

Grupo 10: Isabella Neves, Lucas B. Jorge, Mariana Namie, Paula Maria, Sofia Valadares, Victor Daniel

1. a) Criando as tabelas específicas do grupo e
b) Dando privilégios ao usuário criado 'insertGrupo10'

```
mysql> CREATE DATABASE CottonAnalytics;
Query OK, 1 row affected (0.01 sec)

mysql> USE CottonAnalytics;
Database changed
mysql> CREATE TABLE dadosSensor (
  -> id INT PRIMARY KEY AUTO_INCREMENT,
  -> temp1 DOUBLE,
  -> temp2 DOUBLE,
  -> temp3 DOUBLE,
  -> temp4 DOUBLE,
  -> temp5 DOUBLE,
  -> umid1 DOUBLE,
  -> umid2 DOUBLE,
  -> umid3 DOUBLE,
  -> umid4 DOUBLE,
  -> umid5 DOUBLE,
  -> dtHora DATETIME DEFAULT (NOW())
  -> );
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'insertGrupo10'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'insertGrupo10'@'localhost';
```

2. Tabela de inserção dos dados e
3. Dados inseridos na tabela através do sensor arduíno

```
mysql> SELECT * FROM dadosSensor;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | temp1 | temp2 | temp3 | temp4 | temp5 | umid1 | umid2 | umid3 | umid4 | umid5 | dtHora          |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | 21.79 | 13.07 | 37.47 | 42.26 | 4.79  | 8.74  | 5.24  | 15.03 | 16.95 | 1.92  | 2022-10-18 15:03:24 |
| 2  | 21.79 | 13.07 | 37.47 | 42.26 | 4.79  | 8.74  | 5.24  | 15.03 | 16.95 | 1.92  | 2022-10-18 15:03:26 |
| 3  | 21.79 | 13.07 | 37.47 | 42.26 | 4.79  | 7.11  | 4.26  | 12.22 | 13.78 | 1.56  | 2022-10-18 15:03:31 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

4. Ajustar a API para as credenciais do usuário criado e ajustar para os dados serem recebidos

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

const SERIAL_BAUD_RATE = 9600;
const SERVIDOR_PORTA = 3000;
const HABILITAR_OPERACAO_INSERIR = true;

const serial = async (
  valoresTemp1,
  valoresTemp2,
  valoresTemp3,
  valoresTemp4,
  valoresTemp5,
  valoresUmid1,
  valoresUmid2,
  valoresUmid3,
  valoresUmid4,
  valoresUmid5
) => {
  const poolBancoDados = mysql.createPool(
    {
      host: 'localhost',
      port: 3306,
      user: 'insertGrupo10',
      password: 'grupo10thebest',
      database: 'cottonanalytics'
    }
  ).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) => {
    const valores = data.split(';');
    const temp1 = parseFloat(valores[0]);
    const temp2 = parseFloat(valores[1]);
    const temp3 = parseFloat(valores[2]);
    const temp4 = parseFloat(valores[3]);
    const temp5 = parseFloat(valores[4]);
    const umid1 = parseFloat(valores[5]);
    const umid2 = parseFloat(valores[6]);
    const umid3 = parseFloat(valores[7]);
    const umid4 = parseFloat(valores[8]);
    const umid5 = parseFloat(valores[9]);

```

```

        valoresTemp1.push(temp1);
        valoresTemp2.push(temp2);
        valoresTemp3.push(temp3);
        valoresTemp4.push(temp4);
        valoresTemp5.push(temp5);
        valoresUmid1.push(umid1);
        valoresUmid2.push(umid2);
        valoresUmid3.push(umid3);
        valoresUmid4.push(umid4);
        valoresUmid5.push(umid5);

        if (HABILITAR_OPERACAO_INSERIR) {
            await poolBancoDados.execute(
                'INSERT INTO dadosSensor (temp1, temp2, temp3, temp4, temp5, umid1, umid2, umid3, umid4, umid5) VALUES
                (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)',
                [temp1, temp2, temp3, temp4, temp5, umid1, umid2, umid3, umid4, umid5]
            );
            console.log("valores inseridos no banco: ", temp1 + ", " + temp2 + ", " + temp3 + ", " + temp4 + ", " +
            temp5 + ", " + umid1 + ", " + umid2 + ", " + umid3 + ", " + umid4 + ", " + umid5)
        }
    });
    arduino.on('error', (mensagem) => {
        console.error("Erro no arduino (Mensagem: ${mensagem})");
    });
}

```

```

const servidor = (
  valoresTemp1,
  valoresTemp2,
  valoresTemp3,
  valoresTemp4,
  valoresTemp5,
  valoresUmid1,
  valoresUmid2,
  valoresUmid3,
  valoresUmid4,
  valoresUmid5
) => {
  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/dht11/temp1', (_, response) => {
    return response.json(valoresTemp1);
  });
  app.get('/sensores/dht11/temp2', (_, response) => {
    return response.json(valoresTemp2);
  });
  app.get('/sensores/dht11/temp3', (_, response) => {
    return response.json(valoresTemp3);
  });
  app.get('/sensores/dht11/temp4', (_, response) => {
    return response.json(valoresTemp4);
  });

```

Ln 28, Col 39

```

    app.get('/sensores/dht11/temp5', (_, response) => {
        return response.json(valoresTemp5);
    });
    app.get('/sensores/dht11/umid1', (_, response) => {
        return response.json(valoresUmid1);
    });
    app.get('/sensores/dht11/umid2', (_, response) => {
        return response.json(valoresUmid2);
    });
    app.get('/sensores/dht11/umid3', (_, response) => {
        return response.json(valoresUmid3);
    });
    app.get('/sensores/dht11/umid4', (_, response) => {
        return response.json(valoresUmid4);
    });
    app.get('/sensores/dht11/umid5', (_, response) => {
        return response.json(valoresUmid5);
    });
}

(async () => {
    const valoresTemp1 = [];
    const valoresTemp2 = [];
    const valoresTemp3 = [];
    const valoresTemp4 = [];
    const valoresTemp5 = [];
    const valoresUmid1 = [];
    const valoresUmid2 = [];
    const valoresUmid3 = [];
    const valoresUmid4 = [];
    const valoresUmid5 = [];
    await serial(

```

```

await serial(
  valoresTemp1,
  valoresTemp2,
  valoresTemp3,
  valoresTemp4,
  valoresTemp5,
  valoresUmid1,
  valoresUmid2,
  valoresUmid3,
  valoresUmid4,
  valoresUmid5
);
servidor(
  valoresTemp1,
  valoresTemp2,
  valoresTemp3,
  valoresTemp4,
  valoresTemp5,
  valoresUmid1,
  valoresUmid2,
  valoresUmid3,
  valoresUmid4,
  valoresUmid5
);
})();

```

```

setInterval(() => {
  obterDados(valoresTemp1, '/dht11/temp1');
  obterDados(valoresTemp2, '/dht11/temp2');
  obterDados(valoresTemp3, '/dht11/temp3');
  obterDados(valoresTemp4, '/dht11/temp4');
  obterDados(valoresTemp5, '/dht11/temp5');
  obterDados(valoresUmid1, '/dht11/umid1');
  obterDados(valoresUmid2, '/dht11/umid2');
  obterDados(valoresUmid3, '/dht11/umid3');
  obterDados(valoresUmid4, '/dht11/umid4');
  obterDados(valoresUmid5, '/dht11/umid5');
}, 1000);

```

```

* Bloco do DHT11
*/
float umidade = dht_1.readHumidity();
float temperatura = dht_1.readTemperature();
//aplicando os dados da temperatura à função linear de acordo com os parâmetros do projeto
float temp1 = 1.27*temperatura-10.98;
float temp2 = temp1 * 0.60;
float temp3 = temp1 * 1.72;
float temp4 = temp1 * 1.94;
float temp5 = temp1 * 0.22;

//aplicando os dados da umidade à função linear de acordo com os parâmetros do projeto
float umid1 = (5/3.06)*umidade-105.64;
float umid2 = umid1 * 0.60;
float umid3 = umid1 * 1.72;
float umid4 = umid1 * 1.94;
float umid5 = umid1 * 0.22;

```

```

Serial.print(temp1);
Serial.print(";");
Serial.print(temp2);
Serial.print(";");
Serial.print(temp3);
Serial.print(";");
Serial.print(temp4);
Serial.print(";");
Serial.print(temp5);
Serial.print(";");

```

```

Serial.print(umid1);
Serial.print(";");
Serial.print(umid2);
Serial.print(";");
Serial.print(umid3);
Serial.print(";");
Serial.print(umid4);
Serial.print(";");
Serial.print(umid5);
Serial.println(";");

```