

# Monitoramento domiciliar do abastecimento de água utilizando conceitos de Internet das Coisas

*Title: Home monitoring of water supply using Internet of Things concepts*

**Carlos Alberto Luciano da Silva Júnior**  
Escola Politécnica de Pernambuco  
Universidade de Pernambuco  
50.720-001 - Recife, Brasil  
[calsj@ecomp.poli.br](mailto:calsj@ecomp.poli.br)

**Edison de Queiroz Albuquerque, Dr.**  
Escola Politécnica de Pernambuco  
Universidade de Pernambuco  
50.720-001 - Recife, Brasil  
[edison@ecomp.poli.br](mailto:edison@ecomp.poli.br)

**Resumo** *O número de pessoas que sofrem com a irregularidade no abastecimento de água é grande e em alguns casos a água só fica disponível em certos horários. Isso torna a população refém de um frequente monitoramento manual para verificar a disponibilidade de água na torneira. Assim, este trabalho trata-se de um estudo de caso que utiliza conceitos de Internet das Coisas (do inglês, Internet of Things (IoT)) no qual é proposto um sistema para monitoramento domiciliar do abastecimento de água de uma maneira otimizada, onde o usuário poderia verificar através do seu Smartphone a disponibilidade de água e além disso ele poderia controlar a “torneira”, abrindo-a e fechando-a remotamente.*

**Palavras-Chave:** *IoT, Raspberry, Sensor, Abastecimento de Água*

**Abstract** *The number of people that is suffering with a irregular supply of water is huge and in some of the cases, the water is available only in some part of the day. This leaves the population into a way where they need to perform a manual check in order to verify if there is water flow on tap. This paper is a case-study that uses Internet of Things (IoT) concepts and aims to create a home monitoring system of water supply in a optimized way, where the user shall be able to check using a mobile device if there is water flow on tap and shall take actions to control the “tap”, like opening and closing it remotely.*

**Keywords:** *IoT, Raspberry, Sensor, Water Suplly*

## 1 Introdução

A água é um bem essencial para a vida, seja para higiene ou para alimentação, o ser humano precisa dela para sobreviver. Infelizmente, a situação do abastecimento de água no Brasil não é das melhores e constantemente os problemas relacionados a isso são enfrentados pela população.

A crise hídrica no estado de São Paulo, por exemplo, que começou em 2014 e se estendeu até meados de 2016 chegou a níveis preocupantes e poucas vezes visto no estado. As principais causas foram a diminuição de chuvas na região Sudeste e também a má administração política. Como consequência, a população passou por períodos de racionamento e interrupção no abastecimento de água. [1]

Outras regiões do Brasil, como o Nordeste, sofrem anualmente pelo não abastecimento contínuo de água, o que causa um estado de preocupação na população afetada. Uma alternativa para contornar esse problema é manter reservatórios de água potável nas residências.

Por outro lado, para manter esses reservatórios abastecidos deve-se checar frequentemente a existência de água provinda do sistema de abastecimento da cidade. Uma vez que não é do conhecimento da população o momento que irá chegar a água, torna-se impossível o imediato aproveitamento desse recurso após sua disponibilidade, pois essa verificação é feita de forma manual.

Apesar dos grandes marcos tecnológicos da era digital, muitas tarefas do cotidiano ainda são executadas de modo manual, como, por exemplo, o contexto citado acima. Atualmente quase tudo se resolve atrás de telas e botões, onde objetos que antes eram apenas dedicados a realizarem uma tarefa específica, hoje são capazes de analisar o contexto em que estão, processar esses dados de maneira eficiente e além disso têm a habilidade de se comunicarem entre si.

Tendo esse cenário em vista, propõe-se com este trabalho um estudo de caso para automatizar essa verificação manual, utilizando um sensoriamento da presença de água e conceitos de Internet das Coisas (do inglês, *Internet of Thing* (IoT)).

Para isso, na Seção 2 serão abordados os conceitos teóricos que viabilizam este trabalho, na Seção 3 será apresentada a proposta do sistema, na Seção 4 será descrito como foi o desenvolvimento do sistema, na Seção 5 serão apresentados os resultados, na Seção 6 serão abordadas as conclusões obtidas deste trabalho e por fim na Seção 7 serão descritas algumas melhorias e trabalhos futuros relacionados.

## 2 Fundamentação Teórica

### 2.1 Sensor de Água

Atualmente no mercado já existem alguns sensores que identificam a presença de água, porém nenhum deles foi considerado ideal para o escopo deste trabalho. O sensor de fluxo de água, por exemplo, cuja ativação se dá pelo giro da hélice, utilizado em [2], poderia ser mal interpretado caso houvesse passagem de ar na tubulação.

Dessa forma, para evitar possíveis más interpretações, foi pensado na implementação de um sensor que atendes-se os requisitos deste trabalho, mais precisamente, um sensor que informasse se de fato há ou não água.

Como é de conhecimento geral, a água que chega nas residências contém impurezas, o que a torna condutora de eletricidade. Sendo assim, foi desenvolvido um sensor artesanal composto apenas de um resistor e alguns fios que utiliza a própria água para a ativação do mesmo.

Portanto, conforme é mostrado na Figura 1, com a presença da água na tubulação os polos 1 e 2 estarão conectados eletronicamente fechando assim o circuito mostrado.

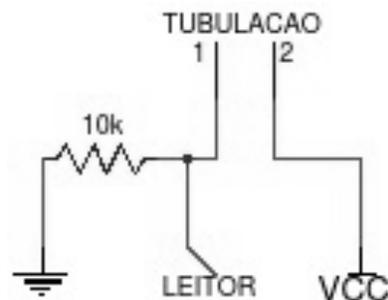


Figura 1: Esquemático do circuito do sensor de água.

Para ler os valores desse sensor foi utilizado uma placa Raspberry Pi, um mini computador de baixo custo e alta portabilidade, pois todo o *hardware* é integrado em uma única placa. Desse modo, com a presença de água na tubulação fecha-se o circuito mostrado acima e para evitar um curto circuito entre o VCC e o terra coloca-se um resistor, para não danificar a Raspberry Pi.

O ponto na Figura 1 identificado como LEITOR é a saída do sensor, ela estará com nível lógico alto com água e nível lógico baixo na ausência da mesma. A Raspberry Pi irá ler esses valores através de um dos pinos de entrada e saída e processará essa informação.

### 2.2 Internet das Coisas

A Internet está atualmente enraizada no cotidiano da sociedade moderna e tornou-se não apenas uma rede de

computadores interligados, mas uma rede de pessoas interligadas. Sendo, assim, um meio de comunicação indispensável, seja no âmbito profissional ou pessoal.

Com a popularização de dispositivos móveis e o fácil acesso a Internet o número de pessoas conectadas só tende a crescer. Possibilitando assim que uma grande parcela da população tenha acesso a serviços voltados para esse meio de comunicação.

O próximo passo na evolução da Internet é a interconectividade de objetos e indivíduos, resultando na otimização de ações do dia a dia de maneira inteligente e sensorial. Hoje em dia já existem muitos acessórios do dia a dia disponíveis em versão *smart*, tais como relógios, geladeiras, lâmpadas, ar condicionados, etc.

Essa evolução é conhecida como Internet das Coisas (IoT). Dispositivos e objetos com sensores, transmissores e receptores, que possibilitam novas formas de comunicação entre pessoas e objetos e entre objetos e objetos, em qualquer lugar e tempo, descrevendo assim um novo paradigma da comunicação. [3]

Objetos capazes de captar aspectos do mundo real, o processamento desses dados em informações úteis e a troca desse contexto com outros objetos são a chave para o desenvolvimento da Internet das Coisas. Para que essa interconectabilidade aconteça utiliza-se frequentemente redes sem fio, tais como Wi-Fi, Bluetooth, NFC, etc, para a troca de informações entre os dispositivos.

Neste trabalho optou-se por usar o protocolo de comunicação chamado *WebSocket*, o qual é transferido através da Internet, podendo ser utilizado, por tanto, redes Wi-Fi como também redes móveis.

### 2.2.1 WebSocket

A *web* tem sido construída com base no conhecido paradigma de solicitação/resposta de HTTP (protocolo de transferência de hipertexto, do inglês, *Hyper Text Transfer Protocol*), onde só há tráfego de dados provindo do servidor se o mesmo tiver sido solicitado anteriormente pelo usuário.

Toda a comunicação HTTP é direcionada pelo usuário, o que exige interação direta dele. Por exemplo, se um usuário carrega uma página da *web*, em seguida, nada acontece até que ele clique na próxima página.

Tecnologias, conhecidas como *Push* ou *Comet*, foram utilizadas no passado para enviar dados do servidor para o cliente sem serem solicitados, criando assim uma ilusão de uma conexão criada pelo servidor. Porém, para que isso funcionasse uma conexão HTTP deveria ser deixada aberta até que algum dado do servidor fosse enviado para o cliente. [4]

O problema dessa abordagem é que ela sobrecarrega a

rede com conexões HTTP, o que não é recomendado quando se tem uma aplicação com baixa latência. Em meados de 2010 surgiu uma especificação chamada *WebSocket*, a qual viria a solucionar problemas como o descrito a cima de uma forma mais otimizada e versátil.

Um soquete de rede é uma das extremidades de um canal de comunicação entre dois computadores atrás de uma rede, sendo definido pelo endereço IP do computador e o número da porta a ser usada. Trazendo esse conceito para a *web*, tem-se a definição de um *WebSocket*, o qual estabelece conexões de soquete entre um navegador da *web* e um servidor. Isto é, há uma conexão persistente entre o cliente e o servidor, onde ambas as partes podem começar a enviar dados a qualquer momento.

Um serviço utilizando *WebSocket* utiliza o protocolo HTTP e seus principais métodos são: *onopen*, *onclose*, *onmessage*, *onerror* e *send*. Os métodos que têm prefixo *on* são invocados automaticamente a depender do estado do objeto de conexão.

Para inicializar um objeto *WebSocket* o cliente deverá chamar o construtor passando como parâmetro o endereço do servidor (host) e, em seguida, utilizar o métodos que for mais adequado. *Onopen* será invocado apenas quando a conexão com o servidor for aberta, já o oposto disso, quando a mesma for fechada, será invocado o *onclose*.

As mensagens enviadas do servidor para o cliente chegarão e serão manipuladas no método *onmessage* e caso algum erro ocorra o método *onerror* será chamado. Já o método *send* deverá ser utilizado a fim de enviar dados para o servidor, o conteúdo da mensagem deverá está no padrão que o servidor espera receber.

*WebSocket* será usado neste trabalho para a comunicação entre o sensor de água e o usuário final. Para que isso ocorra o sensor irá enviar seu estado para um servidor na nuvem e esse servidor irá encaminhar a mensagem para o usuário, o qual poder enviar ações de volta a serem aplicadas no sistema.

## 3 Descrição da proposta

A proposta deste trabalho é um estudo de caso de um sistema que ajudaria seus usuários (pessoas que moram em localidades que sofrem com uma distribuição de água irregular) a otimizar e a mudar a forma como que eles lidam com a falta de informação sobre o abastecimento de água na região.

O fluxo desse sistema funciona da seguinte forma. Um sensor de detecção de água será instalado na tubulação de água que chega na casa do usuário e enviará para o mesmo em tempo real o seu estado atual, ou seja, se há água ou não disponível na tubulação.

Essa informação será recebida pelo usuário atrás do seu *smartphone*, possibilitando assim uma maior comodidade e também mobilidade. Ao ser notificado da disponibilidade de água o usuário poderá, então, enviar um comando para abrir a tubulação, permitindo assim que a água entre na sua casa e conseqüentemente abasteça o seu reservatório.

Além desse cenário descrito, espera-se também como resultado desse sistema um histórico da disponibilidade de água por dia e por quantas horas. Sendo assim, esses dados podem ser usados para promover ações de conscientização para os próximos períodos correlatos ou de prevenção.

## 4 Desenvolvimento do serviço

Nessa seção será explicado todo o processo para o desenvolvimento do sistema, passando por todos os componentes e como os dados trafegam do sensor ao usuário final. Para começar, uma breve introdução a ARTIK Cloud, uma plataforma desenvolvida e distribuída pela Samsung, sendo esta escalável para diversos tipos de aplicações e dispositivos, facilitando assim o desenvolvimento de projetos relacionados a Internet das Coisas. [5]

Com a ARTIK Cloud os clientes podem acessar e agregar dados do histórico de diferentes fontes, abrindo assim uma nova perspectiva para *big data*. Além disso, podem utilizar vários tipos de protocolo diferentes para enviar e receber dados, tais como REST, WebSocket, MQTT e CoAP. A plataforma dispõe de um painel de controle onde pode-se cadastrar e administrar os diversos tipos de aplicações, dispositivos e usuários, restringindo dentre eles o acesso aos dados e as ações que cada um pode realizar.

A plataforma ARTIK Cloud dispõe de SDKs (Kit de Desenvolvimento de *Software*, do inglês, *Software Development Kit*) para diversas linguagens, tais como Java/Android, Objective-C/iOS, Python, C#, Ruby, entre outras. Além disso, é possível encontrar diversos tutoriais com seus respectivos códigos-fonte no *site* oficial da plataforma. Os quais podem ser usados como base para o desenvolvimento de novas aplicações. [6]

Para se utilizar a plataforma é necessário que inicialmente crie-se uma conta na ARTIK Cloud para ter acesso ao painel de configuração, além do acesso a API (Interface de Programação de Aplicação, do inglês, *Application Programming Interface*) e a documentação. Para este trabalho foi escolhido como base o tutorial cujo nome é *An IoT remote control*, o qual tem como finalidade controlar a partir de uma aplicação Android um LED conectado à placa Raspberry. [7]

A aplicação base Android do tutorial foi desenvolvida em Java no Android Studio e a aplicação base Raspberry Pi foi desenvolvida em Node.js. A partir desses códigos-fonte foram desenvolvidas novas funções a fim de alcançar o objetivo deste trabalho.

Para começar deve-se adicionar e configurar na plataforma ARTIK Cloud as três entidades seguintes: a aplicação, o tipo do dispositivo e o dispositivo. Pois cada uma dessas entidades ao serem criadas geram identificadores que serão usados nos códigos-fonte da Raspberry Pi e do Android.

Na entidade “tipo do dispositivo” declara-se variáveis e os tipos de dado dessas variáveis, como por exemplo, o campo `tem_agua` sendo do tipo *boolean* informando se o estado é verdadeiro ou falso. Além disso, são declaradas as ações que um certo tipo de dispositivo poderá executar. Neste projeto foram declaradas as seguintes ações: informar o estado atual do sensor, abrir e fechar a tubulação, sendo elas `getLastState`, `setOn` e `setOff` respectivamente.

Na própria plataforma pode-se definir também regras que executarão essas ações a depender do estado das variáveis pré-definidas, evitando assim qualquer interação com o usuário. Uma outra maneira para ativar essas ações é envia-las de uma aplicação cliente e essa será a forma utilizada neste trabalho.

Assim que a entidade “tipo do dispositivo” estiver configurada, pode-se adicionar a entidade “dispositivo” relacionada a esse tipo. Para cada dispositivo no mundo real, deve haver uma entidade “dispositivo” relacionada, pois serão gerados pela plataforma um identificador e um *token*, os quais servirão para identificar unicamente esse dispositivo.

Para encerrar as configurações na ARTIK Cloud deve-se criar a entidade “aplicação” para obter o identificador que será usado na API da plataforma no Android. Para isso, escolhe-se o método de autenticação OAuth2 que será utilizado e a URL de redirecionamento da autenticação.

Utilizando *WebSocket*, como definido na Seção 2.2.1, pode-se construir uma conexão entre a ARTIK Cloud e os dispositivos compatíveis e suas aplicações, afim de receber e/ou enviar mensagens em tempo real. São utilizados neste projeto dois tipos de *WebSocket*: *FireHose* e *Device Channel*

*FireHose WebSocket* é uma conexão unidirecional que tem como propósito monitorar em tempo real todas as mensagens enviadas para a ARTIK Cloud dos dispositivos específicos. Por outro lado, *Device Channel WebSocket* é uma conexão bidirecional entre a plataforma e um ou mais dispositivos para a troca de mensagens direcionadas.

Um exemplo da arquitetura e do uso das conexões entre a ARTIK Cloud, a Raspberry e a aplicação Android pode ser vista na Figura 2.

O estado do sensor, como descrito na Seção 2.1, muda quando há ou não a presença da água na tubulação e o mesmo é enviado para a ARTIK Cloud pela Raspberry. Através do *FireHose WebSocket* configurado na aplicação Android, o qual monitora o estado do sensor na plataforma, uma vez que um novo estado é detectado, o Android notifica o usuário. Essas duas rotinas são ilustradas na Figura 2 pelas conexões 3 e 4 respectivamente.

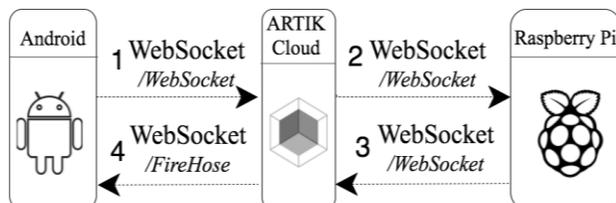


Figura 2: Arquitetura das conexões.

Para a implementação da rotina de monitoramento do sensor na placa Raspberry Pi foi importada uma biblioteca chamada *onoff*. Essa biblioteca auxilia as operações de entrada e saída da placa, dispondo ainda de métodos que especificam funções de *call-back* para toda mudança de estado.[8]

Ao abrir a aplicação Android é solicitado, através da ação *getLastState*, o envio do estado atual do sensor a fim de sincronizar a aplicação. O usuário pode verificar o estado do sensor em tempo real e pode decidir por abrir ou fechar a tubulação comandando assim a ação desejada. Na Figura 3 pode-se ver a captura de tela da aplicação Android, mostrando os dois botões de comando e o retorno da função *getLastState*.

Ao enviar um comando para a Raspberry é utilizado um *Device Channel WebSocket*, o qual contém o identificador do dispositivo que receberá a mensagem. Na Figura 2 isso está ilustrado pelas conexões 1 e 2.

A decisão de abrir a tubulação é tomada por duas condições que devem ser verdadeiras: se há água na tubulação e se o usuário enviou o comando para abri-la. Isso garante que a tubulação só será aberta caso haja água, mesmo que o usuário envie o comando para abrir.

Quando a Raspberry detectar que houve uma mudança no estado do sensor de água e o mesmo está informando que não há mais água, uma decisão automática é tomada para fechar a tubulação. Uma outra maneira para fechar a tubulação é sendo enviada um comando pelo usuário através do aplicativo.

Para viabilizar a abertura da tubulação através da placa Raspberry Pi foi usado um relé e uma válvula para máquina de lavar. Relé é um interruptor eletromecânico

que pode ser configurado de duas maneiras quanto ao seu contato: normalmente fechado e normalmente aberto. Na Figura 4 é mostrado o protótipo final do projeto.



Figura 3: Captura de tela da aplicação Android.

Na configuração normalmente aberto o circuito irá fechar quando há corrente elétrica nas bobinas do relé, já no normalmente fechado quando há corrente elétrica o circuito se abre. A placa irá ativar o relé e esse irá acionar a válvula, permitindo assim que a tubulação se abra. Por isso foi configurado, neste projeto, o relé em normalmente aberto, pois no estado padrão a tubulação permanecerá fechada.

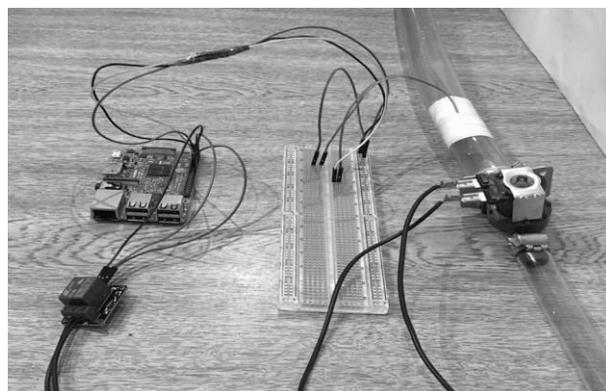


Figura 4: Foto do protótipo final do projeto.

Na Figura 5 está o pseudocódigo que é executado na Raspberry Pi das ações descritas acima.

```

//inicialização da variável
tem_agua = ler_valor_atual()

//Monitoramento do sensor
Função que é chamada quando há
mudança no estado do sensor
(estado){

//variável tem_agua recebe o
valor atual do sensor
tem_agua = estado

Se (tem_agua for falso)
então feche a tubulação, pois
não há água:
tubulacao(fechar)

//função para enviar o estado
atual do sensor para a ARTIK
Cloud
enviar_estado_do_sensor();

}

//Manipulação das ações
Função que é chamada quando
recebe mensagem da ARTIK
Cloud(mensagem) {

Se (mensagem for do tipo ação)
então:

    Se (ação for abrir tubula-
ção)
    então: tubulacao(abrir)

Se (ação for fechar tubulação)
    então: tubulacao(fechar)
}

Função tubulacao (ação){

Se (ação for abrir e tem_agua
for verdadeiro)
então: abra a tubulação

Senão:
feche a tubulação
}

```

Figura 5: Pseudocódigo da Raspberry.

## 5 Resultados

Após o desenvolvimento dos *softwares* no Android e na placa Raspberry Pi e a montagem do *hardware*, foi testado o serviço a fim de avaliar o seu desempenho. Como esperado o sistema mostrou-se eficiente no seu propósito, realizando com exatidão as funções desejadas.

Foram realizados dois experimentos para medir a performance do sistema quanto ao tempo de resposta, os cenários foram os seguintes: um com o celular conectado a uma rede Wi-Fi e o outro com o celular usando uma rede móvel 3G. Para cada cenário foram executados cinco testes e para cada teste foram medidos o tempo que o sistema leva para detectar a chegada da água e o tempo que o sistema leva para detectar a interrupção no abastecimento da água até notificar o usuário.

Esses resultados são mostrados na Tabela 1 sendo a segunda coluna os tempos medidos nos testes da chegada da água no sistema e a terceira coluna os da interrupção.

	Presença	Ausência
Experimento 1: utilizando rede Wi-Fi		
Medição 1	3,3 ms	3,9 ms
Medição 2	2,7 ms	7,3 ms
Medição 3	5,2 ms	4,6 ms
Medição 4	5,9 ms	3,6 ms
Medição 5	4,0 ms	5,9 ms
Experimento 2: utilizando rede móvel 3G		
Medição 1	7,9 ms	4,6 ms
Medição 2	15,8 ms	18,4 ms
Medição 3	13,1 ms	4,0 ms
Medição 4	6,4 ms	5,2 ms
Medição 5	7,2 ms	5,2 ms

Tabela 1: Resultado, em milissegundos, dos intervalos de tempo medidos nos experimentos.

Dos intervalos de tempo coletados nos experimentos pode-se observar que não houve atraso significativo na transmissão da mensagem. Além disso, a diferença na performance utilizando uma rede Wi-Fi e uma rede móvel é irrisória, tendo em vista que as médias dos tempos no cenário da detecção da água foram, respectivamente, 4,2 ms e 10,08 ms, já no cenário da interrupção foram 5,06 ms e 7,48 ms.

Um outro resultado interessante obtido do projeto foi a possibilidade da elaboração de um histórico da disponibilidade de água no sistema, ou seja, um histórico do abastecimento da água pela empresa responsável. Esse

histórico é gerado através da coleta de dados do sensor e com a ajuda da ARTIK Cloud.

Com esses dados pode-se construir conhecimentos tais como quais meses/estações do ano o fornecimento de água é mais prejudicado, além de quantas horas por dia há disponibilidade de água e com isso aplicar programas de prevenção e conscientização, preparando assim a população para essas situações. Um exemplo desses dados pode ser visto na Figura 6.

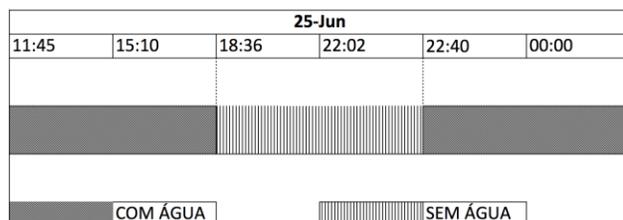


Figura 6: Exemplo do histórico da disponibilidade de água.

## 6 Conclusões

Após o desenvolvimento e testes deste trabalho conclui-se que o sistema poderia ajudar a parte da população, que até hoje sofre com a não regularidade na distribuição de água, a monitorar de forma mais otimizada o abastecimento de água nas residências.

Além desse ponto, é relevante ressaltar também a comodidade e a flexibilidade que o usuário tem ao utilizar um serviço como o proposto neste trabalho. Sendo uma solução viável, relativamente barata e escalável para diversos tipos de residências.

Um outro ponto relevante com a utilização do sistema é o impacto financeiro, pois como a tubulação não permanece aberta após a interrupção do abastecimento de água, não há medição errada no hidrômetro causada pela passagem de ar na tubulação.

Além disso, o sistema também poderia ajudar os órgãos responsáveis contribuindo para o planejamento de políticas de prevenção com base nos meses com pouca disponibilidade de água. Bem como ajudar no planejamento de abastecimentos alternativos, como por exemplo o uso de carros-pipa, aos bairros que mais sofrem com essa situação.

## 7 Melhorias e trabalhos futuros

O objetivo deste trabalho foi uma prova de conceito para otimizar o monitoramento do abastecimento de água nas residências utilizando para isso conceitos de Internet das Coisas. Por essa razão, por se tratar de apenas um protótipo e por falta de recursos não houve nenhuma instalação do sistema em um ambiente real.

Um dos pontos de melhoria mais relevante é o desenvolvimento de um sensor de água de maneira comercial que tenha o mesmo propósito, o qual foi demonstrado nesse trabalho. Além disso, a válvula usada neste trabalho foi uma de máquina de lavar, sendo, portanto, não ideal para uma tubulação com maior vazão de água como é o esperado para um sistema de abastecimento de água domiciliar.

Um melhoria na experiência do usuário quanto a aplicação Android é a implementação de uma função que enviasse não só o estado atual do sensor mas também o estado atual da válvula, garantindo assim para o usuário que a ação enviada por ele foi bem sucedida.

Ainda referente a aplicação Android pode-se implementar uma rotina para mostrar ao usuário o histórico da disponibilidade da água na sua residência, para que através disso ele possa se preparar para meses com pouco fornecimento de água.

Além disso, uma nova funcionalidade poderia ser acoplada ao sistema, caso fosse utilizado um sensor de vazão de água, o qual calcularia a quantidade de água já utilizada no mês, estimando assim uma média do valor atual da conta.

Uma melhoria a fim de evitar o desperdício de água seria adicionar ao reservatório de água potável uma boia eletrônica, então assim que o reservatório atingisse o limite máximo a boia enviaria um comando para a Raspberry Pi para fechar a tubulação. Desta forma seria coberto mais um cenário que pertence ao escopo desse trabalho.

## Referências

- [1] Folha de São Paulo, Racionamento em São Paulo faz 1 ano e amplia transtornos, <http://www1.folha.uol.com.br/fsp/cotidiano/209992-rationamento-em-sp-faz-1-ano-e-amplia-transtornos.shtml>, Junho 2017.
- [2] F. Karray, M. W. Jmal, M. Abid, A. M. Obeid e M. S. BenSaleh. Demo: EarnArdui – A Wireless Sensor Network Testbed for Water Pipeline Monitoring. *International Conference on Embedded Wireless Systems and Networks (EWSN)*, Uppsala, páginas 244-245, 2017
- [3] G. P. Galeale, E. Siqueira, C. B. H. Silva e C. A. Souza. Internet das coisas aplicada a negócios – um estudo bibliométrico. *Revista de Gestão da Tecnologia e Sistemas de Informação*. Vol. 13, No. 3, páginas 423-438, 2016.
- [4] Malte Ubl e Eiji Kitamura, Apresentando WebSockets: trazendo soquetes para a web, <https://www.html5rocks.com/pt/tutorials/websoc>

- kets/basics/, Junho 2017.
- [5] Samsung ARTIK Cloud, *Getting Started*, <https://developer.artik.cloud/documentation/getting-started/>, Junho 2017.
  - [6] Samsung ARTIK Cloud, *Code Samples*, <https://developer.artik.cloud/documentation/tutorials/code-samples/>, Junho 2017.
  - [7] Samsung ARTIK Cloud, *An IoT remote control*, <https://developer.artik.cloud/documentation/tutorials/an-iot-remote-control.html#monitor-light-state/>, Junho 2017.
  - [8] NPM, *onoff*, <https://www.npmjs.com/package/tm-onoff/>, Junho 2017

## MONOGRAFIA DE FINAL DE CURSO

### Avaliação Final (para o presidente da banca)\*

No dia 10 de Julho de 2017, às 10:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **CARLOS ALBERTO LUCIANO DA SILVA JUNIOR**, orientado pelo professor **Edison de Queiroz Albuquerque**, sob título **Monitoramento domiciliar do abastecimento de água utilizando conceitos de Internet das Coisas**, a banca composta pelos professores:

**Ricardo Ataíde de Lima**

**Edison de Queiroz Albuquerque**

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 9,0 ( Nove )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá \_\_\_\_\_ dias para entrega da versão final da monografia a contar da data deste documento.

  
\_\_\_\_\_  
RICARDO ATAÍDE DE LIMA

  
\_\_\_\_\_  
EDISON DE QUEIROZ ALBUQUERQUE

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.