



# Estácio

Campus: Polo Centro – São Bernardo do Campo

Curso: Desenvolvimento Full-Stack

Disciplina: Digital

Turma: Virtual

Semestre letivo: 2024.3

Integrantes: Fabio de Almeida

[Repositório GitHub](#)

## Escopo

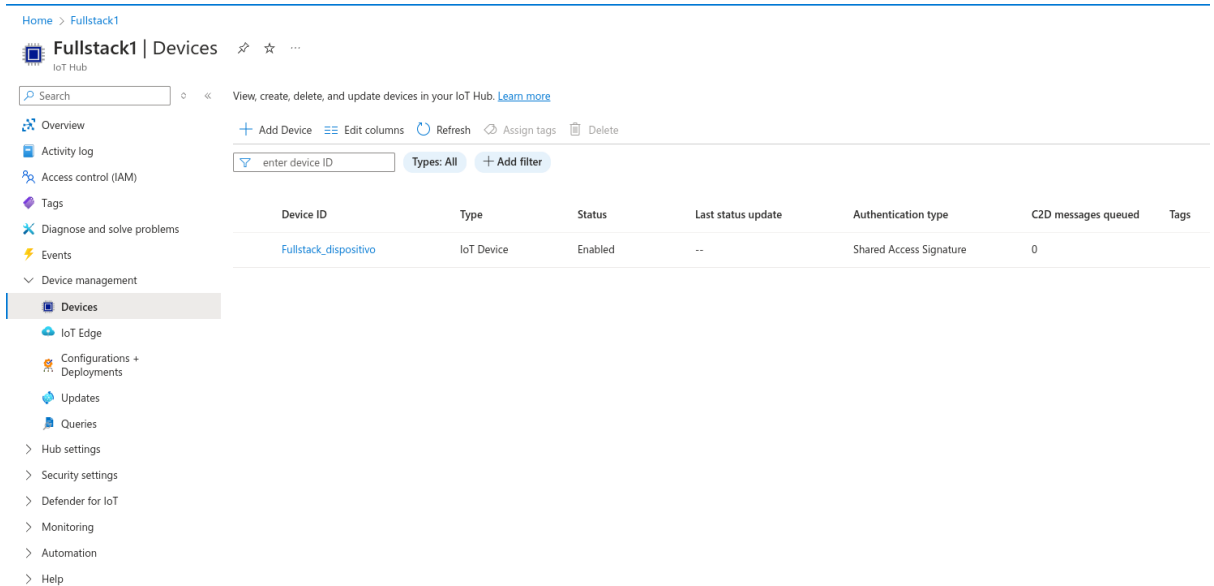
Este projeto visa a criação de um sistema de monitoramento em tempo real de dados de clima (temperatura e umidade) através do Azure IoT Hub, utilizando o Raspberry Pi Azure IoT Online Simulator para simular o dispositivo IoT. O objetivo é visualizar, em tempo real, as informações enviadas pelos sensores do dispositivo, tanto localmente quanto em um aplicativo hospedado na Azure, validando a infraestrutura de monitoramento e visualização de dados fornecida pela plataforma.

## Tecnologia Utilizada

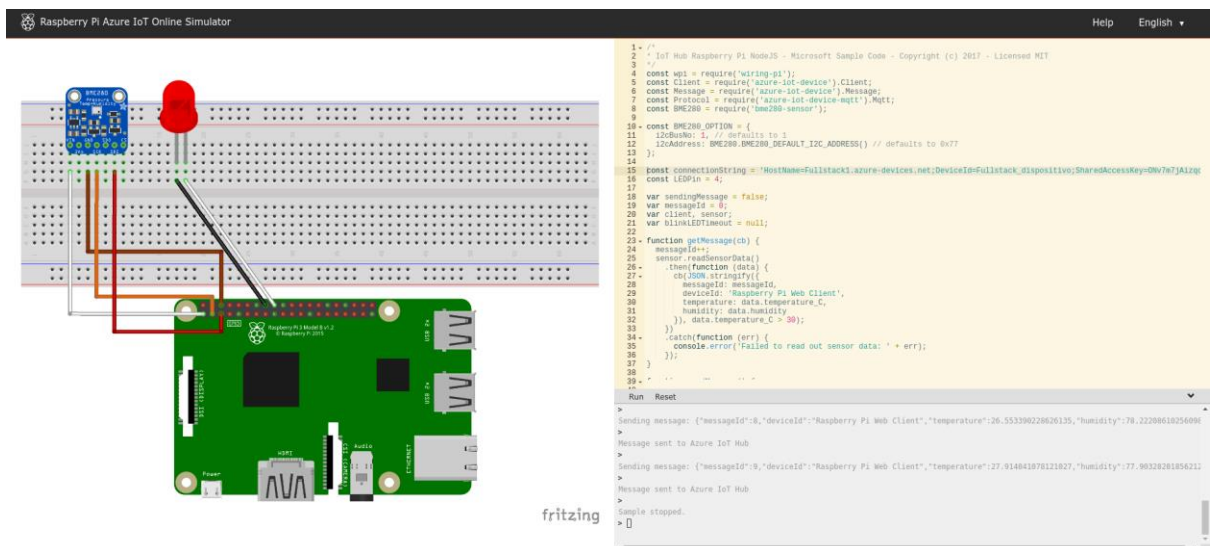
- **Azure IoT Hub:** Serviço utilizado para conectar, monitorar e gerenciar dispositivos IoT.
- **Raspberry Pi Azure IoT Online Simulator:** Ferramenta de simulação de dispositivos que permite simular a leitura e o envio de dados de sensores para o Azure IoT Hub.
- **Azure App Service:** Plataforma de hospedagem para executar e escalar aplicativos web, usada aqui para hospedar o aplicativo que visualiza dados de sensores.
- **Aplicativo “Temperature & Humidity Real-time Data”:** Exemplo fornecido pela Azure, adaptado para simular dados em tempo real.
- **Node.js e Express:** Ferramentas usadas no desenvolvimento do aplicativo web que exibe os dados de sensores.

## Processos Realizados

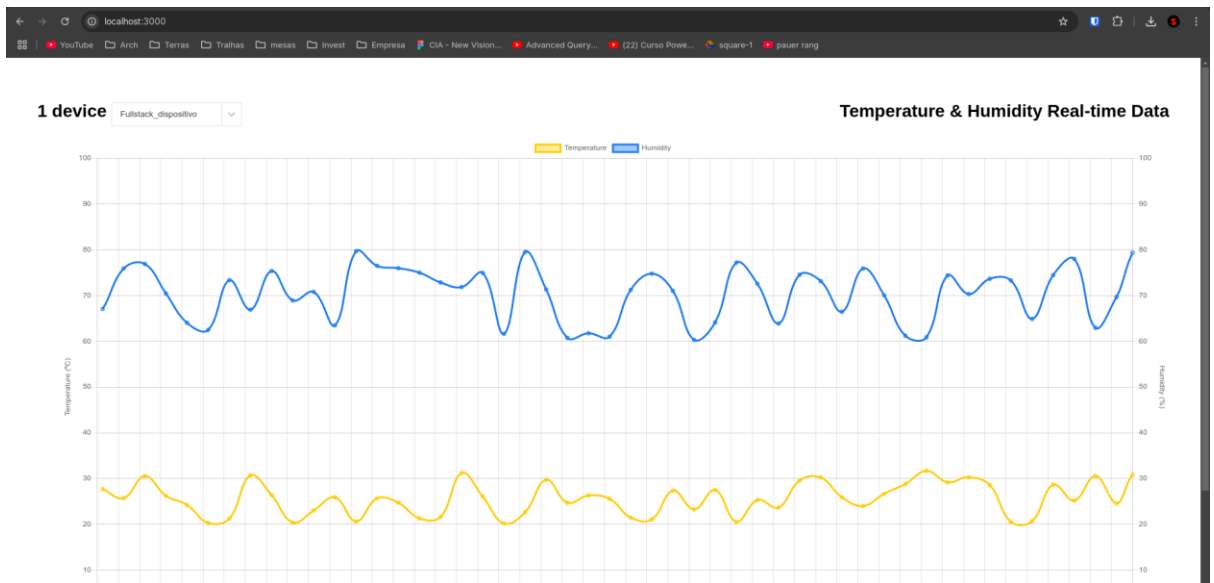
- **Configuração do IoT Hub:** Foi criado um hub IoT no Azure para receber dados de dispositivos. Um dispositivo foi registrado para simular o envio de informações de temperatura e umidade.



- **Simulação de Dispositivo com o Raspberry Pi Online Simulator:** Configurado o simulador para enviar dados periodicamente para o IoT Hub. Com isso, foi possível gerar um fluxo de dados de sensores para o Azure.



- **Desenvolvimento do Web App Local:** Utilizando o exemplo "Temperature & Humidity Real-time Data", foi desenvolvido um aplicativo local que exibe os dados recebidos do Azure IoT Hub em tempo real.



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    x lot.js
  IOT HUB
    node_modules
    lot.js
    lot.py
    package-lock.json
    package.json
  AZURE IOT HUB
    Fullstack1
      Devices
        Fullstack_dispositivo
      Endpoints

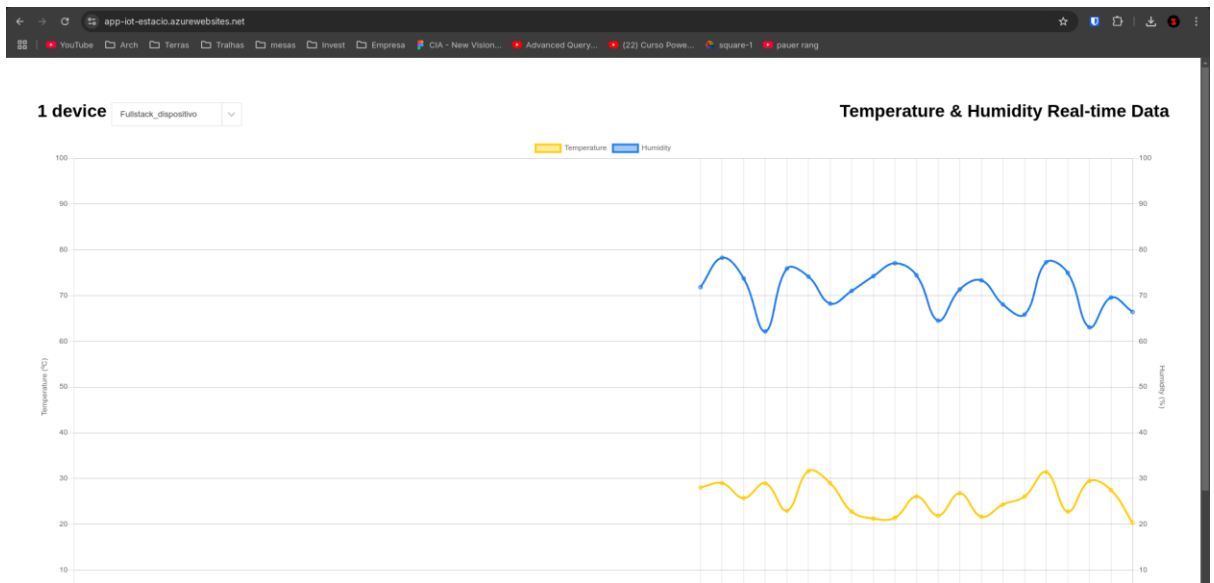
lot.js IoT Hub X
1 // Copyright (c) Microsoft. All rights reserved.
2 // Licensed under the MIT license. See LICENSE file in the project root for full license information.
3
4 "use strict";
5
6 // The device connection string to authenticate the device with your IoT hub.
7 //
8 // NOTE:
9 // For simplicity, this sample sets the connection string in code.
10 // In a production environment, the recommended approach is to use
11 // an environment variable to make it available to your application
12 // or use an X509 or an X509 certificate.
13 // https://docs.microsoft.com/azure/iot-hub/iot-hub-devguide-security
14 var connectionString =
15   "HostName=Fullstack1.azure-devices.net;DeviceId=Fullstack_dispositivo;SharedAccessKey=0Nv7m7jA1zqoCshhIP6KLXmtpj2CUYjojj8VZVEKAXQ=";
```

PROBLEMS OUTPUT TERMINAL PORTS AZURE COMMENTS

TERMINAL

```
send status: MessageEnqueued
Sending message: {"temperature":23.342769572653204,"humidity":78.16908679169904}
send status: MessageEnqueued
Sending message: {"temperature":30.492901168437034,"humidity":78.41396495897057}
send status: MessageEnqueued
Sending message: {"temperature":23.689695715695827,"humidity":60.411507636831715}
send status: MessageEnqueued
Sending message: {"temperature":21.53201007470247,"humidity":79.92807358930258}
send status: MessageEnqueued
^C
~/Documents/Projects/Estacio/IoT Hub 9s
```

- **Hospedagem no Azure App Service:** Após a configuração e validação do aplicativo local, este foi implantado no Azure App Service, possibilitando acesso público via HTTPS e permitindo monitoramento de dados em tempo real de qualquer lugar.



## Declarações Finais

Após a conclusão das etapas descritas, foi possível visualizar os dados de temperatura e umidade enviados pelo simulador em tempo real, tanto localmente quanto no aplicativo hospedado no Azure. Isso demonstrou a eficácia dos serviços Azure na coleta e visualização de dados de dispositivos IoT.

Este projeto demonstrou com sucesso o uso do Azure IoT Hub para receber dados de sensores simulados e exibi-los em um aplicativo em tempo real. A integração entre o Raspberry Pi Online Simulator e o App Service facilita a criação de uma infraestrutura IoT confiável e acessível para monitoramento remoto. Essa atividade permitiu adquirir conhecimentos práticos sobre a criação de uma arquitetura de IoT e a utilização de diversos serviços do Azure para gerenciar dados e serviços em nuvem.