

## Desenvolvimento do sistema de comunicação sem fio para uma tomada inteligente

Development of a wireless communication system for a smart power socket

Andrei da Cunha Lima<sup>1</sup>, Diovane Soligo<sup>2</sup>, Eduardo Silveira Dornelles<sup>3</sup>, Diones de Vargas Dutra<sup>4</sup>,  
Alencar Machado<sup>5</sup> e Vinícius Maranhão<sup>6</sup>

<sup>1,3,4,6</sup>Universidade Federal de Santa Maria – Laboratório de Computação Ubíqua, Móvel e Aplicada, Cachoeira do Sul, Brasil  
cunhalimaandrei1@gmail.com; eduardodornellessilveira@hotmail.com, diones.de@redes.ufsm.br; vinicius.maranhao@ufsm.br

<sup>2</sup>Universidade Federal de Santa Maria – Colégio Politécnico, Santa Maria, Brasil  
diovane.soligo92@gmail.com

<sup>5</sup>Universidade Federal de Santa Maria – Colégio Politécnico, Santa Maria, Brasil  
alencarcomp@gmail.com

### Resumo

*Este artigo apresenta o processo de desenvolvimento e realização de testes de um sistema utilizado para realizar a comunicação sem fio de uma tomada inteligente com um servidor de aplicação. A tomada inteligente utiliza um circuito que faz medições de tensão e corrente da rede elétrica. Os resultados destas medições são enviados a servidores, que a partir destas informações, tomam decisões em ambientes inteligentes. Para desenvolver o sistema de comunicação, foi utilizada a placa ESP8266. Para tal, primeiramente a placa com um código simples que mostrava o estado de acionamento de um botão através de uma página web. Após a realização dos testes iniciais, o sistema de comunicação foi incorporado ao sistema da Tomada Inteligente (Smart Power Socket) e novos testes de recepção e envio de informações foram realizados. Através dos testes, foi possível observar que a utilização do ESP8266 foi satisfatória considerando o uso da Tomada Inteligente em um ambiente residencial.*

**Palavras-chave:** Tomada inteligente; Ambientes inteligentes; Comunicação sem fio

### Abstract

*This article presents a process of developing and testing a system used to perform a wireless communication of a smart socket with an application server. The smart socket current uses a circuit that makes voltage and current measurements instantly. The results of these measurements are sent to servers that take decisions for smart environments. For the development of the communication system, an ESP8266 board was used. To obtain a card with the simple code that shows the trigger state of a button through a web page. After the entry of the initial tests, the communication system was integrated with Smart Socket and new tests of reception and sending of information were realized. Through the testing, it was possible to observe the use of ESP8266 as a satisfactory way of using Smart Decision-making in residential environments.*

**Keywords:** Smart socket; Smart environments; Wireless communication

## 1 Introdução

A redução do consumo de energia elétrica é considerada um dos principais desafios em vários setores da economia atual (Schiefelbein et al., 2018). Na agenda de 2030 para o desenvolvimento sustentável das Nações Unidas (ONU), o tema Cidades Inteligentes e Comunidades Sustentáveis é um dos principais objetivos e nesse aspecto grandes avanços em ambientes inteligentes estão cada vez mais relacionados à criação de ambientes sustentáveis (Nijkamp et al., 2014)(ANEEL, 2008). Esses ambientes têm como função reduzir os impactos ambientais e proporcionar aos seus cidadãos, através de um custo acessível, uma melhora na qualidade de vida (Rosa, 2013).

No que diz respeito ao consumo de energia elétrica no Brasil, o terceiro setor com maior gasto foi residencial (ANEEL, 2008), o que demonstra a necessidade da criação de ambientes inteligentes, com aplicativos capazes de capturar informações do ambiente do usuário e analisá-las. Um dos maiores problemas relacionados ao consumo doméstico é a falta de informação sobre o consumo dos aparelhos e componentes eletrônicos durante o seu uso (Rosa, 2013). Uma razão que evidencia tal situação é a falta de sensores de baixo custo, que realizam a verificação de tensão e corrente de aparelhos para calcular a energia gasta por eles.

Sistemas de controle estão tornando-se cada vez mais empregados em diversos setores da economia como uma boa forma de monitorar o sistema em que estão inseridos, e controlá-lo quando é necessário. Então visando solucionar este problema, a falta de informação sobre o consumo, a criação de uma aplicação capaz de fornecer informações específicas e em tempo real sobre o consumo e o gasto total energético de aparelhos se torna de grande utilidade para que o usuário possa gerenciar seus custos relacionados ao uso de energia elétrica (Schiefelbein et al., 2018).

Nesse contexto, foi projetada a *smart socket*. Uma tomada inteligente capaz de coletar os dados brutos do aparelho conectado a rede elétrica, como corrente e tensão elétrica, interpretá-los e transmitir ao usuário. O usuário então pode programar situações de consumo, em que ele deseja ser notificado para que se necessário tome decisões, como por exemplo: um aparelho de ar-condicionado que está ligado durante horas e atingiu sua cota de consumo diário, estipulado pelo usuário. A tomada então enviará uma notificação ao usuário, que poderá desligar remotamente o aparelho para evitar que ele permaneça consumindo energia (Schiefelbein et al., 2018).

A segunda versão da *smart socket* foi desenvolvida para coletar os dados do aparelho elétrico e enviá-los para servidores de gerenciamento. Este trabalho tem objetivo elaborar um sistema de comunicação sem fio através do módulo ESP8266 para a *Smart Socket*, a fim de que transmissão de dados com servidor seja *wireless*, ou seja, uma comunicação que não tem a necessidade do uso de cabo.

Para realizar a conexão entre o módulo Wi-Fi e a rede local, foi utilizada uma programação baseada pela IDE do Arduino, em que o microcontrolador do Arduino, o ATMEGA328P, é programado e fica responsável para realizar a conexão através da serial com o ESP8266. A integração de um sistema sem fio permite que a *Smart Socket* se torne mais prática e versátil para ambientes residenciais. Com a conexão estabelecida, o Arduino será responsável por realizar as medições e acionar a tomada, além de enviar dados via comunicação serial para o ESP8266.

O artigo está estruturado da seguinte forma: Na Seção 2 é apresentado a *Smart Socket*. Na Seção 3 é apresentado o novo sistema de comunicação sem fio proposto para a *Smart Socket*. Na Seção 4 são apresentados os testes realizados com o sistema de comunicação proposto e na Seção 5 são apresentadas as conclusões.

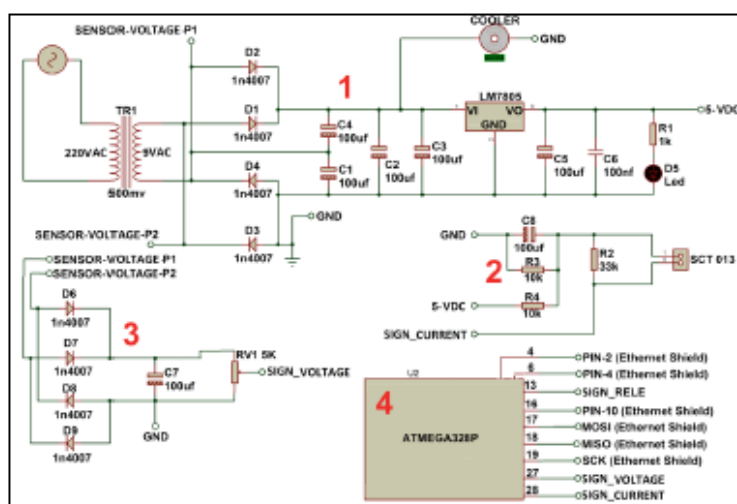
## 2 Smart Socket

Nesta seção é apresentado o circuito da *Smart Socket* e as principais funções da tomada, como a coleta de dados e a conexão com o servidor de gerenciamento.

### 2.1. Circuito da tomada

O circuito da tomada inteligente é responsável por realizar as medições de tensão elétrica e corrente, controlar o fornecimento de energia do aparelho conectado e transmitir essas informações para o servidor de gerenciamento. A estrutura da tomada pode ser dividida em 4 partes, de acordo com a numeração apresentada na Figura 1.

**Figura 1** – Circuito da tomada (Schiefelbein et al., 2018)



Na seção relacionada à energia (1), um transformador é responsável por rebaixar a tensão de 220VAC para 9VAC, logo após uma ponte retificadora e capacitores são utilizados para transformar o sinal em corrente contínua. A seção 2 é responsável pela medição de corrente com o uso de um sensor de corrente, que é ligado em série com a carga, para analisar a quantidade de corrente que a carga demanda. A seção 3 realiza a medição de tensão. A seção 4 realiza o controle, através de um microcontrolador. Observa-se que na Figura 1, 4 pinos são utilizados para ligar um *Shield Ethernet*, que é utilizado para realizar a conexão entre a tomada e o servidor. Todavia, este método de comunicação possui desvantagens, já que precisa da utilização de cabo para realizar a conexão.

## 2.2. Programação utilizada no microcontrolador

O microcontrolador foi programado através da IDE do Arduino, com o uso de bibliotecas padrão e bibliotecas encontradas na internet. A escolha pela IDE do Arduino foi baseada na facilidade que este ambiente de desenvolvimento de software proporciona ao usuário. Um trecho de código correspondente à segunda versão da tomada é apresentado na Figura 2.

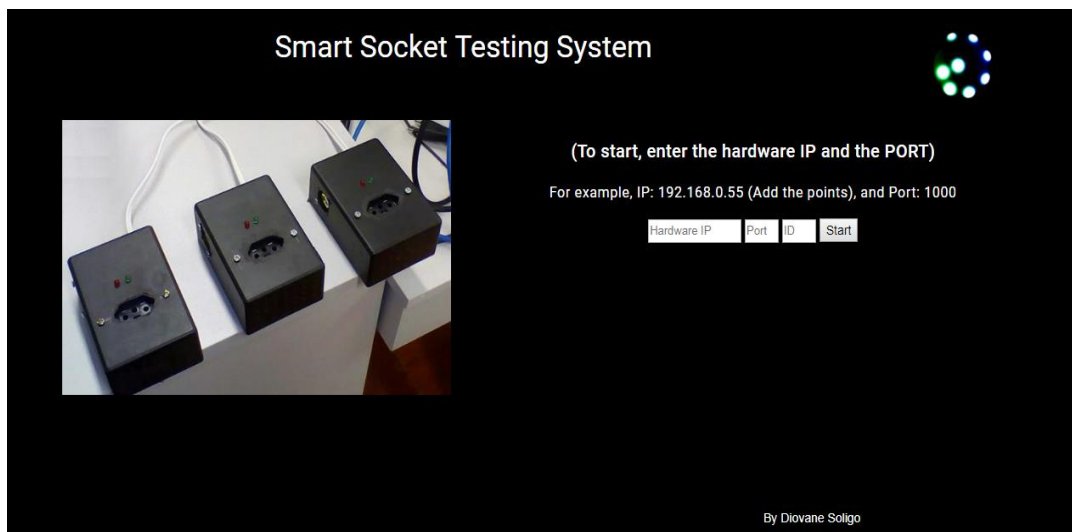
**Figura 2** – Código gravado no ATMEGA328P

```
if (comando == "umidade") {
  client.println(int(DHT.humidity));
} else if (comando == "temperatura") {
  client.println(int(DHT.temperature));
} else if (comando == "hello") {
  client.println("OKDSV2ENCAPH@NTOM");
} else if (comando == "voltagem") {
  client.println(v);
} else if (comando == "corrente") {
  client.println(c);
} else if (comando == "potencia") {
  client.println(v * c);
} else if (comando == "ligar") {
  digitalWrite(7, HIGH);
} else if (comando == "desligar") {
  digitalWrite(7, LOW);
} else if (comando == "status") {
  if (digitalRead(7) == HIGH) {
    client.println("on");
  } else {
    client.println("off");
  }
}
```

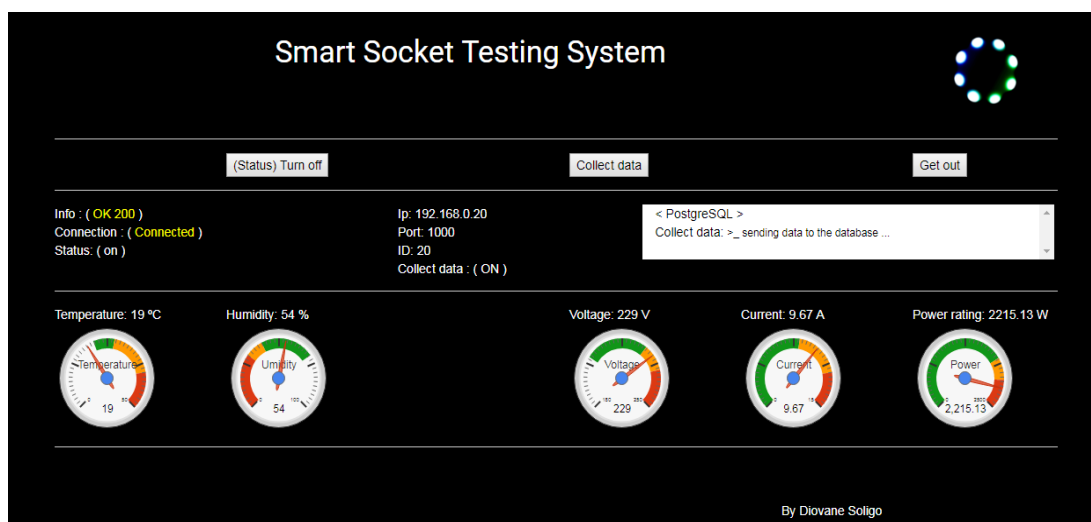
Observa-se que a estrutura do código é baseada em comandos condicionais. O código é responsável por tratar as requisições realizadas pelo servidor, tratá-la de acordo com o requisitado e responder às requisições realizadas utilizando um *Shield Ethernet*.

## 2.3. Servidor de gerenciamento

O servidor é responsável por requisitar comandos à tomada e informar os dados coletados para o usuário. Este servidor foi desenvolvido em linguagem de programação Java (Soligo et al., 2018).

**Figura 3** – Tela de inicialização do sistema (Soligo et al., 2018)

Quando o servidor é inicializado, uma tela para inserção de dados é aberta. Um exemplo é apresentado na Figura 3. Nele, é necessário que o usuário informe o IP, o ID da tomada e a porta de acesso que a comunicação tomada-servidor irá acontecer. Quando o usuário clicar em *start*, a conexão é estabelecida, e o servidor começa a requisitar informações da tomada.

**Figura 4** – Tela de dados enviados pela tomada(Soligo et al., 2018)

Após o usuário fornecer os dados requisitados na tela de inicialização, a tela de dados coletados pela tomada é aberta (Figura 4). Nesta tela, informações de temperatura, umidade relativa da tomada, corrente e potência são apresentadas ao usuário com uma taxa de atualização de 5 segundos. Também é descrito o status da tomada, se está em funcionamento ou não, o IP, a porta de acesso, o ID, e se a coleta de dados está ocorrendo normalmente. O usuário pode ligar ou desligar o aparelho conectado através do botão *'(Status) Turn off'* e também encerrar a conexão com o servidor, pressionando o botão *'Get out'*.

### 3 Sistema de comunicação sem fio

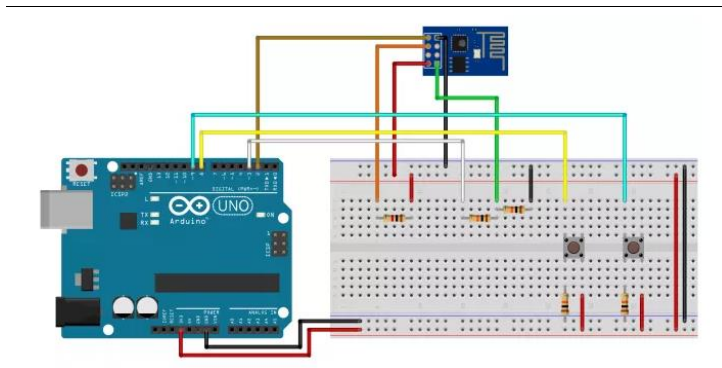
Nesta Seção é apresentado o sistema de comunicação da *Smart Socket* com o servidor de gerenciamento, que é realizado através do módulo ESP8266. Este sistema é composto pelo circuito da tomada, após as mudanças realizadas e o código responsável por estabelecer essa conexão. A primeira etapa na definição do sistema de comunicação foi a construção de um circuito teste com o módulo ESP8266, apresentado na seção a seguir.

#### 3.1. Implementação de Testes com o Módulo ESP8266

Para conectar o ESP8266, responsável por realizar a ligação com a rede Wi-Fi e estabelecer a comunicação sem fio, foi desenhado um circuito na *proto-board* com suas ligações ao Arduino (Figura 5). Foram adicionados fios de coloração vermelha e preta destinados a fornecer a alimentação ao módulo, saindo das portas de alimentação do Arduino de tensão 3.3 volts. Além de alimentar o módulo, precisamos estabelecer uma comunicação entre o módulo e o Arduino.

A comunicação é realizada através da porta serial, ou seja, dois fios são conectados nos pinos 2 e 3 do Arduino e depois ligado ao módulo. Estes fios são chamados de RX e TX do Arduino, sendo que o RX, de coloração marrom, é responsável por receber e o TX, de coloração verde, responsável a enviar os dados ao módulo. Na ligação do fio TX, é necessário o uso de resistores para dividir a tensão, já que a tensão de saída nas portas digitais do Arduino é de 5 volts e a do módulo é de 3.3 volts.

**Figura 5** – Circuito utilizado para realizar a conexão entre ESP8266, Arduino e rede Wi-Fi (De Oliveira, 2017)



Neste circuito, também foram adicionados dois botões e resistores para ligá-los. Os botões foram adicionados temporariamente com objetivo de tornar mais prático os testes de validação de funcionamento da comunicação sem fio, ou seja, enviar sinais indicando se ele está pressionado ou não. Após a construção do circuito de teste com o ESP8266, foi implementado um código de testes, que apresenta uma página HTML à cada mudança de estado dos botões utilizados no circuito. O trecho de código que realiza a apresentação das informações ao usuário à cada nova requisição é apresentado na Figura 6.

**Figura 6** – Trecho de código que apresenta a página HTML com o status do circuito

```

// Verifica se o ESP8266 esta enviando dados
if (esp8266.available())
{
  if (esp8266.find("+IPD, "))
  {
    delay(300);
    int connectionId = esp8266.read() - 48;

    String webpage = "<head><meta http-equiv='refresh' content='3'>"; //Programação em HTML responsável por criar a web page
    webpage += "</head><h1><u>Comunicacao WiFi</u></h1>"; //BODY da programação

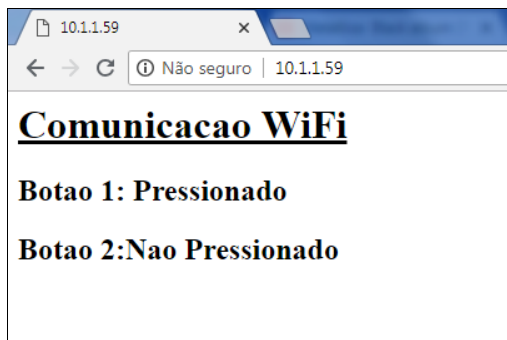
    webpage += "<h2>Botao 1:";
    int a = digitalRead(bot1);
    if (a==1)
      webpage += " Pressionado";
    else
      webpage += "Nao Pressionado";
    webpage += "</h2>";

    webpage += "<h2>Botao 2:";
    int b = digitalRead(bot2);
    if (b==1)
      webpage += " Pressionado";
    else
      webpage += "Nao Pressionado";
    webpage += "</h2>";
    String cipSend = "AT+CIPSEND="; //String responsável pelo comando AT Cipsend, que envia dados
    cipSend += connectionId;
    cipSend += "\n";
  }
}

```

### 3.2. Resultados obtidos com o circuito teste para ESP8266

Após a implementação do circuito teste, a conexão entre o ESP8266 e o Arduino foi estabelecida, assim como a conexão com a rede Wi-Fi. Além disso, o servidor local do módulo está de acordo com o esperado, ou seja, descrevendo a situação atual dos dois botões do circuito teste. Na Figura 7 é apresentada a página gerada através do circuito de testes, apresentando o status dos botões. As informações que são enviadas para a *web Page* possuem uma taxa de atualização de dados de 2 segundos.

**Figura 7** – Interface que apresenta o status dos botões do circuito de testes

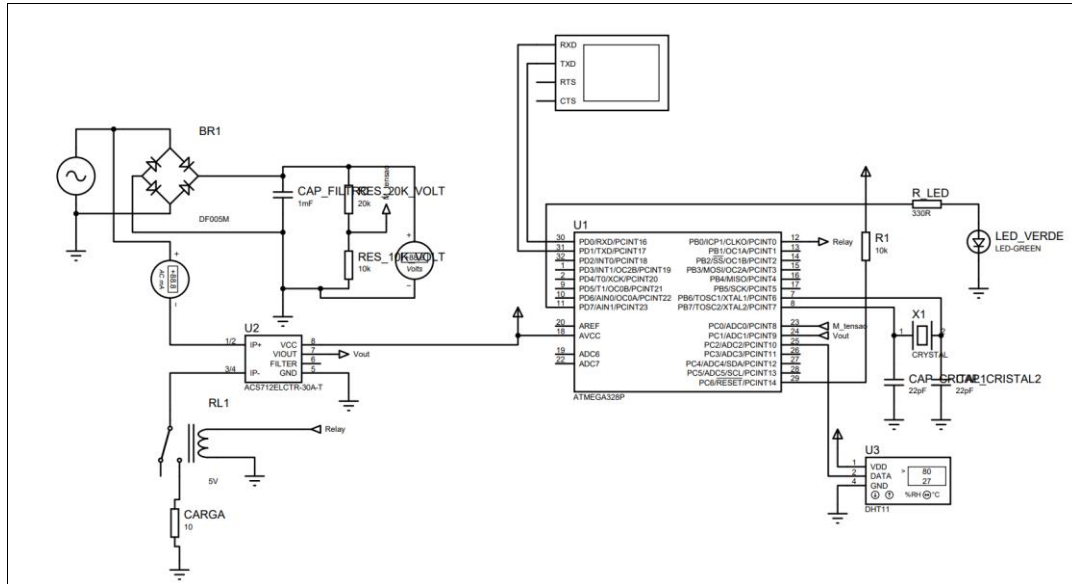
### 3.3. Utilização do Módulo ESP8266 na *Smart Socket*

Ao finalizar os testes com o circuito teste, o processo de implementação da ESP8266 na *smart socket* foi iniciado. A implementação compõe a terceira versão da tomada, que além de um novo de comunicação com servidor, possui uma pequena melhoria na parte de coleta de dados referente a tensão.

O novo circuito inclui o módulo ESP8266, ao invés da *Shield Ethernet*. O ESP8266 se conecta com a rede Wi-Fi da residência e envia os dados para o servidor através da internet. A medição da tensão é realizada a partir de um divisor de tensão, composto por dois resistores colocados em série. Desta forma, o microcontrolador do Arduino é encarregado pela medição da tensão no meio dos resistores. Para rebaixar a tensão, foi utilizado um transformador 220 VAC - 9 VAC. O divisor

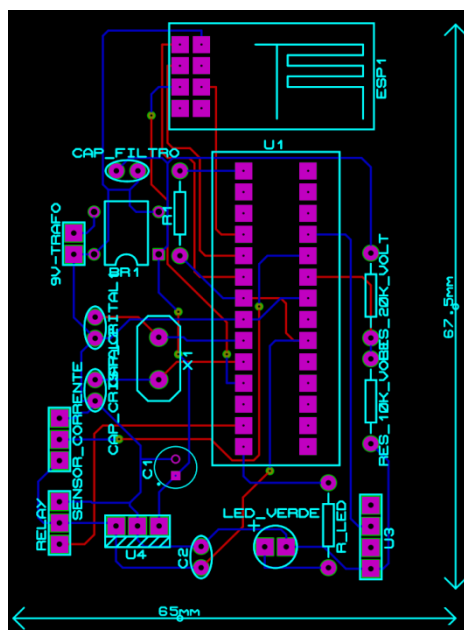
foi inserido após a retificação da tensão e para isso foram utilizados dois resistores de 250K. O esquema do novo circuito é apresentado na Figura 8.

**Figura 8** – O esquema da terceira versão da tomada



Utilizando o esquema apresentado anteriormente, foi projetado o PCB (placa de circuito impresso) que fará com que o circuito fique mais compacto. O PCB é apresentado na Figura 9.

**Figura 9** – O PBC da terceira versão da tomada



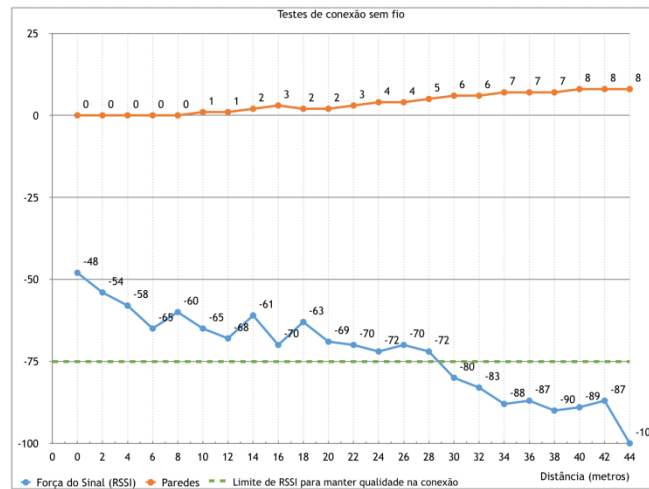
Após a definição do circuito e da programação na nova versão da *Smart Socket*, a mesma foi submetida à testes de comunicação para avaliar a transmissão de dados.



## 4 Testes e Resultados

Na Figura 10 são apresentados os resultados dos testes de conexão realizados com o protótipo da *Smart Socket*. Os testes de alcance foram realizados em ambientes semelhantes a uma residência, ou seja, com paredes e armários interferindo na força do sinal. Os testes foram realizados com um roteador wireless N 150Mbps TL-WR741ND (OpenWRT, 2018). Depois de ocorrer a perda de sinal devido a distância, o módulo não teve problema algum para se reconectar a rede e refazer este processo sozinho.

**Figura 10** – Resultados dos testes de alcance da conexão



No gráfico são apresentadas: (i) a quantidade de paredes entre o roteador e a *smart socket* e (ii) a força do sinal (RSSI - *Received Signal Strength Indicator*) de acordo com a distância e número de paredes (quanto menor o valor, menor a força de sinal). De acordo com Unni *et al.*, (2015), o valor de  $-75\text{dBm}$  é o mínimo considerado aceitável para aplicações com trocas de pacotes constantes. Desta forma, a *smart socket* com o módulo ESP8266 foi capaz de operar sem problemas se conectada à uma distância máxima de 29 metros e com 5 paredes entre ela e o roteador *wireless*.

## 5 Conclusões

A terceira versão da *smart socket* obteve uma melhora significativa em relação a segunda versão. Agora com um sistema de comunicação sem fio, garantindo uma maior mobilidade. Além da melhoria na comunicação, ela conta com um novo método de medição de tensão, que comparado ao anterior é mais compacto, e um design mais *flat*, devido ao uso do PCB no seu desenvolvimento. Como resultados futuros, espera-se construir uma sala inteligente com diversas tomadas, possibilitando o controle preciso de energia elétrica utilizada na sala através do sistema desenvolvido.

## Agradecimentos

Os autores agradecem à Universidade Federal de Santa Maria (Programa FIPE-UFSM), FAPERGS (Programa ARD) e CNPq (Programa PIBIC) pelas bolsas concedidas aos autores e auxílio financeiro para realização do projeto.

## Referências

AGENCIA DE ENERGIA ELÉTRICA DO BRASIL [Internet]. Atlas de Energia Elétrica do Brasil. Available from: <http://www2.aneel.gov.br/arquivos/pdf/atlas3ed.pdf>

DE OLIVEIRA, Sérgio. Internet das Coisas com ESP8266, Arduino e Raspberry Pi. Novatec Editora, 2017.

NIJKAMP, Peter; PERRELS, Adriaan. Sustainable cities in Europe. Routledge, 2014.

OPEN WRT[Internet]. Table of Hardware: TP-Link TL-WR741ND. Available from: <http://wiki.openwrt.org/toh/tp-link/tl-wr741nd>.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA [Internet]. Brasília: Ministério do Planejamento, Orçamento e Gestão (BR) [cited 2009 feb 27]. Pesquisa Nacional por Amostra de Domicílios. Síntese de Indicadores 2005. Available from: <http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2005/default.shtm>

DATASUS [Internet]. Brasília: Ministério da Saúde (BR) [cited 2006 oct 20]. Departamento de Informática do SUS – DATASUS. Available from: <http://w3.datasus.gov.br/datasus/datasus.php>.

ROSA, L.M.F. Sensorização, fusão sensorial e dispositivos móveis: contribuições para a sustentabilidade de ambientes inteligentes.[dissertation]Universidade do Minho/UMINHO; 2013.133p.

SCHIEFELBEIN, U. H. ; SOLIGO, D. ; MARAN, V. ; OLIVEIRA, JOSÉ PALAZZO M. DE ; LIMA, J. C. D. ; MACHADO, A. .Pervasive System Based on Situation-Awareness for Feedback of Energy Efficiency. In: XIV Simpósio Brasileiro de Sistemas de Informação (SBSI), 2018 jun 4-8;Caxias do Sul, Brasil. Anais do Simpósio Brasileiro de Sistemas de Informação (SBSI), 2018.

SOLIGO, D. . Internet das coisas aplicada a eficiência energética. [monography]. Santa Maria: Colégio Politécnico/UFSM; 2018.

UNNI, S., et al. Performance measurement and analysis of long range wi-fi network for over-the-sea communication. In: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on. IEEE, 2015. p. 36-41.