Relatório Técnico

Nº Grupo: 09
Nome dos integrantes: Caio Godinho, Larissa, Gustavo, Vinicius, Samara, Erick
Turma: 1CCOK

Tema do projeto: Sistema de Monitoramento de Queimadas em Propriedades Rurais
Sensor: Sensor capacitivo de umidade do solo

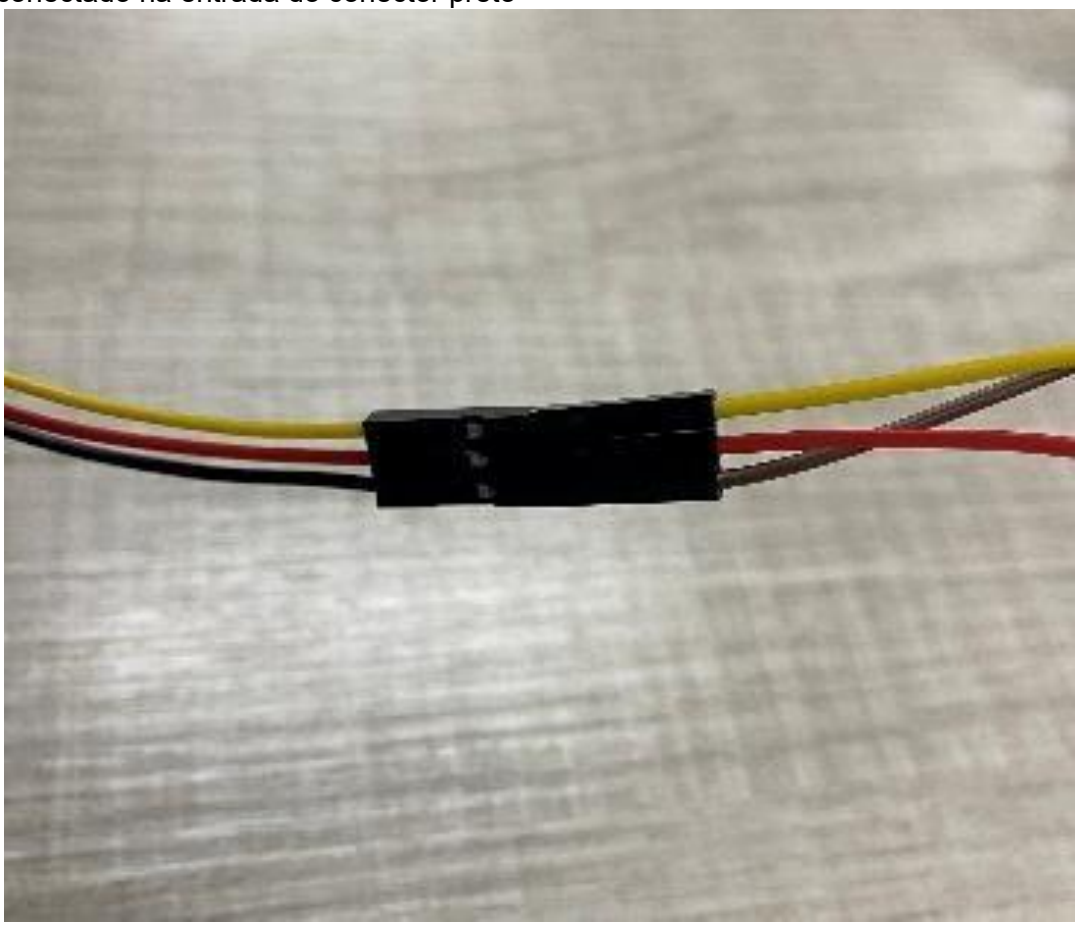
Introdução

O monitoramento da umidade do solo é essencial para a prevenção de queimadas em propriedades rurais. O sistema se baseia na integração de três elementos: o Sensor de Umidade de Solo Capacitivo, que mede o teor de água e gera um sinal de tensão; a placa Arduíno UNO R3, que recebe essa tensão, a converte em um valor digital usando o ADC, calibra os dados e os empacota em formato JSON, e a API, que recebe e valida o pacote de dados via protocolo HTTP (POST). A API, por sua vez, executa um INSERT para registrar os dados no Banco de Dados, criando um registro em série que permite a rápida identificação de quedas críticas na umidade do solo e o acionamento de alertas preventivos contra a seca extrema.

Arquitetura de Montagem do Sensor

1- Conectamos os fios preto, vermelho e amarelo em suas respectivas ordens estabelecidas pelo grupo para melhor memorização, ligando a conexão do fio preto com o GND do sensor, em seguida a conexão do fio vermelho com o VCC do sensor e por fim a conexão do fio amarelo com o AUOT do sensor.

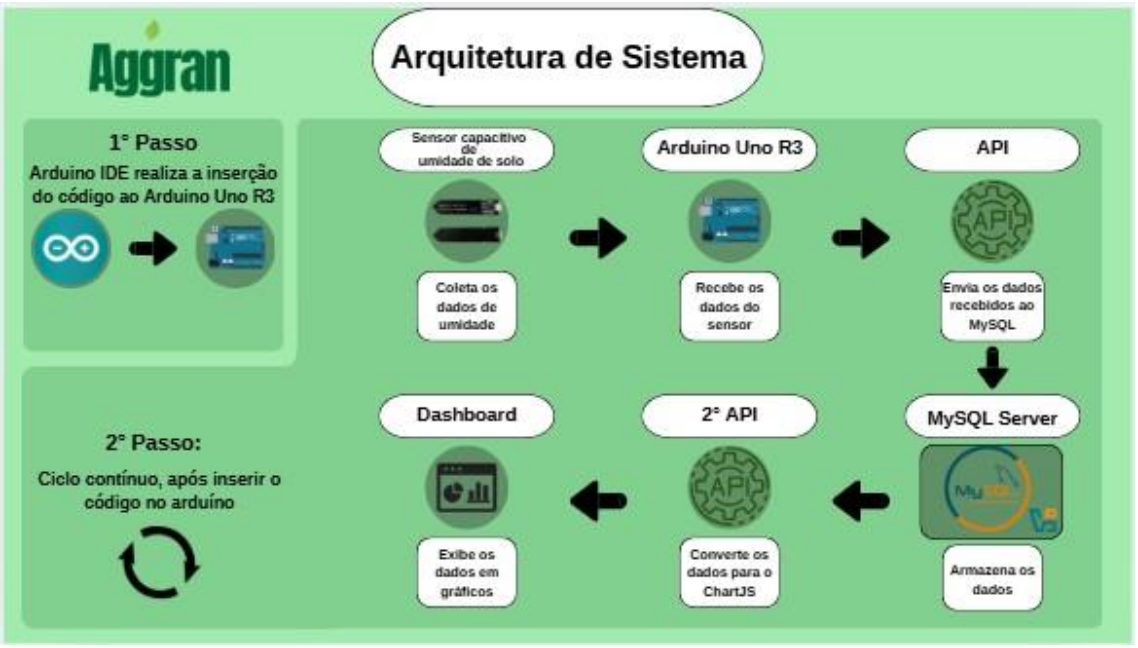
2-) Em seguida conectamos os jumpers em seus devidos espaços separados pelas respectivas cores, sendo jumper amarelo na conexão do fio amarelo e jumper vermelho na conexão do fio vermelho, com exceção do jumper marrom que é conectado na entrada do conector preto



3-) Em seguida conectamos os jumpers macho-macho nas devidas portas analógicas e de energia do Arduino Uno, sendo jumper vermelho na entrada de 5V, jumper marrom na entrada do GND e por fim jumper amarelo na porta analógica A5 definida como a porta utilizada no projeto

4-) Por fim conectamos a alimentação do Arduino via USB no computador.

Arquitetura do Sistema



Código do Projeto

Os dados obtidos pelo sensor de umidade do solo serão interpretados a partir do código inserido no Arduíno a partir do Arduíno IDE. Com base nisso, para a visualização do código, abaixo consta o código e em verde a divisão do código em blocos (variáveis, configuração e execução) e a descrição:

```
// Bloco de definição das variáveis
const int ValorAr = 520; const int
ValorAgua = 200;

int ValorUmidadeSolo = 0; float
porcentagemUmidade = 0;

int random_variable; int
static_variable = 100; int
static_variable2 = 20; int
potentiometer;

// Bloco de definição das configurações  void
setup() {
Serial.begin(9600); // definir a taxa de transferência dos dados em 9600
}

// Bloco de execução
void loop() {
ValorUmidadeSolo = analogRead(A5);
```

```
int faixa = ValorAr - ValorAgua;

int distancia = ValorAr - ValorUmidadeSolo;

porcentagemUmidade = (float)distancia / faixa * 100.0;

if (porcentagemUmidade < 0) porcentagemUmidade = 0; if
(porcentagemUmidade > 100) porcentagemUmidade = 100;

Serial.println(porcentagemUmidade);


delay(1500);

}
```

A partir do momento em que os dados são capturados pelo sensor e interpretados pelo código presente no arduino, será realizado o armazenamento dos dados no banco de dados MySQL Server a partir da API, criada por Marise Miranda com contribuições de Eduardo Verri e Matheus Matos.

Diante disso, o primeiro bloco do código tem como finalidade importar as bibliotecas express, serialport, tal como o mysql que será utilizado para acessar e realizar as operações dentro do banco de dados:

```
const serialport = require('serialport'); const
express = require('express'); const mysql =
require('mysql2');
```

Posteriormente, é realizado as definições das variáveis da taxa de transferência dos dados em 9600, sendo a mesma taxa utilizada no código do arduino descrito anteriormente. Além de definir a variável da porta do servidor em 3300.

```
const SERIAL_BAUD_RATE = 9600; const
SERVIDOR_PORTA = 3300;
```

Além disso, é definido a variável HABILITAR_OPERACAO_INSERIR que quando estiver em true, permitirá que os dados sejam inseridos no banco de dados.

```
const HABILITAR_OPERACAO_INSERIR = true;
```

Posteriormente é criado uma função assíncrona o qual irá definir as informações necessárias para acessar o banco de dados. Por exemplo, o banco de dados será acessado na máquina local com o usuário ‘aluno’, e a senha necessária para acessar é ‘sptech’. Dessa forma, será posteriormente acessado o banco de dados aggran. Sendo ainda, utilizado no Mysql Server a porta 3306 (definida por padrão), para que ocorra a comunicação com o banco de dados. Em seguida é realizado tratativas de erros, tal como, caso não seja encontrado as portas do arduino UNO. Após inserir nas variáveis a porta e baud rate (definidas nas variáveis inicialmente), é realizado o processamento dados recebidos do arduino, o qual irá utilizar com o delimitador ponto e vírgula (;) e os dados serão transformados de string para float, visto que, o modelo do banco de dados é projetado para receber números decimais com até 2 casas decimais.

Em seguida, é realizado a chamada para o banco de dados para executar a inserção dos dados capturados pelo sensor. Dessa forma, os dados serão inseridos na tabela registro no campo umidadeSolo. Conforme esses dados são inseridos no banco de dados aggran (definido anteriormente), será emitido no terminal a mensagem que o dado foi inserido no banco. E em caso de erro, será emitido um alerta de erro.

```
const serial = async (
        valoresSensorDigital,
) => {

        let poolBancoDados = mysql.createPool(
          {
                host: 'localhost',
        user: 'aluno',
        password: 'sptech',
        database: 'aggran',
        port: 3306
          }
        ).promise();
 const   portas   =   await   serialport.SerialPort.list();   const   portaArduino   =
        portas.find((porta) => porta.vendorId == 2341 &&
         porta.productId == 43);
        if (!portaArduino) {
                  throw new Error('O arduino não foi encontrado em nenhuma porta
         serial');
        }

        const arduino = new serialport.SerialPort(
          {
                path: portaArduino.path,
                baudRate: SERIAL_BAUD_RATE
          }
```

```
        );

        arduino.on('open', () => {            console.log(`A leitura do
        arduino foi iniciada na porta
        ${portaArduino.path} utilizando Baud Rate de
        ${SERIAL_BAUD_RATE}`);
        });

        arduino.pipe(new serialport.ReadlineParser({ delimiter: '\r\n' })).on('data',
        async (data) => {  console.log(data);  const valores = data.split(';');  const
        sensorDigital                    =                    parseFloat(valores[0]);
        valoresSensorDigital.push(sensorDigital);                                    if
        (HABILITAR_OPERACAO_INSERIR) {

                await poolBancoDados.execute(
                        'INSERT INTO registro (umidadeSolo) VALUES (?)',
                        [sensorDigital]
                );
                console.log("valores inseridos no banco: ", sensorDigital);
        }

         });

        arduino.on('error', (mensagem) => {      console.error(`Erro no
        arduino (Mensagem: ${mensagem}`)
        });


}
```

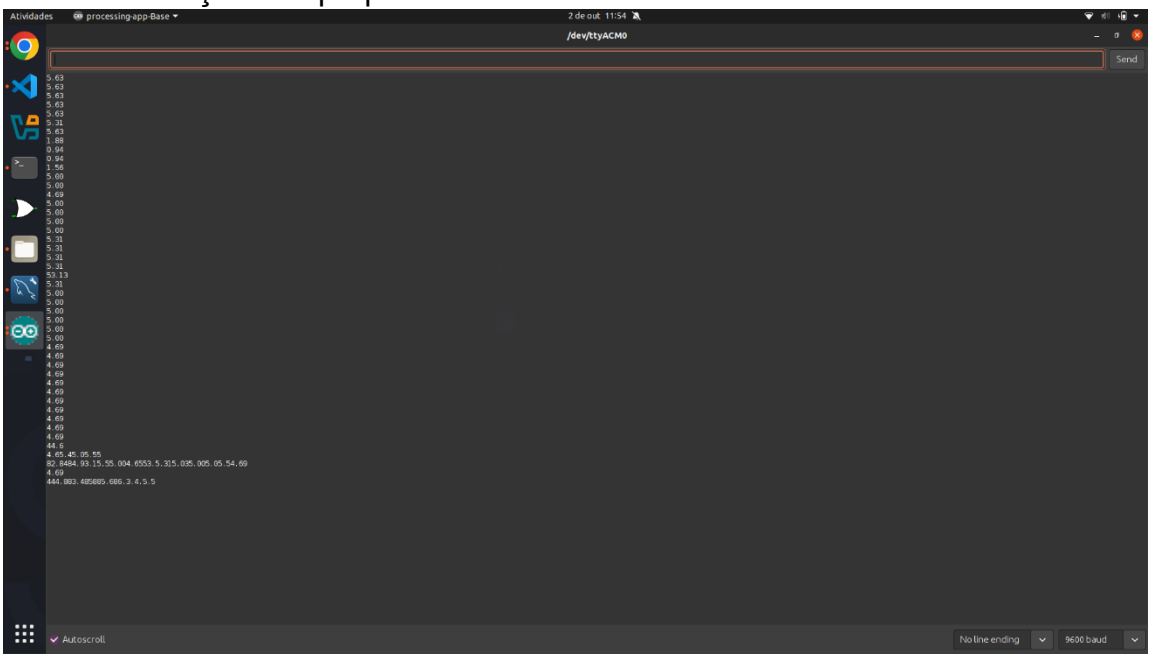
Por fim, contém o código que irá estabelecer a conexão com o servidor web, o qual permite com que a API será executada via navegador. Além de criar a função assíncrona que realiza a transferência dos dados capturados (armazenados na variável valoresSensorDigital em formato de lista) para o serviço web.

```
const servidor = (    valoresSensorDigital
) => {
        const app = express();      app.use((request, response,
next) => {              response.header('Access-Control-Allow-
Origin', '*');
```

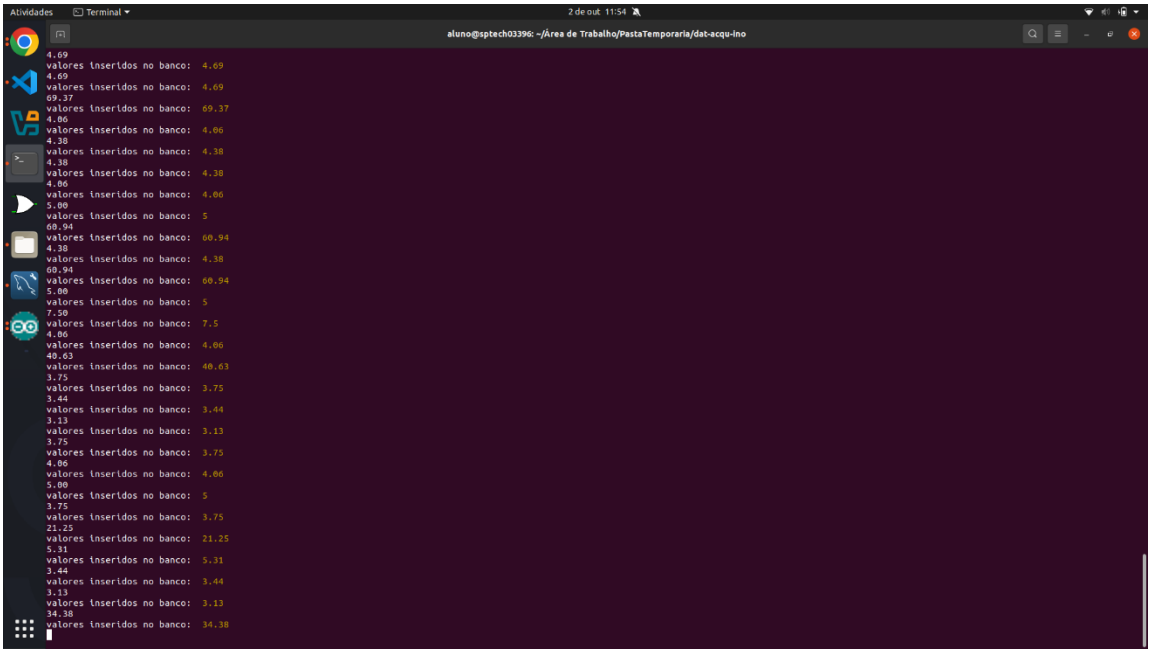
```
                response.header('Access-Control-Allow-Headers', 'Origin,
                Content-Type, Accept');
                next();
        });
        app.listen(SERVIDOR_PORTA, () => {
                console.log(`API executada com sucesso na porta
        ${SERVIDOR_PORTA}`);
        });

        app.get('/sensores/digital', (_, response) => {
                return response.json(valoresSensorDigital);
        });
}



(async () => {

        const valoresSensorDigital = [];

        await serial(
        valoresSensorDigital
        );

        servidor(
        valoresSensorDigital
        );
})();
```

Resultados Iniciais

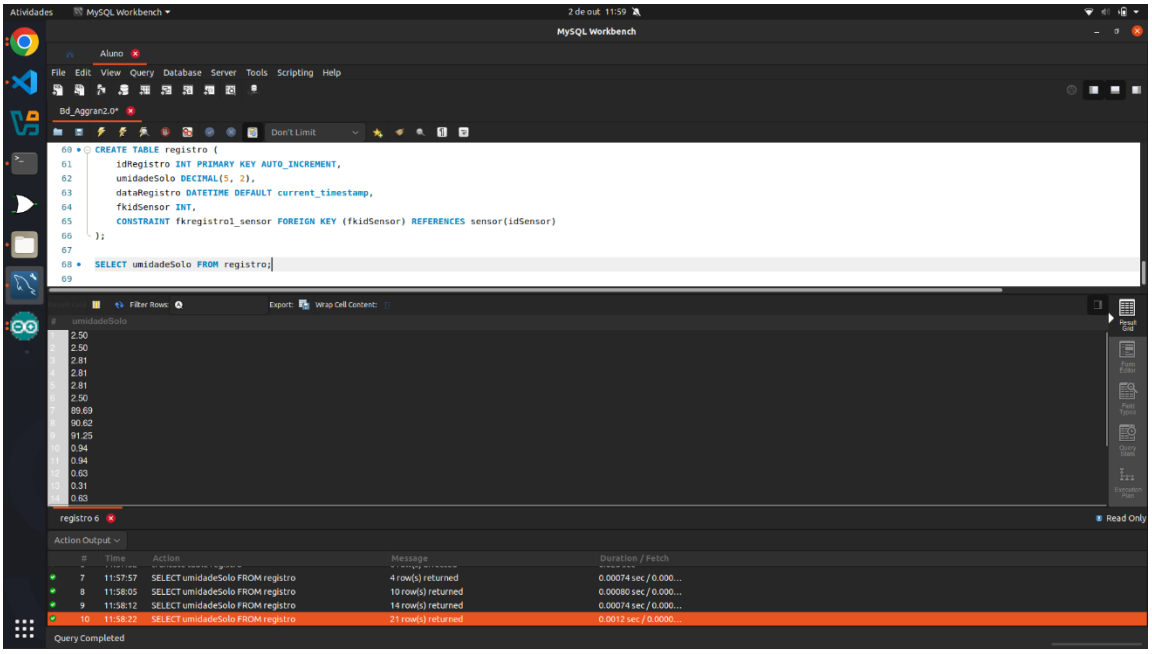
Print 1: Como resultado inicial, conseguimos captar os dados de umidade do solo feitos pelo sensor recebidos pelo arduino, através do arduino IDE, nos permitindo uma visualização dos próprios dados.



Print 2: Com esse dados, conseguimos aplicar a API de forma eficaz, captando os dados disponibilizados pelo sensor e alocá-los em seu devido campo, na tabela registro do banco de dados, armazenando cada dado coletado pelo sensor.



Print 3: Além disso, conseguimos visualizar os registros feitos pela API em ordem crescente no banco de dados, utilizando o comando SELECT.



Print 4: Por fim, conseguimos executar o gráfico dinâmico com os dados captados pelo sensor e recebidos pelo arduino, através de um programa HTML, com finalidade de melhor visualizar a porcentagem de umidade do solo.