

PROD. TEC. ITV. MI- N0017/2021

DOI: 10.29223/PROD.TEC.ITV.MI.2021.17.Matos

PRODUÇÃO TÉCNICA ITV MI

UTILIZAÇÃO DO MICROSOFT AZURE PARA IIOT

Relatório parcial do projeto OptiPlant

Autores ITV:

Saulo Matos

Thomas Vargas

Thiago Euzébio

Autores parceiros:

Vinicius Moreira

Ouro Preto

Minas Gerais, Brasil

Agosto/2021

Título: Utilização do Microsoft Azure para IIoT	
PROD. TEC. ITV. MI- N0017/2021	Versão
Classificação: () Confidencial () Restrita () Uso interno (X) Publico	01

Informações Confidenciais - Informações estratégicas para o Instituto e sua Mantenedora. Seu manuseio é restrito a usuários previamente autorizados pelo Gestor da Informação.

Informações Restritas - Informação cujo conhecimento, manuseio e controle de acesso devem estar limitados a um grupo restrito de empregados que necessitam utilizá-la para exercer suas atividades profissionais.

Informações de Uso Interno - São informações destinadas à utilização interna por empregados e prestadores de serviço.

Informações Públicas - Informações que podem ser distribuídas ao público externo, o que, usualmente, é feito através dos canais corporativos apropriados.

Dados Internacionais de Catalogação na Publicação (CIP)

M425u

Matos, Saulo

Utilização do Microsoft Azure para IIoT. Saulo Matos...[et al.]

Ouro Preto, MG: ITV, 2021.

25 p.:il.

Relatório Técnico (Instituto Tecnológico Vale) – 2021

PROD.TEC.ITV.MI – N0017/2021

DOI: 10.29223/PROD.TEC.ITV.MI.2021.17.Matos

I. Vargas, Thomas. II. Euzébio, Thiago. III. Moreira, Vinícius.

V. Título.

CDD 23. ed. 629.8932

RESUMO EXECUTIVO

São apresentados neste documento a descrição e o funcionamento de serviços do Microsoft Azure, tais como o IoT *Hub*, *Storage Blob* e *Stream Analytics*. Por meio destes, é possível o desenvolvimento de aplicações de IoT e IIoT, nas quais é possível monitorar dados medidos em campo de forma remota via *web*. Além disso, é descrito um tutorial para o uso desses serviços, por via de um exemplo prático, no qual utilizou-se um mini computador de placa única do tipo Raspberry Pi, para o envio de dados à nuvem. Foram enviados dois tipos de dados: um dado randômico, gerado pelo *script* desenvolvido, para simular uma medição de um sensor e fotos oriundas de uma câmera conectada ao Raspberry Pi. Os dados randômicos foram enviados via protocolo MQTT e recepcionados pelo Azure IoT *Hub*. Já os *frames* de captura da câmera enviados via protocolo HTTP. Ambos os dados foram armazenados em contêineres por meio do Azure *Storage Blob*.

RESUMO

São apresentados neste documento a descrição e o funcionamento de serviços do Microsoft Azure, que por meio deles, é possível o desenvolvimento de aplicações de IoT e IIoT. Dessa maneira, é possível monitorar dados medidos em campo de forma remota via *web*. Além disso, é descrito um tutorial para o uso desses serviços, por via de um exemplo prático, no qual utilizou-se um mini computador de placa única, para o envio de dados à nuvem. Foram enviados dois tipos de dados: um dado randômico, para simular uma medição de um sensor e fotos oriundas de uma câmera. Os dados foram enviados à nuvem e armazenados em contêineres do Azure.

Palavras-chave: IIoT. Microsoft Azure. Telemetria. Comunicação Remota. Indústria 4.0.

ABSTRACT

This document presents the description and operation of Microsoft Azure's services, through which is possible the development of IoT and IIoT applications. In this way, it is possible to remotely monitor data measured in the field via web. In addition, a tutorial for the use of these services is described, via a practical example, in which a single-board mini computer was used to send data to the cloud. Two types of data were sent: random data, to simulate a sensor measurement and photos from a camera. Data was uploaded to the cloud and stored in Azure's containers.

Keywords: IIoT. Microsoft Azure. Telemetry. Remote Communication. 4.0 Industry.

LISTA DE FIGURAS

Figura 1 - Princípios essenciais da solução IoT.	11
Figura 2 - Exemplo de arquitetura de nuvem de uma solução IoT.	13
Figura 3 - Criação do recurso.	15
Figura 4 - Configuração do recurso.	16
Figura 5 - Tipos e escalas de preço.	16
Figura 6 - Criação do dispositivo de IoT.	17
Figura 7 - Configurações do dispositivo de IoT.	17
Figura 8 - Criação da conta de armazenamento.	18
Figura 9 - Chaves de acesso ao contêiner.	19
Figura 10 - Contêiner criado.	19
Figura 11 - Criação do <i>Stream Analytics Job</i>	20
Figura 12 - Entrada e saída do <i>Stream Analytics Job</i>	20
Figura 13 - Configuração da saída do <i>Stream Analytics Job</i>	21
Figura 14 - Arquivo do contêiner “sensordata”.	23
Figura 15 - Arquivo do contêiner “fotos”.	23

LISTA DE TABELAS

Tabela 1 - Tabela de identificação dos autores	10
---	----

LISTA DE SÍMBOLOS E ABREVIATURAS

AMQP - *Advanced Message Queuing Protocol*

Blob - *Binary large object*

HTTP - *Hypertext Transfer Protocol*

IIoT - *Industrial Internet of Things*

IoT - *Internet of Things*

ITV - *Instituto Tecnológico Vale*

JSON - *Java Script Object Notation*

MQTT - *Message Queuing Telemetry Transport)*

SUMÁRIO

1	INTRODUÇÃO	9
1.1	CONTEXTO	9
1.2	MOTIVAÇÃO	9
1.3	OBJETIVO	10
1.4	IDENTIFICAÇÃO DOS PARTICIPANTES DO PROJETO	10
2	MICROSOFT AZURE	11
2.1	<i>IOT HUB</i>	11
2.2	<i>AZURE STREAM ANALYTICS</i>	12
2.3	<i>AZURE STORAGE BLOBS</i>	13
3	METODOLOGIA	15
3.1	<i>AZURE IOT HUB</i>	15
3.2	<i>AZURE STORAGE BLOBS</i>	18
3.3	<i>AZURE STREAM ANALYTICS</i>	19
3.4	<i>GATEWAY DE IOT</i>	21
3.5	ENVIO DE BLOBS PARA O CONTÊINER	22
4	COMENTÁRIOS FINAIS	24
	REFERÊNCIAS	25

1 INTRODUÇÃO

1.1 CONTEXTO

O termo “Indústria 4.0” foi mencionado pela primeira vez em 2011, na feira anual Hannover Messe, ocorrida na Alemanha e é normalmente utilizado para definir o momento histórico, o qual representa a 4^a Revolução Industrial. Tal marco tornou-se necessário, uma vez que observou-se uma estagnação no mercado industrial, devido a mudanças que alteraram a dinâmica do mercado, de forma que mudaram as necessidades dos clientes e as estratégias de fabricação. Para retomar ao estado de alta eficiência e produtividade, o setor industrial teve que se adaptar, criando-se o conceito de manufatura ágil, aplicando conceitos de flexibilidade e rápida tomada de decisão. Para isso, foram aproveitadas as invenções oriundas do desenvolvimento tecnológico, anteriormente apenas percebidas em áreas fora da indústria.

Conceitos como IoT (*Internet of Things*), *big data* e computação de borda são muitas vezes associados ao fenômeno supracitado e vêm ganhando destaque atualmente. Eles são capazes de tornar a fábrica inteligente, sendo capaz de se adaptar as adversidades, de forma a trocar informações de maneira transparente com a tomada de decisões descentralizada, efetuar uma conexão entre *hardware*, *software* e inteligência computacional, promover a perfeita colaboração entre robô e operador e utilizar dados históricos para que sejam analisados.

1.2 MOTIVAÇÃO

A indústria da mineração vem se desenvolvendo tecnologicamente, buscando aprimorar a qualidade do produto e aumento de produtividade. Além disso, a extração e o tratamento de minério ocorrem em ambientes desafiadores, muitas vezes insalubres, os quais podem oferecer riscos à saúde e a segurança de profissionais. O uso de tecnologias como o sensoriamento remoto, automação e monitoração de processos de forma remota, são boas alternativas para lidar com os problemas gerados pelo ambiente hostil da indústria da mineração.

Abordagens utilizando a IIoT (*Industrial Internet of Things*), surgem com o intuito de auxiliar esse crescente desenvolvimento tecnológico. Com o uso de ferramentas da Microsoft Azure é possível promover essa tecnologia, de modo a estabelecer uma comunicação bidimensional, rápida e segura entre dispositivos e a nuvem. Dessa maneira, é possível armazenar, processar e analisar os dados obtidos por meio dos dispositivos.

1.3 OBJETIVO

Descrever as ferramentas da Microsoft Azure usadas para envio, processamento e análise de dados oriundos de dispositivos IoT e apresentar um tutorial, cujo intuito é mostrar o desenvolvimento de uma aplicação que promova a comunicação entre dispositivos com a nuvem, bem como o armazenamento de dados utilizando o Microsoft Azure.

1.4 IDENTIFICAÇÃO DOS PARTICIPANTES DO PROJETO

Os autores e instituições vinculadas podem ser identificados por meio da Tabela 1.

Tabela 1: Tabela de identificação dos autores

Nome	Instituição	Departamento
Saulo Matos	ITV	Controle e Robótica
Thomas Vargas	ITV	Controle e Robótica
Thiago Euzébio	ITV	Controle e Robótica
Vinicius Moreira	Vale	X

Fonte: O Autor.

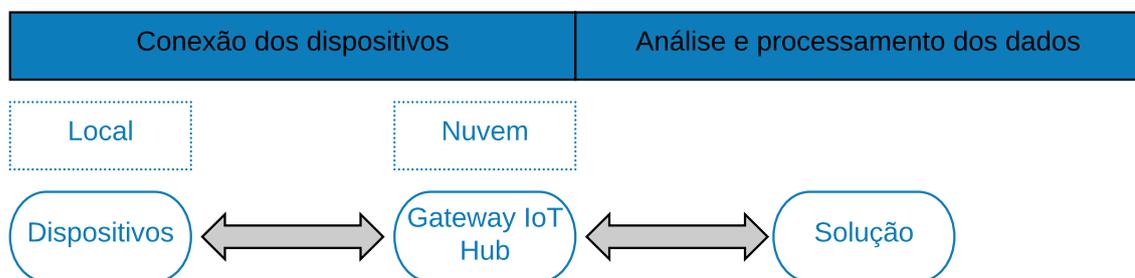
2 MICROSOFT AZURE

Microsoft Azure é uma marca abrangente para os serviços de computação em nuvem da Microsoft. A ferramenta abrange uma ampla, e ainda crescente, gama de serviços que frequentemente formam os elementos fundamentais da computação em nuvem (COPELAND et al., 2015). Nesta seção, serão abordados alguns serviços Azure, os quais são utilizados para comunicação com a nuvem e armazenamento, processamento e análise de dados.

2.1 IOT HUB

Trata-se de um serviço de comunicações totalmente gerenciável que fornece mensagens altamente seguras, confiáveis e escaláveis entre dispositivos IoT e soluções IoT. Um dispositivo IoT é qualquer eletrônico capaz de comunicar com outros sistemas por meio de uma conexão sem fio. O IoT Hub do Azure pode ser visto como um *gateway* de grande escala que atua como um capacitador e gerenciador de toda comunicação bidirecional entre dispositivos entre si e com a nuvem (KLEIN, 2017). Conforme é mostrado na Figura 1, as soluções de IoT podem ser divididas em dois núcleos, que são a conexão do dispositivo e o processamento e análise de dados.

Figura 1: Princípios essenciais da solução IoT.



Fonte: O Autor.

A conectividade do dispositivo representa os dispositivos que geram e coletam dados que são enviados para o *gateway* na nuvem. O *gateway* atua como um mediador que reúne os dados recebidos e os disponibiliza para processamento posterior por outros serviços e processos da solução IoT. O Azure IoT Hub promove uma comunicação bidirecional entre o dispositivo e a nuvem. O uso do IoT Hub como *gateway* em uma solução de IoT, promove diversas vantagens (KLEIN, 2017):

- Mensagens confiáveis de dispositivo para nuvem e vice versa;
- Registro de dispositivos em tempo real;

- Comunicação segura por meio de credenciais de segurança por dispositivo e controle de acesso;
- Monitoramento extensivo da conectividade do dispositivo;
- Monitoramento de eventos;
- Bibliotecas e SDKs disponíveis para a maioria das linguagens de programação, tais como C, C#, Python, Java e Node.js.

Além disso, O IoT *Hub* suporta uma variedade de protocolos IoT populares para enfileiramento e transmissão de dados, incluindo HTTPS, AMQP, AMQP sobre *WebSockets*, MQTT e MQTT sobre *WebSockets* (STACKOWIAK, 2019). Em que, AMQP (*Advanced Message Queuing Protocol*) é um protocolo de camada de aplicação e MQTT (*Message Queuing Telemetry Transport*) é um protocolo de conectividade do tipo máquina a máquina, o qual é o mais recomendável para aplicações de IoT, posto que foi projetado para subscrição e publicação de mensagens de forma mais rápida e segura que os protocolos convencionais (KLEIN, 2017).

2.2 AZURE STREAM ANALYTICS

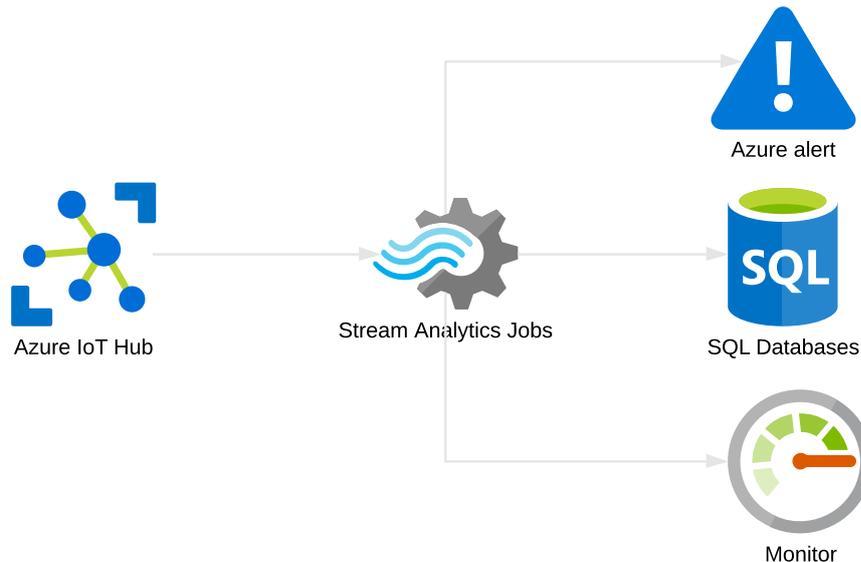
Azure *Stream Analytics* fornece um mecanismo de processamento de eventos que permite o exame e análise de grandes volumes de dados transmitidos por dispositivos. Dessa forma, ele recebe dados do IoT *Hub*, de maneira que pode-se criar trabalhos, de forma a processar os dados usando consultas de transformação SQL para filtrar, classificar, agregar e / ou juntar os dados de *streaming*. Além disso, podem ser visualizados em tempo real por meio de ferramentas como o *Power BI*. Os dados também podem ser enviados ao Azure *Data Lake* para o treinamento de modelos de *machine learning* (STACKOWIAK, 2019). Um exemplo de uso do *Stream Analytics* é mostrado na Figura 2. Klein (2017), descreve os benefícios da utilização do Azure *Stream Analytics*:

- Escala: com centenas e milhares de dispositivos enviando grandes quantidades de dados para a nuvem, o Azure *Stream Analytics* foi projetado e construído para lidar com milhões de eventos por segundo chegando, até 1 GB/segundo. O que torna isso possível é o recursos de particionamento de *Hubs* de eventos;
- Confiabilidade: prevenção contra perda de dados e continuidade de negócios, duas facetas principais do *Stream Analytics* são fornecidos por meio de recursos de recuperação integrados;
- Conectividade: *Stream Analytics* permite conectividade a muitas fontes diferentes e destinos, incluindo *Event Hub* e IoT *Hub* para ingestão de *stream*, bem como

Armazenamento (tabelas e *blobs*), Banco de Dados SQL do Azure, *Azure Data Lake* e *Power BI*;

- Eficácia de custos: O *Stream Analytics* foi projetado para fornecer uma solução analítica que opere em tempo real por um custo baixo.

Figura 2: Exemplo de arquitetura de nuvem de uma solução IoT.



Fonte: O Autor.

2.3 AZURE STORAGE BLOBS

Blob Storage, também conhecido como *Object Storage*, é onde dados de objetos não estruturados podem ser armazenados (DAHER; HAJJDIAB, 2018). *Blob* significa objeto binário grande (*binary large object*). Este tipo de serviço pode armazenar quase tudo, uma vez que armazena dados não estruturados, como documentos, arquivos, imagens, VHDs e assim por diante. Com o serviço de armazenamento de *blobs* do Azure, é possível acessar os objetos armazenados, usando diferentes métodos de acesso, como URLs, APIs REST ou até mesmo os SDKs do Azure (WALY, 2017).

Existem dois tipos de camadas de armazenamento *blob*, que promovem uma melhor relação custo-benefício de acordo com a escolha de uso. A camada de armazenamento quente do Azure é para dados que são acessados com uma alta frequência. Os dados requerem alta durabilidade, disponibilidade, e rápido acesso. Nessa configuração, paga-se menos pelo acesso aos dados, porém o custo é maior de acordo com o tamanho dos dados. Já na camada de armazenamento frio, é para armazenar dados que são de longa duração e raramente acessados. Os dados exigem alta durabilidade de modo que o custo referente ao

tamanho dos dados é inferior e o de acesso aos dados é ligeiramente superior (DAHER; HAJJDIAB, 2018; WALY, 2017). De acordo com Waly (2017), existem o uso de serviços Azure para armazenamento são os seguintes:

- Presença global: é possível hospedar o armazenamento em diversas regiões disponíveis pelo Azure, permitindo o desenvolvimento de aplicações próximas à base do usuário;
- Redundância e recuperação: Azure tem uma presença global que pode ser aproveitada para manter o armazenamento disponibilidade usando replicação de dados, mesmo se ocorrer um desastre em uma região;
- Pagamento conforme o uso: dados podem ser armazenado com um bom custo benefício, sendo de acordo com o uso do usuário.

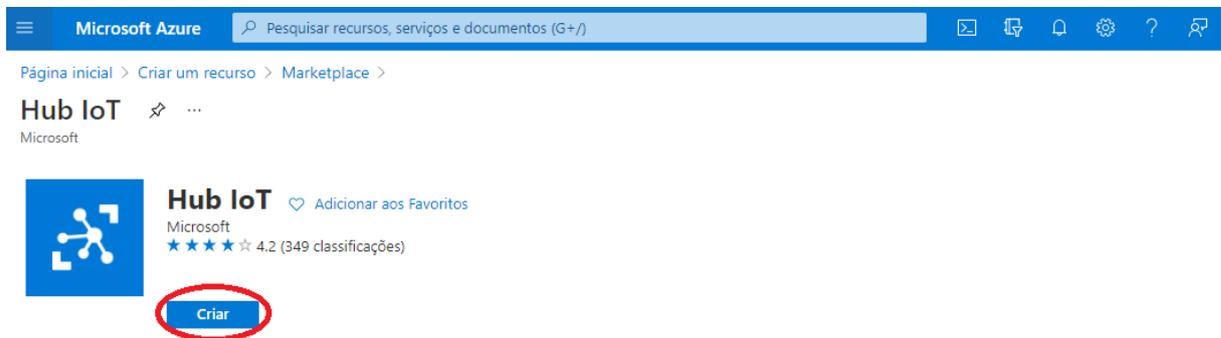
3 METODOLOGIA

Com o intuito de descrever o passo a passo para criar uma solução IoT com o uso do Microsoft Azure, é apresentado um exemplo prático. Foi implementado um *data logger* para um sistema de telemetria, no qual foi utilizado um Raspberry Pi para enviar dados em formato JSON, via MQTT, para a nuvem. O envio dos dados foi feito usando a linguagem de programação Python e utilizou-se os seguintes serviços Azure: o IoT *Hub*, para recepção dos dados, o Azure *Stream Analytics* para coleta dos dados do IoT *Hub* e Azure *Storage Blobs* para armazenamento do *dataset* em contêineres. Além disso, foi configurado o envio de fotos, oriundas de uma câmera conectada ao Raspberry Pi, para contêineres do Azure *Storage Blobs* via protocolo HTTP.

3.1 AZURE IOT HUB

Usando o Portal Azure, foi criada uma aplicação de IoT *Hub*, como mostrado na Figura 3. Após isso, foi configurado o nome da aplicação, o grupo de recursos usado para armazenar os recursos do serviço, a região na qual a aplicação será hospedada, conforme é mostrado na Figura 4.

Figura 3: Criação do recurso.



Fonte: O Autor.

Figura 4: Configuração do recurso.

Microsoft Azure

Pesquisar recursos, serviços e documentos (G+/)

Página inicial > Criar um recurso > Marketplace > Hub IoT >

Hub IoT

Microsoft

Básico Rede Gerenciamento Marcas Revisar + criar

Crie um hub IoT para ajudá-lo a se conectar, monitorar e gerenciar bilhões de ativos de IoT. [Saiba mais](#)

Detalhes do projeto

Escolha a assinatura que você usará para gerenciar as implantações e os custos. Use os grupos de recursos como pastas para ajudar você a organizar e gerenciar recursos.

Assinatura * ⓘ

Grupo de recursos * ⓘ
[Criar novo](#)

Região * ⓘ

Nome do hub IoT * ⓘ

[Revisar + criar](#) < Anterior Avançar: Rede >

Fonte: O Autor.

Na aba gerenciamento pode-se escolher o tipo e a escala do preço, como mostrado na Figura 5 . Neste trabalho optou-se pela camada gratuita, na qual é possível o envio de 8000 mensagens por dia.

Figura 5: Tipos e escalas de preço.

S1: Camada Standard

S1: Camada Standard

S2: Camada Standard

S3: Camada Standard

B1: Tipo básico

B2: Tipo básico

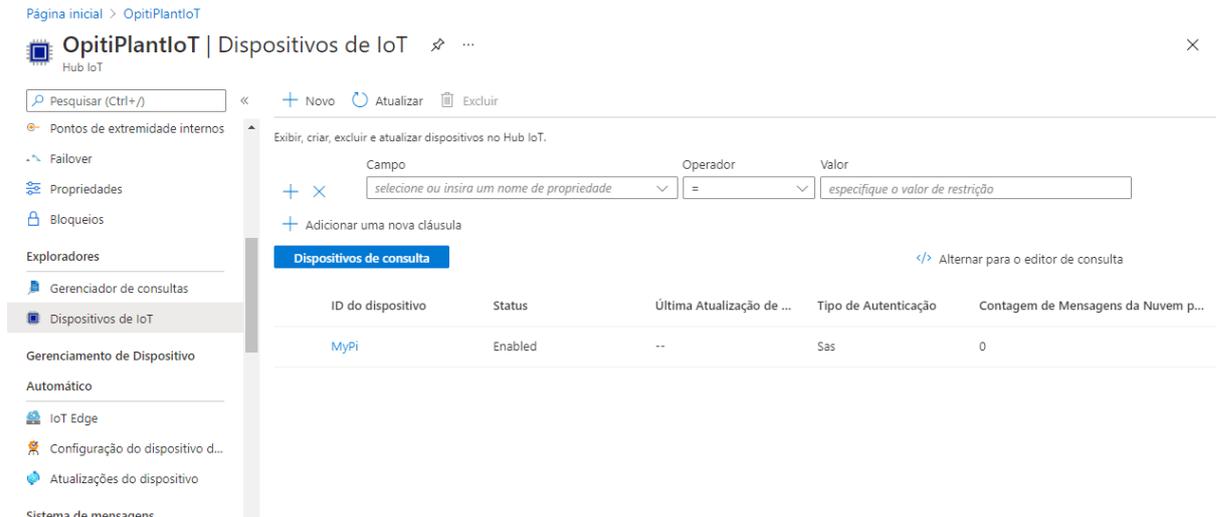
B3: Tipo básico

F1: Camada gratuita

Fonte: O Autor.

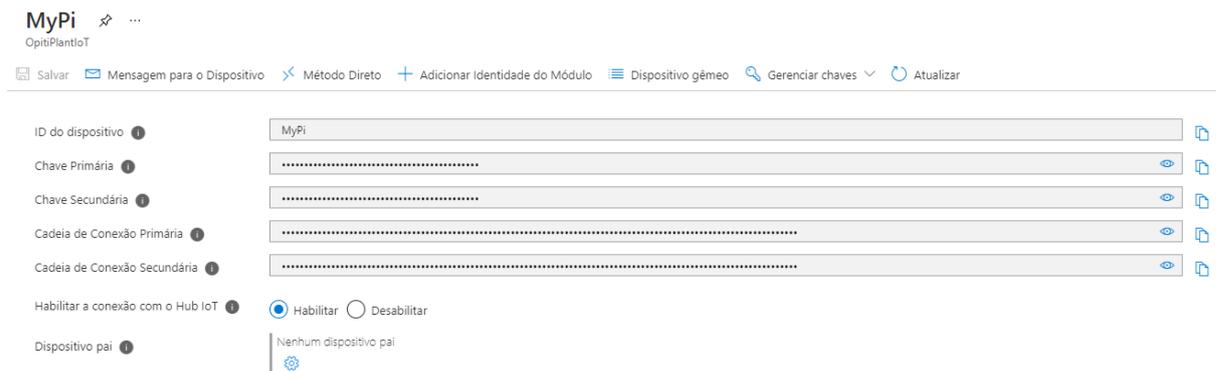
Com o serviço criado, configura-se o dispositivo de IoT que será utilizado como dispositivo de borda, de modo que é possível obter a chave de conexão a ser usada no código exemplo mostrado na Seção 3.4. Na Figura 7 é mostrada a aba de criação do dispositivo de IoT e na Figura 8 são mostradas as chaves de conexão do dispositivo de IoT criado.

Figura 6: Criação do dispositivo de IoT.



Fonte: O Autor.

Figura 7: Configurações do dispositivo de IoT.



Fonte: O Autor.

Inserindo-se a chave primária no código exemplo, a conexão entre o dispositivo de borda e a nuvem está concluída, de forma que o IoT *Hub* da Azure recebe todos os dados randômicos enviados. Usando o Azure CLI é o possível verificar os dados enviados em tempo real usando os comandos listados abaixo. Em que, no primeiro instala-se a extensão para visualizar os dados e no segundo, verifica-se os dados recebidos.

```
1 az extension add --name azure-cli-iot-ext
```

```
1 az iot hub monitor-events --hub-name NomeIoTHub --device-id
  NomeDispositivoIoT
```

3.2 AZURE STORAGE BLOBS

Foi criada uma conta de armazenamento do tipo camada quente e com redundância geográfica, conforme mostrado na Figura 8.

Figura 8: Criação da conta de armazenamento.

Criar uma conta de armazenamento ...

Básico Avançado Rede Proteção de dados Marcas Examinar + criar

O Armazenamento do Azure é um serviço gerenciado pela Microsoft que oferece armazenamento em nuvem altamente disponível, seguro, durável, escalonável e redundante. O Armazenamento do Azure inclui Blobs do Azure (objetos), Arquivos do Azure, Filas do Azure, Tabelas do Azure e o Azure Data Lake Storage Gen2. O custo da sua conta de armazenamento depende do uso e das opções escolhidas abaixo. [Saiba mais sobre as contas de armazenamento do Azure](#)

Detalhes do projeto

Selecione a assinatura na qual a conta de armazenamento será criada. Escolha um grupo de recursos novo ou existente para organizar e gerenciar a conta de armazenamento junto com outros recursos.

Assinatura *	<input type="text" value="ITV Project"/>
Grupo de recursos *	<input type="text" value="OpitPlant-4340050"/>

[Criar novo](#)

Examinar + criar

< Anterior

Avançar: Avançado >

Fonte: O Autor.

Na aba "Chaves de acesso", mostrada na Figura 9, coletou-se a chave a ser usada na configuração da saída do trabalho de *Stream Analytics*.

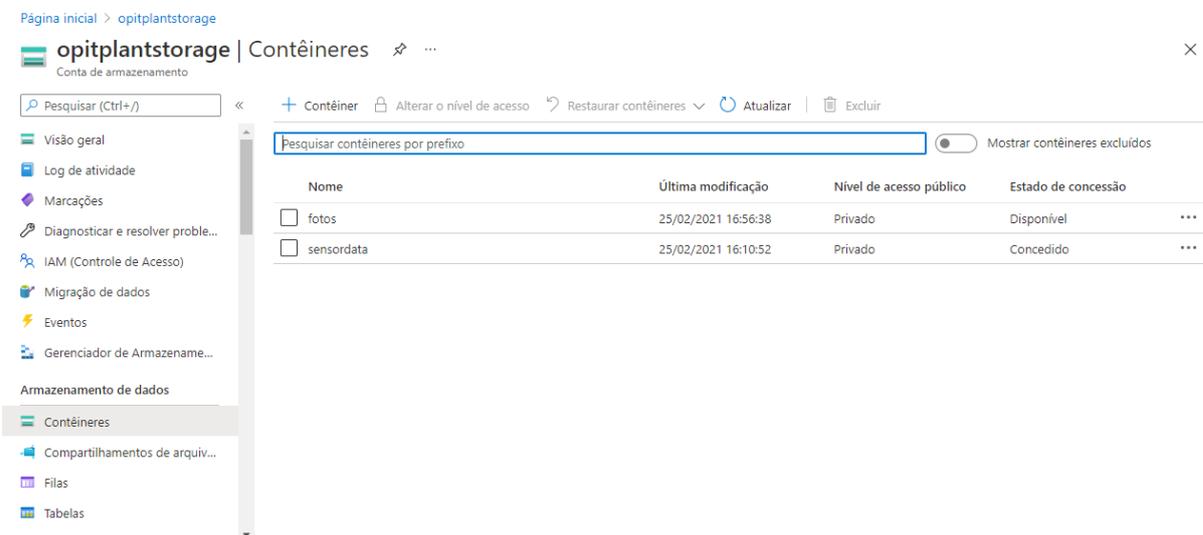
Figura 9: Chaves de acesso ao contêiner.



Fonte: O Autor.

Dentro da aplicação foram criados o contêiner para armazenar os dados em formato JSON são armazenados e o contêiner para o armazenamento das fotos, assim como é mostrado na Figura 10.

Figura 10: Contêiner criado.



Fonte: O Autor.

3.3 AZURE STREAM ANALYTICS

No Portal Azure, foi criado um trabalho do *Stream Analytics*, de modo que ele foi usado para enviar os dados do *IoT Hub* para o armazenamento *blob*. Nessa etapa, configura-se o nome do trabalho, a assinatura, o grupo de recursos a localização e as unidades de *streaming*. Vale ressaltar que a precificação do recurso é feita de acordo com a quantidade

de unidades de *streaming*. Na Figura 11 é mostrada a aba de configuração do *Stream Analytics Job*.

Figura 11: Criação do *Stream Analytics Job*.

Novo trabalho do Stream Analytics ...

! Isso criará um novo trabalho do Stream Analytics. Você será cobrado de acordo com o modelo de cobrança do Azure Stream Analytics. Saiba mais. →

Nome do trabalho *

Assinatura *

ITV Project

Grupo de recursos *

Criar novo

Localização *

Leste dos EUA 2

Ambiente de hospedagem ○

Nuvem Borda

Unidades de streaming (1 a 192) ○

3

Proteger todos os ativos de dados privados necessários para esse trabalho em minha conta de Armazenamento. ○

Criar

Fonte: O Autor.

Foram criados parâmetros de entrada e saída para cada um dos serviços, conforme é mostrado na Figura 12. O parâmetro de entrada representa os dados oriundos do IoT *Hub* do Azure.

Figura 12: Entrada e saída do *Stream Analytics Job*.

▶ Iniciar Interromper Excluir

! Interrompido

Entradas

1

StreamJobInput Hub IoT

Saídas

1

output Armazenamento de blob

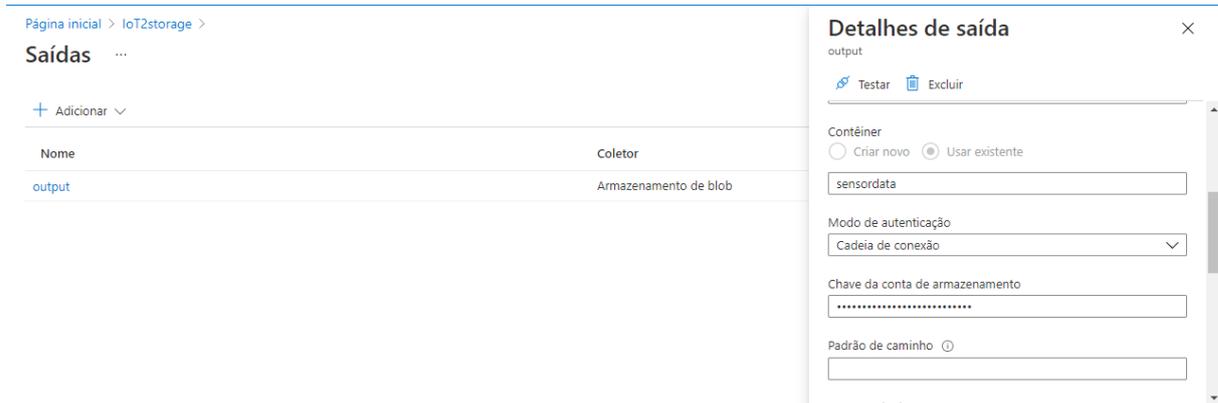
Consulta Editar consulta

```
1 SELECT
2 *
3 INTO
4 output
5 FROM
6 StreamJobInput
7
```

Fonte: O Autor.

O parâmetro de saída foi configurado para envio a um contêiner previamente criado, juntamente com a chave da conta de armazenamento, tipo do arquivo, entre outras configurações. Na Figura 13 é mostrada a configuração do parâmetro de saída, de maneira que esteja condizente com o dado que será enviado.

Figura 13: Configuração da saída do *Stream Analytics Job*.



Fonte: O Autor.

3.4 GATEWAY DE IOT

Um computador de placa única Raspberry Pi usando o sistema operacional Raspbian de *kernel* versão 5.10, no qual representa o dispositivo de borda do sistema IoT, foi usado para implementação um código para envio de dados de telemetria para o Azure IoT *Hub*, por meio do protocolo MQTT. Para tal, foi necessário a instalação das bibliotecas Python: “azure-iot-device”, “azure-iot-hub”, “azure-iothub-service-client” e “azure-iothub-device-client”, disponíveis no SDK de Python para o IoT Hub da Azure.

Para facilitar a manipulação e envio dos dados, optou-se pelo formato JSON (*JavaScript Object Notation*), o qual consiste em um formato leve de troca de informações/dados entre sistemas, muito utilizado na criação de *datasets*. O código utilizado para conectar o dispositivo de borda ao Azure IoT *Hub* e enviar números randômicos via MQTT, é mostrado abaixo.

```

1 # Copyright (c) Microsoft. All rights reserved.
2
3 import random
4 import time
5 from azure.iot.device import IoTHubDeviceClient, Message
6 CONNECTION_STRING = "Chave de conexao do IoT Hub disponivel no portal
   Azure"
7 # Mensagem do tipo JSON a ser enviada
8 MSG_TXT = '{"randomico": {randomico}}'
9 def iothub_client_init():
10     # Cria um cliente IoT Hub

```

```

11     client = IoTHubDeviceClient.create_from_connection_string(
CONNECTION_STRING)
12     return client
13 def iothub_client_telemetry_sample_run():
14     try:
15         client = iothub_client_init()
16         while True:
17             # Cria valores rand micos a serem enviados.
18             randomico = random.random()
19             msg_txt_formatted = MSG_TXT.format(randomico=randomico)
20             message = Message(msg_txt_formatted)
21             # Envio da mensagem.
22             print( "Enviando mensagem: {}".format(message) )
23             client.send_message(message)
24             print ( "Mensagem enviada com sucesso" )
25             time.sleep(3)
26     except KeyboardInterrupt:
27         print ( "IoTHubClient foi parado" )
28 if __name__ == '__main__':
29     print ( "Pressione Ctrl-C para sair" )
30     iothub_client_telemetry_sample_run()

```

3.5 ENVIO DE BLOBS PARA O CONTÊINER

Os *frames* de uma câmera salvos em um diretório do sistema foram enviados via protocolo HTTP para o contêiner “fotos”. Para tal, foram necessárias as bibliotecas “azure.storage.blob” e “os”. Sendo que, a primeira foi usada para o envio dos *blobs* e a segunda para acessar os arquivos presentes no diretório. O código abaixo apresenta a função usada para o envio dos arquivos.

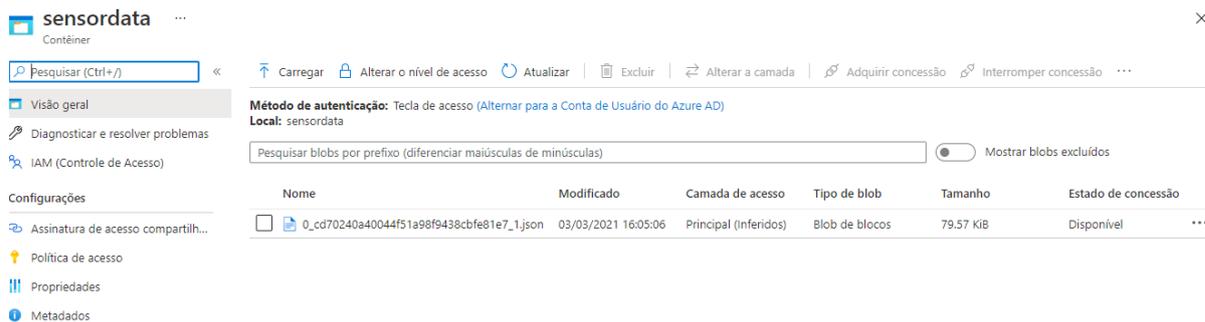
```

1 from azure.storage.blob import BlockBlobService, PublicAccess
2 import os
3 def run_sample():#envia as fotos para o coneteiner no azure
4     block_blob_service = BlockBlobService(account_name='opitplantstorage
', account_key='insira_chave_de_acesso')
5     container_name = 'fotos'
6     local_path = "/home/pi/ros_catkin_ws/src/camera/src"
7     for files in os.listdir(local_path):
8         block_blob_service.create_blob_from_path(container_name,files,os
.path.join(local_path,files))

```

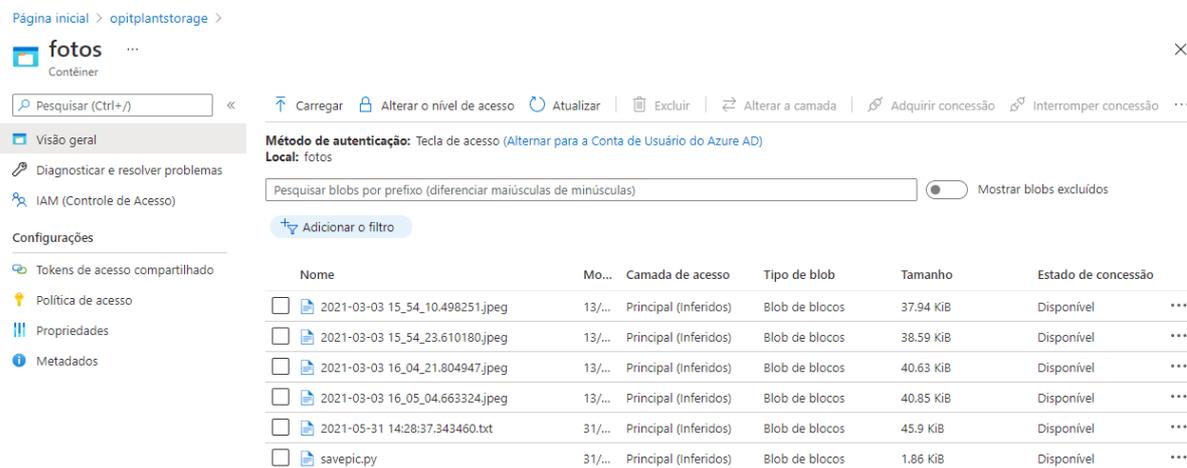
Rodando o código no Raspberry Pi e inicializando o *Stream Analytics Job* foi possível armazenar dados em ambos os contêineres criados. O contêiner “sensordata”, armazenou o arquivo JSON contendo dados aleatórios e o intitulado “fotos”, os arquivos presentes no diretório programado. Nas Figuras 14 e 15 são mostrados os blobs presentes nos contêineres.

Figura 14: Arquivo do contêiner “sensordata”.



Fonte: O Autor.

Figura 15: Arquivo do contêiner “fotos”.



Fonte: O Autor.

4 COMENTÁRIOS FINAIS

A aplicação feita e mostrada neste relatório comprovam como a tecnologia IoT é poderosa e promove segurança e confiabilidade na troca de informações e no armazenamento de dados. A metodologia de criação apresentada é simples e pode ser replicada para qualquer outra aplicação, sendo feitos os devidos ajustes.

Trabalhos envolvendo os temas referentes a IIoT são de extrema importância para a indústria mineral, dado que ela busca se manter produtiva devido as intensas mudanças que estão acontecendo em decorrência do desenvolvimento tecnológico.

REFERÊNCIAS

COPELAND, M. et al. **Microsoft Azure Planning, Deploying, and Managing Your Data Center in the Cloud**. 1. ed. New York - USA: Apress, 2015. ISBN 978-1-4842-1043-7.

DAHER, Zouheir; HAJJDIAB, H. Cloud storage comparative analysis amazon simple storage vs. microsoft azure blob storage. **International Journal of Machine Learning and Computing**, v. 8, p. 85–89, 2018.

KLEIN, Scott. **IoT Solutions in Microsoft's Azure IoT Suite**. 1. ed. New York - USA: Apress, 2017. ISBN 978-1-4842-2143-3.

STACKOWIAK, Robert. **Azure Internet of Things Revealed: Architecture and Fundamentals**. 1. ed. New York - USA: Apress, 2019. ISBN 978-1-4842-5470-7.

WALY, Mohamed. **Learning Microsoft Azure Storage**. 1. ed. Birmingham - UK: Packt Publishing Ltd, 2017. ISBN 978-1-78588-491-7.