



PROJETO IOT APLICADO À CONSTRUÇÃO DE UM ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS

João Marcelo de Lima Rodrigues¹, Idalmir de Souza Queiroz Júnior²

Resumo: O presente trabalho tem como objetivo projetar e construir um alimentador automático para animais domésticos que seja capaz de medir e mostrar para o usuário a quantidade de ração/alimento presente no recipiente. Assim como, ter a função de receber horas e quantidades padrões de ração para que seja depositado para o animal. O protótipo será controlado através de uma plataforma virtual e se conectará à rede Wi-fi da residência. Dessa forma, o usuário poderá automatizar esse processo e ainda controlar a dieta do seu animal de estimação.

Palavras-chave: Alimentador automático; Internet das coisas; Alimentador remoto; NodeMCU

1. INTRODUÇÃO

Animais de estimação são muito comuns atualmente nas residências, contudo, as rotinas das pessoas impedem cada vez mais que os donos tenham tempo para realizar algumas atividades básicas para com seus *pet's*.

Hoje em dia animais domésticos, sejam cães ou gatos, estão presentes em muitas residências. E possuir um animal de estimação em casa requer algumas responsabilidades. Com a globalização e os novos estilos de vida bastante corrido das pessoas fazem com que donos de *pets* não tenham tanto tempo para realizar as responsabilidades básicas para com seu animal [1], como alimentá-los. Ou até mesmo, pessoas que precisam viajar bastante a trabalho e não possam estar colocando a ração de seus animais, mas que ainda assim querem possuir um companheiro em casa. Na *internet* existem formas de desenvolver recipientes armazenadores de ração de modo que nunca falte alimento na vasilha do animal. Contudo, é interessante a ideia de que o dono possa controlar os horários e as porções de alimentação de seus bichos visto que cada raça e tipo físico de animal muda o modo no qual ele deve ser alimentado [2], sendo necessário assim controlar a sua dieta.

Pensando nisso, o presente trabalho tem como objetivo projetar e construir um alimentador automático para animais de estimação. O projeto em questão visa facilitar e automatizar a alimentação de animais domésticos, de forma que os donos possam programar horários de alimentação e quantidades de ração bem definidas através de uma plataforma em qualquer lugar do mundo e o dispositivo seja capaz de atuar conforme essas informações.

2. REFERENCIAL TEÓRICO

2.1. Alimentadores automáticos

No mercado existem alguns produtos que oferecem a comodidade e a praticidade para donos de animais de estimação quando se trata da alimentação. Alimentadores automáticos são dispositivos utilizados para tornar a tarefa de alimentação dos animais de estimação mais simples de forma a não ocupar tempo no dia a dia.

Internacionalmente podem ser encontrados modelos que permitem o controle do horário e da quantidade de alimento, contudo, os preços existentes são bastantes elevados, podendo custar até US\$ 179,00 [3] como é o caso do *PetNet®*. No Brasil podem ser encontrados produtos similares de até R\$ 1400,00 [4]. Esses modelos considerados mais inteligentes e tecnológicos contêm além das funções principais de manter uma dieta regrada, outras funções, como por exemplo, a possibilidade de fotografar o animal através do dispositivo e até realizar vídeo-chamada. Tais funções adicionais elevam o preço de mercado desses alimentadores.

Na *internet*, através de uma breve pesquisa, é possível encontrar tutoriais para a confecção de alimentadores automáticos a partir de tubos de PVC. Tal modelo não é considerado tão tecnológico por utilizar somente da gravidade para que tenha sempre uma certa quantidade de alimento no recipiente alimentador. Com os alimentadores de PVC também não é possível controlar os horários a serem alimentados com o fim de reger a dieta do animal. O custo para a confecção de um dispositivo como esse pode ser bem barato, podendo ser usado materiais de casa ou adquirindo-se facilmente em lojas de materiais de construção, por outro lado é possível comprar no mercado nacional por até R\$ 60,00 [5].

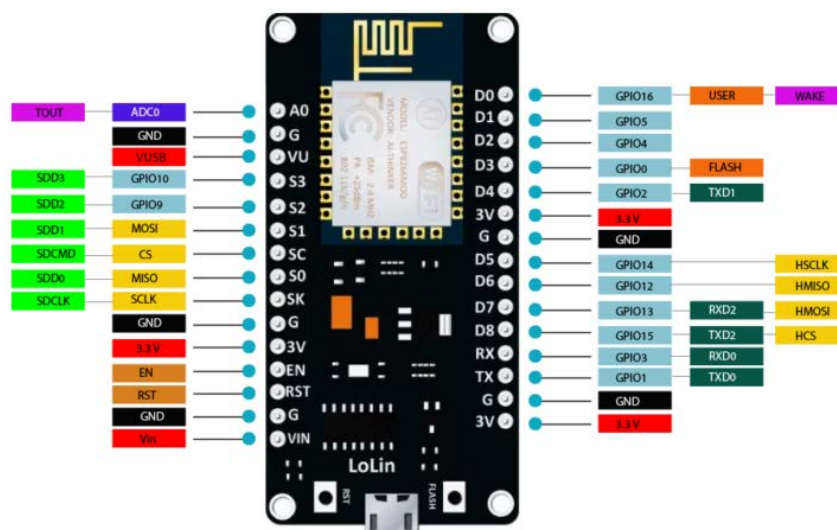
2.2. ESP8266

O ESP8266 é um chip com *Wi-fi* integrado ideal para projetos *IoT* (internet das coisas), seu baixo consumo de energia elétrica e seu formato compacto o torna o chip ideal. O ESP8266 possui um processador de 32 bits da série L106 Diamond, o seu *clock* pode ser de 80 MHz até 160 MHz. Ele suporta o protocolo de rede 802.11 b/g/n sendo possível uma conexão a uma distância de 70 metros por meio da antena de microfita. O módulo ESP8266 pode ser encontrado no mercado nacional por 50 reais [6].

O ESP8266 pode ser alimentado com até 3,3 V sendo essa sua tensão de funcionamento. Ele conta com 11 portas GPIOs sendo 4 do tipo PWM, uma porta ADC com uma precisão de 10 bits e com tensão máxima de saída de 1 V [7].

2.3. NodeMCU

A Node MCU é uma placa de desenvolvimento baseada no modelo ESP-12. Em conjunto com o ESP8266 o NodeMCU possui uma grande gama de aplicações. O seu uso se assemelha a de um Arduino UNO com um módulo *Wi-fi*. Contudo, a placa NodeMCU possui maior quantidade de portas digitais e apenas uma porta analógica. Ela pode ser alimentada com uma tensão de 4,5 V a 9 V, a partir de tensões superiores à máxima recomendada, o módulo processador poderá aquecer mais do que o normal. A alimentação e a comunicação com um computador se dão através de uma porta micro USB, o que facilita a compatibilidade de entrada para realizar a programação. Um ponto a ser considerado para o NodeMCU com o ESP8266 é que só existem saídas de tensão de 3,3 V [8], sendo necessário alimentação externa caso o usuário precise de uma tensão maior em sua aplicação. A Figura 1 mostra a distribuição dos pinos da placa.



uma célula de carga menor é sua precisão, sendo assim não é recomendado células mais robustas para uso que necessite de uma alta precisão.

2.5. Módulo HX711

O HX711 é um módulo amplificador de sinal amplamente utilizado em conjunto de *Strain Gages* na produção de balanças digitais. Ele tem a função de amplificar o sinal fornecido pela célula e possibilitar uma leitura mais precisa do sinal através de um microcontrolador como o Arduino ou outros. O funcionamento do módulo se dá através da conversão de mudanças no valor da resistência elétrica em potência elétrica. A sua tensão de funcionamento é de 4,8 V à 5,5 V DC, sendo a tensão recomendada de operação de 5 V [13]. Sua precisão é de 24 bits e possui uma taxa de atualização de 80 Hz. A Figura 2 mostra o circuito de utilização de uma célula de carga juntamente com o módulo HX711.



Figura 2: Circuito e funcionamento do conjunto célula de carga com o módulo HX711. (Autoria Própria)

2.6. MQTT Broker

O *Message Queue Telemetry Transport* (MQTT) é um protocolo de comunicação entre máquinas através da internet. Ele foi desenvolvido com base no conjunto de protocolos TCP/IP e permite conexão com a internet e consequentemente com outros dispositivos, além de poder usar serviços de *backend*. O protocolo MQTT foi inventado inicialmente pela IBM com a intenção de comunicação entre satélites e *pipelines* de petróleo [14]. Contudo, com a evolução e inovação na área de IoT, o protocolo de comunicação passou a ser amplamente utilizados para projetos desse tipo.

A utilização do MQTT em projetos IoT se torna ideal por vários fatores, um deles é a característica de ser uma rede leve e flexível, ou seja, permite a sua implementação em dispositivos com hardware muito restrito quando se trata de banda limitada e alta latência. Além disso, o MQTT é um protocolo de comunicação assíncrona, isso significa que esse sistema de comunicação desacopla o emissor do receptor [14]. Sem a necessidade de resposta de um servidor para a comunicação, esse protocolo facilita a comunicação entre as partes.

Uma outra característica desse tipo de protocolo é ser um modelo de publicação e assinatura. Dessa forma ele define duas entidades na rede, um *message broker* e os clientes. *Broker* é um servidor que tem a função de receber todos os dados que os clientes enviarem e daí encaminhar esses dados para outros clientes que forem convenientes. Um cliente pode ser qualquer coisa que forneça algum tipo de dado, pode ser um sensor térmico, uma célula de carga ou de qualquer outro tipo. Assim como um cliente também pode ser uma plataforma ou aplicativo que processe os dados recebidos de outro cliente e passe as informações ao usuário [15].

Pelo fato de que o MQTT organiza as informações recebidas em tópicos, os desenvolvedores de plataformas IoT que utilizam esse protocolo tem a facilidade e flexibilidade de desenvolver para usos bem específicos que cada usuário necessita. Por exemplo, o usuário pode enviar informações específicas para um dispositivo através de recursos bem definidos na plataforma e ao mesmo tempo receber dados enviados pelo dispositivo na mesma plataforma, isso tudo sem a necessidade de resposta de um servidor. O *broker* somente encaminha as solicitações e informações de clientes direto para outros clientes.

Ao passo de que ele permite essa troca de informações de forma assíncrona, o MQTT é leve. Ele possui um cabeçalho pequeno e fácil, somente com a funcionalidade de especificar o tipo de mensagem que está sendo enviada ou recebida. O dado pode ser em formatos bastante utilizados como o Base64, XML, binário criptografado ou JSON, dependendo apenas se o cliente final consiga decifrar o formato de dado [15].

2.7. Adafruit.IO

Adafruit.IO é uma plataforma *broker* gratuita para usuários que necessitem de até 10 clientes ou *feeds*, nome utilizado na plataforma. *Feeds* são variáveis que armazenam valores disponibilizados pelo usuário através da interface. A *Adafruit* tem a função de receber informações e armazená-las nos *feeds*, enviar dados armazenados para os dispositivos, e repassar todas essas informações para o usuário que a utiliza. Nela podem ser encontrados

diversos tipos de padrões de entradas de dados para que estes possam ser armazenados e enviados para clientes finais, sendo possível ao usuário manipular a interface da melhor forma para que atenda sua necessidade. A plataforma ainda possui alguns padrões de leitura de dados recebidos, ou seja, gráficos específicos na plataforma que retornam ao usuário informações transmitidas de sensores e clientes.

Para realizar a programação de um dispositivo utilizando o protocolo MQTT e a plataforma, existe no próprio site manuais e tutoriais de utilização [16]. Tanto o protocolo quanto a *Adafruit* necessitam da utilização de bibliotecas que forneçam os artifícios necessários para a programação. É através das bibliotecas adotadas que o usuário pode definir a ligação entre os clientes e o *broker*. A Figura 3 exibe o funcionamento da conexão *Wi-fi* da plataforma com o microcontrolador ESP8266 e, conseqüentemente, a internet.

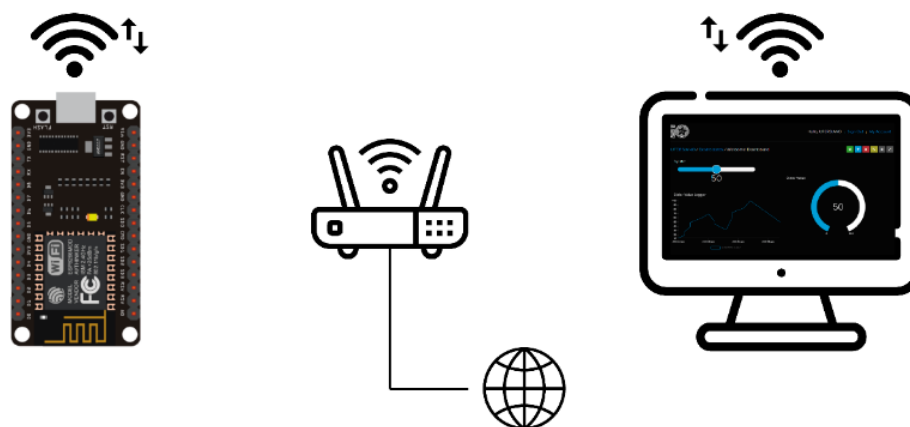


Figura 3: Estrutura das conexões de internet *Wi-fi* do ESP8266 e diálogo com a plataforma *Adafruit.IO*. (Autoria Própria)

3. MATERIAIS E MÉTODOS

O projeto inicialmente foi dividido em três partes de execução: código do programa, projeto do protótipo e construção final do dispositivo. Essas etapas foram bem definidas para que um protótipo fosse inteiramente concluído e visto seus resultados na prática. Uma ênfase foi dada à parte de programação e posteriormente iniciou-se a construção. As fases de execução foram definidas desta forma com base na importância de cada etapa para o projeto. A criação de um código que integrasse todos os circuitos necessários para o funcionamento do equipamento foi a fase mais importante visto que o protótipo só seria classificado como IoT se tivesse todas as funcionalidades do código funcionando.

3.1. Código do programa

Para a construção do código utilizado na automação do dispositivo alimentador se fez necessário a inclusão de algumas bibliotecas nas quais já forneciam funções necessárias para a programação. Sendo assim, iniciou-se a programação de fato. A priori se faz necessário inserir informações referentes a rede *Wi-fi* à qual o dispositivo estará conectado todo o tempo e as credenciais do usuário obtidas na plataforma *Adafruit.IO*. Tais dados serão utilizados no decorrer do código para fazer todos os *logins* necessários. O NodeMCU recebe as informações dos *feeds* e guarda tais informações dentro do código, tais dados serão utilizados como parâmetros de quantidade de ração e horas a serem dispostas as porções de alimento.

Posteriormente, obtidas todas as informações necessárias para a comparação de quantidades e horas pré-definidas, criou-se blocos de código com funções bem definidas, como por exemplo: comparar os horários, verificar a quantidade de ração existente na vasilha de alimentação e comparar com a quantidade mínima necessária, abrir e fechar a porta de passagem do alimento para o recipiente. É importante salientar que estes blocos de códigos seguem uma ordem lógica de ações, por exemplo, somente é verificado a quantidade de ração existente se antes for verificado uma hora pré-definida igual a hora corrente. Através do fluxograma exposto na Figura 4 é possível a visualização da lógica do programa.

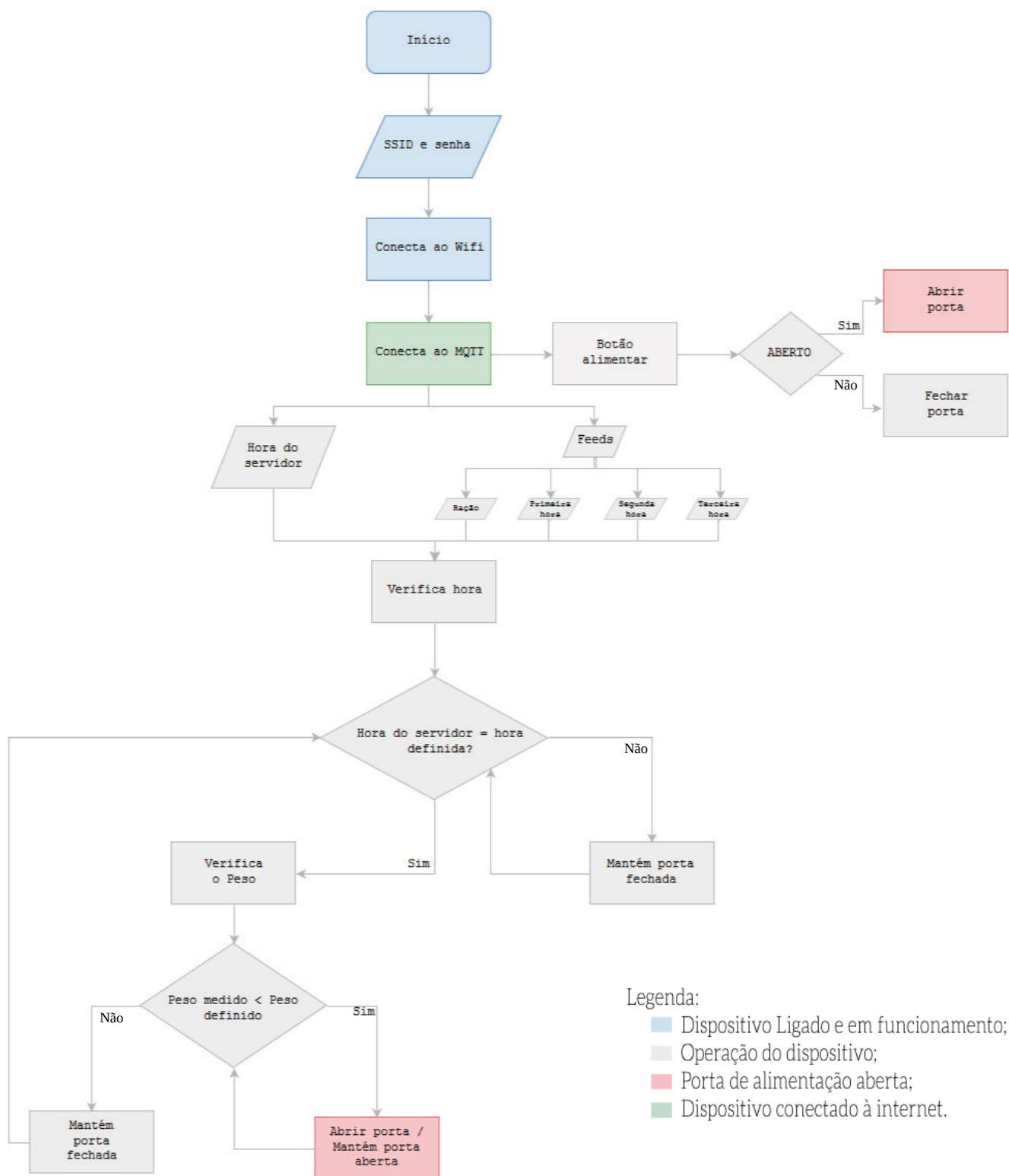


Figura 4: Fluxograma da lógica do programa. (Autoria Própria)

Para controlar e definir os dados de horário e quantidade de alimentação, usou-se a plataforma *Adafruit.IO*. Como visto anteriormente, o site oferece um serviço gratuito excelente para projetos IoT. A partir dela, criou-se um *dashboard* com seis *feeds*, ou seja, seis possibilidades de variáveis para troca de dados. Destas, cinco são responsáveis pelo envio de informação: primeira, segunda e terceira hora para se alimentar, quantidade de ração em gramas a ser despejada no recipiente e um botão alimentador que funciona com acionamento direto do servo não importando a hora corrente nem a quantidade já existente na vasilha. Além do mais, existe um último *feed* responsável por mostrar em tempo real a quantidade em gramas de ração disponível para o animal.

Os *dashboards* da *Adafruit* possuem uma quantidade de botões limitados, porém os existentes apresentam ótimas funcionalidades para o protótipo em questão. Sendo assim, foi possível editar e fazer a customização da interface para uma melhor apresentação ao usuário. A partir da Figura 5 pode-se observar a organização do *dashboard* criado para enviar os dados necessários para NodeMCU e receber informações do mesmo acerca do nível de ração.



Figura 5: Dashboard da interface na plataforma *Adafruit.IO* do alimentador inteligente automático PetFeeder. (Autoria Própria)

3.2. Projeto do protótipo

3.2.1. DESENHO DAS PLANTAS

Inicialmente foram criadas plantas de construção e *design* para o protótipo proposto, tais plantas foram realizadas em um *software* CAD com o intuito de determinar os melhores materiais a serem utilizados, além de todas as dimensões para que fosse possível a execução da terceira etapa do trabalho. As Figuras 6 e 7 mostram, respectivamente, a vista frontal e lateral esquerda da planta do protótipo.

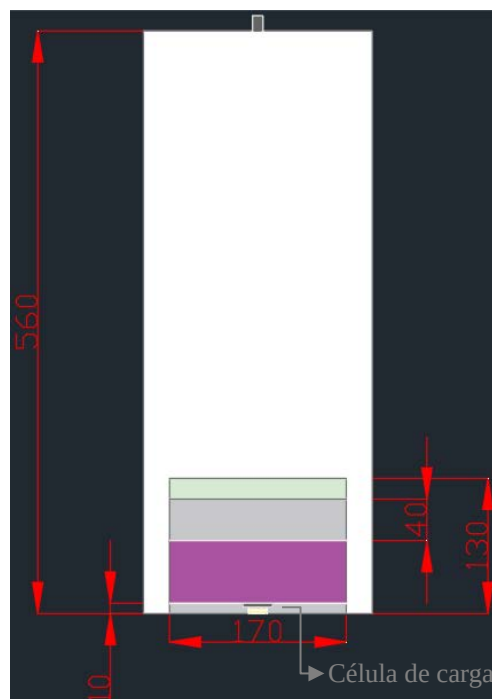


Figura 6: Vista frontal do protótipo desenvolvido. (Autoria Própria)

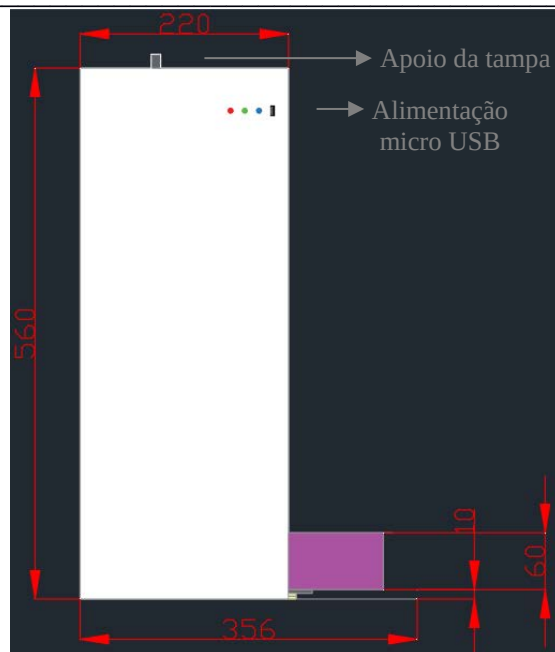


Figura 7: Vista lateral esquerda. (Autoria Própria)

Nota: As unidades de medidas cotadas estão em milímetros.

Assim, decidiu-se que o melhor material a ser utilizado como corpo físico de todo o protótipo seriam chapas de PVC fabricadas manualmente a partir de tubos e conexões de PVC existentes no mercado. Com isso, de acordo com o projeto, se fez necessário $0,706 \text{ m}^2$ de área de chapa de PVC para contemplar toda a construção. Além de que, foi preciso uma redução de PVC de $150 \times 100 \text{ mm}$, uma conexão do tipo curva de 100 mm , e um tubo de 150 mm com $0,2 \text{ m}$ de comprimento para a realização do reservatório da ração. Todas as outras peças foram feitas a partir das chapas de PVC. A Figura 8 ilustra uma perspectiva do protótipo modelado.

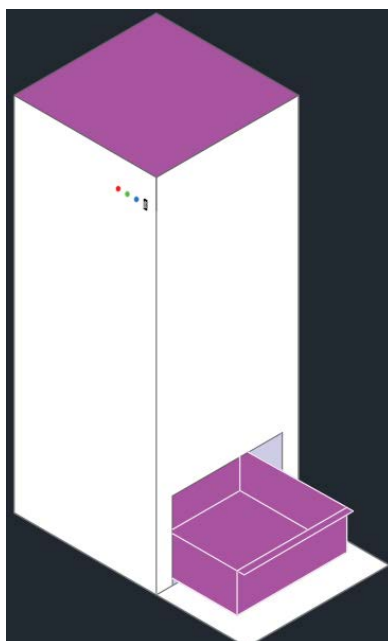


Figura 8: Perspectiva do protótipo modelado. (Autoria Própria)

Vale salientar que o recipiente no qual será depositada a ração está apoiada e contida em um suporte ao qual está fixado a célula de carga. Tal suporte tem a função além de servir como base de apoio para a vasilha principal, fornece a possibilidade da remoção do recipiente alimentador para higienização. A idealização desta parte foi omitida da modelagem apresentada na Figura 8 para facilitar a visualização da vasilha principal. Além disso, a tampa pela qual o usuário tem o acesso ao reservatório também foi omitida.

3.2.2. PROJETO E MONTAGEM DO CIRCUITO

Após a finalização do código fonte, deu-se início aos testes com o circuito completo montado: célula de carga + servo motor. O HX711 necessita de duas portas digitais e de uma alimentação de 3 à 5 V. Com isso, ligou-se a alimentação do módulo à uma das saídas de tensão de 3,3 V da placa microprocessadora. A célula de carga é ligada diretamente ao módulo amplificador.

O servo motor necessita de apenas uma porta digital a qual conecta-se o cabo de sinal. A sua alimentação é a padrão fornecida pela placa, 3,3 V. O dimensionamento do torque do servo motor, uma vez que sua função é abrir e fechar a porta da passagem, foi feito considerando o material da porta a ser levantado (definido anteriormente como chapa de PVC) e o tipo de rolagem que a porta teria. A partir de testes de alguns modelos de diferentes torques, observou-se que seria necessário um servo motor com um torque de 3 kgcm em uma tensão de funcionamento DC 3,3 ~ 6 V.

Para a alimentação da placa microprocessadora foi utilizado um carregador turbo para celular. Experimentalmente percebeu-se que carregadores simples não forneciam a corrente necessária para o perfeito funcionamento do circuito, isto é, os dados emitidos pela célula de carga e o funcionamento do servo motor não eram condizentes com o funcionamento esperado. A Figura 9 fornece o diagrama do circuito montado e testado no protótipo.

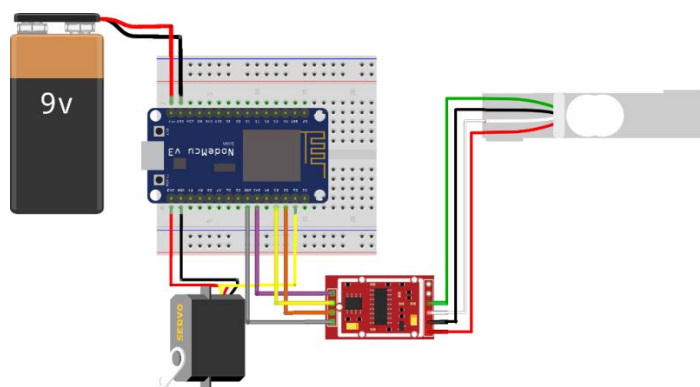


Figura 9: Diagrama de ligações do circuito montado. (Autoria Própria)

Nota: A bateria de 9 V no diagrama do circuito é meramente representativa, sendo utilizado de fato um carregador turbo para celular.

Para o circuito acima, omitiu-se alguns componentes não tão importantes para o real funcionamento do circuito, como por exemplo LEDs que indicam o funcionamento do dispositivo. Esta omissão se fez necessária tendo em vista a facilitação do entendimento do circuito. Contudo, integrou-se ao circuito final 3 LEDs a pinos digitais do NodeMCU. LED vermelho sinaliza que a porta da passagem de ração está aberta. O verde indica ao usuário que o dispositivo se encontra conectado à internet e ao protocolo MQTT, sendo assim, é possível a atualização de informações através da plataforma *Adafruit*. Já o LED azul sinaliza que o aparelho está ligado e em funcionamento.

3.3. Construção do protótipo

Com o circuito completamente montado em funcionamento, deu-se início a construção do protótipo físico que englobaria todas as funções de fato. A partir das plantas e das informações de quais materiais seriam utilizados deu-se o primeiro passo: construir o corpo do dispositivo. A partir de cortes e aquecimentos nos tubos de 150 mm de PVC foi possível a confecção de chapas para uma modelagem até se chegar ao formado e dimensões desejadas. Após toda carcaça externa concluída elaborou-se e fixou-se à estrutura do reservatório responsável pela ração, tal compartimento tem capacidade de 3,5 L, podendo comportar até 2,5 kg de ração.

A etapa final de construção se dá quando todo o protótipo, incluindo o circuito final, estava montado para se realizar os testes necessários.

4. RESULTADOS E DISCUSSÕES

4.1. Custos do Protótipo

Com o intuito de verificar o total gasto na construção do projeto e comparar com os valores existentes no mercado, elaborou-se uma tabela com os custos dos principais materiais utilizados tanto na confecção do circuito quanto para o corpo do dispositivo. A Tabela 1 expõe o custo total detalhado do protótipo.

Tabela 1 – Custo total do protótipo. (Autoria Própria)

Quantidade	Material	Valor Unitário	Valor Total
1	NodeMCU	R\$ 40,00	R\$ 40,00
1	Módulo HX711	R\$ 11,81	R\$ 11,81
1	Célula de carga 15KG	R\$ 12,80	R\$ 12,80
1	Servo AS-170 MG	R\$ 16,40	R\$ 16,40
20	<i>Jumpers</i>	R\$ 10,00	R\$ 10,00
3	LEDs	R\$ 1,00	R\$ 3,00
3	Resistores 1K Ω	R\$ 0,25	R\$ 0,75
1	<i>Horn</i> de fixação	R\$ 0,50	R\$0,50
1	Curva PVC 100 mm	R\$ 20,00	R\$20,00
1	Redução PVC 150 x 100 mm	R\$ 21,70	R\$ 21,70
1	2,5 m de tubo PVC 150 mm	R\$ 33,30	R\$ 33,30
-	Custo Total	-	R\$ 170,26

Ante o exposto, pode ser feita uma comparação entre o modelo construído e os produtos de mercado. Na tabela 2 pode se observar um comparativo.

Tabela 2 – Contraste de valores entre produtos de mercado e do projeto realizado. (Autoria Própria)

Modelo	Valor
<i>SmartFeeder</i> [3]	R\$ 668,62*
Alimentador Eletrônico Hoison [16]	R\$ 1399,90
Protótipo desenvolvido	R\$ 170,26

A tabela 2 ressalta a viabilidade de custos com relação a construção do projeto, o seu valor total equivale entre 12% e 25% dos valores existentes no mercado nacional e internacional.

*Nota: O valor de R\$ 668,62 foi obtido a partir de uma conversão direta sem custos extras de impostos.

4.2. Resultados

Tendo finalizado as três etapas propostas para a construção e funcionamento completo do dispositivo, cumpriu-se alguns testes, dentre estes o mais importante foi a aferição da funcionalidade do alimentador após inseridas as configurações de acordo com a Figura 5. Com os dados de quantidade de ração a ser alimentada e horários de alimentação definidos a partir da plataforma *Adafruit.IO*, foi possível observar na prática o funcionamento do projeto. A Figura 10 exibe um gráfico comparativo entre o nível de ração pré-estabelecido e o nível de ração em tempo real na vasilha.

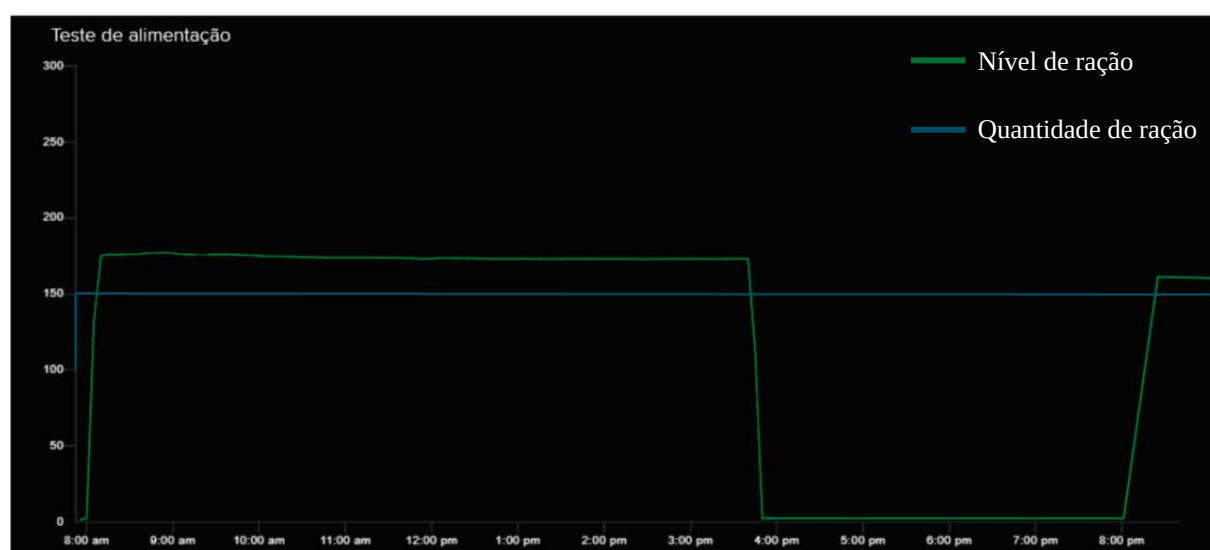


Figura 10: Gráfico comparativo entre o nível de ração em tempo real e o nível pré-estabelecido. (Autoria Própria).

O gráfico exposto foi gerado através da própria plataforma *Adafruit* com os feed de quantidade de ração

estabelecida pelo usuário e o nível de ração existente na vasilha em tempo real. Os gráficos são mais uma forma de visualização possível que o usuário pode utilizar para o monitoramento da alimentação do seu animal. Contudo, o gráfico em questão foi disposto unicamente com a intenção de verificar o êxito ou não dos testes realizados.

A priori, programou-se três horários para alimentação: 8:00 h, 13:00 h e 20:00 h. Assim como, definiu-se a quantidade de ração a ser alimentada, 150 gramas. O teste procedeu-se da seguinte forma, na primeira hora o alimentador entraria em funcionamento uma vez que verificasse uma quantidade de ração inferior a previamente definida. Na segunda hora, observou-se o comportamento do dispositivo quando fosse identificado a presença de uma quantidade de alimento maior ou iguala estabelecida, e de fato, comportou-se como esperado. Com uma medição de 176 gramas de ração existente no recipiente, o servo não é acionado. Próximo às 16:00 h, tirou-se o conteúdo existente na vasilha e aguardou-se até a terceira hora de alimentação. Às 20:00 h, o dispositivo atua novamente fornecendo 160 gramas de ração para a alimentação.

A posteriori, é possível perceber uma diferença entre o valor definido para a quantidade de ração e o peso fornecido pelo dispositivo. Tal diferença pode ser explicada pelo fato de que o servo necessita de um tempo de resposta até que ele interrompa completamente o fluxo de ração para a vasilha. Além de que, a própria célula de carga apresenta variações mínimas em sua leitura e apresentação de dados. Portanto, essa discrepância entre os valores obtidos e o pré-definido pode ser desconsiderado, uma vez que o valor excedente não prejudicará na alimentação do animal. A figura 11 revela o protótipo em seu estágio final de construção e em funcionamento.



Figura 11: Protótipo final em funcionamento. (Autoria Própria)

Sendo assim, os resultados obtidos foram satisfatórios tendo em vista que o dispositivo se comportou conforme o esperado quando submetido a uma simulação da realidade. Situação está na qual o mesmo ficaria um dia, ou dias ligado para automatizar a alimentação diária de animais domésticos. Apesar de o protótipo ter sido projetado e construído para animais de pequeno e médio porte, a célula de carga é capaz de medir até 15 Kg de ração. Portanto, os resultados analisados são suficientes para a comprovação da eficiência do dispositivo.

5. CONSIDERAÇÕES FINAIS

O presente trabalho se propôs a projetar e construir um dispositivo IoT alimentador automático visando uma maior comodidade e automação da tarefa. Das três etapas pré-definidas para o projeto, todas foram contempladas e concluídas, sendo assim, obteve-se êxito no objetivo inicial do projeto. O custo total do dispositivo se encontra

bem abaixo dos valores encontrados no mercado para produtos com funções equivalentes, foi possível observar que através de materiais de baixo custo pode-se construir um protótipo funcional e elegante dependendo da customização aplicada na fabricação.

Outrossim, é possível através da plataforma *Adafruit.IO* a implantação de mais *feeds* na versão gratuita do serviço e com isso aumentar as funcionalidades do produto final para que se torne cada vez mais inteligente e automatizado. Para estudos posteriores em trabalhos IoT nessa área de atuação fica a sugestão de realizar um dispositivo capaz de armazenar, detectar e fazer o abastecimento não somente da ração, mas também de água. Ou até mesmo a realização de um projeto com maiores capacidades, podendo ser usado em proporções maiores como por exemplo em canis ou clínicas veterinárias.

Portanto, o presente projeto se mostrou eficaz em atender as especificações e funções que foram propostas no início do trabalho. Além do mais, por ser idealizado e construído inteiramente do zero, pode ser considerado um produto comercial que poderá ser reproduzido para venda em ambientes *Petshop*.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MAZON, Marcia da Silva; MOURA, Wandgleisom Garcia de. Cachorros e humanos: Mercado de rações pet em perspectiva sociológica. *Civitas*, Porto Alegre, p. 21, 27 abr. 2017. Disponível em: <http://www.scielo.br/pdf/civitas/v17n1/1984-7289-civitas-17-01-138.pdf>. Acesso em: 20 maio 2019.
- [2] MAIA, Wanessa Ferreira. **EXPERIÊNCIA TÉCNICO-COMERCIAL EM ALIMENTAÇÃO DE ANIMAIS DE COMPANHIA**. 2010. Trabalho de conclusão de curso (Graduação em Medicina Veterinária) - Discente, Paraná, 2010. Disponível em: <https://tcconline.utp.br/wp-content/uploads/2011/08/alimentacao-animaais-de-companhia.pdf>. Acesso em: 22 fev. 2019.
- [3] PETNET.IO. **Automatic Feeder for Cats and Dogs**.
- [4] PETLOVE. **Alimentador Eletrônico Hoison Babá Robô para Pet**. [S. l.], 2019. Disponível em: <https://www.petlove.com.br/alimentador-eletronico-hoison-baba-robo-para-pet/p/destaque>. Acesso em: 22 jul. 2019.
- [5] MADERALI. **Comedouro para Cães e Gatos 2 Kg – PVC 60 cm – Maderali**.
- [6] ROBOCORE. **NodeMCU ESP8266-12 V2**.
- [7] SYSTEMS, Espressif. **ESP8266EX: Datasheet**. 6. ed. Espressif Systems. 2018.
- [8] FEZARI, Mohamed. **NodeMCU V3 For Fast IoT Application Development**. October, 2018.
- [9] AQEEL, Adnam. **Introduction to NodeMCU V3**. Gulberg, 11 out. 2018. Disponível em: <https://www.theengineeringprojects.com/2018/10/introduction-to-nodemcu-v3.html>. Acesso em: 23 jun. 2019.
- [10] TOLEDO BRASIL. **Você sabe como funciona uma célula de carga?**. [S. l.], 11 maio 2017. Disponível em: <https://www.toledobrasil.com.br/blog/artigos/detalhe/voce-sabe-como-funciona-uma-celula-de-carga>. Acesso em: 23 jun. 2019.
- [11] ROBOCORE. **Célula de carga 50Kg**.
- [12] ROBOCORE. **Módulo Amplificador HX711**.
- [13] SEMICONDUCTOR, AVIA. **24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales**.
- [14] ENGPROCESS. **O que é MQTT Broker**. [S. l.], 5 maio 2018. Disponível em: <https://engprocess.com.br/mqtt-broker/>. Acesso em: 23 jun. 2019.
- [15] YUAN, Michael. **Conhecendo o MQTT**. [S. l.], 4 out. 2017. Disponível em: <https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>. Acesso em: 23 jun. 2019.
- [16] ADAFRUIT.IO. **Adafruit.IO**. [S. l.], [20--]. Disponível em: <https://io.adafruit.com>. Acesso em: 23 jun. 2019.



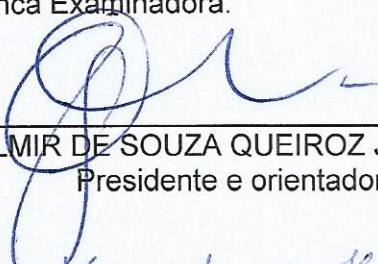
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal Rural do Semi-Árido – UFERSA
Centro de Ciências Exatas e Naturais – CCEN

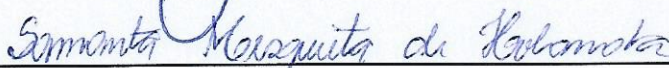
ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

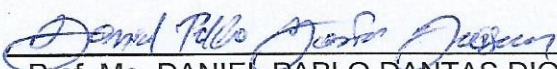
Às 08:00 horas do dia nove de agosto de dois mil e dezenove, no laboratório de Energia, Prédio de Laboratórios de Engenharias I, reuniu-se a Banca Examinadora de defesa de trabalho de conclusão de curso de autoria do aluno **JOÃO MARCELO DE LIMA RODRIGUES**, aluno do curso de Bacharelado em Ciência e Tecnologia desta universidade, N° de matrícula **2016010248**, com o título **“PROJETO IOT APLICADO À CONSTRUÇÃO DE UM ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DOMÉSTICOS”**. A Banca Examinadora ficou assim constituída por três membros: Prof. Dr. IDALMIR DE SOUZA QUEIROZ JUNIOR, presidente da banca e orientador do Trabalho de Conclusão de Curso; Prof. Me. SAMANTA MESQUITA DE HOLANDA e Prof. Me. DANIEL PABLO DANTAS DIOGENES, como membros. Concluída a defesa, procedeu-se o julgamento pelos membros da banca examinadora, tendo o aluno obtido as seguintes notas: 10,0; 10,0; 10,0. Apuradas as notas verificou-se que o aluno foi aprovado com média geral 10,0. E para constar, eu, Idalmir de Souza Queiroz Júnior, lavrei a presente ata que, após lida e aprovada pelos membros da banca examinadora, será assinada por todos.

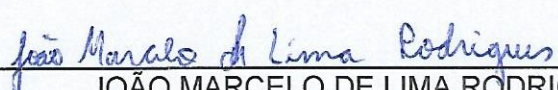
Mossoró, 09 de agosto de 2019.

Assinatura dos membros da Banca Examinadora.


Prof. Dr. IDALMIR DE SOUZA QUEIROZ JUNIOR – UFERSA
Presidente e orientador


Prof. Me. SAMANTA MESQUITA DE HOLANDA – UFERSA
Primeiro Membro


Prof. Me. DANIEL PABLO DANTAS DIOGENES - UFERSA
Segundo Membro


JOÃO MARCELO DE LIMA RODRIGUES
Discente