

## **Desenvolvimento de um sistema de controle remoto para automação de saídas digitais e analógicas por meio de uma aplicação web e uma aplicação local utilizando linguagem python**

### **Development of a remote control system for automation of digital and analog outputs, through a web application and a local application developed in python**

DOI:10.34117/bjdv7n8-010

Recebimento dos originais: 07/07/2021

Aceitação para publicação: 02/08/2021

#### **Lúcio Rogério Júnior**

Mestre em Inovação Tecnológica pela Universidade Federal do Triângulo Mineiro –  
UFTM - Professor pela Universidade de Uberaba - UNIUBE  
E-mail: lucio.rogerio@hotmail.com

#### **Lucas de Oliveira Estevam**

Graduando em Engenharia da Computação pela Universidade de Uberaba - UNIUBE  
Aluno de Iniciação Científica - UNIUBE  
E-mail: lucas.estevam@edu.uniube.br

#### **Mateus Henrique Rodrigues Ribeiro**

Graduando em Sistemas de Informação pela Universidade de Uberaba - UNIUBE  
Aluno de Iniciação Científica - UNIUBE  
E-mail: mateuskb@edu.uniube.br

#### **Guilherme Henrique Alves**

Mestre em Engenharia Elétrica pela Universidade Federal de Uberlândia – UFU  
Doutorando em Engenharia Elétrica pela Universidade Federal de Uberlândia - UFU  
Professor pela Universidade de Uberaba - UNIUBE  
E-mail: guilherme.henrique.alves@outlook.com

#### **Wellington Mrad Joaquim**

Mestre em Ensino de Ciências e Matemática pela Pontifícia Universidade Católica –  
PUC - Professor pela Universidade de Uberaba - UNIUBE  
E-mail: welington.joaquim@uniube.br

#### **Fabiola Eugênio Arrabaça Moraes**

Mestre em Estatística pela Universidade Federal de São Carlos - UFSCar  
E-mail: prof.famoraes@hotmail.br

#### **Antônio Manoel Batista da Silva**

Doutor em Engenharia Elétrica pela Universidade Federal de Uberlândia - UFU  
Professor pela Universidade de Uberaba - UNIUBE  
E-mail: tomanel.tamanel@gmail.com

## RESUMO

O presente artigo apresenta o desenvolvimento e a construção de um protótipo, controlável remotamente, através da internet, a partir de recursos e dispositivos de baixo custo, e de fácil implementação. Desenvolvido com tecnologias de código aberto, o sistema de controle remoto foi construído a partir de um Raspberry Pi, e subdividido em três partes: a aplicação web (desenvolvida em Html, Css, Javascript, PHP e MySQL); a aplicação local codificada em Python e executada no Raspberry; e dois protótipos físicos, compostos por uma placa de disparo de cargas com relés eletromecânicos e uma fonte de alimentação variável. Para acessar remotamente, foi construído um sistema de Login, gerando uma maior segurança e uma interface de botões que, ao serem acionados, realizam requisições a um servidor, que grava o status do botão acionado no Banco de dados SQL (Linguagem de Consulta Estruturada, do inglês, Structured Query Language). Foram criadas, também, barreiras de proteção para simular condições de intertravamentos entre saídas. Foram feitos testes para verificar a integridade de cada sistema. Tendo em vista que não apresentaram falhas, concluiu-se que é possível controlar remotamente elementos digitais (motor e lâmpada) e uma grandeza analógica (fonte de tensão), mostrando um exemplo de aplicação da automação com IoT (Internet das coisas, do inglês, Internet of Things), através de tecnologias de baixo custo, fácil implementação, e sendo referência para outros trabalhos mais complexos.

**Palavras-chaves:** IoT, Raspberry Pi, PWM, Automação.

## ABSTRACT

This article presents the development and construction of a remotely controllable prototype, using the internet, with low-cost and easy-to-implement resources and devices. Developed with open source technologies, the remote control system was built from a Raspberry Pi, and subdivided into three parts: the web application (developed in Html, Css, Javascript, PHP and MySQL), the local application coded in Python and executed on the Raspberry, and two physical prototypes, consisting of a load switching board with electromechanical relays and a variable power supply. To access it remotely, a Login system was built, which results in a greater security, and a button interface that, when triggered, make requests to a server, which records the status of the button pressed in the SQL Database (Structured Query Language). Protection barriers were also created to simulate interlocking conditions between exits. Tests were carried out for each system to check the integrity of it and it did not present any failures, concluding that it is possible to remotely control digital elements (motor and lamp) and an analog quantity (voltage source), showing an example of application of automation with IoT (Internet of the things), through low cost technologies, easy implementation, and being a reference for other more complex works.

**Key-words:** IoT, Raspberry Pi, PWM, Automation.

## 1 INTRODUÇÃO

A ideia de, mecanicamente, reproduzir a ação humana, vem de muito antes da Revolução Industrial. Segundo Guarnieri (2010), a automação deu seus vestígios desde a China antiga, sendo os engenheiros do período helenístico os primeiros a mostrarem

sucesso em suas construções. No entanto, foi apenas na Revolução Industrial que a automação começou a ser utilizada como uma solução para os processos de produção.

Atualmente, com o desenvolvimento da tecnologia, a automação deixou de ser exclusivamente aplicada à indústria e começou a estar presente em casas, para uso doméstico, como mostrado no trabalho de Alkar e Buhur (2005). A aplicação da automação de sistemas e processos está, cada vez mais relacionada ao atual crescimento da internet das coisas. Isso foi constatado por Evans (2011), que projetava um número de 50 bilhões de aparelhos conectados à internet, até ano de 2020, possibilitando o controle de equipamentos e dispositivos de forma remota, via internet.

Com o crescente surgimento de tecnologias embarcadas com conectividade a internet, e a enorme disponibilidade de plataformas de computação cada vez mais acessíveis, inúmeras opções de baixo custo e fácil acesso comercial aparecem como possibilidades para serem utilizadas em projetos de máquinas, segundo Sandeep et al. (2015), equipamentos e dispositivos, para Bermúdez-Ortega et al. (2015), aplicações comerciais como apresentado por Resende, et al. (2021), atendendo à necessidade de realizar automação e controle nesses sistemas.

Normalmente, os equipamentos e dispositivos utilizados para automação em processos industriais são de alto custo, e nem sempre estão disponíveis para acesso rápido, devido às barreiras comerciais que são impostas por representantes e fornecedores. Desta forma, a escolha de tecnologias de baixo custo, acordo com Sonali, Lagu e Sanjay (2014), e que frequentemente são utilizadas em trabalhos acadêmicos, de acordo com Anusha e Mahadevaswamy (2018) e Kumar e Rajasekaran (2016), torna-se muito vantajosa devido à acessibilidade e à grande quantidade de informações disponíveis em trabalhos e projetos já desenvolvidos.

Em consonância com essas premissas, o presente trabalho tem como objetivo apresentar o desenvolvimento de um sistema de automação utilizando IoT, em que duas placas eletrônicas são controladas de forma remota, mediante o uso de uma interface web, com a utilização de dispositivos de hardware de baixo custo e programados com software livre.

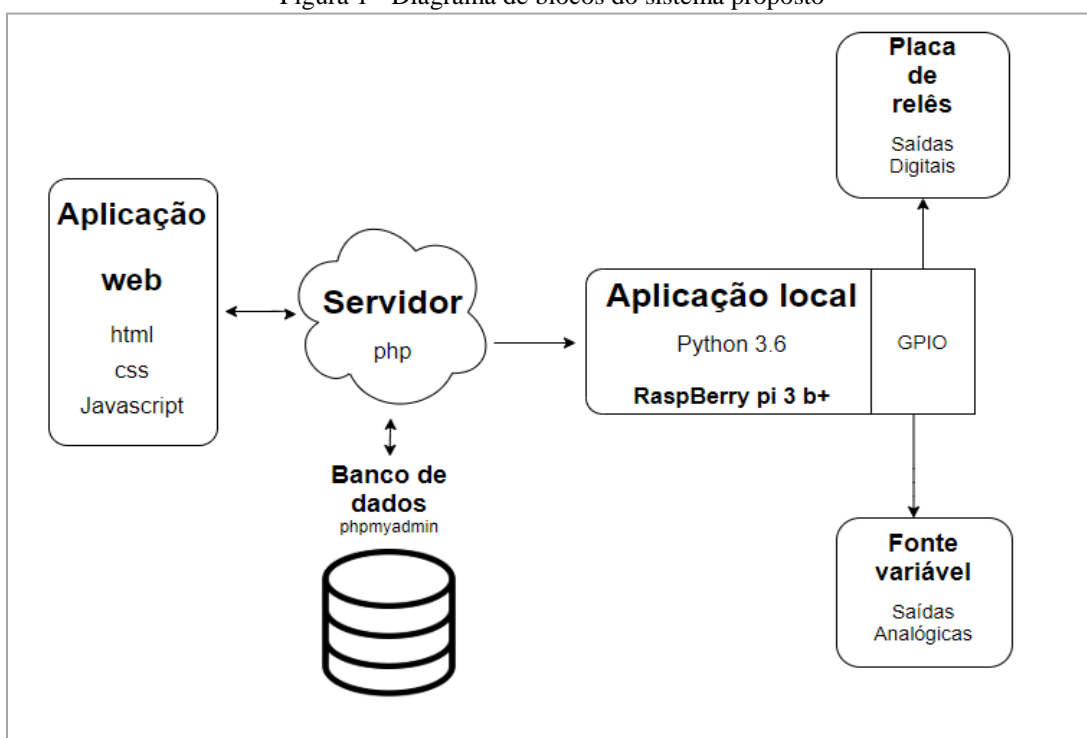
## **2 MATERIAL E MÉTODOS**

O sistema proposto inclui um conjunto de elementos constituídos por hardware e software, e pode ser dividido em três principais partes: uma aplicação web, uma aplicação local e duas placas eletrônicas. A aplicação web é responsável pelo controle

e monitoramento de equipamentos e dispositivos físicos, sendo gerenciados por um usuário, de forma remota, por meio de uma interface gráfica. A aplicação local realiza a comunicação com um servidor de banco de dados e gerencia os dispositivos de hardware.

Um microcomputador embarcado está interligado com duas placas eletrônicas que têm por função demonstrar o controle de variáveis digitais e uma variável analógica, exemplificando dispositivos que poderão estar incluídos em um processo a ser controlado. O diagrama de blocos abrange os processos gerados pelo software e todos os componentes de hardware ligados ao sistema computacional (Figura 1).

Figura 1 - Diagrama de blocos do sistema proposto

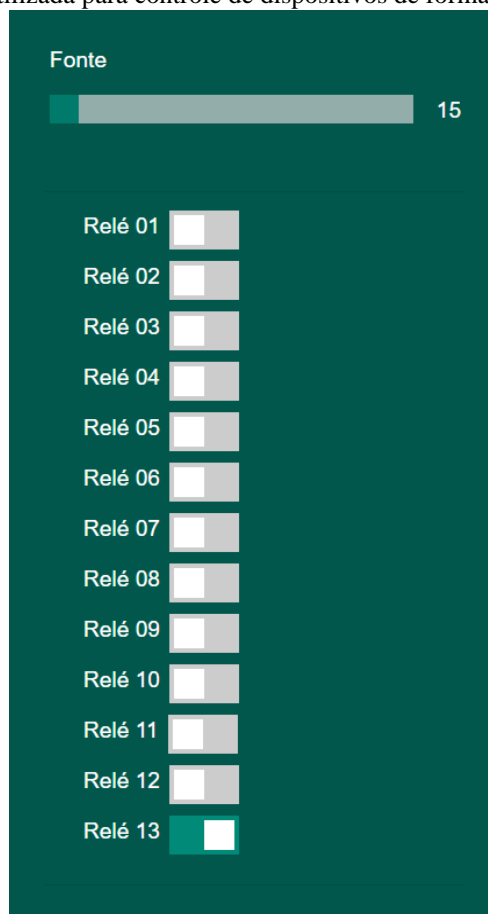


Fonte: Autoria própria (2021)

Uma aplicação web utilizando a linguagem de programação PHP como back-end e JavaScript como linguagem front-end foi construída com o objetivo de controlar remotamente uma placa de relés e uma fonte de alimentação com saída variável. Conforme apresentado na Figura 2, a interface do usuário (UI – do inglês, User Interface) da aplicação web é composta por um conjunto de botões que possibilita o controle individual, de forma remota, de 13 dispositivos discretos e oferece o controle de uma variável analógica por intermédio de um objeto de intervalo (range). O status

dos elementos controlados pode ser verificado por meio da alteração das cores dos componentes da interface.

Figura 2 - UI utilizada para controle de dispositivos de forma remota (mobile)



Fonte: Autoria própria (2021)

A aplicação web é hospedada em um servidor remoto, o que permite o acesso do usuário via internet. Esta aplicação recebe as informações de status dos relés, enviadas pelo usuário, e as transmite para um banco de dados, que é consultado frequentemente pela aplicação local, desenvolvida em Python e sendo executada no Raspberry Pi.

Os códigos para a construção da aplicação web e da aplicação local foram escritos no Ambiente de Desenvolvimento Integrado (do inglês, Integrated Development Environment - IDE) de código livre Visual Studio Code, desenvolvido pela Microsoft. Diferentes bibliotecas foram utilizadas para o desenvolvimento de cada etapa do sistema.

O sistema back-end em PHP foi escrito empregando a microestrutura (Micro Framework) SLIM que funciona em um ambiente APACHE. O sistema front-end foi

escrito em Javascript, utilizando a biblioteca JQuery para a realização de requisições ao servidor de manipulação do HTML.

Um sistema de login para a segurança do componente foi criado. O sistema utilizava as mesmas tecnologias que as outras partes do módulo, adicionando apenas a tecnologia JWT. O JWT é um sistema de token seguro para transmitir informações, garantindo imutabilidade, uma vez que há uma assinatura digital.

O usuário, ao digitar o login e a senha, teria seus dados codificados em base 64 e enviados para o servidor, que os decodifica e compara a senha com o HASH, gravado com segurança no banco de dados. Caso o login esteja correto, é criado um JWT, contendo informações básicas do usuário, que é enviado para o JAVASCRIPT, que o guarda no 'local storage' do navegador, para que possa ser acessado quando necessário.

Caso o usuário clique em 'logout' (desconectar), o navegador terá o JWT apagado e o servidor gravará que o JWT não é mais válido, o que o torna seguro para que não haja fraudes no acesso, caso já tenha sido desconectado.

Figura 3 - UI utilizada para controle de dispositivos de forma remota



Fonte: Autoria própria (2021)

A aplicação em Python realiza funções distintas, sendo responsável por ler os dados do servidor, verificando o status dos relés e do valor da tensão de saída da placa de fonte. Outra função da aplicação é mostrar, na Interface Gráfica do Usuário (do inglês, Graphical User Interface - GUI), disponibilizado através do Raspberry Pi, os dados atuais do sistema, e enviar os novos dados fornecidos pelo usuário para as placas eletrônicas. Dessa maneira, foram utilizadas as seguintes bibliotecas: Requests (Requisição ao servidor), Tkinter (Montagem do GUI), GPIO (Controle das saídas GPIO e PWM), dentre outras não essenciais. Os comandos para importação dos módulos são mostrados na Figura 4.

Figura 4 - Códigos utilizados para importações dos módulos

```
import tkinter as tk
import tkinter.font as tkFont
from PIL import ImageTk, Image
import ast

from RPi import GPIO
from time import sleep

import sys, os
import requests
import base64
import json
```

Fonte: Autoria própria (2021)

A aplicação em Python foi desenvolvida para receber as informações enviadas pelo usuário por meio da aplicação web e transmiti-las para as placas de fonte e relés. Esta aplicação local foi desenvolvida para trabalhar dentro de um microcomputador, e o modelo escolhido para esse projeto foi o Raspberry Pi 3 B+.

A conexão de todas as aplicações é realizada da seguinte forma: a cada clique efetuado na interface do usuário (Figura 2), o JQuery envia uma requisição POST para o servidor PHP informando a desativação ou ativação de um botão. Assim, o PHP recebe essa informação e grava no banco de dados um novo registro (Figura 5), contendo a situação de todos os botões, incluindo o botão recém-clicado e copiando o valor do restante dos botões.

Figura 5 - Exemplo de registro feito no banco de dados

inp_pk	inp_b_rele1	inp_b_rele2	inp_b_rele3	inp_b_rele4
713	0	1	1	0
712	0	1	1	0
711	0	1	1	0
710	0	1	1	0
709	0	1	0	0
708	0	1	1	0
707	0	1	0	0
706	0	1	1	0
705	0	1	0	0
704	0	1	1	0
703	0	1	0	0
702	0	1	1	0
701	0	1	0	0

Fonte: Autoria própria (2021)

A aplicação Python hospedada no Raspberry realiza, a cada 0,5s, uma requisição GET, que recebe o último registro realizado no banco de dados e a compara com o registro anterior. Caso haja qualquer modificação, uma função referente ao botão com valor alterado é executada, além de atualizar a interface gráfica do usuário do Python (Figura 6), mostrando o valor de cada botão.

Figura 6 - Interface gráfica do usuário da aplicação *Python*



Fonte: Autoria própria (2021)

O Raspberry Pi 3 é mostrado na Figura 7. Trata-se de um microcomputador que possibilita a instalação de diferentes sistemas operacionais, sendo composto por 48 pinos GPIO, slot para cartão SD, 4 portas USB. Todo o hardware do sistema está ligado a esse processador, que opera com o fornecimento de 3,3V (RASPBerry, 2021).

O sistema operacional utilizado no Raspberry Pi foi o Raspberry Pi OS, anteriormente chamado de Raspbian. O Raspberry Pi OS foi escolhido justamente pela otimização com o hardware e pela variedade de aplicativos e linguagens pré-instaladas, incluindo a linguagem Python, que foi utilizada para o sistema local, o que permite que o sistema seja utilizado de forma genérica a todos os tipos de projeto.

Figura 7 - Microcomputador *Raspberry Pi 3 B+*



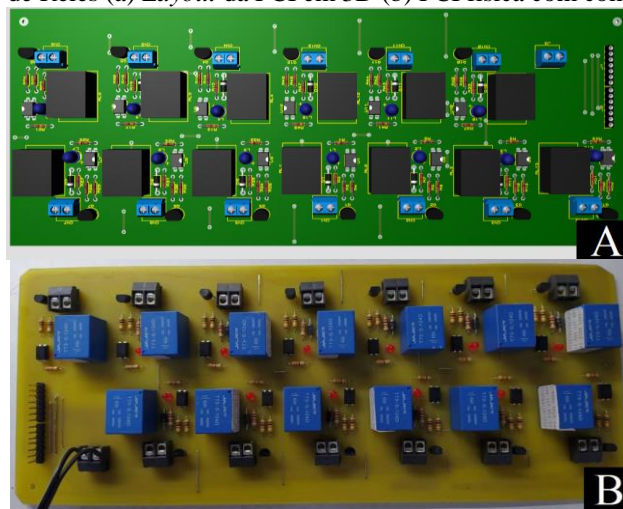
Fonte: Acerbo do autor (2021)



Duas placas eletrônicas foram construídas com o propósito de representar dispositivos digitais e analógicos que podem ser controlados pelo sistema. A placa de relés foi construída para o controle de saídas digitais, utilizando relés eletromecânicos que poderão acionar cargas de até 10A operando com tensões alternadas de até 220V ou 30V em corrente contínua.

O módulo de relés é composto por 13 saídas, disponibilizando, em cada uma delas, um contato normalmente aberto (NA) e normalmente fechado (NF), combinados em um mesmo terminal comum. Os sinais de 3,3V fornecidos pela GPIO do Raspberry Pi são aplicados em optoacopladores presentes na placa. A grande vantagem de um acoplador óptico é o isolamento elétrico entre os circuitos, em que o ponto comum de um circuito é diferente do ponto comum do outro, podendo aterrar um dos circuitos e deixar o outro em flutuação (BOGART, 2001), garantindo a isolação elétrica entre o microcomputador e a placa de disparo de cargas. Na Figura 8 é apresentado o projeto de layout em 3D e a placa eletrônica, depois de construída.

Figura 8 - Módulo de Relés (a) *Layout* da PCI em 3D (b) PCI física com componentes montados



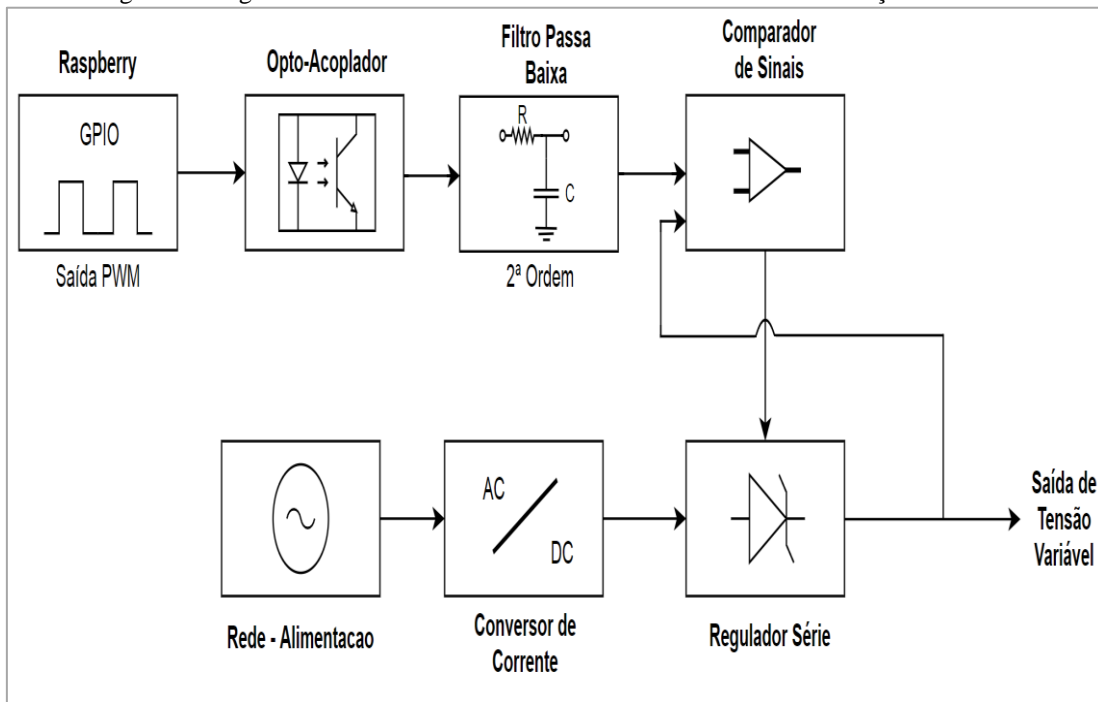
Fonte: autoria própria (2021)

Uma fonte de alimentação linear com saída variável foi desenvolvida e construída, disponibilizando sua tensão de saída de acordo com ajustes realizados de forma remota por meio da aplicação web. Para produzir a variação da tensão da fonte, uma saída digital da GPIO operando em modo de Modulação de Largura de Pulso (do inglês, Pulse Width Modulation - PWM) foi utilizada, possibilitando a variação da tensão de um regulador do tipo série presente na saída do circuito.

Pode-se visualizar em maiores detalhes os blocos principais (Figura 9) que compõem a fonte do tipo linear que foi desenvolvida. O circuito de potência é

composto por um retificador em ponte com filtro capacitivo e um regulador de tensão a transistor. O sinal PWM enviado pela GPIO do Raspberry aciona um acoplador óptico, que realiza o isolamento entre os circuitos. Um filtro Passa Baixa de segunda ordem é aplicado ao sinal, transformando o sinal de onda quadrada em um sinal contínuo (MALVINO; BATES, 2011).

Figura 9. Diagrama de Blocos do circuito eletrônico da fonte de alimentação variável

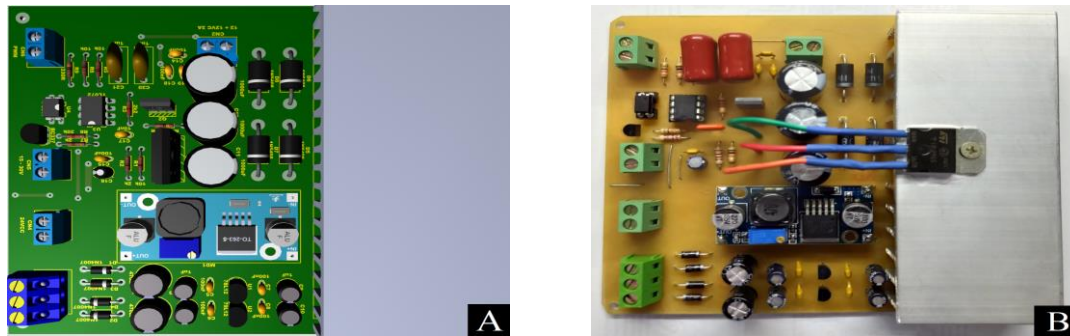


Fonte: Autoria própria (2021)

Um circuito comparador de tensão com realimentação negativa é utilizado para controlar a queda de tensão no regulador série, coletando uma amostragem do sinal de saída para ser comparado com o sinal de controle. O sinal de PWM tem a função de definir o valor da tensão de saída desejada (Set Point) variando a tensão de saída de acordo com a largura de pulso.

As placas eletrônicas da fonte de alimentação e de relés foram realizadas em três etapas: em um primeiro momento, foram elaborados os circuitos eletrônicos e realizadas simulações de funcionamento; na segunda etapa, foram desenvolvidos os layouts das placas de circuito impresso (Figuras 8a e 10a); e, por último, foram confeccionadas placas de circuito impresso, para posterior montagem dos componentes e testes (Figuras 8b e 10b).

Figura 10 - Fonte com saída Variável (a) *Layout* da PCI em 3D (b) PCI física com componentes montados



Fonte: Autoria própria (2020)

### 3 RESULTADOS E DISCUSSÃO

Depois de finalizado o projeto de software e hardware, foram realizados testes de funcionamento, integrando a aplicação Web com o banco de dados que está presente no servidor, por meio de requisições realizadas ao back-end, com variados valores, para ter certeza de que a comunicação estava sem falhas. Foram realizados testes na comunicação com a aplicação local do RaspBerry Pi, por intermédio das mesmas requisições a “endpoints” do servidor, isto é, dos links gerados pelo servidor para executar funções no banco de dados. Os dois testes comprovaram a efetividade da comunicação entre o servidor e as aplicações, alcançando os objetivos iniciais.

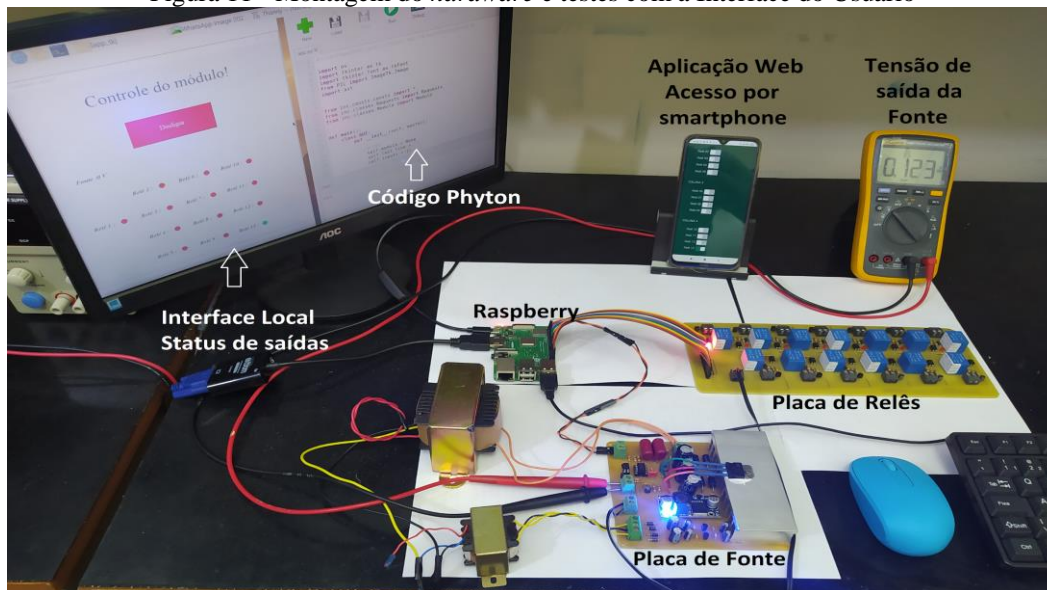
É preciso considerar que, em algumas aplicações que possam utilizar a placa de relés, intertravamentos entre saídas digitais poderão ser necessários; como, por exemplo, no acionamento de um Motor de Indução Trifásico (MIT), em dois sentidos de rotação, em que são utilizadas duas saídas que nunca poderão ser acionadas ao mesmo tempo. Desta forma, foram feitas algumas operações de intertravamento, ou seja, barreiras de proteção, por meio dos códigos, que não permitiriam ao usuário realizar operações que pudessem causar danos ao módulo. Assim, foram criadas barreiras no código Javascript, no servidor com a linguagem PHP, e na aplicação local codificada em Python. As barreiras de intertravamento se mostraram efetivas, como foi visto nos testes realizados.

A aplicação local tem a função de escrever valores no GPIO e, para isso, também foi testada. Foi utilizado um osciloscópio digital para verificar os valores de tensão das saídas digitais do RaspBerry Pi e os testes mostraram valores esperados. Para isso, foi utilizada a biblioteca nomeada GPIO, de código aberto, disponível para importação em Python e com suporte do RaspBerry.

Após o código ser publicado no servidor, foram efetuados os últimos testes, que consistiam em acessar a aplicação web de um computador e observar alterações no módulo devido ao acionamento realizado on line, possibilitando afirmar que o teste foi efetivo, tendo toda a parte de software em funcionamento conforme planejado.

Na Figura 11, pode-se observar todo o conjunto de componentes utilizados para o teste em bancada, de todos os itens de hardware e software. A última parte dos testes constituiu-se no acionamento das duas placas eletrônicas, verificando disparo de cargas e o controle da tensão de alimentação da fonte.

Figura 11 - Montagem do *hardware* e testes com a Interface do Usuário

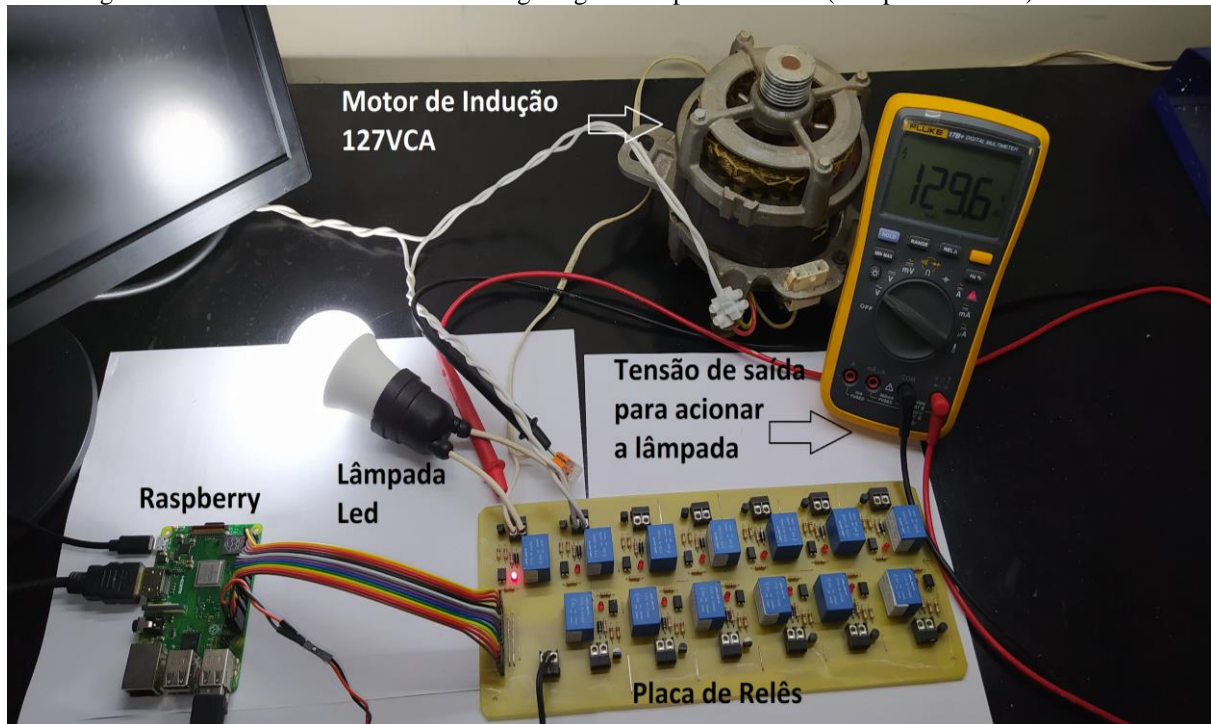


Fonte: Acervo do autor (2021)

A placa de relés foi conectada ao Raspberry Pi por meio de um flat cable, fazendo com que todas as 13 saídas de relés fossem acionadas com êxito, por intermédio de um acoplador óptico, sem nenhuma conexão elétrica entre o circuito de potência e o Raspberry Pi. Uma lâmpada em 127VCA e um motor em 220VCA (Figura 12) foram conectados às saídas a relés, e foram acionados de forma remota, mostrando um pequeno exemplo de possibilidade de controle com o sistema proposto.

A fonte de alimentação variável também recebeu, em sua entrada de controle, o sinal PWM, produzido por uma saída digital do *Raspberry*. Considerando que, para alimentação da fonte, foi utilizado um transformador com secundário com 24VCA, foi possível ajustar a tensão de saída entre 0 a 30VCC. A tensão de saída foi medida utilizando um voltímetro digital, em duas situações: primeiramente, a vazio, e, depois, com uma carga de 2A. Nas duas condições, verificou-se o perfeito funcionamento da fonte e do ajuste produzido na saída.

Figura 12 - Testes de acionamento de cargas ligadas na placa de relés (Lâmpada e Motor)



Fonte: Acervo do autor (2021)

#### 4 CONCLUSÕES

O desenvolvimento e a construção do sistema de automação, controlado remotamente com o propósito de efetuar a comunicação entre duas placas eletrônicas, com uma interface web, foi possível, demonstrando a implementação de um sistema de baixo custo e fácil de ser desenvolvido, podendo ser facilmente acessado em qualquer dispositivo.

Dessa forma, este artigo apresentou, de forma genérica, o controle remoto de dispositivos físicos, utilizando saídas digitais e PWM. Além disso, indica que todos os recursos de software e o Raspberry podem ser usados para controles mais sofisticados, tendo este trabalho como referência para o desenvolvimento de outros, que envolvem automação e Iot.

#### AGRADECIMENTOS

Os autores agradecem a Universidade de Uberaba (UNIUBE) pelo suporte financeiro e por oferecer toda infraestrutura para o desenvolvimento de todas as etapas deste trabalho.

## REFERÊNCIAS

- ALKAR, A. Z., BUHUR, U. An internet based wireless home automation system for multifunctional devices. *IEEE Transactions on Consumer Electronics*, vol. 51, n. 4, p. 1169 - 1174, Nov. 2005. (doi.org/10.1109/TCE.2005.1561840)
- ANUSHA, K; MAHADEVASWAMY, U B. Automatic IoT based plant monitoring and watering system using Raspberry Pi. *International Journal of Engineering and Manufacturing*, vol. 8, n. 6, p. 55-57, 2018. (doi:10.5815/ijem.2018.06.05).
- BERMÚDEZ-ORTEGA, J. et al. Remote web-based control laboratory for mobile devices based on EJS, Raspberry Pi and Node.js. *IFAC-Papers On Line*, vol. 48, n.29, p. 158-163, 2015. (doi.org/10.1016/j.ifacol.2015.11.230)
- BOGART Jr., T. F. *Dispositivos e Circuitos Eletrônicos*, 3. Ed. São Paulo: Editora Makron Books, 2001. 463 p.
- Datasheet Raspberry - Datasheet do Módulo de Computação. Disponível em: <https://www.raspberrypi.org/documentation/hardware/computemodule/datasheet.md>. Acesso em: 10 de março de 2021.
- EVANS, D. *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*. Cisco Internet Business Solutions Group, 2011.
- GUARNIERI, M. The Roots of Automation Before Mechatronics. *IEEE Industrial Electronics Magazine*, vol. 4, n. 2, p. 42-43, junho de 2010. (doi:10.1109/MIE.2010.936772)
- KUMAR, R. & RAJASEKARAN, M. P. An iot based patient monitoring system using Raspberry Pi. In: *International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE)*. IEE, p.1-4, 2016.
- MALVINO, A.; BATES, D. J. *Eletrônica*. Trad. Romeu Abdo. vol.1. 7 ed. Porto Alegre: AMGH, 2011.
- RESENDE, W. J. F. et. Al. Desenvolvimento de um sistema para monitoramento e controle patrimonial, utilizando Rfid e dispositivos Iot. *Brazilian Journal of Development*, vol. 7, n. 7, p. 67357-67368, jul. 2021. (doi:10.34117/bjdv7n7-151)
- SANDEEP, V. et al. Global Accessible Machine Automation using Raspberry Pi. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, p. 1144-1147, 2015. (doi:10.1109/ICACCI.2015. 7275764)
- SONALI S. & LAGU & SANJAY B. D. Raspberry Pi for Automation of Water Treatment Plant. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, p.1999-2003, 2014.