



**BlueAcademy**

CONTROLE DE VERSÃO			
Autor	Versão	Data	Descrição
Marcus Vinicius de Araújo	1.0	22/03/2022	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.

## 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. Anthony Hopkins, estrategista chefe do banco, solicitou a criação de um processo capaz de trazer a cotação do câmbio e suas variações ao longo da semana. Usando o Azure Data Factory, 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 proposta. No Azure SQL crie uma *Stored Procedure* para converter o tipo dos dados para data e *float/money/decimal* e depois retornar um arquivo Parquet no Azure Storage.

## 3. Premissa da solução

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

### Origem e especificações dos dados:

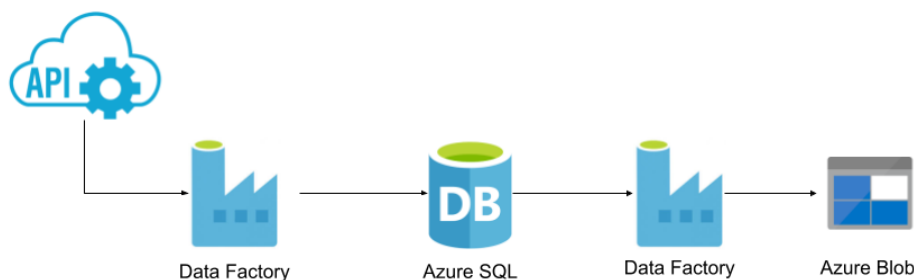
- Através de uma API HTTP será realizado uma ingestão dos dados, ao qual estão disponíveis no Banco Central em formato '.csv';
- O arquivo que será utilizado possui um dicionário com 03 campos.

### 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 e o Azure Blob Storage.

#### 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\_araujo' 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\_araujo' 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\_araujo'.

A *procedure* 'dolar\_procedure\_marcus\_araujo' é formado por uma *truncate* na tabela 'dolar\_final\_marcus\_araujo', 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\_araujo' para a tabela 'dolar\_final\_marcus\_araujo', 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\_araujo' é utilizada para poder deixar os dados mais dispostos a uma nova requisição.

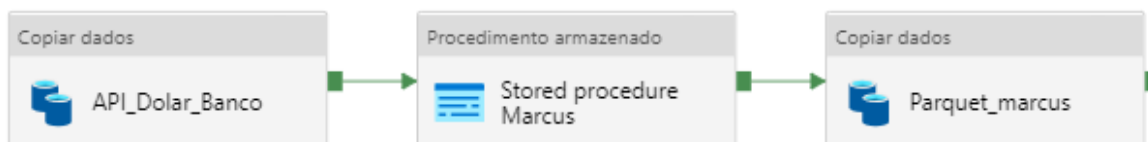
No Data Factory uma *pipeline* foi montada para realizar a orquestração de todo o processo. No qual possui três atividades, sendo:

*Copy Data | API\_Dolar\_Banco*: Este é responsável pela extração dos dados localizado na API e também faz a inserção dos mesmos, no Azure SQL;

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

*Copy Data* | Parquet\_marcus: 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_araujo
cotacaoCompra == VARCHAR
cotacaoVenda == VARCHAR
dataHoraCotacao == VARCHAR

### Tabela Final:

dolar_marcus_araujo.dolar_final_marcus_araujo
cotacaoCompra == FLOAT
cotacaoVenda == FLOAT
dataHoraCotacao == DATETIME

## 7. 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](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)