Dashborad SmartRoom: Web das Coisas para gerenciamento de salas em um Campus Universitário

Jeferson de Almeida, Ramon Costa, Cristhian Carvalho,
Nailton de Andrade Jr, Leandro Andrade e Cássio Prazeres
WISER: Web, Internet and Intelligent Systems Research Lab
Departamento de Ciência da Computação - Universidade Federal da Bahia
Salvador, Bahia, Brasil
{jefersonlimaa, ramondiascosta, cristhian, nailtonjr, leandrojsa, prazeres}@dcc.ufba.br

ABSTRACT

Desde a sua criação, a Web tem evoluído da Web de Documentos inicial à Web atual como plataforma e com foco na interação entre as pessoas. Entretanto, vislumbra-se ainda uma evolução na Web, cujo foco deixará de ser apenas as pessoas. Essa evolução, que visa promover a participação de dispositivos físicos na Web, demanda por soluções para integração de ambientes, como casas e laboratórios, que possuem requisitos peculiares de controle de acesso. Nesse contexto, este trabalho propõe um modelo de aplicação para Salas Inteligentes de um Campus Universitário. Como resultado, foi desenvolvida uma aplicação Dashboard SmartRoom e com um estudo de caso aplicado no laboratório WISER-UFBA.

Keywords

Web das Coisas; Sala Inteligente; Monitoramento

1. INTRODUÇÃO

Em meio a evolução da web 2.0, o conceito de Web das Coisas (*Web of Things* - WoT como será referenciada neste trabalho) começou a ser desenvolvido com a proposta de conectar, por meio de protocolos padrões da Web, todas as coisas à Internet. Coisas conectadas são objetos que vão desde equipamentos do dia a dia, tais como geladeiras, microondas, ar condicionado e outros, a sensores de temperatura e lâmpadas comuns. Dessa forma, a WoT é uma rede de objetos com Internet habilitada, que, por meio de Serviços Web, possibilita a integração de informações a esses dispositivos e o acesso programático aos mesmos [4].

Dado que a Internet das Coisas (Internet of Things - IoT) é um conjunto de diversas tecnologias que oferecem conectividade a dispositivos físicos [2], a WoT, ao se utilizar dos protocolos e padrões da Web, reduz a complexidade envolvida no desenvolvimento de aplicações que usem dos recursos físicos disponibilizados na Internet [4].

Esse novo paradigma (WoT) pode permitir que organizações e domicílios possam ser gerenciados remotamente. Entretanto, é necessário a construção de uma arquitetura própria que se adeque às particularidades de cada ambiente.

Alguns ambientes que possuem requisitos de controle de acesso nem sempre podem estar sendo vigiados por humanos, seja por questões financeiras como, por exemplo, na redução de gastos com serviços de vigilância, ou pelo grau de insalubridade do ambiente como, por exemplo, em salas de produtos químicos, dentre outros motivos.

Nesse contexto, este trabalho visa explorar a infraestrutura da Web das Coisas para poder interligar, de forma transparente, a camada de software da aplicação ao meio físico provendo, assim, escalabilidade à aplicação e baixo nível de acoplamento com o meio físico através do uso de Serviços Web.

Portanto, o objetivo deste trabalho é aplicar os conceitos de WoT para o universo de um Campus Universitário. O estudo de caso será feito através de uma aplicação chamada Dashboard SmartRoom, que é capaz de monitorar e controlar uma sala, através da disponibilização dos dispositivos físicos da rede na Web.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os protocolos IoT e a arquitetura WoT utilizados neste trabalho; a Seção 3 apresenta a ferramenta proposta; por fim, a Seção 5 apresenta algumas considerações finais.

2. INTERNET E WEB DAS COISAS

Atualmente, mais de uma década depois da introdução do termo "Internet of Things" [1], a comunidade de pesquisa e a indústria têm feito significantes progressos em diversas direções, incluindo o desenvolvimento de novas arquiteturas, plataformas, protocolos e aplicações (por exemplo, ambientes inteligentes).

Para suprir a necessidade de interoperabilidade entre dispositivos, protocolos voltados para a Internet das Coisas e comunicação M2M (máquina-máquina) foram propostos na literatura (ver Seção 2.1).

Em termos de arquitetura, baseando-se na especificação da arquitetura de Guinard [4], Prazeres (2013) [6] propôs uma arquitetura dividida em componentes (ver Seção 2.2).

2.1 Protocolos para Internet das Coisas

O termo M2M refere-se a tecnologias de rede, informação e comunicação que permitem a comunicação, com e sem fio, entre dispositivos físicos eletrônicos. O MQTT (Message Queue Telemetry Transport) e CoAP (Constrained Application Protocol) são protocolos criados justamente para suprir essas necessidades. Os dois protocolos, quando utilizados em conjunto, têm como características principais: possuírem padrões abertos, serem adequados para ambientes com restrições de HTTP, fornecerem mecanismos para comunicação assíncrona entre dispositivos, executar sobre o IP e ter um conjunto de implementações disponíveis. Para este trabalho, o protocolo utilizado foi o MQTT, descrito a seguir.

O MQTT é um protocolo de transporte de mensagens, baseado em cliente/servidor e no padrão de *publish/subscribe*.

O protocolo é focado em ambientes com dispositivos limitados (pouco poder de processamento e baixo consumo de energia) e baixa largura de banda disponível. O MQTT utiliza o TCP/IP como protocolo de transporte, é independente do protocolo de aplicação utilizado e possui mecanismos de notificação de desconexão entre clientes [5].

Apesar de suas muitas vantagens, o MQTT não possui uma padronização para as mensagens que são distribuídas aos diversos clientes, diferentemente do HTTP, por exemplo, que possui métodos específicos para requisição e envio de informação.

Dessa forma, com o intuito de facilitar a programação dos dispositivos, o *framework* TATU(*The Acessible Things Universe*), desenvolvido pelos autores deste artigo, propõe a utilização de um protocolo para o controle dos dispositivos. O TATU possui uma biblioteca na linguagem C++ chamada TATUDevice, que é a biblioteca utilizada para a programação dos dispositivos neste trabalho.

O protocolo utilizado pelo TATU, que é chamado TPI(*Thing Protocol for Internet*), estabelece mensagens simples para alteração dos dispositivos e requisições de informações armazenadas nos mesmos.

2.2 Arquitetura para a Web das Coisas

A arquitetura apresentada na Figura 1, proposta por Prazeres (2013) [6], utiliza um barramento de serviços (*ESB - Enterprise Service Bus*) como infraestrutura para configuração, implantação, descoberta, composição, monitoramento, utilização e compartilhamento de dispositivos na Web das Coisas. Tal como detalhado a seguir, essa arquitetura é dividida em 5 componentes maiores, que são: *Communication*; *Enterprise Service Bus*; *OpenID Connect*; *Web of Things Applications*; e *Semantic Web Services*.

A camada mais baixa (parte 1 da Figura1) dessa arquitetura é onde se encontram as formas de comunicação primária dos dispositivos físicos do sistema, tais como Ethernet, ZigBee, WIFi e outros.

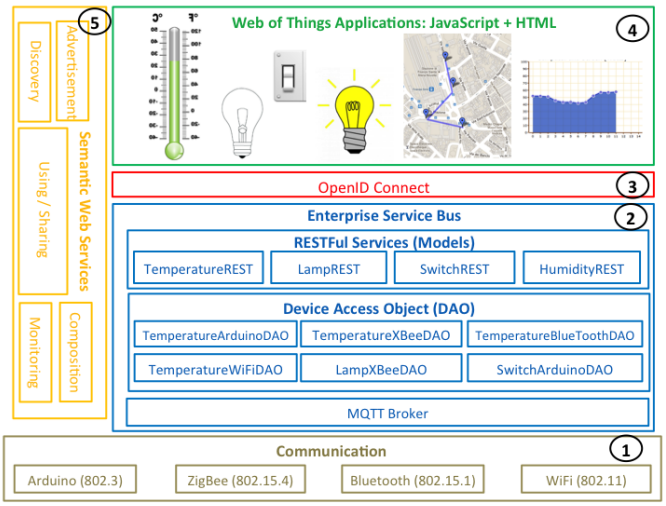

Figura 1: Arquitetura para a Web das Coisas [6]

Na próxima camada (parte 2 da Figura1), existe o barramento de serviços (ESB), que tem como objetivo facilitar o uso dos recursos por aplicações na Web, ao permitir a configuração e instalação de Serviços Web Restful.

Neste trabalho, o ESB é divido em duas partes, a primeira é o conjunto de DAO's(*Device Access Object*) e a segunda são os Serviços Web Restful. Um DAO é basicamente um *driver*, que serve como interface entre um tipo de comunicação específica de um dispositivo e a comunicação mais genérica da infraestrutura IoT implementada. Por sua vez, os Serviços Web Restful servem como a forma de disponibilização destes dispositivos como recursos na Web.

A camada da aplicação se encontra na parte mais externa dessa arquitetura (parte 4 da Figura1) e através dela os usuários podem acessar funcionalidades que foram desenvolvidas baseadas nos recursos disponibilizados pelo ESB.

Na Figura1 ainda podem ser vistos dois outros componentes, partes 3 e 5, que não são tratados neste artigo e são utilizados, respectivamente, para prover segurança e semântica para as aplicações WoT.

3. A FERRAMENTA SMARTROOM

Os dois principais desafios no desenvolvimento da ferramenta Dashboard SmartRoom são: não permitir outros tipos de acesso aos dispositivos, que não seja via aplicação Web e autenticada; possibilitar que a aplicação cliente (Web) tenha acesso aos dispositivos via Serviços Web Restful. Para tratar esses desafios foi implantada uma infraestrutura baseada em VPN (Seção 3.1) e os dispositivos foram implementados como componentes Web em uma Dashboard (Seção 3.2).

3.1 Infraestrutura de Implantação

De acordo com a Figura 2, a infraestrutura implantada é composta de quatro componentes principais: servidor Web; gateways WoT-ESB; redes WoT privadas; e os dispositivos.

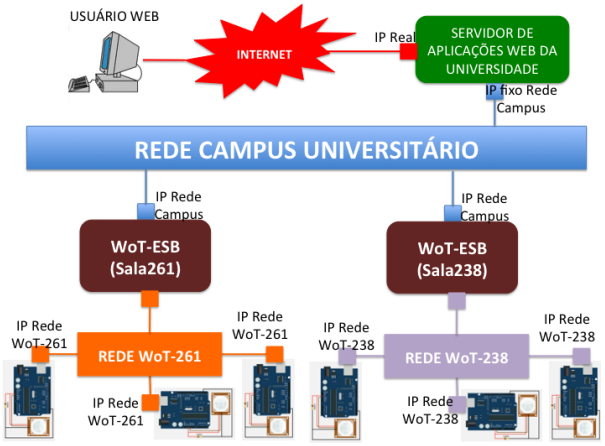

Figura 2: Rede WoT do campus universitário.

Para possibilitar que as aplicações Web tenham acesso aos dispositivos nas redes privadas, uma solução baseada em VPN (*Virtual Private Network*) foi implantada. Nesse caso, a VPN se torna necessária devido ao fato das redes WoT privadas da universidade serem protegidas por um *firewall*, que restringe boa parte das comunicações necessárias ao funcionamento do sistema, ao mesmo tempo em que protege os dispositivos de acessos externos indevidos.

Conectados à VPN estão os dois componentes servidor Web e gateways WoT-ESB. O primeiro é responsável por manter a aplicação Web Dashboard SmartRoom, que tem

acesso aos dispositivos via Serviços Web Restful. O segundo é um *gateway* para Web das Coisas, onde foram implantados os Serviços Web Restful.

O *gateway* pode se comunicar normalmente com os dispositivos, sem a necessidade do VPN, por estarem na mesma rede privada WoT. Neste trabalho, o *gateway* utilizado foi um Raspberry Pi modelo B, onde foram implantados um *broker* MQTT e um ESB.

No gateway, o MQTT é o responsável pela comunicação entre os serviços e os dispositivos da sala. Um driver MQTT pertencente ao *framework* TATU é utilizado como interface para os serviços se comunicarem com os dispositivos da sala.

Por sua vez, os dispositivos da sala são compostos por microcontroladores Arduino[1], que estão conectados aos dispositivos e estão programados através da biblioteca TATUDevice.

3.2 Dispositivos Implementados

A ferramenta Dashboard SmartRoom é em sua maioria executada do lado do cliente (navegador Web) e todos os comandos utilizados são métodos HTTP via Serviços Web Restful.

Para possibilitar a realização de requisições aos Serviços Web Restful, foram utilizadas requisições AJAX (*Asynchronous Javascript and XML*), que são uma categoria de requisições Javascript realizadas de forma assíncrona. Esses serviços, por sua vez, fazem as requisições apropriadas aos dispositivos da sala.

Os serviços esperam por uma resposta do dispositivo, que é dada em formato JSON, tratam essa resposta e enviam um retorno apropriado, também em formato JSON, para a aplicação. Baseando-se na resposta dada, a aplicação atualiza a interface. Com o intuito de evitar possíveis sobrecargas sobre os dispositivos da sala, foi utilizado um cache, que faz requisições eventuais ao sistema e armazena o resultado para devolvê-los quando forem requisitados pela aplicação.

Tal como ilustrado na Figura 3, através de um painel (Dashboard) em uma aplicação na Web, é possível ter acesso aos dispositivos da sala. Dentre os sensores e atuadores utilizados no Dashboard: a Figura 4 apresenta o sensor de temperatura e umidade; a Figura 5 apresenta lâmpadas controladas por relés; e a Figura 6 apresenta um emissor infravermelho usado como controle do ar-condicionado.

Um Arduino controla o sensor de temperatura e umidade e o relé da lâmpada da porta (Figuras 4 e 5). Esse Arduino foi programado com a biblioteca TATUDevice de forma que pudesse ser capaz de gerenciar a comunicação dos vários dispositivos nele implantados. Além disso, o Arduino possui conexão Ethernet e se comunica com os serviços do *gateway* através de um tópico no broker MQTT.

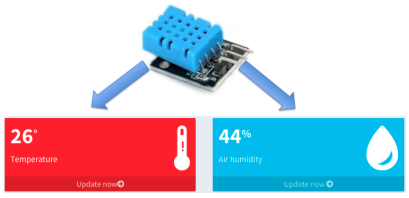

Figura 4: Sensor de umidade e temperatura.

Sempre que uma requisição feita pelo serviço chega através do *broker* MQTT, o Arduino trata a mensagem e, se for uma requisição de desligar ou ligar a lâmpada, executa a ação correspondente e devolve uma resposta de confirmação. Caso a requisição seja de obter uma informação de um sensor ou ainda sobre o estado da lâmpada, o Arduino devolve, através de um JSON, a informação nele armazenada.

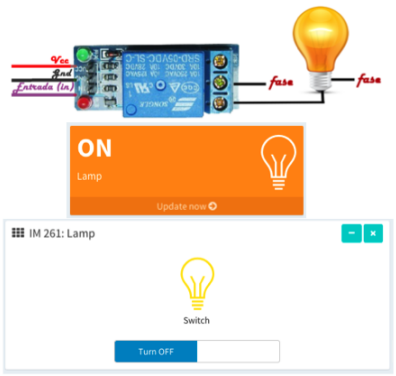

Figura 5: Relé para acender e/ou apagar a lâmpada da sala.

O dispositivo que serve como controle do ar-condicionado (Figura 6) é composto por um outro Arduino, conectado à rede via Wifi. Esse arduino foi programado de forma que pudesse emitir os comando de controle do ar-condicionado via um emissor infravermelho implantado nele.

Figura 6: Controle do ar-condicionado da sala via infravermelho.

Além desses dispositivos, ainda existe uma câmera de segurança, tal como ilustrado na Figura 7. Essa câmera permite movimentos (esquerda, direita, cima e baixo) que são controlados através da interface da aplicação Dashboard SmartRoom. Nesse caso, um software específico, provido pela própria câmera, foi encapsulado em um Serviço Web Restful, mantendo a arquitetura apresentada na Seção 2.

Por fim, com o objetivo de proteger os dispositivos de acessos indevidos, o acesso à aplicação Dashboard SmartRoom (Figura 3) é realizado apenas por usuários autenticados via Twitter, como ilustrado na Figura 8. Dessa forma, apenas seguidores autorizados do Laboratório de Pesquisa que mantém a sala podem acessar a aplicação.

[1]http://www.arduino.cc/

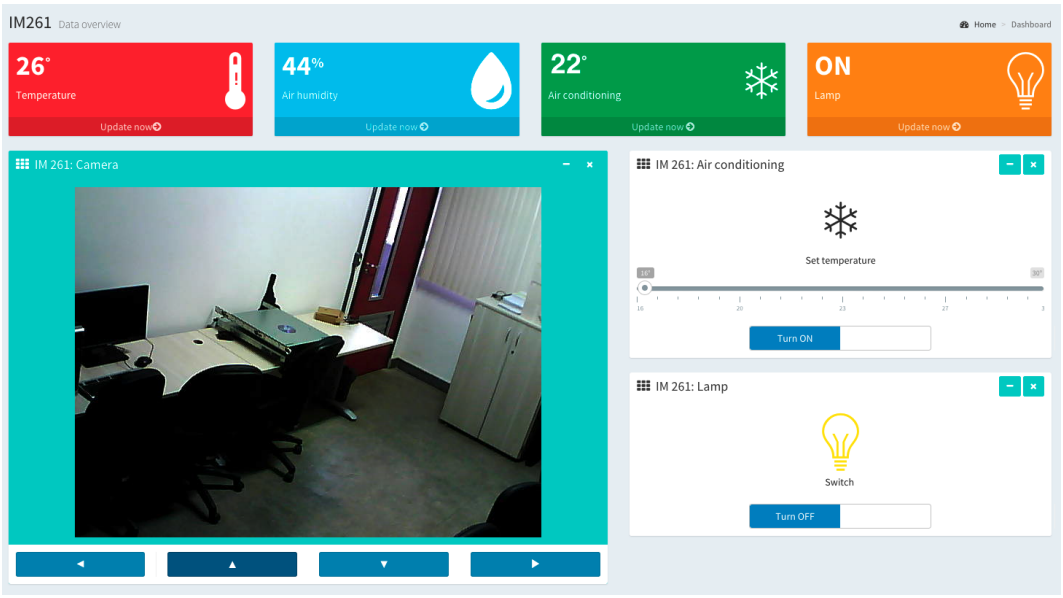

Figura 3: Aplicação Web para Sala Inteligente.

Figura 7: Câmera com rotação vertical e horizontal.

Figura 8: Autenticação via twitter.

4. TRABALHOS RELACIONADOS

O uso de um barramento de serviços como base para disponibilização de recursos do mundo real, é apresentada por Andrade et al. (2014) [3].

Vujovic e Maksimovic [7] utilizam o Raspberry Pi como um nó de uma Rede de Sensores para automação e controle em tempo real de casas inteligentes via Web.

Nos trabalhos de Andrade et al. (2014) [3] e Vujovic e Maksimovic [7] tem-se o uso de um *middleware* que gerencia o acesso aos recursos do mundo real.

5. CONSIDERAÇÕES FINAIS

Com a extensão e barateamento de dispositivos de automação que possuam comunicação com a Internet, é natural que se tenha um aumento de volume na implementação de infraestruturas conectadas à Web em residências e organizações. Entretanto, diversos desafios ainda precisam ser abordados e tratados para disponibilizar esses dispositivos na Web.

Os dois principais desafios abordados e tratados no desenvolvimento da ferramenta Dashboard SmartRoom foram: não permitir outros tipos de acesso aos dispositivos, que não seja via aplicação Web e autenticada; possibilitar que a aplicação cliente (Web) tenha acesso aos dispositivos via Serviços Web Restful.

6. REFERÊNCIAS

[1] K. Ashton. That 'internet of things' thing. http://www.rfidjournal.com/articles/view?4986, June 2009.

[2] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):2787–2805, Oct. 2010.

[3] N. V. de Andrade Junior, D. B. Bastos, and C. V. S. Prazeres. Web of things: Automatic publishing and configuration of devices. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, WebMedia '14, pages 67–74, New York, NY, USA, 2014. ACM.

[4] D. Guinard. *A Web of Things Application Architecture – Integrating the Real-World into the Web*. Ph.d., ETH Zurich, 2011.

[5] D. Locke. Mq telemetry transport (mqtt) v3. 1 protocol specification. http://www.ibm.com/developerworks/ webservices/library/ws-mqtt/index.html, 2010.

[6] C. V. S. Prazeres. Projeto "barramento de serviços como infraestrutura para a web das coisas". http://homes.dcc.ufba.br/ prazeres/ESB-WoT.pdf, Agosto 2013.

[7] V. Vujovic and M. Maksimovic. Raspberry pi as a sensor web node for home automation. *Computers & Electrical Engineering*, 0(0):1–19, 2015.