

**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
GRADUAÇÃO EM ENGENHARIA AGRÍCOLA E AMBIENTAL**

EDSON LUIZ DA SILVA JUNIOR

RELATÓRIO DE ESTÁGIO SUPERVISIONADO OBRIGATÓRIO

**RECIFE
2022.2**

EDSON LUIZ DA SILVA JUNIOR

RELATÓRIO DE ESTÁGIO SUPERVISIONADO OBRIGATÓRIO

**ANÁLISE E MONITORAMENTO DA TEMPERATURA E UMIDADE
DO MILHO EM SILO PILOTO UTILIZANDO ARDUINO**

Relatório apresentado ao Curso de Graduação em Engenharia Agrícola e Ambiental da Universidade Federal Rural de Pernambuco Pernambuco, como pré-requisito para obtenção de nota da disciplina Estágio Supervisionado Obrigatório, sob orientação do Professor Renato Laurenti

**RECIFE
2022.2**

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

S237m Silva Junior, Edson Luiz da
Monitoramento da temperatura e da umidade do milho em silo
piloto utilizando arduino / Edson Luiz da Silva Junior. - 2023.
f.: il.

Orientador: Renato Laurenti.

Trabalho de Conclusão de Curso (Graduação em Engenharia
Agrícola e Ambiental) – Universidade Federal Rural de
Pernambuco, Departamento de Engenharia Agrícola, Recife,
BR-PE, 2023.

Inclui referências.

1. Grãos - Armazenamento 2. Silos 3. Temperatura 4. Umidade
5. Automação 6. Agricultura 7. Arduino 8. Milho I. Laurenti, Renato,
orient. II. Título

CDD

RELATÓRIO DE ESTÁGIO SUPERVISIONADO OBRIGATÓRIO

ANÁLISE E MONITORAMENTO DA TEMPERATURA E UMIDADE DO MILHO EM SILO PILOTO UTILIZANDO *ARDUINO*

Edson Luiz da Silva Junior
Nome e assinatura do aluno

Renato Laurenti
Nome e assinatura do orientador

Renato Laurenti
Nome e assinatura do supervisor (se for o caso)

AGRADECIMENTOS

Agradeço primeiramente a Deus pela vida, pela direção e pelo o conhecimento. Sou grato ao Professor Renato Laurenti e também ao grupo do laboratório de propriedades físicas dos materiais e armazenamento agrícola pela a oportunidade de mais aprendizado e pelo companheirismo.

LISTA DE FIGURAS

Figura 1- Arduino Uno

Figura 2 - sensor de temperatura ds18b20

Figura 3 – sensor de temperatura e umidade DHT22

Figura 4 - Leitura da temperatura do sensor ds18b20 no laptop

Figura 5 - esquema teste com arduino uno e sensor ds18b20.

Figura 6 - esquema de teste com arduino uno e sensor dht22

Figura 7 - Leitura dos valores de umidade e temperatura do sensor dht22

Figura 8 – Silo Piloto do laboratório de armazenamento

SUMÁRIO

1. INTRODUÇÃO.	7
2. O MILHO.	7
3. ARMAZENAMENTO DE GRÃOS.	8
4. AUTOMAÇÃO.	9
5. ARDUÍNO.	9
6. OBJETIVOS.	10
6.1. OBJETIVO GERAL.	10
6.2. OBJETIVOS ESPECÍFICOS.	10
7. REFERENCIAL TEÓRICO.	10
7.1. ARMAZENAMENTO	10
7.2 TIPOS DE UNIDADES ARMAZENADORAS	10
7.3 TEMPERATURA.	11
7.4 UMIDADE	12
7.5.MICROCONTROLADORES.	12
8. METODOLOGIA.	13
9. ÁREA DE ESTUDO	17
10. ATIVIDADES DESENVOLVIDAS.	17
11. CONCLUSÕES/ CONSIDERAÇÕES FINAIS	18
12. REFERÊNCIAS BIBLIOGRÁFICAS.	19
13.ANEXO - CÓDIGOS.	19

1. INTRODUÇÃO

O Brasil, atualmente, é o quarto maior produtor de grãos do mundo, sendo que, com relação ao milho, ocupa a terceira posição, ficando atrás apenas de países como China e Estados Unidos, posição essa não alterada nas últimas décadas. Outro ponto importante é quanto ao seu posicionamento de destaque no comércio internacional como exportador de grãos, reflexo de sua grande produção. (SENAR,2018).

O milho é o segundo grão mais exportado, perdendo apenas para a soja, fazendo com que o Brasil ocupe a segunda posição nos países que mais exportam grãos. (ARAGÃO; CONTINI, 2021). Além disso, o milho é o principal componente para a produção de ração animal, voltado para a produção de carne animal, segmento esse, com grande destaque no comércio exterior. (SOUZA et al.,2018)

2. O MILHO

O cultivo do milho nos últimos anos, o torna um dos cereais mais expressivos da atualidade, ocupando a terceira posição, com, aproximadamente,140 milhões de hectares semeados no mundo. A produção de milho no Brasil tem como característica a realização do plantio em duas épocas, primeira safra e segunda safra. A chamada safrinha representa, dentre as duas safras, a maior produção de milho no Brasil, com a vantagem de ser um cultivo que se dá em sucessão à colheita da soja. (FRIAS; LIMA; TOMAZ, 2019)

Grãos de milho são produzidos em duas safras e carecem de armazenamento durante o restante do período do ano para atender à demanda; entretanto, muitas vezes por déficit de armazenamento ou mesmo por falta de informações, os grãos acabam sendo armazenados de forma incorreta, em condições inadequadas, o que acaba comprometendo a qualidade do produto. A temperatura é um dos principais fatores que interferem na qualidade de armazenamento de grãos (Rehman et al., 2002; Reed et al., 2007; Park et al., 2012) Não somente pensando no aumento de produtividade no campo,

deve-se levar em consideração o aprimoramento tanto do processo de colheita, quanto o de armazenamento dos grãos. Uma vantagem dos 27 grãos é que, é possível que sejam armazenados por um longo período, sem perdas significativas da qualidade. Porém, o armazenamento prolongado só pode ser realizado quando são seguidas corretamente as práticas de colheita, limpeza, transporte, secagem, armazenamento, combate a insetos e prevenção de fungos. (EMBRAPA, 2015).

3. ARMAZENAMENTO DE GRÃOS

O armazenamento é uma etapa de suma importância na cadeia de produção agrícola, pois tem um grande reflexo no custo e afeta diretamente a qualidade do produto que chega à mesa do consumidor. Um dos grandes problemas enfrentados pelo Brasil, em relação à produção de grãos, é o baixo índice de unidades armazenadoras localizadas nas fazendas. Isso gera uma demanda excessiva do setor de transportes (CONAB, 2006), elevando o custo do produto final. Para Zago (2020), a armazenagem de grãos é a forma mais utilizada pelos produtores para conservação e melhor manejo da produção gerada, além de se obter um maior controle, e aguardar o melhor momento de comercialização, ou seja, o mais lucrativo, possibilitando menores perdas na produção. Pode-se dizer que, a armazenagem de grãos nada mais é um processo que compreende a preservação dos grãos desde a colheita até a sua distribuição, passando pelos processos de higienização, secagem, procedimento de defensivo agrícola, carregamento, distribuição, entre outros. De acordo com SENAR (2018), para se determinar qual a unidade armazenadora a ser utilizada, deve-se levar em conta o volume a ser armazenado, bem como os recursos financeiros disponíveis para a sua construção. Além disso é muito importante analisar alguns aspectos como, o custo de instalação e operação, qual o tipo de produto a ser armazenado, quais serão os fatores técnicos e operacionais e qual será a localização.

4. AUTOMAÇÃO

Atualmente a automação, que consiste em uma tecnologia aplicada em sistemas eletrônicos, elétricos ou ainda mecânicos atrelados a meios computacionais, para um maior controle dos sistemas produtivos, vem sendo cada vez mais utilizadas seja para locais onde a presença humana corra algum risco ou ainda para a otimização dos processos. Devido à necessidade de armazenagem e conservação dos grãos, se faz necessárias algumas técnicas de controle, como o uso da termometria, que consiste no controle da temperatura da massa de grãos, e do uso da aeração, que consiste na passagem forçada de ar através da massa de grãos, a fim de prevenir, ou ainda solucionar problemas relacionados à conservação dos grãos armazenados. Oliveira (2014)

5 . ARDUÍNO

Arduino é uma plataforma eletrônica de software e hardware aberto utilizada para a prototipação de projetos, que vem sendo muito utilizado atualmente pela sua facilidade de uso e pelo seu baixo custo. **Arduino** possui pinos de entrada e saída, os de entrada recebem informações (dados) de sensores ligados ao mesmo e os de saída recebem dados do **arduino** para atuadores. Entre a entrada e a saída da plataforma há o micro controlador que faz o controle dos dados de entrada e saída. O micro controlador é programado na linguagem de programação arduino pela IDE de mesmo nome. **IDE** é o software responsável pela escrita do programa e transferência do mesmo para seu micro controlador.

6. OBJETIVOS

6.1. OBJETIVO GERAL

Analisar e comparar dois diferentes modelos de sensores de temperatura, utilizando a plataforma eletrônica Arduino, tendo como finalidade o monitoramento automatizado de um silo piloto 24 horas por dia, para conservação dos grãos de milho.

6.2. OBJETIVOS ESPECÍFICOS

- Pesquisar melhores sensores de temperatura e umidade;
- Montar os sensores no silo;
- Programar micro controlador;
- Monitorar valores no aplicativo e na *IDE*;

7. REFERENCIAL TEÓRICO

7.1. ARMAZENAMENTO

Armazenar é guardar e conservar o produto, diminuindo ao máximo suas perdas, utilizando as melhores técnicas existentes. Para isso existe a rede armazenadora de grãos que é o sistema de armazenagem que permite receber as quantidades produzidas durante as safras agrícolas, retendo a produção que não é comercializada imediatamente, e mantendo suas características tanto qualitativas quanto quantitativas, para posteriormente serem consumidas ou comercializadas. (LORINI et al., 2002).

7.2 TIPOS DE UNIDADES ARMAZENADORAS

Dentre os diferentes tipos de armazenamento, destaca-se o sistema hermético, que se enquadra como um método de controle físico, sem a utilização de

produtos químicos no controle dos agentes prejudiciais ao armazenamento. Este sistema impede a troca gasosa entre o interior e o exterior do compartimento onde são armazenados os grãos. Nesse ambiente, os organismos vivos que compõem o ecossistema (grãos, insetos e microrganismos) consomem o oxigênio (O₂) por meio de processo respiratório e liberam gás carbônico (CO₂) e água (H₂O) e diminuem a concentração de O₂ a níveis letais ou restritivos ao desenvolvimento dos seres vivos associados (Rupollo et al., 2004). Diversos autores têm estudado a qualidade de grãos durante o armazenamento hermético em diversas estruturas, dentre eles; Weinberg et al. (2008) em grãos de milho; Faroni et al. (2009) em grãos de soja. Porém, em tonéis metálicos, voltados para armazenamento de pequenos e médios produtores de milho, não existem estudos relatando a qualidade.

Umidade e temperatura são duas grandezas que estão muito relacionadas quando o assunto é climatização e armazenamento.

7.3 TEMPERATURA

A temperatura é um dos principais fatores que interferem na qualidade de armazenamento de grãos (Rehman et al., 2002; Reed et al., 2007; Park et al., 2012), sendo que nos últimos anos vem crescendo a utilização da tecnologia de resfriamento artificial em grãos, com o objetivo de preservar a qualidade do produto por períodos mais elevados e reduzir a deterioração dos grãos.

Porém poucos trabalhos foram realizados com grãos de milho.

O monitoramento da temperatura da massa granular é importante para que ações preventivas sejam adotadas no sentido de evitar o aquecimento dos grãos e com isso garantir a integridade dos mesmos. Em muitos casos, o monitoramento é realizado de forma manual: onde uma pessoa responsável pela manutenção do sistema de armazenamento dirige-se ao silo empunhando um terminal, que é plugado a tomadas para realizar a leitura dos sensores de temperatura que ficam dispostos em diferentes alturas. Essa informação é transcrita em uma planilha, para posterior análise e tomada de decisão. O processo de aquisição de dados está facilitado pela popularização dos sistemas microcontrolados e com o acesso a sensores eletrônicos que dispensam calibração, além da facilidade do acoplamento desses dispositivos a sistemas de comunicação sem fio. Todas essas condições

permitem a criação de sistemas de monitoramento de temperatura interna em silos com custo atraente e com informação das condições de temperatura e umidade relativa em tempo real, facilitando e otimizando tanto o trabalho do operador, como permitindo o registro dos dados. O sistema contará com uma placa microcontrolada do tipo Arduíno, sensores de temperatura eletrônicos, um sistema de comunicação sem fio e um software 22 especialmente desenvolvido para apresentação das informações na tela de um computador para acompanhamento e o arquivamento.

7.4 UMIDADE

O teor de umidade de um produto exerce grande influência na variação de temperatura de uma massa de grãos durante a armazenagem (Abbouda et al., 1992; Khankari et al., 1994). Por isso, recomenda-se a secagem e limpeza do produto antes do seu armazenamento, pois esses procedimentos diminuem a ação da microflora na massa de grãos, retardando a velocidade das trocas gasosas e o processo de deterioração natural. Devido a mudanças climáticas, variações de temperatura também contribuem para a formação de gradientes de temperatura em uma massa de grãos armazenada. Temperaturas diferentes nos ambientes interno e externo de um silo provocam correntes de ar na massa de grãos, que podem induzir a migração de umidade das áreas de altas temperaturas para as de baixas temperaturas. A migração de umidade pode potencializar o desenvolvimento de insetos, fungos e bactérias e iniciar a deterioração do produto (Converse et al., 1973; Brooker et al., 1992; Chang et al., 1993; Casada & Young, 1994; Chang & Steele, 1995; Gong et al., 1995).

7.5. MICROCONTROLADORES

Microcontroladores Componentes eletrônicos construídos em pastilhas de silício, onde milhões de transistores são dispostos de forma que em sua arquitetura sejam implementados diversos recursos apropriados para realizar tarefas sequenciais e facilitarem o trabalho de aquisição de sinais. Atualmente, o microcontrolador pode ser encontrado e aplicado como um componente em uma plataforma de eletrônica embarcada, ou em plataformas de prototipagem que

ajudam o desenvolvimento de sistemas microcontrolados, sem que o projetista seja um especialista em eletrônica. A plataforma mais popular do momento é o Arduino, composto de uma placa de prototipagem, que pode receber diversos acessórios, além de uma IDE (Integrated Develop Enviroment) que permite ao usuário escrever as funções que serão executadas em linguagem de programação própria e transferir esse programa para a memória do sistema.

8. METODOLOGIA

A plataforma Arduino Uno foi escolhida para desenvolver o equipamento por ser de fácil aquisição, programação simples e por atender os requisitos de hardware, facilitando a execução do protótipo. O Arduino Uno (figura 1) é o elemento intermediário no sistema. Ele é o responsável por receber as informações dos sensores escolhidos, o de temperatura ds18b20 (figura 2) e o sensor de umidade e temperatura DHT 22 (figura 3) e enviar ao Laptop, onde a informação da temperatura e umidade são apresentada ao usuário e armazenadas. Por meio do Arduino também é possível acionar remotamente.

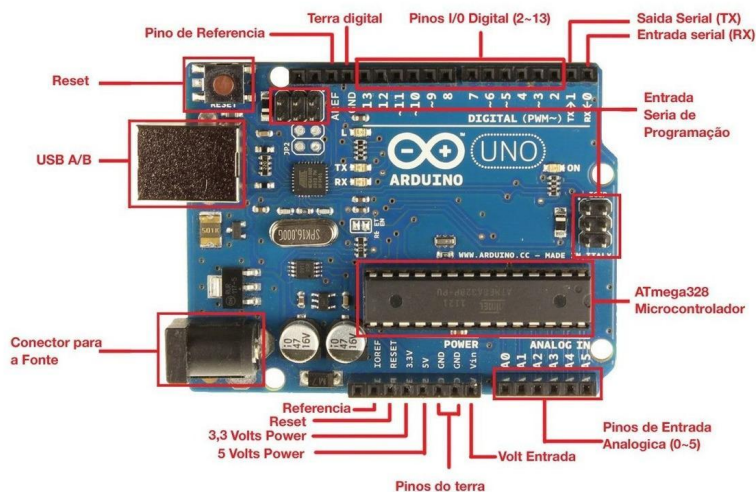


Figura1- Arduino Uno



Figura 2 - Sensor de temperatura ds18b20 mostrando seu encapsulamento e seu cabo de alimentação (VCC), terra (GND) e dados (Data).



Figura 3 – sensor de temperatura e umidade DHT22

A metodologia inicial foi o teste dos sensores nos diferentes código que estão no anexo 13.

Foi verificado o funcionamento dos códigos e dos sensores pelas leituras no laptop conforme as figuras 4 e 7.

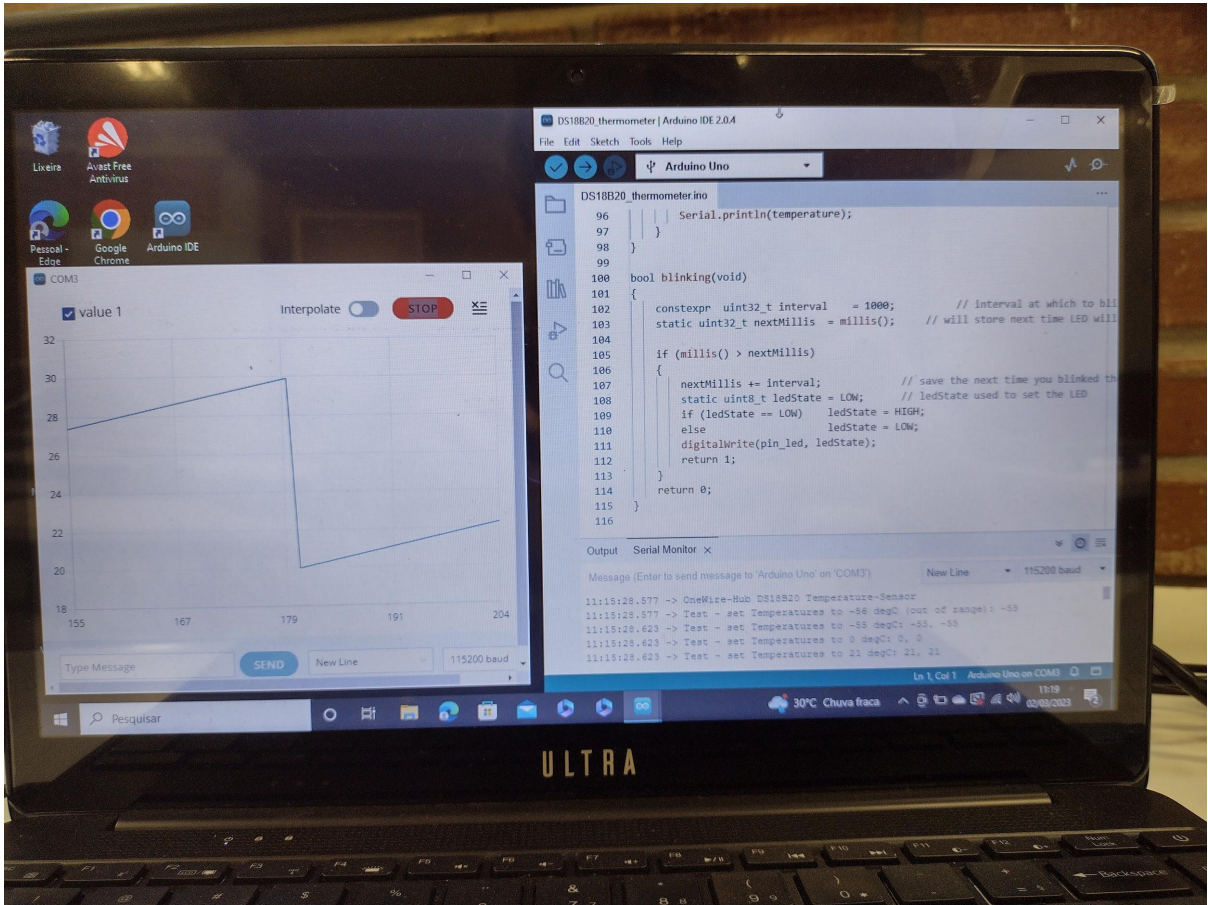


Figura 4 - Leitura da temperatura do sensor ds18b20 no laptop

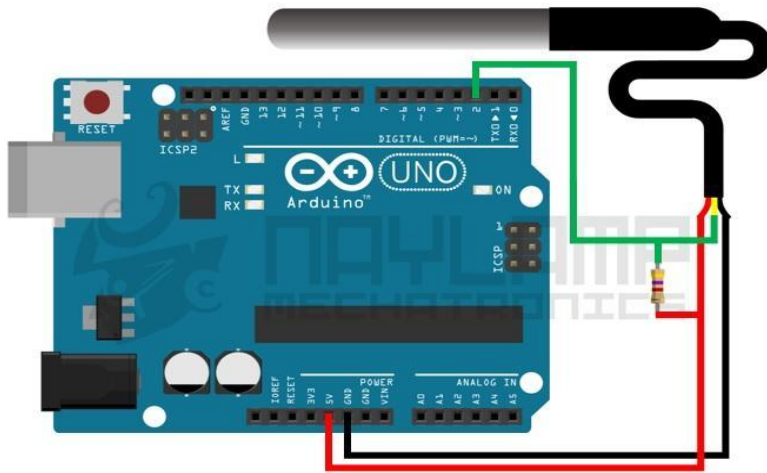


Figura 5 - esquema teste com arduino uno e sensor ds18b20.

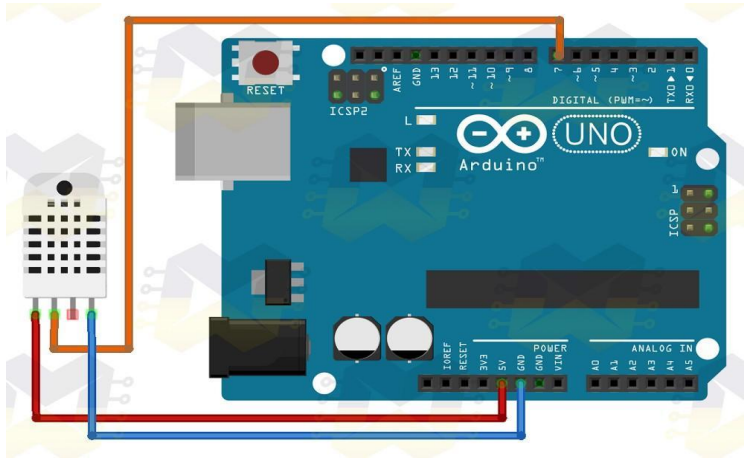


Figura 6 - esquema de teste com arduíno uno e sensor dht22

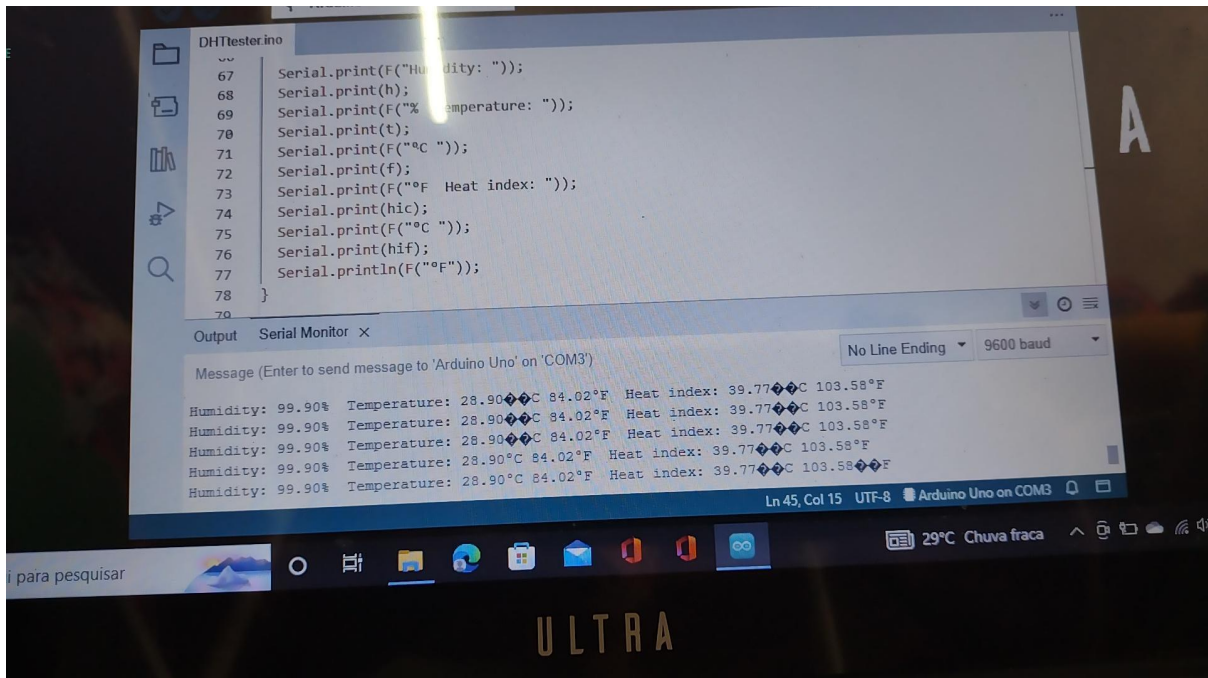


Figura 7 - Leitura dos valores de umidade e temperatura do sensor dht22

O silo utilizado para serve para fins didáticos possui as dimensões: altura útil de 85 cm, diâmetro 50 cm. O mesmo é feito de aço , como mostra figura .



Figura 8 – Silo Piloto do laboratório de armazenamento

9. ÁREA DE ESTUDO

O trabalho foi realizado no laboratório de propriedades físicas dos materiais e armazenagem agrícola, na Universidade Federal Rural de Pernambuco.

10. ATIVIDADES DESENVOLVIDAS

Inicialmente foram realizadas reuniões com o grupo de pesquisa para traçar o direcionamento do projeto.

Foram desenvolvidas pesquisas na literatura com a finalidade de encontrar os melhores componentes a serem utilizados no projeto.

Durante este intervalo foram executadas programações na IDE Arduino.

Com a obtenção dos códigos que é utilizado para carregar a placa Arduino Uno (microcontrolador), foi realizados testes para verificação da funcionalidade dos códigos.

Posteriormente houve a realização de testes nos sensores existentes, entre eles, dez unidades dos sensores ds18b20 e duas unidades do sensor DHT22.

Foi verificado que entre os sensores ds18b20 existentes, oito deles estavam apresentando falha no funcionamento e só duas unidade estavam funcionando porém sem calibração.

Neste ínterim foi executada também a limpeza e desinfecção do silo piloto .

11. CONCLUSÕES / CONSIDERAÇÕES FINAIS

Mesmo ainda em desenvolvimento, o trabalho fornece informações importantes sobre a viabilidade de monitoramento remoto do milho em silo metálico utilizando o microcontrolador arduino, pois esse possui grande versatilidade e precisão na obtenção dos parâmetros desejados (temperatura e umidade) e sendo de custo reduzido.

Também podemos chegar a conclusão que o ideal seria combinação dos dois tipos de sensores testados, pois o ds18b20 é um sensor de temperatura por contato com a massa do grão tendo como vantagem sua robustez e versatilidade, já o sensor DHT 22 é um sensor de temperatura e umidade do ar, tendo como vantagem sua precisão e baixo

custo, e como desvantagem sua fragilidade, sendo necessário seu encapsulamento. Podemos também concluir que a vida útil destes sensores são aparentemente curta, tendo necessidade periódicas de manutenções e substituições dos componentes utilizados, por consequência da sensibilidade dos equipamentos.

Outra verificação para continuidade do projeto é a necessidade de substituição dos sensores por sensores novos para realização do monitoramento no silo piloto.

12. REFERÊNCIAS BIBLIOGRÁFICAS

<https://www.scielo.br/j/rbeaa/a/J4nw69XDYh4SDJfRrtWQLZk/?format=pdf&lang=pt>
<https://repositorio.ufsm.br/bitstream/handle/1/9605/DISSERTACAO%20COMPLETA.pdf?sequence=1&isAllowed=y>
<http://biblioteca.emater.tche.br:8080/pergamumweb/vinculos/000005/000005e0.pdf>
<https://www.scielo.br/j/rbeaa/a/VpMDpzMPscz34JVpfGBDryN/?lang=pt&format=pdf>
https://repositorio.unesp.br/bitstream/handle/11449/242520/mendes_ra_me_botfca.pdf?sequence=5&isAllowed=y
https://repositorio.unesp.br/bitstream/handle/11449/191039/bica_mrr_me_botfca.pdf?sequence=3&isAllowed=y

13. ANEXO - CÓDIGOS

código e do sensor ds18b20

```
/*
 * Example-Code that emulates a DS18B20
 *
 * Tested with:
 * - https://github.com/PaulStoffregen/OneWire --> DS18x20-Example,
  atmega328@16MHz as Slave
 * - DS9490R-Master, atmega328@16MHz and teensy3.2@96MHz as Slave
 */

#include "OneWireHub.h"
#include "DS18B20.h" // Digital Thermometer, 12bit

constexpr uint8_t pin_led      { 13 };
constexpr uint8_t pin_onewire { 8 };

auto hub = OneWireHub(pin_onewire);

auto ds18b20 = DS18B20(DS18B20::family_code, 0x00, 0x00, 0xB2, 0x18, 0xDA,
0x00); // DS18B20: 9-12bit, -55 - +85 degC
auto ds18s20 = DS18B20(0x10, 0x00, 0x00, 0xA2, 0x18, 0xDA, 0x00);
// DS18S20: 9 bit, -55 - +85 degC
auto ds1822 = DS18B20(0x22, 0x00, 0x00, 0x22, 0x18, 0xDA, 0x00);
// DS1822: 9-12bit, -55 - +125 degC
```

```

bool blinking(void);

void setup()
{
  Serial.begin(115200);
  Serial.println("OneWire-Hub DS18B20 Temperature-Sensor");
  Serial.flush();

  pinMode(pin_led, OUTPUT);

  // Setup OneWire
  hub.attach(ds18b20);
  hub.attach(ds18s20);
  hub.attach(ds1822);

  // Test-Cases: the following code is just to show basic functions, can be
  removed any time
  Serial.print("Test - set Temperatures to -56 degC (out of range): ");
  ds18b20.setTemperature(int8_t(-56));
  Serial.println(ds18b20.getTemperature());

  Serial.print("Test - set Temperatures to -55 degC: ");
  ds18b20.setTemperature(int8_t(-55));
  ds18s20.setTemperature(int8_t(-55));
  Serial.print(ds18b20.getTemperature());
  Serial.print(", ");
  Serial.println(ds18s20.getTemperature()); // ds18s20 is limited to signed
  9bit, so it could behave different

  Serial.print("Test - set Temperatures to 0 degC: ");
  ds18b20.setTemperature(int8_t(0));
  ds18s20.setTemperature(int8_t(0));
  Serial.print(ds18b20.getTemperature());
  Serial.print(", ");
  Serial.println(ds18s20.getTemperature());

  Serial.print("Test - set Temperatures to 21 degC: ");
  const int8_t temperature = 21;
  ds18b20.setTemperature(temperature);
  ds18s20.setTemperature(temperature);
  ds1822.setTemperature(temperature);
  Serial.print(ds18b20.getTemperature());
  Serial.print(", ");
  Serial.println(ds18s20.getTemperature());

  Serial.print("Test - set Temperatures to 85 degC: ");

```

```

ds18b20.setTemperature(int8_t(85));
ds18s20.setTemperature(int8_t(85));
Serial.print(ds18b20.getTemperature());
Serial.print(", ");
Serial.println(ds18s20.getTemperature());

Serial.print("Test - set Temperatures to 126 degC (out of range): ");
ds1822.setTemperature(int8_t(126));
Serial.println(ds1822.getTemperature());

Serial.println("config done");
}

void loop()
{
  // following function must be called periodically
  hub.poll();
  // this part is just for debugging (USE_SERIAL_DEBUG in OneWire.h must be
  enabled for output)
  if (hub.hasError()) hub.printError();

  // Blink triggers the state-change
  if (blinking())
  {
    // Set temp
    static float temperature = 20.0;
    temperature += 0.1;
    if (temperature > 30.0) temperature = 20.0;
    ds18b20.setTemperature(temperature);
    ds18s20.setTemperature(temperature);
    ds1822.setTemperature(temperature);
    Serial.println(temperature);
  }
}

bool blinking(void)
{
  constexpr uint32_t interval = 1000; // interval at which to
  blink (milliseconds)
  static uint32_t nextMillis = millis(); // will store next time LED will
  updated

  if (millis() > nextMillis)
  {
    nextMillis += interval; // save the next time you blinked the
  LED
}

```

```

    static uint8_t ledState = LOW;    // ledState used to set the LED
    if (ledState == LOW)    ledState = HIGH;
    else                    ledState = LOW;
    digitalWrite(pin_led, ledState);
    return 1;
}
return 0;
}

```

código do DHT 22 – teste biblioteca arduino

```

#include <DHT.h>
#include <DHT_U.h>

// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

// REQUIRES the following Arduino libraries:
// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library
// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor

#include "DHT.h"

#define DHTPIN 5 // Digital pin connected to the DHT sensor
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whatever type you're using!
// #define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 3 (on the right) of the sensor to GROUND (if your sensor has 3
pins)
// Connect pin 4 (on the right) of the sensor to GROUND and leave the pin 3
EMPTY (if your sensor has 4 pins)
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter
to

```

```

// tweak the timings for faster processors.  This parameter is no longer
needed
// as the current DHT reading algorithm adjusts itself to work on faster
procs.
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
}

```



```

Serial.println(F("°F"));
}

```

código do ds18b20 em série

```

/*****
*****
* Serie sensores - ds18b20 temperatura
*/

// *****Bibliotecas*****
#include <DallasTemperature.h>
#include <OneWire.h>

// ***** Instancias *****
OneWire oneWire (12); // Protocolo One Wire
DallasTemperature sensors (&oneWire); // sensores

// ***** Variaveis *****
DeviceAddress ds18; // Sensor
byte c;

//*****Funções *****
void sensorID(DeviceAddress sensor) {
// Exibe ID do sensor
for (uint8_t i = 0; i < 8; i++) {
if (sensor [i] < 16)
Serial.print ("0");
Serial.print (sensor[i], HEX);
}
}
void alarm(uint8_t* sensor){
// Tratamento de Alarme
Serial.print ("Alarme no sensor");
sensorID(sensor);
Serial.println();
}
// *****Setup *****
void setup(void) {
/* Serial */
Serial.begin (115200);

/* Inicializa sensores */
sensors.begin();
sensors.setResolution(12);

```

```

sensors.setAlarmHandler(&alarm);

c = sensors.getDeviceCount();
Serial.print("DS18B20 identificados: ");
Serial.println(c);

// Define alarmes
for (byte i = 0; i < c; i++) {
sensors.getAddress(ds18, i);
sensors.setLowAlarmTemp(ds18, 5);
sensors.setHighAlarmTemp(ds18, 35);
}
}
//***** Loop *****
void loop() {
// Solicita Temperaturas
sensors.requestTemperatures();

// Exibe dados
Serial.println("n# Serial Resolução AlmBaixo Alalto temp");
for (byte i = 0 ; i <c; i++) {
if (sensors.getAddress(ds18, i)){
Serial.print(" Falha lendo sensor ");
Serial.println(i);
} else {
Serial.print(i);
Serial.print("/t");
sensorID(ds18);
Serial.print("/t");
Serial.print(sensors.getResolution(ds18), DEC);
Serial.print("/t");
Serial.print(sensors.getLowAlarmTemp(ds18), DEC);
Serial.print("/t");
Serial.print(sensors.getHighAlarmTemp(ds18), DEC);
Serial.print("/t");
Serial.println(sensors.getTempC(ds18), 1);

}
}
// Verifica ocorrencia de alarne
sensors.processAlarms();
// Intervalo de 10s
delay(10000);

}

```

