



Cotia, 27 de Novembro de 2020

Controle de Versão			
Autor	Versão	Data	Descrição
Diego	1.0	27/11/2020	Criação do Documento

## 1.Introdução

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

## 2.Solicitação

O cliente solicita um modelo de extração e processamento de dados que resultem em um modelo que alimente uma ferramenta de visualização. O projeto será feito da seguinte forma: Os dados a serem tratados estão alocados no Azure Blob Storage, será feita uma conexão via Nifi para iniciar a migração de dados até ao banco de dados relacional MySQL aonde será feita a ETL dos dados para a criação de um Data warehouse. O desenvolvimento dos Dashboards será feito através do Power Bi.

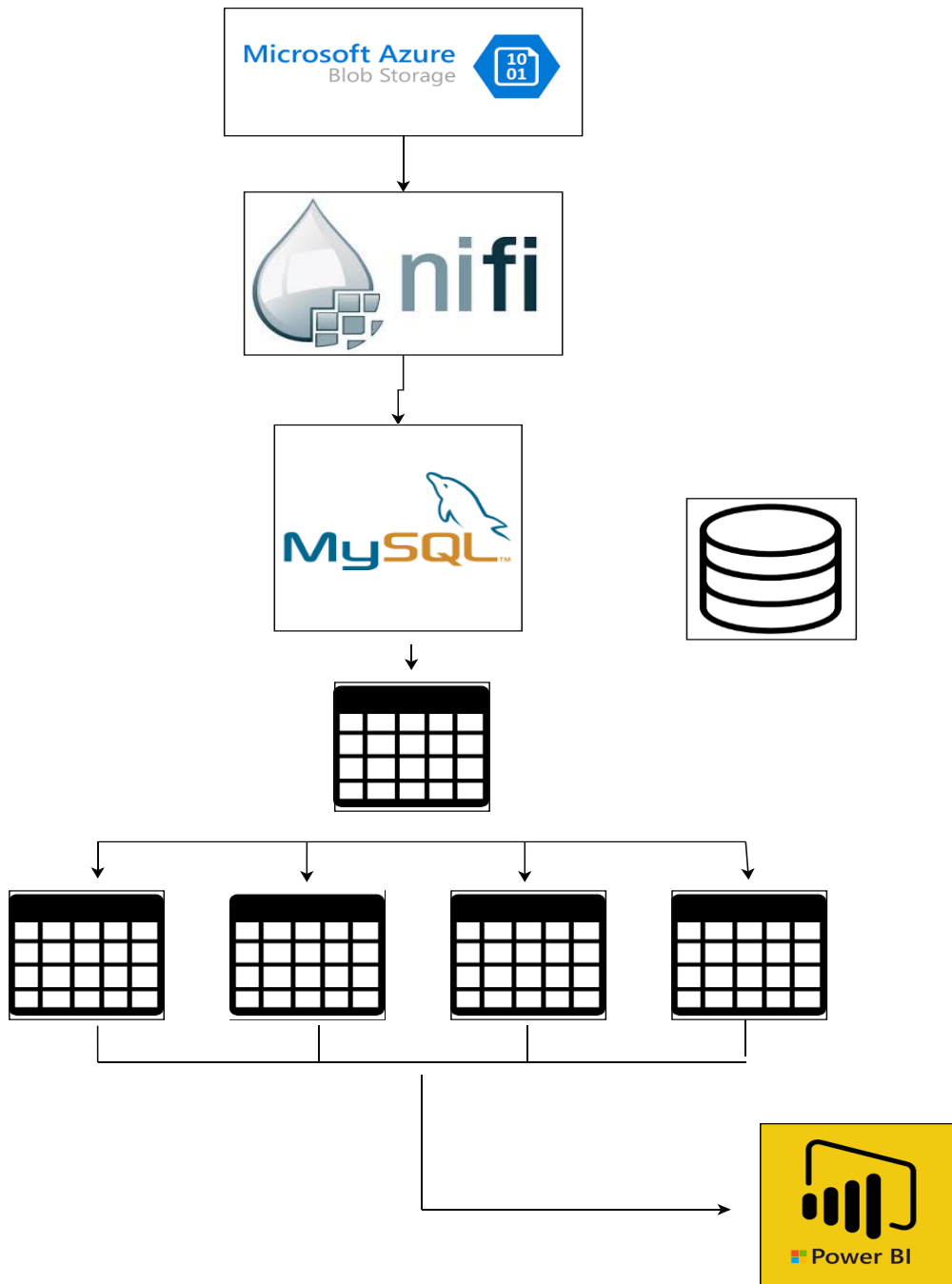
### 3. Premissas da solução

Os itens listados abaixo apresentam as premissas da solução, origem e especificação dos dados.

- O arquivo CSV deverá ser disponibilizado no Azure Bob Storage para o início do processo de extração via Nifi;
- O arquivo CSV deverá seguir o layout pré-definido e especificado neste documento;
- A Pocco deverá disponibilizar acessos aos ambientes de desenvolvimento em todas as ferramentas especificadas na arquitetura proposta neste documento para o time Blueshift.

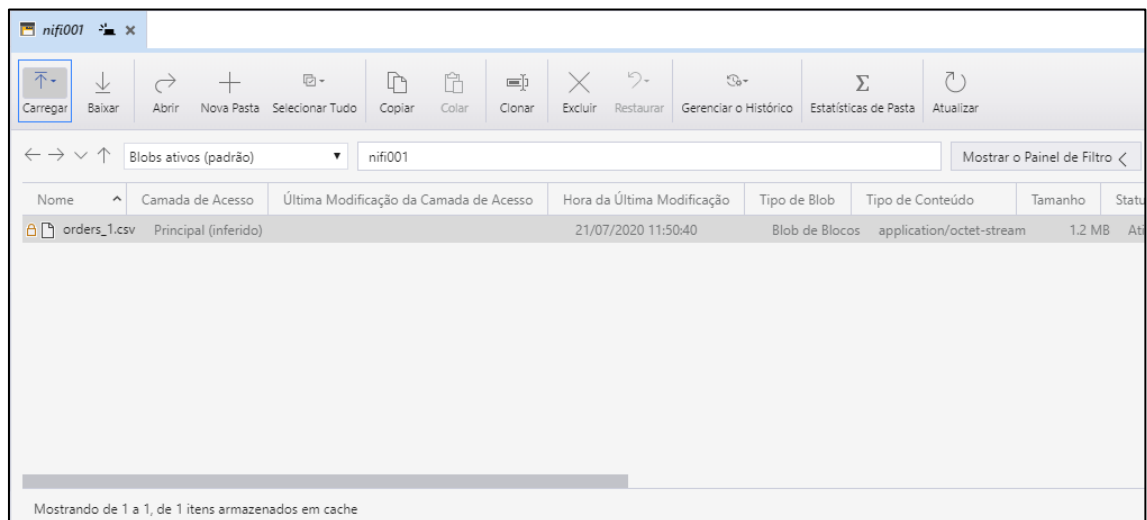
## 4. Modelo da arquitetura sugerida

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



## 5. Disponibilização dos arquivos via Azure Blob Storage

O Arquivo estará disponibilizado no Container nomeado “nifi”, nome do arquivo “orders\_1.csv”



## 6. Criação do processo de flow file – Nifi

Ao todo serão usados 5 processors no processo de migração sendo eles:

- GenerateFlowFile
- FetchAzureBlobStorage
- ReplaceText
- PutDatabaseRecord
- PutSQL

**GenerateFlowFile:** Processor que cria e starta um flow file com dados aleatórios ou conteúdo personalizado. Util. para testes de carga, configuração e simulação.

**FetchAzureBlobStorage:** Faz a conexão com o Azure Blob Storage gravando o conteúdo no flow file.

**ReplaceText:** Atualiza o conteúdo de um flow file avaliando uma expressão regular e substituindo por outra expressão por exemplo: Substitui espaços por vírgulas, pontos, etc.

**PutDatabaseRecord:** É um recorder para inserir registros de um arquivo de fluxo de entrada. Esses registros são transmitidos em instruções SQL e executados em um único lote. Resumindo faz a conexão com seu banco de dados

**PutSQ:** Executa um comando SQL no caso a procedure que faz todo o insert dos dados nas tabelas fato e dimensão,

## Configuração dos Processors

**GenerateFlowFile:** Properties de acordo com a imagem abaixo:

**Configure Processor**

Stopped

SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field +

Property		Value
File Size	?	0B
Batch Size	?	1
Data Format	?	Text
Unique FlowFiles	?	false
Custom Text	?	No value set
Character Set	?	UTF-8
Mime Type	?	No value set

CANCEL APPLY

**FetchAzureBlobStorage:** Existem 4 campos obrigatórios para que possa ser feita a conexão do Nifi com o Azure Blob Storage que são:

FetchAzureBlobStorage / Properties	
Container Name	Nome do container aonde está armazenado o arquivo
Blob	Nome do arquivo com sua extensão
Storage Account Name	Login da conta no Storage
Storage Account Key	Key da Storage Account

Imagem ilustrativa:

### Configure Processor

Stopped

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value	
Container Name	nifi001	
Storage Credentials	No value set	
Storage Account Name	Sensitive value set	
Storage Account Key	Sensitive value set	
SAS Token	No value set	
Common Storage Account Endpoint Suffix	No value set	
Blob	orders_1.csv	
Proxy Configuration Service	No value set	

CANCEL

APPLY



**ReplaceText:** Deverá ser configurado afim de deixar todas expressões regulares da mesma forma:

ReplaceText / Properties	
Search Value	(?s)(^[^\n]*) (.*\$)
Replacement Value	\${'\$1':replace(" ", "")}\$2
Character Set	UTF-8
Maximum Buffer Size	1 MB
Replacement Strategy	Regex Replace
Evaluation Mode	Line-by-Line
Line-by-Line Evaluation Mode	All

**Imagem Ilustrativa:**

### Configure Processor

Stopped

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value	
Search Value	(?s)(^[^\n]*) (.*)	
Replacement Value	\${'\$1':replace(" ", "")}\$2	
Character Set	UTF-8	
Maximum Buffer Size	1 MB	
Replacement Strategy	Regex Replace	
Evaluation Mode	Line-by-Line	
Line-by-Line Evaluation Mode	All	

CANCEL

APPLY

**PutDatabaseRecord:** Deverá conter as informações do banco de dados MySQL e o tipo de arquivo que será inserido;

PutDatabaseRecord / Properties	
Record Reader	CSVReader
Database Type	MySQL
Statement Type	INSERT
Database Connection Pooling Service	DBCPCConnectionPool
Schema Name	pedidos
Table Name	orders
Allow Multiple SQL Statments	true
Quote Table Identifiers	true
Quote Column Identifiers	true

**Imagem Ilustrativa:**

**Configure Processor**

Stopped

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field +

Property	Value	
JDBC Connection Pool	DBCPCConnectionPool	→
SQL Statement	call select_all();	
Support Fragmented Transactions	true	
Database Session AutoCommit	false	
Transaction Timeout	No value set	
Batch Size	100	
Obtain Generated Keys	false	
Rollback On Failure	false	

CANCEL APPLY

**DBCConnectionPool:** Propriedade do PutDatabaseRecord que precisa ser inserido informações de seu banco de dados

Database Connection Pooling Service	
Database Connection URL	URL de conexão com seu banco de dados
Database Driver Class Name	com.mysql.jdbc.Driver
Database Driver Location(s)	Caminho aonde seu banco de dados se encontra

**Imagem Ilustrativa:**

**Controller Service Details**

SETTINGS PROPERTIES COMMENTS

Required field

Property	Value
Database Connection URL	jdbc:mysql://localhost:3306/pedidos?useTim...
Database Driver Class Name	com.mysql.jdbc.Driver
Database Driver Location(s)	C:\Program Files (x86)\MySQL\Connector J 8.0
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Password	No value set
Database User	root
Password	Sensitive value set
Max Wait Time	500 millis
Max Total Connections	8
Validation query	No value set
Minimum Idle Connections	0
Max Idle Connections	8
Max Connection Lifetime	-1

OK

**PutSQ:** Deverá receber como parâmetro o nome da Procedure que irá inserir os dados no banco, (A execução desse procedure será feita somente após a criação das tabelas fato e dimensão no banco de dados)

PutSQL	
JDBC Connection Pool	Database Connection Pooling Service
SQL Statement	Call “nome da procedure”

**Imagem Ilustrativa:**

**Configure Processor** Google Translate

Stopped

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field +

Property	Value
JDBC Connection Pool	DBCPConnectionPool
SQL Statement	call select_all();
Support Fragmented Transactions	true
Database Session AutoCommit	false
Transaction Timeout	No value set
Batch Size	100
Obtain Generated Keys	false
Rollback On Failure	false

CANCEL APPLY

## 7. Criação do Data Warehouse – MySQL

### Criação da tabela de Stage

Após o processo de migração do Azure blob Storage para o MySQL feito pelo nifi, deverá ser criada uma tabela de stage para armazenar as informações do arquivo. Essa tabela de stage deve conter as colunas com os mesmos nomes que as colunas do arquivo migrado.

### Tabelas Fato e Dimensão

De acordo com a solicitação do cliente temos que buscar as seguintes informações:

- O acumulado de vendas do último ano por Região e País. Ele gostaria de ter essa visão através de um Mapa Mundial diretamente no Relatório.
- Quantidade de vendas dos últimos 10 dias através de um gráfico de colunas.
- Quantidade de vendas e a Quantidade acumulada de vendas dos últimos 30 dias.
- Uma visão acumulada das vendas do último ano por Canal e País. De forma que seja possível ver a distribuição das vendas um determinado país por canal.

Sendo assim deverão ser criadas através da tabela stage as tabelas:

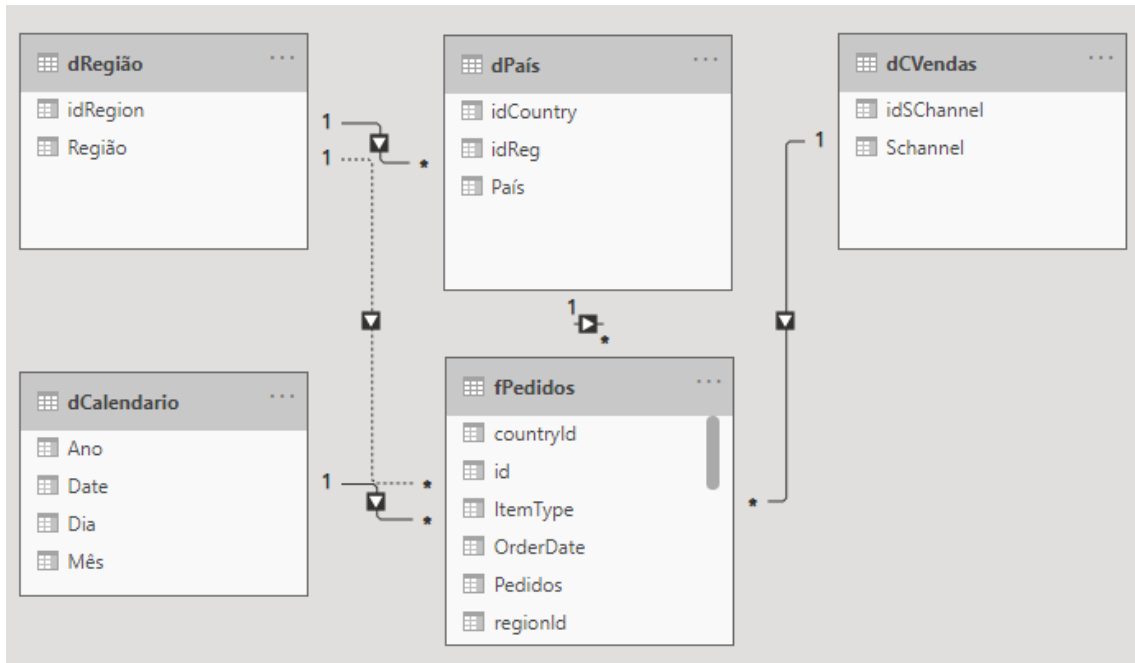
**Fato:** Contendo as vendas;

**Dimensão1:** Com os países;

**Dimensão2:** Com as Regiões;

**Dimensão3:** Com os Canais de venda,

## Relacionamento entre as tabelas:



## Insert dos registros nas tabelas fato e dimensão

Após a criação das tabelas deverá ser feito o insert dos registros da tabela stage, esse insert deverá ser feito através de uma procedure e a mesma passada como parâmetro no processador “**PutSQL**” no Nifi.

Com isso feito os arquivos estarão prontos para serem consumidos no Power BI

## 8. Criação dos Dashboards – Power Bi

### Gráficos utilizados:

**Gráfico de MAPA:** Para a criação do Dashboard que irá trazer o acumulado de vendas do último ano por Região e País será utilizado o gráfico de MAPA.

Nele contendo os campos de Localização, tamanho e dicas de ferramentas

**Localização:** Será inserido das tabelas dimensão de região e país os campos “Região” e “País”

**Tamanho:** Será inserido o campo que contem os pedidos de venda da tabela fato.

**Dicas de ferramentas:** Irá adicionar ao gráfico os valores em \$ através do campo Total de vendas vindo da tabela fato.

### Segmentação de dados:

**Último ano:** Para segmentar as datas e fazer a puxada do ultimo ano utilizando o range de data, utilizando a tabela dCalendário

**Últimos 30 dias:** Para segmentar a data com o range dos últimos 30 dias, utilizando a tabela dCalendário

**Últimos 10 dias:** Para segmentar a data com o range dos últimos 10 dias, utilizando a tabela dCalendário

**Regiões e Países:** Para segmentar as regiões e países utilizando as tabelas de região e país, utilizando as tabelas de dPaís e dRegião

**Canais de distribuição:** Para segmentar os canais de distribuição utilizando a tabela dCVendas

**Cartões:** Irá conter todos os valores acumulados, quantidade de pedidos e valores em \$, utilizando a tabela fato de pedidos

**Gráfico de colunas:** irá juntamente com a segmentação de dados trazer o valor de venda acumulada nos últimos 10 dias através da tabela fato de pedidos

## Mapa e filtros





## Indicadores e filtros

