



Estácio

Tecnólogo em Desenvolvimento Full-Stack

Relatório da missão prática Vamos interagir as coisas com a Nuvem
Aluno: Ricardo Alves dos Santos Junior

15/06/2024
Jaraguá do Sul/SC

Tecnólogo em Desenvolvimento Full-Stack

Relatório da missão prática Vamos interagir as coisas com a Nuvem

Aluno: Ricardo Alves dos Santos Junior

Professor: Alessandro dos Santos Clain

15/06/2024

Jaraguá do Sul/SC

Introdução

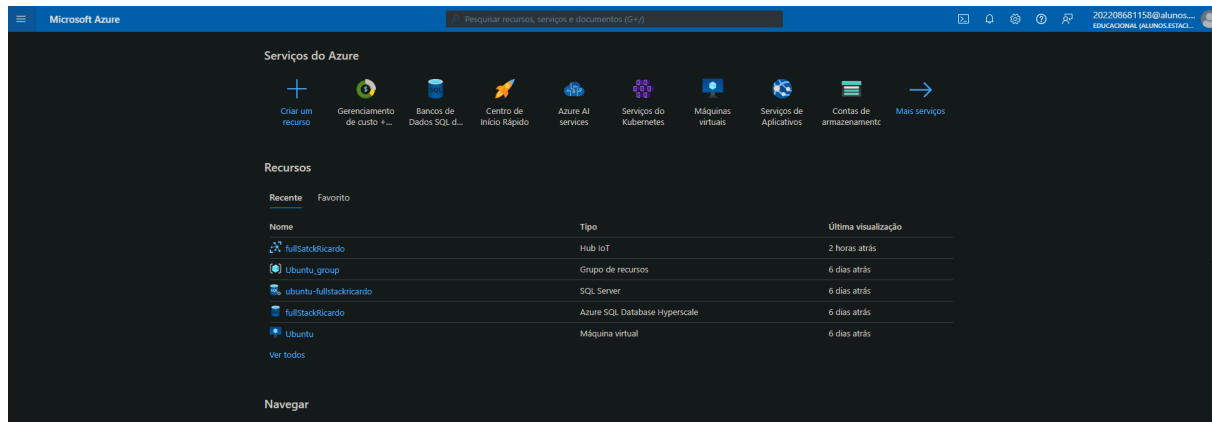
A Internet das Coisas (IoT) permite a interconexão de dispositivos físicos e virtuais, coletando e trocando dados através da internet. Este relatório descreve o desenvolvimento de um sistema IoT utilizando o Azure, com os seguintes objetivos:

- Criar um Hub IoT no Azure.
- Registrar um novo dispositivo no Hub IoT.
- Adicionar a extensão Hub IoT do Azure ao Visual Studio Code.
- Gerenciar e interagir com o Hub IoT.

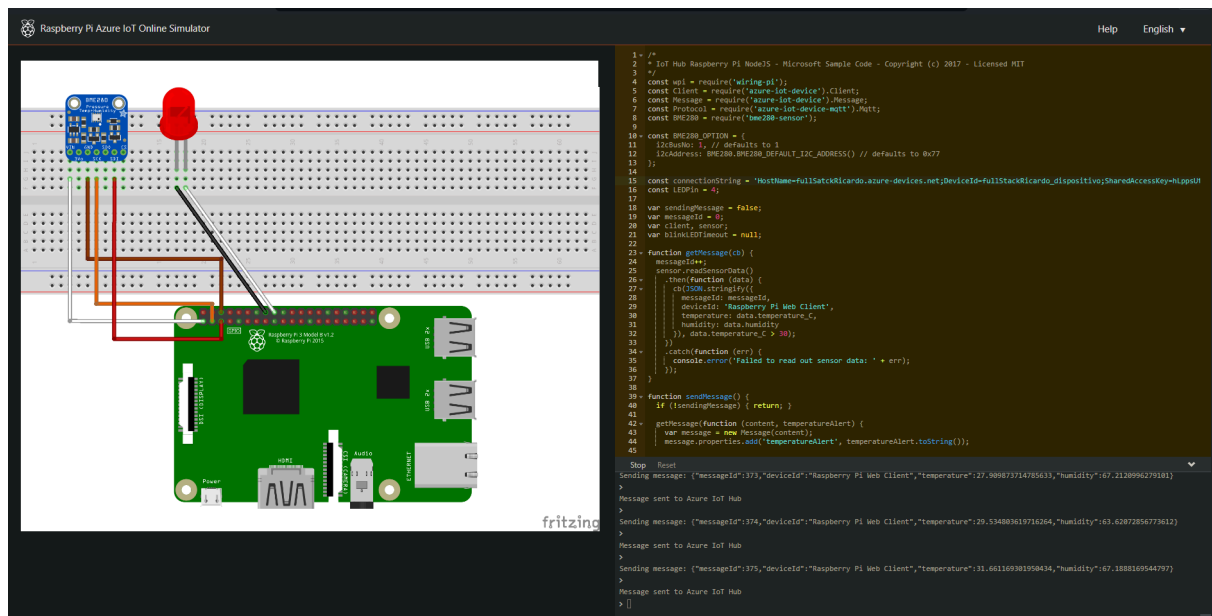
A metodologia envolve o uso de ferramentas da Azure integradas ao Visual Studio Code para desenvolver e gerenciar dispositivos IoT. Este relatório detalha as etapas, configurações, códigos e resultados obtidos, além das dificuldades encontradas e soluções implementadas.

Metodologia

Inicialmente foi criado um novo Hub IOT na Azure com o nome de “fullSatckRicardo”.



Após a criação bem sucedida, foi feito a criação de um novo dispositivo no HUB IOT, criando assim um dispositivo simulado de um raspberry PI com um sensor de umidade e temperatura.

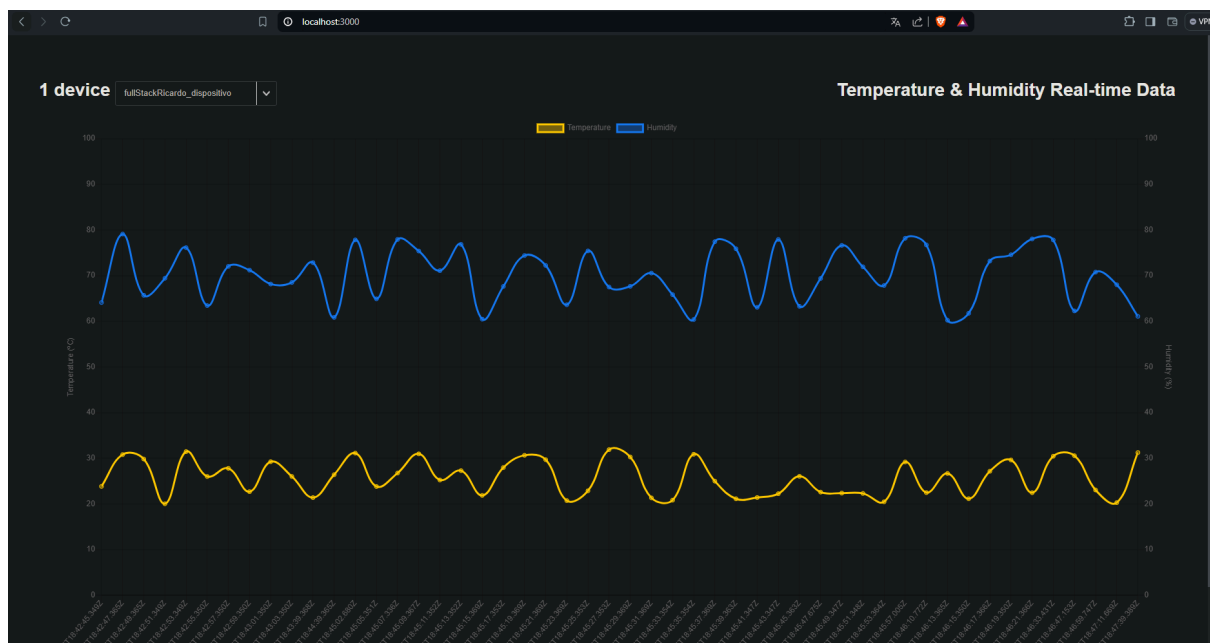


Após estes procedimentos foi implementado um aplicativo usando node.js baixado pelo link “<https://github.com/Azure-Samples/web-apps-node-iot-hub-data-visualization>” em que foi feito a integração do raspberry Pi com um gráfico para apresentar os dados em tempo real.

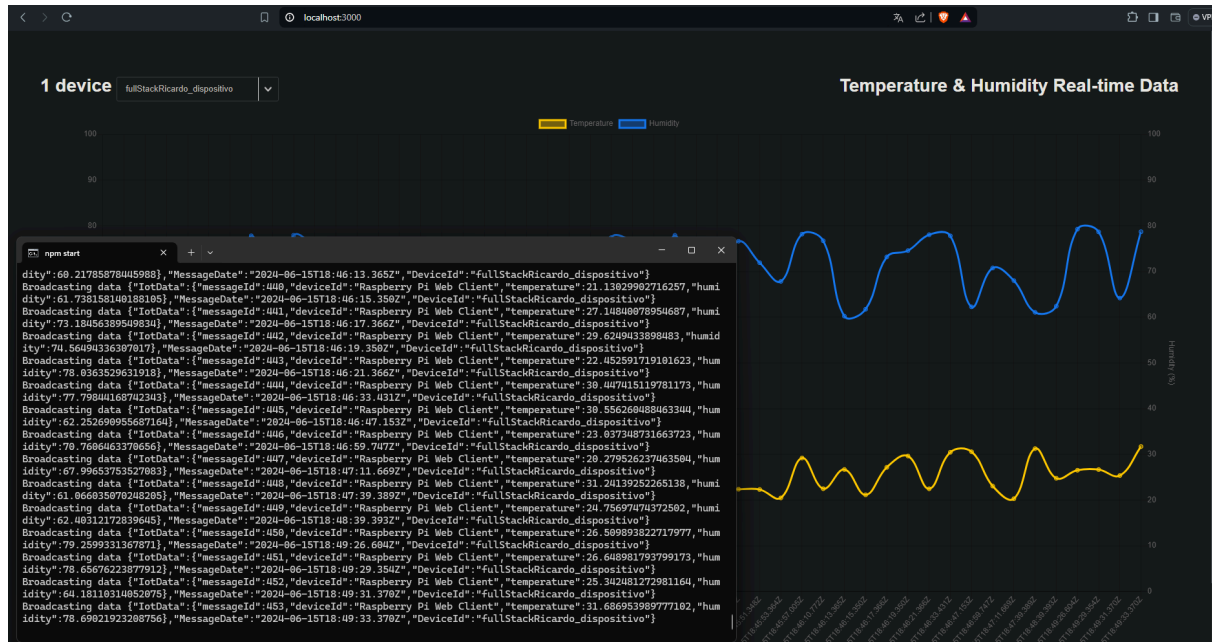
No CMD do windows dentro do repositório git em que se encontrava meu projeto eu executei o comando “npm install” para fazer a instalação do node.js, logo em seguida executei o comando “npm Start” para inicializar o projeto.

```
npm start
}
Node.js v20.14.0
C:\Users\felip\OneDrive\Área de Trabalho\web-apps-node-iot-hub-data-visualization-master\web-apps-node-iot-hub-data-visualization-master>npm install
npm warn old lockfile
npm warn old lockfile The package-lock.json file was created with an old version of npm,
npm warn old lockfile so supplemental metadata must be fetched from the registry.
npm warn old lockfile
npm warn old lockfile This is a one-time fix-up, please be patient...
npm warn old lockfile
npm warn deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if
npm warn deprecated crypto@1.0.1: This package is no longer supported. It's now a built-in Node module. If you've depend
npm warn deprecated glob@7.1.6: Glob versions prior to v9 are no longer supported
npm warn deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when u
npm warn deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams
API instead.
npm warn deprecated jssha@2.3.1: jsSHA versions < 3.0.0 will no longer receive feature updates
npm warn deprecated debug@3.2.6: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when u
npm warn deprecated debug@3.2.6: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when u
```

Ao entrar no navegador no endereço “http:localhost:3000” eu abri o gráfico com as informações do dispositivo raspberry Pi que estava transmitindo as informações de temperatura e Umidade.

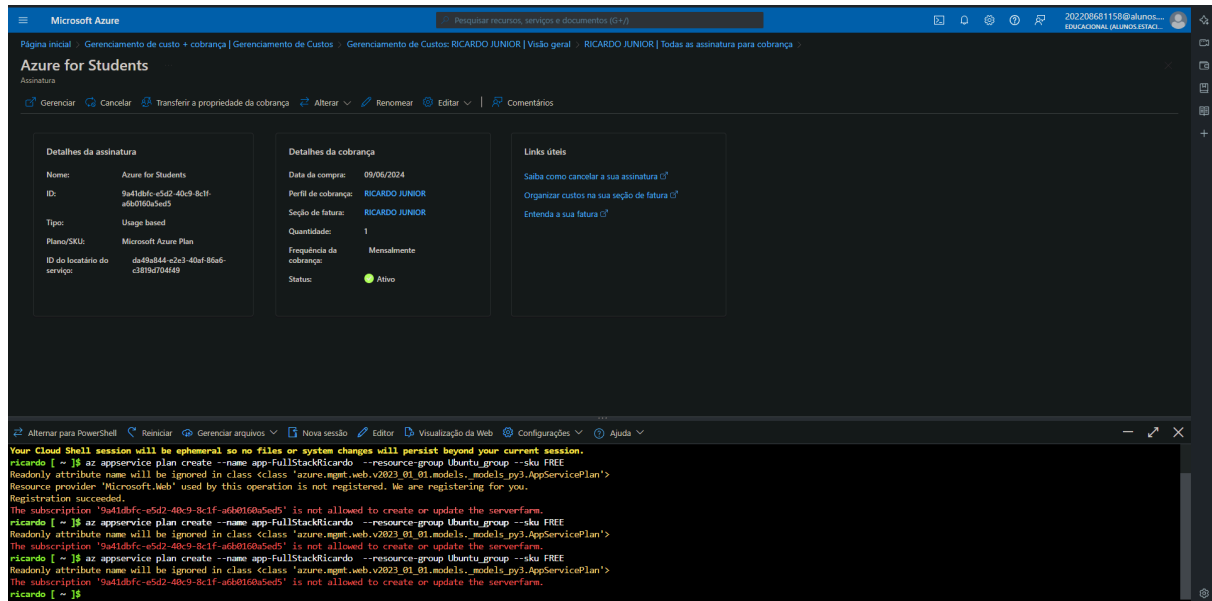


Estes dados que foram plotados no gráfico também puderam ser vistos no cmd do windows em tempo real.



Ao tentar executar o comando “az appservice plan create --name app-FullStackRicardo --resource-group Ubuntu_group --sku FREE” na CLI da Azure não obtive sucesso, pois eu não tinha permissão para mudar o plano e obribe este erro: “The subscription '9a41dbfc-e5d2-40c9-8c1f-a6b0160a5ed5' is not allowed to create or update the serverfarm.”

Devido ao plano grátis de estudante não foi possível desenvolver o resto da aplicação até a hospedagem completa na azure.



Resumo

O desenvolvimento do sistema IoT usando a plataforma Azure foi uma experiência prática muito interessante. Aprendi a criar e gerenciar dispositivos conectados, configurando um Hub IoT, registrando dispositivos e usando o Visual Studio Code para interagir com eles.

Apesar das limitações, como não conseguir hospedar a aplicação na Azure por conta da assinatura gratuita de estudante, consegui alcançar os principais objetivos. A integração do dispositivo Raspberry Pi com a aplicação Node.js permitiu ver os dados de umidade e temperatura em tempo real, mostrando que o sistema funciona bem.

Esse projeto mostrou a importância de conhecer as ferramentas e serviços da Azure para criar soluções IoT. Também ficou claro que é importante planejar os recursos e permissões necessários para completar os projetos. No futuro, vale a pena considerar planos de assinatura mais completos ou alternativas que permitam expandir e hospedar totalmente as aplicações desenvolvidas.