

# Activity 5

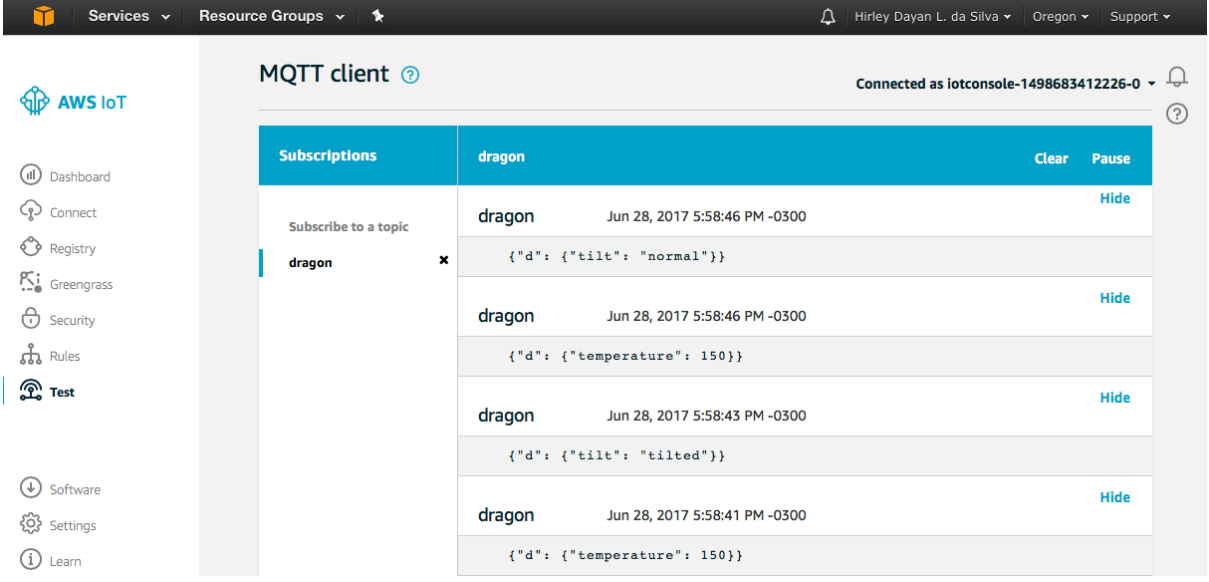
**IoT Platforms Activity**

**AWS, IBM, Microsoft**

**Edvaldo Santos  
Helio Nakazato  
Hirley Dayan**

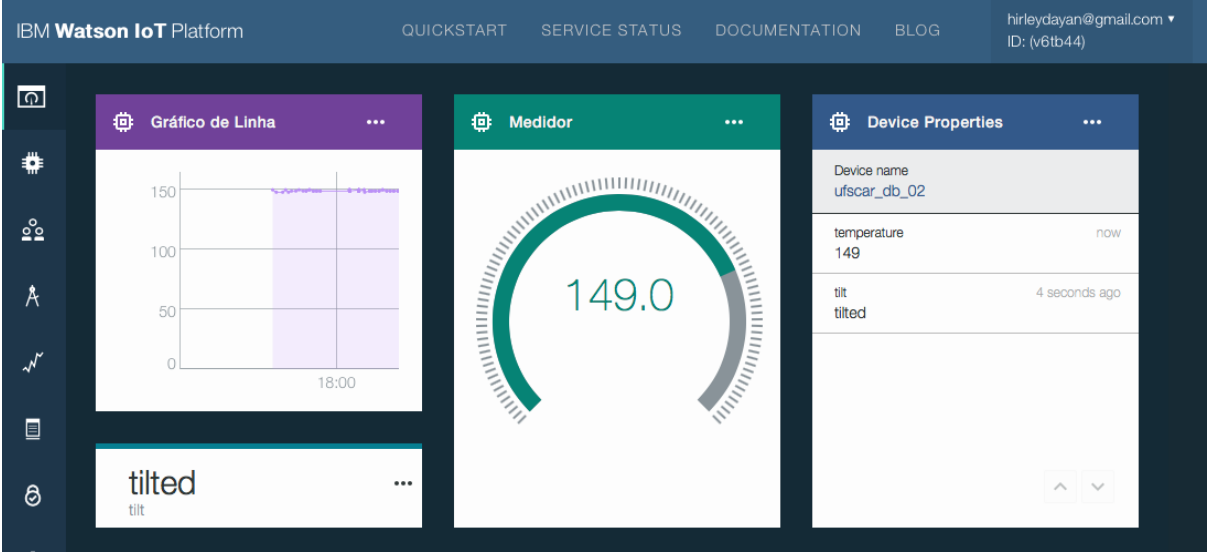
# Exercise 1 and Exercise 3

- Two sensors, one for temperature and another for tilt, had the status collected and published, using MQTT, to AWS IoT Platform and IBM IoT Platforms.
- Instead of using AWS and IBM SDKs, it was chosen to use *paho* library so that code could be reused for both AWS and IBM platforms.



The screenshot shows the AWS IoT console's MQTT client interface. The left sidebar contains navigation links: Dashboard, Connect, Registry, Greengrass, Security, Rules, Test, Software, Settings, and Learn. The main panel is titled 'MQTT client' and shows the user is 'Connected as lotconsole-1498683412226-0'. A table lists subscriptions for the topic 'dragon'.

Subscriptions	dragon	Clear	Pause
Subscribe to a topic	dragon Jun 28, 2017 5:58:46 PM -0300		Hide
	{ "d": { "tilt": "normal" } }		Hide
	dragon Jun 28, 2017 5:58:46 PM -0300		Hide
	{ "d": { "temperature": 150 } }		Hide
	dragon Jun 28, 2017 5:58:43 PM -0300		Hide
	{ "d": { "tilt": "tilted" } }		Hide
	dragon Jun 28, 2017 5:58:41 PM -0300		Hide
	{ "d": { "temperature": 150 } }		Hide

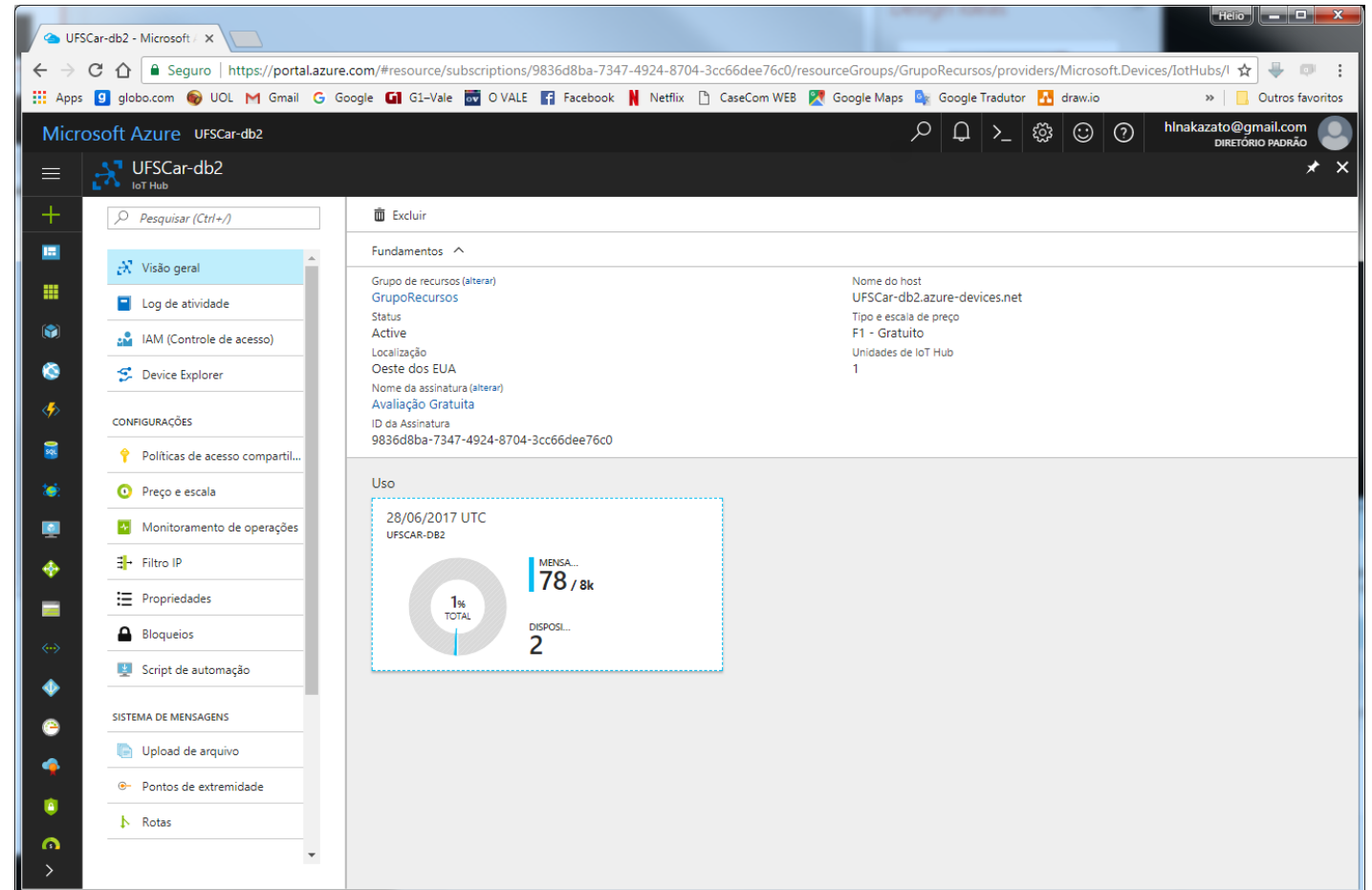


The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes links for QUICKSTART, SERVICE STATUS, DOCUMENTATION, and BLOG, along with the user's email and ID. The dashboard features three main widgets: a line graph titled 'Gráfico de Linha' showing a constant value of 150, a gauge titled 'Medidor' showing a value of 149.0, and a 'Device Properties' panel for the device 'ufscar\_db\_02'.

Device Properties	
Device name	ufscar_db_02
temperature	149 now
tilt	tilted 4 seconds ago

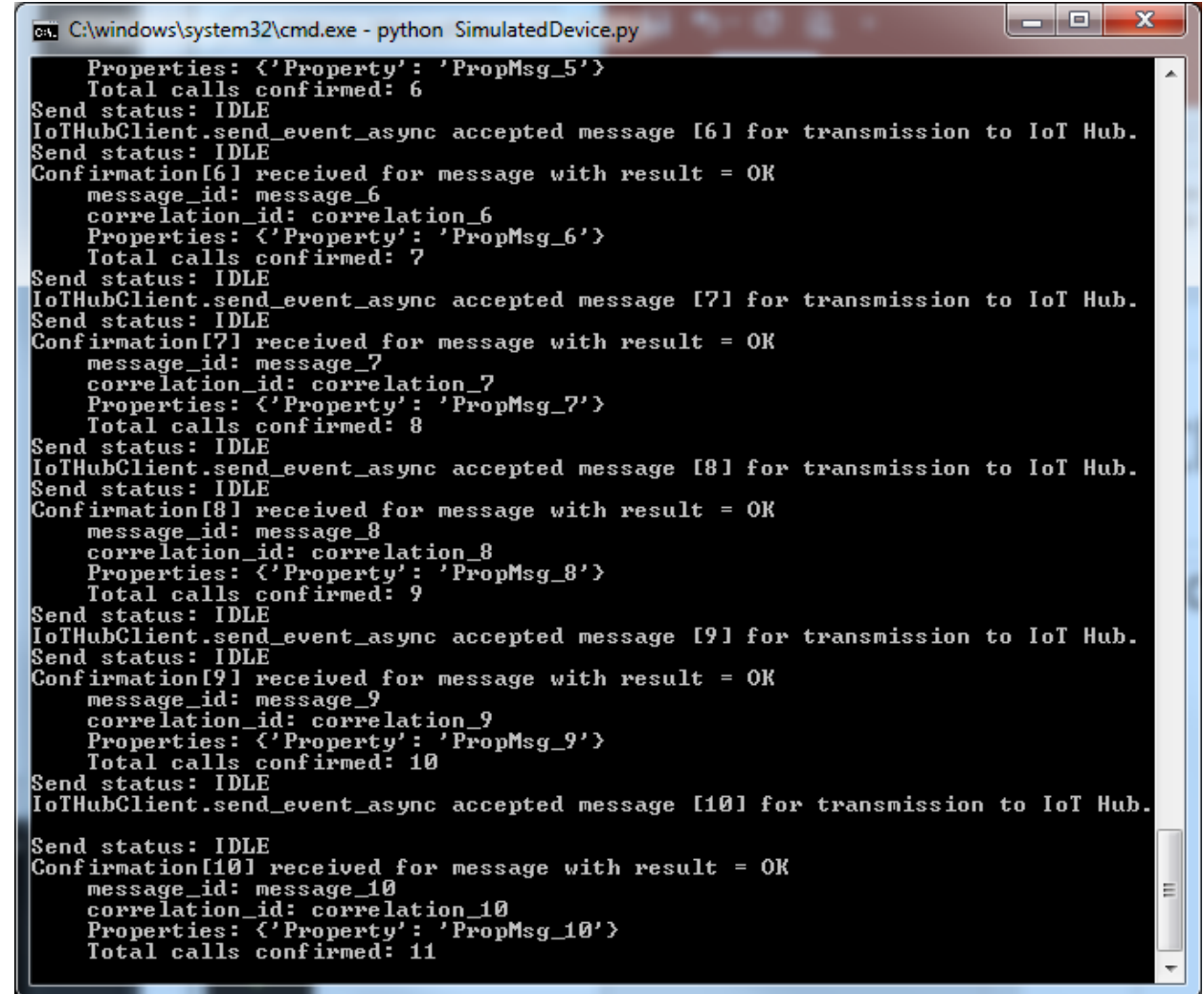
# Exercise 2 - Microsoft IoT Hub

- Account created
- Host created (UFSCar-db2.azure-devices.net)
- New device (TemperatureDevice) created
- Primary key:  
xRQV5mqMaBU46rSoB2  
X8PS4/D98dIED1wJSFjvl  
M6Wo=



# Exercise 2 - Microsoft IoT Hub (Simulated)

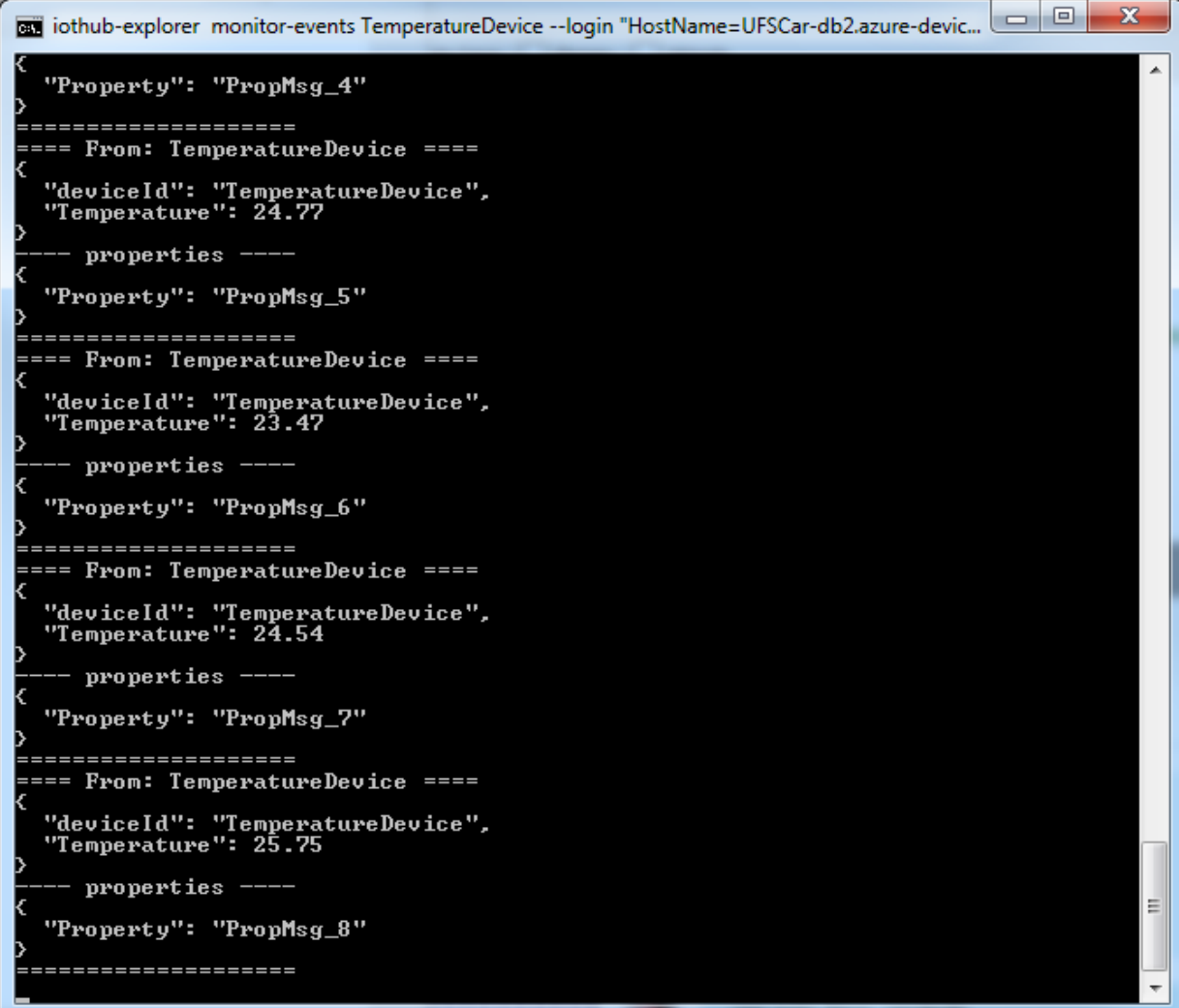
- Python code created (SimulatedDevice.py) to simulate AMQP messages being sent to Microsoft IoT Hub Based on the tutorial: <https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/iot-hub/iot-hub-python-getstarted.md>



```
C:\windows\system32\cmd.exe - python SimulatedDevice.py
Properties: {'Property': 'PropMsg_5'}
Total calls confirmed: 6
Send status: IDLE
IoTHubClient.send_event_async accepted message [6] for transmission to IoT Hub.
Send status: IDLE
Confirmation[6] received for message with result = OK
message_id: message_6
correlation_id: correlation_6
Properties: {'Property': 'PropMsg_6'}
Total calls confirmed: 7
Send status: IDLE
IoTHubClient.send_event_async accepted message [7] for transmission to IoT Hub.
Send status: IDLE
Confirmation[7] received for message with result = OK
message_id: message_7
correlation_id: correlation_7
Properties: {'Property': 'PropMsg_7'}
Total calls confirmed: 8
Send status: IDLE
IoTHubClient.send_event_async accepted message [8] for transmission to IoT Hub.
Send status: IDLE
Confirmation[8] received for message with result = OK
message_id: message_8
correlation_id: correlation_8
Properties: {'Property': 'PropMsg_8'}
Total calls confirmed: 9
Send status: IDLE
IoTHubClient.send_event_async accepted message [9] for transmission to IoT Hub.
Send status: IDLE
Confirmation[9] received for message with result = OK
message_id: message_9
correlation_id: correlation_9
Properties: {'Property': 'PropMsg_9'}
Total calls confirmed: 10
Send status: IDLE
IoTHubClient.send_event_async accepted message [10] for transmission to IoT Hub.
Send status: IDLE
Confirmation[10] received for message with result = OK
message_id: message_10
correlation_id: correlation_10
Properties: {'Property': 'PropMsg_10'}
Total calls confirmed: 11
```

# Exercise 2 - Microsoft IoT Hub (Simulated)

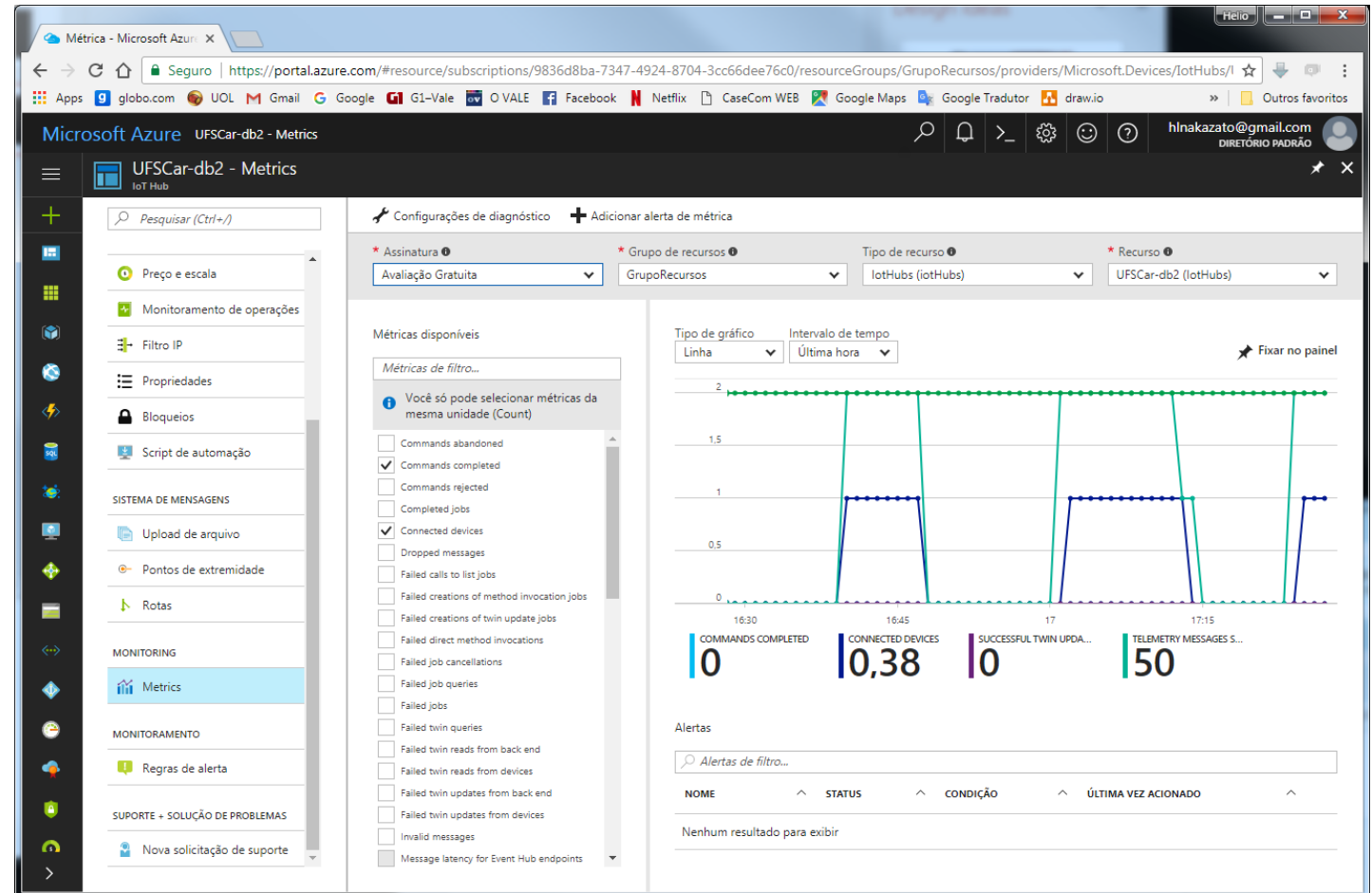
- Install IoT Hub Explorer to receive messages from IoT Hub  
npm install -g iotHub-explorer
- Run the following command to start monitoring  
iotHub-explorer monitor-events TemperatureDevice --login "[IoT Hub connection string]"
- Messages can be seen in the screen



```
iotHub-explorer monitor-events TemperatureDevice --login "HostName=UFSCar-db2.azure-devic...  
< "Property": "PropMsg_4"  
>  
===== From: TemperatureDevice =====  
< "deviceId": "TemperatureDevice",  
  "Temperature": 24.77  
>  
---- properties ----  
< "Property": "PropMsg_5"  
>  
===== From: TemperatureDevice =====  
< "deviceId": "TemperatureDevice",  
  "Temperature": 23.47  
>  
---- properties ----  
< "Property": "PropMsg_6"  
>  
===== From: TemperatureDevice =====  
< "deviceId": "TemperatureDevice",  
  "Temperature": 24.54  
>  
---- properties ----  
< "Property": "PropMsg_7"  
>  
===== From: TemperatureDevice =====  
< "deviceId": "TemperatureDevice",  
  "Temperature": 25.75  
>  
---- properties ----  
< "Property": "PropMsg_8"  
>  
=====
```

# Exercise 2 - Microsoft IoT Hub (Simulated)

- In Microsoft IoT Hub it is possible to check the messages received/sent



# Exercise 2 - Microsoft IoT Hub (Simulated)

- Our impressions after working with Microsoft IoT Hub
  - Not easy to setup
  - Works better (to not say only) on Windows environment since Python libraries “azure-iot-hub-service-client” and “azure-iot-hub-device-client” are currently available only for Windows (ref <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-python-getstarted>). **That is the reason to not run it on Dragon Board.**
  - An attempt was tried to compile the libraries for Linux, following the guideline in <https://github.com/Azure/azure-iot-sdk-python/blob/master/doc/python-devbox-setup.md>, however **“memory unavailable” error was raised even working with 4GB RAM.**
  - Poor dashboard compared to other tools, like IBM and Amazon. For instance, there is no dashboard to see the data coming from the devices.