



Power BI



Power BI



Introducción a Power BI Desktop.....	2
Funcionamiento de Power BI Desktop.....	3
Instalar y ejecutar Power BI Desktop	3
Vistas de Power BI.....	5
Conectar a datos	6
Combinar y dar forma a datos	9
Dar forma a datos.....	9
Combinar datos.....	13
Elaborar informes.....	17
Compartir su trabajo	21
KPI en Power BI	28
Cuándo usar un KPI.....	28
Requisitos de los objetos visuales de KPI.....	28
Cómo crear un KPI.....	28
Crear medidas propias en Power BI Desktop.....	30
¿De qué se tratan estas medidas?	30
Vamos a crear una medida.....	33
Vamos a crear otra medida.....	40
Aspectos básicos de DAX en Power BI Desktop.....	43
¿Qué es DAX?	43
¿Por qué es tan importante DAX?	43
Requisitos previos	43
¡Comencemos!.....	44
Sintaxis	44
Funciones	47
Resumen.....	49
Introducción a Power Query y el Lenguaje M.....	52
El flujo de trabajo – paso a paso.....	52
Cómo entender el lenguaje.....	54
Lenguaje M – recursos de aprendizaje.....	55

Introducción a Power BI Desktop

Esta es la **Guía de introducción de Power BI Desktop**. Donde dará un paseo introductorio por Power BI Desktop se familiarizará con su funcionamiento y sus capacidades. Al final, será capaz de crear modelos de datos eficaces, junto con increíbles informes, que expandirán sus esfuerzos de inteligencia empresarial.



Power BI Desktop le permite crear una colección de consultas, conexiones de datos e informes que se pueden compartir fácilmente con otros usuarios. Power BI Desktop integra tecnologías de eficacia comprobada de Microsoft (un potente motor de consultas, capacidades de modelado de datos y visualizaciones) y funciona en línea sin problemas con el **servicio Power BI**.

Con la combinación de **Power BI Desktop** (donde los analistas y otros usuarios pueden crear conexiones de datos eficaces, modelos e informes) y el **servicio Power BI** (donde se pueden compartir informes de Power BI Desktop para que otros usuarios los vean e interactúen con ellos) se obtiene nueva información del mundo de los datos que es más fácil de modelar, crear, compartir y ampliar.

Los analistas de datos encontrarán en Power BI Desktop una herramienta eficaz, flexible y muy accesible para conectarse con datos y darles forma, crear modelos eficaces y elaborar informes con la estructura adecuada.

Fucionamiento de Power BI Desktop

Con Power BI Desktop, puede *conectarse a datos* (que normalmente son varios orígenes de datos), *dar forma a dichos datos* (con las consultas que crean modelos de datos precisos y convincentes) y usar modelos para *crear informes* (que otros pueden aprovechar, compartir y usar como punto de partida).

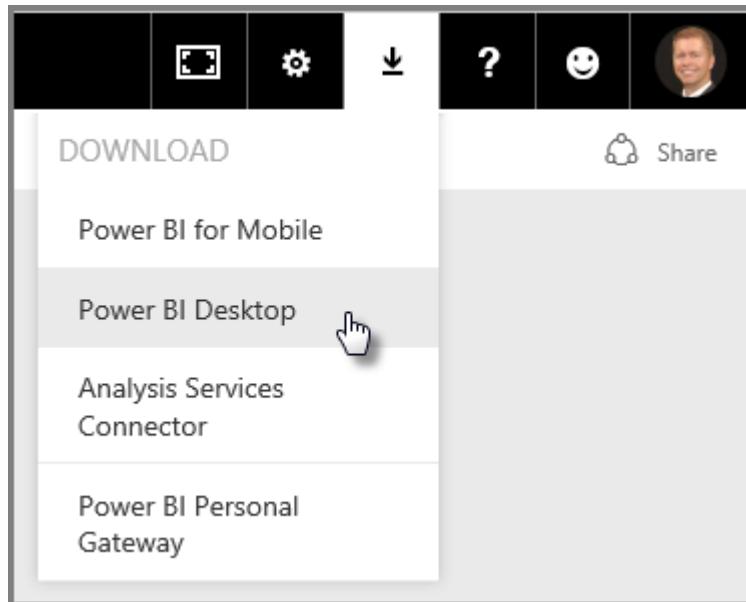
Una vez que haya completado los pasos (conectarse a datos, darles forma, crear el informe) satisfactoriamente, puede guardar el trabajo en el formato de archivos de Power BI Desktop, que es la extensión .pbix. Los archivos de Power BI Desktop se pueden compartir como cualquier otro archivo, pero la forma más contundente de hacerlo es cargarlos (compartirlos) en el **servicio Power BI**.

Power BI Desktop centraliza, simplifica y agiliza lo que de otro modo podría ser un proceso de diseño y creación de repositorios e informes de inteligencia empresarial disperso, arduo y desconectado.

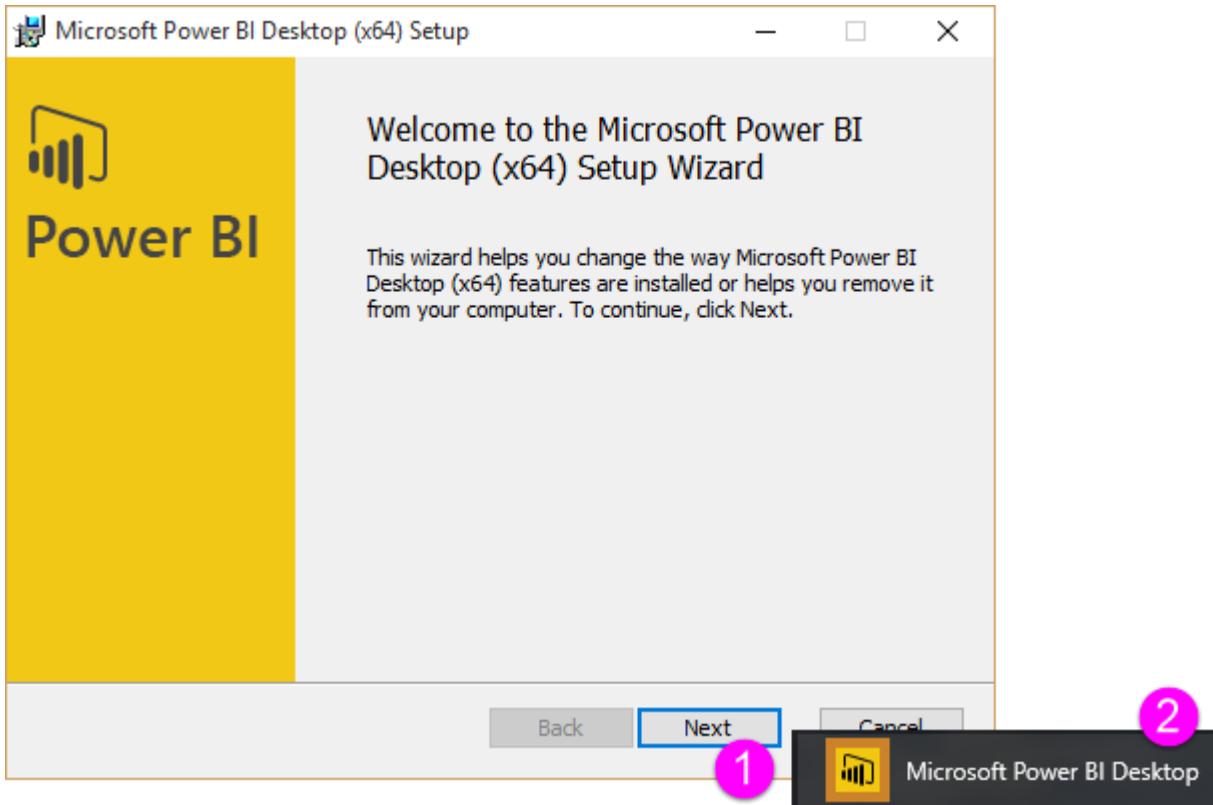
¿Listo para probarlo? Comencemos.

Instalar y ejecutar Power BI Desktop

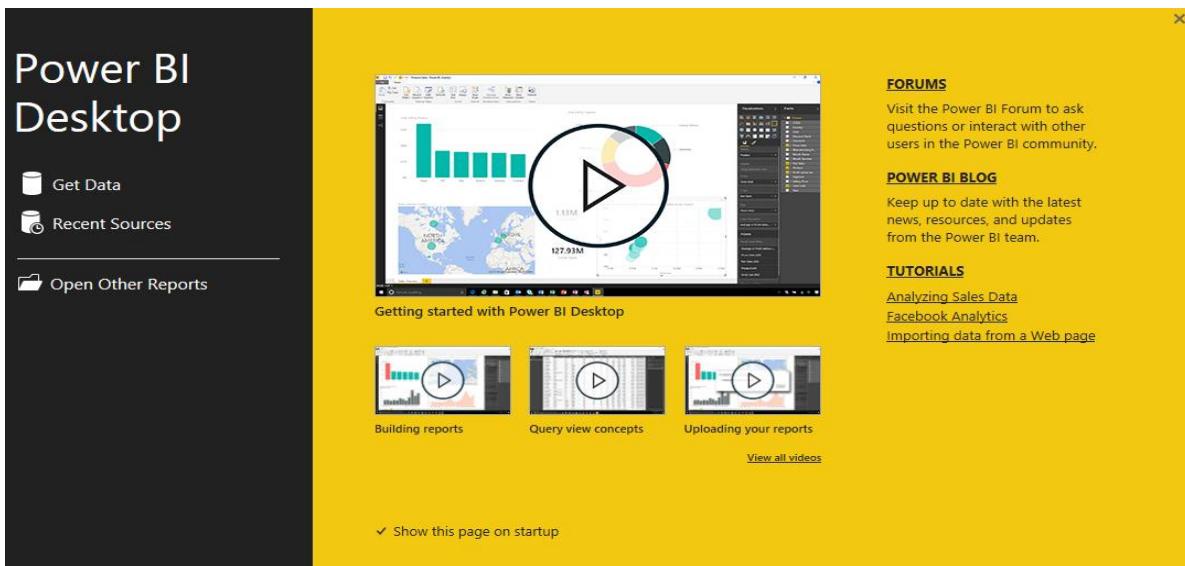
Puede descargar Power BI Desktop desde el servicio **Power BI**, seleccionando el icono de engranaje. A continuación, seleccione **Power BI Desktop**.



Power BI Desktop se instala como una aplicación y se ejecuta en el escritorio.

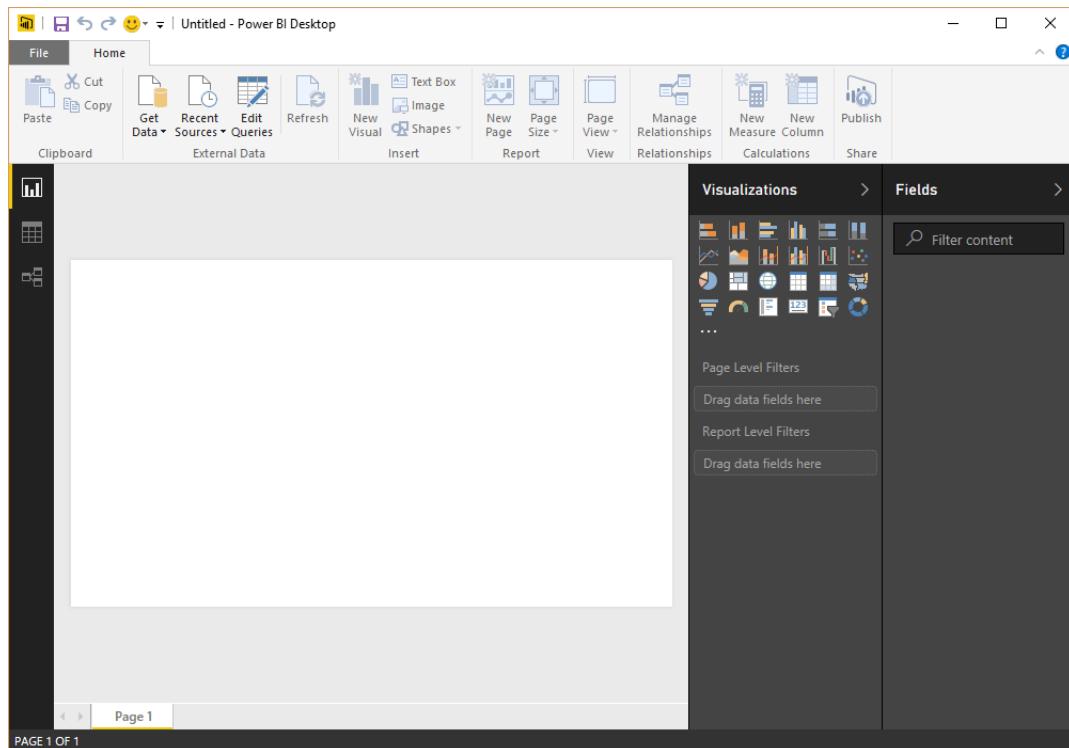


Cuando se ejecuta Power BI Desktop, aparece una pantalla de *bienvenida*.



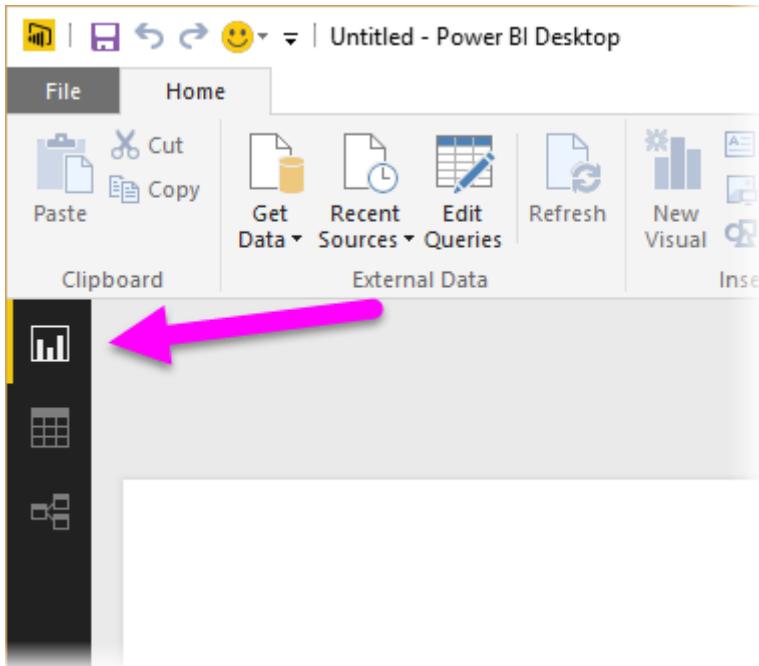
Puede Obtener datos, ver Orígenes recientes o Abrir otros Informes directamente desde la pantalla de bienvenida (desde los vínculos del panel izquierdo). Si cierra la pantalla (seleccione la x situada en la esquina superior derecha),

Vistas de Power BI



Hay tres vistas en Power BI Desktop: vista **Informe**, vista **Datos** y vista **Relaciones**. Power BI Desktop también incluye el **Editor de consultas**, que se abre en una ventana independiente. En el **Editor de consultas**, puede crear consultas y transformar los datos para luego cargar dicho modelo de datos refinados en Power BI Desktop y crear informes.

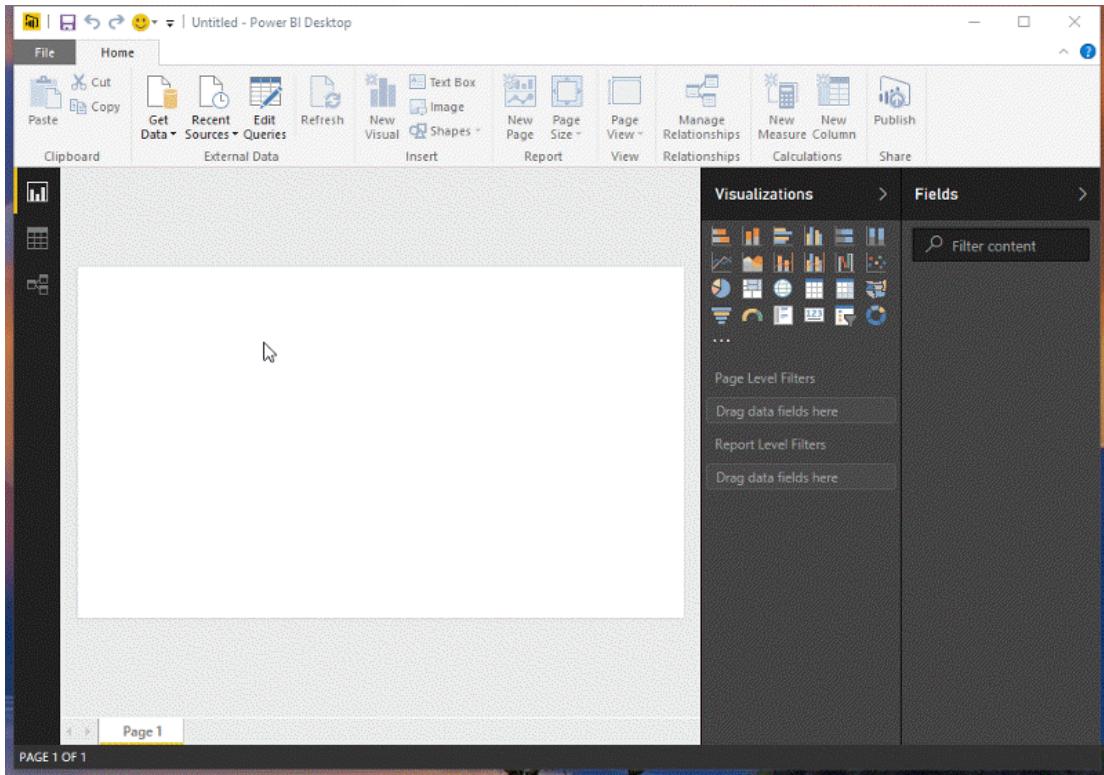
La pantalla siguiente muestra los iconos de las tres vistas en la parte izquierda de Power BI Desktop: **Informe**, **Datos** y **Relaciones**, de arriba a abajo. La vista actual se indica mediante la barra amarilla de la parte izquierda. En este caso, se muestra la vista **Informe**. Puede cambiar las vistas haciendo clic en cualquiera de los tres iconos.



Con Power BI Desktop instalado, está listo para conectarse a datos, darles forma y generar informes (normalmente en ese orden). En las siguientes secciones, se hace un recorrido por cada operación.

Conectarse a datos

Con Power BI Desktop instalado, está listo para conectarse con el mundo en expansión continua de los datos. Hay *todo tipo* de orígenes de datos disponibles en la ventana de consulta. La siguiente imagen muestra cómo conectarse a datos, seleccionando la cinta **Inicio** y, a continuación, **Obtener datos > Más**.



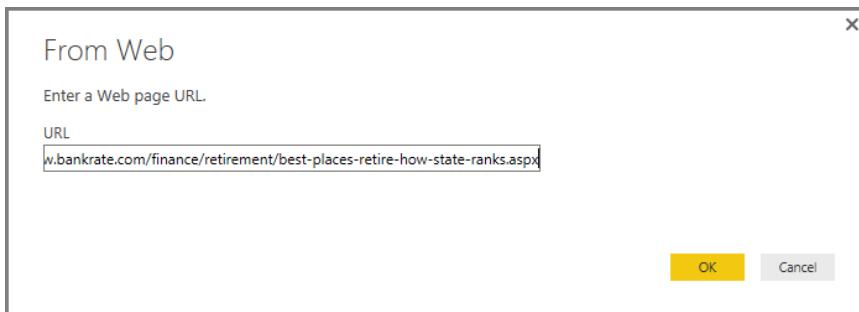
Para este paseo, nos conectaremos a dos orígenes de datos **web** diferentes.

Imagine que se jubila (desea vivir donde hay mucho sol, mejores tasas de impuestos y buena cobertura de servicios médicos), o quizás que es un analista de datos y desea información para ayudar a sus clientes. Por ejemplo, quizás desee ayudar a su distribuidor de gafas de sol a alcanzar su objetivo de ventas en un lugar donde siempre brilla el sol.

Cualquiera que sea el caso, el siguiente recurso web tiene datos interesantes acerca de estos temas y mucho más:

<http://www.bankrate.com/finance/retirement/best-places-retire-how-state-ranks.aspx>

Seleccione **Obtener datos > Web** y pegue la dirección.



Al seleccionar **Aceptar**, la funcionalidad de **consulta** de Power BI Desktop entra en acción. La consulta se conecta con el recurso web y la ventana **Navegador** devuelve la información encontrada en la página web. En este caso, encuentra una tabla (*Tabla 0*) y el documento web general. Nos interesa la tabla, así que la seleccionamos de la lista. La ventana **Navegador** muestra una vista previa.

The screenshot shows the 'Navigator' window with the title 'Table 0'. The preview area displays a list of 22 rows, each containing a link to 'Check out how your state ranks for retirement' followed by the state name and its rank. The columns are labeled 'Header', 'Overall rank', and 'State'. The 'Edit' button at the bottom of the preview area is highlighted with a yellow background.

Header	Overall rank	State
Check out how your state ranks for retirement	1	Wyoming
Check out how your state ranks for retirement	2	Colorado
Check out how your state ranks for retirement	3	Utah
Check out how your state ranks for retirement	4	Idaho
Check out how your state ranks for retirement	5	Virginia
Check out how your state ranks for retirement	6	Iowa
Check out how your state ranks for retirement	7	Montana
Check out how your state ranks for retirement	8	South Dakota
Check out how your state ranks for retirement	9	Arizona
Check out how your state ranks for retirement	10	Nebraska
Check out how your state ranks for retirement	11	Minnesota
Check out how your state ranks for retirement	12	Maine
Check out how your state ranks for retirement	13	North Dakota
Check out how your state ranks for retirement	14	Kansas
Check out how your state ranks for retirement	15	Vermont
Check out how your state ranks for retirement	16	New Hampshire
Check out how your state ranks for retirement	17	Wisconsin
Check out how your state ranks for retirement	18	Massachusetts
Check out how your state ranks for retirement	19	Delaware
Check out how your state ranks for retirement	20	Michigan
Check out how your state ranks for retirement	21	Pennsylvania
Check out how your state ranks for retirement	22	Washington

En este punto, se puede modificar la consulta antes de cargar la tabla (para ello, seleccione **Editar** desde la parte inferior de la ventana) o se puede cargar directamente la tabla.

Si seleccionamos **Editar**, el Editor de consultas inicia una vista representativa de la tabla. Se muestra el panel **Configuración de consulta** (si no es así, puede seleccionar **Vista** desde la cinta de opciones y, a continuación, seleccionar **Mostrar > Configuración de consulta** para mostrar el panel **Configuración de consulta**). Ofrece el siguiente aspecto.

The screenshot shows the Power BI Query Editor interface. On the left, there's a navigation bar with 'File', 'Home', 'Transform', 'Add Column', and 'View' tabs. Below it are buttons for 'Close & Apply', 'New Source', 'Recent Sources', 'Refresh Preview', 'Properties', 'Advanced Editor', 'Choose Columns', 'Remove Columns', 'Reduce Rows', 'Split Column', 'Group By', 'Data Type: Text', 'Use First Row As Headers', and 'Combine'. The main area displays a table titled 'Table 0' with 21 rows. The first column is labeled 'Header' and contains the text 'Check out how your state ranks for retirement' repeated 21 times. The second column is 'Overall rank' and the third is 'State'. The 'State' column lists US states from Wyoming at rank 1 to Pennsylvania at rank 21. A preview window on the right shows the same data. The 'Query Settings' pane on the right shows the table is named 'Table 0' and has one applied step: 'Changed Type'.

Para obtener más información sobre cómo conectarse a datos, consulte [Conectarse a los datos en Power BI Desktop](#).

En la siguiente sección, ajustaremos los datos de acuerdo con nuestras necesidades. El proceso de ajustar datos conectados se conoce como *dar forma* a los datos.

Combinar y dar forma a datos

Ahora que nos hemos conectado a un origen de datos, es necesario ajustar los datos según se requiere. Ajustar a veces significa *transformar* los datos, por ejemplo, al cambiar el nombre de columnas o tablas, convertir texto en números, quitar filas, configurar una primera fila como encabezado, etcétera.

El Editor de consultas de Power BI Desktop hace un amplio uso de los menús contextuales, además de tener las tareas disponibles en la cinta de opciones. La mayor parte de lo que se puede seleccionar en la cinta **Transformar** también está disponible en el menú que aparece al hacer clic con el botón secundario en un elemento (por ejemplo, una columna).

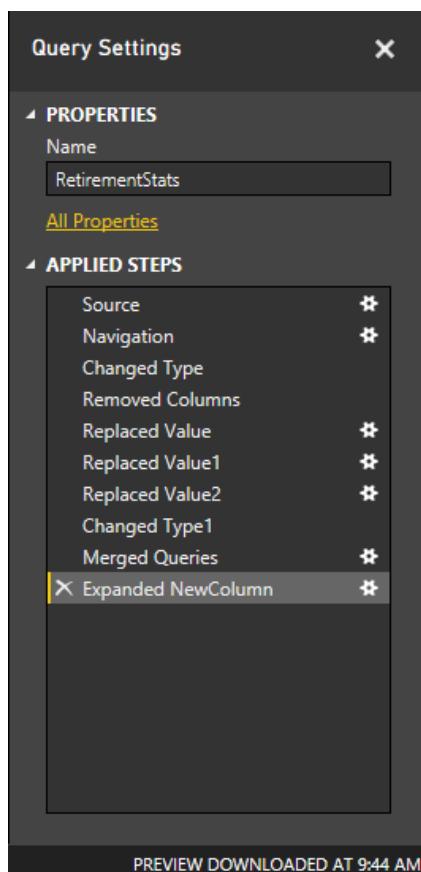
Dar forma a datos

Al dar forma a los datos en el **Editor de consultas**, se proporcionan instrucciones paso a paso (que el **Editor de consultas** lleva a cabo automáticamente) para ajustar los datos a

medida que el **Editor de consultas** los carga y presenta. El origen de datos original no se ve afectado; únicamente la vista de los datos se ajusta o se *forma*.

Los pasos especificados (como cambiar el nombre de una tabla, transformar un tipo de datos o eliminar columnas) se registran en el **Editor de consultas** y, cada vez que este se conecta al origen de datos, dichos pasos se vuelven a aplicar para que los datos siempre muestren la forma que eligió. Este proceso se produce siempre que se use la consulta en Power BI Desktop, o cuando alguien más use su consulta compartida, como en el servicio **Power BI**. Estos pasos se capturan, de manera secuencial, en el panel **Configuración de consulta** en **Pasos aplicados**.

La siguiente imagen muestra el panel **Configuración de consulta** de una consulta a la que se ha dado forma; en los párrafos siguientes se examinará cada uno de los pasos.



Volvamos a los datos para la jubilación que encontramos al conectar con el origen de datos web para darles forma según lo requerimos.

Para empezar, la mayoría de las clasificaciones se presentaron en el **Editor de consultas** como números enteros, pero no todos ellos (una columna contiene texto y números, por lo tanto, no se convirtió automáticamente). Es necesario que los datos sean números. No hay problema, simplemente haga clic con el botón secundario en el encabezado de la columna y seleccione **Cambiar tipo > Número entero** para cambiar el tipo de datos. Si se necesita elegir más de una columna, puede seleccionar una columna, mantener presionada la tecla **MAYÚS**, seleccionar columnas adyacentes adicionales y, a continuación, hacer clic en un encabezado de columna para cambiar todas las columnas seleccionadas. También se puede usar **CTRL** para seleccionar columnas no adyacentes.

The screenshot shows the Power BI Query Editor interface. The ribbon at the top has tabs for File, Home, Transform, Add Column, and View. The Transform tab is selected. Below the ribbon is a toolbar with various icons for file operations and query management. A context menu is open over a column named "Health". The menu includes options like Copy, Remove, Remove Other Columns, Duplicate Column, Remove Duplicates, Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., Split Column, Group By..., Fill, Unpivot Columns, Unpivot Other Columns, Rename..., Move, Drill Down, and Add as New Query. On the right side of the editor, there is a "Query Settings" pane with sections for Properties (Name: Table 0, All Properties) and Applied Steps (Source). The "Source" step is currently selected, showing its properties: Decimal Number, Fixed Decimal Number, Whole Number (which is highlighted), Date/Time, Date, Time, Date/Time/Timezone, Duration, Text (with a checked checkbox), True/False, Binary, and Using Locale... At the bottom of the editor, it says "9 COLUMNS, 50 ROWS" and "PREVIEW DOWNLOADED AT 9:44 AM".

También puede cambiar, o *transformar*, las columnas de texto a encabezado desde la cinta **Transformar**. Esta es la cinta de opciones **Transformar**, con una flecha que señala hacia el botón **Tipo de datos**, que permite transformar el tipo de datos actual.

This screenshot is identical to the one above, showing the Power BI Query Editor with the Transform ribbon selected. The context menu over the "Health" column is also identical, with the "Whole Number" option highlighted under the "Change Type" section. The "Query Settings" pane and the status bar at the bottom are also the same.

Tenga en cuenta que en la **Configuración de consulta**, los **Pasos aplicados** reflejan los cambios realizados. Si desea quitar cualquier paso del proceso de forma, simplemente seleccione el paso y, a continuación, seleccione la X a la izquierda del paso.

The screenshot shows the Power BI Editor interface. On the left, there's a data grid with columns labeled 'Care quality', 'Tax rate', and 'Weather'. The 'Weather' column contains numerical values (1, 19, 23, 27, 21, 22, 13, 3, 17, 26, 45). On the right, the 'Query Settings' pane is open, displaying 'Table 0' under 'Name'. Under 'APPLIED STEPS', there are three entries: 'Source', 'Navigation', and 'Changed Type'. The 'Changed Type' step is highlighted with a yellow border and has a pink arrow pointing to it from the 'Weather' column in the data grid.

Es preciso realizar algunos cambios más para que la consulta llegue al punto donde la queremos:

- *Quitar la primera columna*: no se necesita, solo incluye las filas redundantes que dicen "Mira cómo clasifica su estado para la jubilación", que es un defecto de esto que era una tabla basada en web
- *Corregir algunos errores*: en la página web, una columna tenía texto combinado con los números (varios estados están vinculados en una categoría). Eso funciona bien en el sitio web, pero no en nuestro análisis de datos. El asunto es fácil de corregir, en este caso, y nos permite ver algunas características y capacidades interesantes del **Editor de consultas** y los **Pasos aplicados**
- *Cambiar el nombre de la tabla*: **Tabla 0** no es un descriptor útil, pero cambiarlo es sencillo

Cada uno de estos pasos se muestra en [Combinar y dar forma a datos en Power BI Desktop](#). No dude en consultar la página, o continúe revisando este documento para ver lo que haría a continuación. La siguiente sección retoma el proceso después de que se han aplicado los cambios anteriores.

Combinar datos

Los datos acerca de los diferentes estados son interesantes y serán útiles en la creación de consultas y esfuerzos de análisis adicionales. Pero hay un problema: la mayoría de los datos usa una abreviatura de dos letras para los códigos de estado, no el nombre completo del estado. Se necesita una manera de asociar las abreviaturas con los nombres de los estados.

Estamos de suerte: hay otro origen de datos públicos que hace justamente eso, pero necesita algunos ajustes de forma considerables antes de que podamos conectarlo con la tabla de jubilación. Este es el recurso web de las abreviaturas de los estados:

http://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations

Desde la cinta **Inicio** del **Editor de consultas**, seleccione **Obtener datos > Web** y escriba la dirección, seleccione **Aceptar** y la ventana **Navegador** mostrará la información encontrada en la página web.

The screenshot shows the Microsoft Power BI Navigator dialog box. On the left, there is a tree view of the downloaded content from the URL http://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations. The tree includes a folder named 'Document', a file named 'Mis-matches between USPS and USCG co...', and two tables. The second table, 'Table[edit]', is selected and highlighted with a yellow border. The main pane displays the contents of 'Table[edit]' with the following data:

Region Name	Region Status	Codes ISO	Codes ANSI
Region	Region	Codes	Codes
Name	Status	ISO	ANSI
United States	Federal state	US USA 840	US
Alabama	State	US-AL	AL
Alaska	State	US-AK	AK
Arizona	State	US-AZ	AZ
Arkansas	State	US-AR	AR
California	State	US-CA	CA
Colorado	State	US-CO	CO
Connecticut	State	US-CT	CT
Delaware	State	US-DE	DE
District of Columbia	Federal district	US-DC	DC
Florida	State	US-FL	FL
Georgia	State	US-GA	GA
Hawaii	State	US-HI	HI
Idaho	State	US-ID	ID
Illinois	State	US-IL	IL
Indiana	State	US-IN	IN
Iowa	State	US-IA	IA
Kansas	State	US-KS	KS
Kentucky	State (Commonwealth)	US-KY	KY
Louisiana	State	US-LA	LA

At the bottom right of the dialog box are the 'OK' and 'Cancel' buttons.

Seleccionamos **Table[edit]** porque incluye los datos deseados, pero será necesario aplicar bastante forma para reducir los datos de la tabla. Cada uno de estos pasos también se muestra en [Combinar y dar forma a datos en Power BI Desktop](#). Para resumir los pasos, esto es lo que haremos:

Primero, seleccionamos **Editar** y, a continuación:

- *Quitar las dos primeras filas*: son el resultado de la forma en que se creó la tabla de la página web y no las necesitamos.
- *Quitar las 26 filas inferiores*: son todos los territorios, que no es necesario incluir.
- *Filtrar Washington DC*: la tabla de estadísticas de jubilación no incluye el controlador de dominio, por lo que la excluiremos de nuestra lista.
- *Quitar unas cuantas columnas innecesarias*: únicamente se necesita la correspondencia entre las abreviaturas oficiales de dos letras y los estados, por lo que se pueden eliminar las demás columnas.
- *Usar la primera fila como encabezado*: puesto que se eliminaron las primeras tres filas, queremos que la fila superior actual sea el encabezado.

Nota: es un buen momento para señalar que la secuencia de los pasos aplicados en el **Editor de consultas** es importante y puede afectar a la manera en que se da forma a los datos. También es importante tener en cuenta cómo un paso puede afectar a otro paso posterior; si quita un paso de los **Pasos aplicados**, es posible que los pasos siguientes no tengan el efecto buscado originalmente, debido al impacto de la secuencia de pasos de la consulta.

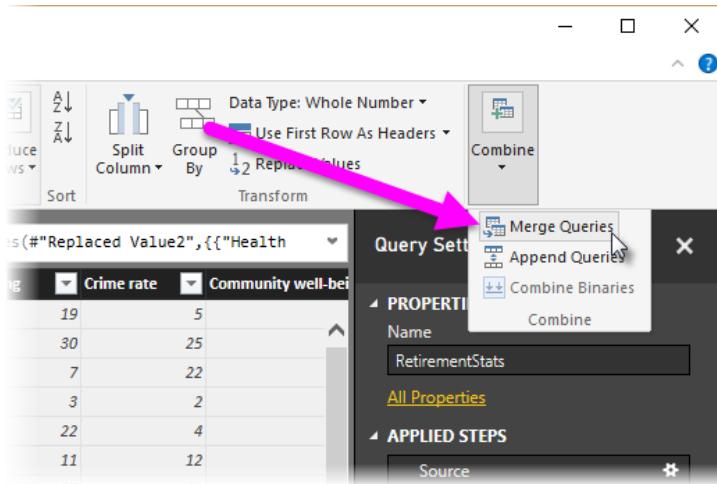
- *Cambiar el nombre de las columnas y la tabla*: como de costumbre, existe un par de formas para cambiar el nombre de una columna. Puede elegir la que prefiera.

Una vez que se ha dado forma a la tabla *StateCodes*, es posible combinar las dos tablas, o consultas, en una; ya que las tablas que ahora tenemos son el resultado de las consultas aplicadas a los datos, a menudo se les denomina *consultas*.

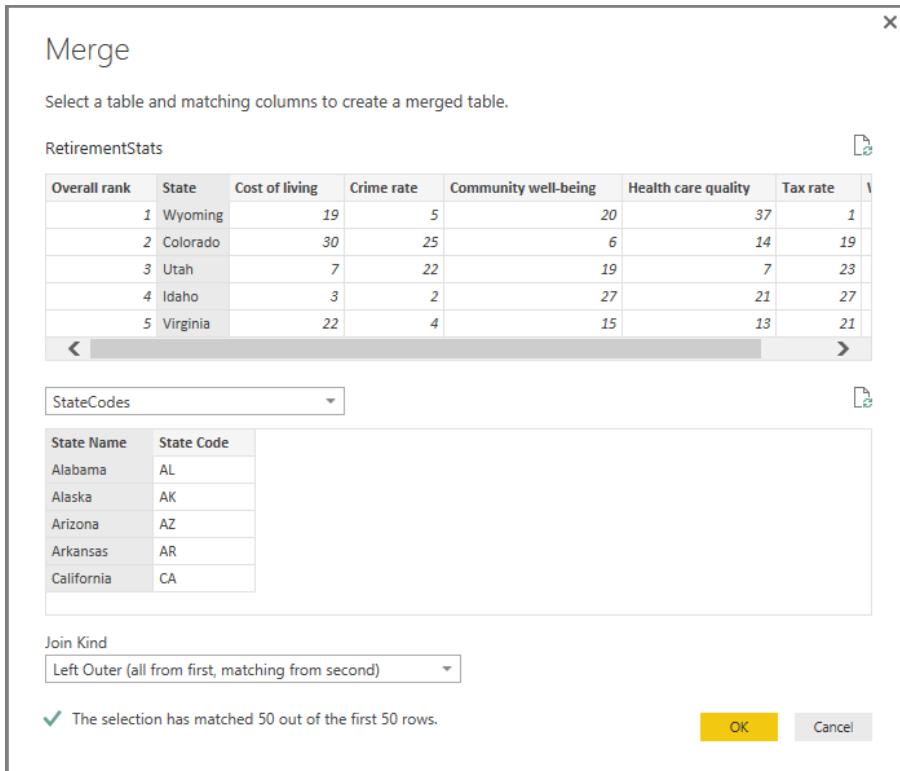
Hay dos métodos principales para combinar las consultas, que son *fusionar* y *anexar*.

Cuando se tienen una o varias columnas para agregar a otra consulta, se **fusionan** las consultas. Cuando se tienen filas de datos adicionales que desea agregar a una consulta existente, se **anexa** la consulta.

En este caso queremos fusionar las consultas. Para empezar, se selecciona la consulta *en la que* se quiere fusionar la otra consulta, a continuación, se selecciona **Fusionar consultas** desde la pestaña **Inicio** en la cinta de opciones.



Aparecerá la ventana **Fusionar**, donde se le pide seleccionar la tabla que se quiere combinar con la tabla seleccionada y, a continuación, las columnas coincidentes que se usarán para la fusión. Seleccione *Estado* desde la tabla (consulta) *RetirementStats*, a continuación, seleccione la consulta *StateCode* (en este caso es fácil, dado que solo hay otra consulta; cuando se conecta a muchos orígenes de datos, hay muchas consultas para elegir). Al seleccionar las columnas coincidentes correctas (Estado de *RetirementStats*, y Nombre del estado de *StateCode*) la ventana **Fusionar** tiene una apariencia similar a la siguiente y el botón **Aceptar** está habilitado.



Se crea una **Nueva columna** al final de la consulta, con el contenido de la tabla (consulta) que se combinó con la consulta actual. Todas las columnas de la consulta combinada se comprimen en la **Nueva columna**, pero puede seleccionar **Expandir** la tabla e incluir

cualquier columna que desee. Para expandir la tabla combinada y seleccionar qué columnas desea incluir, seleccione el ícono de expandir (). Aparecerá la ventana **Expandir**.

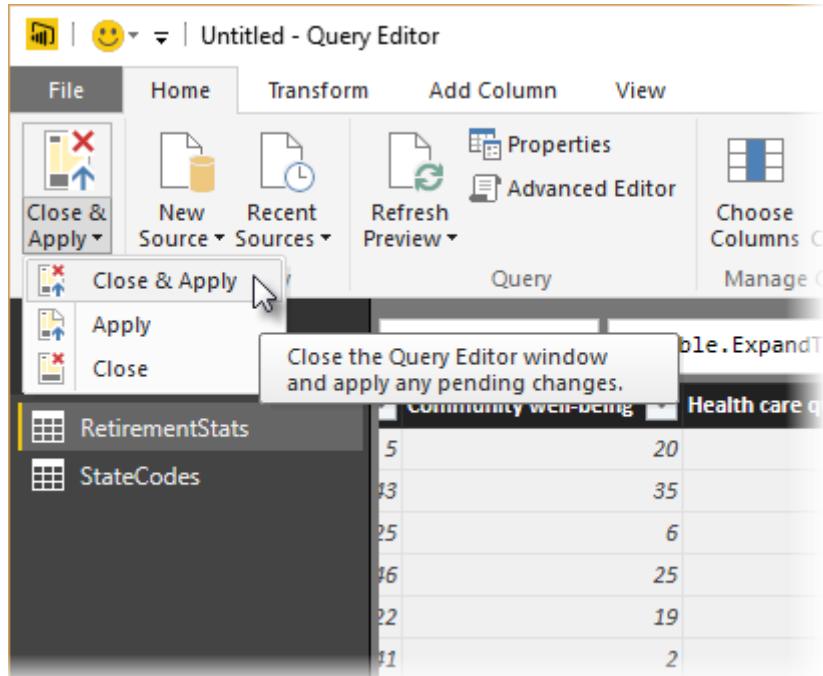
The screenshot shows the Power BI Data Editor interface. On the left, there's a preview of a table with columns like 'Entity well-being', 'Health care quality', 'Tax rate', 'Weather', and 'NewColumn'. The 'NewColumn' column contains numerical values. In the center, a modal dialog titled 'Search Columns to Expand' is open. It has two radio buttons: 'Expand' (selected) and 'Aggregate'. Below them is a list of checked checkboxes: '(Select All Columns)', 'State Name', and 'State Code'. At the bottom of the dialog is a checked checkbox for 'Use original column name as prefix'. There are 'OK' and 'Cancel' buttons at the bottom right. On the right side of the editor, a 'Query Settings' pane is visible, showing 'Properties' and 'Applied Steps' sections. The 'Applied Steps' section lists several steps: Source, Navigation, Changed Type, Removed Columns, Replaced Value, Replaced Value1, Replaced Value2, and 'Changed Type1' (which is currently selected). Below these is a step labeled 'Merged Queries'.

En este caso, solamente queremos la columna *Código de estado*, así que seleccione solo dicha columna y, a continuación, seleccione **Aceptar**. Desactive la casilla de **usar el nombre de la columna original como prefijo** puesto que no es necesaria; si deja la casilla seleccionada, la columna combinada se denominará *NewColumn.State Code* (el nombre de la columna original, o *NewColumn*, un punto, y a continuación el nombre de la columna que se incorporó a la consulta).

Nota: ¿desea experimentar con diferentes maneras de incorporar la tabla *NewColumn*? Puede experimentar un poco y si no le gustan los resultados, elimine ese paso de la lista de **Pasos aplicados** en el panel **Configuración de consulta** y la consulta regresará al estado anterior a la aplicación del paso **Expandir**. Es como una segunda oportunidad, que puede tomar tantas veces como sea necesario hasta que el proceso de expansión tenga la apariencia que deseé.

Ahora tiene una sola consulta (tabla) que combina dos orígenes de datos, a los cuales se dio forma para ajustarse a las necesidades pertinentes. Esta consulta puede servir como base para una gran cantidad de conexiones de datos interesantes, como estadísticas de costes de alojamiento, datos demográficos y oportunidades de trabajo en cualquier estado.

Por ahora, tenemos datos suficientes para crear algunos informes interesantes, todo ello en Power BI Desktop. Puesto que estamos ante un hito, guardemos este archivo de Power BI Desktop con el nombre de **Introducción a Power BI Desktop**. Para aplicar los cambios en el **Editor de consultas** y cargarlos en Power BI Desktop, seleccione **Cerrar y aplicar** en la cinta **Inicio**.

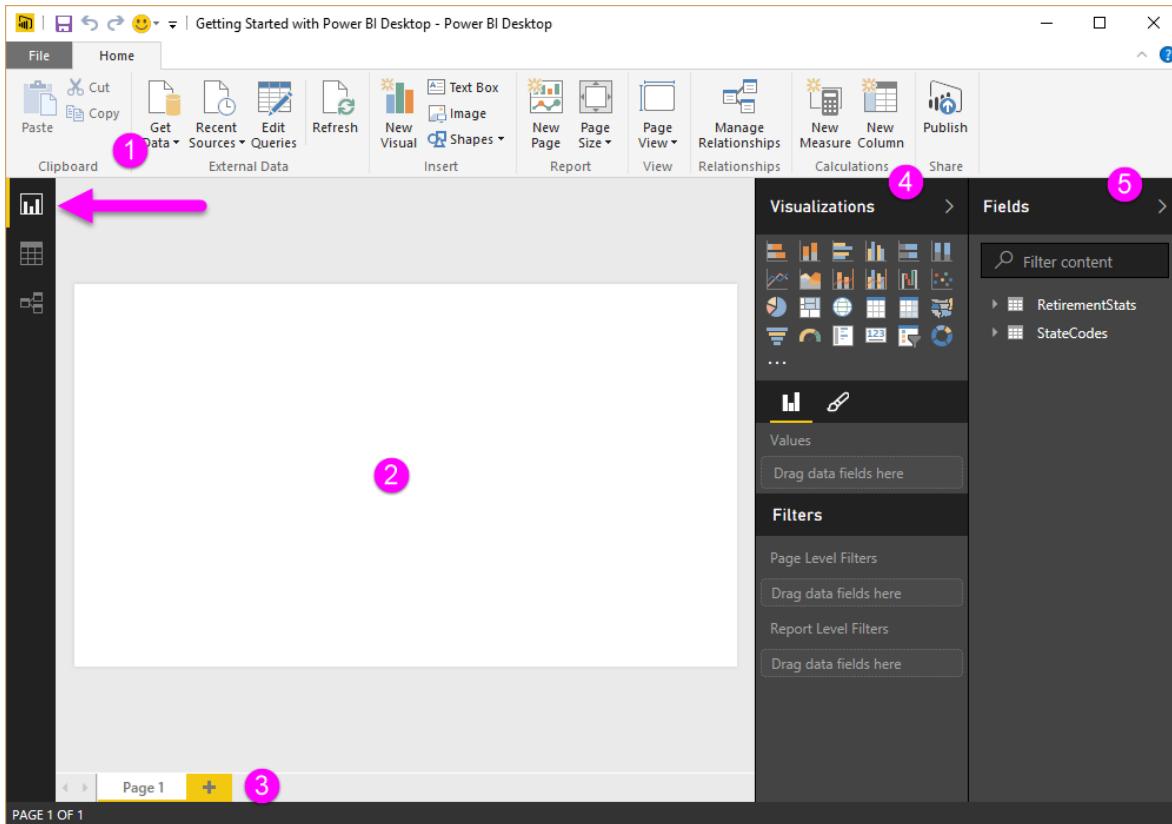


Elaborar informes

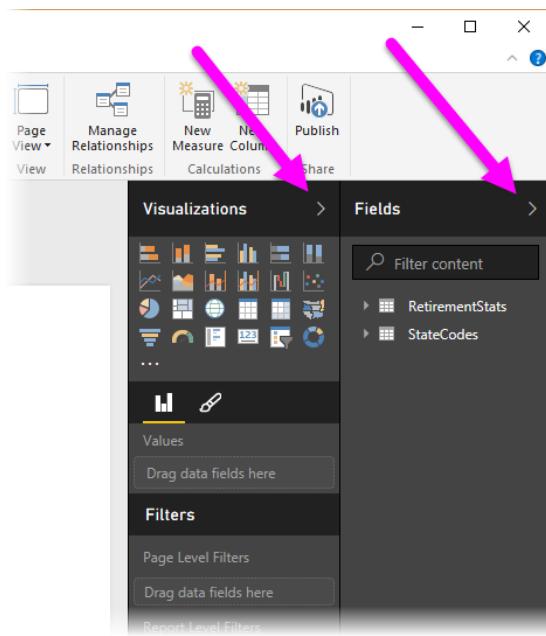
Una vez cargada la tabla, pueden realizarse cambios adicionales y es posible volver a cargar un modelo para aplicar los cambios realizados. Pero por ahora basta con esto. En la vista **Informe** de Power BI Desktop, puede empezar a generar informes.

La vista **Informe** tiene cinco áreas principales:

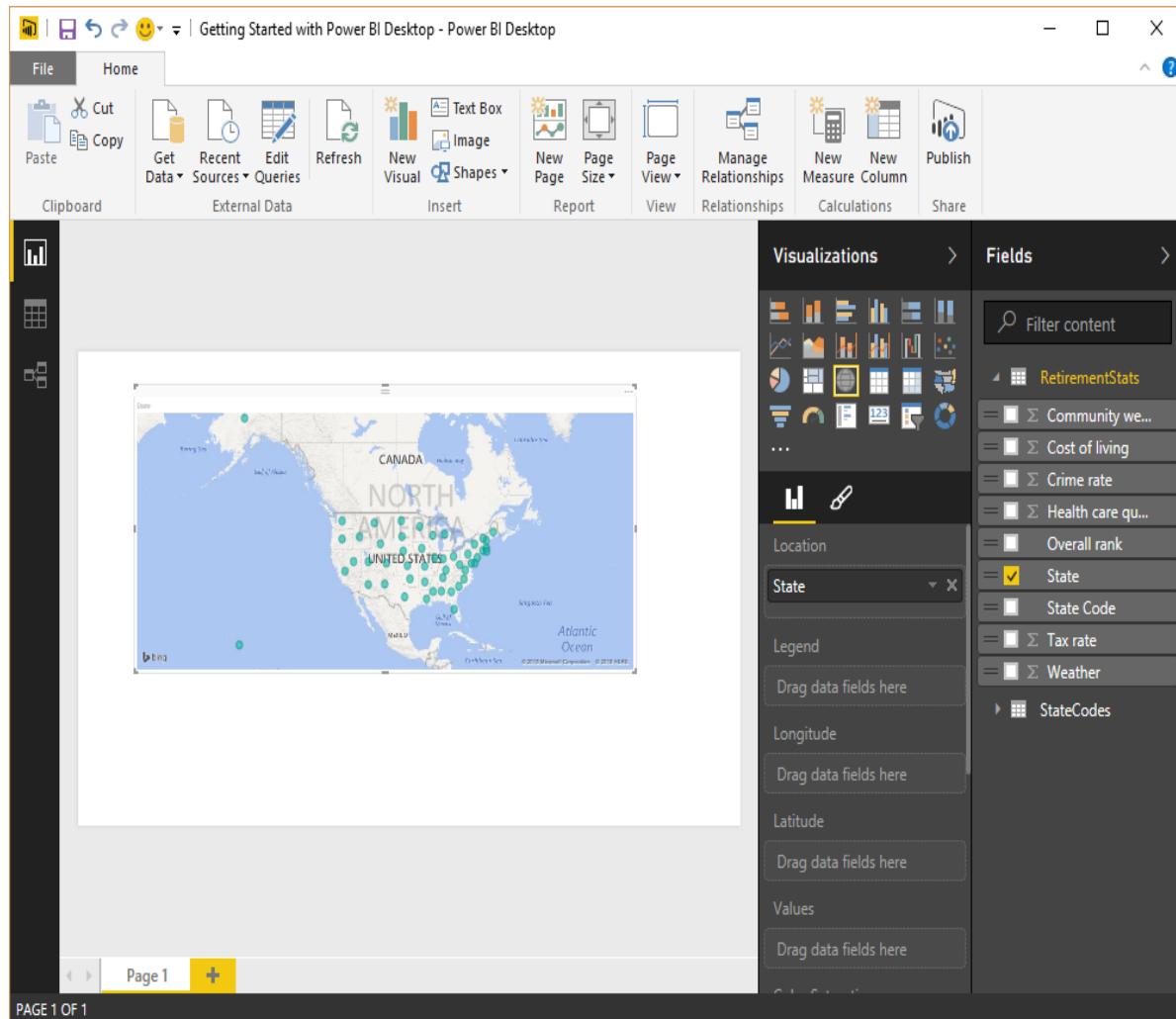
1. La cinta de opciones, que muestra las tareas comunes asociadas con los informes y visualizaciones
2. La vista **Informe**, o lienzo, donde se crean y se organizan las visualizaciones
3. El área de pestaña **Páginas** situada a lo largo de la parte inferior, que permite seleccionar o agregar una página de informe
4. El panel **Visualizaciones**, donde puede cambiar las visualizaciones, personalizar los colores o ejes, aplicar filtros, arrastrar campos, etc.
5. El panel de **Campos**, donde elementos de consulta y filtros se pueden arrastrar a la vista **Informe**, o bien al área **Filtros** del panel **Visualizaciones**.



Los paneles **Visualizaciones** y **Campos** se pueden contraer al seleccionar la flecha pequeña situada junto al borde, lo que proporciona más espacio en la vista **Informe** para crear visualizaciones interesantes. Cuando se modifican visualizaciones, también verá estas flechas hacia arriba o hacia abajo, lo que significa que se puede expandir o contraer la sección, según corresponda.



Para crear una visualización, basta con arrastrar un campo desde la lista **Campos** hasta la vista **Informe**. En este caso, arrastre el campo *Estado* de *RetirementStats* y vea qué sucede.



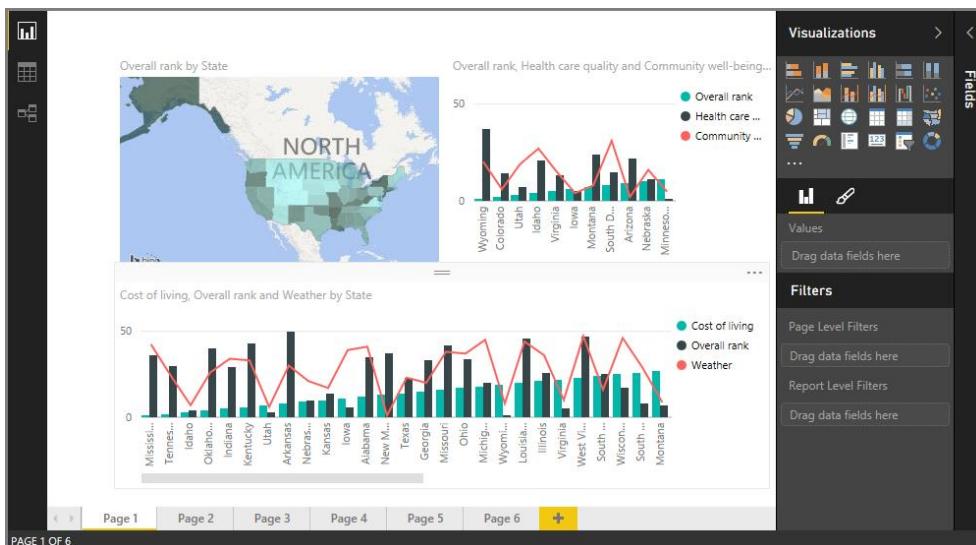
Preste atención... Power BI Desktop crea automáticamente una visualización basada en un mapa, ya que reconoce que el campo *State* contiene datos de geolocalización.

Observe que en el panel **Visualizaciones**, puedo seleccionar diferentes tipos de visualizaciones y en el área situada debajo de los iconos, puedo arrastrar campos a las distintas áreas para aplicar una leyenda o modificar la visualización.

The screenshot shows the Power BI Desktop interface. On the left, there's a map of North America with green dots representing data points. The ribbon at the top has tabs like File, Home, Insert, Page View, Manage Relationships, New Measure, and Publish. A pink arrow points from the 'Manage Relationships' tab towards the Visualizations pane. The Visualizations pane on the right contains a section for 'Location' where 'State' is selected. Below it are sections for Legend, Longitude, Latitude, and Values, each with a 'Drag data fields here' placeholder. To the right of the Visualizations pane is the Fields pane, which lists fields under 'RetirementStats' and 'StateCodes' categories, with 'State' checked.

Avancemos un poco más rápido para ver qué apariencia tiene la vista de **informes** después de agregar un puñado de visualizaciones y algunas páginas de informe nuevas. Para más información sobre los informes, consulte [Vista de informes en Power BI Desktop](#).

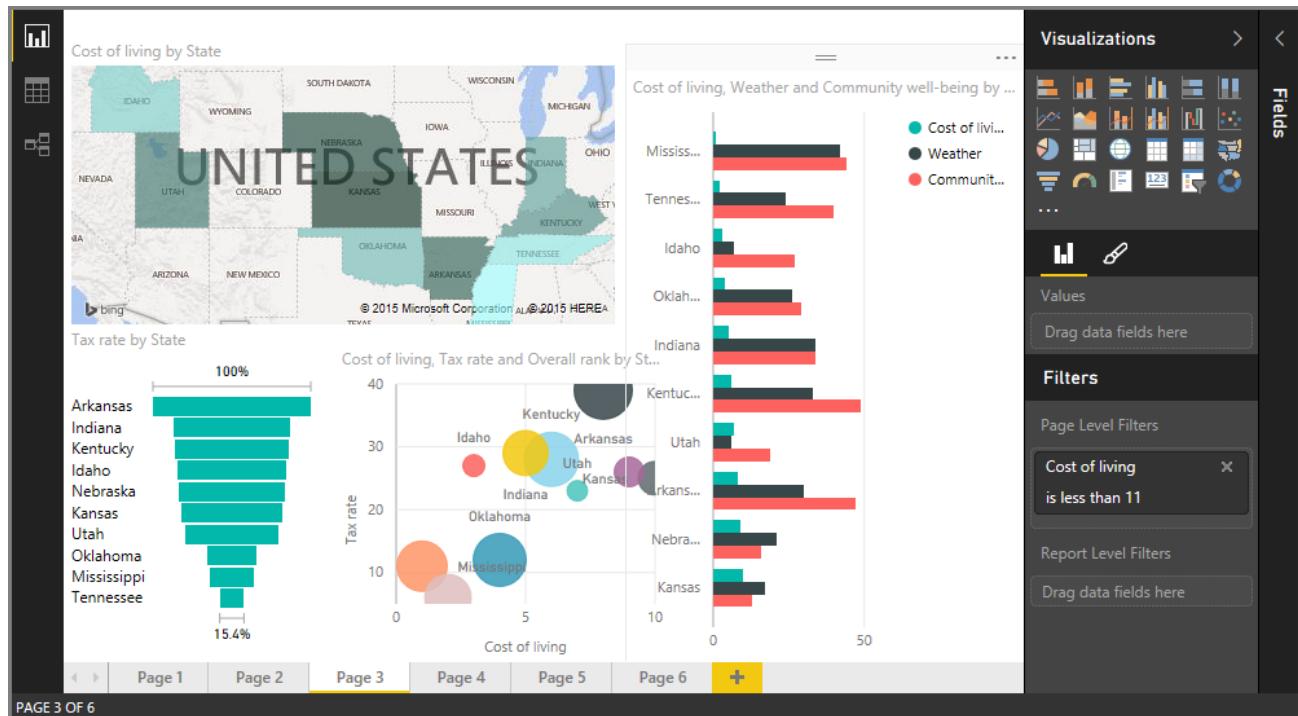
La primera página del informe proporciona una perspectiva de los datos basados en la *Clasificación general*. Cuando se selecciona una de las visualizaciones, el panel **Campos y filtros** muestra los campos seleccionados y la estructura de la visualización (qué campos se aplicaron al **Eje compartido**, los **Valores de columnas** y los **Valores de líneas**).



Hay seis páginas en este informe y en cada una se muestran determinados elementos de los datos.

1. En la primera página, que se muestra anteriormente, se muestran todos los estados según la *Clasificación general*.
2. La segunda página se centra en los 10 principales estados según la *Clasificación general*.
3. En la tercera página se visualizan los 10 mejores estados según el coste de la vida (y los datos asociados).
4. En la cuarta página los datos se abordan desde una perspectiva climática, filtrándose por los 15 estados más soleados.
5. En la quinta página, se muestran y representan gráficamente los 15 estados con mayor bienestar social.
6. Por último, se muestran las estadísticas sobre la delincuencia, que muestran los mejores (y bueno, los últimos) 10 estados.

Este es la apariencia de la página del informe que se centra en el coste de la vida.

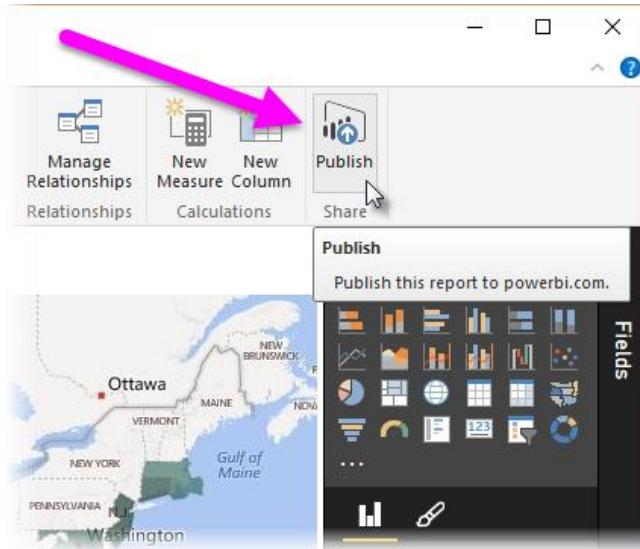


Hay todo tipo de informes y visualizaciones interesantes que puede crear.

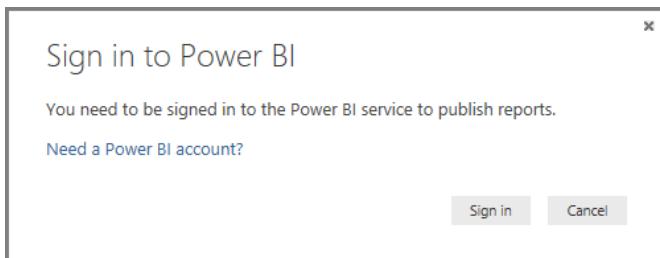
Compartir su trabajo

Ahora que tiene un informe de Power BI Desktop razonablemente completo, puede compartirlo con otras personas en el servicio **Power BI**. Hay varias maneras de compartir su trabajo en Power BI Desktop. Puede publicar en el servicio **Power BI**, puede cargar el archivo .pbix directamente desde el servicio Power BI o puede guardar el archivo .pbix y enviar como cualquier otro archivo.

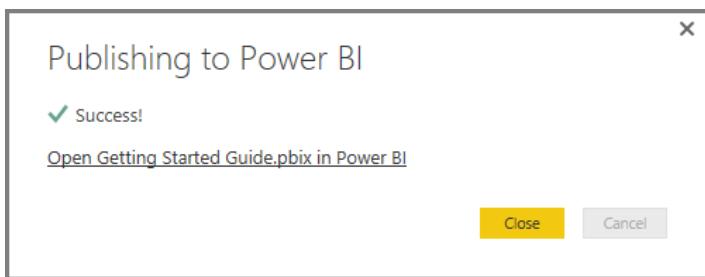
Primero, echemos un vistazo a la publicación en el servicio **Power BI** directamente desde Power BI Desktop. En la cinta de opciones **Inicio**, seleccione **Publicar**.



Se le pedirá que inicie sesión en Power BI.



Cuando haya iniciado sesión y finalizado el proceso de publicación, se mostrará el siguiente cuadro de diálogo.

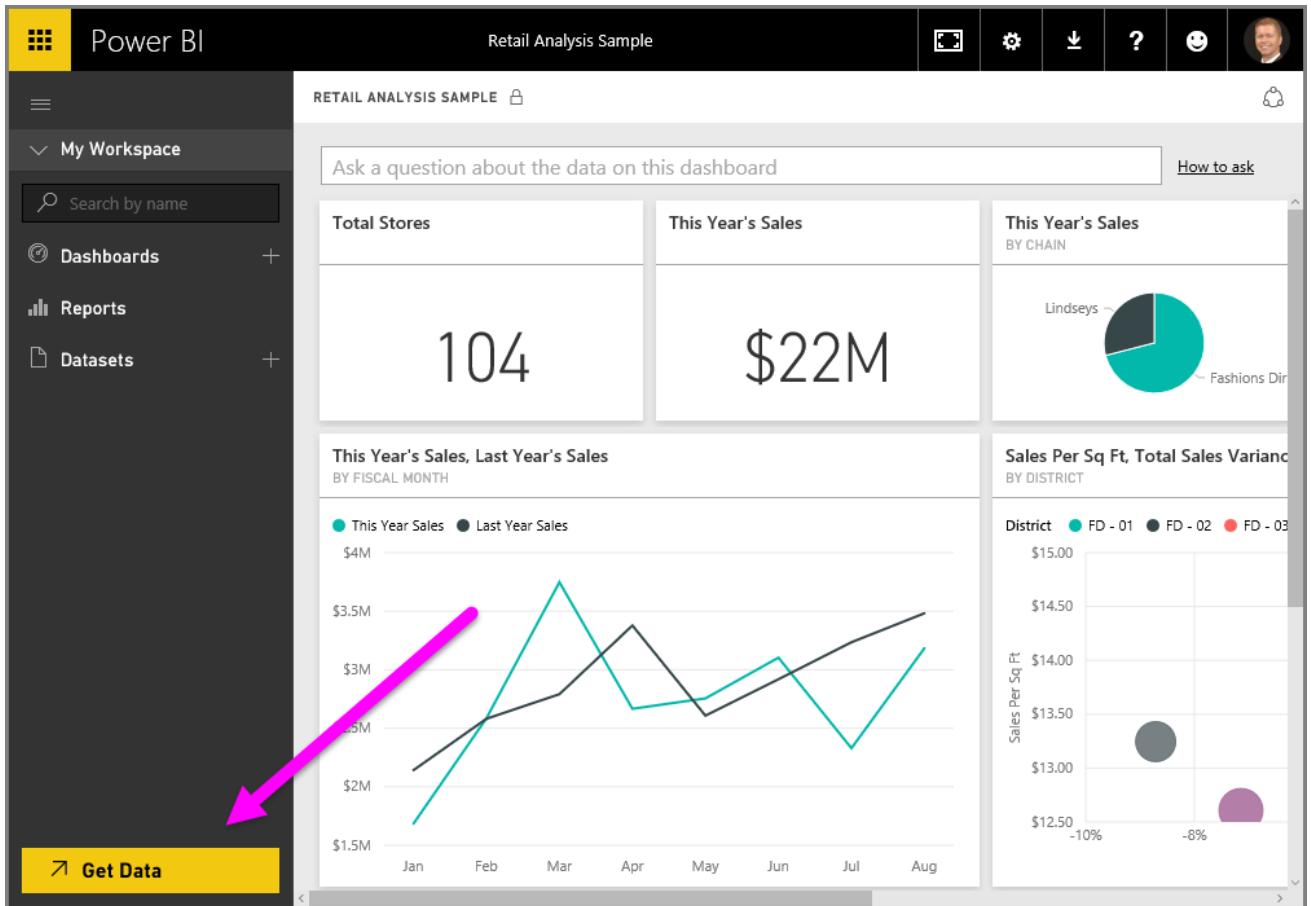


Cuando inicia sesión en Power BI, verá el archivo de Power BI Desktop que acaba de cargar en las secciones **Paneles**, **Informes** y **Conjuntos de datos** del servicio.

Otra manera de compartir su trabajo es realizar la carga desde el servicio **Power BI**. El siguiente vínculo abre el servicio **Power BI** en un explorador:

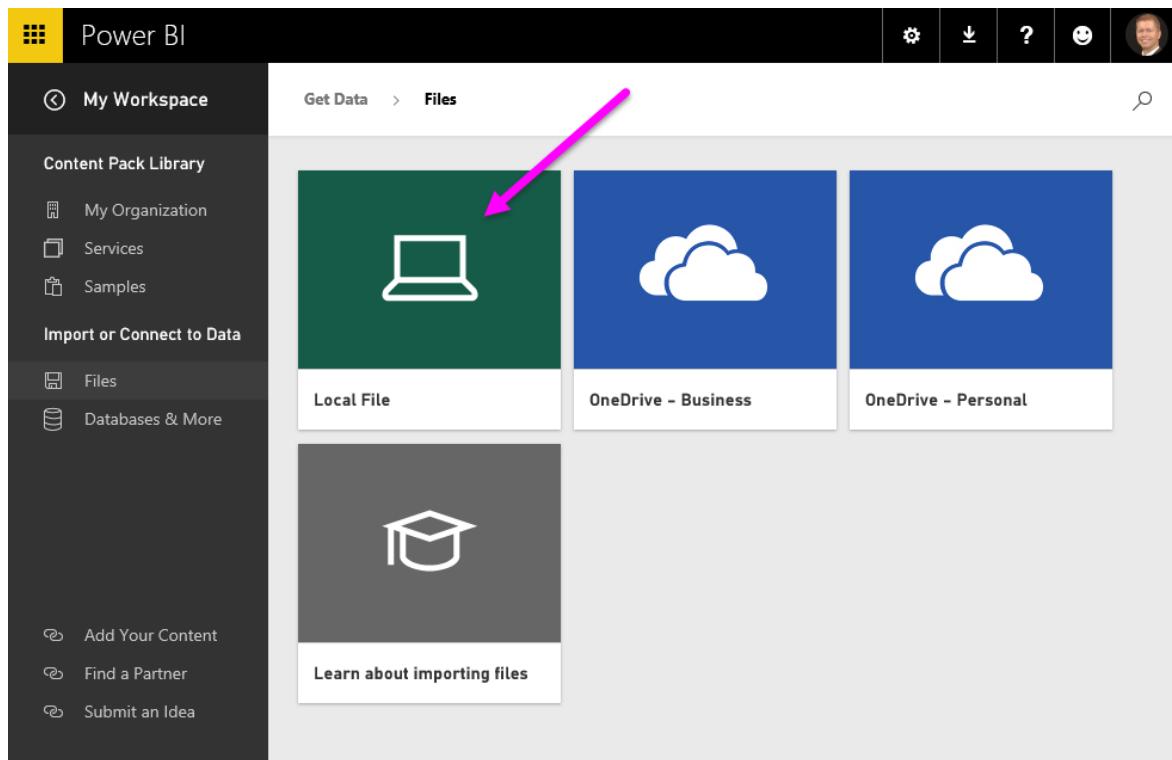
<https://app.powerbi.com>

Seleccione **Obtener datos** para iniciar el proceso de carga del informe de Power BI Desktop.



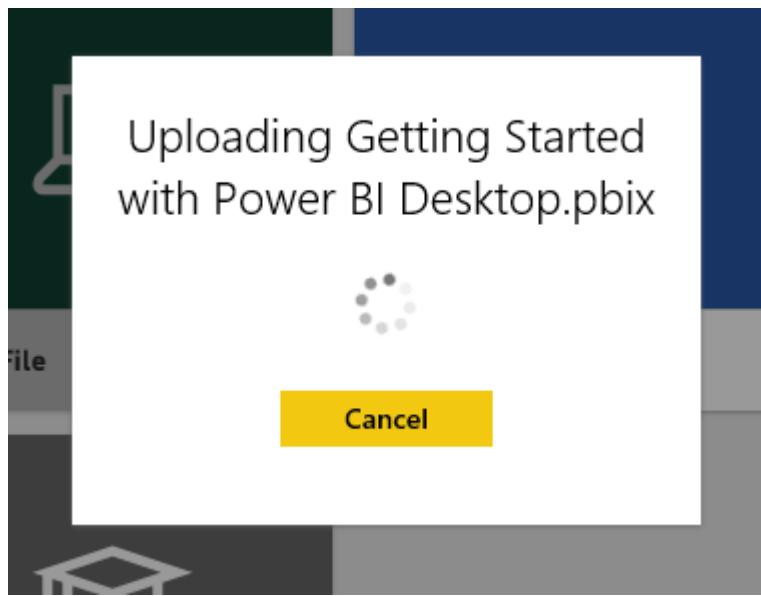
Aparece la página **Obtener datos**, donde puede seleccionar de dónde desea obtener los datos. En este caso, seleccionamos **Obtener** desde el cuadro **Archivos**.

Se muestra la vista **Archivos**. En este caso, seleccionamos **Archivo local**.

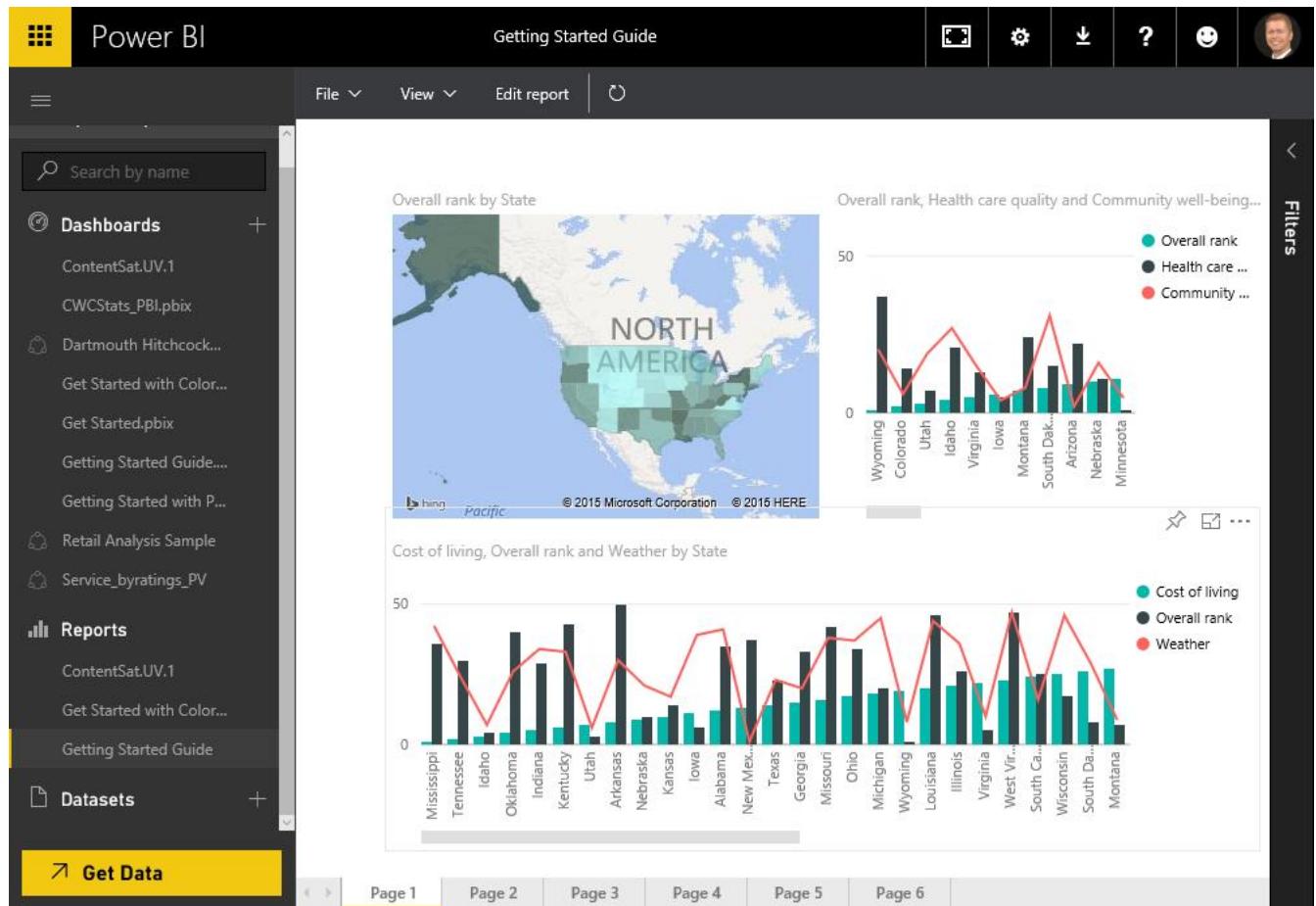


The screenshot shows the 'Get Data' section of the Power BI interface. At the top, it says 'Get Data > Files'. Below this, there are three main options: 'Local File' (represented by a laptop icon), 'OneDrive – Business' (represented by a cloud icon), and 'OneDrive – Personal' (also represented by a cloud icon). A pink arrow points to the 'Local File' icon. To the right of the icons, there is a search bar and a user profile icon.

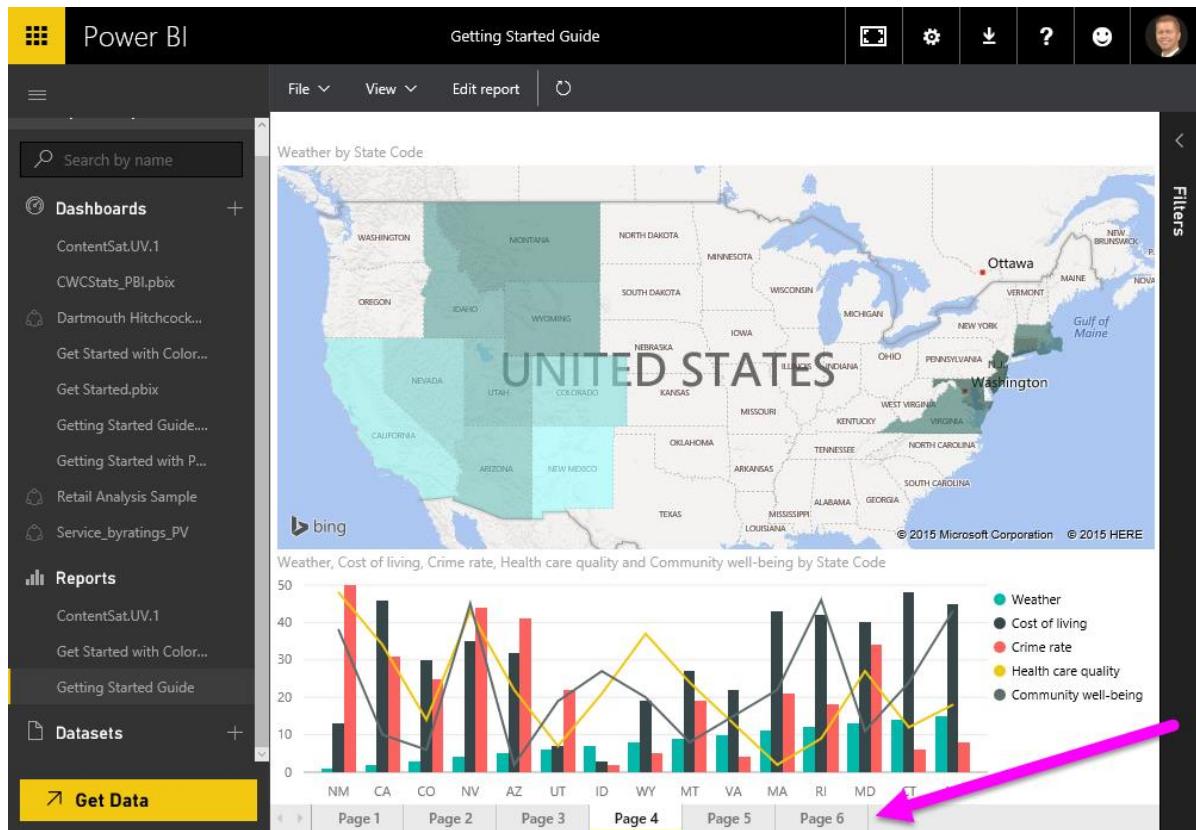
Cuando se selecciona el archivo, Power BI carga el archivo.



Una vez cargado el archivo, puede seleccionarlo en **Informes** en el panel izquierdo del servicio Power BI.

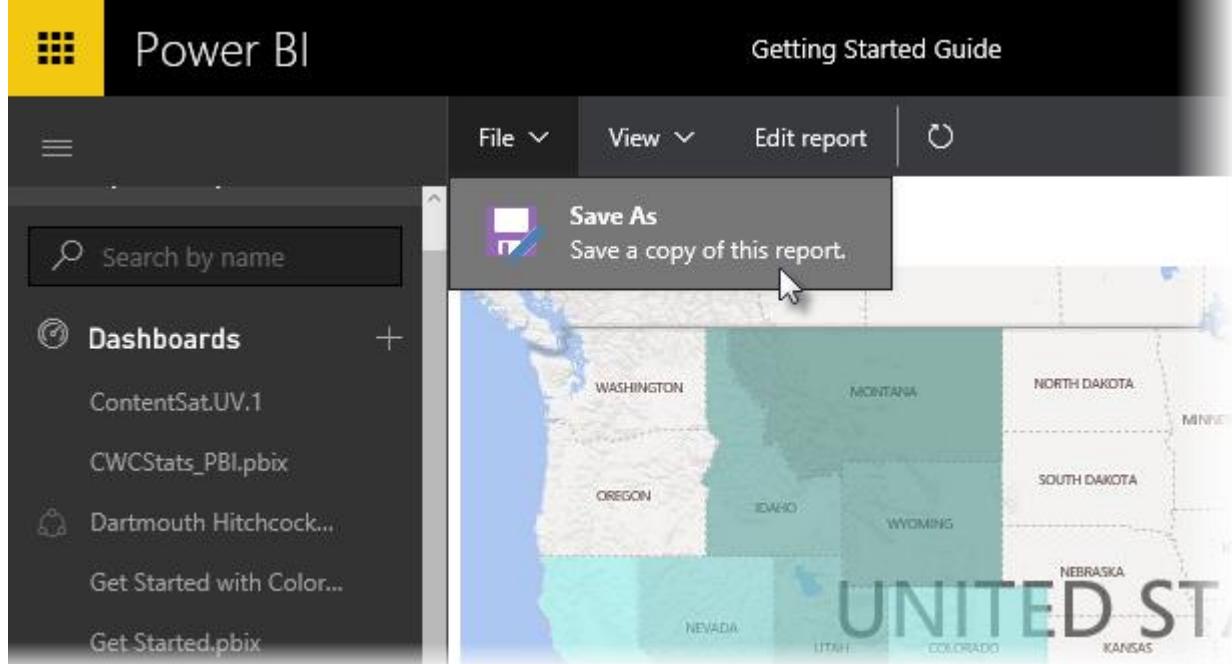


El servicio **Power BI** muestra la primera página del informe. En la parte inferior de la página, puede seleccionar cualquier pestaña para mostrar esa página del informe.

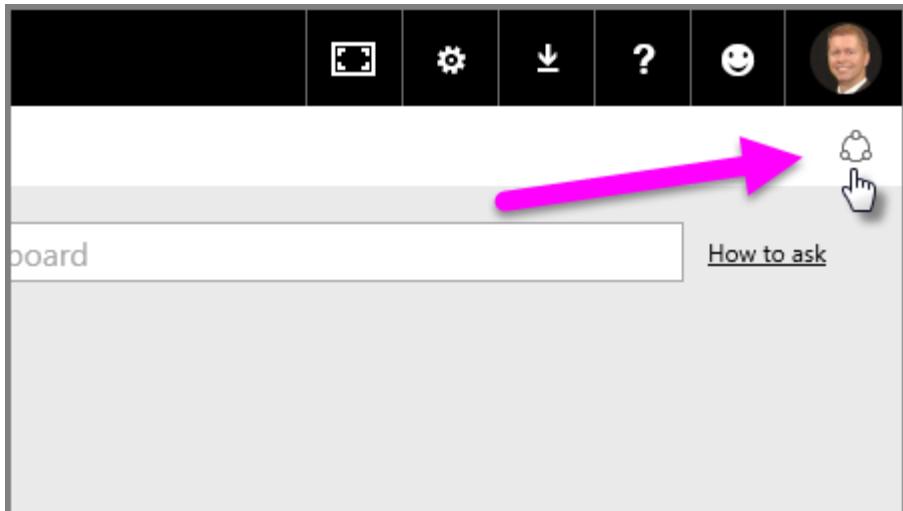


Puede realizar cambios en un informe del servicio **Power BI** seleccionando **Editar informe** desde la parte superior del lienzo de informe.

Para guardar el informe, seleccione **Archivo > Guardar como** desde el servicio. Puede crear todo tipo de efectos visuales interesantes en el servicio **Power BI** de su informe para anclarlos a un *panel*. Para más información sobre los paneles del servicio **Power BI**



Una vez guardado, seleccione el icono **Compartir** de la página principal.



Desde aquí, puede enviar un correo electrónico a los colegas con quienes desea compartir el panel.

A screenshot of the "Share dashboard" dialog in the Power BI web interface. The title bar says "Power BI" and "Get Started.pbix". The main area has a heading "Share dashboard" and a message "Not shared with anyone". It shows two tabs: "Invite" (which is selected) and "Shared with". A yellow banner at the top states: "This dashboard contains Power BI Pro content. Only users that have Power BI Pro will have access to it. [Learn more](#)". Below this is a text input field labeled "Enter email addresses". A descriptive text box says "Check out this dashboard I created to track our data." At the bottom, there is a note "Recipients will have access to the same data and reports as you have in this dashboard. [Learn more](#)" and a checked checkbox "Allow recipients to share your dashboard".

KPI en Power BI

Un indicador clave de rendimiento (KPI) es una indicación visual que comunica el progreso realizado para lograr un objetivo cuantificable. Para más información acerca de los KPI, consulte [Microsoft Developer Network](#).

Cuándo usar un KPI

Los KPI son una excelente opción:

- para medir el progreso (¿voy adelantado o retrasado?)
- para medir la distancia hasta un objetivo (¿cuánto adelanto o retraso tengo?)

Requisitos de los objetos visuales de KPI

Un indicador clave de rendimiento (KPI) se basa en una medida específica y está diseñado para ayudarle a evaluar el valor y el estado actuales de una métrica con respecto al objetivo definido. Por lo tanto, un objeto visual KPI requiere una medida *base* que se evalúa en un valor y una medida o un valor de *destino*, y un umbral u objetivo.

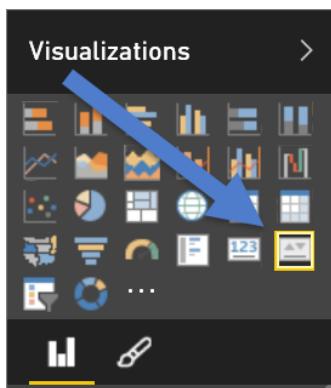
NOTA: Actualmente, un conjunto de datos de KPI debe contener los valores objetivo de un KPI. Esto puede hacerse agregando una hoja de Excel con los objetivos al modelo de datos o el archivo PBIX.

Cómo crear un KPI

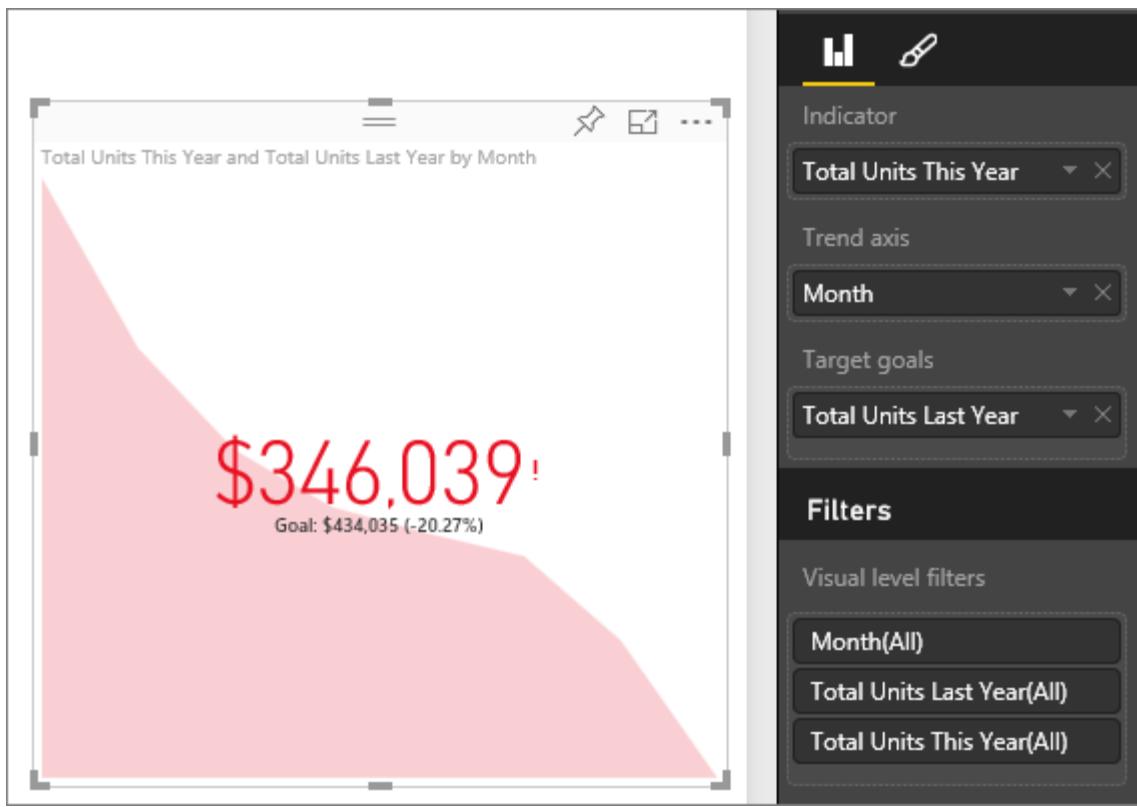
Para continuar, inicie sesión en Power BI y seleccione **Obtener datos > Ejemplos > Ejemplo de análisis de minoristas**. Vamos a crear un KPI que mide el progreso realizado para lograr un objetivo de ventas.

También puede ver otro en el que Will muestra cómo crear objetos visuales de métricas individuales: medidores, tarjetas y KPI.

1. Seleccione **Ventas > Unidades totales de este año**. Este será el indicador.
2. Agregue **Tiempo > Mes**. Esto representará la tendencia.
3. Para convertir el objeto visual en un KPI, seleccione el icono KPI en el panel Visualización.



4. Agregue un objetivo. Agregue los últimos años de ventas como objetivo. Arrastre **Unidades totales del último año** al campo **Objetivos de destino**.



5. Para dar formato al KPI, seleccione el icono de pincel para abrir el panel Formato.
- **Indicador**: controla las unidades de visualización y los decimales del indicador.
 - **Eje de tendencia**: cuando se establece en **Activado**, el eje de tendencia se muestra como el fondo del objeto visual de KPI.
 - **Objetivos**: cuando se establece en **Activado**, el objeto visual muestra el objetivo y la distancia desde el objetivo como un porcentaje.
 - **Estado**: algunos KPI se consideran *mejores* para valores mayores y otros son *mejores* con valores menores. Por ejemplo, ganancias frente a tiempo de espera. Normalmente, un mayor valor de ganancias es mejor que un mayor valor de tiempo de espera, que normalmente se considera como algo peor. Esta opción permite seleccionar el comportamiento del KPI. El estado predeterminado es **alto es mejor**.

6. Cuando tenga el KPI tal y como quiera, ánclelo a un panel.

Los KPI también están disponibles en los dispositivos móviles: manténgase al día del progreso de sus negocios.

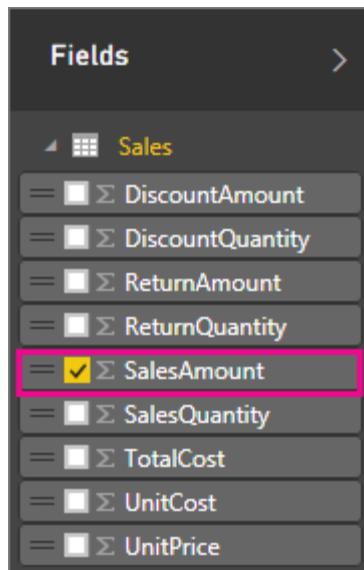
Crear medidas propias en Power BI Desktop

Algunas de las soluciones de análisis de datos más eficaces en el Power BI Desktop se pueden crear mediante el uso de medidas. ¿Cómo nos ayudan las medidas? Realizando cálculos con nuestros datos a medida que interactuamos con nuestros informes.

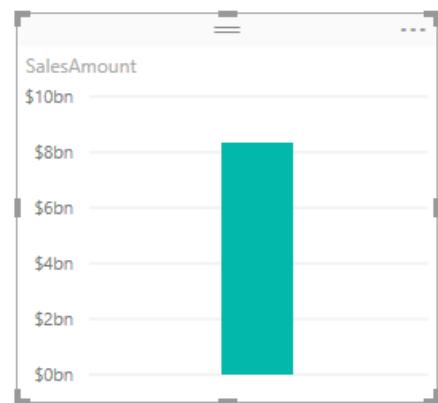
Nota: descargar el archivo de [ventas de muestra de Contoso para Power BI Desktop](#). Ya incluye datos de ventas en línea de la compañía ficticia Contoso, Inc. Dado que los datos en el archivo se importaron desde una base de datos, no podrá conectarse al origen de datos ni verlos en el editor de consultas. Cuando tenga el archivo en su equipo, abra Power BI Desktop.

¿De qué se tratan estas medidas?

En su mayoría, las medidas se crean automáticamente, como cuando se selecciona la casilla junto al campo **SalesAmount** desde la tabla **Sales** en la lista de campos, o bien al arrastrar **SalesAmount** al lienzo del informe.



Aparece una nueva visualización de gráfico, similar a la siguiente:



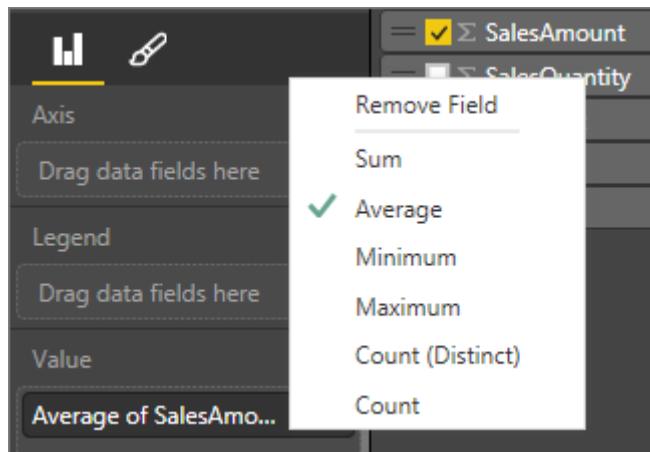
Lo que obtenemos es un gráfico de columnas que muestra el importe de la suma total de los valores del campo SalesAmount. El campo SalesAmount es simplemente una columna denominada SalesAmount en la tabla Ventas que ya se importó a través del editor de consultas.

La columna SalesAmount contiene más de dos millones de filas de valores de ventas. Tal vez se pregunte por qué no ve una tabla con filas de todos esos valores. Bueno, Power BI Desktop detecta que todos los valores de SalesAmount son de un tipo de datos numérico, y que probablemente deseará agregarlos de alguna manera, ya sea para sumarlos, obtener promedios, hacer recuento, etc.

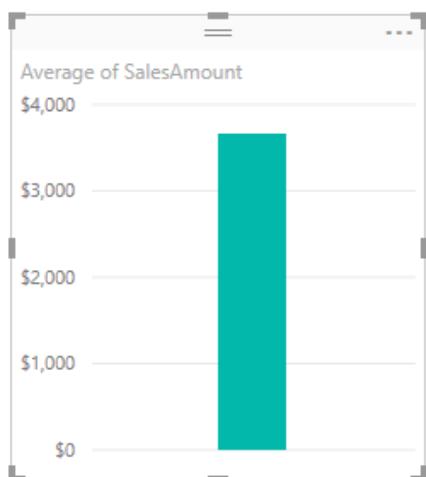
Cada vez vea un campo en la lista de campos con un icono de sigma Σ , significa que el campo es numérico y se pueden agregar sus valores. En este caso, cuando se selecciona SalesAmount, Power BI Desktop crea sus propias medidas y la suma de todos los importes de ventas se calcula y se muestra en el gráfico.

La suma es la agregación predeterminada cuando se selecciona un campo con un tipo de datos numérico, pero se puede cambiar fácilmente a otro tipo de agregación.

En el área **Valor**, si hacemos clic en la flecha abajo junto a **SalesAmount**, a continuación, podremos seleccionar **Media**.



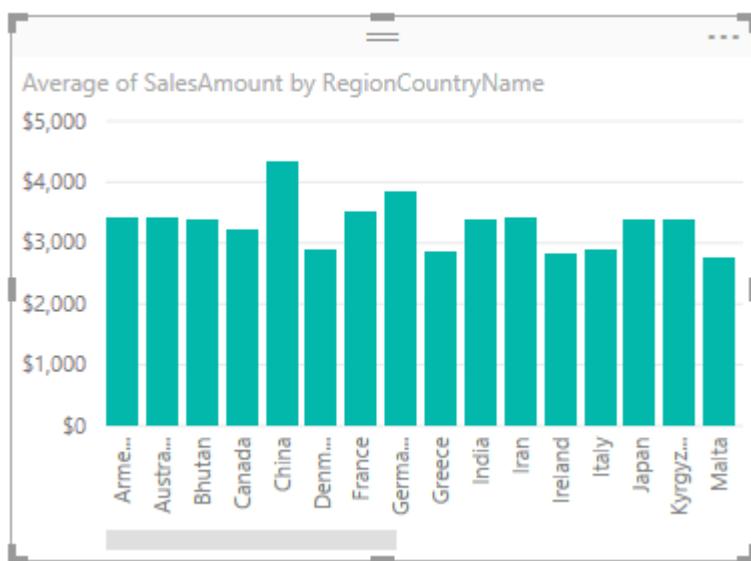
Nuestra visualización cambia a un promedio de todos los valores de ventas en el campo SalesAmount.



Se puede cambiar el tipo de agregación según el resultado que deseamos, pero no todos los tipos de agregación se pueden aplicar a cualquier tipo de datos numérico. Por ejemplo, para el campo SalesAmount, tiene sentido sumar y obtener la media. De igual manera, se pueden usar las funciones de mínimo y máximo. Pero no tiene mucho sentido usar la función de recuento para el campo SalesAmount porque aunque sus valores son numéricos, se trata de valores de moneda.

Comprender las agregaciones es fundamental para entender las medidas, porque cada medida realiza algún tipo de agregación. Veremos más ejemplos del uso de la agregación de suma más adelante, cuando cree algunas medidas propias.

Los valores que se calculan a partir de medidas cambian constantemente en respuesta a interacciones con el informe. Por ejemplo, si se arrastra el campo **RegionCountryName** de la tabla **Geography** a nuestro gráfico, se promedian y se muestran los importes de ventas de cada país.



Cuando el resultado de una medida cambia por una interacción con el informe, se afecta el *contexto* de la medida. De hecho, cada vez que interactúa con el informe, cambia el contexto en el que una medida calcula y muestra los resultados.

En la mayoría de los casos, el diseñador hace lo suyo y calcula y devuelve los valores según los campos que se agregaron y los tipos de agregación que se eligen. Pero en otros casos, tendrá que crear sus propias medidas para realizar cálculos más complejos y únicos.

Con Power BI Desktop, creará sus propias medidas usando el lenguaje de fórmulas de expresiones de análisis de datos (DAX). Las fórmulas DAX son muy similares a las fórmulas de Excel. De hecho, DAX usa muchas de las mismas funciones, operadores y sintaxis que las fórmulas de Excel. Sin embargo, las funciones DAX están diseñadas para trabajar con datos relacionales y realizar cálculos más dinámicos a medida que interactuamos con los informes.

Hay más de 200 funciones DAX que realizan todo tipo de agregaciones simples, como sumas y medias, hasta funciones estadísticas y de filtrado más complejas. No entraremos aquí en muchos detalles sobre el lenguaje DAX, pero hay muchos recursos que le ayudarán a obtener más información. Cuando creamos nuestras propias medidas, estas

se agregan a la lista de campos de la tabla que queremos. Esto se conoce como una medida *explícita* y permanecerá en la tabla como un campo. Las principales ventajas de las medidas explícitas incluyen que se les puede nombrar según se prefiera, lo que facilita su identificación. También se pueden usar como argumentos en otras expresiones DAX y crear medidas para realizar cálculos complejos muy rápidamente.

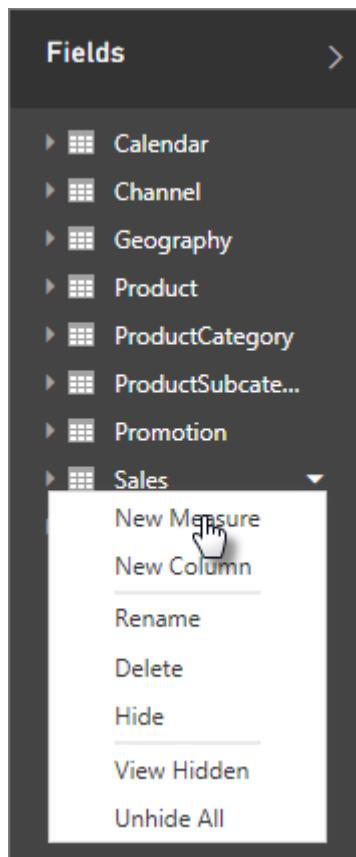
Vamos a crear una medida

Supongamos que queremos analizar nuestras ventas netas. Si revisamos la tabla Sales en la lista de campos, vemos que no hay ningún campo denominado NetSales. Pero tenemos bloques de creación para crear nuestra propia medida para calcular las ventas netas.

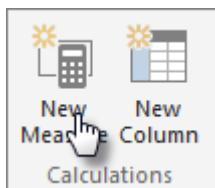
Se necesita una medida para restar los descuentos y devoluciones de los importes de las ventas. Dado que deseamos que nuestra medida calcule un resultado para el contexto que tengamos en nuestra visualización, en efecto, tenemos que restar la suma de DiscountAmount y ReturnAmount de la suma de SalesAmount. Puede parecer un poco confuso en ese momento; no se preocupe, le resultará más claro dentro de poco.

Ventas netas

1. En la lista de campos, haga clic con el botón secundario en la tabla **Sales**, o en la flecha abajo de la tabla, y a continuación haga clic en **Nueva medida**. Esto garantiza que la nueva medida se guarda en la tabla Sales, donde será más fácil encontrar.



Sugerencia : también puede crear una nueva medida haciendo clic en el botón Nueva medida en la cinta de la pestaña Inicio de Power BI Desktop.



Cuando se crea una medida desde la cinta de opciones, la medida se puede crear en cualquiera de las tablas. Aunque no es requisito que una medida pertenezca a una tabla determinada, será más fácil de encontrar si se crea en la tabla que le resulte más lógica. Si desea que pertenezca a una tabla determinada, en primer lugar, haga clic en la tabla para activarla. A continuación, haga clic en Nueva medida. En nuestro caso, vamos a crear nuestra primera medida en la tabla Sales.

La barra de fórmulas aparece en la parte superior del lienzo del informe. Esto es donde podemos cambiar el nombre de la medida y escribir una fórmula DAX.

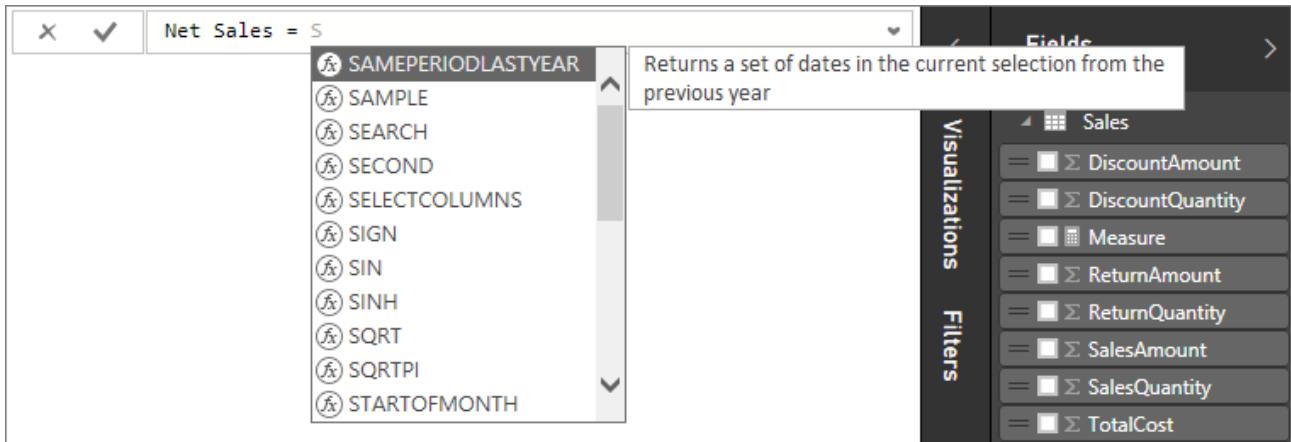
Measure
= SUM(DiscOUNTAmount)
= SUM(DiscOUNTQuantity)
= SUM(Measure)
= SUM(ReturnAmount)
= SUM(ReturnQuantity)
= SUM(SalesAmount)
= SUM(SalesQuantity)
= SUM(TotalCost)
= SUM(UnitCost)
= SUM(UnitPrice)

Elegiremos un nombre para la nueva medida. De forma predeterminada una nueva medida simplemente se denomina Medida. Si no se cambia el nombre, cuando se crea otra, se denominará Medida 2, Medida 3 y así sucesivamente. Queremos que nuestras medidas sean más fáciles de identificar, así que nombraremos esta nueva medida Ventas netas.

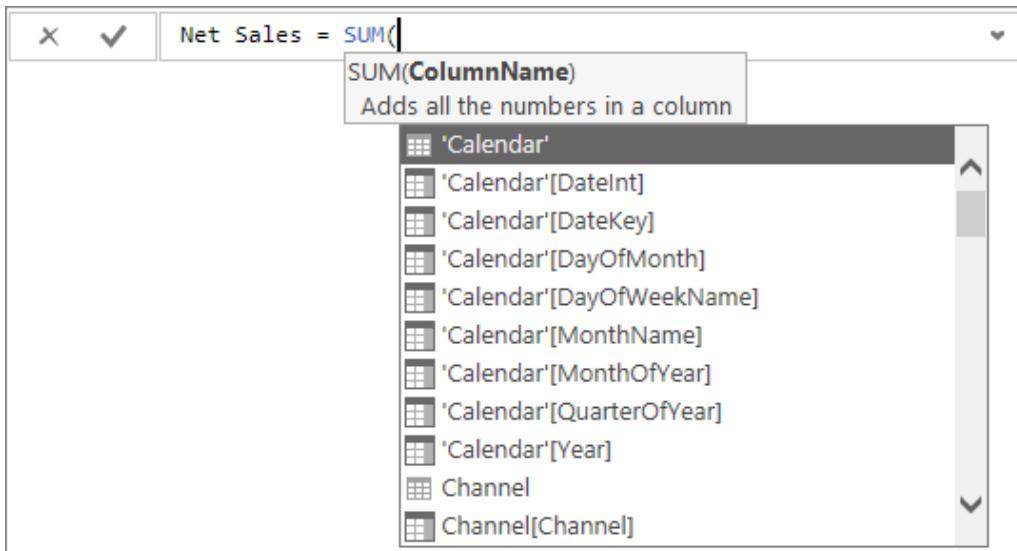
2. **Resalte Medida** en la barra de fórmulas y, después, escriba **Ventas netas**.

Ahora podemos comenzar por introducir la fórmula.

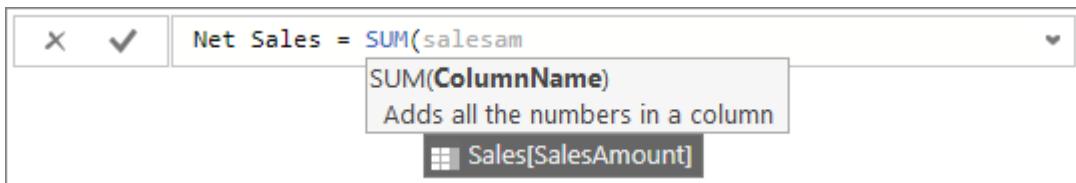
3. Después del signo igual escriba una **S**. Aparecerá una lista desplegable de sugerencias con todas las funciones DAX que comienzan con la letra S. Cuantas más letras se añaden, más se escala la lista de sugerencias acercándose a la función que necesitamos. Desplace hacia abajo y seleccione **SUM** y, a continuación, presione Entrar.



Después de presionar ENTRAR, aparece un paréntesis de apertura junto con otra lista de sugerencias de todas las columnas disponibles para pasar a la función SUM.



Una expresión siempre aparece entre paréntesis de apertura y cierre. En este caso, la expresión contendrá un único argumento para pasar a la función SUM: una columna para sumar. Para reducir la lista de columnas, escriba las primeras letras de lo que desea. En este caso, deseamos la columna SalesAmount, por lo que cuando empezamos a escribir salesam, la lista se hace más pequeña y nos muestra dos elementos que podemos seleccionar. En realidad, se trata de la misma columna. Un elemento muestra solamente [SalesAmount] porque estamos creando la medida en la misma tabla donde se encuentra la columna SalesAmount. En el otro vemos el nombre de tabla precediendo al nombre de la columna.

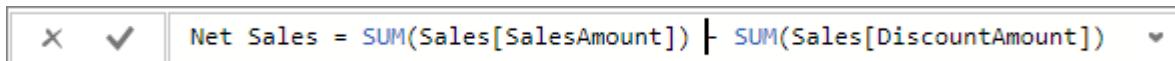


En general, es recomendable que escriba el nombre de la columna completo. Facilitará la lectura de las fórmulas.

4. ** Seleccione **Sales[SalesAmount], después, escriba un paréntesis de cierre.
->**Sugerencia:** los errores de sintaxis suelen deberse a paréntesis de cierre faltantes o en el lugar incorrecto.

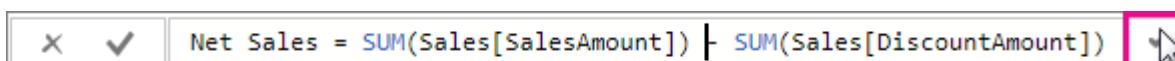
Ahora queremos restar las otras dos columnas.

5. Después del paréntesis de cierre de la primera expresión, escriba un espacio y, a continuación, un operador de signo menos (-), seguido de otro espacio. Luego escriba otra función SUM con la columna **Sales[DiscountAmount]** como su argumento.

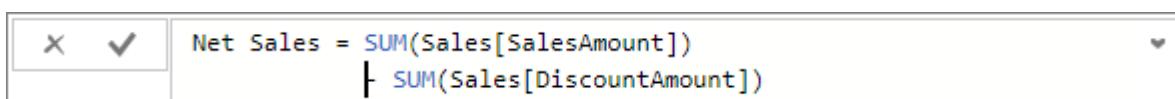


Comienza a faltar espacio para la fórmula. No hay problema.

6. Haga clic en el botón de contenido adicional hacia abajo situado en el lado derecho de la barra de fórmulas.

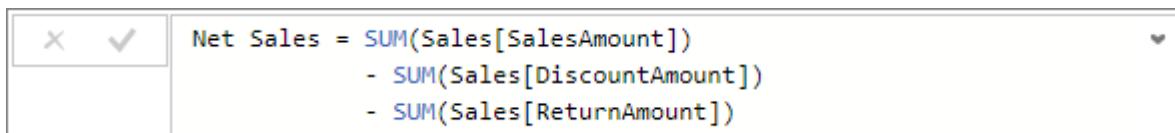


Ahora hay más espacio. Para agregar nuevos elementos a la fórmula en una nueva línea, presione ALT+ENTRAR. También podemos mover elementos con la tecla Tab.



Ya podemos agregar la parte final de la fórmula.

7. Agregue otro operador de signo menos seguido de otra función SUM y la columna **Sales[ReturnAmount]** como su argumento.



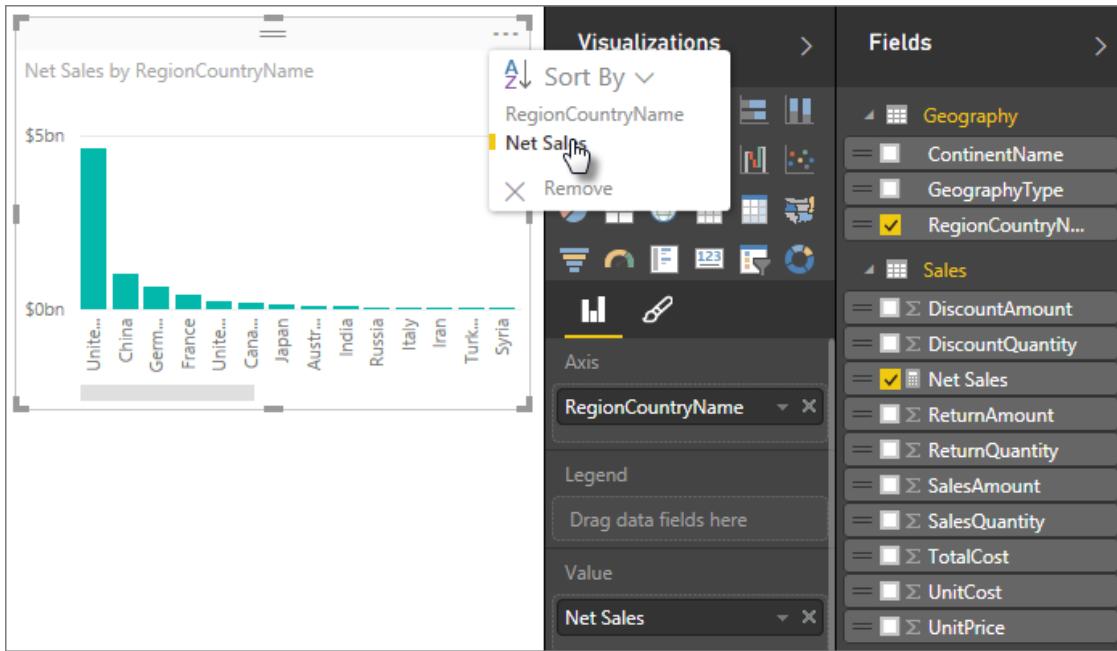
Parece que la fórmula está lista.

8. Para terminar, presione ENTRAR o haga clic en la marca de verificación en la barra de fórmulas. La fórmula se valida y se agrega a la lista de campos en la tabla Sales.

Agreguemos la nueva medida a un informe

Ahora podemos agregar la medida Net Sales al lienzo del informe y las ventas netas se calcularán para los demás campos que se agreguen al informe. Echemos un vistazo a las ventas netas por país.

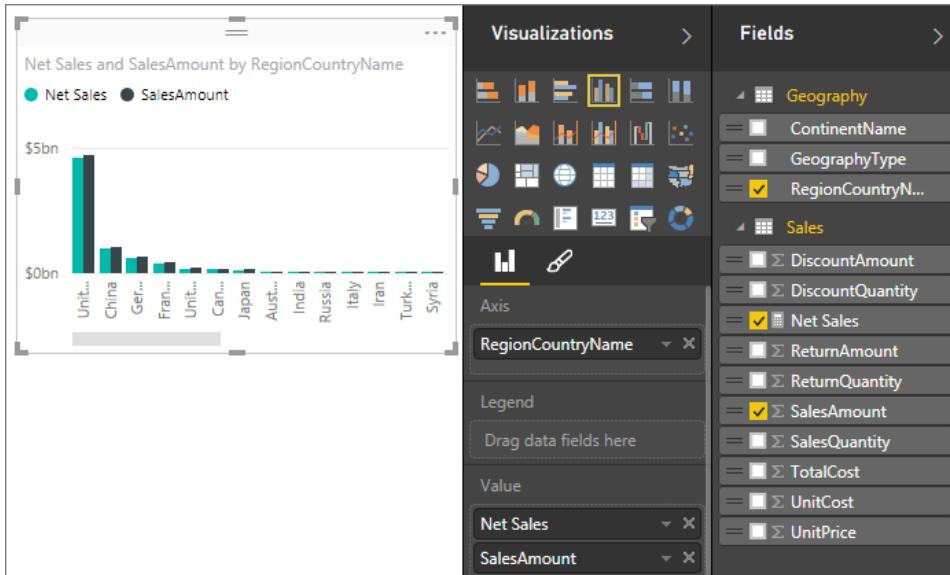
1. Arrastre la medida **Net Sales** desde la tabla **Sales** al lienzo del informe.
2. ** Ahora, arrastre el campo **RegionCountryName desde la tabla **Geography** hasta el gráfico.



Agreguemos algunos datos más.

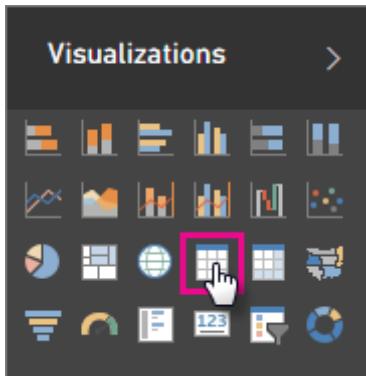
3. Arrastre el campo **SalesAmount** al gráfico para ver la diferencia entre las ventas netas y el importe de las ventas.

Ahora realmente tenemos dos medidas en el gráfico. SalesAmount, que se ha sumado automáticamente, y la medida de ventas netas que creamos. En cada caso, los resultados se calculan en el contexto de otro campo que tenemos en el gráfico, los países en RegionCountryName.

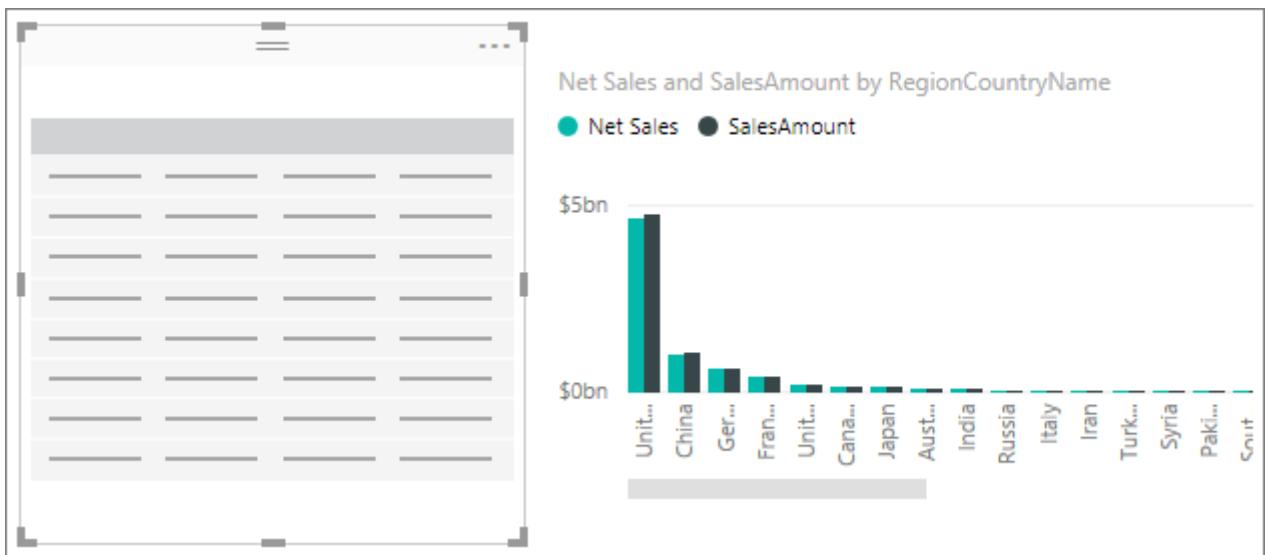


Ahora agreguemos una segmentación de datos, para que podamos desglosar aún más las ventas netas y los importes de ventas por año natural.

4. Haga clic en un área en blanco junto al gráfico y, en **Visualizaciones**, haga clic en la visualización de la tabla.



Esto crea una visualización de tabla en blanco en el lienzo Informe.

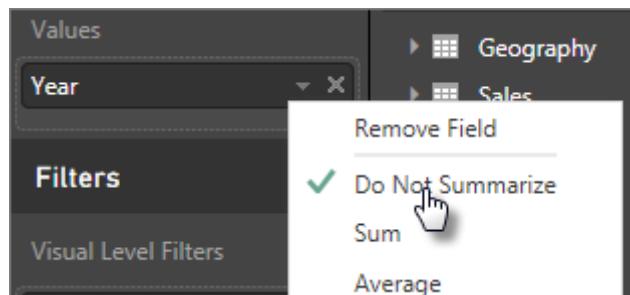


5. Arrastre el campo **Year** desde la tabla **Calendar** a la nueva tabla en blanco.

Year
10284801

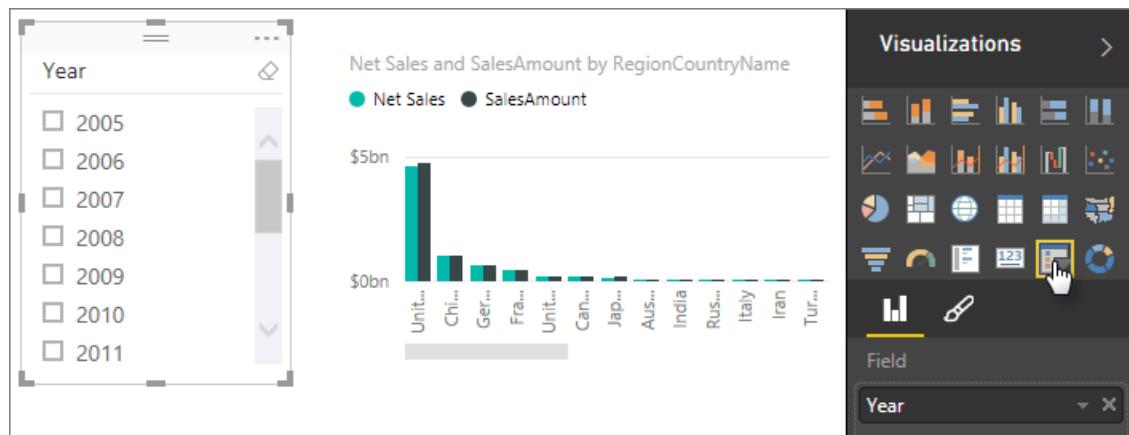
Como Year es un campo numérico, el diseñador sumó sus valores y nos dio un gráfico. Sin embargo, no será de tanta ayuda como una segmentación de datos.

6. ** En **Valores, haga clic en la flecha abajo junto a Año y, a continuación, haga clic en No resumir.



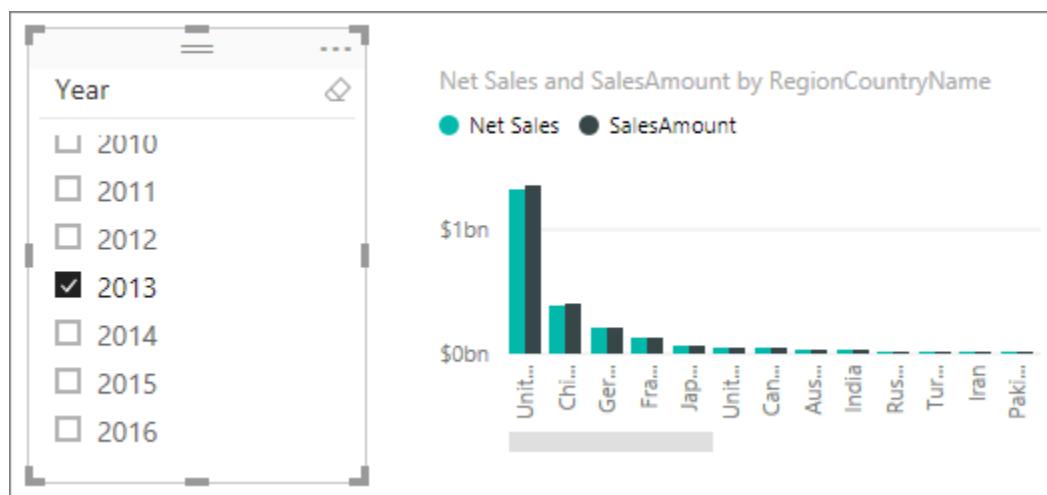
Ahora podemos cambiar el campo Año en la visualización de la tabla a una segmentación de datos.

7. En Visualizaciones, haga clic en la visualización Segmentación.



Ahora tenemos el año como una segmentación de datos. Podemos seleccionar cualquier año o grupo de años y las visualizaciones del informe se segmentarán en consecuencia.

8. ** Continúe y haga clic en **2013. Verá que el gráfico cambia. Las medidas Net Sales y SalesAmount se vuelven a calcular y se muestran nuevos resultados solo para el año 2013. Nuevamente cambiamos el contexto en el que las medidas se calculan y muestran resultados.



Vamos a crear otra medida

Ahora que sabe cómo crear sus propias medidas, vamos a crear otra.

Ventas netas por unidad

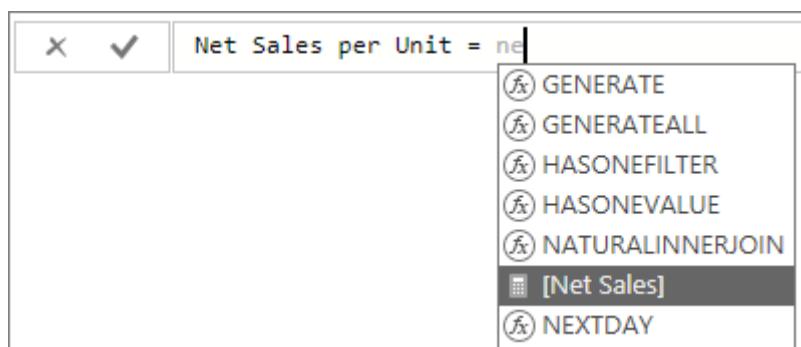
¿Qué pasa si queremos averiguar cuáles son los productos con más ventas por unidad?

Bueno, podemos crear otra medida. En este caso, queremos dividir las ventas netas por la cantidad de unidades vendidas. En efecto, se debe dividir el resultado de la medida Net Sales por la suma de Sales[SalesQuantity].

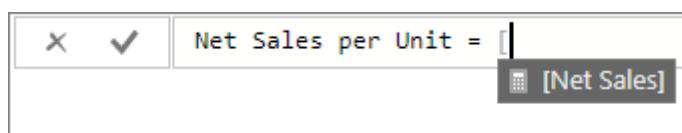
1. Cree una nueva medida denominada **Net Sales per Unit** en la tabla Sales o la tabla Productos.

En esta medida, vamos a usar la medida de ventas netas que se creó anteriormente. Con DAX, podemos hacer referencia a otras medidas en nuestra fórmula.

2. Comience a escribir **Net Sales**. La lista de sugerencias mostrará lo que podemos agregar. Seleccione **[Net Sales]**.



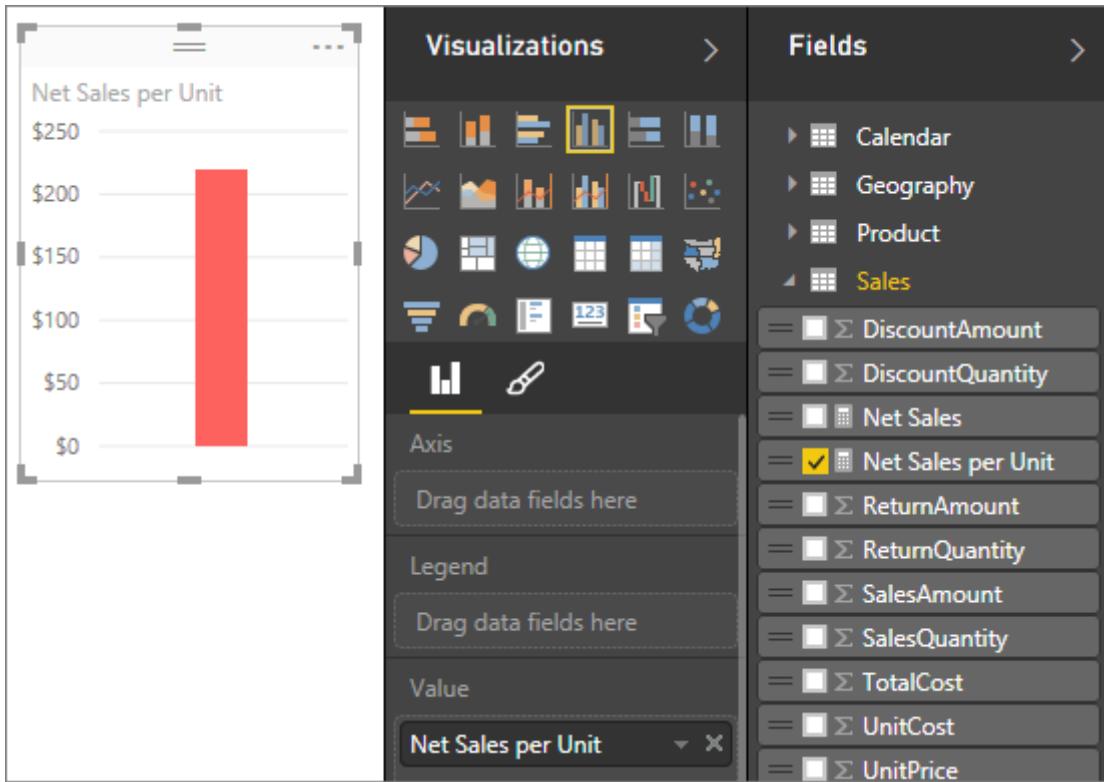
También puede hacer referencia a otra medida, para ello escriba un corchete de apertura (`()`). La lista de sugerencias mostrará únicamente las medidas que podemos agregar a nuestra fórmula.



3. Justo después de **[Net Sales]**, escriba un espacio y, a continuación, un operador de división (`/`), escriba una función SUM y a continuación escriba **Quantity**. La lista de sugerencias muestra todas las columnas con **Quantity** en el nombre. Seleccione **Sales[SalesQuantity]**. Ahora la fórmula debe tener el siguiente aspecto:
> **Net Sales per Unit = [Net Sales] / SUM(Sales[SalesQuantity])**

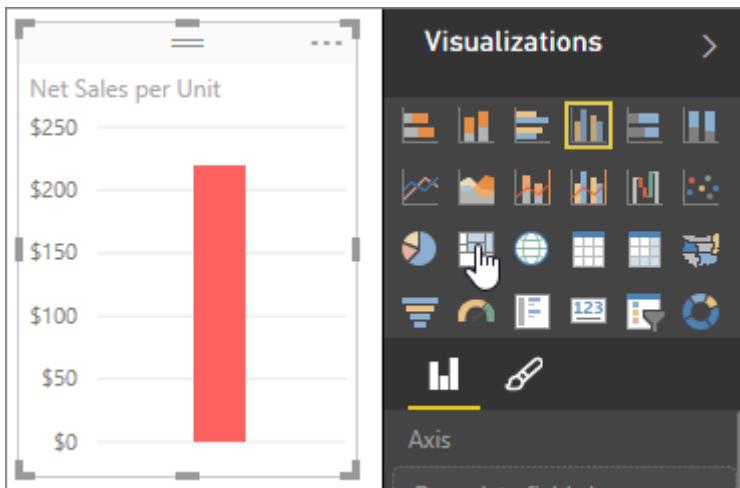
Queda bien, ¿verdad? Introducir fórmulas de DAX es realmente fácil cuando usamos la funcionalidad de búsqueda y sugerencia del Editor de DAX. Ahora, veamos lo que obtenemos con la nueva medida para cantidad las ventas netas por unidad.

4. ** Arrastre la medida ****Net Sales per Unit** a un área en blanco en el lienzo de informes.

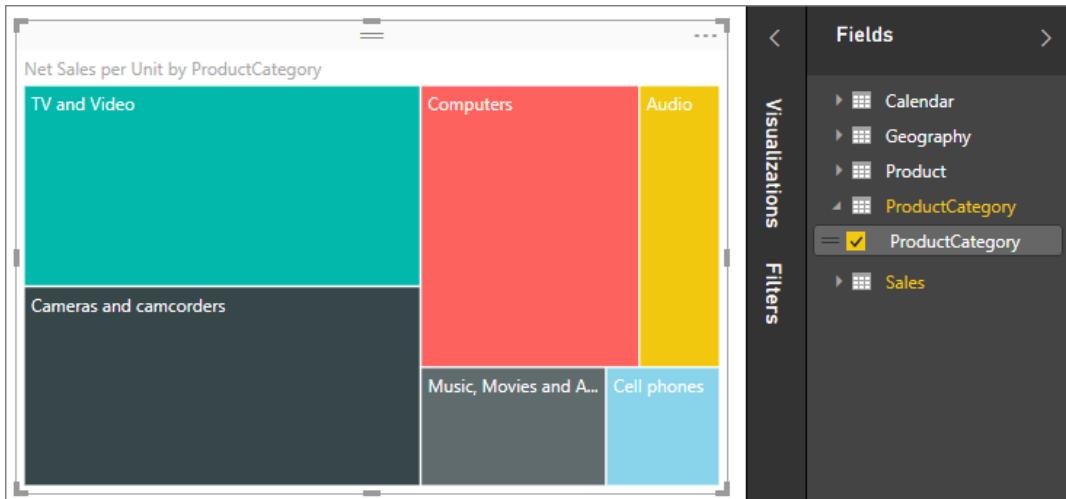


No muy interesante, ¿cierto? No se preocupe.

5. Cambie el tipo de visualización del gráfico a Mapa de árbol.

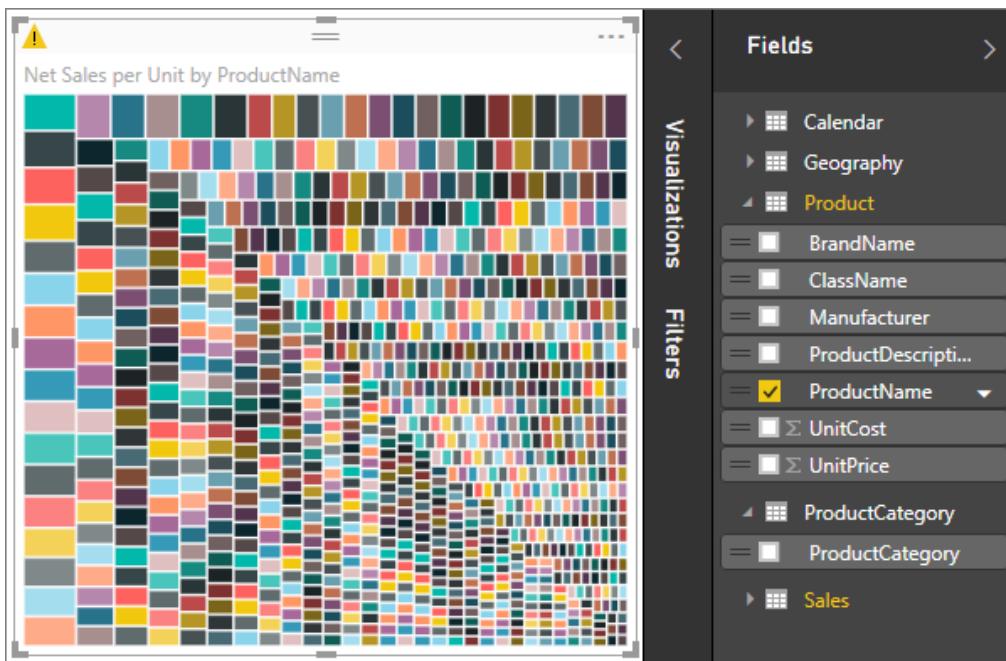


6. ** Ahora, arrastre el campo **ProductCategory desde la tabla ProductCategory hacia el área Grupo.



Esta información es útil, pero ¿qué pasa si queremos ver las ventas netas por producto?

7. ** Quite el campo **ProductCategory y en su lugar agregue el campo **ProductName** arrastrándolo desde la tabla **Products** hacia el área **Grupo**.



Ahora solo estamos jugando, pero tiene que admitir que esto es ¡simplemente genial! Por supuesto, podemos filtrar este mapa de árbol de cualquier forma, pero eso está fuera del ámbito de este tutorial.

Las medidas son de gran ayuda para obtener la información que busca de sus datos. Ahora sabe cómo crear medidas con la barra de fórmulas. Puede asignar a las medidas el nombre que le resulte más conveniente; con las listas de sugerencias es más fácil buscar y seleccionar el elemento correcto para agregar a las fórmulas. También aprendió un poco sobre el contexto, donde el resultado de los cálculos en las medidas cambia en función de otros campos o por otras expresiones en la fórmula de medida.

Aspectos básicos de DAX en Power BI Desktop

Este artículo está destinado a usuarios sin experiencia en Power BI Desktop. Su objetivo es proporcionar una introducción rápida y fácil al uso de expresiones de análisis de datos (DAX) para solucionar una variedad de problemas de análisis de datos y cálculo básico. Abordaremos información conceptual, una serie de tareas que puede completar y algunos cuestionarios para probar lo que ha aprendido. Al finalizar este artículo, debe tener una buena comprensión de los conceptos fundamentales más importantes en DAX.

¿Qué es DAX?

DAX es una colección de funciones, operadores y constantes que se pueden usar en una fórmula o expresión, para calcular y devolver uno o más valores. En pocas palabras, DAX le ayuda a crear información nueva a partir de datos ya incluidos en un modelo.

¿Por qué es tan importante DAX?

Crear un archivo de Power BI Desktop nuevo e importar algunos datos en él es bastante fácil. Incluso puede crear informes que muestren información valiosa sin usar las fórmulas DAX en absoluto. Pero, ¿qué ocurre si necesita analizar el porcentaje de crecimiento por categorías de producto y para intervalos de fechas diferentes? ¿O bien, si debe calcular el crecimiento interanual en comparación con las tendencias del mercado? Las fórmulas DAX proporcionan esta y muchas otras capacidades igual de importantes. Obtener información sobre cómo crear fórmulas DAX eficaces le ayudará a aprovechar los datos al máximo. Cuando obtiene la información que necesita, puede empezar a resolver problemas empresariales reales que afectan los resultados finales. Ese es el potencial de Power BI y DAX le ayuda a alcanzarlo.

Requisitos previos

Posiblemente ya esté familiarizado con la creación de fórmulas en Microsoft Excel. Esta información será útil para comprender DAX, pero incluso si no tiene ninguna experiencia con las fórmulas de Excel, los conceptos descritos aquí le ayudarán a empezar a crear fórmulas de DAX y solucionar problemas de inteligencia empresarial del mundo real inmediatamente.

Nos centraremos en comprender las fórmulas DAX que se usan en los cálculos, más específicamente, en columnas calculadas y medidas. Debe estar familiarizado con Power BI Desktop, la importación de datos y la incorporación de campos a un informe. También debe estar familiarizado con los conceptos fundamentales de [medidas](#) y [columnas calculadas](#).

Libro de trabajo de Excel

La mejor manera de aprender a usar DAX es crear algunas fórmulas básicas, usar las expresiones con algunos datos reales y comprobar los resultados con sus propios ojos. Los ejemplos y tareas que se exponen aquí utilizan el archivo de ventas de muestra de Contoso para Power BI Desktop Preview. Este es el mismo archivo de ejemplo usado en

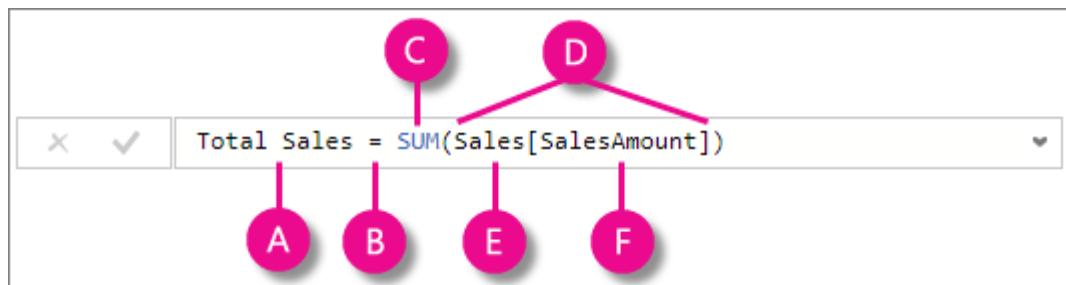
el artículo Tutorial: Crear sus propias medidas en Power BI Desktop. Puede descargarlo [aquí](#).

¡Comencemos!

Ceñiremos nuestro estudio de DAX en torno a tres conceptos fundamentales: *sintaxis*, *funciones**y* *contexto*. Por supuesto, hay otros conceptos importantes en DAX, pero la comprensión de estos tres conceptos le proporcionará una base óptima para desarrollar sus habilidades de DAX.

Sintaxis

Antes de crear sus propias fórmulas, echemos un vistazo a la sintaxis de las fórmulas DAX. La sintaxis incluye los distintos elementos que componen una fórmula, o en términos más simples, cómo se escribe la fórmula. Por ejemplo, echemos un vistazo a una medida de la fórmula DAX simple.



Esta fórmula incluye los siguientes elementos de sintaxis:

- A. El nombre de medida **Total Sales**.
- B. El operador signo igual (=) indica el principio de la fórmula. Al calcular, devolverá un resultado.
- C. La función DAX **SUM** suma todos los números en la columna **Sales[SalesAmount]**. Aprenderá más acerca de las funciones más adelante.
- D. Los paréntesis () envuelven una expresión que contiene uno o más argumentos. Todas las funciones requieren al menos un argumento. Un argumento pasa un valor a una función.
- E. La tabla referenciada **Sales**.
- F. La columna referenciada **[SalesAmount]** en la tabla Sales. Con este argumento, la función SUM detecta en qué columna se agrega una suma.

Al tratar de comprender una fórmula DAX, a menudo resulta útil descomponer cada uno de los elementos en un lenguaje común. Por ejemplo, puede leer esta fórmula como:
> Para la medida denominada *Total Sales*, calcular (=) la *SUMa* de los valores de la columna *[SalesAmount]* en la tabla *Sales*.

Cuando se agrega a un informe, esta medida calcula y devuelve valores que resulten de sumar los importes de las ventas de cada uno de los demás campos que se incluyen, por ejemplo, el de teléfonos móviles de Estados Unidos.

Quizás se pregunte "¿Esta medida no está hace lo mismo que si simplemente agregara el campo SalesAmount a mi informe?" Bueno, sí. Pero hay una buena razón para crear nuestra propia medida que suma los valores del campo SalesAmount: podemos usarla como argumento en otras fórmulas. Esto puede parecer un poco confuso ahora, pero a medida que desarrolle sus habilidades con las fórmulas DAX, saber esto hará sus fórmulas y modelos más eficientes. De hecho, verá que la medida Ventas totales aparece como argumento en otras fórmulas más adelante.

Veamos algunos otros aspectos de esta fórmula. En concreto, introdujimos una función, **SUM**. Las funciones son fórmulas previamente escritas que facilitan el realizar cálculos complejos y manipulaciones con números, fechas, horas, texto y mucho más. Aprenderá más acerca de las funciones más adelante.

También puede ver que la columna [SalesAmount] estaba precedida de la tabla Sales a la que pertenece la columna. Esto se conoce como un nombre de columna completo porque incluye el nombre de la columna precedido por el nombre de la tabla. Las columnas a las que se hace referencia en la misma tabla no requieren que el nombre de la tabla se incluya en la fórmula. De esta manera, las fórmulas largas que hacen referencia a muchas columnas se vuelven más cortas y fáciles de leer. Sin embargo, es recomendable incluir el nombre de la tabla en las fórmulas de medida, incluso cuando estén en la misma tabla.

> **Nota:** Si un nombre de tabla contiene espacios, palabras clave reservadas o caracteres no permitidos, necesitará incluir el nombre de la tabla entre comillas simples. También deberá escribir los nombres de tabla entre comillas si el nombre contiene algún carácter fuera del rango de caracteres alfanuméricos ANSI, independientemente de si la configuración regional es compatible con el conjunto de caracteres o no.

Es importante que las fórmulas tengan la sintaxis correcta. En la mayoría de los casos, si la sintaxis no es correcta, se devolverá un error de sintaxis. En otros casos, la sintaxis puede ser correcta, pero los valores devueltos podrían no ser los que esperaba. El editor de DAX en Power BI Designer incluye sugerencias; una característica que lo ayuda a seleccionar los elementos correctos para crear fórmulas sintácticamente correctas.

Vamos a crear una fórmula simple. Esta tarea le ayudará a entender mejor la sintaxis de fórmulas y la utilidad de la característica de sugerencias en la barra de fórmulas.

Tarea: Crear una fórmula de medida

Para completar esta tarea, deberá abrir el archivo de muestra de ventas de Contoso de Power BI Desktop.

1. En la vista Informes, en la lista de campos, haga clic con el botón derecho en la tabla **Salesy**, a continuación, haga clic en **Nueva medida**.
2. En la barra de fórmulas, reemplace **Medida** con un nuevo nombre de medida: **Ventas del trimestre anterior**.
3. Después del signo igual, escriba **SUM** seguido de un paréntesis de apertura. > En su lugar de escribir un nombre de columna para sumar inmediatamente, vamos a escribir a otra función, para *filtrar* los datos que deseamos sumar.
4. Entre paréntesis, escriba **CALCULATE**, seguido de un paréntesis de apertura. > Usará la función CALCULATE para filtrar los importes que deseamos sumar mediante un argumento que pasamos a la función CALCULATE. Esto es lo que se conoce como funciones anidadas. La función CALCULATE tiene al menos dos argumentos. El primero es la expresión que se evalúa y el segundo, un filtro.

5. Entre los paréntesis () de la función **CALCULATE** escriba **Sales[SalesAmount]**. Este es el primer argumento de expresión de nuestra función CALCULATE.

6. ** Escriba una coma (,) para especificar el primer filtro y, después, escriba ****PREVIOUSQUARTER** seguido de un paréntesis de apertura. > Utilizará la función de inteligencia de tiempo PREVIOUSQUARTER para filtrar los resultados de la suma para el trimestre anterior.

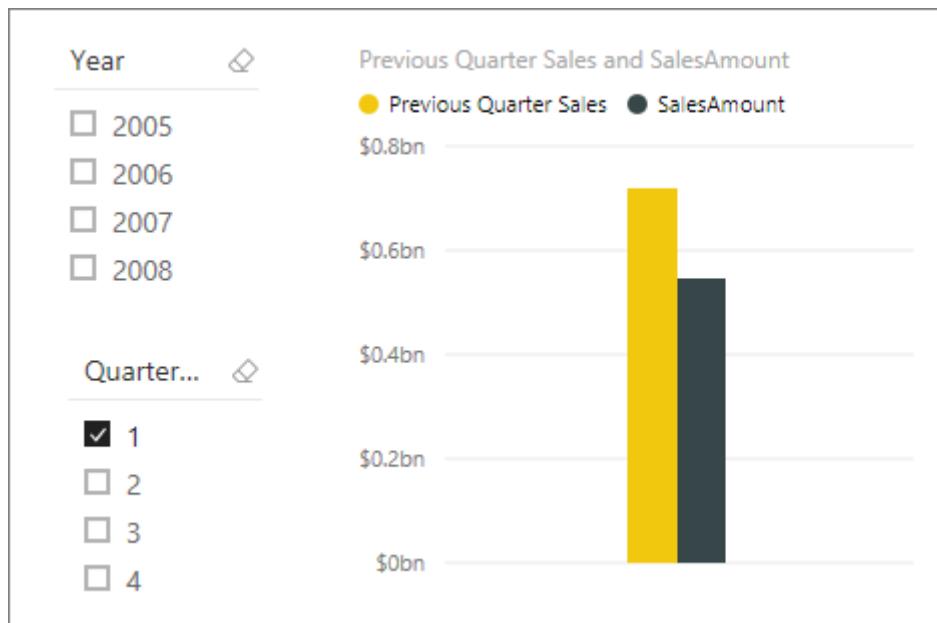
7. ** Entre los paréntesis **** para la función PREVIOUSQUARTER, escriba ****Calendar[DateKey]**. > La función PREVIOUSQUARTER tiene un argumento, una columna que contiene un intervalo de fechas contiguas.

8. ** Asegúrese de que los dos argumentos que se pasan a la función PREVIOUSQUARTER y a la función CALCULATE se cierran con dos paréntesis de cierre **.

La fórmula debe tener el siguiente aspecto: > **Previous Quarter Sales = CALCULATE(SUM(Sales[SalesAmount]), PREVIOUSQUARTER(Calendar[DateKey]))**

**9. ** Haga clic en la marca de verificación ✓ en la barra de fórmulas o presione Entrar para validar la fórmula y agregarla al modelo.

¡Lo logró! Acaba de crear una medida usando DAX (¡y no una de las fáciles!). Lo que hará esta fórmula es calcular el total de ventas del trimestre anterior, según los filtros aplicados en un informe. Por ejemplo, si incluimos SalesAmount y nuestra nueva medida de ventas del trimestre anterior en un gráfico y, a continuación, agregamos Year y QuarterOfYear como segmentaciones de datos, obtendríamos algo parecido a esto:



Le acabamos de presentar varios aspectos importantes de las fórmulas DAX. En primer lugar, esta fórmula incluye dos funciones. Observe que **PREVIOUSQUARTER**, una función de inteligencia de tiempo, se anida como un argumento pasado a la función de filtro **CALCULATE**. Las fórmulas DAX pueden contener hasta 64 funciones anidadas. Es poco probable que una fórmula llegue a contener tantas funciones anidadas. De hecho, una fórmula tal sería muy difícil de crear y depurar, probablemente tampoco sería muy rápida.

En esta fórmula, también usó filtros. Los filtros limitan lo que se calculará. En este caso, se seleccionó un filtro como argumento, que es, de hecho, el resultado de otra función. Aprenderá más acerca de los filtros más adelante.

Por último, usó la función CALCULATE. Esta es una de las funciones más potentes de DAX. A medida que cree modelos y fórmulas más complejas, seguramente volverá a usar esta función muchas veces. Profundizar en la función CALCULATE está fuera del ámbito de este artículo, pero le recomendamos prestar especial atención a esta función conforme aumente sus conocimientos de DAX.

Cuestionario rápido sobre sintaxis

1. ¿Qué hace este botón de la barra de fórmulas? >  >
2. ¿Qué rodea siempre a un nombre de columna en una fórmula DAX?

Encontrará las respuestas al final de este artículo.

Funciones

Las funciones son fórmulas predefinidas que realizan cálculos por medio de valores específicos, denominados argumentos, en un orden o estructura determinados. Los argumentos pueden ser otras funciones, otra fórmula, una expresión, referencias de columna, números, texto, valores lógicos como TRUE o FALSE, o constantes.

DAX incluye las siguientes categorías de funciones: [Fecha y hora](#), [Inteligencia de tiempo](#), [Información](#), [Lógicas](#), [Matemáticas](#), [Estadísticas](#), [Texto](#), [Primarias/Secundarias](#) y [Otras](#). Si está familiarizado con las funciones en las fórmulas de Excel, muchas de las funciones DAX le parecerán semejantes; sin embargo, las funciones DAX son únicas de las maneras siguientes:

- Una función DAX siempre hace referencia a una columna o tabla completa. Si desea usar solo determinados valores de una tabla o columna, puede agregar filtros a la fórmula.
- Si necesita personalizar los cálculos fila por fila, DAX ofrece funciones para usar el valor de la fila actual o un valor relacionado como un tipo de argumento, lo que permite realizar cálculos que varían según el contexto. Aprenderá más acerca del contexto más adelante.
- DAX incluye muchas funciones que devuelven una tabla en lugar de un valor. La tabla no se muestra, pero se usa para proporcionar datos para otras funciones. Por ejemplo, puede recuperar una tabla y después contar los diferentes valores que contenga o calcular sumas dinámicas en tablas o columnas filtradas.
- DAX incluye una variedad de funciones de inteligencia de tiempo. Estas funciones permiten definir o seleccionar intervalos de fechas y realizar cálculos dinámicos con base en ellos. Por ejemplo, puede comparar sumas de períodos paralelos.
- Excel tiene una función muy popular, BUSCARV. Las funciones DAX no toman una celda o rango de celdas como una referencia como BUSCARV en Excel. Las funciones DAX toman una columna o una tabla como referencia. Tenga en cuenta que, en Power BI Desktop, está trabajando con un modelo de datos relacionales. Buscar valores en otra tabla es verdaderamente fácil y en la mayoría de los casos no es necesario crear ninguna fórmula.

Como puede ver, las funciones de DAX pueden ayudarle a crear fórmulas muy eficaces. En realidad, solo hemos tratado los conceptos básicos de las funciones. A medida que desarrolle sus habilidades de DAX, podrá crear fórmulas con muchas funciones diferentes. Uno de los mejores lugares para obtener información detallada sobre cada una de las funciones de DAX está en la [referencia de funciones de DAX](#).

Cuestionario rápido sobre funciones

1. Una función siempre hace referencia a...
2. ¿Puede una fórmula contener más de una función?
3. ¿Qué categoría de funciones usaría para concatenar dos cadenas de texto en una sola cadena?

Encontrará las respuestas al final de este artículo.

Contexto

El contexto es uno de los conceptos de DAX más importantes. Hay dos tipos de contexto en DAX; contexto de fila y contexto de filtro. En primer lugar, examinaremos el contexto de fila.

Contexto de fila

Es más fácil pensar en el contexto de fila como la fila actual. Se aplica siempre que una fórmula tiene una función que usa filtros para identificar una fila individual en una tabla. De manera inherente, la función aplicará un contexto de fila para cada fila de la tabla que está filtrando. Este tipo de contexto de fila se aplica con mayor frecuencia en medidas.

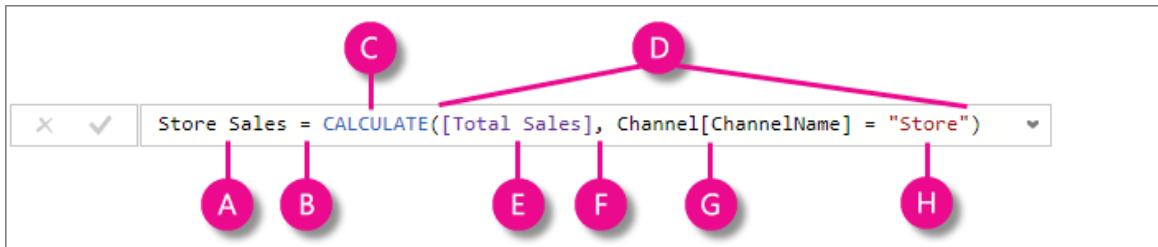
Contexto de filtro

El contexto de filtro es un poco más difícil de entender que el contexto de fila. Para simplificarlo, piense en el contexto de filtro como uno o varios filtros aplicados en un cálculo que determina un resultado o valor.

El contexto de filtro no existe en lugar del contexto de fila; más bien, se aplica además del contexto de fila. Por ejemplo, para restringir aún más los valores que desea incluir en un cálculo, puede aplicar un contexto de filtro que no solo especifique el contexto de fila, sino que también especifique únicamente un valor determinado (filtro) en ese contexto de fila.

El contexto de filtro se puede identificar fácilmente en los informes. Por ejemplo, cuando agrega el costo total a una visualización y, a continuación, el año y región, define un contexto de filtro que selecciona un subconjunto de datos basándose en un año y región determinados.

¿Por qué es tan importante para DAX el contexto de filtro? Porque aunque el contexto de filtro se puede aplicar más fácilmente agregando campos a una visualización, también puede aplicarse en una fórmula DAX al definir un filtro mediante funciones como ALL, RELATED, FILTER, CALCULATE, por relaciones, así como por otras medidas y columnas. Por ejemplo, echemos un vistazo a la siguiente fórmula en una medida denominada Store Sales:



Para entender mejor esta fórmula, podemos desglosarla, similar a como podemos hacerlo otras fórmulas.

Esta fórmula incluye los siguientes elementos de sintaxis:

- El nombre de medida **Store Sales**.
- El operador signo igual (**=**) indica el principio de la fórmula.
- La función **CALCULATE** evalúa una expresión, como argumento, en un contexto modificado por los filtros especificados.
- Los paréntesis (**()**) envuelven una expresión que contiene uno o más argumentos.
- Una medida **[Total Sales]** en la misma tabla como una expresión. La medida Total Sales tiene la fórmula: **=SUM(Sales[SalesAmount])**.
- Una coma (**,**) separa el primer argumento de expresión del argumento de filtro.
- La columna completa a la que se hace referencia, **Channel[ChannelName]**. Se trata de nuestro contexto de fila. Cada fila de esta columna especifica un canal: Store, Online, etc.
- El valor concreto **Store** como filtro. Este es el contexto de filtro.

Esta fórmula garantiza que únicamente se calculen los valores de ventas definidos por la medida Total Sales y solamente para las filas de la columna **Channel[ChannelName]** con el valor "Store" como filtro.

Como puede imaginar, la posibilidad de definir el contexto de filtro dentro de una fórmula ofrece grandes y potentes capacidades. La capacidad de hacer referencia solamente a un valor determinado en una tabla relacionada es solo un ejemplo. No se preocupe si no comprende por completo el concepto de contexto inmediatamente. A medida que cree sus propias fórmulas, comprenderá mejor el contexto y por qué es tan importante en DAX.

Cuestionario rápido sobre el contexto

- ¿Cuáles son los dos tipos de contexto?
- ¿Qué es el contexto de filtro?
- ¿Qué es el contexto de fila?

Encontrará las respuestas al final de este artículo.

Resumen

Ahora que tiene conocimientos básicos de los conceptos más importantes en DAX, puede empezar a crear fórmulas DAX para medidas por su cuenta. En efecto, las expresiones DAX pueden ser un poco complicadas de aprender, pero hay muchos recursos a su

disposición. Después de leer este artículo y experimentar con algunas de fórmulas propias, puede obtener más información acerca de otros conceptos y fórmulas de DAX que le ayudarán a solucionar sus problemas de negocios. Existen muchos recursos de DAX disponibles; el más importante es la [Referencia de expresiones de análisis de datos \(DAX\)](#).

DAX se usa desde hace varios años en otras herramientas de Microsoft BI, como los modelos tabulares de Power Pivot y Analysis Services, por lo que hay una gran cantidad de información disponible. Puede encontrar más información en libros, notas de producto y blogs tanto de Microsoft como de profesionales líderes de BI. El [Wiki del Centro de recursos de DAX de TechNet](#) también es un excelente punto de partida.

Respuestas de los cuestionarios rápidos:

Sintaxis:

1. Valida e introduce la medida en el modelo.
2. Corchetes [].

Funciones:

1. Una tabla y una columna.
2. Sí. Una fórmula puede contener hasta 64 funciones anidadas.
3. [Funciones de texto](#).

Contexto:

1. Contexto de fila y contexto de filtro.
2. Uno o más filtros en un cálculo que determina un valor único.
3. La fila actual.

Introducción a Power Query y el Lenguaje M

Pongamos Power Query y el Idioma M bajo la lupa

Power Query es una herramienta de gran avance para el auto-servicio de inteligencia empresarial y es la nueva mejor herramienta lanzada por Microsoft en el lapso de 5 años. Ya he publicado acerca de lo [que es Power Query y lo que representa para nosotros](#) pero, [¿Cómo puedo empezar con Power Query y el M Idioma?](#)

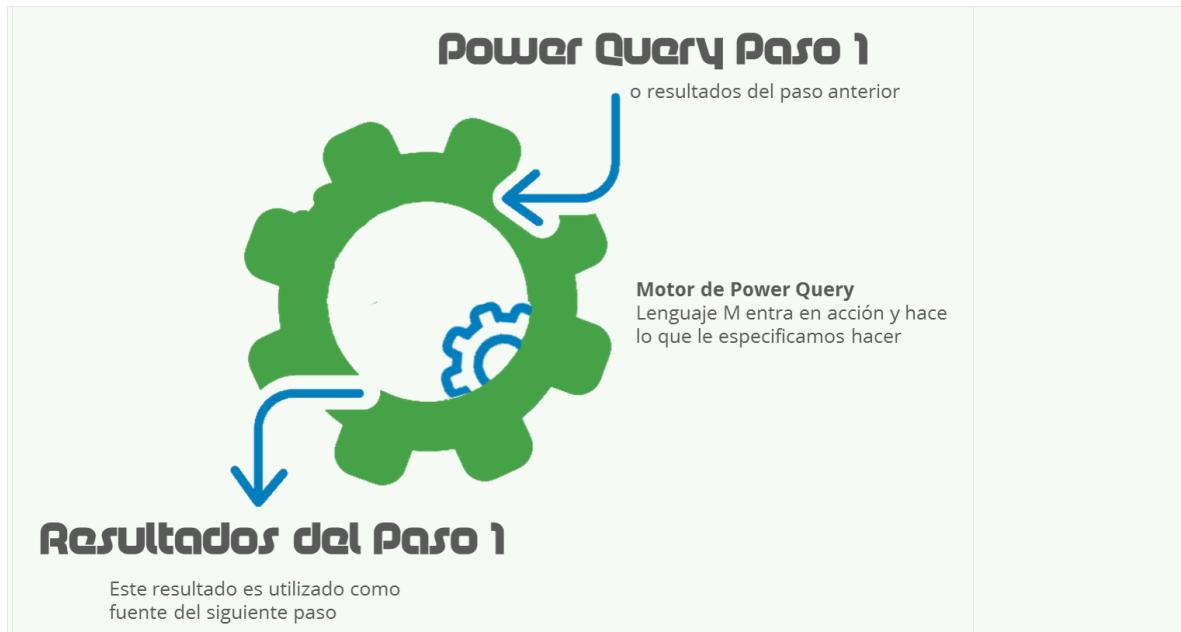
Bueno, este post va a tratar de hacer frente a esa pregunta y le dará una idea de cómo empezar a utilizar Power Query y entender su lenguaje.

He dividido este artículo en 2 secciones principales que pienso firmemente que deben ser el foco principal de nuestro viaje cuando se intenta aprender Power Query y el idioma M. Estos son:

1. Flujo de trabajo en Power Query y el mecanismo M
2. El Lenguaje M

Vamos a empezar con lo básico!

El flujo de trabajo – paso a paso



(La imagen muestra el flujo de trabajo de una consulta Power Query. Funciona en una base por pasos basada en el resultado de algún paso anterior * la mayor parte del tiempo *)

Estas son algunas de las razones por las cuales la gente le encanta este flujo de trabajo con Power Query:

- Trabaja basado en pasos, algo así como el funcionamiento de una macro-grabadora y te invita a trabajar en una forma ordenada.
- Se le mantiene a raya con el fin de aplicar las mejores prácticas de transformación de datos lo cual es impresionante!
- Es fácil de leer y fácil de depurar

La imagen de arriba explica que tenemos un paso, y ese paso tiene algo escrito en ella. El contenido de ese paso podría ser llamado un guión, específicamente una secuencia de comandos M de algo que tiene que suceder. El motor de Power Query aplica dicho guión y el resultado de él se mostrará en la parte izquierda de su pantalla que se podría llamar la vista previa en vista de consulta. Ese resultado puede ser utilizado más tarde para más transformaciones, pero vamos a ver un ejemplo para entenderlo.

Ultimate Combination - Query Editor

APPLIED STEPS

- Source
- Get the files
- Removed Other Columns
- Trans1
- Trans2
- The Formula
- Removed Columns
- Almost there
- MyList
- Here we go
- Expanded

Extraer los datos de una "Source" (fuente)

Transformar los datos paso por paso
(ordenamiento para su fácil lectura)

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
1 Binary	ahora en csv.txt	.txt	12/6/2014 10:47:15 PM	12/6/2014 10:47:15 PM	1/9/2015 4:55:19 AM	Record	C:\Users\Miguel
2 Binary	archivo de prueba.csv	.csv	12/6/2014 8:20:48 PM	12/6/2014 8:20:48 PM	1/9/2015 4:53:51 AM	Record	C:\Users\Miguel
3 Binary	File1.xlsx	.xlsx	11/21/2014 3:32:17 PM	11/21/2014 3:32:17 PM	1/9/2015 4:53:46 AM	Record	C:\Users\Miguel
4 Binary	File10.xlsx	.xlsx	11/21/2014 3:34:32 PM	11/21/2014 3:34:32 PM	1/9/2015 5:00:11 AM	Record	C:\Users\Miguel
5 Binary	File11.xlsx	.xlsx	11/21/2014 3:34:20 PM	11/21/2014 3:34:20 PM	1/9/2015 4:59:02 AM	Record	C:\Users\Miguel
6 Binary	File12.xlsx	.xlsx	11/21/2014 3:34:02 PM	11/21/2014 3:34:02 PM	1/9/2015 4:57:49 AM	Record	C:\Users\Miguel
7 Binary	File7.xlsx	.xlsx	11/21/2014 3:32:42 PM	11/21/2014 3:32:42 PM	1/9/2015 4:55:29 AM	Record	C:\Users\Miguel
8 Binary	File8.xlsx	.xlsx	11/22/2014 9:02:49 PM	11/22/2014 9:02:49 PM	1/9/2015 4:59:44 AM	Record	C:\Users\Miguel
9 Binary	File9.xlsx	.xlsx	11/21/2014 3:33:32 PM	11/21/2014 3:33:32 PM	1/9/2015 5:00:49 AM	Record	C:\Users\Miguel

8 COLUMNS, 9 ROWS

PREVIEW DOWNLOADED AT 12:05 AM.

En la imagen de arriba vemos la consulta que se ha creado en base a mi [anterior entrada del blog](#). Vemos que el primer paso se llama Source, y lo que hace se puede ver en la **barra de fórmulas que comienza con “= Folder.Files (...)”** y de ese paso estaremos agregando un nuevo paso llamado “**“Get the files”**” y después de ese, y en base a este último, creamos un nuevo paso llamado “**“Removed Other Columns”**”.

Sólo para resumir, he aquí cómo diseccionar un solo paso:

- **El nombre de mi paso = “Source”**
- **Lo que hace mi paso = “=Folder.Files(…”**
- **El resultado de este paso =** todas las columnas mostradas en el lado izquierdo de la pantalla a partir de [Content] hasta el final a [Folder Path]

Algo a tomar en consideración es que cualquiera de las entradas o resultados puede ser un registro, lista, tabla o una función, pero por ahora sólo estamos interesados en conocer el flujo de trabajo y no todos los detalles.

Este es el ciclo o flujo de trabajo para Power Query. Usted trabaja más o menos como el uso de la grabadora de macros, grabando todo en una base paso a paso y cuando termine de grabar ejemplo “macro”, usted sólo tiene que ir a “Close & Load” para que pueda guardar la consulta y cargarla ya sea a Excel o su modelo de datos.

Cómo entender el lenguaje

Vamos a diseccionar una vez más la consulta de mi última entrada en el blog y tratar de analizar los patrones de ese código M:

The screenshot shows the Microsoft Power Query Advanced Editor interface. On the left, the 'Advanced Editor' window displays the M code for 'Ultimate Combination'. On the right, the 'Query Settings' pane shows the query name 'Ultimate Combination' and a list of 'APPLIED STEPS' with corresponding orange arrows pointing to the M code. The M code itself is quite complex, involving file reading, table transformations, and formula calculations. The 'APPLIED STEPS' list includes steps like 'Source', 'Get the files', 'Removed Other Columns', 'Trans1', 'Trans2', 'The Formula', 'Removed Columns', 'Almost there', 'MyList', 'Here we go', and 'Expanded'.

```

let
    Source = Folder.Files("C:\Users\Miguel\Google Drive\Best Way to Combine"),
    "Get the files" = Table.SelectRows(Source, each [Extension] = ".csv" or [Extension] = ".txt" or Text.StartsWith([Extension], "#Removed Other Columns")),
    Trans1 = Table.SelectColumns("Get the files", {"Content", "Name", "Extension"}),
    Trans2 = Table.ExpandTableColumn(Trans1, "Custom", {"Name", "Data", "Kind"}, {Custom.Name, Custom.Data, Custom.It}),
    "The Formula" = Table.AddColumn(Trans2, "Custom", each if [Extension] = ".csv" then Table.PromoteHeaders(Csv.Document([Content])) else if [Extension] = ".txt" then Table.PromoteHeaders(Csv.Document([Content],null,"")) else if [Custom.Kind] <> "Table" then Table.PromoteHeaders([Custom.Data])),
    "Removed Columns" = Table.RemoveColumns("The Formula", {"Content", "Custom.Data"}),
    "Almost there" = Table.AddColumn("Removed Columns", "Custom.1", each Table.RowCount([Custom])),
    MyList = Table.ToList(Table.RemoveColumns(Table.Distinct(Table.ExpandListColumn(Table.AddColumn("#Almost there", "ColumnHeader", "#Here we go", Table.RenameColumns("Almost there",{{"Custom.1", "Total Rows"}, {"Custom.Kind", "Kind"}, {"Custom.Item", "Item"}, {"Expanded", "Table.ExpandTableColumn(#Here we go", "Custom", MyList)}})))),
    in
        Expanded

```

Aquí se tiene una imagen lado a lado del Editor avanzado, el código M en su forma cruda , y detrás de ella el Editor de consultas Power Query con la lista de pasos a la derecha. Seguí adelante y dibuje una línea con cada paso correspondiente con el fin de que les muestre cual corresponde a cual.

Así que eso es bastante fácil, sabemos donde cada paso se almacena y donde está, pero te darás cuenta de que los nombres no son exactamente los mismos. Por ejemplo, el segundo paso se llama **Get the files** pero en el Editor avanzado se ve el nombre como **#"Get the files"** y la razón detrás de esto es porque M no puede manejar espacios entre nombres o conflictos con otros nombres (nombres de paso o incluso nombres de las funciones) así que por eso se agrega el # como prefijo y se pone todo dentro de las comillas.Usted no notará esto en la interfaz de usuario ya que le va a mostrar el nombre

que usted le ha dado, pero, bajo el capó se dará cuenta de cómo se traduce al lenguaje M.

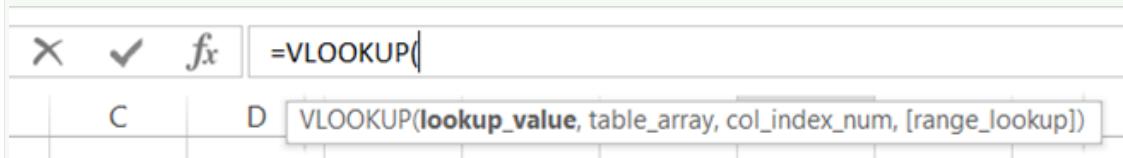
Yo también he destacado 2 palabras:

- **let** – es la sintaxis a utilizar para comenzar una consulta.
- **in** – es la sintaxis para determinar que usted ha terminado el proceso de transformación y, posteriormente, verá el nombre del paso que debe ser entendido como el resultado de dicha consulta. En nuestro caso “**Expanded**” es el resultado de nuestra consulta de PQ.

Lenguaje M – recursos de aprendizaje

Para los recursos oficiales de aprendizaje de Microsoft, favor visite el siguiente [link](#) donde se puede obtener más información sobre las fórmulas y funciones dentro de Power Query.

Además, aquí hay otro truco fácil. Es probable que ya amas esto de Excel y Power Pivot:



y si desea LA misma experiencia, o muy similar a ella, puede simplemente escribir = # shared en la barra de fórmulas de consulta y obtener una lista completa de todas las funciones disponibles en su versión de Power Query:

A screenshot of the Microsoft Power Query Editor window titled 'Query1 - Query Editor'. The formula bar at the top has '= #shared'. A dropdown menu is open, listing various Power Query functions such as Table.Max, Table.MaxN, Table.Min, Table.MinN, Table.Buffer, Table.FindText, Table.ReplaceValue, Table.IsEmpty, Table.SplitColumn, Table.CombineColumns, List.Count, List.Distinct, List.FirstN, List.IsEmpty, List.LastN, List.Select, List.Skip, List.Sort, List.Transform, and List.TransformMany. To the right of the formula bar, there are 'Properties' and 'Applied Steps' panes. The 'Properties' pane shows 'Name: Query1'. The 'Applied Steps' pane shows 'Source'. At the bottom of the editor, it says '599 FIELDS'.

y si desea el artículo de ayuda completo de cualquier función todo lo que tienes que hacer es escribir el nombre de la función de la siguiente manera:

Query1 - Query Editor

Invoke

= Table.Max

```
function (table as table, comparisonCriteria as any, optional default as any) as any
```

Returns the largest row in the table, given the comparisonCriteria. If the table is empty, the optional default value is returned.

Example: Find the row with the largest value in column [a] in the table {{[a = 2, b = 4], [a = 6, b = 8]}}.

Usage:
Table.Max(Table.FromRecords({|a = 2, b = 4|, |a = 6, b = 8|}), "a")

Output:
[a = 6, b = 8]

Example: Find the row with the largest value in column [a] in the table {{ }}. Return -1 if empty.

Usage:
Table.Max(#table("a", {}), "a", -1)

Output:
-1

READY

PREVIEW DOWNLOADED AT 4:48 AM.

Query Settings

PROPERTIES
Name: Query1
All Properties

APPLIED STEPS
Source
Table.ReplaceValue