



# DESENVOLVIMENTO DE UM COMEDOURO AUTOMATIZADO PARA CÃES UTILIZANDO TECNOLOGIAS DE IOT

## *DEVELOPMENT OF AN AUTOMATIC DOG FEEDER USING IOT TECHNOLOGIES*

Marcos Cesar dos Santos<sup>1</sup> 

Evandro Cantú<sup>2</sup> 

Humberto M. Beneduzzi<sup>3</sup> 

**Resumo:** Manter uma dieta saudável e regular dos animais de estimação pode ser um desafio, especialmente para indivíduos que passam períodos prolongados longe de casa. Dispositivos conhecidos como comedouros permitem a alimentação remota e na medida adequada para os animais, mas apresentam um custo elevado e, por muitas vezes, pouca ou nenhuma conectividade e automação. Assim, este projeto propôs uma solução baseada em IoT (Internet das Coisas) de baixo custo para a alimentação de cães, que pode ser controlada tanto de forma automática quanto remota. A solução proposta aplicou o conceito de casa conectada, usando a central de automação de código aberto para casas inteligentes “Home Assistant”, o ambiente de baixa programação “Node-RED”, o protocolo de troca de mensagens “MQTT” e hardware composto por microcontrolador, servo motor, sensores e impressão 3D. O algoritmo foi criado para permitir a troca de informação via protocolo MQTT entre o microcontrolador e o servidor, bem como o acionamento do servo motor e a leitura dos dados de temperatura e umidade do sensor. As automações e interfaces foram implementadas no Node-RED e Home Assistant para controle e apresentação dos dados. A estrutura física do comedouro foi impressa em 3D e montada no final, acomodando ainda o circuito eletrônico desenvolvido. O protótipo construído foi testado e observado, evidenciando-se o acionamento do comedouro em horários agendados e através dos acionamentos remotos, bem como a integração com o sistema de automação residencial. Por fim, os custos de implementação foram calculados e apresentados, assim como as possibilidades de melhorias para trabalhos futuros.

**Palavras-chave:** Comedouro Inteligente. Internet das Coisas. Home Assistant. MQTT. Casa Conectada.

**Abstract:** Maintaining a healthy, regular pet diet can be challenging, especially for individuals who spend extended periods away from home. Devices known as feeders allow remote and adequate feeding for the animals, but have a high cost and, often, little or no connectivity and automation. Thus, this project proposed a low-cost IoT (Internet of Things) based solution for dog feeding, which can be controlled both automatically and remotely. The proposed solution applied the concept of connected home, using the open-source automation center for smart homes Home Assistant, the low programming environment Node-RED, the message exchange protocol MQTT and hardware composed of microcontroller, servo motor, sensors and 3D printing. The algorithm was created to allow the exchange of information via the MQTT protocol between the microcontroller and the server, as well as the activation of the servo motor and the reading of temperature and humidity data from the sensor. Automations and interfaces were implemented in Node-RED and Home Assistant to control and present data. The physical structure of the feeder was printed in 3D and assembled at the end, also accommodating the developed electronic circuit. The built prototype was tested and observed, showing the activation of the feeder at scheduled times and through remote activations, as well as the integration with the home automation system. Finally, the implementation costs were calculated and presented, as well as the possibilities for improvements for future work.

**Keywords:** Smart Feeder. Internet of Things. Home Assistant. MQTT. Connected Home.

## INTRODUÇÃO

Manter uma alimentação saudável e regular dos animais de estimação pode ser uma tarefa difícil para muitas pessoas, principalmente para aquelas que chegam a passar horas fora de casa à trabalho ou estudo. Muitas vezes os pets passam o dia todo aguardando pela chegada de seu dono para receber alguns minutos de carinho e finalmente, alimento (KARYONO *et al.*, 2016; SHIDDIEQY *et al.*, 2020).

Com o advento da tecnologia, diversos dispositivos surgiram para auxiliar o cotidiano das pessoas e tornar tarefas rotineiras em aplicações automatizadas, trazendo conforto, economia e segurança. Entre estes dispositivos estão os comedouros para animais, que permitem uma alimentação remota e na medida adequada (SHIDDIEQY *et al.*, 2021).

Embora existam diversas soluções comerciais, nota-se que estes dispositivos ainda possuem um custo elevado, fato esse que pode afastar os donos de animais de estimação da compra dessas soluções.

Assim, diversos autores propuseram soluções que permitissem fornecer alimentação de maneira automática para animais de estimação, como Manoj (2015), que desenvolveu uma solução baseada em um microcontrolador ATMEGA 8, dois motores de corrente contínua e duas válvulas temporizadas com campainha. O funcionamento da solução deu-se através de um firmware implementado em C e gravado no microcontrolador, responsável por determinar os horários de atuação dos motores e válvulas, controlar o fluxo de ração fornecida e sinalizar a conclusão da tarefa através da campainha.

Razali *et al.* (2021), por sua vez, propuseram um comedouro cujo hardware era composto por um microcontrolador ESP32, um servo motor, um sensor ultrassônico, um sensor infravermelho e uma célula de carga. O sensor infravermelho possuía a função de detectar a presença do pet próxima ao comedouro. O sensor ultrassônico verificava a quantidade de ração disponível. A célula de carga conferia a quantidade de ração fornecida, aferindo o peso de uma tigela. Por fim, o servo motor ficava responsável por dispensar a ração na tigela. Todos os elementos eram controlados através de um ESP32. Além disso, os autores incluíram uma análise de Big Data baseada em Predição de

Série Temporal para prever a escassez de ração e informar o dono do animal de estimação sobre a ocorrência da mesma.

De maneira análoga, Liu (2021) apresentou uma solução para alimentação de animais de estimação utilizando um microcontrolador STC89C52. O microcontrolador era conectado a um RTC (*Real Time Clock* – Relógio de Tempo Real) e botões de controle, pelos quais se determinava o horário de fornecimento da ração. Ao atingir o tempo determinado, uma campainha era acionada para chamar a atenção dos animais e um motor de passo era acionado para fornecer a ração.

Embora todas as soluções apresentadas atendam ao propósito de alimentação dos pets com um baixo custo, há um baixo nível ou inexistência de conectividade, o que impede que as soluções possam ser integradas a outros dispositivos e até mesmo aos sistemas de automação residencial.

Dada a ascensão da aplicação de automação nas residências, assim como da quantidade de fabricantes de dispositivos domésticos inteligentes, surge com isso outro problema relacionado à compatibilidade entre as marcas, o que pode limitar os usuários a dispositivos de fabricantes específicos (GRATSCHEFF, 2022) e afastar os dispositivos do tipo DIY (*Do It Yourself* – Faça Você Mesmo).

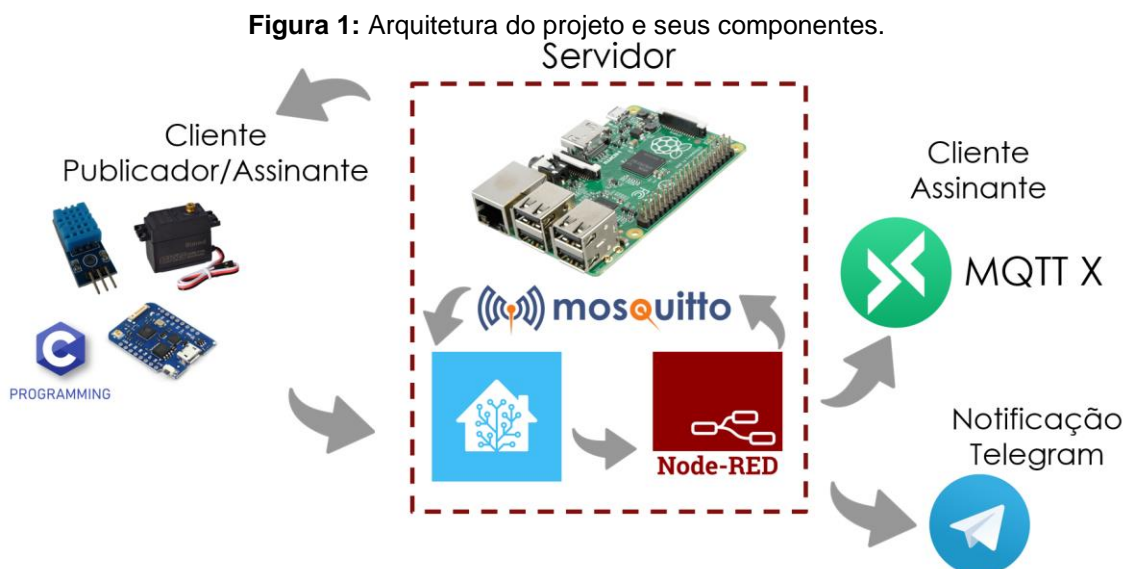
Nesse sentido, esse projeto implementou uma solução para alimentação de cães, baseada em IoT (*Internet of Things* – Internet das Coisas), de baixo custo e que opere tanto de maneira automática quanto remota, sob a aplicação do conceito de casa conectada, utilizando a central de automação de código aberto para casas inteligentes “Home Assistant”, o ambiente de baixa programação (*low code*) “Node-RED”, o protocolo de troca de mensagens “MQTT” (*Message Queue Telemetry Transport* – Transporte de Filas de Mensagem de Telemetria) e hardware composto por microcontrolador, servo motor, sensores e impressão 3D.

# 1. MATERIAIS E MÉTODOS

O desenvolvimento da solução buscou aplicar conceitos de IoT aplicados sob o olhar da domótica, fazendo o uso de tecnologias abertas tanto em hardware quanto em software, a fim de obter a viabilização técnico-econômica do projeto e permitir o controle do acionamento do comedouro de maneira automática e remota.

O cenário construído buscou integrar tanto os elementos de hardware e software essenciais para o funcionamento do comedouro, como também outros dispositivos de automação residencial, aplicando conceitos de IoT e domótica, com intuito de permitir que o usuário gerencie não somente o comedouro criado, mas os possíveis dispositivos de automação presentes na residência, mantendo a viabilização técnico-econômica desejada.

Desta maneira, a arquitetura proposta, representada pela **Figura 1**, apresenta os elementos integrantes da solução, contemplando tanto hardware quanto os sistemas utilizados, cujos detalhes são dissertados nos tópicos seguintes.



**Fonte:** Dos autores.

## 2.1. Servidor

O servidor é o principal componente de todo o sistema. Nele, são executados os principais serviços e sistemas responsáveis por gerenciar a troca de informação entre os dispositivos e o usuário.

Resumidamente, o servidor é composto por um Raspberry Pi 3B executando o sistema operacional Home Assistant, sendo esse o principal elemento responsável pela integração dos sistemas. Por meio dos recursos do Home Assistant, foram instalados o broker MQTT (utilizando o serviço Eclipse Mosquitto) para gerenciamento das mensagens através do protocolo MQTT, e o Node-RED, para implementação das funções de leitura e redirecionamento das mensagens.

### **2.1.1. Raspberry Pi 3B**

O Raspberry Pi é um pequeno computador de baixo custo, cujo tamanho se assemelha ao de um cartão de crédito em que se é possível conectar um monitor de computador ou TV e usar um teclado e mouse padrão. Este pequeno dispositivo permite o fácil aprendizado sobre computação e linguagens de programação. Através do Raspberry Pi, consegue-se realizar qualquer tarefa destinada a um computador convencional, como navegar na internet, assistir vídeos, criar documentos e executar jogos (RASPBERRY PI FOUNDATION, 2022).

Lançado em 2012, várias versões e variações do Raspberry PI foram lançadas desde então. Enquanto o Pi original tinha uma CPU single-core de 700 MHz e apenas 256 MB de RAM, os modelos mais recentes contam com uma CPU quad-core com clock de mais de 1,5 GHz e 4 GB de RAM. (OPENSOURCE, 2022).

O Raspberry Pi 3B (**Figura 2**) é uma dessas variações, sendo também a primeira versão da 3ª geração, substituindo o modelo Raspberry Pi 2 Model B em fevereiro de 2016. O modelo possui processador Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM, placa de rede e Bluetooth Low Energy (BLE) integrados, 40 pinos de entrada e saída (GPIO) para desenvolvimento, 4 portas USB 2.0, conector de áudio e vídeo de 4 polos, saída HDMI, conectores para câmera e display touchscreen, e sistema de armazenamento via cartão Micro SD – onde o sistema operacional é carregado (RASPBERRY PI, 2022).

**Figura 2:** Placa Raspberry Pi 3B

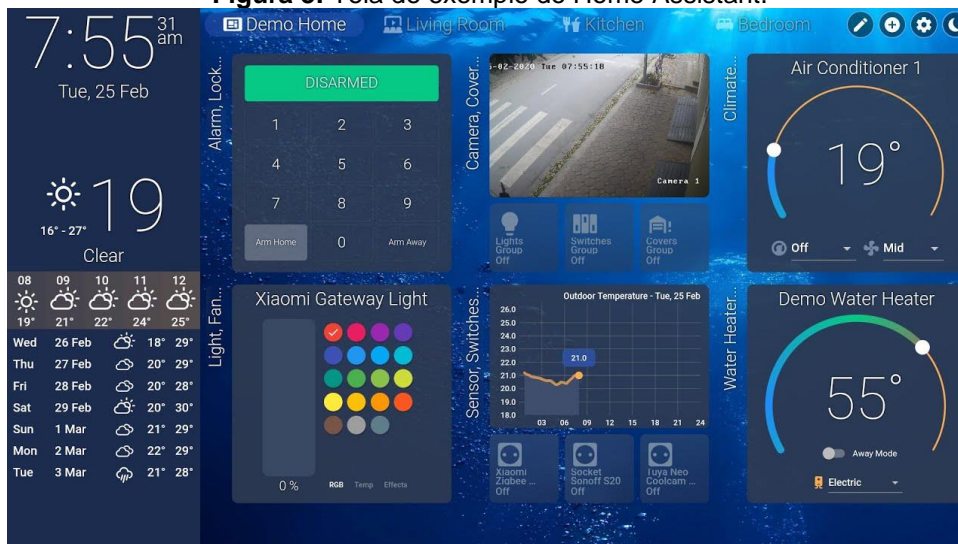


Fonte: Raspberry Pi (2022).

### 2.1.2. Home Assistant

O Home Assistant é uma central de automação de código aberto com foco em controle e privacidade que permite integrar diversas plataformas de automação residencial, gerir serviços e configurar automações. O Home Assistant (**Figura 3**) foi desenvolvido para ser executado em um servidor local. Com ele é possível integrar diversas soluções voltadas para casas inteligentes, desde acionamentos de lâmpadas inteligentes até o controle do sistema de segurança, sensores e cenas automatizadas (HOME ASSISTANT, 2022).

**Figura 3:** Tela de exemplo do Home Assistant.



Fonte: Home Assistant (2022).

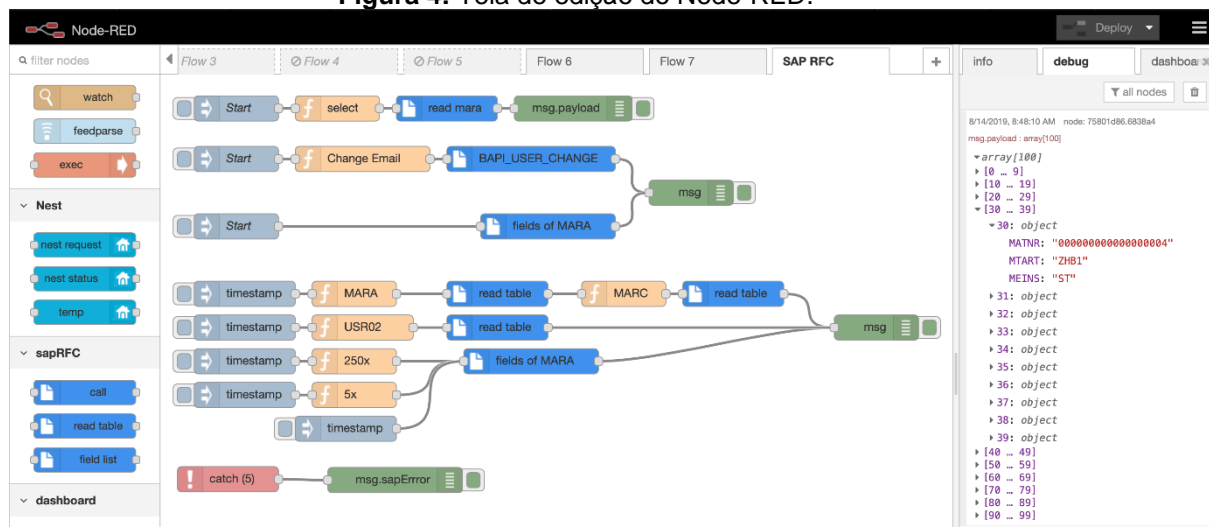
Além disso, é possível realizar a instalação de diversas ferramentas e serviços para apoio no desenvolvimento de soluções e integração entre dispositivos, como é o caso do ambiente *low code* Node-RED, que pode facilmente ser conectado a dispositivos através de protocolos como o MQTT (GRATSCHEFF, 2022).

### 2.1.3. Node-RED

O Node-RED é uma ferramenta de programação visual em fluxograma desenvolvida pela IBM em 2013 por Nick O’Leary e Dave Conway-Jones que surgiu como uma prova de conceito para visualização e manipulação de tópicos MQTT e que se tornou uma ferramenta com ampla conectividade entre diversos serviços, hardwares e APIs (NODE-RED, 2022).

Através de um navegador de internet, é possível ter acesso ao editor (baseado no Node.js) e criar diversas aplicações utilizando uma paleta de nós dentro do espaço de trabalho (**Figura 4**). Os nós nada mais são que funções que podem ser utilizadas dentro do editor. Por padrão, há cerca de 40 nós disponíveis para utilização, porém, outros nós podem ser adicionados através das paletas disponibilizadas por outros desenvolvedores e pela comunidade, tornando o Node-RED uma ferramenta bastante completa (NODE-RED, 2022).

**Figura 4:** Tela de edição do Node-RED.



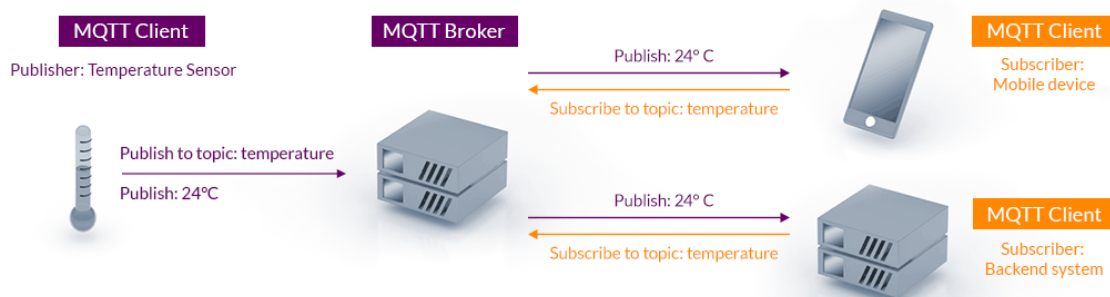
**Fonte:** Node-RED (2022).



#### 2.1.4. O Protocolo MQTT

O protocolo MQTT é um protocolo de troca de mensagens amplamente utilizado para aplicações em IoT desenvolvido pela IBM e Eurotech na década de 90 e padronizado pela OASIS. O protocolo é baseado no modelo publicador-subscritor e, por se tratar de um protocolo extremamente leve, o MQTT é ideal para a conectar dispositivos remotos e com pouca largura de banda. Além disso, o MQTT (**Figura 5**) oferece comunicação bidirecional, entrega confiável de mensagens (definidos através de 3 níveis de qualidade de serviço), possibilidade de encriptação da mensagem e escalabilidade para milhares de dispositivos IoT (MQTT, 2022).

**Figura 5:** Arquitetura padrão de troca de mensagens MQTT.



**Fonte:** MQTT (2022).

#### ***Publish x Subscribe***

O conceito de *publish* e *subscribe* (publicador e subscritor/assinante) define basicamente a maneira que um dispositivo irá se comunicar através do protocolo MQTT. Quando um dispositivo/cliente pretende enviar suas informações, esse deve se conectar ao servidor MQTT (conhecido como *broker*) como um publicador e enviar a mensagem em um determinado tópico. Já quando o dispositivo pretende receber uma mensagem, ele realiza o mesmo processo de conexão ao *broker* mas agora como subscritor no tópico gerado pelo publicador. Através dos tópicos, é possível que o subscritor receba apenas as mensagens que são de seu interesse, assim como o publicador é capaz de enviar a mensagem de maneira a distinguir facilmente a categoria da informação (D'ORTONA, TARCHI e RAFFAELLI, 2022; FERNANDES, 2022).

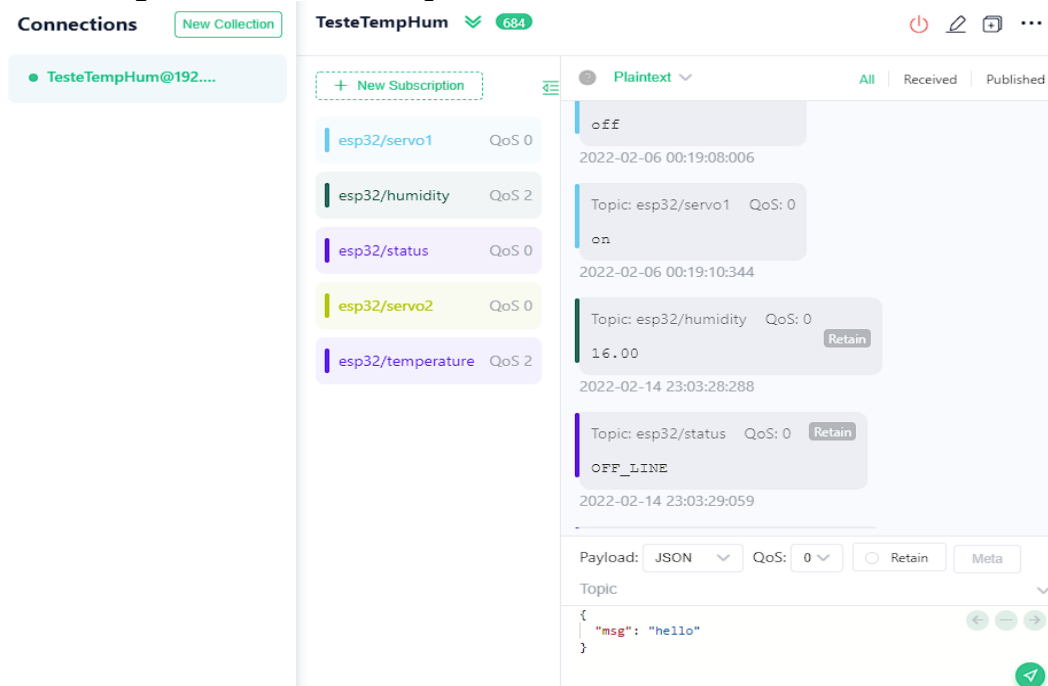
## Broker

O *broker* é um servidor responsável pelo roteamento das mensagens publicadas até os subscritores, intermediando a conexão (MQTT, 2022). Um dos servidores mais utilizados é o Eclipse Mosquitto, que implementa a versão 5.0, 3.1.1 e 3.1 do protocolo MQTT. O Mosquitto recebeu popularidade por se tratar de um servidor leve, robusto e de fácil utilização (MOSQUITTO, 2022).

## MQTT X

O MQTT X é um cliente MQTT para desktop que pode ser executado em sistemas operacionais como Windows, Linux e macOS. Com o MQTT X (Figura 6) é possível visualizar, publicar e subscrever as mensagens MQTT de uma maneira mais prática e rápida, sendo uma ferramenta bastante útil durante o processo de desenvolvimento (MQTT X, 2022).

**Figura 6:** Troca de Mensagens MQTT observadas através do MQTT X.



**Fonte:** Dos autores.

## 2.2. Hardware

O hardware proposto faz uso de uma placa de desenvolvimento Wemos D1 Mini Pro que carrega consigo o processador ESP8266 e cuja função é realizar o controle dos sensores e atuadores, além de realizar a comunicação com o broker MQTT. Para acionamento e fornecimento da ração, fez-se uso do

servo motor 9imod s0600m. Além disso, um sensor de temperatura e umidade foi empregado com intuito de obter mais dados acerca do sistema. A estrutura física foi projetada e impressa em impressoras 3D do tipo FDM (*Fused Deposition Modeling* – Modelagem por Fusão e Deposição). O detalhamento do hardware proposto é explanado nos tópicos seguintes.

### 2.2.1. Wemos D1 Mini Pro

Utilizando um chip ESP8266EX, o Wemos D1 Mini Pro (**Figura 7**) é uma placa de tamanho reduzido utilizada em projetos de IoT de projetos eletrônicos em geral. Com 11 pinos digitais de entrada e saída, 1 pino analógico, 16MB de memória flash, suporte a interrupções, PWM (*Pulse Width Modulation* – Modulação por Largura de Pulso), I2C e one-Wire, o Wemos é uma solução de fácil programação através da interface de desenvolvimento Arduino IDE. Além disso, o dispositivo conta com uma antena WIFI embutida, conector para antena externa e conexão micro-usb (FILIPEFLOP, 2022a).

**Figura 7:** Wemos D1 Mini Pro.



**Fonte:** FilipeFlop (2022a).

### 2.2.2. DHT11

O sensor de umidade e temperatura DHT11 (**Figura 8**) é um sensor de baixo custo e fácil utilização para medições de temperatura entre 0 e 50 °C e umidade entre 20 e 90%. A precisão do sensor de umidade é de  $\pm 5,0\%$  UR e de  $\pm 2,0$  °C para temperatura (FILIPEFLOP, 2022b).

**Figura 8:** Sensor de Temperatura e Umidade Sensor DHT11



Fonte: FilipeFlop (2022b).

### 2.2.3. Servo Motor 9imod s0600m

Utilizado para girar ou movimentar objetos em ângulos e distâncias específicas com garantia de posicionamento e velocidade, o servo motor é um dispositivo eletromecânico de grande precisão. Acionado através de pulsos PWM, o modelo 9imod s0600m (**Figura 9**) possui 180° de rotação, 8 kgf/cm de torque máximo quando operado em 7,4V e velocidade de 0,22seg/60° quando operado sem carga (ALIEXPRESS, 2022).

**Figura 9:** Servo Motor 9imod s0600m.



Fonte: Aliexpress (2022).

## 2. DESENVOLVIMENTO

Em posse de todos os elementos necessários para o desenvolvimento do comedouro, a primeira etapa foi a de construção de um algoritmo que permitisse a troca de informação via protocolo MQTT entre o microcontrolador e o servidor, assim como a leitura dos dados de temperatura e umidade provenientes do sensor e o acionamento do servo motor. Após a validação do

algoritmo e da comunicação, realizou-se a montagem da placa de circuito eletrônico contemplando os elementos de hardware. Seguido a isso, foram implementadas as automações e interfaces no Node-RED e Home Assistant, para controle e apresentação dos dados. Por fim, a estrutura física do comedouro foi impressa em 3D e realizada a montagem final. Detalhes são expostos nos tópicos seguintes.

### 3.1. Algoritmo

Tendo como referência o uso da placa de desenvolvimento Wemos D1 Mini Pro, visto o seu baixo custo e disponibilização de recursos como WiFi e portas de entrada e saída, foi criado um algoritmo baseado em Linguagem C/C++ para início da validação do cenário elaborado.

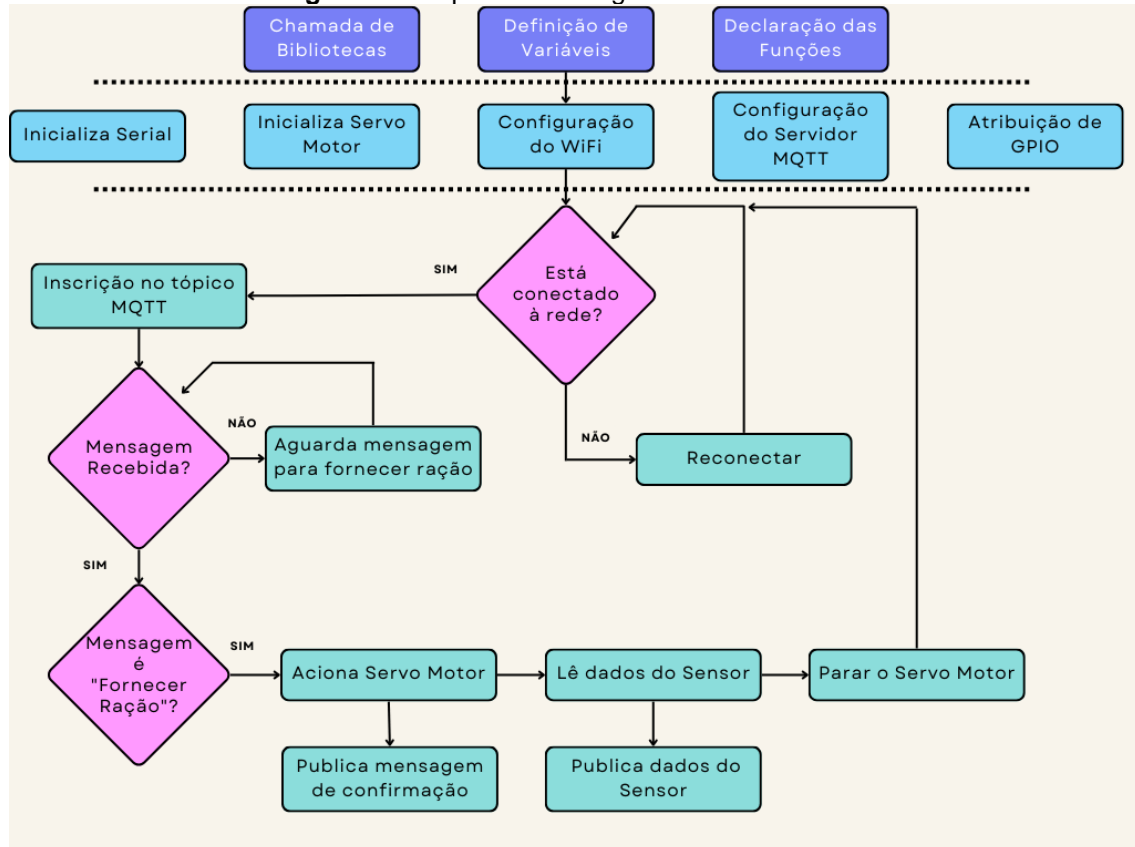
Para isso, utilizou-se da plataforma Arduino IDE, dada sua fácil integração com o microcontrolador utilizado, facilidade de programação, gerenciamento de bibliotecas e amplo acesso à conteúdo para consulta e exemplos (ARDUINO, 2022).

Foi necessário realizar a configuração do microcontrolador Wemos D1 Mini Pro na plataforma Arduino IDE e a inclusão de algumas bibliotecas ao projeto, como a “PubSubClient.h”, uma biblioteca para envio e recebimento de pacotes MQTT; a “ESP8266WiFi.h”, biblioteca para configurar e gerir a conexão WiFi em dispositivos baseados no microcontrolador ESP8266; a “Wire.h”, responsável por gerenciar a transmissão de dados a partir do protocolo I2C; e a biblioteca “Servo.h”, que realiza a conversão do valor de rotação do servo motor para níveis PWM.

A **Figura 10** apresenta o algoritmo proposto, em que primeiramente são realizadas as chamadas das bibliotecas, declarações de variáveis e funções. Nesse ponto também foi definido o tempo em que o servo motor deveria manter seu estado ativo após o acionamento, delimitando assim a quantidade de ração fornecida. Posteriormente, são inicializados elementos como a comunicação Serial, atribuídos os parâmetros do servo motor e GPIO, e realizadas as configurações de WiFi e do protocolo MQTT. Por fim, o laço de repetição verifica a existência de uma conexão WiFi, se inscreve no tópico MQTT para aguardar o recebimento da mensagem para realizar o fornecimento da ração, que em caso de afirmativo, realiza o acionamento do servo motor e a

leitura dos dados de temperatura e umidade enviados pelo módulo DHT11, seguido pela publicação de uma mensagem de confirmação e dos dados do sensor lidos, e o cessar da ação do servo motor.

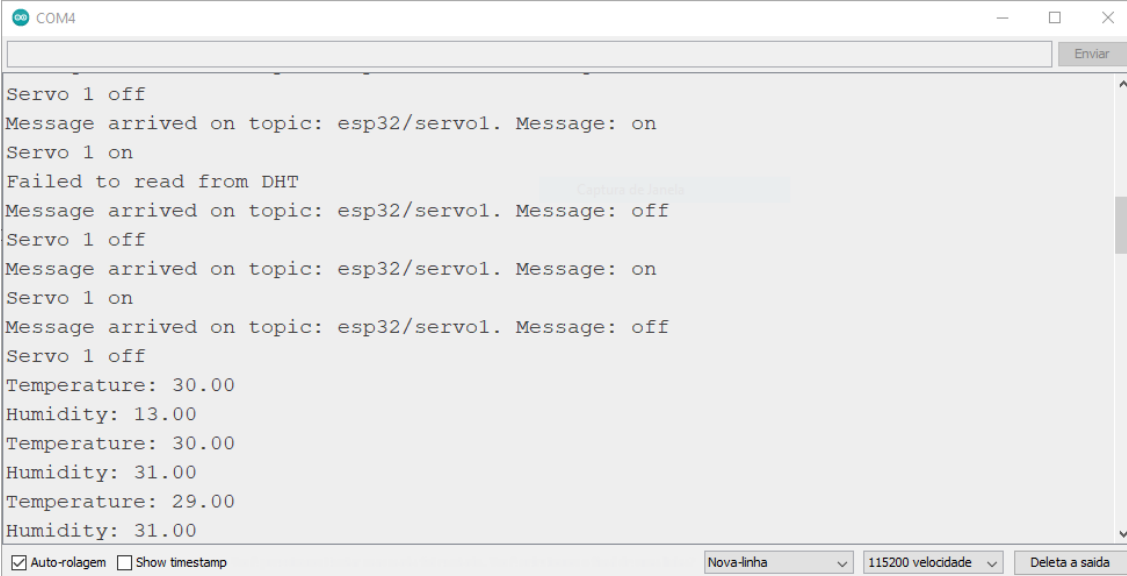
**Figura 10:** Arquitetura do algoritmo desenvolvido.



**Fonte:** Dos autores.

A **Figura 11** apresenta uma captura de tela do Monitor Serial do Arduino IDE, em que se pode observar as trocas de mensagens MQTT entre o microcontrolador e o servidor.

**Figura 11:** Monitor Serial do Arduino IDE - Comunicação MQTT.



```
Servo 1 off
Message arrived on topic: esp32/servo1. Message: on
Servo 1 on
Failed to read from DHT
Message arrived on topic: esp32/servo1. Message: off
Servo 1 off
Message arrived on topic: esp32/servo1. Message: on
Servo 1 on
Message arrived on topic: esp32/servo1. Message: off
Servo 1 off
Temperature: 30.00
Humidity: 13.00
Temperature: 30.00
Humidity: 31.00
Temperature: 29.00
Humidity: 31.00
```

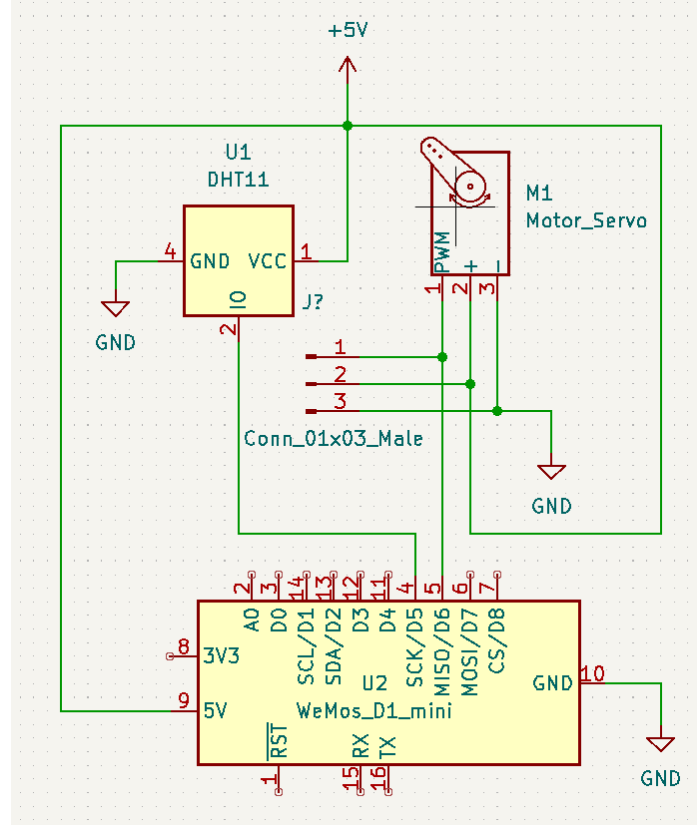
☒ Auto-rolagem ☐ Show timestamp Nova-linha 115200 velocidade Deleta a saída

**Fonte:** Dos autores.

### 3.2. Circuito Eletrônico

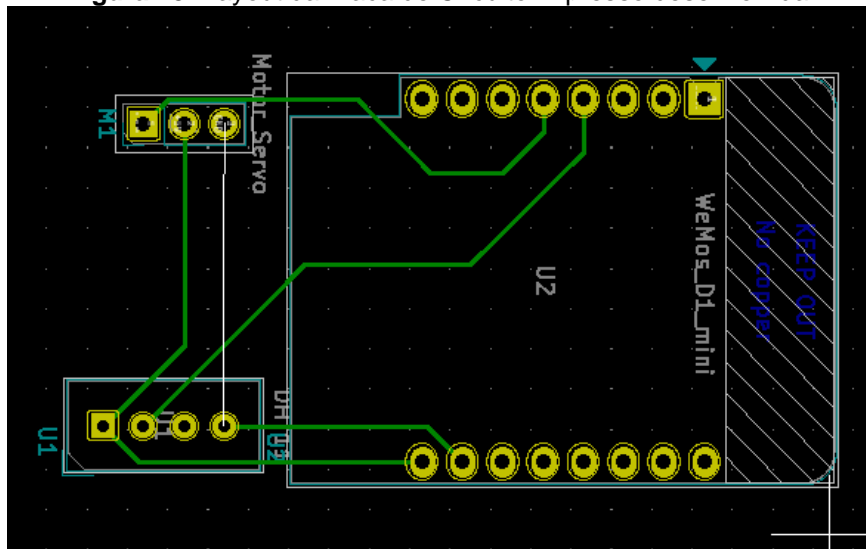
Uma vez concluída a etapa de validação e desenvolvimento do algoritmo, iniciou-se o projeto e confecção do circuito eletrônico. Utilizou-se da ferramenta de código aberto KiCad para construção do circuito esquemático e layout da placa de circuito impresso (**Figura 12** e **Figura 13**, respectivamente). O pino D5 do microcontrolador foi conectado ao pino IO do sensor DHT11 e o pino D6 ao pino de comando PWM do servo motor. A alimentação, representada na **Figura 12** pela fonte 5V, foi provida por uma fonte externa conectada à porta USB do microcontrolador e, por conseguinte, feitas as devidas conexões de alimentação e aterramento.

**Figura 12:** Circuito esquemático do hardware eletrônico desenvolvido.



Fonte: Dos autores.

**Figura 13:** Layout da Placa de Circuito Impresso desenvolvida.

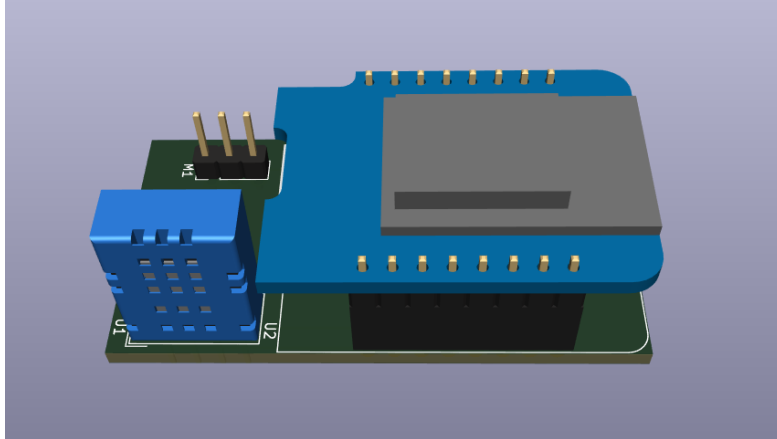


Fonte: Dos autores.

A **Figura 14** e **Figura 15** apresentam o modelo 3D do hardware eletrônico construído a partir do layout de placa de circuito impresso e a placa confeccionada com todos os componentes conectados, respectivamente.



**Figura 14:** Modelo 3D do hardware eletrônico desenvolvida.



Fonte: Dos autores.

**Figura 15:** Hardware eletrônico desenvolvido.



Fonte: Dos autores.

### **3.3. Integração, Automação e Interface Gráfica**

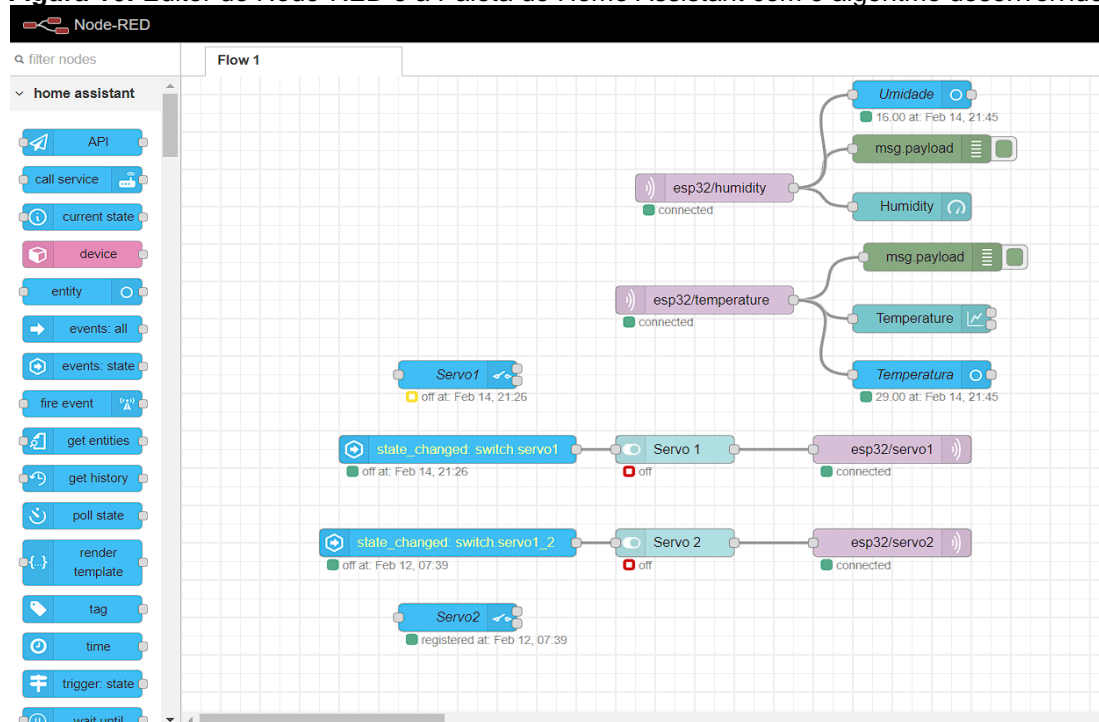
Para realização do acionamento automático e remoto do comedouro, conforme proposto, integrou-se os dispositivos de hardware e os sistemas de software através do mesmo protocolo de comunicação, o MQTT.

Conforme explanado anteriormente, o microcontrolador se coloca como assinante do tópico de acionamento do servo motor e como publicador nos

tópicos de umidade, temperatura e confirmação do acionamento do servo motor.

Um algoritmo foi desenvolvido em Node-RED (**Figura 16**) para realizar o intermédio da comunicação entre o microcontrolador e o Home Assistant através da inscrição nos tópicos de umidade, temperatura e confirmação enviados pelo microcontrolador e a publicação do tópico do servo motor cujo comando é enviado pelo Home Assistant.

**Figura 16:** Editor do Node-RED e a Paleta do Home Assistant com o algoritmo desenvolvido.

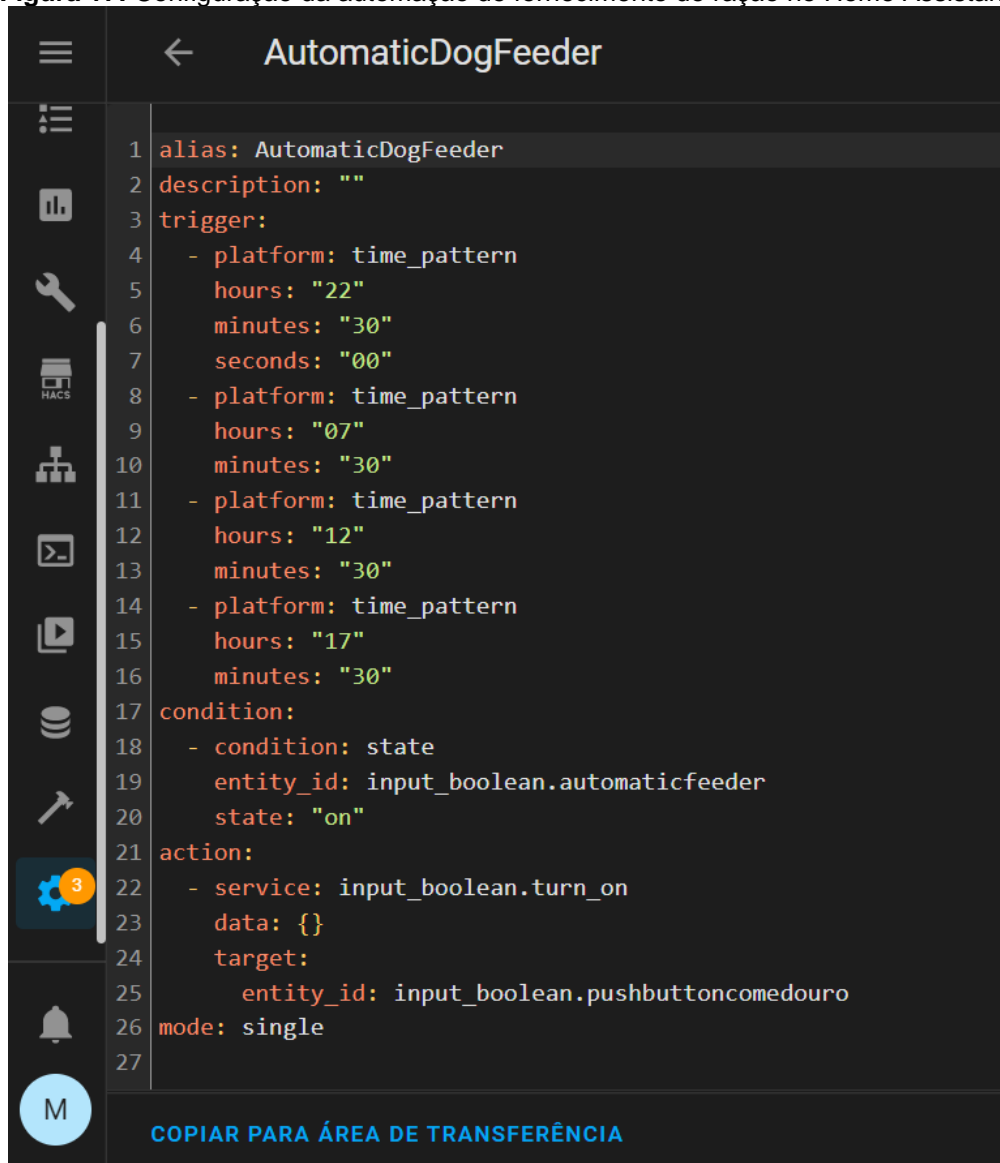


Fonte: Dos autores.

No Home Assistant, por sua vez, foi criada uma dashboard para visualização e controle das informações a respeito do comedouro. Utilizou-se de um botão para realização do acionamento remoto do comedouro e uma chave de seleção para seleção do modo automático de fornecimento de ração, além de outras duas chaves para indicar o estado de conexão do comedouro e o horário da última confirmação de fornecimento de ração recebida. Para gerenciar todos esses eventos, foram desenvolvidas automações (**Figura 17**) para estabelecer os horários de publicação automática das mensagens de fornecimento de ração e tratar os comandos de acionamento remoto, seleção de modo automático e o recebimento das mensagens de confirmação, temperatura e umidade.

Utilizando-se do Home Assistant, foi criada uma automação para envio de notificações ao serviço de mensagens Telegram, de maneira a notificar o usuário que a ração foi servida, assim como sobre o status da conexão com o microcontrolador.

**Figura 17:** Configuração da automação de fornecimento de ração no Home Assistant.



The screenshot shows the Home Assistant interface with a sidebar on the left containing various icons. The main area displays the configuration for an automation named 'AutomaticDogFeeder'. The configuration is as follows:

```
1 alias: AutomaticDogFeeder
2 description: ""
3 trigger:
4   - platform: time_pattern
5     hours: "22"
6     minutes: "30"
7     seconds: "00"
8   - platform: time_pattern
9     hours: "07"
10    minutes: "30"
11   - platform: time_pattern
12     hours: "12"
13     minutes: "30"
14   - platform: time_pattern
15     hours: "17"
16     minutes: "30"
17 condition:
18   - condition: state
19     entity_id: input_boolean.automaticfeeder
20     state: "on"
21 action:
22   - service: input_boolean.turn_on
23     data: {}
24     target:
25       entity_id: input_boolean.pushbuttoncomedouro
26 mode: single
27
```

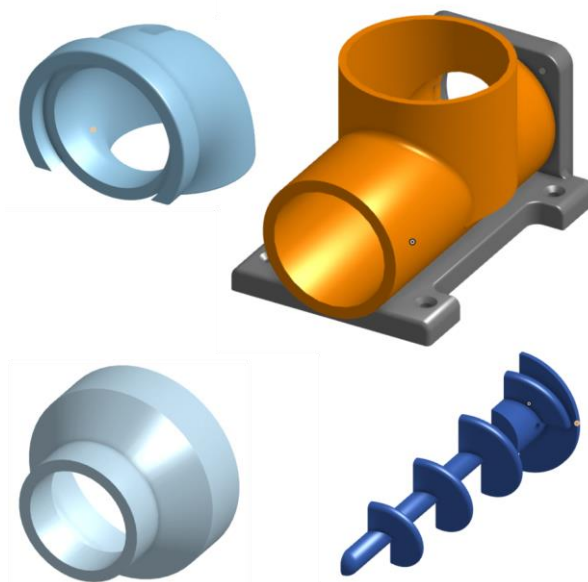
At the bottom of the configuration area, there is a button labeled 'COPIAR PARA ÁREA DE TRANSFERÊNCIA'.

**Fonte:** Dos autores.

### 3.4. Estrutura Física

A estrutura física principal do comedouro proposto foi desenvolvida através da impressão por FDM (*Fused Deposition Modeling* – Modelagem por Fusão e Deposição). Os modelos 3D (**Figura 18**) utilizados no projeto encontram-se disponíveis gratuitamente através do site “Thingiverse” (THINGVERSE, 2022). Na estrutura, tem-se o local para depósito da ração, um eixo espiral para movimentação da ração até a saída do comedouro e uma peça de encaixe do servo motor ao eixo espiral

**Figura 18:** Modelos 3D da estrutura física utilizada.

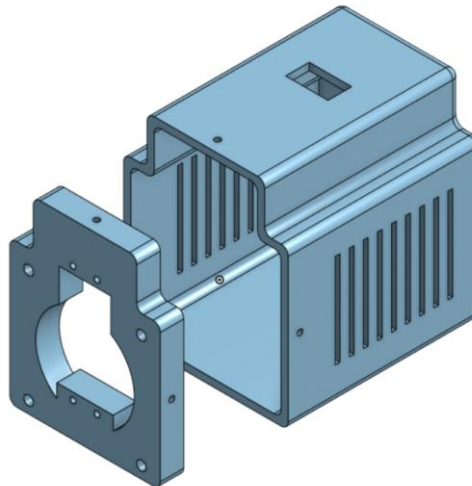


**Fonte:** Dos autores.

Entretanto, como o modelo gratuito não possuía uma estrutura para acomodar o hardware eletrônico, foi criado um novo modelo 3D (**Figura 19**) usando a plataforma Onshape, uma ferramenta CAD online e gratuita de fácil uso. Depois de finalizado o modelo 3D, o mesmo foi impresso através do mesmo processo FDM.

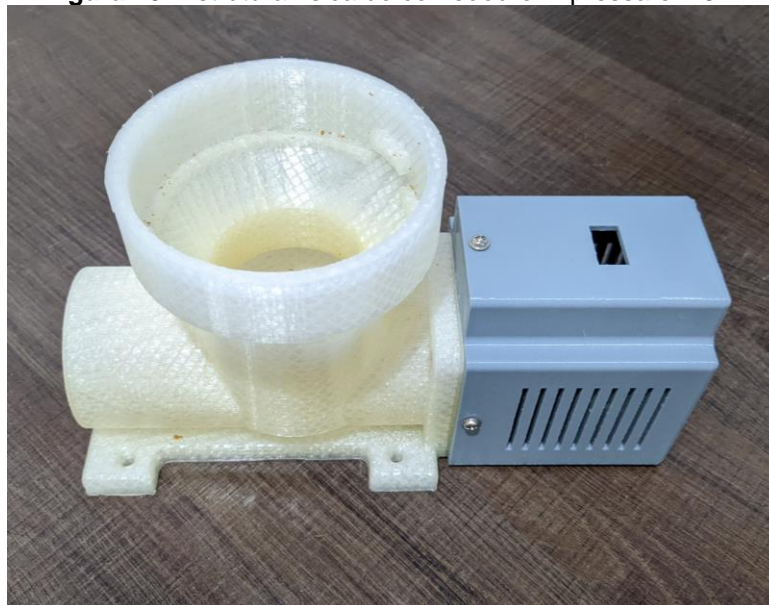
Observa-se através da **Figura 20**, a estrutura do comedouro após a impressão em 3D, juntamente com o hardware eletrônico desenvolvido.

**Figura 19:** Modelo 3D da estrutura de acomodação do hardware eletrônico.



**Fonte:** Dos autores.

**Figura 20:** Estrutura física do comedouro impressa em 3D.



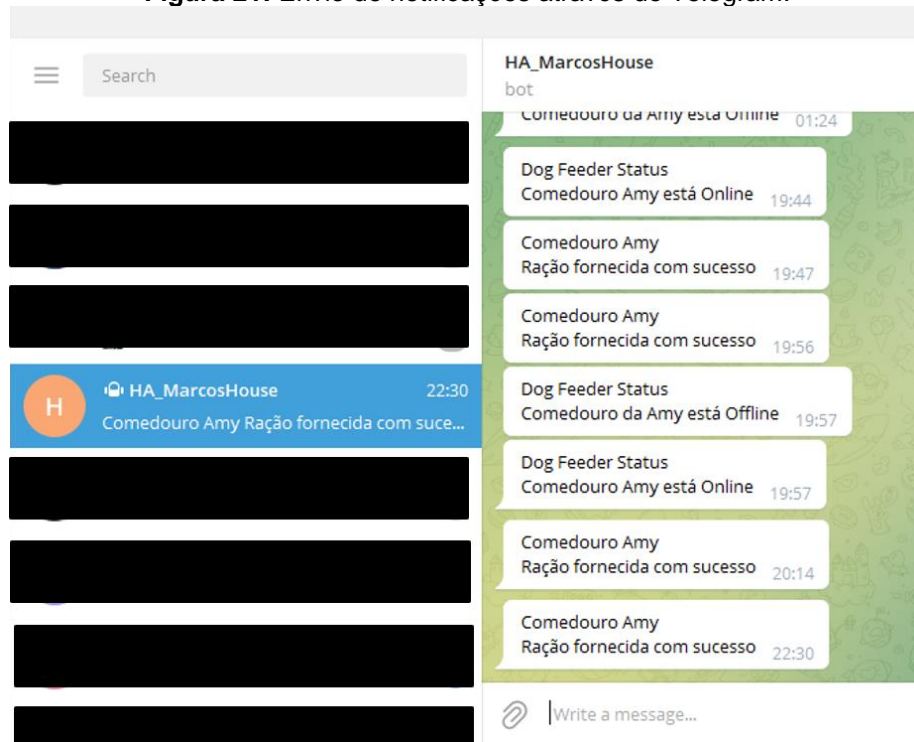
**Fonte:** Dos autores.

### 3. RESULTADOS

Após todas as configurações serem finalizadas, pôde-se acompanhar o funcionamento do projeto como um todo. O primeiro ponto observado foi o de acionamento remoto, em que ao ser acionado o botão de fornecimento de ração no Home Assistant, é publicada uma mensagem no tópico MQTT informando ao microcontrolador que o acionamento do servo motor deve ser feito. Essa mensagem é lida pelo microcontrolador que está conectado como subscritor do tópico do servo motor que, por sua vez, envia o comando para o servo motor.

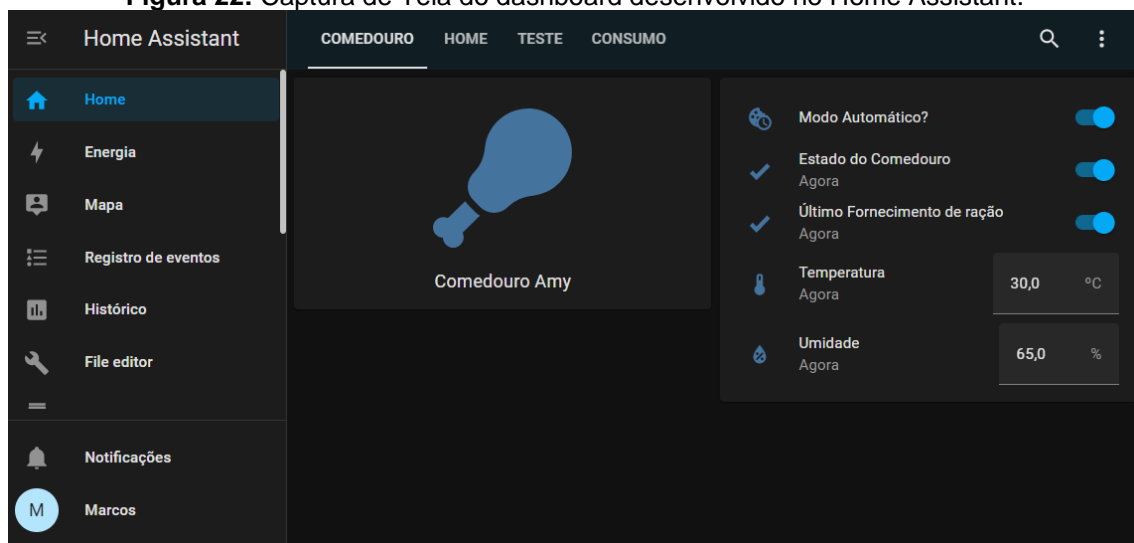
Em seguida, observou-se a execução automática do comedouro. Foram inseridos quatro horários distintos para realização da ação de fornecimento de ração, que ocorreram pontualmente às 7:00h, 12:00h, 19:00h, 00:00h. Em todos os momentos, pôde-se monitorar através do aplicativo de mensageria Telegram (**Figura 21**) e pelo próprio Home Assistant (**Figura 22**) a confirmação do fornecimento de ração de maneira satisfatória.

**Figura 21:** Envio de notificações através do Telegram.



Fonte: Dos autores.

**Figura 22:** Captura de Tela do dashboard desenvolvido no Home Assistant.



Fonte: Dos autores.

Pode-se ainda acompanhar a variação dos valores de umidade e temperatura ao longo do dia, entretanto observou-se que tais valores estavam distorcidos quando comparados a outros sensores de umidade e temperatura da casa.

A quantidade de ração fornecida em cada um dos acionamentos foi de aproximadamente 42g, o que resulta em 168g/dia de ração, valor que fica dentro dos valores recomendados para cães adultos de médio porte e 14kg (SPECIAL DOG, 2022).

Em relação aos custos de implementação, deve-se considerar que o Raspberry Pi foi utilizado como servidor centralizador dos sistemas de automação, podendo ser aplicado a qualquer outra automação que venha a ser utilizada. Dessa maneira, seu custo não foi considerado.

Considera-se também, que todos os sistemas e aplicações utilizadas são gratuitos e de código aberto. Além disso, o custo de impressão 3D por FDM não foi cobrado para o desenvolvimento deste projeto.

Assim, o único custo a ser considerado refere-se à construção do hardware eletrônico. A **Tabela 1** apresenta essa relação de custos médios de implementação, em que se observa o valor total de implementação de R\$ 157,73, que comparado a soluções de mercado com o mesmo nível de automação, possui uma diferença de custo de cerca de 62% menor.

**Tabela 1:** Tabela de Custos de implementação do Hardware Eletrônico.

Quantidade	Descrição	Valor Unitário	Total
1 Unidade	Wemos D1 Mini Pro	R\$ 69,90	R\$ 69,90
1 Unidade	Sensor DHT11	R\$ 15,40	R\$ 15,40
1 Unidade	Servo Motor 9imod s0600m	R\$ 69,43	R\$ 69,43
1 Unidade	Placa Perfurada	R\$ 3,00	R\$ 3,00
Total			R\$ 157,73

Fonte: Dos autores.

## 4. CONSIDERAÇÕES FINAIS

Observou-se através desse projeto que a utilização de softwares como o Home Assistant e Node-RED possibilitam a integração entre diversos dispositivos, tanto para IoT quanto para casas conectadas. Além disso, através da criação do cenário proposto, foi possível observar a facilidade de uso e a abrangência que o protocolo MQTT traz para aplicações em IoT. De maneira geral, pôde-se perceber um eficiente desempenho de todo o software e servidor aplicados no projeto.

A construção do hardware eletrônico e da estrutura física demonstraram grande robustez, embora algumas divergências nos valores de temperatura e umidade tenham sido percebidas. Isso possivelmente esteve relacionado ao fato de o sensor estar diretamente acoplado à placa de circuito impresso, muito próxima da entrada de energia e de outros componentes que podem produzir calor e variar a temperatura e umidade relativa do ar.

Pôde-se concluir ainda que, o sistema de acionamento remoto e automático de fornecimento de ração funcionaram com êxito, permitindo que o usuário controle o fornecimento de ração de acordo com a necessidade de alimentação do cão .

O sistema permite ainda, que outros dispositivos como atuadores e sensores sejam integrados à automação, contemplando o objetivo no início desse desenvolvimento.

Por fim, sugere-se para atividades futuras um aprimoramento do hardware eletrônico construído no que diz respeito ao sensor de temperatura e umidade, a implementação de uma balança para confirmação da quantidade fornecida e de um sensor ultrassônico para verificação da quantidade de ração disponível. Sugere-se ainda a utilização de RTC (*Real Time Clock* – Relógio de Tempo Real), para que o disparo das automações possa ocorrer nos horários pré-determinados independentemente da conexão com a rede.



## 5. REFERÊNCIAS

ALIEXPRESS. **6KG Servo Digital Metal Gear Servo 9imod s0600m**. Especificações. Disponível em: <<https://pt.aliexpress.com/>>. Acesso em: 10 nov. 2022.

ARDUINO. **What is Arduino?** Disponível em: <<https://www.arduino.cc/>>. Acesso em 10 dez. 2022.

D'ORTONA, Cristian; TARCHI, Daniele; RAFFAELLI, Carla. **Open-Source MQTT-Based End-to-End IoT System for Smart City Scenarios**. Future Internet, v. 14, n. 2, p. 57, 2022.

FERNANDES, Natalia. **O que é o protocolo MQTT. HI Tecnologia Automação Industrial**. Disponível em: <<https://www.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/>>. Acesso em 06 mar. 2022.

FILIPEFLOP. **Conheça a Wemos D1 Mini Pro: Mais funcionalidades para o ESP8266**. Disponível em: <<https://www.filipeflop.com/blog/conheca-a-wemos-d1-mini-pro-mais-funcionalidades-para-o-esp8266/>>. Acesso em 06 mar. 2022a.

FILIPEFLOP. **Monitorando Temperatura e Umidade com o sensor DHT11**. Disponível em: <<https://www.filipeflop.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>>. Acesso em 06 mar. 2022b.

GRATSCHEFF, Samuli. **Integrating robotics with a smart home environment**. Tese de Bacharelado. Information and Communications Technology. Turku University of Applied Sciences. 2022.

HOME ASSISTANT. **Awaken your home**. Disponível em: <<https://www.home-assistant.io>>. Acesso em 06 mar. 2022.

KARYONO, Kanisius et al. **Smart dog feeder design using wireless communication, MQTT and Android client**. In: 2016 International Conference

on Computer, Control, Informatics and its Applications (IC3INA). IEEE, 2016. p. 191-196.

LIU, Ruini. **Automatic Pet Feeder based on Single Chip Microcomputer**. In: Journal of Physics: Conference Series. IOP Publishing, 2021. p. 012104.

MANOJ, M. **Automatic Pet Feeder**. International Journal of Advances in Science Engineering and Technology, ISSN, p. 2321-9009, 2015.

MOSQUITTO, Eclipse. **An open source MQTT broker**. Disponível em: <<https://mosquitto.org>>. Acesso em 06 mar. 2022.

MQTT X. **An Elegant Cross-platform MQTT 5.0 Open Source Desktop Client**. Disponível em: <<https://mqttx.app>>. Acesso em 06 mar. 2022.

MQTT. **MQTT: The Standard for IoT Messaging**. Disponível em: <<https://mqtt.org>>. Acesso em 06 mar. 2022.

NODE-RED. **Low-code programming for event-driven applications**. Disponível em: <<https://nodered.org>>. Acesso em 06 mar. 2022.

OPENSOURCE. **What is a Raspberry Pi?**. Disponível em: <<https://opensource.com/resources/raspberry-pi>>. Acesso em: 13 dez. 2022.

RASPBERRY PI. **Raspberry Pi 3 Model B**. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>>. Acesso em: 13 dez. 2022.

RASPBERRY PI FOUNDATION. **What is a Raspberry Pi?** Disponível em: <<https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>>. Acesso em: 12 dez. 2022.

RAZALI, M. K. *et al.* **Smart Pet Feeder System and Big Data Processing to Predict Pet Food Shortage**. Turkish Journal of Computer and Mathematics Education (TURCOMAT), v. 12, n. 3, p. 1858-1865, 2021.

SHIDDIEQY, Rizaldy Hakim Ash *et al.* **Automated Pet Feeder Called Smart Pakan using 3D Printer with Open Source Control System.** IPTEK The Journal of Engineering, v. 6, n. 3, p. 58-62, 2021.

SPECIAL DOG. **Calculadora. Calcule a Quantidade Ideal.** Disponível em: <<https://www.specialdog.com.br/calculadora>>. Acesso em: 14 dez. 2022.

THINGVERSE. **Digital Designs for Physical Objects.** Disponível em: <<https://www.thingiverse.com/>>. Acesso em 20 nov. 2022.