



BlueShift

Business Blueprint

CONTROLE DE VERSÃO			
Autor	Versão	Data	Descrição
Lucas Ferreira B Lima	1.0	07/07/2022	Criação do documento.

1. Introdução

Este documento visa detalhar as necessidades do projeto PoccoBank do ponto de vista técnico, bem como listar as possíveis soluções, suas premissas e atividades a serem executadas durante o projeto.

2. Solicitação

A PoccoBank pretende gerar alguns alguns relatórios semanais para os principais investidores e para isso eles necessitarão da cotação diária do dolar.

Foi solicitada a criação de um processo capaz de trazer a cotação do cambio e suas variações ao longo da semana. Usando o Azure Data Factory, iremos criar um pipeline para coletar os valores do dólar da API do Banco Central e inserir essas informações no Azure SQL.

3. Premissas da solução

A seção abaixo apresenta as premissas da solução

Origem e especificação dos dados

- O arquivo da API é capturado através do seguinte link

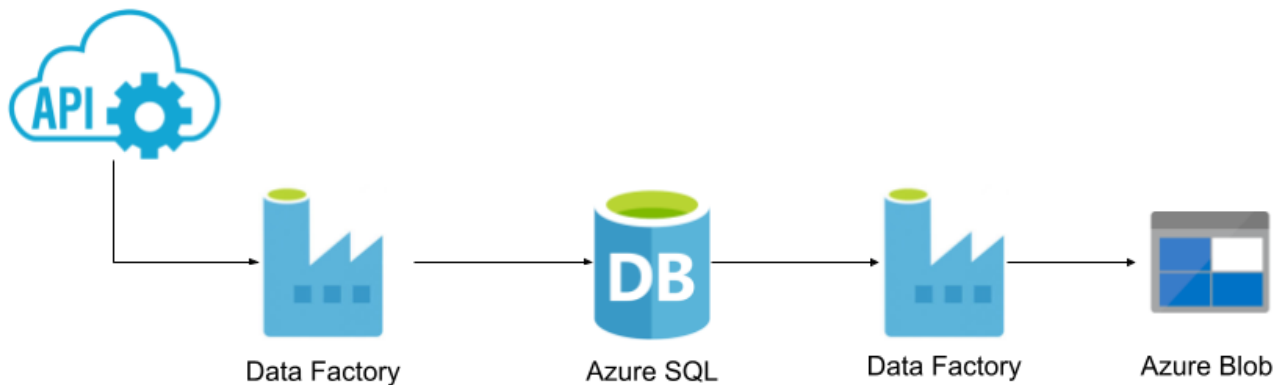
[https://olinda.bcb.gov.br/olinda/servico/PTAX/versao/v1/odata/CotacaoDolarPeriodo\(dataInicial=@dataInicial,dataFinalCotacao=@dataFinalCotacao\)?@dataInicial='01-01-2019'&@dataFinalCotacao='12-31-2025'&\\$top=9000&\\$format=text/csv&\\$select=cotacaoCompra,cotacaoVenda,dataHoraCotacao](https://olinda.bcb.gov.br/olinda/servico/PTAX/versao/v1/odata/CotacaoDolarPe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

Ambiente de desenvolvimento

- Portal Azure
 - Azure SQL
 - Azure Data Factory
 - Azure Blob Storage

4. Modelo da arquitetura sugerida

A figura abaixo apresenta a arquitetura da solução proposta com base no levantamento de requisitos e entendimento do negócio.



5. Extração via Azure Data Factory

Será desenvolvido um processo de extração dos dados através do Azure Data Factory.

O fluxo deverá trazer os dados disponibilizados pela API do Banco Central e levá-los para uma tabela Stage no Azure SQL, na sequência executaremos uma procedure que fará a conversão dos dados (de string para float e datetime), salvando os dados convertidos em uma outra tabela (tabela final).

O fluxo é finalizado com a cópia destes dados da tabela final, para o Azure Blob Storage em um arquivo do formato parquet.

6. Layout dos arquivos

Para este projeto estamos considerando um *layout* principal do arquivo da API que, ao decorrer do processo, alimentarão as tabelas no Azure SQL.

- Tabela de dados da API

cotacaoCompra	cotacaoVenda	dataHoraCotacao
3,8589	3,8595	2019-01-02 13:04:46.568
3,7677	3,7683	2019-01-03 13:04:50.817
3,7621	3,7627	2019-01-04 13:06:29.332
3,7056	3,7062	2019-01-07 13:09:39.652
3,7202	3,7208	2019-01-08 13:09:06.45
3,6925	3,6931	2019-01-09 13:22:28.243
3,6863	3,6869	2019-01-10 13:03:05.634
3,7135	3,7141	2019-01-11 13:09:51.933
3,7255	3,726	2019-01-14 13:07:34.594

7. Processamento dos dados

Para atender a necessidade de armazenamento e conversão dos dados, iremos necessitar de duas tabelas e um procedure no Azure SQL.

- Tabela Stage: Irá armazenar os dados provindos diretamente da API, sem nenhuma conversão e tratamento, e todos os dados no formato VARCHAR

cotacaoCompra	cotacaoVenda	dataHoraCotacao
3,8589	3,8595	2019-01-02 13:04:46.568
3,7677	3,7683	2019-01-03 13:04:50.817
3,7621	3,7627	2019-01-04 13:06:29.332
3,7056	3,7062	2019-01-07 13:09:39.652

- Procedure: Script que irá selecionar os dados da tabela Stage e fazer a conversão, trocando nas colunas cotacaoCompra e cotacaoVenda as vírgulas (,) para ponto(.), transformando a coluna dataHoraCotacao para DATETIME e por fim salvando estes dados na tabela final

- Tabela Final: Irá armazenar os dados convertidos

cotacaoCompra	cotacaoVenda	dataHoraCotacao
3.8589	3.8595	2019-01-02T13:04:46.5670000
3.7677	3.7683	2019-01-03T13:04:50.8170000
3.7621	3.7627	2019-01-04T13:06:29.3330000
3.7056	3.7062	2019-01-07T13:09:39.6530000
3.7202	3.7208	2019-01-08T13:09:06.4500000

8. Scripts.

QUERYS DE CRIAÇÃO DAS TABELAS

Tabela Stage

```
CREATE TABLE [dolar_lucas_fb_lima].[dolar_Stage_lucas_fb_lima] (  
cotacaoCompra VARCHAR NULL,  
cotacaoVenda VARCHAR NULL,  
dataHoraCotacao VARCHAR NULL)
```

Tabela Final

```
CREATE TABLE [dolar_lucas_fb_lima].[dolar_final_lucas_fb_lima] (  
cotacaoCompra FLOAT NULL,  
cotacaoVenda FLOAT NULL,  
dataHoraCotacao DATETIME NULL)
```

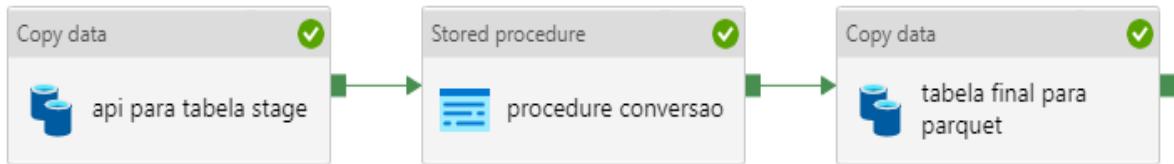
QUERY DO PROCEDURE

A conversão dos dados é feita através do seguinte script SQL

```
CREATE PROCEDURE [dolar_lucas_fb_lima].[procedure]  
AS  
BEGIN  
TRUNCATE TABLE [dolar_lucas_fb_lima].[dolar_final_lucas_fb_lima]  
INSERT INTO [dolar_lucas_fb_lima].[dolar_final_lucas_fb_lima]  
    (cotacaoCompra, cotacaoVenda, dataHoraCotacao)  
    SELECT  
        REPLACE (cotacaoCompra, ',', '.') AS FLOAT,  
        REPLACE (cotacaoVenda, ',', '.') AS FLOAT,  
        dataHoraCotacao AS DATETIME  
    FROM [dolar_lucas_fb_lima].[dolar_Stage_lucas_fb_lima]  
TRUNCATE TABLE [dolar_lucas_fb_lima].[dolar_Stage_lucas_fb_lima]  
END
```

9. Azure Data Factory

Para a captação, desenvolvimento e entrega dos dados utilizamos o Azure Data Factory para criação do nosso pipeline, conforme imagem abaixo:



O pipeline segue sequencialmente da seguinte forma:

1. **Copy data:**
Pega como Source o link de captação da API e como Sink a tabela Stage para armazenar os dados capturados sem conversão
2. **Stored procedure:**
Busca no Azure SQL o procedure que criamos e executa o seu script, que faz as conversões como explicado anteriormente, salvando os dados na tabela final
3. **Copy data:**
Pega os dados da tabela final que estão convertidos e salva no Azure Blob Storage no formato Parquet.

10. Resultados

Podemos observar no Azure Data Factory que o debug do pipeline foi executado com êxito

Name	Type	Run start	Duration	Status
tabela final para parquet	Copy data	2022-07-07T19:07:14.1477805Z	00:00:13	✓ Succeeded
procedure conversao	Stored procedure	2022-07-07T19:06:55.6957066Z	00:00:17	✓ Succeeded
api para tabela stage	Copy data	2022-07-07T19:06:43.9566704Z	00:00:11	✓ Succeeded

E ao verificar no Azure Blob Storage, também vemos que o arquivo parquet foi salvo no contêiner

Nome	Modificado
dolar_lucas_fb_lima.dolar_final_lucas_fb_lima.parquet	07/07/2022 16:07:25