

CENTRO PAULA SOUZA

Etec DE MAUÁ

Curso de Informática

Daniela, Davi, Kevyn e Yuri

Alimentador Automático de Pets

Mauá

2025

Daniela, Davi, Kevyn e Yuri

Alimentador Automático de Pets

Trabalho de Conclusão de Curso (TCC) apresentado ao Curso Técnico em 2025 da ETEC de Mauá, orientadas pelos Prof. Edson Aparecido e Carlos Ronny de Sousa, como requisito para obtenção do título de técnico em Informática.

MAUÁ

2025

SUMÁRIO

1. Dedicatória	1
2. Resumo.....	2
3. Abstract	(3, 4, 5, 6,

7)

4. Introdução ao Desenvolvimento	8
5. Missão, visão, valores	(9, 10)
6. Justificativa	11
7. Problemática	12
8. Oportunidades	13
9. Análise SWOT	14
10. Metodologias	15
11. Ferramentas	16
12. Planejamento Lúdico	17
13. Arduíno código.....	(18, 19, 20)
14. Arduíno código.....	(21, 22, 23)
15. Arduíno código.....	(24, 25, 26)
16. Anexo.....	27
17. Conclusão.....	(28, 29, 30)

Dedicatória

Dedicamos este trabalho acadêmico aos nossos familiares, professores Victor Vicária, Carlos Ronny, Edson e ao orientador Gustavo da ETEC (Escola Técnica Estadual).

Resumo

Vamos desenvolver um projeto de alimentador automático para pets. A programação dos horários de alimentação será feita por meio de um aplicativo, permitindo até 6 horários por dia, com porções que variam de 50 g a 300 g de ração. Nosso reservatório poderá armazenar até 5 kg de ração.

Com o objetivo de economizar tempo dos tutores, desenvolvemos este projeto para melhorar tanto a saúde do pet quanto a eficiência do tempo, já que não será mais necessário reabastecer a tigela de ração toda vez que o pet precisar se alimentar.

Diferente dos modelos existentes no mercado, nosso produto mantém a ração isolada da exposição ao ar na maior parte do tempo, abrindo apenas durante o reabastecimento. Esse design garante um período maior de validade da ração e oferece proteção contra roedores e insetos.

Outro problema comum enfrentado por tutores de pets é o desafio de manter uma rotina de alimentação consistente, especialmente para quem possui um dia a dia corrido. Para isso, a praticidade do nosso aplicativo permitirá aos usuários programar a rotina de alimentação mais adequada, promovendo o bem-estar físico e mental dos seus pets.

ABSTRACT

We will develop a project for an automatic pet feeder. The scheduling of the feeding times will be done through an app, allowing up to 6 feeding times per day, with portions ranging from 50 to 300 grams of pet food.

Our container will be able to store up to 5 kg of pet food.

With the aim of saving time for pet owners, we developed this project to improve both the pet's health and time efficiency, as it will no longer be necessary to refill the food bowl every time the pet needs to eat.

Unlike existing models on the market, our product keeps the pet food isolated from air exposure for the majority of the time, opening only during refilling. This design ensures a longer validation period for the food and offers protection against rodents and insects.

Another common issue faced by pet owners is the challenge of maintaining a consistent feeding routine, particularly for those with busy schedules. To address this, the practicality of our application will enable users to program the most suitable feeding routine, promoting the physical and mental well-being of their pets.

● Mission:

To provide comfort and well-being for pets, while offering peace of mind to pet owners, ensuring regular and personalized feeding, regardless of their presence.

- **Vision:**

To become a reference in technology-driven solutions for pet care, promoting better quality of life for animals and greater peace of mind for their owners.

- **Innovation:**

Integrating technology into the daily care of pets.

- **Reliability:**

Ensuring pets receive food at the scheduled time as programmed.

- **Practicality:**

Facilitating remote feeding control through an intuitive mobile interface.

- **Animal Well-being:**

Prioritizing the health and happiness of pets by offering properly portioned and scheduled meals.

Advantages:

- Feeds pets automatically, even when owners are away.
- Allows for specific scheduling of feeding times and portions.
- Helps maintain a controlled diet, preventing overfeeding.
- Provides consistency in feeding times, reducing pet anxiety.

Disadvantages:

- Devices can be expensive
- May become less effective in the absence of power.
- Due to limited capacity, the product needs to be refilled periodically
- May experience performance issues after prolonged use.
- Setup phase may be difficult for some users.
- Growing awareness of animal welfare and growing demand for products that simplify pet care.

Possible improvements to the prototype:

- Potential for adding additional features, such as smartphone connectivity, cameras for monitoring, and app-controlled feeding.
- Opportunities to develop products for different types of pets and food varieties.
- Possibilities of collaborating with veterinary clinics and pet stores to promote and distribute the product.
- Develop a device that provides regular and controlled feeding for pets, minimizing human intervention. The team will build a prototype designed to simplify pet care and improve the daily routine of pet owners.
- Ability to store pet food.
- Programmable food distribution system.
- Food level sensor.
- User-friendly interface.
- Wi-Fi connectivity (optional).

Arduino IDE: The main platform for programming the Arduino.

- We will use Tinkercad, an online platform that allows for intuitive simulation of electronic circuits and Arduino programming.
- Electronic components.
- Development software.
- Manual materials, such as PVC pipes, among others.
- Data collection through dynamic activities, infographics, and forms.
We will work on developing an accessible app to control the prototype.

1.1 Introdução ao desenvolvimento:

A relevância deste tema está no fato de que os alimentadores automáticos contribuem para a qualidade de vida tanto dos pets, quanto dos tutores.

Com a rotina cada vez mais agitada dos tutores, a utilização de alimentadores automáticos de pets tem se mostrado uma solução prática para garantir uma alimentação regular e controlada, auxiliando aqueles animais que sofrem de problemas alimentares, ausência de seus tutores e alimentação irregular.

Desenvolver um protótipo funcional de um alimentador automático para mascotes que permita programar a quantidade e os horários de alimentação. A programação dos horários de enchimento do pote será feita através do aplicativo, podendo configurar até 6 horários por dia, com uma porção entre 50 e 300 gramas de ração. O nosso pote poderá armazenar até 5kg de ração.

Missão, Visão e Valores:

- **Missão:** Proporcionar conforto e bem-estar para os pets e uma tranquilidade para os tutores, garantindo que a alimentação seja feita de forma regular e personalizada, independentemente da sua presença.
- **Visão:** Ser referência em soluções envolvidas com a tecnologia promovendo cuidado e ajuda na alimentação de pets, proporcionando uma melhor qualidade de vida para os animais e uma maior tranquilidade para os tutores.
- **Inovação:** Integrar tecnologia no cuidado diário dos animais.
- **Confiabilidade:** Garantir que os pets receberão alimentação no horário correto programado.
- **Praticidade:** facilitar o controle remoto da alimentação, com interface intuitiva pelo celular.

- **Bem-Estar animal:** Priorizar a saúde e felicidade dos pets com porções adequadas e alimentações programadas.
- **Segurança:** Garantir que o equipamento seja seguro para os pets e resistentes a falhas.

Justificativa:

Pensando em economizar tempo para os “Pais de Pets”, pensamos neste projeto, unindo uma melhora na saúde do pet, e na economia de tempo, pois não será necessário preencher o pote com ração sempre que o pet for se alimentar.

Praticidade de uma alimentação programada remotamente.

Permitir uma alimentação melhor, mesmo com a falta de energia elétrica, com a nossa bateria de backup.

Diferente dos modelos já existentes no mercado, o nosso produto mantém a ração isolada dos efeitos do ar por maior parte do tempo, se abrindo apenas no momento da recarga, o que permite um maior período de duração da qualidade da ração e a proteção contra animais roedores e insetos.

Problemática:

Outro problema muito presente no cotidiano se trata dos proprietários de animais de estimação que possuem uma rotina cheia, não permitindo uma rotina adequada da alimentação do pet, pensando nisso, a praticidade do nosso aplicativo permitirá que essas pessoas programem a rotina mais adequada, equilibrando a saúde física e mental do animal.

Oportunidades:

Aumento da conscientização sobre o bem-estar animal e a demanda por produtos que facilitam o cuidado com pets;

Potencial para incorporar funcionalidades adicionais, como conectividade com smartphones e câmeras para monitoramento e alimentação controlada por aplicativos;

Oportunidade de desenvolver produtos para diferentes tipos de pets e variedades de ração;

Possibilidade de colaborações com clínicas veterinárias e pet shops para promover e distribuir o produto.

Análise Swot:

• Forças:

- Alimenta os pets automaticamente, mesmo quando os donos estão fora de casa;
- Permite programar horários e quantidades específicas de alimento;
- Ajuda a ter um maior controle da dieta, evitando a superalimentação;
- Com a alimentação monitorada, o animal será alimentado todo dia no mesmo horário, evitando ansiedade.

Fraquezas:

- Os dispositivos podem ter um custo elevado;
- Pode apresentar uma menor eficácia quando não há energia;
- Por ter uma capacidade limitada, o produto deve ser reabastecido de tempos em tempos.

Metodologia:

Objetivo: Desenvolver um dispositivo que forneça alimentação regular e controlada para animais de estimação, minimizando a intervenção humana. O grupo estará construindo um protótipo de algo que possa facilitar o cuidado com seu animal, de tal forma que evite a perda de tempo na rotina do consumidor.

Requisitos:

1. Capacidade de armazenar ração;
2. Sistema de distribuição de ração programável;
3. Sensor de nível de ração;
4. Interface de usuário amigável;
5. Conectividade Wi-Fi (opcional).

Ferramentas:

1. **Arduino IDE:** A plataforma principal para programação do Arduino, utilizaremos o Tinkercad, plataforma online que permite simular circuitos eletrônicos e programar em Arduino de maneira intuitiva;
2. Componentes eletrônicos;
4. Software de desenvolvimento;
5. Materiais manuais como cano PVC entre outros.

Ferramentas softwares:

- flutter;
- Arduino IDE;
- VS Code;
- Firebase;

Planejamento do Lúdico:

Coleta de dados através de dinâmicas com infográficos, e Forms, tentaremos realizar um app acessível para controlar o protótipo.

Paleta de Cores:

Utilizamos a cor prata para simbolizar tecnologia e inovação, já a cor roxo sendo o carinho.

A cor laranja do mascote kyd, foi feita em cartoon para atender todos os públicos.

Bibliotecas:

- WIRE.H;
- LiquidCrystal_I2C.H;
- Wifi.H;
- WebServer.H;
- Adafruit_Sensor.h;
- DHT.h;
- DHT_U.h;
- BLEDevice.h;
- BLEUtils.h;
- BLE2902.h;
- ArduinoJson.h.

Arduino Código:

```
DHTPIN 4 // Pino de dados do sensor DHT
```

```
#define DHTTYPE DHT11 // Tipo de sensor DHT11
```

```
DHT_Unified dht(DHTPIN, DHTTYPE);
```

```
uint32_t delayMS;
```

```
float temperatura;
```

```
float umidade;
```

```
WebServer server(80);
```

```
// Configurações BLE
```

```
BLEServer *pServer = NULL;
```

```
BLECharacteristic *pCharacteristic = NULL;
```

```
bool deviceConnected = false;
```

```
bool oldDeviceConnected = false;
```

```
uint32_t value = 0;
```

```
#define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
```

```
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-  
ea07361b26a8"
```

```
// Configurações do Access Point
```

```
const char* ssid = "ESP32_AP";
```

```
const char* password = "12345678";
```

```
IPAddress local_ip(192, 168, 0, 1);
```

```
IPAddress gateway(192, 168, 0, 1);
```

```
IPAddress subnet(255, 255, 255, 0);
```

```
// Definições do LCD
```

```
LiquidCrystal_I2C lcd(0x3F, 16, 2); // Endereço I2C e dimensões do  
LCD 16x2
```

```
// Funções BLE
```

```
class MyServerCallbacks : public BLEServerCallbacks {
```

```
    void onConnect(BLEServer *pServer) {
```

```
        deviceConnected = true;
```

```
    };
```

```
    void onDisconnect(BLEServer *pServer) {
```

```
        deviceConnected = false;
```

```
}
```

```
};
```

```
// Funções
```

```
void handleRoot();
```

```
void handleTemperature();
```

```
void setupBLE();
```

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    // Inicializa o sensor DHT
```

```
    dht.begin();
```

```
    // Inicializa o LCD
```

```
    lcd.init();
```

```
    lcd.backlight();
```

```
    lcd.setCursor(0, 0);
```

```
    lcd.print("Iniciando...");
```

```
    // Configura a ESP32-S3 como ponto de acesso com endereço IP fixo
```

```
    WiFi.softAP(ssid, password);
```

```
WiFi.softAPConfig(local_ip, gateway, subnet);
```

```
IPAddress IP = WiFi.softAPIP();
```

```
Serial.print("Endereço IP do Access Point: ");
```

```
Serial.println(IP);
```

```
// Configura as rotas do servidor
```

```
server.on("/", handleRoot);           // Página principal
```

```
server.on("/api/temperatura", handleTemperature); // API para JSON  
com dados
```

```
// Inicia o servidor web
```

```
server.begin();
```

```
// Detalhes do sensor
```

```
sensor_t sensor;
```

```
dht.temperature().getSensor(&sensor);
```

```
delayMS = sensor.min_delay / 1000;
```

```
// Configura Bluetooth BLE
```

```
setupBLE();
```

```
}
```

```
void loop() {  
  
    // Delay entre as leituras  
  
    delay(delayMS);  
  
  
    // Leitura da temperatura e umidade  
  
    sensors_event_t event;  
  
  
    // Leitura da temperatura  
  
    dht.temperature().getEvent(&event);  
  
    if (isnan(event.temperature)) {  
  
        Serial.println(F("Erro ao ler a temperatura!"));  
  
        temperatura = NAN;  
  
    } else {  
  
        temperatura = event.temperature;  
  
        Serial.print(F("Temperatura: "));  
  
        Serial.print(temperatura);  
  
        Serial.println(F("°C"));  
  
    }  
  
  
    // Leitura da umidade  
  
    dht.humidity().getEvent(&event);
```

```
if (isnan(event.relative_humidity)) {  
    Serial.println(F("Erro ao ler a umidade!"));  
    umidade = NAN;  
} else {  
    umidade = event.relative_humidity;  
    Serial.print(F("Umidade: "));  
    Serial.print(umidade);  
    Serial.println(F("%"));  
}
```

```
// Atualiza os dados no LCD
```

```
lcd.clear();
```

```
lcd.setCursor(0, 0); // Primeira linha
```

```
lcd.print("Temp: ");
```

```
lcd.print(temperatura);
```

```
lcd.print(" C");
```

```
lcd.setCursor(0, 1); // Segunda linha
```

```
lcd.print("Umid: ");
```

```
lcd.print(umidade);
```

```
lcd.print(" %");
```

```
// Envio dos dados via BLE
```

```
if (deviceConnected) {
```

```
    String bleData = "T: " + String(temperatura) + "C, U: " +  
    String(umidade) + "%";
```

```
    pCharacteristic->setValue(bleData.c_str());
```

```
    pCharacteristic->notify();
```

```
    delay(1000); // Delay para evitar congestionamento de BLE
```

```
}
```

```
// Gerenciamento de conexão BLE
```

```
if (!deviceConnected && oldDeviceConnected) {
```

```
    delay(500); // Espera para garantir desconexão adequada
```

```
    pServer->startAdvertising(); // Reinicia o advertising
```

```
    oldDeviceConnected = deviceConnected;
```

```
}
```

```
if (deviceConnected && !oldDeviceConnected) {
```

```
    oldDeviceConnected = deviceConnected;
```

```
}
```

```
// Lida com as requisições do servidor web
```

```
server.handleClient();
```

```
}
```

```
void handleRoot() {
```

```
    String html = "<html><body><h1>Dados de Temperatura e  
    Umidade</h1>";
```

```
    html += "<p>Temperatura: " + String(temperatura) + " C</p>";
```

```
    html += "<p>Umidade: " + String(umidade) + " %</p>";
```

```
    html += "</body></html>";
```

```
    server.send(200, "text/html", html);
```

```
}
```

```
void handleTemperature() {
```

```
    StaticJsonDocument<200> jsonDoc;
```

```
    jsonDoc["temperatura"] = temperatura;
```

```
    jsonDoc["umidade"] = umidade;
```

```
    String jsonResponse;
```

```
    serializeJson(jsonDoc, jsonResponse);
```

```
    server.send(200, "application/json", jsonResponse);
```

```
}
```

```
void setupBLE() {
```

```
    BLEDevice::init("ESP32_S3_BLE");
```

```
pServer = BLEDevice::createServer();
```

```
pServer->setCallbacks(new MyServerCallbacks());
```

```
BLEService *pService = pServer->createService(SERVICE_UUID);
```

```
pCharacteristic = pService->createCharacteristic(
```

```
    CHARACTERISTIC_UUID,
```

```
    BLECharacteristic::PROPERTY_READ |
```

```
    BLECharacteristic::PROPERTY_WRITE |
```

```
    BLECharacteristic::PROPERTY_NOTIFY
```

```
);
```

```
pCharacteristic->addDescriptor(new BLE2902());
```

```
pService->start();
```

```
BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
```

```
pAdvertising->addServiceUUID(SERVICE_UUID);
```

```
pAdvertising->setScanResponse(false);
```

```
pAdvertising->setMinPreferred(0x0);
```

```
BLEDevice::startAdvertising();
```

```
Serial.println("BLE Ready to notify");
```

```
}
```



Conclusão:

O protótipo de alimentador automático desenvolvido neste trabalho demonstrou-se uma solução viável e eficaz para a alimentação regular de pets, especialmente no contexto de tutores com rotinas atarefadas. O sistema permite programar até seis horários diários de alimentação, controlando porções entre 50 g e 300 g, com capacidade total de armazenamento de 5 kg de ração.

Principais vantagens identificadas:

- Autonomia no cuidado: alimenta regularmente os pets mesmo na ausência dos tutores;
- Controle preciso de dieta: evita sobrealimentação e ajuda no equilíbrio alimentar;
- Redução de ansiedade no pet: refeições previsíveis e consistentes confortam o animal;
- Qualidade da ração preservada: sistema fechado evita contaminação por ar, insetos e roedores;

Limitações reconhecidas:

- Investimento inicial elevado comparado a métodos tradicionais ou modelos simples;
- Dependência de energia elétrica (mesmo com bateria de backup, pode haver limitações);
- Necessidade de reabastecimento frequente, dada a capacidade máxima de 5 kg;
- Complexidade da configuração, especialmente para perfis de usuários com menor familiaridade tecnológica.

Oportunidades de aprimoramento:

- Integração com smartphones e câmeras, com controle remoto em tempo real;
- Diversificação do produto para outras espécies de pets e tipos de alimentação;
- Parcerias estratégicas com veterinários e pet shops, ampliando distribuição e credibilidade;

- Refinamento do software e design para tornar o uso mais intuitivo e acessível.
- Devido à capacidade limitada, o produto precisa ser reabastecido.
- A interface de configuração do aplicativo pode ser complexa para alguns usuários, principalmente para aqueles com idade avançada;
- Crescente conscientização sobre o bem-estar animal e crescente demanda;
- por produtos que simplifiquem os cuidados com os animais de estimação;

REFERÊNCIAS

- **Manual do Mundo; Faça um alimentador automático para cachorros**

11 de maio de 2019

link: https://youtu.be/TdMz7EMKBdY?si=_tC25H9jM50uX6sY

- Distribuidor Automático de Alimentos Petlibro 5 L para 2 gatos;

Alimentador com conexão via app, ideal para duas refeições diárias.

Capacidade de 5 litros.

Preço nas Americanas: ~R\$ 1.127.

- Ocean Tech Alimentador Automático Wi-Fi Otfish Care;

Modelo Wi-Fi, programável remotamente via app.

Preço no Mercado Livre: ~R\$ 428.

- PETLIBRO AIR Wi-Fi Feeder: destacado como “Best Overall” por The Spruce Pets, com bateria de longa duração (~30 dias), app intuitivo e controle fino de porções.