



CENTRO UNIVERSITÁRIO FUNVIC



Luiz Rodrigo de Souza

**SISTEMA EMBARCADO APLICADO À AUTOMAÇÃO
RESIDENCIAL UTILIZANDO INTERNET DAS COISAS**

Pindamonhangaba-SP

2022



CENTRO UNIVERSITÁRIO FUNVIC



Luiz Rodrigo de Souza

SISTEMA EMBARCADO APLICADO À AUTOMAÇÃO RESIDENCIAL UTILIZANDO INTERNET DAS COISAS

Artigo apresentado como parte dos requisitos para obtenção do Diploma de Bacharelado em Engenharia de Computação pelo Curso de Engenharia de Computação do Centro Universitário FUNVIC.

Orientador: Prof. Rodrigo Ramos de Oliveira

Pindamonhangaba-SP

2022

SOUZA, Luiz Rodrigo

Sistema embarcado aplicado á automação residencial utilizando internet das coisas
/ Luiz Rodrigo de Souza / Pindamonhangaba-SP : UniFUNVIC Centro
Universitário FUNVIC, 2022
27f. : il

Monografia (Graduação em Engenharia da Computação) UniFUNVIC- SP.

Orientador: Prof. Rodrigo Ramos de Oliveira

1 IOT. 2 Automação. 3 Arduino. 4 Comunicação Ethernet

I Sistema embarcado IoT na automatização de itens residenciais II Luiz Rodrigo de
Souza



CENTRO UNIVERSITÁRIO FUNVIC



LUIZ RODRIGO DE SOUZA

SISTEMA EMBARCADO APLICADO À AUTOMAÇÃO RESIDENCIAL UTILIZANDO INTERNET DAS COISAS

Artigo apresentado como parte dos requisitos para obtenção do Diploma de Bacharelado em Engenharia de Computação pelo Curso de Engenharia de Computação do Centro Universitário FUNVIC.

Data: 24 /11 / 2022

Resultado: _____

BANCA EXAMINADORA

Prof. Dailton de Freitas

Centro Universitário FUNVIC

Assinatura: _____

Prof. Celio Augusto Machado

Centro Universitário FUNVIC

Assinatura: _____

Prof. Luis Fernando de Almeida

Centro Universitário FUNVIC

Assinatura: _____

AGRADECIMENTOS

Agradeço a Deus por ter me permitido alcançar meu objetivo em todos esses anos de estudo.

Aos meus pais, pelo incentivo e apoio em toda essa trajetória. ao Prof. Rodrigo Ramos de Oliveira, orientador deste trabalho, pelo suporte em todos os momentos em que precisamos. E a todos os docentes que estiveram presentes nestes cinco anos de estudos me ensinando, dando o melhor de si, mesmo com os desafios que surgiram ao longo deste caminho.

Autorizo cópia total ou parcial desta obra, apenas para fins de estudo e pesquisa, sendo expressamente vedado qualquer tipo de reprodução para fins comerciais sem prévia autorização específica do autor. Autorizo também a divulgação do arquivo no formato PDF no banco de monografias da Biblioteca institucional.

Luiz Rodrigo de Souza

Pindamonhangaba, 24 / 11 / 2022

SISTEMA EMBARCADO APLICADO À AUTOMAÇÃO RESIDENCIAL UTILIZANDO INTERNET DAS COISAS

EMBEDDED SYSTEM APPLIED TO HOME AUTOMATION USING INTERNET OF THINGS

Luiz Rodrigo de Souza¹, Rodrigo Ramos de Oliveira²

¹Engenheiro de Computação graduado pelo Centro Universitário FUNVIC, Pindamonhangaba-SP

²Docente do Centro Universitário FUNVIC, Pindamonhangaba-SP.

*luzrs.lrs@outlook.com

Resumo

A automação, em geral, está sendo cada vez mais viabilizada, em razão do barateamento e da evolução da tecnologia dos microcontroladores e sensoriamento. A automação residencial, conhecida como domótica, objeto de estudo deste trabalho, surgiu em decorrência desses fatores e permite monitorar ou acionar os diversos processos em uma residência, com acionamento remoto. Nesse contexto, este trabalho apresenta a implementação de um protótipo de *hardware* utilizando as placas de prototipagem Arduino Uno e Ethernet *shield* e *software* de ambiente *web* para automatizar uma residência. Para o protótipo, utilizou-se a plataforma eletrônica Arduino IDE, para controlar itens residenciais como portas e a iluminação do local. O controle e o acionamento foram providos por meio de uma página *Web*, permitindo acesso por meio de *smatphones*, computadores, *Tablets*, entre outros dispositivos. Os resultados obtidos com o protótipo mostram que a internet das coisas pode ser aplicada com sucesso ao contexto residencial, permitindo implementar tecnologias de automação funcionais e acessíveis.

Palavras-chave: Automação residencial. Plataforma Arduino. Domótica. Internet das coisas.

Abstract

Automation, in general, is being increasingly feasible, due to the cheapness and evolution of microcontroller and sensing technology. Home automation, known as home automation, the object of study of this work, arose as a result of these factors and allows monitoring or triggering the various processes in a residence, with remote drive. . In this context, this work presents the implementation of a hardware prototype using the Arduino Uno and Ethernet shield prototyping boards and web environment software to automate a residence. For the prototype, the Arduino IDE electronic platform was used to control residential items such as doors and site lighting. The control and activation were provided through a web page, allowing access through *smatphones*, computers, *Tablets*, among other devices. The results obtained with the prototype show that the Internet of Things can be successfully applied to the residential context, allowing to implement functional and accessible automation technologies.

Keywords: Home automation. Arduino Platform. Automation. Internet of Things.

INTRODUÇÃO

A expressão “internet das coisas” ou *Internet of Things* (IoT), embora tenha se tornado frequente a partir de 2009, foi proposta pela primeira vez por Kevin Ashton (2009), do Instituto de Tecnologia de Massachusetts (MIT), em 1999.

Com o avanço tecnológico em sensoriamento, acionamento e controle, a automação residencial tornou-se acessível, possibilitando tornar as residências mais autônomas, seguras e

funcionais. Há várias aplicações de segurança na automação residencial para controle de acesso à residência por meio de dispositivos, sensores de abertura de janelas, portas e portões, câmeras de segurança e outros níveis de acesso que informam diretamente ao morador (e à polícia) possíveis invasões. Em um cenário digitalizado e tecnológico, residências automatizadas vêm atraindo cada vez mais adeptos. (BOLZANI, 2004b).

Para Zambarda (2014), a revolução tecnológica representada pela Internet das Coisas é irreversível. O autor acrescenta que todas as coisas utilizadas cotidianamente, como geladeiras, carros ou maçanetas, estão ou estarão, em um futuro muito próximo, conectadas à rede mundial de computadores, integrando o mundo físico e o digital (ZAMBARDA, 2014).

Com a evolução da *IoT* ocorreu um grande impacto no dia a dia das pessoas, visto que diversos equipamentos como televisores, geladeiras, máquina de lavar roupa, sistema de som, podem ser conectados à internet. O objetivo é que os equipamentos se tornem eficientes como a geladeira, que emite avisos de produtos que estão acabando e já realiza cotação de preços informando qual o melhor lugar para realizar a compra. Acrescenta-se também o conforto de poder controlar a residência por meio de um *smartphone*, acendendo luzes, verificando se as fechaduras estão fechadas, entre outras aplicações (GODOI; ARAUJO, 2019).

Internet das coisas ou *Internet of Things* (IoT) faz referência à “rede de objetos físicos incorporados a sensores, *software* e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela internet. Esses dispositivos variam de objetos domésticos comuns a ferramentas industriais sofisticadas” (ORACLE, 2022).

Utilizando sistemas eletrônicos baseados em programação de microprocessadores para automação residência (domótica) e assim facilitar tarefas diárias, este projeto consiste no desenvolvimento de um sistema supervisor *web* baixo custo, capaz de operar diversos equipamentos existentes no protótipo de uma residência.

Internet das Coisas IoT

A constante evolução das tecnologias na área de Internet das Coisas impulsiona projetos e soluções direcionados para a automação residencial, visando não somente o conforto e a praticidade para o cotidiano das pessoas, como também proporcionando a economia de recursos como a água e a energia elétrica. No mercado, as tecnologias existentes para a criação de novos projetos em IoT oferecem padrões bem definidos.

Se faz necessário saber que a IoT é um *framework*, ou seja, é a união de diversos *softwares* para formar uma funcionalidade em comum, logo, não é um dispositivo nem uma tecnologia específica, logo não deriva de tecnologias existentes e sim as utiliza para existir. proporcionando que a utilização não seja restringida, porém torna necessário a criação de novos métodos de comunicação, uma vez que cada dispositivo terá um padrão diferente e precisarão coexistir (FACCIONI FILHO, 2016).

A proposta de criação de uma rede de diversos tipos de dispositivos feita por Costa (2018), capazes de comunicar com o mundo externo, permitem o acesso por qualquer tipo de utilizador. O autor espera padronizar os protocolos inerentes a IoT, integrar as diversas soluções de dispositivos, a simplificação da rede e com hardware de custo reduzido, de modo a ser incluída em qualquer ambiente doméstico.

Silva (2019) descreve o trabalho de *hardware* e *software* projetado para gerenciar a iluminação de um ambiente tendo o objetivo de integrar ao sistema de controle a função acender/apagar as lâmpadas, mantendo o controle manual dos interruptores, compartilhando funções para controlar a iluminação de um ambiente residencial real.

No projeto de Silva (2019) foi utilizado o NodeMCU, MQTT e o Home Assistant, executando testes funcionais para simular um ambiente de infraestrutura de má qualidade e com muitas falhas visando a validação do protótipo e da proposta. Os resultados relatados sob condições de falhas se mostraram estáveis e confiáveis para construir automações robustas.

A rede de comunicação entre os sensores e atuadores em aplicações IoT utiliza o conceito de Internet, existem dois elementos fundamentais na arquitetura da comunicação via Internet: servidor e Cliente. De acordo com o material estudado, ambos conceitos podem ser aplicados em dispositivos IoT, sendo que sensores podem ser representados como clientes e dispositivos podem ser associados aos servidores, mesmo com o *hardware* limitado.

O conceito de cliente e servidor abre espaço para o detalhamento de diversos protocolos e padrões importantes para a implementação de IoT, como TCP/IP e Wi-Fi.

No modelo TCP/IP abordado no projeto, a camada de Internet é responsável pelo roteamento de pacotes de dados e a camada de acesso à rede determina o modo que os dados são trafegados.

Placa de Prototipagem Rápida: Arduino

O Arduino consiste, por definição, em um conjunto de microcontrolador de placa única e um *software* de linguagem *c/c++* que dispõe de várias bibliotecas específicas para diversas aplicações para programá-lo. Trata-se de uma plataforma aberta de prototipagem baseada em *hardware* e *software* flexíveis e de utilização simples. O mesmo pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados do Arduino e atuar sobre eles. (McROBERTS,2011).

O Arduino é comumente chamado de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*. Há uma grande comunidade de pessoas utilizando Arduino, compartilhando seus códigos e diagramas de circuito para que outros os copiem e modifiquem. A maioria dessa comunidade também está muito disposta a auxiliar outros desenvolvedores (McROBERTS, 2011, p. 22).

A plataforma permite conectar botões, interruptores, sensores de temperatura, sensores de pressão, sensores de distância, módulos Ethernet e qualquer dispositivo que receba dados ou possua a função de ser controlado. Por se tratar de um sistema multiplataforma professores e alunos utilizam o Arduino para construir instrumentos científicos de baixo custo, ou para começar a trabalhar com programação e robótica. (ARDUINO, 2021).

Dentre os vários modelos de Arduino, destacam-se o Uno e o Mega. O Arduino Uno é composto por um microcontrolador ATmega328P, 14 pinos de entrada e saída digital (sendo que 6 fornecem saída PWM) e 6 entradas analógicas. A Figura 1, ilustra uma placa do Arduino Uno.

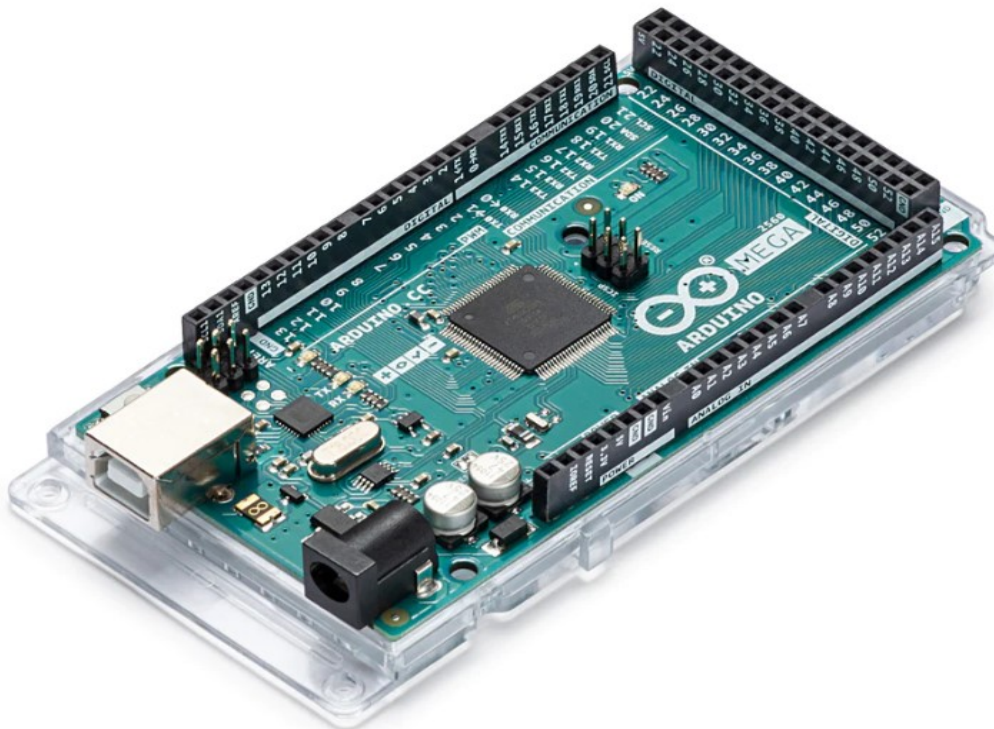
Já a versão Arduino Mega possui um microcontrolador ATmega2560, contando com 54 pinos de entrada e saída (sendo que 15 desses fornecem saída PWM) e 16 pinos de entrada analógica, além da memória interna de 256 kB, muito superior ao tamanho da memória do Arduino Uno, que é de 32 kB, o que possibilita ao Arduino Mega rodar maiores programas. A Figura 2 ilustra uma placa do Arduino Mega.

Figura 1- Arduino Uno



Fonte: Arduino (2021)

Figura 2- Arduino Mega



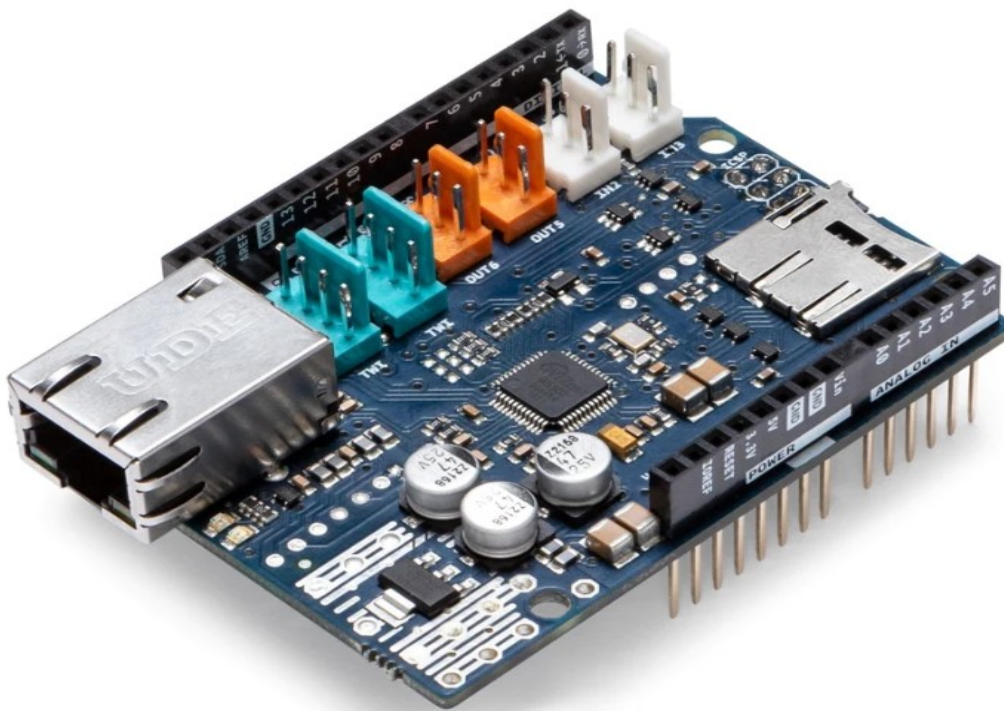
Fonte: Arduino (2021)

Shields para Arduino

Além das entradas e saídas do Arduino, é possível encontrar no mercado placas auxiliares chamadas de *Shields*, que possibilitam ao Arduino novas interações.

O *Arduino Ethernet Shield* permite que uma placa Arduino se conecte à uma rede ou à Internet. Segundo Evans et al. (2013), este *Shield* oficial *Arduino Ethernet* é baseado no WIZnet W5100, possui um conector RJ45 integrado para uma conexão Ethernet e um leitor de cartão micro SD integrado. O *Arduino Ethernet* não possui um chip controlador USB para serial integrado, mas possui um conector de seis pinos que pode ser conectado a um cabo FTDI ou uma porta serial USB para fornecer uma comunicação para que a placa possa ser programada. É ideal para ser utilizado em monitoramento remoto de estações de registros de dados com leitor de cartão micro SD integrados e uma conexão com uma rede Ethernet com fio. A Figura 3, ilustra uma placa do Ethernet Shield.

Figura 3- Ethernet Shield



Fonte: Arduino (2021)

Programação Web

A linguagem de programação HTML (*HyperText Markup Language*), foi criada por Tim Berners-Lee no início de 1990 com o objetivo de padronizar os documentos que seriam distribuídos na rede. Ele desenvolveu protocolos de comunicação para que o tráfego de informações em hipertexto fosse viável. Todo o conteúdo programado em HTML é compilado e exibido em qualquer navegador de internet, seja ele de um computador, tablet ou smartphone (BERNERS-LEE, 1999).

A sigla HTML significa HyperText Markup Language, em português compreendida como: linguagem de marcação de hipertexto. A primeira versão do HTML deriva da linguagem SGML (Standard Generalized Markup Language). O SGML era utilizado para a estruturação de documentos, o HTML herdou diversas tags tais como título <h1> ao <h6>, cabeçalho <head> e parágrafo <p>. A maior diferença entre essas duas linguagens de marcação é que o HTML implementava a tag com o atributo href, permitindo assim a ligação (links) de uma página a outra. Esse conceito de interligação entre documentos é a base de todo o funcionamento da Web (EIS, 2011).

O HTML5 trouxe uma série de novas funcionalidades e tags que permitiam a criação de websites muito mais dinâmica, atrativa e harmoniosa. Dentre as mudanças significativas podemos destacar a separação total entre HTML, o CSS e o JavaScript, o que tornou não só a criação mais simples e dinâmica, mas também garantiu uma enorme facilidade na manutenção e correção de extensos códigos e HTML. Em comparação com o HTML, o 5 também é case insensitive embora ainda seja aconselhado a manter a separação entre maiúsculas e minúsculas pela grande quantidade de pessoas que ainda utilizam o HTML. Surgiram na versão 5 novas tags que ajudam a manter a organização do código, o tornando muito mais fácil até para pessoas que nunca tiveram um contato com esse tipo de linguagem (ANDREWS, 2015, p.?).

Redes e seu funcionamento

De acordo com Tanenbaum (2003), as redes de computadores podem ser classificadas pelo seu tamanho, topologia, meio físico e protocolo utilizado. Os protocolos de comunicação são a linguagem que permite que os diversos elementos de um sistema se comuniquem e se entendam. O protocolo TCP/IP foi desenvolvido para que dois ou mais dispositivos se comuniquem para enviar e

receber informações de outros dispositivos pela rede. A conexão *wi-fi* segue os mesmos princípios da comunicação com fio, porém utiliza como meio físico sinais de rádio.

Os principais tipos de redes levando-se em conta a classificação por seu tamanho têm as seguintes denominações:

- **Redes LAN (*Local Area Network*):** são redes de área local, normalmente utilizadas em um único edifício ou campus universitário com no máximo alguns quilômetros de extensão. São muito usadas para conectar estações de trabalho de empresas e indústrias com o objetivo de compartilhar recursos e trocar informações, tais como compartilhamento de arquivos e impressoras tanto por meio físico (com fios) como também pela conexão *wifi* que segue os mesmos princípios da comunicação com fio, porém utiliza como meio físico sinais de rádio (TANEMBAUM, 2003);
- **Redes MAN (*Metropolitan Area Network*):** são redes que tem abrangência de uma cidade inteira. Elas interligam várias LANs através de nós para que possam se comunicar como se estivessem na mesma LAN. Exemplos de MANs são as interconexões de várias redes LAN de uma empresa em diferentes pontos, redes de TV a cabo e Internet via rádio (CARMONA, 2007);
- **Redes WAN (*Wide Area Network*):** são redes que atingem longa distância, que integram diversos países e continentes. Uma WAN reúne todas as menores redes. A Internet é uma rede WAN por exemplo.

Programação do Arduino

Uma definição de linguagem de programação é uma sequência de códigos padronizados utilizados para definir um *software* de computador, este especifica quais dados um computador irá atuar e como eles serão armazenados ou transmitidos.

Basicamente, uma linguagem de programação é composta pelos códigos propriamente ditos e necessita de um compilador para que o programa seja verificado e executado no computador. O compilador nada mais é do que o responsável por converter todo o código digitado por nós humanos e convertê-lo para a linguagem de máquina.

A linguagem de programação do Arduino é a linguagem C/C++, o que torna sua programação mais acessível, possibilitando o controle de entradas e saídas digitais/analógicas para aplicação em diversos projetos.

MÉTODO

A elaboração deste trabalho fez uso da pesquisa bibliográfica e da pesquisa exploratória, aliadas à técnica de prototipação.

A implementação do sistema de domótica para controle de objetos residenciais foi realizada por meio do Arduino UNO e o *Shield Ethernet*, que possibilita ao Arduino se conectar a uma rede *Ethernet*, o que viabiliza o controle efetivo do sistema através de qualquer navegador de *internet* em diversos dispositivos.

Para o protótipo de uma automação residencial foram utilizados:

- Arduino Uno;
- Ethernet Shield;
- Módulo relé serial
- LEDs;
- Celular;
- Notebook

Em relação à implementação da aplicação web, que permitirá o controle dos itens de uma residência, foram empregados:

- Linguagem de Programação: o aplicativo web, que é o intermediário entre o usuário e a efetiva interação com o sistema automatizado, foi desenvolvido por meio das linguagens: *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript, utilizando o software VScode. HTML permitiu desenvolver o conteúdo a ser exibido na página. Para definir a aparência da aplicação e a formatação do conteúdo, foi utilizada a linguagem CSS e para a leitura e o envio de dados entre a aplicação web e o protótipo Arduino foi utilizada a linguagem JavaScript. No Arduino, o software responsável pelo gerenciamento do sistema foi desenvolvido em linguagem C/C++ na qual o Arduino se baseia, para determinar as funcionalidades a serem executadas.
- Comunicação: A comunicação entre o hardware e a aplicação web foi realizada com rede ethernet por meio do módulo *ethernet shield*, com protocolo TCP/IP.
- O código HTML está armazenado no cartão de memória devido ao tamanho do código e será acessado pelo *Arduino* no pino 4.

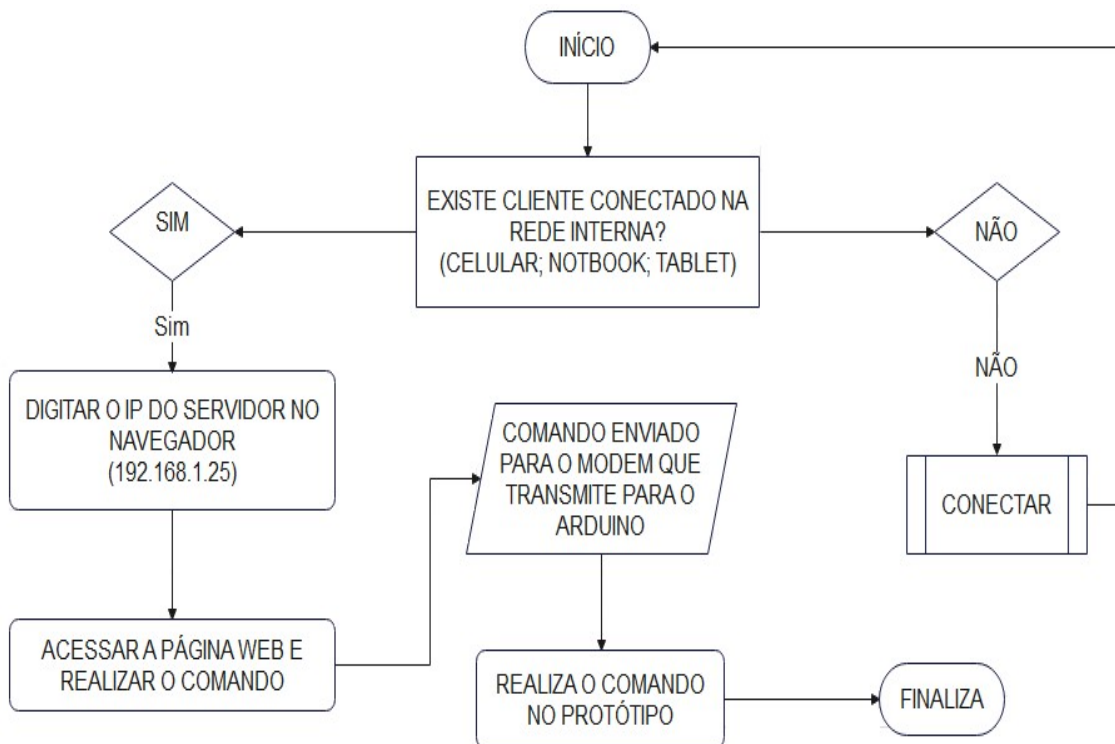
A implementação do código fez uso das seguintes bibliotecas:

- A *Serial Peripheral Interface* (SPI.h), responsável pela comunicação *master slave*, em que determina quem manda e quem obedece.

- A Ethernet_W5500.h, encarregada pela comunicação com a internet.
- A SerialRelay.h, responsável por controlar os relés serial.
- A SD.h responsável por acessar o cartão de memória.

Para ambos, protótipo e aplicação web, foram aplicados os conceitos da Internet das Coisas. A seguir na figura 4 está disposto o diagrama do funcionamento do sistema.

Figura 4 – Diagrama do Sistema



Implementação

A implementação do projeto foi realizada em etapas, para que a cada etapa concluída fossem realizados os testes:

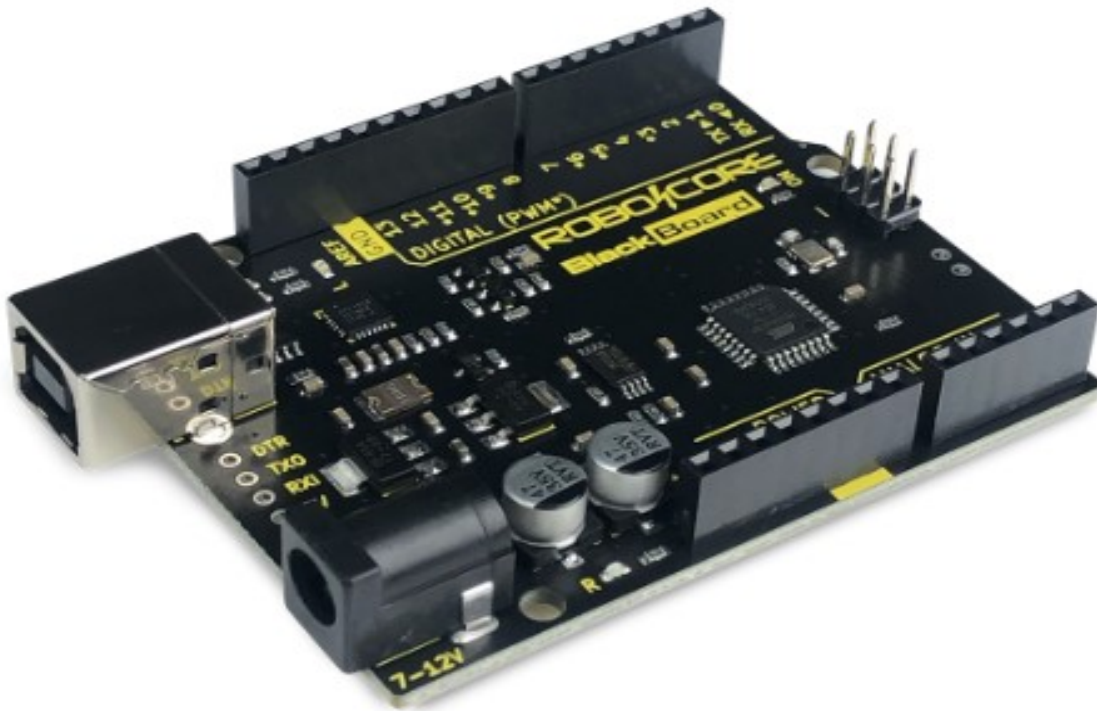
- Compra do Itens
- Inicio da Programação
- Montagem do *hardware*

- Teste do programa com *hardware*
- Teste de acionamento através da página *web* e do *hardware*

Arduino Uno

A placa utilizada no projeto foi a Arduino BlackBoard Uno R3 da Robo Core, que utiliza o micro controlador ATmega238 com um conector USB-serial para a comunicação e gravação do programa, possui 22 pinos de entrada e saída (I/O). Destes, 8 podem ser utilizados como entradas analógicas (A0 a A7), 6 podendo ser utilizados com PWM (D3, D5, D9, D10 e D11). A Figura 5 ilustra a placa Arduino utilizada no projeto.

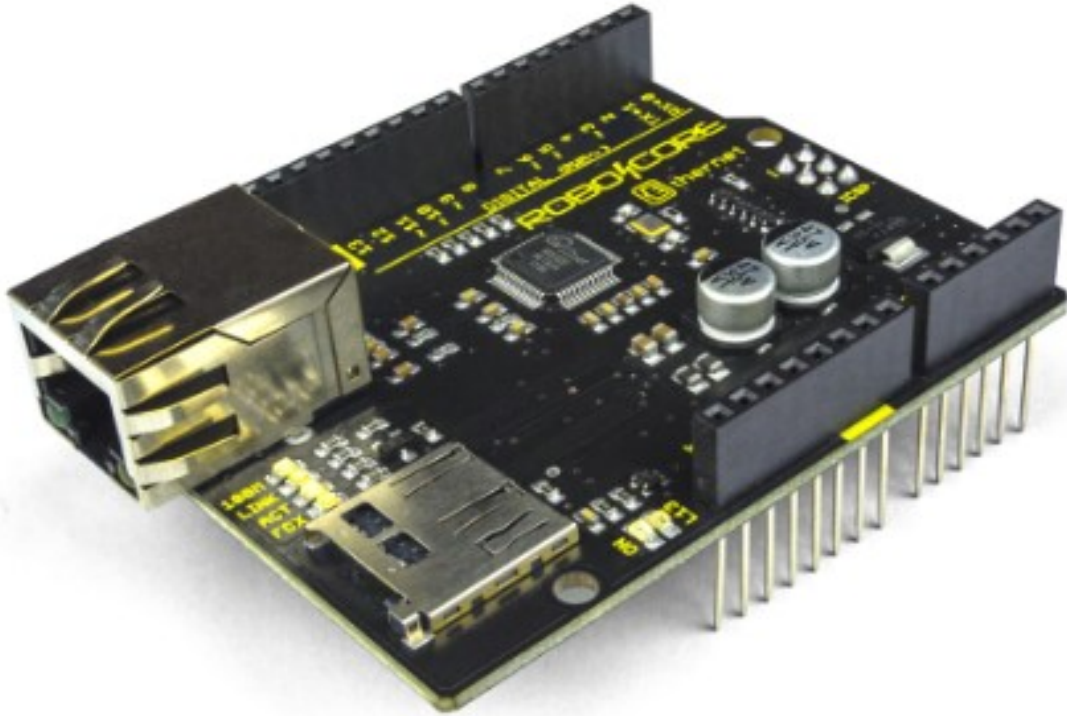
Figura 5 - Arduino BlackBoard Uno R3



Ethernet Shield

No projeto, a comunicação com a rede de internet utiliza o Shield Ethernet da Robo Core com protocolo de comunicação TCP/IP sendo a interface ethernet W5500 da Wiznet, que suporta até 8 comunicações simultâneas e possui um *buffer* de 32 kB, fornecendo um IP com capacidade TCP e UDP. Também possui um *mac address* único habilitado junto a IEEE, garantindo que não haverão conflitos na rede. A Figura 6 ilustra a placa do *Ethernet Shield* utilizada no projeto.

Figura 6 – Ethernet Shield



Web Sever

Um servidor é uma máquina capaz de executar vários serviços simultaneamente e está sempre ligada. Um servidor *web* é responsável pelo gerenciamento, armazenamento e distribuição de páginas de um *website* (GRUPPEN, 2014).

Com a união dos módulos Arduino UNO e da Ethernet Shield tem-se um *web service* permitindo o acesso através da internet por uma página *web* que está armazenada no cartão de memória na *Ethernet Shield*.

Após acessado o *Web Server*, para ligar o equipamento desejado é necessário selecionar a saída equivalente ao equipamento para enviar o comando. Desta forma o Arduino receberá o comando enviado e executará a ação ligando ou desligando o equipamento.

Módulos Relés

Módulos relés são dispositivos que permitem o acionamento de cargas maiores de até 10A, permitindo acionar até grandes cargas como motores de grande porte sendo utilizado como uma interface de acionamento.

Utilizando 2 módulos diferentes para o acionamento das cargas, pode-se observar o quão dinâmico um sistema de IoT com Arduino pode ser. Um dos módulos é o módulo relé serial, que permite o acionamento de até quarenta cargas simultâneas. A Figura 9, ilustra a placa de relés seriais utilizada no projeto.

Figura 7 - Módulo Relé Serial

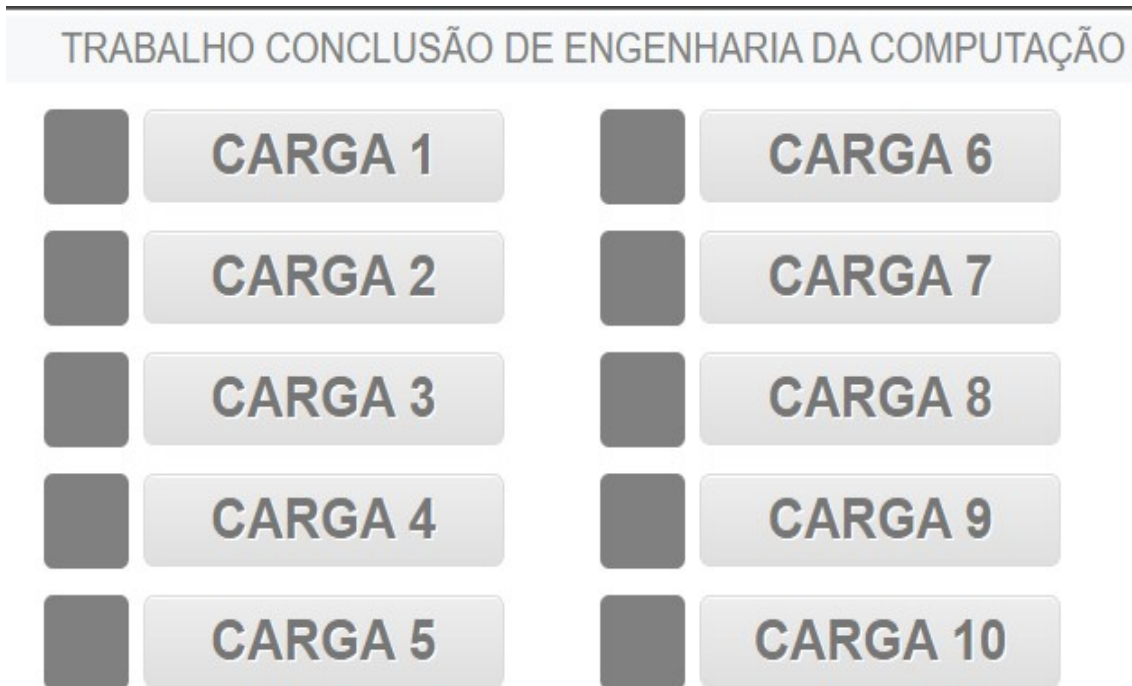


RESULTADO

Para o acesso e o controle da automação foi desenvolvida uma página *web* utilizando a linguagem HTML e o CSS para criar uma interface intuitiva e customizável, podendo ser acessada em qualquer navegador *web*, seja em notebook, Android, IOS. Na interface são apresentados os botões para definir o estado dos atuadores, ligado ou desligado.

A Figura 9 apresenta a interface *online* da aplicação *web*, em que o usuário tem acesso aos acionamentos dos itens presentes na automação em tempo real.

Figura 8 - Interface Web



O *hardware* montado para o acionamento das cargas utilizou duas placas de relé serial em conjunto com o módulo *web server*, sendo que cada relé é acionado por um comando distinto, referente a cada botão disposto na interface, vinda da página web pela rede ethernet usando o protocolo TCP/IP. A figura 9 ilustra o módulo completo montado, para a realização dos testes.

O microcontrolador responsável pela execução das funcionalidades necessárias para o acionamento das cargas espera um comando vindo da página web, indicando ao Arduino qual carga ele deve acionar, de acordo com o botão pressionado na página web, podendo ser de 1 a 10. A comunicação entre a *web server* e a página web é realizada por uma codificação no Arduino, em que é determinado o IP do *web server* para que ele possa ser encontrado na rede. A Figura 10 mostra um trecho do código desenvolvido para o sistema que permite a conexão à rede de internet do microcontrolador *online*.

Figura 9: Modelo inicial do sistema físico utilizado para os testes

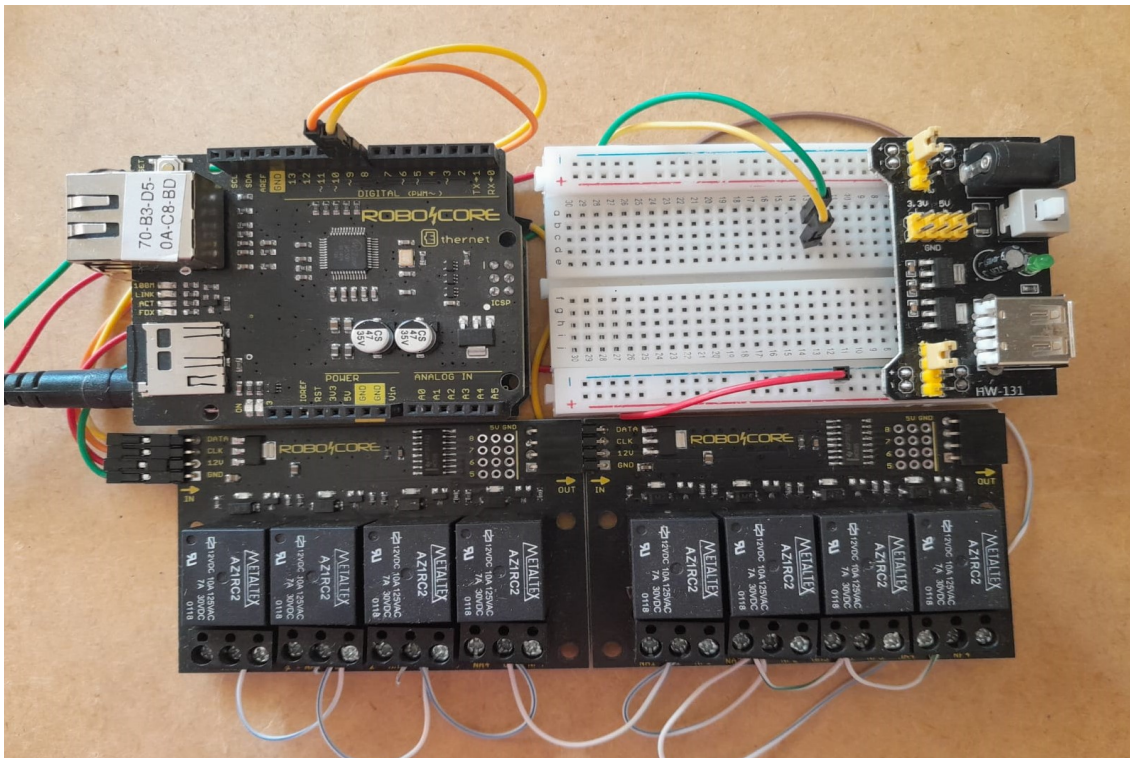


Figura 10 - Código referente a conexão à rede de internet do microcontrolador online

```

void loop()
{
  EthernetClient client = server.available(); //verifica se existe alguém querendo se conectar

  if (client) { // existe cliente?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) { // Existe informacao vinda do cliente
        char c = client.read(); // Le cada byte enviado pelo cliente, ou seja, cada caracter.
        // Por padrao, o ultimo caracter enviado pelo cliente (nosso navegador) é em branco e termina com \n
        // Dessa forma conseguimos saber se o cliente acabou de enviar informacoes para o servidor (Arduino)
        if (req_index < (REQ_BUF_SZ - 1)) {
          HTTP_req[req_index] = c; // salva os caracteres das solicitacoes do browser
          req_index++;
        }
        if (c == '\n' && currentLineIsBlank) {
          // envia para o cliente o protocolo padrao de sucesso HTTP
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");
          client.println();
        }
      }
    }
  }
}

```

Discussão

Por meio da utilização de técnicas semelhantes a observadas em trabalhos anteriores, como o sistema de irrigação doméstico, baseado em internet das coisas, do autor Grehs (2016) e o trabalho referente ao desenvolvimento de um sistema de automação residencial baseado em domótica, dos autores Silva e Oliveira (2019), foi possível a implementação da domótica em um protótipo, possibilitando um avanço na forma como interagimos com o ambiente em nossa residência, podendo ser customizado de acordo com a nossa necessidade. O Arduino Uno mostrou ser uma ferramenta confiável para a gestão deste sistema, sendo sua integração com a página web bem sucedida.

A restrição encontrada neste sistema foi o limite de relés que podem ser acionados de uma só vez que se limita a 40 relés serial. Durante o projeto, a alimentação utilizada para a placa do Arduino Uno, uma fonte de 12Vcc, não fornecia corrente o suficiente para alimentar a web serve, as placas relé serial e os leds, causando uma falha na placa Ethernet Shield, fazendo a comunicação entra em falha, sendo necessário reiniciar todo o sistema. Para este problema foi necessário prover ao sistema de iluminação uma fonte externa de alimentação com tensão de funcionamento de 5Vcc. Futuramente, este trabalho pode ser expandido, adicionando interfaces de acionamentos e comunicação, como por exemplo no controle autônomo do sistema de irrigação de plantas, trabalhando em conjunto com o sensor de umidade do solo. Existem várias possibilidades de expansão deste sistema, podendo se adequar a diversas necessidades do usuário.

Conclusão

Neste trabalho realizou-se um estudo da automação residencial interativa, demonstrando a facilidade de implementação e uso. Não houve um custo elevado para adquirir os componentes utilizados, aumentando a possibilidade de inclusão das classes de menor poder aquisitivo a estes sistemas. Foi construído um protótipo funcional, que recebe os dados vindo da página *web* para executar as ações sobre os componentes de acionamento retornando para o usuário em tempo real o status de cada saída.

Referências

ARDUINO. Disponível em:< <https://www.arduino.cc> >Acesso em: out. 2021.

ANDREWS. Por que utilizar o HTML 5?. 2015. Disponível em: <<http://fabrica.ms.senac.br/2015/05/por-que-utilizar-o-html-5/>>. Acesso em: 27 nov. 2022.

BERNERS-LEE, T. Weaving the Web: **the original design and ultimate destiny of the World Wide Web by its inventor**. 1.ed. San Francisco, CA: Harper San Francisco, 1999.

BOLZANI, C. A. M. **Residências inteligentes**. São Paulo: Livraria da Física. 2004b

BOLZANI, C. A. M. **Desenvolvimento de um Simulador de Controle de Dispositivos Residenciais Inteligentes**: Uma Introdução aos Sistemas Domóticos. 2004 115f. Dissertação (Mestrado) – Escola Politécnica da Universidade de São. Paulo. Departamento de Engenharia de Sistemas Eletrônicos, 2004.

BOLZANI, C. A. M. **Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes**. 2010 155f. Tese (Doutorado) – Escola Politécnica da Universidade de São. Paulo. Departamento de Engenharia de Sistemas Eletrônicos, 2010.

COSTA, Rafael Almeida. **Casa inteligente com recurso a tecnologias open source**. 2018. 104 f. Dissertação (Mestrado) - Curso de Sistemas e tecnologias de Informação para as organizações, Instituto Politécnico de Viseu, Viseu, 2018.

CAIRES, Luís Eduardo; BURANI, Geraldo Francisco; SUETA, Hélio Eiji. **Aplicação de sistemas inteligentes em instalações elétricas residenciais**. Eletricidade Moderna, São Paulo, v. 41, n. 468, p.140-147, mar. 2013. CARMONA, T. Guia Técnico de Redes de Computadores. São Paulo: Digerati Books, 2007.

EIS, Diego. O básico: O que é HTML?. 2011. Disponível em: <<http://fabrica.ms.senac.br/2015/0https://tableless.com.br/o-que-html-basico/5/por-que-utilizaro-html-5/>>. Acesso em: 27 nov. 2012.

FACCIONI FILHO, M. **Internet das coisas. Palhoça**: UnisulVirtual. 2016. 56 p.

GREHS, Daniel Henrique. **Sistema de irrigação doméstico baseado em Internet das Coisas**. 2016. 55 f. TCC (Graduação) - Curso de Engenharia de Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019. Disponível em: <https://www.lume.ufrgs.br/handle/10183/147673>. Acesso em: 22 dezembro. 2021

GODOI, M. G. de; ARAÚJO, L. S. **A INTERNET DAS COISAS**: evolução, impactos e benefícios. Revista Interface Tecnológica, [S. l.], v. 16, n. 1, p. 19–30, 2019. Disponível em:< <https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/538>>. Acesso em: 12 set. 2022.

MCROBERTS, M. **Arduino Básico**. São Paulo: Novatec, 2011.

OLIVEIRA, Sérgio de. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. São Paulo: Novatec, 2017.

ORACLE. O que é IoT? Disponível em: <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acesso em: 4 out. 2022.

SERIALLINK. Disponível em: Acesso em: <<http://www.seriallinke.com.br>>. nov. 2021.

TANEMBAUM, A. S. Redes de Computadores. 4. Ed. Rio de Janeiro: Elsevier,2003.

SILVA, Leonardo José Nascimento. **Utilizando o Home Assistant e o NodeMCU para um modelo genérico de automação moderna**. 2019. 47 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal de Sergipe, Centro de Ciências Exatas e Tecnologia, São Cristóvão, 2019.

SOUZA, Marcelo Varela de. **Domótica de baixo custo usando princípios de IoT**. 2016. 49 f. Dissertação (Mestrado) - Curso de Engenharia de Software, Instituto Metrópole Digital, Universidade Federal do Rio Grande do Norte, Natal, 2016.

ZAMBARDA, Pedro. '**Internet das Coisas**': entenda o conceito e o que muda com a tecnologia. 2014. Acesso em 12.01.2022 Disponível em <http://www.techtudo.com.br/noticias/noticia/2014/08/internet-das-coisas-entenda-o-conceitoe-o-que-ida-com-tecnologia.html>.

Anexo A - Declaração de Direito Autoral

DECLARAÇÃO DE DIREITOS AUTORAIS

Nós, abaixo assinados, transferimos todos os direitos autorais do artigo intitulado SISTEMA EMBARCADO APLICADO À AUTOMAÇÃO RESIDENCIAL UTILIZANDO INTERNET DAS COISAS à Revista Eletrônica de Ciências Exatas.

Declaramos ainda que o trabalho é original e que não está sendo considerado para publicação em outra revista, quer seja no formato impresso ou no eletrônico. Certificamos que participamos suficientemente da autoria do manuscrito para tornar pública nossa responsabilidade pelo conteúdo.

Assumimos total responsabilidade pelas citações e referências bibliográficas utilizadas no texto, bem como pelos aspectos éticos que envolvem os sujeitos do estudo.

Data: 24 / 11 / 2022

Aluno. Luiz Rodrigo de Souza

Centro Universitário FUNVIC

Assinatura: _____

Anexo B - Diretrizes para Autores

Os trabalhos devem ser redigidos em português, com uso obrigatório da norma culta. Durante o preenchimento cadastral, o nome completo de cada autor e respectiva afiliação institucional devem ser inseridos nos campos adequados e devem aparecer no arquivo. A Revista Eletrônica de Ciências Exatas sugere que o número máximo de autores por artigo seja 6 (seis). Artigos com número superior a 6 (seis) serão considerados exceções e avaliados pelo Conselho Editorial que poderá solicitar a adequação. **Pesquisas feitas com seres humanos e animais devem, obrigatoriamente, citar a aprovação da pesquisa pelo respectivo Comitê de Ética, citando o protocolo de aprovação.** O não atendimento implica em recusa da submissão. Da mesma forma, o plágio implicará na recusa do trabalho.

O uso da norma culta da Língua Portuguesa, a obediência às normas da Revista e a originalidade do artigo são de total responsabilidade dos autores. O não atendimento a esses critérios implicará na recusa imediata do trabalho.

APRESENTAÇÃO DO MATERIAL

O número máximo é de 20 páginas, incluindo referências, figuras, tabelas e quadros. Os textos devem ser redigidos em **Fonte Times New Roman, tamanho 12, espaçamento 1,5 cm, justificado.** Devem ser utilizadas margens de 2 cm em cada lado.

As Figuras (gráficos, imagens, desenhos, fluxogramas e esquemas) deverão apresentar boa nitidez, estar em formato JPEG, com resolução de 800dpi e com tamanho de 15cm x 10cm. As figuras deverão ser enumeradas consecutivamente em algarismos arábicos, encabeçadas pelas respectivas legendas; as fontes, logo abaixo das mesmas, com fonte de tamanho 11 e espaçamento simples.

Tabelas e Quadros deverão ser enumeradas consecutivamente com algarismos arábicos e encabeçados pelos respectivos títulos, com fonte de tamanho 11 e espaçamento simples.

As citações deverão aparecer ao longo do texto, conforme Normas da ABNT (NBR 10520, 2002), seguidas pelo ano de publicação, cujas chamadas podem ser pelo sobrenome do autor, pela instituição responsável ou pelo título. As citações podem ser incluídas na sentença: sobrenome (ano). Ex.: Gomes, Faria e Esper (2006) ou entre parênteses: (SOBRENOME, ano). Ex.: (GOMES; FARIA; ESPER, 2006). Quando se tratar de citação direta (transcrição literal), indicar, após o ano, a página de onde o texto foi extraído. O trecho transcrito deverá estar entre aspas quando ocupar até três linhas. As citações diretas com mais de três linhas devem ser destacadas com recuo de 4 cm da margem esquerda, ser escritas com tamanho 11, com espaçamento entre linhas simples e sem aspas. Citações indiretas de vários documentos simultâneos devem constar em ordem alfabética (como nas referências). Citação de citação: deve-se fazer a referência do autor lido. Ex.: Pádua (1996 apud FERNANDES, 2012, p. 5) salienta que “[...] pesquisa é toda atividade voltada para a solução de problemas [...]”.

Teses e dissertações, quando não houver o respectivo artigo científico publicado em periódico, devem ser dos **últimos três anos**; obrigatoriamente indicando o **link** que remeta ao cadastro nacional de teses da CAPES ou das universidades onde esses documentos foram publicados. Não serão aceitas Monografias de Especialização como referência.

Grafia de termos científicos, comerciais, unidades de medida e palavras estrangeiras devem ser grafados por extenso, em vez de seus correspondentes simbólicos abreviados, quando citados pela primeira vez. Deve-se utilizar o Sistema Internacional de Unidades.

Palavras estrangeiras devem ser evitadas, utilizar preferentemente a sua tradução. Na impossibilidade, os termos estrangeiros devem ser grafados em itálico.

ESTRUTURA DO ARTIGO

PESQUISAS ORIGINAIS devem ter no máximo 20 páginas com até 40 citações; organizar da seguinte forma:

Título em português: caixa alta, centralizado, negrito, conciso, com um máximo de 25 palavras.

Título em inglês (obrigatório): caixa alta, centralizado. Versão do título em português.

Autor(es): o(s) nome(s) completo(s) do(s) autor(es) e seus títulos e afiliações à Sociedade ou Instituições. Indicar com asterisco o autor de correspondência. Ao final das afiliações fornecer o e-mail do autor de correspondência.

Resumo: parágrafo único sem deslocamento, fonte tamanho 11, espaço simples, justificado, contendo entre 150 e 250 palavras. Deve conter a apresentação concisa de cada parte do trabalho, abordando objetivos, métodos, resultados, discussão e conclusões. Deve ser escrito sequencialmente, sem subdivisões. Não deve conter símbolos, equações, diagramas, fórmulas e contrações que não sejam de uso corrente.

Palavras-chave: de 3 a 5 palavras-chave, iniciadas por letra maiúscula, separadas e finalizadas por ponto.

Abstract: tradução literal do resumo, com formatação idêntica à do resumo.

Keywords: tradução literal das Palavras-chave em Português.

Introdução: deve apresentar o assunto a ser tratado, fornecer ao leitor os antecedentes que justificam o trabalho, incluir informações sobre a natureza e importância do problema, sua relação com outros estudos correlatos e suas limitações. Essa seção deve representar a essência do pensamento do pesquisador em relação ao assunto estudado e apresentar o que existe de mais significativa na literatura científica. Os objetivos da pesquisa devem figurar como o último parágrafo desse item.

Método: destina-se a expor os meios dos quais o autor se valeu para a execução do trabalho. Pode ser redigido em corpo único ou dividido em subseções. Especificar tipo e origem de produtos e equipamentos utilizados. Citar as fontes que serviram como referência para o método escolhido.

Pesquisas feitas com seres humanos e animais devem, obrigatoriamente, citar a aprovação da pesquisa pelo respectivo Comitê de Ética, citando o protocolo de aprovação.

Resultados: Nesta seção o autor irá expor o obtido em suas observações. Os resultados poderão ser apresentados em quadros, tabelas ou figuras, não podendo ser repetidos em mais de um tipo de ilustração.

Discussão: O autor, ao tempo que justifica os meios que usou para a obtenção dos resultados, deve confrontá-los com a literatura pertinente; estabelecer relações entre causas e efeitos; apontar as generalizações e os princípios básicos que tenham comprovações nas observações experimentais; esclarecer as exceções, modificações e contradições das hipóteses, teorias e princípios diretamente relacionados com o trabalho realizado; indicar as aplicações teóricas ou práticas dos resultados obtidos, bem como, suas limitações; indicar, quando necessário, uma teoria para explicar certas observações ou resultados obtidos; sugerir, quando for o caso, novas pesquisas a partir da experiência adquirida no desenvolvimento do trabalho e visando a sua complementação.

Conclusões: Deve expressar de forma lógica e objetiva o que foi demonstrado com a pesquisa.

Agradecimentos (opcionais): O autor pode agradecer às fontes de fomentos e àqueles que contribuíram efetivamente para a realização do trabalho. Agradecimento a suporte técnico deve ser feito em parágrafo separado.

Referências (e não bibliografia): Espaço simples entre linhas e duplo entre referencias. A lista completa de referências, no final do artigo, deve ser apresentada em ordem alfabética e de acordo com as normas da ABNT (NBR 6023, 2002). Quando a obra tiver até três autores, todos devem ser citados. Mais de três autores, indicar o primeiro, seguido de et al. Alguns exemplos:

APÊNDICE A – LINHAS DE CÓDIGO DO ARDUINO

```

#include <SPI.h>
#include <SD.h>
#include <Ethernet_W5500.h>
#include <SerialRelay.h>

//Define o Mac Address da placa de rede. Essa informação pode ser encontrada em uma
etiqueta colada embaixo da mesma.
byte mac[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

//Define o IP da placa. Caso necessário altere o mesmo para se adequar a sua rede.
IPAddress ip(00, 00, 00, 00);
EthernetServer server(80);
File webFile;

//na linha de baixo está sendo definido as portas em que o módulo relé serial está conectado.
SerialRelay relays(8, 9, 2); // (pino de data, pino de clock, quantidade de módulos)
#define REQ_BUF_SZ 60
char HTTP_req[REQ_BUF_SZ] = {0};
char req_index = 0;
int estado_botao[11];
char * pch;
void setup()
{
  //ESTADO INICIAL DOS BOTOES 0 -> desligado, 1 -> ligado:
  estado_botao[1] = 0;
  estado_botao[2] = 0;
  estado_botao[3] = 0;
  estado_botao[4] = 0;
  estado_botao[5] = 0;
  estado_botao[6] = 0;
  estado_botao[7] = 0;
  estado_botao[8] = 0;
  estado_botao[9] = 0;
  estado_botao[10] = 0;

```

```

Serial.begin(9600);
Serial.println("Inicializando cartao microSD...");
if (!SD.begin(4)) {
  Serial.println("ERRO - inicializacao do cartao falhou!");
}
Serial.println("SUCESSO - cartao microSD inicializado.");
if (!SD.exists("index.htm")) {
  Serial.println("ERRO - index.htm nao foi encontrado!");
  return;
}
Serial.println("SUCESSO - Encontrado arquivo index.htm.");
Ethernet.begin(mac, ip);
server.begin();
}
void loop()
{
  EthernetClient client = server.available(); //verifica se existe alguem querendo se conectar

  if (client) { // existe cliente?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) { // Existe informacao vinda do cliente
        char c = client.read(); // Le cada byte enviado pelo cliente, ou seja, cada caracter.
        // Por padrao, o ultimo caracter enviado pelo cliente (nosso navegador) e em branco e
termina com \n
        // Dessa forma conseguimos saber se o cliente acabou de enviar informacoes para o
servidor (Arduino)
        if (req_index > (REQ_BUF_SZ)) {
          HTTP_req[req_index] = c; // salva os caracteres das solicitacoes do browser
          req_index++;
        }
        if (c == '\n' && currentLineIsBlank) {
          // envia para o cliente o protocolo padrao de sucesso HTTP

```

```

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println();

//caso a request seja pela ação de um botão:
if (StrContains(HTTP_req, "ajax_botao")) {
  for (int i = 1 ; i <= 10 ; i++) {
    //Serial.println(HTTP_req);
    char botao[] = "botaoxx";
    if (i < 10) {
      botao[5] = '0';
      botao[6] = '0' + i;
    } else {
      botao[5] = '0' + (i / 10);
      botao[6] = '0' + (i % 10);
    }
    if (StrContains(HTTP_req, botao)) {
      SetBotao(i, client);
    }
  }
}
else {
  // grava no arquivo webFile a página que temos no microSD
  webFile = SD.open("index.htm");
  if (webFile) {
    while (webFile.available()) {
      client.write(webFile.read()); // envia para o cliente a página - nessa linha de fato o
  Arduino imprime no browser a página
    }
    webFile.close();
  }
}
}

```



```

Serial.println(HTTP_req); //para debug, verifica no monitor serial a requisição
req_index = 0; //reseta o index do buffer e a variável que armazena as requisições
StrClear(HTTP_req, REQ_BUF_SZ);
break;
}
// toda linha de texto recebida do cliente termina com \r\n
if (c == '\n') {
    //verifica se acabou a linha, já que \n é o ultimo caracter
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // o cliente ainda está enviando informações
    currentLineIsBlank = false;
}
}
}
}
delay(1); // dá um tempo para o browser receber os dados
client.stop(); // fecha a conexão
}
}
void SetBotao(int botao, EthernetClient client) {
    int modulo = (botao - 1) / 4 + 1;
    int rele = botao % 4;
    if (rele == 0) {
        rele = 4;
    }

    Serial.print("modulo:");
    Serial.print(modulo);
    Serial.print(" rele:");
    Serial.print(rele);
    Serial.print(" - ");

```

```
if (estado_botao[botao] == 0) {
    relays.SetRelay(rele, SERIAL_RELAY_ON, modulo);

    estado_botao[botao] = 1;
    Serial.println("1");
}
else {
    relays.SetRelay(rele, SERIAL_RELAY_OFF, modulo);
    estado_botao[botao] = 0;
    Serial.println("0");
}
client.print("0|");
for (int i = 1 ; i <= 40 ; i++) {
    client.print(estado_botao[i]);
    client.print("|");
}
}
```