



Conexión a carpeta

01 ETL y Modelo de Datos. PQ Conexión a carpeta con transformación.

Jose Ignacio González Gómez

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

www.jggomez.eu

R26

Contenido

1	Presentación.....	2
1.1	Objetivo de la conexión a carpetas	2
1.2	Ejemplos de uso	2
1.3	Consideraciones previas.....	3
2	Las conexiones a las carpetas, funcionamiento básico.....	3
2.1	Selección de la carpeta y del fichero de ejemplo o de referencia	3
2.2	Carpeta "Transformar archivo de.."	4
2.3	Subcarpeta Consulta Auxiliar (A).....	5
2.4	Otras Consultas (B).....	5
2.5	Conveniencia de crear un parámetro de conexión	6
3	Cambiando el fichero de ejemplo o de referencia.....	6
3.1	Introducción	6
3.2	Archivo de ejemplo, por defecto. Significado de cada parte	7
3.3	Cambiar el origen del "Archivo de ejemplo"	7

1 Presentación

1.1 Objetivo de la conexión a carpetas

La conexión a una carpeta tiene como objetivo principal importar y combinar archivos que se encuentran en una misma carpeta. Esto es especialmente útil cuando disponemos de múltiples archivos con la misma estructura (como informes mensuales) y deseamos consolidar todos esos datos en una sola tabla para su análisis. Esto te permite:

1. **Importar varios archivos al mismo tiempo:** Si tenemos una carpeta con varios archivos del mismo tipo (como Excel, CSV, etc.), Power Query puede combinarlos en un solo conjunto de datos.
2. **Automatizar la actualización de datos:** Si los archivos en la carpeta se actualizan o se agregan nuevos archivos, el editor de consulta puede procesar y actualizar los datos automáticamente sin necesidad de ajustar cada archivo individualmente.
3. **Ahorro de tiempo:** Evita la labor tediosa de importar y transformar cada archivo por separado, lo que es especialmente útil para grandes volúmenes de datos.
4. **Consistencia en el procesamiento de datos:** Todos los archivos en la carpeta se procesan de la misma manera, asegurando que los datos sigan el mismo formato y estructura.

En resumen, la conexión a una carpeta en Power Query es una herramienta poderosa para automatizar la importación, combinación y transformación de datos de múltiples archivos, lo que facilita el análisis y la toma de decisiones informadas.

1.2 Ejemplos de uso

Presentamos a continuación algunos ejemplos de casos de uso:

Análisis de ventas

Nos permite combinar datos de ventas de diferentes archivos mensuales en un solo informe. Por ejemplo, una empresa genera un informe de ventas en formato Excel cada mes. Conectar Power Query a una carpeta donde se almacenan estos archivos permite combinar todos los informes mensuales en un solo informe acumulado.

Consolidación de datos de encuestas

Unir respuestas de encuestas almacenadas en múltiples archivos CSV, es decir combinar todas las respuestas en un único conjunto de datos, ahorrando tiempo y esfuerzo.

Finanzas y Contabilidad

Para empresas que reciben extractos bancarios, reportes financieros, balances por sucursales, archivos de texto con transacciones, bases de datos con información de clientes, etc. en múltiples archivos Excel mensuales, una conexión a carpeta permite reunir toda la información financiera en un único lugar para análisis y reconciliación.

Integración con Datos de Proveedores

Por ejemplo, varios proveedores envían archivos de precios de productos y estos se almacenan en una carpeta. Power Query permite combinarlos y actualizar automáticamente la base de datos de precios.

Datos de Sensores

Si estamos trabajando con datos de sensores que generan archivos CSV cada hora o cada día, podemos usar una conexión a carpeta para consolidar todos estos datos en una sola tabla, facilitando el análisis.

Investigación y Análisis de Datos:

En proyectos de investigación donde se recolectan datos en archivos individuales (por ejemplo, estudios de campo con múltiples registros), conectar una carpeta en Power Query facilita la integración y análisis de todos los datos recogidos.

Seguimiento de Inventarios de Varias Sucursales

Empresas que manejan inventarios en múltiples ubicaciones pueden tener archivos de inventario separados por ubicación. Al conectar una carpeta en Power Query, podemos consolidar todos estos archivos en un solo inventario maestro.

1.3 Consideraciones previas

Antes de empezar recomendamos tomar en consideración los siguientes aspectos:

- Asegúrese de que todos los archivos que desea combinar están contenidos en una carpeta dedicada sin archivos extraños. En caso contrario, todos los archivos de la carpeta y las subcarpetas que seleccione se incluirán en los datos que se combinarán.
- Cada archivo debe tener el mismo esquema con encabezados de columna coherentes, tipos de datos y número de columnas. Las columnas no tienen que estar en el mismo orden que la coincidencia se realiza mediante nombres de columna.
- El fichero debe tener una estructura similar en el sentido de que las hoja/s que contiene la información así como las tablas deben guardar coherencia.
- Si es posible, evite objetos de datos no relacionados para orígenes de datos que pueden tener más de un objeto de datos, como un archivo JSON, un libro de Excel o una base de datos de Access

Todos los pasos de limpieza y transformación que aplicaremos al fichero ejemplo o de referencia se reproducirán al resto de ficheros que añadiremos.

2 Las conexiones a las carpetas, funcionamiento básico

2.1 Selección de la carpeta y del fichero de ejemplo o de referencia

Expondremos a continuación cómo funciona las conexiones a las carpetas. Cuando se realiza una conexión a una carpeta supone que deseamos combinar o anexar el contenido de los distintos ficheros que contiene la misma, lo cual exige que estos no solo sean del mismo tipo (pdf, xls, csv...) sino que además tengan la misma estructura o composición, como son el número y tipo de columnas, el número de tablas, etc... En el caso de que existan ficheros de distinto tipo tendremos que seleccionar un único tipo de fichero (pdf, xls, csv...) con el que queremos trabajar.

Una vez creada la conexión a, tal y como muestra la Ilustración 1 se nos presenta la relación de ficheros disponibles, proponiendo como fichero ejemplo o de referencia el primero, pero podríamos seleccionar otro de los disponibles. Este será el fichero de ejemplo sobre el que se definirán las transformaciones necesarias que se replicarán al resto de ficheros.

Señalar que sería conveniente seleccionar aquel fichero que requiera de más procesos y técnicas de depuración y transformación, por tanto, sería conveniente llevar a cabo un análisis de los ficheros disponibles y de su contenido.

Una vez seleccionado el fichero de ejemplo, el editor de consulta analiza su contenido e identifica los distintos elementos que lo componen (Parámetro), en nuestro caso a identificado 3 elementos, 2 tablas y la página completa.

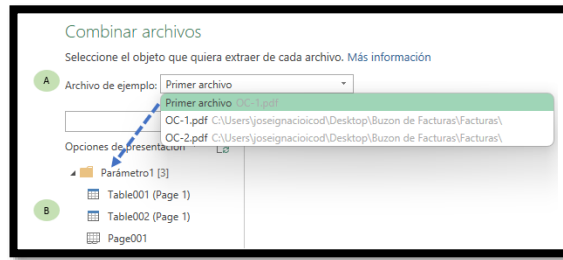


Ilustración 1

Podremos visualizar la información disponible en cada elemento para seleccionar según su contenido, el que contiene la información que nos interesa, si esta se encuentra en una tabla concreta o en toda la página.

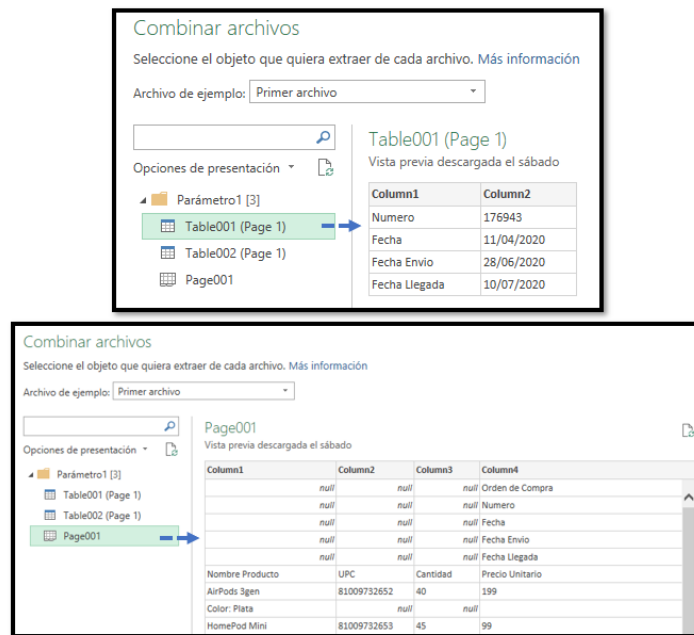


Ilustración 2

El resultado de la conexión es la creación de dos estructuras “Transformar archivo de...” y “Otras Consultas”, ambas vinculadas y que pasamos a analizar (ver Ilustración 3).

2.2 Carpeta “Transformar archivo de..”

Una vez seleccionado el archivo de ejemplo y el elemento (tabla o página) que contiene la información de interes, el editor de consultas genera una estructura para cada conexión que la agrupa en la carpeta “Transformar archivo de Facturas [2]” donde el nombre se corresponde con el nombre de la carpeta a la que estamos conectando (Facturas, por ejemplo) y lo añade a la raíz “Transformar archivo de” generando así la carpeta de conexión.

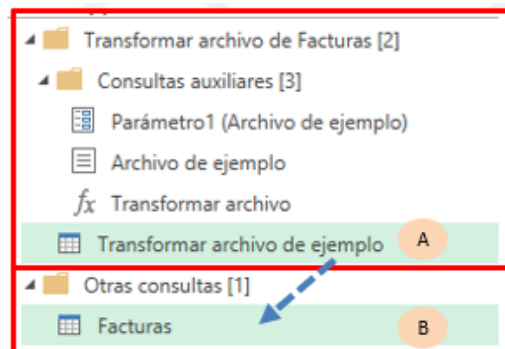


Ilustración 3

El numero 2 se refiere al número de elementos disponibles en la parametrización de la conexión y que son:

1. La subcarpeta Consultas Auxiliares [3], que contiene tres elementos relacionados con los parametros de configuración que definen la consulta.

- El fichero “Transformar archivo de ejemplo” (A), es una copia del seleccionado en la Ilustración 2 y sobre el que se aplicarán todos los ajustes de limpieza y transformación generando una plantilla que se replicarán uno a uno al resto de ficheros contenidos en la carpeta de conexión.

2.3 Subcarpeta Consulta Auxiliar (A)

Centrándonos en la subcarpeta Consultas Auxiliares esta contiene tres elementos (A, B y C):

- Parámetro 1 en el que se define un parámetro que toma como valor “Archivo de Ejemplo” podríamos cambiarlo como, por ejemplo, fichero de referencia.
- Identifica al fichero seleccionado en la Ilustración 1.
- Es una función interna creada.
- Como señalamos el fichero “Transformar archivo de ejemplo” contiene la plantilla de ajustes de limpieza y transformación que se replicaran a cada fichero contenidos en la carpeta de conexión.

Es importante señalar que si creamos un parámetro de conexión debemos actualizarlo en los siguientes elementos en el apartado origen del archivo: B.

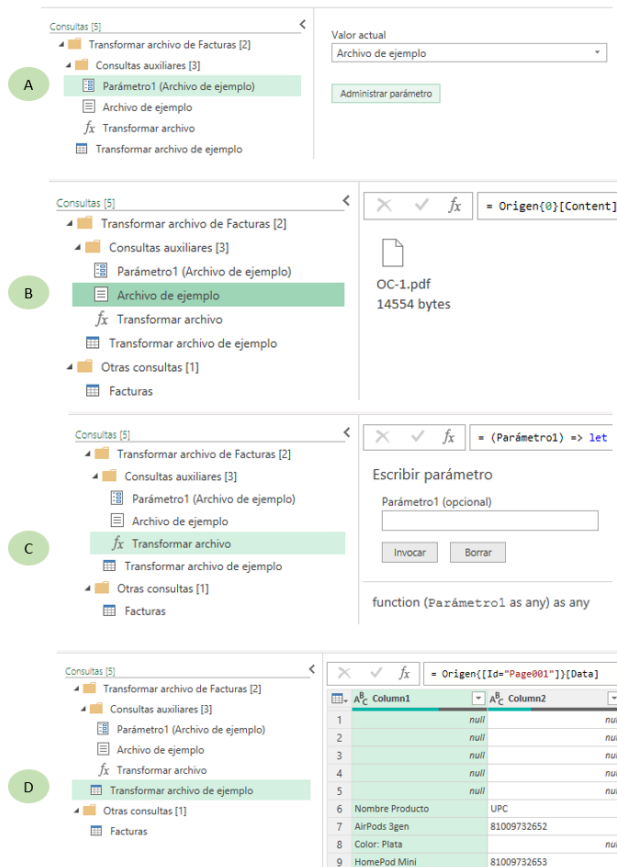


Ilustración 4

2.4 Otras Consultas (B)

En el momento de crear la conexión tal y como hemos mostrado en la Ilustración 3 se ha creado además otra carpeta con el nombre “Otras consultas”, la cual contiene una consulta de agregación consolidación de la información contenido en el resto de ficheros disponibles en la carpeta a la que estamos conectado, y a cada uno de estos ficheros se les ha aplicado la plantilla de limpieza y depuración, s.

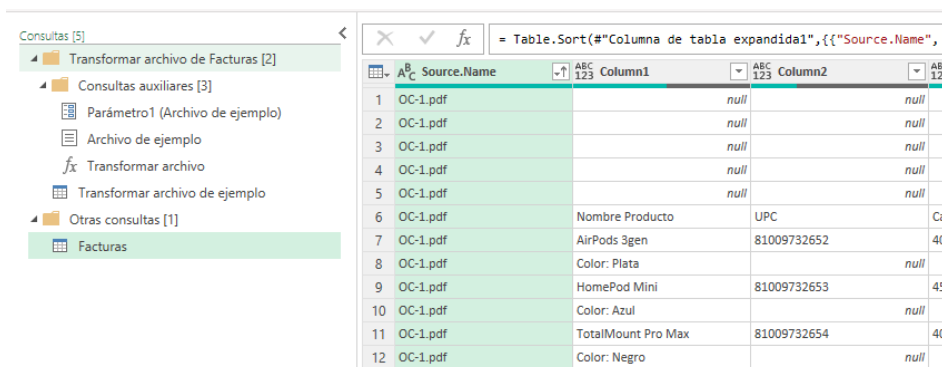


Ilustración 5

Así y según la Ilustración 3, A contiene la plantilla con todos los ajustes necesarios de limpieza y transformación y la consulta B es el resultado de aplicar esta plantilla a todos los

ficheros contenidos en la carpeta de conexión, y según el caso se han consolidado en una sola tabla.

En resumen

De esta forma los ajustes de limpieza y transformación que se definan sobre el fichero de ejemplo o de referencia serán replicados al resto para proceder posteriormente a combinar o anexas el contenido de cada uno de ellos en uno solo que será el resultado u objetivo de la conexión a la carpeta.

2.5 Conveniencia de crear un parámetro de conexión

Es recomendable contar con un parámetro de conexión y aplicarlo a las distintas conexiones de carpetas que hayamos creado ya que esto permite mejorar significativamente la flexibilidad en la actualización del origen de datos al cambiar fácilmente las fuentes (como rutas de archivos o nombres de servidores) sin tener que modificar cada consulta individualmente lo que hace que el mantenimiento sea más sencillo.

En la siguiente ilustración hemos creado el parámetro de conexión denominado "ConexiónCarpeta" y que contiene la ruta a la fuente de datos (carpeta que nos interesa), así una vez creado el parametro tendremos que cambiar el origen de las 2 consultas al citado parámetro, Archivo de ejemplo y Facturas, tal y como se muestra en la Ilustración 6

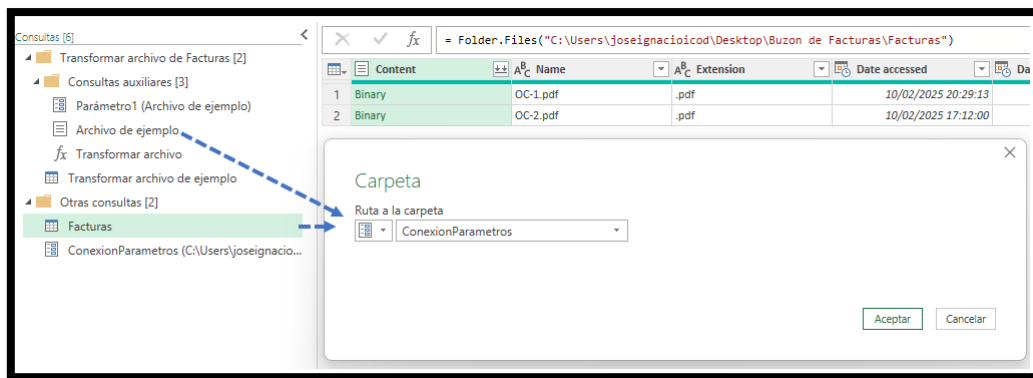


Ilustración 6

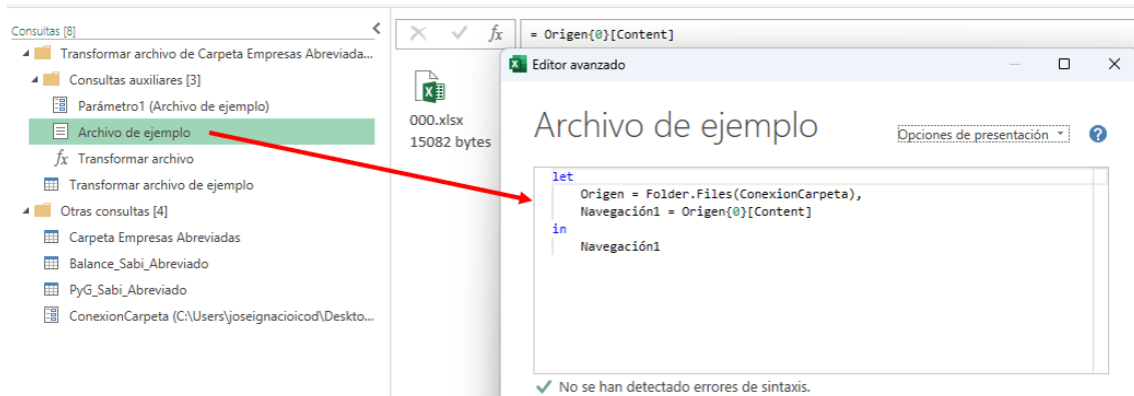
3 Cambiando el fichero de ejemplo o de referencia

3.1 Introducción

Cuando conectas una carpeta, Power Query crea automáticamente un grupo de consultas en el panel izquierdo llamado **Consultas auxiliares** y entre ellas contiene una consulta llamada Archivo de ejemplo, es este el que se ha tomado como ejemplo para hacer los ajustes y configuraciones que posteriormente se replicaran a todos los ficheros que se añadan a la carpeta de conexión o contenedora.

Cambiar el archivo de ejemplo es un proceso común cuando el primer archivo de la carpeta no es el más representativo o cuando necesitas que Power Query base su estructura (columnas, tipos de datos) en uno específico como **Pepe.xlsx**, que contiene otros ajustes mas amplios y sensibles.

Para ello debemos tener en cuenta que cuando se creó la conexión a la carpeta se selecciono el primer fichero que encontró y que se tomo como referencia para hacer los ajustes (Archivo de ejemplo), si accedemos al editor avanzado el resultado es el de la siguiente ilustración.



3.2 Archivo de ejemplo, por defecto. Significado de cada parte

Consulta a Copilot [que significa en power query y lenguaje M lo... | Probar en Copilot Chat](#)

1. Origen = Folder.Files(ConexionCarpeta)
 - Folder.Files devuelve una tabla con todos los archivos (no subcarpetas) encontrados en la ruta indicada por ConexionCarpeta.
 - En resumen: Origen es una tabla con una fila por archivo y el binario del archivo en la columna Content.
2. Navegación1 = Origen{0}[Content]
 - Navegación1 es el binario del primer archivo que devuelve Folder.Files.
3. in Navegación1
 - La consulta devuelve ese binario como resultado final. Normalmente, este binario se pasa a un conector de importación apropiado (por ejemplo, Excel.Workbook(Navegación1, ...), Csv.Document(Navegación1, ...), etc.) en un paso posterior, para leer el contenido del archivo.

Implicaciones y posibles problemas

- Dependencia del orden: Acceder a Origen{0} supone que el primer archivo es el que quieres. El orden de los archivos no está garantizado y puede variar (por cambios en el sistema de archivos, actualizaciones del motor, etc.). Esto puede provocar que, de forma aleatoria, leas otro archivo distinto.
- Error si la carpeta está vacía: Si Folder.Files(ConexionCarpeta) devuelve cero filas, Origen{0} producirá un error de índice fuera de rango.
- Ambigüedad: No filtra por nombre, extensión, fecha, ni ningún criterio, por lo que es frágil si hay múltiples archivos en la carpeta.

Cuándo usar Origen{0}[Content]

Útil únicamente cuando:

- Estás seguro de que la carpeta contendrá un solo archivo válido.
- El orden de los archivos no variará (o te es indiferente cuál se lea).
- Es un prototipo rápido o prueba puntual.

En cualquier otro caso, es preferible filtrar y validar para evitar errores y lecturas incorrectas.

3.3 Cambiar el origen del "Archivo de ejemplo"

Podríamos seleccionar o elegir un nuevo fichero para convertirlo en archivo de ejemplo y sobre el que hacer los ajustes deseados para ello debemos cambiar el código de M de la ilustración por el siguiente.

Codigo propuesto 1

let

```
Origen = Folder.Files(ConexionCarpeta),
Filtro = Table.SelectRows(Origen, each [Name] = "MiArchivo.xlsx"),
Validacion = if Table.RowCount(Filtro) = 1
    then Filtro{0}[Content]
    else error "Archivo no encontrado o hay múltiples coincidencias.",
Binario = Validacion
```

in

Binario

Codigo propuesto 2

Una variante del codigo anterior mas explicita es la siguiente

let

```
Origen = Folder.Files("C:\Users\joseignacioicod\Desktop\MFinanciero jggomez v2\Carpeta
Empresas Abreviadas"),
Filtro = Table.SelectRows(Origen, each [Name] = " MiArchivo.xlsx "),
Validacion = if Table.RowCount(Filtro) = 1
    then Filtro{0}[Content]
    else error "Archivo no encontrado o hay múltiples coincidencias.",
Binario = Validacion
```

in

Binario

Para ello accedemos a la consulta archivo de ejemplo y adaptamos el codigo anterior tal y como aparece en la siguiente ilustración, sustituyendo simplemente MiArchivo.xlsx.

