

Nome: Henrique Pignatari Moraes Rosa RA: 51913401

Este documento servirá como avaliação do primeiro bimestre, onde devem ser adicionadas as atividades relacionadas ao projeto temático, tanto as realizadas em grupo (**mesma resposta para todos os integrantes do grupo**), como individualmente. Fotos do projeto são obrigatórias.

Nome do projeto: SmartEatPet**1. Descrição detalhada:** (grupo)*(Inserir uma descrição do projeto, salientando como será a confecção, principais peças, funcionamento geral)*

SmartEatPet é um projeto de um alimentador automatizado para alimentação de animais, principalmente domésticos. Ele possui duas partes principais, um reservatório de ração que deve ser enchido pelo usuário e uma bandeja acoplada em uma balança onde o animal deve comer. Dentre as principais funcionalidades estão a possibilidade de agendar horários para a alimentação do animal, juntamente com esse horário é possível escolher o peso de ração que deve estar na bandeja nesse horário, esse agendamento é feito através de um aplicativo que será utilizado pelo usuário. Adicionalmente como funções secundárias ele possui um display para comunicação com o usuário, um sensor de presença para verificar se o animal está próximo.

O material de toda a estrutura é madeira MDF, todo o resto dos componentes serão os componentes eletrônicos que atuarão no controle das funcionalidades. O controle central do alimentador será feito por um Arduino Mega, a comunicação com o aplicativo será feita utilizando um ESP32, esses dois se comunicam para trocar as informações recebidas e requisitadas pelo usuário. Para guardar os dados recebidos utilizaremos um módulo de cartão SD, esse módulo também irá guardar arquivos de configuração e adicionais para o funcionamento do alimentador.

Para a confecção da estrutura, as partes separadas foram desenhadas no aplicativo de desenho 3D *Inventor* da Autodesk, posteriormente serão cortadas no corte a laser, ajustadas, finalizadas e montadas para a forma final. Depois disso, todos os componentes eletrônicos devem ser montados dentro da estrutura de madeira e recalibrados.

**2. Objetivo do projeto:** (grupo)*(O objetivo do projeto está ligado a persona que será a usuária. Qual problema irá solucionar? Qual o resultado final esperado?)*

O SmartEatPet tem como principal objetivo facilitar a vida do usuário, no caso o dono de seu animal de estimação, deixando o usuário despreocupado com a alimentação do seu animal. Com as funcionalidades do alimentador o usuário pode, por exemplo, trabalhar sem a preocupação de alimentar o seu pet. Além disso, o projeto também visa cuidar da

saúde do animal, pois ele não passará longos períodos sem alimentação e nem terá a possibilidade de se alimentar demais, evitando inanição e obesidade. Além disso, como ele funciona de maneira cíclica todos os dias é possível recolher dados sobre como o animal tem se alimentado, obtendo um controle e cuidado maior com possíveis doenças que o animal possa ter adquirido.

### 3. Motivação do projeto: (grupo)

*(Porque fazer este projeto?)*

A motivação desse projeto está intimamente ligada com o seu objetivo. Como o projeto visa aumentar a qualidade de vida do bicho de estimação e de seu dono o interesse vem diretamente do usuário. Além disso, o interesse da área da saúde e da etologia animal é a possível aquisição de dados comportamentais de diversos bichos. Alternativamente, esse é um projeto que pode ser escalado, podendo abranger grandes criadores de gado, por exemplo, pois possuiria a função de cortar a mão de obra que manualmente alimenta o gado. Portanto, é possível afirmar que há diversos motivos e interesses de várias partes em viabilizar o projeto, atuando em diversas áreas e podendo ser ferramenta para todos os que o utilizarem.

### 4. Desenho final do projeto: (grupo)

*(A essa altura, todos os grupos já DEVEM ter um desenho final do projeto, anexar o mesmo abaixo, com o máximo de vistas possível bem como todos os detalhes)*



Figura 1 - Desenho do projeto

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

**5. Lista final de componentes:** (grupo)

(Inserir abaixo todos os componentes que serão usados no projeto; apenas o nome e a quantidade)

1) Arduino Mega	qtd: 01
2) ESP32	qtd: 01
3) Módulo cartão SD	qtd: 01
4) Módulo LCD 20x4 com I2C	qtd: 01
5) Células de carga	qtd: 04
6) Módulo HX711	qtd: 01
7) Servo motor	qtd: 01
8) Madeira MDF	qtd: 01
9) Fonte 12V 5A	qtd: 01
10) Sensor de proximidade infravermelho	qtd: 01

**6. Desenvolvimento do projeto:** (grupo)

(Abaixo, descrever o andamento do projeto até a data do preenchimento deste relatório. Quais peças já foram produzidas? Quais programações já foram feitas? Quais atividades já estão concluídas? Quais os próximos passos? Ao final, anexar fotos das atividades realizadas)

Já realizado:

Já foram feitos os desenhos, os modelos 3D, bem como suas montagens. Baseado nisso, fizemos os protótipos de como deveria ficar a montagem. Além disso, foram realizados os testes da balança bem como sua calibragem, foi feito o aplicativo e a comunicação com o ESP32, foram realizados testes de comunicação e dos padrões dos protocolos de dados.

Próximos passos:

Desenvolver o método de como funcionará a portinhola deslizante para cair o alimento com o servo motor que será utilizado. Cortar no laser e montar a estrutura e acomodar os circuitos na montagem. Ajustes no aplicativo e na comunicação, esperar o recebimento dos materiais solicitados para desenvolver o restante da programação.

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

Imagens:

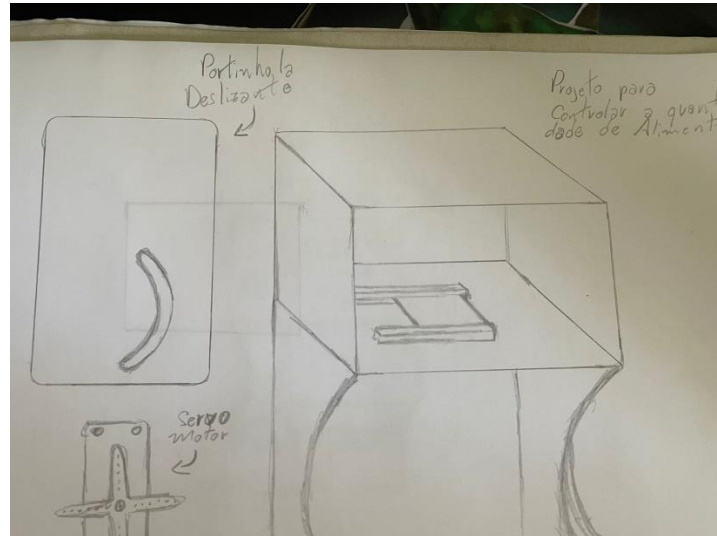


Figura 2 - Desenho à mão do projeto

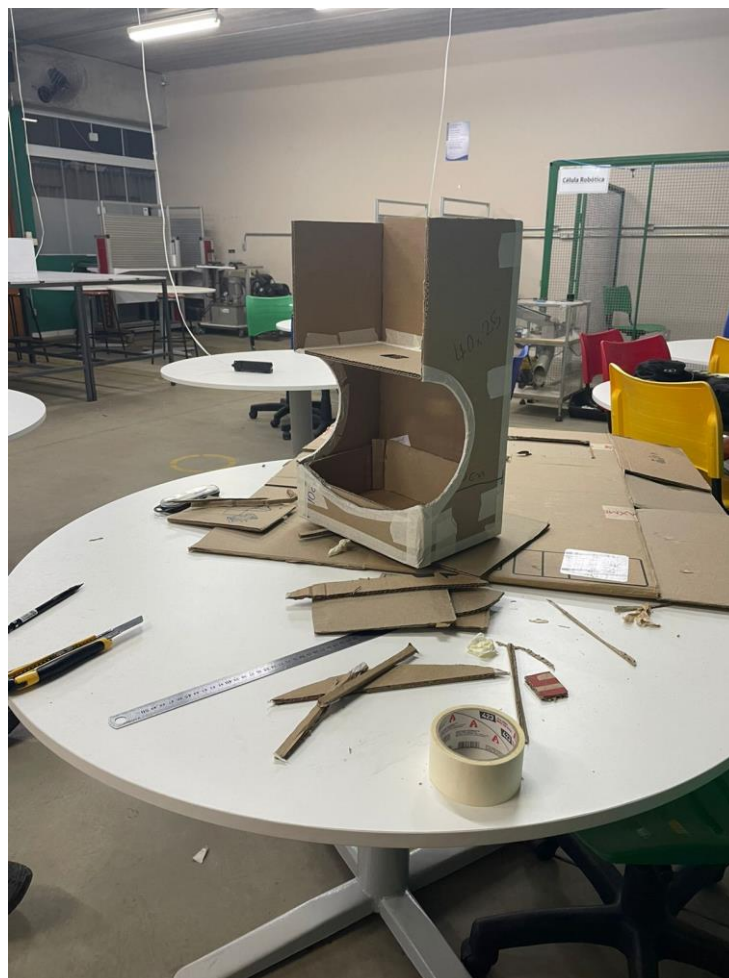


Figura 3 - Protótipo e prova de conceito

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS



Figura 4 - Tela HOME do aplicativo

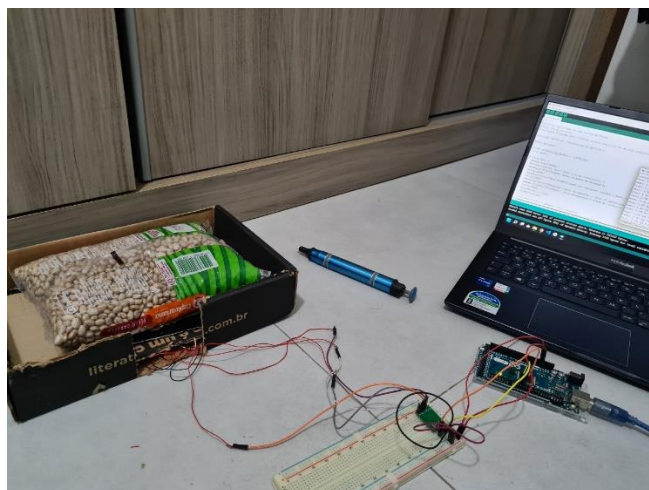
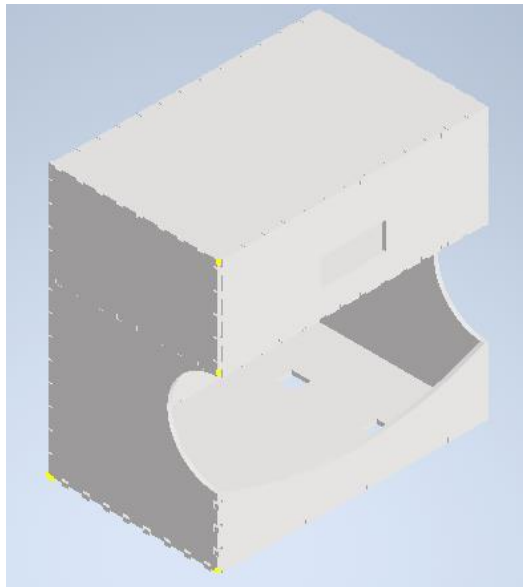
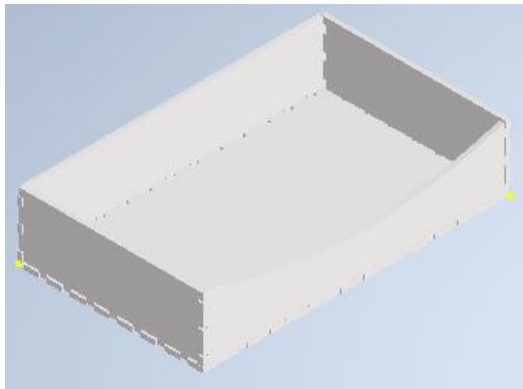


Figura 5 - Protótipo da balança



*Figura 6 - Montagem do projeto via software*



*Figura 7 - Montagem da balança e recipiente para ração*

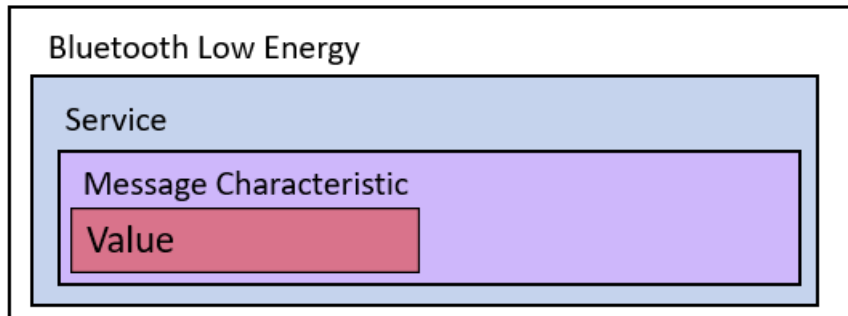
## **7. Atividades teóricas realizadas:** (individual)

(Abaixo, descrever as ideias, sugestões e informações levantadas por você, no momento do desenvolvimento da ideia do projeto)

Durante a fase de desenvolvimento do projeto foram levantados por mim vários pontos que seriam importantes para o desenvolvimento das partes de programação e eletrônica do alimentador. Entre elas os componentes necessários, como eles deveriam funcionar em conjunto, as especificações de hardware e software, as arquiteturas e linguagens de programação que deveriam ser utilizadas.

Primeiramente defini que o aplicativo deveria ser construído do zero utilizando React Native, um framework de TypeScript que serve para a criação de aplicativos nativos tanto para Android quanto para iOS. Utilizando React Native seria possível personalizar tanto a interface quanto as funcionalidades e performance do aplicativo, deixando a criação mais personalizável e bonita.

Após isso, defini os componentes eletrônicos que deveriam ser utilizados. Para a comunicação com o aplicativo seria necessário utilizar uma comunicação bluetooth que consumisse pouca energia e possuísse uma arquitetura de comunicação mais performática e que permitisse a comunicação com o aplicativo, para esse propósito utilizei um BLE (Bluetooth Low Energy), e como queria processar os dados no momento que chegassem ao módulo do alimentador, utilizei um ESP32, uma placa que tem embutido um modulo BLE juntamente com um microcontrolador programável, sendo basicamente um Arduino separado. O protocolo utilizado para a comunicação foi o GATT profile (Generic Attribute Profile), e sua arquitetura segue abaixo:



*Figura 8 - Arquitetura BLE*

Nela temos a estrutura especificada pelo protocolo GATT profile, onde o bluetooth se comporta como um servidor, podendo possuir vários serviços, esses serviços possuiriam diversas características e essas características possuiriam um único valor. Com identificadores únicos dados aos serviços e a características poderíamos acessá-los individualmente, podendo ler ou escrever em cada valor. No caso, utilizaríamos apenas um serviço possuindo apenas uma característica.

Levantei também a questão sobre armazenamento de dados que deveriam ficar guardados mesmo que a energia fosse desligada. Seria possível armazenar os dados na

EEPROM do Arduino, porém esse método possui diversas falhas, sendo muito ruim de catalogar os dados, pois eles são guardados utilizando endereços hexadecimais na EEPROM e se fossem modificados por engano algum dado sensível, poderia corromper o programa em execução, causando falhas inesperadas. Portanto, decidi utilizar um modulo de cartão SD, esse modulo possui um sistema de comunicação que possibilita a criação de sistemas personalizados de arquivos e em formatos diversos, além de ser totalmente desacoplado do sistema de controle, resolvendo assim todos os problemas que tínhamos.



Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

**8. Atividades práticas realizadas:** (individual)

(Descrever as atividades práticas realizadas por você até o momento do preenchimento, na sequência, quais suas próximas atividades até a conclusão do projeto. Imagens valorizarão a avaliação)

Já realizado

No momento boa parte das atividades propostas foram realizadas, em relação ao aplicativo foram realizadas as seguintes: Criação da interface do aplicativo, funcionalidades de criação, edição e remoção de horários, padrão de comunicação entre o aplicativo e o BLE, build do aplicativo para o celular, atualizações de compatibilidade. Em relação ao bluetooth e comunicação: foi escrita a programação para levantar o servidor e suas características, foram feitos os testes de recebimento de dados, definição dos limites de dados enviados por requisição bem como o algoritmo para prevenir a perda de dados, tratamento dos dados recebidos e formatação para os dados enviados. Quanto a balança, foi montado um protótipo para testes de funcionalidade e prova de conceito, foi feito o programa para calibração da balança, foram feitos todos os testes e montagem do circuito.

Para conferir os códigos feitos tanto do Arduino quanto do aplicativo segue a página no GitHub: <https://github.com/henrique-pignatari/snaizenFeeder>. Os arquivos do Arduino se encontram na pasta SnaizenFeeder e os do aplicativo na pasta SnaizenFeederApp. Esse repositório é constantemente atualizado com as versões mais recentes dos programas feitos. Para uma visão mais detalhada das etapas de cada tarefa, ver o Notion: <https://www.notion.so/henriquepignatari/SnaizenFeeder-d5929af1c35a43c48f3f4f918243612e>.

Ação	Descrição	Tempo decorrido (horas)
Programação do aplicativo	Programar a interface do usuário juntamente com as funcionalidades de criar, remover e editar os horários e pesos definidos (Aprox. 1500 linhas de código)	120
Definição das rotinas do Arduino Mega	Definir como quais as sequências que o Arduino deve seguir para realizar todas as tarefas necessárias para o funcionamento do alimentador	2
Estabelecimento da comunicação ESP32 e aplicativo	Estabelecer comunicação do aplicativo com ESP32, passar dados pré estabelecidos, testar envio e recebimento nos dois pontos	15
Criação dos protocolos de comunicação	Criar os padrões de comunicação, a formatação que cada dispositivo deve fazer	3
Prototipagem balança	Montagem de um protótipo para testar as balanças	3
Calibragem da balança	Programa para calibrar a balança e descobrir o valor da escala dela	5
Total:		148

Figura 9 - Tabela de atividades e tempo de execução



#### Próximas atividades

Após concluir essas etapas os próximos passos são principalmente integrar as partes já feitas, por exemplo, estabelecer a comunicação entre o ESP32 e o Arduino, conectar ambos os dispositivos ao modulo de cartão SD, programar o modulo de relógio que irá sincronizar os horários, estabelecer as rotinas de conferência de horários, integrar as rotinas de horários e peso, criar a programação de detecção do animal e verificação da quantidade de ração comida. Porém, muitos desses passos estão aguardando o material solicitado (cartão SD e modulo “*real time clock*”).

Tenho como objetivos opcionais/adicionais, melhorias na interface do aplicativo, validação de dados da criação de horários, refatoração do código do Arduino para aumentar a performance.

#### Imagens:

```
void initiateBLE(){
    // Create the BLE Device
    BLEDevice::init("SnaizenFeeder");

    // Create the BLE Server
    pServer = BLEDevice::createServer();
    pServer->setCallbacks(new MyServerCallbacks());

    // Create the BLE Service
    BLEService *pService = pServer->createService(SERVICE_UUID);

    delay(100);

    // Create a BLE Characteristic
    message_characteristic = pService->createCharacteristic(
        MESSAGE_CHARACTERISTIC_UUID,
        BLECharacteristic::PROPERTY_READ |
        BLECharacteristic::PROPERTY_WRITE |
        BLECharacteristic::PROPERTY_NOTIFY |
        BLECharacteristic::PROPERTY_INDICATE);

    // Start the BLE service
    pService->start();

    // Start advertising
    pServer->getAdvertising()->start();

    message_characteristic->setValue("");
    message_characteristic->setCallbacks(new CharacteristicsCallbacks());

    Serial.println("Waiting for a client connection to notify...");
}
```

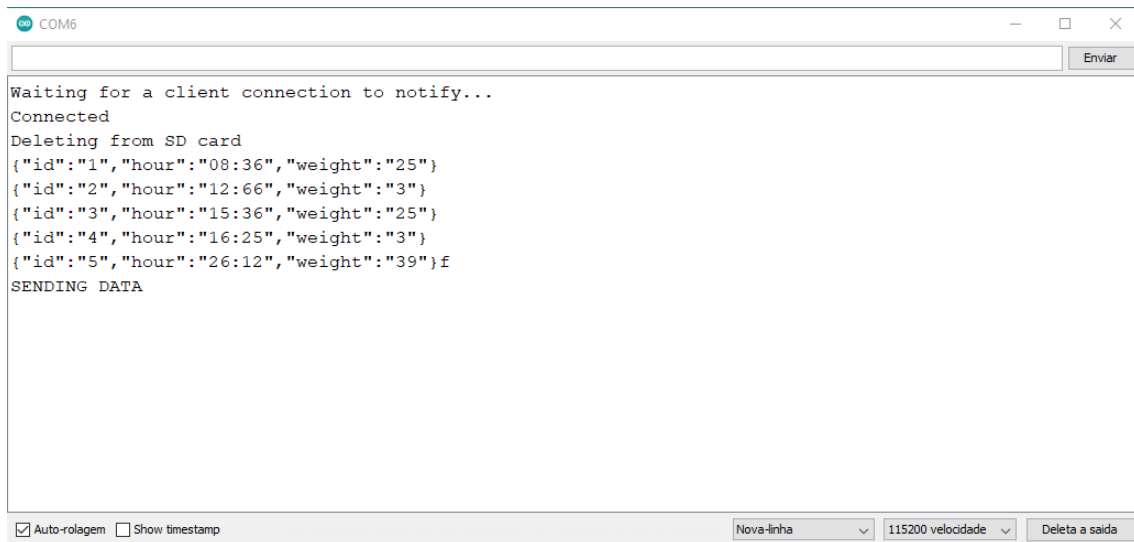
Figura 10 - Programa ESP32 inicialização bluetooth

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

```
class CharacteristicsCallbacks : public BLECharacteristicCallbacks
{
    void onWrite(BLECharacteristic *pCharacteristic)
    {
        if(pCharacteristic == message_characteristic){
            String value = pCharacteristic->getValue().c_str();

            if(value.startsWith("a")){
                String subString = value.substring(1);
                String line = "";
                if(subString.startsWith("d")){
                    Serial.println("Deleting from SD card");
                    subString = subString.substring(1);
                }
                if(!subString.endsWith("f")){
                    line = "l";
                }
                subString.concat(line);
                subString.replace(" ", {"", "("});
                Serial.println(subString);
            }else if(value.startsWith("b")){
                Serial.println("SENDING DATA");
                message_characteristic->setValue("3.5");
                message_characteristic->notify();
            }
        }
    }
};
```

Figura 11 - Call-backs das características



```
Waiting for a client connection to notify...
Connected
Deleting from SD card
{"id":"1","hour":"08:36","weight":"25"}
{"id":"2","hour":"12:66","weight":"3"}
{"id":"3","hour":"15:36","weight":"25"}
{"id":"4","hour":"16:25","weight":"3"}
{"id":"5","hour":"26:12","weight":"39"}f
SENDING DATA
```

Figura 12 - Saída do programa do ESP32 recebendo e enviando dados

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

```
void setup() {  
  Serial.begin(9600);  
  balanca.begin(3, 2); // SCK= pino 2 e DT= pino 3  
  Serial.println("Remova todos os pesos da balança");  
  int seconds = 5;  
  for(int i = 0; i<seconds; i++){  
    Serial.println(seconds-i);  
    delay(1000);  
  }  
  Serial.println("Pressione + para incrementar o fator de calibração");  
  Serial.println("Pressione - para decrementar o fator de calibração");  
  delay(1000);  
  balanca.set_scale();  
  balanca.tare();  
  
  long zero_factor = balanca.read_average();  
}  
  
void loop() {  
  
  balanca.set_scale(calibration_factor);  
  
  Serial.print("Peso: ");  
  peso = balanca.get_units(10);  
  if (peso < 0)  
  {  
    peso = 0.00;  
  }  
  Serial.print(peso);  
  Serial.print(" kg");  
  Serial.print(" Fator de calibração: ");  
  Serial.print(calibration_factor);  
  Serial.println();  
  delay(500);  
  
  if (Serial.available())  
  {  
    char temp = Serial.read();  
    if(temp == '+')  
      calibration_factor += 1;  
    else if(temp == '-')  
      calibration_factor -= 1;  
  }  
}
```

Figura 13 - Programa calibragem célula de carga

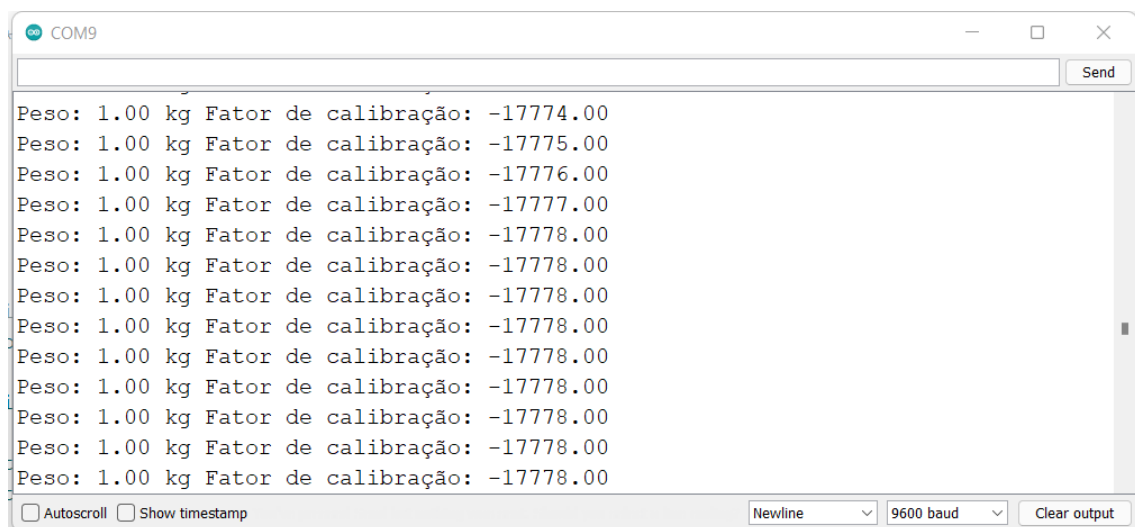


Figura 14 - Saída do programa da balança calibrada

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

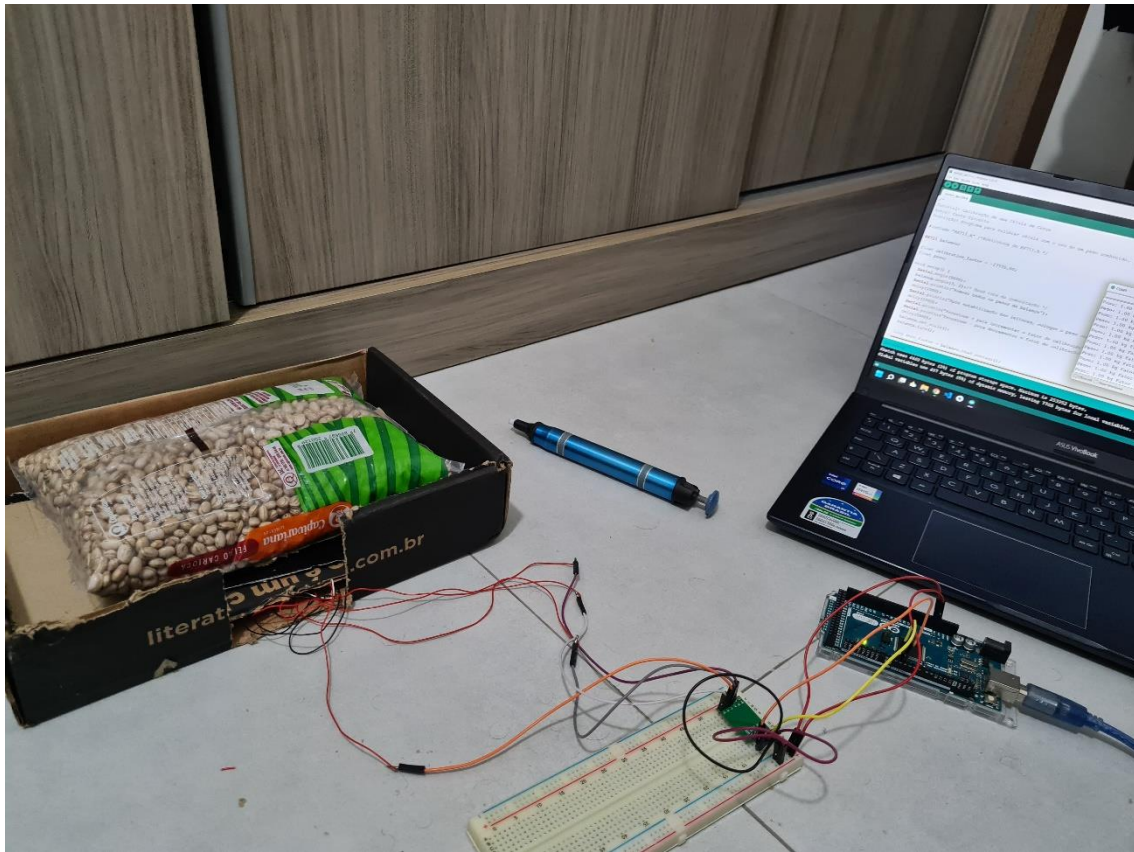


Figura 15 - Teste protótipo balança

Relatório individual do projeto temático  
Avaliação 1º Bimestre  
Projeto Temático 7ºS

