Energesis: *Fog Smart Meter* para Hospedagem Compartilhada

Rafael Sampaio Pereira[1], **José Souza de Jesus**[2], **Matheus Thiago Marques Barbosa**[1], **Eric Bernardes Chagas Barros**[1], **Maycon Leone Maciel Peixoto**[1]

[1]Programa de Pós-Graduação em Ciência da Computação(PGCOMP)
Instituto de Matemática – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

[2]Instituto Federal de Educação Ciência e Tecnologia da Bahia (IFBA)
Jequié – BA – Brasil

{rafaspereira, matheus.thiago, eric.bernardes}@ufba.br,

jose.jesus@ifba.edu.br, mayconleone@dcc.ufba.br

Abstract. *Some homeowners host travelers in their own homes, sharing some rooms for a short period. Along with this practice, there is a need for a resource that allows to measure the energy consumption when the property is being shared. In this way, the guest should only pay for the energy consumed by him. The difficulty of delivering this consumption report is because the houses have only a general measuring system. Thus, this article shows the development of a system for monitoring the consumption of electricity for shared hosting. The system was developed under the Internet of Things (IoT) architecture using Fog and Cloud Computing. The System is composed of a Smart Meter, and a Web application. In addition to the construction of the intelligent measurement architecture, the performance evaluation was performed with the purpose of showing the system's ability to measure electricity consumption and be used as a resource for owners in shared hosting.*

Resumo. *Alguns proprietários de imóveis hospedam viajantes em suas próprias residências, compartilhando apenas parte do imóvel por um curto período. Junto com essa prática, existe a necessidade de um recurso que permita mensurar o consumo de energia quando o imóvel está sendo compartilhado. Dessa forma, o hóspede pagará apenas a energia consumida por ele. A dificuldade de entregar esse relatório de consumo é porque as casas possuem apenas um sistema geral de medição. Dessa forma, este artigo mostra o desenvolvimento de um sistema para monitoramento do consumo de energia elétrica para a hospedagem compartilhada. O sistema foi desenvolvido sob a arquitetura IoT (Internet of Things) utilizando a Fog e Cloud Computing. O Sistema é composto por um Smart Meter, e uma aplicação Web. Além da construção da arquitetura de medição inteligente, foi realizada a avaliação de desempenho com o objetivo de mostrar a capacidade do sistema de medir consumo de energia elétrica e ser usado como um recurso para proprietários em hospedagem compartilhada.*

1. Introdução

A economia compartilhada, chamada também de economia de compartilhamento, é uma tendência, de nível global, onde pessoas compartilham bens com outras que

queiram pagar pela experiência do uso temporário e não queiram adquirir esses bens [Ferreira et al. 2016]. No Brasil hospedagem (e.g., **Airbnb**) e transporte (e.g., **99pop, Uber**) são os setores onde a economia de compartilhamento já está bem difundida, em expansão também em outros setores e, em médio e longo prazo, pode chegar a corresponder até 30% do Produto Interno Bruto (PIB) brasileiro, referente ao segmento de prestação de serviços [O Globo 2017]. Isso demonstra a dimensão e a força que esta forma de economia vem conquistando, e a rapidez com que se propaga, devido ao fato de ser mantida pela ideia de um consumo colaborativo.

A *Internet* dispõe de opções para compartilhamentos [Goidanich 2016]. As plataformas, isto é, sites e aplicativos, são os mediadores no compartilhamento. "A economia compartilhada é um novo fenômeno [...] cujo foco é o compartilhamento de bens ociosos, como carro ou apartamento, através de aplicativos de celulares e computadores." [Gurgel 2017, p.8]. Hospedagem compartilhada caracteriza a prática de locar cômodos ou partes de imóveis residenciais, bem como todo o imóvel. Aplicativos apoiam, e até mesmo intermedeiam os contratos entre proprietário e hóspede. Como afirmam [Soares et al. 2017, p.4] "[...] existe um crescimento nas maneiras inovadoras da economia. O turismo observa [...] como a influência desse modo não convencional cresce no segmento através da hospedagem compartilhada [...]".

A hospedagem compartilhada pode durar menos de 27 dias, período mínimo entre as leituras realizadas pelas concessionárias de energia elétrica para aferir o consumo realizado em um imóvel [ANEEL 2016]. Em algumas situações, a hospedagem acontece de forma parcial, isto é, o hóspede não é responsável por todo o imóvel, apenas por uma parte dele. O ideal é que o hóspede pague pelo consumo de energia elétrica que corresponde apenas à parte do imóvel que ocupou, e que foi realizado durante o período que ocupou, ou seja, o pagamento deve ser proporcional ao consumo realizado.

Quando a hospedagem compartilhada se aplica a todo o imóvel e em período equivalente ao período da leitura feita pela concessionária de energia elétrica, é possível atribuir ao hóspede a fatura emitida pela concessionária, caso contrário não será possível determinar o consumo realizado pelo hóspede, baseando-se apenas na fatura de energia. Para determinar precisamente o consumo elétrico realizado pelo hóspede durante a hospedagem compartilhada, e o valor a ser pago por ele de maneira proporcional ao consumo, faz-se necessário um mecanismo para mensurar consumo elétrico em cômodos específicos por tempo determinado.

Esse mecanismo precisa ser transparente, permitindo ao hóspede o acompanhamento em tempo real do seu consumo. Contudo, considerando a grande quantidade de dados que podem estar sendo transmitidos, a velocidade de conexão do hospedeiro e a quantidade de dados que o serviço pode estar recebendo, esse tipo de acompanhamento pode não ser possível. Assim, é necessário utilizar uma arquitetura onde é possível realizar cálculos em tempo real. Nesse sentido, o conceito de névoa tem sido bastante usado pela comunidade acadêmica, onde procedimentos que antes precisariam ir até a Nuvem(*Cloud*) para serem realizados passaram a serem feitos na borda da rede.

Este artigo descreve o desenvolvimento de um sistema distribuído, chamado de Energesis, que atua como medidor de consumo elétrico. Esse sistema pode ser utilizado como ferramenta de apoio ao proprietário ou administrador de um imóvel, pois é capaz

de coletar e armazenar dados do consumo elétrico realizado pelo hóspede durante hospedagem compartilhada. O proprietário desse imóvel pode gerar uma fatura detalhada com dados do consumo e o valor referente ao consumo total do hóspede.

O Energesis foi desenvolvido sob uma arquitetura de 3 camadas: Sensoreamento, *Fog* e *Cloud Computing*, como meio de oferecer agilidade, disponibilidade e escalabilidade. O sistema é composto por um medidor inteligente (*Smart Meter*), uma API *RestFull*, uma aplicação *web* e utiliza persistência poliglota. A *internet* é uma rede sem sincronia, onde latência e tempo de resposta são variáveis de difícil controle. O *Smart Meter* se comunica através da rede local com um web service que está na Névoa(*Fog*), evitando que os dados de consumo sejam enviados diretamente para *Cloud* pela *internet*. Na *Cloud* há uma aplicação *web*, onde é possível cadastrar e atualizar dados do imóvel, tarifas e hóspedes.

Para verificar a capacidade do sistema de medir consumo elétrico e a viabilidade de adotá-lo como um recurso aos proprietários que compartilham imóveis, foi realizada uma avaliação de desempenho, onde foram observados o valor de corrente elétrica identificado pelo medidor e o tempo gasto para determinar esse valor. A avaliação foi feita no algoritmo de medição embarcado no *Smart Meter* do sistema e no algoritmo de medição disponibilizado como parte do projeto *open source* OpenEnergyMonitor. Essa avaliação permite afirmar que o algoritmo proposto nesse trabalho possui tempo de execução menor em relação ao algoritmo tomado como *base-line*. Os valores de corrente elétrica identificado por ambos os algoritmos são próximos.

Neste artigo são detalhados técnicas, conceitos e materiais utilizados para a implementação do sistema, além da avaliação de desempenho no algoritmo de medição. Este trabalho está dividido em 5 Seções. A Seção 1, introdução, contextualiza o trabalho, explica a problemática e introduz a solução proposta. A Seção 2 trata de trabalhos relacionados. A Seção 3 descreve detalhes da solução proposta. A Seção 4 Avaliação de desempenho do algoritmo de medição, apresenta experimentos realizados para verificar a capacidade do sistema de medir o consumo elétrico. A Seção 5 traz considerações finais acerca do trabalho realizado e a Seção 6 apresenta agradacimentos dos autores.

2. Trabalhos Relacionados

É possível encontrar diversos exemplos de sistemas projetados com o intuito de realizar o monitoramento do consumo elétrico em imóveis. Esses sistemas coletam, armazenam e processam os dados de consumo, além de disponibilizar os dados em aplicações *web*.

O OpenEnergyMonitor[1] é um projeto de *hardware* e *software open-source* fundado pelos americanos Trystan Lea e Glyn Hudson. O projeto tem por objetivo oferecer ferramentas de código aberto que auxiliem as pessoas através da visualização detalhada do consumo, o que pode ajudar nas tomadas de decisões em relação a otimização do consumo, a fim de diminuir eventuais desperdícios [OpenEnergyMonitor 2016]. O emonPi é um medidor elétrico do OpenEnergyMonitor baseado na plataforma Raspberry Pi. Ele não monitora apenas o consumo, mas também a produção de energia elétrica, podendo ser util em imóveis que produzem energia por meio de painéis solares [Advances Wales Magazine 2017].

[1]Disponível em `https://openenergymonitor.org`

[de Melo and Roque 2016] apresentam o SGC (Sistema Gerenciador de Consumo), que é uma ferramenta capaz de monitorar o consumo elétrico realizado por equipamentos eletroeletrônicos de uma residência. O sistema foi desenvolvido para auxiliar os usuários em decisões que ajudem a economizar energia. Esse sistema utiliza *hardware* baseado na plataforma Arduino e sensores invasivos para realizar as medições do consumo elétrico, além de uma aplicação *web* onde é possível gerar gráficos do consumo elétrico e estabelecer metas de consumo para um determinado período.

[Jaiswal and Chaubisa 2017] apresentam um medidor de energia inteligente (IEM – *Intelligent Energy Meter*). Solução desenvolvida tendo em vista o cenário do fornecimento de energia elétrica indiano, onde, segundo os autores, as medições podem apresentar erros devido ao fato de haver intervenção humana. Para os autores o uso da mão de obra humana no processo de suspensão do fornecimento de energia elétrica, é um risco a vida. Esse sistema visa auxiliar o consumidor e o fornecedor de energia. Dados de consumo e faturamento são enviados por e-mail e por SMS para os consumidores. O conceito de IoT é utilizado para implementar uma solução para suspensão do fornecimento a distância.

[Yan and Su 2016] apresentam como o aumento do número de *smart meters* tem influenciado a produção de dados e assim apontam a necessidade de se criar arquiteturas que consigam lidar de forma adequada com o grande volume de dados. Com isso, foi proposto a utilização de uma infraestrutura que utiliza a *fog* para armazenamento e processamento de dados que são enviados por *smart meters*.

[Zahoor et al. 2018] propõem uma arquitetura em três camadas utilizando uma infraestrutura em Névoa para processar dados privados coletados de *smart meters* e armazenar os dados públicos na *Cloud*. Esse trabalho mostrou que é possível controlar equipamentos eletrônicos através do sistema centralizado e realizar cálculos de utilização de recursos elétricos.

[Pant et al. 2017] demonstra uma abordagem integrada entre a névoa e o *IoT* e como as *smart grids* utilizando esses recursos podem fornecer uma transparência maior entre o consumidor e o distribuidor de energia. Essa abordagem permite estender os recursos da névoa para o usuário final, introduzindo técnicas de balanceamento de carga e de consumo de energia.

A solução proposta neste artigo permite o monitoramento do consumo elétrico realizado em um imóvel sem a necessidade de intervenção da concessionária de energia. Diferentemente dos trabalhos encontrados, o sistema Energesis possibilita o cruzamento de dados do consumo elétrico com os dados de um hóspede a fim de determinar o valor que este deve pagar com base no consumo que ele realizou. A Tabela 1 apresenta um comparativo entre os recursos do sistema Energesis e os trabalhos encontrados na literatura.

3. O Sistema Energesis

Com o objetivo de auxiliar proprietários e administradores de imóveis, onde se pratique hospedagem compartilhada, foi desenvolvido o sistema Energesis. Trata-se de um sistema distribuído, que tem como entrada dados do consumo elétrico, coletados por sensores, que monitoram a circulação da corrente elétrica nos circuitos elétricos de um determinado imóvel. As medições realizadas pelo sistema não são sobre aparelhos específicos,

Tabela 1. Comparativo entre proposta e trabalhos relacionados.

	Monitora Equipamento	Monitora Imóvel	Consumo Compartilhado	Smart Meter	Fog Computing	Acesso Remoto
IEM		√				√
emonPI		√		√		√
SGC	√			√		√
Yan and Su 2016		√		√	√	
Zahoor et al. 2018		√		√	√	√
Pant et al. 2017		√		√	√	
ENERGESIS		√	√	√	√	√

tendo em vista que circuito elétrico pode acionar vários aparelhos e um imóvel pode ter vários circuitos. Ao entrarem no sistema os dados são pré-processados e persistidos para que possam ser recuperados e analisados posteriormente. Com os dados armazenados o responsável pelo imóvel pode gerar faturas com informações detalhadas.

O sistema pode ser acessado remotamente por um usuário registrado, por exemplo, o proprietário do imóvel, que doravante será chamado administrador. Desta forma, quando a hospedagem compartilhada englobar todo o imóvel, os responsáveis pelo imóvel poderão emitir um resumo do consumo elétrico ao seu hóspede, sem a necessidade de adentrar ao imóvel que está sob contrato de locação.

O *dashboard* (Figura 1) é a parte da aplicação *web* que permite ao administrador observar dados do consumo, além de cadastrar dados de imóveis, hóspedes, faturas e bandeira tarifária. Para ter acesso a essa parte do sistema, o administrador, precisa estar autenticado utilizando *login* e senha de acesso válidos. As faturas não são cobranças automáticas e não possuem natureza fiscal, sendo apenas listas, onde são detalhados consumo elétrico e o valor a ser pago pelo hóspede. Quando o administrador informar a data, horário de inicio e termino da hospedagem compartilhada, exibe na tela o valor do consumo elétrico.

No momento do cadastro de faturas, a aplicação *web* calcula o consumo em Kwh realizado pelo hóspede. O calculo toma como base os valores da tensão elétrica do imóvel, da soma das correntes elétricas armazenadas no banco de dados e do tempo total em que houve consumo de energia elétrica durante a hospedagem. Ao aplicar sobre esse valor dados de tarifas de consumo e alíquotas de impostos, o sistema exibirá na tela o valor a pagar em reais (R$).

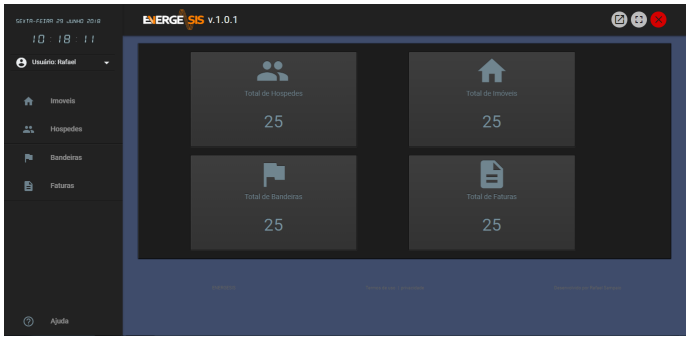

Figura 1. Tela inicial do *Dashboard* do sistema Energesis.

3.1. Arquitetura do Sistema

O sistema Energesis é composto por 3 camadas(Figura 2): A camada de sensoreamento que contém um medidor inteligente (*Smart Meter*); a *Fog* que é composta de um raspberry, que recebe dados do medidor através de uma API *RESTFull*, essa camada responsável por intermediar a transferência de dados entre o medidor e a *Cloud*; e a Nuvem que possui um serviço *web*, um servidor de banco de dados e uma aplicação para visualização dos dados.

A camada mais baixa do sistema utiliza o paradigma de *Internet* das Coisas (IoT), onde ocorre o sensoreamento, nela está presente um medidor inteligente que se comunica através do protocolo HTTP e pode ser identificado em rede através de um endereço IP. Esse medidor utiliza um Arduino MEGA 2560 que serve como plataforma para prototipagem de dispositivos eletrônicos [de Souza et al. 2011]. O medidor é responsável por realizar o tratamento dos sinais de consumo elétrico e estabelecer a comunicação com a camada superior, nesse caso, a Névoa(*fog*).

Os sinais de consumo elétrico são coletados por um sensor de corrente alternada que monitora a instalação elétrica de todo o imóvel, ou de um determinado cômodo, através do eletromagnetismo produzido em volta do condutor que está sendo monitorado. Esse projeto utiliza o sensor de corrente TA 12-100, fabricado pela YHDC, capaz de monitorar correntes alternadas de até 5A [YHDC 2015]. Esse sensor não é invasivo, isto é, ele monitora a instalação elétrica do imóvel sem estar eletricamente ligado a ela.

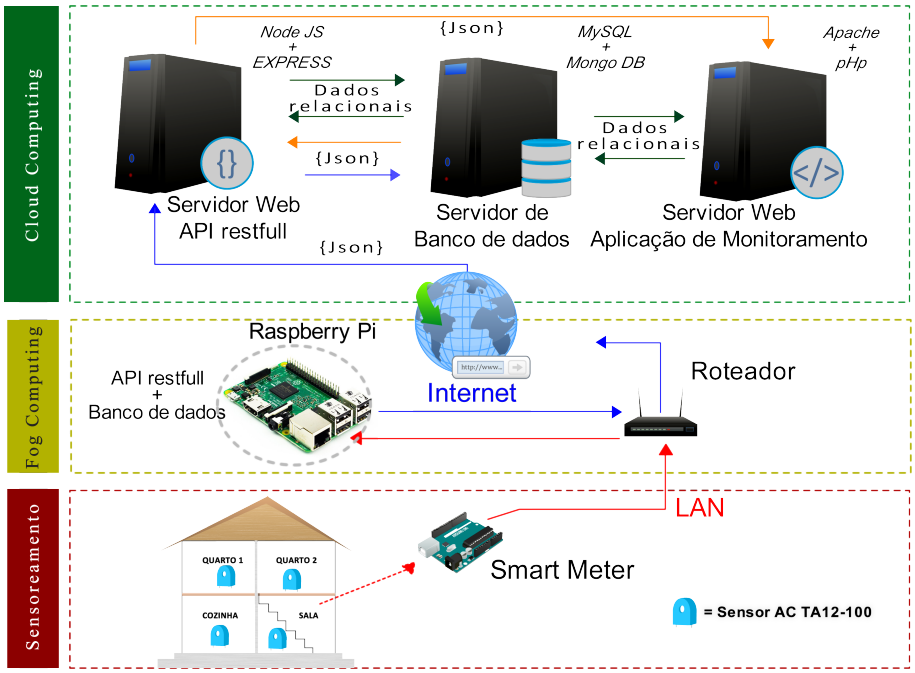

Figura 2. Arquitetura do sistema Energesis.

Os sinais de consumo elétrico coletados pelo sensor são pré-processados pelo sistema embarcado no medidor. Para realizar o tratamento dos sinais o sistema utiliza um algoritmo embarcado que identifica a corrente real (RMS) capturada pelo sensor, ou seja, converte os sinais em valores de corrente elétrica. Esses dados são salvos temporariamente no cartão de memória e depois enviados a um serviço *RESTFull* que está na névoa.

A Fog é camada intermediaria, implementada para evitar perda e atrasos na entrega dos dados ao servidor em *Cloud*. A Névoa disponibiliza recursos com baixa latência, distribuição geográfica, reconhecimento de localização, mobilidade, heterogeneidade e capacidade de processamento devido ao alto número de nós [Mukherjee et al. 2018]. Esse modelo já vem sendo adotado em alguns trabalhos relacionados a *Smart Grids* como é o caso de [Okay and Ozdemir 2016, Barros et al. 2018, Kumar et al. 2018]. Assim o medidor envia os dados a um servidor *Web* no Raspberry PI, dessa forma, em um intervalo de tempo programável, os dados são enviados da Fog a *Cloud*. Caso haja atraso ou distúrbios na comunicação os dados são enviados novamente.

Os dados são adaptados para o formato JSON (*JavaScript Object Notation*) e enviados para uma API desenvolvida com a linguagem JavaScript utilizando o Node.js. Os dados de consumo enviados do medidor a Névoa são: corrente elétrica consumida e o instante de tempo, isto é, horário do consumo. Junto com os dados de consumo, o medidor enviará dois dados: um número identificador do imóvel e uma chave de segurança que irão garantir a autenticidade da informação enviada.

Quando a API do servidor na *Fog* recebe os dados e consegue armazená-los sem nenhum erro ou restrição, é enviado um objeto no formato JSON em resposta ao medidor informando se os dados foram armazenados ou não. Quando a resposta recebida for positiva, o medidor apaga os dados temporários. Em caso de resposta negativa ou serviço indisponível, os dados temporários não são apagados e as leituras continuam até que haja uma nova tentativa de envio.

Em caso de erros, indisponibilidade do serviço *web* da *Fog* ou configurações inválidas o medidor exibe, em um *display* LCD, mensagens de alerta. Nesse *display* o usuário é informado acerca último instante em que os dados foram armazenados na *Fog*, ou em caso de erro exibe o instante da ultima tentativa realizada.

O instante em que os dados são recebidos pela *Cloud* não é o instante real em que o consumo foi realizado. Apesar do envio dos dados ocorrer a cada 2 minutos e os dados serem enviados a nevoa antes de serem enviados a *Cloud*, podem ocorrer atrasos na entrega dos dados, uma vez que os dados são enviados através de uma rede assíncrona. O Arduino não possui um relógio e por isso não tem noção do tempo real, apenas conhece o tempo, em milissegundos, decorrido desde a sua ultima inicialização. Para que consumo e tempo sejam associados e armazenados de maneira correta, o sistema utiliza um modulo RTC (*Real-Time Clock*) DS1307 [Maxim Integrated 2015], que permite ao equipamento trabalhar tempo real.

A *Cloud* é a camada do sistema que armazena permanentemente os dados através do paradigma de persistência poliglota, interagindo com um relacional e outro não relacional. Os dados de hóspede, faturam, administrador, bandeira tarifária e imóveis se relacionam entre si, isso explica a presença de um banco de dados relacional. Nessa camada existe uma API que se conecta ao MySQL, solução reacional de licença gratuita, e verifica se o número identificador e a chave de segurança pertencem ao mesmo imóvel. O consumo elétrico é salvo no banco de dados Mongo DB, escolhido pois o formato de documentos aceito por ele(JSON) é o mesmo utilizado pelo *Smart Meter*, desta forma os dados não precisam de tratamentos antes da persistência. Nessa camada existe uma aplicação *web* desenvolvida na linguagem php por onde os dados são visualizados.

3.2. O transformador de corrente TA 12-100

O sistema realiza as medições de consumo elétrico utilizando um transformador de corrente elétrica alternada, que atua como sensor de corrente não invasivo, isto é, ele transforma valores de correntes elétricas através da indução eletromagnética em sinais analógicos de baixa voltagem (Figura 3). Para que o sinal possa ser medido em *volts* é adicionado um resistor de carga em paralelo à bobina do transformador. Esses sinais são enviados a uma entrada analógica do microcontrolador.

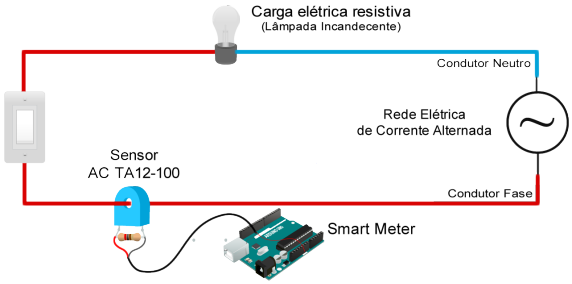

Figura 3. Sensor TA 12-100 atuando em circuito elétrico residencial.

TA 12-100(Figura 4) é um transformador de corrente fabricado pela YHDC, que ao ser utilizado como sensor é capaz de medir correntes de 0A até 5A, e pode ser facilmente substituído por outros transformadores com as mesmas características técnicas. A bobina, isto é, o enrolamento secundário, desses transformadores possui 1000 voltas. O enrolamento primário é o próprio condutor que está sendo monitorado. Para cada 5A que passe pelo primário é disponibilizada uma corrente elétrica de 5mA no secundário. Para compor o resistor de carga foram utilizados resistores em serie com resistência equivalente a $352\,\Omega$.

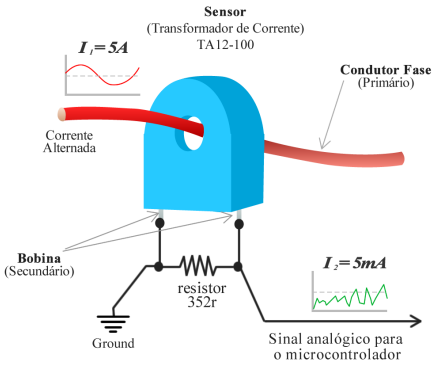

Figura 4. TA 12-100. Características técnicas.

3.3. Algoritmo de medição de corrente elétrica

O sinal analógico enviado pelo sensor de corrente ao microcontrolador precisa ser processado e transformado em valor de corrente para que o consumo possa ser determinado. Nesse trabalho, é proposto um algoritmo desenvolvido em C++, que possibilita ao medidor calcular a corrente elétrica consumida. Para validar a eficácia desse algoritmo foi realizada uma avaliação de desempenho.

3.3.1. A Biblioteca EmonLib

Emonlib é uma biblioteca *open source* disponibilizada como parte do projeto OpenEnergyMonitor (apresentado na Seção 2). Essa biblioteca é composta por um algoritmo dividido em funções que permite dispositivos baseados em Arduino identificar valores de grandezas elétricas, por exemplo, corrente e tensão. A biblioteca não sofreu modificações.

3.3.2. Algoritmo Proposto

O algoritmo proposto para o dispositivo de medição identifica o valor da corrente elétrica convertendo o sinal analógico recebido pelo microcontrolador em voltagem. Para encontrar essa voltagem o algoritmo discretiza o sinal analógico enviado ao ADC do microcontrolador colhendo um determinado número de amostras(Figura 5). O sensor utilizado nesse sistema é unilateral, isto é, monitora apenas um dos sentidos da corrente elétrica alternada, dessa forma são colhidas apenas amostras em um dos ciclos da senoide.

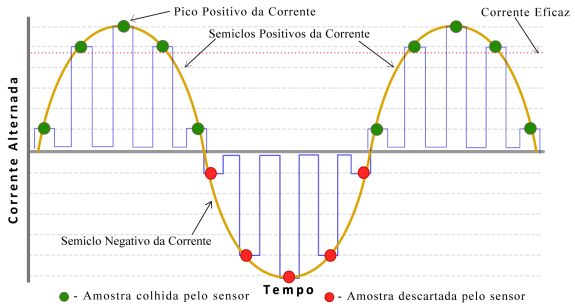

Figura 5. Amostragem realizada pelo algoritmo proposto.

O valor máximo encontrado nas amostras, isto é, o pico de corrente, é o ponto onde o sinal analógico será convertido em *volts*. Conhecido o valor do sinal em *volts*, o algoritmo aplica a regra de eletricidade da primeira lei de *Ohm*. O valor do resistor de carga paralelo ao sensor de corrente é tomado juntamente com o valor em *volts* do sinal analógico para determinar qual a corrente elétrica que passa pelo sensor, utilizando a seguinte formula:

$$\boldsymbol{V} = \boldsymbol{R} \text{ x } \boldsymbol{I},$$

Onde V é o sinal de entrada na porta analógica do microcontrolador, convertido em *volts*, R é o valor do resistor de carga e I é a corrente que passa pelo resistor. O valor de I é multiplicado pelo número de voltas da bobina do transformador. A relação de espiras do transformador é 1000:1, onde há apenas uma volta no primário e mil voltas na bobina do secundário.

$$\boldsymbol{I} = \boldsymbol{I} \text{ x } 1000$$

Para encontrar a corrente RMS(*Root Mean Square*), chamada também de corrente eficaz, que foi consumida o algoritmo divide o valor de corrente encontrado pela raiz quadrada de 2, como na formula a seguir:

$$\boldsymbol{Irms} = \boldsymbol{I} \text{ / } 1.414$$

4. Avaliação de desempenho do algoritmo de medição

O algoritmo de medição de corrente elétrica foi submetido à avaliação de desempenho[2], onde foram realizados experimentos, para que mediante analise de resultados, seja possível verificar o quão confiáveis são as medições realizadas pelo sistema, além de verificar o tempo gasto pelo algoritmo para adquirir, processar e disponibilizar o valor da corrente elétrica consumida. Dessa forma, o tempo gasto no processamento e o *Throughput* (valor da corrente elétrica na saída do processamento) são as variáveis de resposta a serem analisadas.

4.1. Planejamento de Experimentos

A metodologia utilizada no planejamento dos experimentos realizados foi o Projeto Fatorial 2^k, introduzido por Raj Jain, onde k é a quantidade de fatores e 2 a quantidade de níveis por fator. Esse tipo de planejamento permite determinar qual a influência de um determinado fator sobre as variáveis de respostas que estão sendo observadas[Jain 1996]. Para cada fator foram experimentados dois níveis. Nesse trabalho foram considerados 3 fatores: Algoritmo, Carga Elétrica e Amostra. O projeto fatorial resultou na projeção de 8 experimentos (Tabela 2). Cada experimento foi replicado 10 vezes.

Tabela 2. Matriz de planejamento.

	FATORES		
	A	B	C
EXP.	Amostra	Carga Elétrica	Algoritmo
1	100	Lâmpada	EmonLib
2	1000	Lâmpada	EmonLib
3	100	FerroDeSolda	EmonLib
4	1000	FerroDeSolda	EmonLib
5	100	Lâmpada	Proposta
6	1000	Lâmpada	Proposta
7	100	FerroDeSolda	Proposta
8	1000	FerroDeSolda	Proposta

4.1.1. Amostra

O ADC(Conversor Analógico Digital) do Arduino é capaz de realizar até 10.000 leituras por segundo [Arduino 2018]. Foram escolhidos como níveis 1% (100 amostras) e 10% (1000 amostras) do valor máximo que o Arduino é capaz de ler.

4.1.2. Algoritmo

Será analisado o comportamento de um mesmo sistema utilizando dois algoritmos diferentes: O Primeiro é um algoritmo, que faz parte deste trabalho, chamado de ”proposta”durante a avaliação. O segundo é a biblioteca EmonLib, que será utilizada como *base-line* na avaliação de desempenho do algoritmo proposto.

[2]Video com os experimentos disponível em `https://youtu.be/gsKk8rW39GM`

4.1.3. Carga Elétrica

A carga elétrica acionada durante os experimentos será tomada como fator. Serão acionados uma lâmpada incandescente e um ferro de solda. O fator assume Lâmpada e FerroDeSolda como seus níveis.

4.2. Cenários de execução dos experimentos

Os experimentos foram realizados em dois cenários com tensão elétrica de 127v. No primeiro cenário foi monitorado o consumo de uma lâmpada incandescente com potência de 75w e no segundo cenário, foi monitorado o consumo de um ferro de solda com potência de 25w. As potências dos equipamentos são referências informadas pelos fabricantes. Ambos os cenários utilizaram fio flexível com 1,5 mm de espessura.

Para apurar a precisão das medições realizadas pelo algoritmo embarcado, foi utilizado, simultaneamente, um multímetro em serie e com o circuito (Figura 6), isto é, de forma invasiva ao circuito, enquanto o Smart Meter realizou medições de forma não invasiva. O multímetro utilizado foi o Hikari HM-2044 [Hikari 2013], na escala de 20 Amperes.

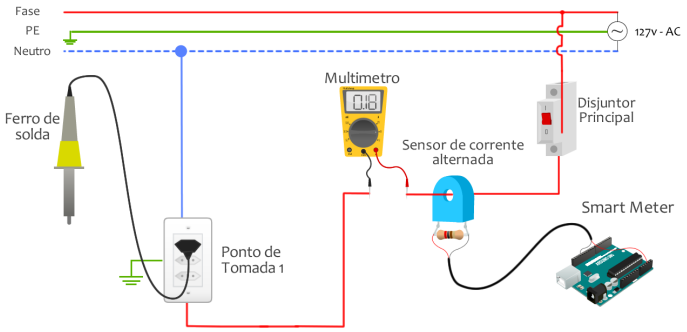

Figura 6. Medição do consumo de um ferro de solda de 25w.

4.3. Análise dos Resultados

O resultado dos experimentos permitem extrair informações sobre o nível de influência que cada fator exerce sobre cada variável de resposta. A avaliação de desempenho não contempla apenas a influencia do algoritmo, e sim a interação do algoritmo com demais fatores e o efeito que essa interação irá produzir na variável de resposta. Os gráficos apresentados na Figura 7 mostram os fatores que mais influenciam nos resultados obtidos em cada variável observada.

O gráfico de efeitos sobre o a variável Tempo Gasto (Figura 7a) mostra que: i) A carga elétrica não influencia no Tempo Gasto; ii) A quantidade de amostras tem maior impacto sobre o Tempo Gasto, ou seja, quanto mais amostras forem colhidas para identificar o valor da corrente elétrica, mais tempo o algoritmo gastará no processamento de dados. 3) O algoritmo Proposto executa, isto é, processa os dados de consumo, mais rápido que o algoritmo *base-line*, a biblioteca EmonLib.

No gráfico de efeitos sobre o a variável *Throughput*(Figura 7b) percebe-se que: 1) A quantidade de amostras não influenciam no *Throughput*. O algoritmo proposto colhe

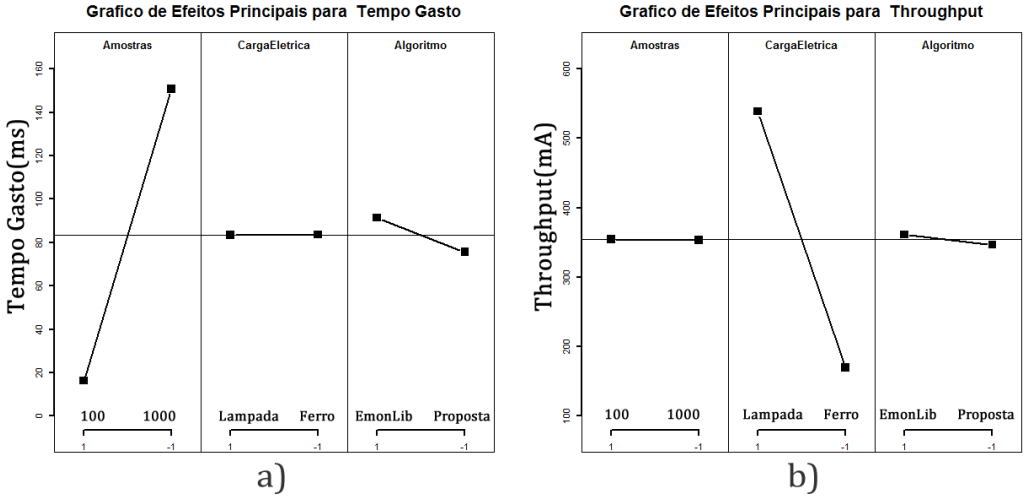

Figura 7. Efeitos dos Fatores nas variáveis. a) Tempo Gasto. b) Throughput.

várias amostras por leitura, porém o valor de corrente eficaz é calculado tomando o valor da corrente de pico, isto é, o mais alto valor lido na amostragem. Dessa forma, independente da quantidade de amostras, a mais alta será considerada, vide item 3.3.2 da seção 3 deste artigo. 2) . A diferença no *Throughput* entre cada algoritmo é pequena. 3) A carga elétrica é o único fator que possui impacto relevante sobre o *Throughput*, ou seja, os algoritmos fazem leituras identificando valores de corrente elétrica próximos. Para que os valores de corrente sofram grandes variações é preciso trocar as cargas elétricas.

5. Considerações finais

Este artigo apresentou o desenvolvimento do sistema Energesis, um solução para medição do consumo elétrico em hospedagem compartilhada utilizando a computação em Névoa, bem como a avaliação do desempenho do algoritmo de medições de consumo elétrico e resultados referentes aos experimentos realizados. Existem na literatura outros Smart Meters que realizam a leitura e processamento de sinais de consumo elétrico a afim de determinar o valor que será pago. Porém em ambientes de hospedagem compartilhada é preciso associar o consumo ao real consumidor. Devido possibilidade de curta duração da hospedagem (e.g. dois dias) em uma semana vários hóspedes podem ocupar o mesmo comodo, sendo essas hospedagens em dias diferentes.

Ao desenvolver um Smart Meter que monitore consumo elétrico uma das etapas mais dificultosas é o preprocessamento de sinais de indução eletromagnética. Medições dessa natureza demandam aprofundado conhecimento em conceitos e grandezas da eletricidade. Além disso os sinais elétricos precisam ser convertidos em sinais digitais e discretizados ainda no Smart Meter antes de ser enviado à camada imediatamente superior. A captura e o preprocessamento desses sinais são feitas por um algoritmo embarcado. A calibragem do sensor, isto é, a identificação do resistor de carga é outro passo onde se deve ter extrema cautela, pois ainda que o algoritmo esteja bem projetado, os valores lidos e processados por ele são recebidos através de entradas analógicas do microcontrolador. Essas entradas são muito sensíveis, o que permitiria valores não-reais entrarem nos sistema caso o resistor de carga do sensor esteja mal dimensionado.Isso tornaria as medições inconsistentes.

O Smart Meter, desenvolvido para o sistema Energesis, permite associar o valor do consumo ao hóspede. A avaliação de desempenho realizada mostra que o algoritmo proposto nesse trabalho possui menor tempo de execução em relação ao algoritmo tomado como *base-line*, e que os valores de corrente elétrica identificado por ambos os algoritmos são próximos, ou seja, o algoritmo proposto possui eficacia no seu proposito, o que habilita os sistema Energesis para ser utilizado como medidor de energia.

Como trabalhos futuros são sugeridos acrescentar o recurso de gráfico de linha, para representar o consumo elétrico de forma gráfica, utilizar um *Shield Wi-fi* para comunicação sem fio do medidor com a *Fog*, identificar separadamente o consumo de monitorado por cada sensor e desenvolver aplicativos moveis para o administrador e hóspede, permitindo ao hóspede acompanhar o consumo realizado por ele antes mesmo de lhe ser apresentada a fatura.

6. Agradecimentos

Os autores agradecem a FAPESB, CAPES, CNPq. Em especial ao MCTI-UFBA, pelo apoio financeiro por meio do Edital PROPCI/PROPG – PROPESQ/UFBA 004/2016.

Referências

Advances Wales Magazine (2017). Energy monitoring unit for a sustainable future. *Advances Wales Magazine*, (82):18.

ANEEL (2016). Como resolver. problemas com recebimento de fatura. http://www.aneel.gov.br/como-resolver/-/asset_publisher/3SAW3SarixVj/content/problemas-com-recebimento-de-fatura/655804, Acesso em 20 jun. 2018.

Arduino (2018). Lenguage reference. analog i/o. https://www.arduino.cc/reference/en/language /functions/analog-io/analogread/, Acesso em 28 nov. 2018.

Barros, E., Peixoto, M., Leite, D., Batista, B., and Kuehne, B. (2018). A fog model for dynamic load flow analysis in smart grids. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6.

de Melo, G. S. and Roque, W. (2016). Sistema de monitoramento de consumo de energia elétrica de aparelhos eletroeletrônicos residenciais. *REVERTE – Revista de Estudos e Reflexões Tecnológicas da Faculdade de Tecnologia de Indaiatuba*. n.14, p.1-19.

de Souza, A. R., Paixão, A. C., Dias, M. A., Uzêda, D. D., Duarte, S., and de Amorim, H. S. (2011). A placa arduino: uma opção de baixo custo para experiências de física assistidas pelo pc. *Revista Brasileira de Ensino de Física*. http://sbfisica.org.br/rbef/pdf/331702.pdf, Acesso em: 6 Dez. 2017.

Ferreira, K. M., Méxas, M. P., Abreu, W., and Mello, P. (2016). Economia compartilhada e consumo colaborativo: Uma revisão da literatura. In: XII Congresso Nacional de Excelência em Gestão & III INOVARSE – Responsabilidade social aplicada, Rio de Janeiro, p.1-21.

Goidanich, M. E. (2016). Airbnb e seus anfitriões: Empreendedorismo, comércio e colaboração na economia de compartilhamento. In: VIII Encontro Nacional de Estudos do Consumo, IV Encontro Luso-Brasileiro de Estudos do Consumo, II Encontro Latino-Americano de Estudos do Consumo Comida e alimentação na sociedade contemporânea, Niterói, p.1-20.

Gurgel, C. G. (2017). A intervenção e regulação estatal sobre o serviço oferecido pelo airbnb no brasil. Master's thesis. Trabalho de Especialização, UFRN, Natal.

Hikari (2013). Hikari: Soluções de confiança. http://sawam.com.br/wp-content/uploads/2015/08/catalogo-hikari-2013.pdf, Acesso: 21 de jun. de 2018.

Jain, R. (1996). In *The art of Computer Systems Performance Analysis*. John Wiley & Sons Inc.

Jaiswal, O. and Chaubisa, D. (2017). Arduino mega and iot based intelligent energy meter (iem) to increase efficiency and accuracy in current billing methodology,. International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, p. 1901-1904.

Kumar, S., Agarwal, S., Krishnamoorthy, A., Vijayarajan, V., and Kannadasan, R. (2018). Improving the response time in smart grid using fog computing. In Kulkarni, A. J., Satapathy, S. C., Kang, T., and Kashan, A. H., editors, *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, pages 563–571, Singapore. Springer Singapore.

Maxim Integrated (2015). Ds1307 64 x 8, serial, i2 c real-time clock. https://datasheets.maximintegrated.com/en/ds/DS1307.pdf, Acesso em: 30 Jun. 2018.

Mukherjee, M., Shu, L., and Wang, D. (2018). Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys Tutorials*, 20(3):1826–1857.

O Globo (2017). Compartilhamento deverá ser 30% do pib de serviços. https://oglobo.globo.com/economia/compartilhamento-devera-ser-30-do-pib-de-servicos-22005402, Acesso: 16 de jun de 2018.

Okay, F. Y. and Ozdemir, S. (2016). A fog computing based smart grid model. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6.

OpenEnergyMonitor (2016). Solar pv monitoring. https://guide.openenergymonitor.org/setup/, Acesso em: 4 jul. 2018.

Pant, V., Jain, S., and Chauhan, R. (2017). Integration of fog and iot model for the future smart grid. In *Emerging Trends in Computing and Communication Technologies (ICETCCT), International Conference on*, pages 1–6. IEEE.

Soares, A., Dias, M., and Filho, L. (2017). A experiência do turista e a hospedagem compartilhada através do uso das novas tecnologias no turismo: O caso do airbnb. *Revista Turismo & Desenvolvimento*. n. 27/28, p.1315 – 1324.

Yan, Y. and Su, W. (2016). A fog computing solution for advanced metering infrastructure. In *Transmission and Distribution Conference and Exposition (T&D), 2016 IEEE/PES*, pages 1–4. IEEE.

YHDC (2015). Current transformer ta 12-100. product specification. http://www.yhdc.us/ENpdf/TA12-100_YHDC.pdf, Acesso em: 24 Abr. 2018.

Zahoor, S., Javaid, N., Khalid, A., Yasmeen, A., and Nadeem, Z. (2018). Fog computing based energy management system model for smart buildings. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 719–727. Springer.