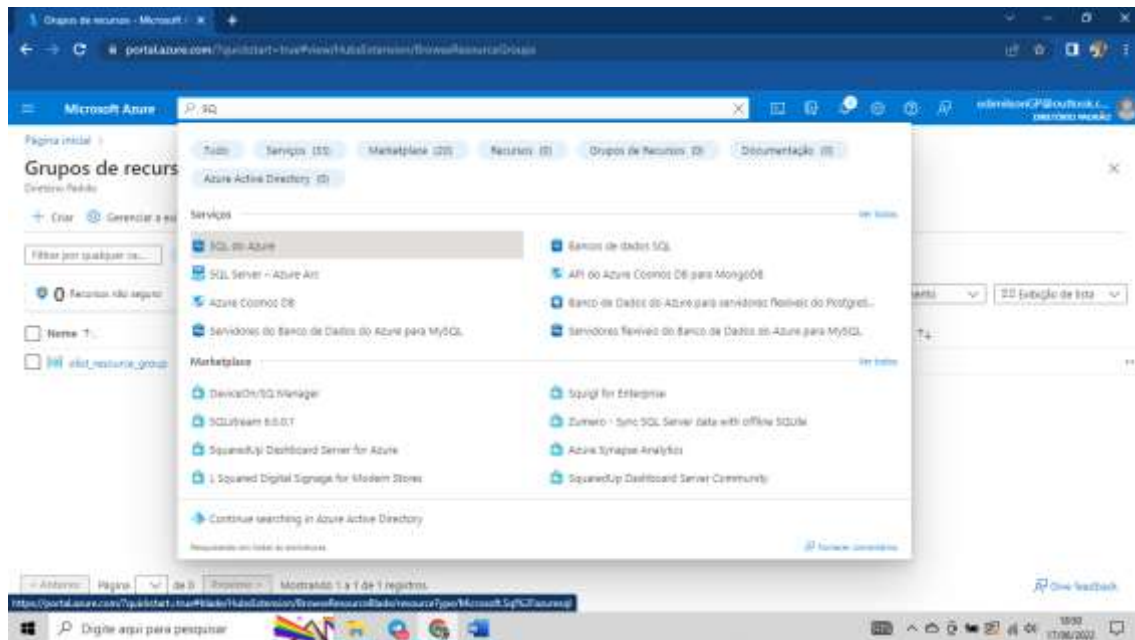
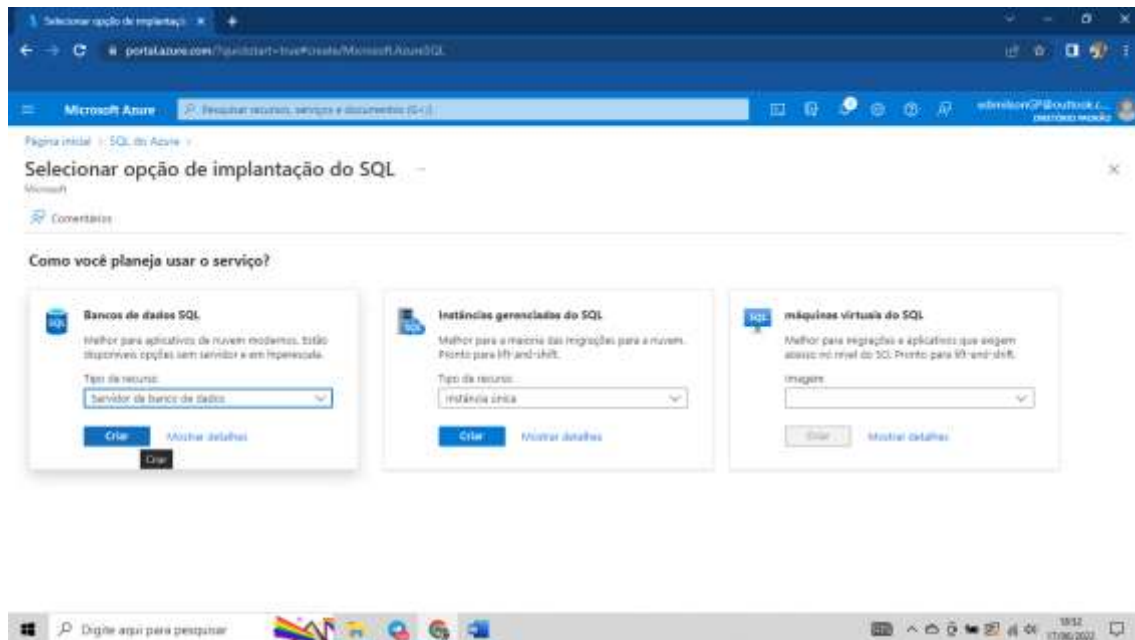


CRIAR BANCO DE DADOS SQL DO AZURE



SELECIONAR EM BANCO DE DADOS SQL

Servidor de banco de dados



Nome do servidor: olist-sql-server1 .database.windows.net

Login do administrador do servidor: olist_admin

Senha:*****

Clickar em AVANÇAR

Microsoft Azure

Página inicial > SQL do Azure > Selecionar opção de implantação do SQL >

Criar Servidor do Banco de Dados SQL

Microsoft

Básico

Rede

Configurações adicionais

Rótulos

Revisar + criar

O servidor do banco de dados SQL é um contêiner lógico para gerenciar bancos de dados e pools elásticos. Conclua a guia [Noções Básicas](#) e acesse [Revisar + Criar](#) para provisionar com padrões inteligentes ou visite cada guia para personalizar. [Saiba mais](#) »

Detalhes do projeto

Selecione a assinatura para gerenciar os custos e os recursos implantados. Use grupos de recursos como pastas para organizar e gerenciar todos os seus recursos.

Assinatura *

Azure subscription 1

Grupo de recursos *

olxst_resource_group

[Criar novo](#)

Detalhes do servidor

Insira as configurações necessárias para este servidor, incluindo um nome e um local.

Nome do servidor *

olxst-sql-server1

database.windows.net

Localização *

(US) East US 2

Autenticação

Selecione seus métodos de autenticação preferidos para acessar este servidor. Crie um logon de administrador servidor e uma senha para acessar seu servidor com autenticação SQL; selecione apenas a autenticação Microsoft Azure Active Directory [Saiba mais](#) usando um usuário, grupo ou aplicativo existente do Microsoft Azure Active Directory como administrador do Microsoft Azure Active Directory [Saiba mais](#) » ou selecione a autenticação do SQL e do Microsoft Azure Active Directory.

Método de autenticação

☒ Usar autenticação SQL

☐ Usar somente a autenticação do Azure Active Directory (Azure AD)

☐ Usar a autenticação do SQL e do Microsoft Azure Active Directory

Logon do administrador do servidor *

olxst_admin

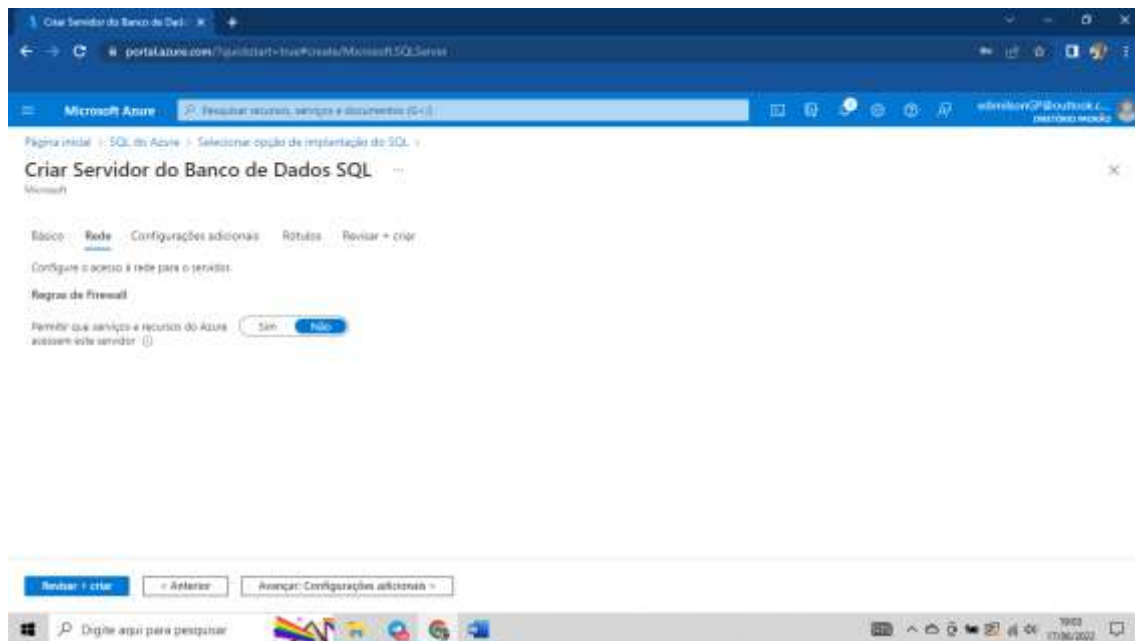
Senha *

Confirmar senha *

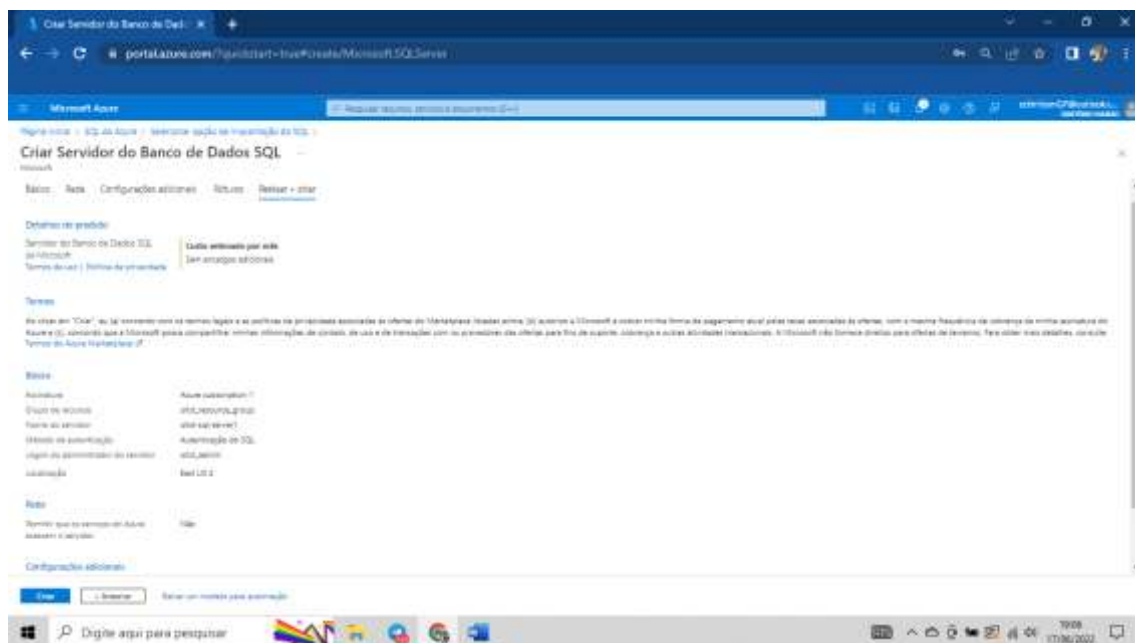
Revisar + criar

Avançar: Rede >

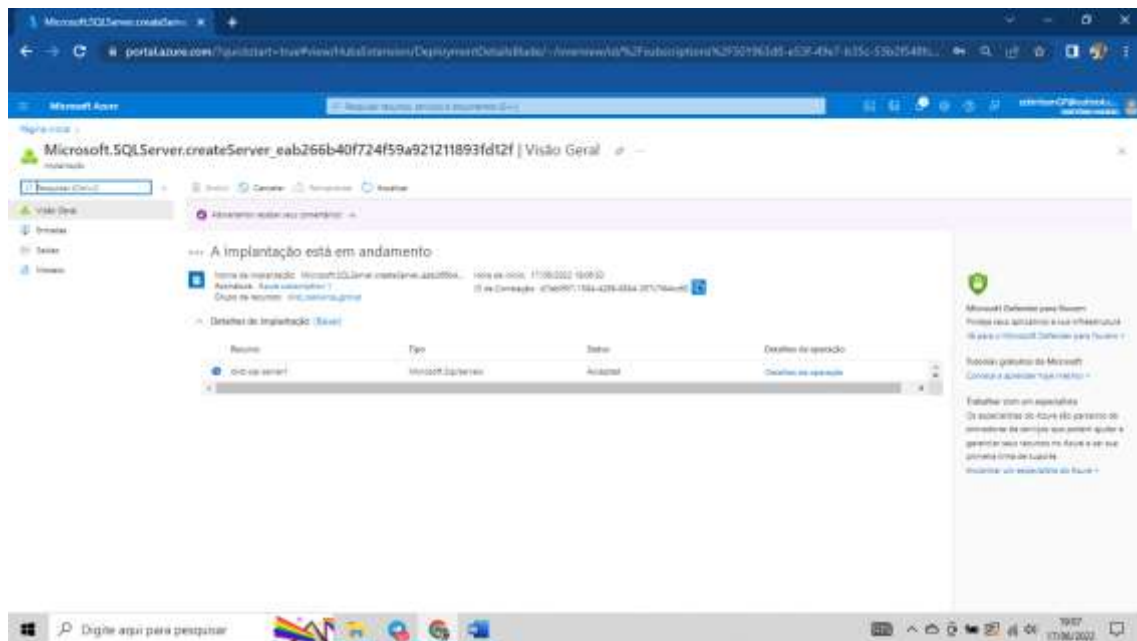
AQUI SÓ MANTER E CLICAR EM AVANÇAR



CLICAR EM CRIAR

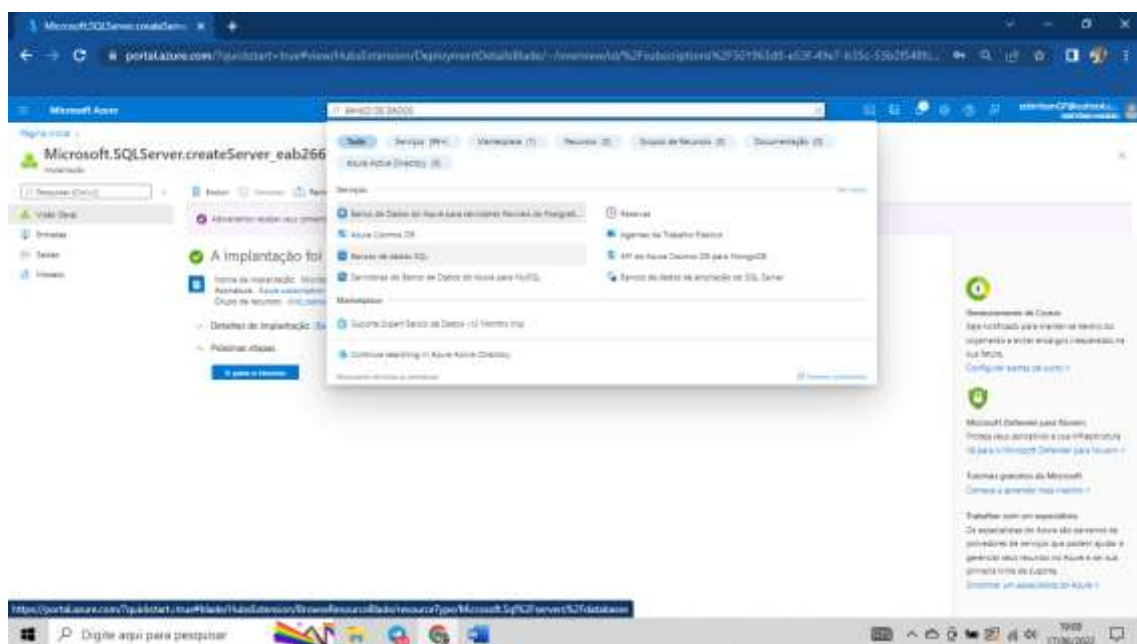


EM ANDAMENTO

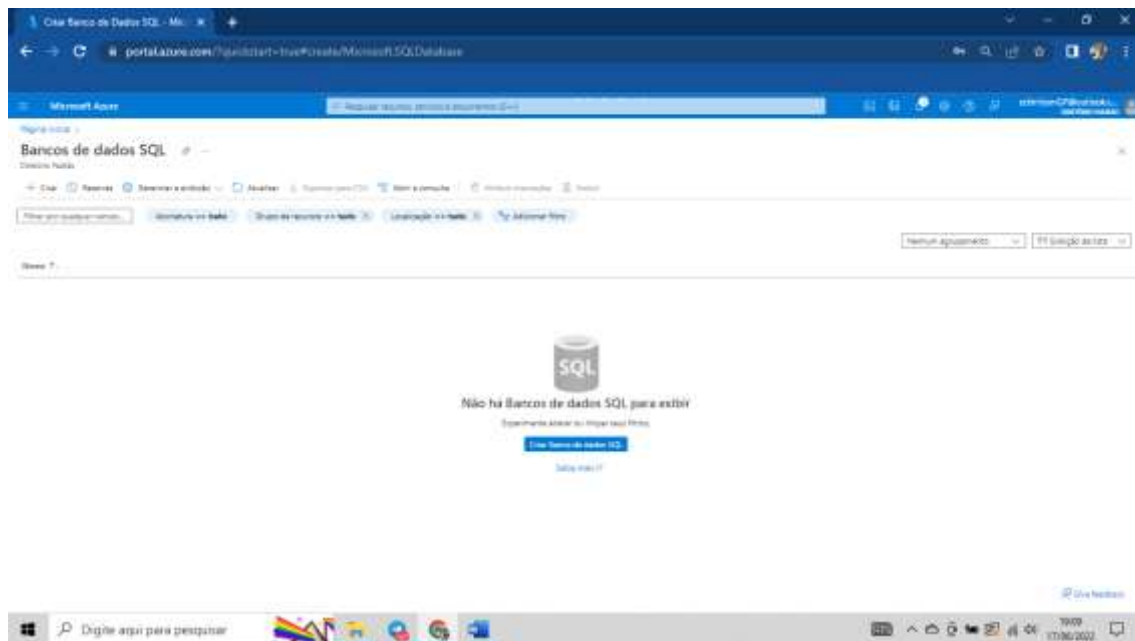


CONCLUIDA

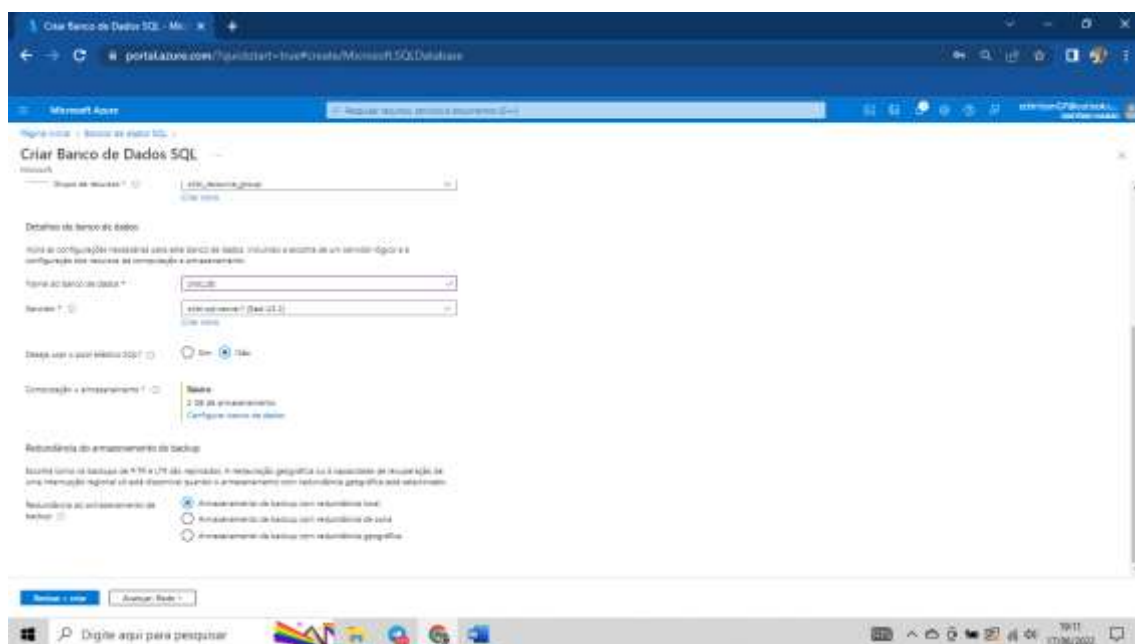
BANCO DE DADOS SQL



CRIAR

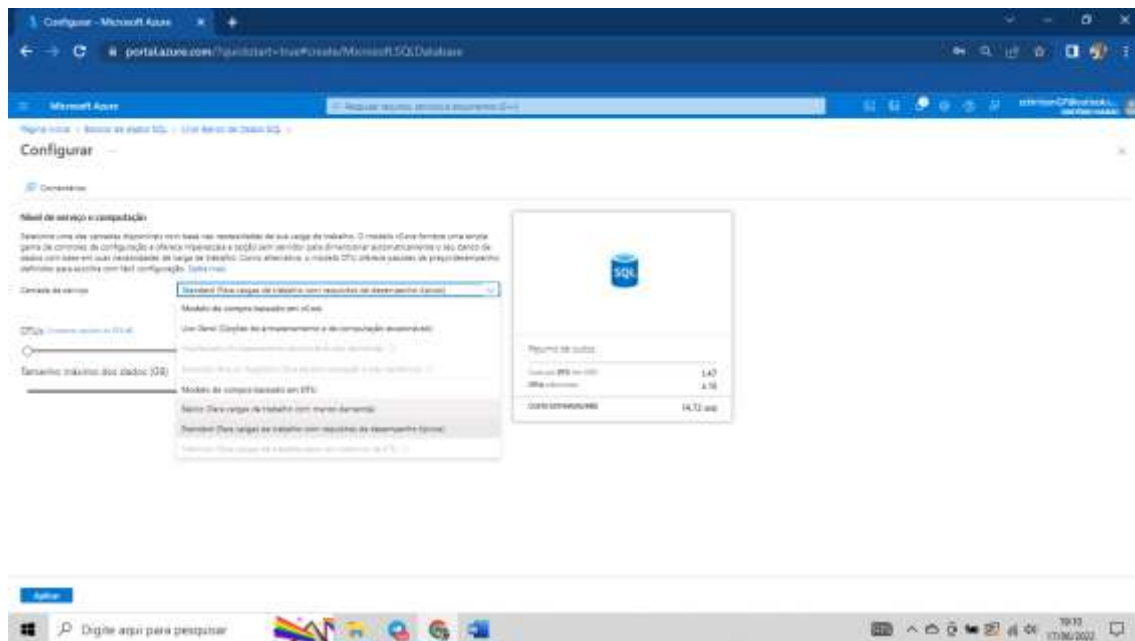


Nome do banco de dados: olist_db

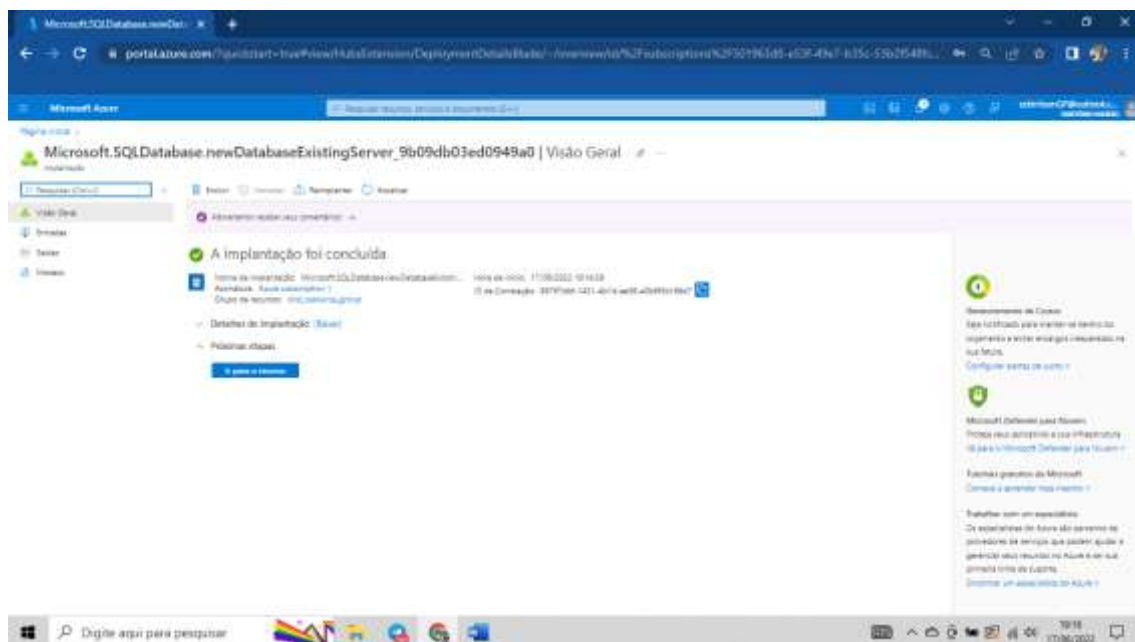


CONFIGURAÇÃO

AVANÇAR ATÉ O FINAL E CLICAR EM CRIAR



CRIADO O DATA BASE

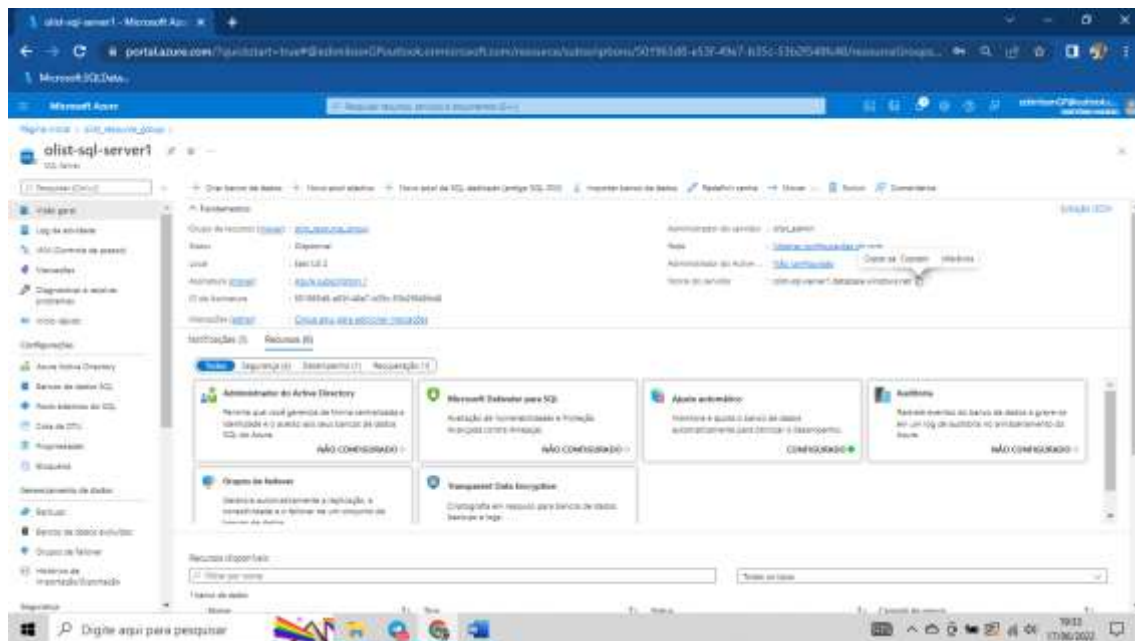


AGORA CARREGAR OS DADOS DENTRO DO DATA BASE

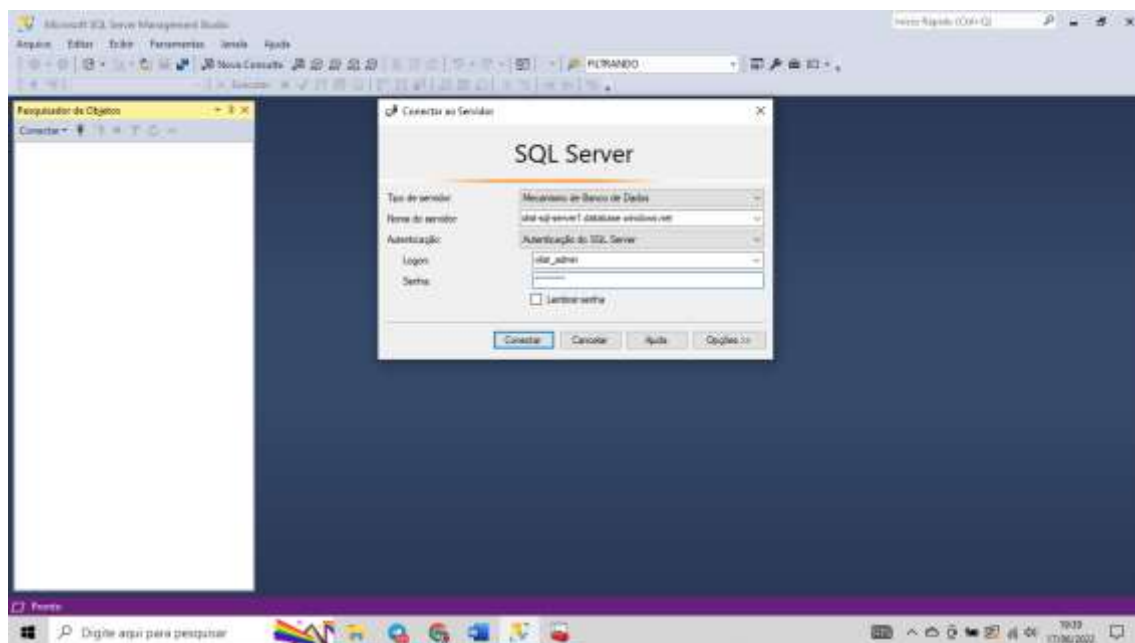
Link do dataset no kaggle: <https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce>

Copiar Nome do servidor: olist-sql-server1.database.windows.net

Para fazer a conexão

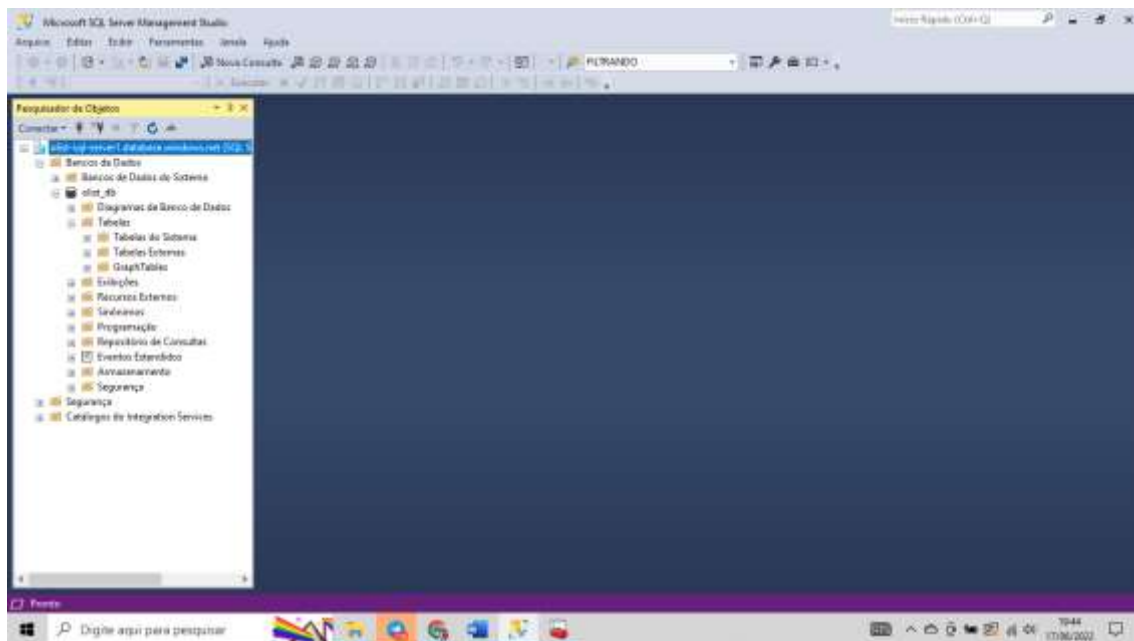


Inserindo os dados de conexão no SQL SERVER

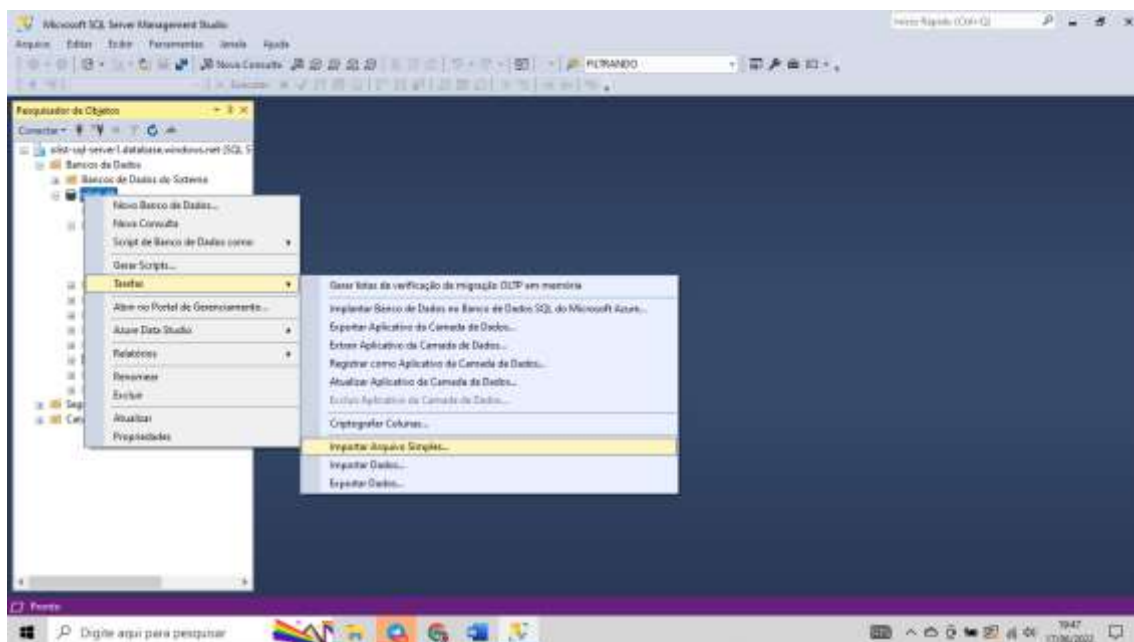


Obrigatoriamente tem que entrar com a conta da Azure

AGORA ESTÁ CONECTADO



IMPORTANDO AS TABELAS

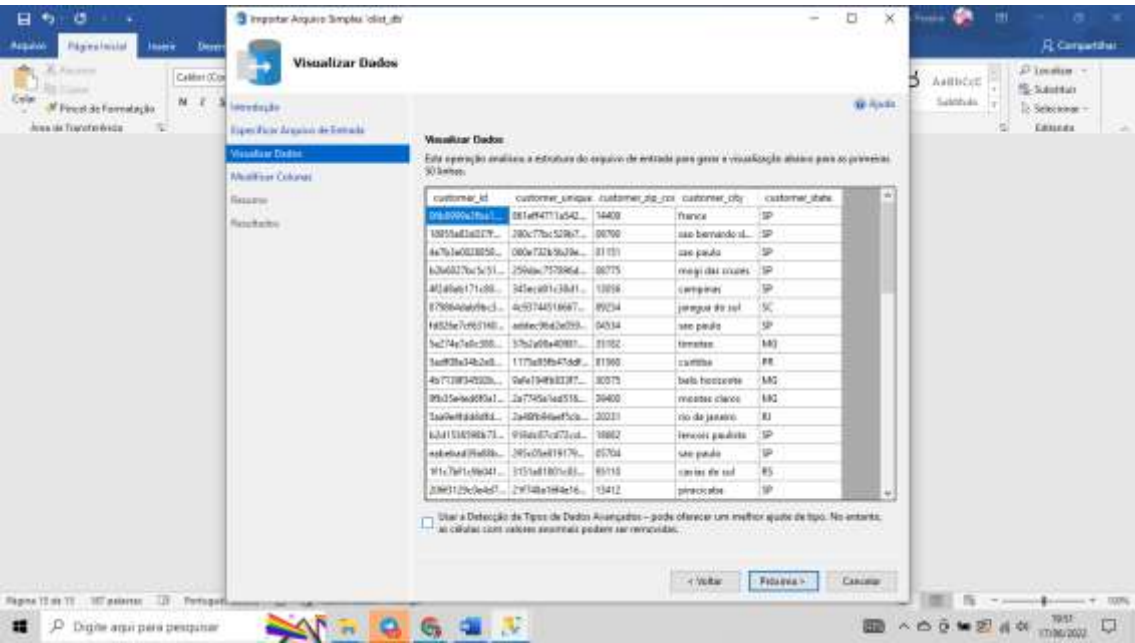


CLICAR EM AVANÇAR

SELECIONAR O ARQUIVO (TABELA)

REMOVER A SELEÇÃO DE DETECÇÃO DE TIPOS DE DADOS PARA NÃO TER ERRO NA IMPORTAÇÃO

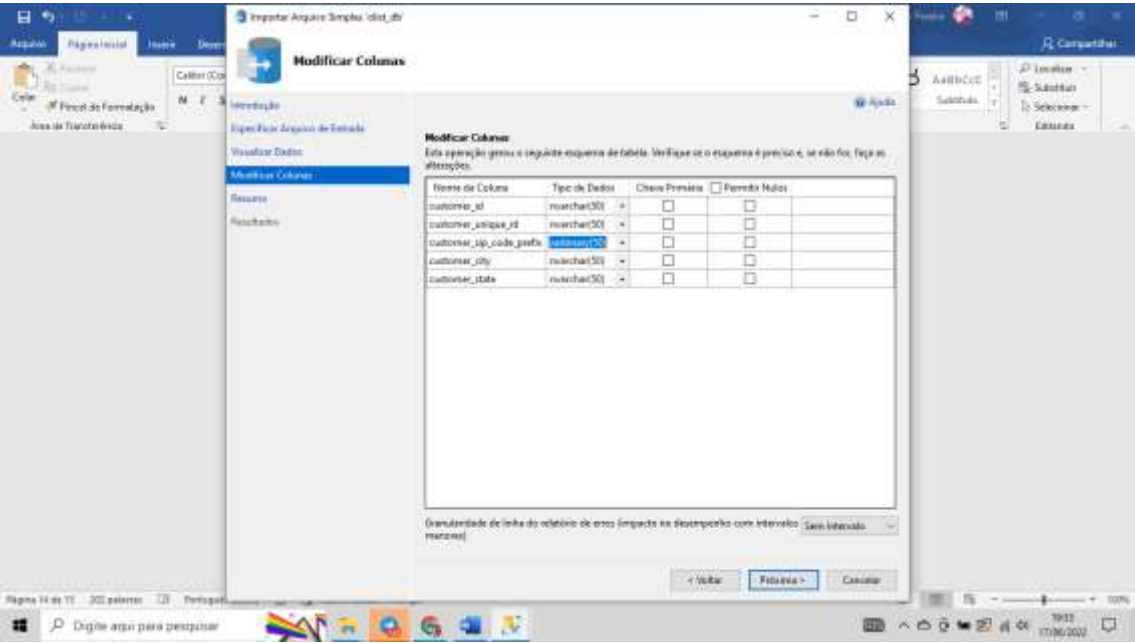
CLICA EM PROXIMO



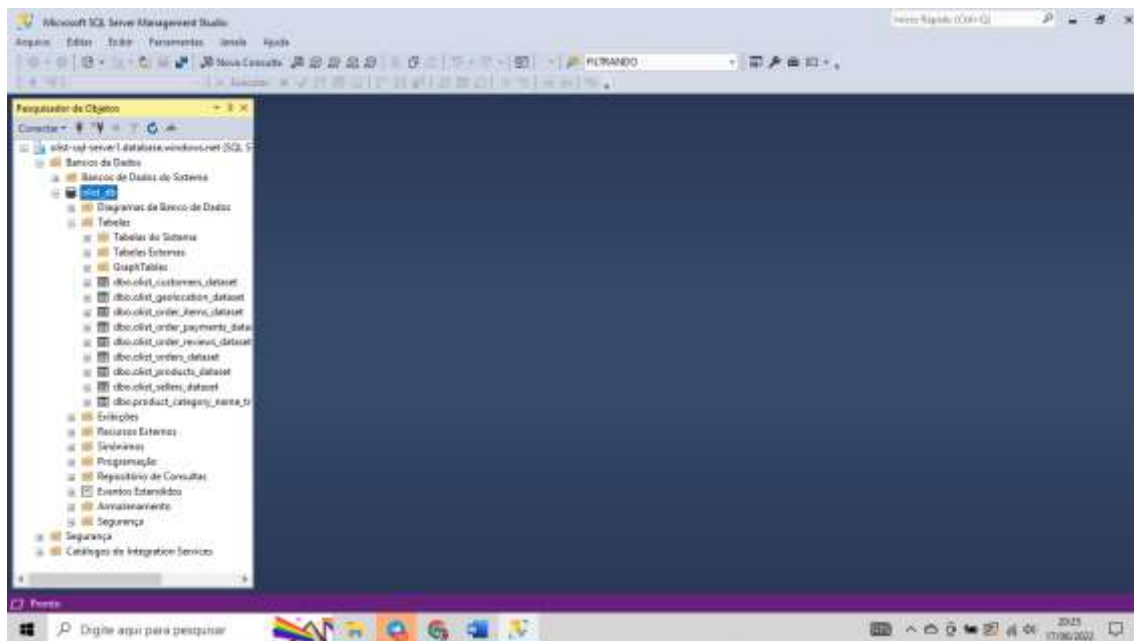
SELECIONAR TODOS OS CAMPOS COMO NVARCHAR (50)

CLICAR EM PROXIMO E FAZER ESSE MESMO PROCESSO PARA TODAS AS TABELAS

EM CASO DE ERRO NO CARREGAMENTO SELECIONAR OS NULOS



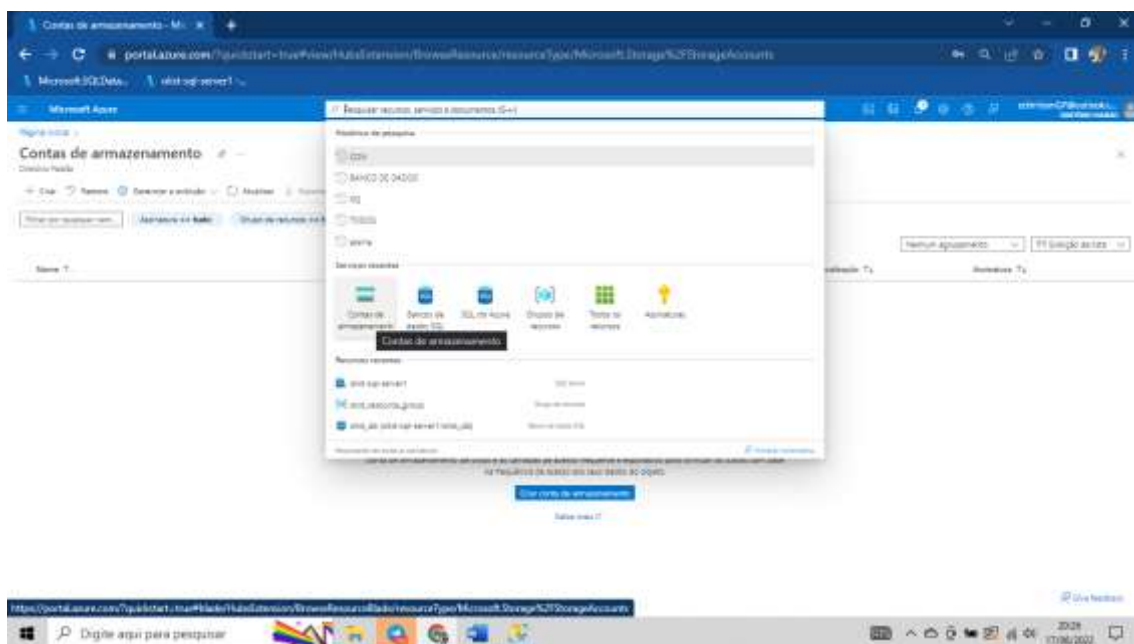
TODAS AS TABELAS CARREGADAS



CRIAR O DATA LAKE

PESQUISAR POR CONTAS DE ARMAZENAMENTO

E CLICAR EM CRIAR



Nome da conta de armazenamento
oliststorageaccount2

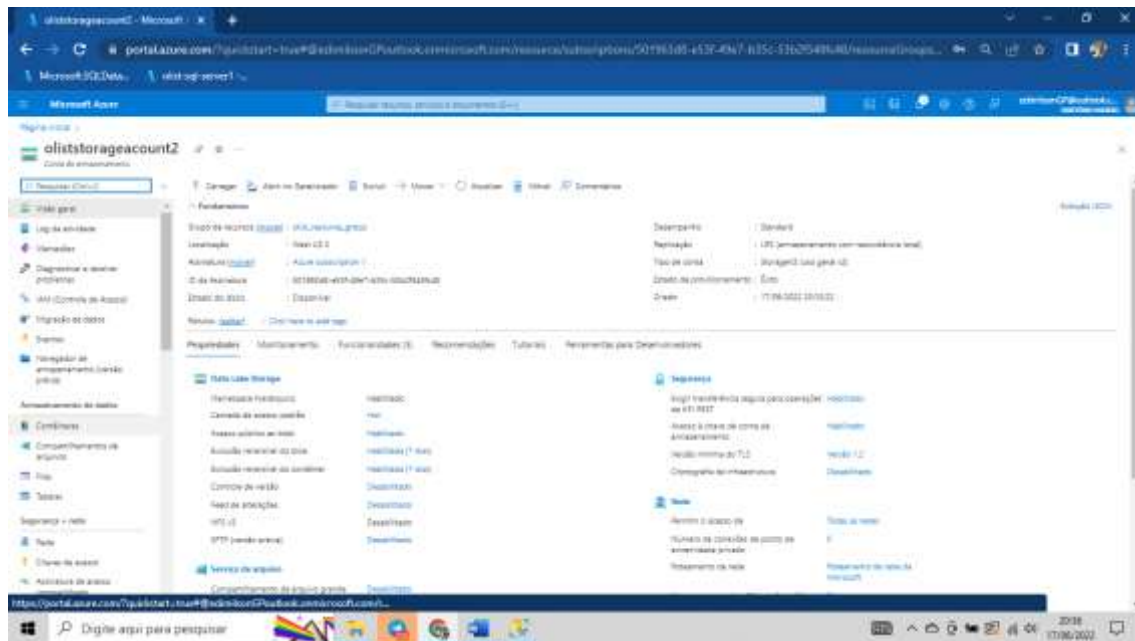
clicar em avançar

selecionar Data Lake Storage Gen2

Agora em oliststorageaccount2

Clicar em Containeres

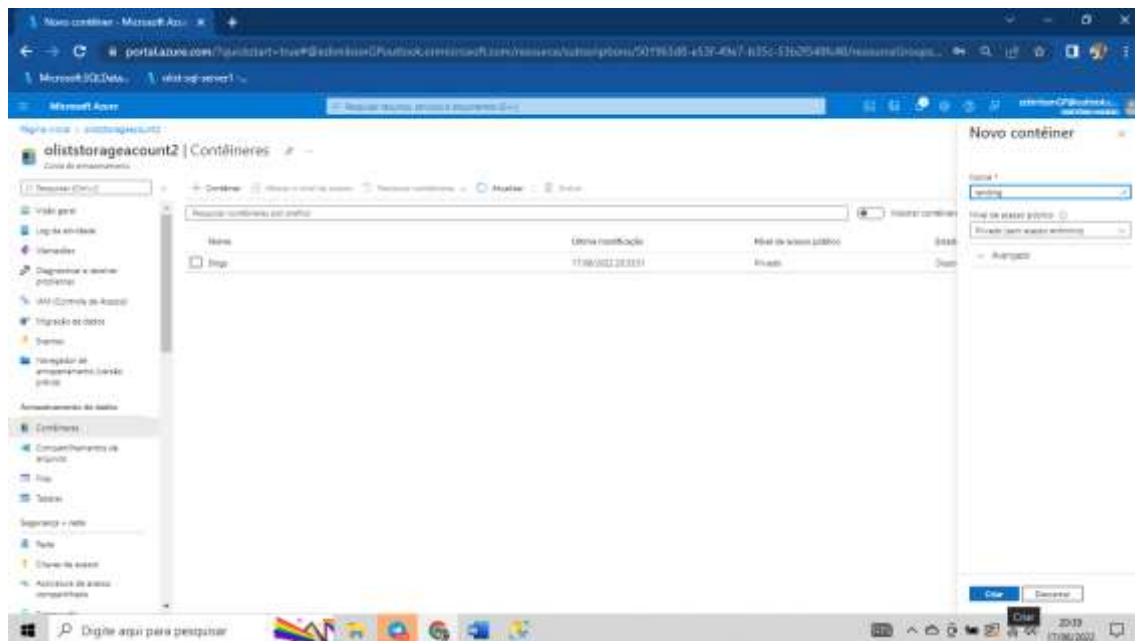
Para criar Landing, Processing, Acureta



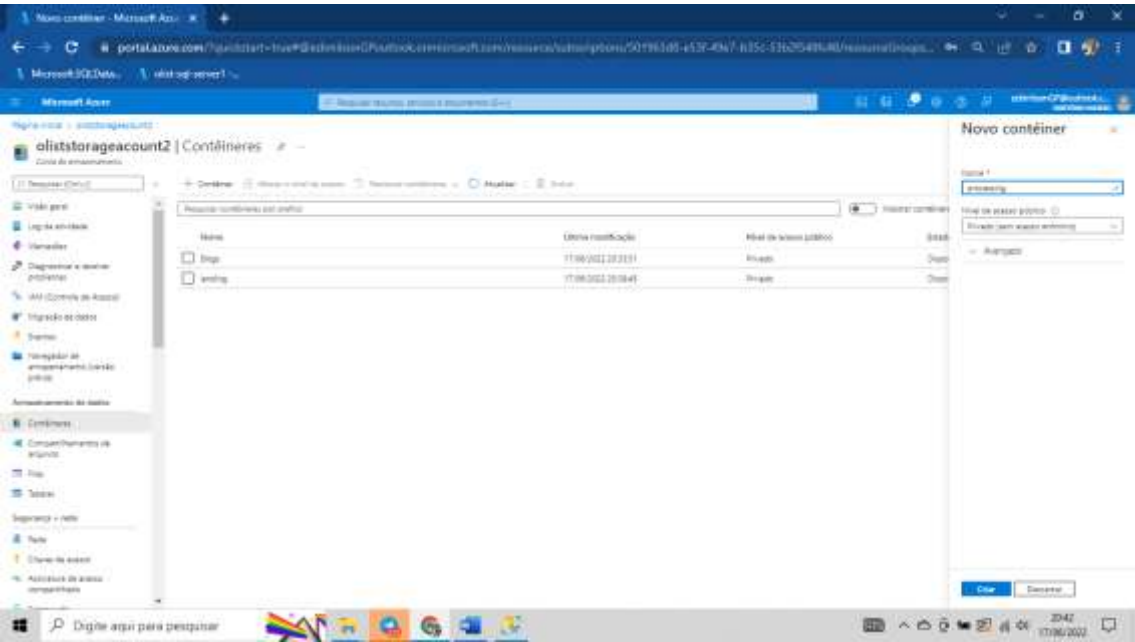
Clicar em +Container e colocar o nome nesse caso é a

LANDING

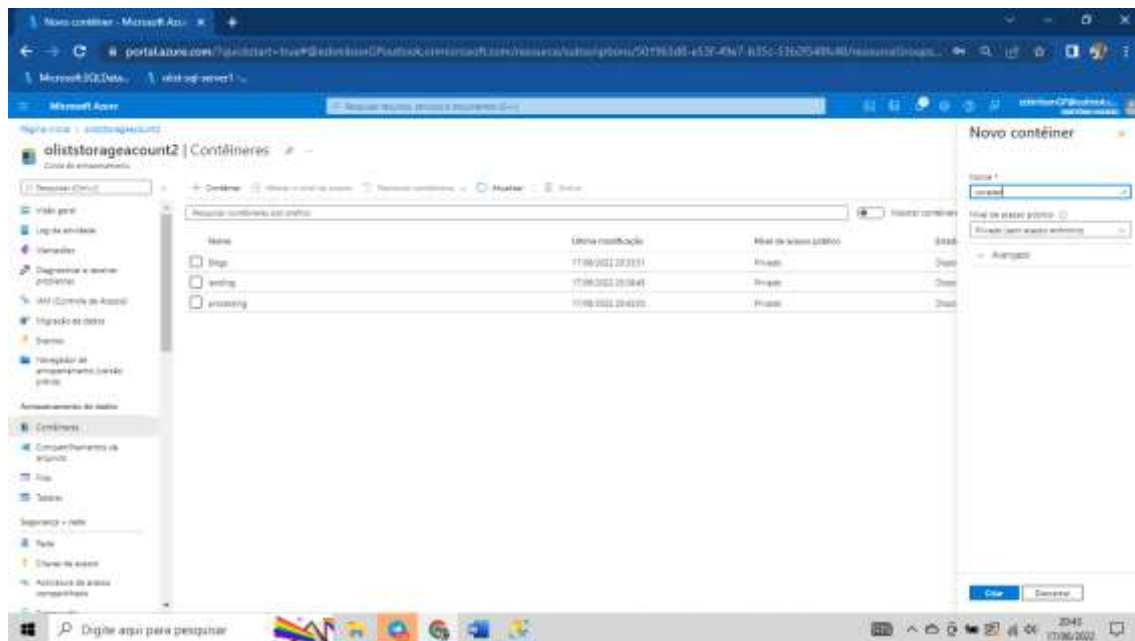
E criar



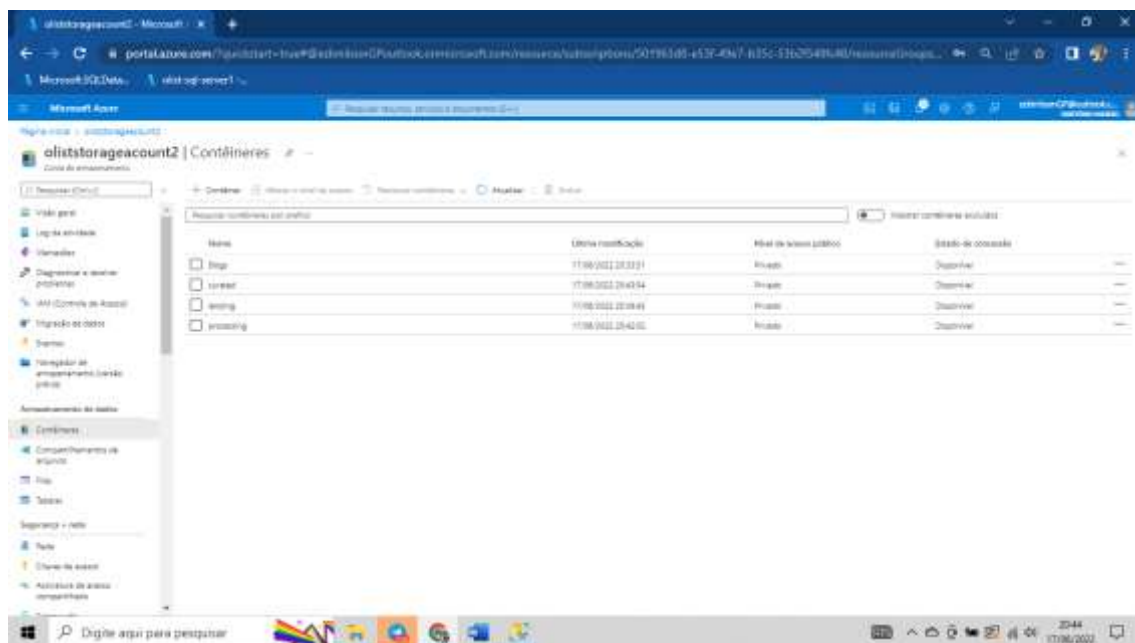
PROCESSING



CURATED

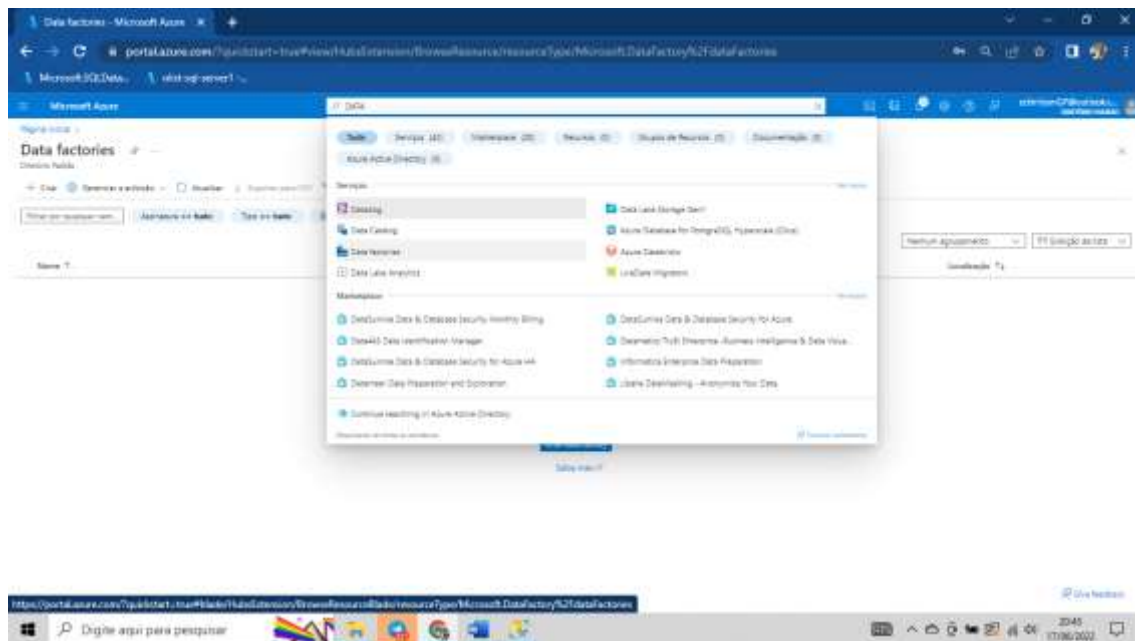


CONTÊINERES CRIADOS

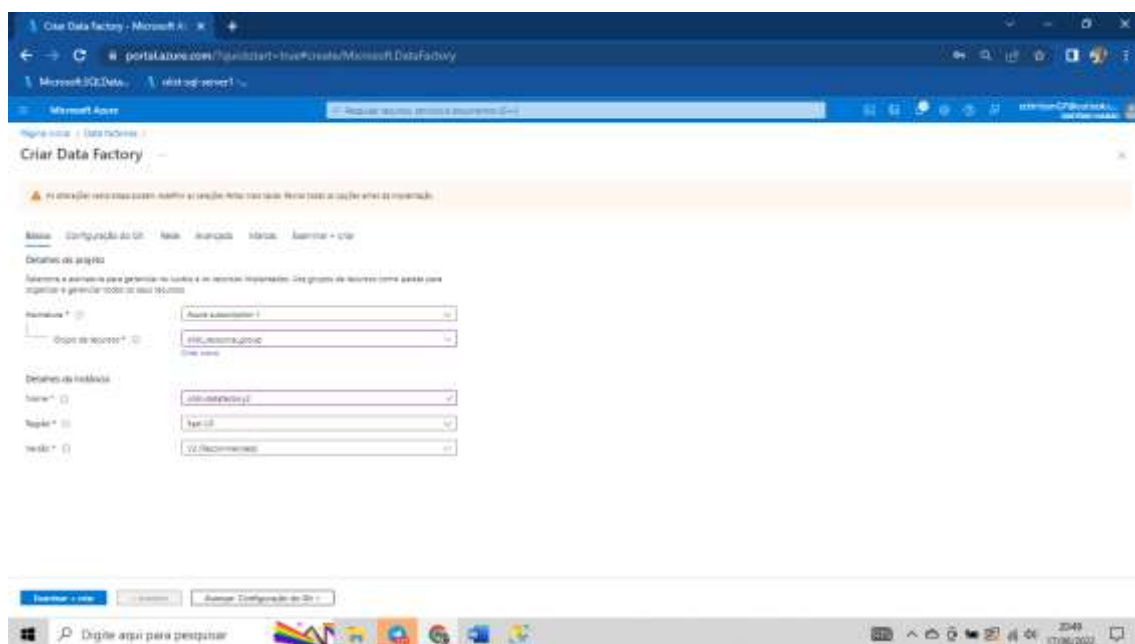


DATAFACTORY

É UMA FERRAMENTA DE ETL ELE AJUDA A TRANSFERIR OS DADOS DE UM LUGAR PARA OUTRO DE DIVERSAS FONTES DISPONÍVEIS

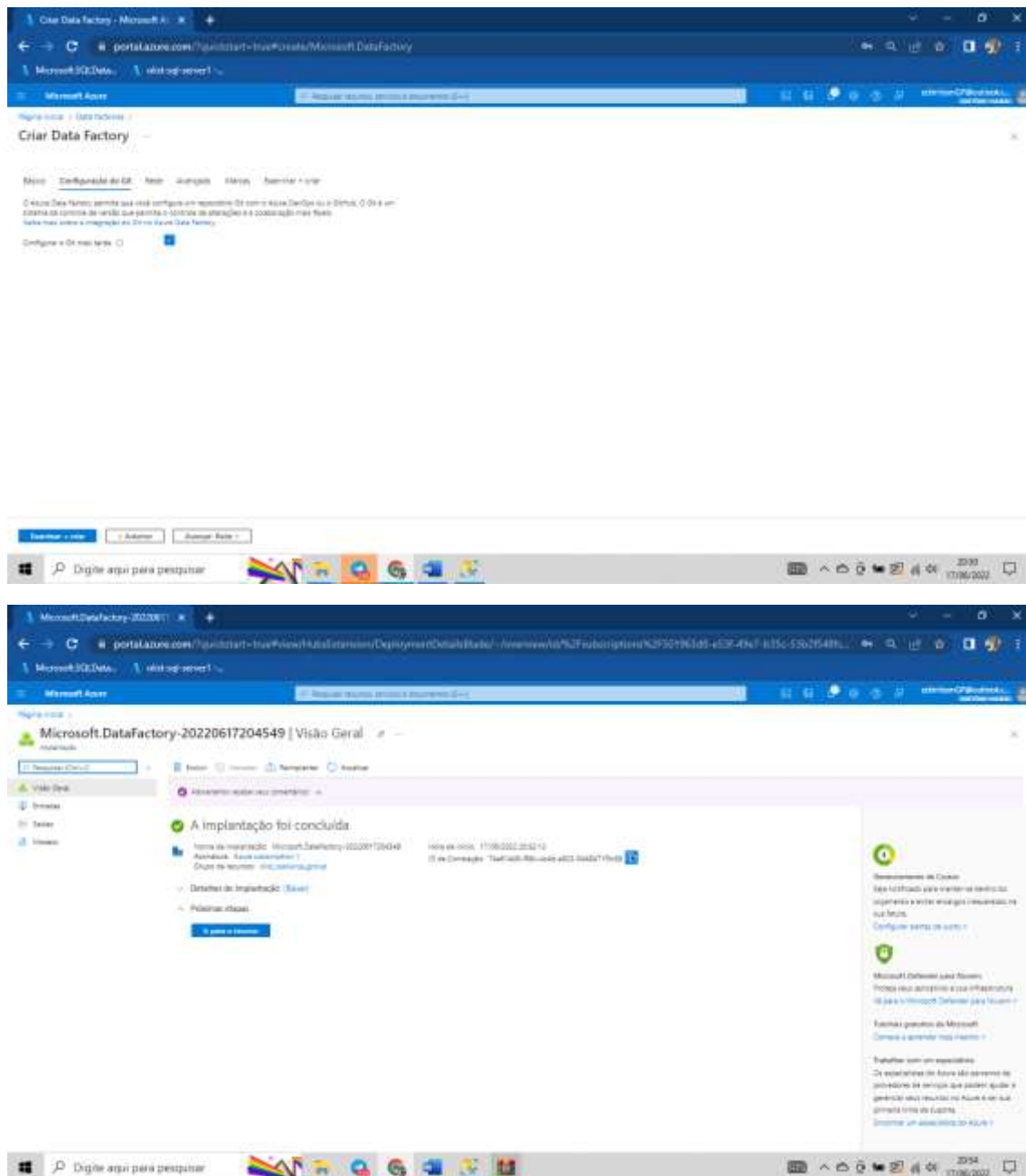


olist-datafactory2



Avançar e selecionar configurar o git mais tarde

E criar



ABRIR O DATA FACTORY – para fazer o ETL

E CLICAR EM ABRIR O ESTUDIO DO AZURE DATA FACTORY

Clicar em NEW PIPELINE

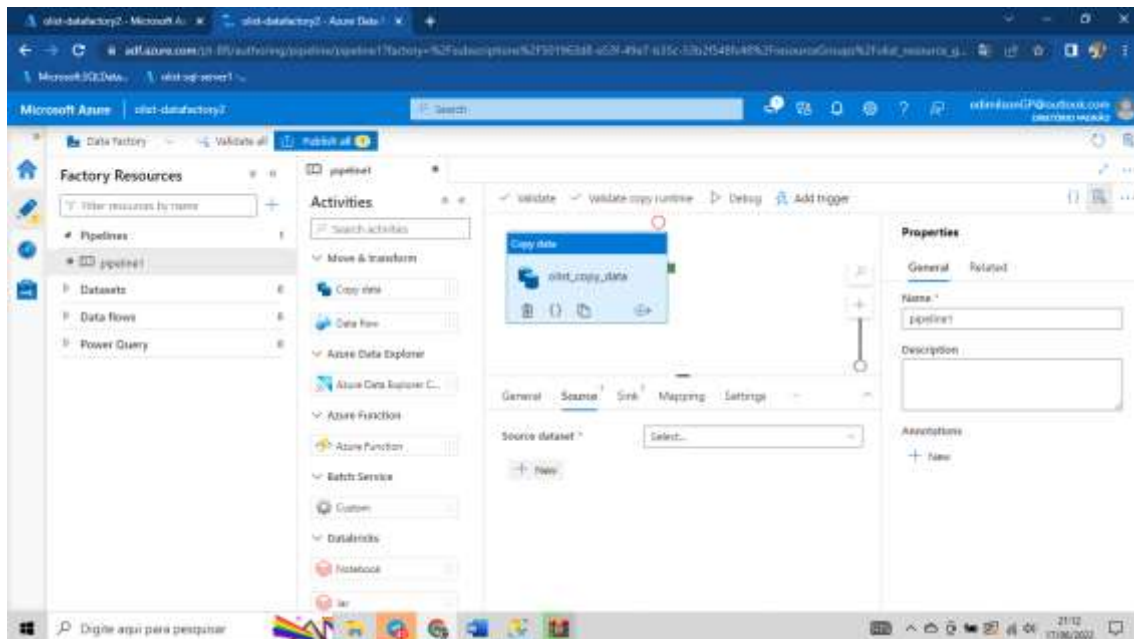
COPIANDO OS DADOS

Clicar e arrastar o ícone para a área e configurar

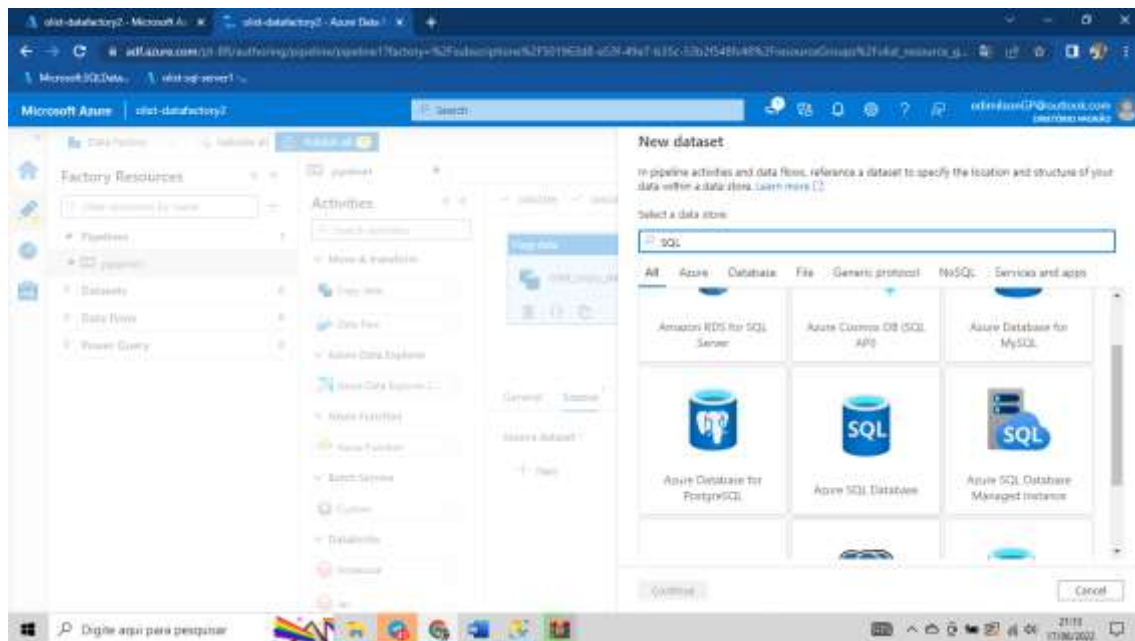
olist_copy_data

SELECIONAR A ORIGEM DOS DADOS EM SOURCE

NEW

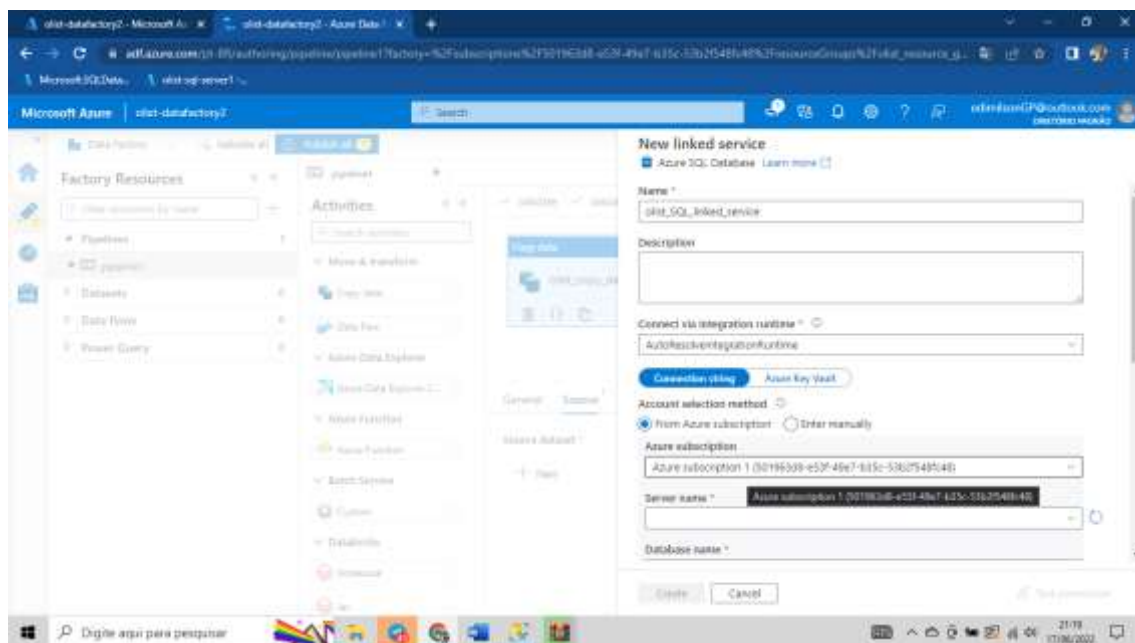


Azure SQL

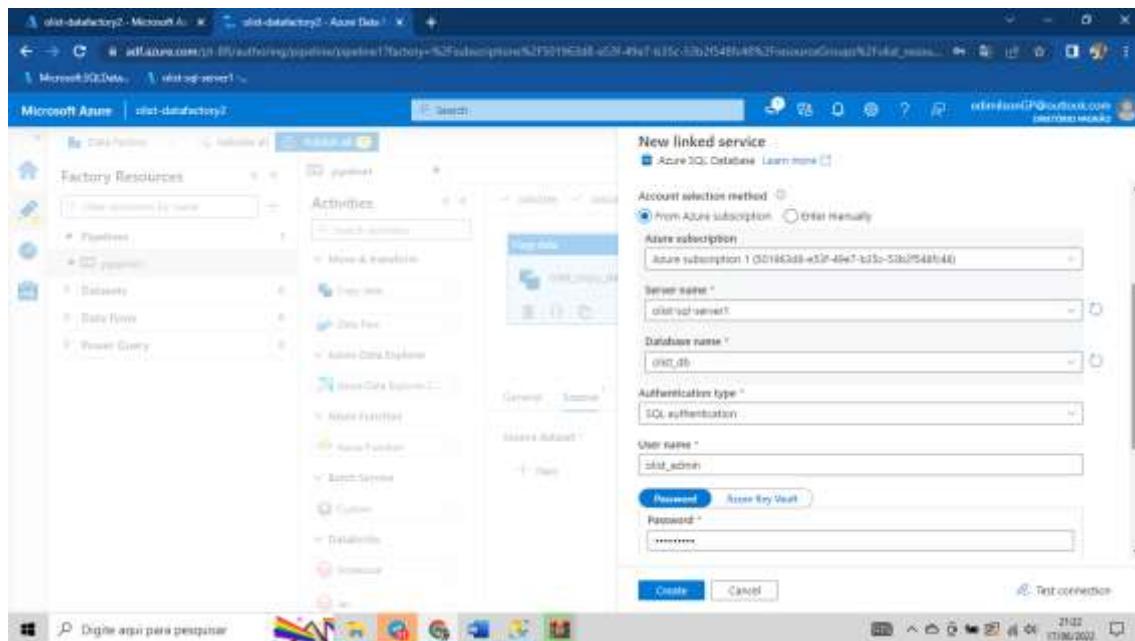


olist_SQL_linked_service1

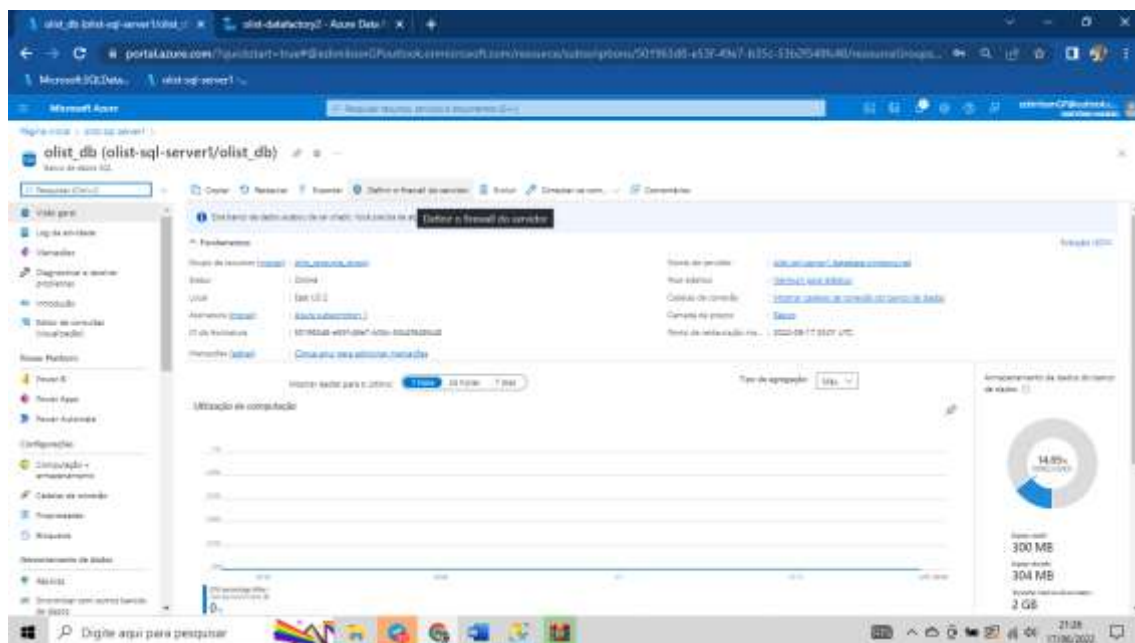
CONEXÃO ASSINATURA DO AZURE



Clicar em testar a conexão



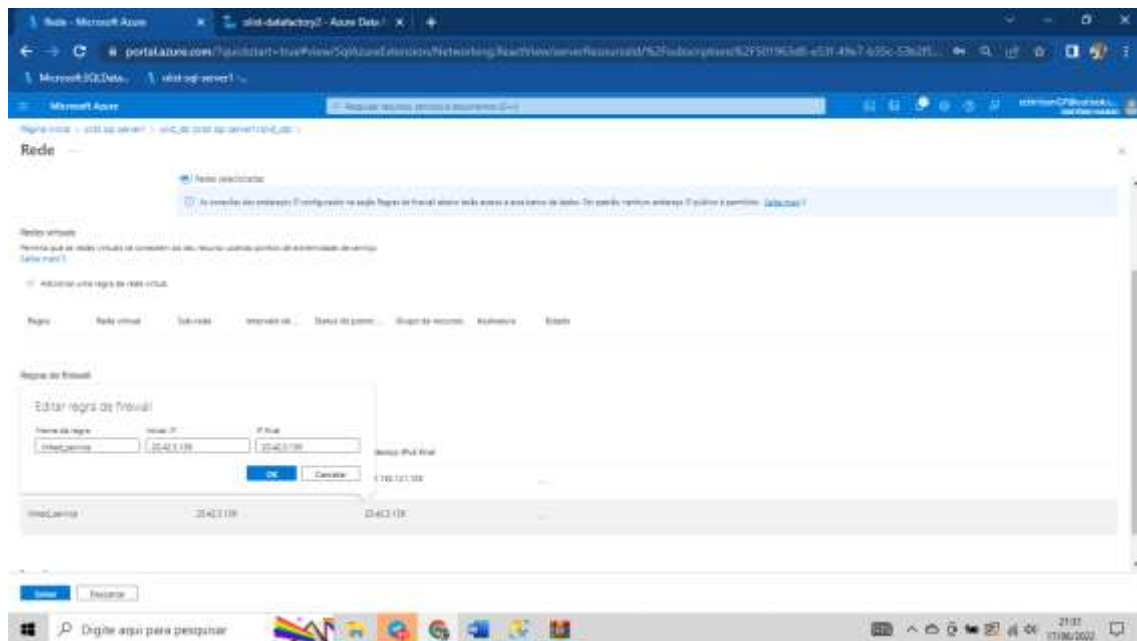
ACESSANDO PERMISSÃO E REGRA DE FIREWALL



ADICIONAR REGRA

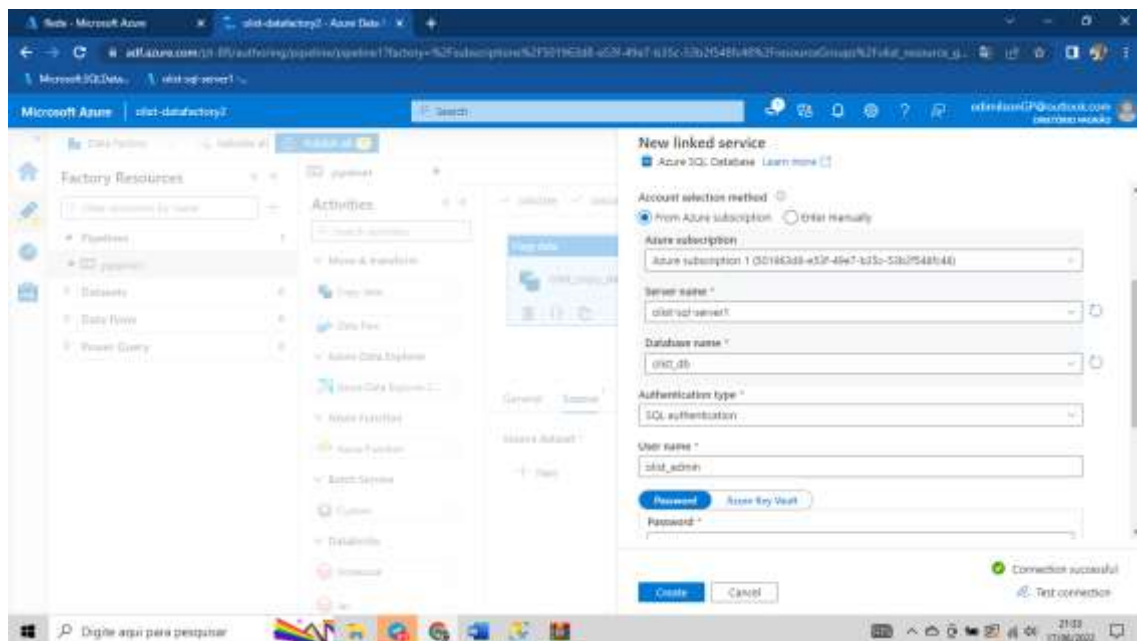
Colocando o IP que aparece lá no erro do teste de conexão '20.42.3.136'

E clicar em salvar

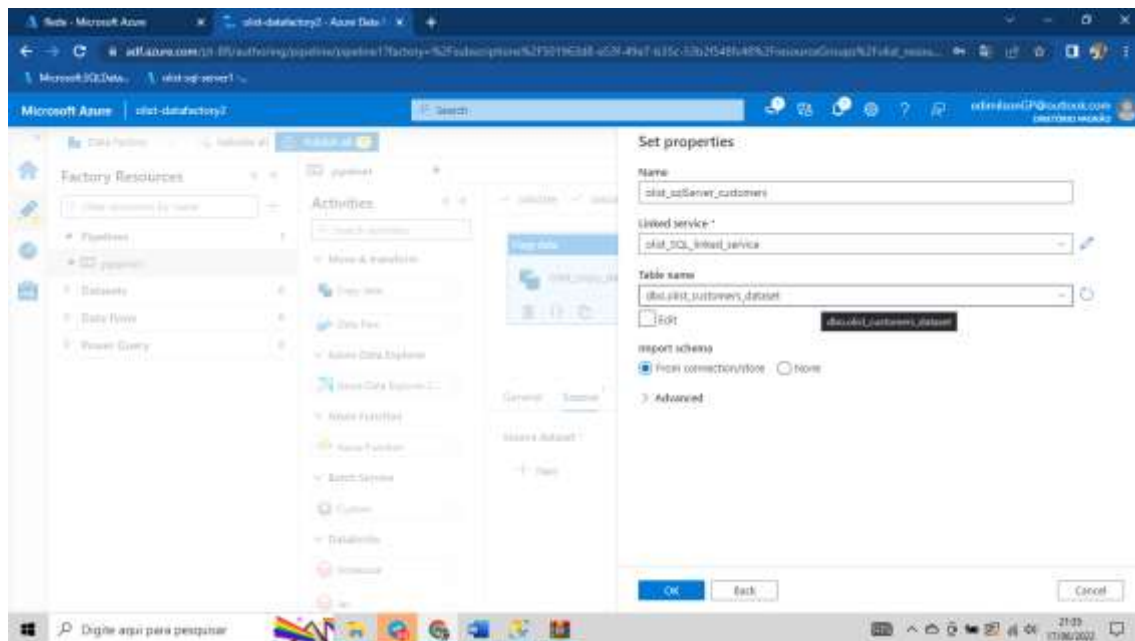


Testando novamente a conexão

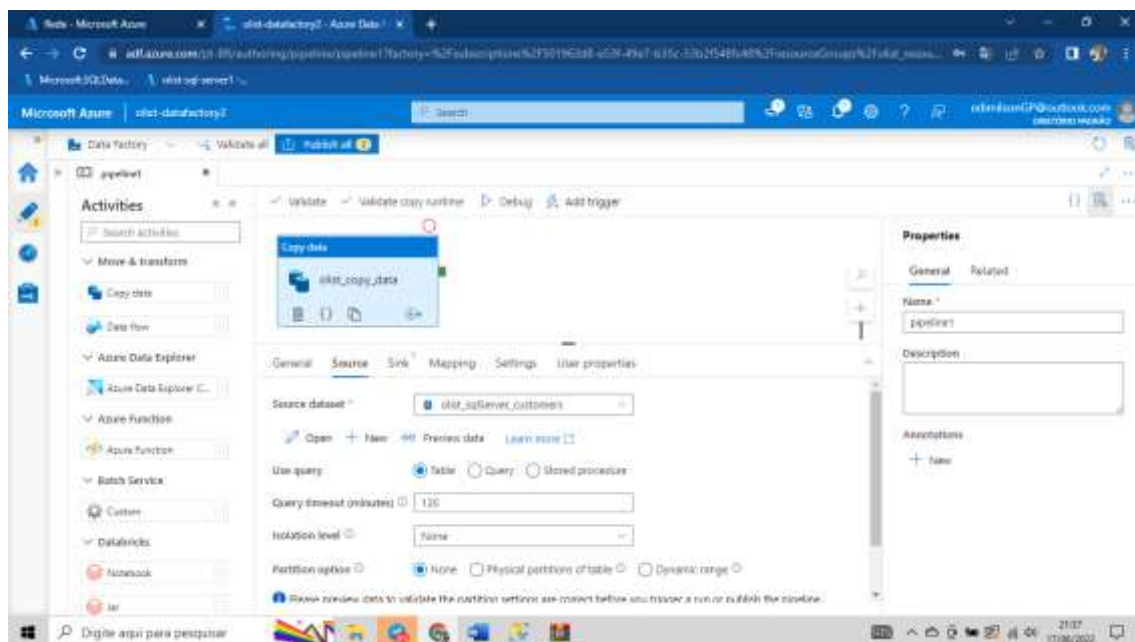
Conectado com sucesso e clicar em Criar



Agora eu escolho a tabela que eu quero



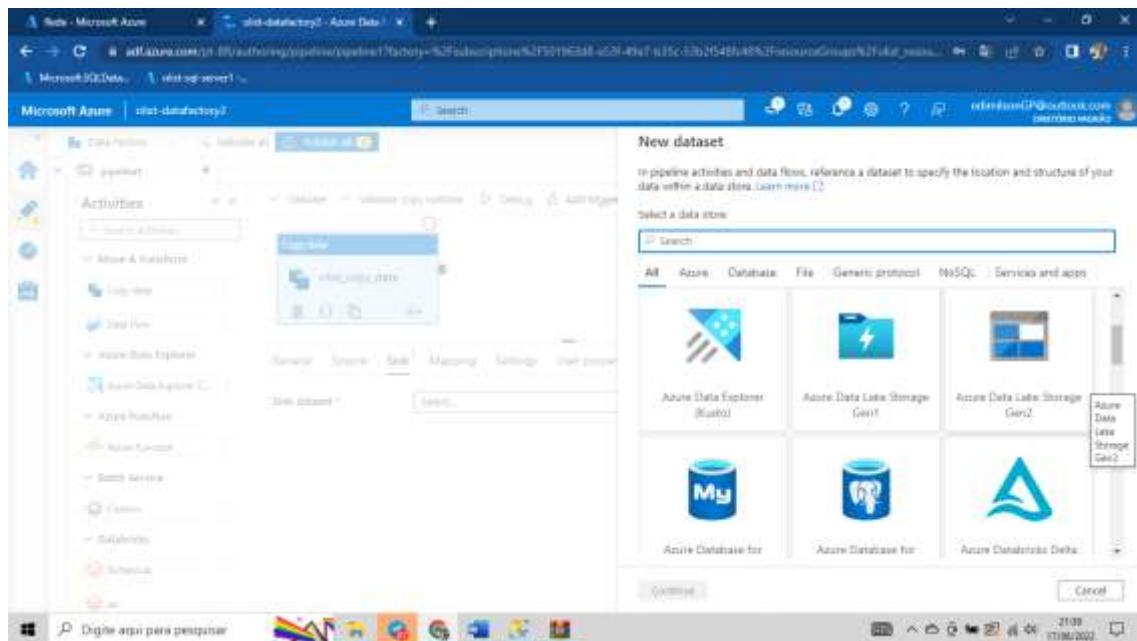
Origem criada



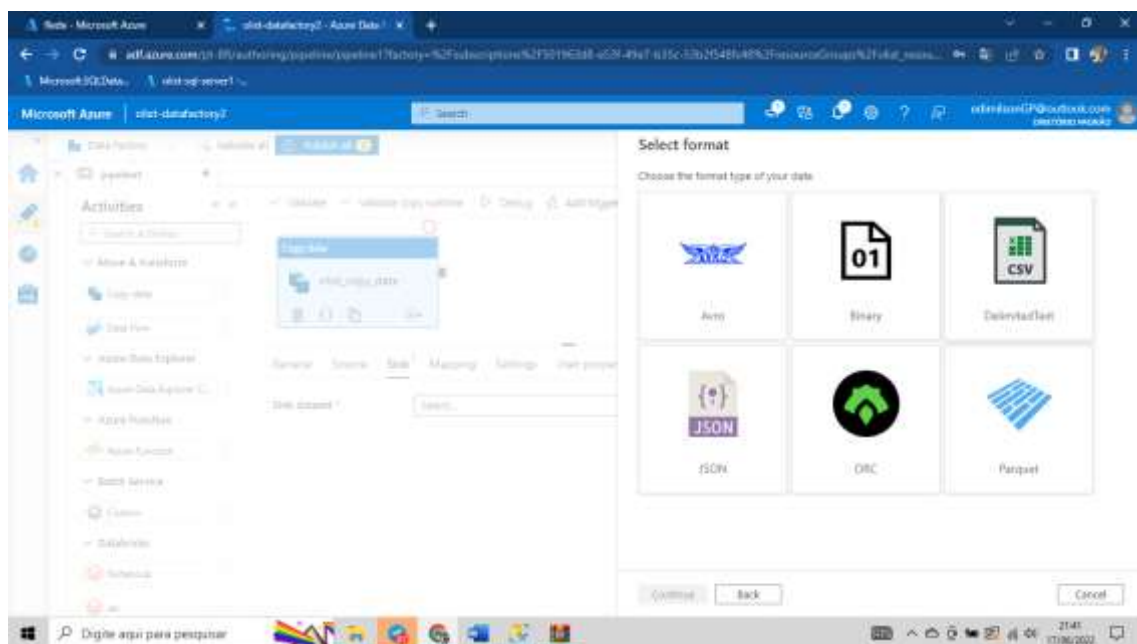
Agora para onde vou enviar os dados clicar em SINK

Clicar em NEW e

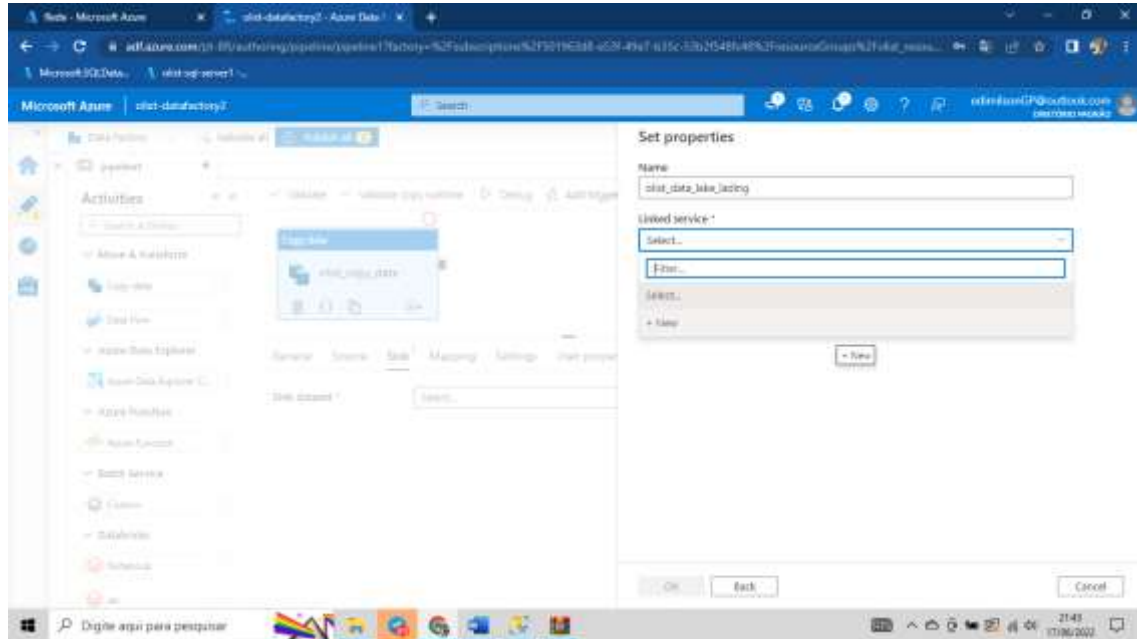
Pesquisar pelo Azure Data Lake Storage Gen2



Selecionar o tipo de arquivo que estamos trabalhando no caso as tabelas estão no formato CSV



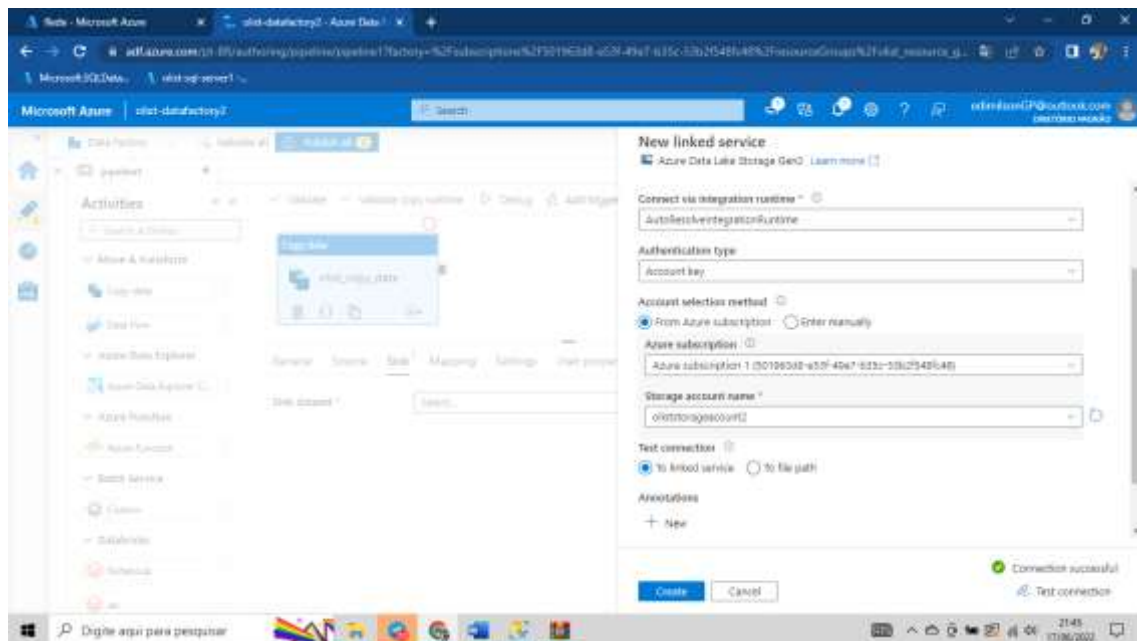
olist_data_lake_landing1

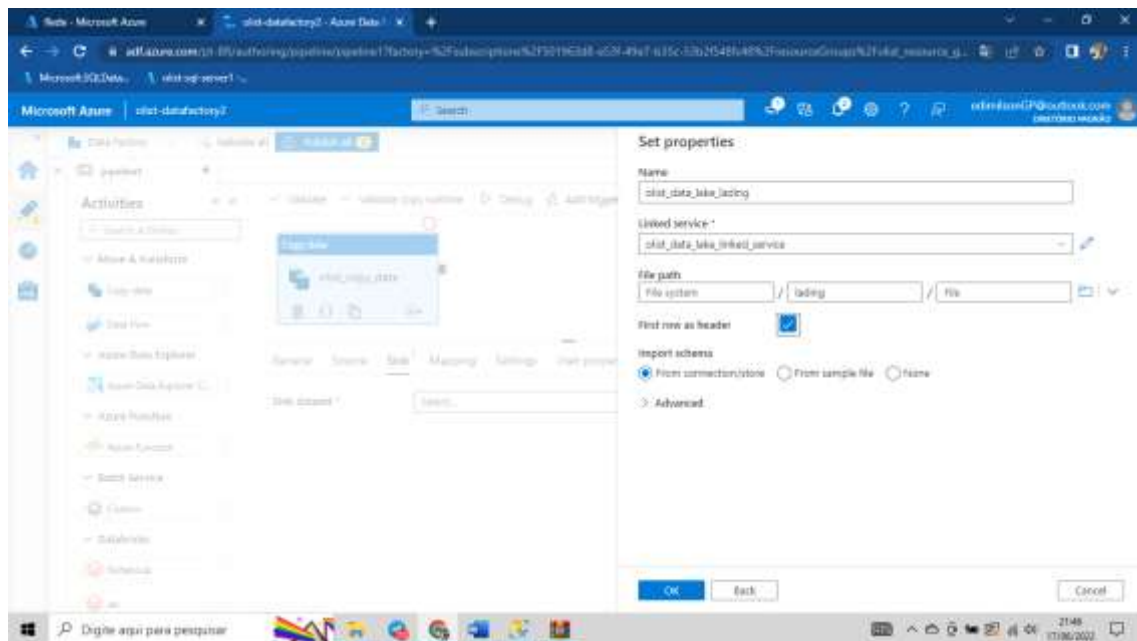


olist_data_lake_linked_service1

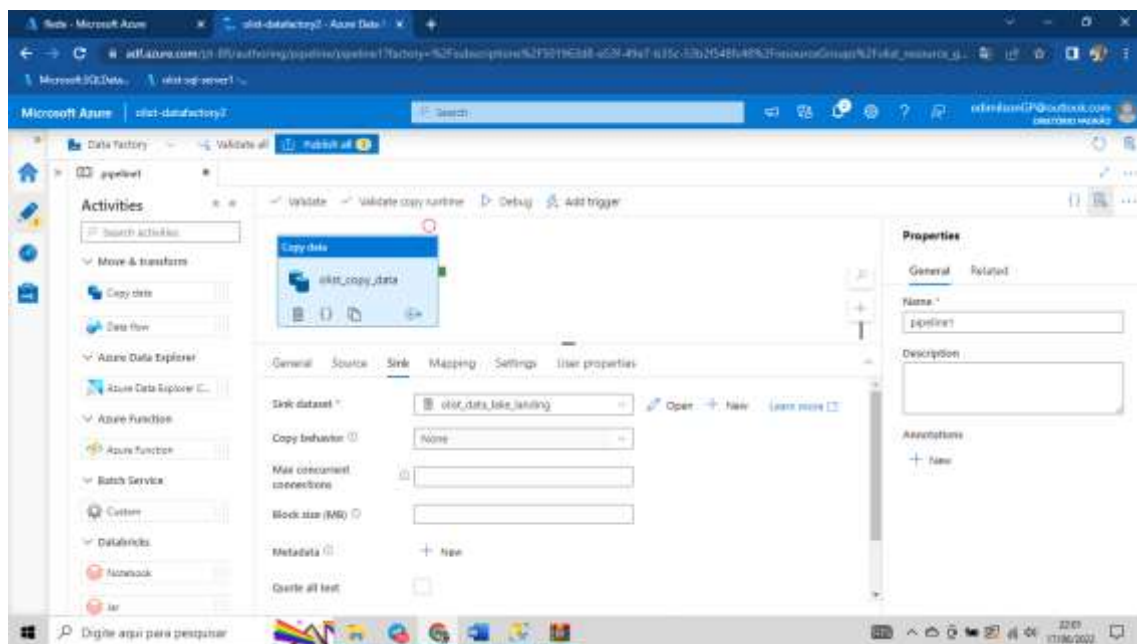
selecionando as informações

testando a conexão e criar

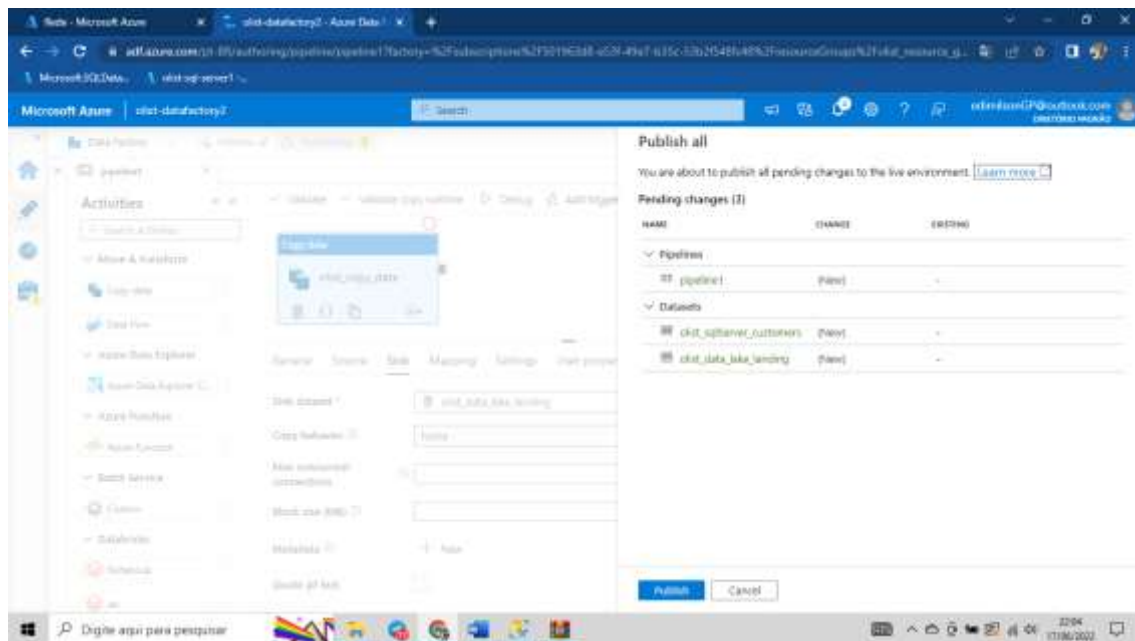




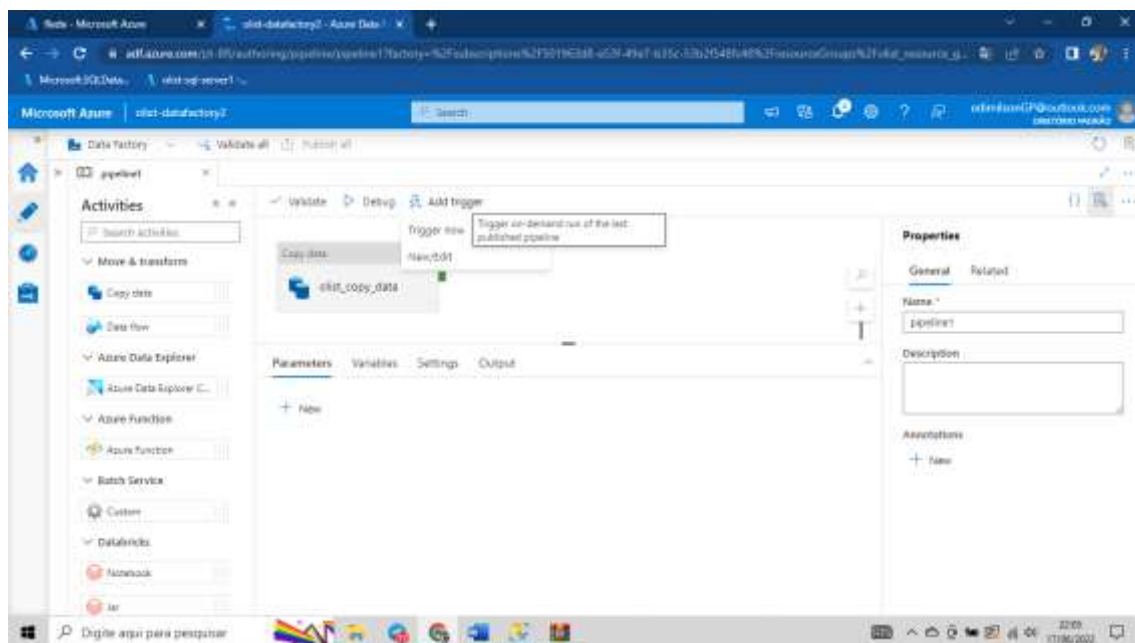
CONFIGURAÇÃO DE DESTINO EFETUADA



PARA TESTARMOS O PIPELINE TEMOS QUE PUBLICAR TODAS AS ATUALIZAÇÕES

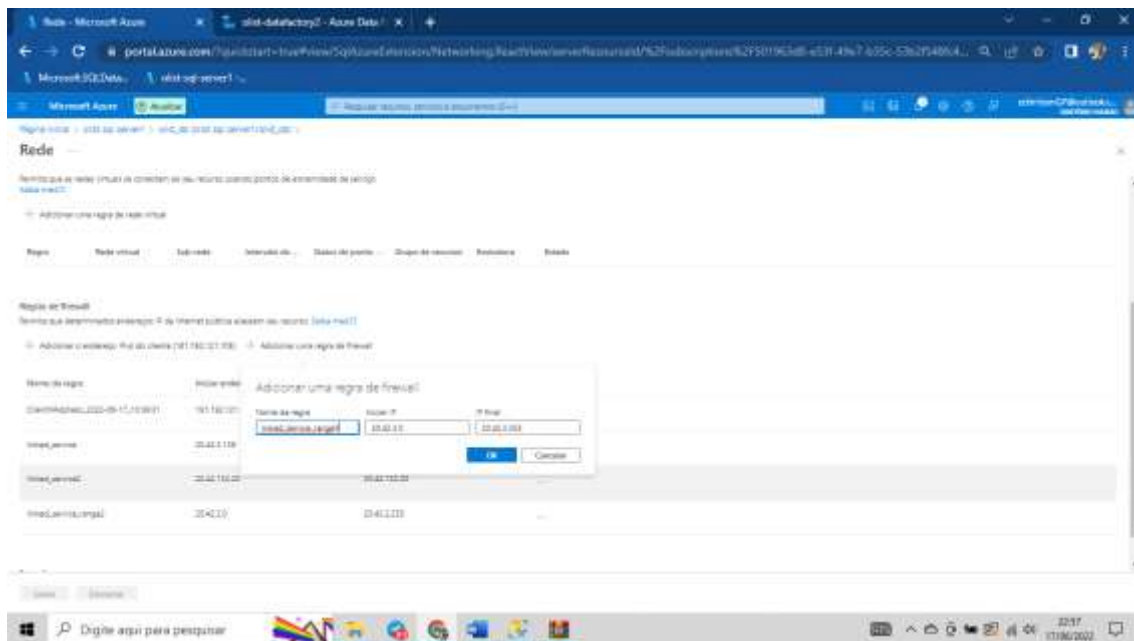


PARA EXECUTAR VAMOS ADICIONAR O GATILHO ADD TRIGGER NOW



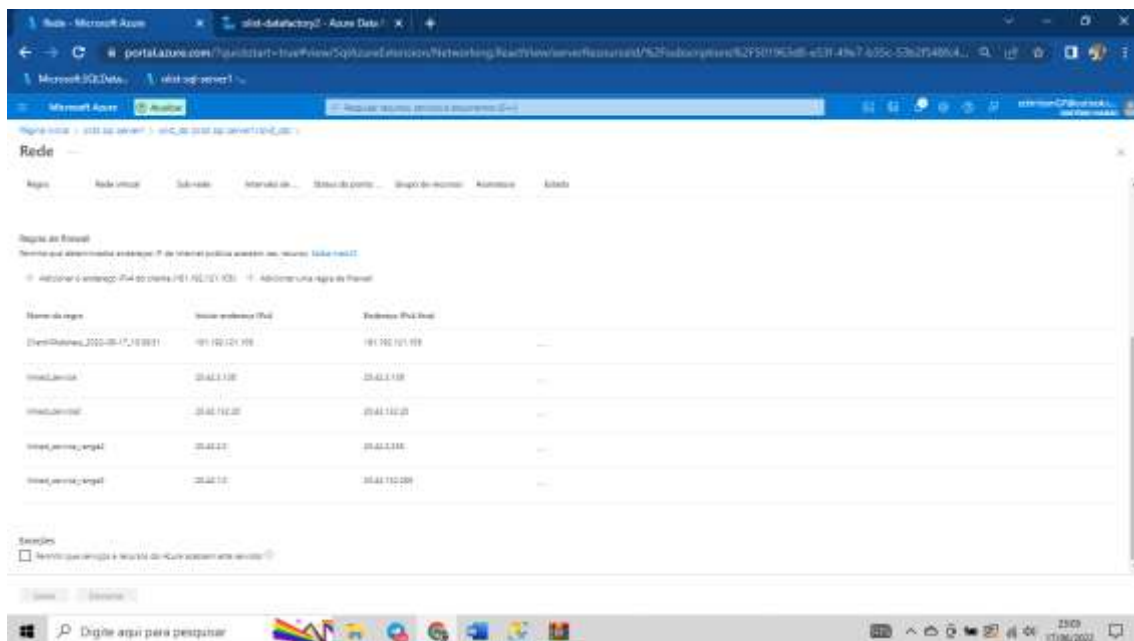
CONTINUA APRESENTANDO ERRO DE IP'S

ENTÃO TEREMOS QUE LIBERAR POR FAIXA



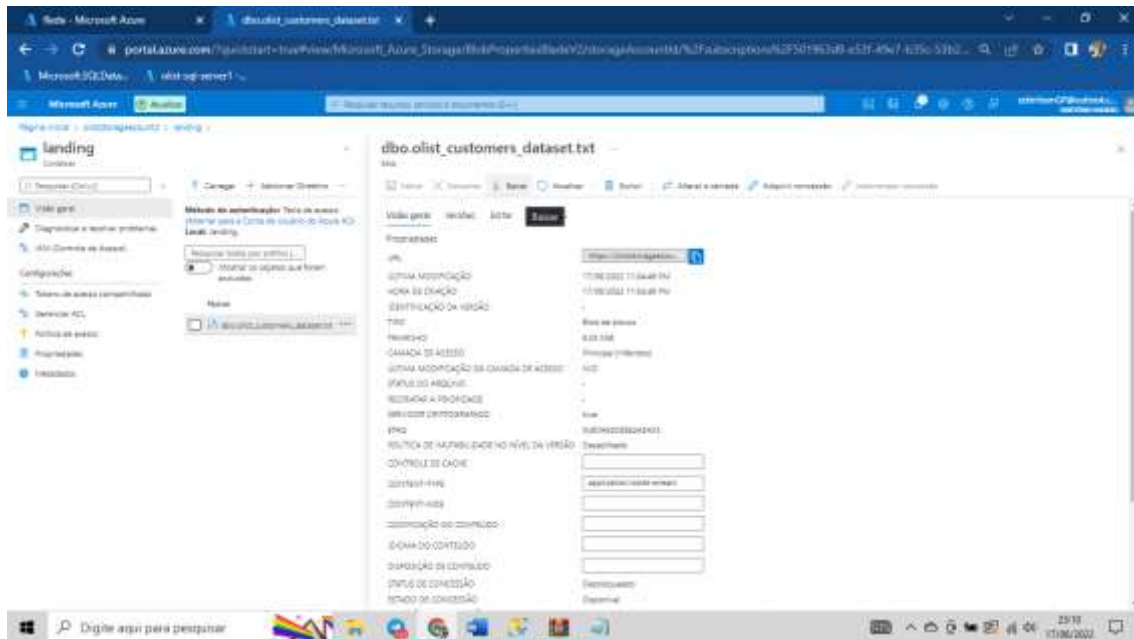
CONEXÃO FUNCIONANDO

COM AS FAIXAS INFORMADAS A BAIXO

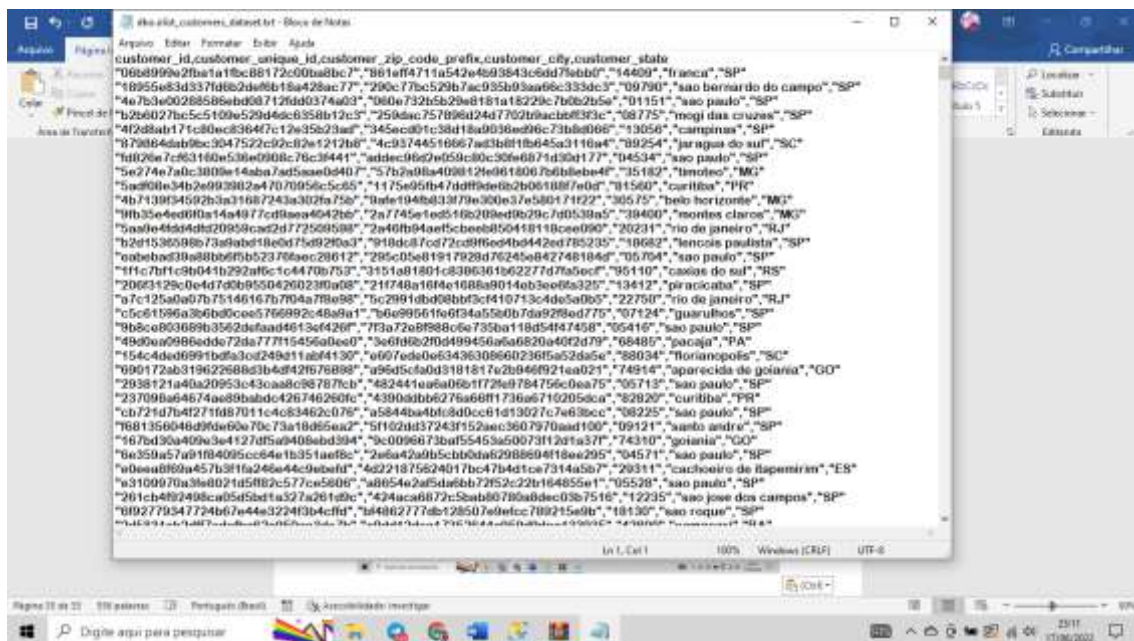


PARA VERIFICAR SE OCORREU TUDO CERTO COM O ETL

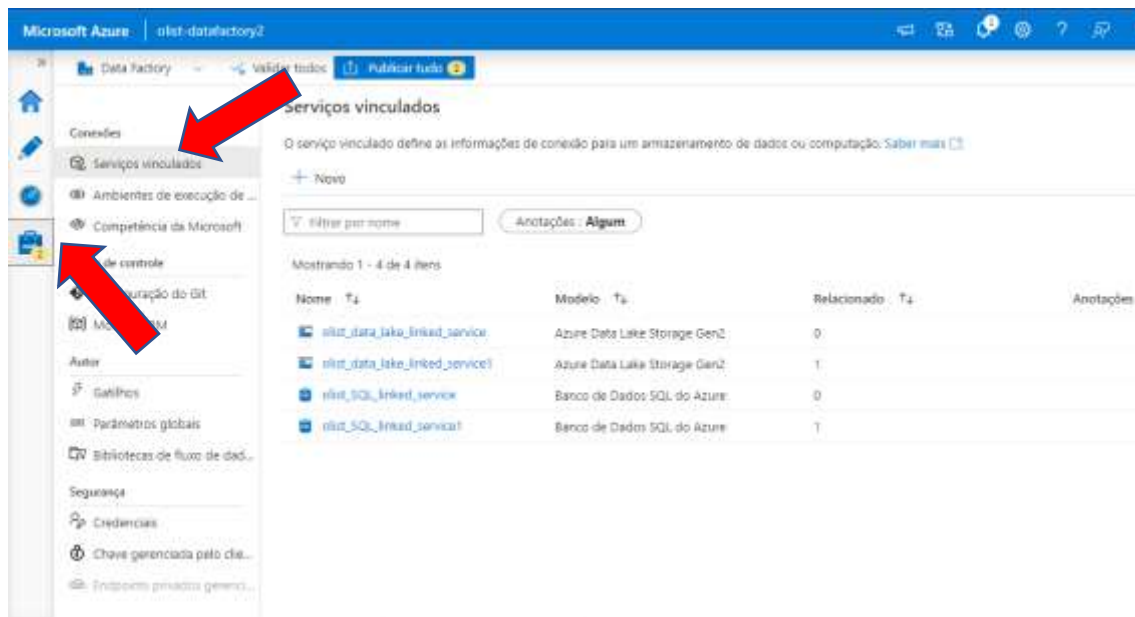
IR PARA PAGINA INICIAL EM OLISTSTORAGEACCOUNT2 EM CONTAINERES E CLICAR NA LANDING E BAIXAR UMA COPIA



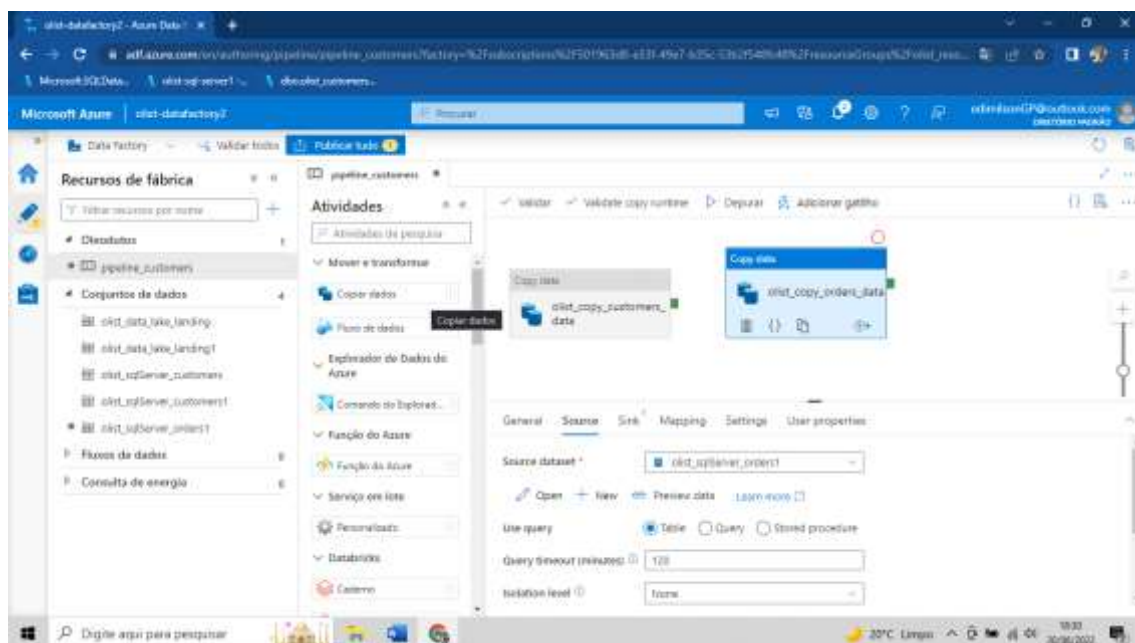
**ABRINDO O ARQUIVO PARA CONFERIR
ENTÃO ESTÁ FUNCIONANDO**



**AGORA VAMOS CARREGAR TODAS AS TABELAS DE UMA ÚNICA VEZ PARA GANHO DE TEMPO
CLICAR EM GERENCIAR E EM SERVIÇOS VINCULADOS**

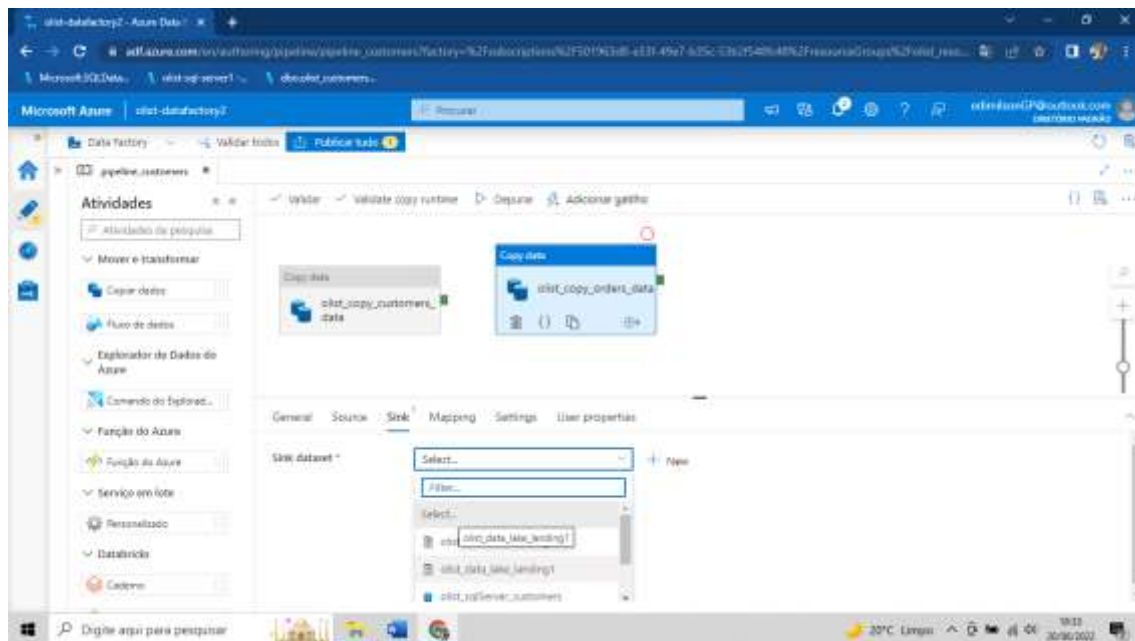


CRIAR UMA NOVA COPIA DE DADOS



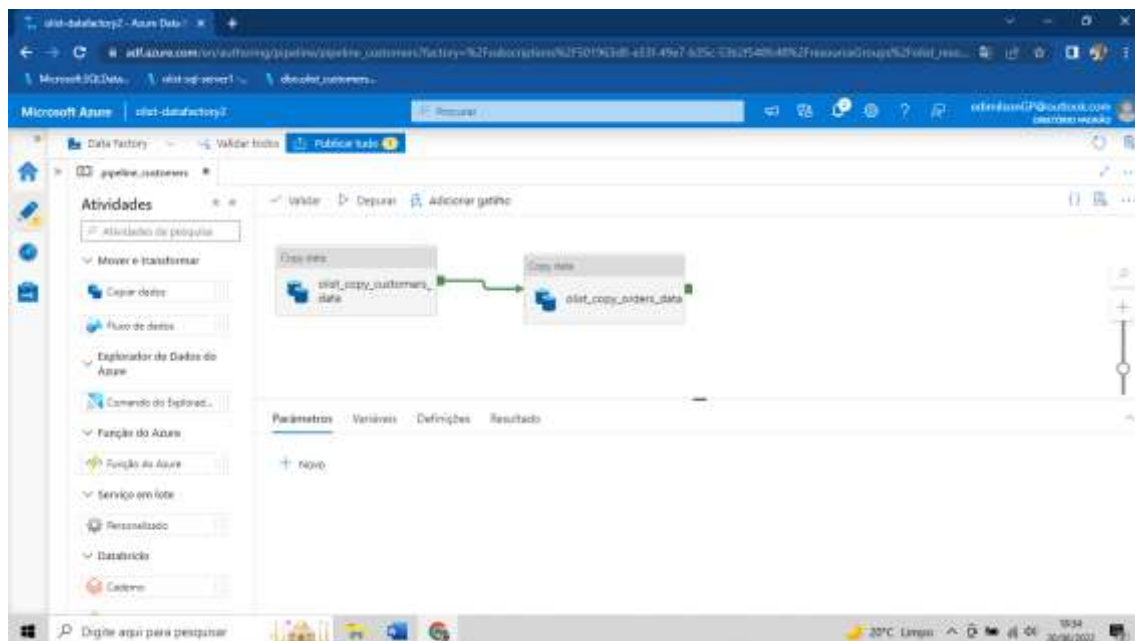
COMO JÁ EXISTE OS CAMPOS AGORA SÓ SELECIONAR OS CAMPOS CORRETOS

O DATALEKE DE LANDING JÁ EXISTE SÓ SELECIONAR NO DESTINO



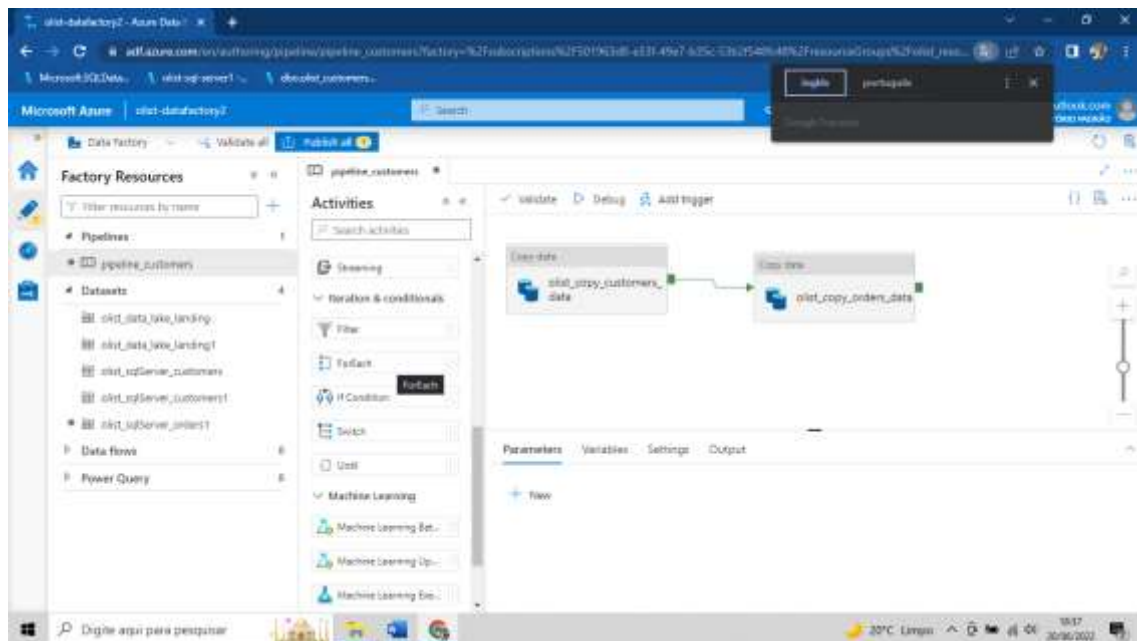
AGORA FAZENDO A CONEXÃO DOS DADOS

CLICANDO E ARRASTANDO



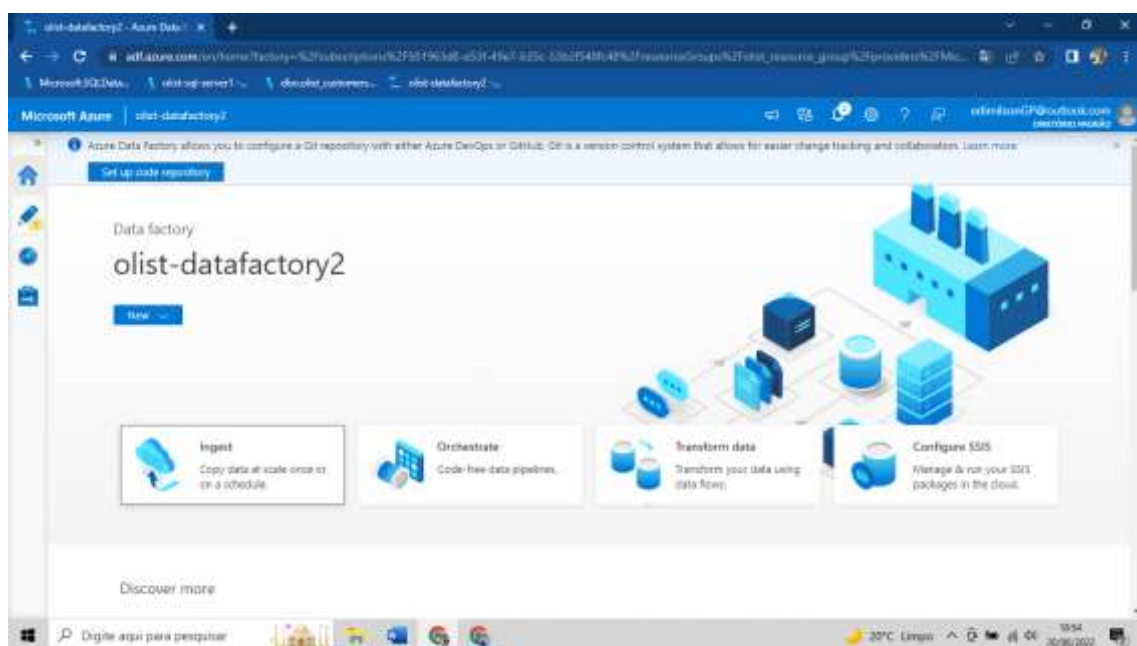
FERRAMENTA PARA CRIAÇÃO MAIS RAPIDA

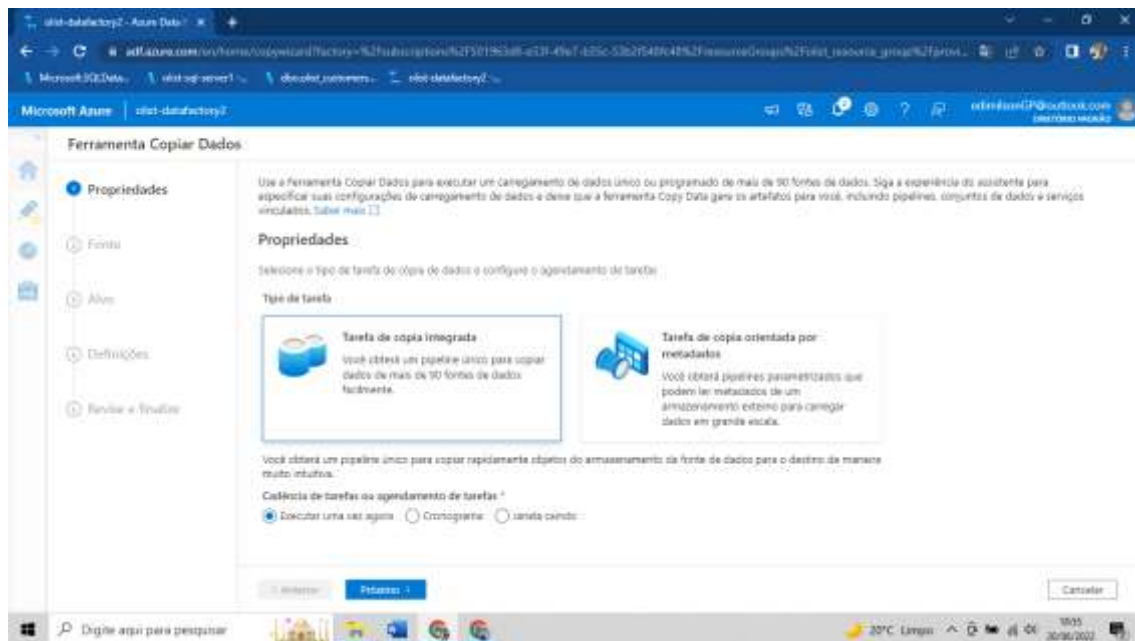
ForEach = ele repete uma tarefa de cada lista que você passa pra ele



Ingestão de dados

CRIANDO PIPELINE PELO WIZARD DO DATA FACTORY





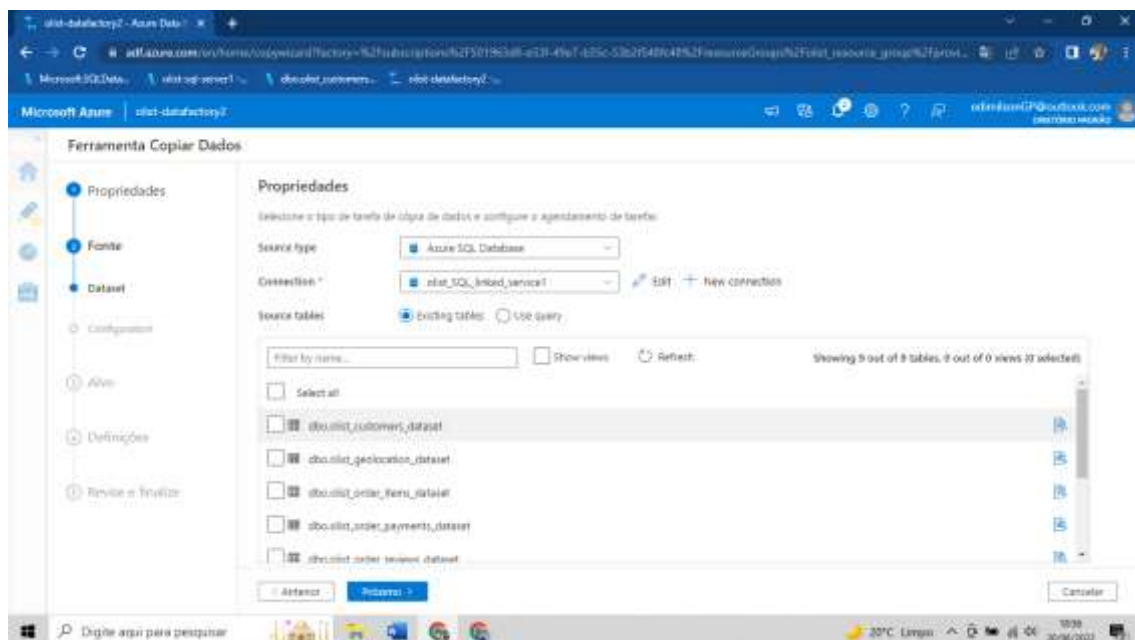
MOSTRANDO TODAS AS TABELAS DISPONÍVEIS

SELECIONANDO O BANCO DE DADOS DO SQL COM AZURE

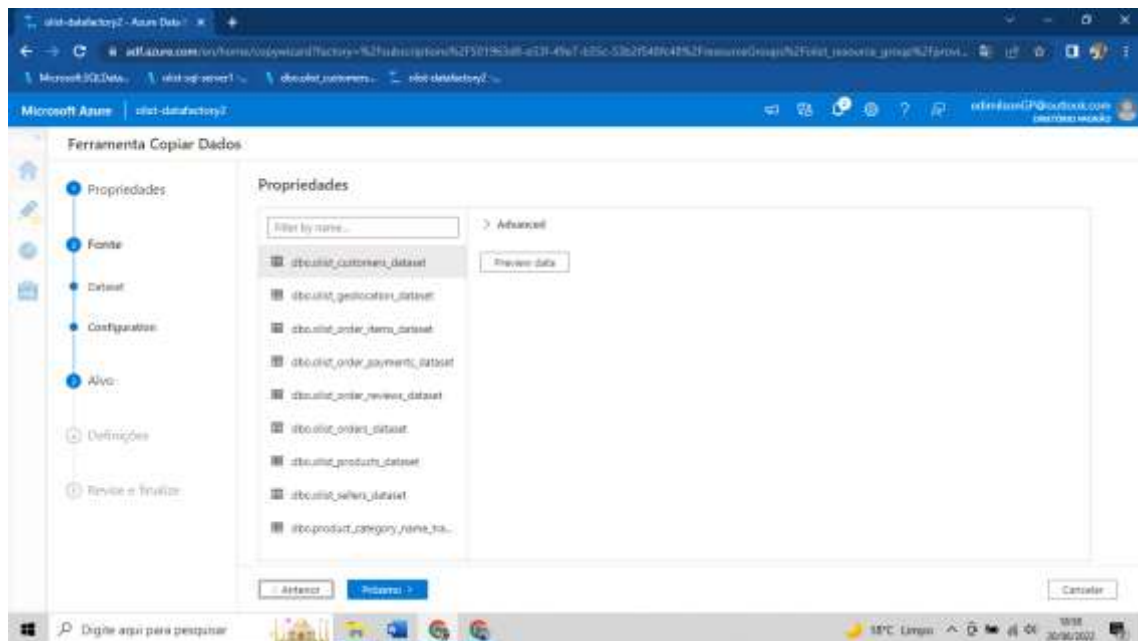
E INFORMANDO A CONEXÃO

ETL

CARGA dos DADOS



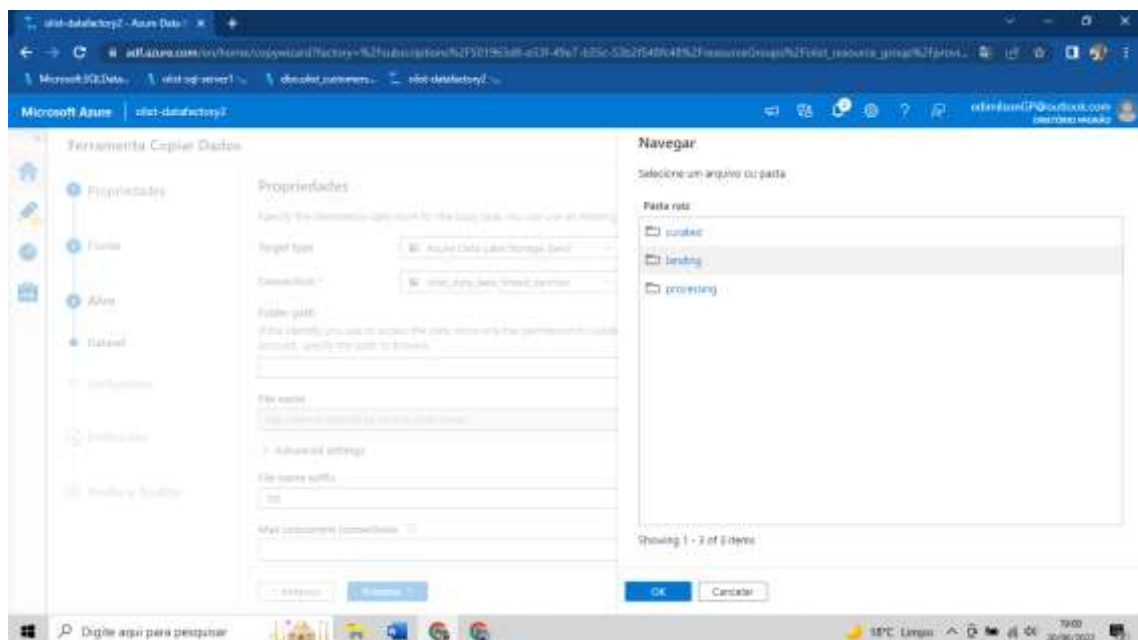
CLICANDO EM PROXIMO



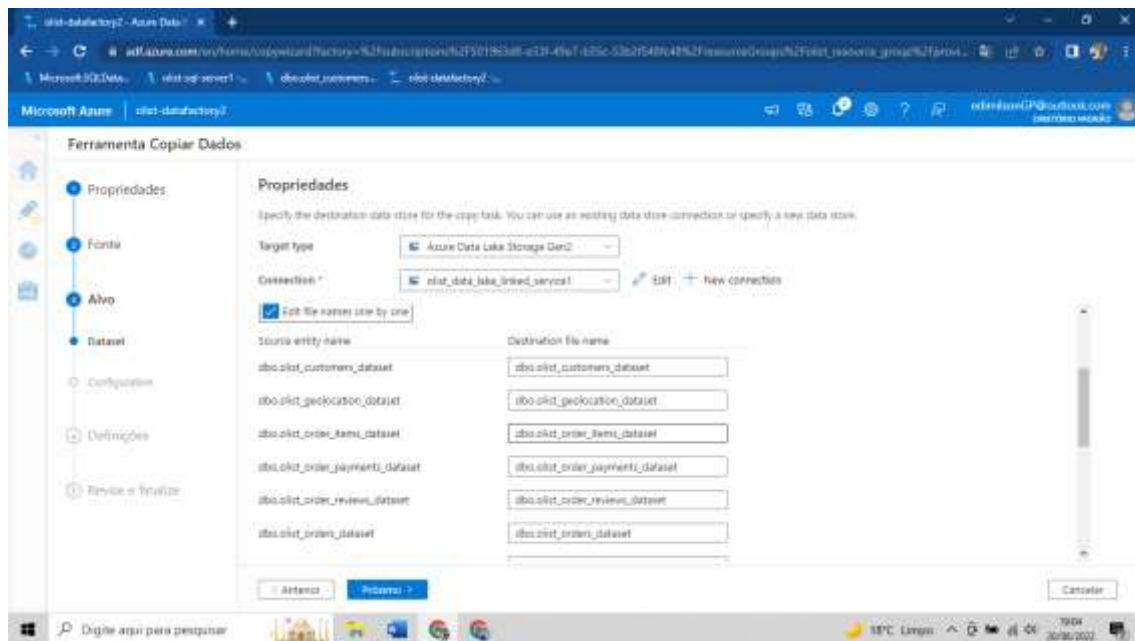
TIPO DE DESTINO

CONEXÃO

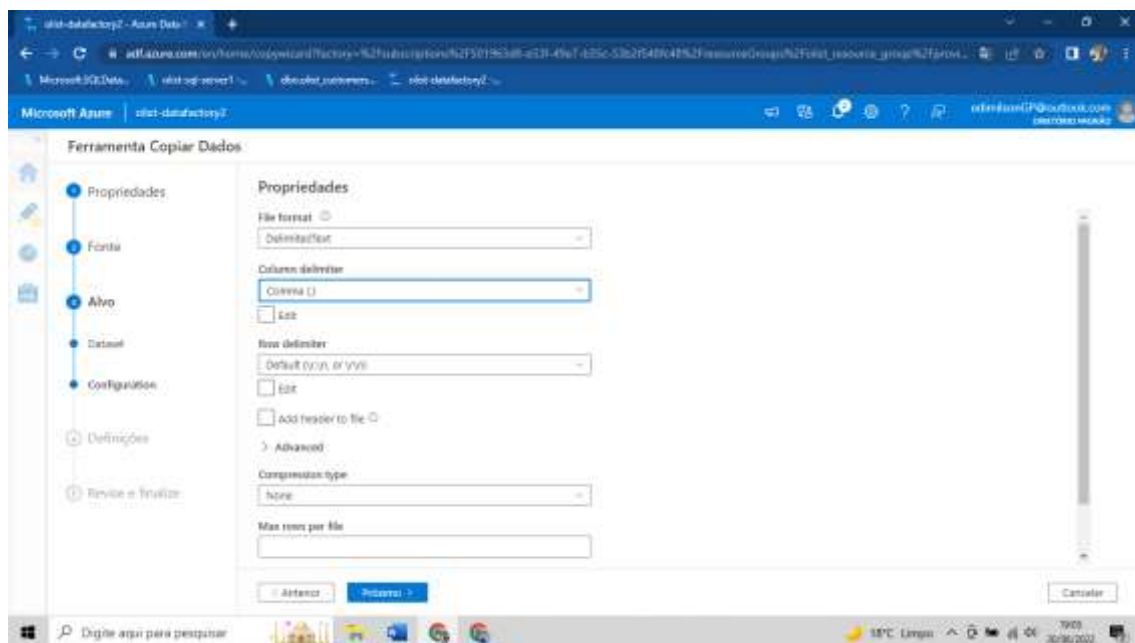
ESCOLHENDO O CONTAINER DE LANDING



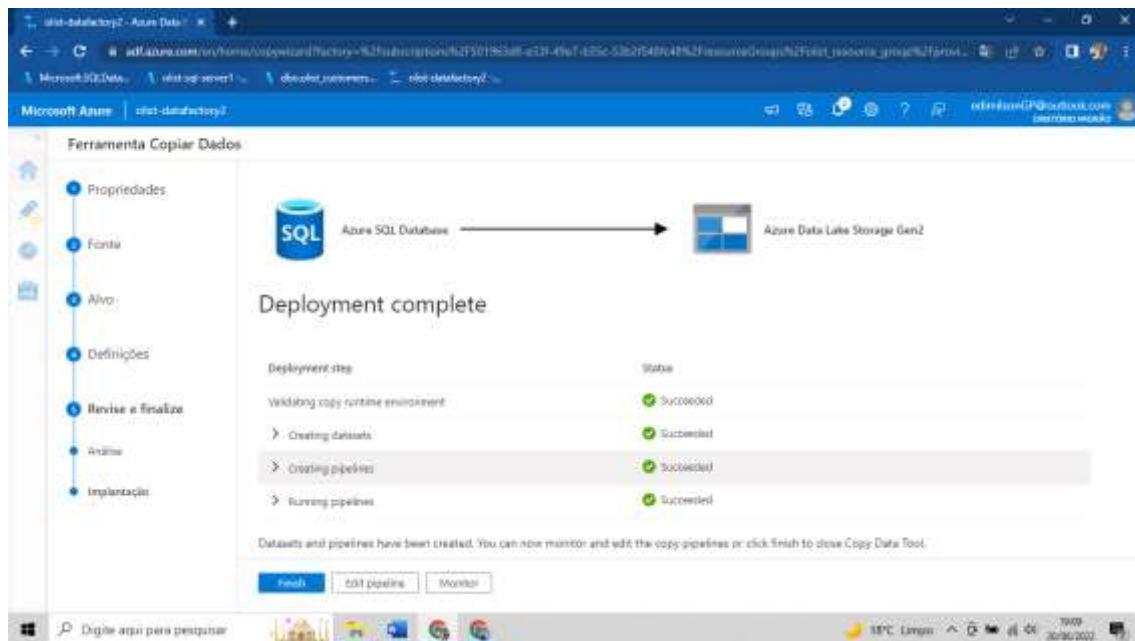
E escolhendo como formato CSV



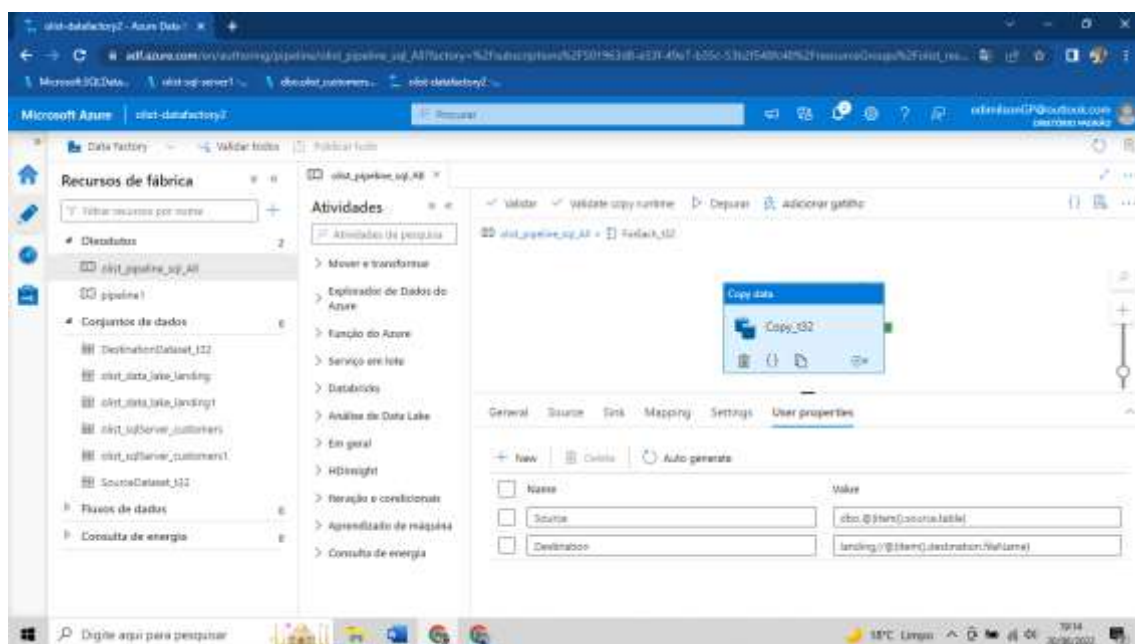
Aqui informa o delimitador que no caso é por virgula se eu quero comprimir o arquivo

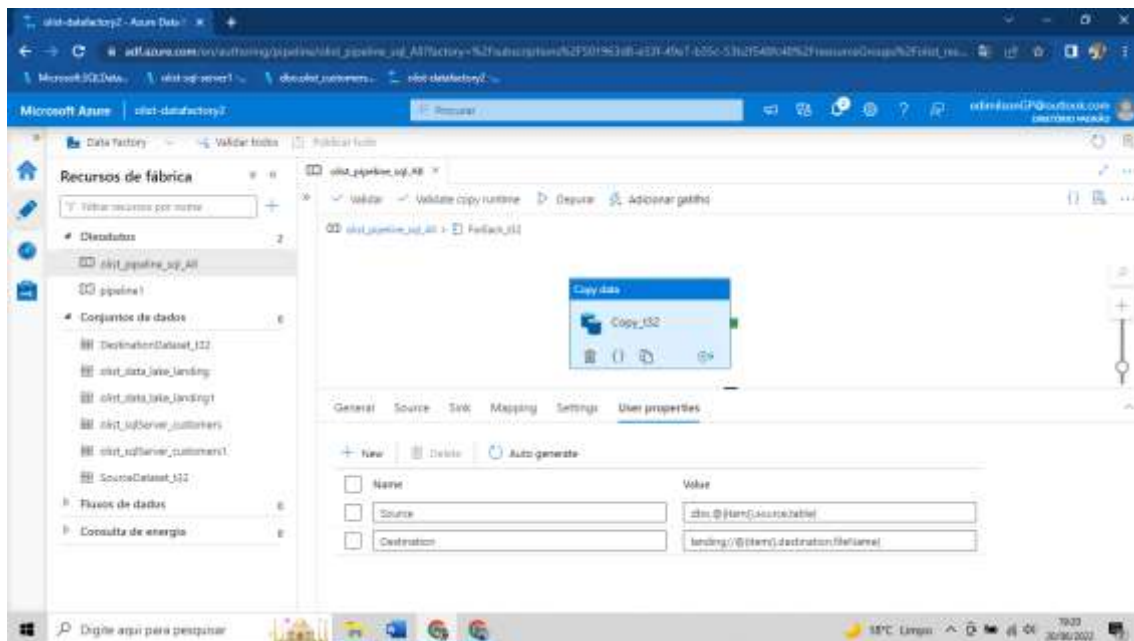


CRIANDO O NOME DO PIPELINE PARA INFORMAR QUE ESTOU ENVIANDO TODAS AS TABELAS



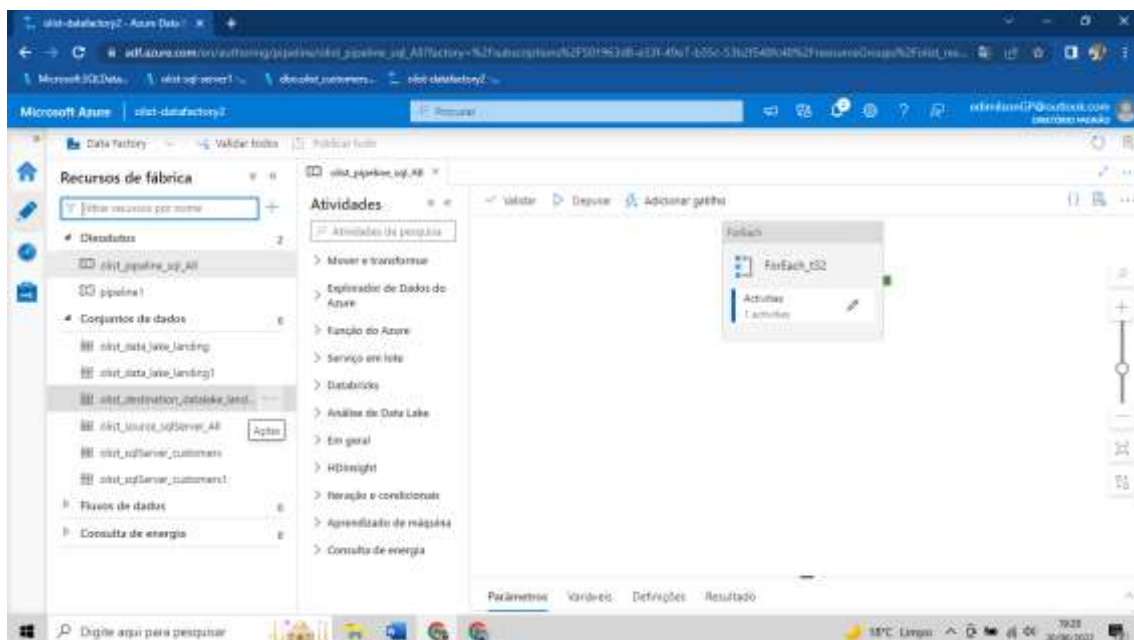
FEZ TODOS O MAPEAMENTO AGORA SÓ EXECUTAR E APERTAR CIONAR GATILHO





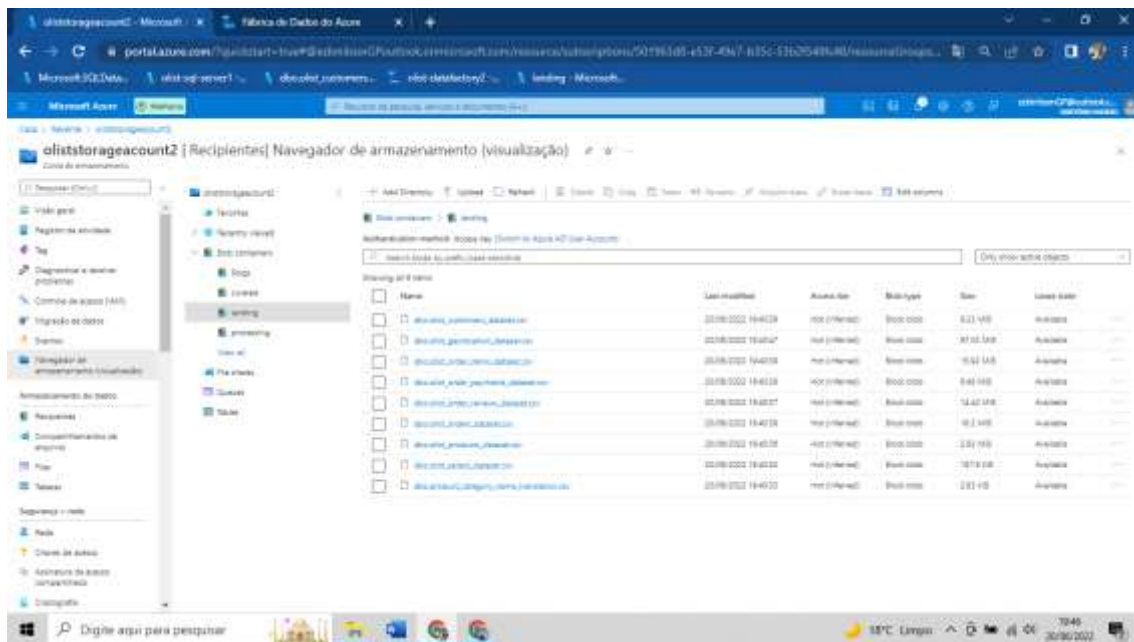
TESTANDO A EXECUÇÃO

Tabelas renomeadas Source e Destination All



Em StorageAccount todas as tabelas foram carregadas em Landing

Tabelas carregadas e gravadas



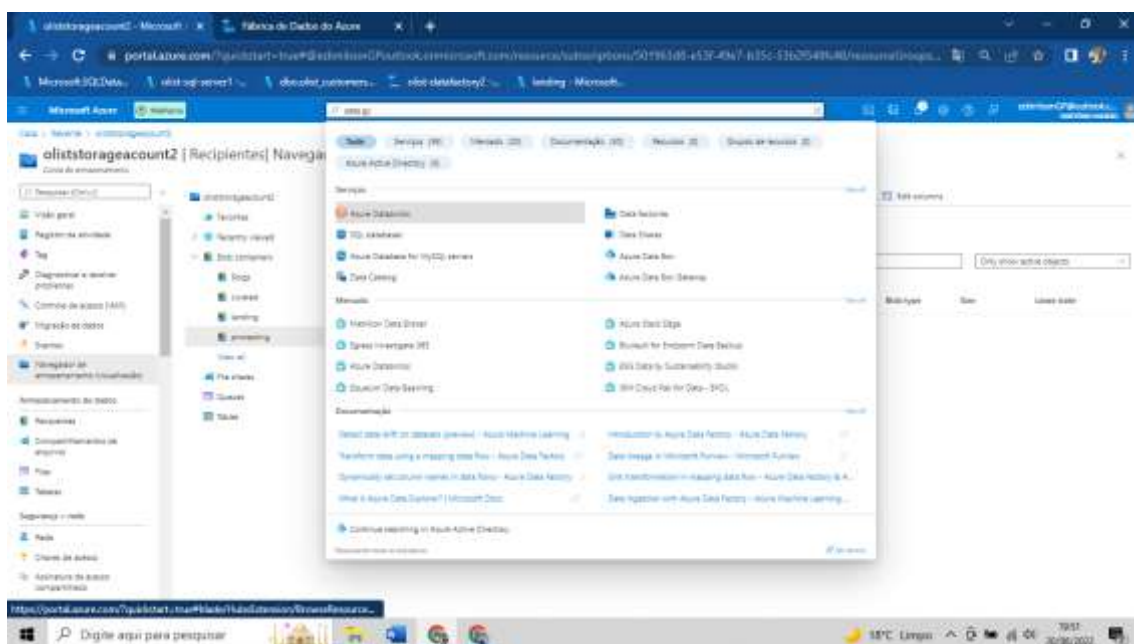
ETL

PROCESSAMENTO = TRANSFORMAÇÃO DOS DADOS

PROCURAR POR AZURE DATABRICKS

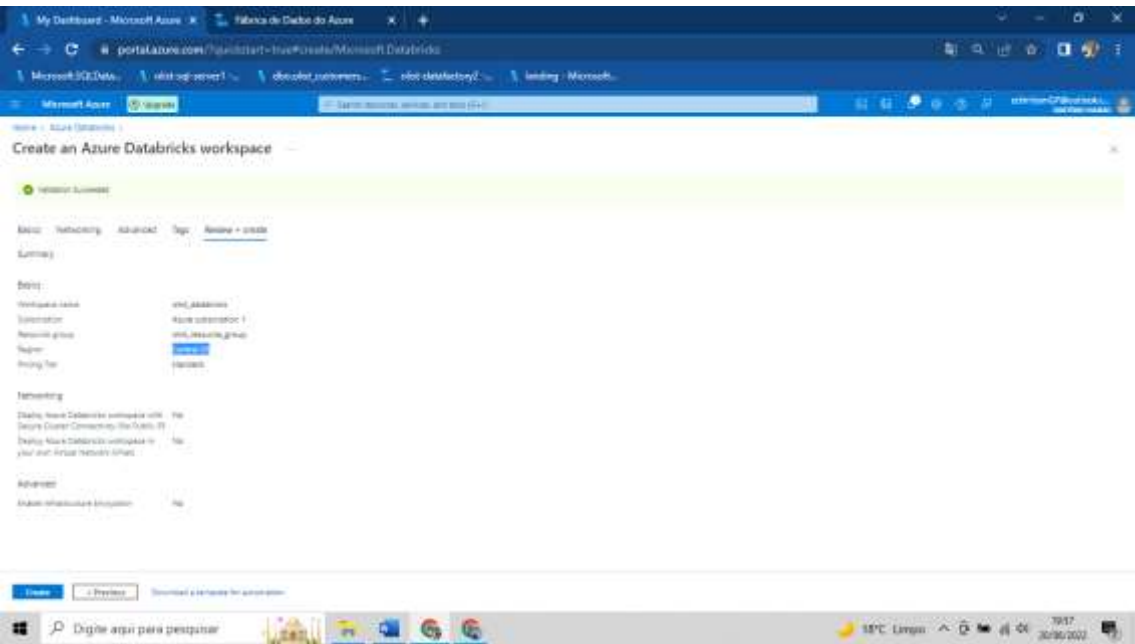
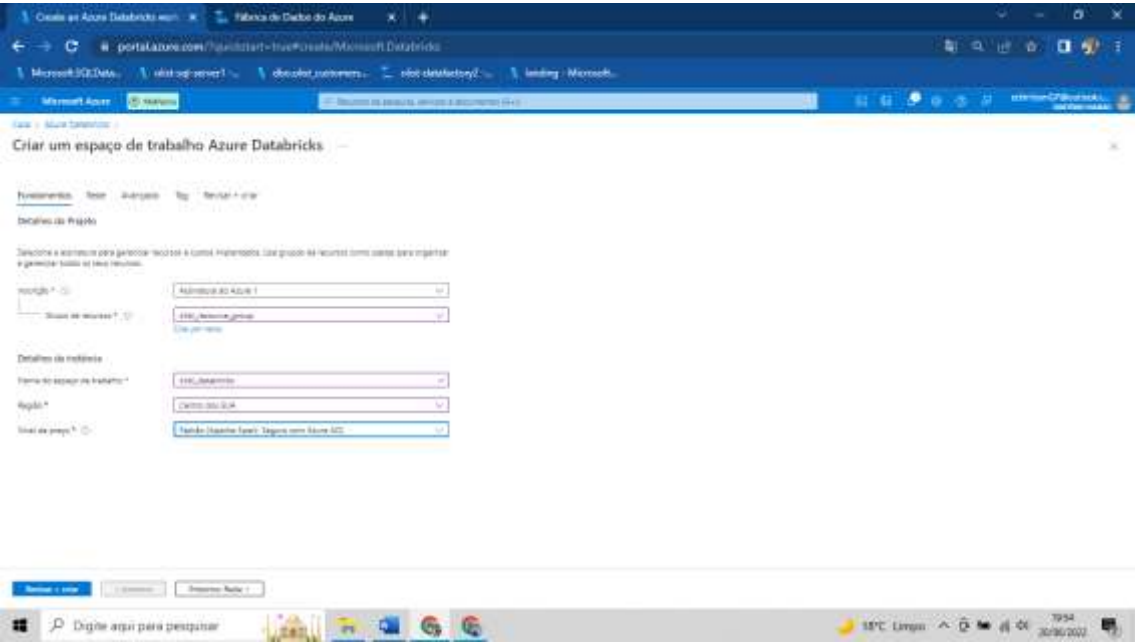
Depois clicar em Criar

CRIANDO O DATABRICKS

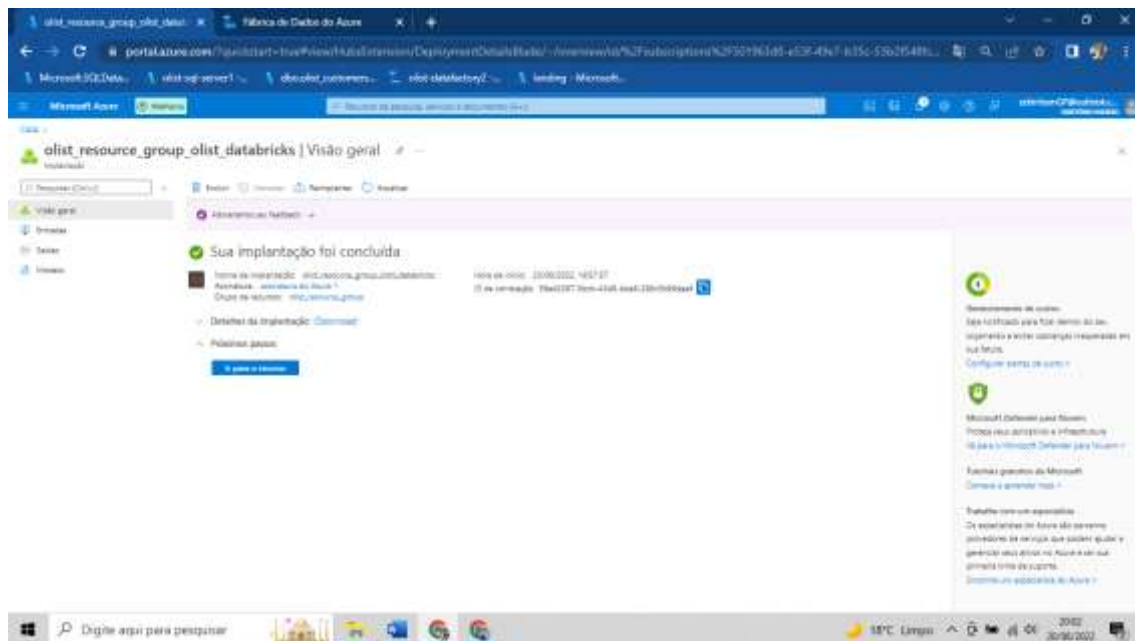


Preenchendo as informações de assinatura e selecionando o grupo de recursos e nome do espaço de trabalho

Região mais barata para o custo com DATABRICKS

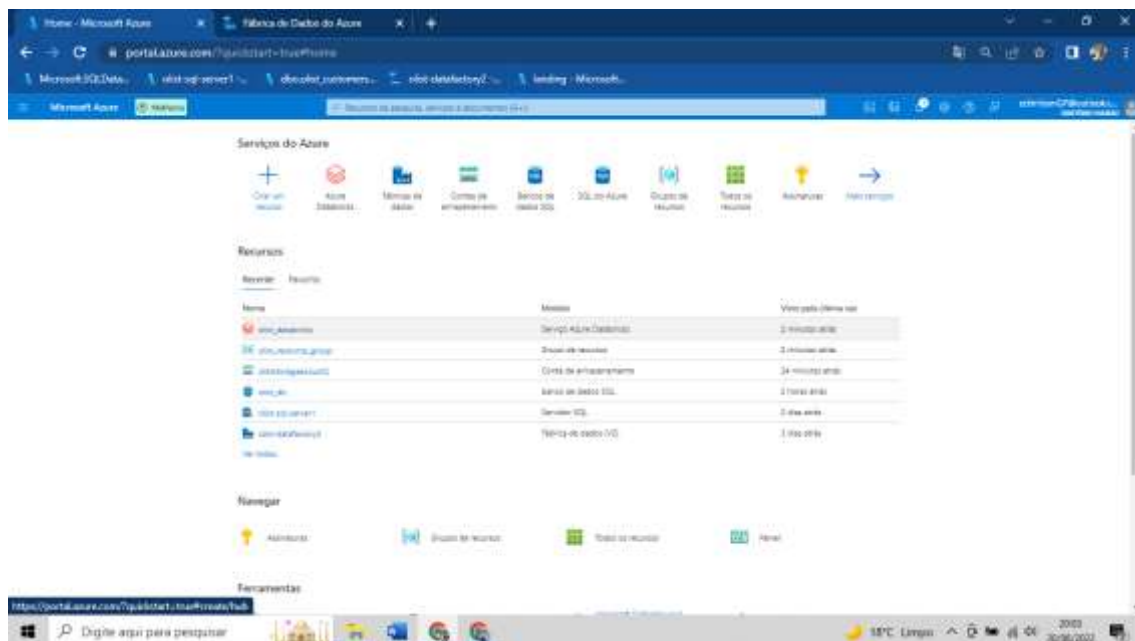


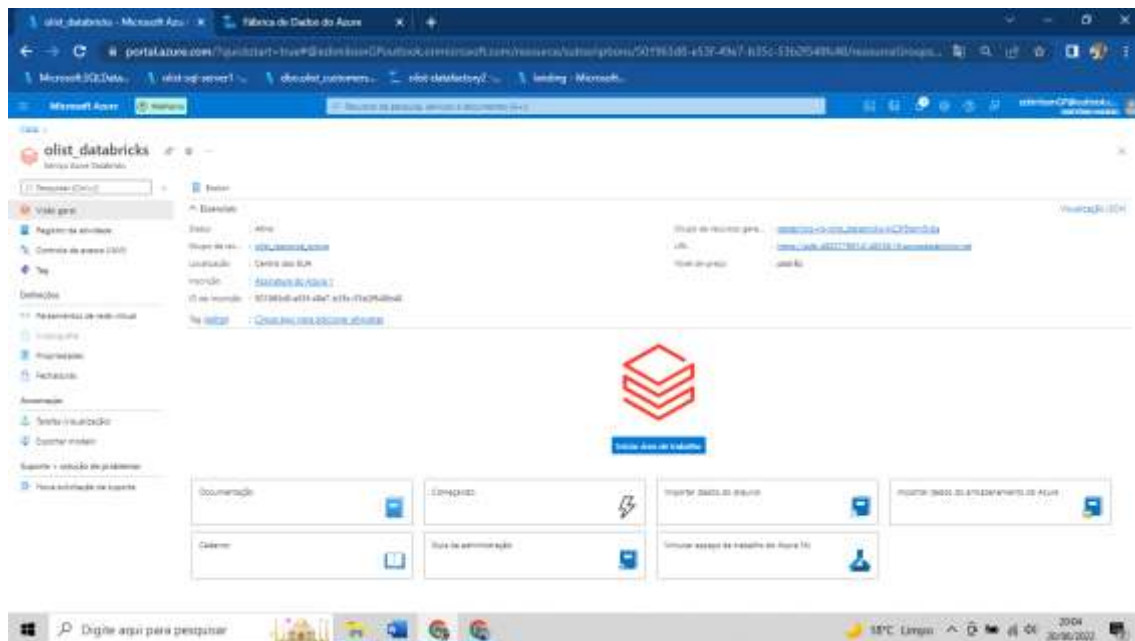
CRIADO O DATABRICKS



VAMOS ACESSAR O RECURSO DO DATABRICKS

INDO NA PAGINA INICIAL



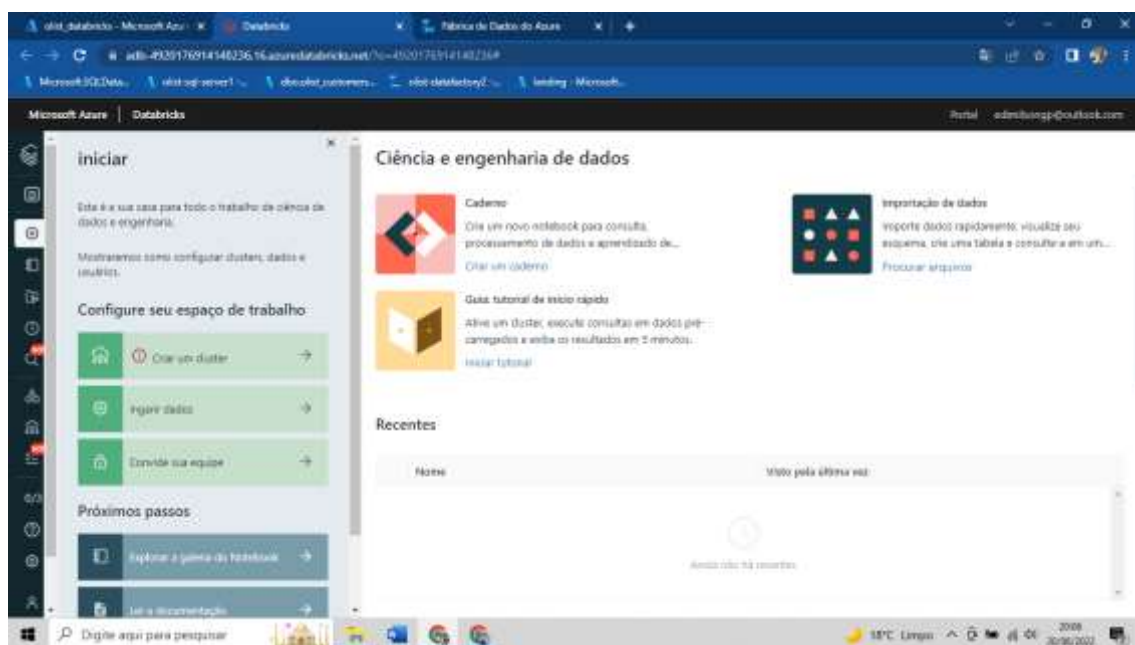
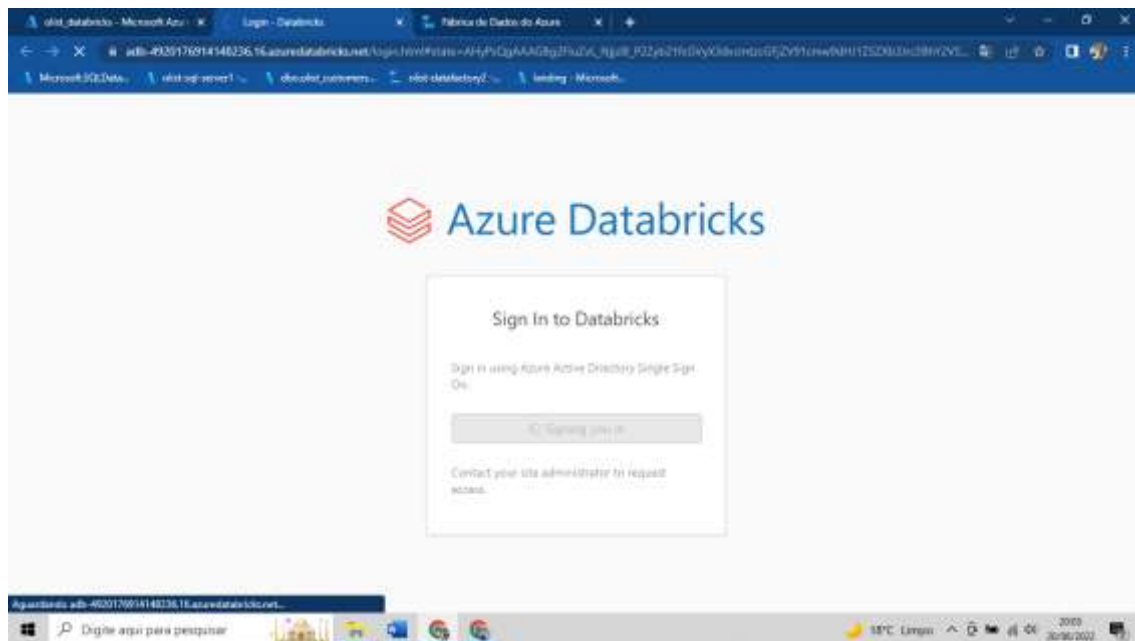


CLICAR EM

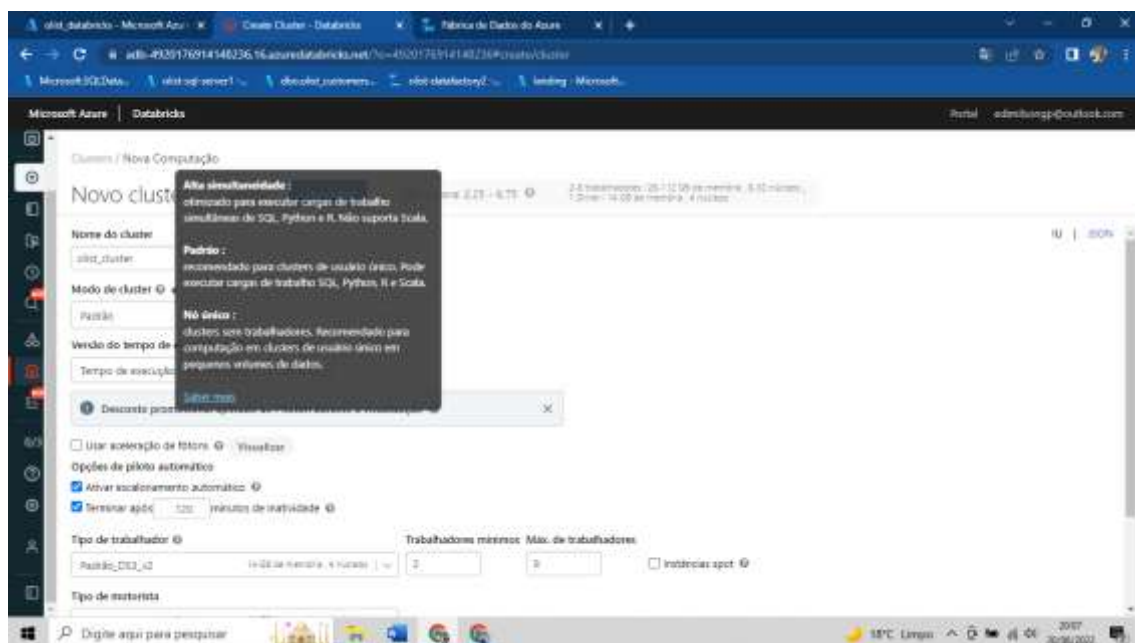
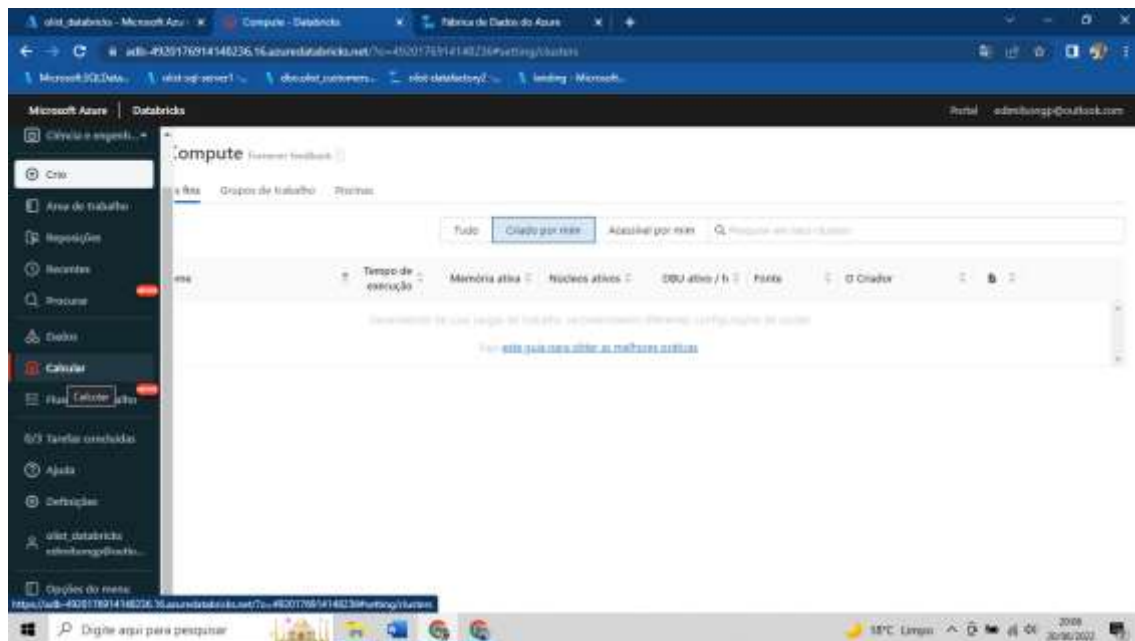


Iniciar área de trabalho

SE CONECTANDO

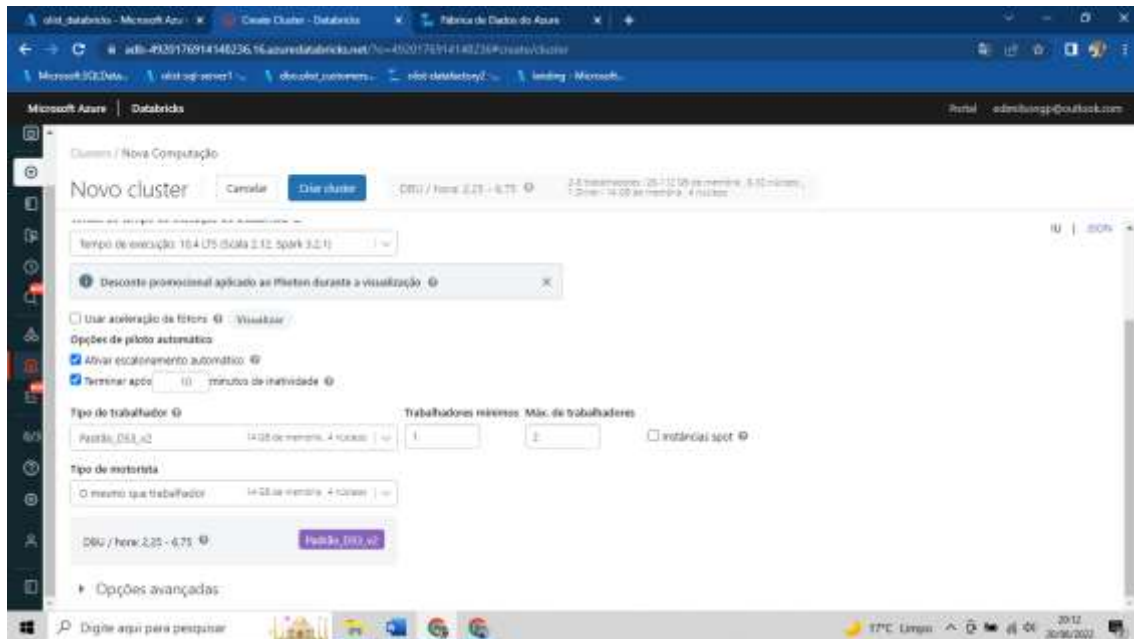


ENTRANDO NO WORKSPACE DO DATABRICKS E CRIANDO UM CLUSTER DE PROCESSAMENTO



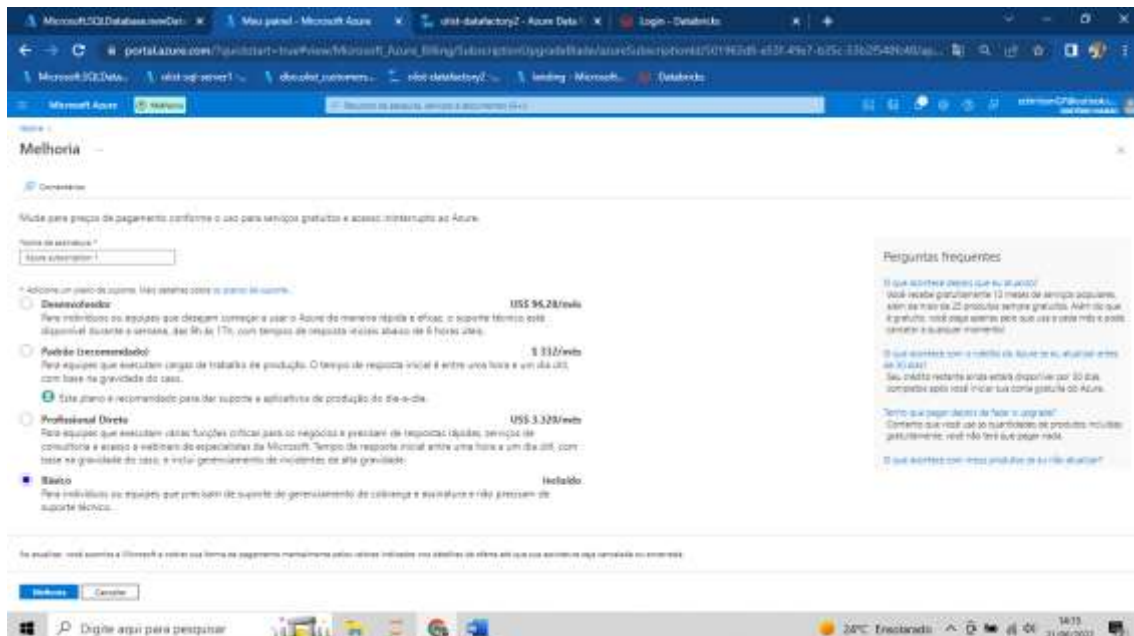
ATENÇÃO PARA O TEMPO

SEMPRE DESATIVAR O CLUSTER

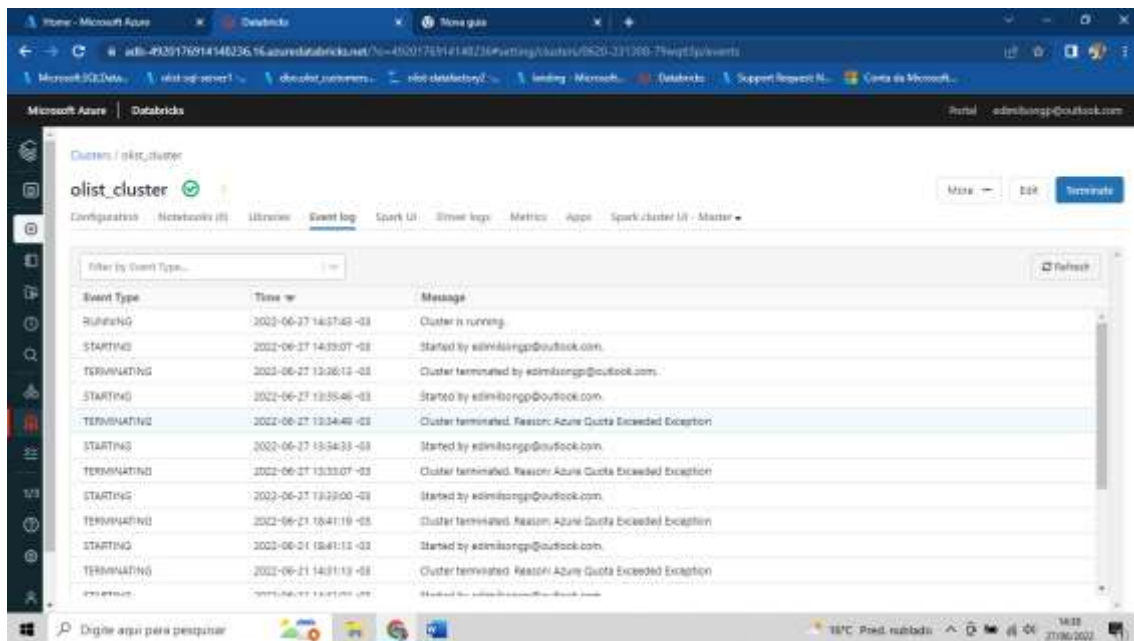


COMO NÃO STARTOU DEVEMOS FAZER UPGRAD DA CONTA PARA PAGO CONFORME O USO, PARA CONSERGUIR UTILIZAR O CLUSTER DO DATABRICKS

FAZER UPGRAD DA CONTA PARA PAGO CONFORME O USO, PARA CONSERGUIR UTILIZAR O CLUSTER DO DATABRICKS



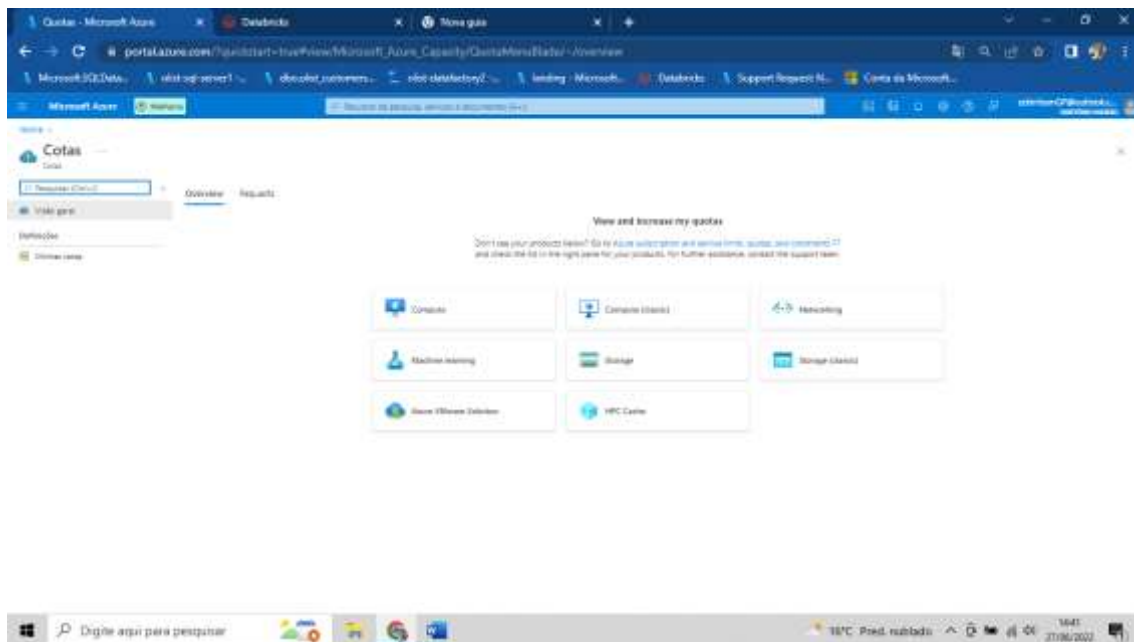
CLUSTER CRIADO E ASSINATURA ATIVADA



CASO MEU CLUSTER NÃO SUBA CORRETAMENTE IREMOS SOLICITAR O AUMENTO DE COTAS

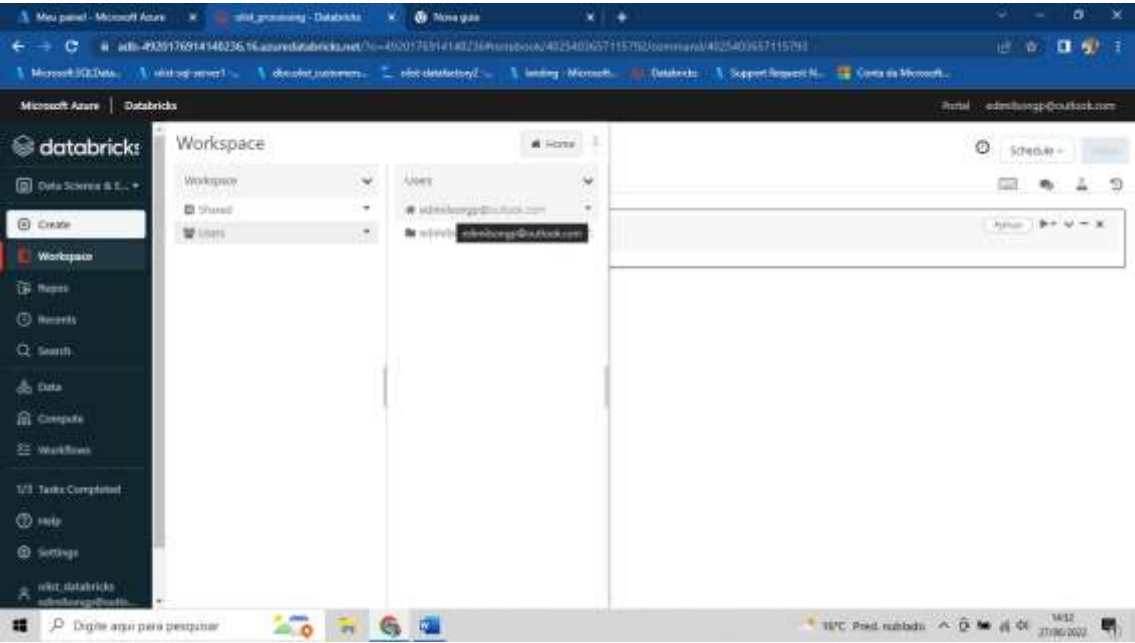
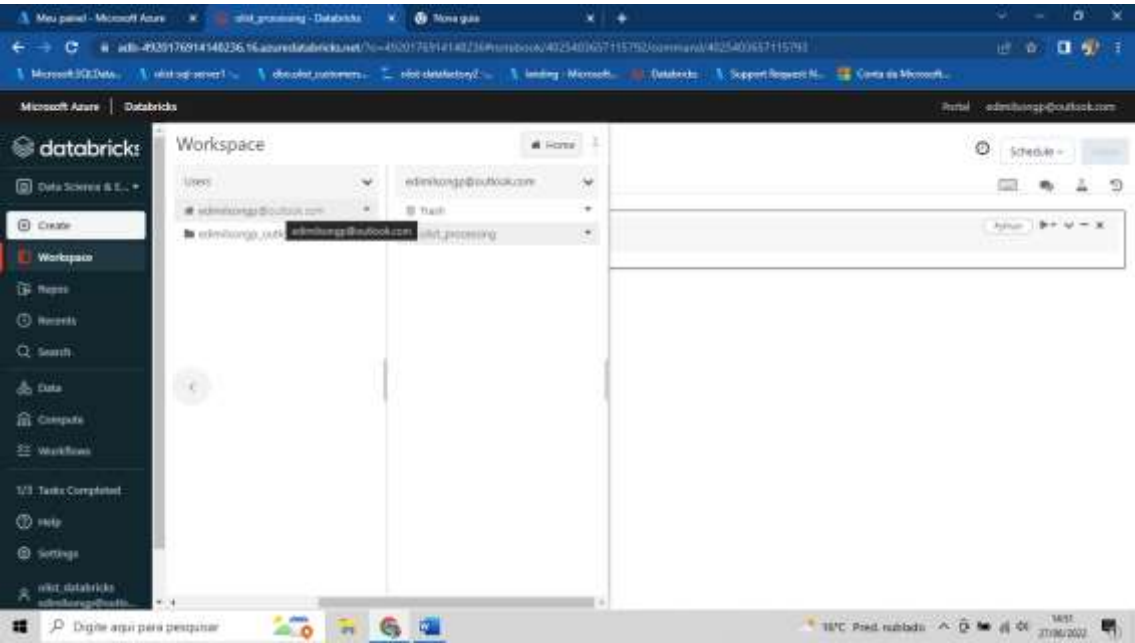
CASO TENHA SUBIDO DE FORMA CORRETA NÃO PRECIS, MAS VAMOS APRENDER COMO FAZER.

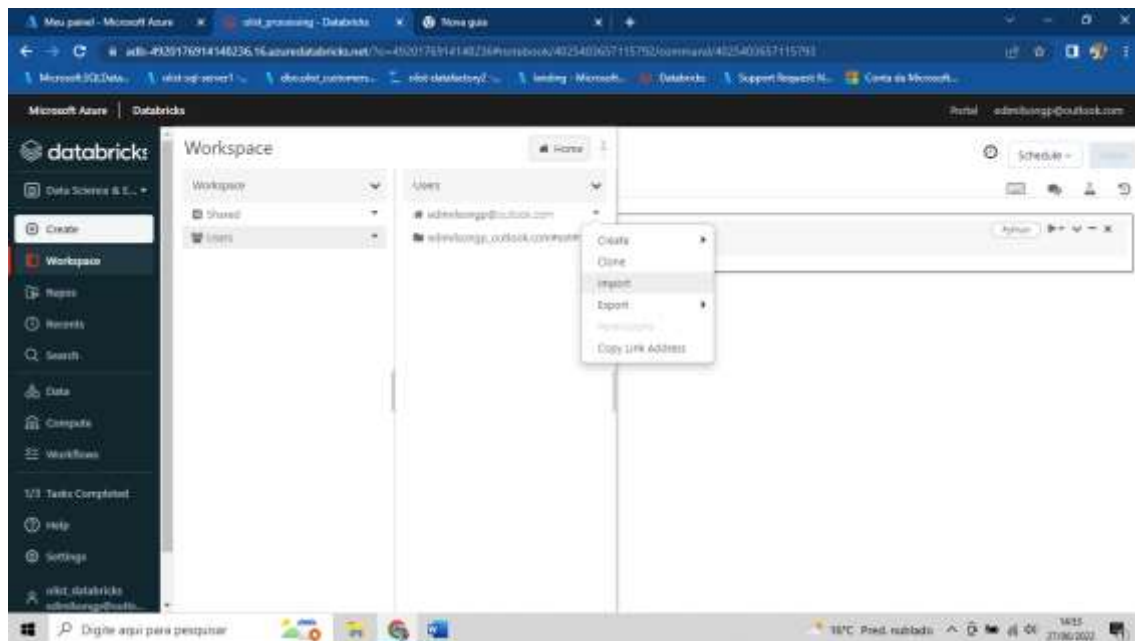
PESQUISAR POR COTA



AUMENTO DE CORE ENTRAR EM COMPUTE

IMPORTANDO O SCRIPT





SCRIP IMPORTADO

PASSO A PASSO DO QUE O SCRIPT FAZ

CONFIGURAÇÃO A SEGUIR CONSTA NA DOCUMENTAÇÃO, NECESSARIO APENAS ADAPTAR AS INFORMAÇÕES AOS SEU ARQUIVOS.

#Carregar, Transformar, Persistir Pipeline

#1 - monte os data lakes

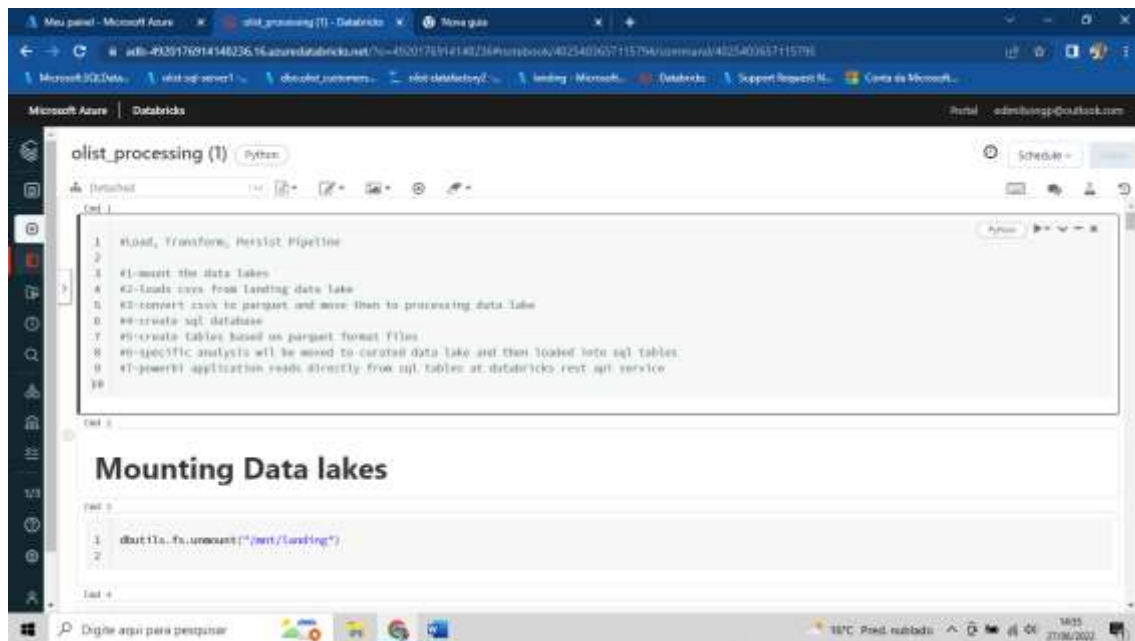
#2-carrega csvs do data lake de desembarque

#3-converter csvs para parquet e passar para o data lake de processamento

#4-criar banco de dados sql

#5-crie tabelas com base em arquivos de formato parquet

A análise específica nº 6 será movida para o data lake com curadoria e, em seguida, carregada em tabelas sql

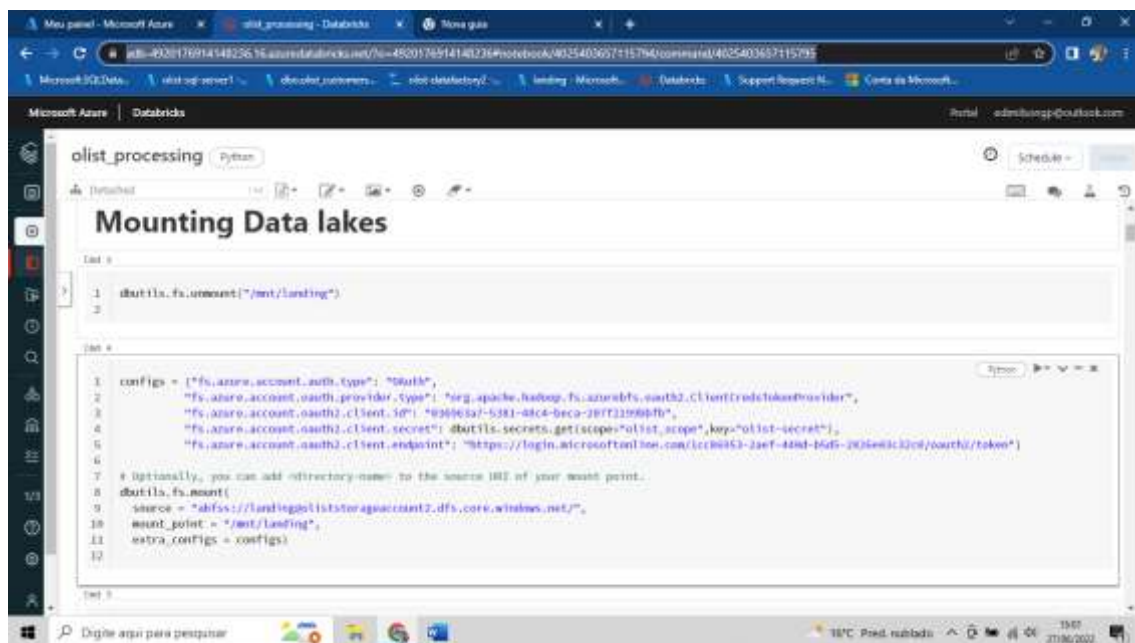


The screenshot shows a Databricks notebook interface. The top bar indicates the notebook is in 'Python' mode. The main content area displays a Python script for data processing, followed by a section titled 'Mounting Data lakes'.

```
1 #Load, Transform, Persist Pipeline
2
3 #1-mount the data lakes
4 #2-loads csvs from landing data lake
5 #3-convert csvs to parquet and move them to processing data lake
6 #4-create sql database
7 #5-create tables based on parquet format files
8 #6-specific analysts will be moved to curated data lake and then loaded into sql tables
9 #7-powerbi application reads directly from sql tables at databricks rest api service
10
```

Mounting Data lakes

```
1 dbutils.fs.unmount("/mnt/landing")
2
```



The screenshot shows a Databricks notebook interface. The top bar indicates the notebook is in 'Python' mode. The main content area displays a Python script for mounting an Azure Data Lake Storage account.

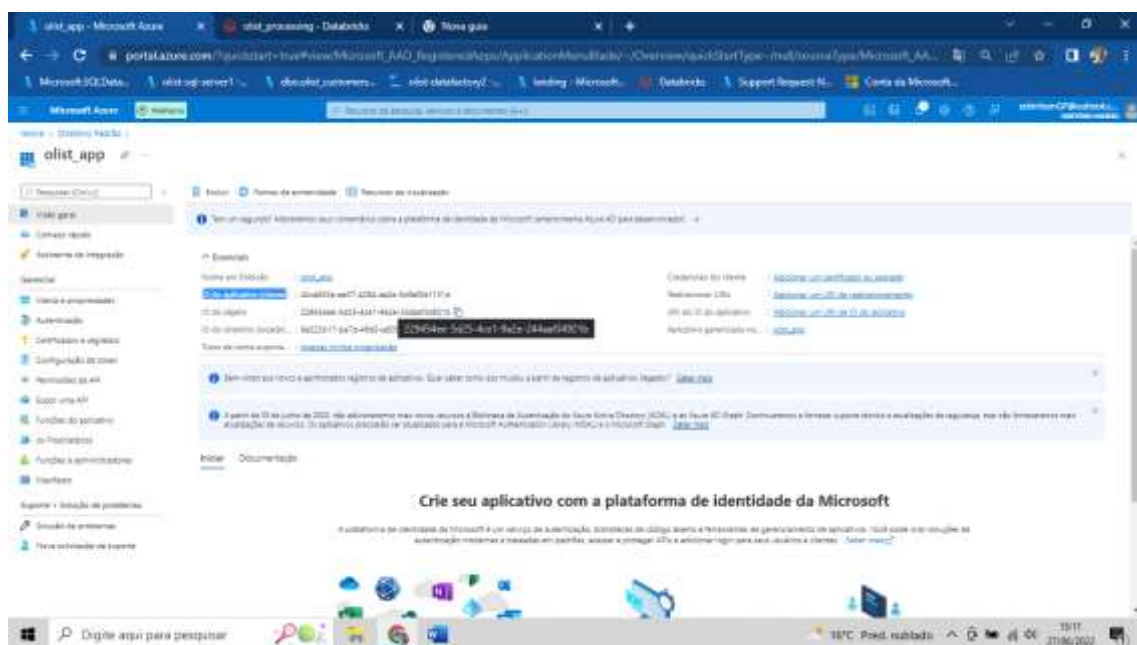
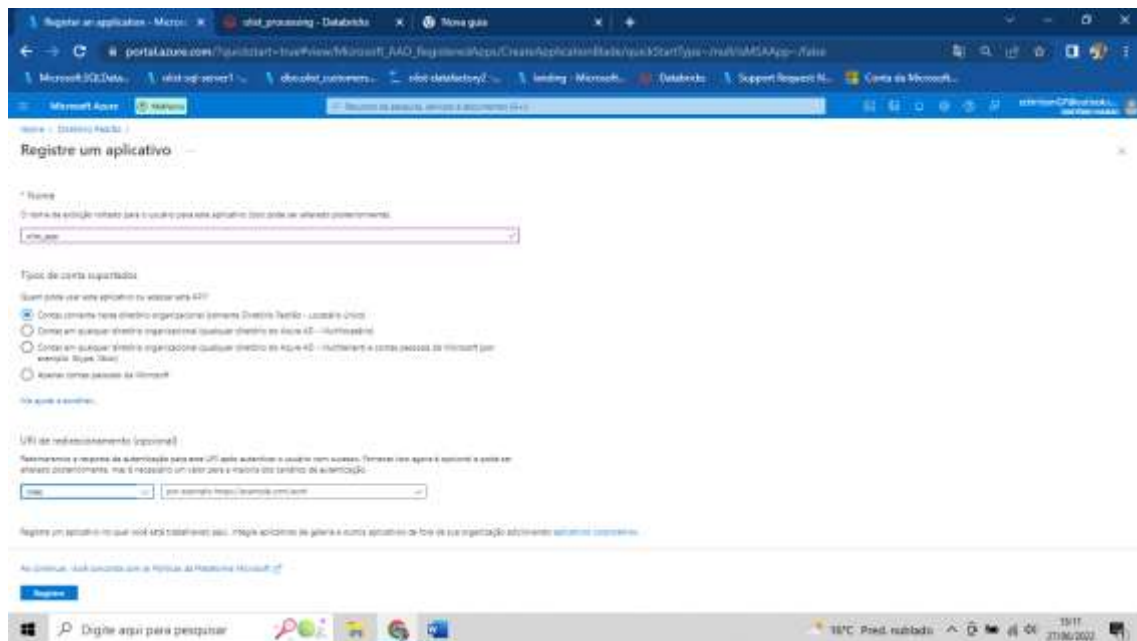
```
1 dbutils.fs.unmount("/mnt/landing")
2
3
4
5
6
7
8
9
10
11
12
```

```
1 configs = [{"fs.azure.account.auth.type": "OAuth",
2             "fs.azure.account.auth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3             "fs.azure.account.oauth2.client.id": "9d966a2-5381-48c4-beca-28f23199bfb",
4             "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="olist_scope", key="olist-secret"),
5             "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/ccc6a53-2aef-448d-b4d5-2036e6c32cd/oauth2/token"}]
6
7 # optionally, you can add <directory-name> to the source URI of your mount print.
8 dbutils.fs.mount(
9     source = "abfss://landingoliststorageeast2.dfs.core.windows.net/",
10     mount_point = "/mnt/landing",
11     extra_configs = configs)
12
```

MONTAR O DATA LAKE COM PYSPARK NO DATABRICKS

PRECISAMOS REGISTRAR UMA APLICAÇÃO

Azure Active Directory



Copiar 43ca923e-ee07-4284-ae2e-5dfe00e1131d

E colar no lugar do 036963a7-5381-48c4-beca-207f2199bbfb

The screenshot shows a Databricks notebook interface with the title "Mounting Data lakes". The code is written in Python and includes the following lines:

```
1 dbutils.fs.unmount("/mnt/landing")
2
3
4
5
6
7
8
9
10
11
12
```

The code defines a configuration dictionary for mounting an Azure Data Lake Storage account. The configuration includes the following keys and values:

- `fs.azure.account.auth.type`: "OAuth2"
- `fs.azure.account.oauth.provider.type`: "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider"
- `fs.azure.account.oauth2.client.id`: "93b632-5381-4848-9c9-287f2196b1b"
- `fs.azure.account.oauth2.client.secret`: `dbutils.secrets.get(scope="olist_scope", key="olist-secret")`
- `fs.azure.account.oauth2.client.endpoint`: "https://login.microsoftonline.com/6c86953-2aef-448d-b6d5-9036e8332ed/oauth2/token"

The code also includes a comment: "Optionally, you can add <directory-name> to the source URI of your mount point." and the following lines:

```
dbutils.fs.mount(
    source = "abfss://landingb01st0r0g0acc0nt2.dfs.core.windows.net/",
    mount_point = "/mnt/landing",
    extra_configs = configs)

```

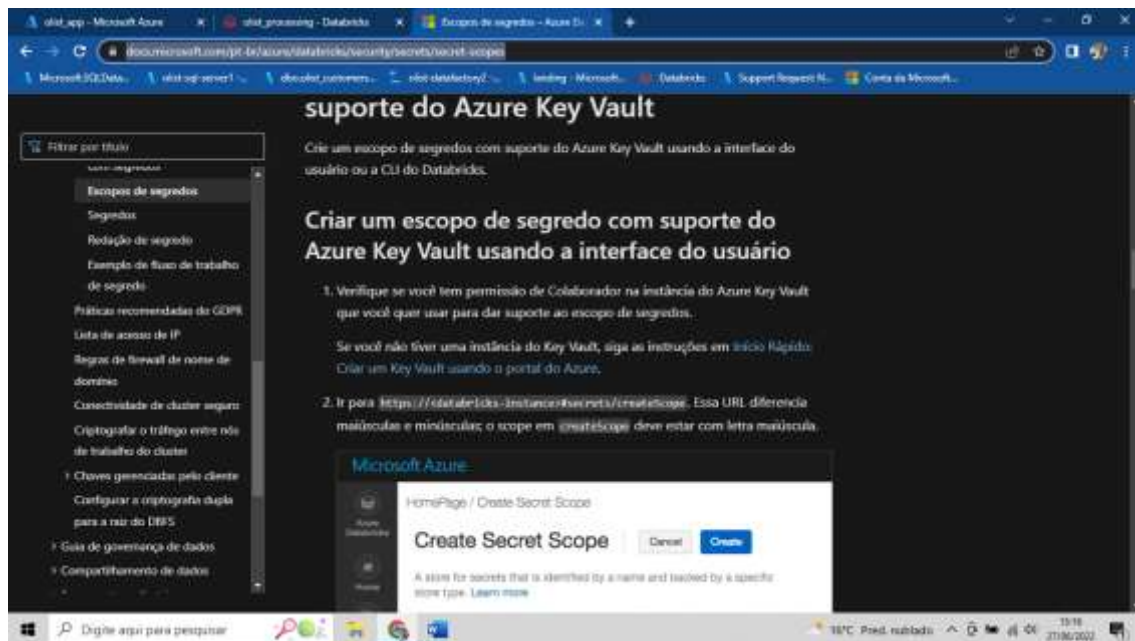
The screenshot shows the same Databricks notebook interface as the first image. The code is identical, but a red box highlights the client ID and secret key in the configuration dictionary:

```
1 configs = [{"fs.azure.account.auth.type": "OAuth2",
2             "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3             "fs.azure.account.oauth2.client.id": "93b632-5381-4848-9c9-287f2196b1b", // client ID
4             "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="olist_scope", key="olist-secret"),
5             "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/6c86953-2aef-448d-b6d5-9036e8332ed/oauth2/token"}]
6
7
8
9
10
11
12
```

<https://docs.microsoft.com/pt-br/azure/databricks/security/secrets/secret-scopes>

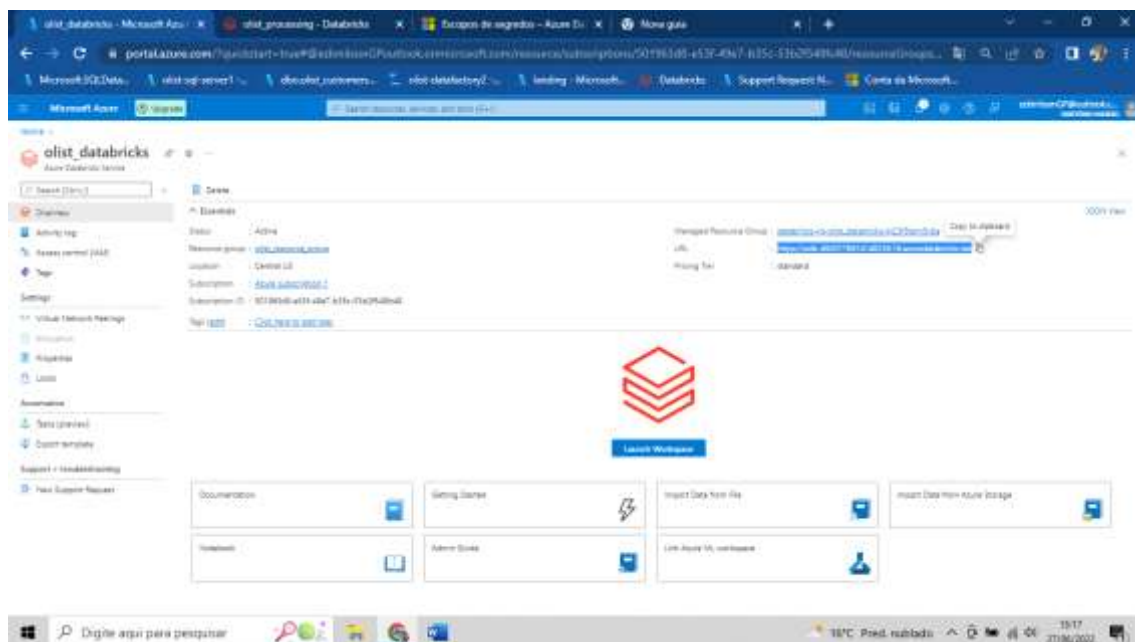
URL DA DOCUMENTAÇÃO

<https://<databricks-instance>secrets/createScope>



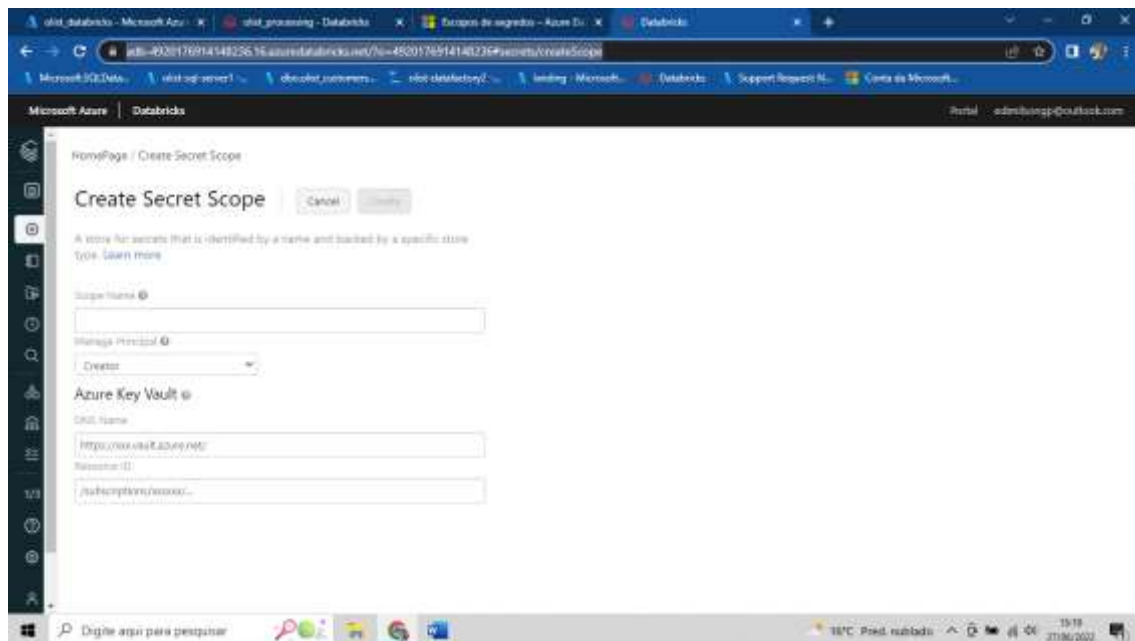
Copiando a URL

<https://adb-4920176914148236.16.azuredatabricks.net>



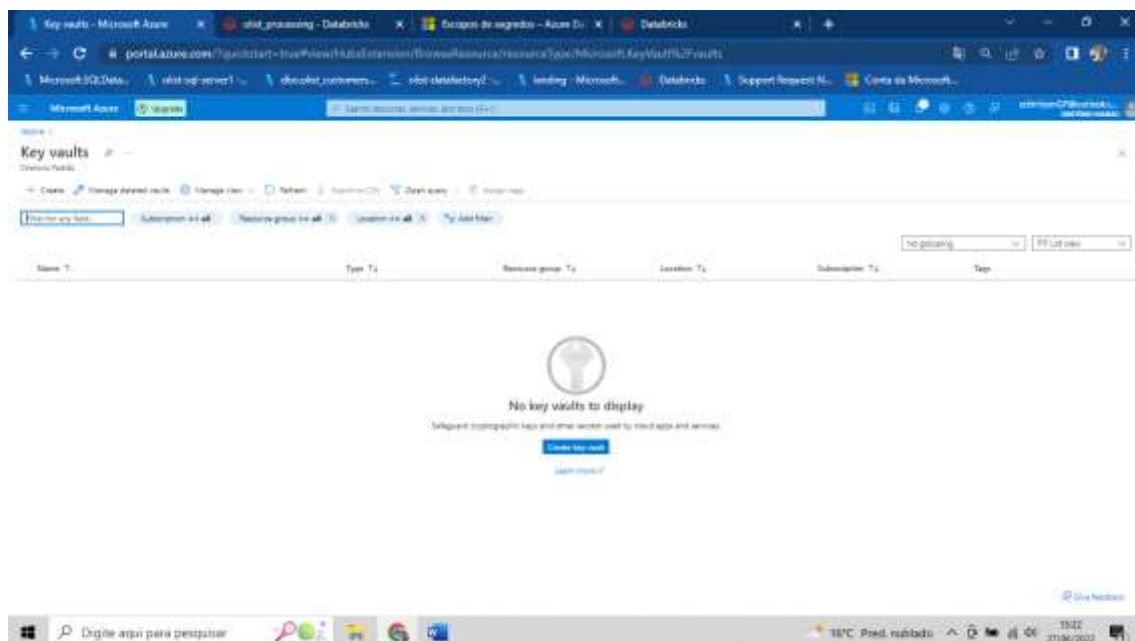
ALTERANDO A URL

<https://adb-4920176914148236.16.azuredatabricks.net/#secrets/createScope>



Criar AZURE KEY VAULT

PESQUISAR PELO KEY VAULT



Abra um cofre de chaves - Microsoft Azure

portal.azure.com/?appid=7f4e0c0a-7b4e-4b4e-8b4e-0b4e0b4e0b4e

Microsoft Azure

Recursos de produtos, serviços e documentos (0/0)

Abra um cofre de chaves

Criar um cofre de chaves

Definir as opções

Defina as opções para gerenciar recursos e configurar o cofre de chaves. Você pode definir o nome do cofre de chaves, o nome do grupo de recursos, a região e o tipo de proteção.

Nome do grupo de recursos:

Nome do cofre de chaves:

Região:

Tipo de proteção:

Definir as opções de proteção

A proteção de software é a opção recomendada para a maioria dos casos. Ela oferece a melhor proteção para os dados armazenados no cofre de chaves e os segredos armazenados no cofre de chaves.

Selecione um período de retenção obrigatório e opcional a ser aplicado permanentemente às cópias de backup dos segredos armazenados no período de retenção. Você pode optar por proteção como imagem. Quando a proteção como imagem está habilitada, os segredos são criados em imagens que são armazenados no cofre de chaves.

Exatidão esperada:

Quais são as opções de acesso?

[Retornar à lista](#) [Criar](#) [Cancelar](#) [Proteger: Habilitar o acesso](#)

CLICAR EM PROXIMO ATÉ CRIAR E VALIDAR

Abra um cofre de chaves - Microsoft Azure

portal.azure.com/?appid=7f4e0c0a-7b4e-4b4e-8b4e-0b4e0b4e0b4e

Microsoft Azure

Recursos de produtos, serviços e documentos (0/0)

Abra um cofre de chaves

Criar um cofre de chaves

Definir as opções

Defina as opções para gerenciar recursos e configurar o cofre de chaves. Você pode definir o nome do cofre de chaves, o nome do grupo de recursos, a região e o tipo de proteção.

Nome do grupo de recursos:

Nome do cofre de chaves:

Região:

Tipo de proteção:

Definir as opções de proteção

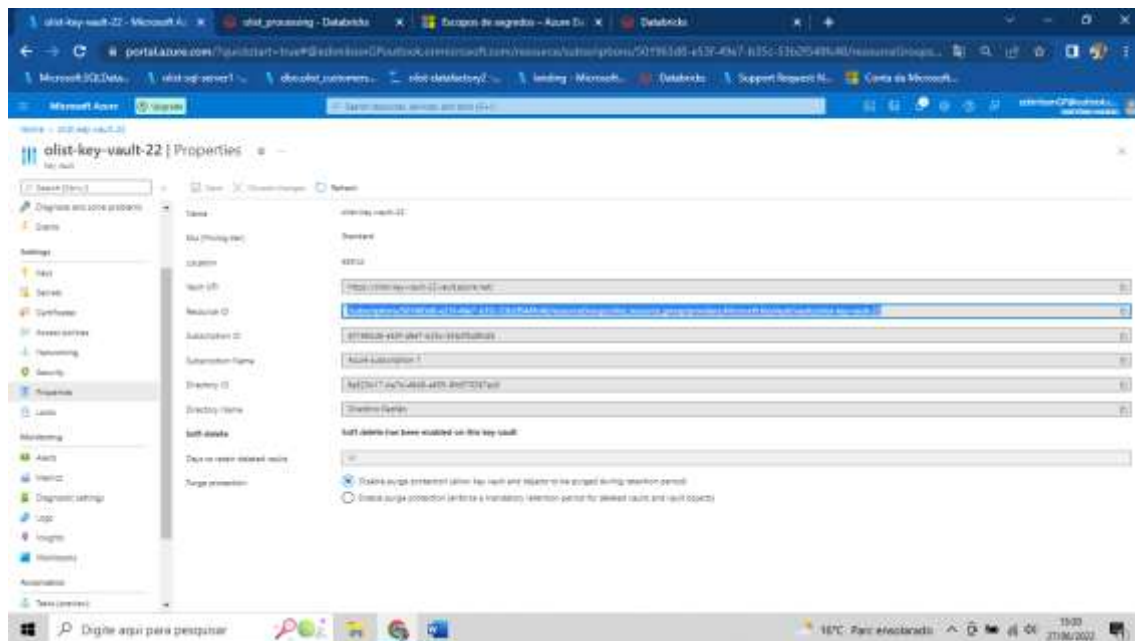
A proteção de software é a opção recomendada para a maioria dos casos. Ela oferece a melhor proteção para os dados armazenados no cofre de chaves e os segredos armazenados no cofre de chaves.

Selecione um período de retenção obrigatório e opcional a ser aplicado permanentemente às cópias de backup dos segredos armazenados no período de retenção. Você pode optar por proteção como imagem. Quando a proteção como imagem está habilitada, os segredos são criados em imagens que são armazenados no cofre de chaves.

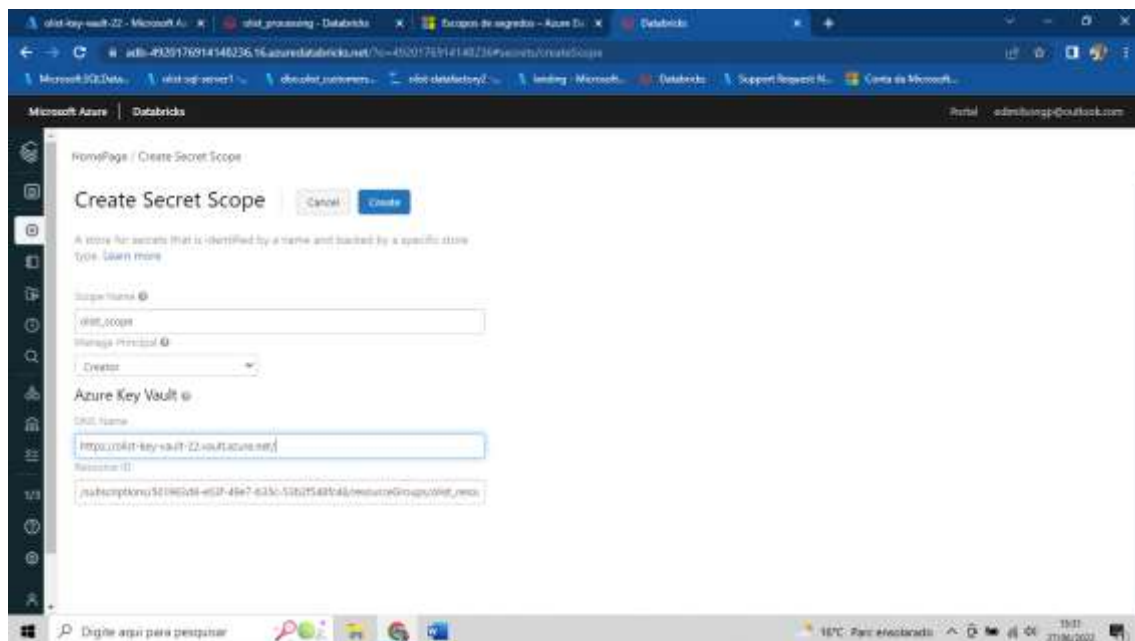
Exatidão esperada:

Quais são as opções de acesso?

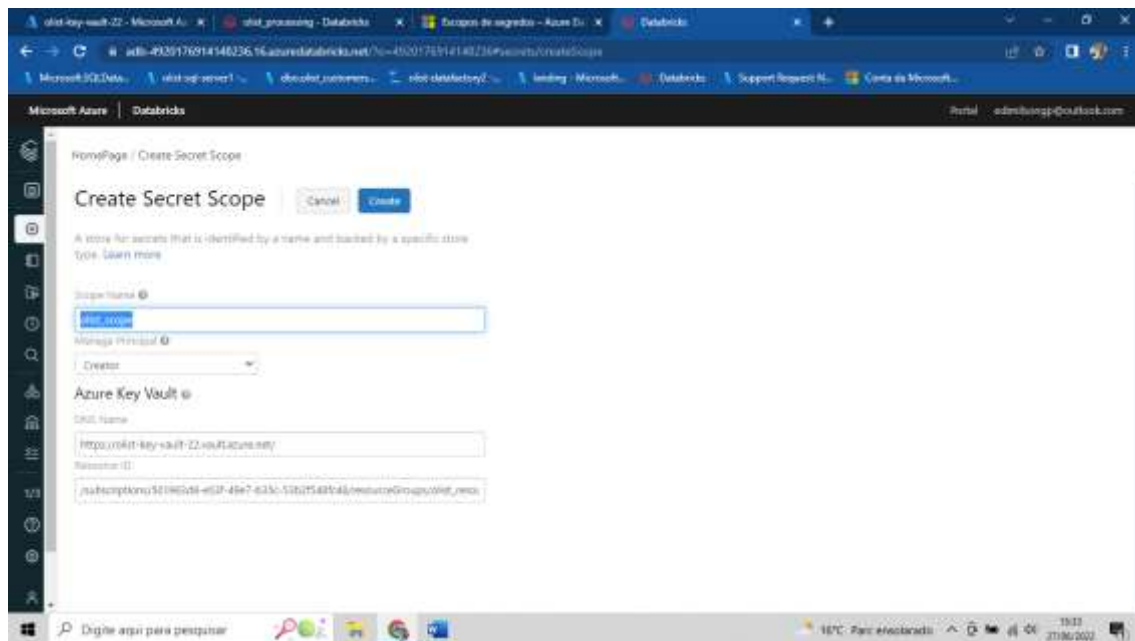
[Retornar à lista](#) [Criar](#) [Cancelar](#) [Proteger: Habilitar o acesso](#)



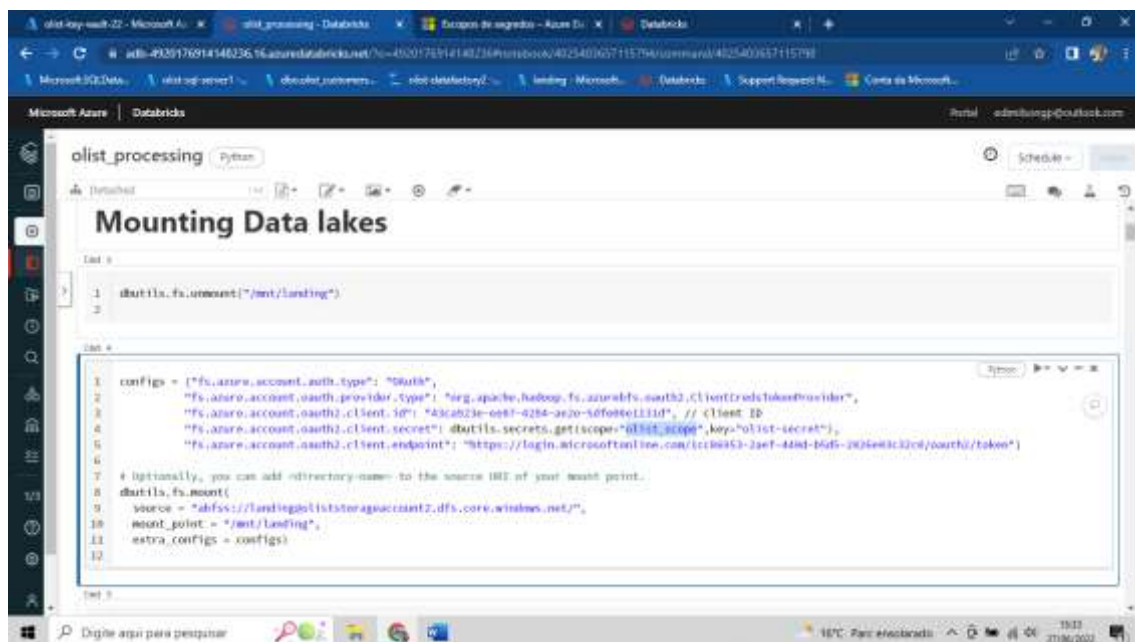
PEGANDO ESSE LINKS E COLOCANDO LÁ NA CRIAÇÃO DE SCOPO



COPIAR O NOME E CLICAR EM CRIAR

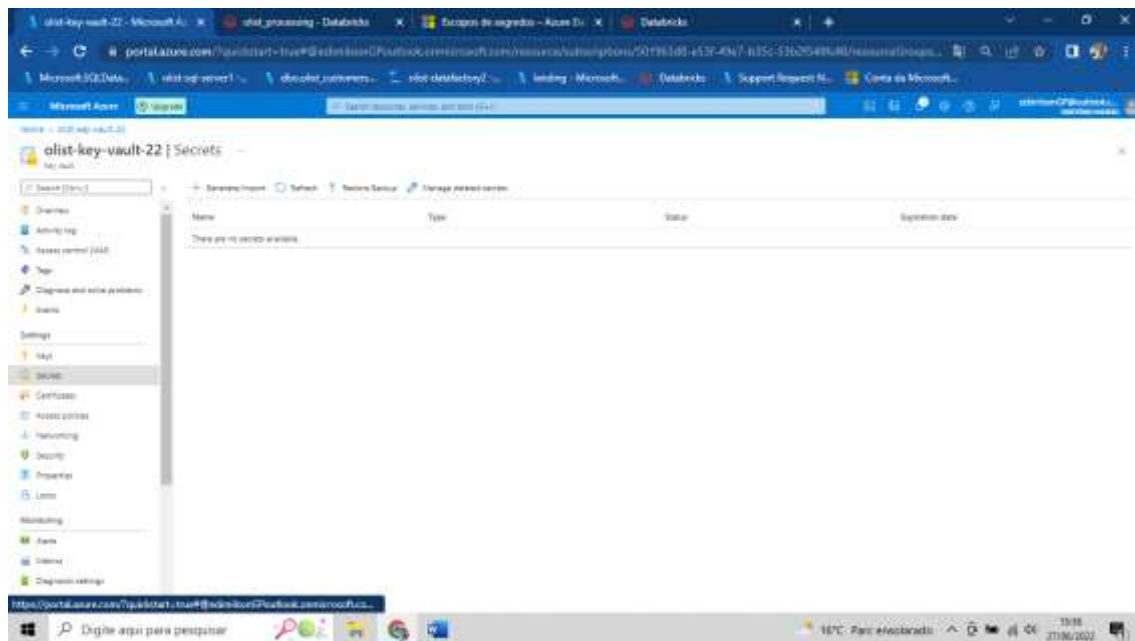


COLAR NO SCRIPT

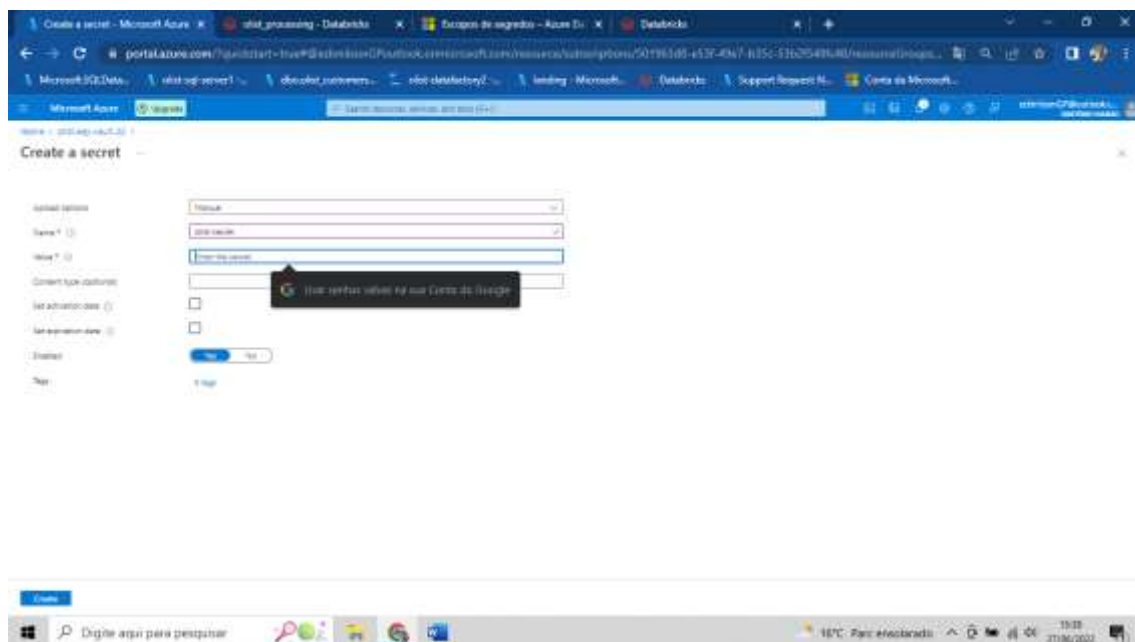


ALTERANDO PARA ALL USERS

CRIADO



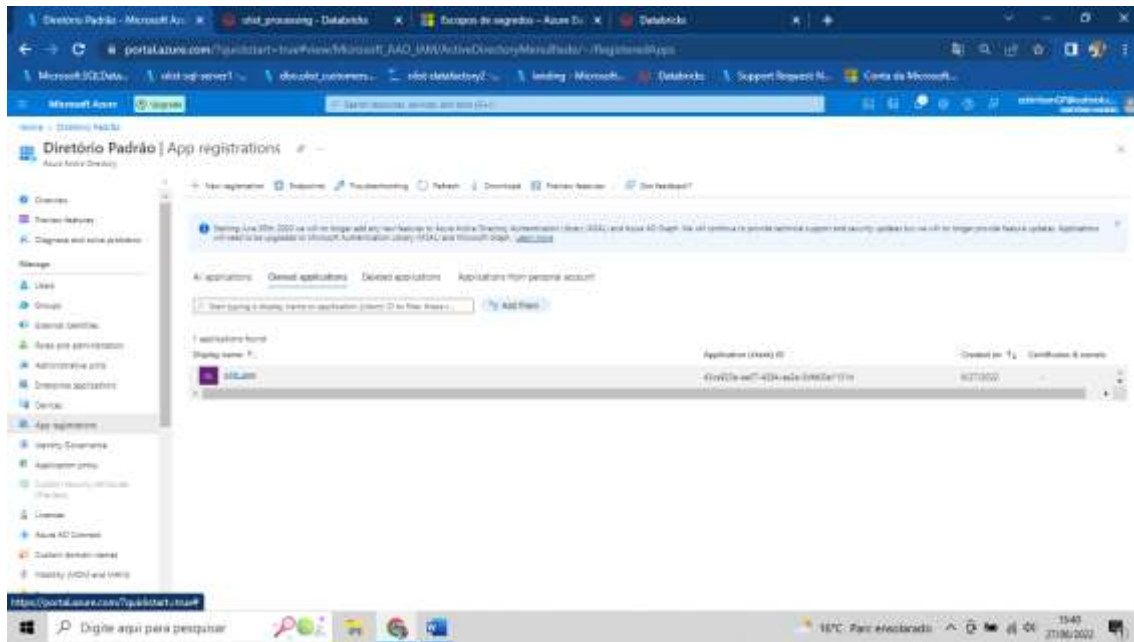
Criando o SECREDO



IR PARA AZURE ACTIVE

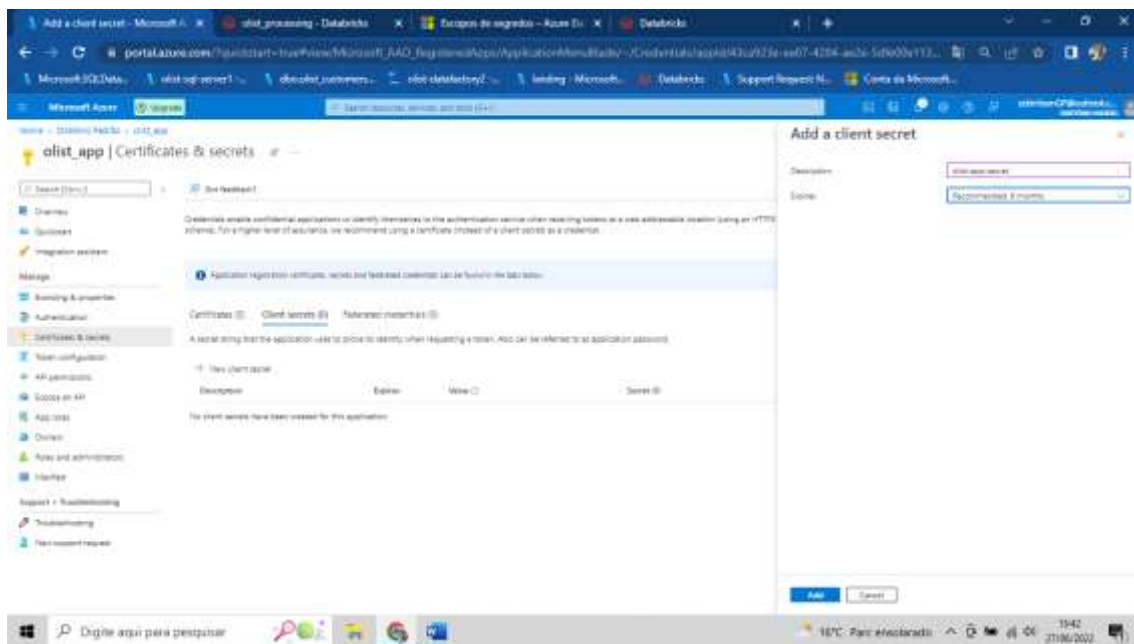
IR EM REGISTRO DE APLICATIVO

CLICAR NO APLICATIVO

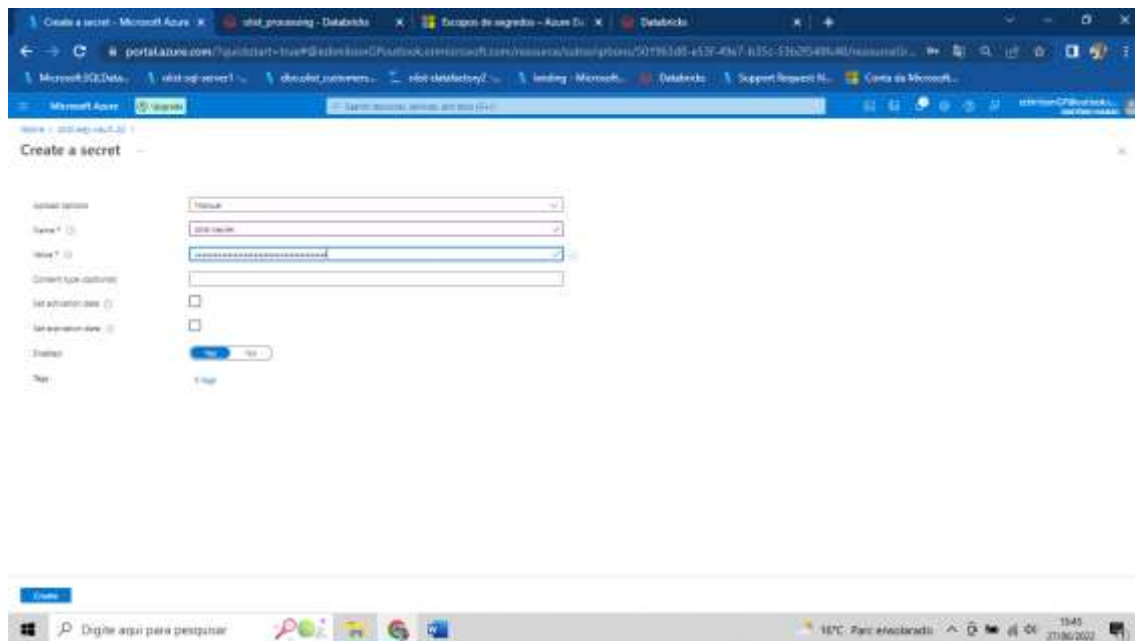


Clicar em novo segredo do cliente e preencher as informações

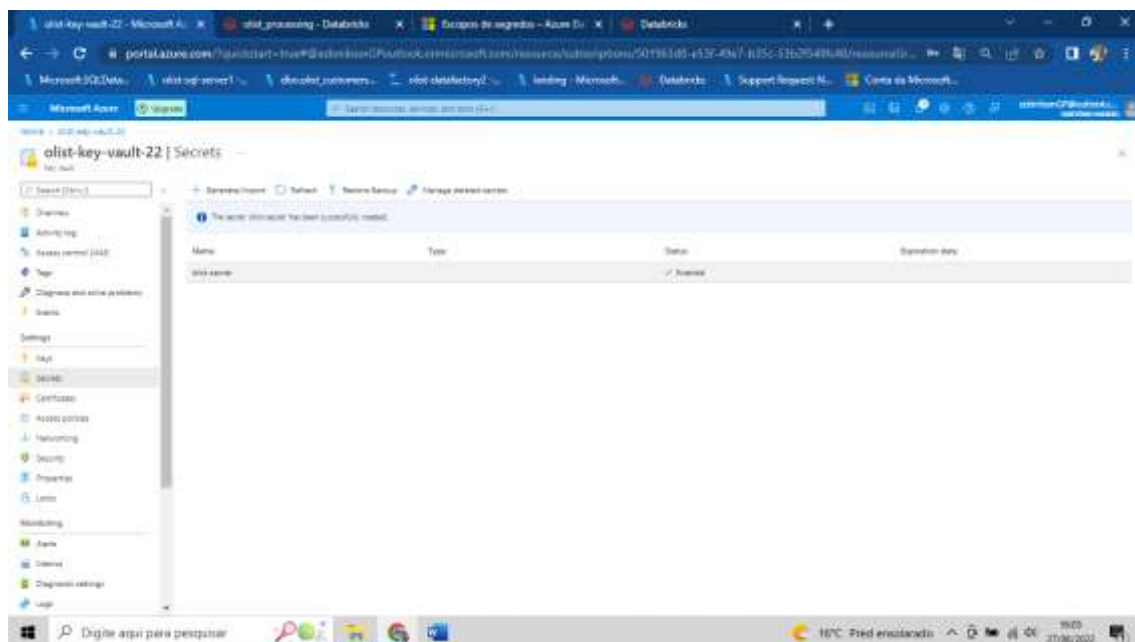
CLICAR EM ADD



SEGREDO CRIADO



CRIADO O OLIST SECRET



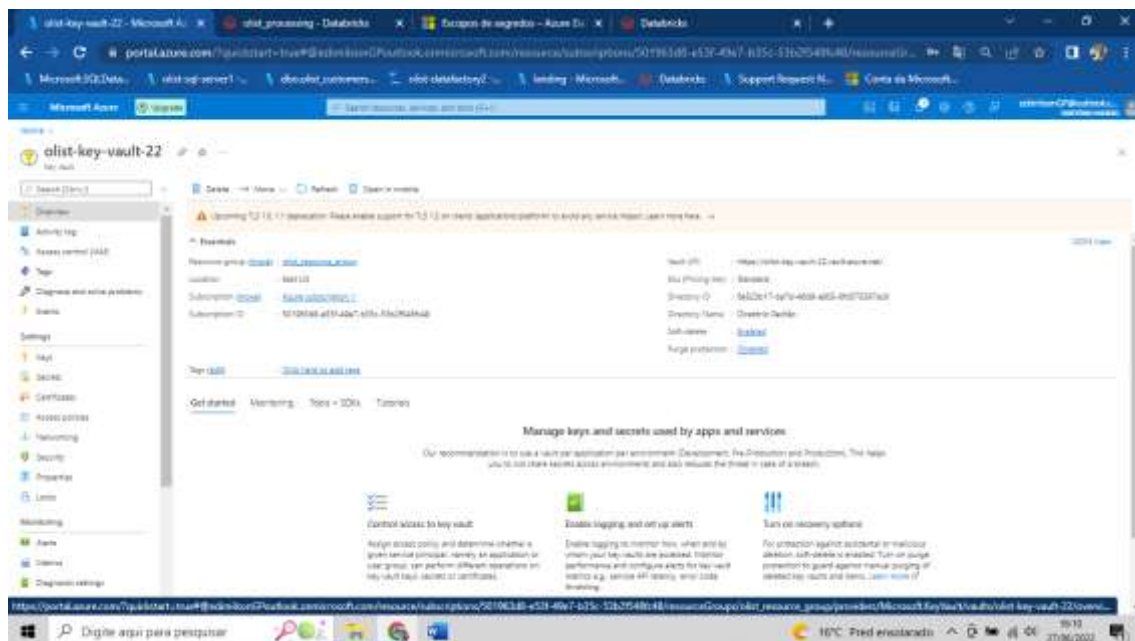
SUBSTITUIR


```
1 dutils.fs.unmount("/mnt/landing")
2
3
4
5
6
7
8
9
10
11
12
```

```
1 configs = [{"fs.azure.account.auth.type": "OAuth",
2             "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3             "fs.azure.account.oauth2.client.id": "43cab23e-9e7e-4254-9e7e-1d6f0b61131d", // client ID
4             "fs.azure.account.oauth2.client.secret": dutils.secrets.get(scope="olist_scope",key="olist-secret"), // scope + segredo
5             "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/62b86133-2aef-428d-b5d5-936e683229/oauth2/token"}] // directory ID
6
7 # optionally, you can add <directory-name> to the source URI of your mount point.
8
9 dutils.fs.mount(
10     source = "abfss://landing@oliststorageaccount2.dfs.core.windows.net/",
11     mount_point = "/mnt/landing",
12     extra_configs = configs)
```

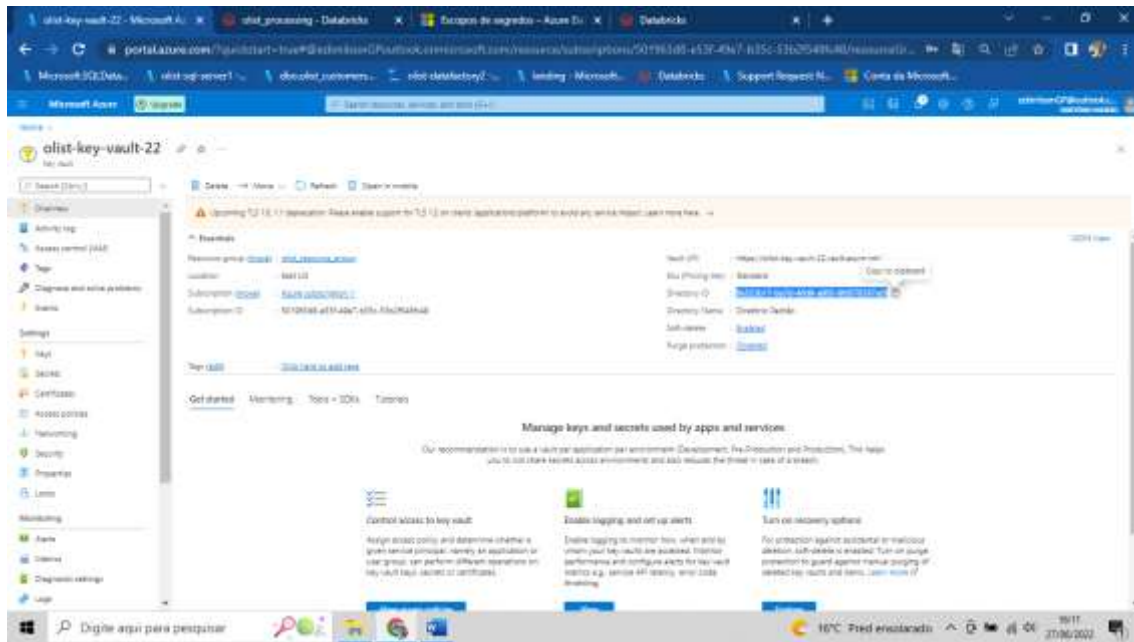
PARA PEGAR ESSAS INFORMAÇÕES DEVEMOS IR EM KEY VAULT

VISÃO GERAL

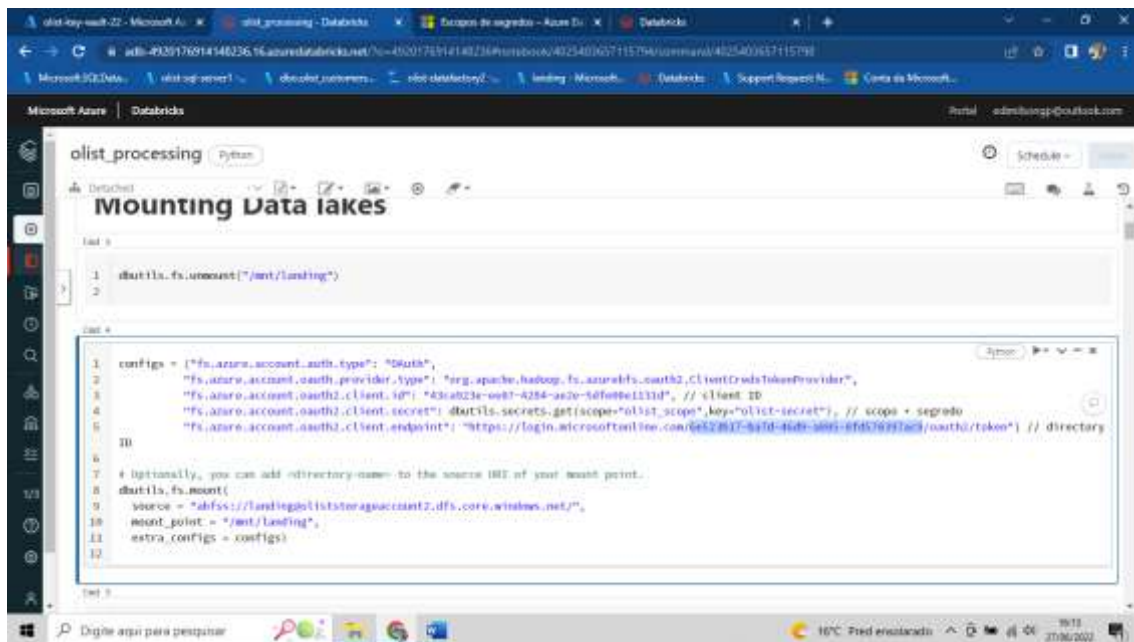


ID DO DIRETORIO

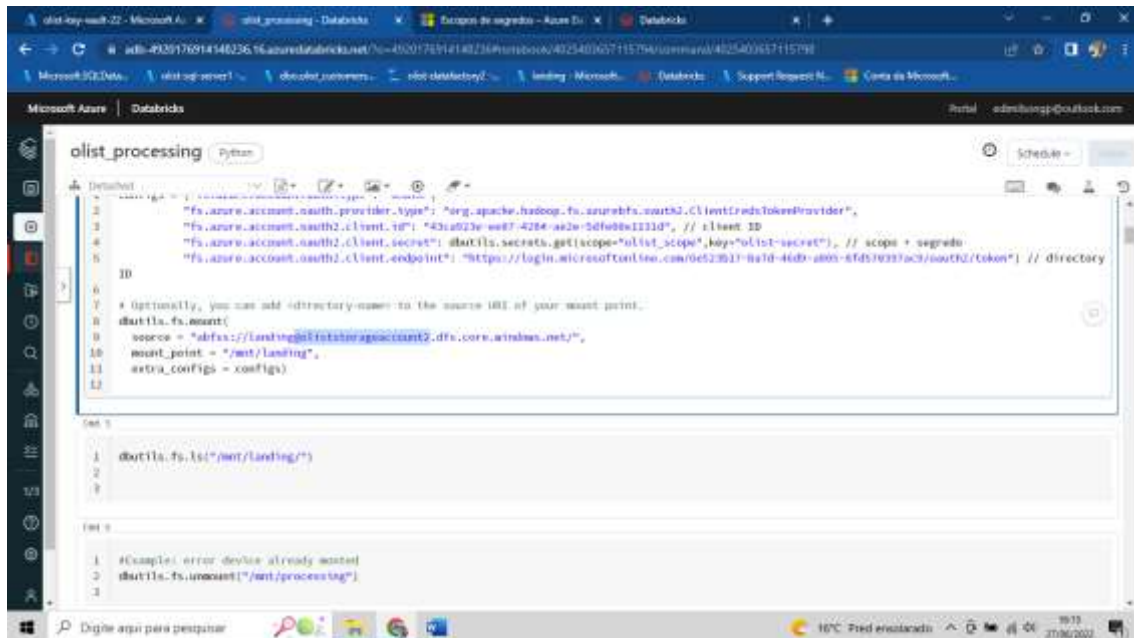
6e523b17-ba7d-46d9-a805-8fd570397ac9



SUBSTITUINDO



AGORA VAMOS ALTERAR A URL DA STORAGE ACCOUNT



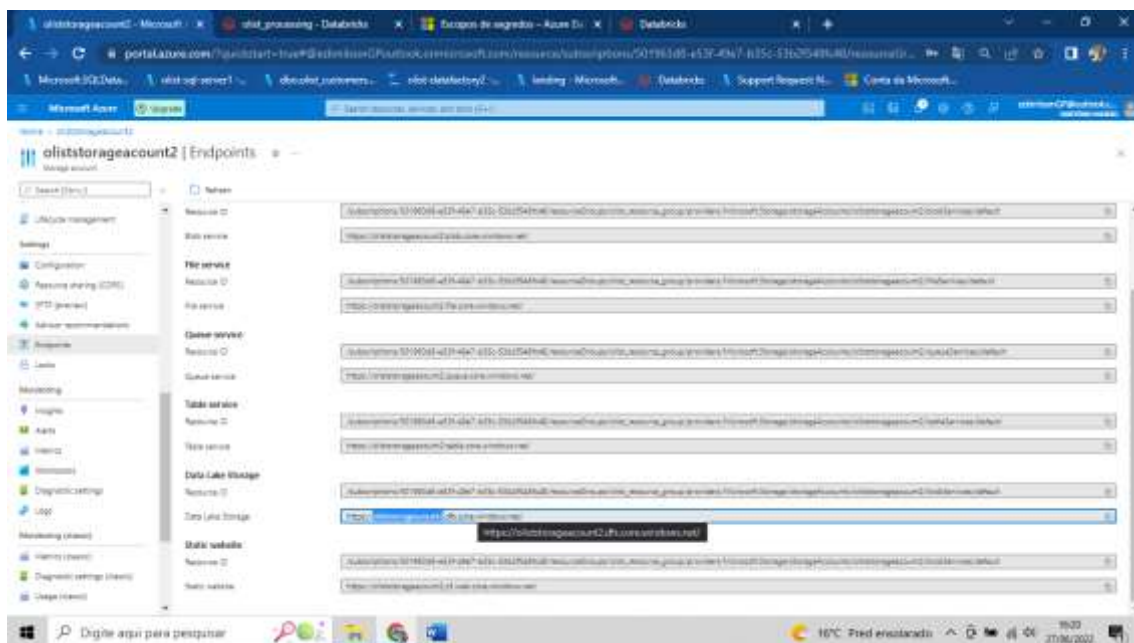
The screenshot shows a Databricks notebook titled 'olist_processing' with a Python language selector. The code defines Azure credentials and mounts a storage account. Below the code, there are two console output sections. The first shows the successful mounting of the storage account at the path '/mnt/landing/'. The second shows an error message: 'Exception: error device already mounted' when attempting to mount the same storage account at a different path, '/mnt/processing/'.

```
1 # Detailed
2 "fs.azure.account.auth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3 "fs.azure.account.auth2.client.id": "43ca023e-ee87-4284-aa2e-5d608a1331d9", // client ID
4 "fs.azure.account.auth2.client.secret": dbutils.secrets.get(scope="olist_scope", key="olist-secret"), // scope + segredo
5 "fs.azure.account.auth2.client.endpoint": "https://login.microsoftonline.com/6e529b17-ba1d-46db-a86b-6fd57937ac37/oauth2/token" // directory
6 ID
7 # Optionally, you can add <directory-name> to the source URI of your mount point.
8 dbutils.fs.mount(
9   source = "abfss://landing@oliststorageaccount2.dfs.core.windows.net/",
10  mount_point = "/mnt/landing",
11  extra_configs = xonfigs)
12
13 Out 1
14
15 dbutils.fs.ls("/mnt/landing/")
16
17 Out 2
18
19 #Example: error device already mounted
20 dbutils.fs.unmount("/mnt/processing/")
21
```

IR PARA PAGINA INICIAL

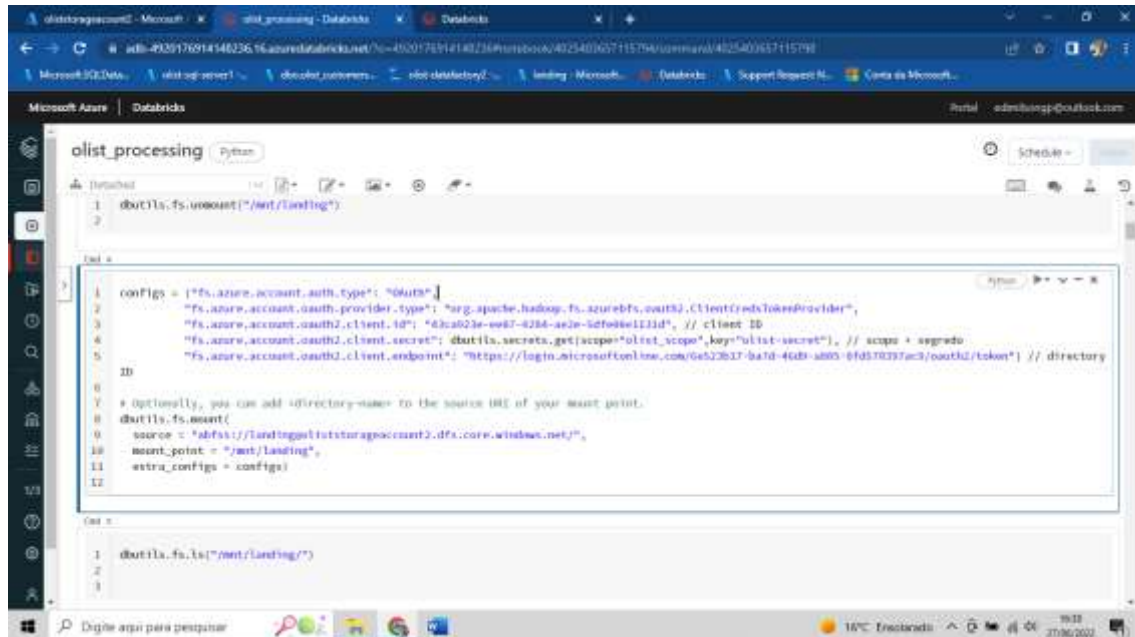
E EM STORAGE ACCOUNT

COPIAR O CODIGO oliststorageaccount2



COLAR

E PRONTO MONTAGEM FINALIZADA



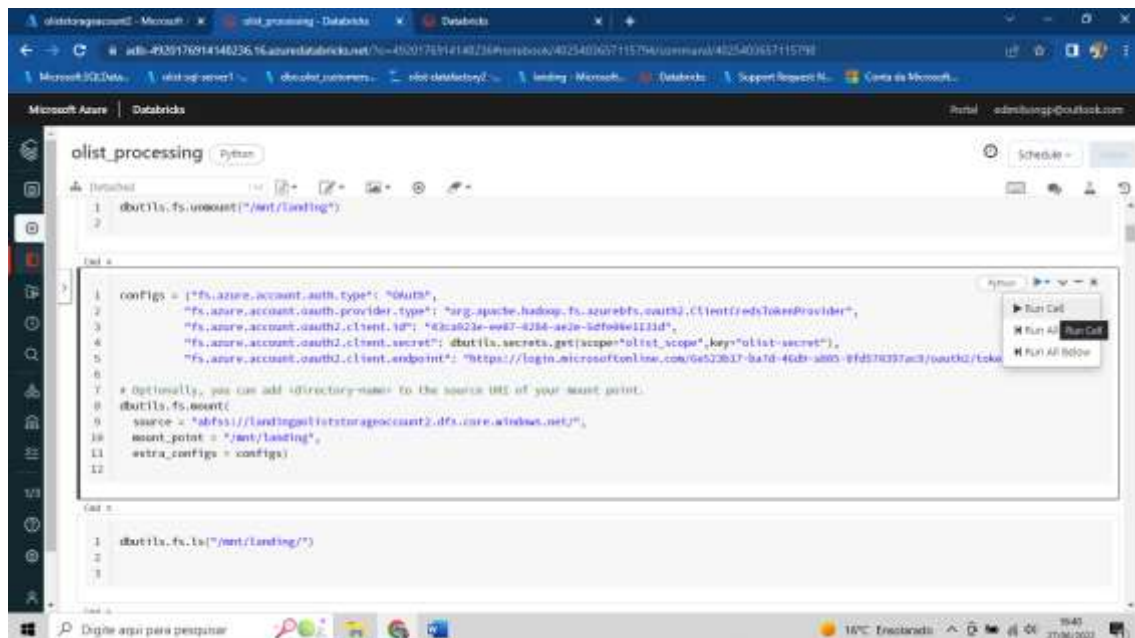
```
1 dbutils.fs.mount("/mnt/landing")
2

def mount:
3     configs = [{"fs.azure.account.auth.type": "OAuth",
4                 "fs.azure.account.auth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
5                 "fs.azure.account.oauth2.client.id": "43ca023e-ee87-4284-ae2e-5ef09e1131d", // client ID
6                 "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="olist_scope", key="olist-secret"), // scope + segredo
7                 "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/6e52b37-ba7d-46db-a805-9f4578397ac3/oauth2/token"} // directory
8     ID
9     # Optionally, you can add <directory-name> to the source URI of your mount point.
10    dbutils.fs.mount(
11        source = "abfs://landingoliststorageaccount2.dfs.core.windows.net/",
12        mount_point = "/mnt/landing",
13        extra_configs = configs)

14
15 dbutils.fs.ls("/mnt/landing/")
16
17
```

EXECUTAR O SCRIPT

CLICANDO EM RUN CELL

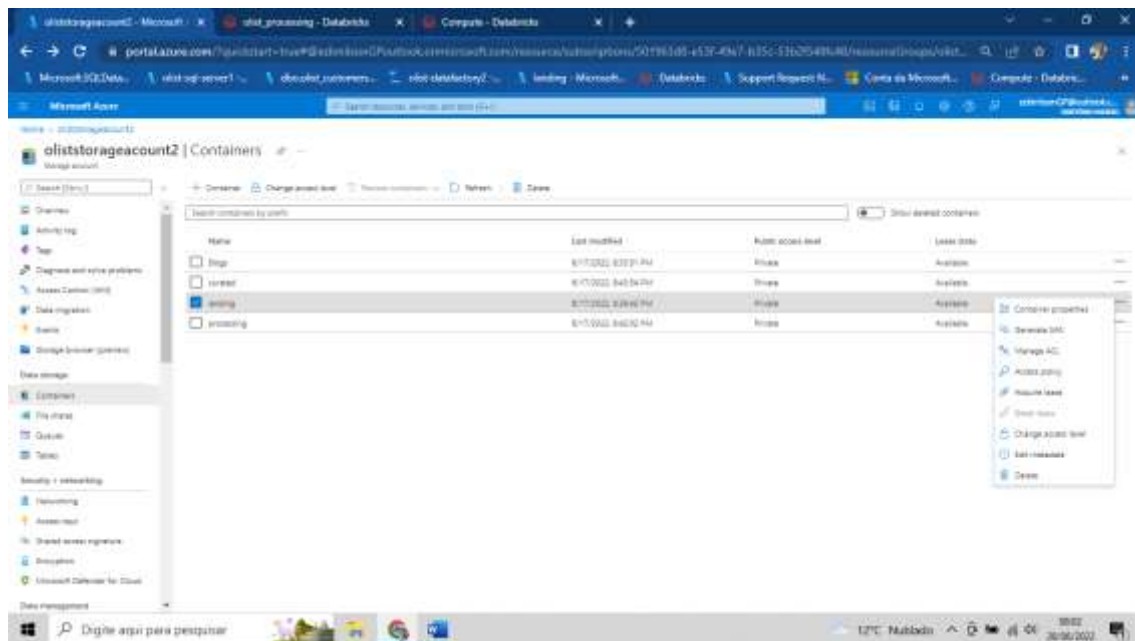


```
1 dbutils.fs.mount("/mnt/landing")
2

def mount:
3     configs = [{"fs.azure.account.auth.type": "OAuth",
4                 "fs.azure.account.auth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
5                 "fs.azure.account.oauth2.client.id": "43ca023e-ee87-4284-ae2e-5ef09e1131d", // client ID
6                 "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="olist_scope", key="olist-secret"), // scope + segredo
7                 "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/6e52b37-ba7d-46db-a805-9f4578397ac3/oauth2/token"} // directory
8     ID
9     # Optionally, you can add <directory-name> to the source URI of your mount point.
10    dbutils.fs.mount(
11        source = "abfs://landingoliststorageaccount2.dfs.core.windows.net/",
12        mount_point = "/mnt/landing",
13        extra_configs = configs)

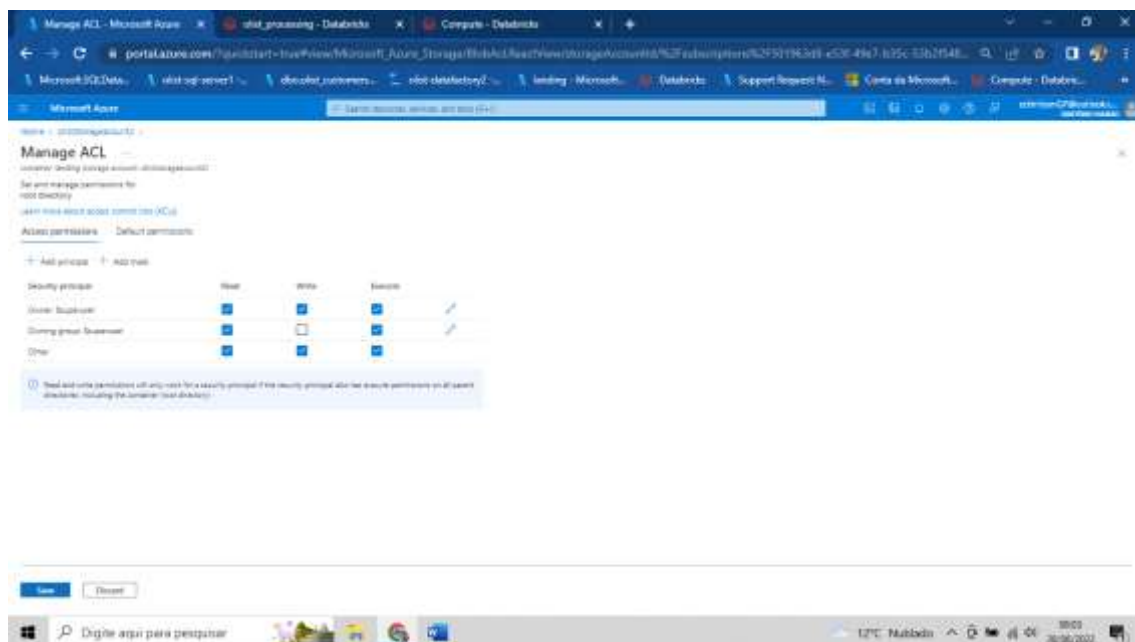
14
15 dbutils.fs.ls("/mnt/landing/")
16
17
```

EM EXECUÇÃO



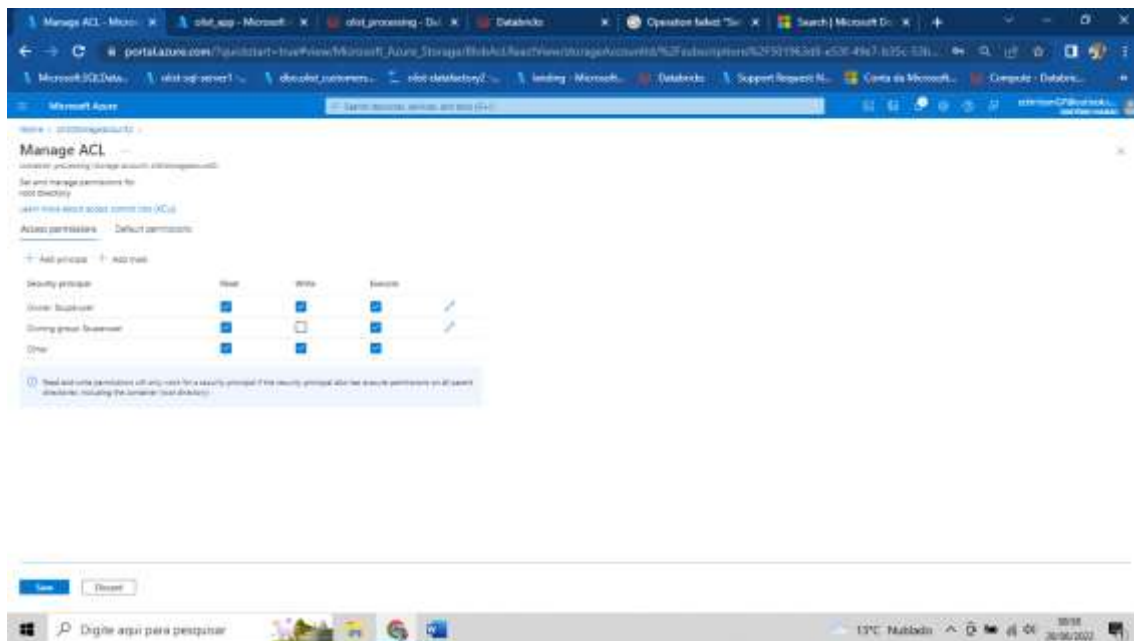
PERMISSÕESATIVADAS PARA OUTROS USUARIOS

SALVAR

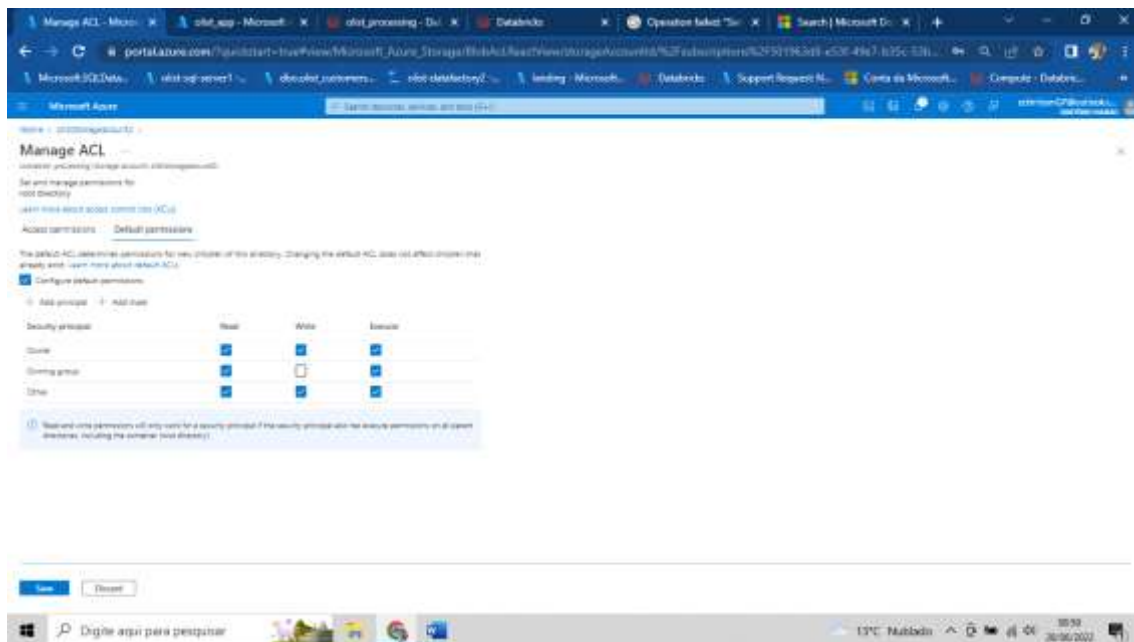


Agora fazer o mesmo processo de permissão para os demais containers

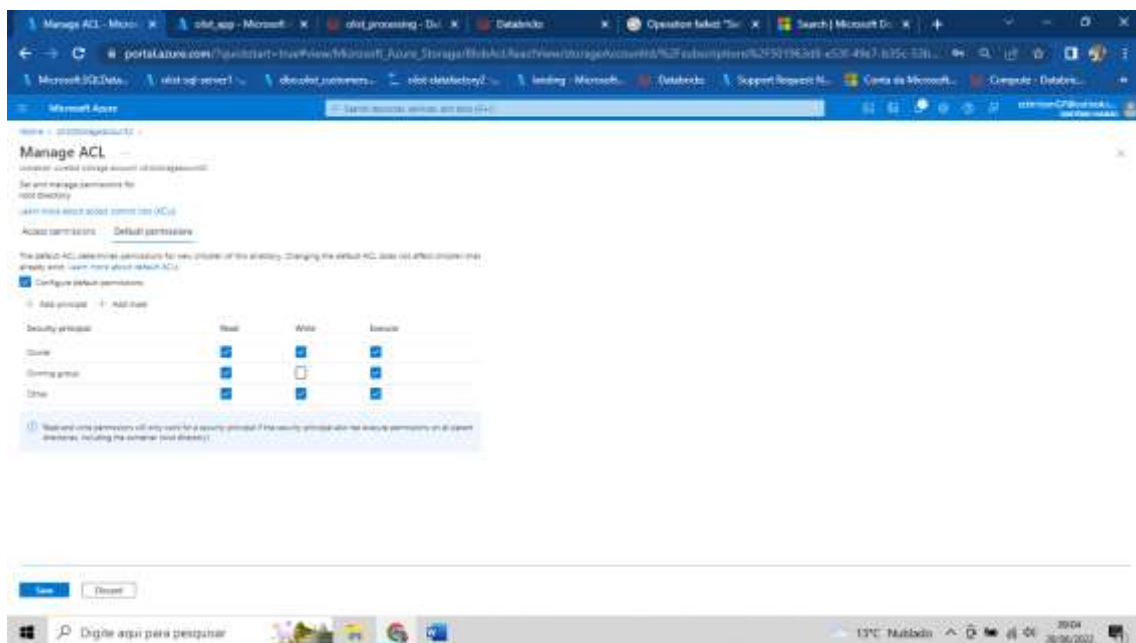
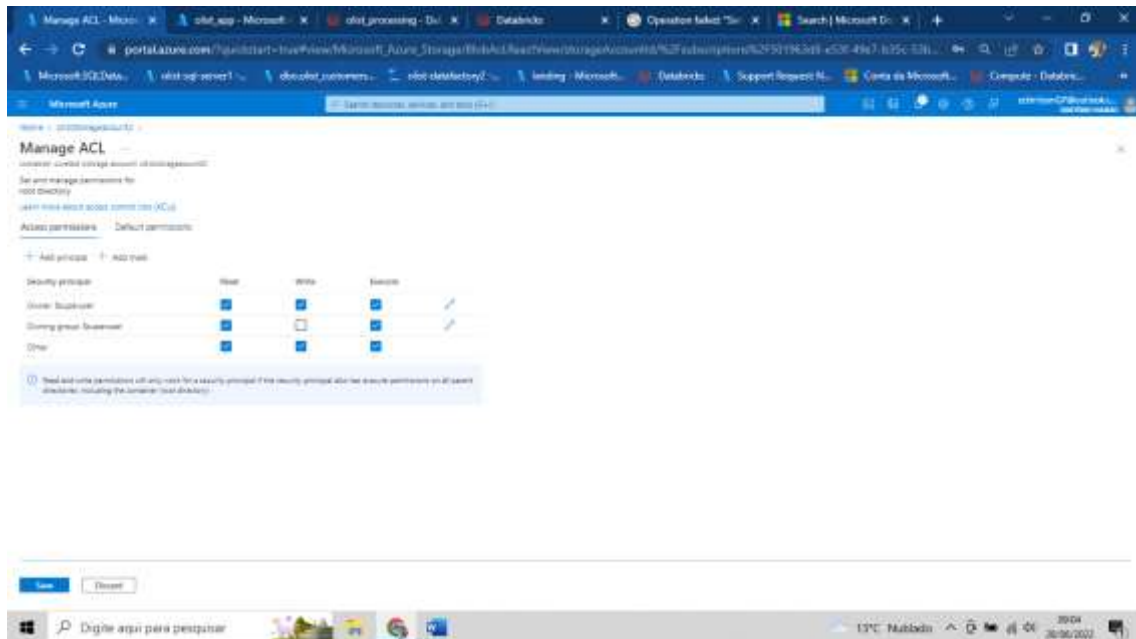
PROCESSING



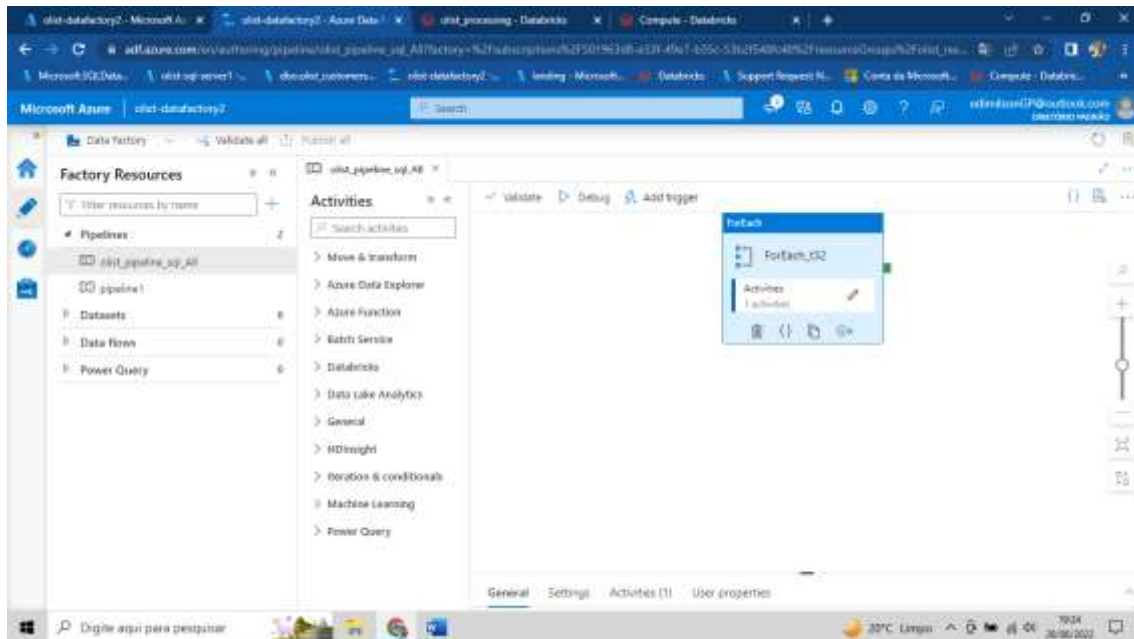
Também em permissão padrão



CURATED



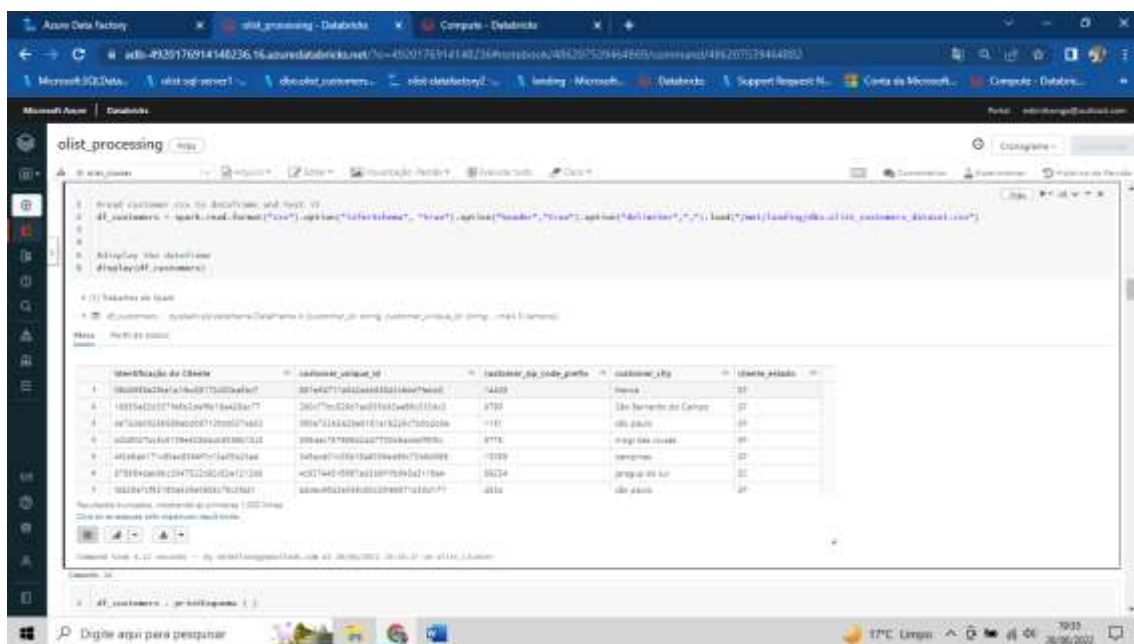
AGORA SÓ RODAR O NOTEBOOK EXECUTANDO UM POR UM,
CASO DE ERRO RODAR O NOTEBOOK DO INICIO



ORIGEM

E DISPARAR GATILHO

EXECUTANDO A LEITURA



EXUTANDO AS OUTRAS CELULAS

Azure Data Factory | olist_processing - Databricks | Compute - Databricks

Microsoft SQL Data... | olist-og-repo1 | dms-ol-repo1 | nls-databricks2 | landing - Microsoft... | Databricks | Support Request N... | Conta da Microsoft... | Compute - Databr...

Microsoft Azure | Databricks

olist_processing | Autor

10:00 AM

Create SQL Database

10:00 AM

```
1 --
2 CREATE DATABASE IF NOT EXISTS customers_db
3
4 --
```

Command took 0.14 seconds -- by olist@onguestbook-001 at 20/06/2022 10:00:00 on olist_cluster

Create SQL Tables

10:00 AM

```
1 --
2 DROP TABLE IF EXISTS customers_db.customers
3
4 --
5 -- customers table
6 --
7 -- order_payments
8 --
9 -- orders
10 --
11 -- address
12 --
13 -- products_category_name_merged
14
15 --
```

Command took 0.17 seconds -- by olist@onguestbook-001 at 20/06/2022 10:00:00 on olist_cluster

17°C Limão

7943 26/06/2022

Azure Data Factory | olist_processing - Databricks | Compute - Databricks

Microsoft SQL Data... | olist-og-repo1 | dms-ol-repo1 | nls-databricks2 | landing - Microsoft... | Databricks | Support Request N... | Conta da Microsoft... | Compute - Databr...

Microsoft Azure | Databricks

olist_processing | Autor

10:00 AM

10:00 AM

```
1 --
2 --
3 --
4 DROP TABLE IF EXISTS customers_db.products
5
6 --
```

Command took 0.14 seconds -- by olist@onguestbook-001 at 20/06/2022 10:00:00 on olist_cluster

10:00 AM

```
1 --
2 --
3 --
4 DROP TABLE IF EXISTS customers_db.order_items
5
6 --
```

Command took 0.14 seconds -- by olist@onguestbook-001 at 20/06/2022 10:00:00 on olist_cluster

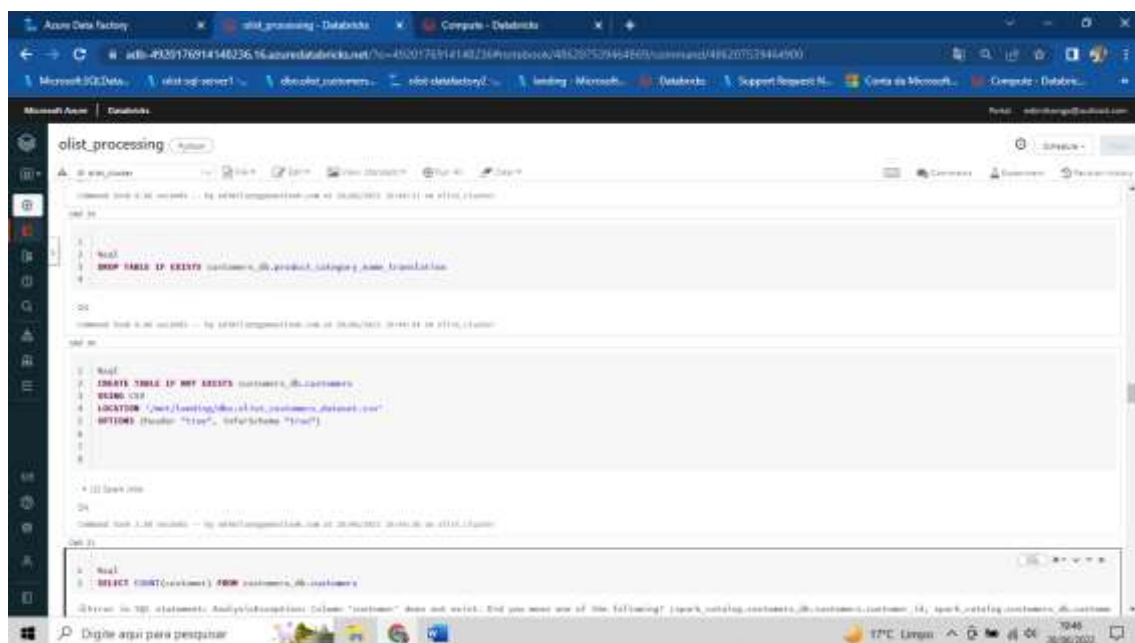
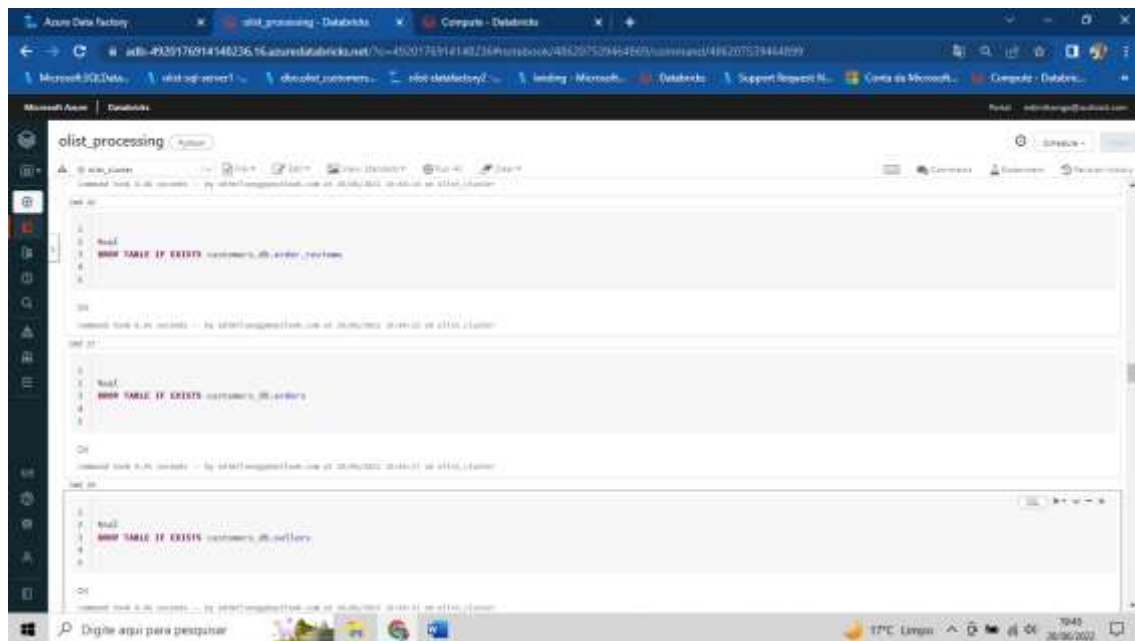
10:00 AM

```
1 --
2 --
3 --
4 DROP TABLE IF EXISTS customers_db.order_payments
5
6 --
```

Command took 0.14 seconds -- by olist@onguestbook-001 at 20/06/2022 10:00:00 on olist_cluster

17°C Limão

7943 26/06/2022



FAZER O MESMO COMANDO PARA AS DEMAIS TABELAS

AGORA VAMOS ACESSAR A TABELA

olist_processing Completed

Job ID: 4f1a1a1a-1a1a-1a1a-1a1a-1a1a1a1a1a1a

Created: 2023-09-26 10:00:00

Completed: 2023-09-26 10:05:00

customer_id	customer_name	customer_email	customer_phone	customer_address	customer_city
1	João da Silva	joao.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
2	Maria da Silva	maria.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
3	Carlos da Silva	carlos.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
4	Ana da Silva	ana.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
5	Pedro da Silva	pedro.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
6	Julia da Silva	julia.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
7	Lucas da Silva	lucas.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
8	Isabella da Silva	isabella.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
9	Gabriel da Silva	gabriel.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo
10	Olivia da Silva	olivia.silva@olista.com	11 9999-9999	Rua da Silva, 123	São Paulo

AGORA VAMOS LER E GRAVAR EM FORMATO PARQUET TODAS AS TABELAS

olist_processing Completed

Job ID: 4f1a1a1a-1a1a-1a1a-1a1a-1a1a1a1a1a1a

Created: 2023-09-26 10:00:00

Completed: 2023-09-26 10:05:00

Write Full Parquet Datasets to Processing Data lake

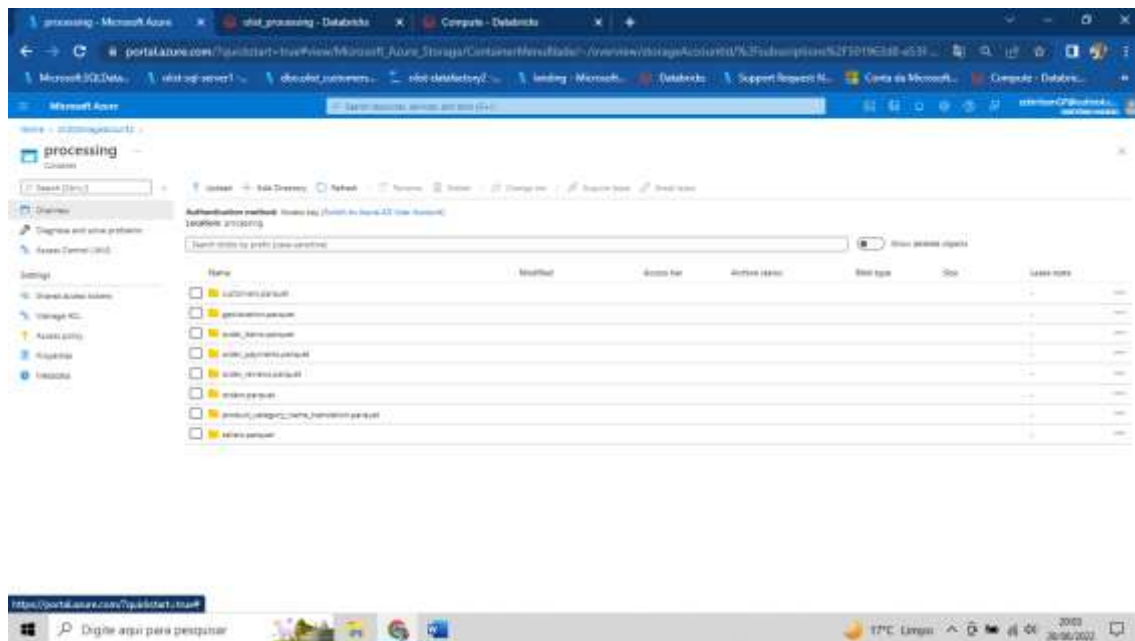
```

1 df.write.mode("overwrite").parquet("/mnt/processing/olist_processing.parquet")
2
3
4 df.write.mode("overwrite").parquet("/mnt/processing/order_items.parquet")
5 df.write.mode("overwrite").parquet("/mnt/processing/order_payments.parquet")
6 df.write.mode("overwrite").parquet("/mnt/processing/order_reviews.parquet")
7 df.write.mode("overwrite").parquet("/mnt/processing/orders.parquet")
8 df.write.mode("overwrite").parquet("/mnt/processing/sellers.parquet")
9 df.write.mode("overwrite").parquet("/mnt/processing/products.parquet")
10 df.write.mode("overwrite").parquet("/mnt/processing/products_category_name.parquet")

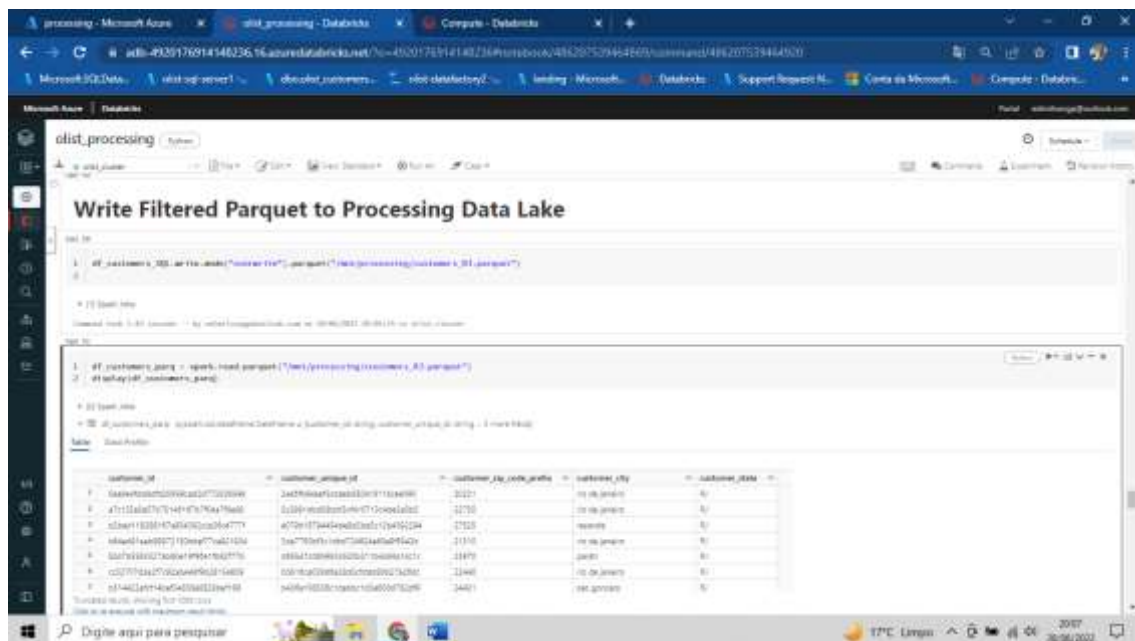
```

Write Filtered Parquet to Processing Data Lake

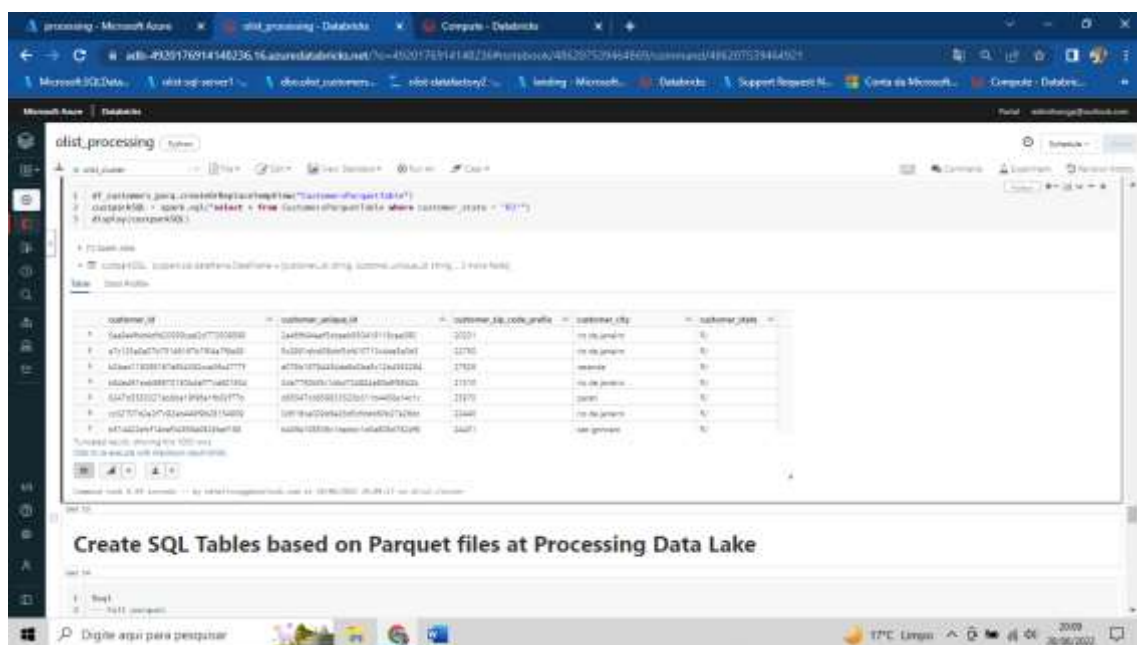
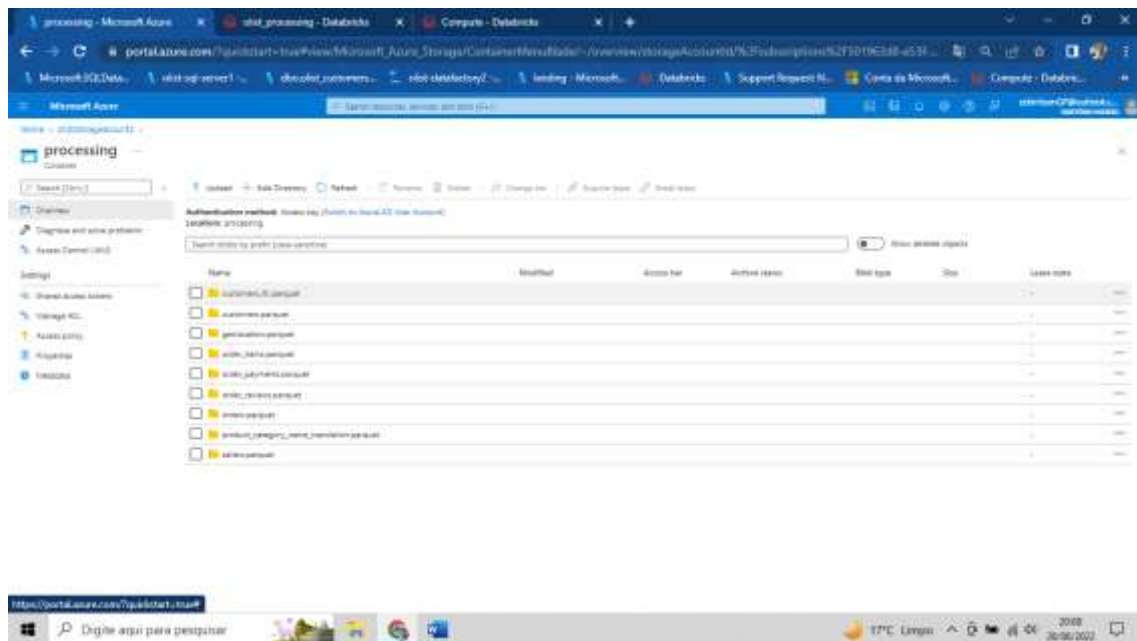
PARA VERIFICAR SE CARREGOU AS TABELAS EM STORAGE ACCOUNT PROCESSING
TABELA CRIADAS COM SUCESSO EM FORMATO PARQUET



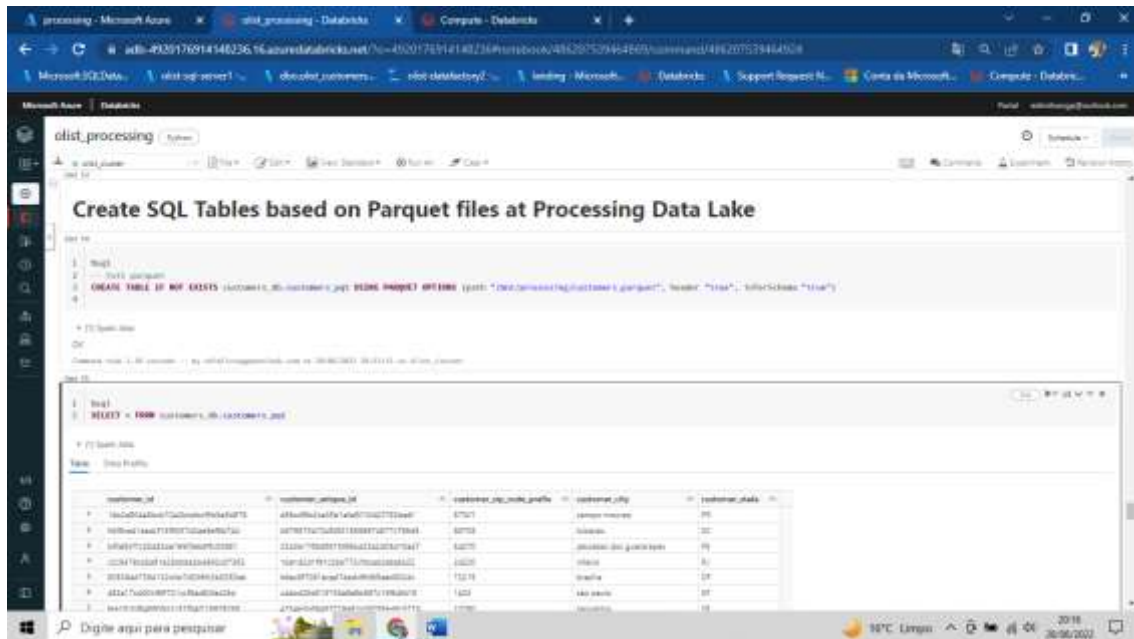
CASO EU QUEIRA SALVAR UM DATAFRAME JÁ FILTRADO
E VERIFICAR SE ESTÁ FUNCIONANDO CORRETAMENTE



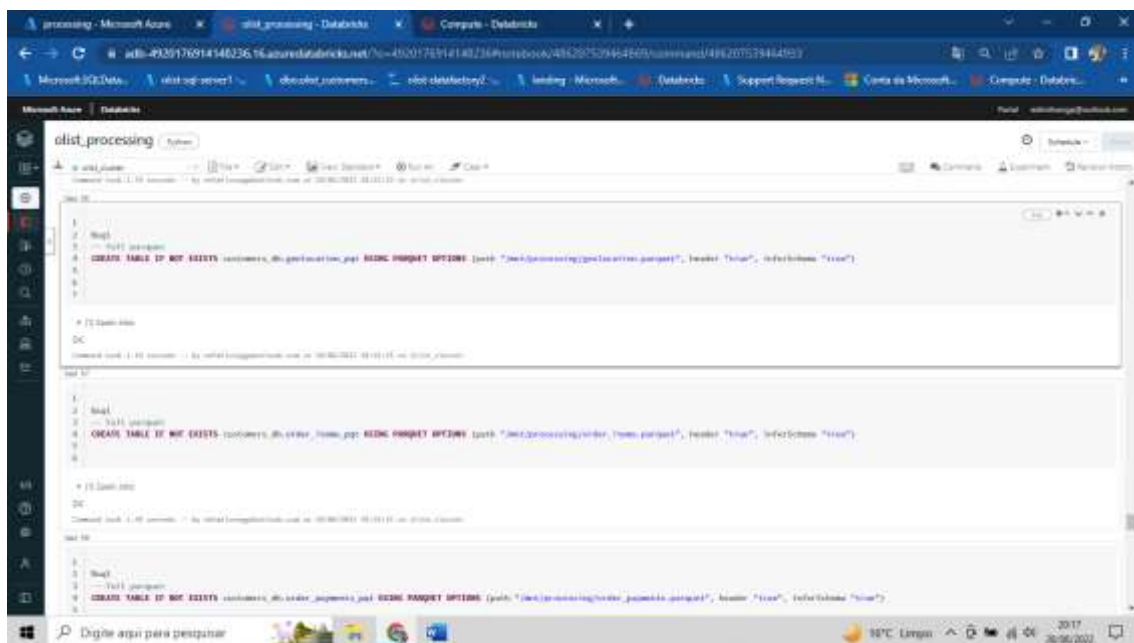
VERIFICANDO SE SALVOU EM PROCESSING
SALVOU COMO RJ.PARQUET O DATAFRAME FILTRADO



AGORA VAMOS CRIAR TABELAS BASEADAS NO PARQUET



FAZER O MESMO PROCESSO PARA AS OUTRAS TABELAS



COMANDO REFRESH VAI BUSCAR OS DADOS MAIS RECENTES

Microsoft Azure | Databricks

olist_processing

Completed: 0.37 seconds

Table: olist_processing

customer_id	customer_segment_id	customer_age_range	customer_city	customer_state
3a3b40e9f09090909090909090909090	3a3b40e9f09090909090909090909090	20-29	rio de janeiro	RJ
4f113a3b40e9f0909090909090909090	4f113a3b40e9f0909090909090909090	30-39	rio de janeiro	RJ
6b3b40e9f09090909090909090909090	6b3b40e9f09090909090909090909090	40-49	rio de janeiro	RJ
8d3b40e9f09090909090909090909090	8d3b40e9f09090909090909090909090	50-59	rio de janeiro	RJ
0e3b40e9f09090909090909090909090	0e3b40e9f09090909090909090909090	60-69	rio de janeiro	RJ
2f3b40e9f09090909090909090909090	2f3b40e9f09090909090909090909090	70-79	rio de janeiro	RJ
5g3b40e9f09090909090909090909090	5g3b40e9f09090909090909090909090	80-89	rio de janeiro	RJ
7h3b40e9f09090909090909090909090	7h3b40e9f09090909090909090909090	90-99	rio de janeiro	RJ

CRIANDO CSV NA CURATED

Microsoft Azure | Databricks

olist_processing

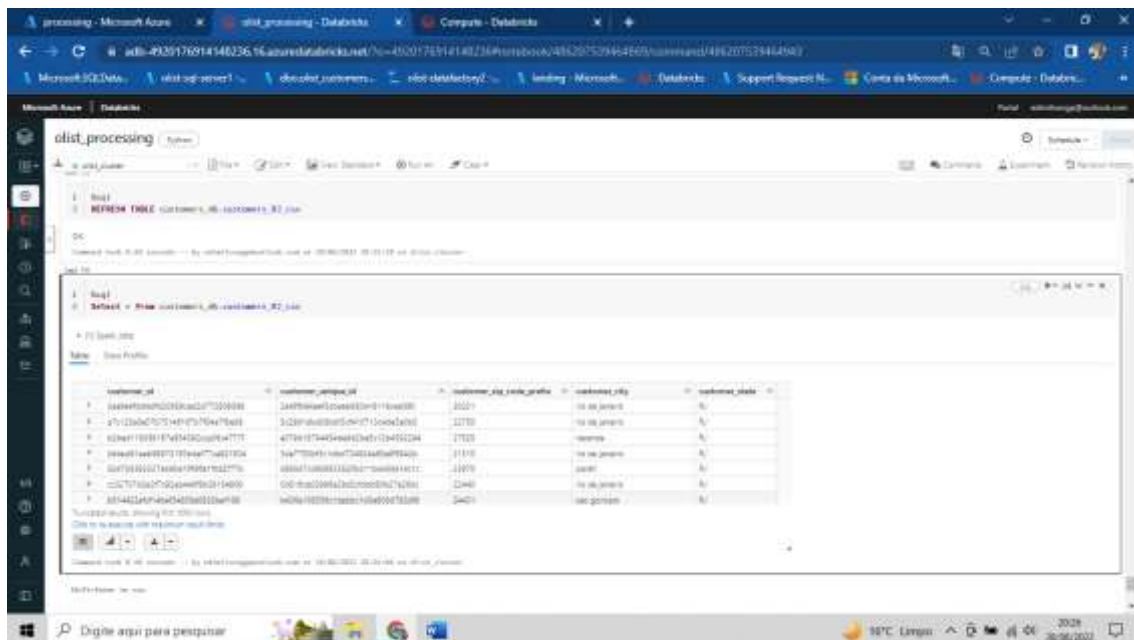
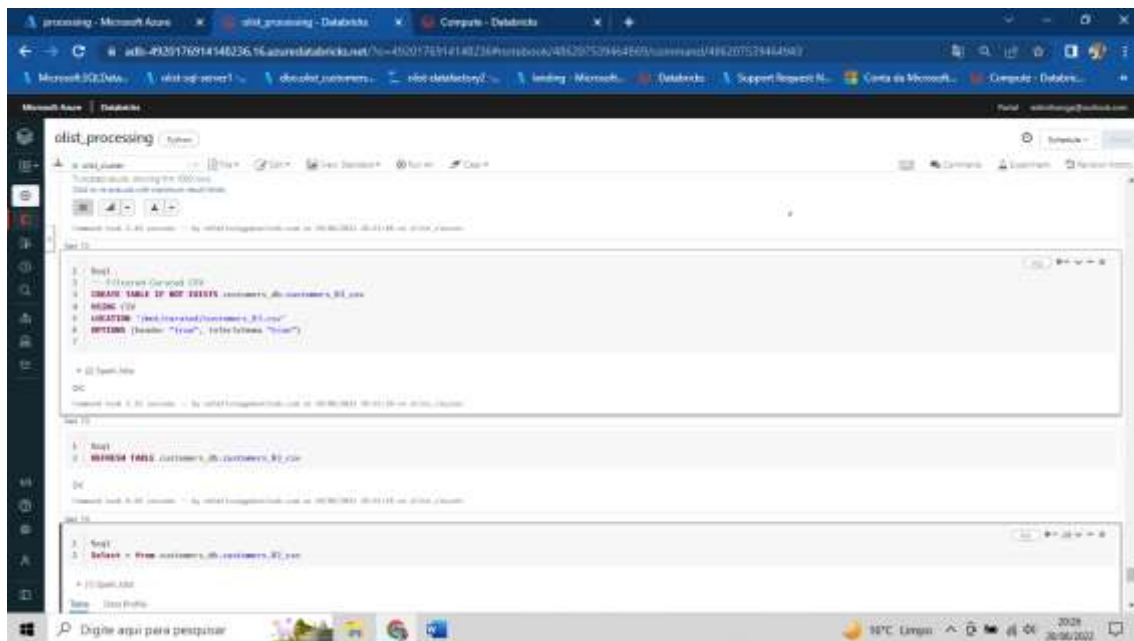
Completed: 0.37 seconds

Table: olist_processing

Write processed CSVs to Curated Data Lake

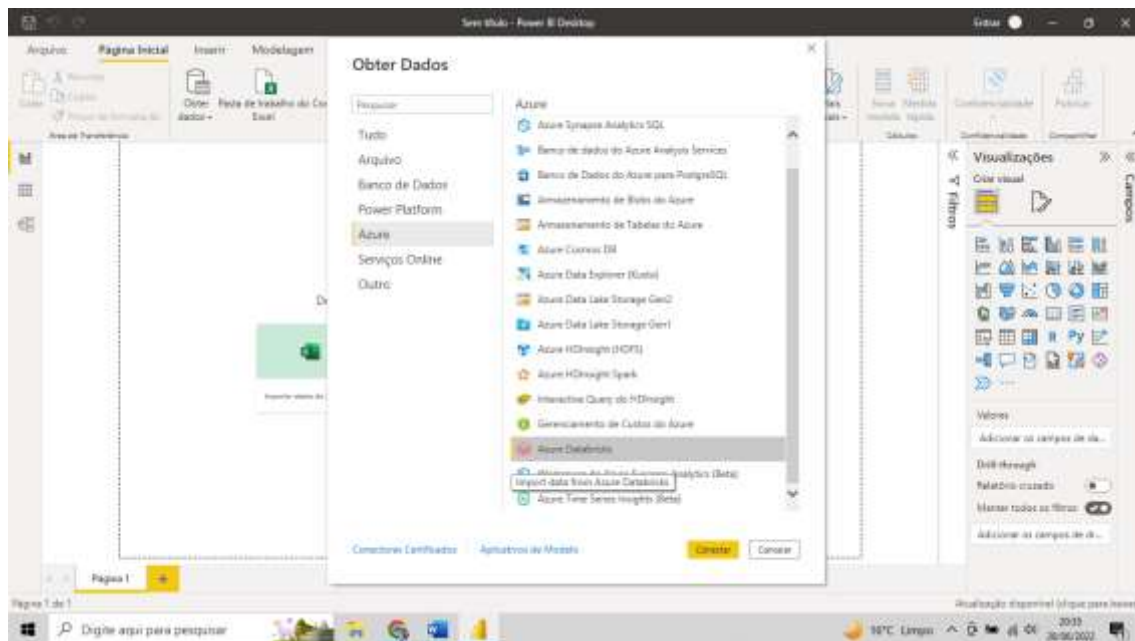
```
df.customer_id, df.customer_segment_id, df.customer_age_range, df.customer_city, df.customer_state
```

VAMOS FAZER O TESTE E LER O ARQUIVO



VERIFICANDO A CRIAÇÃO DO CSV FILTRADO EM CURATED

**ARQUITETURA ESTÁ PRONTA AGORA LEVAR PARA O POWER BI
CONECTANDO DO BANCO DE DADOS DO AZURE DATABRICKS**

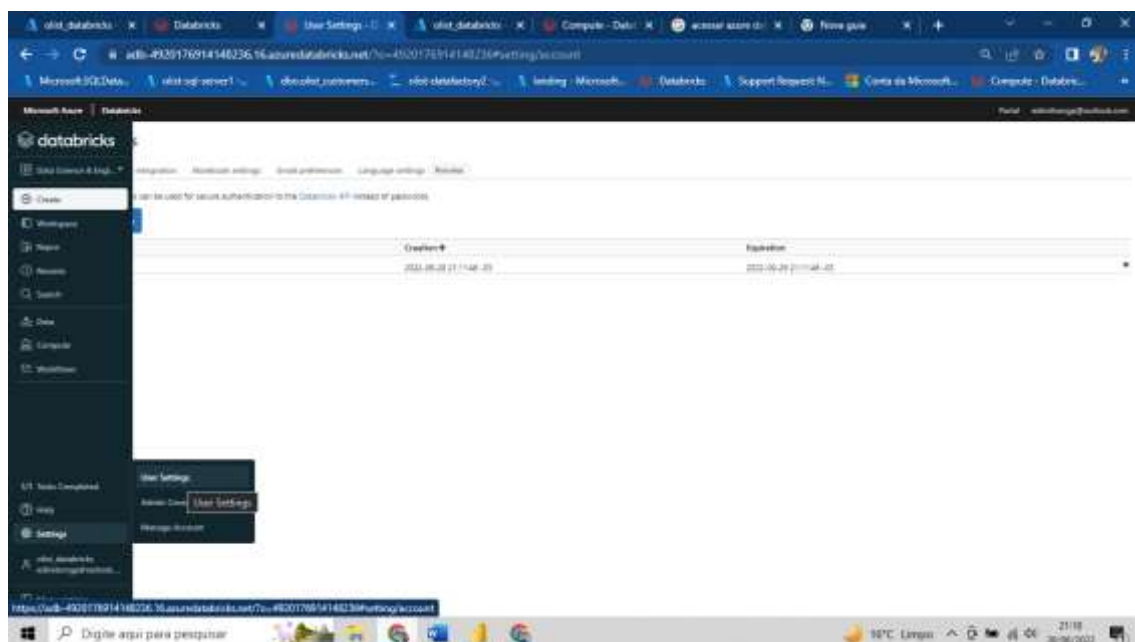


PARA PEGAR ESSAS INFORMAÇÕES ABAIXO

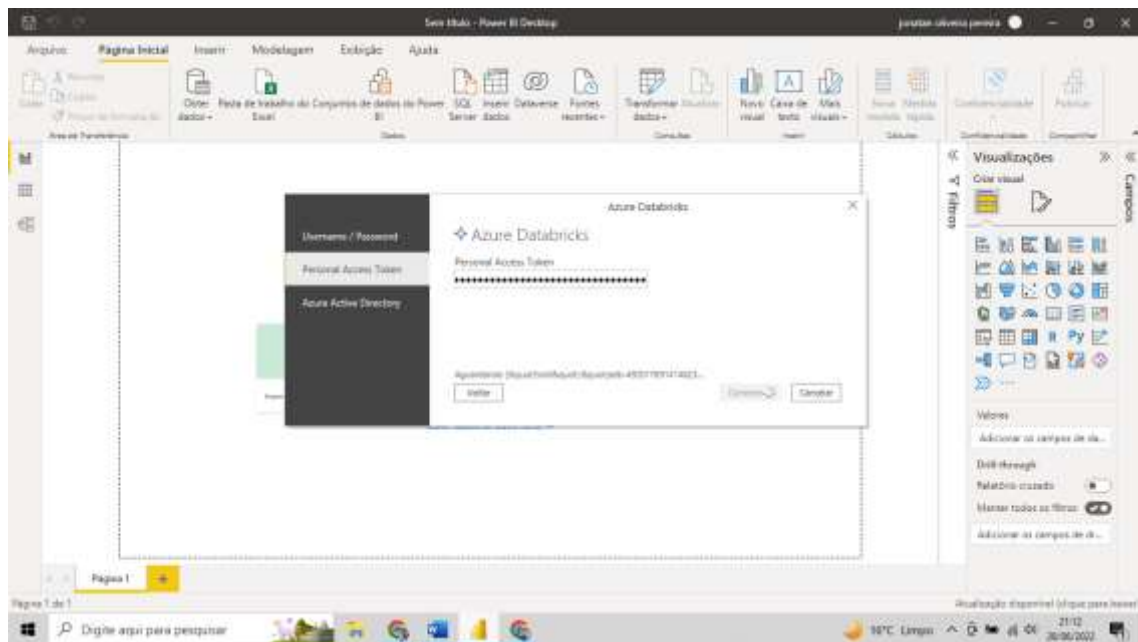
DIRECT QUERY -> PARA PEGAR AS INFORMAÇÕES DO DATA LAKE EM TEMPO REAL

AS INFORMAÇÕES ESTÃO NO CLUSTER DO DATABRICKS

Para gerar o token

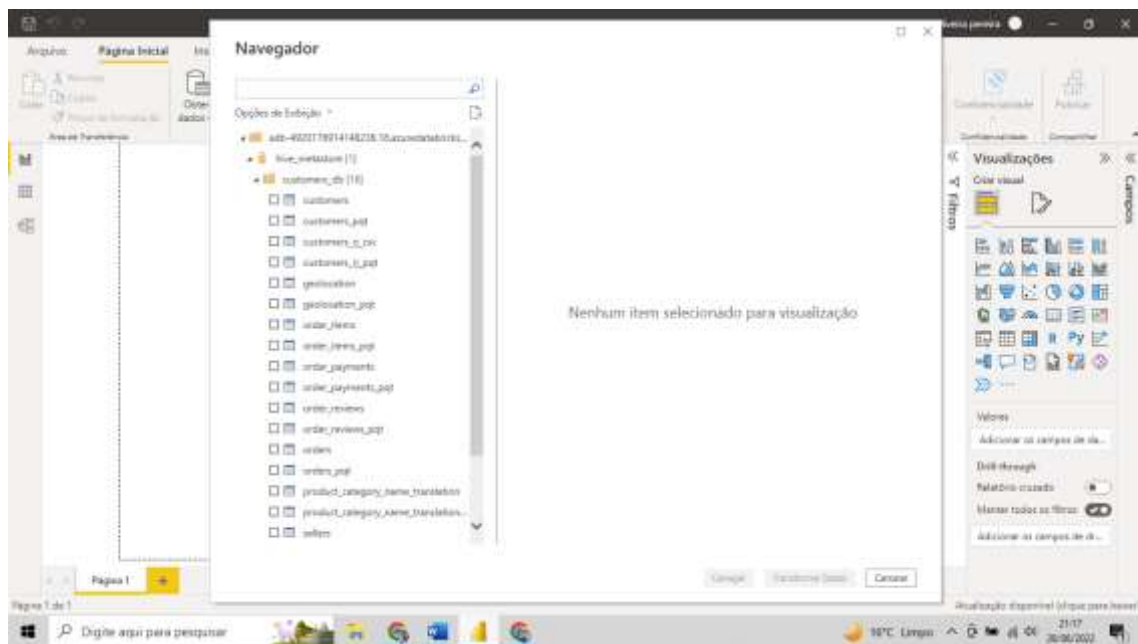


Conectando por token

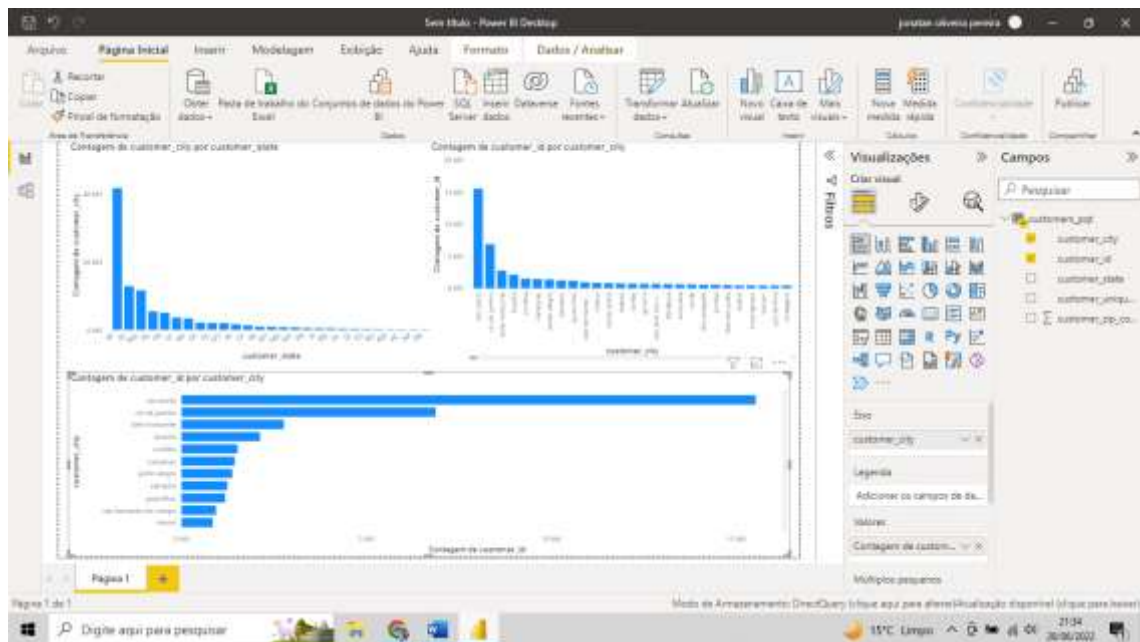


Acesso pelo token funcionando

POWER BI – aqui deve ser feito as Análises



TESTE COM POWER BI



FINALIZANDO A ARQUITETURA DE DADOS – PREPARANDO O NOTEBOOK PARA EXECUÇÃO AUTOMÁTICA COMENTAR TODAS AS LINHAS DE MONTAGEM (mnt)

The screenshot shows the Databricks web interface. At the top, there's a navigation bar with 'Microsoft Azure' and 'Databricks' logos. Below that, a breadcrumb trail shows 'Home' > 'Workspace' > 'Mounting Data lakes'. The main content area displays a notebook titled 'Mounting Data lakes' in a Python environment. The notebook is running on a cluster named 'olist_cluster'. The code in the notebook mounts an Azure Data Lake Storage (ADLS) account. The output shows the mounting was successful.

```

1  ...
2  dbutils.fs.mount("mnt/landing")
3  ...

/mnt/landing has been unmounted.
Out[1]: True

Loading time 20.40 seconds — by edmling@outlook.com on 2023/02/14 14:07:04 on olist_cluster

4  ...
5  configs = {"fs.azure.account.auth.type": "OAuth",
6            "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
7            "fs.azure.account.oauth2.client.id": "43ca023e-0a07-4284-a30e-bdfe96e1231d",
8            "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="olist_scope", key="olist-secret1"),
9            "fs.azure.account.oauth2.endpoint": "https://login.microsoftonline.com/43e28b17-ba1d-46db-a895-f6f57839f3c9/oauth2/token"}
10
11 # optionally, you can add directory-names to the source URI of your mount point.
12 dbutils.fs.mount(
13     source = "abfss://landingstorageaccount2.dfs.core.windows.net/",
14     mount_point = "/mnt/landing",

```

Descomentar todos os DROP e as demais a baixo

The screenshot shows a Databricks notebook interface. The notebook is titled 'olist_processing' and is written in Python. It contains two code blocks. The first code block, labeled 'Cell 23', contains a SQL command to drop a table if it exists: `DROP TABLE IF EXISTS customers_db.geolocation`. The second code block, labeled 'Cell 24', contains a SQL command to drop a table if it exists: `DROP TABLE IF EXISTS customers_db.order_items`. The notebook is running on a Microsoft Azure Databricks cluster.

Agora vamos ativar o Cluster

Para executar novamente o Script

Executando o Script

The screenshot shows a Databricks notebook interface. The notebook is titled 'olist_processing' and is written in Python. It contains a code block labeled 'Cell 1' with a Python script. The script is a Jupyter Notebook cell with a title 'Mounting Data lakes'. The script contains the following code:

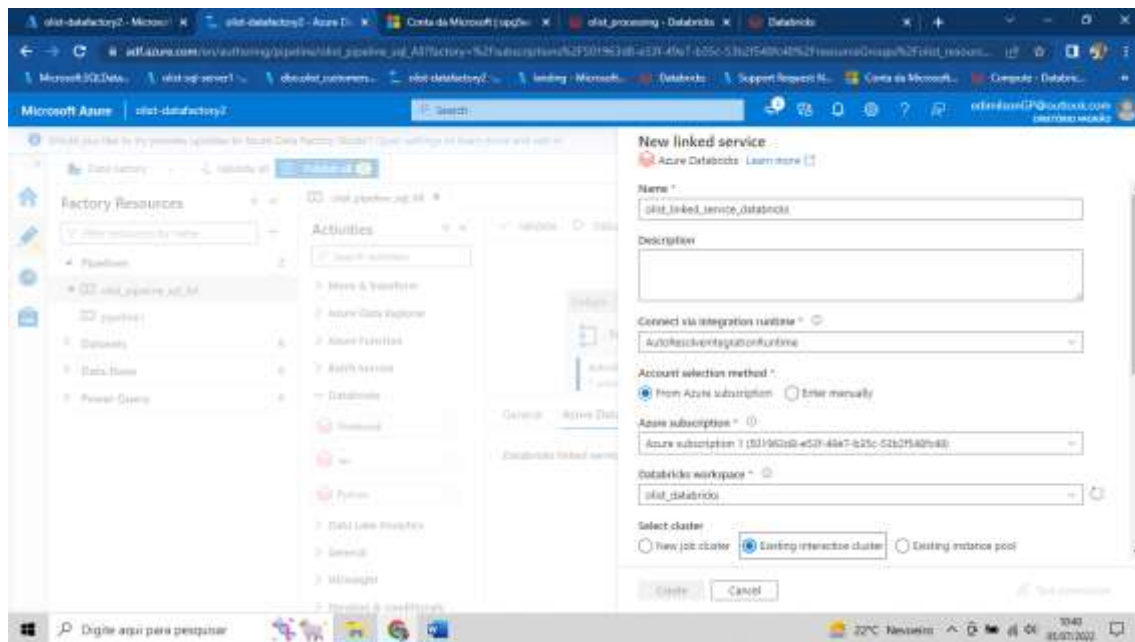
```
1 #Load, Transform, Persist Pipeline
2
3 #1-mount the data lakes
4 #2-loads csvs from landing data lake
5 #3-convert csvs to parquet and move them to processing data lake
6 #4-create sql database
7 #5-create tables based on parquet format files
8 #6-specific analysis will be moved to curated data lake and then loaded into sql tables
9 #7-powerBI application reads directly from sql tables at databricks rest api service
10
11 Uploading notebook
12
13 Mounting Data lakes
14
15 ***
16 dbutils.fs.mount("mnt/landing")
```

FAZER O AGENDAMENTO DO NOTEBOOK(SCRIPT) PARA RODAR AUTOMATICAMENTE DOM DATA FACTORY

EM DATABRICKS TEM O NOTEBOOK

CLICAR E ARRASTAR PARA AREA

CONFIGURANDO

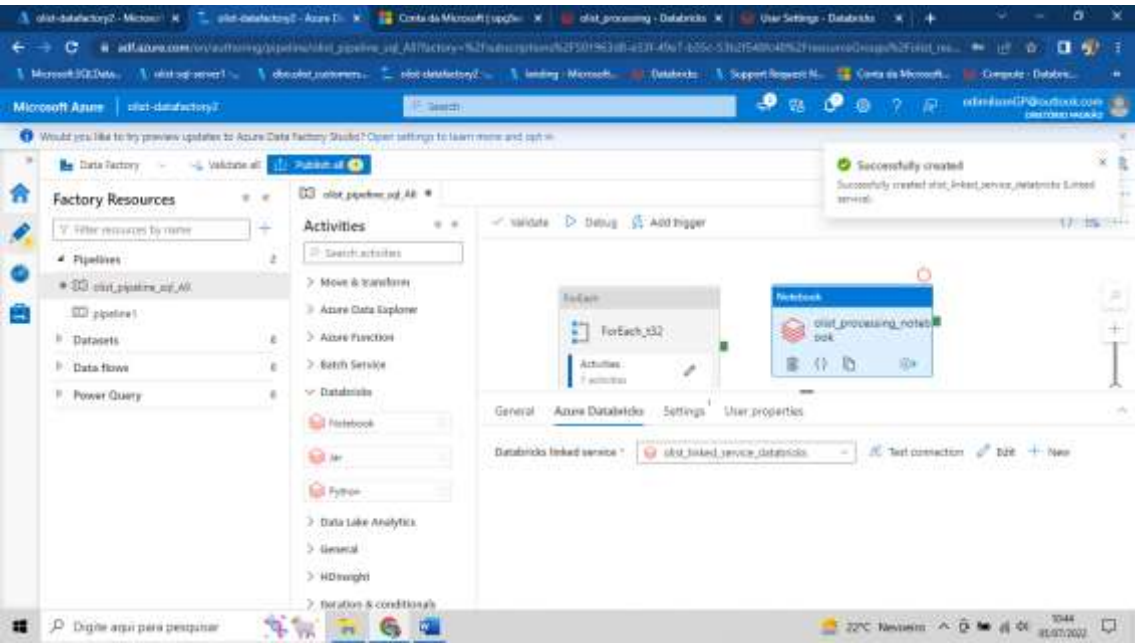


CRIAR TOKEN DE ACESSO

AGORA COPIAR

CLICAR EM CRIAR

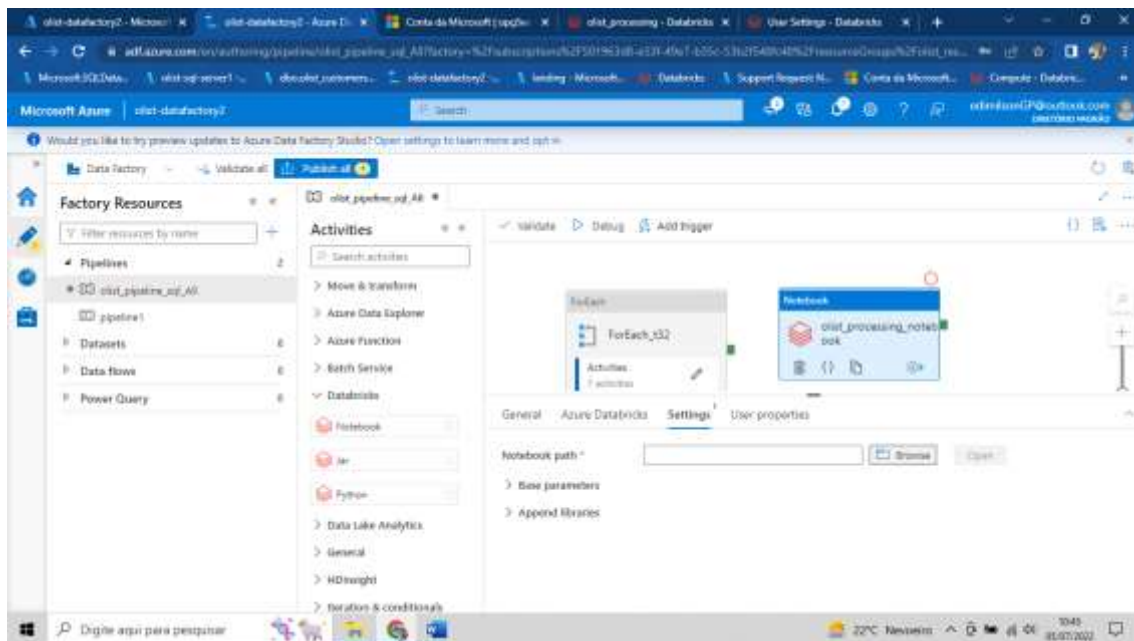
CRIADO



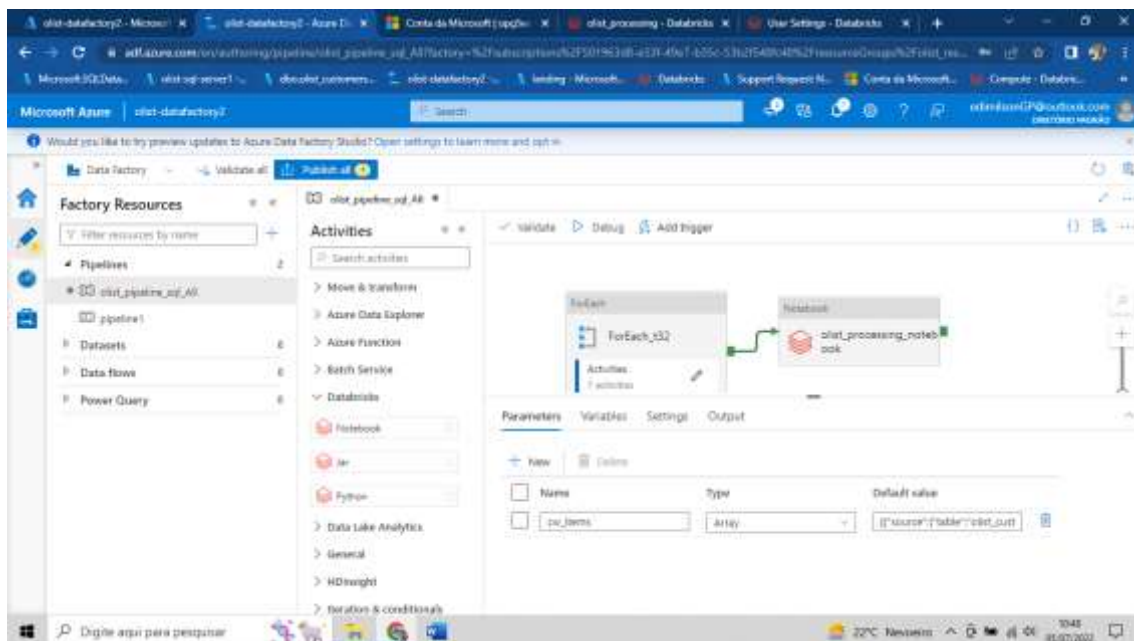
TESTE DE CONEXÃO

AGORA CONECTAR O NOTEBOOK

EM CONFIGURAÇÃO



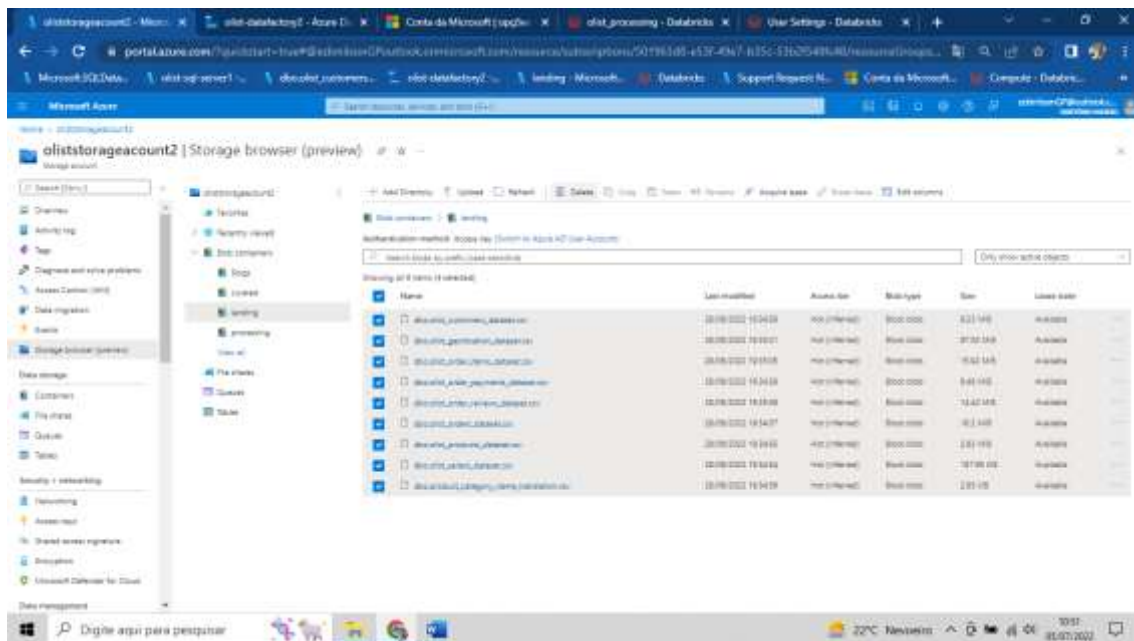
CONECTANDO



AGORA VAMOS FAZER O TESTE

VAMOS DELETAR

EM LANDING

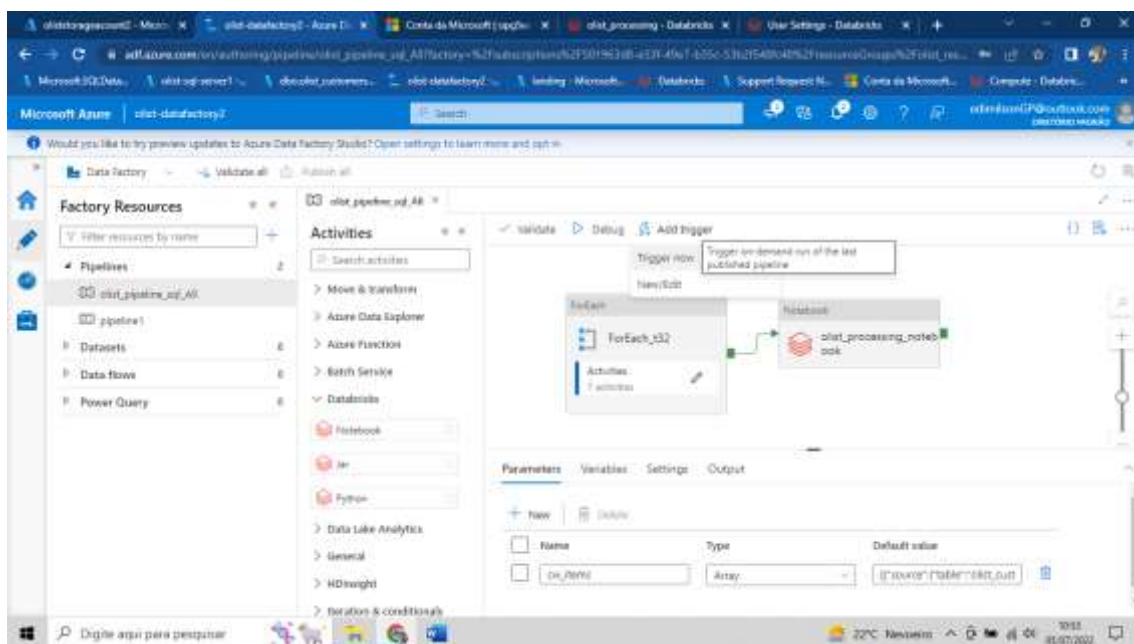


DELETADO

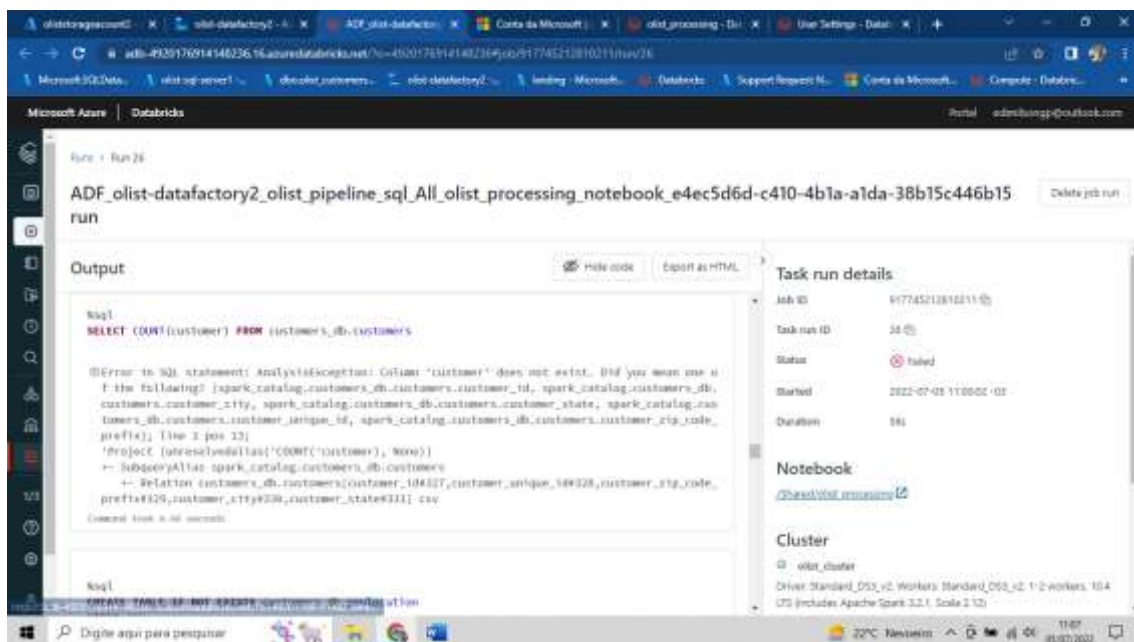
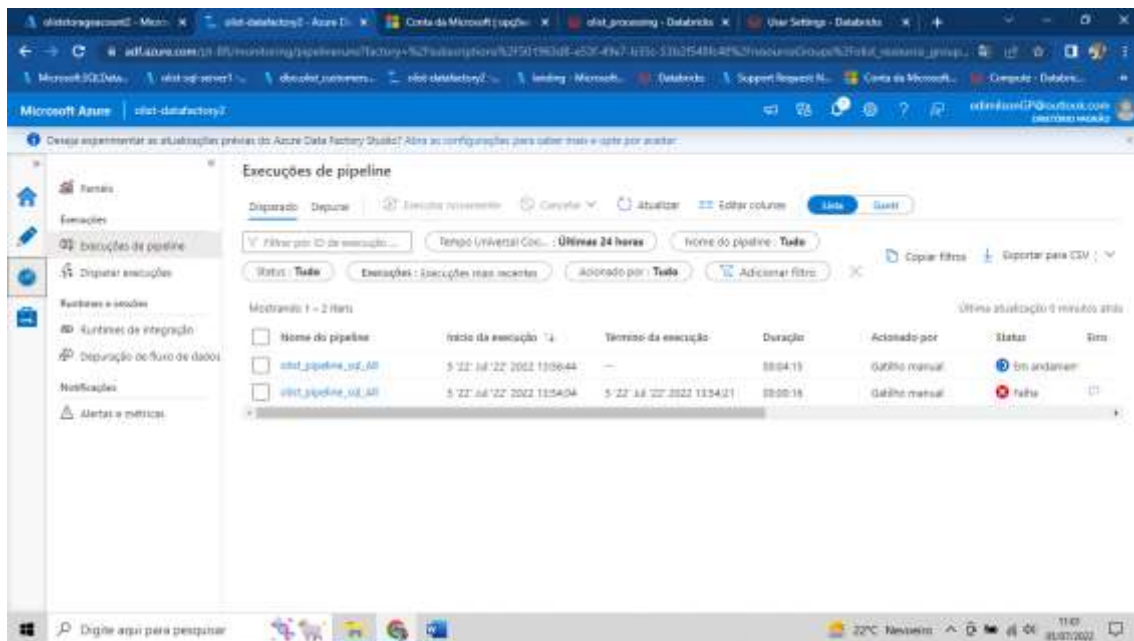
FAZER ISSO PARA OS DEMAIS CONTAINERS

PROCESSING E CURATED

AGORA NO DATA FACTORY VAMOS DISPARAR O GATILHO



APRESENTOU ERRO

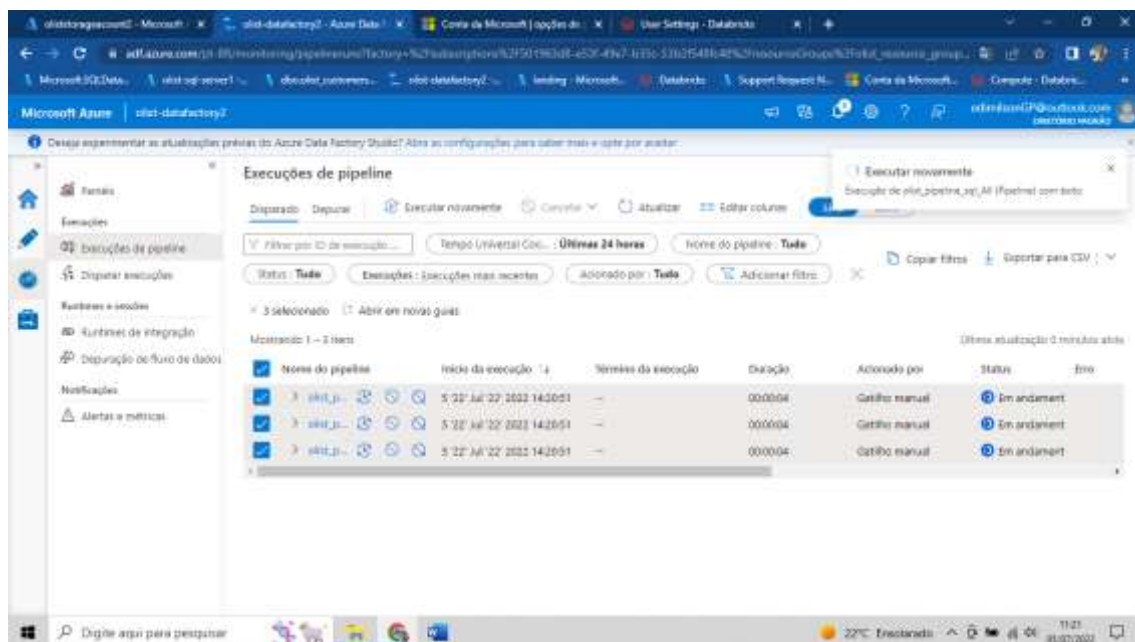
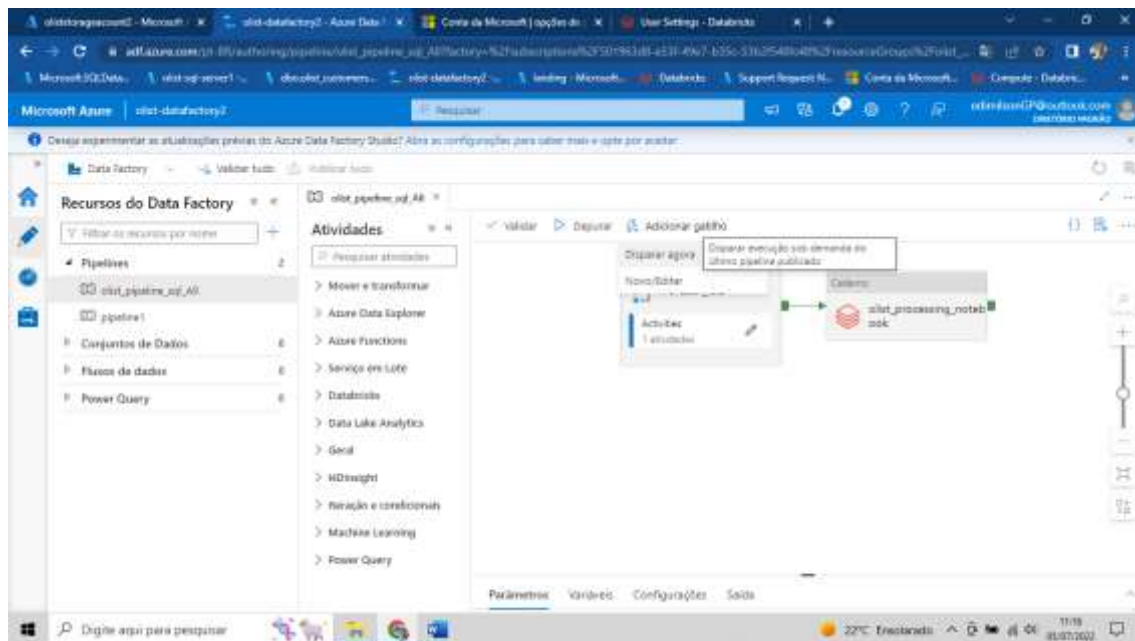


CORREÇÃO REMOVI O COUNT

VERIFICAR SE TEM O DBO NA FRENTE TAMBÉM DE TODOS OS LOCATION

E FECHAR O NOTEBOOK

EXECUTAR O PIPELINE NOVAMENTE NO DATA FACTORY



EM ANDAMENTO

Microsoft Azure | rdt-datalake2

Desaja experimentar as funcionalidades prévia do Azure Data Factory Studio? Abra as configurações para saber mais e apte por aceitar.

Execuções de pipeline

Disparado | Depurar | **Simular novamente** | Cancelar | Atualizar | Editar columnas | **Seleto** | **Quero**

Filtrar por ID de execução... | Tempo Universal Coe...: **Últimas 24 horas** | Nome do pipeline: **Tudo** | Copiar filtro | Exportar para CSV

Status: **Tudo** | Execuções: (execuções mais recentes) | Alocado por: **Tudo** | Adicionar filtro

3 selecionado | Abrir em novas guias

Mostrando 1 - 3 itens

	Nome do pipeline	Início da execução	Terminou da execução	Duração	Alocado por	Status	Final
3	rdt_j1...	5/22 Jul 22 14:24:33	—	00:00:04	Getfile manual	Em andamento	
3	rdt_j1...	5/22 Jul 22 14:24:33	—	00:00:04	Getfile manual	Em andamento	
3	rdt_j1...	5/22 Jul 22 14:24:33	—	00:00:05	Getfile manual	Em andamento	

(Última atualização 2 minutos atrás)

22°C | Trezevidas | 11:28 | 21/07/2022