# Análise de dados do Departamento de LOGISTICA

## Etapa 1 - Exemplos de problemas de negócio envolvendo o departamento de LOGISTICA

## Etapa 2 - Definição da Baseline

## Etapa 3 - Planejamento

Essa etapa tem como objetivo estabelecer a abordagem do projeto, incluindo a decomposição dos problemas, a entrega do produto final, as etapas do processo e as ferramentas a serem utilizadas. Para alcançar essa definição, é necessário realizar reuniões diárias e/ou especificas, em conjunto com os demais membros da equipe de análise de dados, por esse motivo, será adotado um processo de planejamento simplificado para este exemplo de estudo, descrito no arquivo “planejamento.txt.”

## Etapa 4 - Coleta de Dados Brutos de Fontes Confiáveis

Os dados utilizados no projeto são dados primários e foram coletados diretamente do cliente e no formato xlsx. Encontram-se na pasta \_datasets.

## Etapa 5 e 6 - Limpeza e Processamento de Dados e Análise Exploratória de Dados

**Etapas realizadas no Power Query**

#### Tabela Calendário

A tabela DIMCALENDARIO foi criada realizando as etapas abaixo:

1. No Power BI, entre no Power Query, através do Transformar dados.
2. Adicione uma nova fonte > Consulta nula > Renomeie a consulta para DIMCALENDÁRIO.
3. Adicione a fórmula na Barra de fórmula para fazer uma lista de datas:

= List.Dates(#date(2018, 1, 1),Duration.Days(#date(2019, 12, 31) - #date(2018, 1, 1))+1,#duration(1,0,0,0))

1. Na aba transformar, clique em converter para a tabela (Criando uma tabela a partir de uma lista de valores) > Abrirá uma janela > Clique OK
2. Formate a coluna como Data e renomeie o nome da coluna para Data
3. Na aba adicionar coluna > Clique na data e acrescente as colunas: Ano, mês, nome do mês, início do mês e dia > Salvar e fechar (Voltar para o Power BI)
4. Relacionar a tabela DIMCALENDÁRIO com a tabela FATOESTOQUE

Colunas criadas a partir da coluna da Data:

* Data = CALENDAR (DATE (2018,01,01), DATE (2019,12,31))
* Ano = YEAR (TabelaCalendário[Data]) ou Ano = Date.Year([Data])
* Mês = MONTH(TabelaCalendário[Data]) ou Mês = Date.Month([Data])
* Nome do mês = FORMAT(TabelaCalendário[Data], “mmmm”) ou Nome do mês = Date.MonthName([Data])
* Inicio do mês = STARTOFMONTH(TabelaCalendário[Data]) ou Início do mês = Date.StartOfMonth([Data])
* Dia = DAY (TabelaCalendário[Data]) ou Dia = Date.Day([Data])
* Mês/Ano = FORMAT(TabelaCalendário[Data], “mmm/yyyy”)

#### Tabela

A tabela DIMESTOQUEMÍNIMO foi importada do arquivo CSV e teve as seguintes etapas aplicadas: Alteração da primeira linha como Cabeçalho e alteração do tipo de dados das suas 2 colunas, permanecendo assim:

* ID Produto = Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***
* Estoque mínimo = Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***

#### Tabela

A tabela DIMLOJA foi importado do arquivo Excel e teve as seguintes etapas aplicadas:

1. Alteração do tipo de dados
2. Alteração dos nomes das colunas ID Loja e Loja para ID Shopping e Shopping
3. Divisão da coluna Bairro em 2 colunas utilizando o delimitador “-”
4. Alteração dos nomes das 2 colunas para Bairro e Estado

Permanecendo assim:

* ID Loja = Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***
* Shopping = Sem modificação nos dados ***(Alteração no Tipo de Dado: Texto)***
* Bairro = Os dados foram divididos em 2 colunas: Uma coluna Bairro e Uma coluna Estado ***(Alteração no Tipo de Dado: Texto)***
* Estado = Sem modificação nos dados ***(Alteração no Tipo de Dado: Texto)***

#### Tabela

A tabela DIMPRODUTO foi importada do arquivo Excel e teve apenas uma etapa aplicada: Alteração do tipo de dado, permanecendo assim:

* ID Produto: Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***
* Produto: Sem modificação nos dados ***(Alteração no Tipo de Dado: Texto)***
* Categoria: Sem modificação nos dados ***(Alteração no Tipo de Dado: Texto)***
* Subcategoria: Sem modificação nos dados ***(Alteração no Tipo de Dado: Texto)***
* Imagem: Sem modificação nos dados ***(Alteração no Tipo de Dado: Texto e Número)***
* Custo Unit: Sem modificação nos dados ***(Alteração no Tipo de Dado: Decimal)***
* Preço Unit: Sem modificação nos dados ***(Alteração no Tipo de Dado: Decimal)***

#### Tabela

A tabela FATOESTOQUE foi importada do arquivo Excel e teve as seguintes etapas aplicadas:

1. Alteração dos tipos de dados
2. Substituição de S para Saída
3. Substituição de E para Entrada
4. A coluna ID Loja foi renomeada para ID Shopping
5. A coluna Número Movimentação foi excluída

Permanecendo assim:

* ID Produto: Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***
* Data: Sem modificação nos dados ***(Alteração no Tipo de Dado: Data)***
* ID Shopping: Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***
* Movimentação: Sem modificação nos dados ***(Alteração no Tipo de Dado: Inteiro)***
* Tipo: Substituições dos valores S para Saída e E para Entrada ***(Alteração no Tipo de Dado: Texto)***

#### Parâmetro

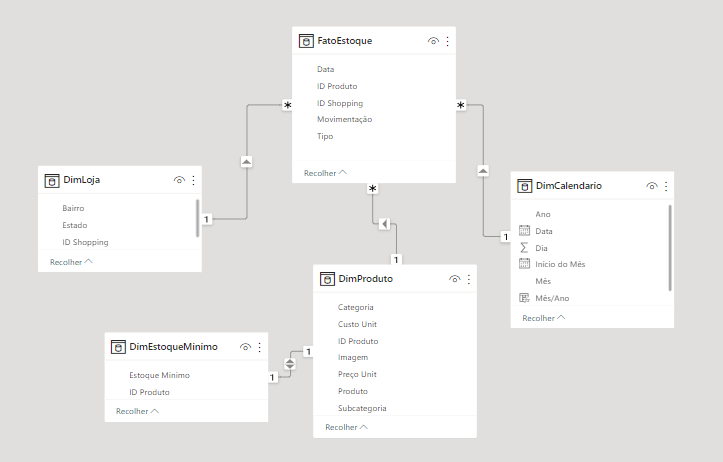
Criação do parâmetro CAMINHO com o caminho das pastas onde se encontra o arquivo em Excel (Base Logistica.xlsx) e a sua substituição no código Fonte. Assim a conexão dos arquivos com o arquivo em Power BI pode ser atualizada com mais facilidade, caso ocorra alguma modificação de pastas.

## Etapa 7 - Preparação dos Dados (Modelagem dos dados)

**Etapas realizadas no Power BI**

Na exibição do modelo, foi realizado as seguintes relações:

* Tabela DimLoja com a tabela FatoEstoque de um para muitos, utilizando a coluna ID Shopping como chave primaria.
* Tabela DimProduto com a tabela FatoEstoque de um para muitos, utilizando a coluna ID Produto como chave primaria.
* Tabela DimEstoqueMinimo com a tabela DimProduto de um para um, utilizando a coluna ID Produto como chave primaria.
* Tabela DimCalendario com a tabela FatoEstoque de um para muitos, utilizando a coluna data como chave primaria.



#### Tabela Medida

A tabela MEDIDA foi criada para armazenar as medidas criadas para serem utilizadas nos Dashboards:

* % de produtos abaixo do estoque: Medida calcula quantos por cento dos produtos estão abaixo do estoque em relação ao seu total = DIVIDE([Quantidade de produtos abaixo do estoque],[Quantidade de produtos distintos])
* % Margem de lucro: Medida calcula a margem de lucro de todos os produtos, ou seja, para cada 1 real vendido quanto de lucro esse produto gera = DIVIDE([Lucro Total], [Receita Total])
* Custo Total: Medida calcula o total da quantidade de entrada realizada de cada produto multiplicada pelo seu preço unitário, ou seja, somatório de cada quantidade compra multiplicada pelo seu preço = CALCULATE(SUMX(FatoEstoque, FatoEstoque[Movimentação] \* RELATED(DimProduto[Custo Unit])), FatoEstoque[Tipo] = "Entrada")
* Lucro Total: Medida calcula o lucro que cada produto obteve ao realizar a subtração da Receita total com o Custo Total = [Receita Total] - [Custo Total]
* Quantidade de estoque: Medida calcula o total da quantidade de saída e de entrada dos produtos que ocorreram em 2018 e 2019. O resultado pode ser positivo ou negativo. Se for positivo significa que ocorreram mais compras do que vendas de produtos durante este período. = SUM(FatoEstoque[Movimentação])
* Quantidade de estoque mínimo: Medida calcula o total da quantidade mínima que cada produto precisa ter = SUM(DimEstoqueMinimo[Estoque Mínimo])
* Quantidade de produtos abaixo do estoque: Medida calcula quantos produtos estão abaixo da quantidade mínima permitida = COUNTX(FILTER(DimProduto,[Quantidade de estoque] - [Quantidade de estoque mínimo] < 0),DimProduto[ID Produto])
* Quantidade de produtos distintos: Medida calcula quantos produtos únicos existem no armazenados no estoque = DISTINCTCOUNT(DimProduto[ID Produto])
* Quantidade vendida: Medida calcula a quantidade de saída realizada em 2018 e 2019, ou seja, a quantidade vendida deste período = COUNTX(FILTER(FatoEstoque,FatoEstoque[Tipo] = "Saída"), FatoEstoque[Data])
* Rank produto: Medida calcula a classificação de cada produto em relação a sua quantidade vendida = RANKX(ALL(DimProduto), [Quantidade vendida]) & "º mais vendido"
* Receita total: Medida calcula o total da quantidade de saída realizada de cada produto multiplicada pelo seu preço unitário, ou seja, somatório de cada quantidade vendida multiplicada pelo seu preço = SUMX(FILTER( FatoEstoque,FatoEstoque[Tipo] = "Saída"), - FatoEstoque[Movimentação] \* RELATED(DimProduto[Preço Unit]))
* Saldo Estoque: Medida calcula a quantidade resultante de cada produto armazenado no estoque = [Quantidade de estoque] - [Quantidade de estoque mínimo]

Funções DAX utilizadas:

* A função **DIVIDE (<numerator>, <denominator> [, <alternateresult>])** usada para dividir um numerador por um denominador
* A função **SUMX(<TABLE>, <EXPRESSION>)** retorna a soma de uma expressão avaliada para cada linha de uma tabela.
* A função **CALCULATE(<EXPRESSION>, [<FILTER1>], [<FILTER2>] …)** avalia uma expressão em um contexto de filtro modificado.
* A função **RELATED(<column>)** retorna um valor relacionado de outra tabela.
* A função **SUM (<COLUMN>)** adiciona todos os números de uma coluna.
* A função **COUNTX(<table>, <expression>)** conta o número de linhas que contêm um valor que não esteja em branco ou uma expressão que é avaliada como um valor que não esteja em branco, ao avaliar uma expressão em uma tabela.
* A função **FILTER(<table>, <filter>)** retorna uma tabela que representa um subconjunto de outra tabela ou expressão.
* A função **DISTINCTCOUNT(<column>)** conta o número de valores distintos de uma coluna.
* A função **RANKX(<table>, <expression> [, <value> [, <order> [, <ties>]]])** retorna à classificação de um número em uma lista de números para cada linha no argumento table.
* A função **ALL([<table> | <column> [, <column> [, <column> […]]]])** retorna todas as linhas de uma tabela ou todos os valores de uma coluna, ignorando todos os filtros que estiverem aplicados. Essa função é útil para limpar filtros e criar cálculos em todas as linhas em uma tabela.

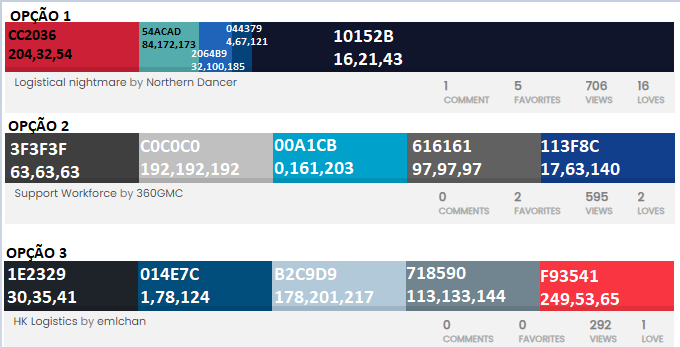
## Etapa 8 - (Pré) Validação dos resultados com os clientes

Essa etapa de pré validação dos resultados com os clientes, para realizar correções e últimos ajustes, não foi feita, pois se trata de um projeto fictício com objetivo de demonstrar as habilidades de análise de dados utilizando Excel e Power BI.

## Etapa 9 - Storytelling dos processos e resultados

**Etapas realizadas no Power BI**

1. Definição das cores, em RGB a serem utilizadas no Dashboards



1. Criação das 4 guias: Inventário, Análise de Estoque, Resultado e Análise por Produto
2. Criação das 4 telas de fundo no PowerPoint
3. Criação da tabela \_Medida
4. Criação das medidas, utilizando a linguagem DAX
   1. % de produtos abaixo do estoque
   2. % Margem de lucro
   3. Custo Total
   4. Lucro Total
   5. Quantidade de estoque
   6. Quantidade de estoque mínimo
   7. Quantidade de produtos abaixo do estoque
   8. Quantidade de produtos distintos
   9. Quantidade vendida
   10. Rank produto
   11. Receita total
   12. Saldo Estoque
5. Criação da segmentação de dados para a guia Análise por Produtos: Filtro Produto
6. Criação dos gráficos e cartões de linhas múltiplas

**DASHBORDS:**

Em conclusão, este projeto demonstra como podemos utilizar o Microsoft Power BI para extrair e analisar dados de um arquivo do Microsoft Excel. A análise fornece insights sobre a movimentação do estoque, um controle do inventário, o retorno financeiro de cada produto e entre outros, que podem ser utilizados ​​para ajudar nas decisões internas de comprar ou não mais produtos dependendo de sua quantidade armazenada, sobre a continuidade ou não de determinado produto devido ao seu retorno baixo e identificação de quais produtos trazem um retorno financeiro satisfatório.

**Exemplos de insights:**