CAIEIRAS
2022

CONTROLE DE VERSÃO			
Autor	**Versão**	**Data**	**Descrição**
Marcus Vinicius de Araújo	1.0	25/03/2022	Criação da documentação

1. Introdução

Este documento tem como objetivo detalhar as necessidades do projeto PoccoBank, tendo como ponto vista técnico e com isso trazer uma lista de soluções, premissas e as atividades que serão executadas ao decorrer do projeto. Nesta nova solicitação será utilizada uma nova ferramenta no auxílio e realização do projeto.

2. Solicitação

A PoccoBank, um dos principais bancos do mundo, pretende gerar alguns relatórios semanais para os principais investidores e para isso eles necessitarão da cotação diária do dólar. Após um grande escândalo envolvendo compra de parmegianas superfaturadas o estrategista chefe da PoccoBannk, Anthony Hopkins, foi afastado. O escândalo ficou conhecido como ParmegiaGate e derrubou metade do alto escalão do banco. Semanas depois o posto de estrategista chefe foi ocupado por Robyn Fenty, a primeira mulher a ocupar a posição. Logo na primeira reunião a Sra. Fenty pediu para que o Databricks fosse incluso na arquitetura do projeto pois, nas palavras dela, “Databricks é o momento, é o queridinho do mercado.”

Usando o Azure Databricks, crie uma *pipeline* para coletar os valores do dólar da API do Banco Central e inserir essas informações no Azure SQL, conforme o desenho da arquitetura. No Azure SQL crie uma *Stored Procedure* para converter o tipo dos dados para data e *float/money/*decimal e depois retornar uma arquivo (via Data Factory) *Parquet* no *Blob* Azure.

3. Premissa da solução

As seções abaixo irão apresentar as soluções da premissa apresentada anteriormente:

Origem e especificações dos dados:

- Através de uma API HTTP será realizado uma ingestão dos dados, ao qual estão disponíveis no Banco Central em formato ‘.csv’;

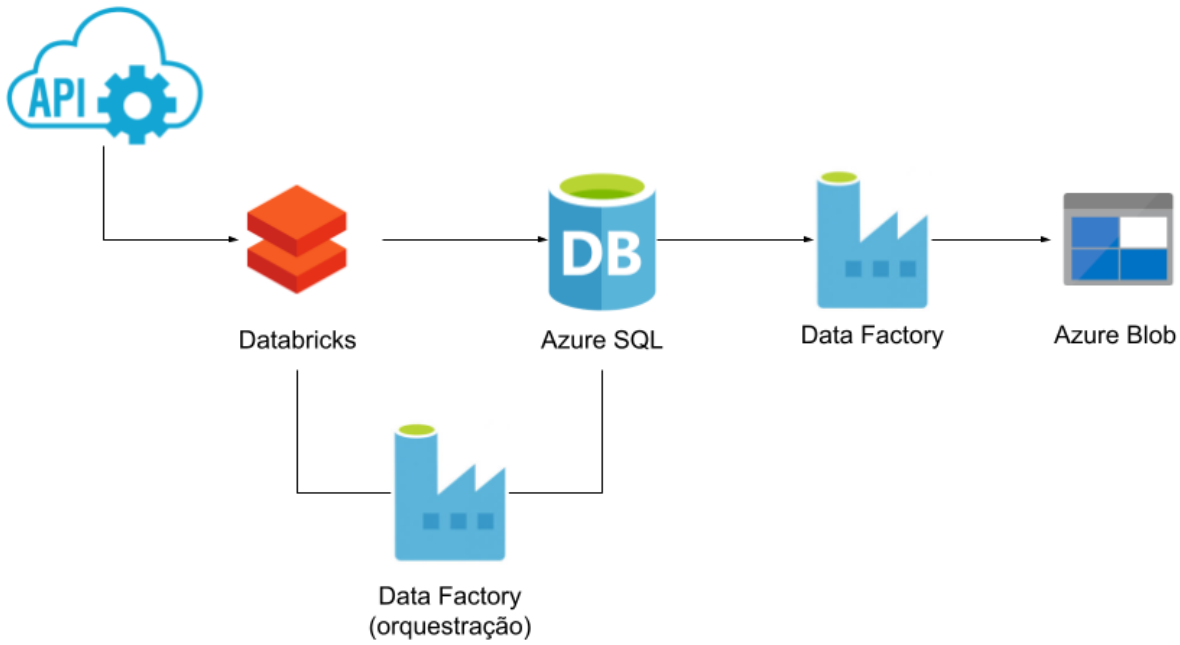
- O arquivo que será utilizado possui um dicionário com 03 campos.

Ambiente do desenvolvimento:

- A solução proposta será desenvolvida utilizando alguns dos recursos que estão provisionados no Grupo de Recursos Azure | Estudos – Azure, localizado no portal do Azure. Os recursos utilizados são o Azure Data Factory, Azure SQL, Azure Blob Storage, Azure Databricks.

4. Modelagem da Arquitetura

A figura a abaixo apresenta a arquitetura da solução proposta para a realização da solicitação:



5. A inserção, transformação e o armazenamento dos dados

Foi criado um *schema* 'dolar_marcus_araujo', no Azure SQL, após a criação do *schema* foi montada a primeira tabela *stage* 'dolar_stage_marcus_araujo02' ao qual possuía a finalidade de receber os dados que provinham da API, nesta API continha três colunas sendo a cotacaoCompra, cotacaoVenda e dataHoraCotacao, sendo elas do tipo varchar. Em seguida foi criado mais uma tabela de nome 'dolar_final_marcus_araujo02' que tem como finalidade o armazenamento dos dados que percorreram pela *Procedure,* que é feito todo o tratamento dos dados da primeira tabela a 'dolar_stage_marcus_araujo02'.

A *procedure* ‘dolar_procedure_marcus_araujo02’ é formado por uma *truncate* na tabela ‘dolar_final_marcus_araujo02’, que tem como o objetivo de fazer a eliminação de algum dado que já havia sido inserido anteriormente, seja por conta dos testes antes da finalização do projeto. Ela possui a finalidade de fazer o processo de inserção dos dados da tabela ‘dolar_stage_marcus_araujo02’ para a tabela ‘dolar_final_marcus_araujo02’, fazendo o tratamento nos dados localizados nas colunas cotacaoCompra e cotacaoVenda alterando a ‘,’ para o ‘.’ e os dados que estão no tipo varchar para o tipo o *float.* Na coluna dataHoraCotacao é alterado o tipo varchar para o tipo datetime. E por fim, a *truncate* na tabela ‘dolar_stage_marcus_araujo02’ é utilizada para poder deixar os dados mais dispostos a uma nova requisição.

No Databricks foi construído um código ao qual faz todo o procedimento de extração dos dados da API, foram utilizadas as bibliotecas Pandas e Pymssql. Pandas é uma biblioteca que tem como função a manipulação e análise de dados com o auxílio da linguagem Python. Sendo ela muito utilizada na área de análise e ciência de dados, por conta da quantidade de ferramentas que ela pode oferece. O pymssql é uma biblioteca Python, é utilizada para acessar o banco de dados MySQL do Python. Importando o módulo Python do pymssql no programa, para que possa usar a API deste módulo para se conectar ao MySQL

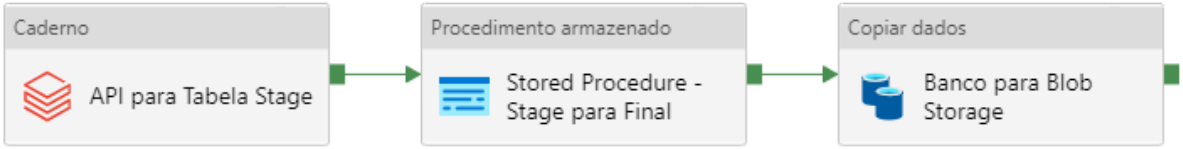
No Data Factory uma *pipeline* foi montada para realizar a orquestração de todo o processo. No qual possui três atividades, sendo:

Caderno | API par Tabela *Stage*: Este é responsável pela extração dos dados localizado na API, através de um código por linguagem Python e por fim faz a inserção dos mesmos, no Azure SQL, na tabela dolar_stage_marcus_araujo02;

Stored Procedure | Stored Procedure – Stage para Final: Faz o procedimento de tratamento dos dados e os envia da tabela *stage* para a tabela final;

Copy Data | Banco para Blob Storage: Faz a extração dos arquivos em formato *Parquet* e são armazenados no Azure Blob Stored.

A figura abaixo apresenta o *Pipeline* criado no Azure Data Factory:



6. ***Layout* dos arquivos utilizados**

Para a realização deste projeto foram desenvolvidas duas tabelas:

Tabela *Stage:*

dolar_marcus_araujo.dolar_stage_marcus_araujo02
cotacaoCompra == VARCHAR
cotacaoVenda == VARCHAR
dataHoraCotacao == VARCHAR

Tabela Final:

dolar_marcus_araujo.dolar_final_marcus_araujo02
cotacaoCompra == FLOAT
cotacaoVenda == FLOAT
dataHoraCotacao == DATATIME

7. ***Link* para acessar o Azure Databricks**

Link:

https://adb-7998781248845980.0.azuredatabricks.net/?o=7998781248845980#notebook/2320329907711638/command/2320329907711639

8. **API**

API do Banco Central:

https://olinda.bcb.gov.br/olinda/servico/PTAX/versao/v1/odata/CotacaoDolarPeriodo(dataInicial=@dataInicial,dataFinalCotacao=@dataFinalCotacao)?@dataInicial='01-01-2019'&@dataFinalCotacao='12-31-2025'&$top=9000&$format=text/csv&$select=cotacaoCompra,cotacaoVenda,dataHoraCotacao