

NOME	TURMA
Izael Reis de Oliveira Junior	1ADSB

Arquitetura Computacional com API

Questões avaliação prática:

1. Inicie o servidor NODE.
 - Após a clonagem do repositório onde está a API configurada, dentro do repositório, abra o terminal (gitbash) e execute o comando: npm start

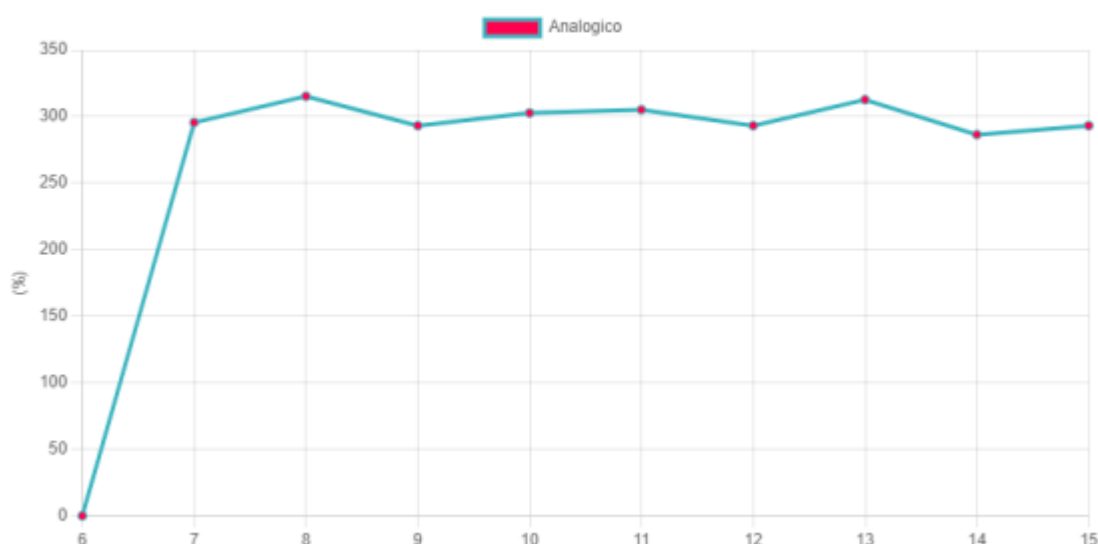
```
PS C:\Users\Master\OneDrive\Documentos\SPTECH CURSOS\Arquitetura Computacional\d
at-acqu-ino> npm start

> arduino-api@2.1.0 start
> node main.js

API executada com sucesso na porta 3300
A leitura do arduino foi iniciada na porta COM10 utilizando Baud Rate de 9600
0.00
0.00
0.00
0.00
0.05
0.00
0.00
295.61
315.35
293.00
```

2. Exiba os gráficos do projeto do grupo no index.html

Graphics



3. Qual comando é utilizado para instalar pacotes dos módulos NODE?

- npm install

```
PS C:\Users\Master\OneDrive\Documentos\SPTECH CURSOS\Arquitetura Computacional\dat-acqu-ino> npm install
```

4. Demonstre no main.js como as variáveis do servidor de serviço foram definidas.

```
// importa as bibliotecas necessários
const serialport = require('serialport');
const express = require('express');
const mysql = require('mysql2');

// constantes para configurações
const SERIAL_BAUD_RATE = 9600;
const SERVIDOR_PORTA = 3300;

// habilita ou desabilita a inserção de dados no banco de dados
const HABILITAR_OPERACAO_INSERIR = false;

// função para comunicação serial
const serial = async (
  valoresSensorAnalogico
) => {

  // conexão com o banco de dados MySQL
  let poolBancoDados = mysql.createPool({
    {
      host: 'localhost',
      user: 'root',
      password: 'Animig0r1@.',
      database: 'luminii',
      port: 3306
    }
  }).promise();

  // lista as portas seriais disponíveis e procura pelo Arduino
  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');
  }

  // configura a porta serial com o baud rate especificado
  const arduino = new serialport.SerialPort({
```

```
    // configura a porta serial com o baud rate especificado
    const arduino = new serialport.SerialPort({
      {
        path: portaArduino.path,
        baudRate: SERIAL_BAUD_RATE
      }
    });
```

```
const valores = data.split(';');
const sensorAnalogico = parseFloat(valores[0]);
```

```
// função para criar e configurar o servidor web
const servidor = (
  valoresSensorAnalogico
) => {
  const app = express();
```

```
const valoresSensorAnalogico = []
```

5. Qual é a distinção entre as portas 3300 e 3306?
- **3300:** É a porta do usb, é usada para receber os dados do Arduino.
 - **3306:** É a porta padrão do MySQL.
6. No index.html, como é feita a chamada à API externa chart.js? Onde é realizada essa chamada e onde um novo gráfico é criado?
- É realizada através da inserção da biblioteca do chart.js dentro do head do index.html. O novo gráfico é criado dentro da tag <canvas>, onde é criado uma tela em branco para que façamos a inserção do gráfico nela.

```
<!doctype html>
<html>

<head>
  <title>Graphics</title>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>

<body>

  <h1>Graphics</h1>
  <div style="display: flex;">
    <div style="width: 50%;">
      <canvas id="sensorAnalogico"></canvas>
    </div>
  </div>
```

```
var sensorAnalogico = new Chart(document.getElementById('sensorAnalogico').getContext('2d'), {
  type: 'line',
  data: {
    datasets: [{
      label: 'Analogico',
      borderColor: '#63B1BC',
      backgroundColor: '#ED145B'
    }]
  },
  options: {
    scales: {
      x: {
        beginAtZero: true
      },
      y: {
        title: {
          display: true,
          text: '(%)'
        },
        beginAtZero: true,
      },
    },
  },
});
```

7. Onde é possível ajustar o tamanho e o tipo de gráfico gerado?

- Dentro da tag div que está o canva, usando o atributo style, é possível estiliza-la, até mesmo alterar o tamanho.

```
<div style="display: flex;">
  <div style="width: 50%;">
    <canvas id="sensorAnalogico"></canvas>
  </div>
```

- Dentro das configurações e atribuições do gráfico, é possível alterar o tipo do gráfico no atributo type.

```
var sensorAnalogico = new Chart(document.getElementById('sensorAnalogico').getContext('2d'), {
  type: 'line',
  data: {
    datasets: [{
      label: 'Analogico',
      borderColor: '#63B18C',
      backgroundColor: '#ED145B'
    }]
  },
  options: {
```

8. O que é representado pelo método 'get' no código main.js?

O get atribui uma propriedade a uma função: nesse caso ele busca dados na API.

```
// função para criar e configurar o servidor web
const servidor = (
  SensorLumini
) => {
  const app = express();

  // configurações de requisição e resposta
  app.use((request, response, next) => {
    response.header('Access-Control-Allow-Origin', '*');
    response.header('Access-Control-Allow-Headers', 'Origin, Content-Type, Accept');
    next();
  });

  // inicia o servidor na porta especificada
  app.listen(SERVIDOR_PORTA, () => {
    console.log(`API executada com sucesso na porta ${SERVIDOR_PORTA}`);
  });

  // define os endpoints da API para cada tipo de sensor
  app.get('/sensores/analogico', (_, response) => {
    return response.json(SensorLumini);
  });
}
```

9. Por que é gerado um arquivo JSON e para que ele é utilizado?

- O arquivo JSON serve para armazenar e transmitir os dados, sendo possível enviar esses dados para onde for necessário.

10. Considerando que no código.ino a saída é:

- a. **DHTH_temp; DHTH_umid; Luminosidade; LM35_temp; chave**, explique como essa estrutura de dados (na forma de lista) é adicionada como um vetor na API Node. Demonstre como o código captura essa lista e a divide ordenadamente dentro de um vetor.
- Os dados coletados pelo sensor são inseridos em uma lista, e dentro desta lista os números são separados por ';', ou seja, após cada ';' é um dado sendo inserido em um vetor diferente

```
arduino.pipe(new serialport.ReadlineParser({ delimiter: '\r\n' })).on('data', async (data) => {  
  console.log(data);  
  const valores = data.split(';');  
  const sensorDigital = parseInt(valores[0]);  
  const sensorAnalogico = parseFloat(valores[1]);  
  
  // armazena os valores dos sensores nos arrays correspondentes  
  valoresSensorAnalogico.push(sensorAnalogico);  
  valoresSensorDigital.push(sensorDigital);  
});
```

```
let sensor = [];  
  
sensor.push("DHTH_temp");  
sensor.push("DHTH_umid");  
sensor.push("Luminosidade");  
sensor.push("LM35_temp");  
sensor.push("chave");
```