

Datalineage - TREATAS

TREATAS



Tabular

Una tabla completa o una tabla con una o más columnas

PBI ≥ Feb 2017 , SSAS ≥ 2012

La función **TREATAS** retorna la tabla proporcionado en su primer parámetro pero heredándole la información de la posición o data lineage de las columnas del su segundo parámetro en adelante, renovando así su etiqueta para que se comporte como una columna en el modelo, de allí su nombre: “tratar a una columna como si se tratará de otra”.

- **SINTAXIS:**

TREATAS (<Expresión> ; <Nombre de Columna> ; [<Nombre de Columna>] ...)

La razón por la que es importante aprender el linaje de datos es porque uno tiene la opción de **mantener o cambiar el linaje de datos a voluntad**. En algunos escenarios, es importante mantener el linaje de los datos, mientras que en otros escenarios uno puede querer cambiar el linaje de una columna.

La función que puede cambiar el linaje de una columna es **TREATAS**. TREATAS acepta una tabla como primer argumento y luego un conjunto de referencias de columna. TREATAS actualiza el linaje de datos de la tabla etiquetando cada columna con la columna de destino adecuada. Por ejemplo, la medida de prueba anterior se puede reescribir de esta manera:

```
Test :=  
CALCULATE (  
    [Sales Amount],  
    TREATAS ( { "Red", "Blue" }, 'Product'[Color] )  
)
```

TREATAS devuelve una tabla que contiene valores etiquetados con la columna *Producto* [*Color*]. Como tal, esta nueva versión de la medida de prueba solo filtra los colores rojo y azul, como se muestra en la Figura.

TREATAS actualiza el linaje de la tabla anónima, por lo que el filtrado ahora funciona como se esperaba.

Color	Sales Amount	Test
Azure	97,389.89	3,545,546.72
Black	5,860,066.14	3,545,546.72
Blue	2,435,444.62	3,545,546.72
Brown	1,029,508.95	3,545,546.72
Gold	361,496.01	3,545,546.72
Green	1,403,184.38	3,545,546.72
Grey	3,509,138.09	3,545,546.72
Orange	857,320.28	3,545,546.72
Pink	828,638.54	3,545,546.72
Purple	5,973.84	3,545,546.72
Red	1,110,102.10	3,545,546.72
Silver	6,798,560.86	3,545,546.72
Silver Grey	371,908.92	3,545,546.72
Transparent	3,295.89	3,545,546.72
White	5,829,599.91	3,545,546.72
Yellow	89,715.56	3,545,546.72
Total	30,591,343.98	3,545,546.72

Ejercicio 1:

```

1 FiltroCB01 =
2     CALCULATE(
3         [Ingresos Tot],
4         FILTER(
5             ALL(Pedidos[SKU]),
6             Pedidos[SKU] = "CB01"
7         )
8     )

```

```

1 FiltroRowCB01 =
2     CALCULATE(
3         [Ingresos Tot],
4         ROW("SKU", "CB01")
5     )

```

```

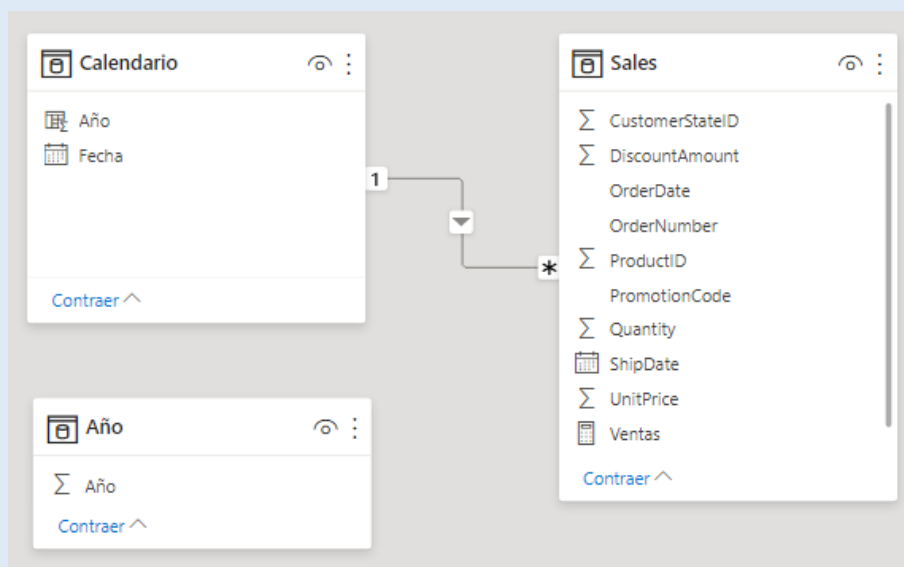
1 FiltroTreatasCB01 =
2     CALCULATE(
3         [Ingresos Tot],
4         TREATAS(
5             ROW("SKU", "CB01"),
6             SKUProductos[SKU]
7         )
8     )

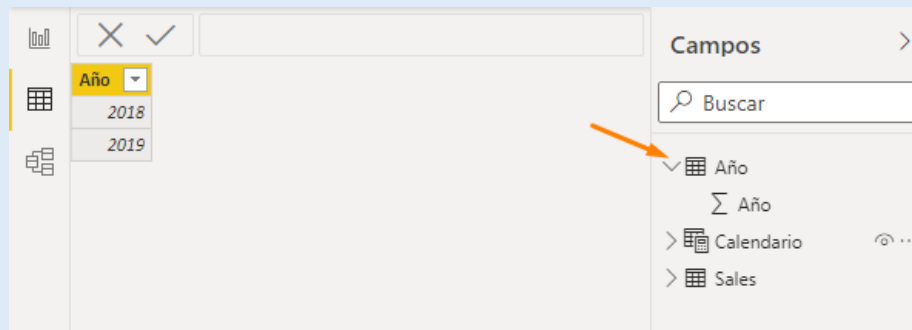
```

País	Ingresos Tot	FiltroCB01	FiltroRowCB01	FiltroTreatasCB01
Argentina	381.278,26	219.831,90	381.278,26	219.831,90
Brasil	362.302,68	203.978,96	362.302,68	203.978,96
Chile	347.945,91	192.222,47	347.945,91	192.222,47
Colombia	372.727,32	212.427,20	372.727,32	212.427,20
Ecuador	373.248,51	212.060,85	373.248,51	212.060,85
Paraguay	375.095,05	213.082,19	375.095,05	213.082,19
Perú	374.136,88	211.472,47	374.136,88	211.472,47
Uruguay	353.314,77	197.773,22	353.314,77	197.773,22
Venezuela	380.170,07	222.385,25	380.170,07	222.385,25
Total	3.320.219,45	1.885.234,53	3.320.219,45	1.885.234,53

Ejercicio 2:

Aplica el resultado de una expresión de tabla como filtros en las columnas de una tabla no relacionada.





1 Ventas = SUMX(Sales, Sales[Quantity]*Sales[UnitPrice])

Año	Ventas
2018	\$9.095.697
2019	\$42.270.311
2020	\$69.230.463

1 Ventas 2018 y 2019 = CALCULATE([Ventas], Calendario[Año] IN {2018,2019})

Año	Ventas	Ventas 2018 y 2019
2018	\$9.095.697	\$51.366.009
2019	\$42.270.311	\$51.366.009
2020	\$69.230.463	\$51.366.009

1 Ventas 2018 y 2019 2 = CALCULATE([Ventas], TREATAS({2018,2019}, Calendario[Año]))

Año	Ventas	Ventas 2018 y 2019	Ventas 2018 y 2019 2
2018	\$9.095.697	\$51.366.009	\$51.366.009
2019	\$42.270.311	\$51.366.009	\$51.366.009
2020	\$69.230.463	\$51.366.009	\$51.366.009

“Tratar” la lista la tabla de una sola columna AÑO “como” el campo “Año” de la tabla Calendario

1 Ventas 2018 y 2019 3 = CALCULATE([Ventas], TREATAS('Año', Calendario[Año]))

Año	Ventas	Ventas 2018 y 2019	Ventas 2018 y 2019 2	Ventas 2018 y 2019 3
2018	\$9.095.697	\$51.366.009	\$51.366.009	\$51.366.009
2019	\$42.270.311	\$51.366.009	\$51.366.009	\$51.366.009
2020	\$69.230.463	\$51.366.009	\$51.366.009	\$51.366.009

Ejercicio 3:

Se optimizo el ejercicio 2 de la función LASTNONBLANK.

```
1 UltimoBalanceOptimizado =
2 VAR Filtro =
3     ADDCOLUMNS(
4         VALUES(Balance[Nombre]),
5         "@Fecha", CALCULATE(MAX(Balance[Fecha]))
6     )
7 VAR FiltroConEtiquetasActualizadas =
8     TREATAS(
9         Filtro,
10        Calendario[Fecha]
11    )
12 VAR Resultado =
13     CALCULATE(
14         SUM(Balance[Balance]),
15         FiltroConEtiquetasActualizadas
16     )
17 RETURN
18     Resultado
```

No hereda el Datalineage. Por tanto, hay que aplicar un TREATAS.

! En la función TREATAS, el número de columnas de salida (1) no coincide con el número de columnas de la tabla de entrada (2).

Por tanto:

```
1 UltimoBalanceOptimizado =
2 VAR Filtro =
3     ADDCOLUMNS(
4         VALUES(Balance[Nombre]),
5         "@Fecha", CALCULATE(MAX(Balance[Fecha]))
6     )
7 VAR FiltroConEtiquetasActualizadas =
8     TREATAS(
9         Filtro,
10        Balance[Nombre],
11        Calendario[Fecha]
12    )
13 VAR Resultado =
14     CALCULATE(
15         SUM(Balance[Balance]),
16         FiltroConEtiquetasActualizadas
17     )
18 RETURN
19     Resultado
```

Tabla con todas sus columnas

El primer argumento de TREATAS puede recibir una tabla de una o más columnas. El segundo argumento (*Balance[Nombre]*), se va a corresponder siempre con la primera columna de la tabla indicada en el primer argumento del TREATAS, en este caso, *Filtro*. Y así sucesivamente. En este ejemplo, nosotros necesitábamos modificar las fechas, pero dado que la columna de fechas se ubica como segunda columna en Filtro, debemos hacer referencia a las columnas que le anteceden.

Año	Bod	Karol	Lita	Total
2018	3800	3700	4000	11500
Trim. 1	3800	3700	4000	11500
enero	1000	2600	2500	6100
febrero	700	400	850	1950
marzo	3800	3700	4000	11500
Total	3800	3700	4000	11500

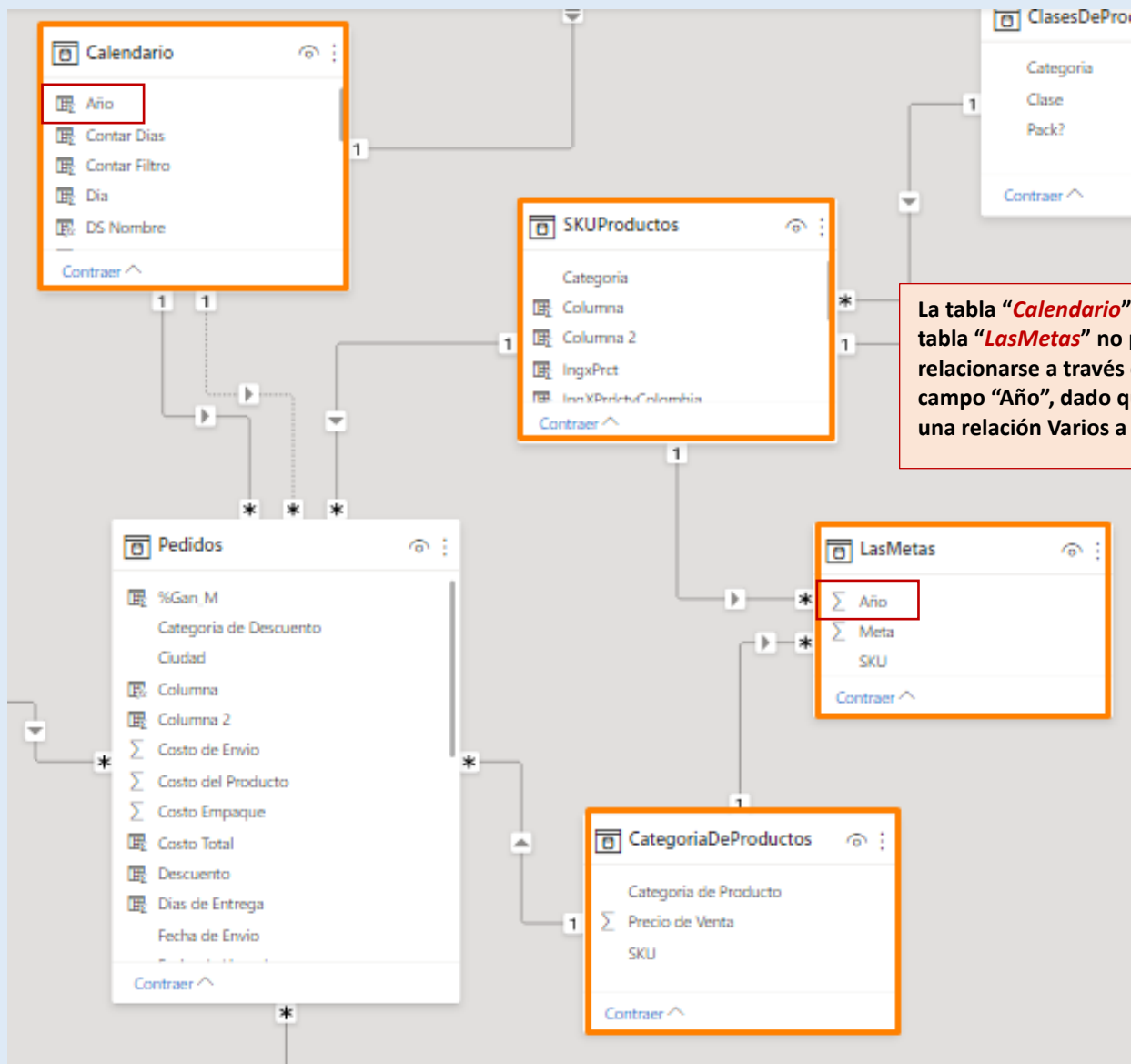
Ejercicio 4:

Ejercicio 1:

Para cada SKU y para cada año se estableció una meta de ingresos reales de acuerdo a diversos análisis respecto a periodos previos y respecto al mercado, estos datos esta registrados en la tabla *Metas*.

Para un *Bullet Chart* (by OKVIZ) situar los SKU en el área *Category* y la media *IngresosReales* en el área *Value*, adicionalmente, agregar una segmentación de datos a manera de lista desplegable donde el campo *Año* de la tabla de *Calendario* se encuentre seleccionado para el año 2004.

- a) Crear una medida que devuelva la meta establecida según el contexto y se puede ver su cambio dinámico en el área *Target* en el *Bullet Chart*. La tabla metas se encuentra en: <https://www.excellfreeblog.com/wp-content/uploads/2021/04/LasMetas.txt>



Podríamos explicarlo de esta manera: "Tome estos valores que retorna VALUES para filtrar la columna Año de la tabla LasMetas."

Tabla de una sola columna, por tanto, no hay problema de llamar en el segundo argumento de TREATAS solo a la columna que queramos modificar.

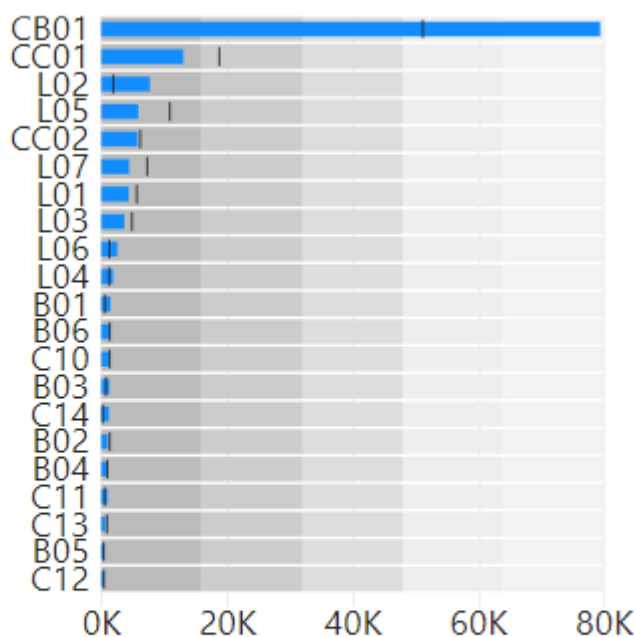
```
MetaIdeal =
CALCULATE(
    SUM(LasMetas[Meta]),
    TREATAS(
        VALUES(Calendario[Año]),
        LasMetas[Año]
    )
)
```

```
Ingresos Reales =
SUMX(
    FILTER(
        Pedidos ,
        Pedidos[Tipo de Compra] = "Normal"
    ) ,
    Pedidos[Ingresos]
)
```

Tabla "LasMetas"

SKU	Año	Meta
CB01	1999	4985,721995
CC01	1999	135,9796679
L02	1999	262,8320941
CC02	1999	345,2906533
L05	1999	159,4965874
L01	1999	148,8769678
L07	1999	76,25901043
L03	1999	53,35581511
L04	1999	63,05325975
L06	1999	437,9508838
B06	1999	30,57401496
C10	1999	184,3710206
CB01	2000	88107,92512
CC01	2000	16871,87639
L02	2000	5058,211417
CC02	2000	5787,474606
L05	2000	2830,821321
L01	2000	5288,526172
L07	2000	6507,041996
L03	2000	2191,64374
L04	2000	3106,926274
L06	2000	466,1214297
B01	2000	1445,336674
B06	2000	895,2715845
C13	2000	2160,194437
C10	2000	617,1560213

Ingresos Reales y Metaideal por SKU



Podemos usar el campo desde la tabla SKUProductos o Categoría de productos.

Category

SKU

Value

Ingresos Reales

Comparison value

Agregar campos de datos a...

Targets

Metaideal

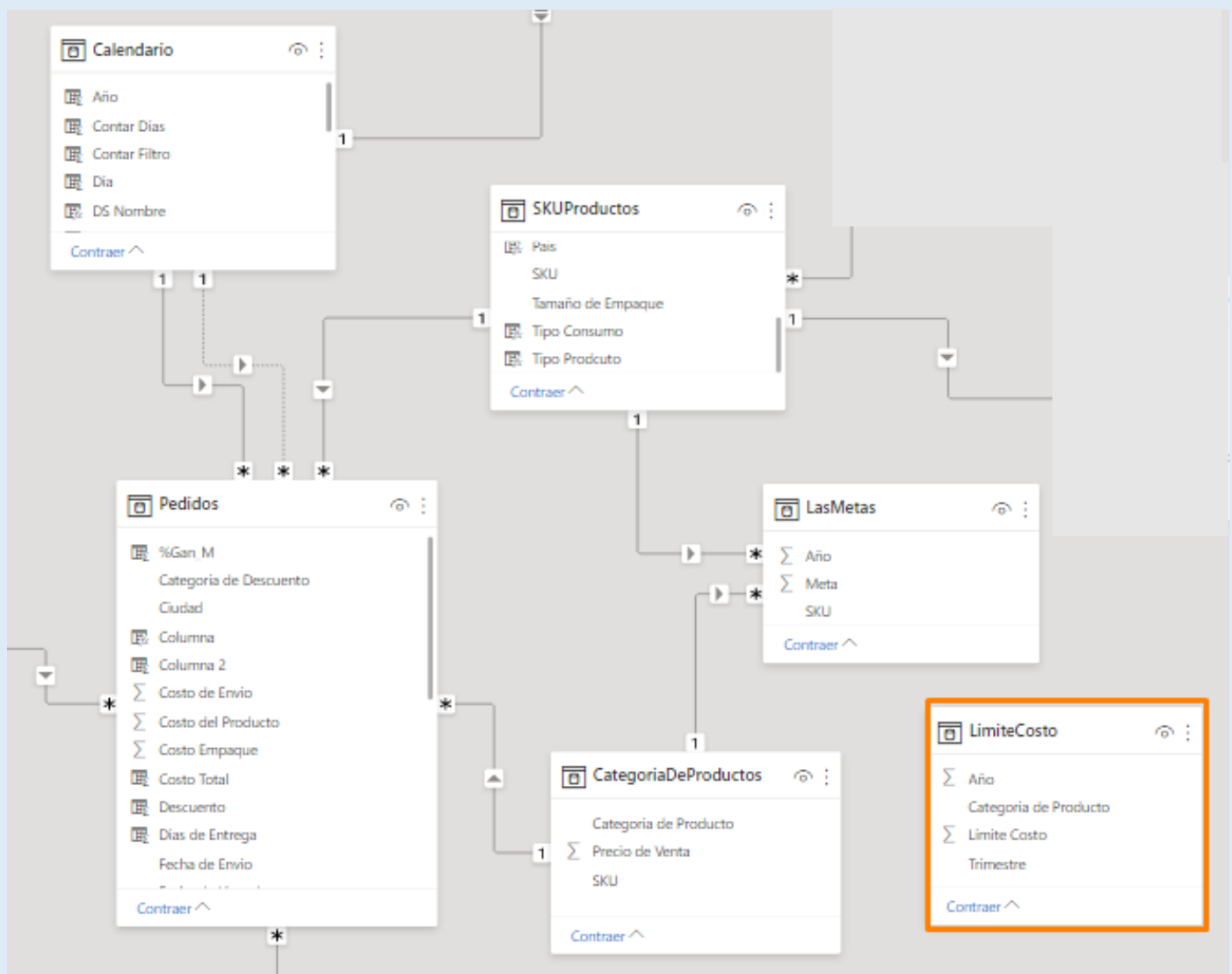
Ejercicio 5:

Ejercicio 2:

La compañía define un límite de costo total (*Costo Real*) por trimestre para cada una de las distintas categorías de productos desde la fundación de la empresa.

Para un *Tabla* con el total desactivado y con los campos: *Categoría de Producto* de la tabla *Categoría de Producto* y *Trimestre* de la tabla de *Calendario* en el área de *Valores*; donde además, exista una segmentación de datos del *Año* de la tabla de *Calendario* a modo de lista desplegable con 2000 seleccionado.

a) Crear una medida el límite estableció para la tabla: <https://www.excelfreeblog.com/wp-content/uploads/2020/07/LimiteCosto.txt>

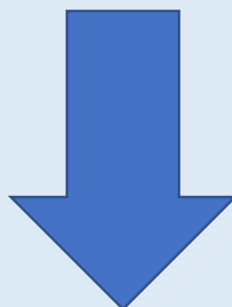



```

Limite Costo =
VAR FiltroAño =
|   TREATAS(VALUEs(Calendario[Año]),LimiteCosto[Año])
VAR FiltroTrimestre =
|   TREATAS(VALUEs(Calendario[Trimestre]),LimiteCosto[Trimestre])
VAR FiltroCategoría =
|   TREATAS(VALUEs(CategoríaDeProductos[Categoría de Producto]),LimiteCosto[Categoría de Producto])
VAR Resultado =
|   CALCULATE(
|       SUM(LimiteCosto[Límite Costo]),
|       FiltroAño,
|       FiltroTrimestre,
|       FiltroCategoría
|   )
RETURN
|   Resultado

```

Año	Categoría de Producto	Trimestre	Costos Tot	Limite Costo
<input type="checkbox"/> (En blanco)				
<input type="checkbox"/> 1999	Blue-Ray	Trim. 1	283,72	469,13
<input checked="" type="checkbox"/> 2000	Blue-Ray	Trim. 2	298,22	483,25
<input type="checkbox"/> 2001	Blue-Ray	Trim. 3	234,80	383,26
<input type="checkbox"/> 2002	Blue-Ray	Trim. 4	412,50	321,98
<input type="checkbox"/> 2003	CD	Trim. 1	239,40	302,59
<input type="checkbox"/> 2004	CD	Trim. 2	281,26	227,72
<input type="checkbox"/> 2005	CD	Trim. 3	210,46	247,68
<input type="checkbox"/> 2006	CD	Trim. 4	209,26	262,48
<input type="checkbox"/> 2007	Combo	Trim. 1	1.702,56	1.654,11
<input type="checkbox"/> 2008	Combo	Trim. 2	1.784,62	1.464,22
<input type="checkbox"/> 2009	Combo	Trim. 3	1.728,68	3.112,65
<input type="checkbox"/> 2010	Combo	Trim. 4	1.586,04	633,55
<input type="checkbox"/> 2011	Libro	Trim. 1	1.588,98	3.216,66
<input type="checkbox"/> 2012	Libro	Trim. 2	1.543,60	1.457,19
<input type="checkbox"/> 2013	Libro	Trim. 3	1.963,66	3.564,93
<input type="checkbox"/> 2014	Libro	Trim. 4	1.704,12	3.094,79
<input type="checkbox"/> 2015				
<input type="checkbox"/> 2016				
<input type="checkbox"/> 2017				
<input type="checkbox"/> 2018				

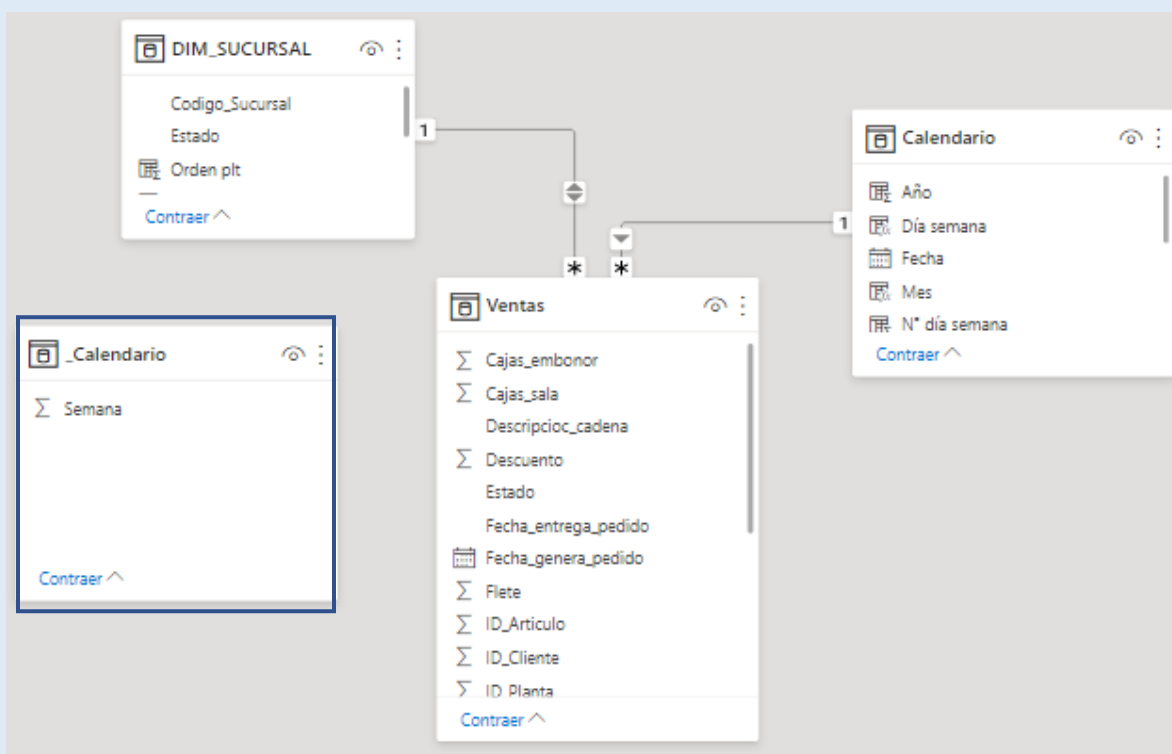


Otra forma de desarrollarlo utilizando ROW:

```
1 LmtCst_2 =
2 VAR Filtro =
3     ROW(
4         "Año", VALUES( Calendario[Año] ),
5         "Trimestre", VALUES( Calendario[Trimestre] ),
6         "Categoría", VALUES( CategoriaDeProductos[Categoría de Producto] )
7     )
8 VAR FiltroActualizarEtiquetas =
9     TREATAS(
10         Filtro,
11         LimiteCosto[Año],
12         LimiteCosto[Trimestre],
13         LimiteCosto[Categoría de Producto]
14     )
15 VAR Resultado =
16     CALCULATE(
17         SUM( LimiteCosto[Límete Costo] ),
18         FiltroActualizarEtiquetas
19     )
20 RETURN
21     Resultado
22
```

Ejercicio 6:

Este ejercicio se encuentra en el video Dia 5 – Dax Nivel 2 (1:00:00 – 2:00:00)



1 _Calendario =	
2 GENERATESERIES(1, 53, 1)	
Semana	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
...	

```
1 Medida =
2 VAR SMAS =
3     MAX(Calendario[Semana]) -- 16
4 VAR SemanaEnLaSeleccion =
5     SELECTEDVALUE(_Calendario[Semana])
6 VAR Comprobacion =
7     SemanaEnLaSeleccion IN {SMAS,SMAS-1,SMAS-2,SMAS-3,SMAS-4}
8 VAR Calculo =
9     CALCULATE(
10         [Valor],
11         TREATAS(
12             VALUES(_Calendario[Semana]),
13             Calendario[Semana]
14         ),
15         REMOVEFILTERS(Calendario)
16     )
17 VAR Resultado =
18     IF(Comprobacion, Calculo, BLANK())
19 RETURN
20 Resultado
```

Calendario (53 filas) Columna: Semana (53 valores distintos)

Fecha de consulta

01-04-2020

17-04-2020

PRUEBA

Año	Mes	Semestre	Trimestre	Semana	Planta	Sucursal
Todas	Todas	Todas	Todas	Todas	Todas	Todas

SOLICITADO POR SEMANA

Semana Valor

14	516
15	54
16	204
Total	774

SOLICITADO ULTIMAS 5 SEMANAS

Semana

14
15
16

Fecha de consulta

01-04-2020

17-04-2020

PRUEBA

Año	Mes	Semestre	Trimestre	Semana	Planta	Sucursal
Todas	Todas	Todas	Todas	Todas	Todas	Todas

SOLICITADO POR SEMANA

Semana Valor

14	516
15	54
16	204
Total	774

SOLICITADO ULTIMAS 5 SEMANAS

Semana Medida

12	2408
13	30
14	991
15	54
16	204
Total	

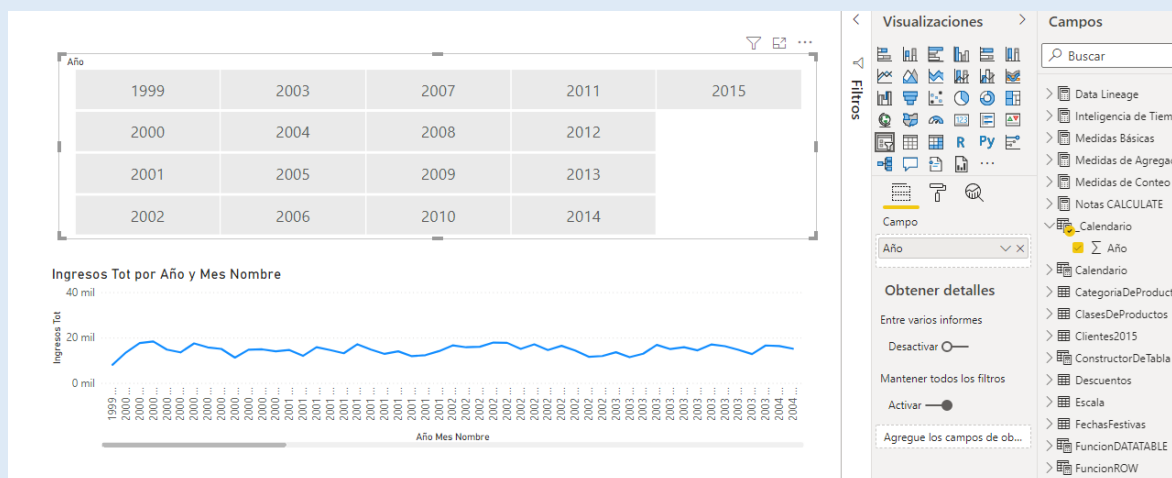
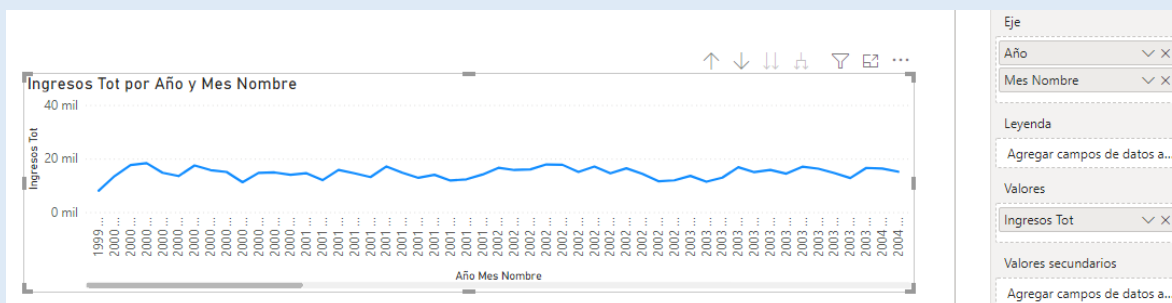
Ejercicio 7:

Este ejercicio se encuentra en el video Dia 5 – Dax Nivel 2 (2:11:00 – 2:00:00)

1 _Calendario =
2 GENERATESERIES(1999, 2015, 1)

Año

1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015



```

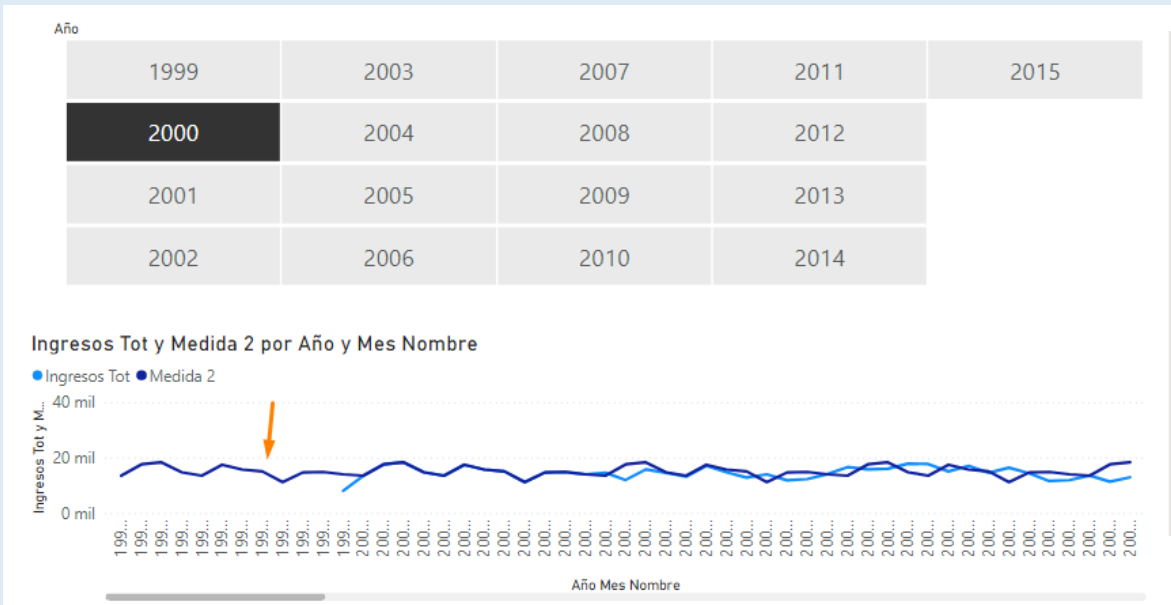
1 Medida 2 =
2 CALCULATE(
3     [Ingresos Tot],
4     TREATAS(
5         VALUES(_Calendario[Año]),
6         Calendario[Año]
7     )
8 )

```

Se repite el mismo valor de cada uno de los meses en todos los años.

Año	Ingresos Tot	Medida 2
+		180.680,24
1999	8.019,74	180.680,24
2000	180.680,24	180.680,24
enero	13.508,96	13.508,96
febrero	17.592,04	17.592,04
marzo	18.328,51	18.328,51
abril	14.726,52	14.726,52
mayo	13.507,17	13.507,17
junio	17.466,96	17.466,96
julio	15.729,11	15.729,11
agosto	15.056,15	15.056,15
septiembre	11.207,04	11.207,04
octubre	14.678,72	14.678,72
noviembre	14.862,94	14.862,94
diciembre	14.016,11	14.016,11
2001	166.860,36	180.680,24
enero	14.558,15	13.508,96
febrero	11.967,01	17.592,04
Total	3.320.219,45	180.680,24

Año	Ingresos Tot	Medida 2
+		180.680,24
1999	8.019,74	180.680,24
2000	180.680,24	180.680,24
2001	166.860,36	180.680,24
enero	14.558,15	13.508,96
febrero	11.967,01	17.592,04
marzo	15.791,46	18.328,51
abril	14.528,67	14.726,52
mayo	13.129,91	13.507,17
junio	17.100,43	17.466,96
julio	14.712,75	15.729,11
agosto	12.854,18	15.056,15
septiembre	13.988,98	11.207,04
octubre	11.852,06	14.678,72
noviembre	12.296,38	14.862,94
diciembre	14.080,39	14.016,11
2002	184.876,83	180.680,24
2003	177.061,95	180.680,24
Total	3.320.219,45	180.680,24



Eje

Año

Mes Nombre

Leyenda

Agregar campos de datos a...

Valores

Ingresos Tot

Medida 2

```

1 Medida 2 =
2 CALCULATE(
3     [Ingresos Tot],
4     KEEPFILTERS(
5         TREATAS(
6             VALUES(_Calendario[Año]),
7             Calendario[Año]
8         )
9     )
10 )

```

Recordar que con CALCULATE modificamos el contexto de filtro, por lo que al elegir un año de la tabla Calendario que creamos, sus valores se impondrán. Mientras que al usar KEEPFILTERS el filtro del año que seleccionemos no se impondrá y generará una intersección con el año del contexto de filtro del gráfico. Por tanto, por ejemplo, al elegir el año 2001 y el contexto de filtro de gráfico es el año 2000, existirá intersección lo que genera vacío. Al elegir el año 2000 y el contexto de filtro es el año 2000 habrá intersección y la línea del gráfico aparecerá.

