

Medidas rápidas

The screenshot shows the Power BI desktop interface. At the top, the ribbon has tabs: Arquivo, Página Inicial (selected), Inserir, Modelagem, Exibição, and Ajuda. The 'Calculos' (Calculus) tab is highlighted with a yellow box. Below the ribbon, there's a toolbar with icons for Recortar (Cut), Colar (Paste), Pincel de formatação (Format brush), Obter dados (Get Data), Excel, BI Server, Dados, Fontes recentes (Recent Sources), Transformar dados (Transform Data), Atualizar (Update), Consultas (Queries), Novo visual (New Visual), Caixa de texto (Text Box), Mais visuais (More Visuals), and Nova medida rápida (New Quick Measure). A blue arrow points from the 'Nova medida rápida' icon in the ribbon to the 'Novas medidas rápidas' (New Quick Measures) option in the context menu of the 'fVendas' field in the 'Valores' (Values) pane.

Medidas rápidas

Cálculo: Média por categoria

Campos: Pesquisar

Agregar por categoria: Média por categoria (selected)

Filtros: Valor filtrado, Diferença do valor filtrado, Diferenca_percentual do valor filtrado

OK Cancelar

Medidas rápidas são cálculos “pré-moldados” que permitem você arrastar e soltar campos ao invés de escrever uma função DAX do zero na barra de fórmulas

Essa pode ser uma opção útil para medidas mais complexas, como inteligência de tempo, porém não te ajuda a compreender os fundamentos da linguagem DAX.

Sugestão: não utilizar, você pode mais que isso! 😊

Funções de Agregação

Úteis para agregar valores – funcionam apenas com colunas numéricas e agregam **apenas UMA coluna por vez.**

SUM()

Adiciona todos os números de uma coluna.

=SUM(NomeColuna)

AVERAGE()

Retorna a média aritmética de todos os números de uma coluna.

=AVERAGE(NomeColuna)

MAX()

Retorna o maior valor numérico de uma coluna ou entre duas expressões escalares.

=MAX(NomeColuna)

MIN()

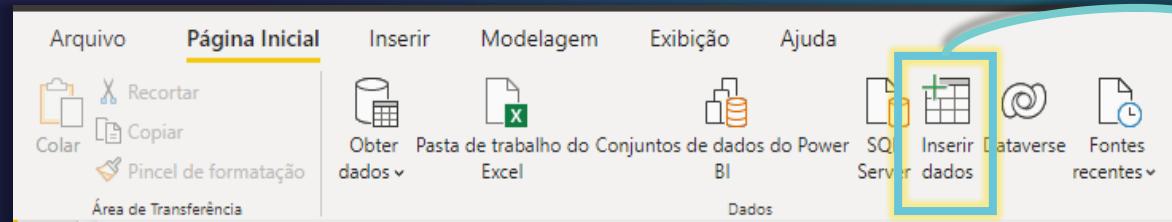
Retorna o menor valor numérico de uma coluna ou entre duas expressões escalares.

=MIN(NomeColuna)

Organizando as medidas

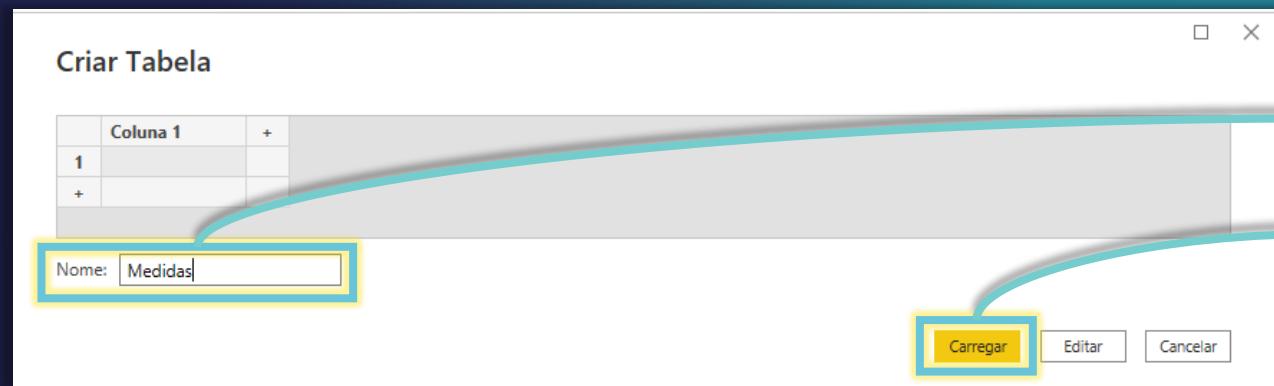
Uma **boa prática** interessante é criar uma **tabela separada para organizar/centralizar todas as suas medidas DAX** em um único local no seu modelo e elas não ficarem “espalhadas” entre tabelas:

1



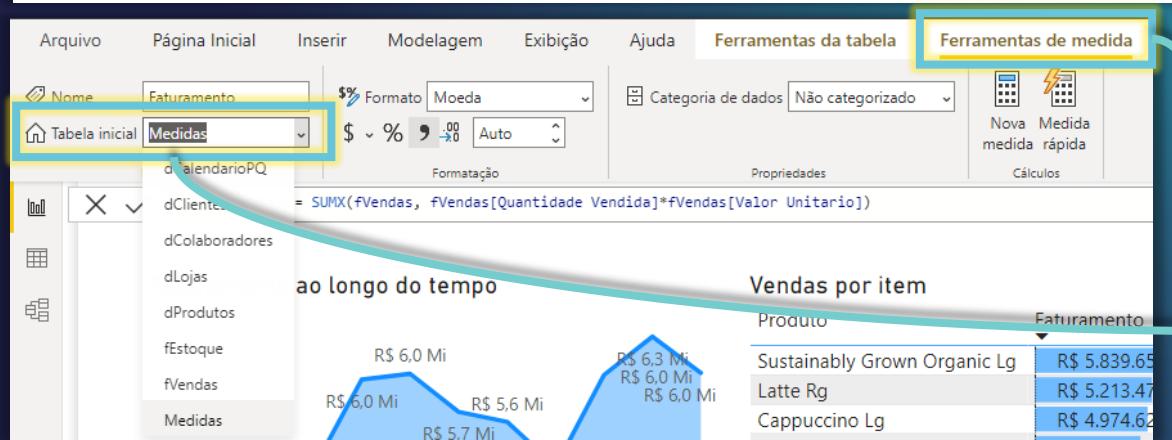
Inserir dados
para criar uma
nova tabela

2



Renomeie para
“Medidas”
Carregue para
o Power BI

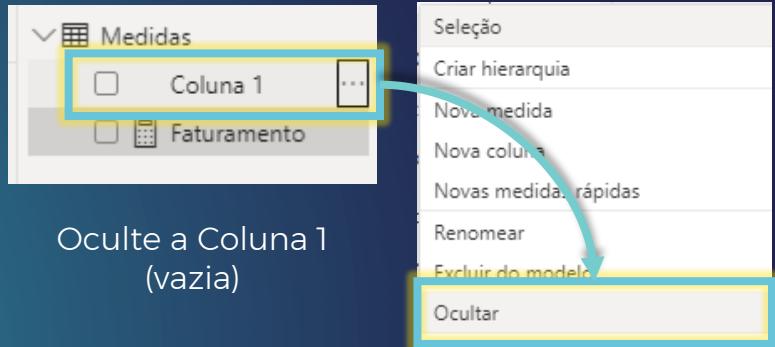
3



Selecionar as medidas já
criadas (uma de cada vez)
e depois vá para
Ferramentas da medida

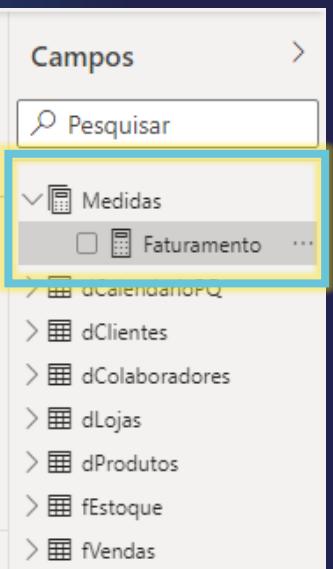
Altere a Tabela Inicial
para a tabela “Medidas”
criada anteriormente

4



Oculte a Coluna 1
(vazia)

Feito. Sua tabela
de medidas
ficará assim!
Note o símbolo
de calculadora



Funções Estatísticas

Diferenças entre as diferentes funções de contagem no Power BI

COUNT()

Conta o número de células de uma coluna que contém **números**

=COUNT(**NomeColuna**)

DISTINCTCOUNT()

Conta o número de **valores distintos/únicos** (sem repetições) de uma coluna.

=DISTINCTCOUNT(**NomeColuna**)

DISTINCTCOUNTNOBLANK()

Conta o número de **valores distintos/únicos** (sem repetições) de uma coluna, mas **sem considerar os valores vazios/em branco**

=DISTINCTCOUNTNOBLANK(**NomeColuna**)

COUNTROWS()

Conta o **número de linhas na tabela especificada** ou em uma tabela definida por uma expressão

=COUNTROWS(**Tabela**)

Funções Iteradoras

Funções iteradoras permitem que você percorra a mesma expressão a cada linha de uma tabela e no final traga um único valor agregado (ou seja: máx., mín., média).

Imagine a função adicionando uma **nova coluna temporária** à sua tabela, **calculando o valor para cada linha** individualmente (com base na expressão – que pode ser uma multiplicação, por exemplo) e, em seguida, aplicando uma agregação a essa nova coluna (como uma soma)

=SUMX(Tabela, Expressão)

Agregação a ser aplicada no final
(depois do cálculo a nível de linha)

SUMX()

COUNTX()

AVERAGEX()

RANKX()

MAX()/MIN()

Tabela em que a expressão/cálculo
será considerado

- fVendas
- fEstoque

Expressão a ser avaliada para cada uma das
linhas da tabela mencionada anteriormente

- [Faturamento Total] (medida)
- fVendas[PreçoUnitario] * [QuantidadeVendida]

```
1 Faturamento = SUMX([fVendas, fVendas[Quantidade Vendida]*fVendas[Valor Unitario]])
```



ALL

ALL()

Retorna todas as linhas de uma tabela,
ou todos os valores de uma coluna,
IGNORANDO qualquer filtro aplicado

=ALL(**NomeTabela ou NomeColuna**, [Coluna1], [Coluna2], ...)



Nome da tabela **ou** da coluna que você quer limpar os filtros impostos pelo visual ou segmentação de dados no relatório

- fVendas
- dProdutos[Categoria]

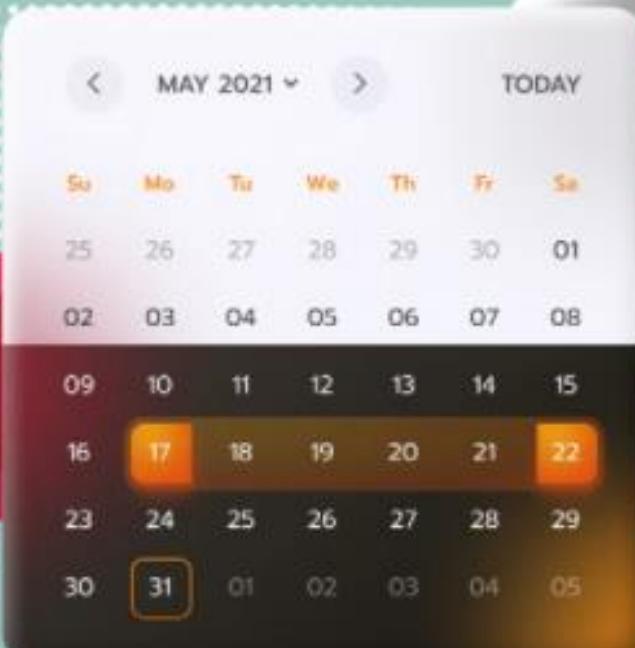
Opcional – lista de colunas adicionais que você quer remover o filtro

Obs: se o seu primeiro parâmetro for um NomeTabela, você não pode especificar colunas adicionais. Todas as colunas devem incluir o nome da tabela antes e devem vir de uma mesma tabela



Ao invés de adicionar um contexto de filtro, a função ALL remove os filtros daquele valor.
Ela é muito utilizada quando você precisa de valores que não irão reagir as mudanças de filtro no dashboard (ex: **% do total**, onde o denominador deve permanecer um valor “fixo”)

por que criar tabela calendário em projetos de BI



“

"Mas Letícia, se eu já tenho uma coluna com a data certinha na minha base de dados, por que devo criar uma tabela específica para calendário? Não vejo sentido nisso!"

Esse questionamento é muito comum, principalmente no início (atire a primeira pedra que nunca pensou isso, né?). Mas saiba que ter uma **tabela dimensão calendário em seus projetos é sim **algo fundamental!****

Existem 4 motivos principais (e que fazem total diferença nos resultados das suas análises!)



Arrasta para o lado para entender quais são e ver o passo a passo de como criar uma tabela calendário!

• Motivos para ter uma dCalendário

1

Para trabalhar com **funções de inteligência de tempo** (ex: SAMEPERIODLASTYEAR, ou fazer qualquer cálculo que envolve data/periódico), você precisa ter um range completo de datas.

Caso contrário, não vai funcionar! E somente com a tabela calendário conseguimos **garantir que não haverá valores repetidos e nem faltantes nas suas datas**, ou seja, nenhum "furo" ou redundância.

2

A dimensão calendário é uma forma de **permitir o relacionamento de mais de uma tabela fato** na sua modelagem de dados.

Vamos supor que você tem uma tabela fVendas e outra fEstoque. Você poderá relacioná-las com a dCalendário, utilizando a coluna de data existente em cada uma!

3

Trabalhar com **datas customizadas** (ano fiscal que não começa em Janeiro, feriados locais, dias úteis, etc).

4

Criar **filtros e segmentações de data** nos seus **dashboards** que se aplicam a todos os dados e visuais de forma consistente!



Existem duas formas mais recomendadas para criar uma tabela dimensão calendário no Power BI:

- Utilizando a linguagem DAX (funções CALENDAR ou CALENDARAUTO)
- Utilizando a linguagem M (dentro do Power Query Editor - guia "Transformar dados")

Não existe jeito melhor ou pior, eu normalmente escolho o seguinte:

CALENDARAUTO

Ideal quando não tenho tempo e quero praticidade



CALENDAR

A que eu mais utilizo.
Acho simples, fácil e me atende bem!

LING. M

Ótima quando preciso de flexibilidade para atender algum critério mais específico/customizado

dCALENDARIO em DAX

Como criar sua tabela calendário no Power BI

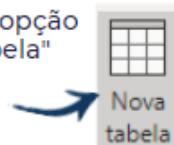
em DAX - CALENDAR AUTO

1 Vá até a guia "Dados", do lado esquerdo da tela



2 Em Pág. Inicial > Selecione a opção
"Nova Tabela"

Página Inicial



3 Na barra de fórmulas, digite:

```
1 dCalendario = CALENDAR(AUTO())
```

4 Aperte Enter e o Power BI já vai listar de modo automático todas as datas existentes no seu modelo (do primeiro ao último dia dos anos encontrados, sempre de jan-dez):

Date
01/01/2017 00:00:00
02/01/2017 00:00:00
03/01/2017 00:00:00
04/01/2017 00:00:00

5 Altere o formato para (dd/mm/yyyy)

Formato: dd/mm/yyyy

Formatos comuns: *14/03/2001 13:30:55 (General Date)
'mm/yyyy'
*14/03/2001 (dd/mm/yyyy)

6 Feito isso, vamos adicionar novas colunas para complementar essa tabela (uma coluna por vez, usando as fórmulas abaixo):

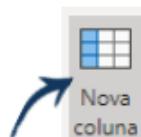
```
1 Ano = YEAR(dCalendario[Date])
```

```
1 Mês = MONTH(dCalendario[Date])
```

```
1 Dia = DAY(dCalendario[Date])
```

```
1 Nome do mês = FORMAT(dCalendario[Date], "mmm")
```

```
1 Mês abrev = FORMAT(dCalendario[Date], "mm")
```



4x a letra M ou D = traz nome completo do mês (M) ou dia (D)
3x a letra M ou D = 3 primeiras letras do mês (M) ou dia (D)

```
1 Dia da semana = FORMAT(dCalendario[Date], "ddd")
```

```
1 Trimestre = "T" & QUARTER(dCalendario[Date])
```

7

Prontinho! A sua tabela vai ficar assim:

Date	Ano	Mês	Dia	Nome do mês	Mês abrev	Dia da semana	Trimestre
01/01/2017	2017	1	1	janeiro	jan	dom	T1
02/01/2017	2017	1	2	janeiro	jan	seg	T1
03/01/2017	2017	1	3	janeiro	jan	ter	T1
04/01/2017	2017	1	4	janeiro	jan	qua	T1
05/01/2017	2017	1	5	janeiro	jan	qui	T1
06/01/2017	2017	1	6	janeiro	jan	sex	T1
07/01/2017	2017	1	7	janeiro	jan	sáb	T1
08/01/2017	2017	1	8	janeiro	jan	dom	T1
09/01/2017	2017	1	9	janeiro	jan	seg	T1
10/01/2017	2017	1	10	janeiro	jan	ter	T1
11/01/2017	2017	1	11	janeiro	jan	qua	T1
12/01/2017	2017	1	12	janeiro	jan	qui	T1
13/01/2017	2017	1	13	janeiro	jan	sex	T1
14/01/2017	2017	1	14	janeiro	jan	sáb	T1
15/01/2017	2017	1	15	janeiro	jan	dom	T1
16/01/2017	2017	1	16	janeiro	jan	seg	T1
17/01/2017	2017	1	17	janeiro	jan	ter	T1
18/01/2017	2017	1	18	janeiro	jan	qua	T1
19/01/2017	2017	1	19	janeiro	jan	qui	T1
20/01/2017	2017	1	20	janeiro	jan	sex	T1
21/01/2017	2017	1	21	janeiro	jan	sáb	T1



- Lembrando que você pode **personalizar e adicionar mais colunas** para atender outros critérios de negócio
- Não esqueça de ordenar o Nome do mês e Mês Abrev** pela coluna Mês (numérica)
- Por fim, **crie o relacionamento da tabela calendário** com as suas demais tabelas utilizando a coluna DATA.

dCALENDARIO em DAX

Qual é a diferença entre as duas funções para criar uma tabela dimensão Calendário utilizando a linguagem DAX?

CALENDAR AUTO()

Retorna uma tabela com apenas uma coluna chamada "Date" que contém um conjunto contíguo de datas. **O intervalo de datas é calculado automaticamente** com base nos dados no modelo.

=CALENDAR AUTO()

CALENDAR()

Retorna uma tabela com apenas uma coluna chamada "Date" que contém um conjunto contínuo de datas. O intervalo de datas é da **data de início especificada até a data de término especificada**.

=CALENDAR (DataInicio, DataFim)

Função FORMAT para datas

Código	Descrição	Exemplo
dddd	nome completo do dia	segunda-feira, terça-feira, quarta-feira ...
ddd	primeiros 3 dígitos do dia	seg, ter, qua, qui, sex ...
mmmm	nome completo do mês	janeiro, fevereiro, março, abril ...
mmm	primeiros 3 dígitos do mês	jan, fev, mar, abr, mai, jun, jul ...
mm	núm. do mês com dois digitos	01, 02, 03, 04, 05 ...
q	quarter (= trimestre)	1, 2, 3, 4
yyyy	ano com quatro digitos	2018, 2019, 2020, 2021 ...

Funções de Texto

LEN()

Retorna o número de caracteres em uma cadeia de texto.

=LEN(Texto)

CONCATENATE()

Une duas cadeias de texto em uma única palavra.

=CONCATENATE(Texto1, Texto2)

LEFT/MID/RIGHT()

Retorna o número especificado de caracteres do início/meio/fim de uma cadeia de texto.

=LEFT/RIGHT(Texto, [NumCaracteres])
=MID(Texto, PosiçãoInicial, NumCaract)

UPPER/LOWER/PROPER()

Converte todas as letras de um texto em maiúsculas (UPPER), minúsculas (LOWER) ou somente as primeiras letras como maiúsculas (PROPER)

=UPPER/LOWER/PROPER(Texto)

SUBSTITUTE()

Substitui o texto original pelo novo texto em uma cadeia de texto.

=SUBSTITUTE(Texto, TextoAntigo, NovoTexto)

VALUE()

Converte em número uma cadeia de texto que representa um número.

=VALUE(Texto)

CALCULATE

CALCULATE()

Avalia uma expressão em um contexto de **filtro** modificado

=CALCULATE(Expressão, [Filtro1], [Filtro2], ...)

Nome de uma medida existente ou uma outra fórmula DAX válida

- [Total de orders]
- SUM([Valor Venda])

Lista de filtros desejados

- Categoria = Café
- Loja = 1
- Preço atacado >= 12

A função CALCULATE funciona como a “SOMASE ou COUNT.SE” no Excel. A diferença é que aqui podemos criar medidas baseado em QUALQUER tipo de cálculo (não somente Soma, Contagem, etc), facilita pensar a função como “CALCULATESE”

Você calcula algo de acordo com o critério de filtro especificado.

Funções de Inteligência de tempo

DATEADD()

Retorna o resultado daquela medida correspondente a um **período anterior**.

=CALCULATE(Medida, DATEADD(dCalendario[Data], -1, MONTH))

Intervalo de comparação



Pode alterar o período para DAY, MONTH, QUARTER ou YEAR

SAMEPERIODLASTYEAR()

Retorna o resultado daquela medida correspondente a um **ano anterior**.

=CALCULATE(Medida, SAMEPERIODLASTYEAR(dCalendario[Data]))

Exatamente mesmo resultado que:

=CALCULATE(Medida, DATEADD(dCalendario[Data], -1, YEAR))

DATESYTD()

Retorna o **resultado acumulado** daquela medida até a presente data/momento.

=CALCULATE(Medida, DATESYTD(dCalendario[Data]))

Utilize DATESQTD para quarters (trimestre) ou DATESMTD para meses

DATESINPERIOD()

Retorna o **resultado acumulado móvel** de um período especificado

Conhecido como
“running total”

=CALCULATE(Medida,
DATESINPERIOD(dCalendario[Data], MAX(dCalendario[Data]), -3, MONTH))

Intervalo de comparação



Pode alterar o período para DAY, MONTH, QUARTER ou YEAR



Funções Lógicas

As funções lógicas agem sobre uma expressão para retornar informações sobre os valores ou conjuntos dela. Funcionam com Datas, Textos, Números (e qualquer tipo de dado). Essas são as mais comuns:

IF()

Verifica uma condição e retorna um valor quando é VERDADEIRO; caso contrário, retorna um segundo valor.

=IF(Teste Logico, ValorSeVerdadeiro, ValorSeFalso)

IFERROR()

Avalia uma expressão e retorna um valor especificado se a expressão retorna um erro; caso contrário, retorna o valor da própria expressão – ideal para **combinar com outras funções**

=IFERROR(Valor, ValorSeErro)

SWITCH()

Avalia uma expressão em relação a uma lista de valores e retorna uma das **várias expressões** de resultado Possíveis.

=SWITCH(Expressão, Valor, Resultado (...), Valor, Resultado (...), Outro/Se não)

= SWITCH([Mês], 1, "Janeiro", 2, "Fevereiro", 3, "Março", 4, "Abril", 5, "Maio", 6, "Junho", 7, "Julho", 8, "Agosto", 9, "Setembro", 10, "Outubro", 11, "Novembro", 12, "Dezembro", "Mês desconhecido")

Ideal para substituir o uso da função IF quando você tem uma situação com **muitas condições** (é possível combiná-la com a expressão TRUE (verdadeiro))

Funções de Relação

As funções nessa categoria são para o gerenciamento e a utilização de relações entre tabelas.

RELATED()

Essa função não realiza nenhuma operação matemática, apenas “abre a porta” para acessarmos colunas que estão em outra tabela relacionada no modelo.

Retorna um valor relacionado de outra tabela (traz valor do lado 1 – da tabela DIMENSÃO)

RELATEDTABLE()

Essa função normalmente é utilizada com outros agregadores, como COUNTROWS, SUMX, AVERAGEX, etc

Retorna uma tabela relacionada, filtrada apenas com as linhas que possuem conexão (lado muitos - da tabela FATO)

USERELATIONSHIP()

A função não retornará nenhum valor, apenas habilitará a relação indicada durante o cálculo. Interessante para quando temos múltiplas tabelas de data no modelo (em tabelas diferentes) conectadas a uma única tabela dCalendário.

Força a ativação de um relacionamento inativo (---).

CROSSFILTER()

Ao invés de utilizar relacionamentos bi-direcionais, utilize a CROSSFILTER para ativar a direção desejada em um caso específico.

Modifica a direção (<->) do filtro – é mais seguro do que deixar como “Both” e evita ambiguidade no modelo

=RELATED(Coluna)

Nome da coluna eu você quer retornar valores (deve referenciar o lado “1” da tabela 1:*)

- ‘dProdutos’[Custo Unitario]
- ‘dClientes’[Loja]

=RELATEDTABLE(NomeTabela)

Nome da tabela eu você quer retornar valores (deve referenciar o lado “muitos” da tabela 1:*)

- COUNTROWS(RELATEDTABLE(‘fEstoque1’))

=USERELATIONSHIP(Coluna1, Coluna2)

São as colunas que contém as chaves primárias e secundárias/estrangeiras do seu modelo de dados

=CROSSFILTER(Coluna1, Coluna2, DireçãoFiltro)

OneWay, Both, None

DIVIDE

Função DIVIDE vs. operador (/)

DIVIDE()

Executa a divisão e retorna o resultado alternativo ou BLANK() na divisão por 0.

=DIVIDE(Numerador, Denominador, [Resultado Alternativo])

A função DIVIDE foi criada para lidar automaticamente com casos de divisão por zero. Se um resultado alternativo não for passado e o denominador for zero ou ficar EM BRANCO, a função retornará “EM BRANCO”. Quando um resultado alternativo é definido na função, ele será retornado, ao invés da palavra EM BRANCO (ou infinito).