uma sala do Instituto de Informática da Universidade Federal de Goiás. Esses nós estão conectados à VM na infraestrutura da RNP através da Internet. Para o sensoriamento de presença, usamos o sensor ultrassônico, modelo HC-SR204, integrado ao nó Raspberry Pi.

A versão FIWARE-Lab, foi desenvolvida e implantada no ambiente FIWARE-Lab@RNP. Entretanto, na versão FIWARE-Lab, as aplicações Relógio e de Notificação foram instanciadas localmente em um notebook e o acesso ao laboratório virtual foi feito via *Virtual Private Network* (VPN). Contudo, devido a alguns restrições impostas pelo cliente VPN, optamos por usar nós de sensoriamento virtuais nos testes com a versão FIWARE-Lab. Dessa forma, os nós virtuais simulam os nós Raspberry Pi e a aplicação de Monitoramento. Em ambas as versões, os dados de cada reserva, obtidos pela aplicação Relógio, foram simulados e não obtidos diretamente da base de dados da RNP. Entretanto, a integração com o Sistema de Agendamento de Salas da RNP já foi realizada.

Validamos as duas versões da aplicação piloto em diversas situações. Em ambas as versões e em todas as situações testadas, a aplicação piloto se comportou como esperado. A Figura 6 ilustra os *logs* gerados pela Aplicação de Notificação (na versão FIWARE) durante o monitoramento do ambiente, para o caso em que o escritório é reservado e a reunião acontece do início ao fim da reserva. Durante o período de monitoramento, todos os nós implantados no ambiente identificam a presença de pessoas na sala. A sala está reservada no intervalo das 11h35 até as 12h do dia 29 de Dezembro de 2020.

```
2020-12-29 11:34:39.823041 - INFO - Aplicação em execução.
-----
2020-12-29 11:39:42.751997 - INFO - Monitoramento iniciado.
2020-12-29 11:55:00.672656 - INFO - Monitoramento encerrado.
-----
```

Figura 6. Logs da Aplicação de Notificação - Caso 1.

Resultados

De maneira geral, percebemos que a plataforma FIWARE é uma ótima solução para aplicações simples. Entretanto, a plataforma completa é uma solução complexa e vários componentes GEs estão disponíveis no seu catálogo. Como a plataforma não apresenta uma forma concisa e integrada de implantar os GEs, fica a cargo do desenvolver escolher os GEs, implantá-los e integrá-los de forma correta. Essa operação é mais simples para os GEs básicos da plataforma, mas mais complexas para os GEs que proveem serviços mais elaborados.

O FIWARE-Lab@RNP, por sua vez, oferece um ambiente Web e um conjunto de funcionalidades que, de fato, facilitam a construção de aplicações IoT baseadas na plataforma FIWARE. A instanciação de componentes a partir de imagens pré-configuradas e um ambiente gráfico para criar, configurar e interagir com essas instâncias contribui significativamente para que o ambiente FIWARE-Lab@RNP alcance seu objetivo. Este fato ficou evidenciado pela facilidade de adicionar o componente de armazenamento histórico na versão FIWARE-Lab. Nossa tentativa de instalar esse mesmo componente manualmente na versão FIWARE, seguindo as instruções disponíveis na documentação, não foi bem sucedida. Em nossas tentativas, o processo de instalação levou mais de 40 minutos, após o qual um erro pouco intuitivo foi reportado. O mesmo foi observado usando a opção de instalação via contêiner. Outra vantagem do FIWARE-Lab@RNP é o tempo necessário para criar e executar uma instância de um componente FIWARE neste ambiente. Esse tempo é significativamente menor que aquele observado quando essas tarefas são realizadas de forma manual.

Entretanto, a documentação oficial do ambiente FIWARE-Lab@RNP ainda contém poucas informações. De fato, a criação de algumas instâncias e dispositivos, durante a execução deste trabalho, só foi possível com a ajuda direta da equipe de desenvolvimento do ambiente. Finalmente, é importante ressaltar que, vários problemas enfrentados durante a execução deste trabalho foram relacionados ao acesso ao ambiente FIWARE-Lab@RNP. Esses problemas poderiam ser evitados com um sistema federado de identidades como a CAFé.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou um estudo comparativo envolvendo a plataforma FIWARE e o ambiente FIWARE-Lab@RNP. Especificamente, desenvolvemos e validamos uma aplicação piloto para escritórios inteligentes usando as duas abordagens. Resultados confirmam a que a abordagem usando o FIWARE-Lab@RNP facilita significativamente a construção de aplicações baseadas na FIWARE.

Como trabalhos futuros, pretendemos fazer uso de outros GEs fornecidos pela plataforma FIWARE, o que possibilitará expandir as funcionalidades e o suporte às aplicações desenvolvidas. Além disso, iremos avaliar outras plataformas de middleware IoT, como por exemplo, a Dojot. A Dojot possui um ambiente virtualizado disponibilizado pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) [CPqD 2021] para experimentação com aplicações IoT. Nele, os desenvolvedores podem fazer uso dos microsserviços disponibilizados pela plataforma, tais como protocolos de comunicação, modelagem de dispositivos, histórico de dados, entre outros.

Referências

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.
- Araujo, V., Mitra, K., Saguna, S., and Åhlund, C. (2019). Performance evaluation of fiware: A cloud-based iot platform for smart cities. *Journal of Parallel and Distributed Computing*, 132:250 – 261.
- CPqD (2021). Dojot. Disponível em: <http://www.dojot.com.br>. Acessado em 24/05/2021.
- da Cruz, M. A., Rodrigues, J. J., Sangaiah, A. K., Al-Muhtadi, J., and Korotaev, V. (2018). Performance evaluation of iot middleware. *J. Netw. Comput. Appl.*, 109(C):53–65.
- Dantas, L., Cavalcante, E., and Batista, T. (2019). A development environment for fiware-based internet of things applications. In *Proceedings of the 6th International Workshop on Middleware and Applications for the Internet of Things*, pages 21–26.
- Domingue, J., Galis, A., Gavras, A., Zahariadis, T., Lambert, D., Cleary, F., Daras, P., Krco, S., Müller, H., and Li, M. e. a. (2011). The future internet. future internet assembly 2011: Achievements and technological promises, 6656. Disponível em: <https://link.springer.com/book/10.1007/978-3-642-20898-0>. Acessado em 10/08/2020.
- FIWARE (2020). fiware. Disponível em: <https://www.fiware.org/>. Acessado em 10/08/2020.
- Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, 3(1):70–95.
- Singh, K. J. and Kapoor, D. S. (2017). Create your own internet of things: A survey of iot platforms. *IEEE Consumer Electronics Magazine*, 6(2):57–68.
- Trindade, C., Souza, A., Pereira, J., Oliveira, J., Cacho, N., Batista, T., Cavalcante, E., Lopes, F., Maia, M., and Loss, S. (2017). Usando o fiware para desenvolvimento de aplicações de cidades inteligentes.