



**UNIVERSIDADE ESTÁCIO DE SÁ  
POLO ALCANTARA - SÃO GONÇALO/RJ**

**ALUNO: LUIZ IZAIAS TEIXEIRA P.G.**

**TECNOLOGIA EM DESENVOLVIMENTO FULL STACK  
RPG0027- Vamos Interligar as Coisas Com a  
Nuvem**

## **RELATÓRIO DA MISSÃO PRÁTICA**

### **1 OBJETIVO**

O objetivo desta missão prática é configurar, executar e visualizar em tempo real os dados de sensores conectados ao Azure IoT Hub, utilizando um aplicativo web Node.js localmente e depois hospedá-lo no Serviço de Aplicativo do Azure.

### **MATERIAIS UTILIZADOS**

- Conta no Microsoft Azure.
- Navegador Web (Google Chrome, Firefox, MS Edge, Safari ou Opera).
- Visual Studio Code (VS Code).
- Raspberry Pi Azure IoT Online Simulator.
- GitHub para baixar o aplicativo web: [Azure IoT Web App](#).

## PROCEDIMENTOS REALIZADOS

### Configuração do ambiente local

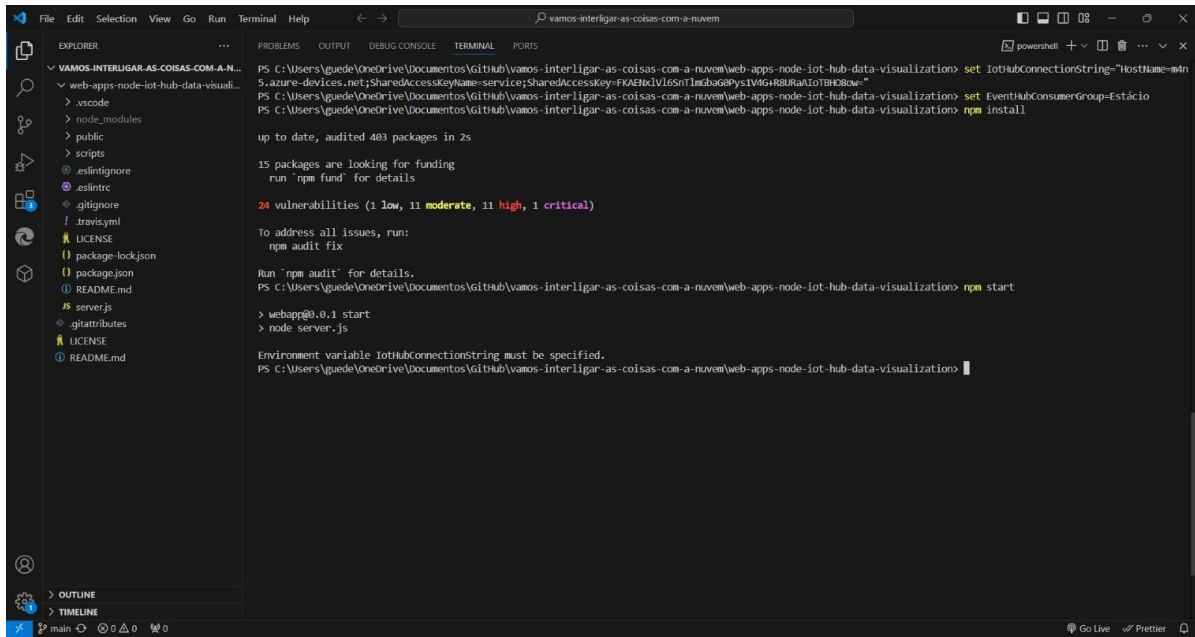
- 📁 Baixar e descompactar o código do GitHub: O código-fonte do aplicativo foi baixado e extraído em uma pasta local. O projeto foi aberto no Visual Studio Code para análise dos arquivos principais.
- 📁 Exploração do código: Foram analisados os principais arquivos do projeto:
  - 📁 server.js: Inicializa a conexão com o hub IoT.
  - 📁 event-hub-reader.js: Lê os dados enviados pelo hub IoT.
  - 📁 chart-device-data.js e index.html: Lógica do lado do cliente para exibir os dados em gráficos.

### Conexão com o Azure IoT Hub

- Criação de um grupo de consumidores no IoT Hub: Utilizando o comando `az iot hub consumer-group create`, foi criado um grupo de consumidores dedicado para ler os dados do hub IoT.
- Configuração das variáveis de ambiente: Foram definidas as variáveis `IotHubConnectionString` e `EventHubConsumerGroup` para permitir que o aplicativo se conecte ao hub IoT.

### Execução local do aplicativo

- Instalação dos pacotes e execução: No diretório do projeto, os pacotes foram instalados com o comando `npm install`, seguido pela execução do servidor com `npm start`.
- Visualização de dados em tempo real: A interface foi acessada via <http://localhost:3000>, onde os dados de temperatura e umidade dos sensores foram exibidos em gráficos.



```
File Edit Selection View Go Run Terminal Help
vamos-interligar-as-coisas-com-a-nuvem

EXPLORER
vamos-interligar-as-coisas-com-a-nuvem
  web-apps-node-iot-hub-data-visuali...
    > .vscode
    > node_modules
    > public
    > scripts
    > .eslintignore
    > .eslintrc
    > .gitignore
    > .travis.yml
    > LICENSE
    > package-lock.json
    > package.json
    > README.md
    > server.js
    > .gitattributes
    > LICENSE
    > README.md

TERMINAL
PS C:\Users\guede\OneDrive\Documentos\github\vamos-interligar-as-coisas-com-a-nuvem\web-apps-node-iot-hub-data-visualization> set IotHubConnectionString="HostName=5.azure-devices.net;SharedAccessKeyName=service;SharedAccessKey=FKAEHdVl6SnTlmbagaPysIVAGH8URaaIOTBHR0aw="
PS C:\Users\guede\OneDrive\Documentos\github\vamos-interligar-as-coisas-com-a-nuvem\web-apps-node-iot-hub-data-visualization> set EventHubConsumerGroup=Estácio
PS C:\Users\guede\OneDrive\Documentos\github\vamos-interligar-as-coisas-com-a-nuvem\web-apps-node-iot-hub-data-visualization> npm install

up to date, audited 403 packages in 2s

15 packages are looking for funding
  run `npm fund` for details

24 vulnerabilities (1 low, 11 moderate, 11 high, 1 critical)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
PS C:\Users\guede\OneDrive\Documentos\github\vamos-interligar-as-coisas-com-a-nuvem\web-apps-node-iot-hub-data-visualization> npm start

> webapp@0.0.1 start
> node server.js

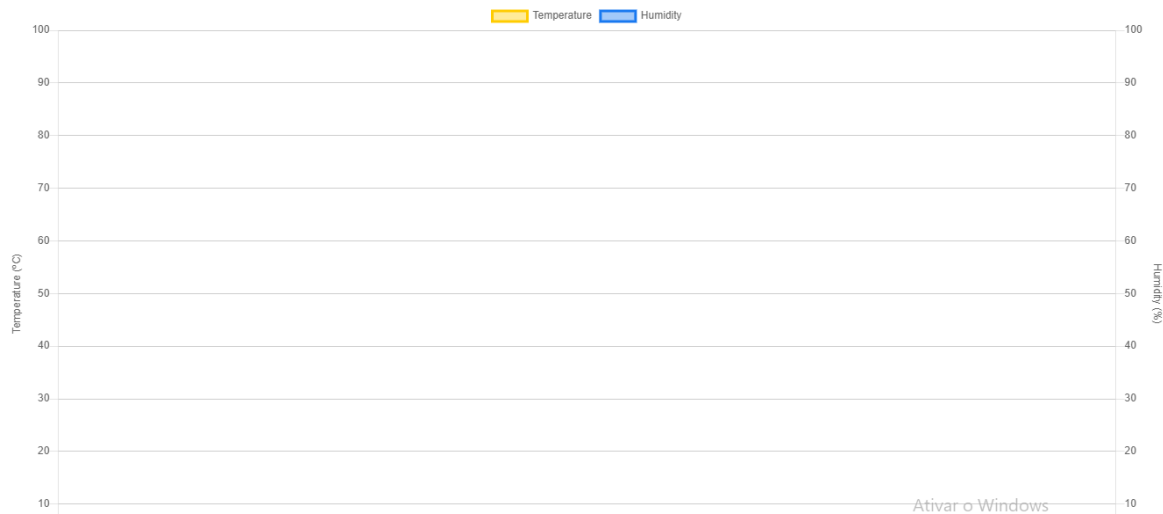
Environment variable IotHubConnectionString must be specified.
PS C:\Users\guede\OneDrive\Documentos\github\vamos-interligar-as-coisas-com-a-nuvem\web-apps-node-iot-hub-data-visualization>
```

## Execução local do aplicativo

- Instalação dos pacotes e execução: No diretório do projeto, os pacotes foram instalados com o comando `npm install`, seguido pela execução do servidor com `npm start`.
- Visualização de dados em tempo real: A interface foi acessada via <http://localhost:3000>, onde os dados de temperatura e umidade dos sensores foram exibidos em gráficos.

0 devices

## Temperature & Humidity Real-time Data



### Hospedagem na nuvem com Azure App Service

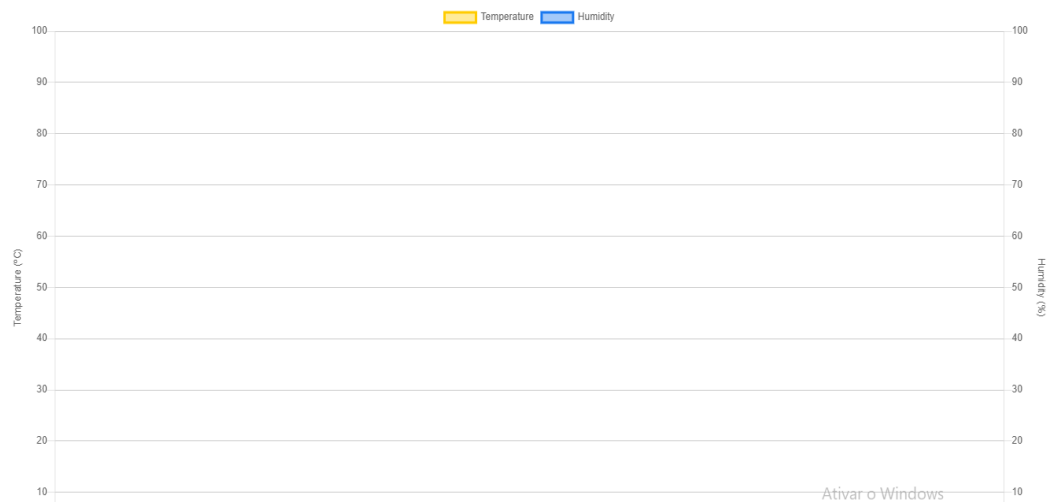
- Criação de um plano de serviço do Azure: Foi provisionado um plano de serviço gratuito no Azure para hospedar o aplicativo.
- Provisionamento do aplicativo: Utilizando o comando `az webapp create`, o aplicativo foi criado e configurado para rodar na plataforma Azure.
- Configuração de variáveis no Azure: As mesmas variáveis de ambiente configuradas localmente foram aplicadas ao serviço no Azure.
- Ativação de WebSockets e HTTPS: Configurações foram ajustadas para suportar WebSockets e garantir segurança via HTTPS.

### Acesso ao aplicativo na nuvem

- Verificação do status: O comando `az webapp show` confirmou que o aplicativo estava rodando corretamente no Azure.
- Acesso via URL pública: O aplicativo foi acessado via `temperatura.azurewebsites.net`, exibindo os dados dos sensores em tempo real.

0 devices

## Temperature & Humidity Real-time Data



### - RESULTADOS OBTIDOS

- Ao final da prática, foi possível visualizar dados de temperatura e umidade enviados pelos sensores conectados ao Azure IoT Hub, tanto localmente quanto pela aplicação hospedada na nuvem Azure. O gráfico em tempo real refletiu com precisão as leituras dos sensores, confirmando o sucesso da integração.

### - CONCLUSÃO

- A missão prática foi concluída com sucesso. Os objetivos de interligar dados IoT com a nuvem Azure e visualizar esses dados em tempo real foram atingidos. Além disso, o uso do Serviço de Aplicativo do Azure possibilitou a escalabilidade e fácil acesso à aplicação, oferecendo um ambiente robusto para o gerenciamento de dados IoT.