



UNIVERSIDADE  
**Estácio de Sá**

<b>Universidade</b>	Estácio de Sá
<b>Campus</b>	Polo de Cobilândia / Vila – Velha/ES
<b>Nome do Curso</b>	Desenvolvimento Full Stack
<b>Nome da Disciplina</b>	RPG0027- Vamos interligar as coisas com a nuvem
<b>Turma</b>	9001
<b>Semestre</b>	Primeiro Semestre de 2025
<b>Integrantes do Grupo</b>	Tiago de Jesus Pereira Furtado
<b>Matrícula</b>	202306189045

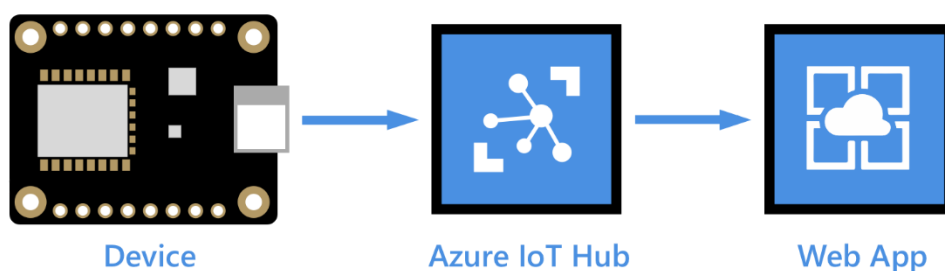
**VILA VELHA  
2025**

## Missão Prática | Vamos interligar as coisas com a nuvem! ☁️

Nessa atividade revisaremos tudo o que utilizamos nas microatividades anteriores. Além disso, veremos também que os [Aplicativos Lógicos do Azure](#) podem ajudar você a orquestrar fluxos de trabalho em serviços locais e de nuvem, em uma ou mais empresas e em vários protocolos.

### - Contextualização

Nesta Missão Prática você aprenderá a visualizar dados em tempo real provenientes de sensores conectados ao seu hub IoT. Isso será realizado por meio da execução de um aplicativo web Node.js em seu computador local. Após a configuração bem-sucedida e execução do aplicativo web local, você terá a opção de hospedá-lo no Serviço de Aplicativo do Azure para facilitar o acesso e a escalabilidade. O fluxo de dados seguirá o caminho delineado na figura abaixo. O dispositivo simulado coletará dados de temperatura e umidade, os quais serão enviados para o Azure IoT Hub e exibidos através do Serviço de Aplicativo do Azure (Web App).



Quando dados de telemetria são recebidos pelo ponto de extremidade do Hub IoT é possível, além de gerar a visualização em tempo real, configurar um aplicativo lógico que pode desencadear uma série de ações. Estas incluem o armazenamento dos dados em um blob no Armazenamento do Azure, o envio de alertas por e-mail em casos de anomalias nos dados e até mesmo a programação de visitas técnicas em resposta a falhas relatadas pelo dispositivo.

As mensagens recebidas pelo seu hub IoT seguem um formato semelhante ao apresentado abaixo, contendo os dados de telemetria no corpo (body) e a propriedade temperatureAlert nas propriedades do aplicativo (applicationProperties). As propriedades do sistema não são exibidas. Confira a seguir a mensagem no formato JSON:

```
{
  "body": {
    "messageId": 18,
    "deviceId": "Raspberry Pi Web Client",
    "temperature": 27.796111770668457,
    "humidity": 66.77637926438427
  },
  "applicationProperties": {
    "temperatureAlert": "false"
  }
}
```

Com os dados de temperatura disponíveis no formato JSON apresentado acima, diversas ações podem ser desencadeadas para otimizar o gerenciamento e a resposta a condições específicas. Por exemplo, ao monitorar a propriedade "temperatureAlert" no conjunto de propriedades do aplicativo, é possível automatizar a detecção de anomalias térmicas. Caso a temperatura detectada ultrapasse um limiar crítico, configurando a propriedade como "true", sistemas automatizados podem ser acionados para notificar por e-mail sobre a situação, alertar equipes responsáveis e armazenar os dados em um blob no Armazenamento do Azure para referência futura ou análise. Essa abordagem proporciona uma resposta proativa a condições adversas, permitindo a implementação eficiente de medidas corretivas ou preventivas com base nos dados em tempo real coletados dos sensores.

Nesta missão prática, o foco inicial será na visualização dos dados de temperatura provenientes dos sensores, conforme apresentados no formato JSON fornecido. No entanto, é importante ressaltar que as ações implementadas durante esta etapa serão inicialmente limitadas à visualização em tempo real. O objetivo principal é proporcionar uma compreensão sólida do fluxo de dados e do processo de exibição no Serviço de Aplicativo do Azure.

Entretanto, encorajamos os alunos a explorar e expandir ainda mais as funcionalidades. Com base no conhecimento adquirido, os participantes têm a oportunidade de desenvolver e incorporar novas funcionalidades ao aplicativo, como a integração de alertas automáticos, armazenamento avançado ou a implementação de outros recursos personalizados. Esta abordagem permite uma aprendizagem prática e flexível, incentivando a criatividade e a aplicação de conceitos em cenários mais complexos, à medida que os alunos aprimoram suas habilidades no gerenciamento de dados IoT.

## Roteiro de prática

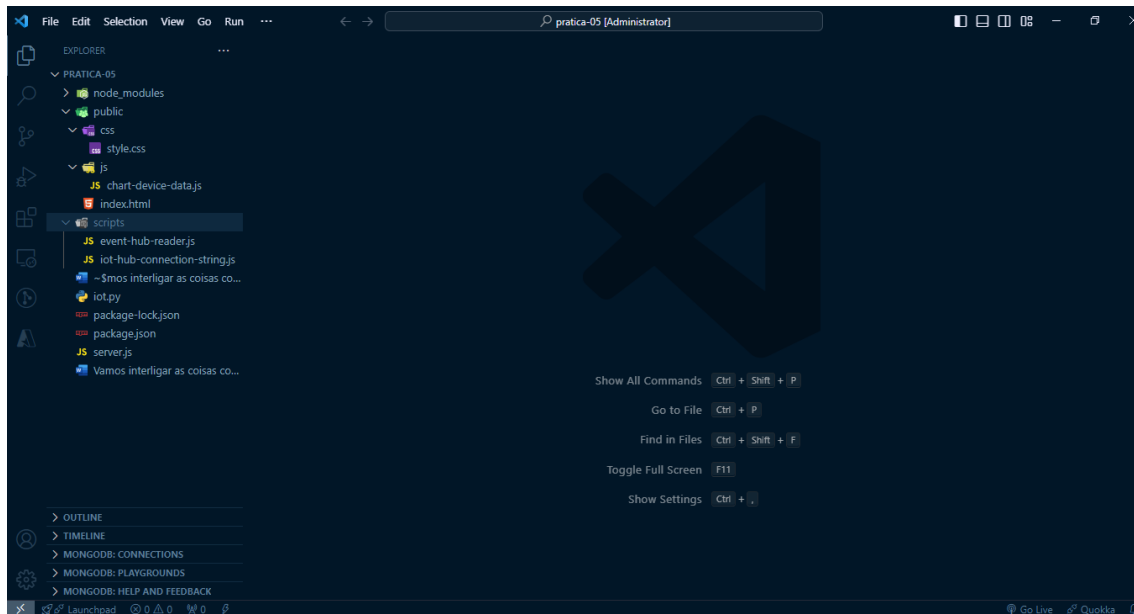
### - Material necessário para a prática

- 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.
- Clone ou baixe o aplicativo Web do GitHub, disponível em <https://github.com/Azure-Samples/web-apps-node-iot-hub-data-visualization>

### - Procedimentos

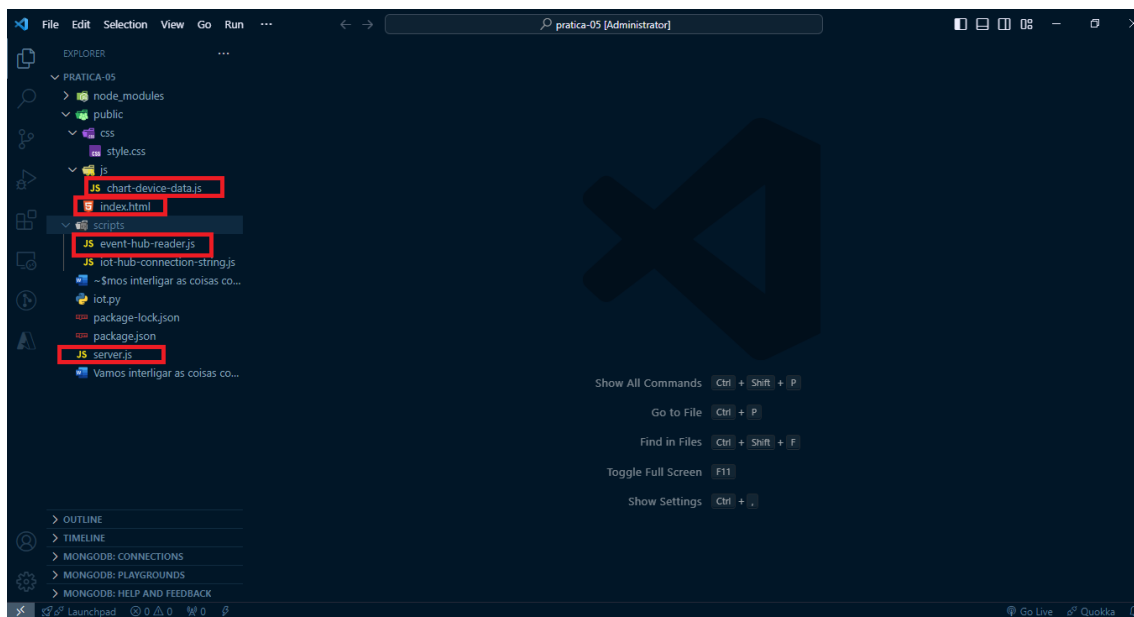
1- Para iniciar o processo, é necessário primeiro descompactar o código-fonte do aplicativo Web disponível no GitHub. Acesse o repositório em <https://github.com/Azure-Samples/web-apps-node-iot-hub-data-visualization> e faça o download do código-fonte. Após o download, descompacte os arquivos em uma pasta de sua escolha.

2- Em seguida, abra a pasta do aplicativo Web na sua IDE preferida, como o Visual Studio Code. Navegue até o diretório "web-apps-node-iot-hub-data-visualization" e explore os arquivos relacionados ao aplicativo. Esta etapa inicial é crucial para garantir um ambiente de desenvolvimento adequado antes de prosseguir para a análise e configuração detalhadas do código. Considerando o uso do VS Code, você terá uma visualização semelhante a imagem a seguir.



Reserve um tempo para examinar os seguintes arquivos:

- **server.js:** Este script do lado do serviço inicializa o soquete de rede e a classe wrapper dos Hubs de Eventos. Fornece um callback para a classe wrapper dos Hubs de Eventos, que utiliza esse callback para transmitir mensagens de entrada para o soquete de rede.
- **scripts/event-hub-reader.js:** Este script do lado do serviço conecta-se ao ponto de extremidade integrado do hub IoT usando a cadeia de conexão e o grupo de consumidores especificados. Extrai informações como DeviceId e EnqueuedTimeUtc de metadados em mensagens de entrada e, em seguida, retransmite a mensagem utilizando o método de retorno de chamada registrado por server.js.
- **public/js/chart-device-data.js:** Este script do lado do cliente escuta no soquete de rede, rastreia cada DeviceId e armazena os últimos 50 pontos de dados de entrada de cada dispositivo. Associa, então, os dados do dispositivo selecionado ao objeto de gráfico.
- **public/index.html:** Este arquivo lida com o layout da interface do usuário da página da web e menciona os scripts necessários para a lógica do lado do cliente.



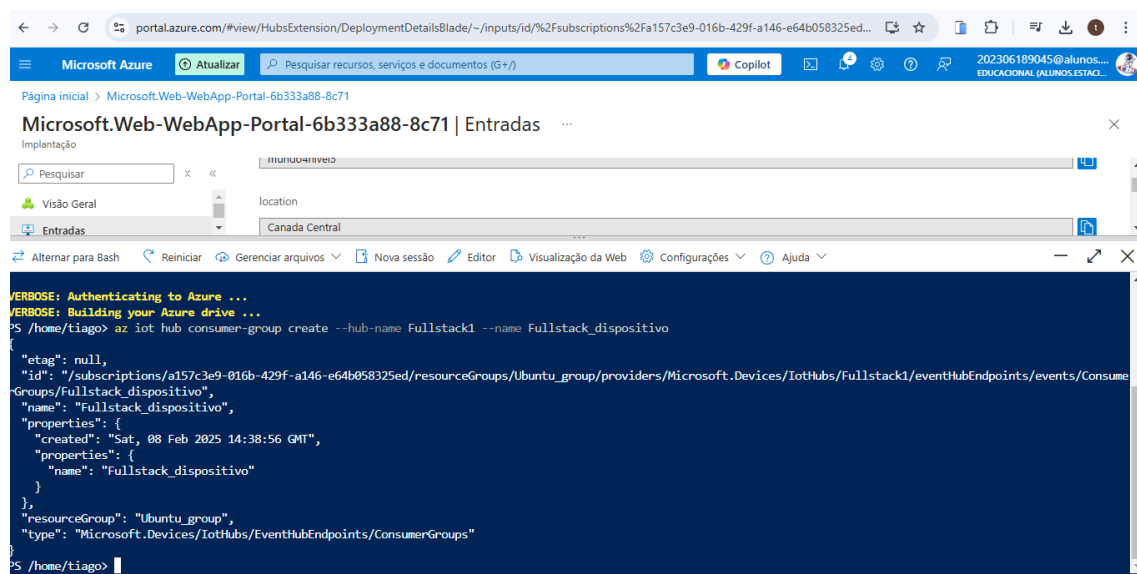
3. Adicionar um grupo de consumidores ao hub IoT é essencial para fornecer visualizações independentes do fluxo de eventos. Essas visualizações permitem que aplicativos e serviços do Azure consumam dados do mesmo ponto de extremidade do hub de eventos.

Nesta etapa, você configurará um grupo de consumidores no ponto de extremidade integrado do seu hub IoT, que será utilizado pelo aplicativo web para ler dados. Execute o seguinte comando na CLI do Azure para adicionar um grupo de consumidores ao ponto de extremidade interno do hub IoT:

```
az iot hub consumer-group create --hub-name  
YOUR_IOT_HUB_NAME --name  
YOUR_CONSUMER_GROUP_NAME
```

Certifique-se de substituir "YOUR\_IOT\_HUB\_NAME" pelo nome do seu hub IoT e "YOUR\_CONSUMER\_GROUP\_NAME" pelo nome desejado para o grupo de consumidores que está sendo criado. Este comando estabelecerá um grupo de consumidores dedicado ao seu hub IoT, proporcionando uma segmentação eficiente do fluxo de eventos para as necessidades específicas do seu aplicativo web. Anote o nome que você escolher porque você precisará dele mais à frente.

A imagem a seguir ilustra a execução do comando.



The screenshot shows the Azure portal interface. At the top, the browser address bar displays the URL: `portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/inputs/id/%2Fsubscriptions%2Fa157c3e9-016b-429f-a146-e64b058325ed...`. The page title is "Microsoft.Web-WebApp-Portal-6b333a88-8c71 | Entradas". Below the title, there is a search bar and a navigation menu with options like "Visão Geral", "Entradas", and "Gerenciar arquivos". The main content area shows a terminal window with the following text:

```
VERBOSE: Authenticating to Azure ...  
VERBOSE: Building your Azure drive ...  
PS /home/tiago> az iot hub consumer-group create --hub-name Fullstack1 --name Fullstack_dispositivo  
{  
  "etag": null,  
  "id": "/subscriptions/a157c3e9-016b-429f-a146-e64b058325ed/resourceGroups/Ubuntu_group/providers/Microsoft.Devices/IotHubs/Fullstack1/eventHubEndpoints/events/Consumers/Fullstack_dispositivo",  
  "name": "Fullstack_dispositivo",  
  "properties": {  
    "created": "Sat, 08 Feb 2025 14:38:56 GMT",  
    "properties": {  
      "name": "Fullstack_dispositivo"  
    }  
  },  
  "resourceGroup": "Ubuntu_group",  
  "type": "Microsoft.Devices/IotHubs/EventHubEndpoints/ConsumerGroups"  
}  
PS /home/tiago>
```



4. Configurar as variáveis de ambiente para o aplicativo Web é crucial para a leitura de dados do seu hub IoT. O aplicativo Web requer a cadeia de conexão do seu hub IoT e o nome do grupo de consumidores que deve ser lido. Essas informações são obtidas do ambiente de processo, como mostrado nas seguintes linhas em `server.js`:

```
const iotHubConnectionString = process.env.iotHubConnectionString;

if (!iotHubConnectionString) {

  console.error(`Environment variable iotHubConnectionString must be specified.`);

  return;
}

console.log(`Using IoT Hub connection string [${iotHubConnectionString}]`);

const eventHubConsumerGroup = process.env.EventHubConsumerGroup;

console.log(eventHubConsumerGroup);

if (!eventHubConsumerGroup) {

  console.error(`Environment variable EventHubConsumerGroup must be specified.`);

  return;
}

console.log(`Using event hub consumer group [${eventHubConsumerGroup}]`);
```

5. Obter uma cadeia de conexão de serviço para seu hub IoT é um passo crucial para conceder permissões necessárias a serviços que precisam ler e gravar nos pontos de extremidade do hub IoT. Os hubs IoT são criados com várias políticas de acesso padrão, e uma dessas políticas é a política de serviço.

Execute o seguinte comando para obter uma cadeia de conexão para o hub IoT, seguindo a política de serviço:

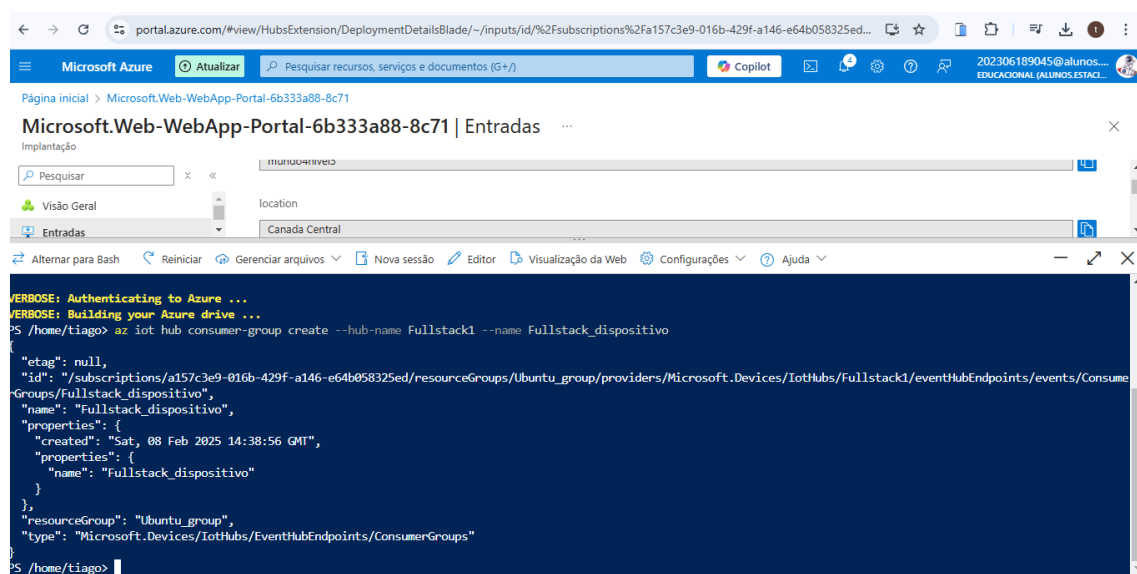
```
az iot hub show-connection-string --hub-name YOUR_IOT_HUB_NAME --policy-name service
```

Certifique-se de substituir "YOUR\_IOT\_HUB\_NAME" pelo nome do seu hub IoT. Este comando fornecerá a cadeia de conexão de serviço necessária para garantir que o aplicativo web tenha as permissões adequadas para interagir com o hub IoT de maneira segura e eficiente.

A cadeia de conexão do serviço deve ser semelhante ao seguinte exemplo:

```
"HostName=YOUR_IOT_HUB_NAME.azure-devices.net;SharedAccessKeyName=service;SharedAccessKey=YOUR_SHARED_ACCESS_KEY"
```

A imagem a seguir ilustra a execução do comando. Anote a cadeia de conexão porque você precisará dele mais à frente.

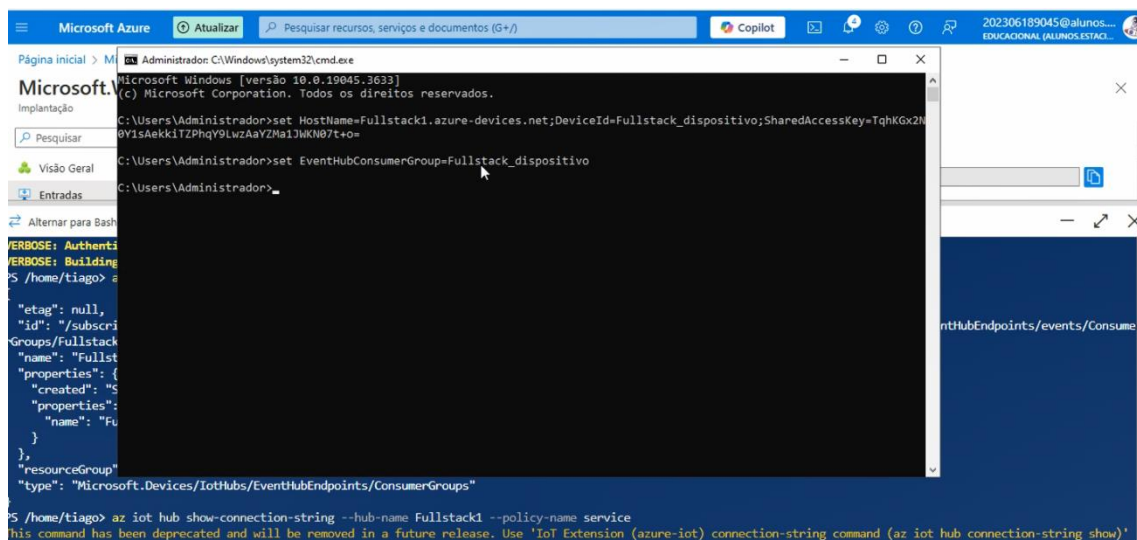


```
portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~:/inputs/id/%2Fsubscriptions%2Fa157c3e9-016b-429f-a146-e64b058325ed...
Microsoft Azure
Página inicial > Microsoft.Web-WebApp-Portal-6b333a88-8c71
Microsoft.Web-WebApp-Portal-6b333a88-8c71 | Entradas
Implantação
Pesquisar
Visão Geral location
Entradas Canada Central
Alternar para Bash Reiniciar Gerenciar arquivos Nova sessão Editor Visualização da Web Configurações Ajuda
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/tiago> az iot hub consumer-group create --hub-name Fullstack1 --name Fullstack_dispositivo
{
  "etag": null,
  "id": "/subscriptions/a157c3e9-016b-429f-a146-e64b058325ed/resourceGroups/Ubuntu_group/providers/Microsoft.Devices/IotHubs/Fullstack1/eventHubEndpoints/events/Consum
Groups/Fullstack_dispositivo",
  "name": "Fullstack_dispositivo",
  "properties": {
    "created": "Sat, 08 Feb 2025 14:38:56 GMT",
    "properties": {
      "name": "Fullstack_dispositivo"
    }
  },
  "resourceGroup": "Ubuntu_group",
  "type": "Microsoft.Devices/IotHubs/EventHubEndpoints/ConsumerGroups"
}
PS /home/tiago>
```

O comando a seguir exemplifica o uso no prompt de comando do Windows:

```
set EventHubConsumerGroup=YOUR_CONSUMER_GROUP_NAME
```

A imagem a seguir apresenta a execução dos comandos em um ambiente Windows



7. Para executar o aplicativo web, primeiro, certifique-se de que o dispositivo está em execução e enviando dados. Na janela do terminal, execute as seguintes linhas para baixar e instalar os pacotes referenciados e iniciar o site: Para usuários do Prompt de Comando do Windows, execute os comandos abaixo.

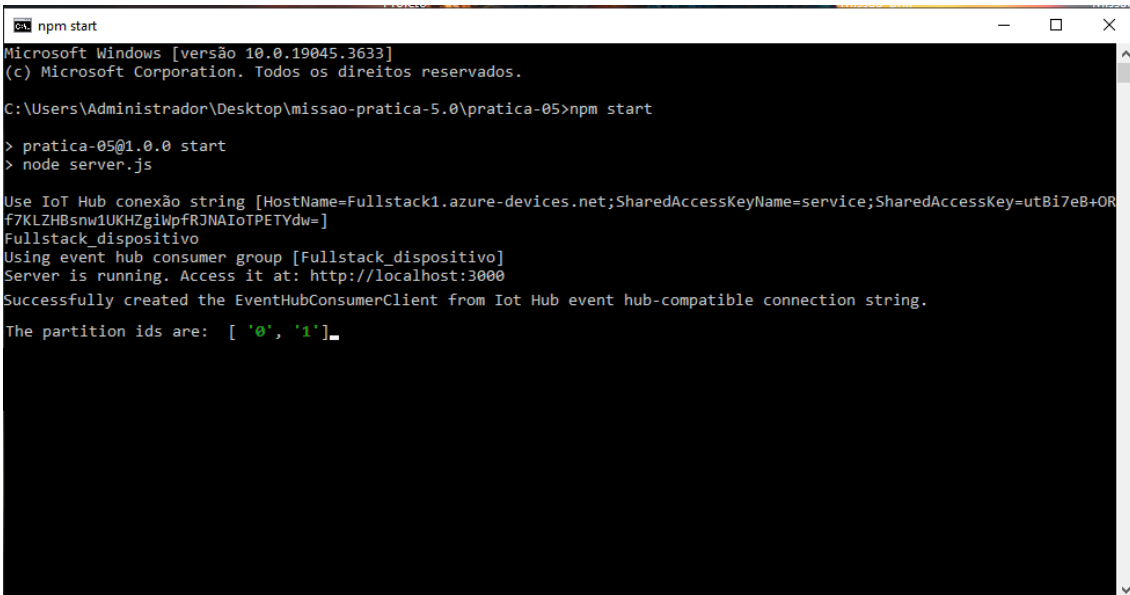
```
npm install
```

```
npm start
```

**IMPORTANTE !**: os comandos acima devem ser executados no diretório que

**contém os arquivos que foram copiados do Git.**

Você deve ver a saída no console, que indica que o aplicativo Web foi conectado com êxito ao hub IoT e está escutando na porta 3000, conforme figura a seguir.



```
npm start
Microsoft Windows [versão 10.0.19045.3633]
(c) Microsoft Corporation. Todos os direitos reservados.

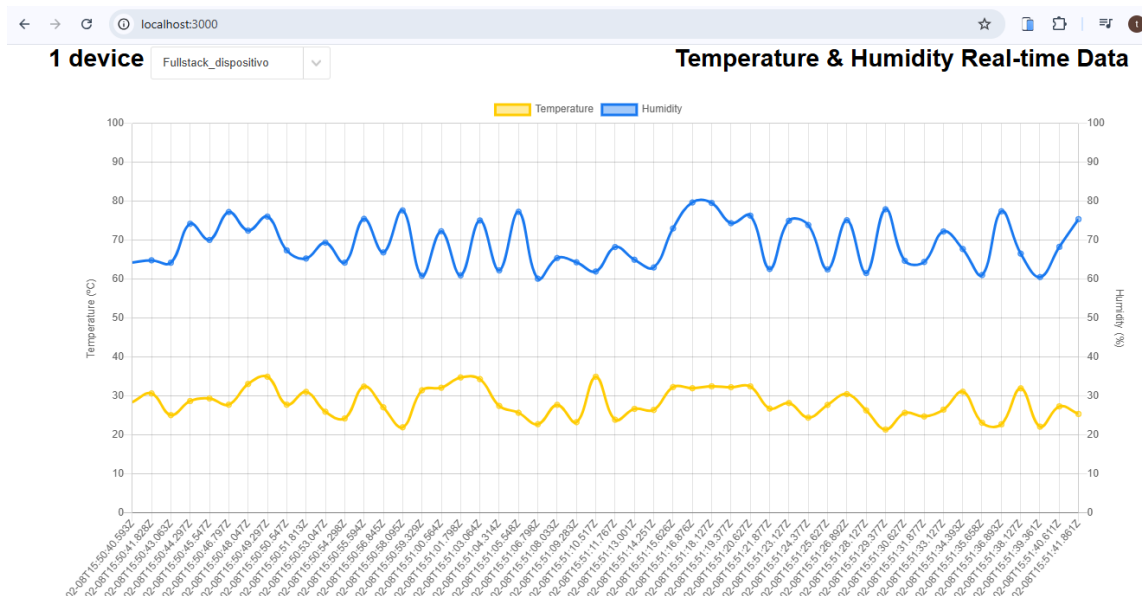
C:\Users\Administrador\Desktop\missao-pratica-5.0\pratica-05>npm start

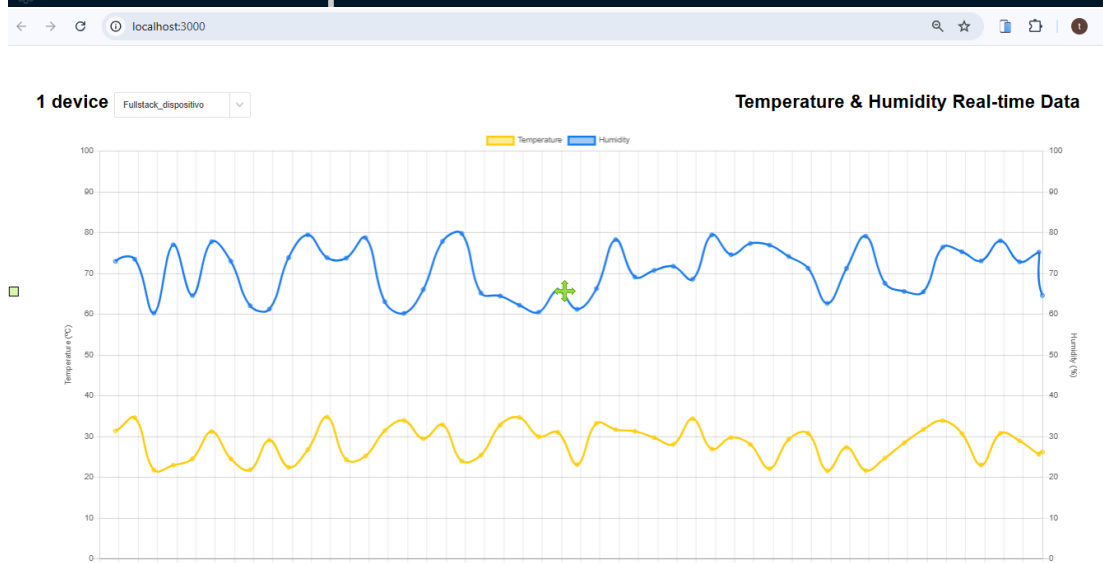
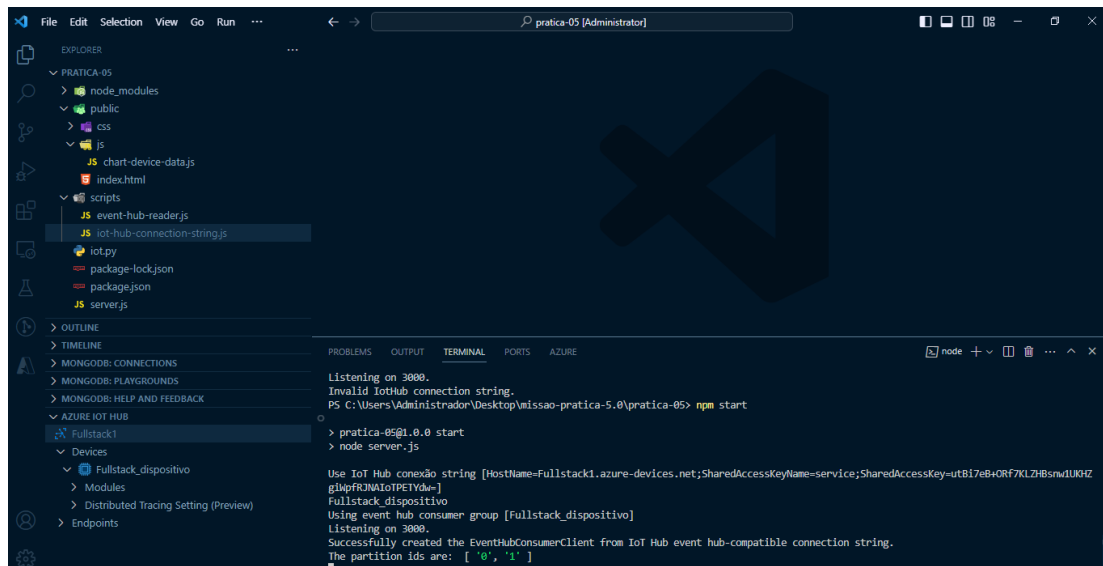
> pratica-05@1.0.0 start
> node server.js

Use IoT Hub conexão string [HostName=Fullstack1.azure-devices.net;SharedAccessKeyName=service;SharedAccessKey=utBi7eB+ORf7KLZHBsnw1UKHZgiWpfRJNAIoTPETVdw=]
Fullstack dispositivo
Using event hub consumer group [Fullstack_dispositivo]
Server is running. Access it at: http://localhost:3000
Successfully created the EventHubConsumerClient from Iot Hub event hub-compatible connection string.

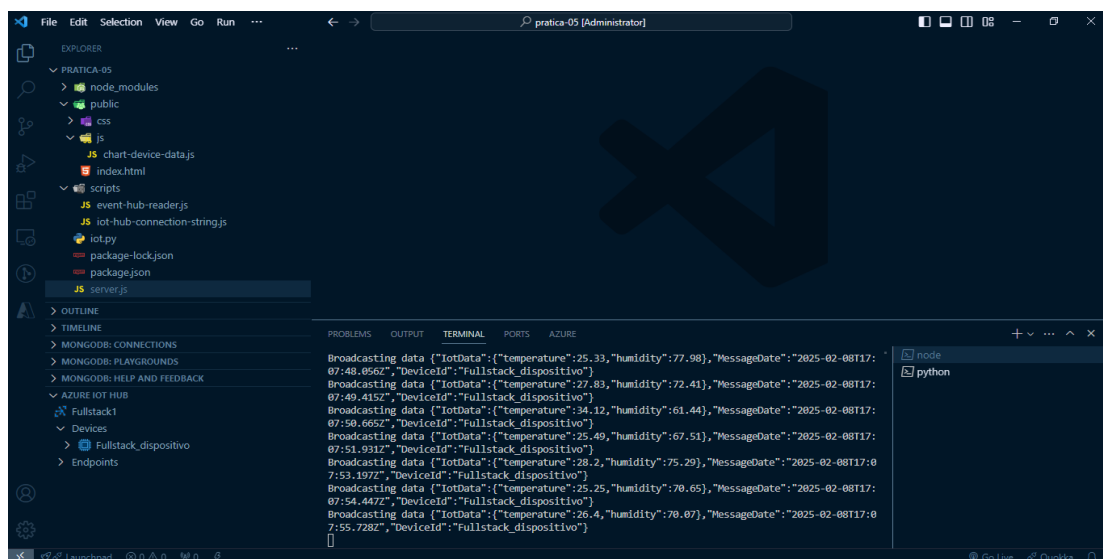
The partition ids are: [ '0', '1' ]
```

8. Com todo o ambiente configurado, agora é possível visualizar a coleta de dados através de um navegador. Para visualizar os dados do hub IoT, abra uma página da web acessando <http://localhost:3000> em seu navegador. Na lista "Selecionar um dispositivo", escolha o dispositivo desejado para visualizar um gráfico em tempo real dos últimos 50 pontos de dados de temperatura e umidade enviados pelo dispositivo para o hub IoT. Essa interface intuitiva proporciona uma análise eficaz e interativa dos dados coletados, permitindo uma compreensão detalhada das condições do dispositivo em tempo real. A figura a seguir ilustra o gráfico com os dados que foram coletados do sensor e apresentados no navegador.





Você também deve ver a saída no console, que mostra as mensagens que seu aplicativo Web está difundindo para o cliente de navegador:



Até este ponto, o sensor e o Hub IoT estão em execução na nuvem, enquanto o servidor web opera localmente em sua máquina. Agora, avançaremos para a segunda fase da Missão Prática, onde o objetivo é hospedar o servidor na nuvem Azure por meio do Serviço de Aplicativo do Azure.

O Serviço de Aplicativo do Azure oferece uma plataforma como serviço (PAAS) para a hospedagem de aplicativos web. Ao utilizar esse serviço, os aplicativos web podem aproveitar os recursos robustos do Azure, incluindo segurança avançada, balanceamento de carga e escalabilidade.

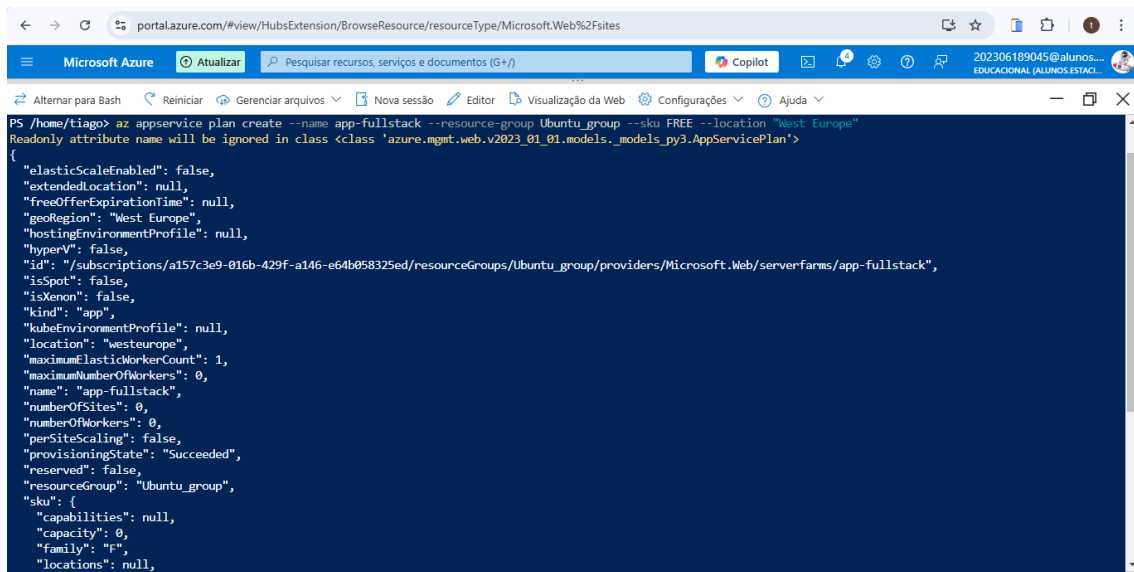
Nesta etapa, você provisionará um aplicativo web no Serviço de Aplicativo e implantará o código nele utilizando os comandos da CLI do Azure. Os detalhes dos comandos utilizados podem ser encontrados na documentação oficial do `az webapp`. Este processo garantirá que o servidor web esteja agora disponível na nuvem, proporcionando benefícios adicionais de desempenho e escalabilidade.

9. Um plano de Serviço de Aplicativo define um conjunto de recursos de computação para a execução de um aplicativo hospedado no Serviço de Aplicativo. Para criar um novo plano do Serviço de Aplicativo usando o nível gratuito do Windows, utilize o comando **`az appservice plan create`**. Certifique-se de usar o mesmo grupo de recursos em que seu hub IoT está localizado. O nome do plano do serviço pode conter letras maiúsculas e minúsculas, números e hifens.

Na CLI do Azure, execute o seguinte comando no Cloud Shell, substituindo NEW\_NAME\_FOR\_YOUR\_APP\_SERVICE\_PLAN pelo seu nome do plano de serviço de aplicativo e YOUR\_RESOURCE\_GROUP\_NAME pelo nome do grupo de recursos

```
az appservice plan create --name NEW_NAME_FOR_YOUR_APP_SERVICE_PLAN --resource-group YOUR_RESOURCE_GROUP_NAME --sku FREE
```

A imagem a seguir ilustra a execução do comando.



```
PS /home/tyago> az appservice plan create --name app-fullstack --resource-group Ubuntu_group --sku FREE --location "West Europe"
Readonly attribute name will be ignored in class <class 'azure.mgmt.web.v2023_01_01.models._models_py3.AppServicePlan'>
{
  "elasticScaleEnabled": false,
  "extendedLocation": null,
  "freeOfferExpirationTime": null,
  "geoRegion": "West Europe",
  "hostingEnvironmentProfile": null,
  "hyperV": false,
  "id": "/subscriptions/a157c3e9-016b-429f-a146-e64b058325ed/resourceGroups/Ubuntu_group/providers/Microsoft.Web/server-farms/app-fullstack",
  "isSpot": false,
  "isXenon": false,
  "kind": "app",
  "kubeEnvironmentProfile": null,
  "location": "westeurope",
  "maximumElasticWorkerCount": 1,
  "maximumNumberOfWorkers": 0,
  "name": "app-fullstack",
  "numberOfSites": 0,
  "numberOfWorkers": 0,
  "perSiteScaling": false,
  "provisioningState": "Succeeded",
  "reserved": false,
  "resourceGroup": "Ubuntu_group",
  "sku": {
    "capabilities": null,
    "capacity": 0,
    "family": "F",
    "locations": null,
```

Obs.: No meu caso foi utilizado o “West Europe” Porque o “sku FREE”, não Estava dando certo.



10. Utilize o comando **az webapp create** para provisionar um aplicativo web em seu Plano do Serviço de Aplicativo, utilizando o comando a seguir:

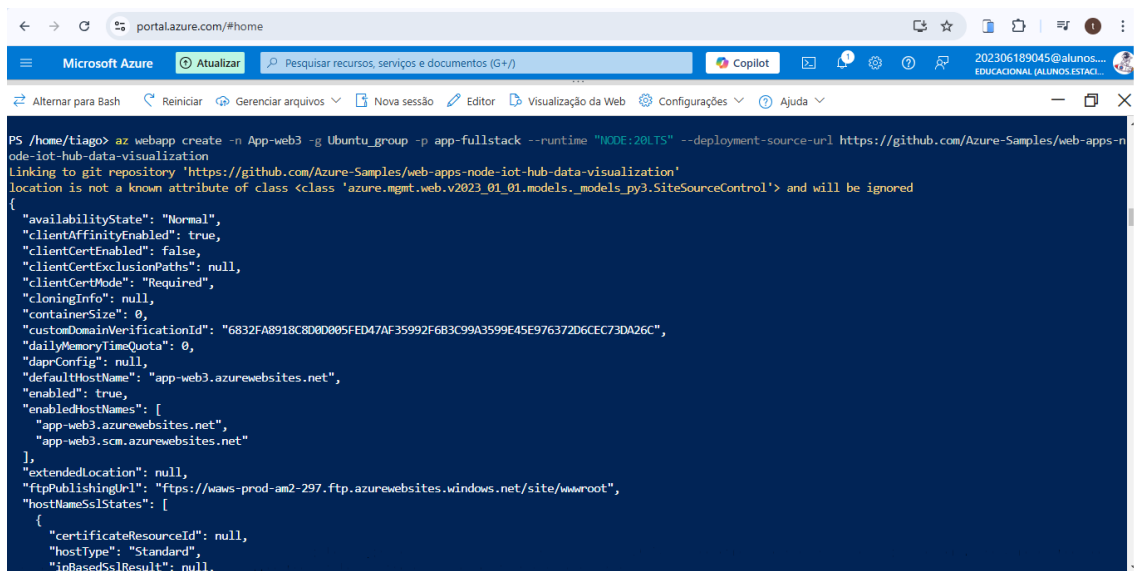
```
az webapp create -n NEW_NAME_FOR_YOUR_WEB_APP -g  
YOUR_RESOURCE_GROUP_NAME -p YOUR_APP_SERVICE_PLAN_NAME --runtime  
"NODE:16LTS" --deployment-local-git
```

O parâmetro **--deployment-local-git** permite que o código do aplicativo web seja carregado e implantado a partir de um repositório Git em seu computador local. Caso o código do aplicativo seja carregado de um repositório Git na internet utilize o parâmetro **--deployment-source-url** seguido da URL do repositório desejado.

O nome do aplicativo web deve ser globalmente exclusivo e pode conter letras maiúsculas e minúsculas, números e hifens. O nome do aplicativo deve ser anotado porque será utilizado posteriormente.

Certifique-se de especificar a versão 16 ou posterior do Node para o parâmetro **--runtime**, dependendo da versão do runtime do Node.js que você estiver utilizando. Você pode utilizar o comando **az webapp list-runtimes** para obter uma lista dos runtimes disponíveis.

A imagem a seguir ilustra a execução do comando.



```
PS /home/tiago> az webapp create -n App-web3 -g Ubuntu_group -p app-fullstack --runtime "NODE:20LTS" --deployment-source-url https://github.com/Azure-Samples/web-apps-node-iot-hub-data-visualization
Linking to git repository 'https://github.com/Azure-Samples/web-apps-node-iot-hub-data-visualization'
location is not a known attribute of class <class 'azure.mgmt.web.v2023_01_01.models._models_py3.SiteSourceControl'> and will be ignored
{
  "availabilityState": "Normal",
  "clientAffinityEnabled": true,
  "clientCertEnabled": false,
  "clientCertExclusionPaths": null,
  "clientCertMode": "Required",
  "cloningInfo": null,
  "containerSize": 0,
  "customDomainVerificationId": "6832FA8918C80D005FED47AF35992F683C99A3599E45E976372D6CEC73DA26C",
  "dailyMemoryTimeQuota": 0,
  "daprConfig": null,
  "defaultHostName": "app-web3.azurewebsites.net",
  "enabled": true,
  "enabledHostNames": [
    "app-web3.azurewebsites.net",
    "app-web3.scm.azurewebsites.net"
  ],
  "extendedLocation": null,
  "ftpPublishingUrl": "https://waaws-prod-am2-297.ftp.azurewebsites.windows.net/site/wwwroot",
  "hostNameSslStates": [
    {
      "certificateResourceId": null,
      "hostType": "Standard",
      "ipBasedSslResult": null,

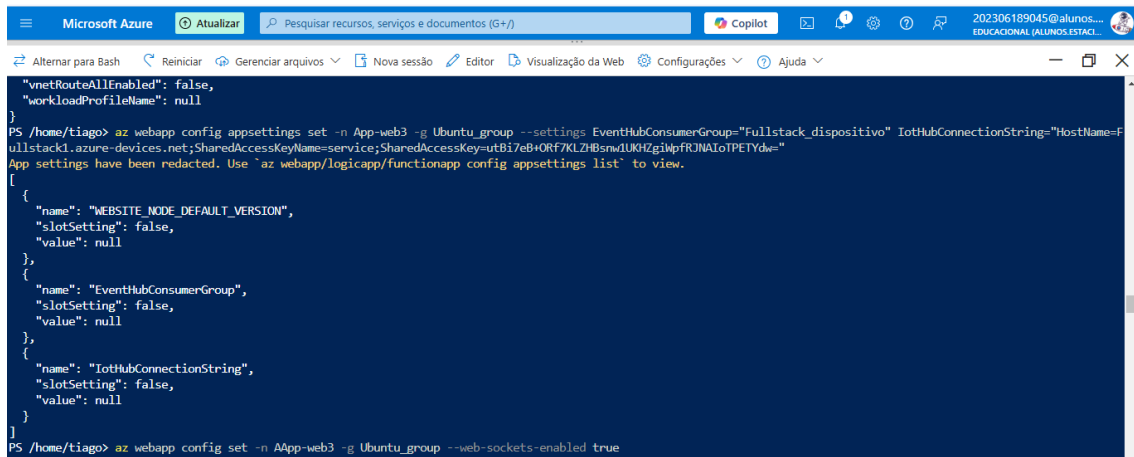
```

11. Utilize o comando **az webapp config appsettings set** para adicionar configurações de aplicativo referentes às variáveis de ambiente que especificam a cadeia de conexão do hub IoT e o grupo de consumidores do hub de eventos.

```
az webapp config appsettings set -n YOUR_WEB_APP_NAME -g  
YOUR_RESOURCE_GROUP_NAME --settings  
EventHubConsumerGroup=YOUR_CONSUMER_GROUP_NAME  
IotHubConnectionString="YOUR_IOT_HUB_CONNECTION_STRING"
```

As configurações individuais são delimitadas por espaço no parâmetro `-settings`. Utilize a cadeia de conexão de serviço do hub IoT e o grupo de consumidores criado anteriormente nesta prática.

A imagem a seguir ilustra a execução do comando.



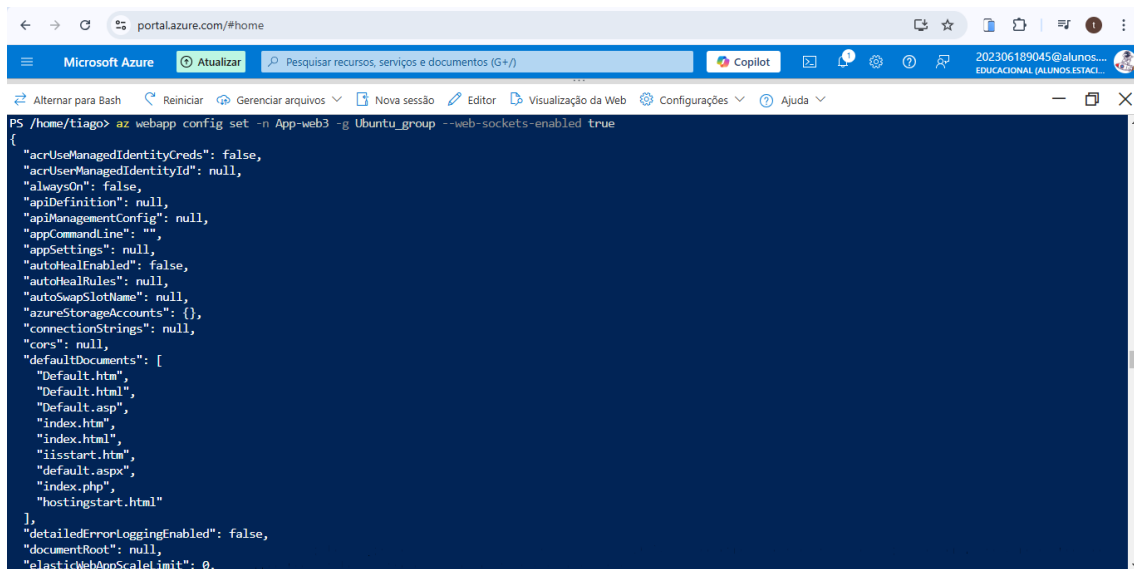
```
Microsoft Azure | Atualizar | Pesquisar recursos, serviços e documentos (G+/) | Copilot | 202306189045@alunos... EDUCACIONAL (ALUNOS.ES...)
```

```
PS /home/tiago> az webapp config appsettings set -n App-web3 -g Ubuntu_group --settings EventHubConsumerGroup="Fullstack_dispositivo" IotHubConnectionString="HostName=Fullstack1.azure-devices.net;SharedAccessKeyName=service;SharedAccessKey=utB17eB+ORF7KLZH8snw1UKHZg1MpFRJNAIoTPETYdw="
App settings have been redacted. Use 'az webapp/logicapp/functionapp config appsettings list' to view.
[
  {
    "name": "WEBSITE_NODE_DEFAULT_VERSION",
    "slotSetting": false,
    "value": null
  },
  {
    "name": "EventHubConsumerGroup",
    "slotSetting": false,
    "value": null
  },
  {
    "name": "IotHubConnectionString",
    "slotSetting": false,
    "value": null
  }
]
PS /home/tiago> az webapp config set -n App-web3 -g Ubuntu_group --web-sockets-enabled true
```

12. Ative o protocolo WebSocket para o aplicativo web e configure-o para aceitar exclusivamente solicitações HTTPS (solicitações HTTP serão redirecionadas para HTTPS) utilizando os comandos abaixo. Lembre-se de substituir as opções YOUR\_WEB\_APP\_NAME e YOUR\_RESOURCE\_GROUP\_NAME pelas configurações do seu aplicativo.

```
az webapp config set -n YOUR_WEB_APP_NAME -g YOUR_RESOURCE_GROUP_NAME  
--web-sockets-enabled true
```

```
az webapp update -n YOUR_WEB_APP_NAME -g YOUR_RESOURCE_GROUP_NAME --  
https-only true
```



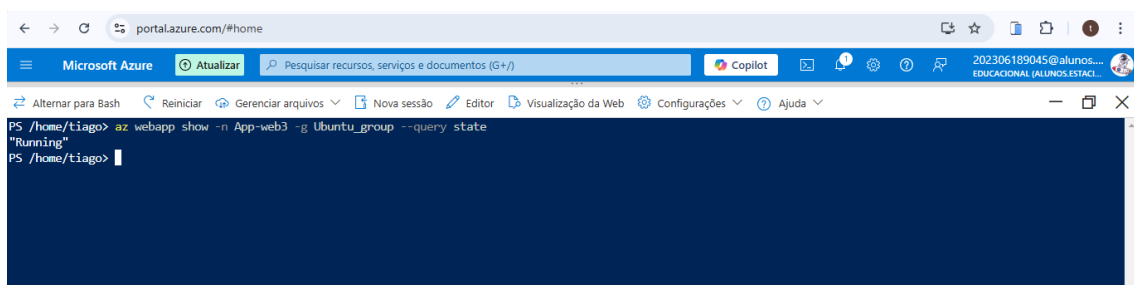
The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a user profile. The main content area displays the configuration for a web application. The command prompt shows the execution of the 'az webapp config set' command, which sets the 'web-sockets-enabled' property to 'true'. The resulting JSON configuration is displayed below the command, showing various properties such as 'acrUserManagedIdentityCreds', 'apiDefinition', 'appSettings', 'autoHealEnabled', 'autoHealRules', 'autoSwapSlotName', 'azureStorageAccounts', 'connectionStrings', 'cors', 'defaultDocuments', 'detailedErrorLoggingEnabled', 'documentRoot', and 'elasticWebAppScaleLimit'.

```
PS /home/tiago> az webapp config set -n App-web3 -g Ubuntu_group --web-sockets-enabled true  
{  
  "acrUserManagedIdentityCreds": false,  
  "acrUserManagedIdentityId": null,  
  "alwaysOn": false,  
  "apiDefinition": null,  
  "apiManagementConfig": null,  
  "appCommandLine": "",  
  "appSettings": null,  
  "autoHealEnabled": false,  
  "autoHealRules": null,  
  "autoSwapSlotName": null,  
  "azureStorageAccounts": {},  
  "connectionStrings": null,  
  "cors": null,  
  "defaultDocuments": [  
    "Default.htm",  
    "Default.html",  
    "Default.asp",  
    "index.htm",  
    "index.html",  
    "iisstart.htm",  
    "default.aspx",  
    "index.php",  
    "hostingstart.html"  
  ],  
  "detailedErrorLoggingEnabled": false,  
  "documentRoot": null,  
  "elasticWebAppScaleLimit": 0,  
}
```

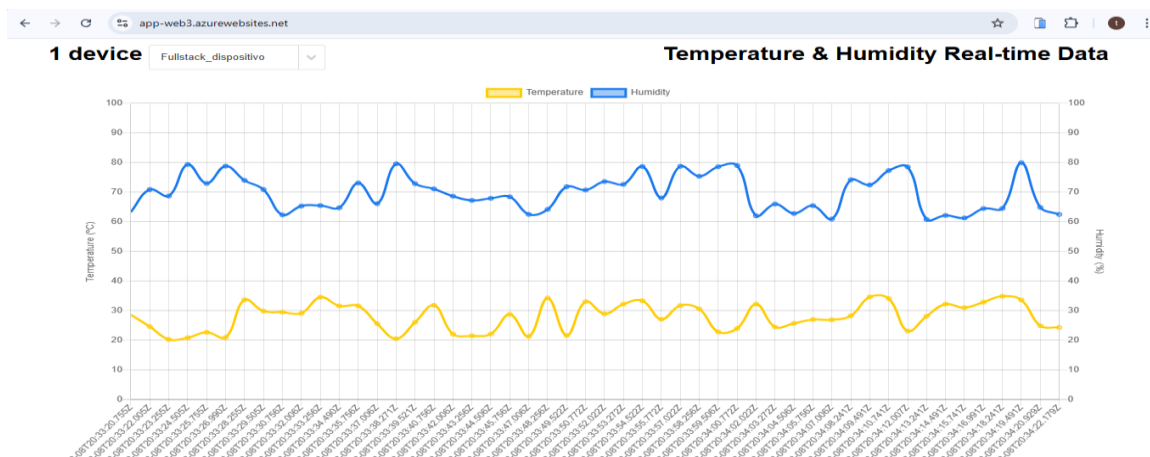
13. Execute o comando a seguir para consultar o status do seu aplicativo web e assegurar-se de que está em execução, lembrando de ajustar os parâmetros.

```
az webapp show -n YOUR_WEB_APP_NAME -g YOUR_RESOURCE_GROUP_NAME --query state
```

Se a resposta for “Running”, a aplicação está funcionando perfeitamente!



14. Após isso, acesse <https://<seu nome de aplicativo web>.azurewebsites.net> em um navegador. Você verá uma página da web semelhante àquela que visualizou ao executar o aplicativo web localmente. Supondo que o dispositivo esteja em execução e enviando dados, você deverá observar um gráfico em tempo real das 50 leituras mais recentes de temperatura e umidade enviadas pelo dispositivo.



## - Resultados esperados 🌟

Ao finalizar esta atividade, espera-se que o aluno tenha realizado com sucesso a visualização em tempo real de dados provenientes de sensores conectados ao Azure Hub IoT. A visualização deverá ser realizada através de aplicativos locais como através do Serviço de Aplicativo do Azure.

Segue abaixo um Link onde mostra passo a passo do trabalho e da ligação tanto Local Pela Porta 3000 como pela Web através da plataforma Azure:

<https://youtu.be/m4TdKs2gxv8>