

SALVADOR RAMOS

Excel 2013, Power Pivot Y DAX

Tus análisis elevados
a la enésima potencia



SolidQ

Analiza tu Negocio con Excel y Power BI
Aprende de tus datos

VOL. IV

ADVERTENCIA LEGAL

Todos los derechos de esta obra están reservados a SolidQ™ Press.

El editor prohíbe cualquier tipo de fijación, reproducción, transformación o distribución de esta obra, ya sea mediante venta, alquiler o cualquier otra forma de cesión o comunicación pública de la misma, total o parcialmente, por cualquier sistema o en cualquier soporte, ya sea por fotocopia, medio mecánico o electrónico, incluido el tratamiento informático de la misma, en cualquier lugar del mundo.

La vulneración de cualquiera de estos derechos podrá ser considerada como una actividad penal tipificada en los artículos 270 y siguientes del Código Penal.

La protección de esta obra se extiende al mundo entero, de acuerdo a las leyes y convenios internacionales.

© SolidQ™ Press, 2015

Título:

Excel 2013, Power Pivot y DAX.

Tus análisis elevados a la enésima potencia

Autor: **Salvador Ramos**

Serie:

Analiza tu Negocio con Excel y Power BI.

Aprende de tus datos

SolidQ Global S.A.

Apartado de correos 202

03340 Albufera, Alicante, España

<http://www.solidq.com>

AUTOR



Soy **experto en BI & Analytics**
Director de Formación en SolidQ
SQL Server MVP desde el año 2003

¿En qué te puedo ayudar?

Quiero acompañarte en tu transición hacia el mundo del BI & Analytics y Big Data. Tanto si estás dando un giro a tu carrera profesional, como si buscas una mayor especialización en estas áreas.

Estoy especializado en Bases de datos, Data Warehousing, ETL, técnicas de Visualización y en el uso de tecnologías Microsoft (*SQL Server, Integration Services, Analysis Services, Reporting Services, Performance Point Services, Self-service BI, Excel, Power BI, Microsoft Azure*).

Te invito a que conozcas mi [blog personal](#) (pincha [aquí](#)), a que te [suscribas a él](#) para recibir tu regalo (pincha [aquí](#)) y estés al tanto de todas las novedades (*libros y artículos que publico, conferencias en las que participo, cursos y masters que imparto, material que regalo ...*).

Por confiar en mí, te ofrezco gratis mi libro gratuito “Microsoft Business Intelligence: Vea el cubo medio lleno”. Descárgalo [aquí](#).

Sígueme también en: <http://www.salvador-ramos.com>

Linkedin: <http://www.linkedin.com/in/SalvadorRamos>

Twitter: [@salvador_ramos](#)

SOLIDQ



SolidQ, desde el año 2002, suministra servicios para plataformas Microsoft que le ayudan a diseñar, integrar y optimizar su utilización de datos.

Combina una amplia experiencia técnica y de implementación en el mundo real, con un compromiso firme en la transferencia de conocimiento, dada la combinación única de dotes lectivas y experiencia profesional que nuestros mentores ofrecen. De este modo, no solamente ayudamos a nuestros clientes a solventar sus necesidades tecnológicas, sino que somos capaces de incrementar la capacidad técnica de sus profesionales, dándoles una ventaja competitiva en el mercado. Por eso llamamos **Mentores** a nuestros expertos: por su compromiso en asegurar el éxito de su empresa y de sus equipos profesionales a largo plazo.

Nuestros expertos son profesionales reconocidos en el mercado, con más de 100 premios Microsoft MVP (Most Valuable Professional) obtenidos hasta la fecha. Se trata de autores y ponentes en las conferencias más importantes del sector, con varios centenares de ponencias presentadas en conferencias nacionales e internacionales durante los últimos años.

Nuestra misión es la de **transmitir todo el conocimiento adquirido** resolviendo problemas del mundo real para miles de clientes, escribiendo artículos y libros, publicando whitepapers, creando contenidos educativos y formando a decenas de miles de trabajadores de TI en todo el mundo, para que los proyectos de nuestros clientes obtengan los mayores éxitos. Esta transferencia de conocimiento la realizamos fundamentalmente con dos tipos de servicios:

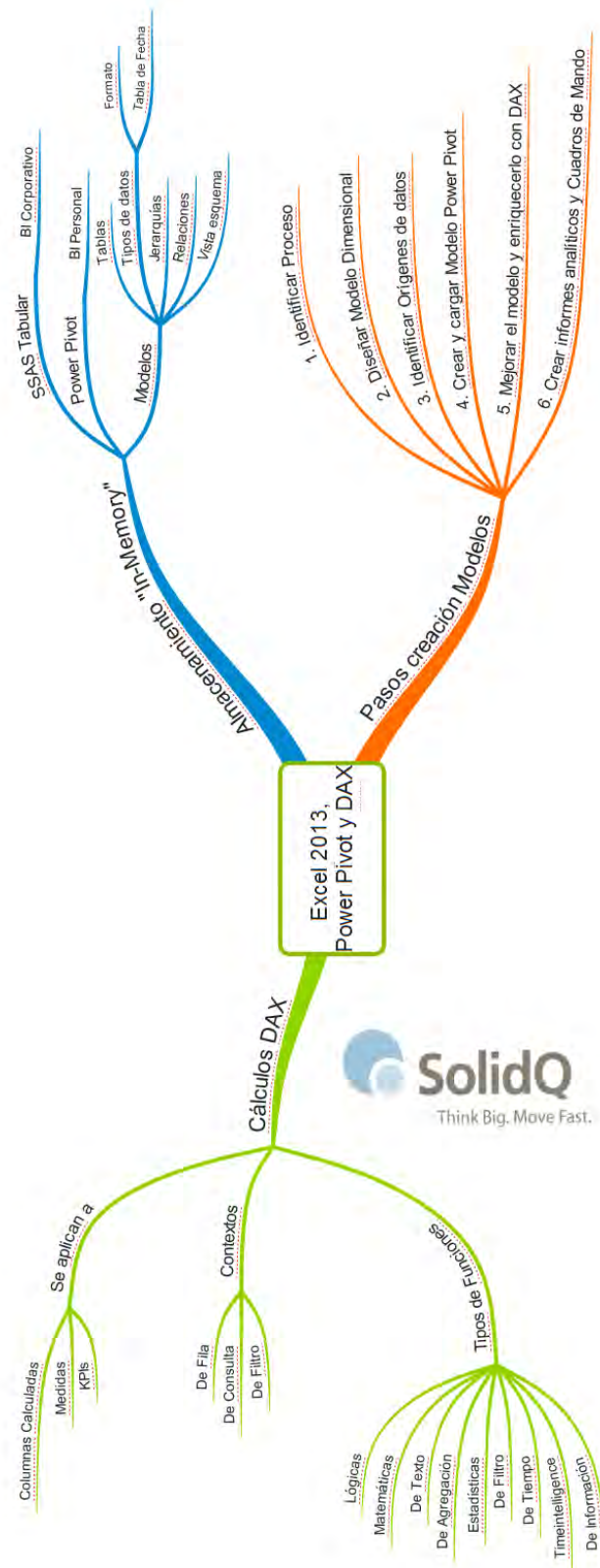
Consultoría: hazlo bien la primera vez (haz clic [aquí](#))

Mentoring: conoce tu potencial personal y mejora tus decisiones (haz clic [aquí](#))

Formación: la mejor inversión posible es pagar por el conocimiento de otros. Conoce nuestro **Plan Formativo** (haz clic [aquí](#)) y nuestro **Calendario** (haz clic [aquí](#))

Publicaciones: ponemos nuestros conocimientos a su alcance. Acceda a nuestros **blogs** (haz clic [aquí](#)) y a nuestras **publicaciones, la mayoría gratuitas** (haz clic [aquí](#))

MAPA MENTAL



ÍNDICE

ADVERTENCIA LEGAL	0
AUTOR	1
SOLIDQ	2
MAPA MENTAL.....	3
Introducción a Power Pivot.....	6
Crear modelos de análisis con Power Pivot.....	12
Menú 'Home'	14
Menú 'Design'	20
Menú 'Advanced'	24
Cargar datos en Power Pivot	28
Iniciación a DAX.....	33
¿En qué punto del camino nos encontramos?	33
¿Qué es DAX?	35
Sintaxis DAX	35
Columnas Calculadas	40
Campos Calculados (Medidas)	43
Uso de columnas calculadas vs campos calculados (medidas)	44
Funciones DAX	46
Funciones heredadas de Excel	46
Funciones de Agregación	47
Funciones de navegación entre tablas a través de las relaciones	51
DAX en la práctica	53
Entendiendo los contextos	53
Contexto de Fila (<i>Row Context</i>)	54
Contexto de Consulta (<i>Query Context</i>)	55
Contexto de Filtro (<i>Filter Context</i>)	56
Funciones DAX	57
Funciones de Agregación terminadas en "X"	57
Funciones ALL() y ALLEXCEPT()	60
Función FILTER().....	61

Función CALCULATE().....	62
Funciones de Inteligencia de Tiempo (Time Intelligence)	65
Ejemplo práctico DAX	68
ENLACES ESENCIALES.....	70
GRACIAS	71

Introducción a Power Pivot

Power Pivot es una tecnología que permite procesar y analizar información compleja de forma muy eficiente y rápida. Su principal característica y la que le aporta una enorme velocidad es que es una tecnología *"In-Memory"*, es decir que tiene todos los datos en memoria con una alta compresión, para disminuir el espacio que ocupa y aumentar la velocidad de análisis. Estos modelos de datos están basados en tablas y relaciones.

Tiene tres "sabores" o presentaciones:

- **Power Pivot para Excel:** es un complemento (*add-in*) para Excel que contiene toda esta tecnología y permite gestionar estos modelos desde unos nuevos menús integrados en el propio Excel. *Su uso está orientado al BI Personal.*
- **Power Pivot para SharePoint:** Permite gestionar hojas de trabajo Power Pivot en un servidor SharePoint. Tanto el procesamiento de datos como la visualización es gestionada en dicho servidor. *Su uso está orientado al BI departamental.*
- **Modelo Tabular de SQL Server Analysis Services (SSAS):** son bases de datos *"In-Memory"* gestionadas por Analysis Services a través de su motor analítico, un servicio orientado a servidores y a su administración por parte de personal de TI cualificado. Su objetivo es disponer de la información corporativa centralizada y servirla a todos los usuarios de la empresa de forma simultánea con unos tiempos de respuesta muy rápidos. Soporta mayores volúmenes de almacenamiento, la gestión está optimizada para uso simultáneo por parte de muchos usuarios. Por último, destacar que permite la automatización y agendado de la carga de información desde los orígenes de datos al modelo. *Su uso está orientado al BI Corporativo.*

A nivel un poquito más técnico, por si el lector desea investigar sobre este modelo de almacenamiento, Microsoft lo ha llamado **"xVelocity"** (anteriormente lo llamó "Vertipaq"); puede buscar en internet por estos dos términos y encontrar gran cantidad de información técnica.

En este momento nos vamos a centrar en el complemento (*add-in*) para Excel, que es el que permite a los usuarios de negocio crear sus propios modelos analíticos y poblarlos con información procedente de muy diversos orígenes de datos, con total independencia del departamento de TI. Además un modelo creado en Excel, se puede compartir en SharePoint o importar desde SSAS (con ayuda de personal

cualificado del departamento de TI), reutilizando todo lo que el usuario de negocio ha desarrollado en Excel.

Power Pivot para Excel es un complemento (*add-in*) para Excel que nos permite integrar y estructurar en tablas la información procedente de diversos orígenes de datos, dicha información se almacena en los archivos “.xlsx” con un alto grado de compresión, y en el momento que se abren los lee y carga por completo en memoria, lo que permite una rapidez de cálculo no vista hasta ahora en Excel, así como la posibilidad de almacenar grandes cantidades de filas, muchos millones de filas, superando así las limitaciones (1 millón de filas aproximadamente) y las lentas velocidades que tenemos cuando hay un gran número de filas en las hojas Excel tradicionales.

Para que se haga una idea de la alta compresión que se puede llegar a alcanzar, vamos a poner un ejemplo con datos reales, realizado por SolidQ. Tenemos un modelo con cuatro tablas:

Clientes	18.848 filas	29 columnas
Fecha	2.191 filas	19 columnas
Productos	606 filas	36 columnas
Ventas	6.341.790 filas	26 columnas

*Ocupando un tamaño de **tan sólo 26,3 MB**. Tenga en cuenta que la compresión puede ser menor, ya que depende de la redundancia de valores de cada columna y en este caso esa redundancia es bastante alta.*

Adicionalmente incluye un potente lenguaje de fórmulas, llamado DAX (Data Analysis Expressions) que incrementa enormemente la capacidad de cálculo analítico y la velocidad de respuesta.

Tanto si usted es un usuario de nivel intermedio de Excel y obtiene los datos de su empresa mediante exportación de informes y consultas desde sus aplicaciones y/o mediante operaciones de copiar y pegar, como si es un usuario avanzado que descubre y lucha día a día con el lenguaje VBA para tratar de obtener soluciones analíticas con cierto dinamismo y sin tener que hacer tareas manuales tediosas y repetitivas para obtener datos actualizados de su empresa y transformarlos para su análisis, Power Pivot es la herramienta que necesita para hacer su vida más fácil, dedicando menos tiempo a tareas tediosas y más tiempo a analizar su información y tomar decisiones más acertadas.

*En resumen, **mayor almacenamiento, facilidad para obtener datos externos, gran aumento de la velocidad de cálculo y nuevo lenguaje de fórmulas con mucha más potencia de cálculo.***

Power Pivot es la nueva herramienta de almacenamiento y análisis de la información orientada al usuario final y que está totalmente integrada con Excel. Como puede comprobar en la siguiente imagen su interfaz es un nuevo menú en la “cinta” (Ribbon) de Excel.

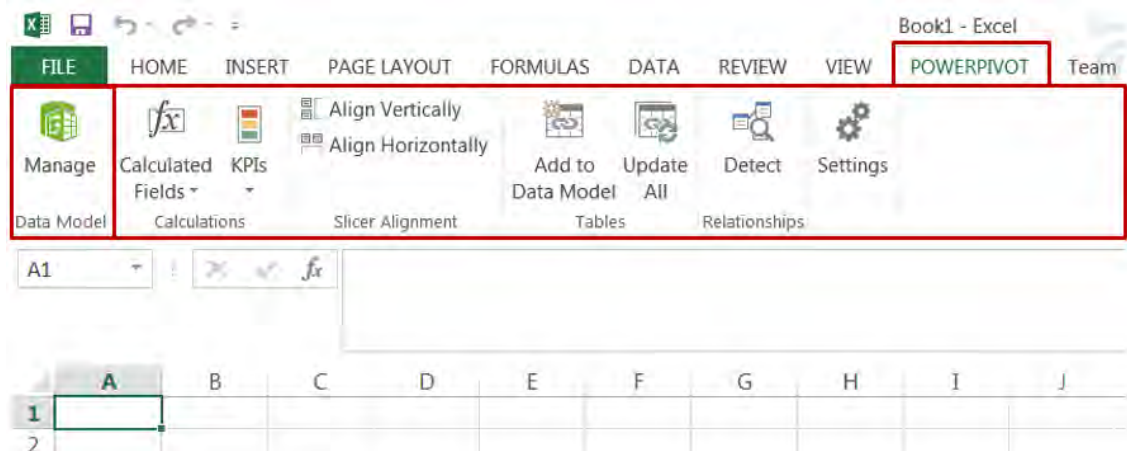


Figura 0-1 Power Pivot Ribbon en Excel.

El complemento Power Pivot no viene habilitado por defecto en Excel 2013. Para habilitarlo debe seguir una serie de pasos que se resumen en la siguiente imagen:

- Excel 2013

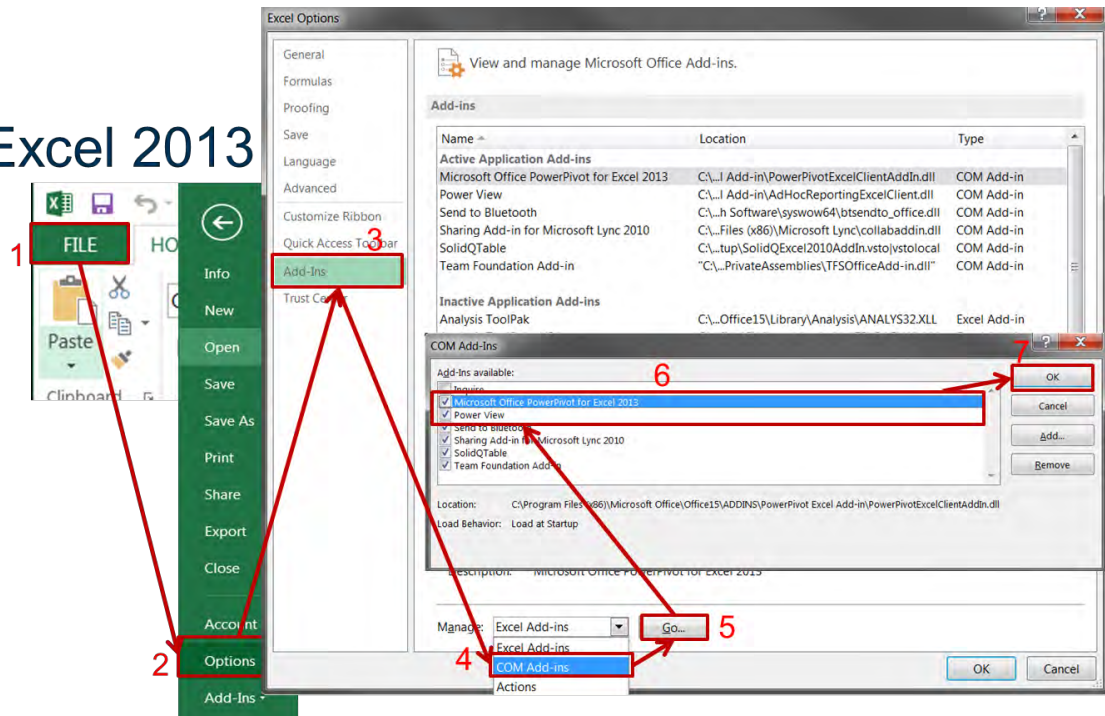


Figura 0-2 Pasos a realizar para habilitar Power Pivot en Excel 2013.

Uno de los principales elementos es la opción “Administrar” (*Manage*) que da acceso a una nueva ventana con todos los elementos necesarios para su gestión. En la siguiente imagen puede ver dicha ventana.

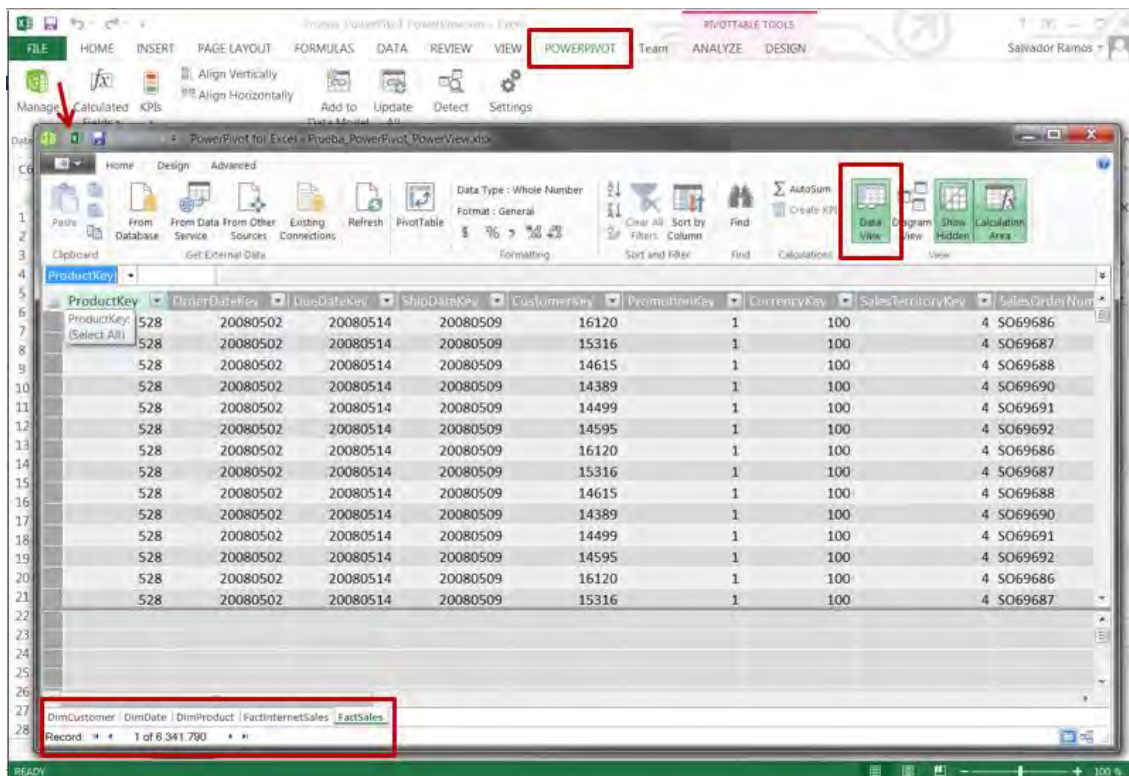


Figura 0-3 Ventana de Administración en Power Pivot.

Fíjese en la parte inferior izquierda, donde indica que hay en esa tabla 6.341.790 registros, algo imposible en una hoja Excel tradicional.

Toda la información, tanto de Power Pivot como de Excel tradicional se almacena en el mismo fichero “.xlsx”. Tenga en cuenta que la función de autoguardado de Excel no aplica a la ventana de Power Pivot, por lo que deberá guardar los cambios de forma manual para evitar perderlos ante un cierre inesperado de la aplicación.

Por último vamos a ubicar **Power Pivot** junto con el resto de componentes de Power BI para Excel en la siguiente imagen, para seguir trabajando en la comprensión desde una perspectiva global. Observe que hemos marcado con un círculo rojo su ubicación:

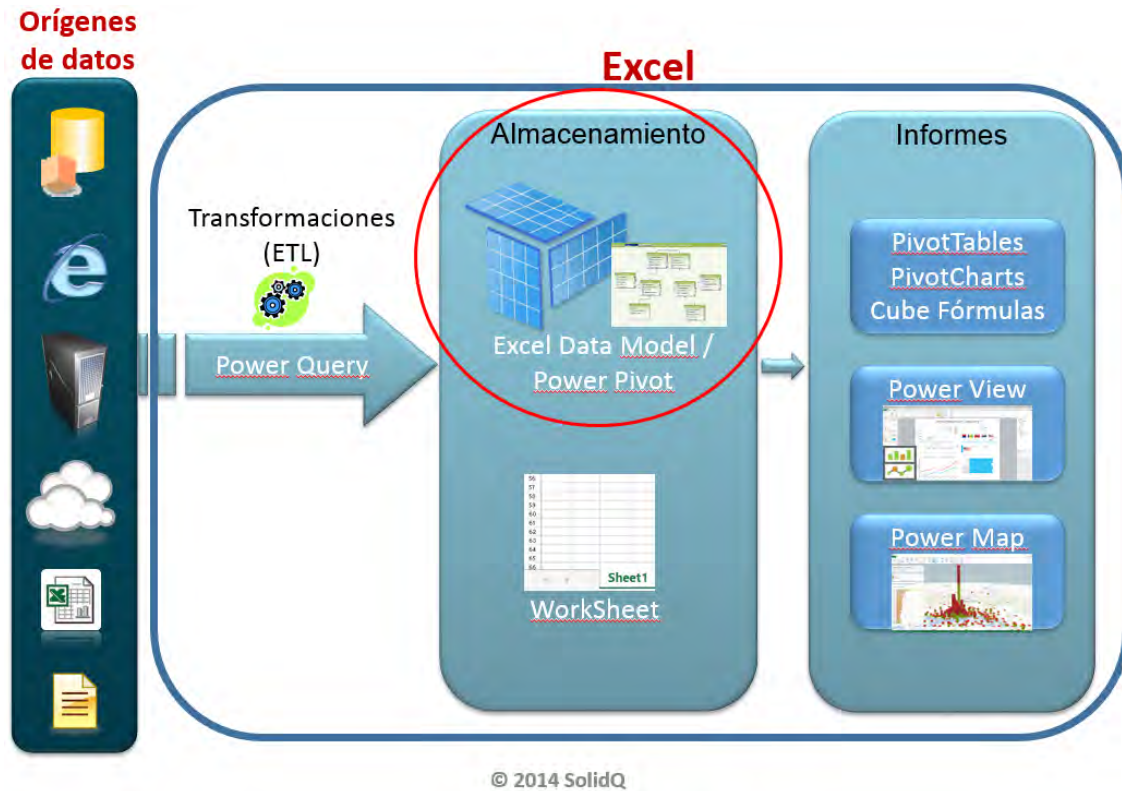


Figura 0-4 Componentes de Excel Power BI. Power Pivot.

Power Pivot es el núcleo de los componentes de Power BI para Excel, sin él no tiene sentido la existencia del resto de componentes, ya que todas las demás herramientas la necesitan como punto de partida o de destino:

1. **Power Query**: lee datos de los orígenes, los transforma y los almacena en *Power Pivot*.
2. **Power View**: es una herramienta para visualizar información almacenada en *Power Pivot* y en servidores con *SQL Server Analysis Services*.
3. **Power Map**: es una herramienta para visualizar y analizar información geolocalizada en diversos tipos de mapas, a partir de información almacenada en *Power Pivot* y en servidores con *SQL Server Analysis Services*.

Crear modelos de análisis con Power Pivot

A partir de este momento, ya centrados en Power Pivot para Excel, vamos a ir conociendo todos los detalles necesarios para la creación de modelos analíticos.

Power Pivot tiene la peculiaridad, con respecto a otras herramientas similares que hay en el mercado, que no creamos como tal y desde cero una estructura de tablas y columnas, con sus tipos de datos y características desde un editor específico para ello. Para que entienda el lector a qué tipo de herramientas me refiero, voy a poner una pantalla de Microsoft Access donde se van definiendo las características de cada tabla de la base de datos, y las columnas que la componen, de forma visual:

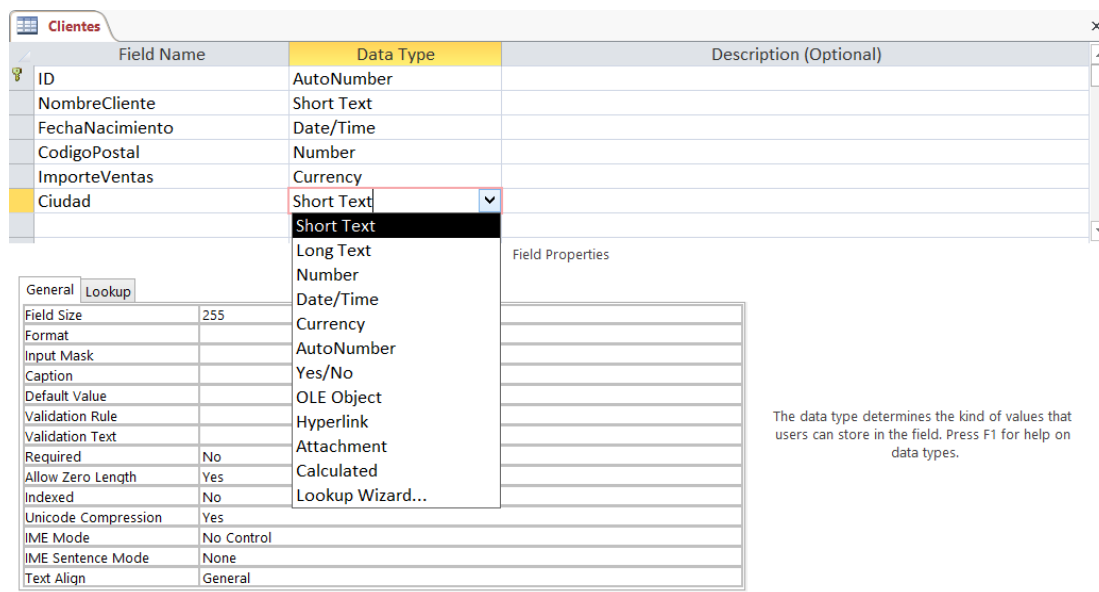


Figura 0-1 Diseño de Tablas en Microsoft Access.

En Power Pivot tenemos dos formas de generar estas estructuras de almacenamiento:

- Mediante **Power Query**, herramienta que hemos estudiado anteriormente, que permite extraer datos de los orígenes, aplicar una serie de transformaciones sobre ellos y guardarlos en el destino. Pues bien, cuando indicamos que el proceso utilice como destino Power Pivot, lo que ocurre es que se crea una estructura de almacenamiento, tal y como la hemos definido en Power Query y una vez creada importa los datos seleccionados en dicho proceso.
- Mediante las opciones para **“Obtener datos externos”** que nos proporciona Power Pivot (aún no las hemos estudiado, se estudiarán más adelante).

Una vez creadas esas estructuras, podemos acceder a ellas abriendo la ventana de administración de Power Pivot, para ello pulsamos es la opción “Administrar” (*Manage*), que se encuentra en la parte izquierda del menú “POWERPIVOT”, desde la que tenemos acceso a todas las opciones que nos proporciona.

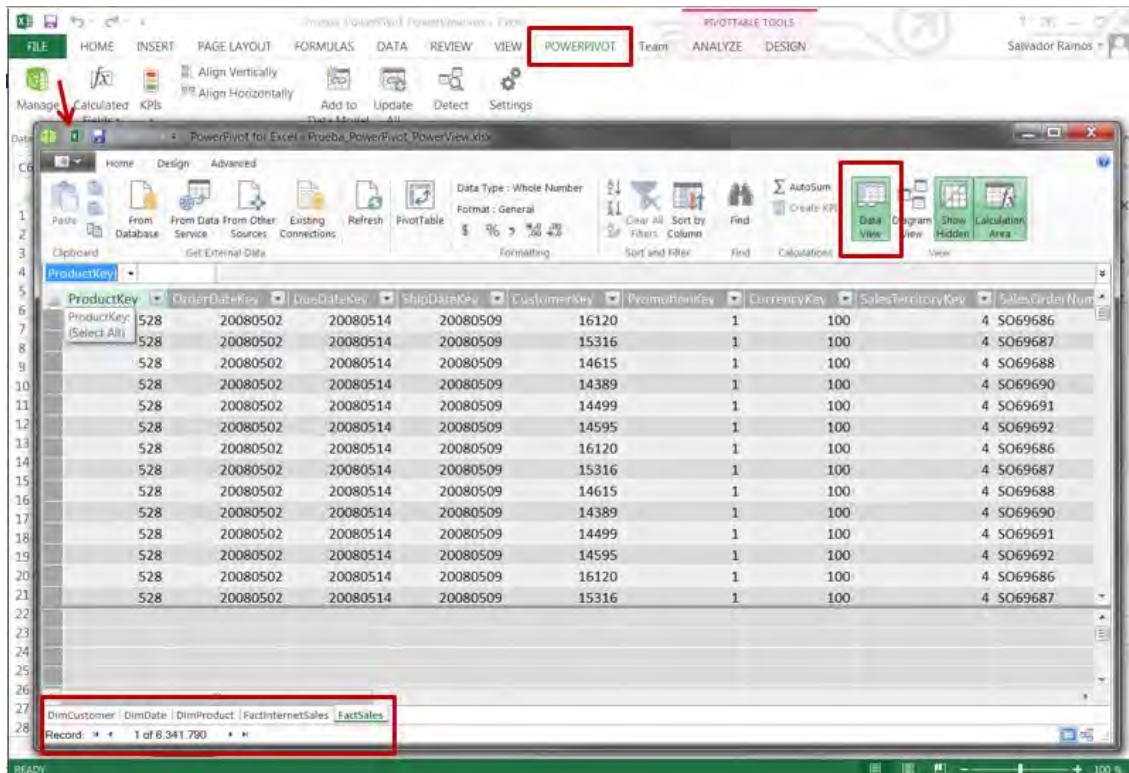


Figura 0-2 Ventana de Administración en Power Pivot. Vista de Datos.

A continuación estudiaremos cada uno de esos menús de la “cinta” (*Ribbon*) y cada uno de los grupos de opciones y opciones que lo componen.

Menú 'Home'

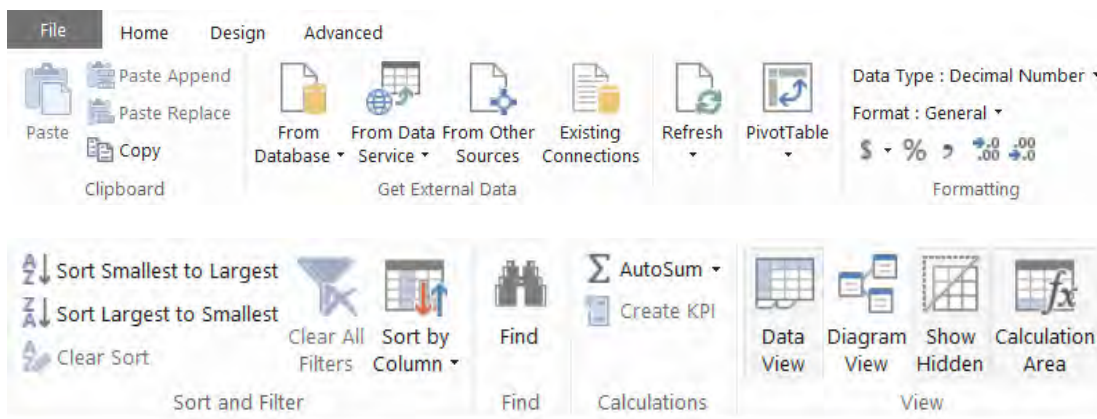


Figura 0-3 Ribbon, menú 'Home'.





Grupo de opciones '**Clipboard**': son opciones relativas a la obtención de datos desde el portapapeles. Se estudiará más adelante.

Grupo de opciones '**Get External Data**': son opciones relativas a la obtención de datos desde diversos orígenes desde el propio Power Pivot, sin necesidad de utilizar Power Query. Se estudiará más adelante.

Grupo de opciones '**Refresh**': son opciones relativas a la actualización de datos del modelo, permite actualizar los datos de todo el modelo (opción **Refresh All**) o sólo de la tabla seleccionada (opción **Refresh**). Elimina totalmente el contenido y lo vuelve a extraer del origen, transformar y cargar. No permite cargar incrementales, sólo cargas completas.

Grupo de opciones '**PivotTable**': permite crear tablas dinámicas y/o gráficos dinámicos conectados al modelo con un solo clic, abriéndonos de nuevo la ventana de Excel y mostrándonos las tablas y gráficos listos para hacer consultas sobre el modelo. Simplemente es una opción que mejora la usabilidad y nos evita hacer todo lo anterior paso a paso.

Grupo de opciones '**Formatting**': son opciones relativas a la consulta al origen de datos.

- **Data Type:** permite cambiar el tipo de datos que se ha asignado a cada columna. Es muy importante aplicar los tipos de datos correctos, es una buena práctica y ayuda a optimizar el almacenamiento y los tiempos de respuesta. Los tipos de datos disponibles son:
 - Texto
 - Fecha
 - Número Decimal
 - Número Entero
 - Moneda
 - TRUE/FALSE
- **Format:** permite elegir entre una serie de formatos que dependerán del tipo de datos de dicha columna.
Para tipos de datos numéricos se puede elegir entre los siguientes formatos:
 - General
 - Número Decimal
 - Número Entero
 - Moneda
 - TRUE/FALSEPara tipos de datos Date, Date/Time y Time se puede elegir entre una gran variedad de formatos, más o menos extensos, para representar esa fecha/hora.
-  : asignación de símbolo de moneda. Este y los siguientes iconos, se comportan de igual forma que en las hojas Excel tradicionales.
-  : visualización como porcentaje, tenga en cuenta que multiplica por 100 el valor almacenado en Power Pivot.
-  : formato con separador de miles.
-  : aumenta o disminuye el número de decimales.

Vamos a profundizar un poquito más en lo relacionado a los tipos de datos. Cada columna tiene asignado un tipo de datos, que por defecto es asignado automáticamente según el tipo de datos de cada columna del origen, pero debe revisarlo manualmente, a fin de mejorarlo si es posible, y reasignarle un tipo de datos más apropiado si lo considera oportuno y, por supuesto, si es compatible y permite almacenar los valores obtenidos del origen. Con esto me refiero a que por ejemplo si tiene una columna con nombres de clientes no podrá convertirla en tipo

de dato numérico. Estos cambios de tipos de datos se pueden realizar se han de realizar de forma manual en Power Pivot, una vez creada la tabla y sus columnas tras un primera importación de los datos (que es cuando se asignan los tipos de datos). Debe tener en cuenta los siguientes aspectos en cuanto a tipos de datos y formato:

- El tipo de datos afecta al almacenamiento, mientras que el formato sólo afecta a la visualización. No puede asignar tipos de datos que no permitan almacenar los valores que tienen los datos de esa columna.
- No asigne formatos incoherentes para el tipo de datos elegido, aunque en algunos casos esté permitido. No tiene sentido tener un tipo de datos 'Número Entero' y en formato asignarle dos decimales (aunque siempre valdrán ".00" es confuso).
- El formato 'Moneda' no hace conversiones: ¿ 142,84€ = 142,84\$?

Grupo de opciones '**Sort and Filter**': son opciones relativas a la ordenación y filtrado de los datos.

- **A-Z:** realiza una ordenación ascendente por la columna selecciona.
- **Z-A:** realiza una ordenación descendente por la columna selecciona
- **Clear Sort:** elimina cualquier ordenación que hayamos habilitado con las opciones anteriores.
- **Clear All Filters:** elimina los filtros establecidos, quedando visibles todas las filas de la tabla.
- **Sort by Column:** permite ordenar los datos de una columna por los valores de otra columna diferente. Aunque a priori pueda parecer algo muy extraño, la verdad que es muy útil, veamos un ejemplo de uso: tenemos una columna con el nombre del mes, en orden ascendente, por ella el primer mes sería abril y el segundo sería agosto (orden alfabético de los nombres de los meses), pero necesitamos el orden cronológico y que el primer mes sea enero, el segundo febrero, y así sucesivamente. ¿Cómo lo solucionamos? Pues debemos tener una columna 'NombreMes' con los nombres de los meses y otra 'Mes' con los números de los meses del 1 al 12; entonces utilizamos esta opción y le decimos que la columna 'NombreMes' la ordene por los valores de la columna 'Mes'.

Opción '**Find**': permite hacer búsquedas en el contenido de las tablas del modelo. Es equivalente a dicha opción en Excel.

Grupo de opciones '**Calculations**': son opciones que nos permiten crear de forma rápida campos calculados y KPIs. Los estudiaremos en la parte dedicada a DAX.

Grupo de opciones '**View**': da acceso a las diferentes opciones de visualización:

- **Data View**: muestra la vista de datos (vea la figura anterior donde se muestra *Ventana de Administración en Power Pivot*).
- **Diagram View**: muestra la vista diagrama (la puede ver en la siguiente imagen). En ella se ven las tablas y relaciones en forma de esquema. Además es en esta ventana donde se gestionan las *Jerarquías*, a través del icono que hay en la parte superior izquierda de cada tabla (recuerde que sólo aparece cuando dicha tabla está seleccionada).
- **Show Hidden**: muestra u oculta las columnas que hemos marcado como "ocultas en herramientas cliente". Estas columnas son aquellas que necesitamos internamente, habitualmente para cálculos y establecimiento de relaciones, pero que no aportan nada al usuario durante el análisis, sino al contrario introducen "ruido" al ser columnas que ni conoce ni va a utilizar.
- **Calculation Area**: muestra u oculta el "área de cálculo", que es esa rejilla que hay en la parte inferior y en la que definimos los campos calculados mediante DAX (esta área la estudiaremos más adelante).

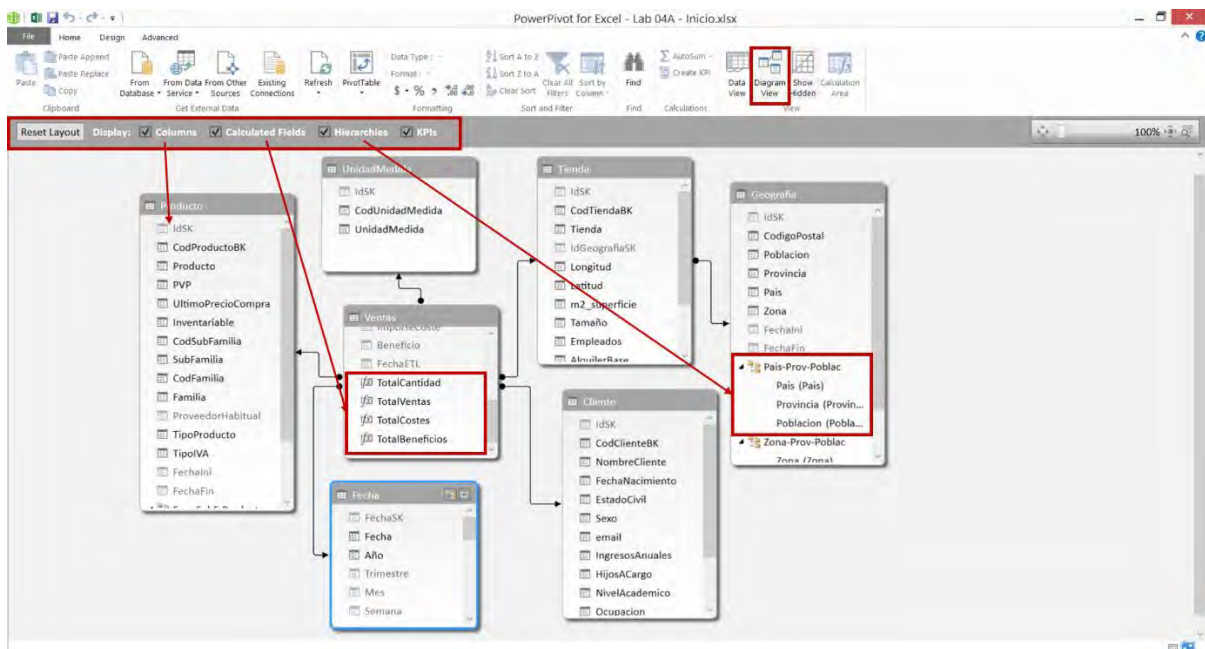
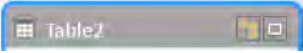



Figura 0-4 Ventana de Administración en Power Pivot. Vista de Diagrama.

Creación de Jerarquías

Para crear una jerarquía debe situarse en la vista diagrama y hacer clic sobre la tabla en concreto sobre la que quiere crear la jerarquía, verá que se marca en diferente color y que aparecen dos iconos en la parte superior, a la derecha del nombre de la tabla , debe hacer clic sobre el icono , a continuación debe asignar nombre a dicha jerarquía e ir arrastrando columnas de esa misma tabla para crear los diferentes niveles que deseamos que tenga. Tenga en cuenta que una jerarquía no puede utilizar columnas que pertenezcan a otra tabla diferente a donde la ha creado.

En los Laboratorios guiados paso a paso de los [cursos y masters de SolidQ](#) se verá con detalle y se practicará, tanto la creación de jerarquías, como el resto de opciones estudiadas hasta el momento aplicándolas sobre casos reales.

Menú 'Design'

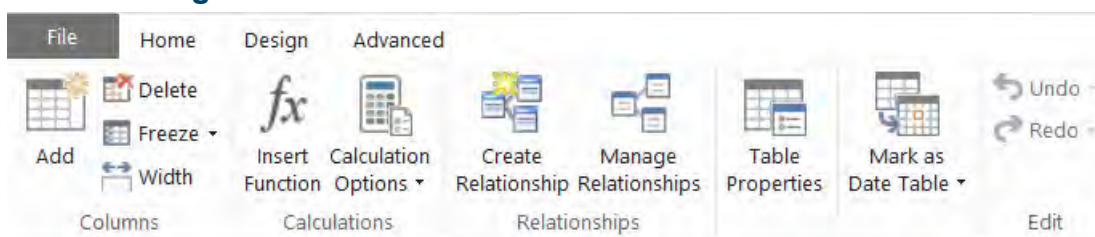


Figura 0-5 Ribbon, menú 'Design'.

Grupo de opciones '**Columns**': son opciones relativas al tratamiento de las columnas de una tabla en la ventana de Administración de Power Pivot.

- **Add**: permite agregar columnas calculadas a una tabla. Esta opción se verá con detalle cuando estudiemos la parte de DAX.
- **Delete**: permite eliminar columnas de una tabla.
- **Freeze**: permite fijar columnas, para que se mantengan visibles al hacer *scroll* horizontal. No afecta en absoluto a cómo lo verán los usuarios en las tablas dinámicas, sólo afecta a la visualización de la tabla en la ventana de Administración de Power Pivot
- **Width**: permite ajustar el ancho de la columna en píxeles. Sólo afecta a la visualización de la tabla en la ventana de Administración de Power Pivot.

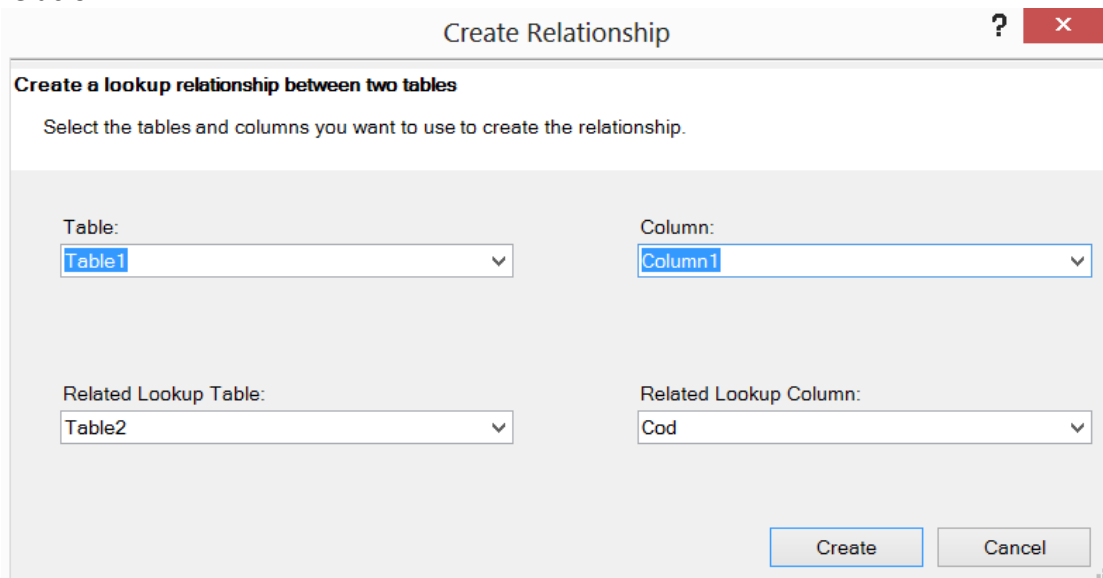
Grupo de opciones '**Calculations**': son opciones relativas a la realización de cálculos mediante expresiones DAX.

- **Insert Function**: aplica a la barra de fórmulas, nos sirve de ayuda a la hora de escribir expresiones DAX, permitiéndonos usar un formulario de búsqueda sobre las funciones disponibles.
- **Calculation Options**: permite que los cálculos se realicen de forma automática, es decir, cada vez que introduzca o modifique una expresión DAX, se recalculan los valores. La ventaja es que tiene inmediatamente el resultado, pero conforme va teniendo más expresiones DAX en el modelo, más lento se va haciendo cualquier cambio (recalcula todas, no sólo la que ha modificado). Por ello, desde aquí se puede cambiar la opción a 'Modo de cálculo Manual', con lo que sólo se realizarán los cálculos cuando usted decidas, es decir, cuando pulse la opción que hay en ese mismo botón, llamada 'Calcular ahora'. Es una práctica habitual quitar el 'Cálculo

automático' mientras estamos diseñando los modelos, consiguiendo así menores tiempos de espera al escribir DAX.

Grupo de opciones '**Relationships**': son opciones relativas a la gestión de Relaciones entre tablas.

- **Create Relationship**: una relación se establece en base a una columna de una tabla que se relaciona con otra columna de otra tabla. Ambas han de ser del mismo tipo de datos. Este es el formulario que nos permite crear una relación:



- **Manage Relationships**: permite crear, modificar o borrar relaciones.

También se pueden crear relaciones desde la '**Vista Diagrama**' pinchando en la columna origen de la relación y arrastrando hasta soltar sobre la columna destino de la relación, tal y como se ve en la imagen:



Al soltar el ratón creará la relación entre dichas columnas de ambas tablas.

Las **Relaciones**, además de ser entre columnas de tablas diferentes y que tengan el mismo tipo de datos, debemos crearlas de forma coherente. Por ejemplo, aunque el código de producto sea numérico y el código de proveedor también, no tiene ningún sentido establecer una relación entre ambos (aunque la herramienta me lo permita), además, obtendríamos unos datos analíticos totalmente falsos e incoherentes. Por ello, a la hora de establecer relaciones, vamos a hacerlo en base a las relaciones definidas en nuestra documentación del modelo, que es donde ya está estudiado con detalle y plasmado por escrito qué relaciones debe haber entre las tablas del modelo y a través de qué columnas se han de realizar, todo ello aplicando las buenas prácticas de modelado estudiadas. Aquí meramente haremos la parte operativa de crearlas físicamente en Power Pivot en base a la documentación.

A continuación vamos a citar una serie de limitaciones que tenemos a la hora de crear estas relaciones, y que en parte afectan al diseño del modelo, aunque si seguimos las buenas prácticas de modelado estudiadas, se nos dará esta casuística en menos ocasiones:

- Sólo se puede utilizar una columna por cada tabla que interviene en la relación. Aunque este aspecto se ve reflejado en el propio formulario, donde sólo puede seleccionar una columna, queremos destacarlo aquí, porque hay otras herramientas que permiten que en la relación intervengan varias columnas de cada tabla.
- Sólo puede haber una relación activa entre dos tablas. Por ejemplo, si tengo una tabla de ventas con tres fechas (fecha venta, fecha envío y fecha cobro) tendré que establecer una relación por cada una de estas columnas con la tabla de la dimensión 'Fecha', pero sólo una de ellas podrá estar activa, quedando las otras dos como inactivas. Una relación activa es la que utiliza por defecto Power Pivot para obtener información de dos tablas relacionadas. Si necesitamos utilizar una relación inactiva, tendremos que hacerlo mediante expresiones DAX en las que se indique expresamente la relación inactiva a utilizar.
- Sólo admite relaciones uno a uno o uno a muchos.
- No se permiten relaciones entre dos columnas de la misma tabla (relaciones auto referenciadas).
- No se permiten relaciones circulares, es decir, si tabla1 se relaciona con tabla2 y tabla2 con tabla3, tabla3 no la podemos relacionar nuevamente con tabla1.

La correcta creación de las Relaciones entre tablas, es uno de los aspectos que más debemos cuidar. Cualquier error cometido aquí tiene la grave consecuencia de que obtendremos información errónea y estaremos falseando los resultados de nuestros análisis.

Opción '**Table Properties**': permite editar las propiedades de la tabla. Como ya explicamos anteriormente, no hay un diseñador para ello, sino que la estructura de la tabla proviene del resultado de la obtención y transformación de datos desde Power Query o desde las opciones de 'obtener datos externos' incluidas en Power Pivot. Si hemos utilizado Power Query, en el formulario no se podrán hacer modificaciones del proceso realizado con Power Query, sino que habrá que ir a Power Query y hacer los cambios oportunos. Si el proceso se hizo con alguna de las opciones del menú 'Obtener datos externos' (*Get External Data*) de Power Pivot, desde aquí tendremos acceso para consultarlo y modificarlo.

Opción '**Mark as Date Table**': permite seleccionar cuál es la tabla de Fecha. Esta tabla es la correspondiente a la dimensión Fecha, que ya estudiamos con detalle anteriormente. Mediante esta opción debemos asignar dicha tabla y marcar cuál es la columna con tipo de datos Fecha. A partir de ese momento, quedan habilitadas las funciones de inteligencia de tiempo que forman parte del lenguaje DAX. *Básicamente, en Power Pivot, a nivel de creación del modelo, sólo hay que hacer esta asignación. El grueso del trabajo está en incluir en el modelo dimensional una dimensión 'Fecha' que recopile todas las necesidades de tratamiento del tiempo propias de cada negocio y, posteriormente, incluir los cálculos mediante expresiones DAX que ayuden a enriquecer el análisis.*

Grupo de opciones '**Edit**': son opciones para rehacer y deshacer.

- **Undo**: deshacer.
- **Redo**: rehacer.

Menú 'Advanced'

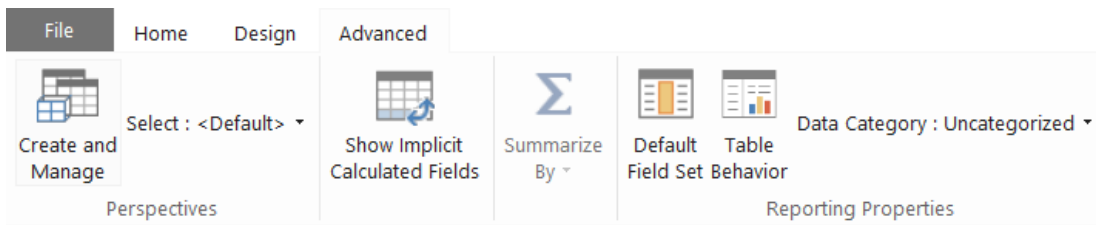


Figura 0-6 Ribbon, menú 'Advanced'.

Grupo de opciones '**Perspectives**': son opciones relativas a la gestión de perspectivas. Las **Perspectivas** son subconjuntos de elementos del modelo que permiten simplificar la navegación. Supongamos que tenemos un modelo de ventas con diversa información relativa a las Tiendas, pero no toda ella aporta valor para el análisis que va a realizar nuestro equipo comercial, aunque sí es útil para otros departamentos. Por tanto podemos hacer una vista personalizada que sólo muestre los elementos seleccionados que necesitan los comerciales, evitando así que les cueste encontrar lo que necesitan entre mucha información que nunca usarán. En la siguiente imagen se ve que la *perspectiva* llamada 'Comerciales', donde en vez de tener acceso a todos los datos de las tiendas, sólo se les ofrecerán los campos: CodTiendaBK, Tienda, Empleados y Tamaño:

Campos	Comerciales
- Tablas	<input type="checkbox"/>
- Tienda	<input type="checkbox"/>
AlquilerAñoContrato	<input type="checkbox"/>
AlquilerBase	<input type="checkbox"/>
CodTiendaBK	<input checked="" type="checkbox"/>
Empleados	<input checked="" type="checkbox"/>
FechaFin	<input type="checkbox"/>
FechaIni	<input type="checkbox"/>
IdGeografiaSK	<input type="checkbox"/>
IdSK	<input type="checkbox"/>
Latitud	<input type="checkbox"/>
Longitud	<input type="checkbox"/>
m2_superficie	<input type="checkbox"/>
Tamaño	<input checked="" type="checkbox"/>
Tienda	<input checked="" type="checkbox"/>

Figura 0-7 Selección de campos de una Perspectiva.

- **Create and Manage:** permite crear y administrar *perspectivas*.
- **Select:** permite seleccionar una de las perspectivas existentes. Cuando se selecciona una perspectiva sólo se ven los campos que hay marcados en ella.
- **Show implicit Calculated Fields:** muestra u oculta los campos calculados implícitos. Estos campos son aquellos que se crean cuando arrastramos una columna numérica a la zona de valores de una tabla dinámica o gráfico dinámico, utilizando por defecto la función de agregación suma. A partir de su primer uso, aparecen en la rejilla de la ventana de administración de forma automática. Se puede evitar, en gran medida, la aparición de campos calculados implícitos (*Implicit Calculated Field*), aplicando una buena práctica que consiste en crear todos los campos calculados que necesitemos y asignándoles el formato apropiado, para a continuación, marcar las columnas a partir de las que se calculan, como no visibles en herramientas cliente (al no estar visibles no las podremos utilizar y, por tanto, no se crearán campos calculados implícitos a partir de ellas).
- **Sumarize By:** indica la función de agregación (suma, min, max, ...) que utilizará por defecto cuando cree un campo calculado implícito por esa columna.
- **Default Field Set:** es el conjunto de campos predeterminado que mostrará en las herramientas cliente cuando seleccionemos directamente una tabla del modelo, además lo hará en el orden que aparezcan en el cuadro de la parte derecha del formulario. Si no definimos este conjunto de campos, mostrará todos los campos de dicha tabla, en el mismo orden que aparecen en ella.
- **Table Behavior:** permite definir ciertas opciones avanzadas de visualización en herramientas cliente. Por ejemplo, podemos configurar el comportamiento de algunas características de esta tabla cuando la utilicemos en Power View.

A modo de resumen mostramos el camino andado hasta el momento con lo visto sobre Power Pivot, para que se haga una idea del camino que nos queda por andar, y que lo recorreremos en el apartado dedicado al Lenguaje DAX:

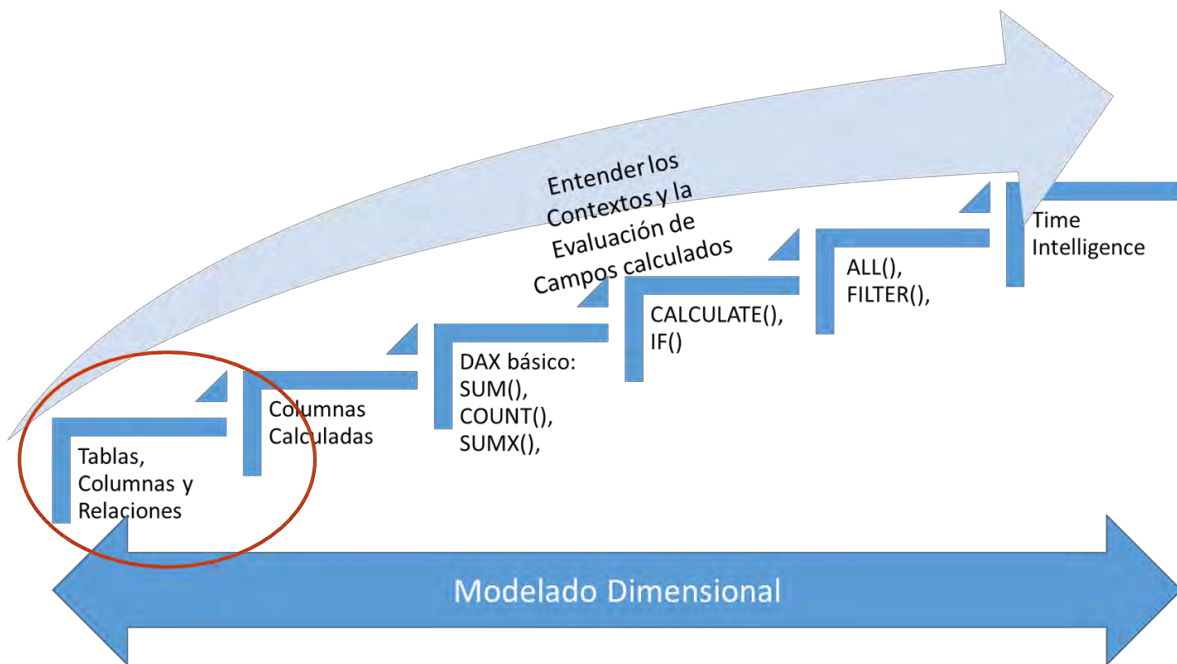


Figura 0-8 Evolución y proceso de aprendizaje de Power Pivot y DAX.

Todas estas opciones avanzadas, para entender con detalle todas sus posibilidades, necesitan ir las definiendo e ir comprobando a continuación su comportamiento en las herramientas cliente, principalmente tablas dinámicas, gráficos dinámicos y Power View. Es complicado definir las aquí sin ir detallando todo paso a paso, con ejemplos de uso, y ese no es el objetivo de este libro. Si desea alcanzar dicho objetivo, le recomiendo que realice algunos de los [cursos y Masters ofrecidos por SolidQ](#), donde en los “Laboratorios guiados paso a paso” se abordarán este tipo de prácticas didácticas sobre casos reales; o bien sobre uno de los modelos que haya diseñado, vaya haciendo cambios en la configuración y comprobando los resultados y cambios en la visualización que generan de forma autodidacta.

Exponemos aquí un resumen de los pasos que recomendamos que siga para la creación de un modelo de análisis con Power Pivot:

1. Disponer del diseño lógico del modelo dimensional a crear. Es muy importante haber aplicado las buenas prácticas estudiadas, nos evitará muchos problemas, así como complejidades innecesarias en las expresiones DAX que se realizarán más adelante.
2. Utilizar Power Query y/o las herramientas de obtención de datos propias de Power Pivot para extraer los datos de los orígenes, transformarlos según el modelo establecido y guardarlos en Power Pivot.
3. Utilizar las opciones de diseño que ofrece Power Pivot, aplicando las buenas prácticas de diseño estudiadas, para mejorar el modelo, es decir:
 - Asignar tipos de datos idóneos y definir formatos que mejoren la visualización.
 - Administrar las relaciones entre tablas (basándonos en lo definido en la documentación del modelo).
 - Crear jerarquías (basándonos en lo definido en la documentación del modelo).
 - Hacer un tratamiento adecuado del tiempo (*Tabla de Fecha*).
4. Enriquecer el modelo con DAX, creando columnas calculadas, campos calculados (medidas) y KPIs.

Hasta el momento hemos estudiado cómo realizar la mayor parte del proceso, sabemos crear modelos dimensionales y hacer el diseño lógico, sabemos realizar procesos ETL (extracción, transformación y carga) mediante Power Query y hacer las mejoras oportunas en el modelo.

Nos queda pendiente la obtención de datos externos desde el propio Power Pivot, y conocer el lenguaje DAX para así poder enriquecer el modelo. Estos puntos serán los que veamos a continuación.

Cargar datos en Power Pivot

Llegados a este punto el lector se preguntará *¿Por qué tengo disponible un menú para obtener datos externos dentro del propio Power Pivot, si dispongo de Power Query?*

La verdad que es una pregunta totalmente lógica y que nosotros también nos hemos hecho, el principal motivo que encontramos es que cuando salió Power Pivot en el año 2010 no existía Power Query (que apareció en 2014), por tanto, la propia herramienta incorporaba la posibilidad de obtener datos externos de diferentes fuentes.

A día de hoy, hay una gran cantidad de archivos Excel que han utilizado estas opciones en ellos y por otro lado muchos usuarios que no se han instalado Power Query, bien por desconocimiento (hay que descargarlo e instalarlo por separado), bien por evitar un nuevo aprendizaje. Por tanto, a día de hoy, y esto es opinión personal: *es complicado eliminar estas opciones, aunque creemos que en un futuro sí las eliminarán e irán alineando todo en un mismo lugar*, porque básicamente, las funcionalidades disponibles en este menú son un subconjunto de las que hay disponibles en Power Query; salvo pequeñas excepciones, habiendo algunos orígenes de datos (como informix y algún otro) que sí que están en Power Pivot y no en Power Query.

Si usted ya conoce Power Query, tiene experiencia con la herramienta y lo tiene instalado en los equipos con los que trabaja, no tiene necesidad de utilizar ninguna de estas funcionalidades desde Power Pivot. De todas formas si decide utilizarlas, con los conocimientos adquiridos sobre Power Query, no habrá ni tan siquiera curva de aprendizaje, simplemente pulse sobre las opciones del menú y vaya siguiendo las indicaciones, no encontrará ningún problema en entender lo que la herramienta le va a ir solicitando. Lo único que encontrará es que en el proceso de obtención de datos habrá menos pasos y menos opciones, con lo que se verá limitado y no podrá tener tanta flexibilidad como cuando utiliza Power Query.

Siempre habrá ocasiones, que tanto en Power Pivot como en Power Query, necesite apoyarse en el personal de TI para que le facilite los datos de conexión a los servidores corporativos.

Aun habiendo comentado que su uso es opcional, vamos a ir viendo las opciones disponibles.

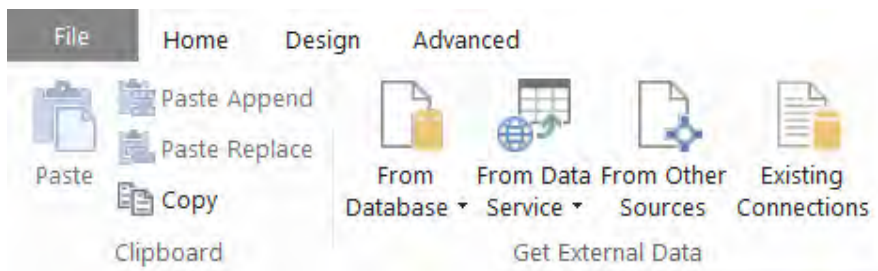


Figura 0-1 Ribbon, menú 'Home', grupos de opciones 'Clipboard' y 'Get External Data'.

Grupo de opciones '**Clipboard**': permiten pegar tablas de datos desde el 'portapapeles', quedando almacenadas como tablas dentro de Power Pivot. Es una opción disponible y que a priori puede parecer útil y cómoda de utilizar para usted, pero le aconsejo que no la utilice porque tiene más inconvenientes que ventajas. Una vez creada la tabla no tiene ninguna posibilidad de hacer cambios en ella (le recuerdo que esta es una característica de Power Pivot), la forma de actualizarla es refrescando con los datos que hay actualmente en origen. Pero hay una excepción, que es precisamente cuando proceden del portapapeles, ya que éste sólo se almacena en la memoria RAM y por ello no queda ningún vínculo con el origen una vez creada la tabla. Si desea actualizar la tabla deberá volver a copiar los datos al portapapeles y utilizar una de las siguientes opciones:

- **Paste**: crea una nueva tabla a partir de los datos del portapapeles, en ella sólo podemos elegir su nombre y si la primera fila contiene encabezados o no. Posteriormente podemos hacer mejoras igual que sobre cualquier otra tabla (cambiar nombres de columnas, tipos de datos, asignar formatos, etc.).
- **Paste Append**: pega los datos, pero agregándolos y manteniendo los datos que ya había en la tabla. Sólo funciona sobre tablas que se crearon pegando desde el portapapeles.
- **Paste Replace**: pega los datos dejando sólo éstos y eliminando todos los datos que hubiese anteriormente en la tabla. Sólo funciona sobre tablas que se crearon pegando desde el portapapeles.
- **Copy**: permite copiar las celdas seleccionadas al portapapeles. Funciona como en Excel, o como en cualquier otra aplicación que permita copiar al portapapeles.

Grupo de opciones '**Get External Data**':


- **From Database:** tiene a su vez un menú desplegable con las siguientes opciones:
 - **From SQL Server:** permite obtener datos de Microsoft SQL Server.
 - **From Access:** permite obtener datos de Microsoft Access.
 - **From Analysis Services or PowerPivot:** permite obtener datos desde SQL Server Analysis Services (SSAS) o desde modelos PowerPivot de SharePoint. Es importante que recuerde que no se pueden obtener datos almacenados en el PowerPivot de otro archivo Excel.





- **From Data Service:** tiene a su vez un menú desplegable con las siguientes opciones:
 - **From Windows Azure Marketplace:** permite obtener datos de Microsoft SQL Server.
 - **Suggest Related Data:** hace sugerencias de Fuentes de datos relacionados existentes en el Azure Marketplace.
 - **From OData Data Feed:** permite obtener datos almacenados en fuentes de datos Open Data.


- **From Other Sources:** aparece un formulario con una gran variedad de fuentes, incluyendo bases de datos relacionales, bases de datos de Analysis Services, 'Data Feeds', ficheros de texto, etc. Aquí se incluyen todas las fuentes vistas anteriormente y otras adicionales. No vamos a citarlas aquí, sino que vamos a poner una imagen con el formulario que aparece, donde ya están descritas:


Bases de datos relacionales


**Microsoft SQL Server**
Cree una conexión a una base de datos de SQL Server. Importe tablas o vistas desde la base de datos o datos que haya devuelto una consulta.


**Microsoft SQL Azure**
Cree una conexión a la base de datos SQL Azure. Importe tablas o vistas desde la base de datos o datos que haya devuelto una consulta.


**Almacenamiento de datos paralelos de Microsoft SQL Server**
Cree una conexión a un almacenamiento de datos paralelos de SQL Server. Importe las tablas o vistas de la base de datos o bien, los datos que haya devuelto una consulta.


**Microsoft Access**
Cree una conexión a la base de datos de Microsoft Access. Importe tablas o vistas desde la base de datos o datos que hayan devuelto una consulta.


**Oracle**
Cree una conexión a una base de datos de Oracle. Importe tablas o vistas de la base de datos o datos que haya devuelto una consulta.

**Teradata**
Cree una conexión a una base de datos de Teradata. Importe tablas o vistas desde la base de datos o datos que haya devuelto una consulta.


**Sybase**
Cree una conexión a una base de datos de Sybase. Importe tablas o vistas desde una base de datos o datos que haya devuelto una consulta.

**Informix**
Cree una conexión a una base de datos de Informix. Importe tablas o vistas desde la base de datos o datos que haya devuelto una consulta.

**IBM DB2**
Cree una conexión a la base de datos DB2. Importe las tablas o vistas desde la base de datos o bien, los datos que haya devuelto una consulta.

**Otros (OLEDB/ODBC)**
Cree una conexión a un origen de datos mediante un proveedor OLE DB o una OLE DB para un proveedor ODBC. Importe datos de tablas o vistas que haya devuelto el proveedor.

Orígenes multidimensionales

**Microsoft Analysis Services**
Crear una conexión a un cubo de SQL Analysis Services. Importar los datos devueltos de una consulta MDX.

Fuentes de distribución de datos



Informe

Cree una conexión a un informe de Microsoft SQL Server Reporting Services. Importe datos de la fuente.



De Windows Azure Marketplace

Obtenga datos externos de Windows Azure Marketplace.



Sugerir datos relacionados

Obtenga sugerencias de datos externos.



Otras fuentes

Cree una conexión a una fuente de distribución de datos. Importe datos desde la fuente.

Archivos de texto



Archivo de Excel

Importe datos desde un archivo de Excel.



Archivo de texto

Importar datos de un archivo de texto.

- **Existing Connections:** muestra una relación de conexiones que ya hemos creado en nuestro equipo, para poder reutilizarlas sin tener que ir introduciendo de nuevo todos los datos necesarios para la conexión. Estos datos están almacenados en archivos con extensión “.odc”.

Iniciación a DAX

¿En qué punto del camino nos encontramos?

Llegado este momento y antes de abordar un tema, que es con seguridad el que mayor grado de complejidad tiene y a su vez el que mayor valor aporta a nuestros análisis de negocio, nos gustaría hacer un pequeño alto en el camino, salir de la evolución del aprendizaje que estamos haciendo en el día a día, alejarnos como si fuésemos un pájaro y tomar una vista general del camino recorrido y del que tenemos pendiente de recorrer. Para ello vamos a revisar el ciclo de desarrollo de una solución de BI Personal:

Comenzamos identificando un Proceso de Negocio y documentándolo. A partir de ahí diseñamos el Modelo Dimensional que nos permitirá responder a las preguntas de negocio. Como este modelo está vacío (aún no contiene información) debemos diseñar un proceso ETL (Extracción transformación y Carga) que incorpore la información a dicho modelo. A partir de este punto, y dependiendo de la herramienta que estemos utilizando (en nuestro caso Power Pivot), debemos hacer las mejoras que estimemos oportunas al modelo.

Todo lo citado anteriormente ya lo hemos estudiado y practicado con diferentes casos prácticos reales. Por tanto nos quedan dos fases por descubrir: *Enriquecer la información mediante el lenguaje de expresiones DAX* y *Realizar la capa de Visualización*. Todo el ciclo de desarrollo del BI Personal se muestra en la siguiente imagen:

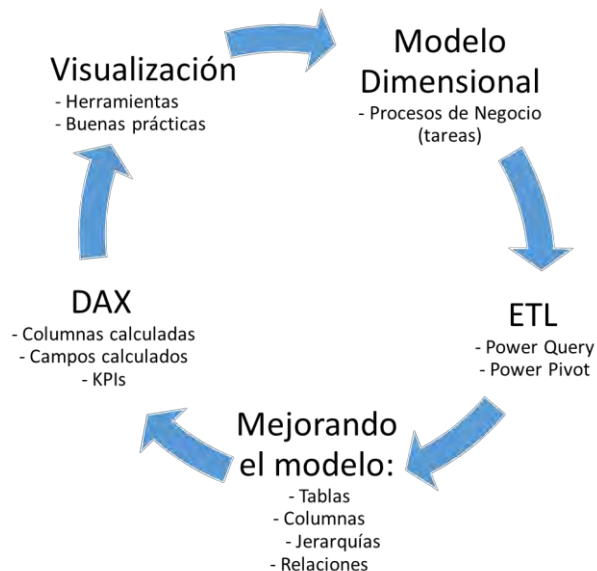


Figura 0-1 BI Personal. Ciclo de desarrollo.

Una vez puestos en contexto vamos a centrar el foco en Power Pivot, e igualmente veremos el camino recorrido y que nos queda por recorrer.

Con lo visto hasta el momento, hemos realizado el modelo dimensional, que es la base de toda nuestra solución, hemos creado el proceso ETL (bien con Power Query, bien con Power Pivot) que nos ha permitido cargar la información en nuestro almacenamiento en Power Pivot, hemos sido capaces de mejorar el modelo (tipos de datos, relaciones, jerarquías, etc.) y llega el momento de enriquecer dicho modelo mediante DAX.

Ha llegado el momento de subir el resto de peldaños que hemos marcado en nuestro proceso de aprendizaje:

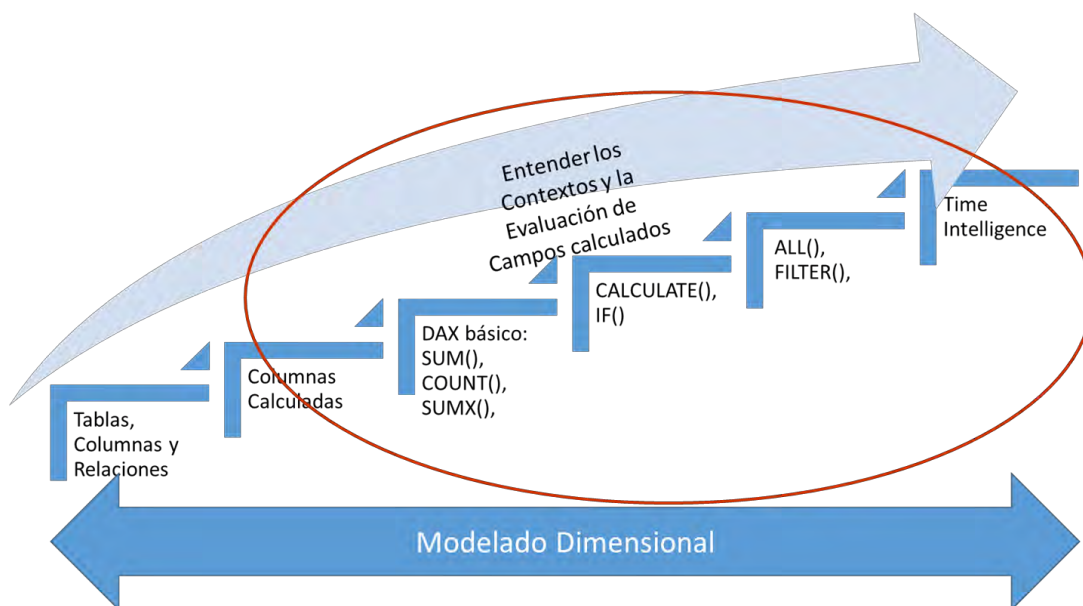


Figura 0-2 Evolución y proceso de aprendizaje de Power Pivot y DAX.

¿Qué es DAX?

DAX (Data Analysis eXpressions) es un lenguaje de expresiones que podemos encontrar en el interior de Power Pivot, está basado en fórmulas que permiten definir cálculos personalizados. Está compuesto por una biblioteca de funciones y operadores que se pueden combinar para generar fórmulas y expresiones. Se caracteriza por su simplicidad en la sintaxis y su similitud a las fórmulas de Excel, incluso ha heredado muchas funciones con el mismo nombre y misma funcionalidad (excepto pequeñas variaciones en contadas excepciones).

Sintaxis DAX

Las fórmulas de Excel están orientadas a cálculos *basados en celdas y rangos de éstas*, por ejemplo, en la celda A21 podemos escribir la siguiente expresión:
=SUMA(C1:G1)

✕		✓		fx		=SUMA(C1:G1)	
	C	D	E	F	G	H	
	1	2	3	4	5	15	

Figura 0-3 Sintaxis Excel tradicional (basada en celdas).

También tenemos una nueva sintaxis, incorporada en las últimas versiones de Excel, que se aplica en el caso de que hayamos formateado un conjunto de celdas como tabla, que nos permite hacer referencia a columnas de la tabla usando el formato "[@NombreColumna]". Tenga en cuenta que en las tablas se puede usar tanto la sintaxis tradicional basada en celdas, como la nueva sintaxis basada en columnas.




						=[@Cantidad]*[@Precio]	
C		D		E		F	
Fecha		Cantidad		Precio		Importe	
01/01/2014		5		8		40	
02/01/2014		23		8		184	
03/01/2014		6		8		48	
04/01/2014		3		8		24	
05/01/2014		8		8		64	

Figura 0-4 Sintaxis Excel para formato de tablas.

¿Por qué hablamos aquí de estas dos sintaxis? Pues para facilitar la introducción a la sintaxis de DAX, que es más similar a la sintaxis que acabamos de ver, basada en tablas, aunque con algunas peculiaridades:

- Sólo se permite hacer referencia a columnas de las tablas, *no existe el concepto de celda*.
- Se puede aplicar tanto en columnas calculadas, como a campos calculados (medidas) y KPIs.
- En el caso de aplicarse a campos calculados (medidas), la sintaxis incluye el nombre del nuevo campo.
- Los KPIs no permiten utilizar columnas de la tabla ni columnas calculadas, sólo permiten el uso campos calculados (medidas).

Campos calculados es el nuevo nombre que se da a partir de la versión 2012 a lo que antes se llamaba **Medidas**. Es simplemente un cambio de nombre; el concepto, funcionalidad y características siguen siendo las mismas. Aquí utilizaremos el nuevo nombre y entre paréntesis el antiguo, ya que debe conocer ambos y encontrará mucha documentación en internet que utilice el término "Medidas" (*Measures*).

En DAX: Campos Calculados = Medidas

¿Dónde se escriben las fórmulas DAX? Dentro de la ventana de Power Pivot, aunque se ubica en un lugar diferente y con ciertos matices en la sintaxis, dependiendo de que se trate de una columna calculada o un campo calculado (medida).

Sintaxis para columnas calculadas:

Es similar a la de los conjuntos de celdas formateados como tablas:

```
= "Expresión"  
= FUNCION( "Expresión1" ; "Expresión2" ; ... )
```

Para hacer referencia en cualquier expresión a las columnas de una tabla utilizaremos el formato 'Tabla'[Columna], y cumpliremos las siguientes reglas:

- Las comillas simples “'” son opcionales al indicar la tabla, sólo son obligatorias si el nombre de tabla contiene espacios o caracteres especiales:
 - o ‘Tabla Clientes’, Clientes
- Si la columna de la tabla utilizada en la expresión de columna calculada pertenece a esa misma tabla es opcional poner el nombre de la tabla:
 - o 'Tabla Ventas'[ImporteVenta],
Ventas[ImporteVenta], [ImporteVenta]
- Los corchetes son obligatorios. Es la forma que identifica que nos estamos refiriendo a una columna de la tabla.
 - o 'Tabla Ventas'[ImporteVenta],
Ventas[ImporteVenta], [ImporteVenta]

La introducción de las expresiones de las columnas calculadas se hace en un cuadro de texto que está justo en la parte superior de la tabla, muy similar a dónde se introducen las fórmulas en Excel. Una vez introducida la expresión, para cambiar el nombre de la columna calculada pulsamos botón derecho sobre ella y elegimos la opción “Cambiar nombre de la columna” (esta es la forma de hacerlo, ya que el nombre de la columna calculada no forma parte de la expresión DAX).

Cuando estamos escribiendo una expresión para una columna calculada, siempre nos aparece una ayuda contextual que nos facilita la tarea:

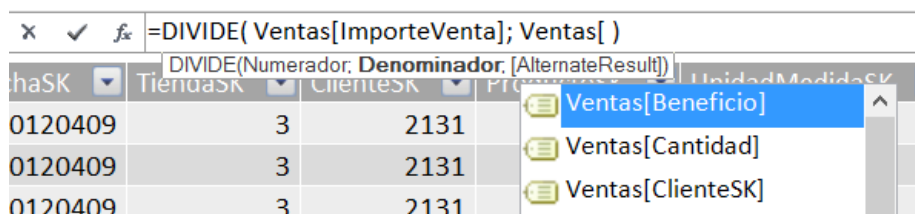


Figura 0-5 Ayuda a la escritura de DAX.

Veamos algunos ejemplos de uso de DAX en columnas calculadas:

<i>f_x</i>	=Ventas[ImporteVenta] / Ventas[Cantidad]		
	PrecioNeto	CalculatedColumn1	
01...	0,74		
01...	0,74		

<i>f_x</i>	=DIVIDE(Ventas[ImporteVenta]; Ventas[Cantidad])		
	PrecioNeto2	CalculatedColumn 2	
74	0,74	1	
74	0,74	1	
74	0,74	1	

Figura 0-6 Expresiones DAX en Columnas Calculadas.

Sintaxis para campos calculados (medidas):

Campo Calculado := "Expresión"

Campo Calculado := FUNCION("Expresión1" ;
"Expresión2" ; ...)

Se utilizan las mismas reglas indicadas anteriormente para las columnas calculadas, salvo que en este caso el nombre del campo calculado (medida) sí que forma parte de la expresión DAX y se indica en la parte izquierda de los símbolos " := ".

PowerPivot para Excel - Tema04_DAX_ParaSacarlmaç

File Inicio Diseñar Avanzadas

Pegar y anexar Pegar y reemplazar Copiar

De base de datos De servicio de datos De otros orígenes Conexiones existentes Actualizar Tabla dinámica

Tipo de datos: Formato: General \$ % +00 -00

Ordenar de A a Z Ordenar de Z a A Borrar orden Borrar todos los filtros Ordenar por columna Buscar

Portapapeles Obtener datos externos

[Producto...] f_x PromedioVentas:=AVERAGE([ImporteVenta])

linea	FechaSK	TiendaSK	CienteSK	ProductoSK	UnidadMedidaSK	Cantidad	ImporteVenta	ImporteCoste
300337	20120409	3	2131	985	1	1	0,74	0,44
300395	20120409	3	2131	985	1	1	0,74	0,44
301104	20120409	3	2131	985	1	1	0,74	0,44
301135	20120409	3	2131	985	1	1	0,74	0,44
301151	20120409	3	2131	985	1	1	0,74	0,44
301152	20120409	3	2131	985	1	1	0,74	0,44
301192	20120409	3	2131	985	1	1	0,74	0,44
							TotalVentas: 10...	
							PromedioVen...	TotalCoste: 811...

Figura 0-7 Expresiones DAX en Campos Calculados (Medidas).

El proceso de introducción de campos calculados consiste en posicionarse en cualquier celda libre del área de campos calculados (la rejilla que hay en la parte inferior de la tabla), pulsar sobre el área de expresiones DAX o la tecla “F2”, escribir la expresión DAX y pulsar la tecla “Intro”.

Una vez creada una columna calculada o un campo calculado (medida) queda disponible para ser utilizado en cualquier expresión DAX que creamos. De hecho, *nuestra recomendación es que siempre que le sea posible los reutilice y evite volver a escribir esa parte de la expresión*, tal y como se ve en el siguiente ejemplo:

f_x =DIVIDE([PrecioNeto]; [Cantidad])

f_x NuevoCalculo:=DIVIDE([PromedioVentas] ; [TotalVentas])

Las funciones DAX no tienen traducción al español. Independientemente de que esté utilizando Excel en inglés, en español o en otro idioma, sólo serán válidos los nombres originales de la función en inglés. Por ejemplo, siempre escribiremos SUM(... AVERAGE(..., etc. Las traducciones como SUMA(..., PROMEDIO(..., etc. no serán válidas.

Columnas Calculadas

Una **Columna Calculada** (*Calculated Column*) es una columna que se agrega a una tabla de Power Pivot existente, mediante una expresión DAX que la define. El valor de la expresión utilizada se calcula para cada fila de la tabla en el mismo momento que se crea (en tiempo de diseño del modelo) y posteriormente se va actualizando en el momento en que se actualizan los datos del origen (al pulsar el botón “Actualizar” (*Refresh*)).

Puede crear columnas calculadas basadas en otros campos calculados (medidas) y otras columnas calculadas.

Las columnas calculadas se van agregando a la derecha de las columnas que hemos extraído del origen de datos y automáticamente les asigna los nombres CalculatedColumn1, CalculatedColumn2 y así sucesivamente. Aunque puede reorganizar el orden de las columnas pulsando sobre ellas y arrastrándolas a la posición deseada. También puede renombrar cualquier columna calculada, para ello pulse botón derecho sobre ella y elija la opción “Cambiar nombre de la columna”.

Sus valores se almacenan en disco y en memoria de la misma forma que cualquier otra columna de la tabla. Técnicamente se conocen como “materializadas” o “persistentes”. Están designadas para ser estáticas por naturaleza.

Las columnas calculadas se pueden utilizar posteriormente en Excel en:

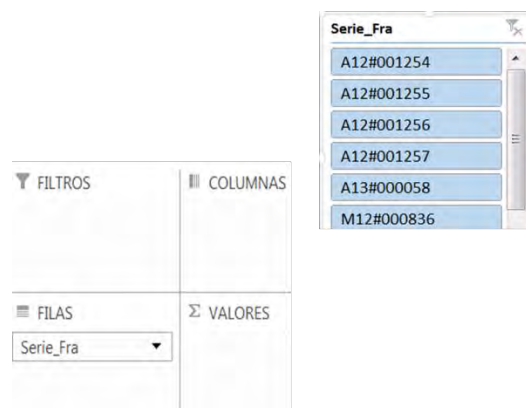
Segmentadores (*Slicers*)

Filtros

Filas

Columnas

Valores



Veamos algunos ejemplos de uso:

- Calcular el año (o cualquier otro cálculo relativo a la fecha) en una nueva columna =YEAR([Fecha])
- Concatenar el número de serie y factura =[Serie]&"#[Factura]
- Calcular el Margen =[PrecioNeto] - [PrecioCoste]
- Obtener una columna de otra tabla =RELATED(Provincia[Provincia])

- Es una técnica más para desnormalizar

Cuando arrastramos una columna al área de valores ocurre algo internamente que debemos entender, realmente lo que hace Power Pivot es crear un “Campo Calculado Implícito”, esto ocurre siempre que arrastremos una columna de una tabla, bien sea calculada o no, al área de valores.

Un **Campo Calculado Implícito** es aquel que no existe originariamente, pero que se crea automáticamente en el momento de arrastrar una columna al área de valores. A partir de ese momento existe como un campo más al que por defecto se le asigna la función de agregación SUM y su nombre por defecto es “sum of Nombre_Columna_Calculada”. Por ejemplo, si tenemos la columna ImporteVenta y la arrastramos por primera vez al área de valores se creará automáticamente el campo calculado “Sum of ImporteVenta” (o si utiliza la versión en español “Suma de ImporteVenta”).

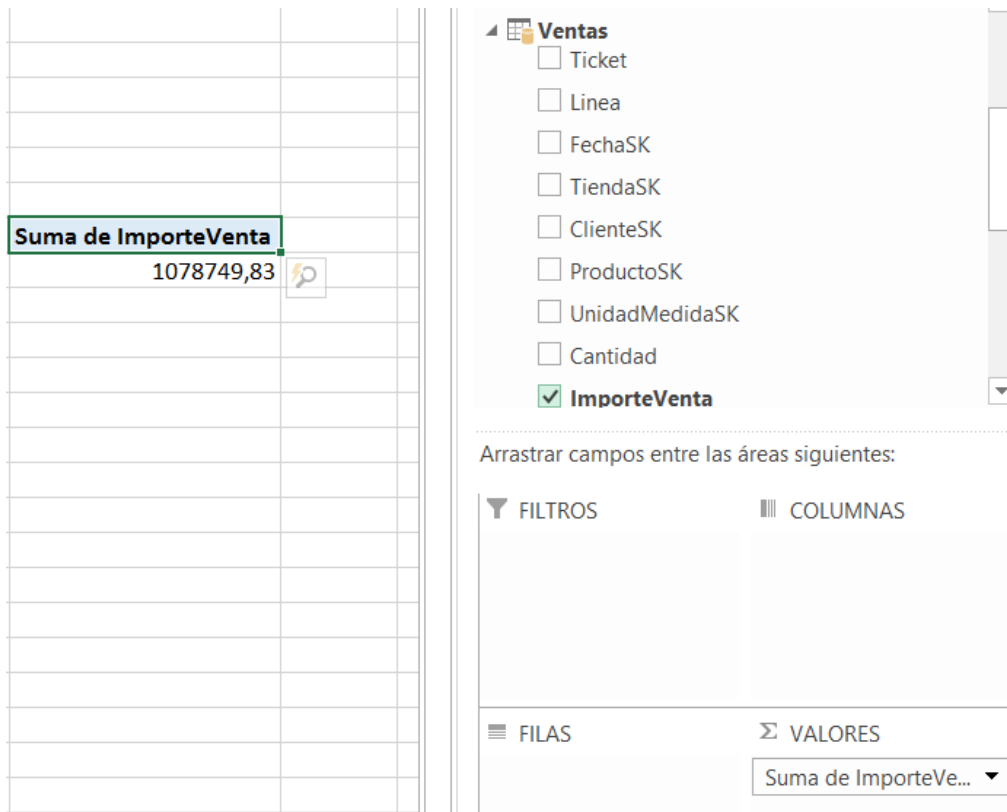


Figura 0-8 Campos Calculados Implícitos (Medidas Implícitas).

Nuestra recomendación es que **no utilice campos calculados implícitos**, básicamente, porque no permiten decidir ni el nombre ni el formato que tendrán. Para evitar su uso debe usar la opción 'Ocultar en herramientas cliente' para estas columnas, con ello evita que estén visibles y por tanto que puedan ser utilizadas en las tablas y gráficos dinámicos. Y a continuación crear un campo calculado explícito, aunque sea una simple suma, ya que sobre él sí que podrá decidir el nombre que tendrá y asignarle el formato que desee para obtener una mejor visualización.

Un **Campo Calculado Explícito** es aquel que creamos nosotros en tiempo de diseño en Power Pivot, en la rejilla diseñada para ello. A continuación estudiaremos con detalle los campos calculados (medidas).

Llegados a este punto nos planteamos la siguiente pregunta que tiene varias respuestas, con sus pros y sus contras: ¿Dónde debo realizar los cálculos, con SQL al pedir la información desde el origen, con Power Query, con columnas calculadas?

- Con SQL tenemos dos inconvenientes, necesitamos saber lenguaje SQL (es evidente) y además la información se calcula en origen consumiendo recursos del servidor y viajando por la red una vez calculada. No os la recomendamos en este caso.
- Con Power Query, se calcula durante la ejecución del proceso de actualización de los datos, se hace ya en el destino, en el equipo donde se está ejecutando y la columna queda almacenada como cualquier otra columna del origen (realmente no somos capaces de distinguir cuando estamos viendo la información columnar en Power Pivot si una columna existe en el origen o se calcula con Power Query. Esta es una buena alternativa, dejando toda esa lógica de cálculo en Power Query, tiene la ventaja de que si necesito ciertas columnas sólo para el cálculo las puedo utilizar en el proceso y no tienen por qué llegar al modelo en Power Pivot.
- Utilizando columnas calculadas, los cálculos DAX son bastante eficientes, pero por el contra me obliga a llevar las columnas que intervienen en el cálculo al modelo y por tanto ocupar un espacio innecesario. Lo que sí podemos hacer, es ocultarlas al usuario para que no las vea.

Campos Calculados (Medidas)

Los **Campos Calculados** (*Calculated Fields*), también conocidos como Medidas (*Measures*) en versiones anteriores, son los cálculos que se usan para el análisis de información.

Los campos calculados (medidas) sólo se pueden utilizar posteriormente en Excel en:



Se crean al escribir una expresión DAX tras seleccionar una celda en el área de cálculo (este proceso se ha visto con detalle anteriormente).

Sus valores no se almacenan en disco (como ocurre con las columnas calculadas), sino que el motor los calcula cuando recibe las consultas. Gracias a ellos vamos a tener una potencia de cálculo casi infinita. No, no estoy exagerando. Para que desde el inicio sea consciente de la importancia y envergadura que tiene esta incorporación, voy a citar a uno de los mayores gurús en la materia, **Bill Jelen (Mr. Excel)**:

“Los Campos Calculados (Medidas) es Lo Mejor que la sucedido a Excel en los últimos 20 años”

Aparentemente nos pueden parecer unas fórmulas que se agregan y permiten ser utilizadas en una tabla o gráfico dinámico, y así es, pero lo realmente relevante es que ofrecen una potencia y flexibilidad de cálculo sin precedentes.

Si ya ha utilizado las tablas dinámicas y conoce las fórmulas que se pueden crear en ellas, el nuevo mensaje es: “va a dejar de utilizarlas”, ya no es necesario, dispone de DAX que hace todo lo que esas fórmulas y muchas cosas más. Si no ha utilizado dicha característica no hay problema, no es muy conocida y a partir de ahora tampoco la va a necesitar.

Uso de columnas calculadas vs campos calculados (medidas)

Consideramos que es una cuestión fundamental saber cuándo debe escribir una columna calculada y cuando un campo calculado (medida). Aunque no hay una respuesta única sí que vamos a dar una serie de pautas que le ayuden en su decisión. Mostraremos qué utilizar en los siguientes casos:

1) Utilizar Columnas Calculadas cuando sólo intervienen en el cálculo valores de la propia fila donde se va a representar.

Si por cada fila de la tabla tenemos la cantidad de unidades vendidas y el importe total de la línea, pero no el precio neto unitario y queremos calcularlo:

fx =ROUND(Ventas[ImporteVenta] / Ventas[Cantidad]; 6)			
Cantidad	ImporteVenta	PrecioNeto	
7	5,55	0,792857	
15	12,99	0,866	
15	13,33	0,888667	
7	1,9	0,271429	
15	13,33	0,888667	
7	1,65	0,235714	
15	14,05	0,936667	
7	6,55	0,935714	

Figura 0-9 Uso de columnas calculadas.

Otro ejemplo habitual es cuando tenemos una fecha y queremos extraer de ella nuevas columnas con el año, el mes, el trimestre, el día de la semana, etc.

2) Utilizar Columnas Calculadas si necesita utilizarlas en segmentadores (slicers), área de filtro, filas o columnas.

Como hemos visto anteriormente los campos calculados (medidas) sólo se pueden utilizar en el área de valores (no hay forma de colocarlos en otro área), mientras que las columnas calculadas se pueden utilizar además de en el área de valores, en segmentadores (slicers), área de filtro, filas o columnas. No siempre tenemos esta necesidad, además las columnas calculadas tienen menor flexibilidad que los campos calculados, por lo tanto, piense detenidamente a la hora de tomar la decisión si realmente necesita usarla más allá del área de valores.

3) Utilizar Campos Calculados (Medidas) en el resto de casos.

Los campos calculados (medidas) son especialmente útiles y flexibles para la realización de cálculos que afectan a conjuntos de filas. Por ejemplo, conocer los clientes activos (aquellos que les hemos vendido en un periodo de tiempo)

			Formato		
	f_x	Clientes Activos:=DISTINCTCOUNT(Ventas[ClienteSK])			
	ClienteSK		Pr...		UnidadMedi...
	2131		985		1
	2131		985		1
	Clientes Activos				
	f_x	% Importe Cliente:=DIVIDE([TotalVentas]; CALCULATE([TotalVentas]; ALLEXCEPT(Ventas; Ventas[Cantidad])))			
		ClienteSK		ProductoSK	UnidadMedi...
	4	2131		985	
	4	2131		985	
		Clientes Activos		Productos Activos	
				RankingProductosVentas	
		% Importe Cliente		RankingProductosBfos	

Figura 0-10 Uso de campos calculados (medidas).

En la imagen anterior ha visto una serie de expresiones DAX que no entiende, es lógico, no las hemos explicado aún. Por ello el siguiente paso es ir conociendo algunos conceptos fundamentales y las funciones que componen el lenguaje DAX, practicar con ellas e ir mostrando ejemplos de uso.

Funciones DAX

Vamos a comenzar con el estudio de algunas de las funciones DAX básicas y a su vez más habituales, para posteriormente ir profundizando en detalles y conceptos adicionales que hemos de tener claros para obtener el mejor resultado analítico.

Funciones heredadas de Excel

Vamos a comenzar por las **funciones heredadas de Excel**. Si ve el nombre de una función que ya conoce de Excel, la puede usar directamente, ya que en Power Pivot tendrá el mismo comportamiento y los mismos parámetros que tenga en las hojas tradicionales. Sólo habrá muy contadas excepciones que veremos llegado el momento de estudiar a fondo cada función. Otra diferencia, ya comentada anteriormente, es que no existirá su traducción sino sólo la función con su nombre en inglés.

Exponemos en la siguiente imagen una relación de estas funciones:

Date and Time	Information	Math and Trig	Math and Trig	Text
DATE	ISBLANK	ABS	ROUND	CONCATENATE
DATEVALUE	ISERROR	CEILING	ROUNDDOWN	EXACT
DAY	ISLOGICAL	EXP	ROUNDUP	FIND
EDATE	ISNONTEXT	FACT	SIGN	FIXED
EOMONTH	ISNUMBER	FLOOR	SQRT	LEFT
HOUR	ISTEXT	INT	SUM	LEN
MINUTE		LN	SUMSQ	LOWER
MONTH	Logical	LOG	TRUNC	MID
NOW	AND	LOG10		REPLACE
SECOND	IF	MOD	Statistical	REPT
TIME	IFERROR	MROUND	AVERAGE	RIGHT
TIMEVALUE	NOT	PI	AVERAGEA	SEARCH
TODAY	OR	POWER	COUNT	SUBSTITUTE
WEEKDAY	FALSE	QUOTIENT	COUNTA	TRIM
WEEKNUM	TRUE	RAND	COUNTBLANK	UPPER
YEAR			MAX	VALUE
YEARFRAC			MAXA	
			MIN	
			MINA	

Figura 0-11 Funciones heredadas de Excel.

Funciones de Agregación

Hay otro grupo de funciones, las **Funciones de Agregación**, que parte de ellas provienen de la relación de funciones heredadas de Excel y nos van a facilitar el comenzar a entender las diferencias entre el Excel tradicional y Power Pivot.

Las agregaciones permiten contraer, resumir o agrupar datos. Se encargan de realizar un cálculo sobre una columna a partir del máximo nivel de detalle de sus filas, afectando en principio a todas ellas, salvo que indiquemos algún tipo de filtro o desglose descriptivo.

Veamos un ejemplo, si queremos saber el total de ventas de la empresa, crearemos una columna calculada utilizando la función SUM():

fx TotalVentas:=SUM(Ventas[ImporteVenta])	
Cantidad	ImporteVenta
1	0,74
1	0,74
1	0,74
1	0,74
1	0,74
TotalVentas: 1.078.749,83	

Figura 0-12 Ejemplo de uso de la función SUM().

Hasta aquí nada nuevo, esta función ya la hemos visto en ejemplos anteriores y nos ofrece la suma de todos los valores de la columna “[ImporteVenta]”.

Veamos ahora lo que hemos conseguido escribiendo esa función tan simple desde el punto de vista de la visualización y uso en una tabla dinámica. Tenga en cuenta que también hemos aplicado formato para que muestre siempre dos decimales y los separadores de miles (el cómo hacer esto se estudió anteriormente).

Si creamos una tabla dinámica conectada a nuestro modelo en Power Pivot y arrastramos “TotalVentas” al área de valores, nos muestra el resultado de sumar “[ImporteVenta]” para todas las filas de la tabla y ya formateado:

		<input checked="" type="checkbox"/> TotalVentas
		Arrastrar campos entre las áreas siguientes:
		<div> <div>FILTROS</div> <div>COLUMNAS</div> </div>
		<div> <div>FILAS</div> <div> Σ VALORES <div>TotalVentas</div> </div> </div>

TotalVentas
1.078.749,83

Figura 0-13 TotalVentas.

Ahora lo desglosamos con una fila por cada Tienda:

Etiquetas de fila	TotalVentas
Tienda Alicante 01	408.818,05
Tienda Alicante 02	54.236,44
Tienda Castelldefels	320.580,88
Tienda La Coruña	103.264,14
Tienda Murcia 01	48.624,71
Tienda Principal	143.225,61
Total general	1.078.749,83

		Arrastrar campos entre las áreas siguientes:
		<div> <div>FILTROS</div> <div>COLUMNAS</div> </div>
		<div> <div>FILAS</div> <div> Σ VALORES <div>Tienda</div> </div> </div>
		<div> <div>FILAS</div> <div> Σ VALORES <div>TotalVentas</div> </div> </div>

Figura 0-14 TotalVentas por Tienda.

Y lo volvemos a desglosar con una columna por cada año:

TotalVentas	Etiquetas de columna		
Etiquetas de fila	2012	2013	Total general
Tienda Alicante 01	322.513,71	86.304,34	408.818,05
Tienda Alicante 02	42.516,38	11.720,06	54.236,44
Tienda Castelldefels	244.255,51	76.325,37	320.580,88
Tienda La Coruña	75.213,03	28.051,11	103.264,14
Tienda Murcia 01	36.336,70	12.288,01	48.624,71
Tienda Principal	113.390,62	29.834,99	143.225,61
Total general	834.225,95	244.523,88	1.078.749,83

Arrastrar campos entre las áreas siguientes:

FILTROS

COLUMNAS

Año

FILAS

VALORES

Tienda

TotalVentas

Figura 0-15 TotalVentas por Tienda y Año.

Y finalmente agregamos un filtro para tener sólo las ventas de los clientes “solteros” (EstadoCivil=Soltero/a):

EstadoCivil	Soltero/a
-------------	-----------

TotalVentas	Etiquetas de columna		
Etiquetas de fila	2012	2013	Total general
Tienda Alicante 01	4.491,76	824,18	5.315,94
Tienda Alicante 02	1.301,18	219,06	1.520,24
Tienda Alicante 02 (Tienda)	3.826,05	804,32	4.630,37
Tienda Alicante 02 (Tienda)	1.778,53	607,59	2.386,12
Tienda Murcia 01	4.968,99	1.750,51	6.719,50
Tienda Principal	5.129,81	1.166,93	6.296,74
Total general	21.496,32	5.372,59	26.868,91

Fechalni

FechaFin

Arrastrar campos entre las áreas siguientes:

FILTROS

COLUMNAS

EstadoCivil

Año

FILAS

VALORES

Tienda

TotalVentas

Figura 0-16 TotalVentas a clientes Solteros, por Tienda y Año.

¿Cuántas fórmulas DAX ha tenido que realizar para obtener toda esta información relativa al importe de ventas de la empresa?

¡¡¡ SÓLO UNA !!! Y manteniendo el mismo valor en el parámetro

Pero, si hubiese utilizado la función SUM (o SUMA) en el Excel tradicional ¿Cuántas veces hubiese tenido que cambiar los rangos de celdas del parámetro que le pasa a dicha función para obtener ese resultado? No merece la pena ni contarlos, es evidente que uno por cada cambio.

Ahí es donde reside uno de los elementos que dan esa potencia de cálculo y flexibilidad para realizar análisis. Pero no es el único elemento, más adelante seguiremos profundizando y entendiendo el porqué de este comportamiento tras el hecho de crear un campo calculado (medida) tan simple en DAX.

A continuación vamos a explicar los distintos **tipos de agregaciones** que existen y las funciones que hay en cada uno de ellos:

- **Aditivas:** Agregan sin restricciones.
 - Funciones: **SUM, COUNT, COUNTROWS, AVERAGE**
- **Semi-Aditivas:** Agregan sólo en parte, veamos un ejemplo: necesitamos saber el total de clientes distintos que compran en nuestras tiendas, sabemos que en enero han sido 200 clientes, en febrero han sido 225 clientes y en marzo 210; en cambio si lo que queremos saber el total de clientes distintos que compran en nuestras tiendas en el primer trimestre, no podemos sumar tal cual, porque habrá clientes que nos hayan comprado durante los tres meses de este trimestre y otros que no, por tanto, será un número comprendido entre 225 y 635.
 - Funciones: **DISTINCTCOUNT**
- **No Aditivas:** No pueden agregar.
 - Funciones: **MIN, MAX**
- **Personalizadas:** decidimos nosotros mediante diversas expresiones DAX cómo deben actuar. Por ejemplo, si queremos calcular un balance de situación, habrá cuentas que sumen y cuentas que resten.
 - Funciones: no hay ninguna específica, tendremos que crear expresiones DAX más complejas que resuelvan de forma individual cada necesidad.

En la siguiente tabla mostramos la sintaxis de estas funciones y una breve descripción. Para conocer más detalles puede ver la ayuda del producto.

SUM(<Columna>)	Suma todas las filas de una columna. = SUM([ImporteVenta])
COUNT(<Columna>)	Cuenta el número de filas de una columna que tenga datos numéricos o fechas. = COUNT([ImporteVenta])
COUNTROWS(<Tabla>)	Cuenta el número de filas de una tabla. = COUNTROWS('Cliente')

	<code>= COUNTROWS (RELATEDTABLE ('Ventas'))</code>
AVERAGE(<Columna>)	Devuelve el promedio de todos los números de una columna. <code>= AVERAGE ([ImporteVenta])</code>
DISTINCTCOUNT(<Columna>)	Cuenta el número de filas con valores distintos de una columna. <code>= DISTINCTCOUNT (Ventas[Ticket])</code>
MIN(<Columna>)	Devuelve el menor valor numérico de una columna que tenga datos numéricos o fechas. <code>= MIN (Ventas[TiendaSK])</code>
MAX(<Columna>)	Devuelve el mayor valor numérico de una columna que tenga datos numéricos o fechas. <code>= MAX (Ventas[TiendaSK])</code>

Funciones de navegación entre tablas a través de las relaciones

Estas funciones nos permiten navegar y obtener valores de columnas de diversas tablas del modelo siempre que existan relaciones entre ellas.

En la siguiente tabla mostramos la sintaxis de estas funciones y una breve descripción. Para conocer más detalles puede ver la ayuda del producto.

RELATED(<Columna>)	Devuelve un valor relacionado de la columna de otra tabla siguiendo la relación M->1. Ejemplo: agregar una columna calculada con la “[Poblacion]” a la tabla “Tienda” (Muchas Tiendas -> 1 Población): <code>= RELATED (Geografia[Poblacion])</code>
RELATEDTABLE(<Tabla>)	Devuelve una tabla en un contexto especificado por los filtros indicados, siguiendo la relación 1->M. Ejemplo: agregar una columna con el número de clientes a la tabla “Geografia”: <code>= COUNTROWS (RELATEDTABLE ('Cliente'))</code>


La función RELATED se puede utilizar, por ejemplo, para desnormalizar, creando columnas calculadas que permitan añadir columnas de varias tablas del origen, que estén relacionadas, a una sola tabla.

Sobre la función RELATEDTABLE volveremos más adelante, ya que se utiliza en conjunto con otras funciones y hay que entender qué son los contextos y cómo actúan.

DAX en la práctica

Entendiendo los contextos

Anteriormente hemos visto que con una expresión DAX muy simple, como el campo calculado (medida) siguiente:

	<code>TotalVentas:=SUM(Ventas[ImporteVenta])</code>
---	---

Nos ha permitido analizar en una tabla dinámica, las ventas totales, las ventas por tienda y año, las ventas a clientes solteros por tienda y año, ..., y así habríamos podido seguir mostrando decenas, de informes basados en dicho campo calculado.

Pero también se encontrará a veces en el caso de que la expresión DAX que realice no le devuelva el valor que espera, buena parte de la culpa la tienen **los contextos**. Por ello, los vamos a estudiar a continuación, ya que es imprescindible conocerlos y saber cómo se comportan para entender cualquier función DAX.

Los contextos permiten realizar análisis dinámicos, en los que los resultados de una expresión van variando en función de las selecciones realizadas en las áreas de filas, columnas y filtros. Debemos conocerlos y usarlos de forma eficaz, tanto para evitar obtener resultados erróneos, como para generar expresiones eficientes.

Tenemos tres tipos contextos:

- Contexto de Fila (*Row Context*)
- Contexto de Consulta (*Query Context*)
- Contexto de Filtro (*Filter Context*)

Contexto de Fila (Row Context)

Aplica a los elementos de la fila en cuestión, a la “fila actual”. Se utiliza, entre otros, en las columnas calculadas. Abarca la fila que se está calculando y las filas de otras tablas relacionadas con dicha fila. Veamos un ejemplo:

fx =ROUND(Ventas[ImporteVenta] / Ventas[Cantidad]; 4)			
Cantidad	ImporteVenta	PrecioNeto	
-32	-12,37	0,3866	
15	16,8	1,12	
-23	-18,11	0,7874	
18	21,25	1,1806	
13	12,64	0,9723	
13	9,56	0,7354	
13	3,07	0,2362	
15	12,99	0,866	

Figura 0-1 Contexto de Fila, columna calculada [PrecioNeto].

Se comporta como las fórmulas de Excel aplicadas en el interior de una tabla, haciendo referencia automáticamente a valores de la misma fila. Si la tabla está relacionada con otras, se tiene acceso a cualquier valor de dichas tablas que estén relacionados con la fila actual. Para acceder a dichas tablas relacionadas, disponemos de la función RELATED explicada anteriormente.

Hay otros tipos de casuísticas para las que DAX dispone de una serie de funciones que iteran, en una especie de bucle interno, los cálculos sobre una tabla. De momento es suficiente con entender bien las expuestas y asentar una buena base.

Contexto de Consulta (Query Context)

Evalúa un conjunto de datos para una celda concreta. Hace referencia a un subconjunto de datos que se recuperan implícitamente para una fórmula y permiten agregar elementos a las áreas de filtros, filas y columnas.

Al colocar un campo calculado o una columna con valores numéricos en el área de valores de una tabla dinámica, el motor de Power Pivot examina los encabezados de columna y fila, segmentos de datos y los filtros de informe para determinar el contexto. A continuación, Power Pivot realiza los cálculos necesarios para rellenar cada celda en la tabla dinámica. El conjunto de datos que se recupera es el contexto de la consulta para cada celda.

Para un campo calculado (medida), es el definido por la fila y columna, más los filtros aplicados desde segmentadores (*slicers*) y áreas de filtro.

Veamos un ejemplo de ventas simplificado, con un número de filas y columnas muy reducido para entenderlo mejor. Para analizar las ventas hemos puesto en filas las fechas, en valores el campo calculado “Sum of importe” (campo calculado implícito) y hemos agregado dos segmentadores, uno para Cliente y otro para Producto. Finalmente hemos filtrado sólo las ventas del Cliente “C1”:

Fecha	Cliente	Producto	Cantidad	Precio	Importe
01/01/...	C1	P1	5	8	40
01/01/...	C1	P2	23	10	230
01/01/...	C2	P1	6	8	48
01/01/...	C2	P2	3	10	30
02/01/...	C1	P1	8	8	64
02/01/...	C2	P2	11	10	110
02/01/...	C1	P1	7	8	56
02/01/...	C2	P2	4	10	40
03/01/...	C1	P1	1	8	8
03/01/...	C1	P2	2	10	20

Cliente	Producto
C1	P1
C2	P2

Row Labels	TotalImporte
01/01/2014	270,00
02/01/2014	120,00
03/01/2014	28,00
Grand Total	418,00

Figura 0-2 Contexto de Consulta.

En la tabla de la izquierda (en color gris) se ven las celdas que lee el motor de Power Pivot para obtener el valor que aparece calculado en la tabla dinámica que hay en la parte derecha (color celeste y blanco).

Contexto de Filtro (*Filter Context*)

Se aplica sobre los contextos de fila y consulta, con el fin de generar excepciones y permitirnos variar las celdas evaluadas en base a nuestras necesidades. Es decir, permite reducir o ampliar los contextos de fila (*Row context*) y consulta (*Query context*) mediante filtros adicionales aplicados en las funciones DAX

Hay multitud de casos en los que debemos aplicar contextos de filtro adicionales, pero con el fin de entenderlo vamos a hacer un ejemplo sencillo en el que vamos a calcular el ratio de ventas de cada cliente con respecto al total de ventas de la empresa, para ello vamos a definir los siguientes campos calculados:

```
TotalImporte:=SUM([Importe])
```

```
TotalImporteEmpresa:=CALCULATE([TotalImporte];ALL(Ventas2))
```

```
RatioVentas:=DIVIDE([TotalImporte]; [TotalImporteEmpresa])
```

Row Labels	TotalImporte	TotalImporteEmpresa	RatioVentas
C1	394,00	656	60,06 %
C2	262,00	656	39,94 %
Grand Total	656,00	656	100,00 %

Figura 0-3 Contexto de Filtro.

Observe que “TotalImporteEmpresa” es siempre 656. Más adelante estudiaremos las funciones CALCULATE y ALL, así como otras funciones que nos permiten realizar contextos de filtro.

Como puede apreciar, hemos aplicado las buenas prácticas de crear campos calculados con cálculos simples e ir reutilizándolos. Es preferible hacerlo con los tres campos calculados mostrados anteriormente que con la siguiente expresión:

```
RatioVentas:=DIVIDE(SUM([Importe]);  
CALCULATE([TotalImporte];ALL(Ventas2)))
```

Tanto por la simplicidad a la hora de entenderlos, como por su posterior reutilización. Si lo hacemos como propusimos inicialmente, podremos reutilizar los campos calculados “TotalImporte” y “TotalImporteEmpresa” en cualquier otro cálculo que necesitemos realizar, y si hubiese cualquier error con sólo corregirlo ahí quedarían corregidos automáticamente todos los campos calculados que los incluyen.

Funciones DAX

Una vez sentadas las bases, estamos capacitados para ir aprendiendo cualquier función DAX e ir aplicándola correctamente. A continuación vamos a conocer algunas de uso frecuente, que además se suelen utilizar en combinación con otras de las funciones existentes.

Funciones de Agregación terminadas en “X”

Tenemos una serie de funciones de agregación terminadas en “X”, como SUMX, COUNTX, AVERAGEX, MINX, MAXX, etc. Estas funciones iteran fila a fila sobre la tabla que le pasamos como primer parámetro y hacen la operación indicada como segundo parámetro, para finalmente aplicar al resultado la función de agregación indicada.

Veamos algo que a priori nos parece muy sencillo, que además es una expresión que podemos utilizar en las columnas calculadas (recuerde, contexto de fila), pero que no está permitido utilizarla en los campos calculados (medidas).

En la siguiente tabla disponemos de “Cantidad” e “ImporteVenta”, pero no del “PrecioNeto”, esto no es ningún problema es una simple división a la que además vamos a aplicar un redondeo a cuatro dígitos:

fx =ROUND([ImporteVenta]/[Cantidad];4)			
Cantidad	ImporteVenta	PrecioNeto	
18	21,25	1,1806	
17	14,71	0,8653	
16	3,77	0,2356	
17	15,1	0,8882	

Figura 0-4 “PrecioNeto” como columna calculada.

Si más adelante necesitamos saber la suma total de “PrecioNeto” podemos crear el siguiente campo calculado:

```
TotalPrecioNeto:=SUM([PrecioNeto])
```

Ahora nos ponemos en la situación de que lo que necesitamos realizar ese cálculo en un campo calculado (medida), prescindir de la columna calculada y ahorrarnos ese espacio en disco y memoria, por tanto, vamos a la rejilla y lo creamos:

```
PrecioNetoX:=ROUND([ImporteVenta]/[Cantidad];4)
```

Pero nos encontramos con el siguiente error:




Semantic Error: The value for 'ImporteVenta' cannot be determined.

Bueno, realmente lo que queremos es un campo calculado que nos de la suma, por tanto incluimos la función SUM de la siguiente forma:

```
TotalPrecioNetoX:=SUM(ROUND([ImporteVenta]/[Cantidad];4))
```

Y de nuevo nos encontramos con un error, que ahora es diferente:



Semantic Error: The SUM function only accepts a column reference as an argument.

¿Realmente no podemos hacer un campo calculado que sume el resultado de una operación? Pues sí que podemos, pero no con la función SUM, para ello deberemos usar la función SUMX.

```
TotalPrecioNetoX:=SUMX('Ventas';  
ROUND([ImporteVenta]/[Cantidad];4))
```

Tenga mucho cuidado con la realización de cálculos incorrectos, que tendemos a realizar al iniciarnos en el lenguaje DAX. Veamos el siguiente campo calculado:

```
TotalPrecioNetoMal:=ROUND(SUM([ImporteVenta])/SUM([Cantidad]);  
4)
```

Este segundo caso, lo que hace es, primero sumar “ImporteVenta” para todas las filas, luego sumar “Cantidad” para todas las filas y finalmente dividir ambos resultados.

Veamos ahora el resultado de los tres cálculos sobre la tabla dinámica, el error de cálculo cometido en el último caso es abismal:

Valores	
TotalPrecioNeto	1.008.513,9789
TotalPrecioNetoX	1.008.513,9789
TotalPrecioNetoMal	1,6020

Figura 0-5 Errores de cálculo.

Este es un caso extremo, pero sí que debe tener en cuenta que conforme vamos haciendo campos calculados (medidas) más sofisticados, puede obtener totales generales y subtotales que no sean igual a la suma o el promedio de su detalle. En estos casos le serán de ayuda las funciones de agregación terminadas en “X”.

En la siguiente tabla mostramos la sintaxis de estas funciones y una breve descripción. Para conocer más detalles puede ver la ayuda del producto.

SUMX(<Tabla>; <Expresión>)	Devuelve la suma de una expresión evaluada por cada fila de una tabla.
COUNTX(<Tabla>; <Expresión>)	Devuelve el número de filas que cumplen la expresión evaluada.
AVERAGEX(<Tabla>; <Expresión>)	Calcula el promedio de una expresión evaluada por cada fila de la tabla
MINX(<Tabla>; <Expresión>)	Evalúa una expresión por cada fila de la tabla y devuelve el menor valor numérico.
MAXX(<Tabla>; <Expresión>)	Evalúa una expresión por cada fila de la tabla y devuelve el mayor valor numérico.

Funciones ALL() y ALLEXCEPT()

La función ALL se utiliza para eliminar filtros del contexto. Tiene dos sintáxis:

```
ALL(<Tabla>)
```

```
ALL(<Columna1>;<Columna1>;...;<Columna1>)
```

No puede combinar ambas opciones. Si se indica una tabla, quita todos los filtros de la tabla especificada. Si se indica una o varias columnas, quita los filtros a las columnas especificadas, manteniendo los filtros aplicados para el resto de las columnas de la tabla.

Esta función no se usa por sí sola, sino que se utiliza como parámetro de otras funciones para cambiar el conjunto de resultados sobre los que se realizarán otros cálculos.

La función ALLEXCEPT es la opuesta en cuanto a filtros del contexto a eliminar, es decir, mantiene los filtros del contexto aplicados a la tabla o columnas especificadas y elimina todos las demás. También tiene dos sintaxis:

```
ALLEXCEPT(<Tabla>)
```

```
ALLEXCEPT(<Columna1>;<Columna1>;...;<Columna1>)
```

Y sobre ella aplica todo lo explicado para la función ALL.

Veamos un ejemplo de cuando usar una u otra, ya que en muchos casos es más una cuestión de escribir menos. Si tenemos una tabla con 15 columnas y necesitamos aplicar ALL sobre 14 de ellas, es más práctico utilizar ALLEXCEPT sobre esa única columna sobre la que no quiero que aplique ALL. La diferencia estaría en el comportamiento que tendría si se incluyese una nueva columna en la tabla, en el primer caso ALL seguiría aplicando sobre las 14 columnas indicadas como parámetros y no aplicaría ahora sobre 2 de ellas, mientras que en el segundo caso ALLEXCEPT haría que aplicase sobre 15 columnas (se incluiría la nueva) y no aplicaría sobre la única columna que incluimos como parámetro.

Función FILTER()

La Función FILTER nos permite realizar un filtro sobre una tabla. Su sintaxis es:

```
FILTER(<Tabla>;<Filtro>)
```

<Tabla>: es la tabla que se va a filtrar. Tenga en cuenta que bien puede ser directamente una tabla o una expresión que devuelva una tabla como resultado.

<Filtro>: Es una expresión booleana, es decir, el resultado de evaluarla ha de ser Verdadero o Falso (*True or False*). Por ejemplo: [ImporteVenta] > 0, [Producto]="P2", ...

Esta función no se usa por sí sola, sino que se utiliza como parámetro de otras funciones para cambiar el conjunto de resultados sobre los que se realizarán los cálculos.

La función FILTER se puede utilizar como parámetro en funciones como CALCULATE, COUNTROWS, SUMX, etc.

Si conoce el lenguaje SQL, le conviene saber que tiene ciertas similitudes con la condición WHERE.

Veamos un ejemplo de uso:

```
FILTER(Ventas2; [Producto]="P2")
```

Más adelante veremos ejemplos adicionales utilizando FILTER como parámetro de otras funciones que iremos estudiando.

Función CALCULATE()

La función CALCULATE evalúa una expresión en un contexto que ha sido modificado por los filtros que se le han pasado como parámetros. Su sintaxis es:

<code>CALCULATE(<Expresión>;<Filtro1>;<Filtro2>;...;<FiltroN>)</code>

Todos los parámetros de “Filtro” aplicados funcionan como si estuvieran dentro de una función Y() (AND()), es decir, la expresión aplicará sólo a las filas que cumplan todos los filtros.

Como esta definición le puede decir poco en estos momentos, veamos con otras palabras más entendibles para cualquier usuario de Excel. CALCULATE es como la función SUMAR.SI(), pero su potencia y flexibilidad se ha aumentado de forma casi inagotable. Pero, ¿Esto qué quiere decir realmente? Pues que la función CALCULATE en lugar de aplicar sólo a “SUMAR” aplica a cualquier expresión DAX que le pasemos como primer parámetro, y que en lugar de la condición “SI” aplica a todos los filtros que le pasemos como parámetro. Recuerde que el concepto de filtro se utiliza sólo para seleccionar una parte de los valores seleccionados, sino que es un cambio a contexto de filtro, por lo que puede seleccionar un número de filas superior a las filtradas en un momento dado.

Vamos a mostrar un ejemplo muy básico emulando la función SUMAR.SI(), que aunque es poco útil, nos ayudará a entender mejor la función CALCULATE:

```
SumarImporteSiProductoP2:=CALCULATE (SUM ( [Importe] ) ;  
Ventas2 [Producto]="P2")
```

```
SumarImporteSiProductoP2:=CALCULATE (SUM ( [Importe] ) ;  
FILTER (Ventas2; [Producto]="P2") )
```

Las dos expresiones anteriores son equivalentes. Hemos utilizado una expresión de suma y hemos aplicado un filtro para que sólo sume los importes del producto “P2”. Puede comprobar que funciona correctamente en la siguiente imagen:

[Cantidad] ✕ ✓ fx SumarImporteSiProductoP2:=CALCULATE(SUM([Importe]);Ventas2[Producto]="P2")					
Fecha	Cliente	Producto	Cantidad	Precio	Importe
01/01/...	C1	P1	5	8	40
01/01/...	C1	P2	23	10	230
01/01/...	C2	P1	6	8	48
01/01/...	C2	P2	3	10	30
02/01/...	C1	P1	8	8	64
02/01/...	C2	P2	11	10	110
02/01/...	C1	P1	7	8	56
02/01/...	C2	P2	4	10	40
03/01/...	C1	P1	1	8	8
03/01/...	C1	P2	2	10	20
			TotalImporte: 646,00		
			TotalImporteEmpresa: 646		
			RatioVentas: 100,00 %		
			SumarImporteSiProductoP2: 430		

Figura 0-6 Cálculo equivalente a SUMAR.SI() realizado con CALCULATE().

Retomemos el ejemplo que vimos anteriormente cuando explicamos el Contexto de Filtro, ahora seremos capaces de entenderlo con todo detalle:

TotalImporte:=SUM([Importe])

TotalImporteEmpresa:=CALCULATE([TotalImporte];ALL(Ventas2))

Hemos utilizado una expresión de suma y hemos aplicado un contexto de filtro, para que en lugar de aplicar los filtros que haya definido el usuario en la tabla dinámica, aplique siempre a todas las filas de la tabla 'Ventas2', ya que la función ALL() aquí significa “devolver todas las filas de la tabla ventas2 sin filtrar”.

A continuación explicaremos las claves del funcionamiento de CALCULATE, para ello debemos entender cómo se comportan los filtros:

1. Los parámetros de “Filtro” modifican el contexto de la tabla dinámica, cambiándolo a contexto de filtro:
 - a. Si el argumento de filtro actúa sobre un campo que “Ya” está en la tabla dinámica, anula ese contexto y activa el contexto de filtro.
 - b. Si el argumento de filtro actúa sobre un campo que “No” está siendo utilizado en la tabla dinámica, lo agrega al contexto de filtro.

Vamos a exponer un tercer ejemplo de uso de la función CALCULATE. En este caso algo que tiene cierta complejidad y un pobre rendimiento en sistemas de bases de datos relacionales, pero que aquí se realiza con mucha facilidad y muy buen rendimiento. Necesitamos conocer las ventas acumuladas desde el inicio de los tiempos hasta una fecha dada, para ello utilizamos la siguiente expresión DAX:

```
VentasAcumuladas:=CALCULATE([TotalVentas]; FILTER(ALL(Fecha); Fecha[FechaSK] <= MAX(Ventas[FechaSK])) )
```

Veamos el resultado en una tabla dinámica para entenderlo mejor:

Row Labels	TotalVentas	VentasAcumuladas
2012	834.225,95	834.225,95
+ Ene	62.149,33	62.149,33
+ Feb	61.406,32	123.555,65
+ Mar	65.998,70	189.554,35
+ Abr	74.044,43	263.598,78
+ May	73.150,30	336.749,08
+ Jun	73.089,58	409.838,66
+ Jul	83.011,32	492.849,98
+ Ago	94.368,31	587.218,29
+ Sep	68.597,90	655.816,19
+ Oct	59.877,92	715.694,11
+ Nov	58.341,94	774.036,05
+ Dic	60.189,90	834.225,95
2013	244.523,88	1.078.749,83
+ Ene	57.696,34	891.922,29
+ Feb	50.743,58	942.665,87
+ Mar	63.749,86	1.006.415,73
+ Abr	59.620,20	1.066.035,93
+ May	12.713,90	1.078.749,83
Grand Total	1.078.749,83	1.078.749,83

Figura 0-7 Ventas y VentasAcumuladas.

Funciones de Inteligencia de Tiempo (Time Intelligence)

La mayoría de los sistemas analíticos actuales incorporan una serie de características avanzadas para el tratamiento del tiempo. En este caso Power Pivot y DAX no van a ser menos, también las incorporan. Este conjunto de funciones nos permiten manipular los periodos de tiempo, que es un requerimiento muy habitual en BI.

Hay un requerimiento sencillo, pero importante, que debemos tener en cuenta, para poder aplicarlas necesitamos disponer de una columna cuyo tipo de datos sea "Date".

Las funciones de tiempo nos permiten:

- Obtener fechas concretas.
- Obtener un conjunto de fechas.
- Evaluar una expresión sobre un periodo de tiempo especificado.

Hay un grupo devuelve un día concreto, algunas de éstas son:

FIRSTDATE(DateColumn)
LASTDATE(DateColumn)
FIRSTNONBLANK(DateColumn, Expression)
LASTNONBLANK(DateColumn, Expression)
STARTOFMONTH(DateColumn)
ENDOFMONTH(DateColumn)
Etc.

No vamos a estudiarlas una por una, ya que son funciones que bien por su nombre, bien por existir en Excel, son muy sencillas de consultar en la ayuda y comenzar a utilizarlas.

Hay otro grupo que lo que devuelve es un conjunto de fechas, algunas de éstas son:

DATEADD(DateColumn, Numero_de_intervalos, intervalo)
DATESBETWEEN(DateColumn, FechaInicio, FechaFin)
DATESINPERIOD(DateColumn, Numero_de_intervalos, intervalo)
PARALLELPERIOD(DateColumn, Numero_de_intervalos, intervalo)
Etc.

Veamos a continuación un ejemplo que nos ayudará a entender la utilidad de este tipo de funciones. Es muy habitual que cuando estemos trabajando con un periodo de tiempo queramos ver el resultado que se obtuvo en el mismo periodo del año anterior, para ello, una de las funciones que podemos utilizar es PARALLELPERIOD:

```
TotalVentas:=SUM([ImporteVenta])
```

```
VentasAñoAnterior:=CALCULATE([TotalVentas];  
PARALLELPERIOD(Fecha[Fecha];-1;year))
```

```
VentasTrimestreAnterior:=CALCULATE([TotalVentas];  
PARALLELPERIOD(Fecha[Fecha];-1;quarter))
```

Que nos devuelven el siguiente resultado:

Row Labels	TotalVentas	VentasAñoAnterior	VentasTrimestreAñoAnterior
2012	834.225,95		655.816,19
⊕ Ene	62.149,33		
⊕ Feb	61.406,32		
⊕ Mar	65.998,70		
⊕ Abr	74.044,43		189.554,35
⊕ May	73.150,30		189.554,35
⊕ Jun	73.089,58		189.554,35
⊕ Jul	83.011,32		220.284,31
⊕ Ago	94.368,31		220.284,31
⊕ Sep	68.597,90		220.284,31
⊕ Oct	59.877,92		245.977,53
⊕ Nov	58.341,94		245.977,53
⊕ Dic	60.189,90		245.977,53
2013	244.523,88	834.225,95	422.933,64
⊕ Ene	57.696,34	834.225,95	178.409,76
⊕ Feb	50.743,58	834.225,95	178.409,76
⊕ Mar	63.749,86	834.225,95	178.409,76
⊕ Abr	59.620,20	834.225,95	172.189,78
⊕ May	12.713,90	834.225,95	172.189,78
⊕ Jun		834.225,95	172.189,78
⊕ Jul		834.225,95	72.334,10
⊕ Ago		834.225,95	72.334,10
⊕ Sep		834.225,95	72.334,10

Figura 0-8 Comparativa con año y trimestre anteriores.

Y finalmente veremos un tercer grupo que evalúan una expresión en un periodo de tiempo:

TOTALYTD(Expresión, DateColumn, Filtro) -- Año
 TOTALQTD(Expresión, DateColumn, Filtro) -- Trimestre
 TOTALMTD(Expresión, DateColumn, Filtro) -- Mes
 Etc.

Veamos un ejemplo de uso de TOTALYTD y TOTALQTD, que nos permite realizar un acumulado de ventas dentro del año actual, inicializándose a cero automáticamente para cada año.

TotalVentas:=SUM([ImporteVenta])

TotalAñoActual:=TOTALYTD([TotalVentas]; Fecha[Fecha])

En la siguiente imagen puede ver el resultado obtenido:

Row Labels	TotalVentas	TotalAñoActual	TotalTrimestreActual
2012	834.225,95	834.225,95	178.409,76
+ Ene	62.149,33	62.149,33	62.149,33
+ Feb	61.406,32	123.555,65	123.555,65
+ Mar	65.998,70	189.554,35	189.554,35
+ Abr	74.044,43	263.598,78	74.044,43
+ May	73.150,30	336.749,08	147.194,73
+ Jun	73.089,58	409.838,66	220.284,31
+ Jul	83.011,32	492.849,98	83.011,32
+ Ago	94.368,31	587.218,29	177.379,63
+ Sep	68.597,90	655.816,19	245.977,53
+ Oct	59.877,92	715.694,11	59.877,92
+ Nov	58.341,94	774.036,05	118.219,86
+ Dic	60.189,90	834.225,95	178.409,76
2013	244.523,88	244.523,88	
+ Ene	57.696,34	57.696,34	57.696,34
+ Feb	50.743,58	108.439,92	108.439,92
+ Mar	63.749,86	172.189,78	172.189,78
+ Abr	59.620,20	231.809,98	59.620,20
+ May	12.713,90	244.523,88	72.334,10

Figura 0-9 Totales por año y trimestre.

Ejemplo práctico DAX

A continuación vamos a ver algunos ejemplos, que nos permiten definir un informe de análisis del ticket medio en nuestra empresa.

Estos son los cálculos que hemos utilizado:

```
TotalVentas:=SUM(Ventas[ImporteVenta])
```

```
TotalCantidad:=SUM([Cantidad])
```

```
NºOperaciones:=COUNTROWS (VALUES (Ventas[Ticket]))
```

```
TicketMedio (Importe) :=DIVIDE ([TotalVentas]; [NºOperaciones])
```

```
TicketMedio (Cantidad) :=DIVIDE ([TotalCantidad];  
[NºOperaciones])
```

```
VentasAcumuladas:=CALCULATE ([TotalVentas]; FILTER (ALL (Fecha);  
Fecha[Fecha] <= MAX (Fecha[Fecha])))
```

Con esta serie de cálculos podemos obtener el siguiente informe:



Figura 0-10 Informe sobre el ticket medio.

Como puede observar, no es suficiente con hacer los cálculos en DAX. Una vez llegados a este punto, tenemos que ser capaces de mostrarlos con el mayor impacto visual posible. Más adelante iremos profundizando en esas técnicas de visualización,

conociendo las posibilidades de las herramientas de las que disponemos y mostrando múltiples ejemplos de uso.

Si se encuentra realizando alguno de los [cursos o Másters ofertados por SolidQ](#), podrá disponer de diversos ejemplos completos, en los que podrá ver todos los detalles, desde la definición del modelo, pasando por el proceso de carga (ETL), siguiendo por el enriquecimiento de la información y finalizando con las técnicas a seguir para obtener una visualización adecuada. Se le entregarán como parte del material una serie de ejemplos en Excel para que los pueda reutilizar y consultar en todo momento.

Si no es su caso, le recomiendo que se forme con nuestro curso:

Analiza tu Negocio con Excel y Power BI.

Aprende de tus datos

[Quiero obtener información detallada y registrarme](#)

ENLACES ESENCIALES

Visita el [blog de Salvador Ramos](#)

Mi libro “Microsoft BI: Vea el cubo medio lleno” ([descárgalo gratis](#))

Linkedin [Salvador Ramos](#)

Twitter [Salvador Ramos](#)

Visita [los blogs de SolidQ](#)

Libros y publicaciones de SolidQ, la mayoría gratuitos ([descárgalos aquí](#))

Accede a los [Cursos de SolidQ](#)

Accede al [Calendario de cursos de SolidQ](#) (continuamente actualizado)

Accede a los [Servicios de SolidQ](#)

Twitter de [SolidQ](#), y también nuestro [canal en español](#)

GRACIAS

Antes de que te vayas, quiero decirte “Gracias por leer mi libro”.

Sé que hay otros muchos libros en la materia, pero apostaste por este. Ahora sólo quiero pedirte un “pequeño” favor:

Dedica unos minutos y deja un testimonio en Amazon, lo necesito para seguir mejorando y aportando contenido que sea de tu interés.

Y, por último, si crees que merece la pena compartir este libro, ¿podrías tomarte unos segundos y mostrárselo a tus seguidores en las Redes Sociales? El boca a boca es crucial para hacer llegar estos conocimientos al mayor número posible de personas interesadas en la materia, les serás de gran ayuda. Si tienes un momento, te estaré muy agradecido.

¡Gracias!