

ETEC JORGE STREET

CURSO DE ELETRÔNICA



BINIOT, A LIXEIRA INTELIGENTE

Davi Abreu de Carvalho
Gabryell Francesco Reis Pereira
Guilherme Lundquist da Costa
Robson Rioki Nakama

SÃO CAETANO DO SUL – SP
DEZEMBRO/2022

Davi Abreu de Carvalho
Gabryell Francesco Reis Pereira
Guilherme Lundquist da Costa
Robson Rioki Nakama

BINIOT, A LIXEIRA INTELIGENTE

Trabalho de Conclusão de Curso apresentado
como requisito para obtenção do título do Ensino
Médio com Educação Técnica em Eletrônica pela
ETEC Jorge Street.

Orientador(a): Profa. Cristina de Moura Ramos

SÃO CAETANO DO SUL – SP
DEZEMBRO/2022

Dedicamos este trabalho de conclusão de curso a Deus e aos nossos pais e amigos que sempre nos incentivaram.

AGRADECIMENTOS

Agradecemos, em primeiro lugar, a Deus que sempre nos conduziu com os devidos ensinamentos de amor, fraternidade e compaixão hoje e sempre.

Agradecemos aos nossos pais e aos nossos familiares que nos acompanharam e incentivaram a perseverar na busca pelo conhecimento.

Agradecemos aos nossos professores e colegas por nos ajudarem e apoiarem este trabalho.

Agradecemos aos nossos amigos que nos apoiaram durante o percurso.

“Se você acha que a internet mudou a sua vida, pense novamente. A Internet das coisas está prestes a mudar tudo de novo!” (Brendan O’Brien)

RESUMO

Observou-se em escolas, empresas e condomínios, que o sistema de coleta não é planejado eficientemente, pois a falta de informação das lixeiras para os funcionários torna o gerenciamento de descarte mais trabalhoso. O objetivo do projeto é otimizar as lixeiras de ambientes privados possibilitando um diagnóstico em tempo real delas. Este produto se baseia em uma lixeira de médio porte com sensores que identificam o nível, o peso e a quantidade de vezes que foi aberta. Esses dados são registrados em uma nuvem que são redirecionados a um aplicativo, por conexão wi-fi, que notifica quando o recipiente encher. A metodologia consiste em pesquisas bibliográficas e levantamentos de campos, relacionadas a sistemas de coleta nas cidades inteligentes, lixeiras automáticas e entrevistas com faxineiras. Vale ressaltar que esta obra procura apontar caminhos em eletrônica aos interessados em conhecer como acontece o processo de desenvolvimento de um protótipo que implementa IoT. Em síntese, este trabalho de conclusão de curso visou fomentar o aprimoramento das variadas situações e objetos que podem facilitar o cotidiano ao introduzir sistemas inteligentes.

Palavras-chave: Lixeira; Sensores; Aplicativo móvel; IoT.

ABSTRACT

It was observed in schools, companies, and condominiums that the collection system is not planned efficiently, because the lack of information about the garbage cans for the employees makes the waste management more laborious. The objective of the project is to optimize the wastebaskets in private environments by enabling a real-time diagnosis of them. This product is based on a medium-sized trash can with sensors that identify the level, weight, and number of times it has been opened. This data is registered in a cloud that is redirected to an application, via wi-fi connection, that notifies when the container fills up. The methodology consists of bibliographic research and field surveys, related to collection systems in smart cities, automatic rubbish bin, and interviews with cleaners. It is worth mentioning that this work seeks to point out paths in electronics to those interested in knowing how the development process of a prototype that implements IoT happens. In summary, this course completion work aimed to promote the improvement of the various situations and objects that can facilitate daily life by introducing intelligent systems.

Keywords: Trash can; Sensors; Mobile application; IoT.

LISTA DE ILUSTRAÇÕES

Figura 1 – Dimensões do Sensor Ultrassônico	15
Figura 2 – Ponte de Wheatstone.....	16
Figura 3 – Ligação de Duas Células de Carga.....	17
Figura 4 – Componente e Simbologias do LDR	17
Figura 5 – Fluxograma da Comunicação via Internet.....	21
Figura 6 – Fluxograma do Funcionamento.....	21
Figura 7 – Idealização do Circuito no TinkerCad.....	22
Figura 8 – Projeção 3D da Estrutura no TinkerCad.....	23
Figura 9 – Programação em Blocos do Aplicativo no App Inventor.....	24
Figura 10 – Interface do Aplicativo	25
Figura 11 – Logo	26
Figura 12 – Lixeira.....	27
Figura 13 – Circuito no Interior da Lixeira	27
Figura 14 – Cronograma	28
Figura 15 – Tabela de Custos	29

LISTA DE SIGLAS

VCC – *Voltage Common Collector*; Tensão em corrente contínua.

GND – *Ground*; Terra.

DOUT – *Digital Output*; Saída digital.

SCK – *Serial Clock*; Relógio serial.

LED - *Light-Emitting Diode*; Diodo Emissor de Luz.

ID – *Identity*; Identidade.

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivos	12
1.1.1 Objetivo Geral	12
1.1.2 Objetivos Específicos	13
1.2 Estrutura do Trabalho	13
2 MATERIAIS E MÉTODOS.....	14
2.1 Módulo Wi-fi ESP8266 NodeMCU	14
2.2 Sensor Ultrassônico HC-SR04	14
2.2.1 Funcionamento	15
2.3 Célula de Carga com Módulo Hx711	16
2.4 Resistor Dependente de Luz (LDR)	17
2.5 ThingSpeak.....	18
2.6 MIT App Inventor	18
3 METODOLOGIA.....	20
3.1 Levantamento de Dados.....	20
3.1.1 Entrevista	20
3.1.2 Comunicação e Funcionamento.....	20
3.2 Etapas do Projeto	22
3.2.1 Projeção Inicial do Circuito no TinkerCad	22
3.2.2 Projeção Inicial da Estrutura no TinkerCad	23
3.2.3 Desenvolvimento do Aplicativo	23
3.2.4 Desenvolvimento do Nome e da Logo	25
3.2.5 Configuração do ThingSpeak.....	26
3.2.6 Montagem	26
3.3 Custos e Cronograma.....	28
4 CONCLUSÃO.....	30

REFERÊNCIAS.....	31
ANEXOS	32
ANEXO A – Código do Circuito no Tinkercad.....	33
ANEXO B – Código do Circuito no Arduino	36

1 INTRODUÇÃO

A Internet das Coisas (IoT) pode ser descrita como uma grande rede de objetos físicos que incorporam sensores, softwares e tecnologias semelhantes a fim de transmutar dados entre dispositivos e sistemas pela internet. Esses dispositivos tendem a variar de objetos domésticos a industriais dos mais sofisticados [ORACLE 2022].

Na última década, a IoT teve grande crescimento, tornando-se umas das tecnologias mais importantes do atual século. Podendo-se conectar utensílios do cotidiano à Internet através de dispositivos incorporados, nota-se uma comunicação praticamente perfeita entre pessoas, processos e outros elementos [ORACLE 2022].

Visando isso, decidiu-se desenvolver um projeto que constituísse a IoT devido ao seu explosivo crescimento e aplicação. Conseguir informações em tempo real é de extrema valia para praticamente qualquer produto, sobretudo do cotidiano, sendo neste caso uma lixeira.

Para a construção do dispositivo foi necessário escolher seus componentes, desenvolver o hardware com os sensores necessários para a coleta de dados, elaborar o aplicativo, fazer a comunicação via wi-fi pela nuvem e, por fim, montar o projeto em uma lixeira.

Inicialmente foi realizado o levantamento de dados, obtendo informações em cursos e com profissionais da limpeza. Em seguida, foram selecionados os componentes para o desenvolvimento, como o microcontrolador e os sensores. E por último, um protótipo do dispositivo foi construído.

Ao final, o dispositivo desenvolvido, nomeado BinIoT, atingiu seu objetivo ao tornar-se pronto para a apresentação. A lixeira cumpriu todas as funções que se propôs.

1.1 Objetivos

A seguir, apresentam-se os objetivos desse trabalho.

1.1.1 Objetivo Geral

Utilizar conceitos de Eletrônica e Internet das coisas para criar um sistema simples de monitoramento em tempo real de lixeiras em estabelecimentos, ao fornecer dados da altura interna do lixo, peso e quantidade de vezes que foi aberta.

1.1.2 Objetivos Específicos

1. Avaliação do desempenho da equipe;
2. Criação dos softwares;
3. Criação do protótipo e validação do sistema.

1.2 Estrutura do Trabalho

Este primeiro capítulo introduziu o trabalho desenvolvido. O capítulo 2 apresenta os materiais e métodos para o desenvolvimento dele. Em seguida, a metodologia utilizada para a realização do projeto, desde a etapa de levantamento de dados até a montagem do dispositivo, descrita no capítulo 3. Por fim, o capítulo 5 apresenta as considerações finais recapitulando todo o percurso.

2 MATERIAIS E MÉTODOS

O capítulo atual explica os softwares e hardwares utilizados de forma descritiva, além da definição dos métodos aplicados escolhidos para a confecção do protótipo e seu contexto.

2.1 Módulo Wi-fi ESP8266 NodeMCU

O Módulo Wi-fi ESP8266 NodeMCU é uma plataforma de código aberto que se combina ao chip ESP8266 – um microcontrolador desenvolvido pela Expressif. Pode-se programar usando o software livre LUA ou a plataforma do Arduino e se comunicar via cabo micro USB. Também possui uma antena embutida, 11 pinos de I/O e conversor analógico-digital [FILIPEFLOP 2017].

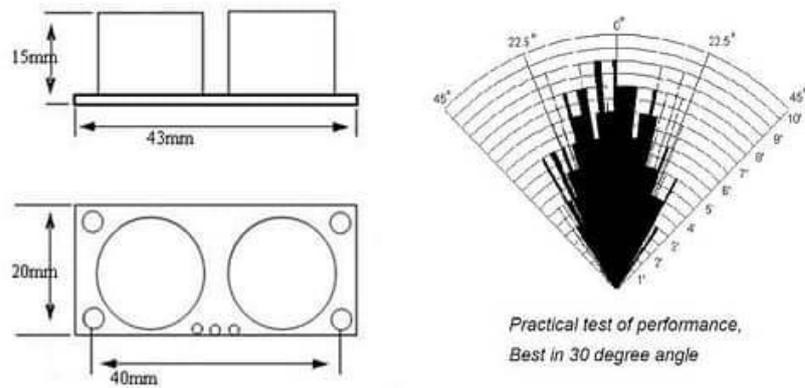
O módulo utilizado no projeto vem com o ESP-12F, que mantém total compatibilidade com o ESP-12E e tem um aumento no alcance do sinal wireless entre 30 e 50% [FILIPEFLOP 2017].

Dessa forma, a escolha desse dispositivo foi conveniente devido a sua disponibilidade de ferramentas, seu fácil e simples uso e um custo-benefício proveitoso para este projeto.

2.2 Sensor Ultrassônico HC-SR04

O Sensor de Distância Ultrassônico HC-SR04 consegue medir distâncias de 2 centímetros até 4 metros mantendo uma ótima precisão e um preço baixo. Este possui um circuito pronto com emissor e receptor, além de quatro pinos de medição para VCC, Trigger (do inglês, gatilho. Pino usado para gatilho de pulsos sonoros), ECHO (do inglês, eco. Pino usado para recepção dos pulsos sonoros) e GND [FILIPEFLOP 2017].

Figura 1 – Dimensões do Sensor Ultrassônico



Fonte: FilipeFlop, 2017.

2.2.1 Funcionamento

Para iniciar o funcionamento é necessário alimentá-lo e colocar o pino Trigger em nível alto por mais de 10 microssegundos. Destarte, o sensor emitirá uma onda sonora que diante de um obstáculo, ela refletirá de volta em direção ao módulo [FILIPEFLOP 2017].

Durante o tempo de emissão e recepção do sinal, o pino ECHO passará a ficar em nível alto. Portanto, o cálculo da distância pode ser feito de acordo com o tempo em que o pino ECHO permaneceu em nível alto após o pino Trigger ter sido colocado em nível alto, a qual pode ser visualizado na fórmula a seguir [FILIPEFLOP 2017].

$$Distância = [Tempo ECHO em nível alto * Velocidade do Som] / 2$$

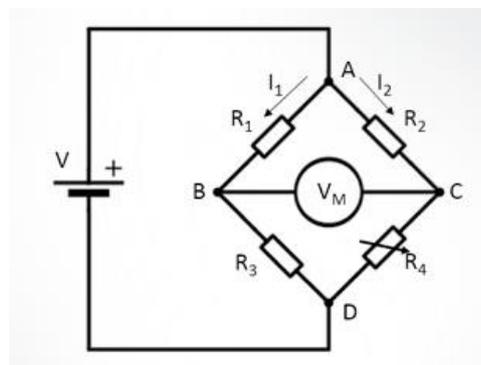
A velocidade do som é de aproximadamente 340 metros por segundo, portanto o resultado é dado em metros se considerado o tempo em segundos. Na fórmula, a divisão por 2 se deve ao fato de que a onda é lançada e rebatida, ou seja, ela percorre duas vezes a distância buscada [FILIPEFLOP 2017].

Baseando-se nessas informações, para obter ciência da altura dos resíduos, a escolha do ultrassônico calhou bem, pois a sua função é suficientemente eficaz para cumprir esta necessidade.

2.3 Célula de Carga com Módulo Hx711

A Célula de carga nada mais é que um sensor de força. São capazes de medir deformações mecânicas, ou seja, à medida que o corpo sofre uma deformação a resistência fixada no corpo dessa célula é alterada. Com a alteração da resistência é possível determinar a deformação causada pela força aplicada, essa resistência é um circuito ponte de Wheatstone [BAU 2018].

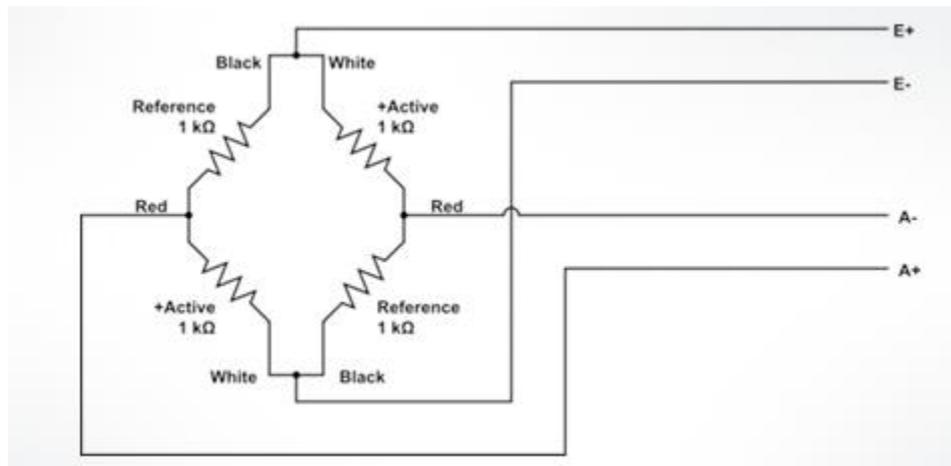
Figura 2 – Ponte de Wheatstone



Fonte: Baú da Eletrônica, 2018.

O módulo conversor HX711 é um conversor analógico digital de 24 bits que também funciona como um amplificador, assim mantendo o nível de sinal dentro da faixa de operação do conversor. Sua pinagem baseia-se em 2 pinos (E) para alimentar a célula, 2 pinos (A) para entradas analógicas da célula, 2 pinos (Vcc, GND) para alimentação do módulo, 1 pino (DOUT) para saída digital e 1 pino (SCK) para entrada digital - que se conectam ao NodeMCU [BAU 2018].

Figura 3 – Ligação de Duas Células de Carga



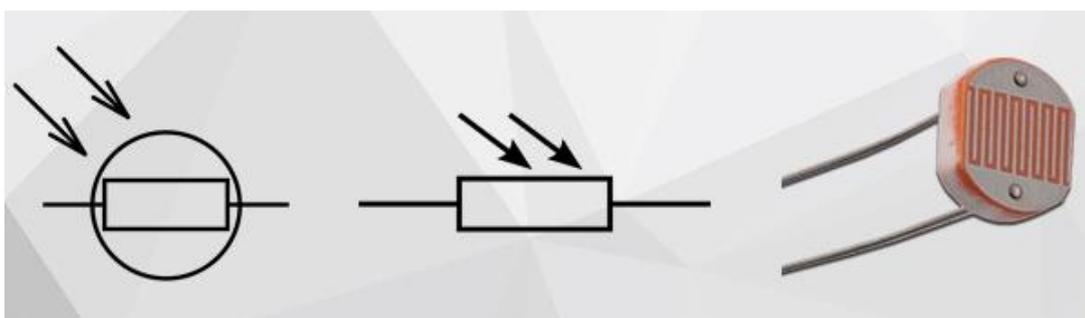
Fonte: Baú da Eletrônica, 2018.

Decidiu-se usar uma célula de carga do tipo plataforma, baseada em deformação mecânica, com capacidade máxima de 50 quilogramas por se adaptar melhor à estrutura e ao propósito do protótipo.

2.4 Resistor Dependente de Luz (LDR)

LDR é a sigla em inglês para *Light Dependent Resistor*, que traduzindo significa resistor dependente de luz, ou também fotoresistor, é um tipo de resistor capaz de gerar uma variação de resistência sobre si em função de uma intensidade de luz lançada na sua direção. Os fotoresistores mais comuns possuem maior sensibilidade à luz visível, porém existem aqueles cuja sensibilidade é dada em relação à luz infravermelha. O LDR é um componente que possui apenas dois terminais e não tem polaridade definida [ALVES 2022].

Figura 4 – Componente e Simbologias do LDR



Fonte: Manual da Eletrônica, 2022.

Baseado nisso, é perceptível que seu funcionamento seja consideravelmente simples, de maneira que no momento no qual as partículas de luz (fótons) incidem sobre a superfície do sensor, os elétrons do material semiconductor são liberados, tornando a condutividade do LDR maior e diminuindo sua resistência [ALVES 2022].

O LDR foi usado no projeto como uma forma de identificar quantas vezes a lixeira foi utilizada, baseando-se no momento em que ela for aberta, assim, gerando uma iluminação para a leitura do fotoresistor.

2.5 ThingSpeak

ThingSpeak é um serviço de plataforma de análise de IoT que permite agregar, visualizar e analisar fluxos de dados ao vivo na nuvem. Ele fornece visualizações instantâneas de dados postados por seus dispositivos. Nele, é possível realizar análise e processamento online dos dados à medida que eles chegam. O ThingSpeak é frequentemente usado para prototipagem e prova de conceito de sistemas IoT que exigem análise [THE MATHWORKS 2022].

Há muitos serviços semelhantes ao ThingSpeak, porém durante o processo de pesquisa, encontrou-se uma plataforma de cursos oferecida pela Samsung e pelo Laboratório de Sistemas Integráveis Tecnológico chamada CodeloT, a qual tem um curso especificamente sobre esse tópico utilizando o ThingSpeak. Logo, optou-se por seguir as instruções dele.

2.6 MIT App Inventor

O MIT App Inventor é um ambiente de programação visual intuitivo que permite até mesmo crianças criarem aplicativos totalmente funcionais para Android, iPhones e tablets Android/iOS. Mesmo sendo iniciante, é possível ter um primeiro aplicativo simples instalado e funcionando em menos de 30 minutos. Além disso, pela interface ser baseada em blocos, ela facilita a criação de aplicativos complexos e de alto impacto em um tempo significativamente menor do que em ambientes de programação tradicionais. O projeto tem o objetivo de democratizar o desenvolvimento de software, capacitando todas as pessoas a passar do consumo de tecnologia para a criação de tecnologia [MIT 2022].

Como foi falado no assunto anterior, o mesmo se aplica aqui. Pela plataforma de cursos CodeLoT, o curso escolhido igualmente orienta usar o App Inventor para a confecção do aplicativo, contendo ainda um curso exclusivamente para ensinar a como programar aplicativos nele.

3 METODOLOGIA

Este capítulo apresenta todas as etapas para o desenvolvimento da lixeira.

3.1 Levantamento de Dados

Para a construção desta monografia acadêmica foi empregado o tipo exploratório de pesquisa e o método bibliográfico, a qual permitiu adquirir informações relacionados aos sistemas de coleta pública em cidades inteligentes, aprofundamento da aplicação de sensores a IoT e programação no NodeMCU. As técnicas de análise escolhidas foram a documentação direta de uma entrevista e professores e a documentação indireta em sites como YouTube e CodeloT.

3.1.1 Entrevista

Foi feita uma pesquisa com profissionais da limpeza, na qual registramos uma entrevista, em que perguntamos como funciona o sistema de coleta na escola, quem retira os lixos e de quanto em quanto tempo e se a possibilidade de monitorar e ser notificada, quando a lixeira encher, fosse útil.

Os lixos são separados pelos próprios alunos com as lixeiras recicláveis e, também, tem lixos espalhados pelos ambientes da escola. [...] Eu os retiro todos os dias, praticamente toda hora. [...] Com certeza! Seria ótimo e facilitaria bastante.

E baseado nisso, assegurou-se o valor que o projeto pode alcançar na sociedade.

3.1.2 Comunicação e Funcionamento

A comunicação foi planejada com suporte dos ensinamentos do curso da CodeloT, começando pelo NodeMCU mandando a informação, em seguida o ThingSpeak recebe e o aplicativo a busca.

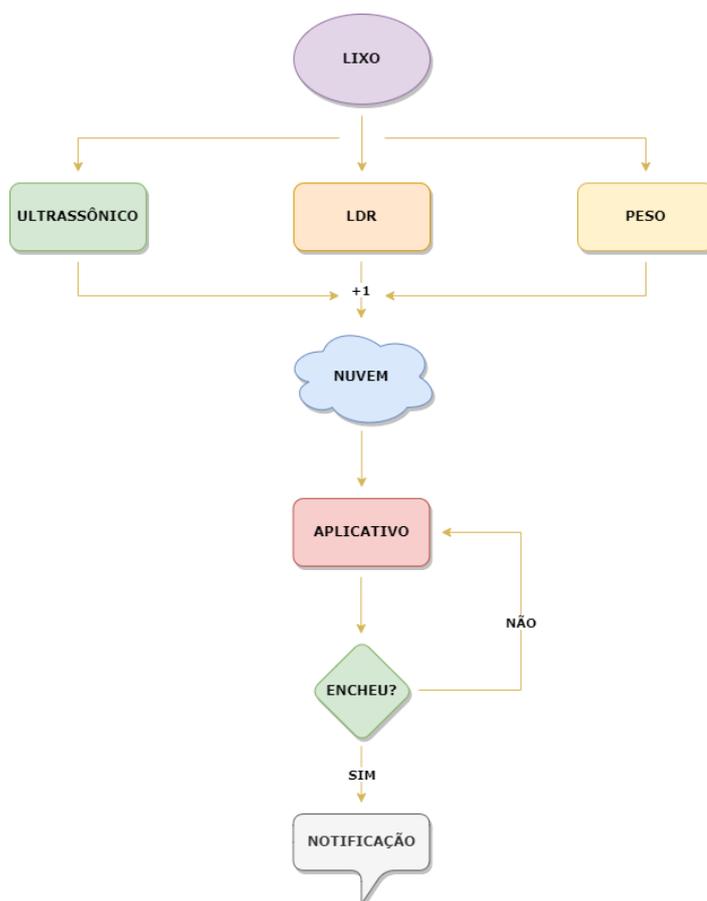
Figura 5 – Fluxograma da Comunicação via Internet



Fonte: Elaboração Própria, 2022.

A sistematização do funcionamento para criar a programação foi elaborada com a ajuda de um professor, que pode ser vista no fluxograma da Figura 6, para fundamentar a lógica a ser seguida.

Figura 6 – Fluxograma do Funcionamento



Fonte: Elaboração Própria, 2022.

3.2 Etapas do Projeto

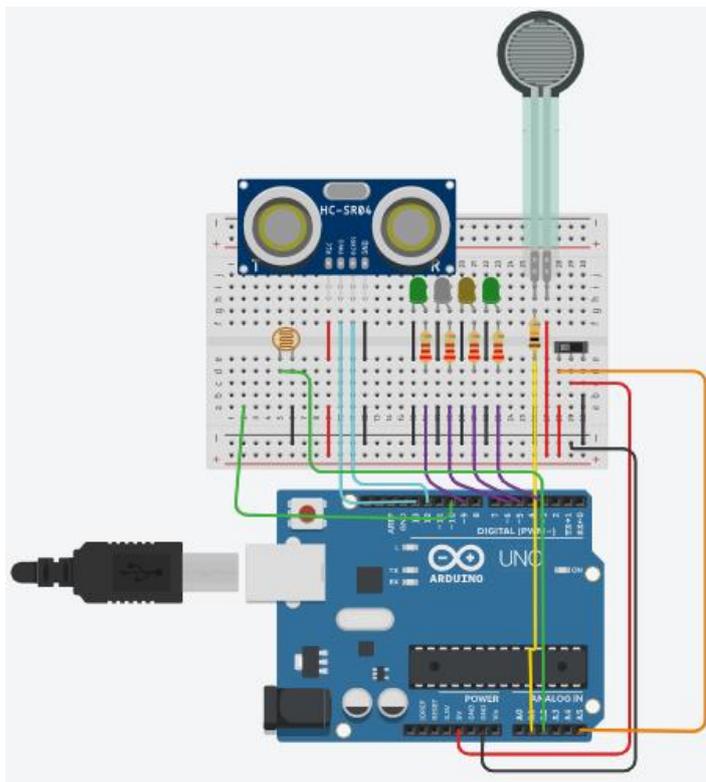
O desenvolvimento do protótipo pode ser dividido em etapas apresentadas na lista abaixo.

1. Projeção Inicial do Circuito no TinkerCad
2. Projeção Inicial da Estrutura no TinkerCad
3. Desenvolvimento do Aplicativo
4. Configuração do ThingSpeak
5. Aperfeiçoamento
6. Montagem

3.2.1 Projeção Inicial do Circuito no TinkerCad

De início, o projeto foi construído virtualmente pela plataforma do TinkerCad, para garantir quais componentes seriam realmente necessários. Os LEDs eram apenas para teste e o interruptor foi descontinuado.

Figura 7 – Idealização do Circuito no TinkerCad



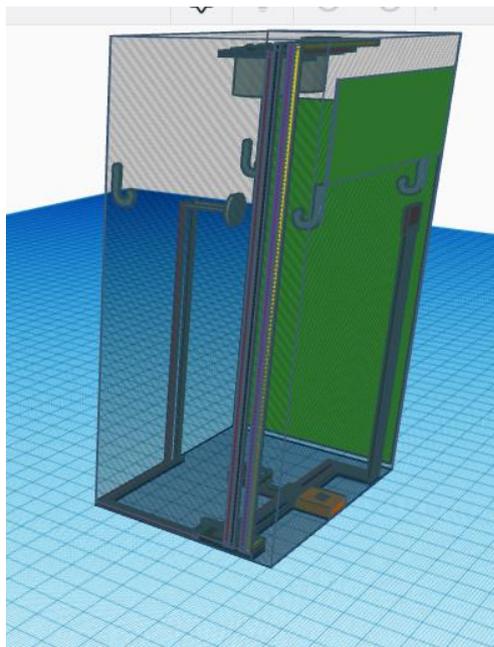
Fonte: Elaboração Própria, 2022.

3.2.2 Projeção Inicial da Estrutura no TinkerCad

Posteriormente, ao confirmar os componentes, iniciou-se o desenvolvimento primordial da estrutura, como o tamanho e a posição mais proveitosa de cada dispositivo.

A lixeira utilizada para o protótipo não conteve garras – que tinham a função de segurar a sacola de lixo – e a tampa é no topo, não na lateral. Também, como pode ser visualizado na Figura 8, o retângulo verde, na lateral direita, era uma porta destinada para o retiro do lixo. Porém, ele foi descontinuado da mesma forma que as garras.

Figura 8 – Projeção 3D da Estrutura no TinkerCad



Fonte: Elaboração Própria, 2022.

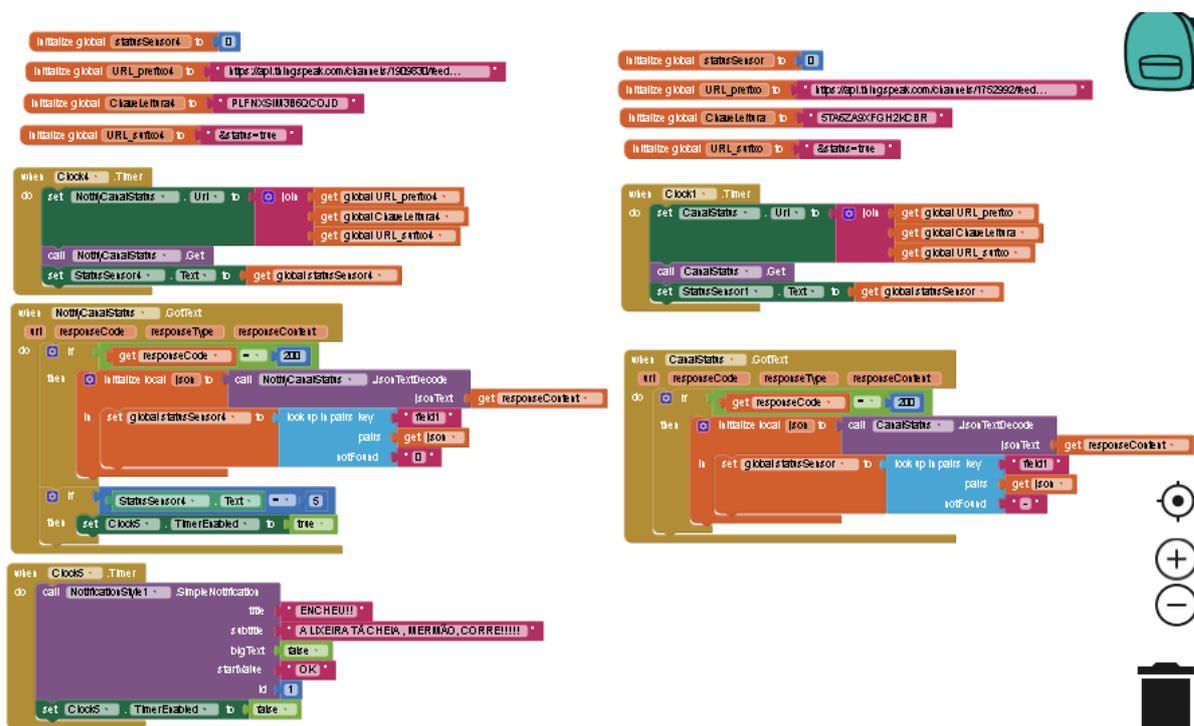
3.2.3 Desenvolvimento do Aplicativo

O aplicativo foi elaborado sincronamente com o circuito, a estrutura e a logo – que inspirou as cores e será abordada no próximo subtópico. A programação foi desenvolvida baseada no curso do CodeloT, na qual busca a informação na nuvem e configura a notificação.

Primeiramente, ela cria um link do ThingSpeak, acessa e lê o valor contido, repetindo isso para os 4 sensores e a notificação – que também tem o código que define a mensagem que aparece.

Na Figura 9, no canto direito é o código da notificação e na direita o do sensor ultrassônico, que se repete para os demais.

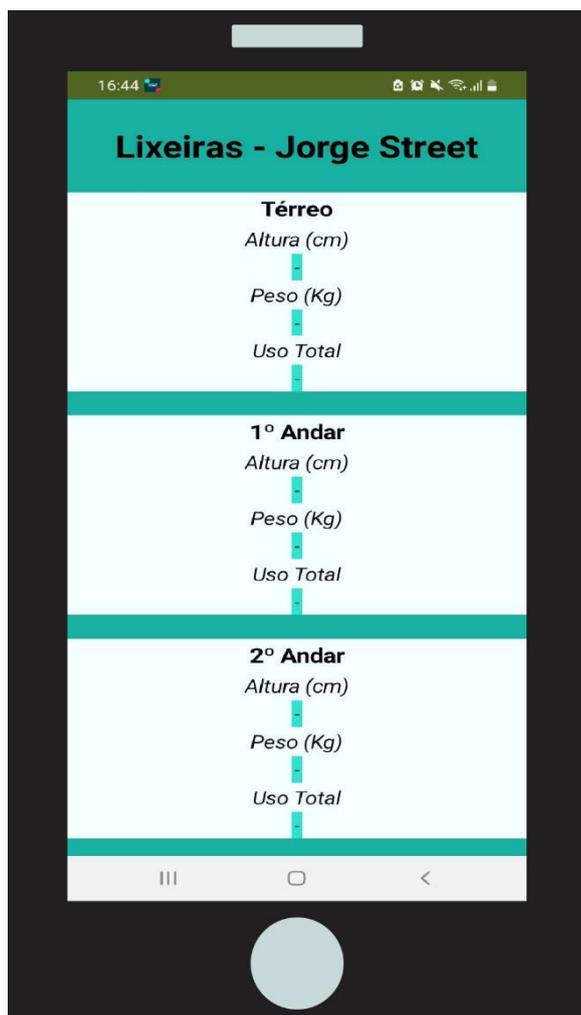
Figura 9 – Programação em Blocos do Aplicativo no App Inventor



Fonte: Elaboração Própria, 2022.

E a interface pode ser vista na Figura 10.

Figura 10 – Interface do Aplicativo



Fonte: Elaboração Própria, 2022.

3.2.4 Desenvolvimento do Nome e da Logo

O nome se deu pela junção da palavra *bin*, do inglês, que tem o significado de lixeira e a sigla IoT, também do inglês, que significa Internet das Coisas. Esse nome representa muito bem o que o projeto se trata.

Na figura 11, é possível ver a logo em seu estado final.

Figura 11 – Logo



Fonte: Elaboração Própria, 2022.

3.2.5 Configuração do ThingSpeak

Após criar a conta e os canais responsáveis por cada sensor, para a comunicação do NodeMCU e do aplicativo, as seguintes informações foram necessárias: Chaves Escritas e de Leituras, IDs dos canais e os links que contêm os valores.

3.2.6 Montagem

Durante a montagem, foi efetuado diversos testes – nos quais modificamos o código, e pode ser visto o código finalizado no Anexo B – e além das peças principais, foram necessários uma placa de ensaio, uma placa, para servir como plataforma para a célula de carga, e um bloco pequeno de madeira.

Figura 12 – Lixeira



Fonte: Elaboração Própria, 2022.

Figura 13 – Circuito no Interior da Lixeira

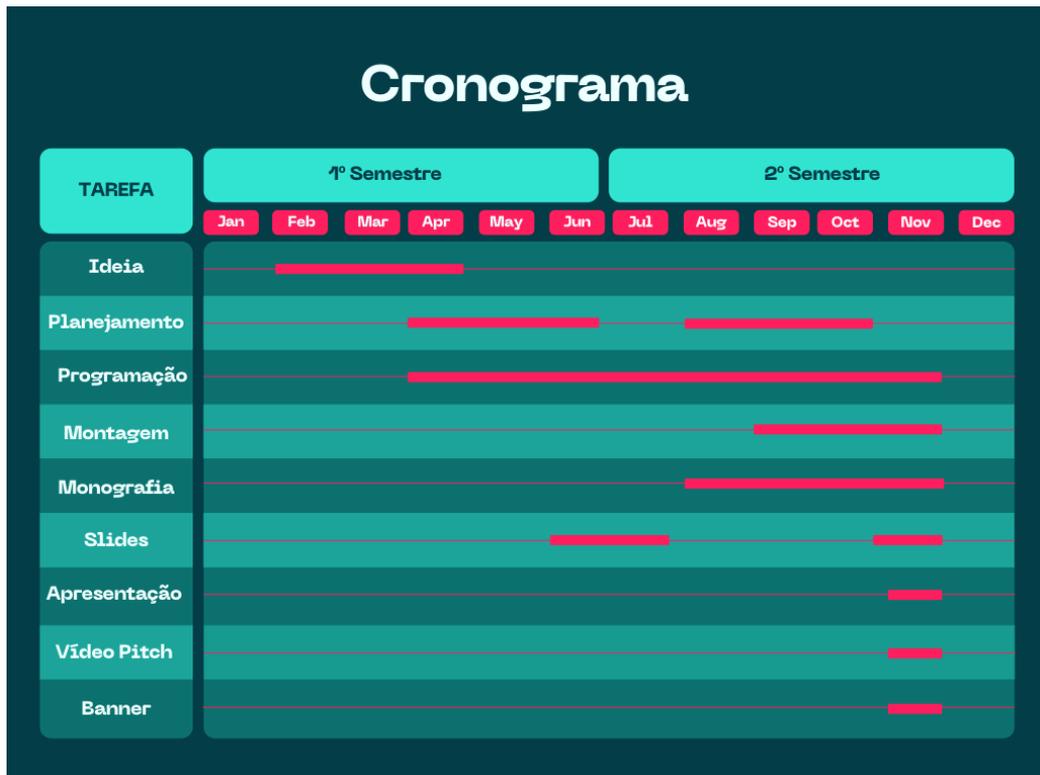


Fonte: Elaboração Própria, 2022.

3.3 Custos e Cronograma

O primeiro semestre foi utilizado principalmente para o planejamento do projeto, mas também houve progresso no desenvolvimento, exceto na montagem e na monografia.

Figura 14 – Cronograma



Fonte: Elaboração Própria, 2022.

Em relação aos custos, mesmo o projeto contendo 4 integrantes, o preço para cada um foi bem acessível, o que era esperado, levando em conta que todos os dispositivos não são de alto valor.

Figura 15 – Tabela de Custos



Componente	Valor (R\$)
Ultrassônico	14,16
Célula de Carga	12,66
Hx711	2,42
LDR	0,48
NodeMCU	26,91
Protoboard	15,90
Fios	12,26
Lixeira	22
Total	106,79
Por Integrante	26,70

Fonte: Elaboração Própria, 2022.

4 CONCLUSÃO

O objetivo geral deste projeto foi concluído com o desenvolvimento de um sistema, para lixeiras, que permite saber a altura, peso e quantidade de vezes que foi aberta a lixeira, utilizando os conhecimentos adquiridos, previamente e posteriormente, de Eletrônica e IoT.

O primeiro objetivo específico foi contemplado pois todos os integrantes participaram ativamente e pertinentemente. Cada um teve suas tarefas isoladas e conjuntas. Todos foram cruciais para a realização deste trabalho.

Outra finalidade alcançada foi o desenvolvimento da programação do microcontrolador e do aplicativo, a configuração da nuvem e a comunicação entre todos eles.

O último objetivo também foi realizado, a montagem do protótipo foi concluída, operando satisfatoriamente.

Portanto, vê-se que a apresentação pode ser feita e alcançar bons resultados.

Com relação a aplicações futuras, este projeto é apenas um trabalho de conclusão de curso, caso nenhuma empresa tenha interesse, ele não terá mais usabilidade.

REFERÊNCIAS

ORACLE. **O que é Internet of Things (IoT)?** Disponível em: <<https://www.oracle.com/br/internet-of-things/what-is-iot/>>. Acesso em: 19 set. 2022, 2022. Citado 2 vezes na página 13.

FILIFELOP. **Módulo WiFi ESP8266 NodeMcu ESP-12.** Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/>>. Acesso em: 29 set. 2022, 2017. Citado 2 vezes na página 13.

FILIFELOP. **Sensor de Distância Ultrassônico HC-SR04.** Disponível em: <<https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>>. Acesso em: 03 out. 2022, 2017. Citado 4 vezes na página 14 e 15.

BAU. **Conversor HX711: Balança Digital com Sensor de Peso e Arduino Uno.** Disponível em: <<https://blog.baudaeletronica.com.br/conversor-hx711-para-balanca-eletronica/>>. Acesso em: 10 out. 2022, 2018. Citado 2 vezes na página 15.

ALVES, P. **LDR – O que é e como funciona! - Manual da Eletrônica.** Disponível em: <<https://www.manualdaeletronica.com.br/ldr-o-que-e-como-funciona/>>. Acesso em: 10 out. 2022, 2022. Citado 2 vezes na página 16 e 17.

THE MATHWORKS. **Learn More - ThingSpeak IoT.** Disponível em: <https://thingspeak.com/pages/learn_more>. Acesso em: 10 out. 2022, 2022. Citado 1 vez na página 17.

MIT. **About Us.** Disponível em: <<https://appinventor.mit.edu/about-us>>. Acesso em: 10 out. 2022, 2022. Citado na página 18.

ANEXOS

ANEXO A – Código do Circuito no Tinkercad

```
int Peso = 0;
int Luz = 0;
int trig = 13;
int echo = 12;
float distancia;
bool whatever = 0;
unsigned long abre = 0;

void setup()
{
  pinMode(9, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  Serial.begin(8600);
  pinMode(A2, INPUT_PULLUP);
}

void loop()
{
  if(Peso > 512)
    digitalWrite(9, HIGH);

  else(digitalWrite(9, LOW));
  Peso = analogRead(A1);

  if(Luz > 512)
    digitalWrite(3, HIGH);

  else(digitalWrite(3, LOW));
  Luz = analogRead(A2);
```

```
digitalWrite(trig, LOW);  
delay(0005);  
digitalWrite(trig, HIGH);  
delay(0010);  
digitalWrite(trig, LOW);
```

```
distancia = pulseIn (echo, HIGH);  
distancia = distancia/58;
```

```
if (distancia > 85)  
digitalWrite(5, HIGH);
```

```
else (digitalWrite(5, LOW));
```

```
if((Peso > 512) && (Luz > 512) && (distancia > 85))  
digitalWrite(6, HIGH);
```

```
else(digitalWrite(6, LOW));
```

```
if(digitalRead(A2)) whatever = 1;
```

```
if(!digitalRead(A2) && whatever)
```

```
{  
  whatever = 0;  
  abre += 1;  
}
```

```
Serial.print ("Peso: ");
```

```
Serial.println (Peso);
```

```
Serial.print ("Luminosidade: ");
```

```
Serial.println (Luz);
```

```
Serial.print ("Distancia: ");
```

```
Serial.println (distancia);
```

```
Serial.print ("abreu: ");
```

```
Serial.println (abre);  
delay(1000);  
}
```

ANEXO B – Código do Circuito no Arduino

```
int Luz = 0;
int trig = 16; //D0
int echo = 05; //D1
float distancia;
bool whatever = 0;
unsigned long abre = 0;
int notificacao = 0;

#include <ThingSpeak.h>
#include <ESP8266WiFi.h>
#include <HX711.h>

const int PINO_DT = 00; // D3
const int PINO_SCK = 02; // D4
const int TEMPO_ESPERA = 1000; //declaracao da variavel de espera

HX711 escala; //declaracao do objeto escala na classe HX711 da biblioteca

float fator_calibracao = -45000; //pre-definicao da variavel de calibracao

char comando; //declaracao da variavel que ira receber os comandos para alterar o
fator de calibracao

char ssid[] = "RelampagoMarquinhos"; // Nome da rede SSID
char pass[] = "katiaauw"; // Nome da senha

int status = WL_IDLE_STATUS;
WiFiClient client;
int sensorValueUt; // variável que armazena o valor lido pelo sensor ultrassônico
unsigned long canalUt = 1752992;
const char * ChaveEscritaAPIUt = "U0R3ID62UXR7KZC4";
```

```

int sensorValueOp; // variável que armazena o valor lido pelo sensor óptico
unsigned long canalOp = 1752993;
const char * ChaveEscritaAPIOp = "UQMFYLYW9CXRPUAO";

int sensorValueC; // variável que armazena o valor lido pelo sensor de carga
unsigned long canalC = 1752991;
const char * ChaveEscritaAPIc = "DRCDYHX2WWMXMKT1";

int sensorValueN; // variável que armazena o valor lido pela notificação do
ultrassônico
unsigned long canalN = 1909830;
const char * ChaveEscritaAPIN = "ID8JPBW63RWB23OZ";

void setup()
{
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  Serial.begin(115200);
  pinMode(04, INPUT_PULLUP);

  WiFi.begin(ssid, pass);
  ThingSpeak.begin(client);

  escala.begin (PINO_DT, PINO_SCK);
  float media_leitura = escala.read_average();
  escala.tare(); //zera a escala
}

void loop()
{
  Luz = analogRead(04); //D2
  digitalWrite(trig, LOW);
  delay(0005);
  digitalWrite(trig, HIGH);

```

```
delay(0010);
digitalWrite(trig, LOW);

distancia = pulseIn (echo, HIGH);
distancia = distancia/58;
// Contador
if(digitalRead(04)) whatever = 1;
if(!digitalRead(04) && whatever)
{
  whatever = 0;
  abre += 1;
}

// Célula de Carga
escala.set_scale(fator_calibracao); //ajusta a escala para o fator de calibracao

//verifica se o modulo esta pronto para realizar leituras
if (escala.is_ready())
{
  //mensagens de leitura no monitor serial
  Serial.print("Leitura: ");
  Serial.print(escala.get_units(), 1); //retorna a leitura da variavel escala com a
  unidade quilogramas
  Serial.print(" kg");
  Serial.print(" \t Fator de Calibracao = ");
  Serial.print(fator_calibracao);
  Serial.println();

  //alteracao do fator de calibracao
  if(Serial.available())
  {
    comando = Serial.read();
    switch (comando)
    {
```

```
    case 'x':
        fator_calibracao = fator_calibracao - 100;
        break;
    case 'c':
        fator_calibracao = fator_calibracao + 100;
        break;
    case 'v':
        fator_calibracao = fator_calibracao - 10;
        break;
    case 'b':
        fator_calibracao = fator_calibracao + 10;
        break;
    case 'n':
        fator_calibracao = fator_calibracao - 1;
        break;
    case 'm':
        fator_calibracao = fator_calibracao + 1;
        break;
    }
}
}
else
{
    Serial.print("HX-711 ocupado");
}
delay(TEMPO_ESPERA);

// Notificacoes
if (distancia < 10 && distancia != 0)
{
    if (notificacao > 6)
    {
        notificacao = 0;
    }
}
```

```
    notificacao += 1;
    delay(1000);
}
else
    notificacao = 0;

// Teste
Serial.print ("Luminosidade: ");
Serial.println (Luz);
Serial.print ("Distancia: ");
Serial.println (distancia);
Serial.print ("abreu: ");
Serial.println (abre);
Serial.print ("Notificação: ");
Serial.println (notificacao);
delay(1000);

sensorValueUt = distancia; // Sensor Ultrassônico
ThingSpeak.writeField(canalUt, 1, sensorValueUt, ChaveEscritaAPIUt); // Sensor
Ultrassônico

sensorValueOp = abre; // Sensor Óptico
ThingSpeak.writeField(canalOp, 1, sensorValueOp, ChaveEscritaAPIOp); // Sensor
Óptico

sensorValuedC = escala.get_units(); // Sensor de Carga
ThingSpeak.writeField(canalDC, 1, sensorValuedC, ChaveEscritaAPIIdC); // Sensor
de Carga

sensorValueN = notificacao; // Notificação do Ultrassônico
ThingSpeak.writeField(canalN, 1, sensorValueN, ChaveEscritaAPIN); // Notificação

delay(1000); // ThingSpeak aceita atualizações dos valores cada 5 segundos.
}
```