



213 FRUVE I

1 PQ+PP+DAX

Jose Ignacio González Gómez

Departamento de Economía, Contabilidad y Finanzas - Universidad de La Laguna

www.jggomez.eu

V.2.5

Ejercicio Basado: [David Asurmendi](#)

DAX aplicado: IF, ISBLANK, COUNTROWS, DISTINCTCOUNT, HASONEVALUE, SUM, SUMX, KPI básico...

Resumen: necesidad de crear un parámetro de conexión, evitar totalizar una columna, contar distintos, mostrar clientes a 0, etc..

De las funciones de agregación analizamos AVERAGE, como se utiliza y sus limitaciones así como el cálculo de %:

Vamos a estudiar cómo obtener valores promedio con la función AVERAGE y calculándolo indirectamente a través de medidas con la función DIVIDE ya que en ocasiones no es posible aplicar la función. AVERAGE solo vale para calcular la media de los valores de una columna, pero cuando intervienen otros valores como resultado del precio por cantidad es decir de dos o más columnas necesitamos calcularlo diseñando una medida sin utilizar la función promedio. Funciones de iteración. SUMX

Técnicas y palabras clave: Crear parámetro de conexión y relacionar 8 tablas.

Funciones DAX Aplicadas. IF, ISBLANK, COUNTROWS, DISTINCTCOUNT, HASONEVALUE, SUM, SUMX, KPI básico ...

Contenido

Presentación.....	3
Se Pide.....	3
Realización de proceso ELT y modelo datos con las medidas necesarias para los informes y visualizaciones requeridas.....	3
Proceso ETL: Conexión, crear parámetro de conexión	3
Proceso ETL: Transformación y carga.....	3
Modelo de datos, tabla calendario y tabla para las medidas	3
Informes y Medidas	4
Informes y medidas I. DAX (IF, ISBLANK, COUNTROWS, DISTINCTCOUNT ...)	4
Informes y medidas II. DAX (DISTINCTCOUNT, HASONEVALUE, SUM ...)	6
Informes y medidas III. DAX (Funciones de iteración, SUMX..)	8
Consideraciones y consejos relacionados.....	9
Orientación. Pasos sugeridos	12
Proceso ETL con Power Query Modelado de datos con Power Pivot	12
Modelo de datos, tabla calendario y tabla para las medidas.....	12
Crear tabla para las organizar las medidas	12
Relaciones, administrador de relaciones del modelo	12
Informes y medidas I.....	13
Informes y medidas II.....	14
Informes y medidas III.....	14
Diseño de KPI	16

Informes y medidas I Objetivo como medida valor, Campo base: en del promedio de margen bruto.....	16
Informes y medidas II. KPI DAX (DISTINCTCOUNT, HASONEVALUE, SUM ...)	16
Informes y medidas III: KPI Objetivo valor absoluto, Campo base: en del promedio de margen bruto.....	17
Resumen de medidas aplicadas y otras fuentes	18

Presentación

FRUVE es una cooperativa agrícola dedicada a la comercialización de productos perecederos de frutas y verduras que opera en diversos países europeos.

Disponemos de la información comercial de los últimos ejercicios y nos interesa analizar los pedidos de los clientes y especialmente de los que no han realizado ninguno con el objeto de dirigir nuestras políticas de marketing hacia este segmento para mejorar las ventas.

Información disponible

Los datos están disponibles en ocho ficheros que deben examinarse previo a crear la conexión y posterior incorporación al modelo de datos y además sería conveniente identificar el tipo de tabla, es decir tabla de hechos o tabla de dimensión lo nos facilitara el diseño del modelo de datos.



Ilustración 1

Respecto a las tablas disponibles, señalar lo siguiente:

1. Calendario. Se nos facilita una tabla calendario correspondiente a los periodos objeto de estudio, por tanto, no debemos crear ninguna, pero si garantizar que es la tabla calendario por defecto.
2. Categorías, tabla simple que contiene las categorías de los productos.
3. Clientes. Contiene información relacionada con el propio nombre de la tabla.
4. Comerciales, información relacionada.
5. Detalle de pedidos, contiene el detalle de cada uno de los pedidos (producto, cantidad, precio, descuento, etc)
6. Pedidos, información básica del pedido como es la fecha, cliente, comercial...
7. Productos, información relacionada.
8. Subcategorías, información relacionada.

Se Pide

Realización de proceso ELT y modelo datos con las medidas necesarias para los informes y visualizaciones requeridas.

Proceso ETL: Conexión, crear parámetro de conexión

Ante los múltiples ficheros a los que queremos conectar se hace necesario crear un parámetro de conexión que facilite la actualización de los enlaces a las fuentes de datos.

Proceso ETL: Transformación y carga

Llevar a cabo el proceso ETL (Conexión-Transformación y Carga) de tal forma que permita crear el modelo de datos base de las visualizaciones propuestas con las medidas necesarias.

Modelo de datos, tabla calendario y tabla para las medidas

Una vez finalizado el proceso ETL y cargado las tablas al modelo tendremos que establecer la tabla de fechas por defecto así como crear las relaciones entre las tablas del modelo.

Con el fin de tener organizado las medidas que vamos a diseñar a lo largo del trabajo vamos a crear una tabla vacía que llamaremos en nuestro caso como "Medidas_jggomez" y que agregaremos al modelo. Esto lo podríamos crear en una hoja nueva y ocultarla posteriormente.

El resultado de este proceso deberá ser similar al siguiente:

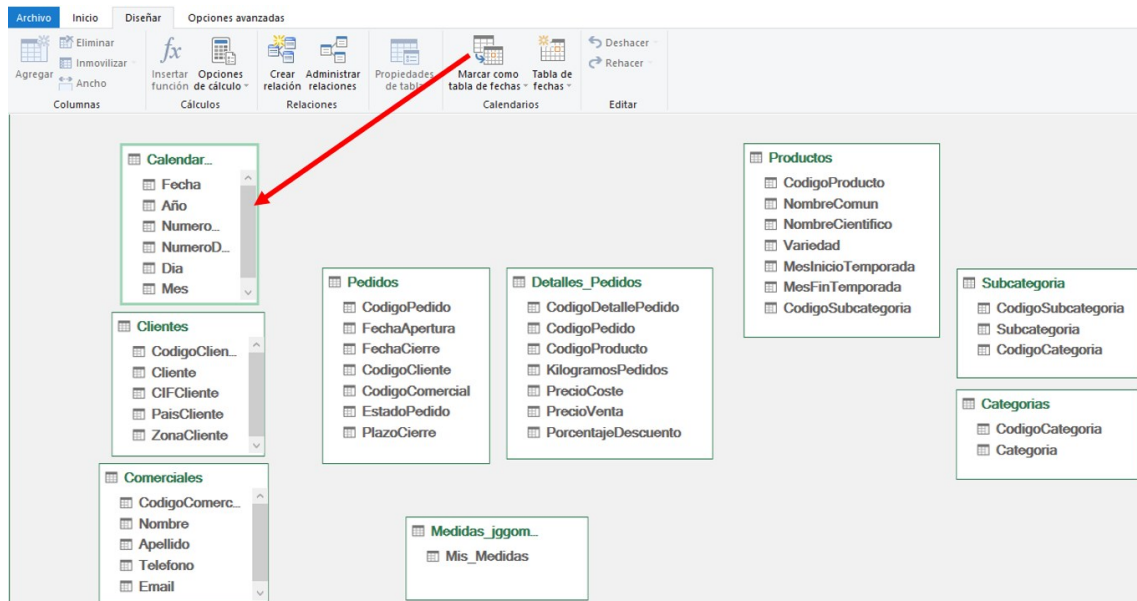


Ilustración 2

Informes y Medidas

Creadas las bases para el análisis pasamos a continuación al desarrollo de un conjunto de informes basados en tablas y gráficos dinámicos

Informes y medidas I. DAX (IF, ISBLANK, COUNTROWS, DISTINCTCOUNT ...)

Se solicita los siguientes informes, donde los segmentadores deben afectar a todos los elementos del informe, tanto tablas dinámicas como gráficos.

Nº de pedidos por cliente

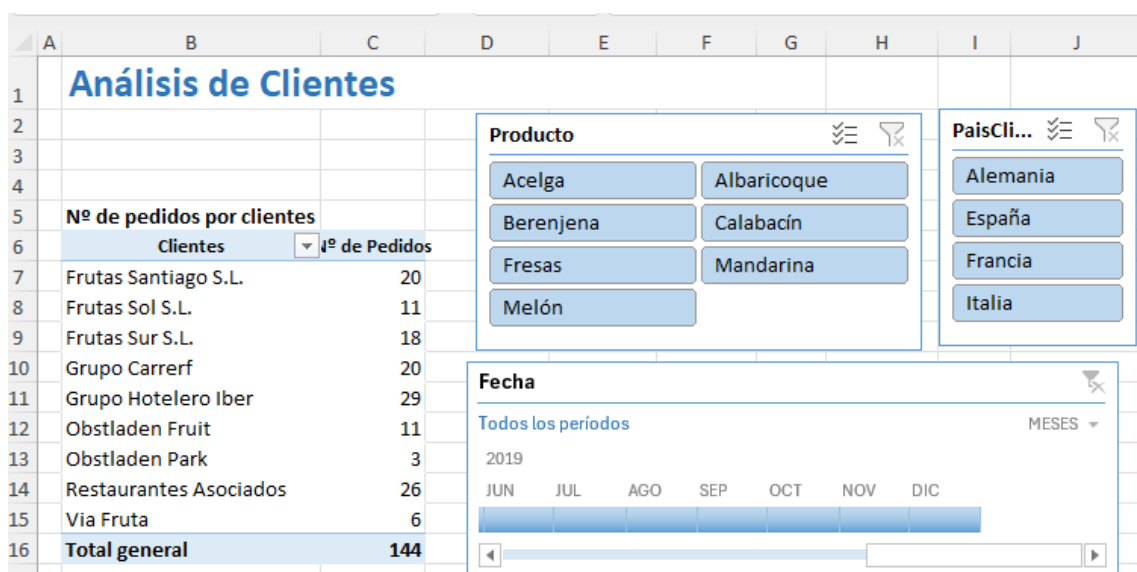


Ilustración 3

En este caso se aplica una medida en la que no muestra los clientes que no hacen pedidos.

El resultado lo podemos ver en la ilustración anterior en la que de los 11 clientes que disponemos solo muestra 9, es decir aquellos clientes que han realizado pedidos, es decir, no muestra los clientes que no han realizado pedidos, por tanto, la función COUNTROWS tiene como limitación que los clientes que no tienen ventas no se muestran por que la función COUNTROWS devuelve vacío o en blanco y ese es un comportamiento que por defecto tienen las TD en el sentido que si no tienen un elemento con valores para mostrar cómo es vacío o blanco, no lo muestra en la TD.

En circunstancias normales podría ser correcto este comportamiento pero en otras ocasiones como las que nos ocupa si interesa mostrar estos valores cero correspondiente a los clientes sin pedidos.

Cientes vs Productos (nº de Pedidos) v.1 sin los clientes que no piden

Clientes vs Productos (nº de Pedidos) v.1 sin los clientes que no piden								
Nº de Pedidos	Productos v1	Nº de P						
Cliente vs Producto	Acelga	Albaricoque	Berenjena	Calabacín	Fresas	Mandarina	Melón	Total general
Frutas Santiago S.L.	84	74	73	67	40	23	37	398
Frutas Sol S.L.	56	67	51	64	50	39	50	377
Frutas Sur S.L.	62		60					122
Grupo Carrerf	56	47	55	81	20	23	22	304
Grupo Hotelero Iber	103	131	105	98				437
Obstladen Fruit	37	54	44	41	40	34	33	283
Obstladen Park					23	18	21	62
Restaurantes Asociados	70	87	82	85	34	29	30	417
Via Fruta	43							43
Total general	511	460	470	436	207	166	193	2.443

Ilustración 4

Igualmente, que en el caso anterior pero aplicado a productos, es decir nos cuenta el número de veces que se cuenta los productos vendidos por cliente, para ello hacemos uso de COUNTROWS.

Cientes vs Productos (nº de Pedidos) v.2 con los clientes que no piden a 0

Clientes vs Productos (nº de Pedidos) v.2 con los clientes que no piden a 0								
Nº de Pedidos	Productos v2	Nº de P						
Cliente vs Producto	Acelga	Albaricoque	Berenjena	Calabacín	Fresas	Mandarina	Melón	Total general
Fruits Fruits S.A.	0	0	0	0	0	0	0	0
Frutas Santiago S.L.	84	74	73	67	40	23	37	398
Frutas Sol S.L.	56	67	51	64	50	39	50	377
Frutas Sur S.L.	62	0	60	0	0	0	0	122
Fruterías Paco	0	0	0	0	0	0	0	0
Grupo Carrerf	56	47	55	81	20	23	22	304
Grupo Hotelero Iber	103	131	105	98	0	0	0	437
Obstladen Fruit	37	54	44	41	40	34	33	283
Obstladen Park	0	0	0	0	23	18	21	62
Restaurantes Asociados	70	87	82	85	34	29	30	417
Via Fruta	43	0	0	0	0	0	0	43
Total general	511	460	470	436	207	166	193	2.443

Ilustración 5

En esta ocasión podremos analizar aquellos clientes que no nos piden y por tanto buscar las razones.

Nº de pedidos por cliente y productos distintos

Basado en el informe de la Ilustración 3 se pide completarlo con el número de productos diferentes que ha comprado cada cliente.

Nº de pedidos por cliente y productos distintos		
Cientes	Nº de Pedi	Nº Productos Distintos
Frutas Santiago S.L.	20	7
Frutas Sol S.L.	11	7
Frutas Sur S.L.	18	2
Grupo Carrerf	20	7
Grupo Hotelero Iber	29	4
Obstladen Fruit	11	7
Obstladen Park	3	3
Restaurantes Asociados	26	7
Via Fruta	6	1
Total general	144	7

Ilustración 6

En este caso debemos tener en cuenta que en la tabla DetallesPedidos disponemos de los productos que configuran cada pedido y además puede darse el caso que en un mismo pedido existe productos duplicados por tener diferentes calidades y por tanto precios, por ejemplo, plátanos óptimos y plátanos maduros, los primeros serán más caros. Por tanto, podemos tener productos duplicados en el mismo pedido, según el estado de conservación.

En definitiva, queremos una TD similar a la siguiente con los productos únicos de cada cliente ya que nos interesa además conocer cuántos productos diferentes me ha comprado cada cliente, me puede interesar saber quién me compra menos productos de una determinada categoría y así hacer ofertas especiales.

Nº de pedidos por cliente y KPI basado en un medida como objetivo

Nº de pedidos por cliente con KPI		
Cientes	Nº de Pedido	Estado NºdePedidos
Frutas Santiago S.L.	20	●
Frutas Sol S.L.	11	●
Frutas Sur S.L.	18	●
Grupo Carrerf	20	●
Grupo Hotelero Iber	29	●
Obstladen Fruit	11	●
Obstladen Park	3	●
Restaurantes Asociados	26	●
Via Fruta	6	●
Total general	144	●

Ilustración 7

La medida ha crear es "Objetivo nº de pedidos por cliente" = 15 y el intervalo establecido es 60% vs 80%

Informes y medidas II. DAX (DISTINCTCOUNT, HASONEVALUE, SUM ...)

Se nos ha solicitado un primer informe de evaluación de la demanda de productos similar al siguiente:

Ilustración 8

En este caso debemos tener en cuenta que en la tabla DetallesPedidos disponemos de los productos que configuran cada pedido y además puede darse el caso que en un mismo pedido existe productos duplicados por tener diferentes calidades y por tanto precios, por ejemplo, plátanos óptimos y plátanos maduros, los primeros serán más caros. Por tanto, podemos tener productos duplicados en el mismo pedido, según el estado de conservación y solo nos interesa contar el número de pedidos de cada producto independientemente de su estado, así el caso de que en un mismo pedido tengamos cinco precios de plátano distintos debido a su diferencia de calidad, debe ser considerado solo una vez, no cinco.

Por otro lado es necesario en ocasiones **modificar el total general de una columna** al carecer sentido, por ejemplo como vemos en Ilustración 8, el único total que tiene relevancia o sentido es el de total de kg, el resto de totales de columnas son incoherentes, por ejemplo el número de veces de total general

Una forma de solucionarlo poco doctrinal es ocultar este total a través de “opciones de tabla dinámica” pestaña “Totales y Filtros” desactivamos “Mostrar totales de las columnas”

Esto afectaría a todos los totales de la tabla dinámica y seguramente solo nos interesaría para un total concreto.

Otra opción sería ocultarlo cambiando el color de fuente y hacerlo coincidir con el fondo.

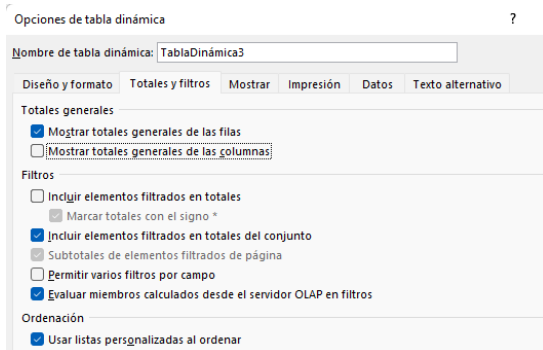


Ilustración 9

Vamos a presentar otra opción más correcta mediante una fórmula es decir modificar el total general de una columna desde la fórmula de una medida con el uso de la función **HASONEVALUE** y detectar cuando se calcula un total general contado filas.

Recordemos que los totales generales no son el resultado de las celdas superiores, sino que se le aplica la formula o campo calculado y lo que vamos a hacer es dejar ese resultado vacío que es distinto que oculto.

Este informe contiene 3 filtros o segmentadores que solo afectan a la tabla dinámica y hemos desarrollado tres medidas con sus respectivas variantes:

1. NºVecesProducto v1, evalúa el número de veces que se ha pedido un producto ajustado al contexto de filtro definido por los segmentadores.
Hemo creado una versión de esta primera medida con el fin de evitar que se muestre un total al carecer de sentido, uso de HASONEVALUE y que sera NºVecesProducto v2
2. Total kg, calcula el total de kg
3. Promedio, calcula el promedio de kg por pedido, pero en este caso con tres versiones, la primera usando la función average que calcula el promedio de cada pedio de producto considerando cada vez que se registra aunque se repita en el pedido, Promedio v1 de la Ilustración 8 (ver tambien A de la Ilustración 10), y Promedio v2 de la Ilustración 8 considerando que son el mismo producto del pedido (ver b de la Ilustración 10)
Se puede observar que hay notables diferencias en el promedio. La ultima versión Promedio v3 es igual a la anterior pero aplicando HASONEVALUE para evitar totalizar.

	Plátano Optimo	Plátano Maduro	Total
Pedido nº 20112	50 kg	30 kg	80 kg
Pedido nº 20113	20 kg	10 kg	30 kg
A) Promedio de kg por pedido sobre 4 (área azul):			28 kg
B) Promedio de kg por pedido sobre 2 (área verde):			55 kg

Ilustración 10

Informes y medidas III. DAX (Funciones de iteración, SUMX..)

Presentacion de las funciones de iteración

Vamos a seguir avanzando en los conceptos básicos de medidas, funciones y tipos de cálculos más frecuentes.

Con este ejercicio vamos a presentar las funciones de interacción que nos permiten hacer operaciones a nivel de celda, es decir si queremos calcular un resultado a nivel de celda en el que intervengan varias columnas (el cálculo no depende de una sola columna), entonces debemos usar una función iteradora que corresponda a cada operación, SUMX, MINX, etc..

En resumen, la diferencia entre las funciones de agregación básica como **SUM** respecto a las funciones de iteración como **SUMX** es que nos permite realizar operaciones a nivel de celda con diversas columnas de la misma tabla u otras columnas con la función **RELATED**.

Estas funciones iteradoras van recorriendo la tabla fila a fila y aplicando la expresión o calculo establecido.

En diversas ocasiones podremos sustituir el uso de las funciones de iteracion operando directamente con columnas calculadas y obtendríamos el mismo resultado, pero en nuestro caso vamos a tratar de evitar trabajar con columnas calculadas excepto que sean estrictamente necesarias y sustituirla por medidas que empleen funciones de agregación o de iteración como son SUMX, MINX, etc.

Se pide

Asi se solicita los siguientes informes, donde los segmentadores deben afectar a todos los elementos del informe, tanto tablas dinámicas como gráficos.

Análisis Financiero: Valores Absolutos: Ingresos Netos, márgenes brutos absolutos y porcentuales

Análisis financiero

Valores Absolutos: Ingresos Netos, márgenes brutos absolutos y porcentuales

Producto	T.Kg	Subtotal	Descuento	T. Ingres.Net	Total Coste	Margen Bruto	% Marg.Bruto.Cost	% Marg.Bruto.IngNeto
Acelga	97.300 kg	100.899 €	5.647 €	95.252 €	75.130 €	20.122 €	26,78 %	21,12 %
Albaricoque	82.950 kg	85.589 €	4.949 €	80.640 €	70.382 €	10.258 €	14,57 %	12,72 %
Berenjena	93.820 kg	99.710 €	5.592 €	94.118 €	72.922 €	21.196 €	29,07 %	22,52 %
Calabacín	86.340 kg	92.012 €	5.504 €	86.508 €	66.821 €	19.687 €	29,46 %	22,76 %
Fresas	19.070 kg	20.552 €	895 €	19.657 €	16.162 €	3.495 €	21,62 %	17,78 %
Mandarina	14.510 kg	15.611 €	961 €	14.650 €	11.884 €	2.766 €	23,27 %	18,88 %
Melón	15.960 kg	17.249 €	951 €	16.298 €	11.529 €	4.769 €	41,36 %	29,26 %
Total general	409.950 kg	431.622 €	24.500 €	407.122 €	324.830 €	82.292 €	25,33 %	20,21 %

Fecha: Todos los periodos (2018, 2019) TRIMESTRES

Cliente: Frutas Santiago S.L., Frutas Sol S.L., Frutas Sur S.L.

Comercial: Alex, Elena, Fran, Luis, María, Paula

PaisCliente: Alema..., España, Francia, Italia

Ilustración 11

Análisis Financiero: Valores Medios: Promedio de Kg por Pedido, Promedio Margen Bruto por Kg

Valores Medios: Promedio de Kg por Pedido, Promedio Margen Bruto por Kg

Producto	T. Ingres.Net	Margen Bruto	T.Kg	Nº de Pedidos	PromKg.Pedido	PromMargBruto.K	Estado KPI
Acelga	95.252 €	20.122 €	97.300 kg	511	190 kg/pedido	0,207 €/kg	●
Albaricoque	80.640 €	10.258 €	82.950 kg	460	180 kg/pedido	0,124 €/kg	●
Berenjena	94.118 €	21.196 €	93.820 kg	470	200 kg/pedido	0,226 €/kg	●
Calabacín	86.508 €	19.687 €	86.340 kg	436	198 kg/pedido	0,228 €/kg	●
Fresas	19.657 €	3.495 €	19.070 kg	207	92 kg/pedido	0,183 €/kg	●
Mandarina	14.650 €	2.766 €	14.510 kg	166	87 kg/pedido	0,191 €/kg	●
Melón	16.298 €	4.769 €	15.960 kg	193	83 kg/pedido	0,299 €/kg	●
Total general	407.122 €	82.292 €	409.950 kg	2.443	168 kg/pedido	0,20 €/kg	●

Ilustración 12

Consideraciones y consejos relacionados

Sobre columnas calculadas

Evitar el uso de columnas calculadas a no ser que sea estrictamente necesario por el consumo excesivo de recursos, afectando a la eficiencia del modelo de datos.

Reutilización de las medidas. Evitar Uso de Medidas Creadas.

Otra cuestión a tener en cuenta es el consejo relacionado con la reutilización de medidas y como esto afecta al rendimiento, especialmente cuando estas medidas que se reutilizan contienen funciones iteradoras.

Es decir, debemos crear la medida evitando el uso de medidas intermedias ya creadas para así evitar las interacciones o funciones de agregación y mejorar el rendimiento.

Medida ? X

Nombre de la tabla:

Nombre de la medida:

Descripción del valor:

Fórmula:

```
=SUMX(DetallesPedidos;DetallesPedidos[KilogramosPedidos] * DetallesPedidos[PrecioVenta] -
DetallesPedidos[KilogramosPedidos] * DetallesPedidos[PrecioVenta] * DetallesPedidos[PorcentajeDescuento])
```

Ilustración 13

Básicamente lo que hacemos es repetir cálculos anteriores (por ejemplo, medidas ya creadas) que habíamos diseñado por separado previamente, pero ahora lo hacemos todo junto para obtener el total de cada fila que se ira totalizando para ser devuelto por la función SUMX.

Es decir, hemos creado la medida haciendo las operaciones detalladamente para cada fila se hace el calculo de:

$$\text{DetallesPedidos;DetallesPedidos[KilogramosPedidos]} * \text{DetallesPedidos[PrecioVenta]} - \text{DetallesPedidos[KilogramosPedidos]} * \text{DetallesPedidos[PrecioVenta]} * \text{DetallesPedidos[PorcentajeDescuento]}$$

La primera parte de la función antes del menos, es el caculo del subtotal:

$$\text{DetallesPedidos;DetallesPedidos[KilogramosPedidos]} * \text{DetallesPedidos[PrecioVenta]}$$

La segunda parte, lo que va despues del – es el calculo del descuento que se resta al subtotal

$$- \text{DetallesPedidos[KilogramosPedidos]} * \text{DetallesPedidos[PrecioVenta]} * \text{DetallesPedidos[PorcentajeDescuento]}$$

Y al final se obtiene el resultado de la suma, por tanto, la agregación o interacción solo se realiza una vez, solo se aplica una vez, mientras que en el primer metodo el que hemos denominado Simple en codigo lo que se hace para cada celda son dos interacciones por celda una para la primera medida (€ Subtotal) y para el descuentos lo que genera un consumo de recursos extraordinario y si los datos que estamos trabajando son ficheros de peso, podría verse afectado el rendimiento del informe.

Hemos intentado evitar el uso de medidas que teníamos ya creada porque estaríamos duplicando el uso de SUMX con el consiguiente esfuerzo y consumo de recursos y tiempo, estamos usando dos funciones cuando se puede utilizar una sola.

Por tanto podemos reutilizar medidas pero con cierto rigor y teniendo en cuenta donde y como puede afectar al rendimiento general.

Uso de la función RELATED

Por ultimo, a veces en estos cálculos necesitamos acceder a datos o columnas que esta en otra tabla, por tanto para hacer referencia a esa columna de la otra tabla empleamos la función RELATED. Veamos el siguiente ejemplo donde el descuento de los productos en vez de estar en una columna concreta de la tabla Detalles_Pedidos estuviera en una tabla de Descuento que tiene el codigo del producto y la tasa de descuento, por tanto para obtener el valor del descuento en este caso seria igual que como vimos anteriormente pero haciendo referencia a esa ubicación.

Medida	
Nombre de la tabla:	DetallesPedidos
Nombre de la medida:	€ Descuento
Descripción del valor:	
Fórmula:	<input type="button" value="fx"/> Comprobar fórmula DAX
<code>=SUMX(DetallesPedidos; DetallesPedidos[KilogramosPedidos] * DetallesPedidos[PrecioVenta] * RELATED(Descuentos[PorcentajeDescuento]))</code>	

Ilustración 14

Esto exige, evidentemente que las tablas estén correctamente relacionadas.

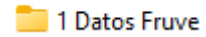
Orientación. Pasos sugeridos

- Creamos nuestra hoja de solución al caso planteado

Proceso ETL con Power Query Modelado de datos con Power Pivot

1. Como siempre el primer paso es analizar el contenido, estructura y formato de los ficheros disponibles.
2. Crear Conexión

Proponemos crear conexión a todos los ficheros a la carpeta donde están los datos, en concreto la llamada 1 Datos Fruve, de esta forma se creará la conexión a todos los ficheros presentes y futuros que este en la citada carpeta.



Vamos a crear el parámetro de conexión y posteriormente configurar las conexiones, partiendo que estamos trabajando con una carpeta en el escritorio que contiene los datos, el parámetro de conexión será:

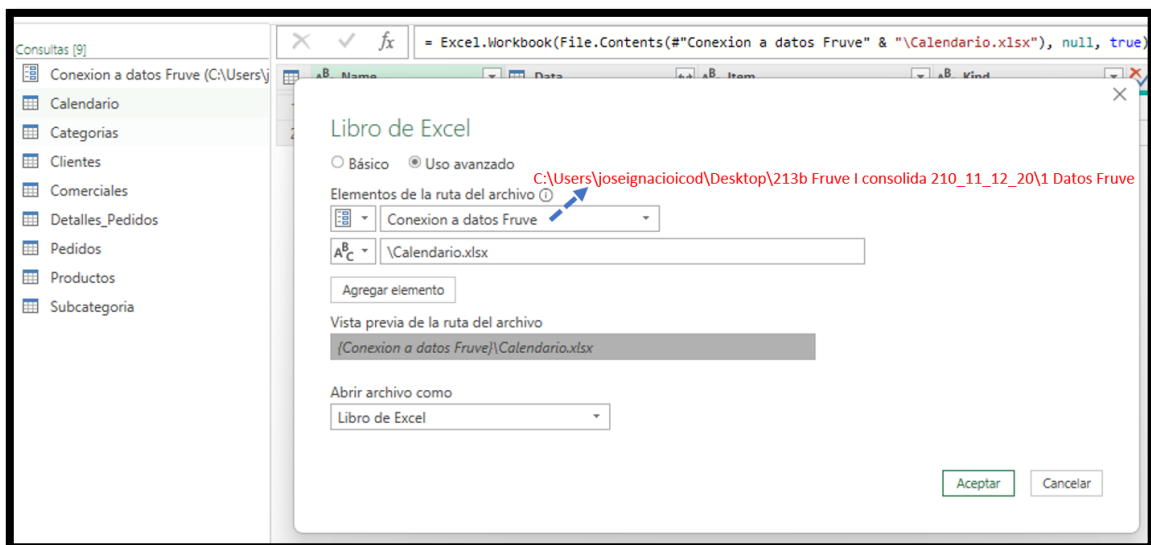


Ilustración 15

Basado en este parámetro creamos las conexiones al resto de ficheros y aplicamos las transformaciones necesarias de los ficheros. Terminamos este proceso con la carga de las conexiones (tablas) al modelo de datos.

Modelo de datos, tabla calendario y tabla para las medidas

Crear tabla para las organizar las medidas

Recordemos que una vez cargado las tablas al modelo tendremos que establecer la tabla de fechas por defecto así como crear las relaciones entre las tablas del modelo, pero previamente creamos la tabla para organizar las medidas y ocultamos la hoja donde la hemos creado.

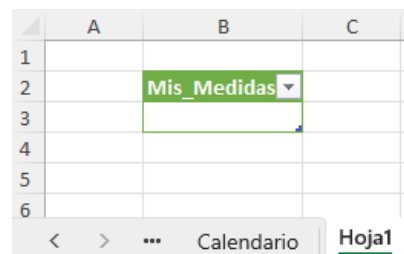


Ilustración 16

Relaciones, administrador de relaciones del modelo

Con las tablas cargadas procedemos a establecer las relaciones correspondientes, podemos observar que la única tabla que queda sin relacionar es la de Medidas_jggomez.

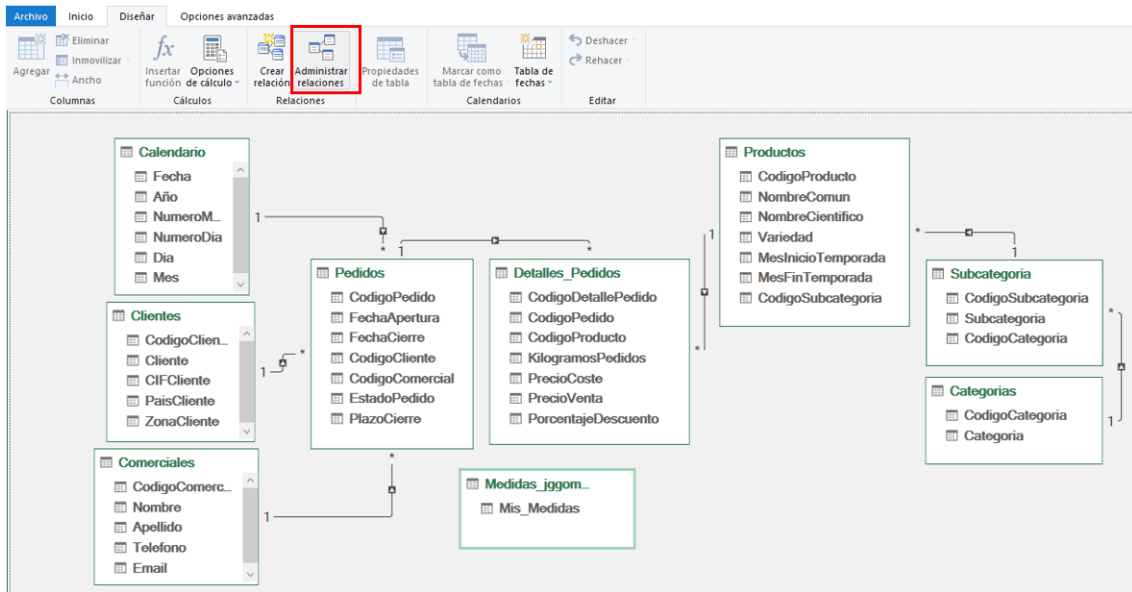


Ilustración 17

Si accedemos al Administrador de relaciones (ver Ilustración 17), nos muestra todas las relaciones establecidas en el modelo (Ilustración 18)

Administrar relaciones

Crear Editar Eliminar

Activo	Tabla 1	Cardinalidad	Dirección de filtro	Tabla 2
Sí	Detalles_Pedidos [CodigoPedido]	Varios a uno (*:1)	<< A Detalles_Pedidos	Pedidos [CodigoPedido]
Sí	Detalles_Pedidos [CodigoProducto]	Varios a uno (*:1)	<< A Detalles_Pedidos	Productos [CodigoProducto]
Sí	Pedidos [CodigoCliente]	Varios a uno (*:1)	<< A Pedidos	Clientes [CodigoCliente]
Sí	Pedidos [CodigoComercial]	Varios a uno (*:1)	<< A Pedidos	Comerciales [CodigoComercial]
Sí	Pedidos [FechaApertura]	Varios a uno (*:1)	<< A Pedidos	Calendario [Fecha]
Sí	Productos [CodigoSubcategoria]	Varios a uno (*:1)	<< A Productos	Subcategoria [CodigoSubcategoria]
Sí	Subcategoria [CodigoCategoria]	Varios a uno (*:1)	<< A Subcategoria	Categorías [CodigoCategoria]

Ilustración 18

- Vemos por ejemplo que la tabla Detalles_Pedidos se relaciona con la tabla Pedidos a través de la columna CodigoPedido.
- Igualmente la Pedidos se relaciona con la de Calendario a través de las columnas FechaApertura y Fecha respectivamente.

Informes y medidas I

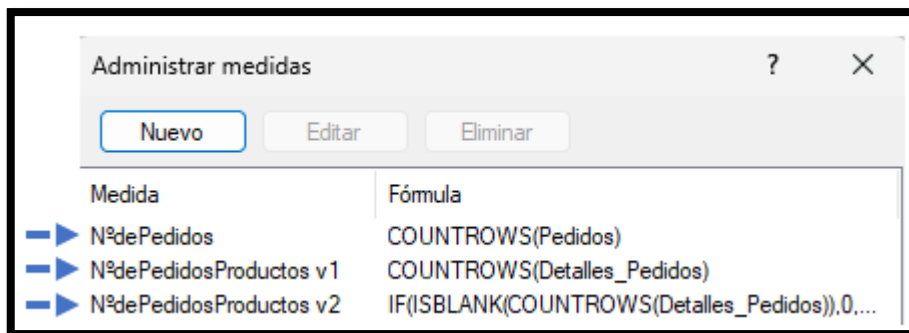


Ilustración 19

- N°dePedidos: =COUNTROWS(Pedidos)
Cuenta de la tabla de pedidos el total realizado pedidos
- N°dePedidosProductos v1= COUNTROWS(Detalles_Pedidos)

Cuenta el nº de registro en la tabla detalle de pedidos

- `NºdePedidosProductos v2=`
`=IF(ISBLANK(COUNTROWS(Detalles_Pedidos));0;COUNTROWS(Detalles_Pedidos))`
 Nos da el nº de pedidos de productos incluyendo lo que no piden a 0
- `NºdeProductosDistintos=` `DISTINCTCOUNT(Detalles_Pedidos[CodigoProducto])`

Informes y medidas II

- `NºVecesProducto v1 =` `DISTINCTCOUNT(Detalles_Pedidos[CodigoPedido])`
 Nº veces que se pide un producto
- `NºVecesProducto v2=`
`=IF(HASONEVALUE(Productos[NombreComun])=TRUE();DISTINCTCOUNT(Detalles_Pedidos[Codi`
`goPedido]);BLANK())`
 Calcula el promedio de productos unicos sin distinguir precios, no totaliza
- `T.Kg=` `SUM(Detalles_Pedidos[KilogramosPedidos])`
 Devuelve el total de Kg
- `Promedio v1 =` `AVERAGE(Detalles_Pedidos[KilogramosPedidos])`
 Calcula el promedio de kg por pedido
- `Promedio v2 =` `DIVIDE([T.Kg];[NºVecesProducto v1])`
 Calcula el promedio teniendo en cuenta la repetición de productos en un mismo pedido
- `Promedio v3=`
`=IF(HASONEVALUE(Productos[NombreComun])=TRUE();[Promedio v2];BLANK())`
 Calcula el promedio teniendo en cuenta la repetición de productos en un mismo pedido y evitando totalizar la columna

Informes y medidas III

- `Subtotal =`
`=SUMX(Detalles_Pedidos;Detalles_Pedidos[KilogramosPedidos]*Detalles_Pedidos[PrecioVenta])`
 Calcula el total de ingresos multiplicando para cada detalle de pedido, el total de Kg por precio unitario sin descuento.

Nos interesa conocer para cada pedido de producto (Tabla Detalles_Pedidos) el subtotal como resultado del cálculo de las columnas Kilogramos de pedido por PrecioVenta.

Para ello vamos a crear una medida que sería la intuitiva en la tabla dinámica,

`=SUM(Detalles_Pedidos[KilogramosPedidos]*Detalles_Pedidos[PrecioVenta])`

Esto genera un error porque la función SUM solo permite una columna de datos y no expresiones como multiplicar dos columnas. Es decir, con SUM lo que vamos a obtener es el total de una columna de una tabla.

Por tanto, vamos a crear la medida correctamente con la función SUMX y que según sintaxis en primer lugar debemos definir los parámetros correspondientes como son:

`SUMX(table;expresión)`

- $\text{Descuento} = \text{SUMX}(\text{Detalles_Pedidos}; \text{Detalles_Pedidos}[\text{PorcentajeDescuento}] * [\text{Subtotal}])$
Calcula el total de descuento aplicado en cada detalle de pedido.
- $\text{Total Ingreso Neto} = [\text{Subtotal}] - [\text{Descuento}]$
Obtenemos el total de ingreso neto como diferencia de las dos medidas calculadas anteriormente.
- $\text{Total Coste} =$
 $= \text{SUMX}(\text{Detalles_Pedidos}; \text{Detalles_Pedidos}[\text{KilogramosPedidos}] * \text{Detalles_Pedidos}[\text{PrecioCoste}])$
Calculamos el total del coste como paso previo para determinar el margen bruto
- $\text{Margen Bruto} = [\text{Total Ingreso Neto}] - [\text{Total Coste}]$
Calculamos el total de margen bruto.
- $\% \text{ Marg.Bruto.Coste} = \text{Divide}([\text{Margen Bruto}]; [\text{Total Coste}])$
Determinamos el % de margen bruto que obtenemos sobre el coste.
- $\% \text{ Marg.Bruto.IngNeto} = \text{DIVIDE}([\text{Margen Bruto}]; [\text{Total Ingreso Neto}])$
Determinamos el % de margen bruto sobre el ingreso neto.
- $\text{PromKg.Pedido} = \text{Divide}([\text{T.Kg}]; [\text{N}^\circ \text{de Pedidos Productos v1}])$
Promedio de Kg por Pedido
- $\text{PromMargBruto.Kg} = \text{Divide}([\text{Margen Bruto}]; [\text{T.Kg}])$
Promedio Margen Bruto por Kg

Diseño de KPI

Se propone crear y aplicar los siguientes KPI en los informes y tabla dinámicas siguientes:

Informes y medidas I Objetivo como medida valor, Campo base: en del promedio de margen bruto

Se establece como objetivo el numero de pedidos por clientes asignándole un valor de 15 y se establece como intervalo 60% vs 80%

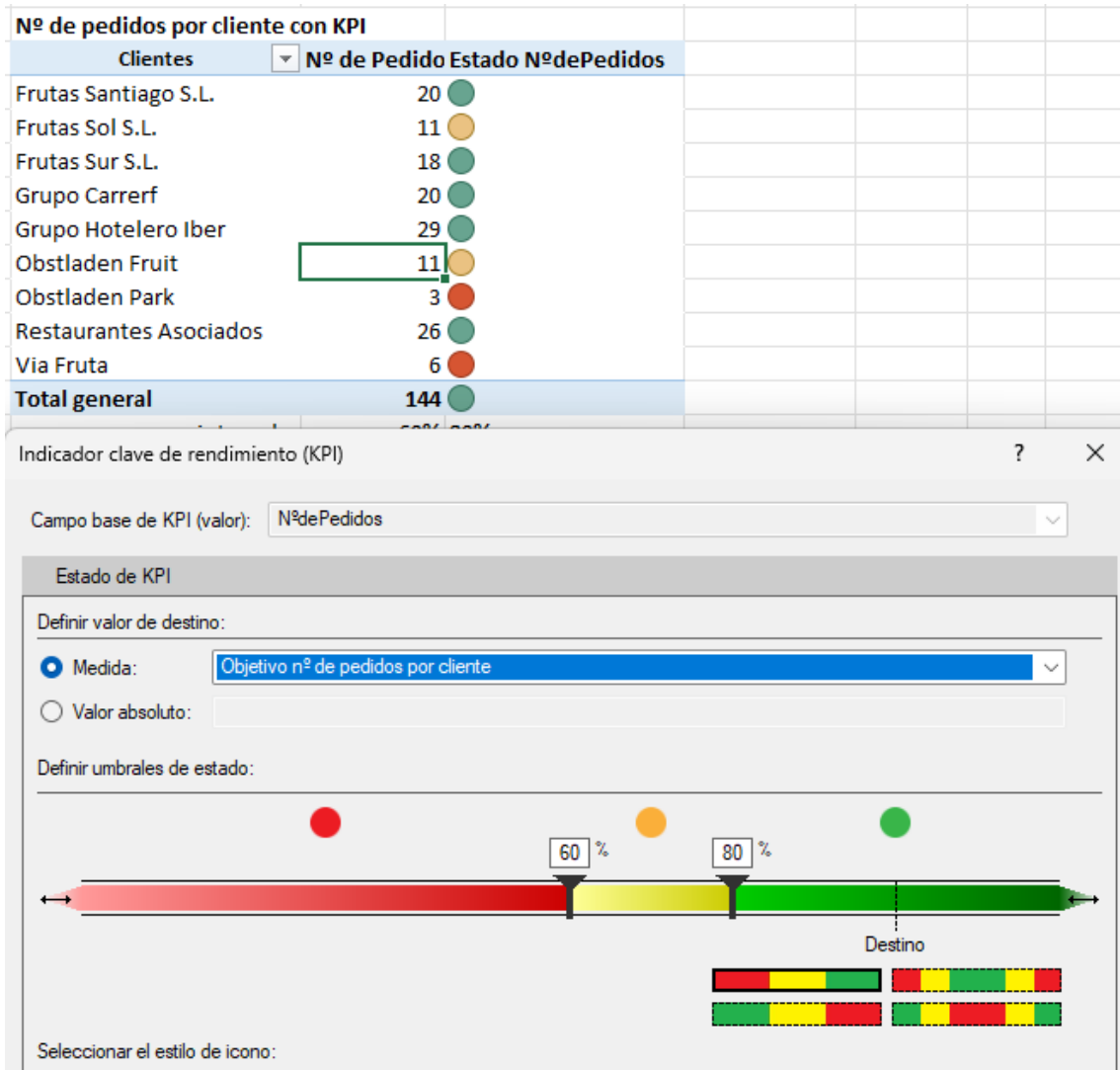


Ilustración 20

Informes y medidas II. KPI DAX (DISTINCTCOUNT, HASONEVALUE, SUM ...)

Informes y medidas III: KPI Objetivo valor absoluto, Campo base: en del promedio de margen bruto

Valores Medios: Promedio de Kg por Pedido, Promedio Margen Bruto por Kg							
Producto	T. Ingres.Net	Margen Bruto	T.Kg	Nº de Pedidos	PromKg.Pedido	PromMargBruto.K	Estado KPI
Acelga	95.252 €	20.122 €	97.300 kg	511	190 kg/pedido	0,207 €/kg	●
Albaricoque	80.640 €	10.258 €	82.950 kg	460	180 kg/pedido	0,124 €/kg	●
Berenjena	94.118 €	21.196 €	93.820 kg	470	200 kg/pedido	0,226 €/kg	●
Calabacín	86.508 €	19.687 €	86.340 kg	436	198 kg/pedido	0,228 €/kg	●
Fresas	19.657 €	3.495 €	19.070 kg	207	92 kg/pedido	0,183 €/kg	●
Mandarina	14.650 €	2.766 €	14.510 kg	166	87 kg/pedido	0,191 €/kg	●
Melón	16.298 €	4.769 €	15.960 kg	193	83 kg/pedido	0,299 €/kg	●
Total general	407.122 €	82.292 €	409.950 kg	2.443	168 kg/pedido	0,20 €/kg	●

Indicador clave de rendimiento (KPI) ? X

Campo base de KPI (valor): PromMargBruto.Kg

Estado de KPI

Definir valor de destino:

Medida:

Valor absoluto: 0.2

Definir umbrales de estado:

Destino

Seleccionar el estilo de icono:

Resumen de medidas aplicadas y otras fuentes

Administrar medidas	
<input type="button" value="Nuevo"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>	
Medida	Fórmula
% Marg.Bruto.Cost	Divide([Margen Bruto],[Total Coste])
% Marg.Bruto.IngNeto	DIVIDE([Margen Bruto],[Total Ingreso Neto])
Descuento	SUMX(Detalles_Pedidos,Detalles_Pedidos[PorcentajeDescuento]*[Subtotal])
Margen Bruto	[Total Ingreso Neto]-[Total Coste]
NºdePedidos	COUNTROWS(Pedidos)
NºdePedidosProductos v1	COUNTROWS(Detalles_Pedidos)
NºdePedidosProductos v2	IF(ISBLANK(COUNTROWS(Detalles_Pedidos)),0,COUNTROWS(Detalles_Pedidos))
NºdeProductosDistintos	DISTINCTCOUNT(Detalles_Pedidos[CodigoProducto])
NºVecesProducto v1	DISTINCTCOUNT(Detalles_Pedidos[CodigoPedido])
NºVecesProducto v2	IF(HASONEVALUE(Productos[NombreComun])=TRUE(),DISTINCTCOUNT(Detalles_Pedidos[CodigoPedido]),BLANK())
Promedio v1	AVERAGE(Detalles_Pedidos[KilogramosPedidos])
Promedio v2	DIVIDE([T.Kg],[NºVecesProducto v1])
Promedio v3	IF(HASONEVALUE(Productos[NombreComun])=TRUE(),[Promedio v2],BLANK())
PromKg.Pedido	Divide([T.Kg],[NºdePedidosProductos v1])
PromMargBruto.Kg	Divide([Margen Bruto],[T.Kg])
Subtotal	SUMX(Detalles_Pedidos,Detalles_Pedidos[KilogramosPedidos]*Detalles_Pedidos[PrecioVenta])
T.Kg	SUM(Detalles_Pedidos[KilogramosPedidos])
Total Coste	SUMX(Detalles_Pedidos,Detalles_Pedidos[KilogramosPedidos]*Detalles_Pedidos[PrecioCoste])
Total Ingreso Neto	[Subtotal]-[Descuento]

Ilustración 21

[Funciones DAX | Interactive Chaos](#)

[Escenarios DAX | Interactive Chaos](#)

[SUMX | Interactive Chaos](#)

[RELATED | Interactive Chaos](#)