

ARQUITETURA GERAL DE UM SISTEMA PARA AUTOMAÇÃO RESIDENCIAL UTILIZANDO PLATAFORMAS ABERTAS

DANIEL ARAÚJO*, JUDSON COSTA*, DIEGO R. C. SILVA*, MARCELO B. NOGUEIRA*, MARCONI C. RODRIGUES*

*UFRN Campus - Lagoa Nova
Natal, RN, Brazil

Emails: danielaraujo@bct.ect.ufrn.br, judson.scosta@bct.ect.ufrn.br,
diego@ect.ufrn.br, marcelonogueira@ect.ufrn.br, marconicamara@ect.ufrn.br

Abstract— This article proposes an architecture to meet the need for new technologies in the area of residential automation. The proposed architecture allows the storage of domiciliary data and the achievement of strategies to better manage the resources, in order to seek greater efficiency in spending and allow a sustainable model. The architecture is built from open platforms and results in an effective solution that allows bidirectional data flow with low latency, fully compatible with the Wi-Fi network, works with the local network and, consequently, there is the possibility of use in environments that do not have an internet connection.

Keywords— Home automation, Domotics, Web Service, WebSocket.

Resumo— Este artigo propõe uma arquitetura para atender à necessidade de novas tecnologias na área de automação residencial. A arquitetura proposta permite o armazenamento de dados referentes ao domicílio e a obtenção de estratégias para melhor administrar os recursos, a fim de buscar uma maior eficiência nos gastos e permitir um modelo sustentável. A arquitetura é construída a partir de plataformas abertas e resulta em uma solução eficaz, que permite fluxo de dados bidirecional com baixa latência, totalmente compatível com a rede Wi-Fi, funciona com a rede local e, conseqüentemente, há a possibilidade de uso em ambientes que não possuem conexão com a internet.

Palavras-chave— Automação residencial, Domótica, Web Service, WebSocket.

1 Introdução

A automação residencial utiliza tecnologias diversas para prover um estilo de vida diferente do que jamais foi visto, alinhando a comodidade do controle dos gastos com água, energia, dentre outros recursos, à possibilidade de interações físicas com os equipamentos inteligentes. Nessa perspectiva vislumbra-se facilitar a vida de idosos, pessoas com deficiência física que necessitam de acompanhamento a todo momento. Esse avanço tecnológico tem muitas vantagens para o usuário, demonstradas nos aspectos a seguir: na **segurança** - permite a vigilância domiciliar, o acompanhamento do estado atual de portas, janelas, entre outros; no **controle** - possibilita a tomada de decisões e acionamento de quaisquer dispositivos conectados a uma rede; na **manutenção** - é possível ter previsibilidade dos gastos a partir dos dados gerados pelos sensores; no **acesso remoto** - independentemente de onde esteja, o usuário pode monitorar e controlar sua residência sem a necessidade de estar presente no local.

O desenvolvimento progressivo da infraestrutura da eletrônica computacional, com alta capacidade de armazenamento e processamento alinhado a um baixo consumo de energia, propiciou o desenvolvimento de pequenos dispositivos, chamados de microcontroladores, capazes de monitorar atividades corriqueiras e também servidores com competência para grande acúmulo e gerenciamento de dados, conhecidos como *big data* (Provost and

Fawcett, 2013). Em conjunto, houve o aperfeiçoamento de comunicação entre equipamentos eletrônicos que permite maior velocidade, alcance e segurança alinhado a uma menor perda energética na transferência de dados abrindo portas para o conceito de IoT (*Internet of Things*), internet das coisas (Santos et al., n.d.), que utiliza de todos esses processos para transformar simples objetos em sensores inteligentes (Miorandi et al., 2012).

Com o alto poder de processamento e armazenamento nos microeletrônicos, há uma grande transferência de dados entre equipamentos. No âmbito da comunicação sem fio (ou *wireless*) foram desenvolvidos diversas especificações para o envio de informações entre dispositivos eletrônicos com o objetivo de diminuir o tráfego de rede, interferências, consumo de energia e perda de dados, e aumentar a segurança, alcance do sinal e muitos outros aspectos. Alguns dos protocolos existentes são: *Z-Wave*; *Bluetooth*; *Zigbee*; *Wi-Fi*. Os protocolos de comunicação também evoluíram para satisfazer, por exemplo, a necessidade de baixo tempo de resposta; novos padrões de segurança na transferência de dados e; comunicação em tempo-real. O *Hyper Text Transfer Protocol Version 2* (HTTP/2) foi criado para oferecer um uso mais eficiente de rede, compressão de dados e outras melhorias (Belshe et al., 2015), e a adição da camada de segurança com o SSL/TLS, resultar no protocolo HTTPS (Rescorla, 2000), que criptografa a conexão cliente-servidor, dificultando a visualização dos dados transferidos por terceiros.

Portanto, ao longo deste artigo será caracterizado uma arquitetura para automação residencial que facilita instalar dispositivos inteligentes sem complicações e conhecimentos técnicos; monitorar seus dados e visualizá-los em relatórios gráficos que podem ter suas escalas de tempo alteradas, por exemplo, monitorar por dia ou mês; criar relatórios automaticamente e; facilitar o controle para todo e qualquer dispositivo conectado a partir dos conceitos de grupos e predefinições de forma intuitiva para o usuário, além de possibilitar o seu uso sem estar necessariamente conectado externamente à internet, ou seja, usar apenas na rede interna da residência. A aplicação mostrará os dados de consumo de energia a partir de tomadas inteligentes e de água por meio de hidrômetros conectados, além de acionamento de lâmpadas e grupos com diversos dispositivos. Tudo isso a partir de um aplicativo web (*WebApp*) e uma central servindo como *Web Service* conectada a todos os dispositivos da residência. Esta arquitetura adota: o protocolo **Wi-Fi** - por se tratar de uma rede de comunicação sem fio que apresenta diversas vantagens como: mobilidade, compatibilidade, compartilhamento, acessibilidade, baixo custo de manutenção, segurança e popularidade; a tecnologia **WebSocket** - para comunicação em tempo-real e bidirecional; e o protocolo **HTTPS** - para a transferência de dados com conexão criptografada. Por isso, esta arquitetura diferencia-se das demais arquiteturas como (Leu et al., 2014): utiliza somente o protocolo HTTP através do processo de *polling*; e (Girau et al., 2013): utiliza somente o HTTP.

2 Arquitetura

Neste artigo desenvolveu-se uma arquitetura para automação residencial em que um conjunto de sensores e atuadores podem ser instalados em uma casa de tal forma que estes consigam se comunicar entre si e o usuário consiga configurar as regras que envolvem um conjunto arbitrário de sensores e acionem outro conjunto de quaisquer atuadores automaticamente. Além disso, é possível que o usuário consiga acionar manualmente os atuadores e visualizar de forma amigável os dados produzidos pelos sensores através de uma aplicação web disponível no servidor localmente e acessível por computador, celular ou *tablet* conectado à rede. Como mostrado na Figura 1, a arquitetura consiste de uma unidade central (*middleware*) que através da rede *Wi-Fi* da residência conecta-se aos dispositivos inteligentes, sensores e atuadores, para monitoramento do local, medição do consumo de energia e água, e acionamento de tomadas e sistemas de iluminação. A interface da aplicação web é mostrada ao acessar a unidade central, onde pode-se visualizar os dados monitorados, acessar os atuadores e, assim, controlar o ambiente sem estar necessariamente no local.

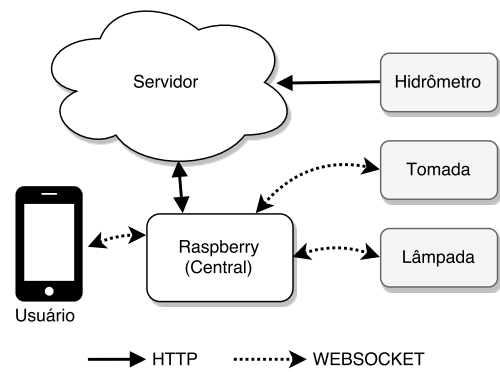


Figura 1: Arquitetura

2.1 Desenvolvimento Web

Em princípio a escolha pelas bibliotecas *React.js* e *Node.js* se deu por conta do eficiente desenvolvimento de interfaces web e melhor aplicação ao lado do servidor local. Portanto, ressalta-se que ambas as bibliotecas são implementadas na linguagem *JavaScript*, exatamente por permitir a entrada e saída de dados de forma assíncrona com extensa precisão e velocidade. Vale salientar que além do software, foi necessário selecionar outros hardwares para que fosse possível complementar a experiência. Entre eles o *Raspberry PI 3* é usado como um servidor central e local por oferecer uma velocidade de processamento adequado, conexão à internet sem fio e um tamanho reduzido; o ESP8266 possibilitou o uso de um microcontrolador com baixas dimensões, alta capacidade de processamento e com acesso ao *Wi-Fi* intrínseco ao chip, além do baixo consumo. Todos esses dispositivos comunicam-se no meio web com *Wi-Fi* e os protocolos HTTP e *WebSocket*.

O meio de desenvolvimento web foi escolhido por ser uma tecnologia difundida, de fácil acesso e também pela facilidade de implementar comparando-se com a programação nativa para *desktop* ou *mobile*, como *C/C++* ou *Java*. A adoção do padrão web traz diversos benefícios por se tratar um receptor universal, onde pode transitar qualquer tipo de informação que seja necessária. Para isso, o formato de envio de dados escolhido foi o *JSON (JavaScript Object Notation)* por ser um formato condensado para transferência de dados, diminuindo o tráfego desnecessário de dados que não fazem parte da informação enviada, como em outros formatos existentes. Atualmente, o *JSON* é amplamente utilizado, sendo suportado por diversas linguagens de programação.

Uma das grandes vantagens do meio web é por ele ser multiplataforma, permitindo que a aplicação web seja executada em qualquer sistema operacional que tenha suporte a um navegador, independente se é um *desktop* ou *smartphone*. Com isso, resulta-se uma manutenção mais simples do

que de uma aplicação dedicada para cada um dos sistemas e dispositivos existentes no mercado, pois o suporte da aplicação web dependerá apenas da versão do navegador disponível.

2.2 Interface

Diversas linguagens de programação e *frameworks* foram criados para agilizar o desenvolvimento de interfaces web. A escolha pela linguagem *JavaScript* deve-se por ter uma proposta de executar seu código em navegadores e em servidores, com sua característica de fluxo de execução assíncrono, permite que a entrada e saída de dados de funções sejam executados em paralelo, não bloqueando sua aplicação caso o usuário forneça dados muito extensos.

O *framework* selecionado foi o *React.js* que torna fácil criar *User Interface* (UIs) complexas. A biblioteca é baseado em componentes que podem atualizar seu próprio estado e podem passar dados através da aplicação usando o virtual DOM que é uma abstração do HTML DOM sendo mais rápido e leve. O *React.js* necessita de diversos módulos para suprir um ambiente de desenvolvimento completo, como: *React-router*, que cria rotas que alteram somente a *view* da aplicação *one-page*; *Redux*, que permite armazenar o estado dos componentes em um dicionário externo à árvore de componentes; *Socket.io-client* que implementa o *WebSocket* como uma abstração do mesmo, facilitando sua utilização; *Redux-socket.io*, que interliga o dicionário do *Redux* com os eventos do *Socket.io-client*.

Redux é uma implementação da arquitetura criada pelo *Facebook* chamada *Flux*, onde complementa e reestrutura a característica padrão da passagem unidirecional de dados no *React*. Dessa forma, o fluxo de dados fornece ferramentas para ajudar o desenvolvedor e simultaneamente tornar a aplicação mais dinâmica para o usuário.

O usuário interage com a interface web formada pelos componentes do *React* e, se for requisitado uma alteração nos dispositivos, chama uma função denominada *action*. A *action* serve somente para carregar as informações para uma outra função, que poderá solicitar ao servidor ou ao *reducer*. Se for ao servidor, é disparado um evento na conexão *WebSocket* que retorna ao *reducer* os novos dados. Como é apresentado na Figura 2, o *reducer* é um grande bloco de funções referentes a cada *action* interna ou eventos do servidor. Ele recebe o que deverá ser mudado na *store* (*dictionary* do *Redux*) e em seguida altera de forma imutável os dados e, após a mudança na *store*, despacha ao *React* para renderizar novamente os componentes que forem necessários, isso sendo feito instantaneamente, interferindo de forma mínima na experiência do usuário. A Figura 2 apresenta o fluxo descrito anteriormente.

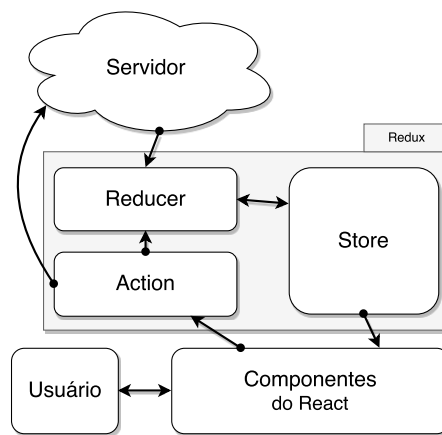


Figura 2: Fluxo dos dados na interface web

2.3 Unidade Central

A unidade central (ou *middleware*) é o principal componente desta arquitetura, é o intermediador entre a aplicação web acessada pelo usuário e todos os dispositivos inteligentes conectados na residência. Além de permitir o controle sem que o usuário esteja obrigatoriamente conectado à internet, a não ser para acesso externo, ela é responsável pelo gerenciamento de todos os pedidos da interface web; pelas solicitações do bloco de regras; pelo armazenamento de todos os dados gerados pelos sensores; e pelo acionamento dos atuadores por usuários e/ou bloco de regras. A central é um *Web Service* que recebe dados seguindo um formato pré-definido e retorna as informações solicitadas. É a partir desse formato de saída e entrada de dados que é possível universalizar essa aplicação para se comunicar com requisições de dados a partir de qualquer cliente e dispositivo, independente de quem esteja solicitando ou inserindo esses dados.

A aplicação segue os princípios do padrão *Representational State Transfer (RESTful)* que utiliza os métodos do HTTP (Richardson and Ruby, 2008) para acesso externo à central que recebe os dados gerados dos seguintes componentes: consumo de água a partir de hidrômetros; consumo energético a partir de tomadas; registro de acionamento do sistema de iluminação e das tomadas. O *middleware* é constituído de várias ferramentas e bibliotecas que funcionam integradas entre si, como é mostrado na Figura 3.

O usuário, por meio da interface web, e os vários dispositivos inteligentes conectam-se à unidade central através dos protocolos HTTPS ou *WebSocket*, depende do volume de dados e da necessidade de monitoramento em tempo real. As requisições são recebidas pelo *Node.js* que trata o que é pedido e guarda o que é recebido. O *Node.js* é um *framework* para construção de aplicações de alta performance e fluxo assíncrono que permite

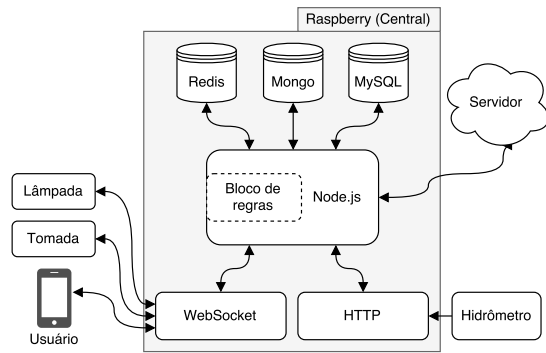


Figura 3: Ferramentas utilizadas na Unidade Central

receber múltiplas conexões e processá-las simultaneamente (Tilkov and Vinoski, 2010). Toda a lógica de leitura e gravação de dados, a execução do conjunto de regras, a conexão com a nuvem para *backup* de dados e a comunicação com os três tipos de bancos de dados é feita pelo *Node.js*.

Sabe-se que o armazenamento dos dados é de suma importância, mas para arquivá-los com segurança, se faz necessário saber seu tipo e prioridade de processamento. Nesta arquitetura é utilizado três diferentes bancos de dados com propósitos distintos de processamento e cada um tem vantagens e desvantagens quanto aos seus desempenhos (de Souza and dos Santos, 2015). O *Redis* é um banco não relacional que armazena os dados em memória e apresenta um alto desempenho de leitura e escrita se comparado aos Sistema de Gerenciamento de Banco de Dados (SGBD) de armazenamento em disco. É utilizado o *Redis* para armazenar dados que precisam de acesso rápido, por exemplo, estado atual dos sensores e atuadores, armazenamento das multi-conexões gerenciadas pelo *Socket.io*, além de possibilitar a execução de vários processos (*clusters*) de aplicações *Node* centralizando as conexões *sockets*. O *MongoDB* também é um banco não relacional e utiliza o armazenamento em disco. Mesmo assim é excelente para armazenamento de grande quantidade de dados, como os registros de: medição de água; consumo de energia; acesso de usuário à aplicação e dispositivos. Diferente dos anteriores, o *MySQL* é um SGBD relacional com bom desempenho na indexação de dados (de Souza and dos Santos, 2015) e também com armazenamento em disco, no entanto, não permite o escalonamento horizontal, não suportando aplicações com grande fluxo de dados. Sabendo disso, utiliza-se o *MySQL* para armazenar os dados de usuários, casas, dispositivos, grupos e seus relacionamentos, como: uma casa só pode estar cadastrada se tiver algum usuário dono da mesma; um dispositivo só existe se estiver relacionado a alguma residência.

2.4 Dispositivos

Os dispositivos utilizados são todos desenvolvidos com base no microcontrolador (MCU) ESP8266 devido a conectividade *Wi-Fi* nativa do chip que possibilita a integração direta entre os sensores e atuadores, com o servidor (Kolban, 2015). Essas propriedades desse componente eletrônico satisfazem, hoje, as necessidades desta arquitetura devido ao ambiente em que o mesmo está inserido ser baseado em protocolos de comunicação web. O chip suporta os protocolos web e mais os padrões de segurança *Wi-Fi*, como WEP, WPA-PSK, WPA2-PSK e WPS.

A utilização de um MCU nos dispositivos torna grande a gama de sensores e atuadores que podem ser instalados para automatização de tarefas gerais como: controle do cultivo na agricultura (Roeck et al., 2016); sistemas de iluminação e controle de eficiência energética (Vassoler, 2016), etc. Em virtude disso, utiliza-se o MCU tanto como sensor “puro” (envia os dados captados), atuador “puro” (recebe comandos para acionamento) e a junção dos dois, sensor-atuador. Exemplificando no âmbito desta arquitetura, exemplos destes podem ser, respectivamente, um hidrômetro, uma lâmpada e uma tomada.

A possibilidade de diversas formas de uso do mesmo microcontrolador juntamente com a função de regras do software, pode-se exemplificar a automação de tarefas simples como monitoramento e acionamento rápido. Uma das situações pode ser ao adicionar uma tomada ao sistema que monitora o consumo de um dispositivo qualquer que está sendo alimentado pela mesma, e uma regra de negócio que desliga a tomada caso o consumo do dia passe de um valor pré-definido; Um sensor de presença que pode ligar/desligar determinado grupo de luzes de um cômodo da casa.

2.5 Integração

Os dispositivos conectam-se à unidade central através da rede *Wi-Fi* da residência e utilizam os protocolos HTTPS e/ou *WebSocket*. A escolha por qual protocolo utilizar depende principalmente do tempo de resposta que se exige nas transferências dos dados, como a ação de ativar determinada lâmpada e o envio de uma medição realizada por um hidrômetro têm prioridades diferentes, enquanto o primeiro caso necessita que a ação seja realizada instantaneamente, o segundo não precisa ter uma resposta tão imediata. Todos os dispositivos reportam-se à central ao identificarem alguma anormalidade, como problemas no envio dos dados, na conexão com a internet ou seu próprio reinício inesperado. Os dispositivos e usuários são autenticados seguindo a especificação do *OAuth 2.0* (Hardt, 2012) que padroniza a autenticação através de chaves de segurança temporárias, diminuindo a exposição de dados sensíveis,

como senhas.

Como é visto na Figura 4, os dispositivos inteligentes (tomada, hidrômetro e lâmpada) comunicam-se com a unidade central ou a nuvem, recebendo ações a serem executadas e enviando dados monitorados. Devido ao baixo nível de urgência de resposta, o hidrômetro é o único que conecta-se diretamente à nuvem e utiliza unicamente o protocolo HTTP, representado pelo POST, que é um dos métodos desse protocolo. Utiliza-se o método POST para enviar dados para o servidor, sendo essa a finalidade do hidrômetro, fazer o monitoramento do consumo de água e armazená-las no servidor. Diferente do hidrômetro, a interface do usuário, a tomada e a lâmpada tem alto nível de urgência de resposta, exigindo uma resposta imediatamente após o acionamento. Para suprir essa necessidade, ambos os dispositivos conectam-se diretamente à unidade central por meio do protocolo *WebSocket* para comunicação em tempo-real. Além de gerenciar todas essas conexões com diferentes dispositivos e protocolos, o *middleware* também conecta-se à nuvem em casos específicos em que a requisição solicita alguma informação que não é encontrada localmente, ou quando necessita, por segurança, fazer *backup* dos dados já armazenados, ou ainda possibilitar o acesso externo do usuário à própria residência, sendo isso opcional.

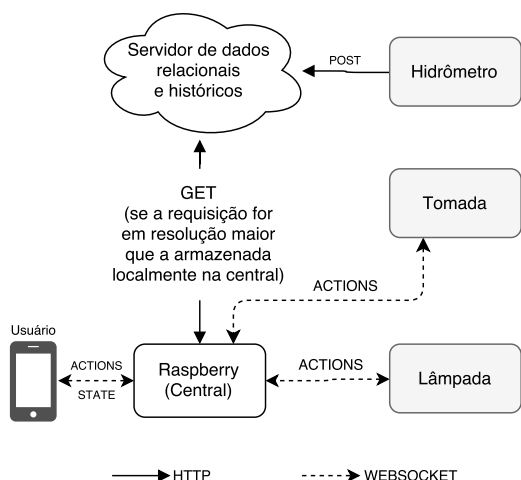


Figura 4: Comunicação entre os dispositivos

3 Resultados

3.1 Interface

A interface da aplicação atendeu aos requisitos descritos na introdução. Com rápida renderização e carregamento pelo cache, os arquivos contabilizam em média 400 kb, levando em consideração uma velocidade de rede de 750 kb/s a 1,5 Mb/s a tela foi carregada já com as informações advin-

das do servidor em 1.23 segundos, possibilitando uma boa experiência para o usuário. A aplicação pode ser adicionada à tela padrão de celulares simulando um aplicativo nativo à partir do navegador padrão dos sistemas operacionais mobile. O tráfego de rede foi reduzido ao máximo pois a interface guarda os dados que já foram requisitados do servidor.

A Figura 5 mostra a interface da aplicação web que fornece o histórico do consumo energético do dispositivo conectado à uma tomada inteligente. A interface permite visualizar o histórico de qualquer dispositivo inteligente cadastrado e o período de amostragem com resolução diária, mensal e anual. Além disso, a interface possui outras telas que possibilitam o cadastro de dispositivos e criação de grupos, e o controle individual ou em grupo dos mesmos, assim como a criação do conjunto de regras que serão executadas automaticamente.

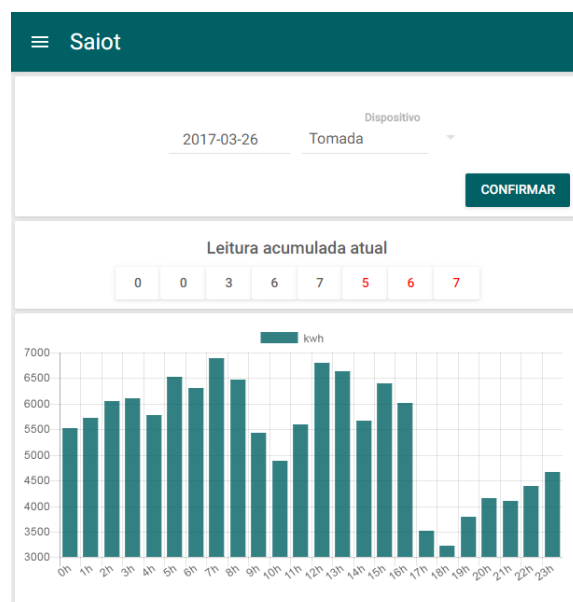


Figura 5: Tela da interface mostrando o gráfico do consumo de energia de uma tomada

3.2 Validação da arquitetura

Para validar a arquitetura proposta, os dispositivos mostrados na Figura 4 foram desenvolvidos e utilizados em ambiente experimental que permite monitorar o consumo de energia e água de aparelhos reais. É representado na Figura 5, o consumo real de um desses equipamentos, no caso, a tomada. O ambiente de teste consiste em um aquário com água, uma pequena bomba de aquário, um hidrômetro e uma tomada inteligente. A bomba, localizada dentro do aquário, bombeia a água para o hidrômetro através de um pequeno cano. A bomba está plugada à tomada para assim permitir a medição do seu consumo energético. Todos os equipamentos ficam ligados 24 ho-

ras por dia, enviando os dados captados durante todo esse tempo para a nuvem.

A partir dos dados obtidos, pode-se comparar as medições com os valores fornecidos pela fabricante da bomba de aquário, verificando se o consumo obtido pela tomada inteligente é coerente com o consumo teórico disponibilizado. Quanto ao hidrômetro, compara-se o valor acumulado registrado analogicamente com o somatório dos pulsos contabilizados digitalmente, resultando no mesmo valor total. Ao fim da análise, verificou-se que a unidade central se comportou de forma correta, não perdeu nenhum dado recebido dos dispositivos conectados e disponibiliza as informações para a interface do usuário sem nenhuma inconsistência.

4 Conclusões

A arquitetura proposta para automação residencial com uso de rede *Wi-Fi* e *Web Service* foi implementado e validado de forma experimental com dispositivos inteligentes e uma unidade central implementada diretamente na nuvem. Um ponto a ser destacado é a confiabilidade obtida pelo *middleware* por não apresentar perda de dados no processamento das requisições recebidas e enviadas. A escolha por utilizar comunicação por rede *Wi-Fi* torna a arquitetura compatível com os roteadores residenciais comuns e simplista por não necessitar de nenhum dispositivo conversor entre diversos protocolos existentes de redes sem-fio, não perdendo performance durante essa conversão. A utilização do HTTP e *WebSocket* seguindo o padrão *RESTful* torna essa aplicação universalmente acessível, podendo ser acessada por qualquer interface de usuário devidamente autenticado e receber dados de qualquer dispositivo inteligente que siga esse mesmo padrão.

Uma característica importante dessa arquitetura é a possibilidade de implementar-se em lugares que não possuem conexão com a internet, ou seja, existe apenas uma rede doméstica. Nesse cenário, a arquitetura continua a funcionar normalmente, como explicado anteriormente, os dispositivos se reportam apenas à unidade central que está conectada localmente, independente da existência de redes que possibilitam o acesso externo. Essa propriedade também é importante quando tem-se uma conexão com a internet que apresenta oscilações ou fica indisponível durante um determinado período, enquanto isso, os dispositivos estão se comunicando normalmente com a central local.

Referências

- Belshe, M., Thomson, M. and Peon, R. (2015). Hypertext transfer protocol version 2 ([http/2](http://http2.org/)).
- de Souza, V. C. O. and dos Santos, M. V. C. (2015). Maturing, consolidation and performance of nosql databases-comparative study, *Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective-Volume 1*, Brazilian Computer Society, p. 32.
- Girau, R., Nitti, M. and Atzori, L. (2013). Implementation of an experimental platform for the social internet of things, *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 500–505.
- Hardt, D. (2012). The oauth 2.0 authorization framework.
- Kolban, N. (2015). Kolban’s book on esp8266, *an introductory book on ESP8266*.
- Leu, J.-S., Chen, C.-F. and Hsu, K.-C. (2014). Improving heterogeneous soa-based iot message stability by shortest processing time scheduling, *IEEE Transactions on Services Computing* **7**(4): 575–585.
- Miorandi, D., Sicari, S., Pellegrini, F. D. and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges, *Ad Hoc Networks* **10**(7): 1497 – 1516.
- Provost, F. and Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making, *Big Data* **1**(1): 51–59.
- Rescorla, E. (2000). Http over tls.
- Richardson, L. and Ruby, S. (2008). *RESTful web services*, ”O’Reilly Media, Inc.”.
- Roeck, A. G. W., Silva, L. W. d. et al. (2016). Smart gardens-automação e monitoramento inteligente da agricultura.
- Santos, B. P., Silva, L. A., Celes, C. S., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N. and Loureiro, A. A. (n.d.). Internet das coisas: da teoria à prática.
- Tilkov, S. and Vinoski, S. (2010). Node.js: Using javascript to build high-performance network programs, *IEEE Internet Computing* **14**(6): 80–83.
- Vassoler, V. (2016). Mudança de tecnologia e como isso pode ajudar na redução de consumo na iluminação das empresas, *Revista Gestão & Sustentabilidade Ambiental* **5**(2): 382–397.