

Relatório 4 – Iteradores

Variáveis

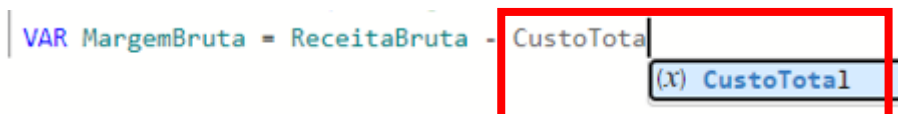
Como nas linguagens de programação, o DAX também é possível criar variáveis e para fazer isso é utilizando a palavra reservada VAR. Para mostrar vou fazer o cálculo da margem bruta usando variáveis.

```
1 Margem Bruta em % VAR =  
2   VAR ReceitaBruta = SUM(Livros[Receita Bruta])  
3   VAR CustoTotal = SUM(Livros[Custo Total])  
4   VAR MargemBruta = ReceitaBruta - CustoTotal
```

Criei três variáveis, os próprios nomes já especificam o que as variáveis querem dizer, uma para guardar a **receita bruta**, a segunda para o **custo total** e a terceira para fazer o cálculo da **receita total menos o custo total**.

Perceba que na linha 4, eu chamei as variáveis que declarei nas linhas 2 e 3, isso significa que eu posso chamar variáveis dentro de uma outra variável.

Para declarar uma variável no DAX, é uma boa prática usar letras maiúsculas para separar as palavras, como você pode ver nas três variáveis que criei. Isso se chama Camel Case.



```
VAR MargemBruta = ReceitaBruta - (x) CustoTotal
```

Perceba que quando uma variável é criada e o intellisense mostra essa variável recém criada, significa que ela está salva na memória e o seu ícone é a letra x, a letra fx significa função.

No DAX, declarar uma variável não é o bastante, quando declaramos uma variável e queremos usá-la em um cálculo, temos que usar outra palavra reservada, o **return**.

```
1 Margem Bruta em % VAR =  
2   VAR ReceitaBruta = SUM(Livros[Receita Bruta])  
3   VAR CustoTotal = SUM(Livros[Custo Total])  
4   VAR MargemBruta = ReceitaBruta - CustoTotal  
5   Return  
6   MargemBruta / ReceitaBruta  
7
```

Como eu quero saber a margem bruta, eu peço para retornar o resultado da divisão da margem bruta pela receita bruta, em seguida pressiono a tecla **enter**.

Um detalhe, posso criar várias variáveis e não usar todas, somente será processado as variáveis que eu chamar. O que realmente importa é o que eu quero que retorne.

Próxima Demanda – Lead time

O termo lead time se refere ao tempo necessário para a conclusão de uma atividade e a próxima demanda da Buscante é saber o lead time da entrega dos livros, em outras palavras, quanto tempo demorou para um livro ser entregue para um cliente que fez seu pedido na Buscante.

Mas fica a seguinte pergunta, como fazer isso? Preciso saber quanto tempo levou para o cliente receber o seu livro depois de ter feito um pedido na Buscante e na tabela **registro_vendas** temos duas colunas chamadas **data_compra** e **data_entrega**.

Para chegar nessa resposta preciso saber a diferença entre a data de compra e data de entrega e para fazer isso eu faço uma subtração entre essas duas colunas.

Crio esse cálculo através de uma coluna calculada.

```
LeadTime = registro_vendas[Data_Entrega] - registro_vendas[Data_Compra]
```

Depois de pressionar a tecla enter o cálculo foi feito, porém algo esquisito apareceu.

LeadTime
25/12/1899 00:00:00
28/12/1899 00:00:00
23/12/1899 00:00:00
28/12/1899 00:00:00
26/12/1899 00:00:00
27/12/1899 00:00:00

Não é isso o que eu quero, isso aconteceu pela razão que as colunas que mencionei no cálculo tem o tempo como domínio, ou seja, seu tipo é date/time. Para saber a diferença entre a compra e a entrega eu preciso fazer na coluna calculada transformar o date/time em int.

```
LeadTime = INT(registro_vendas[Data_Entrega] - registro_vendas[Data_Compra])
```

Para fazer isso no DAX eu coloco a função **INT** dentro do cálculo, essa função irá transformar o domínio data/time em inteiro.

Agora veja o resultado com a função aplicada. Agora sim é da maneira que quero.

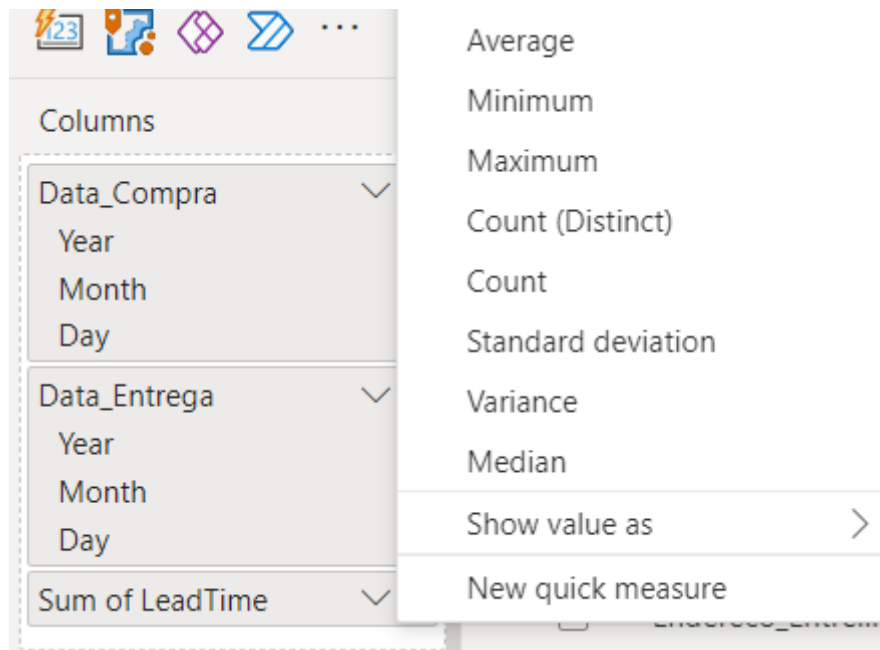
LeadTime
5
2
7
2
4

Em seguida coloquei, no canvas, uma tabela contendo a data de compra, data de entrega mais o lead time e o resultado foi o seguinte.

Year	Month	Day	Year	Month	Day	Sum of LeadTime
2022	May	31	2022	June	5	5
2022	June	1	2022	June	3	2
2022	June	1	2022	June	8	7
2022	June	2	2022	June	4	2
2022	June	2	2022	June	5	3
2022	June	2	2022	June	6	4
2022	June	3	2022	June	10	14
2022	June	4	2022	June	7	3
2022	June	4	2022	June	8	4
2022	June	4	2022	June	10	6
2022	June	4	2022	June	13	18
2022	June	5	2022	June	7	2
2022	June	5	2022	June	9	4
2022	June	5	2022	June	15	10
2022	June	6	2022	June	7	1
2022	June	6	2022	June	9	3
2022	June	6	2022	June	13	7
2022	June	6	2022	June	15	9
2022	June	6	2022	June	16	10
2022	June	7	2022	June	9	2
2022	June	7	2022	June	13	6
2022	June	7	2022	June	17	10
2022	June	8	2022	June	12	4
Total						5472

No lado esquerdo consta as datas de compras, no meio as datas de entrega e no lado esquerdo o lead time das entregas, mas veja no total o BI somou todas as diferenças

entre os pedidos e as entregas e isso não está no meu objetivo, o mais apropriado seria uma média de lead time.



Uma solução seria mudar a opção de **sum** para **average**, vou aplicar e ver o que acontece.

Year	Month	Day	Year	Month	Day	Average of LeadTime
2022	May	31	2022	June	5	5,00
2022	June	1	2022	June	3	2,00
2022	June	1	2022	June	8	7,00
2022	June	2	2022	June	4	2,00
2022	June	2	2022	June	5	3,00
2022	June	2	2022	June	6	4,00
2022	June	3	2022	June	10	7,00
2022	June	4	2022	June	7	3,00
2022	June	4	2022	June	8	4,00
2022	June	4	2022	June	10	6,00
2022	June	4	2022	June	13	9,00
2022	June	5	2022	June	7	2,00
2022	June	5	2022	June	9	4,00
2022	June	5	2022	June	15	10,00
2022	June	6	2022	June	7	1,00
2022	June	6	2022	June	9	3,00
2022	June	6	2022	June	13	7,00
2022	June	6	2022	June	15	9,00
2022	June	6	2022	June	16	10,00
2022	June	7	2022	June	9	2,00
2022	June	7	2022	June	13	6,00
2022	June	7	2022	June	17	10,00
2022	June	8	2022	June	12	4,00
Total						5,47

Consegui mais ou menos o que eu queria, eu tenho a média total do lead time, porém essa alteração fez com que a coluna inteira fosse calculada, nesse caso não seria problema pois só tenho um número então a média será a mesma.

Outra solução seria criar uma medida para ver a média total do lead time.

```
Média do Lead Time = AVERAGE(registro_vendas[LeadTime])
```

No DAX existe uma função chamada AVERAGE e dentro dessa função eu coloco a coluna do qual eu quero saber a média que é a do lead time, depois de pressionar o enter eu tenho o resultado.

Média do Lead Time	
	5,00
	2,00
	7,00
	2,00
	3,00
	4,00
	7,00
	3,00
	4,00
	6,00
	9,00
	2,00
	4,00
	10,00
	1,00
	3,00
	7,00
	9,00
	10,00
	2,00
	6,00
	10,00
	4,00
	5,47

A medida trouxe os mesmos resultados.

Porém eu descobri que poderia ter feito tudo isso sem a necessidade de criar a coluna calculada que mostrei antes. Eu posso descobrir a média do lead time usando um conceito chamado **iterador**.

Iterador é um processo onde a coluna de uma tabela é avaliada linha a linha. Depois de analisar todas as linhas, a função utilizada agrega o resultado e o entrega como uma medida.

Uma medida que funciona com a iteração traz no final, um resultado agregado, tendo a capacidade de avaliar linha a linha da tabela.

No Power BI existem as funções iteradoras, e em sua grande maioria, tem que colocar um “x” no final do nome da função, como por exemplo, **SUM()** é **SUMX()**, **AVERAGE()** é **AVERAGEX()**, **MIN()** é **MINX()**.

É importante dizer que não são todas as funções iteradoras que tem que colocar o “x” no final, algumas são desse tipo, mas não tem o “x”.

```
1 Média Lead Time Iterador = AVERAGEX('registro_vendas',  
2 INT('registro_vendas'[Data_Entrega] - 'registro_vendas'[Data_Compra]))]
```

As funções iteradoras pedem no mínimo dois parâmetros. Como você pode ver, eu chamei a função **AVERAGEX()** e no primeiro parâmetro coloquei a tabela que será analisada linha a linha. Sim, o primeiro parâmetro você informa a tabela que será analisada.

```
1 Média Lead Time Iterador = AVERAGEX('registro_vendas',  
2 INT('registro_vendas'[Data_Entrega] - 'registro_vendas'[Data_Compra]))]
```

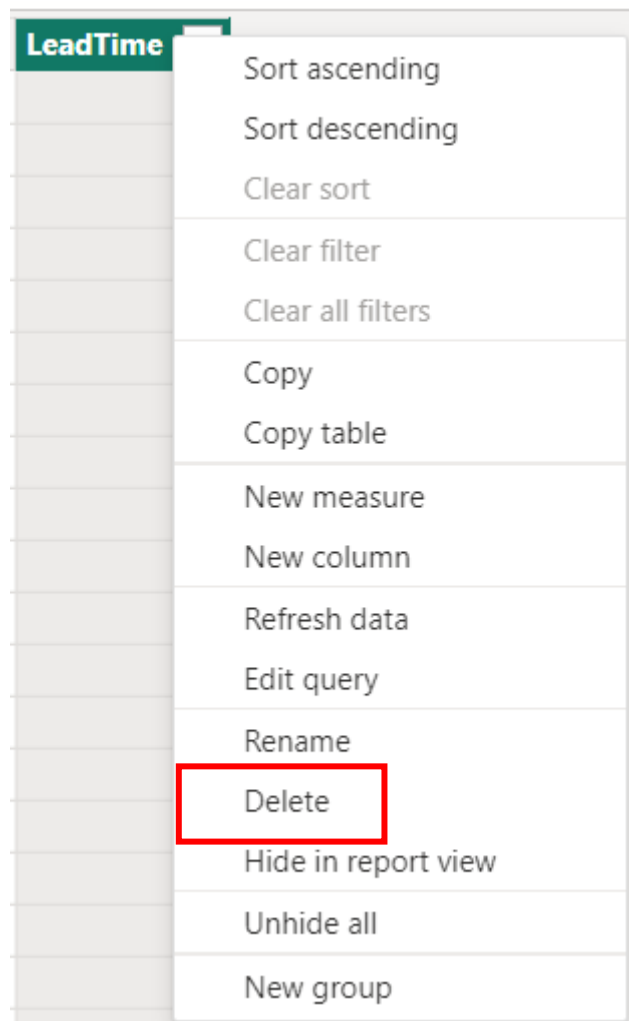
No segundo parâmetro eu coloco o cálculo para descobrir a diferença entre a data da entrega e a data da compra, eu fiz esse cálculo quando eu criei a coluna calculada.

Lembrando que eu uso a função INT, pois estou fazendo um cálculo cujo as colunas são do tipo date/time. Em seguida pressiono enter e coloco essa medida na tabela, e olha o resultado.

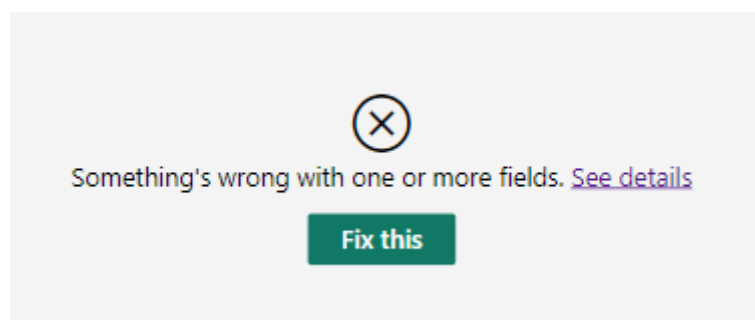
Average of LeadTime	Média do Lead Time	Média Lead Time Iterador
5,00	5,00	5,00
2,00	2,00	2,00
7,00	7,00	7,00
2,00	2,00	2,00
3,00	3,00	3,00
4,00	4,00	4,00
7,00	7,00	7,00
3,00	3,00	3,00
4,00	4,00	4,00
6,00	6,00	6,00
9,00	9,00	9,00
2,00	2,00	2,00
4,00	4,00	4,00

O que está destacado em vermelho é a medida feita com a função iteradora. Perceba que ela trouxe os mesmos resultados dos outros dois procedimentos e uma diferença importante, a medida iteradora não tem nenhuma ligação com a coluna calculada criada.

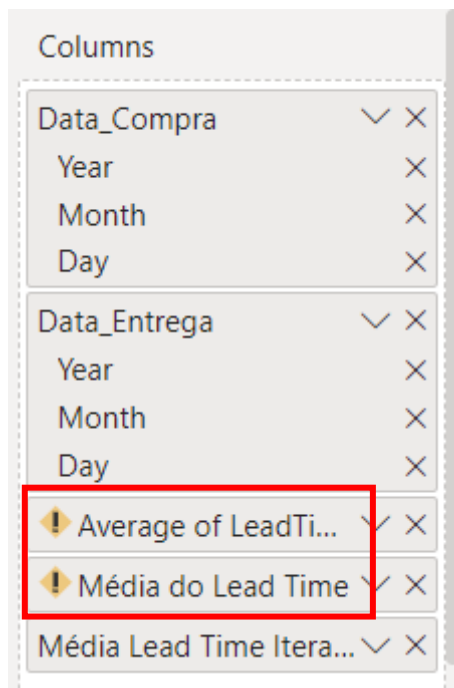
Em outras palavras, se eu apagar a coluna calculada, a medida feita com a função iteradora não será afetada, agora as outras medidas que tem como base a coluna calculada **Lead time** serão afetadas.



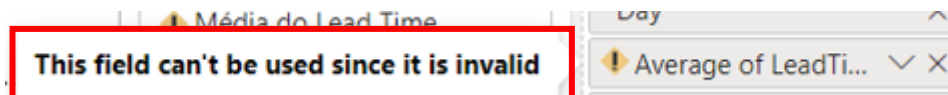
Excluí a tabela Lead time, veja o que aconteceu no Canvas.



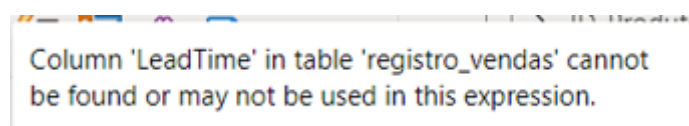
O BI já detectou que há algo de errado com a tabela e a resposta está na aba visualizations.



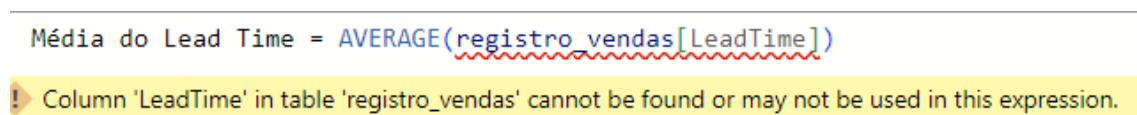
Vê esses dois itens destacados em vermelho? Que estão com um aviso em amarelo? Então se você arrastar o ícone do mouse e coloca-lo em cima de um desses dois itens irá aparecer o seguinte recado:



Ele diz que o campo não pode ser utilizado pois está inválido, mas por que? Na aba data possuí a resposta.



A resposta é simples, não podemos usar a medida pois a tabela que ela utiliza como referência não existe mais, aquela tabela foi apagada.



O DAX também avisa para os usuários que a tabela não existe mais, portanto a medida não pode ser usada.

Apagando as medidas inúteis o visual volta ao normal, com a medida que criei usando a função de iteração.

Year	Month	Day	Year	Month	Day	Média Lead Time Iterador
2022	May	31	2022	June	5	5,00
2022	June	1	2022	June	3	2,00
2022	June	1	2022	June	8	7,00
2022	June	2	2022	June	4	2,00
2022	June	2	2022	June	5	3,00
2022	June	2	2022	June	6	4,00
2022	June	3	2022	June	10	7,00
2022	June	4	2022	June	7	3,00
2022	June	4	2022	June	8	4,00
2022	June	4	2022	June	10	6,00
2022	June	4	2022	June	13	9,00
2022	June	5	2022	June	7	2,00
2022	June	5	2022	June	9	4,00
2022	June	5	2022	June	15	10,00
2022	June	6	2022	June	7	1,00
2022	June	6	2022	June	9	3,00
2022	June	6	2022	June	13	7,00
2022	June	6	2022	June	15	9,00
2022	June	6	2022	June	16	10,00
2022	June	7	2022	June	9	2,00
2022	June	7	2022	June	13	6,00
2022	June	7	2022	June	17	10,00
2022	June	8	2022	June	12	4,00
2022	June	9	2022	June	14	5,00
2022	June	9	2022	June	16	7,00
2022	June	9	2022	June	18	9,00
Total						5,47