



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

YURE THALLES COSTA DANTAS

MODELO DE HARDWARE PARA O SISTEMA DE MONITORAMENTO DE
LIXEIRAS – *GREENTRASH*

PAU DOS FERROS/RN

2023

YURE THALLES COSTA DANTAS

**MODELO DE HARDWARE PARA O SISTEMA DE MONITORAMENTO DE
LIXEIRAS – *GREENTRASH***

Monografia apresentada à Universidade Federal Rural do Semi-Árido (UFERSA), como requisito parcial para a obtenção do título de Bacharel em Tecnologia da Informação.

Orientador: Prof. Dr. Pedro Thiago Valério de Souza

PAU DOS FERROS/RN

2023

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tornar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

CC837 Costa, Yure Thalles Costa Dantas.
m Modelo de Hardware Para o Sistema de
Monitoramento de Lixeiras- Greentrash / Yure
Thalles Costa Dantas Costa. - 2023.
34 f. : il.

Orientador: Pedro Thiago Valério de Souza
Souza.

Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Tecnologia da
Informação, 2023.

1. ESP8266. 2. HC-SR04. 3. Greentrash. 4.
Sistema. 5. Protótipo. I. Souza, Pedro Thiago
Valério de Souza, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).

Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

YURE THALLES COSTA DANTAS

**MODELO DE HARDWARE PARA O SISTEMA DE MONITORAMENTO DE
LIXEIRAS – *GREENTRASH***

Monografia apresentada à Universidade Federal Rural do Semi-Árido (UFERSA), como requisito parcial para obtenção do título de Bacharel em Tecnologia da Informação.

Defendida em: 13/10/2023.

BANCA EXAMINADORA

Prof. Dr. Pedro Thiago Valério de Souza (UFERSA)
Presidente

Prof. Dr. Francisco Carlos Gurgel da Silva Segundo (UFERSA)
Membro Examinador

Prof. Dr. Reudismam Rolim de Sousa (UFERSA)
Membro Examinador

Prof. Mr. Alex Pinheiro Feitosa (UFERSA)
Membro Examinador

“Winter is coming!”

Game of Thrones

AGRADECIMENTOS

Primeiramente a Deus, por me dar as forças necessárias para superar os desafios acadêmicos, pelas suas orientações e bênçãos, que têm sido uma fonte constante de inspiração e motivação em minha jornada.

Ao meu orientador, o Prof. Dr. Pedro Thiago Valério de Souza, pela sua paciência e orientação dadas durante a preparação da metodologia deste trabalho, de forma crucial. A sua experiência com a pesquisa e a orientação acadêmica me proporcionou a segurança necessária para que eu conduzisse este trabalho de conclusão de curso da melhor forma possível.

RESUMO

No contexto atual, a tecnologia desempenha um papel essencial em nossa sociedade, moldando significativamente nossa forma de vida e interação com o mundo. À medida que as cidades crescem e a produção de resíduos aumenta, a gestão eficiente desses materiais se torna uma necessidade urgente para promover a sustentabilidade ambiental. É nesse cenário que um sistema de monitoramento de lixeiras emerge como solução inovadora e relevante. Este estudo explora a crescente importância da tecnologia na resolução dos desafios urbanos, destacando o papel fundamental do *Greentrash* na gestão de resíduos. O sistema capacita os usuários a adotar práticas conscientes em relação aos resíduos, o *Greentrash* atende as demandas de uma sociedade que prioriza cada vez mais a sustentabilidade e a qualidade de vida. Assim, esse sistema ilustra, de forma exemplar, como a tecnologia pode ser aplicada de maneira inovadora, para abordar questões ambientais, contribuindo para um futuro mais limpo e sustentável. A plataforma evidencia como a tecnologia possui o potencial de influenciar, de maneira positiva, nossa forma de viver e de conexão com o ambiente, apresentando visão otimista para a gestão de resíduos no horizonte futuro.

Palavras-chave: Sistema, ESP8266, HC-SR04, Greentrash, Protótipo, Gestão de resíduos.

ABSTRACT

Currently, technology is essential in society, influencing people's lives and interaction with the world. As cities grow and waste production increases, efficient management of these materials becomes necessary and urgent for environmental sustainability. In this scenario, a trash monitoring system emerges as an innovative and relevant solution. This study explores the importance of technology in solving urban challenges, highlighting the role of Greentrash in waste management. The system enables users to adopt conscious practices in relation to waste, Greentrash meets the demands of a society that prioritizes sustainability and quality of life. This system illustrates how to apply technology to address environmental issues, contributing to a sustainable future. The platform highlights the potential of technology in influencing the way of living and connecting with the environment, presenting an optimistic vision for waste management in the future.

Keywords: System, ESP8266, HC-SR04, Greentrash, Prototype, Waste management.

LISTA DE FIGURAS

Figura 1	–	ESP8266	15
Figura 2	–	Sensor ultrassônico HC-SR04.....	18
Figura 3	–	Conversor de nível lógico bidirecional.....	20
Figura 4	–	Diagrama do protótipo do sistema.....	20
Figura 5	–	Circuito montado parte 1.....	21
Figura 6	–	Circuito montado parte 2.....	22
Figura 7	–	Circuito montado parte 3.....	22
Figura 8	–	Circuito montado parte 4.....	23

LISTA DE TABELAS

Tabela 1	– Tabela geral de especificações do ESP8266.....	16
Tabela 2	– Tabela geral de especificações do HC-SR04.....	17

LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet</i> das Coisas
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
ADC	Conversor Analógico-Digital
PWM	Pulse Width Modulation

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVO GERAL	13
1.2 OBJETIVOS ESPECÍFICOS	13
1.3 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 <i>INTERNET</i> DAS COISAS (IoT)	14
2.2 ESP8266	15
2.3 SENSOR ULTRASSÔNICO (HC-SR04)	16
3 SISTEMA DE MONITORAMENTO DE LIXEIRAS - <i>GREENTRASH</i>.....	18
3.1 COMPONENTES.....	18
3.2 DIAGRAMA	20
3.3 PREPARAÇÃO E MONTAGEM.....	21
3.4 PROGRAMAÇÃO DO ESP8266	23
3.5 INTEGRAÇÃO COM <i>SOFTWARE</i>	25
4 CONSIDERAÇÕES FINAIS.....	33
REFERÊNCIAS	34

1 INTRODUÇÃO

No contexto presente, a tecnologia desempenha um papel crucial na sociedade, sendo frequentemente utilizada para melhorar a qualidade de vida da humanidade. De acordo com uma pesquisa conduzida pela ABRELPE/IBGE, sobre a situação dos resíduos sólidos no Brasil, em 2022, o país gerou 81,8 milhões de toneladas de resíduos, o que equivale a 224 mil toneladas diárias. Como resultado, a média de produção de resíduos por habitante no Brasil foi de 1,043kg por dia. Vale destacar que 20,2 milhões de toneladas desses resíduos foram produzidas na região Nordeste do país. Do total de resíduos gerados no Nordeste, apenas 82,70% foram coletados, enquanto 17,3% não foram recolhidos.

Esses dados destacam a urgente necessidade de uma gestão de resíduos mais eficaz, uma vez que a ausência de um plano de coleta eficiente resulta na presença de resíduos nas vias públicas, criando desafios ambientais e de saúde pública que não podem mais ser ignorados. Em cidades como Pau dos Ferros/RN, a falta de implementação de políticas públicas para o manejo de resíduos sólidos é evidente, e seus efeitos adversos são observados no dia a dia da comunidade. Embora os níveis de poluição em Pau dos Ferros não se comparem aos das grandes cidades brasileiras, a desorganização na coleta de lixo é uma realidade preocupante. Nesse contexto, lixeiras públicas inadequadas e veículos de coleta são os principais meios de controle da poluição, mas a coleta de lixo não é seletiva.

Portanto, a utilização da tecnologia emerge com abordagem eficaz, para enfrentar problemas como o acúmulo de resíduos nas ruas. É evidente que a sociedade passa por transformações significativas em diversas áreas, e nesse contexto surge a ideia de desenvolver um sistema web adaptável para lixeiras convencionais. Esse sistema é capaz de fornecer informações sobre o status da lixeira, indicando se está cheia ou vazia. Esses dados são obtidos por meio de um dispositivo e um sensor instalados na tampa da lixeira, que, em seguida, são transmitidos para o sistema web. Esse sistema desempenhará um papel fundamental na gestão eficiente dos dados coletados, prometendo aprimorar a gestão de resíduos, contribuindo para a preservação do meio ambiente e melhor qualidade de vida nas cidades.

O *Greentrash* apresenta uma abordagem sustentável. Por meio do monitoramento contínuo das lixeiras, será possível otimizar o serviço de coleta, evitando a sobrecarga dos recipientes e reduzindo os impactos ambientais causados pelo descarte irregular. Além disso, o sistema permitirá um planejamento mais eficaz das rotas de coleta, otimizando o uso do tempo e recursos disponíveis.

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo geral: Desenvolver um *hardware* que se comunique com um sistema de monitoramento de lixeiras, para efetuar o gerenciamento eficiente do lixo.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos são fundamentais para alcançar o objetivo geral do projeto. Eles foram definidos de forma cuidadosa, para que pudessem ajudar a concluir o projeto com sucesso.

Objetivos específicos, que guiarão este trabalho:

- a) Desenvolver um *hardware* que monitore o status da lixeira e envie os dados para um sistema de gestão;
- b) Estabelecer comunicação eficiente entre a lixeira e o sistema;
- c) Realizar testes e validações, para garantir eficiência e confiabilidade do *hardware*.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma.

O segundo Capítulo apresenta a fundamentação teórica e as discussões realizadas, abordando as tecnologias utilizadas em cada componente selecionado, assim como o seu funcionamento e a sua descrição.

O terceiro Capítulo mostra o desenvolvimento e a montagem do protótipo, demonstrando todo o raciocínio lógico utilizado e implementado para a utilização e programação dos componentes.

No último Capítulo, é evidenciada a conclusão do projeto, apresentando os resultados e as discussões, além da possibilidade de abordar questões futuras relacionados ao aprimoramento do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

O propósito deste capítulo é estabelecer um contexto sólido e proporcionar uma compreensão aprofundada acerca do sistema de monitoramento de lixeiras. Nesta Seção, serão explorados os princípios, as teorias, as pesquisas relevantes e a utilização de ferramentas sem necessidade de programação.

Para os próximos anos, a previsão aponta para um aumento significativo na geração de resíduos urbanos, estimando mais de 2 bilhões de toneladas até 2025. Isso indica um acentuado crescimento na produção de resíduos urbanos, quando comparado à taxa de urbanização.

O modelo de gestão tradicional geralmente se baseia em rotas predefinidas, frequentemente ignorando se as lixeiras estão realmente cheias ou não. Ter o conhecimento antecipado do estado das lixeiras resultaria em economia de tempo e de combustível (ALMEIDA *et al.*, 2021).

Muitas empresas ainda adotam o método manual de coleta de lixo, no qual seus funcionários percorrem os diferentes setores da cidade para localizar e esvaziar as lixeiras. No entanto, esse modelo causa um desgaste considerável para os profissionais da limpeza, que perdem muito tempo procurando recipientes que estejam cheios e que necessitem ser esvaziados (BRANDÃO-JÚNIOR *et al.*, 2018).

No entanto, com o avanço da *Internet das Coisas* (IoT), uma tendência cada vez mais presente em nosso cotidiano, uma variedade de dispositivos capazes de se comunicar e interagir de várias maneiras com os usuários está disponível. Exemplos como relógios inteligentes, que monitoram a frequência cardíaca, carros inteligentes, drones e muitos outros casos de aplicação da IoT (BRANDÃO-JÚNIOR *et al.*, 2018).

Com o objetivo de aliviar o trabalho dos funcionários encarregados de verificar o estado de todas as lixeiras nos locais, e considerando o conceito da IoT juntamente com os desafios relacionados à coleta de lixo, está sendo desenvolvida uma solução por meio de lixeiras automatizadas. Isso visa melhorar a qualidade do trabalho das empresas, identificando em tempo real, quais recipientes estão cheios ou vazios.

2.1 *INTERNET DAS COISAS* (IoT)

No âmbito desta pesquisa, é fundamental compreender o papel desempenhado pela *Internet das Coisas* (IoT), que representa uma verdadeira revolução tecnológica.

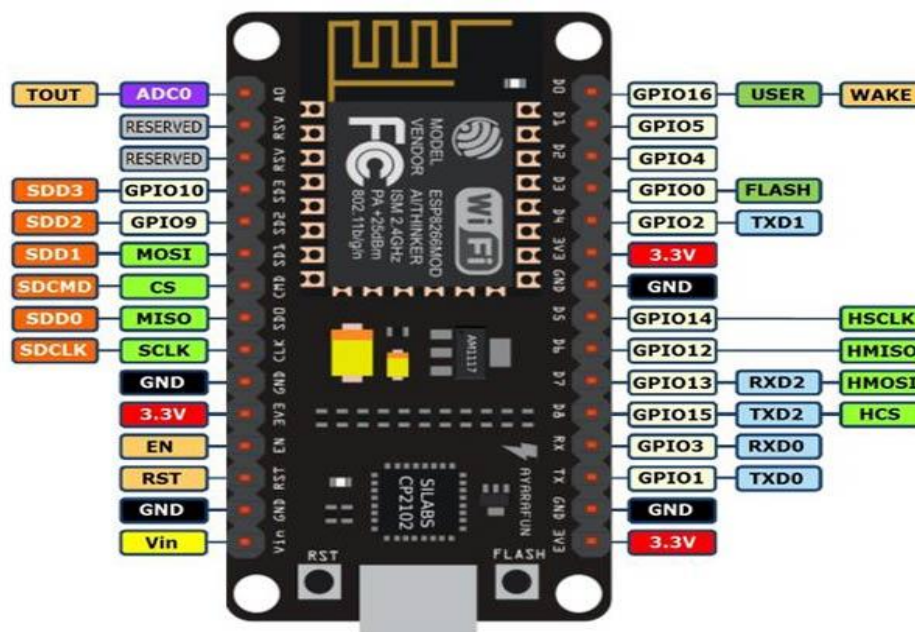
A sigla IoT refere-se à “*Internet das Coisas*,” que é uma rede interligada de objetos físicos, dispositivos, veículos e outros elementos incorporados com sensores, *software* e tecnologia de conectividade. Essa interconexão possibilita a coleta, a troca e a análise de dados, para aprimorar processos, tomar decisões e aumentar a eficiência em diversas áreas. A IoT tem o potencial para transformar setores como indústria, saúde, agricultura, transporte e muito mais, permitindo uma maior automatização e controle (ATZORI et al., 2010).

O sistema *Greentrash* é uma inovação fundamentada na concepção da *Internet das Coisas*, que envolve a interligação de objetos físicos por meio de sensores e microcontroladores. O sistema foi concebido com base na arquitetura de cliente-servidor. O servidor corresponde ao sistema web que recebe as requisições do cliente, processa essas solicitações e disponibiliza os dados necessários, permitindo o controle e o monitoramento remoto das lixeiras.

2.2 ESP8266

O ESP8266 é um microcontrolador com conectividade *Wi-Fi* utilizando o protocolo 802.11 com conectividade sem fio e de custo acessível, desenvolvido pela empresa Espressif Systems, com sede em Xangai, China. Este dispositivo é amplamente empregado em projetos relacionados à *Internet das Coisas* (IoT), ele inclui diversas funcionalidades, como pinos GPIO, interfaces I2C, SPI e UART, capacidade de entrada analógica ADC, saída PWM e até um sensor interno de temperatura.

Figura 1: ESP8266.



Fonte: <https://blog.smartkits.com.br/esp8266-como-programar-o-nodemcu-atraves-da-arduino-ide/>. Acesso em: 18 set. 2023.

O ESP8266 está no mercado desde 2014. A principal fonte de referência sobre o ESP8266 é o livro *Kolban's Book on ESP8266*, de autoria de N. Kolban, publicado em 2016. Esse livro aborda, em detalhes, as técnicas de programação do ESP8266, suas aplicações práticas e os seus requisitos de eletrônica necessários para a realização dos projetos apresentados.

Tabela 1: Tabela geral de especificações do ESP8266.

Consumo de Corrente	10 μ A
Voltagem	3.3V
Memória Flash	16MB max (512k normal)
Velocidade do processador	80-160MHz
Processador	Tensilica L106 32bit
RAM	32K + 80K
Analógico para digital	1 entrada com 1024 de resolução
GPIOs	17 (multiplexada com outras funções)
Suporte 802.11	b/g/n/d/e/i/k/r
Máxima corrente de conexão TCP	5

Fonte: Informações do fabricante.

Existem alguns trabalhos disponíveis na *internet* que fazem uso do ESP8266 como *Internet das Coisas com ESP8266, Arduino e Raspberry pi* de Sérgio de Oliveira, entre outros. Porém, há uma carência de documentação mais formal e abrangente sobre o assunto, embora a comunidade de usuários tenha crescido consideravelmente, pois sua adoção no âmbito acadêmico ainda não está definida.

2.3 SENSOR ULTRASSÔNICO (HC-SR04)

Os sensores ultrassônicos utilizam ondas sonoras para calcular a distância entre o sensor e o objeto mais próximo em seu alcance. Eles conseguem isso emitindo pulsos de som em uma

frequência que está acima do espectro de audição humana, tornando-se, essencialmente, sensores de som de alta frequência.

O sensor HC-SR04, que foi selecionado para a execução do projeto, emite uma onda sonora em uma frequência determinada e, em seguida, detecta quando essa onda sonora específica reflete em um objeto e retorna. O sensor registra o tempo decorrido desde o momento em que a onda sonora foi emitida até o momento em que ela retorna.

Desse modo, com essa informação sobre a velocidade do som e o tempo de viagem, é possível calcular a distância percorrida usando a seguinte equação:

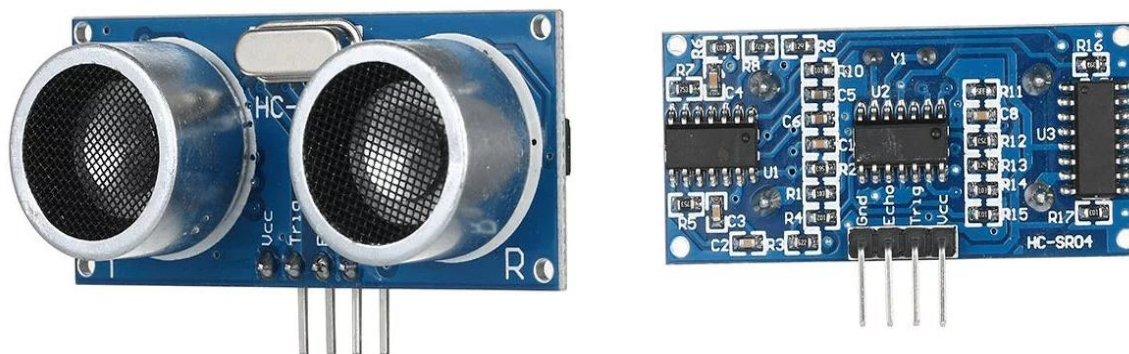
$$\text{Equação: } d = v * t.$$

Tabela 2: Tabela geral de especificações do HC-SR04.

Fonte de Alimentação	+5V CC
Corrente quiescente	<2mA
Corrente de trabalho	15mA
Ângulo de alcance	<15°
Precisão	0,3cm
Ângulo de medição	30°
Largura de pulso de entrada do gatilho	10uS
Distância de alcance	2.400 cm
Dimensão	45mm x 20mm x 15mm
Peso	10g

Fonte: Informações do fabricante

Figura 2: Sensor ultrassônico HC-SR04.



Fonte: <https://capsistema.com.br/index.php/2020/12/19/radar-de-reversao-ultrassonico-com-arduino/>. Acesso em: 18 set. 2023.

3 SISTEMA DE MONITORAMENTO DE LIXEIRAS – *GREENTRASH*

Este capítulo apresenta como ocorreu o desenvolvimento do protótipo do sistema *Greentrash*.

O sistema *Greentrash*, uma inovação criada com uma integração entre *hardware* e *software*, emerge como uma solução eficaz para os desafios contemporâneos relacionados à gestão de resíduos sólidos.

O *Greentrash* está redefinindo nossa abordagem frente à administração de resíduos. Este sistema não apenas simplifica a coleta de dados sobre o estado das lixeiras, mas também oferece uma visão abrangente em tempo real, capacitando a tomada de decisões proativas na gestão de resíduos.

O *Greentrash* foi desenvolvido com base em uma integração entre um *hardware* e um *software*, em que o protótipo é informar o status da lixeira, utilizando integrações com uma API desenvolvida na plataforma Bubble.io utilizando como método de requisição HTTPS, para fazer a comunicação entre ambos.

3.1 COMPONENTES

O sistema *Greentrash* se destaca pela seleção cuidadosa de componentes que desempenham papéis cruciais em sua funcionalidade. Primeiramente, a escolha do ESP8266 é

uma decisão inteligente, uma vez que esse microcontrolador é equipado com um módulo *Wi-Fi* integrado possibilitando uma conectividade sem fio utilizando o protocolo 802.11, tornando-o uma escolha ideal para soluções de conectividade no *Greentrash*.

Além disso, o suporte à programação no Arduino IDE amplia consideravelmente a acessibilidade e a flexibilidade na criação de lógica personalizada para o sistema. Com essas capacidades combinadas, o ESP8266 possibilita a criação de aplicações de automação residencial, que podem ser facilmente integradas em redes *Wi-Fi* existentes, tornando o *Greentrash* altamente adaptável e poderoso.

O sensor ultrassônico HC-SR04 desempenha um papel fundamental no sistema *Greentrash* devido à sua capacidade de detecção de distâncias curtas. Sua capacidade de captar objetos em proximidade imediata é essencial para aplicações que requerem precisão na detecção de obstáculos ou medição de distâncias no contexto desse sistema. Essa característica torna o HC-SR04 uma escolha para a automação residencial, em que a detecção precisa de objetos próximos é, muitas vezes, necessária.

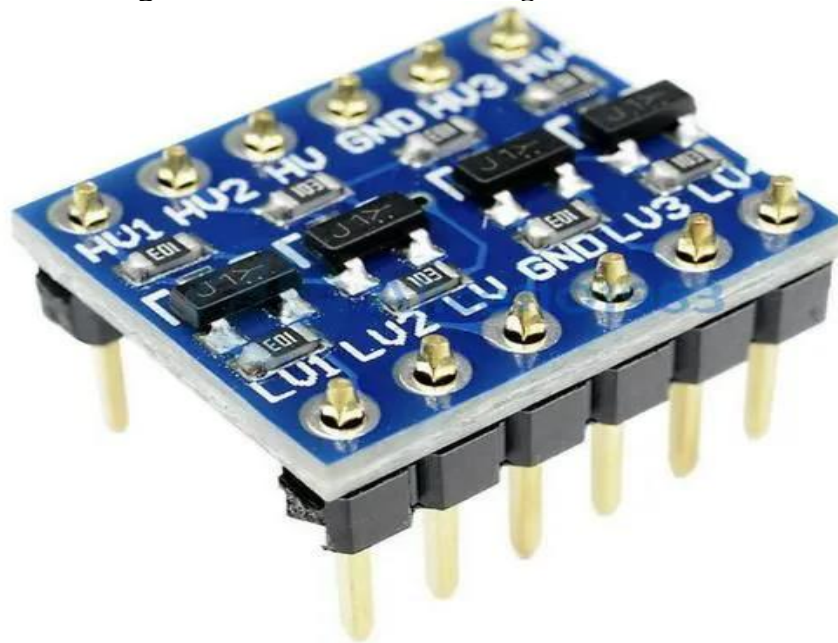
O sensor utiliza ondas ultrassônicas que voltam ao contato com o primeiro objeto, assim possível medir o tempo que a onda levou para ir e voltar podendo assim calcular a distância com base na velocidade do som de 340m/s. Com isso é possível identificar se a lixeira estará cheia ou vazia com base na distância do objeto mais próximo do sensor.

No entanto, a integração perfeita desses componentes no *Greentrash* envolve considerações elétricas importantes. O uso de um conversor bidirecional é uma solução essencial, para harmonizar as diferenças de tensão entre o HC-SR04 e o ESP8266 no *Greentrash*. Enquanto o sensor opera com uma tensão de 5V, o ESP8266 opera a 3,3V.

Assim, o conversor bidirecional atua como uma ponte, garantindo que o sinal seja traduzido de forma confiável entre os componentes, tornando o *Greentrash* seguro e eficaz em seu funcionamento.

Portanto, a combinação destes componentes – o ESP8266, o sensor ultrassônico HC-SR04 e o conversor bidirecional – cria uma base para o sucesso do sistema *Greentrash*, proporcionando conectividade, precisão e compatibilidade elétrica, itens essenciais para seu funcionamento eficaz.

Figura 3: Conversor de nível lógico bidirecional.

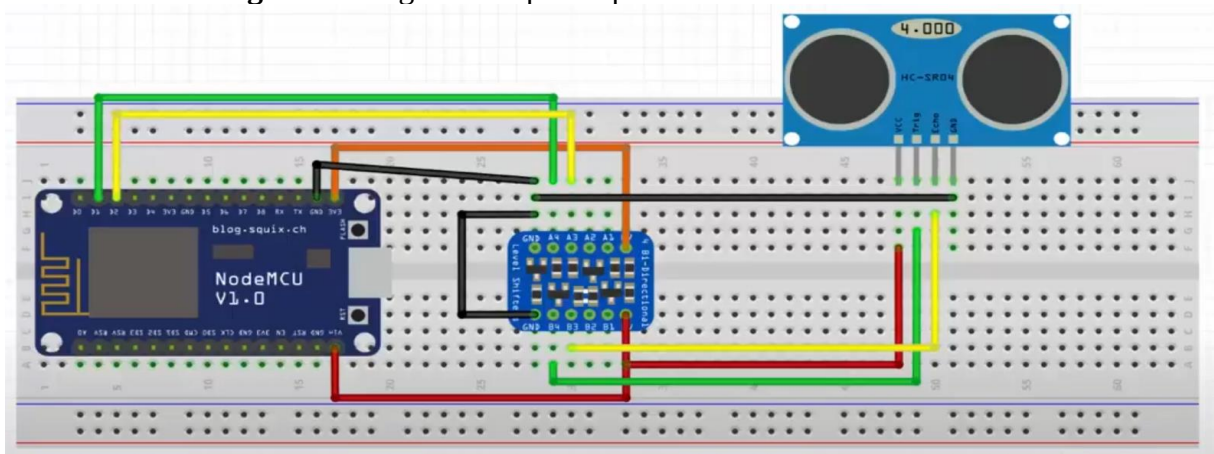


Fonte: <https://www.smartkits.com.br/modulo-conversor-de-nivel-logico-3-3-5v-bidirecional/>. Acesso em: 18 set. 2023.

3.2 DIAGRAMA

A montagem do diagrama do sistema *Greentrash*, que utiliza o ESP8266, o sensor ultrassônico HC-SR04 e um conversor bidirecional, segue uma série de etapas para garantir que os componentes funcionem em conjunto e de maneira eficaz. O projeto iniciou o processo com o diagrama feito utilizando o *software* Fritzing.

Figura 4: Diagrama do protótipo do sistema *GREENTRASH*.



Fonte: Autoria própria (2023).

3.3 PREPARAÇÃO E MONTAGEM

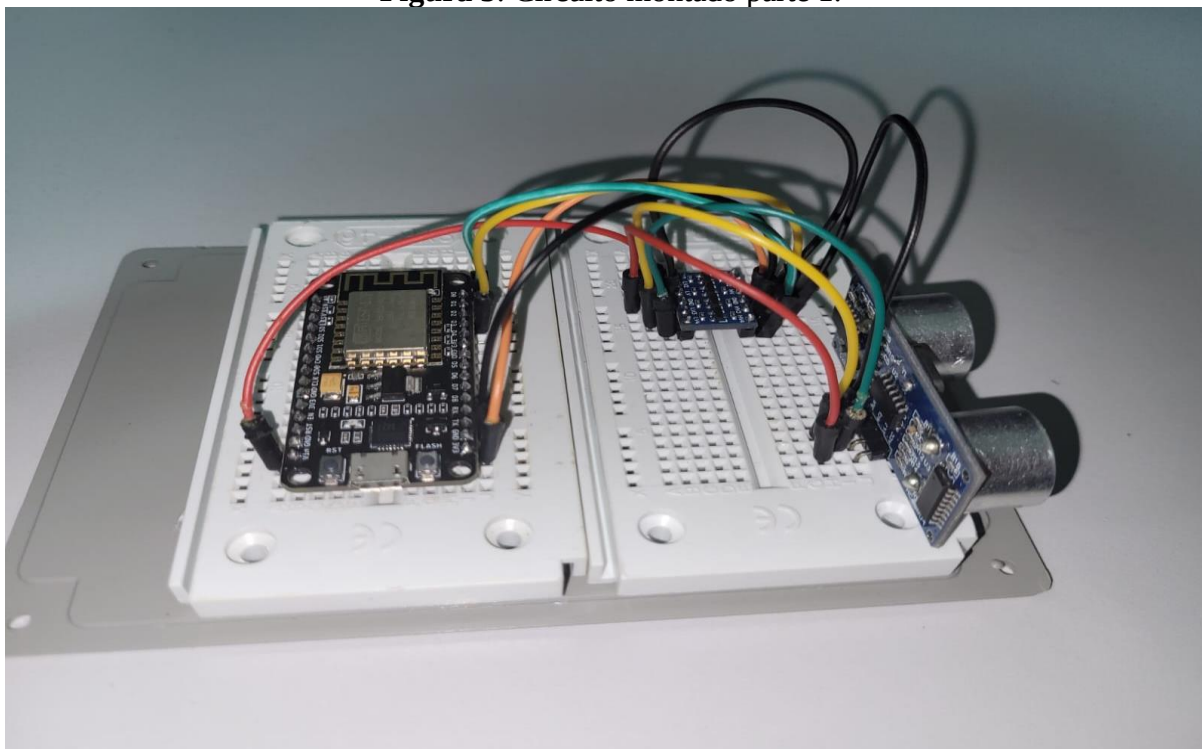
Antes de iniciar o processo de montagem do diagrama, todos os componentes necessários estavam em boas condições de uso e funcionamento. Feito isso, certifica-se de ter o ESP8266, o sensor HC-SR04 e o conversor bidirecional disponíveis à mão para uso.

Verifica-se também o *datasheet* de um dos componentes, para identificar os pinos de entrada e saída necessários. Por exemplo, no HC-SR04, há pinos para *Trigger* (gatilho), *Echo* (eco), VCC (alimentação) e GND (terra).

Em seguida, conecta-se os componentes eletricamente. Por exemplo, conectou-se a saída de 5V do sensor HC-SR04 ao conversor bidirecional, para a obtenção dos 3,3V que são necessários para o ESP8266. Depois disso, os pinos GND dos componentes em comum foram ligados.

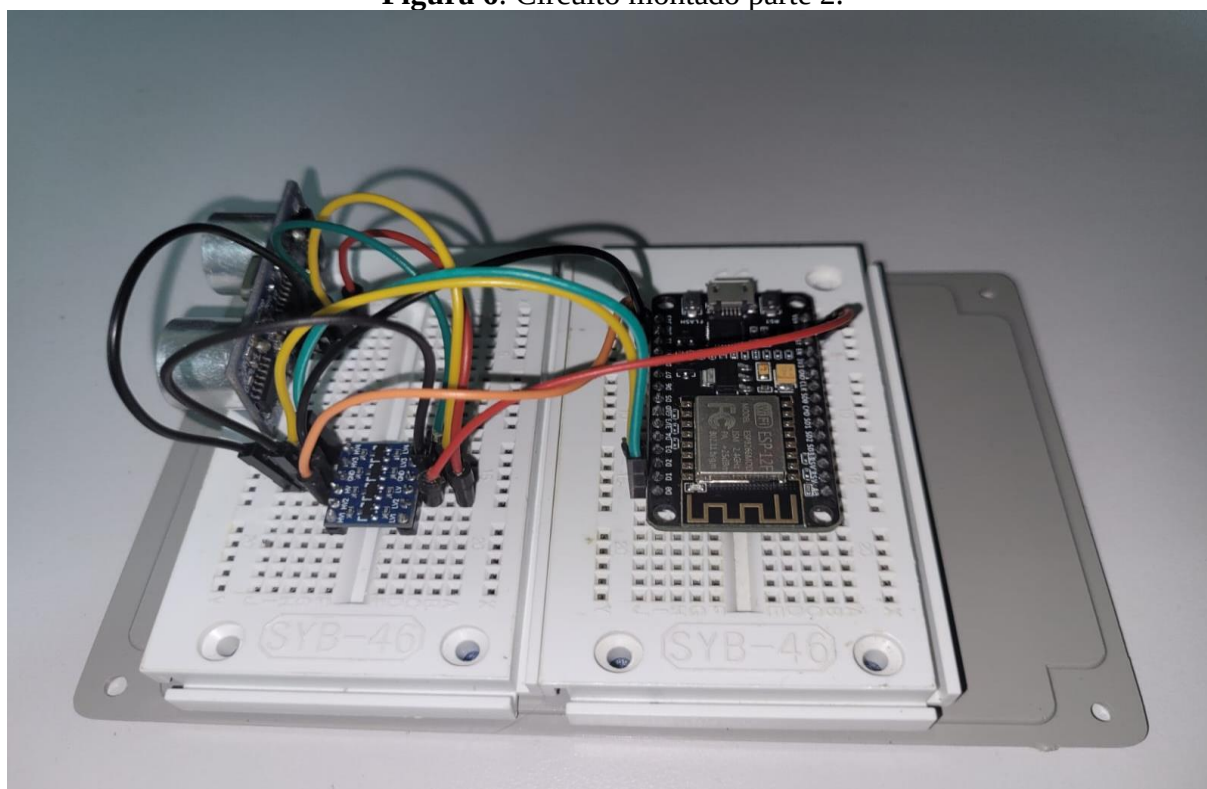
Seguimos conectando os pinos de sinal entre os componentes. Por exemplo, conectou-se o pino Trigger do sensor HC-SR04 a um pino de saída digital do ESP8266, e o pino Echo do sensor a um pino de entrada digital do ESP8266, por intermédio do sensor lógico bidirecional.

Figura 5: Circuito montado parte 1.



Fonte: Autoria própria (2023).

Figura 6: Circuito montado parte 2.



Fonte: Autoria própria (2023).

Figura 7: Circuito montado parte 3.



Fonte: Autoria própria (2023).

Figura 8: Circuito montado parte 4.



Fonte: Autoria própria (2023).

Diante disso, temos o circuito montado, havendo necessidade de execução da próxima etapa, que se constitui da programação do ESP8266.

3.4 PROGRAMAÇÃO DO ESP8266

A programação do ESP8266, utilizando o Arduino IDE, é uma maneira popular e acessível de aproveitar, ao máximo, o poder desse microcontrolador *Wi-Fi*.

Programar o ESP8266 no Arduino IDE oferece uma interface familiar e ampla flexibilidade para criar projetos IoT com conectividade *Wi-Fi*. Nos certificamos de explorar a quantidade de recursos e tutoriais disponíveis *online*, para aproveitarmos, ao máximo, as capacidades do ESP8266 neste projeto utilizando a linguagem de programação do Arduino IDE.

Bibliotecas utilizadas:

- ESP8266WiFi.h
- WIFIClientSecure.h
- ESP8266WebServer.h
- ESP8266HTTPClient.h
- ArduinoJson.h

Conforme definido no diagrama, o pino do sensor ultrassônico Trigger se conecta com o D1 no ESP8266, e o pino Echo se conecta com o D2.

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

#define trigPin D1
#define echoPin D2

/* Set these to your desired credentials. */
const char *ssid = "Yure"; // ENTER YOUR WIFI SETTINGS
const char *password = "96028076";

// Link to read data from
// https://jsonplaceholder.typicode.com/comments?postId=7
// Web/Server address to read/write from
const char *host = "regiotcc.bubbleapps.io";
const int httpsPort = 443; // HTTPS= 443 and HTTP = 80

// SHA1 finger print of certificate use web browser to view and copy
const char fingerprint[] PROGMEM = "64 7C BE 85 BB 17 62 1F 07 88 5F F5 CE C1
99 67 D2 5B 75 75";

//=====
//                               Power on setup
//=====

void setup() {
  delay(1000);
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
  WiFi.mode(WIFI_OFF); // Prevents reconnection issue (taking too long
to connect)
  delay(1000);
  WiFi.mode(WIFI_STA); // Only Station No AP, This line hides the
viewing of ESP as wifi hotspot

  WiFi.begin(ssid, password); // Connect to your WiFi router
  Serial.println("");

  Serial.print("Connecting");
  // Wait for connection
```

```

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

// If connection successful, show IP address in serial monitor
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); // IP address assigned to your ESP
}

```

Com essas especificações, o ESP8266 já é capaz de se conectar a uma rede *Wi-Fi* estabelecida. O que aponta para a execução da próxima etapa, que é desenvolver uma função para calcular a distância para o objeto mais próximo.

```

int calcular_distancia(){

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duracao = pulseIn(echoPin, HIGH, 1000000);
    distancia = (duracao*340/10000)/2;
    Serial.println("distancia: ");
    Serial.println(distancia);
    return distancia;
}

```

Concluindo esse processo, já temos tudo que é necessário para o sistema funcionar corretamente e nos retornar a distância do sensor ultrassônico para o objeto mais próximo.

3.5 INTEGRAÇÃO COM O SOFTWARE

A integração do *software Greentrash* com o ESP8266 envolveu o uso de HTTPS. O uso do protocolo HTTPS (*Hypertext Transfer Protocol Secure*) é fundamental para garantir a segurança e a privacidade dos dados transmitidos pela rede, o que é particularmente importante em aplicações IoT, como o *Greentrash*, em que informações sensíveis podem ser transmitidas ou recebidas.

A seguir, estão alguns aspectos importantes sobre a integração com HTTPS no contexto do *Greentrash*.

- Segurança de dados. Ao utilizar HTTPS, todas as informações trocadas entre o ESP8266 e servidores remotos são criptografadas, impedindo que terceiros mal-intencionados interceptem ou acessem os dados.
- Certificados SSL/TLS. Para estabelecer uma conexão HTTPS, é necessário o uso de certificados SSL/TLS (*Secure Sockets Layer/Transport Layer Security*). Isso requer a configuração adequada de certificados tanto no servidor quanto no ESP8266, para garantir a autenticidade da conexão.
- Libraries. No Arduino IDE, utilizamos bibliotecas específicas, como a “ESP8266HTTPClient” e a “WiFiClientSecure”, para facilitar a comunicação HTTPS. Essas bibliotecas permitem enviar e receber dados com segurança por meio de URLs HTTPS.
- Autenticação e chaves. Certificamos de que o sistema *Greentrash* esteja configurado para lidar com autenticação, como senhas ou chaves de API, de forma segura. Isso pode ser essencial para acessar serviços remotos protegidos por autenticação.
- Requisitos de *hardware*. Certifique-se de que o ESP8266 tenha conectividade *Wi-Fi* confiável e sua infraestrutura de rede seja robusta para suportar comunicações HTTPS.
- Testes e depuração. Realize testes abrangentes e depure cuidadosamente a integração com HTTPS, para garantir que todas as conexões e transferências de dados funcionem.

A integração com HTTPS no *software Greentrash* não apenas protege os dados sensíveis, mas também garante a integridade e a autenticidade das comunicações, tornando seu sistema mais seguro e confiável em ambientes de IoT, em que a segurança dos dados é uma prioridade. Nos certificamos de seguir as melhores práticas de segurança, ao configurarmos e implementarmos a comunicação HTTPS neste projeto.

Foi determinado que cada dispositivo *Greentrash* terá um ID único definido dentro do seu código fonte, esse ID será enviado para uma API, para que o dispositivo seja identificado pelo *software*. Desse modo, utilizando todos os requisitos necessários para uma conexão HTTPS, e com todos os requisitos para uma chamada POST para uma API, foi desenvolvida uma função para tal:

```
void requisicao (int distancia){
  WiFiClientSecure httpsClient;    // Declare object of class WiFiClient

  Serial.println(host);

  Serial.printf("Using fingerprint '%s'\n", fingerprint);
```

```

httpsClient.setFingerprint(fingerprint);
httpsClient.setTimeout(15000); // 15 Seconds
delay(1000);

Serial.print("HTTPS Connecting");
int r = 0; // Retry counter
while ((!httpsClient.connect(host, httpsPort)) && (r < 30)) {
    delay(100);
    Serial.print(".");
    r++;
}
if (r == 30) {
    Serial.println("Connection failed");
} else {
    Serial.println("Connected to web");
}

String postData;

// Create JSON object
DynamicJsonDocument jsonDocument(1024);
jsonDocument["distancia"] = distancia;
jsonDocument["id"] = "1234";

// Serialize JSON to string
serializeJson(jsonDocument, postData);

Serial.print("Requesting URL: ");
Serial.println(host);

// Construct the POST request
String request = String("POST ") + "/version-test/api/1.1/wf/teste" + "
HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Content-Type: application/json\r\n" +
    "Content-Length: " + String(postData.length()) + "\r\n" +
    "Connection: close\r\n\r\n" +
    postData + "\r\n";

httpsClient.print(request);

Serial.println("Request sent");

while (httpsClient.connected()) {
    String line = httpsClient.readStringUntil('\n');
    if (line == "\r") {
        Serial.println("Headers received");
        break;
    }
}

```

```

    }
  }
}

```

Essa função recebe a distância como parâmetro, realiza a conexão HTTPS com o servidor e realiza uma requisição POST para uma API definida dentro do *software*, enviando parâmetros como a distância e o ID do dispositivo em formato Json. Assim, com todas as funções definidas o código final ficou desta forma.

```

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

#define trigPin D1
#define echoPin D2

/* Set these to your desired credentials. */
const char *ssid = "Yure"; // ENTER YOUR WIFI SETTINGS
const char *password = "96028076";

//          Link          to          read          data          from
https://jsonplaceholder.typicode.com/comments?postId=7
// Web/Server address to read/write from
const char *host = "regiotcc.bubbleapps.io";
const int httpsPort = 443; // HTTPS= 443 and HTTP = 80

// SHA1 finger print of certificate use web browser to view and
copy
const char fingerprint[] PROGMEM = "64 7C BE 85 BB 17 62 1F 07
88 5F F5 CE C1 99 67 D2 5B 75 75";

//=====
=====

```

```

//                                Power on setup
//=====
=====

void setup() {
    delay(1000);
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    digitalWrite(trigPin, LOW);
    WiFi.mode(WIFI_OFF);           // Prevents reconnection issue
    (taking too long to connect)
    delay(1000);
    WiFi.mode(WIFI_STA);           // Only Station No AP, This line
    hides the viewing of ESP as wifi hotspot

    WiFi.begin(ssid, password);    // Connect to your WiFi router
    Serial.println("");

    Serial.print("Connecting");
    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // If connection successful, show IP address in serial monitor
    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP()); // IP address assigned to your

```

ESP

```

}

//=====
=====

//                      Main Program Loop
//=====
=====

long duracao;
int distancia;

void loop() {

    requisicao(calcular_distancia());

    delay(5000); // POST Data every 2 seconds
}

int calcular_distancia(){
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duracao = pulseIn(echoPin, HIGH, 1000000);
    distancia = (duracao*340/10000)/2;
    Serial.println("distancia: ");
    Serial.println(distancia);
    return distancia;
}

```

```

void requisicao (int distancia){
    WiFiClientSecure httpsClient;    // Declare object of class
WiFiClient

    Serial.println(host);

    Serial.printf("Using fingerprint '%s'\n", fingerprint);
    httpsClient.setFingerprint(fingerprint);
    httpsClient.setTimeout(15000); // 15 Seconds
    delay(1000);

    Serial.print("HTTPS Connecting");
    int r = 0; // Retry counter
    while ((!httpsClient.connect(host, httpsPort)) && (r < 30)) {
        delay(100);
        Serial.print(".");
        r++;
    }
    if (r == 30) {
        Serial.println("Connection failed");
    } else {
        Serial.println("Connected to web");
    }

    String postData;

    // Create JSON object
    DynamicJsonDocument jsonDocument(1024);
    jsonDocument["distancia"] = distancia;
    jsonDocument["id"] = "1234";

    // Serialize JSON to string
    serializeJson(jsonDocument, postData);

```



```

Serial.print("Requesting URL: ");
Serial.println(host);

// Construct the POST request
String request = String("POST ") + "/version-
test/api/1.1/wf/teste" + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Content-Type: application/json\r\n" +
    "Content-Length: " +
String(postData.length()) + "\r\n" +
    "Connection: close\r\n\r\n" +
    postData + "\r\n";

httpsClient.print(request);

Serial.println("Request sent");

while (httpsClient.connected()) {
    String line = httpsClient.readStringUntil('\n');
    if (line == "\r") {
        Serial.println("Headers received");
        break;
    }
}
return;
}

```

Todo o código foi desenvolvido na linguagem padrão do Arduino IDE enquanto a API foi desenvolvida na plataforma Bubble utilizando linguagem no-code, com isso tem-se o suficiente para o funcionamento eficaz e eficiente do sistema.

4 CONSIDERAÇÕES FINAIS

Diante da constituição desta pesquisa, constatou-se que todos os objetivos elencados neste projeto foram alcançados, como desenvolver um *hardware* que monitore o status da lixeira e envie para um sistema de gestão de lixeiras (*software*). Além disso, foi estabelecida uma comunicação eficiente entre o *hardware* e o *software*, a partir da realização de testes e das validações para garantir o efeito ou o resultado esperado.

Após realizados esses objetivos, obtivemos os resultados classificados como positivos e eficientes, uma vez que o protótipo apresentou um desempenho excelente, estando em conformidade com o que foi proposto como sua finalidade.

Desse modo, este projeto de pesquisa representa uma contribuição significativa para a área de gestão de resíduos sólidos, uma vez que oferece uma forma de solução inovadora para a otimização da coleta de lixo urbano.

Nesse contexto, o sistema *Greentrash*, por meio do protótipo que foi construído neste projeto, provou ser altamente usável e eficiente, além de poder ser aplicado em ambientes urbanos, como em parques, empresas e outros locais com necessidades de gerenciamento de resíduos.

Nesse sentido, o sistema desenvolvido neste projeto se destaca em comparação com as abordagens tradicionais de coleta de lixo, proporcionando uma coleta mais direcionada e eficiente.

Assim, reconhecemos que o sistema atual de uso para a coleta de resíduos nas cidades tem muitas limitações, como o alcance de comunicação com o *Wi-Fi*, o que representa uma carência de inovação tecnológica.

Isso revela que é necessário haver futuras melhorias, para que possam incluir a implementação de tecnologias de comunicação de *internet*, utilizando outros dispositivos tecnológicos, como também a utilização de fontes de alimentações diferentes, como energia solar ou baterias.

Por fim, o desenvolvimento deste projeto pode ser visto também como uma forma de pesquisa que abre novas discussões acerca das inovações a serem implementadas na área de coleta de resíduos urbanos, por meio do uso da tecnologia para aprimorar e tornar esse tipo de serviço mais eficiente.

REFERÊNCIAS

ALMEIDA, M. S. T.; MELO, M. C. de; BOARETO, M. D.; REZENDE, R. M. Análise da correlação do volume de resíduos sólidos provenientes da construção civil com o produto interno bruto para os municípios no estado de Minas Gerais. **Revista Augustu**, v. 24, n. 49, p. 320-331. Rio de Janeiro, 2020. Disponível em: <https://doi.org/10.15202/1981896.2019v24n49p320>. Acesso em: 18 out. 2023.

ATZORI, L.; EIRA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, v. 54, n. 15, p. 2787-2805, 28 out. 2010.

BRANDÃO-JÚNIOR, E. L.; BERNARDO, G. P; BERNARDO, L. P; NASCIMENTO, S. I. B. do; LIMA, B. F. R; SILVA, K. V. C. C.; CAVALCANTE, G. M. E; RULIM, A. L. L; DUARTE, J. O. Queima Inadequada de Resíduos Sólidos Domésticos, Principais Gases Tóxicos e Manifestação Clínicas: Uma Revisão de Literatura. **Revista Multidisciplinar e de Psicologia**, v. 12, n. 42, p. 602-612, 2018.

KOLBAN, N. **Kolban's Book on ESP8266**. 2016. Disponível em: https://leanpub.com/ESP8266_ESP32. Acesso em: 17 out. 2023.

MORGAN, E. J. **HC-SR04 Ultrasonic Sensor**. 2014. Disponível em: https://nafcom.es/fotos_pdt/108.MD0017/HC-SR04-ETC2.pdf. Acesso em: 18 out. 2023.

RODRIGUES, O. R. **Uso do Microcontrolador ESP8266 para Automação Residencial**. 2017. Monografia. (Graduação). Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação.