



# Prácticas recomendadas para el diseño de libros de trabajo de Tableau eficaces

Edición para Tableau 10

Alan Eldridge  
Tableau Software

## Acerca de este documento

Una vez más, cabe aclarar que este documento es una síntesis del material escrito por diferentes autores. Básicamente, lo que hice fue recopilarlo en un solo informe y estructurar la información. Algunas de las personas que lean esto reconocerán su huella en varias secciones (de hecho, algunas reconocerán fragmentos enteros de texto). Estoy agradecido con todos, porque este documento existe gracias a su excelente trabajo continuo y a sus denuncias por infracción de derechos de autor.

También quisiera agradecer a la gran cantidad de personas que lo revisaron para cerciorarse de su precisión y legibilidad. El cuidado por el detalle y las explicaciones pacientes hicieron que este informe sea mucho más comprensible de lo que habría sido sin su intervención.

Este documento se actualizó para reflejar las funcionalidades de Tableau 10. Versiones futuras de Tableau proporcionarán características y funcionalidades nuevas que pueden cambiar algunas de estas recomendaciones.

¡Gracias!

Alan Eldridge  
Junio de 2016

## Demasiado largo para leer

A menudo, quienes leyeron o recomendaron este informe me han comentado que es demasiado largo. Mi respuesta es que es lo suficientemente largo como para abarcar un material tan amplio con la profundidad adecuada.

Sin embargo, este es el resumen de los puntos clave tratados en este documento:

- No hay soluciones mágicas para los libros de trabajo ineficaces. Comience echando un vistazo al registro de rendimiento para comprender en qué se está yendo el tiempo. ¿Consultas que tardan mucho en ejecutarse? ¿Demasiadas consultas? ¿Cálculos lentos? ¿Representaciones complejas? Utilice esta información para orientar sus esfuerzos en la dirección correcta.
- Las recomendaciones que aparecen en este documento son solo eso, recomendaciones. Si bien son prácticas recomendadas, debe comprobar si mejorarán el rendimiento en su caso particular. Muchas de ellas pueden depender de la estructura de sus datos y de la fuente de datos que esté utilizando (p. ej., un archivo sin formato, un sistema de administración de bases de datos relacionales [RDBMS] o una extracción de datos).
- Las extracciones son una forma sencilla y rápida de ejecutar con mayor velocidad la mayoría de los libros de trabajo.
- Cuanto más limpios estén sus datos y mejor concuerden con la estructura de sus consultas (es decir, que requieran menos preparación y manipulación), más rápido se ejecutarán sus libros de trabajo.
- La causa principal de que un dashboard se ejecute lentamente es un diseño defectuoso. En particular, cuando un solo dashboard contiene demasiados gráficos o se intenta mostrar muchos datos al mismo tiempo. Simplifique. Permita que sus usuarios exploren los datos con una profundidad incremental, en vez de intentar mostrarles toda la información para que después filtren.
- Trabaje solo con los datos que necesite (tanto en relación con los campos a los que hace referencia, como con la granularidad de los registros que devuelva). Esto permite que Tableau genere menos y mejores consultas, con una velocidad mayor, y reduzca la cantidad de datos que se deben transferir desde la fuente de datos al motor de Tableau. También reduce el tamaño de sus libros de trabajo, por lo que serán más fáciles de compartir y se abrirán más rápidamente.
- Asegúrese de utilizar filtros de manera eficaz para reducir los datos.
- Las cadenas y las fechas son lentas, mientras que los números y los valores booleanos son rápidos.

Por último, algunas de las recomendaciones de este documento solo tienen un impacto real en el caso de que trabaje con conjuntos de datos grandes o complejos. ¿Qué los define como grandes o complejos? Eso depende. Sin embargo, no será dañino seguir estas recomendaciones en todos sus libros de trabajo, puesto que nunca se sabe si sus datos incrementarán en algún momento futuro. La práctica hace al maestro.

## Contenido

Acerca de este documento .....	2
Demasiado largo para leer .....	3
Introducción .....	6
¿Para qué sirve Tableau? .....	6
¿Para qué no está preparado Tableau? .....	6
El concepto de eficacia .....	8
¿Qué es un libro de trabajo “eficaz”? .....	8
¿Por qué debería importarnos la eficacia? .....	8
Las leyes de la física .....	9
Caja de herramientas básica .....	11
Registro del rendimiento .....	11
Registros.....	13
Vistas de rendimiento en Tableau Server .....	14
Supervisión y pruebas .....	15
Otras herramientas .....	16
¿El problema es el diseño de mi libro de trabajo?.....	18
Un buen diseño de dashboard .....	18
Modifique el dashboard para mejorar el rendimiento .....	22
Un buen diseño de hoja de trabajo .....	27
Filtros eficaces.....	33
¿El problema son mis cálculos? .....	43
Tipos de cálculo.....	44
Análisis .....	48
Cálculos y características nativas .....	48
Impacto de los tipos de datos.....	49
Técnicas para mejorar el rendimiento .....	49
¿El problema son mis consultas? .....	55
Optimizaciones automáticas.....	55
Uniones .....	62
Combinaciones.....	62
Integración de datos .....	66
SQL personalizado.....	68
Alternativas a SQL personalizado.....	69
¿El problema son mis datos? .....	71
Consejos generales .....	71

Fuentes de datos.....	72
Preparación de datos.....	81
Extracciones de datos.....	81
Administración de datos.....	88
¿El problema es mi entorno?.....	90
Actualice.....	90
Pruebe Tableau Desktop en el servidor.....	90
Separe las actualizaciones y las cargas de trabajo interactivas.....	90
Supervise y ajuste su servidor.....	91
Infraestructura.....	91
Conclusión.....	93

## Introducción

### ¿Para qué sirve Tableau?

En Tableau, queremos cambiar la manera en que las personas ven y comprenden los datos y cómo interactúan con ellos. Como resultado, no intentamos ofrecer el mismo tipo de experiencia que ofrecen las plataformas de BI empresariales tradicionales. Tableau funciona mejor cuando se usa para crear libros de trabajo que sean:

- **Visuales:** hay muchísima evidencia que muestra que para los seres humanos la manera más eficaz de entender conjuntos de datos grandes y complejos es mediante representaciones visuales. El procedimiento predeterminado de Tableau consiste en presentar los datos a través de gráficos, diagramas y dashboards. Las tablas y las tabulaciones cruzadas tienen su lugar (y se admiten), y hablaremos sobre cómo aprovecharlas más adelante.
- **Interactivos:** los documentos de Tableau están diseñados para proporcionar resultados con los que los usuarios puedan interactuar, tanto en sus escritorios como en la web y en dispositivos móviles. A diferencia de otras herramientas de BI que producen, principalmente, resultados de impresión (ya sean para papel o para documentos de tipo PDF), el objetivo es diseñar experiencias útiles e interactivas que permitan a los usuarios explorar los datos y resolver consultas de negocios.
- **Iterativos:** el descubrimiento es un proceso cíclico por naturaleza. Tableau está diseñado para acelerar el ciclo de consulta-información-consulta para que los usuarios puedan desarrollar hipótesis rápidamente, probarlas con los datos disponibles, corregirlas, volverlas a probar y así sucesivamente.
- **Rápidos:** históricamente, el proceso de BI ha sido lento. Ha sido lento para instalar y configurar software, lento para poner los datos a disposición para el análisis y lento a los usuarios instalarlo, conectarse y crear documentos más rápido que nunca y, en muchos casos, reduce el tiempo de respuesta a las consultas de meses o semanas a horas o minutos.
- **Sencillos:** las herramientas de BI empresariales tradicionales suelen exceder las capacidades de la mayoría de los usuarios profesionales debido a su costo o a su complejidad. En muchos casos, los usuarios necesitan asistencia de TI o de algún usuario avanzado para crear las consultas y los documentos que desean. Tableau ofrece una interfaz intuitiva para que los usuarios sin conocimientos técnicos puedan hacer consultas y análisis de datos complejos sin necesidad de convertirse en expertos en bases de datos u hojas de cálculo.
- **Bonitos:** dicen que la belleza depende de los ojos con que se la mire, pero en el mundo de la comunicación visual existen prácticas recomendadas que deben respetarse. Mediante funciones como “Mostrarme”, Tableau guía a los usuarios sin conocimientos técnicos en la creación de gráficos eficaces y comprensibles basados en los datos que se usen.
- **Ubicuos:** cada vez es menos frecuente que los usuarios creen documentos para una única plataforma de trabajo. Ellos necesitan ver los datos e interactuar con estos en sus escritorios, en la web y en dispositivos móviles, además de insertarlos en otras aplicaciones y documentos, y más. Tableau permite publicar documentos y, después, usarlos en todas estas plataformas sin necesidad de migración ni rediseño.

### ¿Para qué no está preparado Tableau?

Tableau es una herramienta sofisticada y eficaz, pero se debe comprender desde el principio que probablemente no sea la mejor solución para resolver ciertos problemas. Esto no significa que no pueda usarse la herramienta: Tableau puede manipularse con el fin de que realice muchas tareas para las que no fue diseñado originalmente. Lo que queremos decir es que, si usa Tableau para

tratar de resolver problemas para los cuales la herramienta no está diseñada, la relación entre esfuerzo y recompensa probablemente no será favorable, y la solución resultante puede ser poco o nada satisfactoria.

Le sugerimos que piense en modificar sus requisitos o considere otro enfoque si:

- Necesita un documento que esté diseñado para el papel, no para la pantalla. Con esto nos referimos a los casos en los que necesite controlar diseños de página complejos, deba incluir características como encabezados o pies de página, sección o grupo, o requiera un formato WYSIWYG preciso. Tableau puede generar informes de varias páginas, pero estos carecen del nivel de control de formato que está disponible en las herramientas dedicadas de generación de informes con bandas.
- Necesita un mecanismo instantáneo y complejo para documentos personalizables (también llamado “envío en ráfaga”) que se envíen con distintos modos de entrega. En Tableau Server se incluye el concepto de suscripciones de informes, que permite que un usuario se suscriba a sí mismo (y, en Tableau 10, a otros) para recibir un informe por correo electrónico. Sin embargo, en ocasiones los clientes prefieren una solución más flexible. Se puede usar Tableau para crear otras formas de sistemas instantáneos, pero esta no es una característica nativa de Tableau. Se requiere el desarrollo de una solución personalizada, generada a partir de la utilidad TABCMD o mediante la incorporación de soluciones de terceros, como *VizAlerts* (<http://tabsoft.co/1stldFh>) o *Push Intelligence for Tableau* de Metric Insights (<http://bit.ly/1HACxul>).
- El caso de uso principal para el lector es exportar los datos en otro formato (frecuentemente un archivo CSV o de Excel). Por lo general, se requieren informes tabulares con muchas filas de datos detallados. Debe quedar claro que Tableau permite a los usuarios exportar los datos de una vista o un dashboard a Excel, tanto en resumen como de manera detallada. Sin embargo, cuando el caso de uso principal es exportar, el proceso consiste en extraer, transformar y cargar (ETL) copias. Existen soluciones mucho más eficaces que una herramienta de generación de informes para lograr esto.
- Necesita documentos de tabulación cruzada muy complejos que, quizás, imiten informes de hoja de cálculo existentes con cálculos de subtotalet difíciles, referencias cruzadas complicadas, etc. Los ejemplos más comunes de esto son los informes financieros, como estados de resultados, balances generales, etc. Además, puede tener la necesidad de realizar modelos de situaciones, análisis hipotéticos e, incluso, modificaciones de datos supuestos. Si los datos minuciosos subyacentes no están disponibles o si la lógica para generar informes se basa en “referencias de celdas”, en vez de combinar los registros y calcular totales, puede ser apropiado continuar usando una hoja de cálculo para este tipo de informes.

## El concepto de eficacia

### ¿Qué es un libro de trabajo “eficaz”?

Existen varios factores que hacen que un libro de trabajo sea “eficaz”. Algunos de estos factores son técnicos, y otros se centran más en el usuario. Pero, en general, un libro de trabajo eficaz es:

- **Simple:** ¿el libro de trabajo es fácil de crear? ¿Y será sencillo realizarle mantenimiento en el futuro? ¿Saca provecho de los principios de análisis visual para comunicar con claridad el mensaje del autor y de los datos?
- **Flexible:** a través del libro de trabajo, ¿se pueden contestar varias preguntas que el usuario quiera hacer? ¿O solo se puede contestar una? ¿Involucra al usuario en una experiencia interactiva o es simplemente un informe estático?
- **Rápido:** ¿el libro de trabajo proporciona respuestas lo suficientemente rápido para los usuarios? En esto están comprendidos los tiempos de apertura, actualización y respuesta a la interacción. Esta medida es subjetiva, pero, en general, se espera que el libro de trabajo proporcione una visualización inicial de la información y una respuesta a las interacciones del usuario en pocos segundos.

El rendimiento de un dashboard se ve afectado por:

- El diseño visual tanto en el nivel del dashboard como en el de las hojas de trabajo. Por ejemplo, la cantidad de elementos, la cantidad de puntos de datos, la utilización de filtros y acciones, etc.
- Los cálculos; por ejemplo, el tipo de cálculo, el lugar en el que se ejecuta el cálculo, etc.
- Las consultas; por ejemplo, la cantidad de datos que se devuelven, si es SQL personalizado, etc.
- Las conexiones de datos y las fuentes de datos subyacentes.
- Algunas diferencias entre Tableau Desktop y Tableau Server.
- Otros factores del entorno, como la configuración y la capacidad del hardware.

### ¿Por qué debería importarnos la eficacia?

La eficacia es importante por varias razones:

- Cuando un analista o autor del libro de trabajo trabaja de manera eficaz, se llega a la respuesta con mayor velocidad.
- Cuando trabaja de manera eficaz, se puede concentrar en un “flujo” continuo de análisis. Esto implica pensar en las preguntas y en los datos, en vez de distraerse con cómo debe manipular la herramienta para conseguir un resultado.
- La creación de un libro de trabajo con un diseño flexible disminuye la necesidad de crear múltiples libros de trabajo y mantenerlos para satisfacer requisitos similares.
- Con la creación de un libro de trabajo de diseño sencillo, es más fácil para otras personas tomar el libro de trabajo que usted creó y realizar iteraciones adicionales en su trabajo.
- La forma en que se percibe la capacidad de respuesta es un factor de éxito importante para los usuarios finales a la hora de ver informes y dashboards. Por lo que, para hacer más felices a los usuarios, es necesario lograr que los libros de trabajo funcionen lo más rápido posible.

En nuestra vasta experiencia, observamos que la mayoría de los problemas de rendimiento se deben a los errores en el diseño del libro de trabajo. Si podemos corregir estos errores (o, mejor aún, prevenirlos a través de la capacitación), podemos solucionar los problemas.

Cuando se trabaja con volúmenes de datos pequeños, muchas de estas recomendaciones no son cruciales. Puede superar el problema a la fuerza. Sin embargo, cuando tiene cientos de millones



de registros, diversos libros de trabajo o varios autores, se amplifican los efectos de un diseño deficiente. En esta situación, se hace necesario considerar con mayor detenimiento las indicaciones de este informe.

Por supuesto, la práctica hace al maestro; por eso, se recomienda seguir estas pautas en todos los libros de trabajo. Recuerde que el diseño no está completo hasta que haya probado el libro de trabajo con los volúmenes de datos que se espera tener en la producción.

Una nota importante: en todo este documento, hacemos referencia a Tableau Server. Sin embargo, en la mayoría de los casos, las pautas también son adecuadas para Tableau Online si es que prefiere utilizar nuestra solución hospedada, en vez de una implementación en las instalaciones. Las excepciones obvias son las pautas que versan sobre el ajuste de los parámetros de la configuración del servidor y la instalación o actualización del software en el nivel de servidor (en el mundo del software como servicio [SaaS], no se tiene que preocupar por estas cuestiones).

## Las leyes de la física

Antes de sumergirnos en los detalles técnicos de las distintas características que afectan el rendimiento de los libros de trabajo, presentamos algunos principios básicos que lo ayudarán a crear dashboards e informes eficaces:

### Si funciona lento en la fuente de datos, también será lento en Tableau

Si el libro de trabajo de Tableau se basa en una consulta que se ejecuta lentamente, su libro de trabajo también será lento. En las secciones siguientes ofreceremos sugerencias para ajustar sus bases de datos, a fin de contribuir a reducir el tiempo de ejecución de las consultas. Además, analizaremos cómo usar el veloz motor de datos de Tableau para mejorar el rendimiento de las consultas.

### Si funciona lento en Tableau Desktop, (casi siempre) funcionará lento en Tableau Server

Los libros de trabajo que sean lentos en Tableau Desktop no serán más rápidos si los publica en Tableau Server. A menudo, los usuarios tienen la sensación de que los libros de trabajo se ejecutarán más rápido en Tableau Server, porque el servidor tiene más CPU/RAM/etc. que su equipo local. En general, los libros de trabajo funcionarán un poco más lento en Tableau Server, por los siguientes motivos:

- Hay varios usuarios que comparten de manera simultánea los recursos del servidor para generar libros de trabajo. (Aunque parezca contradictorio, es posible que note que el libro de trabajo responda con mayor velocidad a medida que se comparte; esto se debe a los mecanismos de caché de Tableau Server).
- El servidor debe hacer trabajo extra para generar los dashboards y los gráficos, a diferencia de cuando esto se realiza en la estación de trabajo del cliente.

Antes de comenzar a ajustar el rendimiento en Tableau Server, debe orientar sus esfuerzos al ajuste del libro de trabajo en Tableau Desktop.

La excepción a esta regla es cuando Tableau Desktop tiene límites de recursos que no se encuentran en el servidor. Por ejemplo, cuando el equipo no tiene memoria RAM suficiente para procesar el volumen de datos que está analizando o el servidor tiene una latencia de conexión con la fuente de datos más rápida o menor. Algunos usuarios se topan con un rendimiento reducido o, incluso, con errores por “falta de memoria” cuando trabajan con conjuntos de datos en estaciones de trabajo con especificaciones básicas de 2 GB de RAM. Sin embargo, consideran que el funcionamiento del libro de trabajo publicado es aceptablemente rápido porque el servidor tiene mucha más memoria y poder de procesamiento.

### Lo nuevo es mejor

El equipo de desarrollo de Tableau trabaja continuamente para mejorar el rendimiento y la capacidad de uso de nuestro software. En algunas ocasiones, la actualización a la última versión de Tableau Desktop y Tableau Server puede producir mejoras significativas en el rendimiento, sin necesidad de implementar cambios en el libro de trabajo. Por ejemplo, muchos clientes comunicaron que el rendimiento de sus libros de trabajo era tres veces mejor (o más) después de actualizar de la versión 8 a la 9. La mejora en el rendimiento continúa siendo un objetivo para la versión de Tableau 10 y las posteriores. Por supuesto, esto no es un inconveniente en Tableau Online, puesto que siempre está actualizado a la última versión publicada.

Esto se aplica tanto a las versiones de mantenimiento como a las versiones principales y secundarias. Puede obtener más información acerca del ciclo de versiones de Tableau y los detalles específicos de cada versión en la página de notas de la versión:

<http://www.tableau.com/es-es/support/releases>

Además, los proveedores de bases de datos están trabajando para mejorar sus productos. Asegúrese también de estar utilizando la última versión del controlador de fuente de datos correspondiente, según aparecen en la página web siguiente:

<http://www.tableau.com/es-es/support/drivers>

### Menos es más

Como en todos los aspectos de la vida, algo bueno en exceso puede ser malo. No intente incluir absolutamente todo en un libro de trabajo único y monolítico. Si bien un libro de trabajo de Tableau *puede* tener 50 dashboards, cada uno con 20 objetos gráficos, y comunicarse con 50 fuentes de datos diferentes, casi con seguridad será lento.

Si se encuentra con un libro de trabajo de este tipo, considere dividirlo en varios archivos distintos. Es fácil de hacer. Simplemente tiene que copiar los dashboards entre los libros de trabajo y Tableau traerá todas las hojas de trabajo y fuentes de datos asociadas. Si sus dashboards son extremadamente complejos, considere simplificarlos y usar interacciones para guiar a los usuarios finales de un informe a otro. Recuerde que no fijamos el precio de nuestro software por cantidad de documentos, así que distribuya los datos con tranquilidad.

### La escalabilidad no es lo mismo que el rendimiento

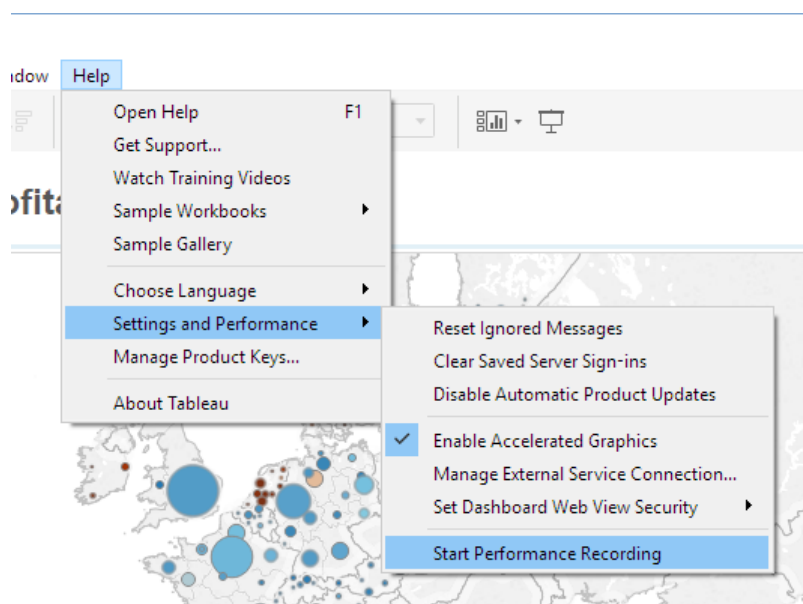
La escalabilidad implica garantizar que múltiples usuarios puedan ver libros de trabajo compartidos. El rendimiento implica garantizar que un libro de trabajo se ejecute lo más rápido posible. A pesar de que muchas de las recomendaciones que se ofrecen en este documento tienen una influencia positiva en la escalabilidad de los libros de trabajo publicados en Tableau Server, el objetivo principal de este documento es mejorar el rendimiento.

## Caja de herramientas básica

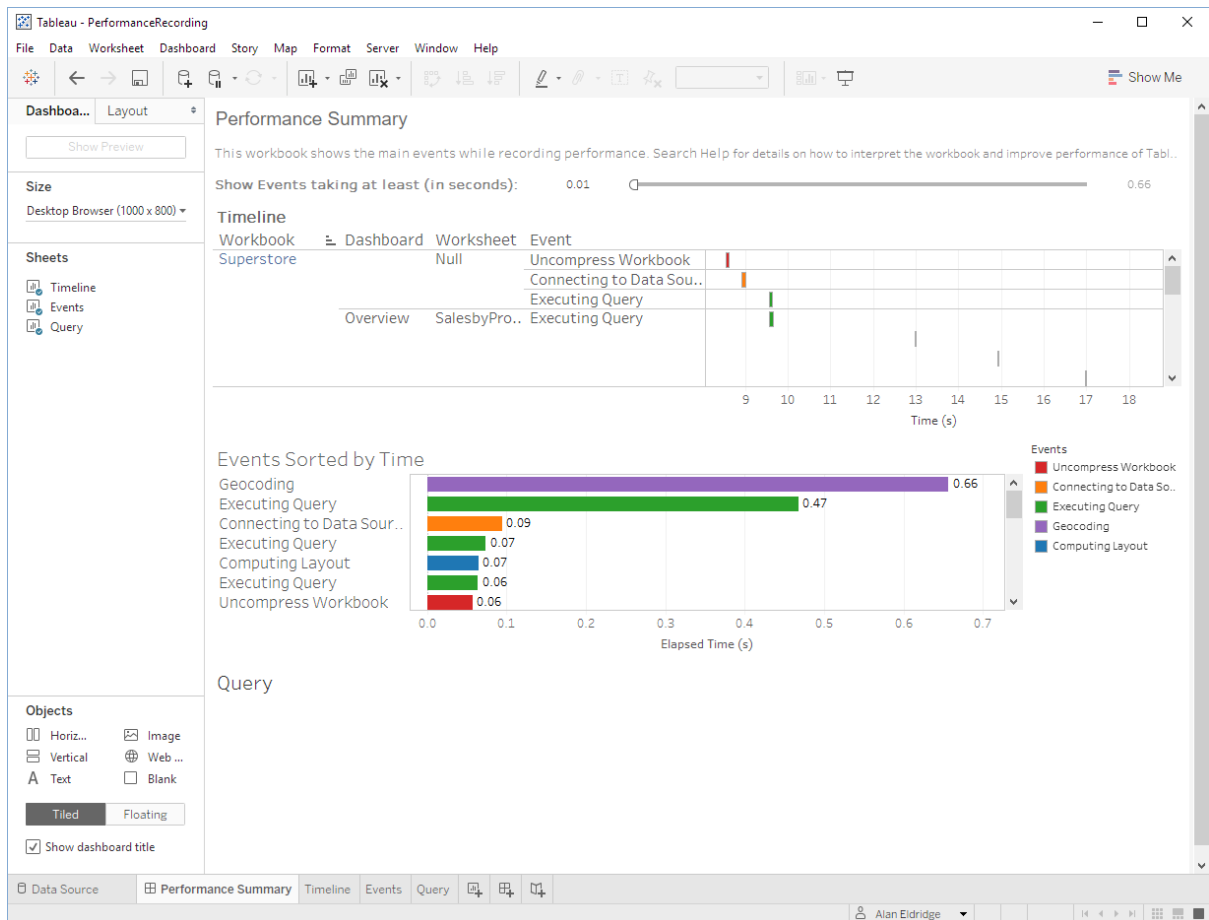
Para poder comprender el rendimiento de sus libros de trabajo, primero debe saber a) lo que está sucediendo y b) cuánto se tarda en hacerlo. Esta información se registra en varios lugares, dependiendo de dónde esté ejecutando el libro de trabajo (es decir, si utiliza Tableau Desktop o Tableau Server), y con diversos niveles de detalle. En esta sección se presentan las distintas opciones disponibles.

### Registro del rendimiento

El primer lugar en el que debería consultar la información de rendimiento es la funcionalidad Registro del rendimiento de Tableau Desktop y Tableau Server. En Tableau Desktop, esta funcionalidad se activa desde el menú Ayuda:



Inicie Tableau y comience a registrar el rendimiento. Luego, abra el libro de trabajo (la práctica recomendada es no tener otros libros de trabajo abiertos mientras hace esto, para que no compitan involuntariamente por los recursos). Interactúe con él como si usted fuera un usuario final y, cuando sienta que recopiló suficientes datos, vuelva al menú Ayuda y detenga el registro. Se abrirá otra ventana de Tableau Desktop con los datos recopilados:



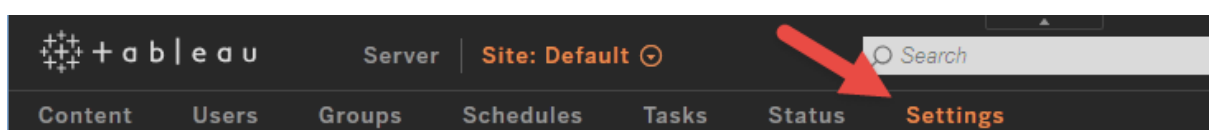
Ahora puede identificar las acciones del libro de trabajo que llevan más tiempo; por ejemplo, en la imagen de arriba, la consulta seleccionada de la hoja de trabajo Escala de tiempo tarda 30,66 segundos en completarse. Si hace clic en la barra, verá el texto de la consulta que se está ejecutando. Puesto que el resultado del registro del rendimiento es un libro de trabajo de Tableau, también puede crear vistas adicionales para explorar esta información de otras formas:

*Nota: de manera predeterminada, no se muestran los eventos que tardan menos de 0,1 segundos. Para incluirlos, ajuste el filtro de la parte superior del dashboard. Sin embargo, debería concentrarse en las tareas con tiempos de ejecución mayores. Una buena práctica es configurarlo en 1 segundo.*

También es posible crear registros del rendimiento en Tableau Server con el fin de identificar los problemas que surgen cuando se publica el libro de trabajo. De manera predeterminada, el registro del rendimiento no está habilitado en Tableau Server. Es una funcionalidad que se puede controlar para cada sitio.

Un administrador del servidor puede habilitar el registro del rendimiento para cada uno de los sitios.

1. Navegue hasta encontrar el sitio para el cual desea habilitar el registro del rendimiento.
2. Haga clic en **Configuración**:



3. En Métricas de rendimiento de libro de trabajo, seleccione **Registrar métricas de rendimiento de libro de trabajo**.
4. Haga clic en **Guardar**.

Cuando desee crear un registro del rendimiento:

1. Abra la vista para la cual desea crear un registro del rendimiento: Cuando abre una vista, Tableau Server agrega “:iid=<n>” a continuación de la URL. Es un ID de sesión. Por ejemplo:

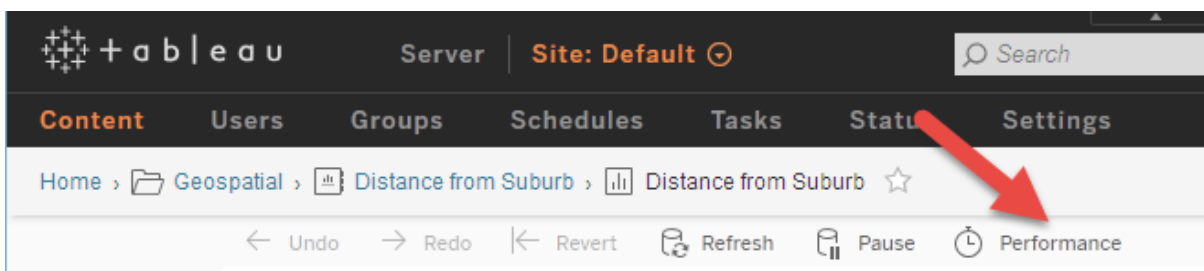
```
http://<tableau_server>/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?iid=1
```

2. Escriba `:record_performance=yes&` al final de la URL de vista, inmediatamente antes del ID de sesión. Por ejemplo:

```
http://<tableau_server>/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?record_performance=yes&iid=1
```

3. Cargue la vista.

La opción **Rendimiento** en la barra de herramientas de la vista es una confirmación visual de que se está ejecutando el registro del rendimiento:



Cuando haya terminado y esté listo para ver el registro del rendimiento:

1. Haga clic en **Rendimiento** para abrir un libro de trabajo de rendimiento. Es una instantánea en tiempo real de los datos de rendimiento. Puede continuar tomando instantáneas adicionales a medida que continúa trabajando con la vista. Los datos de rendimiento son acumulativos.
2. Pase a una página diferente o quite `:record_performance=yes` de la URL para detener el registro.

Debe usar esta información para identificar las secciones del libro de trabajo que más se presten a revisión, es decir, dónde puede obtener la mejora más significativa con respecto al tiempo que emplea. Puede encontrar más información sobre la interpretación de estos registros en el siguiente enlace:

[http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#perf\\_record\\_create\\_desktop.html](http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#perf_record_create_desktop.html)

## Registros

En Tableau puede encontrar el texto completo de la consulta en el archivo de registro. La ubicación predeterminada es `C:\Users\\Documents\My Tableau Repository\Logs\log.txt`. Este archivo tiene mucho texto y está escrito en lenguaje codificado JSON, por lo que se recomienda utilizar un buen editor de texto, como Notepad++ o Sublime, para poder interpretarlo. Si hace una búsqueda con el texto “begin-query” o “end-query”, podrá encontrar la cadena de consulta transmitida a la fuente de datos. El registro de “end-query” también le mostrará el tiempo que tardó en ejecutarse

la consulta y cuántos registros fueron devueltos a Tableau:

```
{ "ts": "2015-05-24T12:25:41.226", "pid": 6460, "tid": "1674", "sev": "info", "req": "-", "sess": "-", "site": "-", "user": "-", "k": "end-query", "v": { "protocol": "4308fb0", "cols": 4, "query": "SELECT [DimProductCategory].[ProductCategoryName] AS [none:ProductCategoryName:nk], \n [DimProductSubcategory].[ProductSubcategoryName] AS [none:ProductSubcategoryName:nk], \n SUM(CAST([FactSales].[ReturnQuantity]) as BIGINT) AS [sum:ReturnQuantity:ok], \n SUM([FactSales].[SalesAmount]) AS [sum:SalesAmount:ok] \n FROM [dbo].[FactSales] \n INNER JOIN [dbo].[DimProduct] [DimProduct] ON ([FactSales].[ProductKey] = [DimProduct].[ProductKey]) \n INNER JOIN [dbo].[DimProductSubcategory] [DimProductSubcategory] ON ([DimProduct].[ProductSubcategoryKey] = [DimProductSubcategory].[ProductSubcategoryKey]) \n INNER JOIN [dbo].[DimProductCategory] [DimProductCategory] ON ([DimProductSubcategory].[ProductCategoryKey] = [DimProductCategory].[ProductCategoryKey]) \n GROUP BY [DimProductCategory].[ProductCategoryName], \n [DimProductSubcategory].[ProductSubcategoryName]", "rows": 32, "elapsed": 0.951 } }
```

Si está realizando la búsqueda en Tableau Server, los registros se encuentran en C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Logos.

## Vistas de rendimiento en Tableau Server

Tableau Server viene con varias vistas para administradores, con el fin de ayudar a supervisar la actividad en Tableau Server. Las vistas se ubican en la tabla Análisis en la página Mantenimiento del servidor:

Analysis	
Dashboards that monitor Tableau Server activity.	
Dashboard	Analysis
Traffic to Sheets	Usage and users for published sheets.
Traffic to Data Sources	Usage and users for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	Sheet load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.
Server Disk Space	Current and historical disk space usage, by server node.
Tableau Desktop License Usage	Summary of usage for Tableau Desktop licenses
Tableau Desktop License Expirations	Expiration information for Tableau Desktop licenses

Puede encontrar más información sobre estas vistas en el siguiente enlace:

<http://onlinehelp.tableau.com/current/server/es-es/adminview.htm>

Además, puede crear vistas administrativas personalizadas conectándose a la base de datos PostgreSQL que forma parte del repositorio de Tableau. Puede encontrar instrucciones aquí:

[http://onlinehelp.tableau.com/current/server/es-es/adminview\\_postgres.htm](http://onlinehelp.tableau.com/current/server/es-es/adminview_postgres.htm)

## Supervisión y pruebas

### TabMon

TabMon es un supervisor de clústeres de código abierto para Tableau Server, que le permite recopilar en el tiempo estadísticas de rendimiento. TabMon cuenta con el soporte de la comunidad. Publicaremos el código fuente completo bajo licencia de código abierto del MIT.

TabMon inmediatamente lleva un registro del estado del sistema y las métricas de las aplicaciones. Recopila métricas incorporadas, como las que proporcionan los contadoresd Windows Perfmon, Java Health y Java Mbean (JMX), en los equipos de toda la red en los que se ejecuta Tableau Server. Puede utilizar TabMon para supervisar los componentes físicos (CPU, RAM), la red y el uso del disco duro. Puede llevar un registro de la frecuencia de aciertos de caché, la latencia de solicitud, las sesiones activas y mucho más. Los datos se muestran con una estructura clara y unificada, lo que facilita su visualización en Tableau Desktop.

TabMon le proporciona un control total sobre las métricas que desea recopilar y los equipos que quiere supervisar, sin necesidad de scripts ni código. Todo lo que necesita es el nombre del equipo y el de la métrica. TabMon se puede ejecutar de manera remota y con independencia del clúster. Puede supervisar, agregar y analizar el estado de los clústeres desde cualquier equipo de la red, casi sin agregarle carga a los equipos de producción.

Puede encontrar más información sobre TabMon aquí:

<http://bit.ly/1ULFelf>

### TabJolt

TabJolt es una herramienta para pruebas de rendimiento y carga diseñada para trabajar fácilmente con Tableau Server. A diferencia de las herramientas tradicionales de prueba de carga, TabJolt puede impulsar la carga de manera automática a la instancia de Tableau Server sin necesidad de desarrollar scripts ni mantenerlos. Gracias a que TabJolt conoce el modelo de presentación de Tableau, puede cargar visualizaciones de manera automática e interpretar las interacciones posibles durante la ejecución de la prueba.

Con solo indicar a TabJolt la ubicación de uno o más libros de trabajo en su servidor, cargará y ejecutará interacciones automáticamente en las vistas de Tableau. Además recopila métricas clave, como el tiempo de respuesta promedio, la capacidad de proceso, el tiempo de respuesta en el percentil 95, y captura las métricas de rendimiento de Windows como punto de comparación.

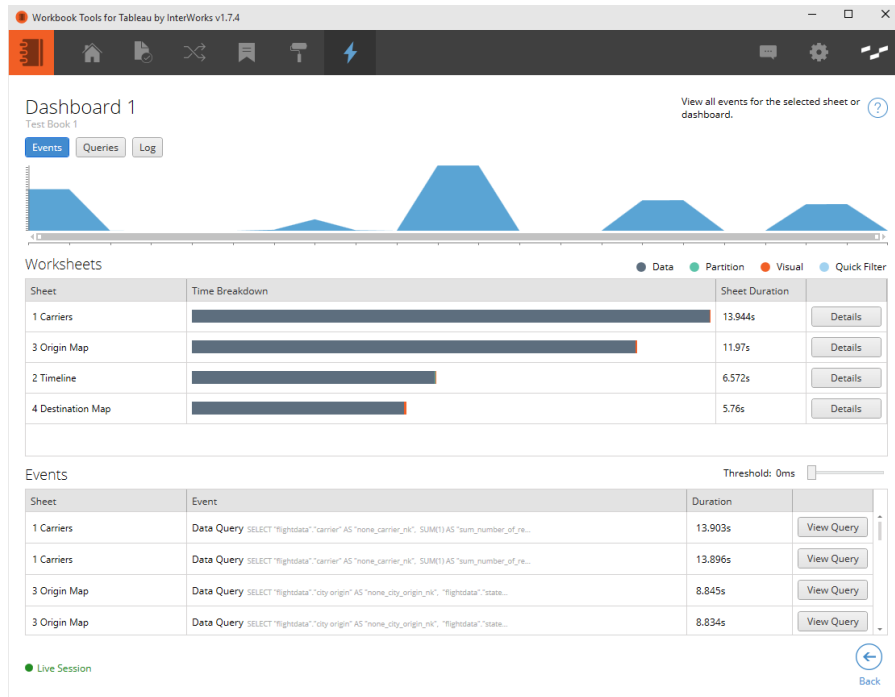
Por supuesto, incluso con TabJolt, los usuarios deben tener el conocimiento suficiente de la arquitectura de Tableau Server. No se recomienda tratar la instancia de Tableau Server como una caja negra en las pruebas de carga. Muy probablemente, devuelva resultados no acordes con sus expectativas. TabJolt es una herramienta automática y no puede replicar fácilmente la variedad de interacciones humanas. Por este motivo, debe considerar si los resultados de TabJolt se ajustan a la realidad.

Puede encontrar más información sobre TabJolt aquí:

<http://bit.ly/1ULFtgi>

## Otras herramientas

Hay otras herramientas disponibles de terceros que lo pueden ayudar a identificar las características de rendimiento de sus libros de trabajo. Una de las opciones es “Power Tools for Tableau” de Interworks, que incluye un analizador de rendimiento (similar al registro del rendimiento incorporado) que le permite explorar las hojas y consultas que llevan más tiempo y comprender los motivos.



Palette Software también cuenta con un producto, denominado Palette Insight, que captura la información de rendimiento de Tableau Server y le permite planear la capacidad, identificar usuarios y libros de trabajo que consumen muchos recursos, auditar los accesos de los usuarios y generar modelos de contracargos.





Además, las plataformas más modernas de sistema de administración de bases de datos (DBMS) incluyen herramientas de administración que permiten llevar un registro de las consultas que ejecutan y analizarlas. Si sus registros del rendimiento indican que el tiempo de ejecución de las consultas es un factor clave, su administrador de bases de datos puede ser de gran ayuda.

Si cree que el problema es la interacción con el navegador del cliente y el servidor, también puede probar con herramientas como Telerik Fiddler o las herramientas para el desarrollador de su navegador. Su empleo le puede proporcionar una visión más detallada del tráfico entre el cliente y el servidor.

## ¿El problema es el diseño de mi libro de trabajo?

Trabajar con Tableau es una experiencia nueva para muchos usuarios, y existen técnicas y prácticas recomendadas de diseño que estos deben aprender para crear libros de trabajo eficaces. Sin embargo, hemos notado que muchos usuarios nuevos intentan aplicar enfoques de diseño antiguos con Tableau y obtienen resultados deslucidos. El propósito de esta sección es revisar los principios de diseño que reflejan las prácticas recomendadas.

### Un buen diseño de dashboard

Con Tableau, usted crea una experiencia interactiva para sus usuarios finales. El resultado final que proporciona Tableau Server es una aplicación interactiva que permite a los usuarios explorar los datos, en lugar de solo verlos. Por lo tanto, para crear un dashboard de Tableau eficaz, necesita dejar de pensar como si estuviera desarrollando un informe estático.

Veamos un ejemplo de un tipo de dashboard que muchos novatos crean, especialmente si antes trabajaron con herramientas como Excel o Access, o si tienen experiencia en el uso de herramientas “tradicionales” de generación de informes. Comenzamos con un informe tabular que muestra “todo” y una serie de filtros que permiten al usuario restringir la tabla hasta que muestre los registros que le interesan:

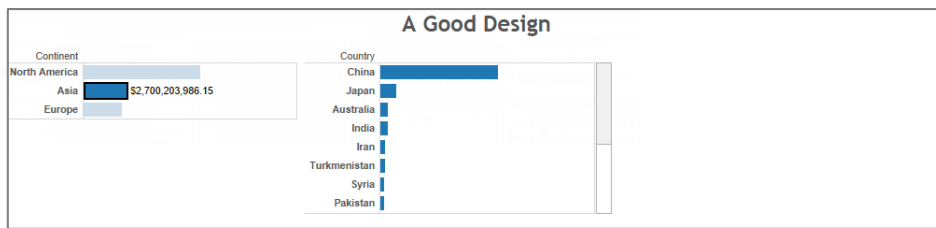
A Bad Design															
Continent	Country	State/Province	Product Category	Product Subcate..	Product Name	Sales Qty	Total Cost	Sales Amou..	Continent	Country	City	Product Category	Product Subcategory	Product Name	
Asia	Turkmenistan	Ahal Province	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	64	\$5,547.52	\$11,790.68	<input checked="" type="checkbox"/>	Asia		<input checked="" type="checkbox"/>	Audio	<input checked="" type="checkbox"/>	A. Datum Advanced Digi.
North America	United States	Alaska	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	62	\$5,200.80	\$11,536.20	<input checked="" type="checkbox"/>	Europe		<input checked="" type="checkbox"/>	Cameras and camcorders	<input checked="" type="checkbox"/>	A. Datum Advanced Digi.
North America	Canada	Alberta	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	40	\$3,467.20	\$7,540.00	<input checked="" type="checkbox"/>	North America		<input checked="" type="checkbox"/>	Cell phones	<input checked="" type="checkbox"/>	A. Datum Advanced Digi.
Europe	France	Alpes-Maritim.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	31	\$2,687.08	\$5,734.17	<input type="checkbox"/>		Natl	<input checked="" type="checkbox"/>	Computers	<input checked="" type="checkbox"/>	A. Datum Advanced Digi.
Asia	Armenia	Armenia	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	19	\$1,560.24	\$3,468.40	<input type="checkbox"/>		Alexandria	<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	France	Bas-Rhin	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	36	\$3,033.80	\$6,710.60	<input type="checkbox"/>		Amsterdam	<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Germany	Bavaria	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	60	\$5,114.12	\$10,819.90	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	China	Beijing	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	2,276	\$194,683.28	\$421,825.30	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Germany	Berlin	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	1,193	\$102,022.36	\$220,330.11	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Switzerland	Bern	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	38	\$3,207.16	\$6,936.80	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
North America	Canada	British Colum.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	107	\$9,274.76	\$20,075.25	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Romania	Bucuresti	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	10	\$866.80	\$1,885.00	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Greece	Central Greec.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	18	\$1,560.24	\$3,317.60	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Japan	Chubu	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	13	\$1,040.16	\$2,337.40	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Kyrgyzstan	Chuy Province	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	66	\$5,547.52	\$12,280.78	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
North America	United States	Colorado	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	699	\$59,635.84	\$130,093.28	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
North America	United States	Connecticut	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	293	\$25,050.52	\$54,533.05	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Syria	Damascus	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	104	\$8,928.04	\$19,311.83	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	United Kingdo.	England	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	478	\$40,912.96	\$88,702.45	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
North America	United States	Florida	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	423	\$35,885.52	\$78,745.88	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Germany	Hesse	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	26	\$2,253.68	\$4,674.80	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Japan	Hokkaido	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	22	\$1,906.96	\$4,109.30	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	China	Hong Kong	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	111	\$9,448.12	\$20,574.78	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Pakistan	Islamabad Ca.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	63	\$5,287.48	\$11,724.70	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Japan	Kansai	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	119	\$10,228.24	\$22,158.18	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Japan	Kanto	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	176	\$15,255.68	\$32,648.20	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Asia	Thailand	Krung Thep	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	89	\$7,541.16	\$16,522.03	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.
Europe	Ireland	Leinster	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	31	\$2,687.08	\$5,824.65	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	A. Datum Advanced Digi.

Este no es un “buen” dashboard de Tableau (de hecho, no es nada “bueno”). En el peor de los casos, es un proceso de extracción de datos glorificado porque el usuario desea llevar los datos a otra herramienta, como Excel, para hacer análisis y gráficos adicionales. En el mejor de los casos, indica que, en realidad, no entendemos cómo el usuario final desea explorar los datos, de modo que tomamos el siguiente enfoque: “En función de su criterio inicial, le proporcionamos todo... e incluimos algunos filtros para que pueda restringir el conjunto de resultados y encontrar lo que realmente busca”.

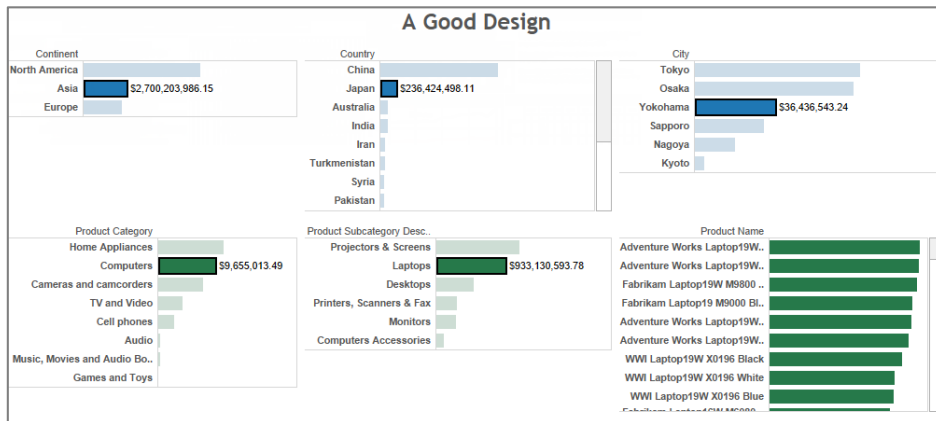
Ahora, considere el siguiente reprocesamiento con los mismos datos. Comenzamos con el nivel de agregación más alto:



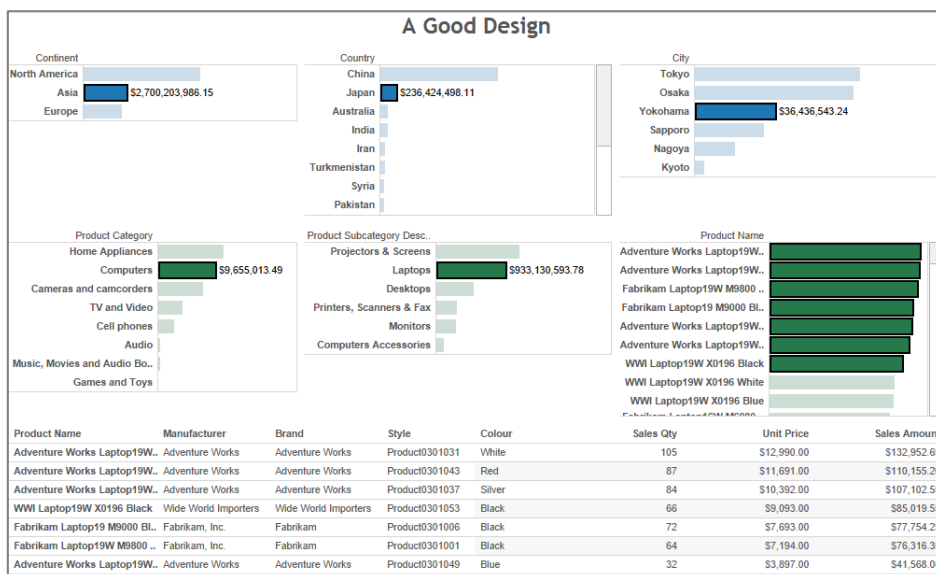
Si seleccionamos uno o más elementos, se muestra el siguiente nivel de detalle:



Continuamos haciendo esto y revelamos cada vez más detalles:



Finalmente, llegamos a revelar el máximo nivel de datos, los mismos datos que se mostraron en el dashboard de tabulación cruzada de arriba.



No se detenga en la presentación de los datos (es un tema importante, pero lo trataremos más adelante). En lugar de eso, reflexione acerca de la experiencia de usar este dashboard. Observe cómo fluye naturalmente, de izquierda a derecha, de arriba a abajo. Este ejemplo puede tener muchos datos subyacentes, pero el dashboard guía al usuario final a explorar en profundidad y de manera gradual para que pueda encontrar el conjunto de registros detallados y precisos que busca.

La diferencia principal de los dos ejemplos es el modo en que se guía al usuario final en el proceso de análisis. El primer ejemplo arroja un resultado inicial extenso (muestra todos los registros posibles)

y permite al usuario final reducir la cantidad de registros que se muestran mediante la aplicación de filtros. Pero existen problemas inherentes a esta técnica:

- La consulta inicial que debe ejecutarse antes de que se muestre algo al usuario final es, básicamente, la consulta más general que se puede hacer: “quiero todos los registros”. Con cualquier conjunto de datos real, ejecutar esta consulta y transmitirla de vuelta al motor de Tableau llevaría un tiempo considerable. La experiencia del “contacto inicial” es fundamental para establecer la percepción del usuario final con respecto a la solución; y si transcurren más de unos pocos segundos antes de que algo suceda, la percepción será negativa.
- Crear una vista con cientos de miles o millones de marcas (cada celda de una tabulación cruzada se llama marca) implica una gran exigencia sobre la CPU y la memoria. También demanda tiempo, lo que contribuye a la percepción negativa de la capacidad de respuesta del sistema. En Tableau Server, el hecho de que muchas personas generen tabulaciones cruzadas grandes al mismo tiempo puede provocar un rendimiento reducido y, en el peor de los casos, agotar la memoria del sistema. Esto puede generar problemas de estabilidad en el servidor, errores y todo tipo de situaciones desagradables para los usuarios finales. Por supuesto, puede agregar más memoria al servidor para minimizar esto, pero estaría tratando el síntoma, no la causa.
- Por último, los usuarios no tienen ninguna guía contextual que les indique si su conjunto inicial de filtros producirá resultados muy extensos o muy reducidos. ¿Cómo puede saber el usuario de un informe que si selecciona todas las categorías disponibles, la consulta inicial producirá decenas de miles de registros y agotará toda la memoria RAM disponible del servidor? No puede saberlo si no lo experimenta.

Compare esto con el segundo enfoque, en el que la consulta inicial muestra únicamente el nivel más alto de agregación:

- La primera consulta que se debe ejecutar es altamente agregada y, en consecuencia, proporciona solo una cantidad mínima de datos. Para las bases de datos bien diseñadas, esta es una actividad muy eficaz, y el tiempo de respuesta del “primer contacto” es muy rápido, lo que conduce a una percepción positiva del sistema. A medida que profundizamos, cada consulta posterior se agrega y restringe por las selecciones del nivel más alto. Estas continúan ejecutándose y regresando al motor de Tableau rápidamente.
- Aunque contamos con más vistas cuando el dashboard está completo, cada vista muestra solo unas pocas decenas de marcas. Incluso si muchos usuarios finales están usando el sistema, los recursos necesarios para generar cada una de estas vistas son insignificantes, y es mucho menos probable que el sistema se quede sin memoria.
- Por último, se puede observar que aprovechamos la oportunidad para mostrar el volumen de ventas de cada categoría en los niveles de “navegación” más altos. Esto proporciona algo de contexto al usuario acerca de la cantidad de registros que incluye la selección. También indicamos la rentabilidad de cada categoría con colores. Esto es extremadamente relevante, ya que le permite ver qué áreas requieren de su atención y evita que tenga que navegar a ciegas.

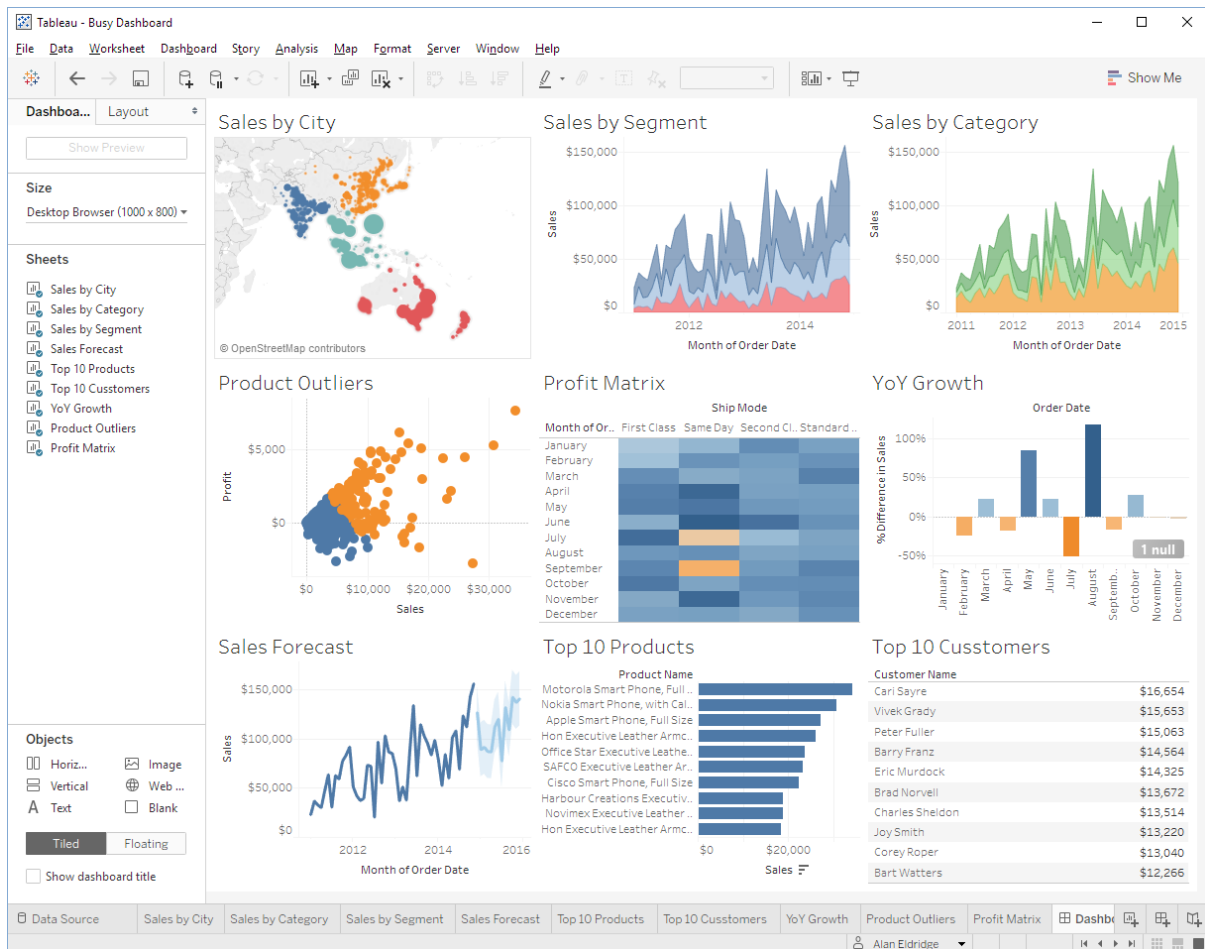
### Simplifique

Un error común de los usuarios nuevos es crear dashboards demasiado “complejos”. Quizás estén tratando de recrear un documento que usaron anteriormente en otra herramienta o, probablemente, estén intentando crear algo diseñado específicamente para ser impreso. El resultado final es un libro de trabajo que se ejecuta lentamente y es ineficaz.

Los puntos a continuación contribuyen a la complejidad:

### Demasiadas hojas de trabajo por dashboard

Un error común de los usuarios nuevos es agregar demasiados gráficos u hojas de trabajo en un solo dashboard.



Tenga en cuenta que cada hoja de trabajo ejecutará una consulta (y, probablemente, más de una) en las fuentes de datos. Por ese motivo, a mayor cantidad de hojas, mayor será el tiempo que se tarde en generar el dashboard. Aproveche que Tableau está diseñado para proporcionar dashboards interactivos a usuarios finales y disemine los datos entre múltiples dashboards y páginas.

### Demasiadas tarjetas de filtros

Las tarjetas de filtros son una funcionalidad muy eficaz, que nos permite crear dashboards ricos e interactivos para los usuarios finales. Sin embargo, es posible que cada filtro deba ejecutar una consulta para enumerar las opciones. Por eso, agregar demasiados filtros a su dashboard puede ralentizar inesperadamente la representación de la vista. Además, cuando utiliza la opción “mostrar valores relevantes” en un filtro, se debe ejecutar una consulta para actualizar los valores mostrados cada vez que se modifican otros filtros. Use esta funcionalidad con moderación.

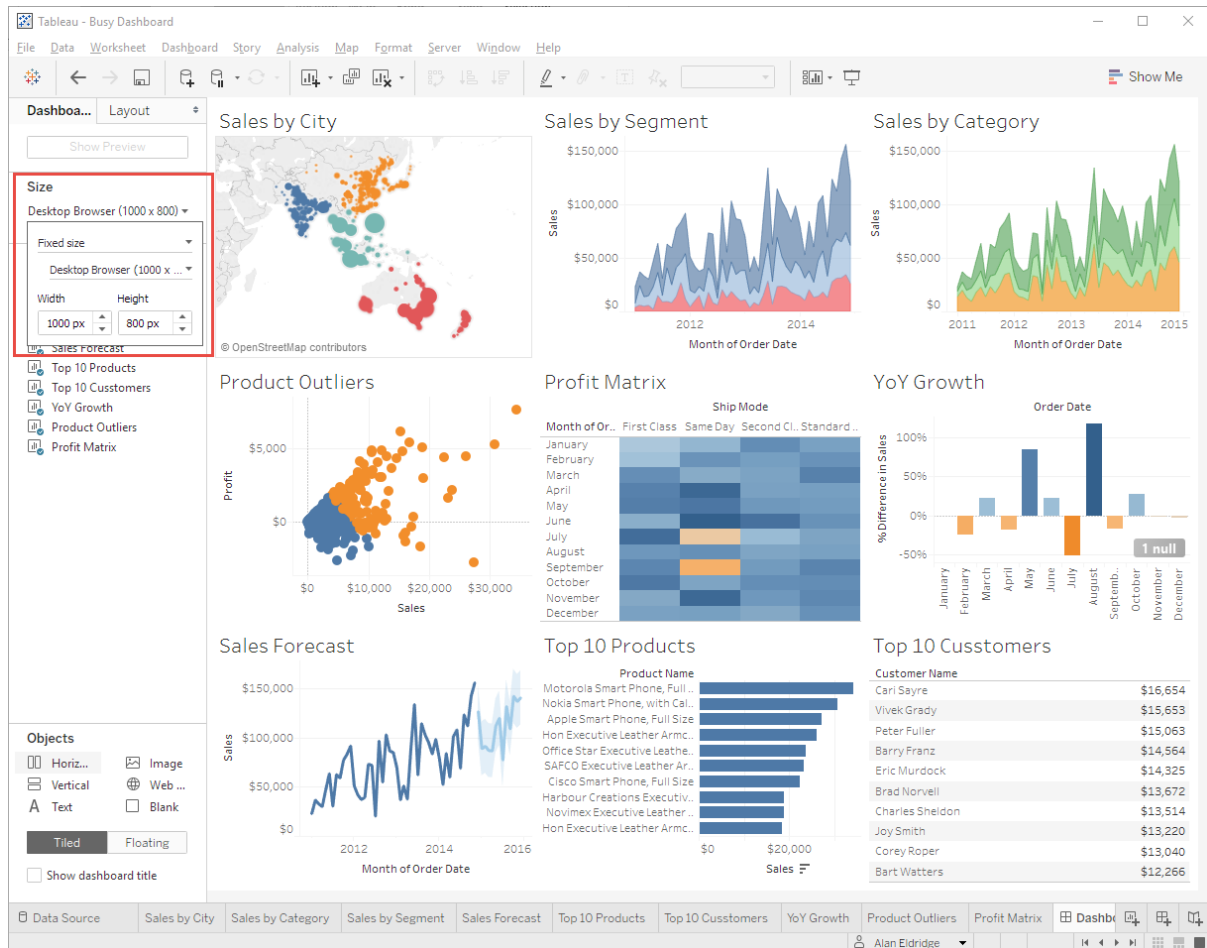
Además, si aplicó filtros a múltiples hojas de trabajo, tenga en cuenta que cada cambio generará varias consultas, debido a que se actualizarán todas las hojas de trabajo afectadas y visibles (las hojas de trabajo no visibles no se ejecutan). Si esta operación tarda segundos en completarse, puede dar como resultado una experiencia insatisfactoria para el usuario. Si el diseño ofrece la posibilidad de que los usuarios realicen varios cambios a un tipo de filtro de selección múltiple, considere la opción de mostrar el botón “aplicar”, para que el usuario ejecute la actualización una vez hechas todas las selecciones.

## Modifique el dashboard para mejorar el rendimiento

Tras asegurarnos de haber simplificado el dashboard lo más posible, podemos modificar nuestro diseño para aprovechar el almacenamiento en caché con el propósito de mejorar el rendimiento aún más.

### Dashboards de tamaño fijo

Una de las medidas más sencillas que podemos tomar para mejorar el rendimiento es comprobar que el dashboard tenga un tamaño fijo.



En Tableau, parte del proceso de representación es la creación de una disposición (la cantidad de filas y columnas que se mostrarán para los múltiples pequeños y las tabulaciones cruzadas; la cantidad y el intervalo de graduaciones de eje y líneas de cuadrícula que se deben visualizar; la cantidad y la ubicación de las etiquetas de marca que se deben mostrar, etc.). Esto se determina por el tamaño de la ventana en la que se visualizará el dashboard.

Si tenemos varias solicitudes para un mismo dashboard, pero de ventanas de diferentes tamaños, debemos generar una disposición para cada solicitud. Al configurar la disposición del dashboard en un tamaño fijo, nos aseguramos de que solo tengamos que crear una única disposición que se puede utilizar para todas las solicitudes. Esto es aún más importante para las representaciones del lado del servidor. El motivo es que los dashboards de tamaño fijo también permiten almacenar en caché y compartir mapas de bits representados en el servidor, lo que mejora tanto el rendimiento como la escalabilidad.

## Dashboards específicos para dispositivos

Con Tableau 10, presentamos una nueva funcionalidad denominada “dashboard específicos para dispositivos”. Se puede utilizar para crear disposiciones de dashboards personalizadas, seleccionadas automáticamente según el dispositivo que se utilice.

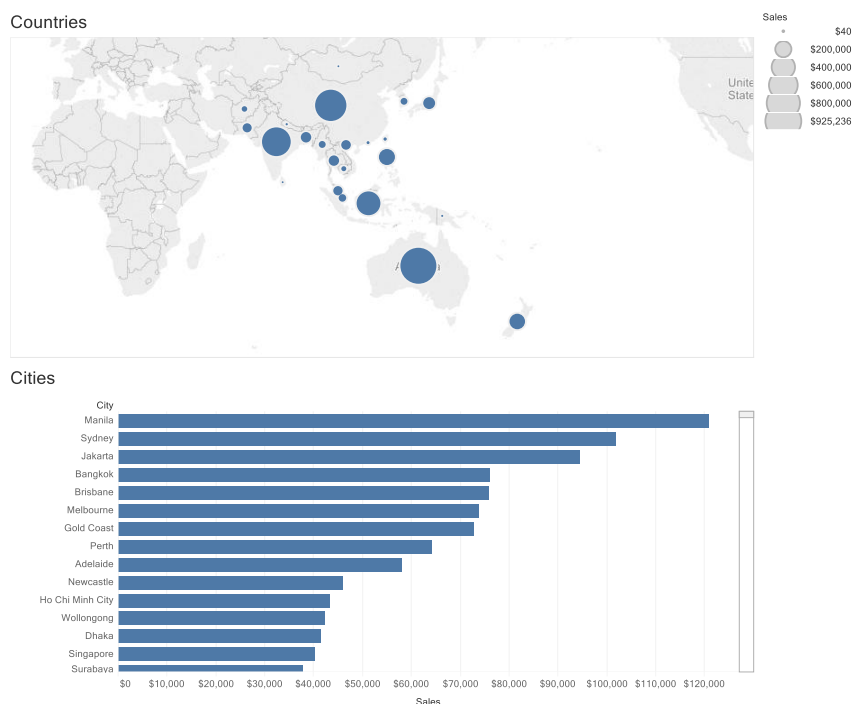
La disposición se selecciona según el tamaño de la pantalla:

- $\leq 500$  píxeles en el eje más corto: teléfono
- $\leq 800$  píxeles en el eje más corto: tableta
- $> 800$  píxeles : escritorio

Puesto que los diferentes dispositivos tendrán diferentes tamaños de pantalla, comprendidos en esos intervalos, y gracias a que los dispositivos pueden rotar la imagen, por lo general es recomendable que las disposiciones para teléfonos y tabletas estén configuradas en tamaño automático. Esto proporcionará la mejor experiencia de visualización para todos los dispositivos. Sin embargo, tendrá un impacto en la reutilización de la memoria caché (tanto para la memoria caché de modelos de presentación como para la de mosaicos de imágenes en las representaciones del lado del servidor). En términos generales, el beneficio de configurar el tamaño de manera adecuada para los distintos dispositivos es mayor que el impacto en el almacenamiento en caché, aunque es algo que se debe tener en cuenta. Hay que tener en cuenta que, una vez que los usuarios comiencen a utilizar el libro de trabajo, irá rellendo los modelos y los mapas de bits para los tamaños de pantalla más comunes y, así, el rendimiento mejorará.

## Uso del nivel de detalle de la visualización para reducir la cantidad de consultas

A pesar de que la práctica recomendada indica generalmente utilizar solo los campos que se necesitan para cada libro de trabajo, en algunas ocasiones podemos mejorar el rendimiento extrayendo más información en una hoja de trabajo para evitar que se realicen consultas en otra. Observe el dashboard siguiente:



Si se genera del modo que cabría esperar, el plan de ejecución dará como resultado dos consultas, una para cada hoja de trabajo:



## Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10  2.11

### Timeline

Workbook	Dashboard	Worksheet	Event	Time (s)
Book1	Dashboard 1	Countries	Executing Query	807 - 808
		Cities	Executing Query	813 - 815

```
SELECT [Superstore APAC].[City] AS [City],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[City]
```

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Country]
```

Si hacemos un cambio en el diseño de nuestro dashboard y agregamos Country a la hoja de trabajo Cities (en el estante Detalle), Tableau puede completar el dashboard con una sola consulta. El software de Tableau es lo suficientemente inteligente como para, primero, ejecutar la consulta para la hoja de trabajo Cities y, luego, utilizar la memoria caché de resultados de consultas para proporcionar los datos para la hoja de trabajo Countries. Esta funcionalidad se denomina “procesamiento por lotes de consultas”.

## Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10  3.02

### Timeline

Workbook	Dashboard	Worksheet	Event	Time (s)
Book1	Dashboard 1	Cities	Executing Query	126.5 - 129.5

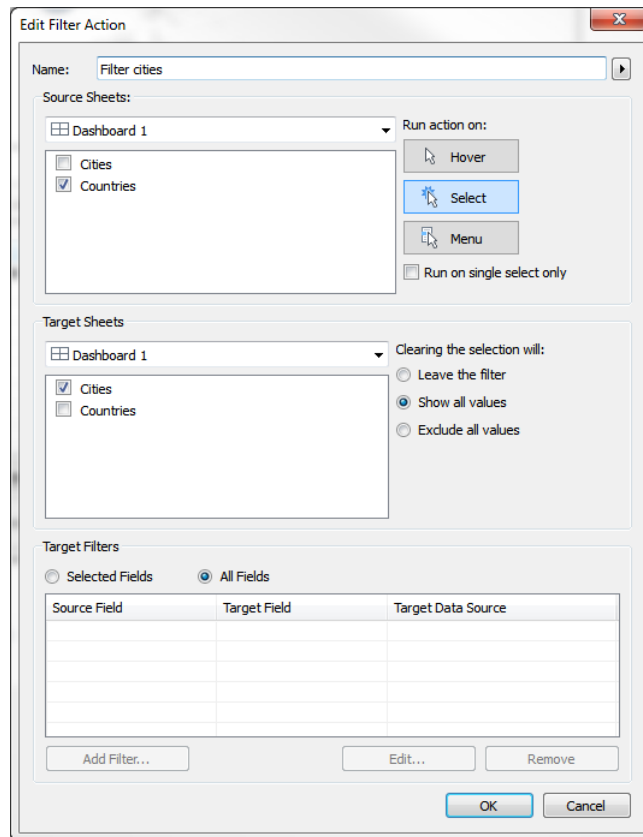
```
SELECT [Superstore APAC].[City] AS [City],
       [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[City],
         [Superstore APAC].[Country]
```

Está claro que esto no se puede realizar en todos los casos, porque cuando se agrega una dimensión a la visualización, se altera el nivel de detalle. Por eso, es posible que se muestre una mayor cantidad de marcas. Sin embargo, cuando cuenta con una relación jerárquica en sus datos, como en el ejemplo anterior, es una técnica útil, puesto que no afectará el nivel de detalle visible.

### Uso del nivel de detalle de la visualización para optimizar las acciones

Podemos emplear un enfoque similar con las acciones, para reducir la cantidad de consultas que ejecutamos. Imagine que usamos el mismo dashboard anterior (previo a la optimización) y decidimos agregar una acción de filtro de la hoja de trabajo Countries a la hoja de trabajo Cities.





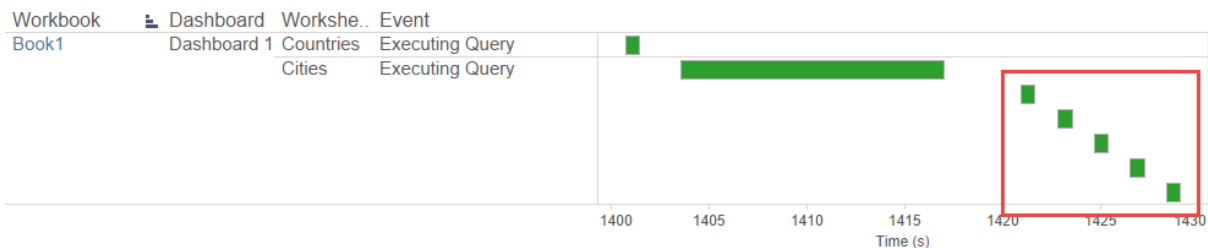
Cuando ejecutamos la acción al hacer clic en una marca del mapa, vemos que Tableau debe hacer una consulta para determinar los valores de la hoja de trabajo Cities. Esto es porque no hay datos en la memoria caché de resultados de consultas acerca de la relación ciudad-país.

### Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10 13.47

#### Timeline



```
SELECT [Superstore APAC].[City] AS [City],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Country] = 'Australia')
GROUP BY [Superstore APAC].[City]
```

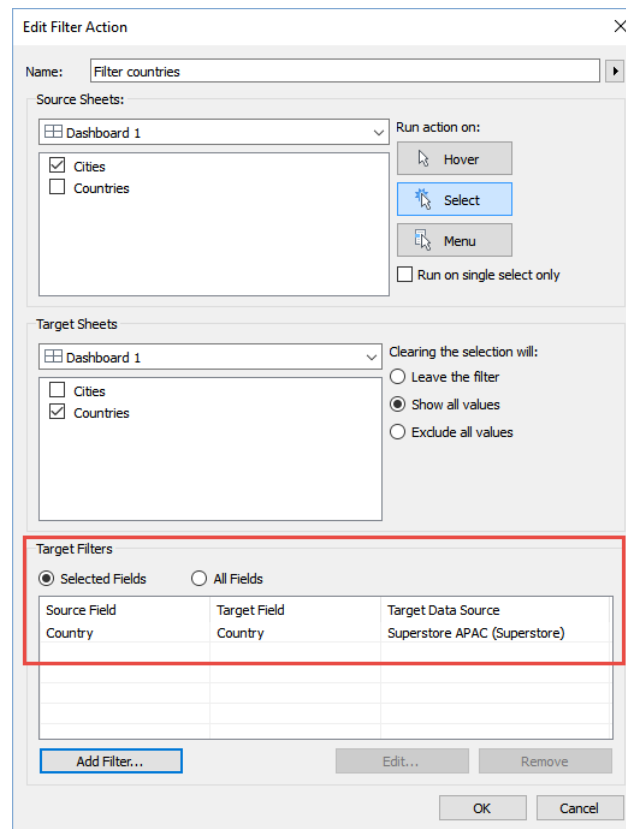
Si agregamos Country a la hoja de trabajo Cities, ahora tenemos información suficiente en la memoria caché de resultados de consultas como para filtrar sin necesidad de volver a la fuente de datos.

De la misma manera, podemos implementar esta optimización en situaciones en las que la hoja de trabajo de origen tenga más detalles que la de destino. Si adoptamos la definición de acción predeterminada, en la que se filtra mediante "Todos los campos":

Esto genera que el libro de trabajo ejecute una consulta por cada acción, porque la cláusula del filtro hace referencia a Country y a City, que no pueden obtenerse de la memoria caché de resultados de consultas para la hoja de trabajo Countries.

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE (([Superstore APAC].[City] = 'Sydney') AND ([Superstore APAC].[Country] =
'Australia'))
GROUP BY [Superstore APAC].[Country]
```

Si cambiamos la acción para que solo filtre sobre la base de Country:



Ahora podemos satisfacer los requisitos de este filtro desde la memoria caché de resultados de consultas, por lo que no necesitamos volver a consultar la fuente de datos. Al igual que antes, debe evaluar si la modificación del nivel de detalle tendrá un efecto en el diseño de su hoja. De no ser así, esta puede ser una técnica útil.

### Un buen diseño de hoja de trabajo

El primer nivel inferior al dashboard es la hoja de trabajo. En Tableau, el diseño de la hoja de trabajo está intrínsecamente relacionado con las consultas que se realizan. Cada hoja de trabajo generará una o más consultas, por lo que, en este nivel, debemos asegurarnos de generar las consultas que sean más convenientes.

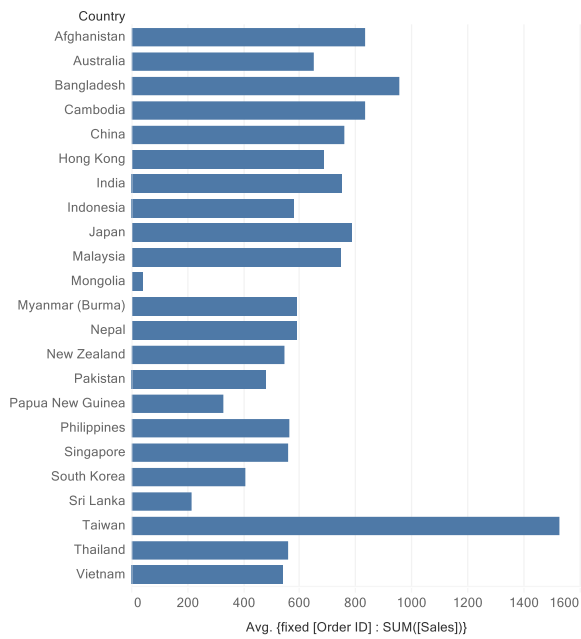
### Solo incluya los campos que necesita

Échele un vistazo al estante Detalle y quite todos los campos que no se utilicen directamente en la visualización, los que no sean necesarios para las descripciones emergentes y los que no sean requeridos para el nivel de detalle de las marcas. Esto agilizará la consulta en la fuente de datos y hará que se devuelvan menos datos en los resultados de la consulta. Hay algunas excepciones a esta regla, como ya hemos explicado (el procesamiento por lotes de consultas para eliminar consultas similares en otras hojas de trabajo). Sin embargo, son situaciones poco comunes y dependen de que no se modifique el nivel de detalle de la hoja.

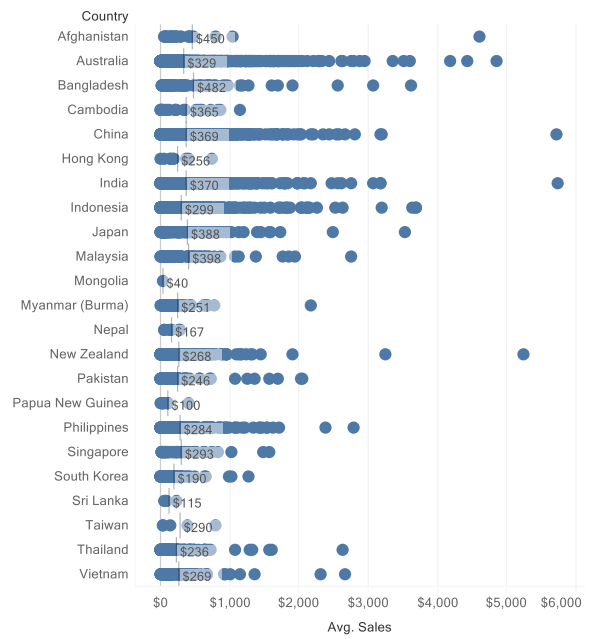
### Muestre la cantidad de marcas mínima para contestar la pregunta

En Tableau, a menudo hay varias maneras de calcular el mismo número. Observe el dashboard a continuación. Ambas hojas de trabajo contestan la pregunta: “¿Cuál es el tamaño promedio de pedido por país?”.

Avg Order Size 1



Avg Order Size 2

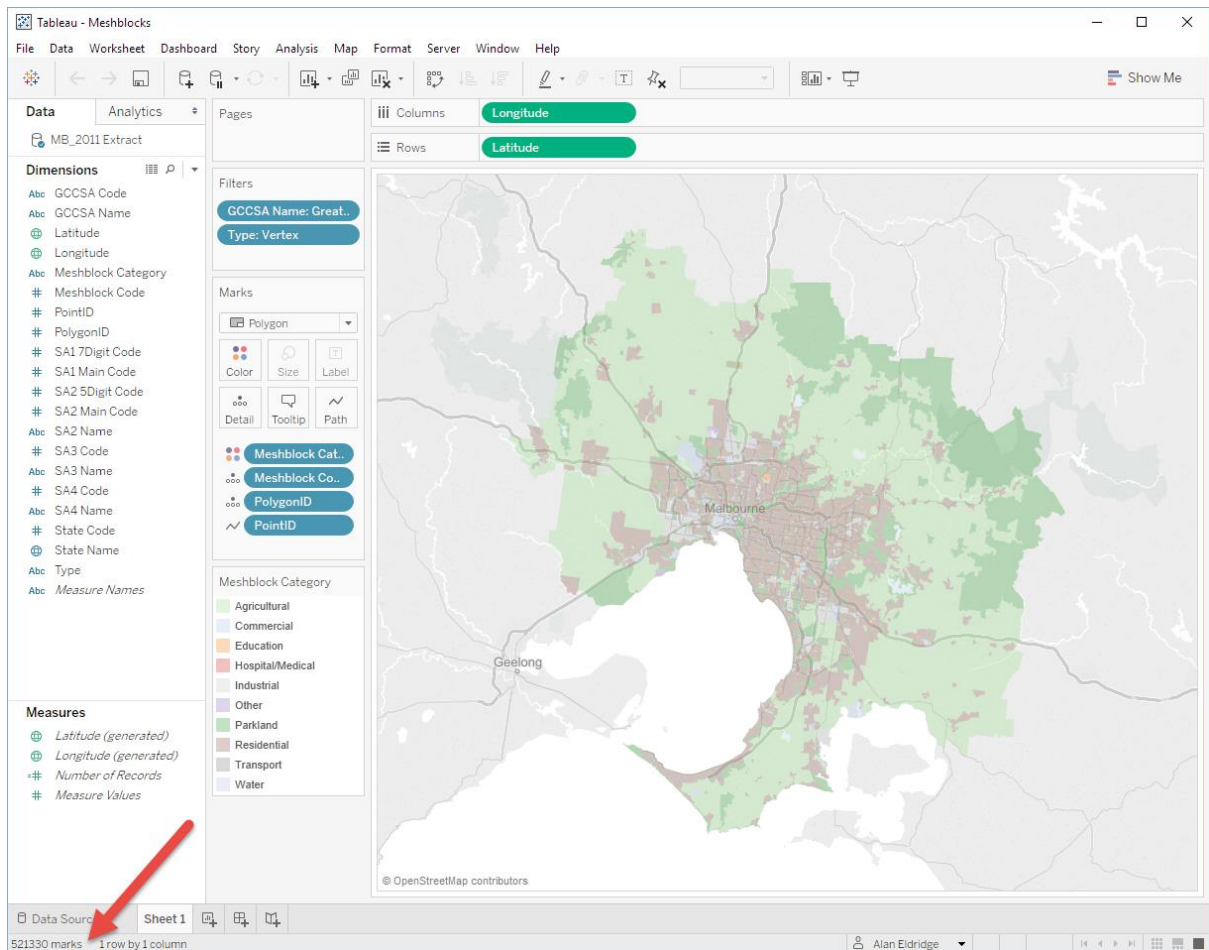


En la Hoja 1 solo se muestra una única marca, que representa el tamaño promedio de pedido para cada país. La consulta devuelve solo 23 registros de la fuente de datos. Sin embargo, en la Hoja 2 se muestra una marca por cada pedido dentro de cada país y, luego, se calcula el promedio, representado por la línea de referencia. Esto requiere la extracción de 5436 registros de la fuente de datos.

La Hoja 1 es una solución mejor si solo estamos interesados en la pregunta original: “¿Cuál es el tamaño promedio de pedido por país?”. Sin embargo, la Hoja 2 proporciona también información más detallada sobre el rango de tamaños de pedido, lo que nos permite identificar valores atípicos.

Evite generar visualizaciones extremadamente complejas

Una métrica importante que hay que tener en cuenta es la cantidad de puntos de datos que se representan en cada visualización. Puede encontrar esta información fácilmente en la barra de estado de la ventana de Tableau Desktop:

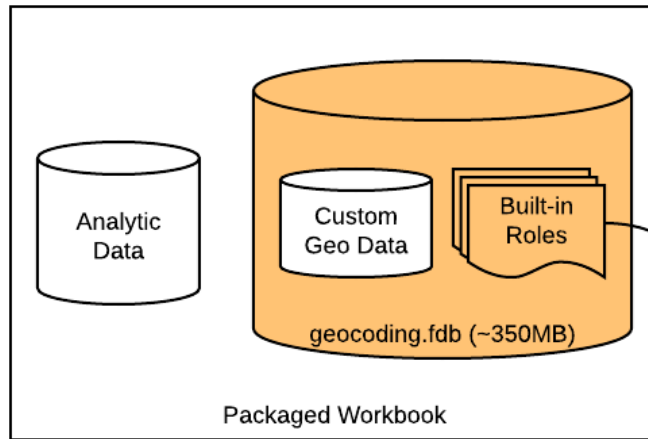


A pesar de que no hay ninguna regla establecida que defina cuántas son “demasiadas marcas”, tenga en cuenta que la cantidad de marcas es directamente proporcional al trabajo de la CPU y la memoria RAM que se necesitan para visualizarlas. Tenga precaución cuando trabaje con tabulaciones cruzadas grandes, diagramas de dispersión o mapas con polígonos personalizados complejos.

## Mapas

### *Geocodificación personalizada*

Cuando se importa una función de geocodificación personalizada, se almacena en la base de datos de geocodificación (un archivo de base de datos de Firebird, ubicado de manera predeterminada en C:\Archivos de programa\Tableau\Tableau 10.0\Local\data\geocoding.fdb). Si utiliza la función en un libro de trabajo y lo guarda como un libro de trabajo empaquetado, se comprimirá todo el archivo de base de datos en el archivo TWBX (la totalidad de los 350 MB).

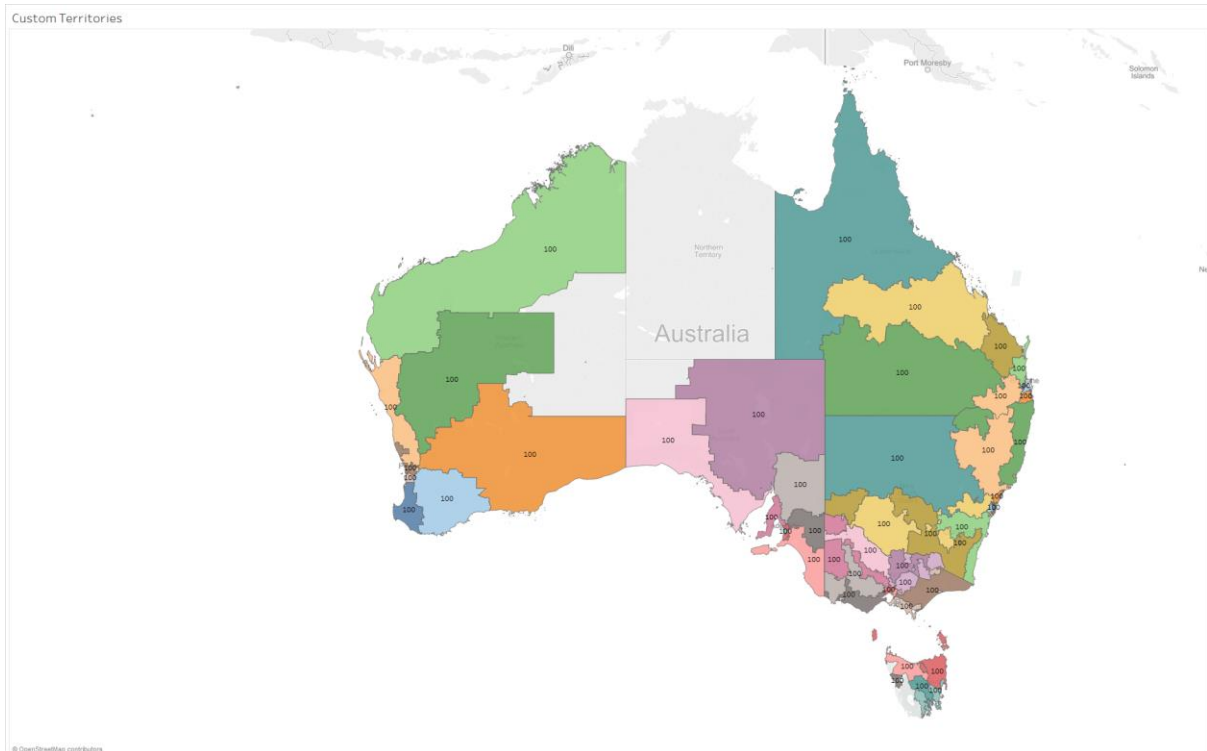


Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
AreaCode.tds	Tableau Datasource	2 KB	No	9 KB	82%	9/06/2016 1:29 PM
City.tds	Tableau Datasource	2 KB	No	13 KB	85%	9/06/2016 1:29 PM
CMSA.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Congress.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Country.tds	Tableau Datasource	3 KB	No	17 KB	87%	9/06/2016 1:29 PM
County.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
GEOCODING.FDB	FDB File	118,407 KB	No	359,280 KB	68%	9/06/2016 1:28 PM
State.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Suburb.tds	Tableau Datasource	3 KB	No	15 KB	86%	9/06/2016 1:29 PM
ZipCode.tds	Tableau Datasource	2 KB	No	8 KB	81%	9/06/2016 1:29 PM

Esto produce que el archivo TWBX resultante sea a) demasiado grande, puesto que el tamaño de los datos de geocodificación, una vez comprimidos, es de 110 MB; y b) muy lento de abrir, puesto que la descompresión inicial debe trabajar con un conjunto de datos mucho más grande. Un método más eficaz es no importar los datos como una función de geocodificación personalizada, sino utilizar la combinación dentro del libro de trabajo, para unir los datos analíticos con los datos geoespaciales. Con este método, el archivo geocoding.fdb no estará incluido en el archivo TWBX, que solo tendrá los datos analíticos y de geocodificación personalizada.

#### *Territorios personalizados*

Los territorios personalizados son una nueva funcionalidad de Tableau 10, con la que los usuarios pueden combinar áreas de bases de datos internas de geocodificación para crear regiones agregadas.



La representación inicial de los territorios personalizados, basados en muchas regiones de menor nivel, puede ser muy lenta. Por este motivo, utilice esta función con precaución. Sin embargo, una vez realizada, los territorios personalizados se almacenan en caché y se puede esperar un buen rendimiento.

#### *Mapas completos y mapas de puntos*

Las marcas de los mapas completos (tanto si se usan en un mapa o como una marca de algún otro tipo de gráfico) son costosas cuando se utiliza la representación del lado del cliente. Esto se debe a que es necesario enviar los datos de polígonos para la forma, lo que puede ser bastante complejo. Si el rendimiento de la representación es lento, considere utilizar un mapa de símbolos como alternativa.

#### *Marcas de polígonos*

Cualquier visualización que utilice marcas de polígonos obligará a Tableau Server a realizar una [representación del lado del servidor](#), que puede afectar la experiencia del usuario final. Utilícelas con moderación.

#### *Otros factores*

##### *Grandes tabulaciones cruzadas*

En versiones anteriores de este documento, le recomendábamos evitar grandes tabulaciones cruzadas, puesto que las representaciones resultantes eran muy lentas. En las últimas versiones de la herramienta, se mejoró la mecánica subyacente de este tipo de visualización. Ahora, las tabulaciones cruzadas se pueden representar tan rápido como otros tipos de gráficos múltiples y pequeños. Sin embargo, aún le recomendamos que valore la utilización de grandes tabulaciones cruzadas, puesto que requieren que la base de datos subyacente lea muchos datos que no son útiles para el análisis.

##### *Descripciones emergentes*

De manera predeterminada, la colocación de una dimensión en el estante Descripción emergente produce que esta se agregue mediante la función de atributo ATTR(). Esto requiere que se realicen dos agregaciones en la fuente de datos subyacente –MIN() y MAX()– y que ambos resultados se pasen

nuevamente al conjunto de resultados. Consulte la sección acerca del [uso de ATTR\(\)](#) (más adelante en el documento) para obtener más información.

Si no le preocupa la posibilidad de tener múltiples valores de dimensión, una solución más eficaz es simplemente utilizar una agregación, en vez de la predeterminada ATTR. Elija una, MIN() o MAX(), y solo use esa para maximizar sus posibilidades de emplear los datos almacenados en caché.

Otra opción (si sabe que no afectará el nivel de detalle visual de su visualización) es ubicar la dimensión en el estante Nivel de detalle, en vez del estante Descripción emergente. Como resultado, el campo de dimensión se utiliza directamente en las cláusulas SELECT y GROUP BY de la consulta. Le recomendamos comprobar si esta opción ofrece un rendimiento mejor que la agregación simple, puesto que puede depender del rendimiento de la plataforma de sus datos.

### Leyendas

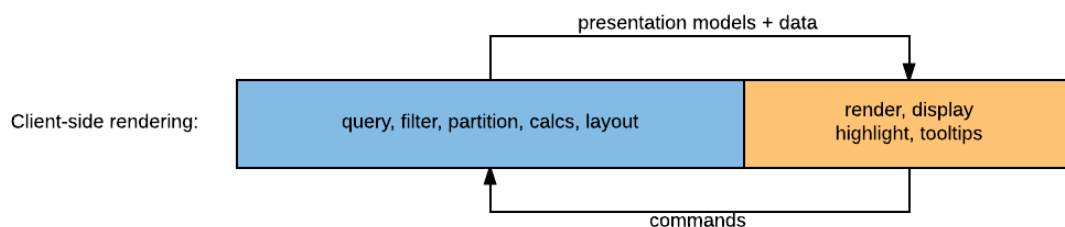
Por lo general, las leyendas no son una causa de problemas de rendimiento, puesto que sus dominios se extraen de la memoria caché de resultados de consultas. Sin embargo, pueden sobrecargar la representación si el dominio enumerado es grande, porque los datos deben transferirse al navegador del cliente. Si este es el caso, la leyenda por lo general no es útil; por lo tanto, simplemente quítela.

### Estante Página

Algunos usuarios piensan que el estante Página funciona de la misma manera que el estante Filtro; es decir, reduce el número de registros que devuelve la fuente de datos. Esto no es correcto. La consulta de la hoja de trabajo devolverá registros para todas las marcas de todas las páginas. Si cuenta con una dimensión de página con un alto grado de cardinalidad (es decir, muchos valores únicos), esto puede aumentar significativamente el tamaño de la consulta de la hoja de trabajo, lo que afecta el rendimiento. Use esta función con moderación.

### Representación del lado del cliente y representación del lado del servidor

Antes de que las marcas y los datos de una vista se muestren en el navegador web de un cliente, se recuperan, interpretan y representan. Tableau Server puede realizar este proceso en el navegador web del cliente o en el servidor. La representación del lado del cliente es el modo predeterminado porque manejar la representación y todas las interacciones en el servidor puede producir más transferencias de datos en la red y retrasos de recorrido de ida y vuelta. Con la representación del lado del cliente, muchas de las interacciones con la vista son más rápidas porque se interpretan y generan en el navegador.

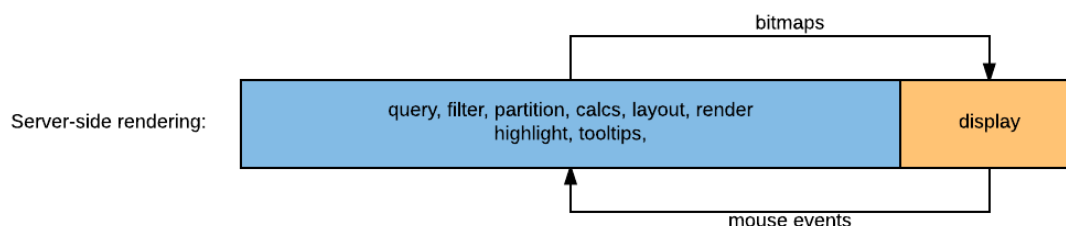


*(Azul: realizado en el servidor; naranja: realizado en el navegador del cliente).*

Sin embargo, algunas vistas se generan con mayor eficacia en el servidor, donde hay más poder de procesamiento. La representación del lado del servidor tiene sentido para vistas complejas, ya que los archivos de imágenes consumen significativamente menos ancho de banda que los datos usados para crear esas imágenes. También, como las tabletas suelen tener un rendimiento mucho más reducido que los equipos informáticos, estas admiten vistas menos complejas. El lado negativo es que las interacciones



simples, como las descripciones emergentes y los resaltados, pueden ser lentas en la representación del lado del servidor, puesto que requieren un recorrido de ida y vuelta con el servidor.



*(Azul: realizado en el servidor; naranja: realizado en el navegador del cliente).*

La configuración de Tableau Server incluye un umbral de complejidad para resolver automáticamente todas estas situaciones. Este umbral desencadena la representación de la vista en el servidor, en vez de en el navegador web. El umbral es diferente para equipos de escritorio y dispositivos móviles. Por eso, hay casos en los que la vista abierta desde el navegador web de un equipo puede representarse del lado del cliente, mientras que la misma vista abierta desde el navegador web de una tableta se representará del lado del servidor. Los filtros también pueden variar el comportamiento de la representación. Un libro de trabajo puede abrirse inicialmente con una representación del lado del servidor y puede cambiar a una representación del lado del cliente cuando se aplica un filtro. Además, si una visualización utiliza marcas de polígonos o el estante Páginas, solo utilizará la representación del lado del servidor, aunque cumpla con todos los otros requisitos para la representación del lado del cliente. Por este motivo, utilice estas funciones con precaución.

Como administrador, usted puede probar o ajustar esta configuración, tanto para equipos como para tabletas. Consulte el siguiente enlace para obtener más información:

[https://onlinehelp.tableau.com/current/server/es-es/browser\\_rendering.htm](https://onlinehelp.tableau.com/current/server/es-es/browser_rendering.htm)

## Filtros eficaces

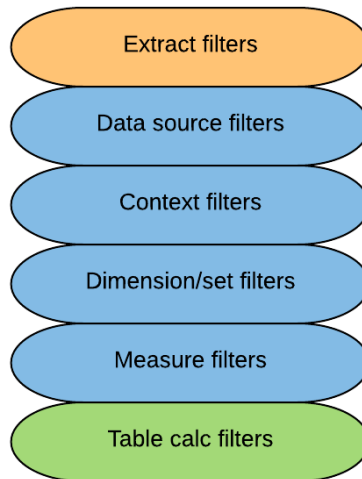
El filtrado en Tableau es extremadamente eficaz y descriptivo. No obstante, los filtros ineficaces son una de las causas más comunes del mal rendimiento de los libros de trabajo y los dashboards. Las secciones siguientes presentan una serie de prácticas recomendadas para trabajar con filtros.

Nota: la eficacia de los filtros se ve drásticamente afectada por la presencia y el mantenimiento de índices en la fuente de datos. Consulte la sección sobre índices para obtener más detalles.

## Tipos de filtros

### *Prioridad de los filtros*

En Tableau, los filtros se aplican con el orden siguiente:



### *Filtros de extracción*

Estos filtros solo funcionan con extracciones de datos; en esos casos, se aplican lógicamente antes de todos los demás filtros. Limitan los datos que se extraen de la fuente de datos subyacente, y pueden ser filtros de dimensiones o medidas. Además, pueden realizar operaciones de “superior” o “muestra” para reducir la cantidad de registros devueltos, dependiendo de la plataforma de la fuente de datos.

### *Filtros de fuentes de datos*

Los filtros de fuentes de datos conforman el nivel más alto disponible en las conexiones en tiempo real. Una diferencia clave entre los filtros de fuentes de datos y los filtros de contexto es que los primeros tienen un alcance de fuente de datos completa, mientras que los segundos se configuran por hoja de trabajo. Esto implica que, cuando se utilizan en una fuente de datos publicada, se pueden utilizar los filtros de fuentes de datos, mientras que los filtros de contexto se aplican en el nivel de la hoja de trabajo.

Los filtros de las fuentes de datos pueden ser una manera eficaz de configurar una limitación en una fuente de datos para evitar que los usuarios finales ejecuten accidentalmente una consulta masiva. Por ejemplo, se puede implementar un filtro de fuente de datos para limitar las consultas a una tabla de transacciones de manera que solo se consulten los últimos 6 meses.

### *Filtros de contexto*

De manera predeterminada, todos los filtros que configura en Tableau se calculan de forma independiente. Es decir, cada filtro accede a todas las filas de su fuente de datos independientemente de los demás filtros. Sin embargo, al especificar un filtro de contexto, puede establecer una dependencia en cualquier otro filtro que defina, porque estos procesarán solo los datos que pasen a través del filtro de contexto.

Debe utilizar los filtros de contexto cuando sean necesarios para obtener la respuesta correcta (por ejemplo, un filtro para los N superiores). Por ejemplo, imagine una vista en la que se muestren los 10 productos principales por SUM(Sales), con un filtro de región. Sin un filtro de contexto, se ejecuta la consulta siguiente:

```
SELECT [Superstore APAC].[Product Name] AS [Product Name],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
INNER JOIN (
  SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
               SUM([Superstore APAC].[Sales]) AS [$__alias__0]
  FROM [dbo].[Superstore APAC] [Superstore APAC]
```

```

GROUP BY [Superstore APAC].[Product Name]
ORDER BY 2 DESC
) [t0] ON ([Superstore APAC].[Product Name] = [t0].[Product Name])
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Product Name]

```

Esto devuelve la contribución de Oceanía para los 10 productos principales a nivel internacional. Si lo que en realidad desea es conocer los 10 productos principales dentro de la región de Oceanía, debe agregar un filtro de región al contexto. En dicho caso, se ejecuta la siguiente consulta:

```

SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok],
SUM([Superstore APAC].[Sales]) AS [$_alias_0]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Product Name]
ORDER BY 3 DESC

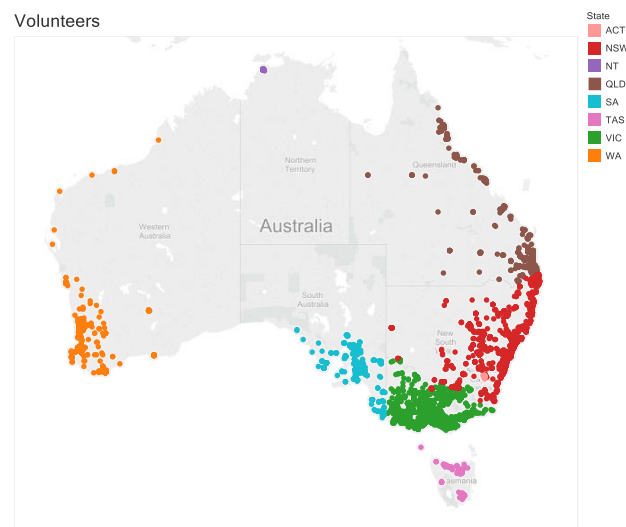
```

Históricamente, los filtros de contexto se implementaban como una tabla temporal en la fuente de datos. Ya no es así. En la mayoría de las situaciones, los filtros de contexto se implementan como parte de la consulta de fuente de datos (como se mostró anteriormente) o se procesan de manera local en el motor de datos.

Ya no debe utilizar los filtros de contexto como mecanismo para mejorar el rendimiento de las consultas.

### Filtrado de dimensiones categóricas

Observe la siguiente visualización, un mapa de Australia con marcas por código postal:



Podemos filtrar el mapa de diferentes maneras para que solo muestre los códigos postales de WA (los puntos naranjas):

- Podemos seleccionar todas las marcas de WA y conservar solo la selección.
- Podemos seleccionar todas las marcas fuera de WA y excluir la selección.
- Podemos conservar otro atributo, como la dimensión de estado.
- Podemos filtrar por intervalo, en los valores de código postal o los de latitud/longitud.

### Discreto

Si implementáramos las primeras dos opciones, la de conservar y la de excluir, descubriríamos que no son eficaces. De hecho, a veces pueden ser más lentas que el conjunto de datos sin filtros. Esto se debe

a que están expresadas en formato de listas discretas de valores de códigos postales, filtrados por el sistema de administración de bases de datos (DBMS); esto se efectúa mediante una cláusula WHERE IN o, si hay muchos valores, a través de la creación de una tabla temporal con los valores seleccionados y el uso de una unión interior entre esta y las tablas principales. En un conjunto grande de marcas, esto puede producir una consulta muy costosa de evaluar.

La tercera opción es rápida en este ejemplo porque el filtro resultante (WHERE STATE="Western Australia") es muy sencillo, y la base de datos puede procesarlo eficazmente. Sin embargo, este enfoque se vuelve menos eficaz a medida que la cantidad de miembros de la dimensión necesarios para expresar el filtro aumenta y, en última instancia, se acerca al rendimiento de las opciones de lazo y de conservar.

#### *Intervalo de valores*

El uso del filtro de intervalo de valores también permite que la base de datos evalúe una cláusula de filtro simple (WHERE POSTCODE >= 6000 AND POSTCODE <= 7000 O WHERE LONGITUDE < 129), lo que acelera la ejecución. Pero este enfoque, a diferencia de los filtros en dimensiones relacionadas, no se vuelve más complejo a medida que se incrementa la cardinalidad de las dimensiones.

La conclusión es que los filtros de intervalo de valores suelen ser más rápidos de evaluar que largas listas detalladas de valores discretos. Por eso, en la medida de lo posible, deben usarse antes que las opciones de conservar y excluir para grandes conjuntos de marcas.

#### *Filtros de segmentación*

Los filtros de segmentación son filtros que se aplican a las dimensiones que no se utilizan en la visualización (es decir, no son parte del nivel de detalle de la visualización). Por ejemplo, es posible tener una visualización en la que se muestren las ventas totales por país, pero que esté filtrada por región. En este ejemplo, la consulta que se ejecuta es la siguiente:

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Country]
```

Estos filtros se vuelven cada vez más complejos si segmentamos los resultados de una agregación. Por ejemplo, si filtramos la visualización anterior no por región, sino para que se muestren las ventas de los 10 productos más rentables, Tableau debe ejecutar dos consultas: una en el nivel de producto, para aislar los 10 productos más rentables, y otra en el nivel de país, a partir de la restricción resultante de la primera consulta.

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
INNER JOIN (
  SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
               SUM([Superstore APAC].[Profit]) AS [$_alias_0]
  FROM [dbo].[Superstore APAC] [Superstore APAC]
  GROUP BY [Superstore APAC].[Product Name]
  ORDER BY 2 DESC
) [t0] ON ([Superstore APAC].[Product Name] = [t0].[Product Name])
GROUP BY [Superstore APAC].[Country]
```

Tenga precaución cuando utilice los filtros de segmentación, puesto que su evaluación puede ser costosa. Además, tenga en cuenta que, debido a que la dimensión no forma parte de la memoria caché de resultados de consultas, no se puede realizar un filtrado rápido en el navegador sobre los filtros

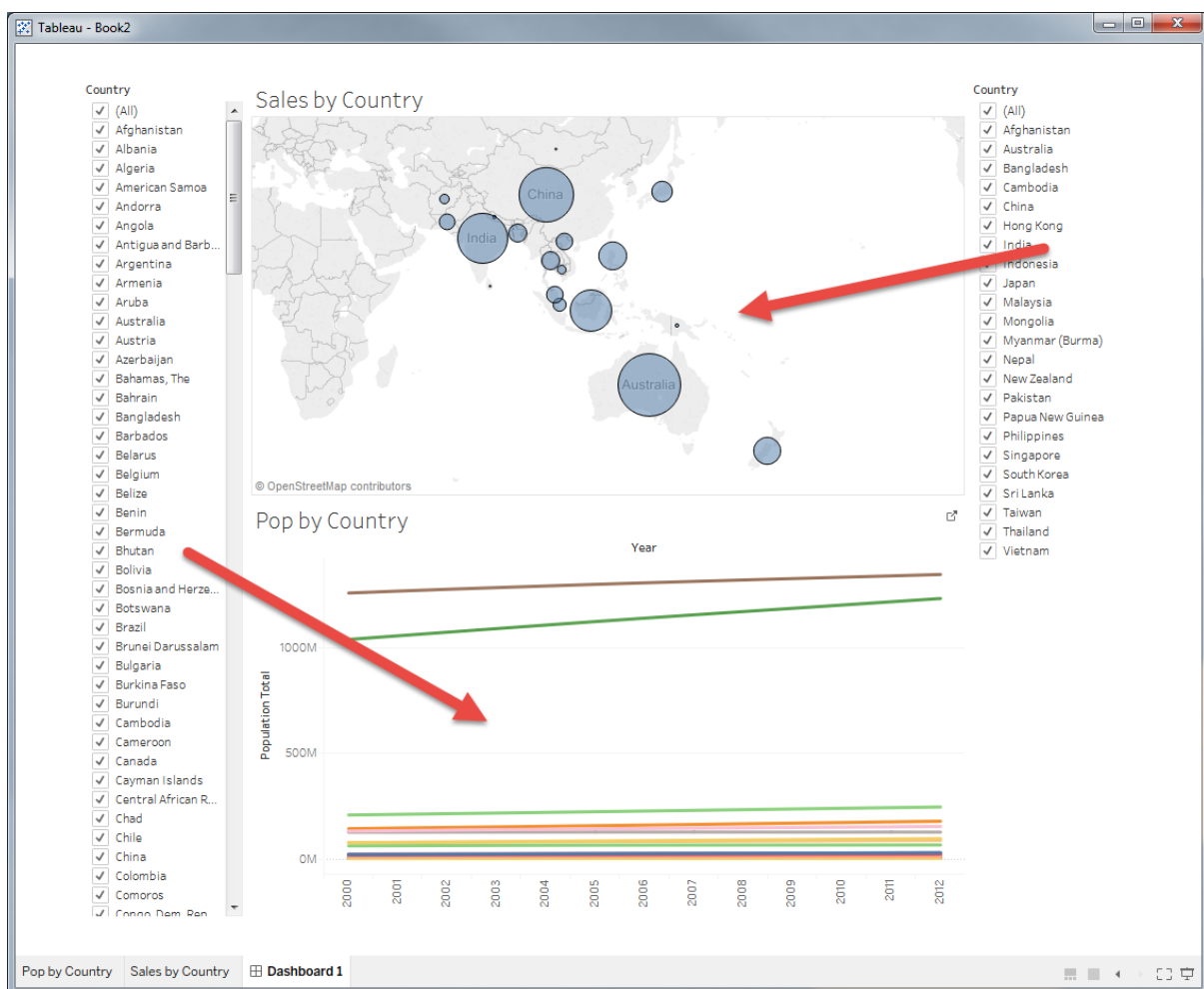
de segmentación (consulte la [sección anterior](#) acerca de las representaciones del lado del cliente y del lado del servidor).

### Filtros para distintas fuentes de datos

El filtro para distintas fuentes de datos es una nueva funcionalidad de Tableau 10. Permite agregar un filtro en múltiples fuentes de datos que tienen uno o más campos en común. Las relaciones se definen de la misma forma que las combinaciones: de manera automática, sobre la base de la concordancia de nombre o tipo; o manualmente, a través de una relación personalizada, que se configura desde el menú Datos.

Los filtros para distintas fuentes de datos tienen los mismos efectos en el rendimiento de un dashboard que los filtros rápidos. Cuando se modifican, pueden generar la actualización de varias zonas, lo que, potencialmente, puede requerir múltiples consultas. Utilícelos de manera razonable. Además, si espera que los usuarios apliquen varios cambios, considere la posibilidad de mostrar el botón “aplicar”, de forma que solo se envíen las consultas una vez que se hayan hecho todas las selecciones.

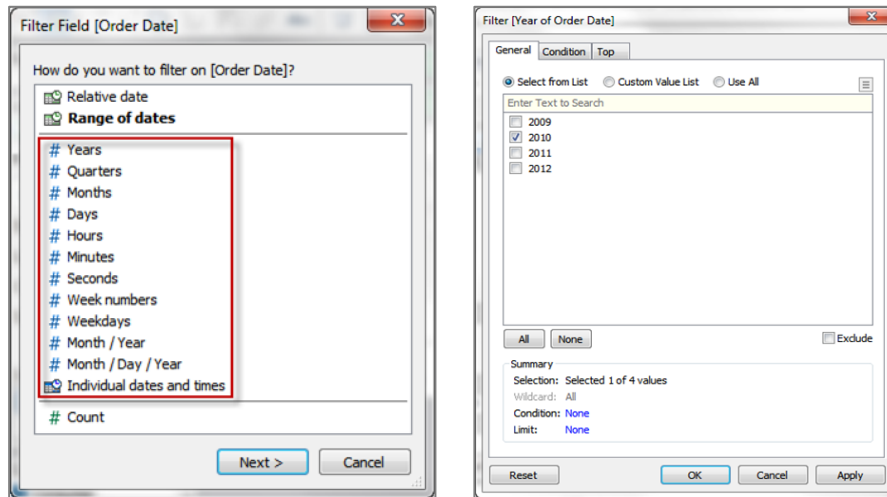
Además, tenga en cuenta que el dominio del filtro se extrae de la fuente de datos “principal”. Es decir, la primera fuente de datos utilizada en la hoja donde se creó el filtro. Si un campo relacionado cuenta con diversos dominios en diferentes fuentes de datos, debe tener cuidado con cuál utiliza, puesto que el mismo filtro puede mostrar valores diferentes, como se muestra a continuación:



## Filtrado de fechas: discreta, por intervalo, relativa

Los campos de fecha son un tipo especial de dimensión que Tableau a menudo controla de manera diferente que los datos categóricos estándar. Esto es así especialmente cuando crea filtros de fecha. Los filtros de fecha son extremadamente comunes y se dividen en tres categorías: filtros de fecha relativa, que muestran un intervalo de fechas relativo a un día específico; filtros por intervalo de fechas, que muestran un intervalo definido de fechas discretas; y filtros de fecha discreta, que muestran fechas individuales que usted haya seleccionado de una lista. Como se muestra en la sección anterior, el método usado puede tener un impacto significativo en la eficacia de la consulta resultante.

### Discreta



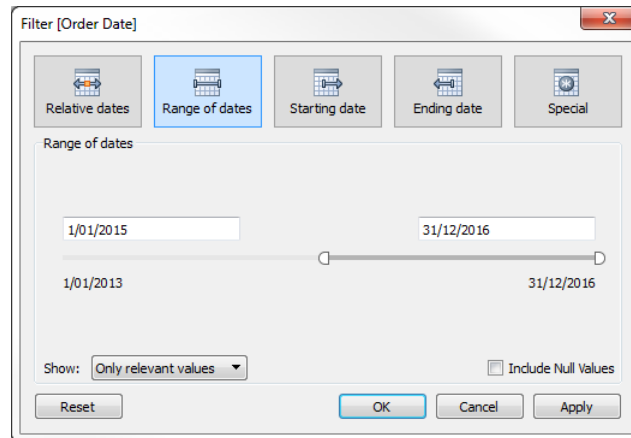
Algunas veces, es posible que desee filtrar para incluir fechas individuales específicas o fechas completas. Este tipo de filtro se denomina filtro de fecha discreta porque se definen valores discretos en lugar de un intervalo. Este tipo de filtro hace que la expresión de fecha se transmita a la base de datos como un cálculo dinámico:

```
SELECT [FactSales].[Order Date], SUM([FactSales].[SalesAmount])
FROM [dbo].[FactSales] [FactSales]
WHERE (DATEPART(year,[FactSales].[Order Date])) = 2010)
GROUP BY [FactSales].[Order Date]
```

En la mayoría de los casos, los optimizadores de consultas evaluarán de forma inteligente el cálculo DATEPART, pero existen ciertas situaciones en las que el uso de filtros de fecha discreta puede desencadenar una ejecución ineficaz de la consulta. Un ejemplo de esto es hacer una consulta en una tabla con particiones con un filtro de fecha discreta en la clave de partición de fecha. Como la tabla no está particionada según el valor de DATEPART, algunas bases de datos evaluarán el cálculo en todas las particiones para encontrar registros que concuerden con el criterio, aunque esto no sea necesario. En este caso, es posible que vea una mejora considerable en el rendimiento si utiliza un filtro de "intervalo de fechas" o un filtro de fecha relativa con una fecha de anclaje especificada.

Una manera de optimizar el rendimiento para este tipo de filtros es realizar el cálculo con una extracción de datos. Primero, cree un campo calculado que implemente la función DATEPART explícitamente. Si después crea una extracción de datos de Tableau, este campo calculado se materializará como valores almacenados en la extracción (porque el resultado de la expresión es determinístico). Filtrar el campo calculado en vez de la expresión dinámica será más rápido porque el valor simplemente se puede buscar, en vez de calcularse en el momento de consulta.

## Intervalo de fechas

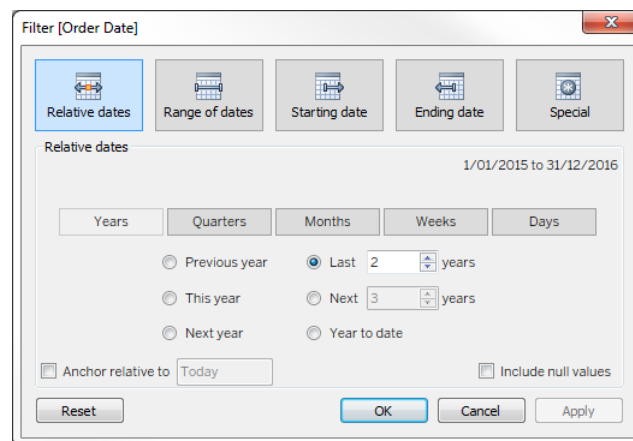


Este tipo de filtro se usa cuando se desea especificar un intervalo de fechas continuas. La siguiente estructura de consulta se transmite a la base de datos:

```
SELECT SUM([factOrders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[factOrders] [factOrders]
WHERE (([factOrders].[Order Date] >= {d '2015-01-01'}) AND
([factOrders].[Order Date] <= {d '2016-12-31'}))
GROUP BY ()
```

Este tipo de cláusula WHERE es muy eficaz para los optimizadores de consulta, ya que permite que los planes de ejecución aprovechen por completo los índices y las particiones. Si observa tiempos de consulta lentos cuando agrega filtros de fecha discreta, considere reemplazarlos por filtros por intervalo de fechas y vea si eso funciona.

## Relativa



Un filtro de fecha relativa le permite definir un intervalo de fechas que se actualiza sobre la base de la fecha y la hora en que abre la vista. Por ejemplo, quizás quiere ver las ventas del año hasta la fecha, todos los registros de los últimos 30 días o los errores que se cerraron la semana pasada. Los filtros de fecha relativa también pueden ser relativos respecto de una fecha de anclaje específica y no de hoy.

```
SELECT SUM([factOrders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[factOrders] [factOrders]
WHERE (([factOrders].[Order Date] >= {ts '2015-01-01 00:00:00'}) AND
([factOrders].[Order Date] < {ts '2017-01-01 00:00:00'}))
GROUP BY ()
```

Como puede ver, la cláusula WHERE resultante usa una sintaxis de intervalo de fechas, de modo que esta también es una forma eficaz de filtro de fecha.

Tenga en cuenta que, debido a la naturaleza variable de los filtros de fechas que son relativos a la fecha u hora actual (ya sea a través de un filtro de fecha relativa, como la anterior, o mediante el uso explícito de los operadores NOW o TODAY en la fórmula del filtro), la caché de consultas no resulta tan eficaz para las consultas que la utilizan. Tableau las marca como “consultas transitorias” en el servidor de caché y sus resultados no se conservan durante tanto tiempo como los resultados de otras consultas.

### Tarjetas de filtros

Mostrar demasiadas tarjetas de filtros lo retrasarán, especialmente si los configura para que usen “solo valores relevantes” y usted tiene muchas listas discretas. En lugar de ello, pruebe un enfoque analítico más guiado y use filtros de acciones en un dashboard. Si genera una vista con una gran cantidad de filtros para hacerla muy interactiva, piense en si varios dashboards con diferentes niveles y temas funcionarían mejor (una pista: sí, probablemente lo harían).

### Enumeradas y no enumeradas

Las tarjetas de filtros enumeradas requieren que Tableau haga consultas a la fuente de datos acerca de todos los valores de campo posibles para poder representar la tarjeta de filtro. Son las siguientes:

- Lista de valores múltiples: *todos los miembros de dimensión*.
- Lista de valores individuales: *todos los miembros de dimensión*.
- Lista compacta: *todos los miembros de dimensión*.
- Deslizador: *todos los miembros de dimensión*.
- Filtros de medida: *valores MÍNIMOS y MÁXIMOS*.
- Filtros de intervalo de fechas: *valores MÍNIMOS y MÁXIMOS*.

Por otro lado, las tarjetas de filtros no enumeradas no requieren conocimiento de los valores de campo posibles. Son las siguientes:

- Lista de valores personalizada
- Coincidencia de comodín
- Filtros de fecha relativa
- Filtros de fecha de período de navegación

En consecuencia, las tarjetas de filtros no enumeradas reducen la cantidad de consultas relacionadas con filtros rápidos que la fuente de datos debe ejecutar. Además, las tarjetas de filtros no enumeradas son más veloces cuando hay muchos miembros de dimensión para mostrar.

El uso de tarjetas de filtros no enumeradas puede mejorar el rendimiento, pero lo hace a expensas de contexto visual para el usuario final.

### Valores relevantes

Se pueden configurar las tarjetas de filtros enumeradas para que muestren los valores de campo posibles de tres maneras diferentes:

- Todos los valores de la base de datos: cuando selecciona esta opción, se muestran todos los valores de la base de datos, independientemente de los otros filtros de la vista. No hay necesidad de que el filtro vuelva a consultar la base de datos cuando hay otros filtros cargados.



- Todos los valores en contexto: esta opción solo está disponible cuando tiene filtros de contexto activos. En la tarjeta de filtros se mostrarán todos los valores del contexto, independientemente de los otros filtros de la vista. El filtro no debe volver a hacer la consulta a la base de datos cuando se modifican los filtros de dimensión o medida. Sin embargo, sí debe volver a hacer la consulta en el caso de que cambie el contexto.
- Solo valores relevantes: cuando selecciona esta opción, se tienen en cuenta otros filtros y se muestran solo los valores que pasan estos filtros. Por ejemplo, un filtro por estado solo mostrará los estados orientales cuando se haya configurado un filtro por región. Por lo tanto, el filtro debe volver a consultar la fuente de datos cuando otros filtros cambian.

Como puede ver, la configuración de “solo valores relevantes” puede ser muy útil para ayudar al usuario a hacer selecciones relevantes, pero puede aumentar significativamente la cantidad de consultas que deben ejecutarse mientras interactúan con el dashboard. Debe usarse con moderación.

### Alternativas a las tarjetas de filtros

Existen alternativas al uso de tarjetas de filtros que proporcionan un resultado analítico similar, pero sin la sobrecarga de consultas adicionales. Por ejemplo, puede crear un parámetro y un filtro sobre la base de las selecciones de los usuarios.

- VENTAJAS:
  - Los parámetros no requieren una consulta a la fuente de datos antes de la representación.
  - Los parámetros, junto con los campos calculados, pueden implementar lógicas más complejas que las posibles con un filtro de campo simple.
  - Los parámetros se pueden usar para filtrar varias fuentes de datos (en las versiones anteriores a Tableau 10, los filtros solo operan dentro de una fuente de datos). En Tableau 10 y versiones posteriores, los filtros se pueden configurar para que funcionen en fuentes de datos relacionadas.
- DESVENTAJAS:
  - Los parámetros son valores únicos: no los puede usar si desea que el usuario seleccione varios valores.
  - Los parámetros no son dinámicos: la lista de valores se define cuando estos se crean y no se actualiza en función de los valores del DBMS.

Otra alternativa es usar acciones de filtro entre vistas:

- VENTAJAS:
  - Las acciones admiten la selección de múltiples valores, usando lazos visuales o presionando CTRL o MAYÚS.
  - Las acciones muestran una lista dinámica de valores que se evalúa en tiempo de ejecución.
  - Las acciones se pueden usar para filtrar varias fuentes de datos (en las versiones anteriores a Tableau 10, los filtros solo operan dentro de una fuente de datos). En Tableau 10 y versiones posteriores, los filtros se pueden configurar para que funcionen en fuentes de datos relacionadas.
- DESVENTAJAS:
  - Las acciones de filtro son más complejas de establecer que las tarjetas de filtros.

- Las acciones no tienen la misma interfaz de usuario que los parámetros ni las tarjetas de filtros; generalmente requieren más espacio en la pantalla para mostrarse.
- La hoja de origen de la acción aún debe hacer las consultas a la fuente de datos, pero se beneficia del almacenamiento en caché de los canales de procesamiento de Tableau.

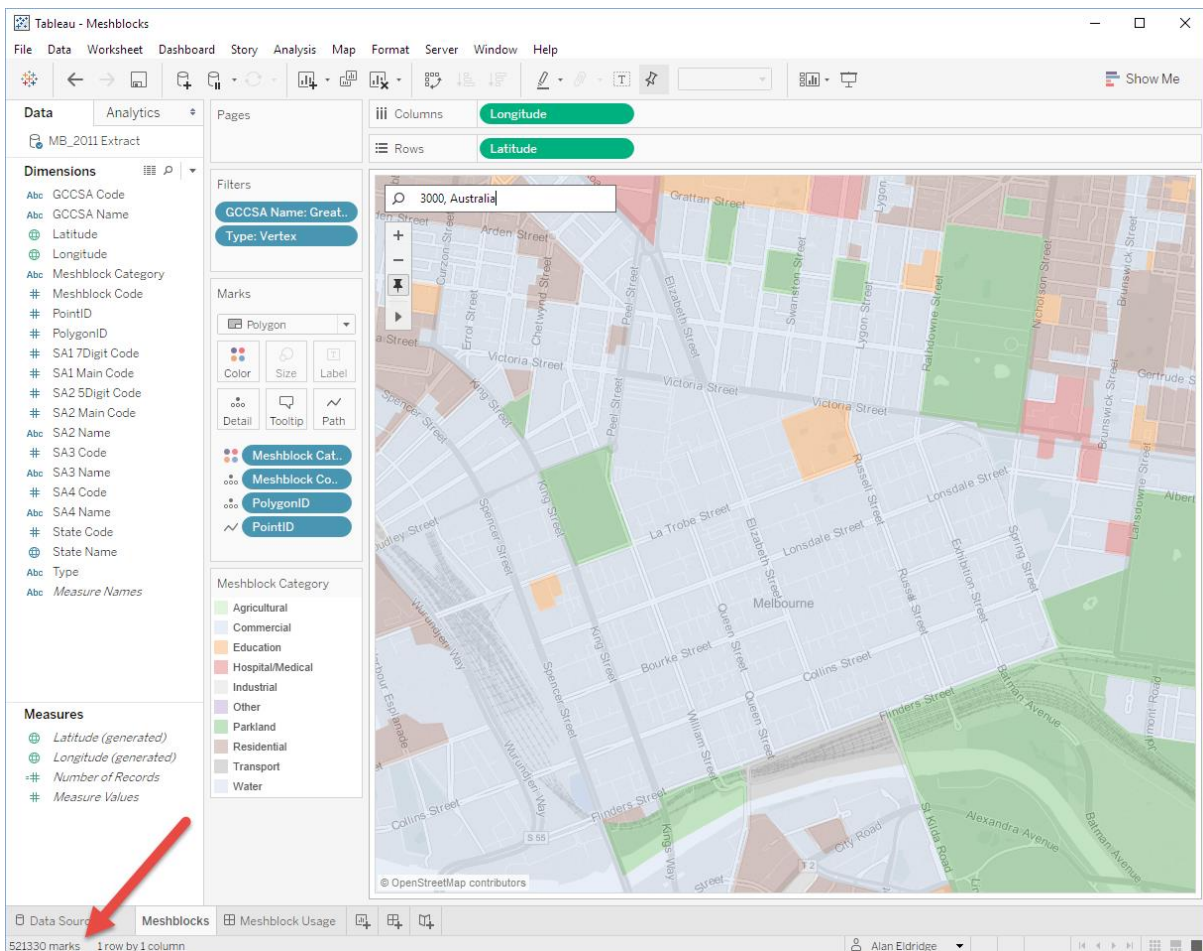
Para saber más sobre las técnicas de diseño alternativas que no dependen en gran medida de las tarjetas de filtros, consulte la [sección anterior](#) sobre un buen diseño de dashboards.

### Filtros de usuarios

Cuando un libro de trabajo contiene filtros de usuarios (creados a través del cuadro de diálogo “Crear Filtro de usuarios” o mediante campos calculados que utilizan cualquiera de las funciones de usuario incorporadas, como ISMEMBEROF), la caché de modelos no se comparte entre las diversas sesiones de usuarios. Esto puede traer como resultado una tasa de reutilización de la caché mucho menor, lo que, a su vez, implicará más procesamiento de Tableau Server. Utilice estos tipos de filtros con precaución.

### Zoom y filtros

Cuando usa el zoom para acercar una visualización con una gran cantidad de marcas, no se excluyen las marcas que no puede ver. Lo único que se modifica es la ventana gráfica a través de la cual se ven los datos. No se modifica la cantidad total de marcas que se están manipulando, como puede ver en la imagen a continuación:



Si solo necesita un subconjunto de datos, filtre los datos no deseados y deje que la funcionalidad de zoom automático de Tableau configure la ventana gráfica.

## ¿El problema son mis cálculos?

En muchos casos, sus datos de origen no proporcionarán todos los campos que necesita para responder todas sus preguntas. Los campos calculados lo ayudarán a crear todas las dimensiones y medidas necesarias para su análisis.

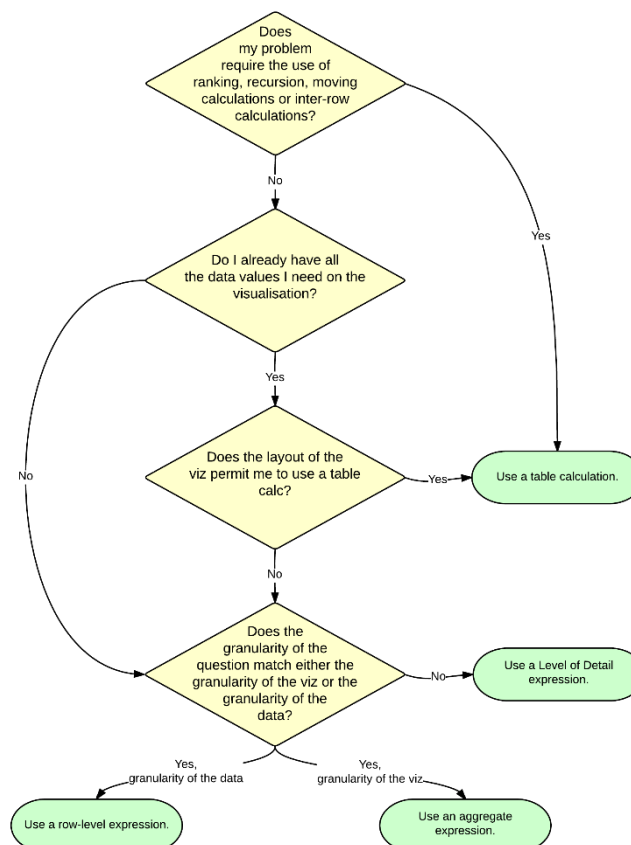
En un campo calculado puede definir una constante codificada de forma rígida (p. ej., tasa impositiva), hacer operaciones matemáticas muy simples, como restas o multiplicaciones (p. ej., ingresos menos costos), usar fórmulas matemáticas más complejas, hacer pruebas de lógica (p. ej., IF/THEN, CASE), hacer conversiones de tipo y mucho más.

Una vez definido, el campo calculado está disponible en todo el libro de trabajo, siempre que las hojas de trabajo usen la misma fuente de datos. Puede usar campos calculados en su libro de trabajo de la misma manera que usa dimensiones y medidas de sus datos de origen.

Hay cuatro tipos de cálculos diferentes en Tableau:

- Cálculos de nivel de fila
- Cálculos de agregación
- Cálculos de tablas
- Expresiones de nivel de detalle (LOD)

Utilice el siguiente diagrama de flujo como guía para seleccionar el mejor enfoque:



La Calculation Reference Library (Biblioteca de referencia de cálculos) de Tableau aporta un excelente material de referencia para aprender a realizar cálculos complejos y contiene un foro donde los usuarios comparten soluciones a problemas comunes: <http://tabsoft.co/110SsWz>

## Tipos de cálculo

### Cálculos de nivel de fila y de agregación

Los cálculos de nivel de fila y los de agregación se expresan como parte de la consulta que se envía a la fuente de datos y, por lo tanto, la base de datos los resuelve. Por ejemplo, una visualización en la que se muestra la suma total de ventas anual de los datos sobre pedidos enviaría la consulta siguiente a la fuente de datos:

```
SELECT DATEPART(year,[OrdersFact].[Order Date]) AS [yr:Order Date:ok],
SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY DATEPART(year,[OrdersFact].[Order Date])
```

El cálculo de **YEAR** es de nivel de fila, y el de **SUM(SALES)** es de agregación.

En general, los cálculos de nivel de fila y los de agregación permiten un correcto ajuste de escala, y existen muchas técnicas de ajuste de bases de datos que pueden emplearse para mejorar su rendimiento.

Tenga en cuenta que, en Tableau, es posible que la consulta de base de datos real no sea una traducción directa de los cálculos básicos utilizados en la visualización. Por ejemplo, la consulta siguiente se ejecuta cuando la visualización contiene un campo calculado de porcentaje de beneficio, definido como `SUM([Profit])/SUM([Sales])`:

```
SELECT DATEPART(year,[OrdersFact].[Order Date]) AS [yr:Order Date:ok],
SUM([OrdersFact].[Profit]) AS
[TEMP(Calculation_0260604221950559)(1796823176)(0)],
SUM([OrdersFact].[Sales]) AS
[TEMP(Calculation_0260604221950559)(3018240649)(0)]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY DATEPART(year,[OrdersFact].[Order Date])
```

En realidad, Tableau toma los elementos del cálculo y realiza la función de división en el nivel de cliente. Esto asegura que `SUM([Profit])` y `SUM([Sales])` en el nivel de año se almacenen en la memoria caché y se puedan utilizar para otras operaciones dentro del libro de trabajo, sin tener que acceder nuevamente a la fuente de datos.

Por último, la fusión de consultas (combinación de múltiples consultas lógicas en una única consulta) puede modificar la consulta que se ejecuta en la fuente de datos. Puede dar como resultado la combinación de varias medidas de diversas hojas de trabajo en una única consulta cuando esas otras hojas de trabajo comparten la misma granularidad.

### Uso de ATTR()

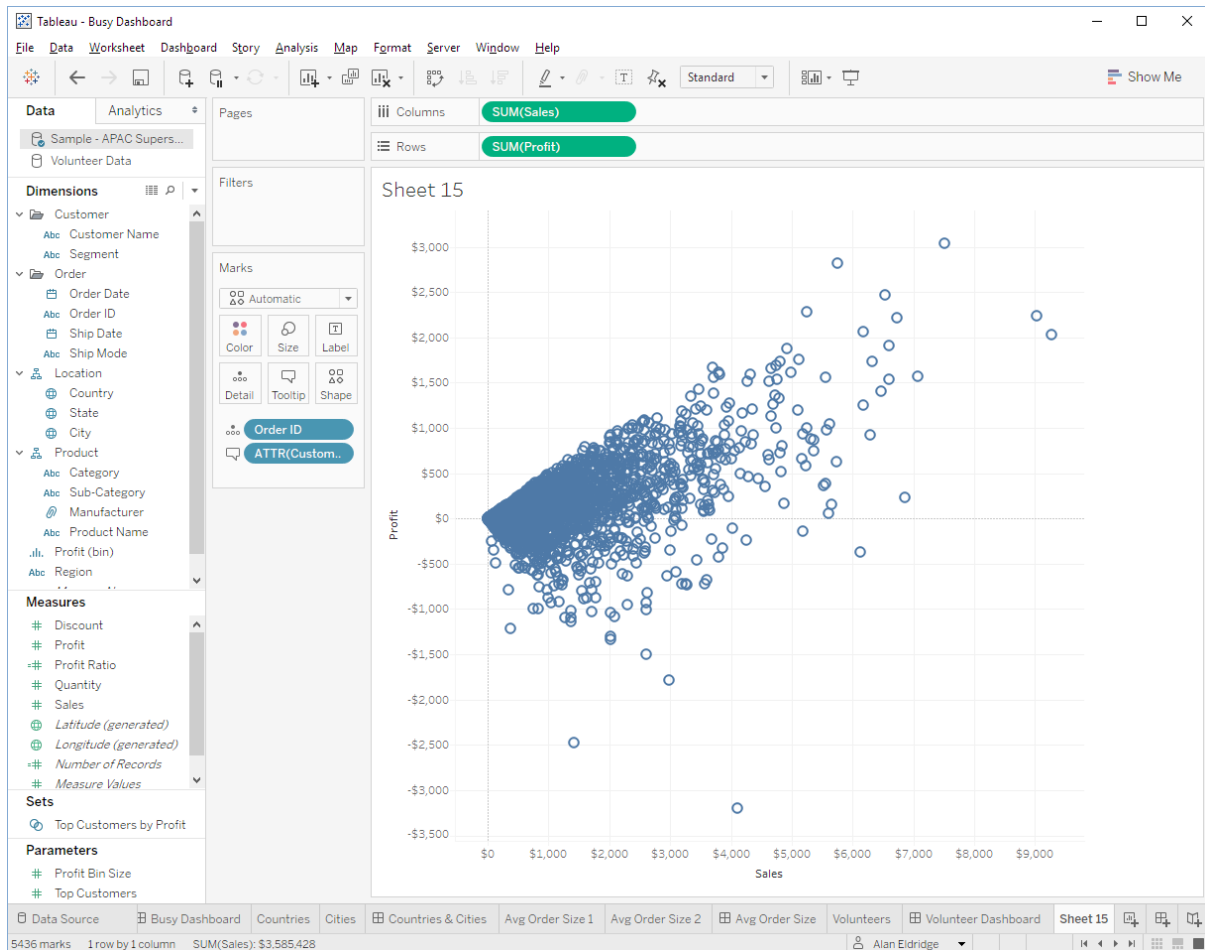
ATTR() es una función eficaz, que a menudo se usa como agregación en las dimensiones categóricas. En términos sencillos, devuelve un valor si es el único; de lo contrario, devuelve \*. Técnicamente, la función lleva a cabo la prueba lógica siguiente:

```
IF MIN([dimension]) = MAX([dimension])
THEN [dimension]
ELSE "*"
END
```

Esto requiere que se realicen dos agregaciones en la fuente de datos subyacente –MIN() y MAX()– y que ambos resultados se pasen nuevamente al conjunto de resultados. Si no le preocupa la posibilidad de tener múltiples valores de dimensión, una solución más eficaz es simplemente utilizar una sola

agregación: MIN() o MAX(). Independientemente de la que utilice, una vez elegida, solo debe usar esa para maximizar sus posibilidades de emplear los datos almacenados en caché.

Para mostrar el impacto de ATTR() en la fuente de datos, creamos la visualización a continuación:



De forma predeterminada, cuando se coloca una dimensión en el estante de descripción emergente, se emplea ATTR() como agregación predeterminada. Esto produce la consulta siguiente:

```
SELECT [Superstore APAC].[Order ID] AS [Order ID],
       MAX([Superstore APAC].[Customer Name]) AS [TEMP(attr:Customer
Name:nk) (2542222306) (0)],
       MIN([Superstore APAC].[Customer Name]) AS [TEMP(attr:Customer
Name:nk) (3251312272) (0)],
       SUM([Superstore APAC].[Profit]) AS [sum:Profit:ok],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Order ID]
```

Si sabemos que no hay varios clientes para cada ID de pedido, podemos cambiar la agregación a MIN() y, así, simplificar la consulta:

```
SELECT [Superstore APAC].[Order ID] AS [Order ID],
       MIN([Superstore APAC].[Customer Name]) AS [min:Customer Name:nk],
       SUM([Superstore APAC].[Profit]) AS [sum:Profit:ok],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Order ID]
```

En la tabla a continuación, se muestran los diferentes rendimientos de las dos opciones (los resultados expresan los segundos que tarda la consulta en completarse, tomados del archivo de registro):

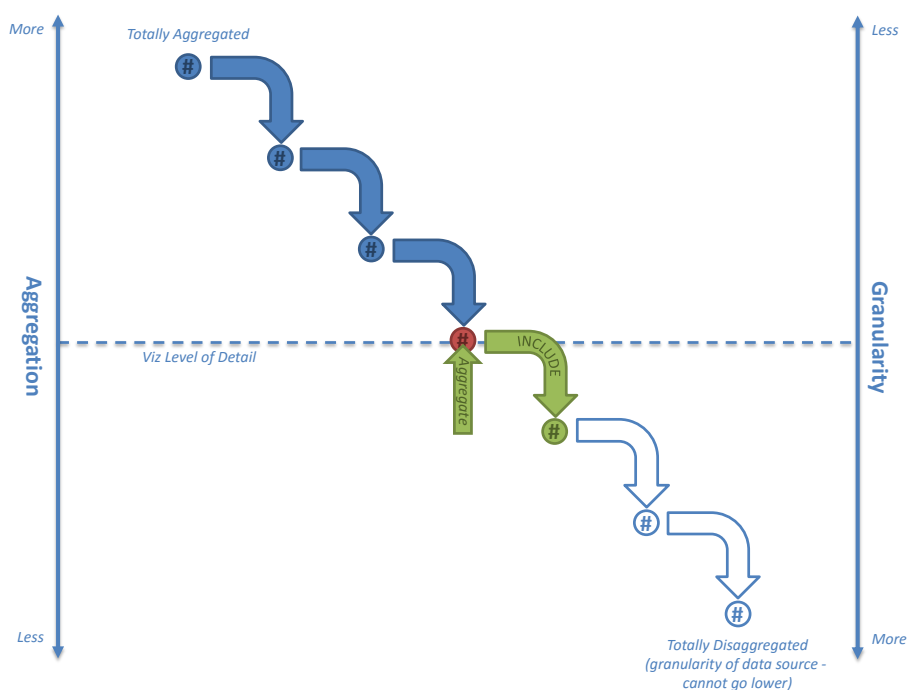
Test #	ATTR	MIN
1	2.824	1.857
2	2.62	2.09
3	2.737	2.878
4	2.804	1.977
5	2.994	1.882
<b>Average</b>	<b>2.7958</b>	<b>2.1368</b>
<b>% improvement</b>		<b>24%</b>

Puede leer más información acerca de la función ATTR() en la publicación de blog siguiente de InterWorks:

<http://bit.ly/1YEuZhX>

### Cálculos de nivel de detalle

Las expresiones de nivel de detalle (LOD) le permiten definir el nivel de detalle que debe tener un cálculo, independientemente del nivel de detalle de la visualización.



En algunos casos, es posible que las expresiones LOD le permitan reemplazar cálculos que quizás haya creado con mucha más dificultad en versiones anteriores de Tableau:

- Es posible que haya intentado encontrar el primer o el último período de una partición con cálculos de tablas. Por ejemplo, puede que haya deseado calcular la cantidad de integrantes de una organización el primer día de cada mes.
- Es posible que haya intentado agrupar campos agregados con cálculos de tablas, campos calculados y líneas de referencia. Por ejemplo, puede que haya deseado encontrar el promedio de un recuento específico de clientes.

- Es posible que haya intentado aplicar filtros de fechas relativas a la máxima fecha de los datos mediante la combinación de datos. Por ejemplo, si sus datos se actualizan todas las semanas, puede que haya deseado calcular los totales del año hasta la fecha de acuerdo a la fecha máxima.

A diferencia de los cálculos de tablas, las expresiones LOD se generan como parte de la consulta a la fuente de datos subyacente. Se expresan como selecciones anidadas, de modo que dependen del rendimiento del sistema de administración de bases de datos (DBMS):

```
SELECT T1.[State],SUM(T2.[Sales per County, State])
FROM [DB Table] T1 INNER JOIN
    (SELECT [State], [County], AVG([Sales]) AS [Sales per County, State]
    FROM [DB Table] GROUP BY [State],[County]) T2
ON T1.[State] = T2.[State]
GROUP BY T1.[State]
```

Esto implica que puede haber situaciones en las que es más eficaz solucionar un problema mediante uno u otro enfoque. Es decir, es posible que un cálculo de tablas o una combinación tenga mejor rendimiento que una expresión LOD, o viceversa. Si sospecha que el rendimiento es lento debido a una expresión LOD, intente reemplazarla con un cálculo de tablas o una combinación de datos (de ser posible) para determinar si mejora el rendimiento. Además, la selección de uniones puede tener un gran impacto en las expresiones LOD, por lo que le sugerimos que relea dicha sección si descubre que las consultas se ejecutan con lentitud cuando se utilizan expresiones LOD.

Para comprender mejor la mecánica de las expresiones LOD, lea el informe “Comprensión de las expresiones de nivel de detalle (LOD)”:

<http://www.tableau.com/es-es/learn/whitepapers/understanding-lod-expressions>

También puede leer la publicación de blog “Las 15 expresiones LOD principales”, escrito por Bethany Lyon, que proporciona ejemplos para una serie de problemas comunes:

<http://www.tableau.com/es-es/about/blog/LOD-expressions>

Finalmente, hay muchas publicaciones de blog de la comunidad sobre este tema, que son muy útiles. Puede encontrar una selección maravillosa en *data + science*:

<http://bit.ly/1MpkFV5>

### Cálculos de tablas

A diferencia de los cálculos de nivel de fila o de agregación, los cálculos de tablas no los ejecuta la base de datos, sino que los calcula Tableau dentro del conjunto de resultados de la consulta. Si bien esto implica más trabajo para Tableau, este trabajo suele realizarse sobre un conjunto de registros mucho más pequeño que el que se encuentra en la fuente de datos original.

Si tiene problemas con el rendimiento de los cálculos de tablas (posiblemente porque el conjunto de resultados que se presenta a Tableau es muy grande), considere enviar algunos aspectos de los cálculos al nivel de fuente de datos. Una manera de hacerlo es a través de una expresión de nivel de detalle. Otra manera es aprovechar una extracción de datos agregados. Imagine que usted desea conocer el promedio semanal del total de ventas diarias en varias tiendas. Puede usar un cálculo de tablas con:

```
WINDOW_AVG(SUM([Sales]))
```



Sin embargo, si la cantidad de días o de tiendas es muy grande, este cálculo puede ser lento. Para enviar SUM([Sales]) a la capa de datos, cree una extracción agregada que resuma la dimensión de fecha al nivel del día. Luego, el cálculo se puede hacer simplemente con AVG([Sales]), puesto que la extracción ya calculó los totales diarios.

En algunos casos de uso, es posible que sepamos que el valor del factor agregado no cambia en la partición. Cuando esto ocurra, utilice MIN() o MAX() como función de agregación, en vez de AVG() o ATTR(), ya que son más rápidas de evaluar. Y, una vez que haya elegido una, siempre utilice la misma para mejorar sus probabilidades de reutilizar los resultados almacenados en la memoria caché.

### Cálculos externos

Tableau cuenta con un mecanismo en el que las operaciones lógicas complejas se transfieren a un motor externo para su cálculo. Entre estas, se incluyen las llamadas a R (a través de la conexión Rserve) o Python (mediante una conexión del servidor Dato).

Dentro de Tableau, estas operaciones son similares a los cálculos de tablas, por lo que se calculan entre particiones. Esto implica que es posible que deban llamarse varias veces para una única visualización, lo que puede sobrecargar el rendimiento. Tampoco están optimizadas ni fusionadas de ninguna manera, así que piense si puede obtener múltiples valores de retorno en una sola función (por ejemplo, en una cadena concatenada), en lugar de generar varias llamadas de función. Por último, también puede significar una sobrecarga el tiempo de transferencia de datos hacia el motor de cálculo externo y desde él.

### Análisis

El panel Análisis proporciona un acceso rápido a una variedad de análisis avanzados, como los que se presentan a continuación:

- Totales
- Líneas de referencia
- Líneas de tendencia
- Estimaciones
- Agrupación en clústeres (novedad de Tableau 10)

Estos análisis generalmente no requieren la ejecución de consultas adicionales. Se comportan de manera similar a los cálculos de tablas en los datos de la memoria caché de resultados de consultas. Al igual que los cálculos de tablas, si tiene un conjunto de datos muy grande en los resultados de la consulta, el proceso de cálculo puede llevar un tiempo considerable. Sin embargo, por lo general, no es un factor que afecte demasiado el rendimiento general de un libro de trabajo.

Otro factor que puede afectar el rendimiento, para los totales y las líneas de referencia, es si las agregaciones de las medidas son acumulativas o no. Para las agregaciones acumulativas (p. ej., SUM, MIN, MAX), el cálculo de totales o líneas de referencia se puede realizar de manera local en Tableau. Para las agregaciones no acumulativas (p. ej., COUNTD, TOTAL, MEDIAN, PERCENTILE), debemos volver a la fuente de datos para calcular los valores del total o la línea de referencia.

### Cálculos y características nativas

A veces, los usuarios crean campos calculados para llevar a cabo funciones que pueden realizarse fácilmente con características nativas de Tableau. Por ejemplo:

- Para agrupar miembros de dimensiones, considere usar [grupos](#) o [conjuntos](#).
- Para agrupar valores de medidas en bandas con intervalos, considere usar [agrupaciones](#).



- Para truncar fechas con una granularidad mayor (p. ej., por mes o semana), considere usar [campos de fecha personalizada](#).
- Para crear un conjunto de valores que sea la concatenación de dos dimensiones diferentes, considere usar [campos combinados](#).
- Para mostrar fechas con un formato específico o convertir valores numéricos en indicadores KPI, considere utilizar las funcionalidades de formato incorporadas.
- Para cambiar los valores mostrados de miembros de dimensiones, considere usar [alias](#).

Esto no es siempre posible (p. ej., puede necesitar agrupaciones de ancho variable, que no son posibles con agrupaciones básicas), pero considere usar las características nativas cuando sí lo sea. Usarlas suele ser más eficaz que hacer cálculos manuales y, a medida que nuestros desarrolladores continúen mejorando el rendimiento de Tableau, usted se beneficiará de sus esfuerzos.

### Impacto de los tipos de datos

Cuando se crean campos calculados, es importante entender que el tipo de datos tiene un impacto significativo en la velocidad de los cálculos. Como regla general:

- Los valores enteros y booleanos son mucho más rápidos que las cadenas y las fechas.

Los cálculos de cadena y fecha son muy lentos; suele haber 10-100 instrucciones básicas que deben ejecutarse en cada cálculo. En comparación, los cálculos numéricos y los booleanos son muy eficaces.

Estas afirmaciones no solo son verdaderas para el motor de cálculo de Tableau, sino que lo son para la mayoría de las bases de datos. Como los cálculos de nivel de fila y los agregados se envían a la base de datos, si se expresan de manera numérica, en vez de con la lógica de cadenas, se ejecutarán mucho más rápido.

### Técnicas para mejorar el rendimiento

Considere las siguientes técnicas para garantizar que sus cálculos sean lo más eficaces posible:

#### Uso de valores booleanos para cálculos de lógica básica

Si tiene un cálculo que produce un resultado binario (p. ej., sí/no, aprobado/no aprobado, arriba/abajo), asegúrese de presentar un resultado booleano en lugar de una cadena. Por ejemplo:

```
IF [Date] = TODAY() THEN "Today"
  ELSE "Not Today"
END
```

Esto será lento porque usa cadenas. Una forma más rápida de hacer lo mismo sería con un valor booleano:

```
[Date] = TODAY()
```

Después, use alias para cambiar el nombre de los resultados TRUE y FALSE a "Today" y "Not Today".

#### Búsqueda de cadenas

Imagine que desea poder mostrar todos los registros en los que el nombre del producto contenga alguna cadena de búsqueda. Puede usar un parámetro para obtener la cadena de búsqueda del usuario y, después, crear el siguiente campo calculado:

```
IF FIND([Product Name],[Product Lookup]) > 0
  THEN [Product Name]
  ELSE NULL
END
```

Este cálculo es lento y no es eficaz para evaluar la relación de contención. Una mejor manera de hacer esto sería usar la función específica CONTAINS, ya que se convertirá en SQL óptimo cuando pase a la base de datos:

```
CONTAINS ([Product Name], [Product Lookup])
```

No obstante, en este caso, la mejor solución sería no usar un campo calculado en absoluto, sino usar una tarjeta de filtros de concordancia con carácter comodín.

### Parámetros para cálculos condicionales

Una técnica popular en Tableau es proporcionar al usuario final un parámetro para que pueda seleccionar un valor que determinará cómo se realiza un cálculo. Como ejemplo, imagine que desea permitir al usuario final controlar el nivel de agregación de fecha que se muestra en la vista mediante la selección a partir de una lista de valores posibles. Muchas personas crearían un parámetro de cadena:

```
Value
Year
Quarter
Month
Week
Day
```

Luego, lo usarían en un cálculo como este:

```
CASE [Parameters].[Date Part Picker]
  WHEN "Year" THEN DATEPART('year', [Order Date])
  WHEN "Quarter" THEN DATEPART('quarter', [Order Date])
  ..
END
```

Un método mejor sería utilizar la función incorporada DATEPART() y crear el cálculo como se muestra a continuación:

```
DATEPART(LOWER([Parameters].[Date Part Picker]), [Order Date])
```

En las versiones anteriores de Tableau, el último enfoque era mucho más rápido. Sin embargo, ya no hay diferencias en el rendimiento entre estas dos soluciones, puesto que contraemos la lógica condicional de la declaración CASE a partir del valor del parámetro y solo se pasa la función DATEPART() adecuada a la fuente de datos.

No olvide que también debe pensar en la capacidad de mantenimiento de la solución en el futuro. En ese caso, el último cálculo probablemente sea la mejor opción.

### Conversión de fechas

A menudo, los usuarios tienen datos de fechas que no están almacenados en formato de fecha nativo, por ejemplo, pueden ser cadenas o marcas de tiempo numéricas. La función DATEPARSE() facilita estas conversiones. Simplemente puede especificar una cadena de formato:

```
DATEPARSE("yyyyMMdd", [YYYYMMDD])
```

Tenga en cuenta que DATEPARSE() solo se admite en un subconjunto de fuentes de datos:

- Conexiones con archivos de Excel o de texto no heredados
- MySQL
- Oracle

- PostgreSQL
- Extracción de datos de Tableau

Si DATEPARSE() no se admite para su fuente de datos, una técnica alternativa de conversión a fechas compatibles con Tableau es redistribuir el campo en una cadena de fecha (p. ej., "2012-01-01"; tenga en cuenta que se prefieren cadenas ISO porque se ajustan a la internacionalización) y, después, incluirlo en la función DATE().

Si los datos originales son campos numéricos, la conversión primero a una cadena y después a una fecha es muy ineficaz. Es mucho mejor mantener los datos en formato numérico y usar DATEADD() y los valores literales de fechas para hacer el cálculo.

Por ejemplo, un cálculo lento podría ser convertir un campo numérico con formato ISO:

```
DATE(LEFT(STR([YYYYMMDD]), 4)
+ "-" + MID(STR([YYYYMMDD]), 4, 2)
+ "-" + RIGHT(STR([YYYYMMDD]), 2))
```

Una forma más eficaz de hacer este cálculo es:

```
DATEADD('day', INT([yyyymmdd])% 100 - 1,
DATEADD('month', INT([yyyymmdd]) % 10000 / 100 - 1,
DATEADD('year', INT([yyyymmdd]) / 10000 - 1900,
#1900-01-01#)))
```

Una manera incluso mejor sería utilizar la función MAKEDATE() si es compatible con su fuente de datos:

```
MAKEDATE(2004, 4, 15)
```

Tenga en cuenta que las mejoras de rendimiento pueden ser extraordinarias con grandes conjuntos de datos. Para una muestra de más de mil millones de registros, el primer cálculo llevó 4 horas, mientras que en el segundo se tardó alrededor de un minuto.

## Declaraciones lógicas

### *Pruebe primero el resultado más frecuente*

Cuando Tableau evalúa una prueba lógica, detiene el procesamiento de los resultados posibles cuando encuentra una concordancia. Esto implica que usted debería probar primero el resultado más probable, de manera que, para la mayoría de los casos que se evalúan, se detengan después de la primera prueba.

Observe el ejemplo siguiente: Esta lógica:

```
IF <unlikely_test>
THEN <unlikely_outcome>
ELSEIF <likely_test>
THEN <likely_outcome>
END
```

Tiene una evaluación más lenta que esta:

```
IF <likely_test>
THEN <likely_outcome>
ELSEIF <unlikely_test>
THEN <unlikely_outcome>
END
```

## ELSEIF > ELSE IF

Cuando trabaje con declaraciones de lógica compleja, recuerde que ELSEIF > ELSE IF. Esto sucede porque una declaración IF anidada procesa una declaración CASE anidada en la consulta resultante, en vez de procesarse como parte de la primera. Entonces, este campo calculado:

```
IF [Region] = "East" AND [Segment] = "Consumer"
  THEN "East-Consumer"
  ELSE IF [Region] = "East" and [Segment] <> "Consumer"
    THEN "East-All Others"
  END
END
```

Genera el código SQL siguiente, con dos declaraciones **CASE** anidadas y cuatro pruebas condicionales:

```
(CASE
  WHEN (([Global Superstore].[Region] = 'East')
  AND ([Global Superstore].[Segment] = 'Consumer'))
  THEN 'East-Consumer'
  ELSE (CASE
    WHEN (([Global Superstore].[Region] = 'East')
    AND ([Global Superstore].[Segment] <> 'Consumer'))
    THEN 'East-All Others'
    ELSE CAST(NULL AS NVARCHAR)
  END)
END)
```

Esto se ejecutaría con mayor velocidad si se reescribiera del siguiente modo, con ELSEIF en vez de una declaración IF anidada y con un uso más eficaz de las pruebas condicionales:

```
IF [Region] = "East" AND [Segment] = "Consumer"
  THEN "East-Consumer"
  ELSEIF [Region] = "East"
    THEN "East-All Others"
  END
```

Esto produce una única declaración CASE en el código SQL:

```
(CASE
  WHEN (([Global Superstore].[Region] = 'East')
  AND ([Global Superstore].[Segment] = 'Consumer'))
  THEN 'East-Consumer'
  WHEN ([Global Superstore].[Region] = 'East')
  THEN 'East-All Others'
  ELSE CAST(NULL AS NVARCHAR)
END)
```

Sin embargo, esto sigue siendo rápido. A pesar de que no usa una declaración IF anidada, es el uso más eficaz de las pruebas condicionales:

```
IF [Region] = "East" THEN
  IF [Segment] = "Consumer"
    THEN "East-Consumer"
    ELSE "East-All Others"
  END
END
```

El código SQL resultante:

```
(CASE
  WHEN ([Global Superstore].[Region] = 'East')
  THEN (CASE
```

```

        WHEN ([Global Superstore].[Segment] = 'Consumer')
            THEN 'East-Consumer'
            ELSE 'East-All Others'
        END)
    ELSE CAST(NULL AS NVARCHAR)
END)

```

### *Evite las verificaciones de lógica redundante*

De manera similar, evite las verificaciones de lógica redundante. El siguiente cálculo:

```

IF [Sales] < 10 THEN "Bad"
ELSEIF [Sales] >= 10 AND [Sales] < 30 THEN "OK"
ELSEIF [Sales] >= 30 THEN "Great"
END

```

Genera el código SQL siguiente:

```

(CASE
    WHEN ([Global Superstore].[Sales] < 10)
        THEN 'Bad'
    WHEN (([Global Superstore].[Sales] >= 10)
        AND ([Global Superstore].[Sales] < 30))
        THEN 'OK'
    WHEN ([Global Superstore].[Sales] >= 30)
        THEN 'Great'
    ELSE CAST(NULL AS NVARCHAR)
END)

```

Esto sería más eficaz así:

```

IF [Sales] < 10 THEN "Bad"
ELSEIF [Sales] >= 30 THEN "Great"
ELSE "OK"
END

```

Genera el código SQL siguiente:

```

(CASE
    WHEN ([Global Superstore].[Sales] < 10)
        THEN 'Bad'
    WHEN ([Global Superstore].[Sales] >= 30)
        THEN 'Great'
    ELSE 'OK'
END)

```

### Otras pequeñas medidas

Hay muchas pequeñas medidas que pueden tomarse y que tienen un impacto en el rendimiento.

Tenga en cuenta estas sugerencias:

- Trabajar con lógicas de fechas puede ser complicado. En vez de escribir pruebas elaboradas con múltiples funciones de fecha (p. ej., MONTH() y YEAR()), considere usar algunas de las otras funciones incorporadas, como DATETRUNC(), DATEADD() y DATEDIFF(). Pueden disminuir significativamente la complejidad de la consulta generada para la fuente de datos.
- Los diversos valores de conteo son uno de los tipos de agregaciones más lentas en casi todas las fuentes de datos. Utilice la agregación COUNTD con moderación.
- El uso de parámetros con un amplio alcance de impacto (p. ej., en declaraciones de SQL personalizado) pueden afectar el rendimiento de la memoria caché.

- El filtrado con base en cálculos complejos puede causar que se desestimen índices en la fuente de datos subyacente.
- Use NOW() solo si necesita el nivel de detalle de marca de tiempo. Use TODAY() para cálculos con nivel de fecha.
- Recuerde que todos los cálculos básicos se procesan a través de la fuente de datos subyacente, incluso los cálculos literales, como las cadenas de etiquetas. Si debe crear etiquetas (p. ej., para los encabezados de columnas) y su fuente de datos es muy grande, genere una fuente de datos simple en un archivo Excel o de texto, con solo un registro para almacenarlas, para que no sobrecargue la fuente de datos grande. Esto es especialmente importante si su fuente de datos utiliza procedimientos almacenados. Consulte [esta sección](#) para obtener más información.

## ¿El problema son mis consultas?

Una de las funcionalidades más importantes de Tableau es que puede utilizar tanto datos de la memoria (es decir, conexiones de datos extraídos) como datos *in situ* (es decir, conexiones de datos en tiempo real). Las conexiones en tiempo real son una funcionalidad muy eficaz de Tableau, puesto que nos permiten aprovechar el poder de procesamiento de la propia fuente de datos. Sin embargo, debido a que el rendimiento depende de la plataforma de la fuente de datos, es crucial que generemos las consultas a la fuente de datos de la forma más eficaz y óptima posible.

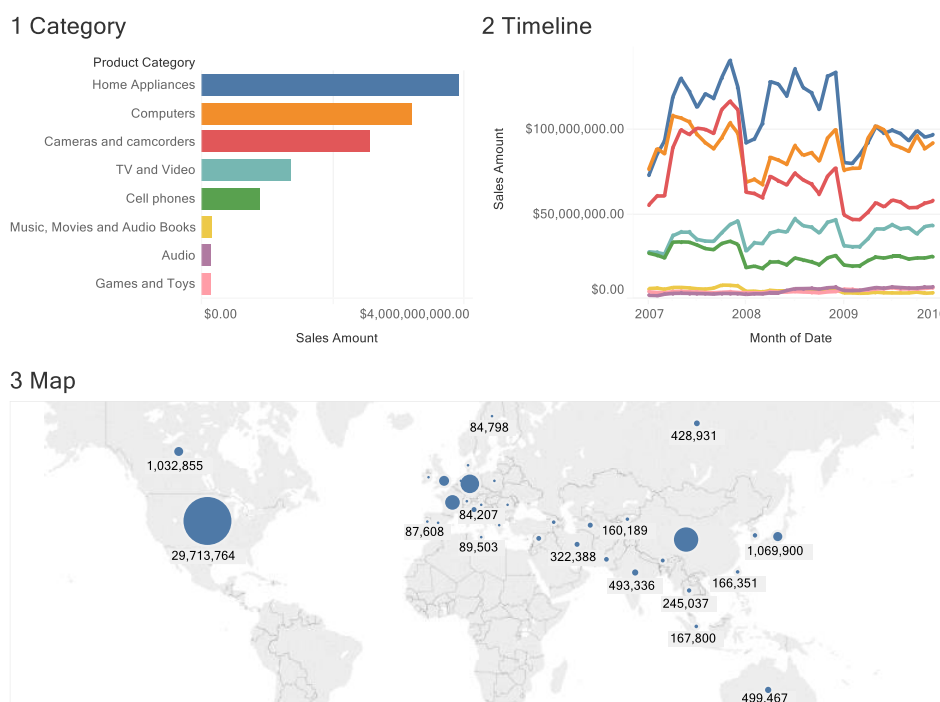
Como ya hemos señalado, cuanto más compleja sea la pregunta, más cantidad de datos se mostrarán y más elementos se incluirán en el dashboard. Todo esto implica mayor trabajo por parte de la fuente de datos. Para que nuestros libros de trabajo sean lo más eficaces posible, debemos minimizar la cantidad de consultas, asegurarnos de que las consultas sean evaluadas con eficacia, minimizar la cantidad de datos devueltos por las consultas y reutilizar los datos entre solicitudes tanto como sea posible.

### Optimizaciones automáticas

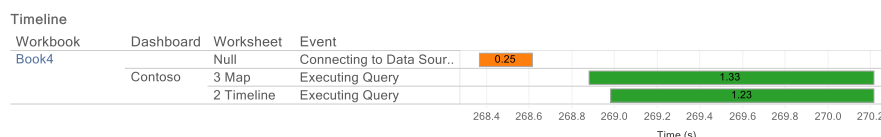
A pesar de que hay muchas medidas que podemos tomar para mejorar de forma deliberada el rendimiento de nuestras consultas, hay muchas optimizaciones que Tableau realizará de manera automática para asegurar que sus libros de trabajo se ejecuten eficazmente. No puede controlar directamente la mayoría de estas optimizaciones y, en la mayoría de los casos, no es necesario que se preocupe por ellas. Pero, si las comprende bien, puede aprovecharlas para mejorar el rendimiento.

### Ejecución paralela: ejecutar varias consultas simultáneamente

Tableau aprovecha la capacidad de las fuentes de datos para ejecutar múltiples consultas a la vez. Observe el dashboard siguiente:



Al abrir este libro de trabajo en Tableau, se muestran las consultas siguientes:



Como puede ver, una ejecución en serie de las consultas tardaría 2,66 segundos. Pero, cuando las ejecuta en paralelo, solo se tiene en cuenta la duración de la consulta que más tardó (1,33 segundos).

El nivel de paralelismo de consultas varía según el sistema de origen con el que se trabaje, puesto que algunas plataformas manejan las consultas simultáneas mejor que otras. De forma predeterminada, se ejecuta un máximo de 16 consultas paralelas para todas las fuentes de datos, sin contar los archivos de texto, Excel ni los estadísticos (cuyo límite es una consulta a la vez). Es posible que otras fuentes de datos tengan configurados límites por debajo del predeterminado. Consulte el enlace siguiente para obtener más detalles:

<http://kb.tableau.com/articles/HowTo/Configuring-Parallel-Queries-in-Tableau-Desktop?lang=es-es>

En la mayoría de los casos, esta configuración no es necesaria y debería dejarla sin modificaciones. Sin embargo, si tiene una necesidad específica de controlar el grado de paralelismo, se puede configurar como:

- Un límite global en la cantidad de consultas paralelas para Tableau Server.
- Límites para un tipo de fuentes de datos en particular, como SQL Server.
- Límites para un tipo de fuentes de datos en particular en un servidor específico.
- Límites para un tipo de fuentes de datos en particular en un servidor específico, cuando se conecta a una base de datos específica.

Esta configuración está establecida en un archivo XML llamado `connection-configs.xml`. Para Tableau Desktop, lo debe crear y almacenar en la carpeta de la aplicación en Windows (`C:\Program Files\Tableau\Tableau 10.0`) y Mac (haga clic con el botón secundario en la aplicación, haga clic en `Mostrar contenido del paquete` y coloque el paquete allí). Para Tableau Server, en el directorio “`config`” ubicado en la carpeta “`vizqlserver`” (por ejemplo: `C:\ProgramData\Tableau\TableauServer\data\tabsvc\config\vizqlserver`). Debe copiar este archivo de configuración a todos los directorios de configuración “`vizqlserver`” en todos los equipos de trabajo.

Puede leer más acerca de cómo configurar las consultas paralelas aquí (se incluye la sintaxis que debe utilizar en el archivo `connection-configs.xml`):

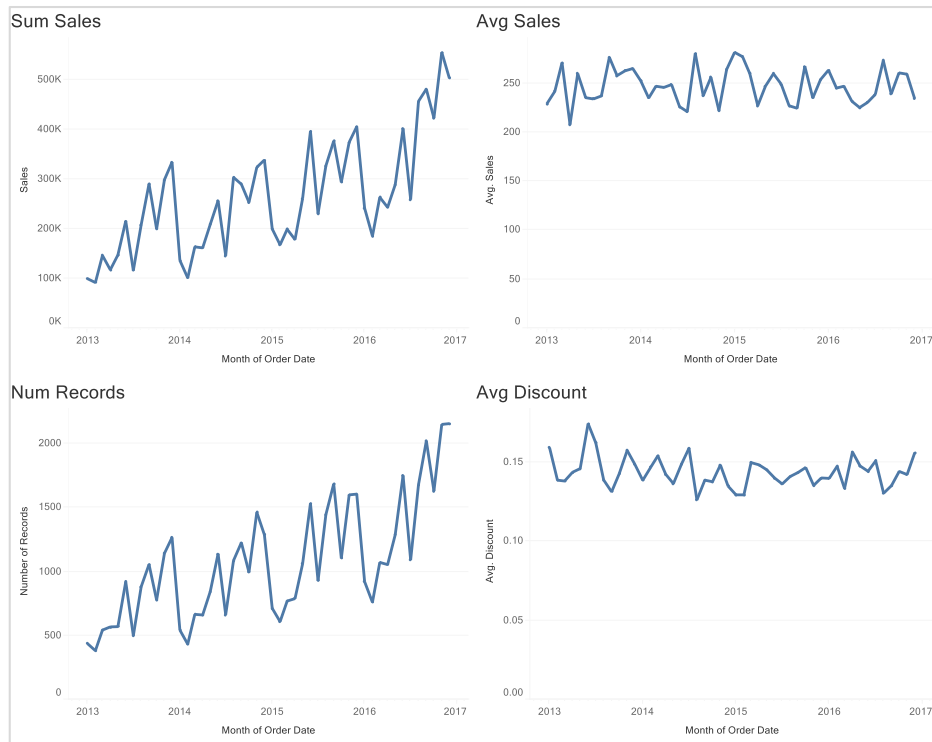
<http://kb.tableau.com/articles/knowledgebase/parallel-queries-tableau-server?lang=es-es>

**Eliminación de consultas: ejecutar menos consultas**

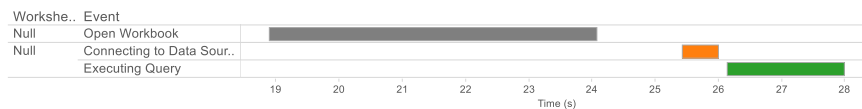
En el ejemplo anterior, también puede ver que solo se ejecutaron dos consultas en vez de tres. Mediante el procesamiento por lotes de consultas, Tableau puede eliminar las consultas redundantes. El optimizador de consultas de Tableau ordenará las consultas para ejecutar las más complejas primero, con la esperanza de que las consultas siguientes puedan reutilizar los datos de la memoria caché de resultados. En el ejemplo, debido a que la escala de tiempo incluye la categoría de producto y porque la suma de ventas es completamente acumulativa, se pueden resolver los datos del gráfico de categoría a partir de la memoria caché de consultas de la hoja de trabajo Escala de tiempo. No es necesario volver a consultar la fuente de datos.

El optimizador de consultas también buscará consultas que tengan el mismo nivel de detalle (es decir, que estén especificadas por el mismo conjunto de dimensiones) y las contraerá en una sola consulta que devuelva todas las medidas solicitadas. Observe el dashboard siguiente:





Como puede ver, este dashboard contiene cuatro hojas. En cada una se muestran diversas medidas en una escala de tiempo. Todas tienen el mismo nivel de detalle, puesto que en ellas se muestran los datos con un mes continuo. En vez de realizar solicitudes diferentes a la base de datos, Tableau las combina en una sola consulta:



La consulta es la siguiente:

```
SELECT AVG(cast([Global Superstore].[Discount] as float)) AS
[avg:Discount:ok],
AVG(cast([Global Superstore].[Sales] as float)) AS [avg:Sales:ok],
SUM(CAST(1 as BIGINT)) AS [sum:Number of Records:ok],
SUM([Global Superstore].[Sales]) AS [sum:Sales:ok],
DATEADD(month, DATEDIFF(month, CAST('0001-01-01 00:00:00' AS datetime2),
[Global Superstore].[Order Date]), CAST('0001-01-01 00:00:00' AS datetime2))
AS [tmn:Order Date:ok]
FROM [dbo].[Global Superstore] [Global Superstore]
GROUP BY DATEADD(month, DATEDIFF(month, CAST('0001-01-01 00:00:00' AS
datetime2), [Global Superstore].[Order Date]), CAST('0001-01-01 00:00:00' AS
datetime2))
```

Como se puede ver, esta optimización (denominada “fusión de consultas”) puede mejorar el rendimiento general de manera significativa. Cuando sea posible, considere la posibilidad de configurar el mismo nivel de detalle para varias hojas de un dashboard.

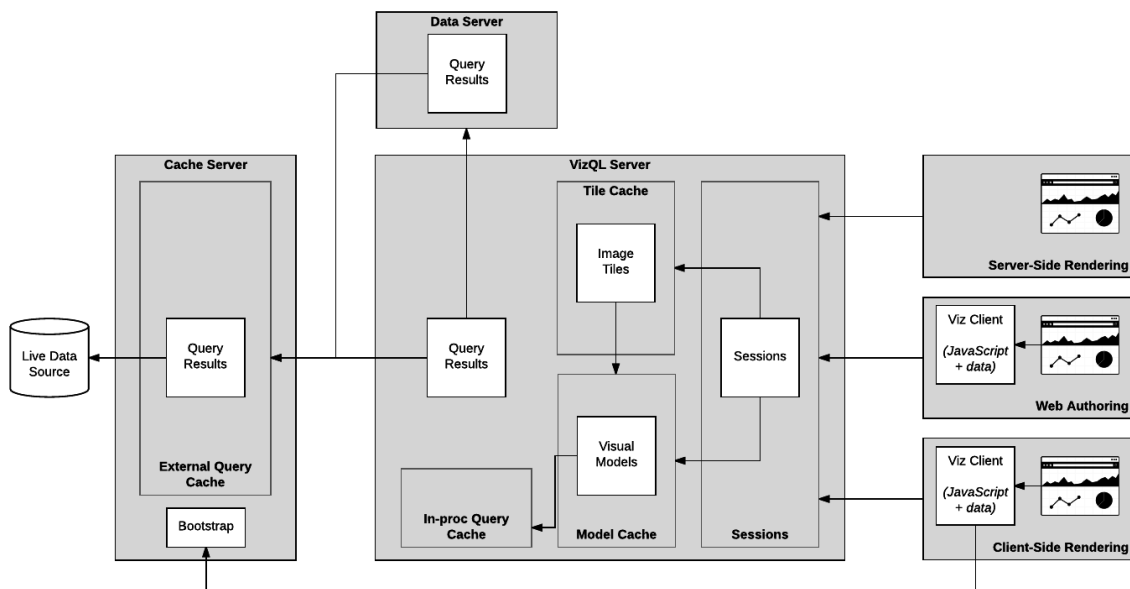
Tenga en cuenta que la fusión de consultas no funciona en las consultas que se realizan en fuentes de datos TDE.

## Almacenamiento en caché: no ejecutar ninguna consulta

¿Qué puede ser mejor que ejecutar menos consultas? No ejecutar ninguna. Tableau realiza un almacenamiento en caché exhaustivo, tanto en Tableau Desktop como en Tableau Server. Esto puede reducir significativamente la cantidad de consultas que se deben ejecutar en la fuente de datos subyacente.

### Tableau Server

El almacenamiento en caché en Tableau Server tiene múltiples capas:



La primera capa del almacenamiento en caché depende de si la sesión utiliza representación del lado del cliente o del lado del servidor. Consulte esta [sección anterior](#) para obtener más información sobre los dos modelos de representación y sobre cómo Tableau determina cuál debe utilizar.

Si la sesión utiliza la representación del lado del cliente, lo primero que necesita el navegador es cargar el cliente del visor. Esto implica cargar un conjunto de bibliotecas de JavaScript y datos, para poder representar la vista inicial (un proceso denominado “arranque”, que puede tardar varios segundos). Sabiendo que es muy probable que los dashboard sean vistos por varios usuarios, y para reducir el tiempo de espera de las solicitudes de vistas siguientes, las respuestas de arranque se almacenan en caché. En primera instancia, cada sesión verifica los datos de arranque almacenados en caché, para determinar si hay alguna respuesta creada que pueda utilizar. Si encuentra una, el navegador simplemente carga la respuesta desde la memoria caché, lo que agiliza considerablemente la representación de la vista inicial. Una vez que el cliente del visor se carga en el navegador del cliente, algunas interacciones se pueden realizar en su totalidad con datos locales (p. ej., el resaltado, las descripciones emergentes), lo que da como resultado una experiencia de usuario rápida y fluida.

Si el dashboard se genera a través de la representación del lado del servidor, el servidor representa los elementos del dashboard como una serie de archivos de imagen (denominados “mosaicos”). Estos mosaicos se transfieren al navegador, donde se ensamblan para mostrar la visualización. Como antes, se espera que el dashboard sea visto por diversos usuarios, por lo que el servidor almacena estas imágenes en la caché del disco. Cada solicitud verifica los mosaicos almacenados en la memoria caché para determinar si la imagen ya se ha representado. En el caso de que la encuentre, simplemente carga el archivo de la caché. Una consulta de los mosaicos almacenados en caché agiliza el tiempo

de respuesta y reduce la carga de trabajo en el servidor. Una forma simple de incrementar la eficacia de la caché de mosaicos es configurar el dashboard para que tenga un [tamaño fijo](#).

En general, la representación del lado del cliente produce interacciones fluidas y con mayor capacidad de respuesta. Además, genera menos carga de trabajo para Tableau Server. Cuando sea posible, debe diseñar sus libros de trabajo para que se representen del lado del cliente.

La capa siguiente de almacenamiento en caché se denomina modelo visual. Un modelo visual describe la manera de representar una hoja de trabajo individual; por esto, la visualización de un dashboard puede implicar varios modelos visuales (uno por cada hoja de trabajo utilizada). Incluye los resultados de los cálculos locales (p. ej., cálculos de tablas, líneas de referencia, estimaciones, agrupaciones en clústeres, etc.) y la disposición visual (cantidad de filas y columnas que se deben mostrar para los múltiplos pequeños y las tabulaciones cruzadas; la cantidad y el intervalo de graduaciones de eje y líneas de cuadrícula que se deben visualizar; la cantidad y la ubicación de las etiquetas de marca que se deben mostrar, etc.).

VizQL Server crea y almacena en la memoria caché los modelos visuales; además, optimiza el proceso de compartir resultados entre las sesiones de usuarios cuando es posible. Los factores clave que determinan si un modelo visual se puede compartir son los siguientes:

- El tamaño del área de disposición de la visualización. Los modelos solo se pueden compartir entre sesiones con un mismo tamaño de visualización. Si configura el dashboard para que tenga un tamaño fijo beneficia tanto a la memoria caché de modelos como a la memoria caché de mosaicos. Esto permite una mayor reutilización de los datos y reduce la carga de trabajo en el servidor.
- Si hay concordancia entre las selecciones y los filtros. Dependiendo de si el usuario cambia filtros, parámetros, realiza exploraciones u otras acciones, el modelo únicamente se compartirá con las sesiones que tengan el mismo estado de vista. Intente no publicar libros de trabajo que tengan activada la opción “Mostrar selecciones”, puesto que esto puede reducir la probabilidad de concordancia entre diferentes sesiones.
- Las credenciales utilizadas para conectarse a la fuente de datos. Si se le solicita a los usuarios que especifique las credenciales para conectarse a la fuente de datos, el modelo solo se puede compartir entre sesiones de usuarios con las mismas credenciales. Use esto con precaución, puesto que puede reducir la eficacia de la memoria caché de modelos.
- Si se habilitó el filtro de usuarios. Si los libros de trabajo contienen filtros de usuarios o tienen cálculos con funciones como USERNAME() o ISMEMBEROF(), el modelo no se comparte con ninguna otra sesión de usuario. Use estas funciones con precaución, puesto que pueden reducir significativamente la eficacia de la memoria caché de modelos.

La última capa de almacenamiento en caché es la memoria caché de consultas. Almacena los resultados de las consultas que se ejecutaron anteriormente y que pueden ser útiles en futuras consultas. El empleo de esta memoria caché es muy eficaz, porque nos permite evitar la ejecución repetida de consultas en la fuente de datos. Solo se cargan los datos de la memoria caché. Además nos permite, en algunos casos, responder consultas con el resultado de otra consulta. Ya mencionamos los beneficios de esto [anteriormente](#), en la sección sobre la eliminación de consultas y la fusión de consultas.

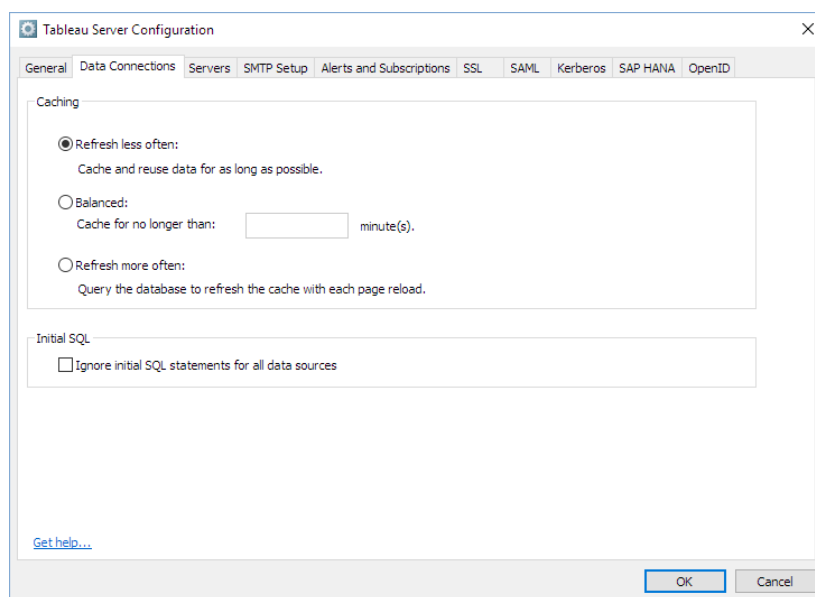
Esta memoria caché tiene dos partes: una pertenece al proceso de VizQL (denominada “caché en proceso”) y la otra se comparte entre los diferentes procesos a través del Servidor de caché (denominada “caché externa”). Si una consulta se envía a una instancia de VizQL que ya ejecutó la misma consulta

anteriormente, la solicitud obtiene la información de la caché en proceso. Tenga en cuenta que esta caché es local para cada proceso de VizQL y se conserva en la memoria.

Además de estas memorias caché en proceso, existe una caché externa compartida, no solo entre las instancias de VizQL, sino también entre TODOS los procesos que accedan a la fuente de datos subyacente (p. ej., procesos en segundo plano, servidores de datos, etc.). Un servicio llamado Servidor de caché administra la memoria caché externa en todo el clúster. Tenga en cuenta que no todas las consultas se escriben en la caché externa. Si una consulta se ejecuta muy rápidamente, es posible que sea más rápido ejecutarla de nuevo que comprobar la memoria caché, por lo que se establece un umbral de tiempo mínimo para las consultas. Además, si el resultado de una consulta es muy extenso, es posible que no sea eficaz escribirla en el Servidor de caché, por lo que también hay un umbral de tamaño máximo.

La caché externa aumenta significativamente la eficacia del almacenamiento en caché en las implementaciones con varias instancias de VizQL. A diferencia de la caché en proceso, que es volátil, la caché externa es duradera (el Servidor de caché almacena los datos en el disco) y se mantiene entre las instancias de los servicios.

Para maximizar la eficacia de las memorias caché, puede ajustar la configuración de la instalación de Tableau Server con el propósito de conservarlas tanto como sea posible:



Si tiene capacidad de servidor disponible, puede aumentar el tamaño de las memorias caché.

El incremento de este valor de configuración no está penalizado, por lo que si cuenta con memoria RAM suficiente, puede aumentar el número de manera significativa para evitar que el contenido se extraiga de la caché. Puede supervisar la eficacia de las memorias caché a través de los [archivos de registro](#) o mediante [TabMon](#).

- **Memoria caché de modelos:** la configuración predeterminada admite 60 modelos en caché por instancia de VizQL. Puede modificar esta configuración a través del comando *tabadmin*. Si tiene memoria RAM disponible, puede incrementar el valor para que coincida con la cantidad de vistas publicadas en el servidor:
  - `tabadmin set vizqlserver.modelcachesize 200`
- **Memoria caché en proceso:** la configuración predeterminada admite 512 MB de resultados de consultas en la memoria caché por instancia de VizQL. Esto parece poco, pero recuerde

que estamos almacenando en caché los resultados de las consultas provenientes de las consultas agregadas. Puede ajustar esta configuración mediante el comando *tabadmin*:

- `tabadmin set vizqlserver.querycachesize 1024`

También puede incrementar la capacidad y el procesamiento de la memoria caché de consultas externas mediante la incorporación de más instancias al Servidor de caché. La práctica recomendada es ejecutar una instancia de Servidor de caché por cada instancia de VizQL Server.

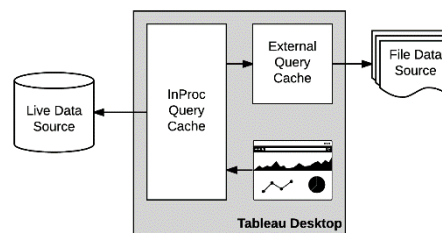
- **Memoria caché externa:** la configuración predeterminada es almacenar 512 MB de resultados de consultas en la memoria caché por cada instancia de Servidor de caché.

Por último, un efecto secundario útil de compartir la caché es que, si sabemos que un libro de trabajo tarda en mostrar la vista inicial, podemos ejecutar una suscripción en el libro de trabajo como “calentamiento” de la memoria caché. Si lo hace temprano por la mañana, antes de que los usuarios interactivos lleguen a la oficina, se asegurará de que el libro de trabajo se ejecute y de que los resultados de las consultas se carguen en la memoria caché. Siempre que los resultados no se eliminen de la memoria caché ni venzan durante el tiempo de intervención, se consultará la caché cuando los usuarios vean el libro de trabajo, lo que hace que la vista inicial sea más rápida.

### Tableau Desktop

En Tableau Desktop, el almacenamiento en caché es mucho más simple que en Tableau Server, porque es una aplicación de un solo usuario y no es necesario administrar varias sesiones.

Tableau Desktop solo cuenta con la memoria caché de consultas:



Al igual que Tableau Server, hay una caché en proceso y una externa. La caché en proceso, volátil y basada en la memoria, se utiliza para las conexiones con todas las fuentes de datos. Por su parte, la caché externa solo se utiliza para las fuentes de datos basadas en archivos (p. ej., extracciones de datos, archivos de Excel y Access, archivos de texto, archivos estadísticos, etc.). Como en Tableau Server, la caché externa es duradera. Esto implica que los resultados almacenados en caché se conservan entre las diferentes sesiones de Tableau Desktop. Por este motivo, si utiliza una fuente de datos basada en archivo, reinicia Tableau Desktop y la vuelve a usar, seguirá beneficiándose de la memoria caché.

En Windows, la caché externa se almacena en `%LOCALAPPDATA%\Tableau\Caching`; en Mac, se almacena en `~/Library/Caches/com.tableau/`. De manera predeterminada, tiene un límite de alrededor de 750 MB en total y se invalidará en el caso de que un usuario fuerce una actualización de la fuente de datos (p. ej., al presionar F5, ⌘R).

Por último, en Tableau Desktop, los datos de la caché de consultas externa se incluyen cuando el archivo se almacena como un libro de trabajo empaquetado. Esto permite a Tableau Desktop y Tableau Reader generar de manera rápida una representación de la vista inicial del libro de trabajo mientras se siguen desempaquetando archivos de fuentes de datos más grandes en segundo plano. Para un usuario final, esto puede mejorar drásticamente la capacidad de respuesta en el momento de abrir el libro de trabajo. Tenga en cuenta que los datos en caché solo se incluyen si los datos representan, como máximo,

el 10 % del tamaño del archivo de la fuente de datos. Además, las consultas que utilizan filtros de fechas relativas no se incluyen.

### Conexiones diferidas

En las versiones anteriores a Tableau 10, cuando un usuario abría un libro de trabajo con múltiples fuentes de datos, inicialmente se conectaba a todas las fuentes de datos sin credenciales (es decir, fuentes de datos que no exigen un nombre de usuario y una contraseña). Esto implicaba una posible pérdida de tiempo en la conexión con fuentes de datos que no se empleaban en la hoja, dashboard o historia que el usuario estuviese viendo.

En Tableau 10, ahora solo se produce la conexión con una fuente de datos cuando un usuario la necesita para ver una hoja, dashboard o historia que haya seleccionado. Este cambio reducirá el tiempo de carga de un libro de trabajo que incluya una vista con pestañas (en Tableau Desktop), por lo que los usuarios pueden comenzar a explorar antes sus datos.

### Uniones

Por lo general, si trabaja con varias tablas en Tableau, el método preferido es definir las uniones en la ventana de conexión de datos. Cuando lo hace, no está definiendo una consulta específica; simplemente está definiendo la forma en la que las tablas se relacionan entre sí. Cuando se arrastran y se sueltan campos en una visualización, Tableau utiliza esta información para generar una consulta específica, necesaria para extraer solo los datos requeridos.

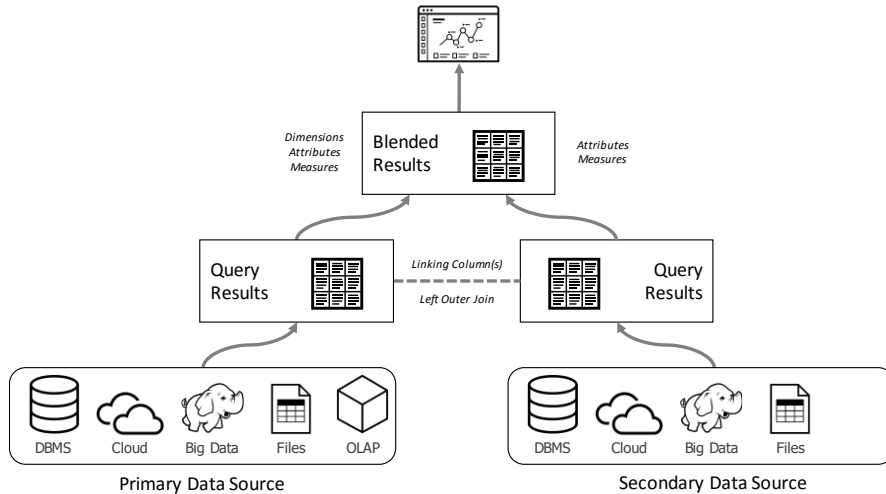
Como regla general para el rendimiento, se debe minimizar la cantidad de tablas unidas e incluir únicamente las necesarias para una hoja de trabajo o visualización específica. Dependiendo de la complejidad de la conexión de datos y de las hojas de trabajo, también puede ser recomendable separar las conexiones de datos para cada hoja de trabajo y crear patrones de unión específicos para dichas hojas.

Las uniones funcionan mejor cuando cuentan con limitaciones apropiadas entre tablas. Las limitaciones proporcionan a Tableau Desktop la libertad de simplificar las consultas para que extraigan datos solo de las tablas necesarias para contestar ciertos subconjuntos de preguntas (leyendas, tarjetas de filtros, etc.).

### Combinaciones

Al decidir entre la unión o la combinación de tablas de datos en Tableau, piense de dónde vienen los datos, la cantidad de conexiones de datos y la cantidad de registros que hay en los datos.

Si opta por la combinación, uno de los factores principales que influyen sobre el rendimiento de esta no es la cantidad de registros de cada fuente de datos sino la cardinalidad de los campos de combinación que vinculan los dos conjuntos de datos. Cuando se usa la combinación, se hacen las consultas en ambas fuentes de datos en el nivel de los campos vinculantes y, después, se fusionan los resultados de ambas consultas en la memoria local de Tableau.



Si hay muchos valores únicos, podría necesitarse una gran cantidad de memoria. Se recomienda utilizar la versión de Tableau Desktop de 64 bits cuando realice combinaciones de datos, a fin de evitar que se agote la memoria. Sin embargo, es posible que incluso en esas circunstancias las combinaciones de este tipo tarden más en calcularse.

La práctica recomendada para la combinación es evitar combinar dimensiones con grandes cantidades de valores únicos (p. ej., ID de pedido, ID de cliente, fecha/hora exacta, etc.).

### Grupos y alias primarios

Si tiene la necesidad de hacer combinaciones entre dos fuentes de datos porque una contiene los registros de “hechos” y la otra contiene atributos dimensionales, podría mejorarse el rendimiento creando un grupo o alias primario.

Funcionan mediante la creación de grupos o alias en la fuente de datos principal, para que reflejen los atributos enumerados en la fuente de datos secundaria. Los grupos primarios se utilizan para relaciones de 1 a muchos, mientras que los alias primarios se usan para relaciones 1 a 1. A continuación, puede utilizar el objeto pertinente de la fuente de datos principal, lo que elimina la necesidad de realizar la combinación durante la visualización.

Para los siguientes ejemplos, tenga en cuenta estas tres tablas:

ID	Value
A	89
B	94
C	74
D	88
E	58
F	89
G	95

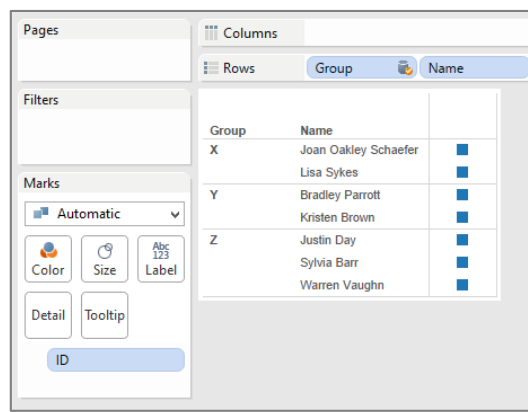
ID	Name
A	Lisa Sykes
B	Joan Oakley Schaefer
C	Bradley Parrott
D	Kristen Brown
E	Sylvia Barr
F	Warren Vaughn
G	Justin Day

ID	Group
A	X
B	X
C	Y
D	Y
E	Z
F	Z
G	Z

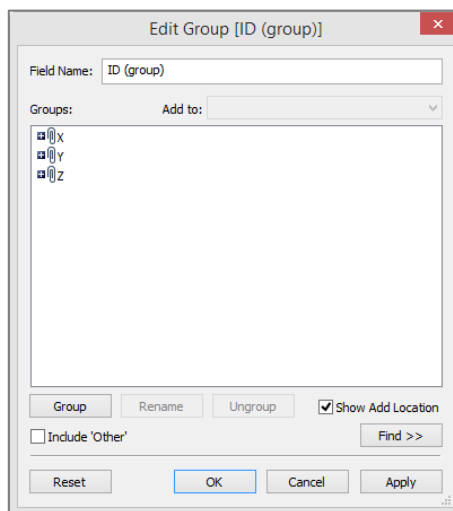
Los grupos primarios son útiles cuando la fuente de datos secundaria contiene un atributo en el que cada valor se asigna a varios de los miembros de dimensiones de la fuente de datos principal. Imagine que lo que nos interesa generar con los datos anteriores es esto:

Group	Name
X	Lisa Sykes
	Joan Oakley Schaefer
Y	Bradley Parrott
	Kristen Brown
Z	Sylvia Barr
	Warren Vaughn
	Justin Day

Podríamos crearlo mediante la combinación, pero, como ya mencionamos, esto ralentizaría mucho el funcionamiento si hubiera una gran cantidad de ID:

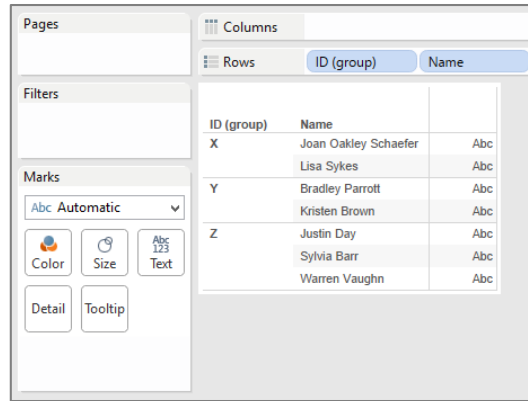


Si hace clic con el botón secundario en el campo Grupo y selecciona “Crear grupo primario”, Tableau creará un objeto de grupo en la fuente de datos principal que asigna el campo de vinculación (en este caso, ID) a la dimensión seleccionada de la fuente de datos secundaria (en este caso, Grupo).



Ahora podemos crear esta tabla sin la necesidad de una combinación:

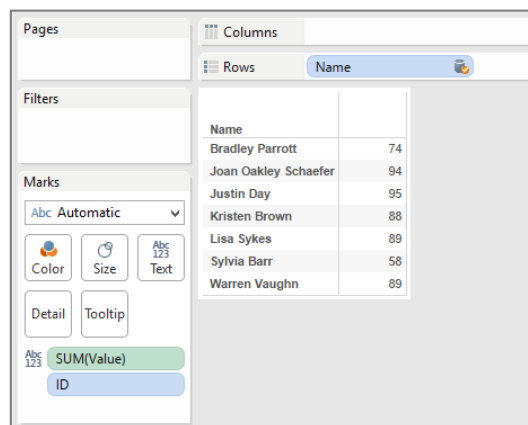




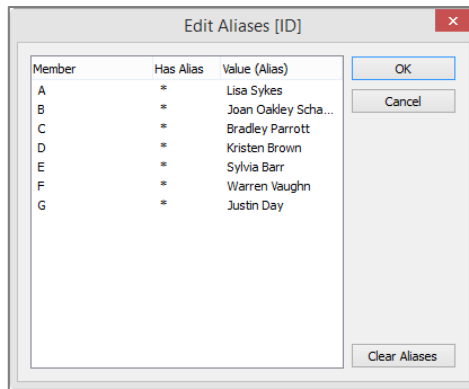
Los alias primarios son útiles cuando la fuente de datos secundaria contiene un atributo en el que cada valor se asigna a uno de los miembros de dimensiones de la fuente de datos principal. Imagine que lo que nos interesa generar con los datos anteriores es esto:

Name	Value
Lisa Sykes	89
Joan Oakley Schaefer	94
Bradley Parrott	74
Kristen Brown	88
Sylvia Barr	58
Warren Vaughn	89
Justin Day	95

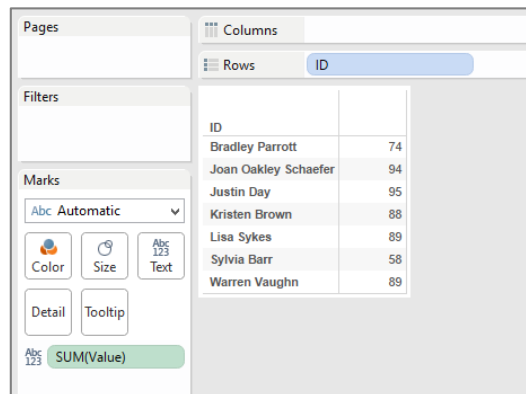
Podríamos crearlo mediante la combinación entre ambas fuentes de datos, pero, como ya mencionamos, esto ralentizaría el funcionamiento si hubiera una gran cantidad de ID:



Si hacemos clic con el botón secundario en el campo Nombre y seleccionamos “Editar alias primario”, podemos hacer una asignación única del campo Nombre al campo ID como valores de alias:



Entonces, podemos crear la visualización deseada sin ninguna combinación, lo que será más rápido:



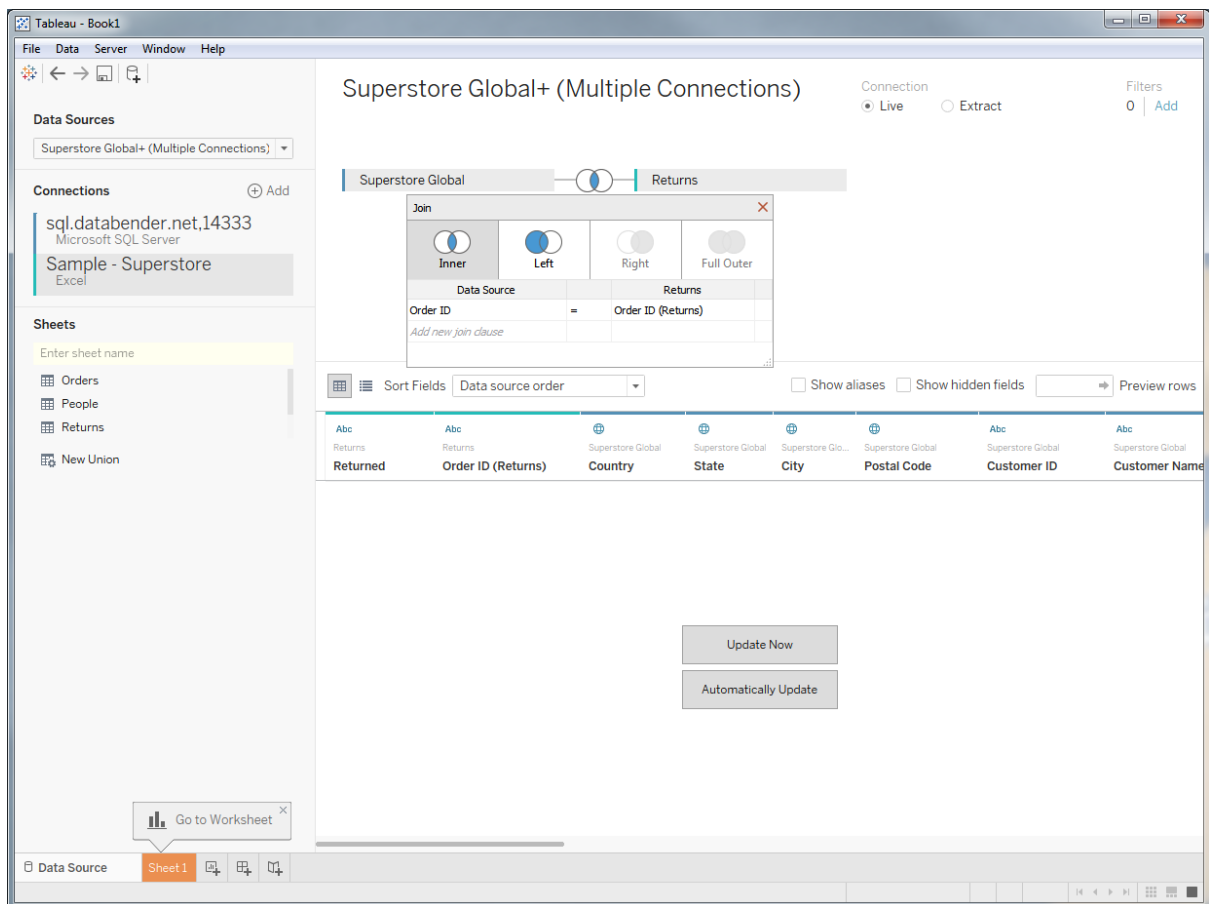
Tenga en cuenta que tanto los grupos como los alias primarios no son dinámicos y deberán actualizarse si los datos cambian. Por lo tanto, no son una solución excelente para los datos que se actualizan con frecuencia; pero si necesita una asignación rápida, pueden llegar a evitar la necesidad de combinaciones costosas.

Puede encontrar más información en este artículo de la Base de conocimientos:

[http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#multipleconnections\\_create\\_primary\\_group.html](http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#multipleconnections_create_primary_group.html)

## Integración de datos

La integración de datos es una funcionalidad nueva de Tableau 10 que permite que las fuente de datos los combinen a partir de múltiples conexiones de datos potencialmente heterogéneas.



Una diferencia clave entre la integración de datos y la combinación de datos es que la primera es una unión en el nivel de la fila, mientras que la segunda se realiza en un conjunto de resultados agregados de cada fuente de datos. Esto implica que la integración de datos es sensible al tamaño de las fuentes de datos subyacentes. Hay 4 aspectos que debe tener en cuenta:

- La demora es mayor cuantos más datos se deban transferir. Los datos se extraen en el nivel de la fila de cada conexión de datos, por lo que el volumen de datos es un factor clave.
- La demora es mayor cuanto mayor sea la distancia de transferencia de los datos. El rendimiento se verá afectado si ejecuta una unión a una fuente de datos que tiene una conexión de latencia alta.
- La demora es mayor cuanto más lenta sea la tasa de transferencia de datos. El rendimiento se verá afectado si ejecuta una unión a una fuente de datos que tiene una conexión con un ancho de banda limitado.
- La demora es mayor cuantas más concordancias deba haber. De manera similar al punto uno, el rendimiento se ve afectado por la cantidad de registros que se deban unir.

En Tableau 10, la integración de datos solo se puede utilizar con fuentes de datos extraídos; es decir, los resultados de cada conexión de datos se extraen para luego ser almacenados en un archivo TDE. A partir de allí, esta fuente de datos extraídos se puede publicar en Tableau Server para que otros la puedan usar.

En Tableau 10, no es posible integrar datos en fuentes de datos con conexiones en tiempo real; tampoco es posible integrar datos de fuentes publicadas. Son capacidades que estamos intentando incorporar a futuras versiones.

## SQL personalizado

En ocasiones, quienes usan Tableau por primera vez intentan emplear técnicas anticuadas en sus libros de trabajo, como crear fuentes de datos con declaraciones SQL escritas a mano. Muchas veces, esto es contraproducente, puesto que Tableau puede generar consultas con mucha mayor eficacia simplemente si definimos las relaciones de unión entre las tablas y dejamos que el motor de consultas escriba el código SQL específico para la vista que se está creando. Sin embargo, hay ocasiones en las que la especificación de uniones en la ventana de conexión de datos no ofrece toda la flexibilidad que necesita para definir las relaciones de sus datos.

La creación de una conexión de datos que emplee declaraciones SQL escritas a mano puede ser muy poderosa. Pero, como advierte un famoso superhéroe: “Un gran poder conlleva una gran responsabilidad”. En algunos casos, el código SQL personalizado puede efectivamente tener un impacto negativo en el rendimiento. Esto se debe a que, a diferencia de la definición de uniones, el SQL personalizado nunca se deconstruye y siempre se ejecuta atómicamente. Esto implica que no se produce una selección de uniones y puede conducir a una situación en la que se solicite a la base de datos que procese una consulta completa por una sola columna, como en el ejemplo siguiente:

```
SELECT SUM([TableauSQL].[Sales])
FROM (
    SELECT [OrdersFact].[Order ID] AS [Order ID],
           [OrdersFact].[Date ID] AS [Date ID],
           [OrdersFact].[Customer ID] AS [Customer ID],
           [OrdersFact].[Place ID] AS [Place ID],
           [OrdersFact].[Product ID] AS [Product ID],
           [OrdersFact].[Delivery ID] AS [Delivery ID],
           [OrdersFact].[Discount] AS [Discount],
           [OrdersFact].[Cost] AS [Cost],
           [OrdersFact].[Sales] AS [Sales],
           [OrdersFact].[Qty] AS [Qty],
           [OrdersFact].[Profit] AS [Profit]
    FROM [dbo].[OrdersFact] [OrdersFact]
) [TableauSQL]
HAVING (COUNT_BIG(1) > 0)
```

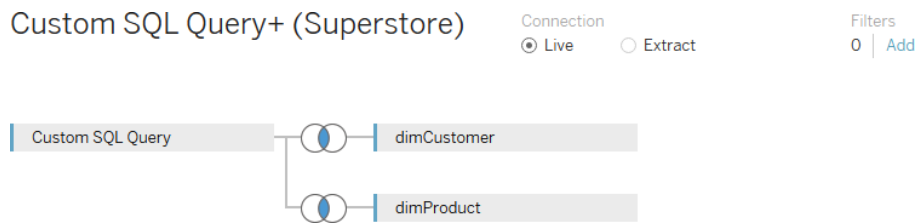
Es importante asegurarse de que la declaración de SQL personalizado no contenga ninguna cláusula innecesaria. Por ejemplo, si la consulta contiene cláusulas GROUP BY u ORDER BY, por lo general producirán sobrecarga, puesto que Tableau creará sus propias cláusulas basándose en la estructura de la visualización. Elimínelas de su consulta de ser posible.

Una buena recomendación es usar SQL personalizado en combinación con extracciones de datos. Así, la consulta atómica se ejecuta una sola vez (para cargar los datos en la extracción de datos), y todo el análisis siguiente con Tableau se hace mediante consultas dinámicas optimizadas a la extracción de datos. Por supuesto, todas las reglas tienen excepciones y esta también la tiene. Cuando utilice SQL personalizado, cree una extracción de datos A MENOS QUE el código SQL personalizado contenga parámetros.

En algunas circunstancias, el uso de parámetros en una declaración de SQL personalizado puede mejorar el rendimiento de las conexiones en tiempo real, ya que la consulta de base puede ser más dinámica (p. ej., las cláusulas de filtro que utilizan parámetros se evaluarán adecuadamente). Además, los parámetros pueden usarse para introducir valores de delimitación de rendimiento, como TOP o SAMPLE, a fin de restringir la cantidad de datos que devuelve la base de datos. No obstante, si utiliza una extracción de datos, se purgará y regenerará cada vez que modifique el parámetro y esto puede ralentizar el proceso. Además, tenga en cuenta que los parámetros solo pueden usarse

para introducir valores literales y no pueden usarse para cambiar de forma dinámica las cláusulas SELECT ni las FROM.

Por último, también es posible unir tablas a SQL personalizado:



Esto le permite escribir un código SQL personalizado más específico, que haga referencia a un subconjunto del esquema total. Luego, las uniones entre esto y las otras tablas se pueden seleccionar (potencialmente) como uniones de tablas comunes, lo que creará consultas más eficaces.

### Alternativas a SQL personalizado

En lugar de utilizar directamente SQL personalizado en Tableau, a veces es preferible trasladar la consulta a la fuente de datos subyacente. En muchos casos, esto puede mejorar el rendimiento, puesto que permite que la fuente de datos analice más eficazmente la consulta. O bien, podría dar como resultado una única ejecución de una consulta compleja. Además se adecua a las prácticas recomendadas de administración, puesto que permite compartir una sola definición entre múltiples libros de trabajo y fuentes de datos.

Hay varias maneras de llevar esto a cabo:

#### Vistas

Casi todos los sistemas de administración de bases de datos (DBMS) admiten el concepto de vistas: una tabla virtual que representa los resultados de una consulta a una base de datos. Para algunos sistemas de bases de datos, la simple extracción de una consulta desde una declaración de SQL personalizado y la creación de una instancia en la base de datos como una vista mejora el rendimiento considerablemente. Esto se debe a que el optimizador de consultas es capaz de generar un plan de ejecución mejor cuando la consulta está incluida en una declaración SELECT que la contenga. Para obtener más detalles, consulte el siguiente debate de la comunidad:

<http://tabsoft.co/1su5YMe>

Definir la lógica de una consulta personalizada en una vista, en vez de hacerlo en el libro de trabajo de Tableau, también permite que pueda reutilizarse en múltiples libros de trabajo y fuentes de datos. Además, muchos DBMS admiten el concepto de vistas materializadas o instantáneas, en las que los resultados de la consulta de la vista se calculan previamente y se almacenan en caché, lo que permite respuestas mucho más veloces en el momento de realizar la consulta. Pueden ser similares a las tablas de resumen (ver más adelante), pero reciben mantenimiento del DBMS de forma automática.

#### Procedimientos almacenados

Los procedimientos almacenados son parecidos a las vistas. Sin embargo, pueden incluir lógica mucho más compleja y pueden realizar varios pasos de la preparación de datos. También se pueden parametrizar, lo que les permite devolver un conjunto dirigido de datos a partir de las entradas del usuario.

Los procedimientos almacenados son compatibles con Sybase ASE, SQL Server y Teradata. Para evitar la ejecución repetida de un procedimiento almacenado para una única visualización, Tableau lo ejecuta y almacena el conjunto de resultados en una tabla temporal en la base de datos. Luego, se ejecutan

las consultas reales en la tabla temporal. La ejecución del procedimiento almacenado y el relleno de la tabla temporal se realizan durante la apertura inicial del libro de trabajo y siempre que se modifiquen los parámetros del proceso almacenado. Es posible que este proceso lleve un tiempo y que las interacciones sean lentas.

Si extrae resultados de un procedimiento almacenado parametrizado y los guarda en una extracción de datos, la extracción se purgará y actualizará cada vez que modifique los valores de los parámetros.

Puede encontrar más información sobre cómo utilizar los procedimientos almacenados en la Ayuda en línea de Tableau:

[http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#connect\\_basic\\_stored\\_procedures.html](http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#connect_basic_stored_procedures.html)

### Tablas de resumen

Si tiene un conjunto de datos muy grande y detallado que suele resumir cuando hace consultas (p. ej., guarda transacciones individuales, pero generalmente usa los datos resumidos por día, región, cliente, producto, etc.), considere crear una tabla de resumen y usar Tableau con ella para reducir los tiempos de consulta.

Nota: puede usar extracciones de datos de Tableau para lograr un resultado similar creando una extracción de datos agregada. Consulte la sección sobre extracciones para obtener más detalles.

### SQL inicial

Otra alternativa al uso de SQL personalizado (si su fuente de datos la admite) consiste en usar la declaración de SQL personalizado en un bloque SQL inicial. Esto le permite crear una tabla temporal, que será la tabla seleccionada para su consulta. Debido a que el código SQL inicial se ejecuta una sola vez cuando se abre el libro de trabajo (en lugar de ejecutarse cada vez que se cambia la visualización del código SQL personalizado), esto podría mejorar considerablemente el rendimiento en algunos casos.

Tenga en cuenta que, en Tableau Server, un administrador puede configurar una restricción en SQL inicial para que no se ejecute. Es posible que deba comprobar si esto es adecuado para su entorno en el caso de que esté planeando publicar su libro de trabajo para compartirlo con otros.

Puede encontrar más información sobre SQL inicial en la documentación en línea:

[http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#connect\\_basic\\_initialsql.html](http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#connect_basic_initialsql.html)

## ¿El problema son mis datos?

Una de las valiosas características de Tableau es su capacidad de conectarse a datos de una gran variedad de plataformas. En términos generales, estas plataformas pueden ser de los siguientes tipos:

- Fuentes de datos basadas en archivos, como archivos de Excel y CSV.
- Fuentes de bases de datos relacionales, como Oracle, Teradata y SQL Server, además de las aplicaciones analíticas especializadas, como HP Vertica, IBM Netezza y otras.
- Fuentes de datos OLAP, como Microsoft Analysis Services y Oracle Essbase.
- Fuentes de datos “big data”, como Hadoop.
- Fuentes de datos basadas en la nube, como Salesforce, Google y otras.

Todos los tipos de fuentes de datos tienen sus propias ventajas y desventajas, y se tratan de manera independiente.

Tenga en cuenta que Tableau Desktop es compatible tanto con el entorno Windows como con el de Mac OS X, y que el conjunto de fuentes de datos compatibles en Mac no es el mismo que en Windows. Tableau tiene el propósito de minimizar las diferencias entre las plataformas. No obstante, actualmente solo algunas fuentes de datos son compatibles con una u otra.

### Consejos generales

#### Utilice controladores nativos

Tableau 10 admite conexiones nativas con más de 40 fuentes de datos diferentes. Esto implica que Tableau implementó técnicas, capacidades y optimizaciones específicas para estas fuentes de datos. El diseño de ingeniería y las pruebas realizadas en estas conexiones aseguran la mayor eficacia que Tableau puede ofrecer.

Tableau también admite ODBC genéricos para acceder a fuentes de datos que no están en la lista de conectores nativos. Como un estándar definido públicamente, muchos proveedores de bases de datos proporcionan controladores ODBC para sus bases de datos. Tableau también puede utilizar estos controladores para conectarse a datos. Puede haber diferencias en la forma en la que el proveedor de bases de datos interpreta o implementa las capacidades del ODBC estándar. En algunos casos, Tableau le recomendará o le exigirá que cree una extracción de datos antes de continuar trabajando con un controlador concreto. También hay otros controladores ODBC y bases de datos a los que Tableau no puede conectarse.

Si hay un controlador nativo para la fuente de datos que está consultando, debe usarlo en lugar de la conexión ODBC, puesto que, por lo general, proporcionan un mejor rendimiento. Además, tenga en cuenta que las conexiones ODBC están solo disponibles en Windows.

#### Realice pruebas tan cerca del entorno donde estén hospedados los datos como sea posible

Como se señaló antes, un principio general es que, si las consultas se ejecutan lentamente en la fuente de datos, la experiencia en Tableau será también lenta. Una buena manera de comprobar el nivel de rendimiento de la fuente de datos de manera directa es, de ser posible, instalar Tableau Desktop en el equipo donde está almacenada la fuente de datos y ejecutar algunas consultas. Esto eliminará algunos factores que afectan el rendimiento, como el ancho de banda de la red o su latencia, y le permitirá comprender con mayor detalle el nivel de rendimiento de la consulta en la fuente de datos de manera directa. Además, al utilizar el nombre *localhost* para la fuente de datos en lugar del nombre de DNS, podrá determinar con mayor facilidad si los factores del entorno, como la lentitud de resolución de nombres o de los servidores proxy, son también causa de un rendimiento deficiente.

## Fuentes de datos

### Archivos

En esta categoría se incluyen todas las fuentes de datos basadas en archivos: archivos de texto (como CSV), hojas de cálculo de Excel y archivos de MS Access. Estos son los ejemplos más comunes, aunque esta categoría también incluye archivos de datos provenientes de las plataformas estadísticas SPSS, SAS y R. Frecuentemente, los usuarios corporativos utilizan datos en este formato, porque es un método común para obtenerlos de conjuntos de datos “administrados” (tanto al crear informes como al generar una consulta para la extracción).

En general, la práctica recomendada es importar la fuente de datos basada en archivos al motor de datos de Tableau. Esto hará que las consultas se ejecuten mucho más rápido y que los valores de datos se almacenen en archivos mucho más pequeños. Sin embargo, si el archivo es pequeño o se necesita una conexión en tiempo real con el archivo para reflejar datos cambiantes, puede conectarse en vivo.

### *Extracciones de respaldo*

Cuando se conecta con archivos de Excel o de texto no heredados, o archivos estadísticos, Tableau crea un archivo de extracción de manera transparente como parte del proceso de conexión. Se lo denomina extracción de respaldo y agiliza considerablemente el trabajo con los datos, a diferencia de si consultara directamente el archivo.

Es posible que, cuando use un archivo grande por primera vez, tarde algunos segundos en cargar los datos al panel de vista previa. Esto se debe a que Tableau está extrayendo los datos del archivo y almacenándolos en un archivo de extracción. De manera predeterminada, estos archivos se almacenan en `C:\Users\\AppData\Local\Tableau\Caching\TemporaryExtracts` con un nombre hash basado en el nombre de la ruta y la última fecha de modificación del archivo de datos. Tableau conserva en este directorio las extracciones de respaldo de las últimas cinco fuentes de datos utilizadas y elimina la menos reciente cuando se crea una nueva. Si, posteriormente, vuelve a utilizar un archivo que tiene una extracción de respaldo, Tableau simplemente abrirá el archivo de extracción y la vista previa de datos aparecerá casi al instante.

A pesar de que las extracciones de respaldo contienen datos subyacentes y otra información similar a la de una extracción estándar de Tableau, las extracciones de respaldo se almacenan en un formato diferente (con una extensión .ttde), lo que implica que no se pueden utilizar de la misma manera que las extracciones de Tableau.

### *Conectores heredados para archivos de Excel y de texto*

En las versiones anteriores de Tableau, las conexiones con archivos de Excel y de texto empleaban el controlador del motor de datos JET de Microsoft. Sin embargo, a partir de Tableau 8.2, comenzamos a utilizar de manera predeterminada un controlador nativo que proporciona un rendimiento mejorado y funciona con archivos más grandes y complejos. No obstante, hay algunas situaciones en las que es preferible utilizar los controladores heredados (p. ej., si desea utilizar SQL personalizado). En estos casos, los usuarios tienen la opción de utilizar el controlador JET heredado. Tenga en cuenta que los archivos de MS Access todavía requieren el uso del controlador JET.

Puede encontrar una lista detallada de las diferencias entre los dos controladores aquí:

[http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#upgrading\\_connection.html](http://onlinehelp.tableau.com/current/pro/desktop/es-es/help.htm#upgrading_connection.html)



Tenga en cuenta que los controladores JET no están disponibles para Mac OS y, por lo tanto, Tableau Desktop para Mac no admite archivos MS Access ni proporciona una opción de conexión heredada para archivos de Excel ni de texto.

### Bases de datos relacionales

Las fuentes de datos relacionales son las fuentes de datos más comunes para los usuarios de Tableau, y Tableau proporciona controladores nativos para una amplia gama de plataformas. Estos pueden basarse en filas o columnas; pueden ser personales o empresariales, y puede accederse a ellos mediante controladores nativos u ODBC genérico. Técnicamente, esta categoría también incluye las fuentes de datos de MapReduce, ya que se accede a ellas a través de niveles de acceso SQL, como Hive o Impala. Analizaremos estas fuentes en detalle en la sección de “big data” más adelante.

Existen muchos factores internos que afectan la velocidad de las consultas en los sistemas de administración de bases de datos relacionales (RDBMS). Para cambiarlos o modificarlos, probablemente necesite ayuda de su administrador de bases de datos, pero puede obtener mejoras de rendimiento significativas.

### Sistemas basados en filas y basados en columnas

Los sistemas de administración de bases de datos relacionales (RDBMS) tienen dos configuraciones: los basados en filas y los basados en columnas. Las disposiciones de almacenamiento basadas en filas son adecuadas para cargas de trabajo de tipo OLTP, que tienen cargas más intensas con las transacciones interactivas. Las disposiciones de almacenamiento basadas en columnas son adecuadas para las cargas de trabajo analíticas (p. ej., almacenes de datos), lo que, por lo general, implica realizar consultas complejas en grandes conjuntos de datos.

Actualmente, muchas de las soluciones de análisis con alto rendimiento funcionan con RDBMS basados en columnas. Verá que sus consultas se ejecutarán con mayor velocidad cuando utiliza dichas soluciones. Algunos ejemplos de bases de datos basadas en columnas compatibles con Tableau son Actian Vector, Amazon Redshift, HP Vertica, IBM Netezza, MonetDB, Privotal Greenplum, SAP HANA y SAP Sybase IQ.

### SQL como interfaz

Existen muchos sistemas que no están basados en las tecnologías tradicionales de RDBMS, pero aún aparecen como fuentes relacionales, puesto que proporcionan una interfaz basada en SQL. Para Tableau, esto incluye varias plataformas NoSQL (p. ej., MarkLogic, DataStax, MongoDB, etc.), además de las plataformas de aceleración de consultas, como Kognitio, AtScale y JethroData.

### Índices

La indexación correcta de la base de datos es fundamental para que las consultas sean eficaces:

- Asegúrese de que las columnas que formen parte de las uniones de la tabla estén indexadas.
- Asegúrese también de que las columnas utilizadas en los filtros estén indexadas.
- Tenga en cuenta que el uso de filtros de fecha discretos en algunas bases de datos puede provocar que las consultas no usen los índices de las columnas de fecha y fecha/hora. Esto se analiza en profundidad en la sección de filtros, pero sepa que usar un filtro por intervalo de fechas garantiza que se use el índice de las fechas. Por ejemplo, en vez de utilizar `YEAR([DateDim])=2010`, exprese el filtro de la siguiente manera: `[DateDim] >= #2010-01-01# and [DateDim] <= #2010-12-31#`.
- Asegúrese de que las estadísticas de sus datos estén habilitadas para permitir que el optimizador de consultas cree planes de consulta de alta calidad.

Muchos entornos de sistemas de administración de bases de datos (DBMS) tienen herramientas de administración que recomiendan índices en función de las consultas.

#### *Valores nulos*

Los valores nulos en las columnas de dimensiones pueden reducir la eficacia de las consultas. Imagine una visualización en la que desee mostrar las ventas totales por país para los 10 productos principales. En primer lugar, Tableau generará una subconsulta para encontrar los 10 productos principales y, luego, unirá estos resultados con la consulta por país para generar los resultados finales.

Si la columna de productos está configurada para ADMITIR valores nulos, debemos ejecutar una consulta para verificar si hay algún valor nulo entre los 10 productos principales que devolvió la subconsulta. De ser así, debemos realizar una unión que preserve los valores nulos, lo que es más costoso de calcular que una unión habitual. Si la columna de productos está configurada para NO ADMITIR valores nulos, sabemos que el resultado de la consulta no puede contener valores nulos. Por ese motivo, podemos realizar una unión habitual y evitar la consulta para “verificar la presencia de valores nulos”.

Por esta razón, cuando sea posible, debe configurar las columnas de dimensiones de manera que NO ADMITAN valores nulos.

#### *Integridad referencial*

Cuando se unen varias tablas en una fuente de datos, Tableau tiene una función estupenda (y, generalmente, invisible para el usuario) llamada “selección de uniones”. Dado que se debe destinar tiempo y dinero a procesar las uniones en el servidor de la base de datos, no se recomienda enumerar cada unión que se declaró en la fuente de datos todo el tiempo. La selección de uniones nos permite hacer consultas solo en las tablas relevantes, en lugar de hacerlas en todas las tablas definidas en una unión.

Imagine la siguiente situación en la que se unieron varias tablas en un esquema de estrella pequeño:

#	Abc	#	Abc	#	#	Abc
CustomerDim	CustomerDim	DeliveryDim	DeliveryDim	DeliveryDim	LocationDim	LocationDim
Customer ID (Cust...	Name	Delivery ID (Delive...	Shipping Mode	Days	Place ID (Location...	Region
75.00	Altd	8900.00	First Class	3.00000	1051.00	Europe
1195.00	Systed Systems, Inc	8300.00	First Class	2.00000	661.00	Asia
1004.00	Seas	2400.00	Standard Class	5.00000	885.00	Asia
1439.00	Vellutems Corporation	0.00	Standard Class	4.00000	595.00	Asia
883.00	Mics Consulting	0.00	Standard Class	4.00000	786.00	Asia
984.00	Quan	2400.00	Standard Class	5.00000	173.00	Africa
1029.00	Sems Group	4200.00	Standard Class	6.00000	31.00	Africa
953.00	Nettpaq USA	2400.00	Standard Class	5.00000	130.00	Africa
683.00	Inetems Research	6000.00	Second Class	2.00000	282.00	America
1302.00	TER	5400.00	Standard Class	7.00000	541.00	Asia
1054.00	Sillian Corporation	0.00	Standard Class	4.00000	151.00	Africa

Con la selección de uniones, al hacer doble clic en la medida Sales, se genera la consulta siguiente:

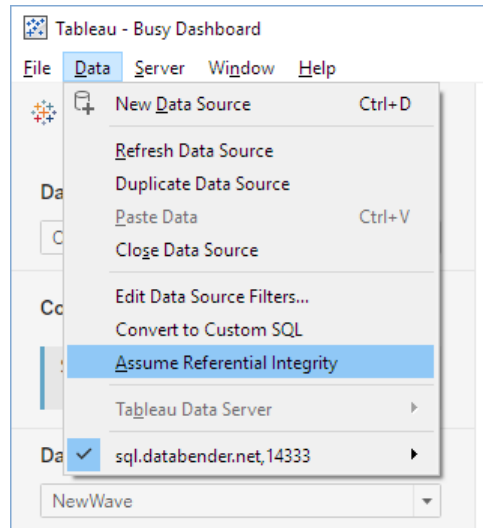
```
SELECT SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY ()
```

Sin ella, se genera una consulta mucho menos eficaz:

```
SELECT SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
INNER JOIN [dbo].[CustomerDim] [CustomerDim]
ON ([OrdersFact].[Customer ID] = [CustomerDim].[Customer ID])
INNER JOIN [dbo].[DeliveryDim] [DeliveryDim]
ON ([OrdersFact].[Delivery ID] = [DeliveryDim].[Delivery ID])
INNER JOIN [dbo].[LocationDim] [LocationDim]
ON ([OrdersFact].[Place ID] = [LocationDim].[Place ID])
INNER JOIN [dbo].[ProductDim] [ProductDim]
ON ([OrdersFact].[Product ID] = [ProductDim].[Product ID])
INNER JOIN [dbo].[TimeDim] [TimeDim]
ON ([OrdersFact].[Date ID] = [TimeDim].[Date ID])
GROUP BY ()
```

Todas las tablas de dimensiones deben unirse para garantizar que se calculen las sumas de medidas correctas desde el principio. Por ejemplo, si nuestra tabla de hechos contiene datos del período 2008-2012, pero la tabla de dimensiones de tiempo solo tiene valores del intervalo 2010-2012, el resultado de `SSUM([Sales])` posiblemente cambie dependiendo de si se incluye la tabla de tiempo.

Anteriormente, se requería una integridad referencial “dura” cuando la regla era impuesta por el DBMS. Sin embargo, muchos clientes tienen fuentes de datos en las que la integridad referencial se cumple en la capa de la aplicación o mediante un proceso de ETL. Esto se conoce como integridad referencial “suave”. Los usuarios pueden indicar a Tableau que se configuró una integridad referencial suave y que la selección de uniones se puede utilizar de manera segura activando la opción “Asumir la integridad referencial”:



Tenga en cuenta que, a pesar de que Tableau puede utilizar tanto la integridad referencial dura como la suave, en la mayoría de las situaciones es mucho mejor emplear la integridad referencial dura, porque la base de datos también puede seleccionar uniones. Para obtener más información, consulte la siguiente serie de artículos de Russell Christopher en su blog Tableau Love:

<http://bit.ly/1HACmPV>

<http://bit.ly/1HACqPn>

### *Particionamiento*

El particionamiento de bases de datos mejora el rendimiento mediante la división de tablas grandes en tablas individuales más pequeñas (llamadas particiones o divisiones). Esto permite que las consultas se ejecuten con más rapidez, ya que la cantidad de datos que se debe examinar es menor o hay más controladores que dan servicio en la entrada y la salida. El particionamiento es una estrategia recomendada para grandes volúmenes de datos y es transparente para Tableau.

El particionamiento funciona bien en Tableau si se hace en una dimensión (por ejemplo, tiempo, región, categoría, etc.) que normalmente se filtre, de modo que las consultas individuales solo tengan que leer registros de una sola partición.

Tenga en cuenta que para algunas bases de datos se necesitan filtros por intervalo de fechas (no filtros discretos) para garantizar que los índices de partición se usen correctamente; de lo contrario, la exploración de una tabla completa puede tener un rendimiento muy bajo.

### *Tablas temporales*

Muchas operaciones en Tableau pueden implicar el uso de tablas temporales; por ejemplo, la creación de grupos y conjuntos ad-hoc, y la combinación de datos. Se recomienda asignar a los usuarios el permiso para crear y soltar tablas temporales, y asegurarse de que el entorno tenga el espacio en cola suficiente para las consultas que se están ejecutando.

## OLAP

Tableau admite varias fuentes de datos OLAP:

- Microsoft Analysis Services
- Microsoft PowerPivot (PowerPivot para Excel y PowerPivot para SharePoint)
- Oracle Essbase
- SAP Business Warehouse
- Teradata OLAP

Hay diversas diferencias funcionales cuando se conecta a fuentes OLAP en contraste con las relacionales, debido a las diferencias en el lenguaje subyacente entre MDX/DAX y SQL. Los puntos clave que debe tener en cuenta son que ambos tipos de fuentes tienen la misma interfaz de usuario en Tableau, las mismas visualizaciones y el mismo lenguaje de expresiones para las medidas calculadas. Las diferencias tienen que ver, sobre todo, con los metadatos (cómo y dónde se definen), los filtros, el funcionamiento de los totales y las agregaciones, y la forma en que se puede usar la fuente de datos para la combinación de datos.

Puede encontrar más información acerca de las diferencias entre el uso de fuentes de datos relacionales y el uso de fuentes de datos OLAP en Tableau en este artículo de la Base de conocimientos:

<http://kb.tableau.com/articles/knowledgebase/functional-differences-olap-relational?lang=es-es>

### *Extracciones de datos de SAP BW*

SAP BW es única entre nuestras fuentes de datos OLAP, porque puede extraer datos de los cubos SAP BW e incorporarlos en el motor de datos de Tableau. (Nota: esto requiere un código de clave especial que puede solicitar a Tableau). Tableau recupera los nodos de nivel de hoja (no los datos detallados) y los convierte en una fuente de datos relacional. Como la transformación de multidimensional a relacional no conserva todas las estructuras de cubo, las extracciones con cubos no admiten la posibilidad de cambiar libremente una y otra vez entre una extracción y una conexión en tiempo real sin que el estado de su visualización se vea afectado. Deberá elegir antes de comenzar a diseñar su visualización. Sin embargo, no debe decidir todo desde un principio. Puede alternar entre opciones de alias (clave, nombre largo, etc.) después de la extracción.

Puede encontrar información detallada sobre cómo crear extracciones de SAP BW aquí:

<http://tabsoft.co/1SuYf9d>

## Big data

Big data es un término con muchas acepciones en el mundo del análisis de datos. Sin embargo, en este documento, lo usamos particularmente para referirnos a las plataformas basadas en Hadoop. En Tableau 10, hay cuatro distribuciones compatibles de Hadoop que admiten conexiones Hive o Impala:

- Amazon EMR
  - HiveServer
  - HiveServer2
  - Impala
- Cloudera Hadoop
  - HiveServer
  - HiveServer2
  - Impala

- Hortonworks Hadoop Hive
  - HiveServer
  - HiveServer2
  - Hortonworks Hadoop Hive
- MapR Hadoop Hive
  - HiveServer
  - HiveServer2
- Spark SQL
  - SharkServer \*
  - SharkServer2 \*
  - SparkThriftServer

*\* Tenga en cuenta que se proporcionan las conexiones SharkServer y SharkServer2 para su uso, pero no son admitidas por Tableau.*

Hive actúa como capa de conversión de SQL-Hadoop: convierte la consulta a MapReduce, la cual después se ejecuta con los datos de HDFS. Impala ejecuta la declaración de SQL directamente con los datos de HDFS (no necesita MapReduce). Tableau también es compatible con Spark SQL, un motor de procesamiento de código abierto para big data que puede tener un rendimiento 100 veces más rápido que MapReduce cuando se ejecuta en la memoria en vez de en el disco.

Impala, por lo general, es mucho más rápido que Hive, y Spark ha demostrado ser incluso más rápido.

Incluso con estos componentes adicionales, por lo general, Hadoop no responde adecuadamente a las consultas analíticas como las que crea Tableau. Las extracciones de datos de Tableau suelen usarse para mejorar los tiempos de respuesta de las consultas (más adelante encontrará información sobre las extracciones y sobre su uso con “big data”).

Puede encontrar más detalles para mejorar el rendimiento con fuentes de datos de Hadoop en:

<http://kb.tableau.com/articles/knowledgebase/hadoop-hive-performance?lang=es-es>

## Nube

Actualmente, Tableau admite las siguientes fuentes de datos basadas en la nube:

- Salesforce.com
- Google Analytics
- oData (incluye Windows Azure Marketplace DataMarket)

Las fuentes de este primer grupo leen un conjunto de registros de datos de un servicio web y los cargan en un archivo de extracción de datos de Tableau. La “conexión en tiempo real” no es posible para estas fuentes de datos, pero el archivo extraído puede actualizarse para que también se actualicen los datos que contiene. Con Tableau Server, este proceso de actualización puede automatizarse y programarse.

Tableau también admite las siguientes plataformas de datos basadas en la nube:

- Amazon Aurora
- Amazon Redshift
- Amazon RDS
- Google BigQuery
- Google Cloud SQL

- Hojas de cálculo de Google
- Almacén de datos SQL de Microsoft Azure
- Snowflake

A diferencia del grupo anterior de fuentes de datos, estas plataformas funcionan como fuentes de datos relacionales y admiten tanto conexiones en tiempo real como extracciones. No ahondaremos aquí en este tema (consulte la sección anterior sobre fuentes de datos relacionales); simplemente mencionaremos que, por lo general, es conveniente que utilice estas plataformas como conexiones en tiempo real para evitar la transferencia de grandes volúmenes de datos provenientes de la nube.

Por último, Tableau también proporciona un conector de datos genérico para la importación de datos de servicios basados en la web que publican datos en formatos JSON, XML o HTML: el Conector de datos web.

### *Salesforce*

Cuando se conecta con Salesforce, debe tener en cuenta las siguientes limitaciones del conector:

#### No hay opción de “conexión en tiempo real”

Hay varias razones por las que elegimos limitar su uso solo a extracciones y no incluir las conexiones en tiempo real:

- Rendimiento: las consultas analíticas en tiempo real realizadas en Salesforce son, por lo general, lentas.
- Cuotas de las API: realizar consultas en Salesforce en tiempo real con mucha frecuencia puede causar la suspensión de la cuenta si se alcanzó la cuota diaria. Mediante las extracciones, hacemos un uso eficaz de las API para minimizar la cantidad de llamadas necesarias y evitar sobrepasar los límites. A fin de mantener un rendimiento óptimo y asegurar que la API de Force.com esté disponible para todos sus clientes, Salesforce.com equilibra las cargas de transacciones al imponer dos tipos de límites: Límites de solicitudes simultáneas de la API (<http://sforce.co/1f19cQa>) y Límites de solicitudes totales de la API (<http://sforce.co/1f19kiH>).

#### La extracción inicial puede ser muy lenta

La primera vez que extrae datos de Salesforce, la extracción puede tardar bastante, dependiendo del tamaño de la tabla, la carga de Force.com, etc. Esto se debe a que los objetos se descargan en su totalidad.

#### Las opciones de unión son limitadas

Cuando elige la opción de tablas múltiples, tenga en cuenta que solo puede unir objetos en sus claves PK/FK (solo uniones izquierda e interior).

#### Los datos no se pueden filtrar previamente

No cuenta con la posibilidad de filtrar datos previamente mediante el conector de Salesforce. Si este requisito es fundamental para usted, puede utilizar un controlador ODBC para Salesforce de terceros (por ejemplo, de Simba o DaraDirect) que admita conexiones en tiempo real. Luego, puede crear una extracción a partir de esta conexión.

DBAmp también proporciona una solución que le permite almacenar datos de Salesforce en una base de datos de SQL Server. Luego, puede conectarse a Tableau mediante el conector de SQL Server.

#### Las columnas de fórmulas no se pueden migrar

Si tiene campos calculados, deberá volver a crearlos en Tableau después de extraer los datos.

### Las consultas tienen un límite de 10 mil caracteres

La API de Force.com restringe las consultas a un total de 10 000 caracteres. Si se conecta a una o más tablas muy anchas (muchas columnas con nombres posiblemente muy largos), puede llegar al límite cuando intente crear una extracción. En estos casos, debe seleccionar menos columnas para reducir el tamaño de la consulta. Es posible que Salesforce.com sea capaz de ampliar el límite de consulta de su empresa en ciertos casos. Comuníquese con su administrador de Salesforce.com para obtener más información.

### Google Analytics

Google Analytics (GA) toma una muestra de sus datos cuando un informe incluye gran cantidad de dimensiones o datos. Si sus datos para una propiedad web específica y un intervalo de fechas dado superan (para una cuenta de GA normal) las 50 000 visitas, GA agregará los resultados y proporcionará un conjunto de muestra de esos datos. Cuando GA haya proporcionado un conjunto de muestra de los datos a Tableau, Tableau mostrará el siguiente mensaje en la esquina inferior derecha de la vista:

“Google Analytics devolvió datos de muestra. Las muestras ocurren cuando la conexión incluye una gran cantidad de dimensiones o de datos. Consulte la documentación de Google Analytics para obtener más información acerca de cómo las muestras afectan su informe de resultados”.

Si se toman muestras de sus datos, es importante saberlo porque agregar ciertos conjuntos de muestra de datos puede producir inferencias muy sesgadas e imprecisas. Por ejemplo, suponga que agrega un conjunto de muestra de datos que describe una categoría inusual de sus datos. Las inferencias basadas en ese conjunto de muestra agregado pueden estar sesgadas por haber un número insuficiente de elementos de muestra en esa categoría. Para crear vistas de GA que le permitan hacer inferencias precisas sobre los datos, asegúrese de contar con una muestra lo suficientemente grande para la categoría que le interesa. El tamaño de muestra mínimo recomendado es de 30.

Para obtener información sobre cómo ajustar el tamaño de muestra de GA y aprender más acerca de la técnica de muestreo de GA, consulte la documentación de GA:

<https://support.google.com/analytics/answer/1042498?hl=es>

Para evitar el muestreo se puede proceder de dos maneras:

- Ejecute varios informes de GA en el nivel de sesión o de resultado para dividir los datos en partes no muestreadas. Después, descargue los datos en un archivo de Excel y use el motor de extracción de Tableau para “agregar los datos de la fuente de datos...” a fin de reensamblarlos en un único conjunto.
- Cambie su cuenta de GA por una cuenta Premium e incremente, así, la cantidad de registros que se pueden incluir en un informe. Esto facilitará mucho la separación de los datos en distintas partes para el análisis. Para el futuro, Google anunció que permitirá a los clientes Premium de GA exportar sus datos de nivel de sesión o de resultado a Google BigQuery para un análisis más profundo. Este sería un enfoque mucho más sencillo, ya que Tableau se puede conectar directamente a BigQuery.

Por último, tenga en cuenta que la API que Tableau usa para hacer consultas a GA limita la consulta a un máximo de 7 dimensiones y 10 medidas.

### Conector de datos web

El Conector de datos web de Tableau le proporciona una forma de conectarse con datos que aún no disponen de un conector. A través de este, puede crear y utilizar conexiones con casi cualquier tipo de datos a los que se pueda acceder a través del protocolo HTTP. Esto puede incluir servicios web



internos, datos JSON, datos XML, las API de REST y muchas otras fuentes. Puesto que usted es quien controla cómo se recopilan los datos, puede incluso combinar datos de diversas fuentes.

Un conector de datos web se crea al desarrollar una página web que contenga JavaScript y HTML. Después de crear un conector de datos web, puede compartirlo con otros usuarios de Tableau publicándolo en Tableau Server.

Para ayudarlo a crear conectores de datos web, hemos confeccionado un kit de desarrollo de software (SDK), que incluye plantillas, código de muestra y un simulador que le permite probar sus conectores. En esta documentación, también se incluye un tutorial que le proporciona los pasos para crear un conector de datos web de cero. El SDK de conectores de datos web es un proyecto de código abierto. Para obtener más información, consulte la página de conectores de datos web de Tableau en GitHub.

Aquí tiene más detalles sobre los conectores de datos web:

[http://onlinehelp.tableau.com/current/api/wdc/es-es/help.htm#WDC/wdc\\_resources.htm?TocPath=3](http://onlinehelp.tableau.com/current/api/wdc/es-es/help.htm#WDC/wdc_resources.htm?TocPath=3)

## Preparación de datos

A veces, los datos que debemos usar para nuestros libros de trabajo no son perfectos. Es posible que no estén limpios ni tengan la forma adecuada. También es posible que estén distribuidos entre varios archivos y bases de datos. Históricamente, Tableau funcionaba mejor cuando trabajaba con datos limpios y normalizados, lo que implicaba que, en ocasiones, era necesario preparar los datos por medio de otras herramientas antes de cargarlos a Tableau.

Esto ya no siempre es así. Actualmente, Tableau cuenta con varias funcionalidades que lo ayudan a cargar datos desorganizados. Las funcionalidad son las siguientes:

- Tabla dinámica
- Unión
- Intérprete de datos
- Combinación de columnas

La preparación de datos instantánea a) lo ayuda a permanecer en un flujo continuo de análisis y b) permite a los usuarios sin acceso a otras herramientas continuar analizando sus datos en Tableau. Sin embargo, por lo general estas operaciones requieren un costoso proceso de cálculo (especialmente para grandes volúmenes de datos) y pueden tener un impacto en el rendimiento para los usuarios de informes. Siempre que sea posible, se recomienda utilizar una extracción de datos que materialice los datos después de usar estas operaciones.

También debería considerar estas funcionalidades como complementarias a los procesos de extracción, transformación y carga (ETL). Además, evalúe si procesar previamente los datos antes de insertarlos en Tableau es un enfoque mejor y más eficaz en el caso de que los datos se utilicen en varios informes y se compartan con otras herramientas de BI.

## Extracciones de datos

Hasta ahora hemos analizado técnicas para mejorar el rendimiento de las conexiones de datos en las que los datos permanecen en su formato original. Estas conexiones se denominan conexiones de datos en tiempo real, y, en estos casos, dependemos de la plataforma de la fuente de datos respecto a rendimiento y funcionalidad. Para mejorar el rendimiento de las conexiones en tiempo real, generalmente es necesario hacer cambios a la fuente de datos, y muchos clientes no pueden hacerlos.

Una alternativa disponible para todos los usuarios es aprovechar el veloz motor de Tableau y hacer una extracción de datos de Tableau a partir del sistema de datos de origen. Para la mayoría de los usuarios, esta es la forma más rápida y sencilla de mejorar significativamente el rendimiento de un libro de trabajo por encima de cualquier fuente de datos.

Una extracción:

- Es una memoria caché de datos duradera que se graba en el disco y puede reproducirse.
- Es un almacén de datos en columnas (un formato en el que los datos están optimizados para realizar consultas para análisis).
- Está completamente desconectada de la base de datos durante la consulta. De hecho, la extracción reemplaza la conexión de datos en tiempo real.
- Puede actualizarse, mediante la generación de una nueva extracción o la incorporación progresiva de filas de datos a una extracción existente.
- Reconoce la arquitectura (a diferencia de la mayoría de las tecnologías en memoria, no se ve restringida por la cantidad de RAM física disponible).
- Es portátil (las extracciones se almacenan como archivos, por lo que pueden copiarse en un disco duro local o usarse cuando el usuario no está conectado a la red empresarial; también pueden usarse para insertar datos en libros de trabajo empaquetados que se distribuyan para su uso con Tableau Reader).
- Suele ser más rápida que la conexión de datos en tiempo real subyacente.

Tom Brown de The Information Lab escribió un artículo excelente donde explica distintos casos de uso en los que las extracciones resultan beneficiosas (asegúrese de leer también los comentarios para conocer más ejemplos de otros usuarios):

<http://bit.ly/1F2iDnT>

Hay una cuestión importante que debe conocer acerca de las extracciones de datos: no reemplazan el almacén de datos, sino que lo complementan. Aunque pueden usarse para recopilar y agregar datos a lo largo del tiempo (es decir, agregar datos progresivamente en función de un ciclo periódico), esto debe tomarse como una solución táctica, no a largo plazo. Las actualizaciones progresivas no admiten acciones de actualización o eliminación de registros que ya se hayan procesado (para cambiar estos registros se debe volver a cargar la extracción por completo).

Por último, no se pueden crear extracciones de fuentes de datos OLAP, como SQL Server, Analysis Services u Oracle Essbase. La excepción a esta regla es que puede crear extracciones de SAP BW (consulte la sección pertinente anterior).

**Cuándo usar extracciones y cuándo usar conexiones en tiempo real**

Como en todos los ámbitos, hay un momento y un lugar para las extracciones de datos. Las siguientes son algunas situaciones en las que las extracciones pueden resultar beneficiosas:

- Lentitud en la ejecución de consultas: si su sistema de datos de origen tarda en procesar las consultas que genera Tableau Desktop, crear una extracción puede ser una manera sencilla de mejorar el rendimiento. El formato de los datos extraídos está diseñado intrínsecamente para proporcionar una respuesta rápida a las consultas de análisis, por lo que, en este caso, se puede pensar en la extracción como una memoria caché de aceleración de consultas. Para algunos tipos de conexión, esta es una práctica recomendada (p. ej., archivos de texto grandes, conexiones de SQL personalizado), y algunas fuentes solo funcionan con este modelo (consulte la sección sobre fuentes de datos en la nube).

- **Análisis sin conexión:** si necesita trabajar con los datos cuando la fuente de datos original no está disponible (p. ej. está desconectado de la red cuando viaja o trabaja desde su casa). Las extracciones de datos persisten como los archivos y pueden copiarse fácilmente en dispositivos portátiles. Es muy sencillo alternar entre una extracción y una conexión en tiempo real si entra y sale de la red.
- **Libros de trabajo empaquetados para Tableau Reader, Tableau Online o Tableau Public:** si planea compartir sus libros de trabajo para que otros usuarios puedan abrirlos en Tableau Reader o si planea publicarlos en Tableau Online o Tableau Public, deberá insertar los datos en un archivo de libro de trabajo empaquetado. Incluso si el libro de trabajo usa fuentes de datos que también puedan insertarse (es decir, fuentes de datos de archivos), las extracciones de datos proporcionan, intrínsecamente, un alto nivel de compresión de datos, de modo que el libro de trabajo empaquetado resultante es considerablemente menor.
- **Funcionalidad adicional:** algunas fuentes de datos (p. ej., fuentes de datos de archivos mediante el controlador JET heredado) no admiten ciertas funciones de Tableau Desktop (p. ej., mediana, recuento de valores diferenciados, jerarquización, agregaciones de percentiles, operaciones de entrada y salida de conjuntos, etc.). Una manera sencilla de habilitar estas funciones es extraer los datos.
- **Seguridad de datos:** si desea compartir un subconjunto de los datos del sistema de datos de origen, puede crear una extracción y ponerla a disposición de otros usuarios. Puede limitar los campos y las columnas que se incluyen y compartir datos agregados cuando desee que los usuarios vean valores resumidos, pero no los datos individuales registrados.

Las extracciones son muy eficaces, pero no son soluciones mágicas para todos los problemas.

Existen algunas situaciones en las que el uso de extracciones podría no ser adecuado:

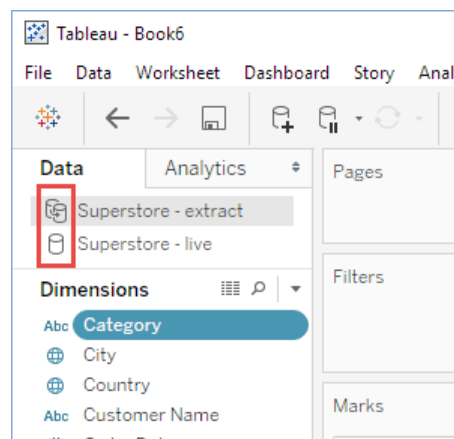
- **Datos en tiempo real:** debido a que las extracciones son una foto instantánea de los datos en un momento específico, no son apropiadas para análisis que requieren datos en tiempo real. Es posible actualizar automáticamente las extracciones con Tableau Server, y muchos clientes lo hacen varias veces al día, pero el verdadero acceso a los datos en tiempo real requiere una conexión del mismo tipo.
- **Datos inmensos:** si tiene que trabajar con una cantidad inmensa de datos (la definición de "inmensa" depende del usuario, pero generalmente será de millones o miles de millones de registros), extraerlos puede no ser práctico. El archivo de extracción resultante puede ser extremadamente grande o el proceso de extracción puede llevar muchísimas horas. Pero hay un par de excepciones a esta regla. Si tiene un conjunto de datos de origen inmenso, pero va a trabajar con un subconjunto filtrado, muestreado o agregado de esos datos, usar una extracción puede ser una excelente idea. En términos generales, el motor de extracción de Tableau está diseñado para funcionar bien con hasta unos cientos de millones de registros, pero esta cantidad se verá afectada por la forma y la cardinalidad de sus datos.
- **Funciones de paso a través de SQL:** si su libro de trabajo usa funciones de paso, estas no responderán con una extracción de datos.
- **Seguridad sólida de nivel de usuario:** si tiene que proporcionar seguridad sólida de nivel de usuario, debe implementarla en la fuente de datos. Si se aplicaron filtros de usuario en el libro de trabajo, los usuarios pueden quitarlos si se les otorga acceso a todos los datos de la extracción. La excepción a esta regla sucede cuando la extracción se publica en Tableau Server con filtros definidos para la fuente de datos y otros usuarios acceden a la extracción mediante el servidor de datos. Nota: deberá garantizar que se revoken los permisos de descarga de los usuarios para impedir que omitan los filtros establecidos.

## Creación de extracciones en Tableau Desktop

En la mayoría de los casos, la creación inicial de una extracción se realiza en Tableau Desktop y es muy sencilla. Después de conectarse a sus datos, vaya al menú Datos y haga clic en “Extraer datos”, luego, acepte las opciones predeterminadas del cuadro de diálogo (volveremos a tratar este tema más adelante). Tableau le pedirá que especifique dónde quiere guardar la extracción, elija el lugar en el que desee guardar el archivo, aunque Tableau probablemente lo dirija a “My Tableau Repository | Datasources”, que también es una ubicación apropiada.

Ahora, espere a que se cree la extracción. El tiempo de espera depende de la tecnología de la base de datos que use, la velocidad de la red, el volumen de datos, etc.; además, depende de la velocidad y la capacidad de su estación de trabajo, ya que la creación de una extracción es una tarea que hace un uso intensivo de la memoria y el procesador.

Sabrás que la extracción finalizó cuando vea que el icono de la fuente de datos cambió y tiene otro icono de base de datos detrás, que representa una “copia”, que es exactamente lo que es una extracción.



Cuando crea una extracción de esta forma (a través de Tableau Desktop), el procesamiento se realiza en su estación de trabajo, por lo que deberá asegurarse de que tenga la capacidad suficiente para completar la tarea. La creación de extracciones usa recursos de todos los tipos (CPU, RAM, almacenamiento en disco, E/S de la red), y procesar grandes volúmenes de datos en un equipo pequeño puede generar errores si cualquiera de ellos se agota. Se recomienda hacer extracciones grandes en una estación de trabajo adecuada (CPU rápida con múltiples núcleos, mucha RAM, E/S rápidas, etc.).

El proceso de extracción requiere espacio temporal en el disco para almacenar los archivos de trabajo (hasta dos veces el tamaño del archivo de extracción final). Este espacio de trabajo se ubica en el directorio especificado por la variable de entorno TEMP (habitualmente `C:\WINDOWS\TEMP` o `C:\Users\USERNAME\AppData\Local\Temp`). Si esta unidad no es lo suficientemente grande, dirija la variable de entorno a una ubicación con más espacio.

Si es imposible (o no es práctico) llevar a cabo un proceso de extracción inicial en una estación de trabajo, se puede hacer lo siguiente para crear una extracción vacía que después se publicará en Tableau Server. Cree un campo calculado que contenga `DateTrunc("minute", now())`. A continuación, agréguelo a los filtros extraídos y excluya el único valor que muestra (hágalo rápido porque después de un minuto el filtro pierde validez). Si necesita más tiempo, amplíe el intervalo de publicación (p. ej., 5 o 10 minutos, o una hora si la necesita). Esto creará una extracción vacía en su escritorio. Cuando la publique en el servidor y active el programa de actualización, llenará toda la extracción, ya que la marca de tiempo que excluimos ya no es la misma.

## Creación de extracciones mediante la API de extracción de datos

Tableau también proporciona una interfaz de programación de aplicaciones (API) que permite a los desarrolladores crear directamente un archivo de Extracción de datos de Tableau (TDE). Los desarrolladores pueden usar esta API para generar extracciones a partir de software en las instalaciones o de software como servicio. La API permite a Tableau incorporar datos de manera sistemática cuando no exista un conector nativo para la fuente de datos que esté utilizando.

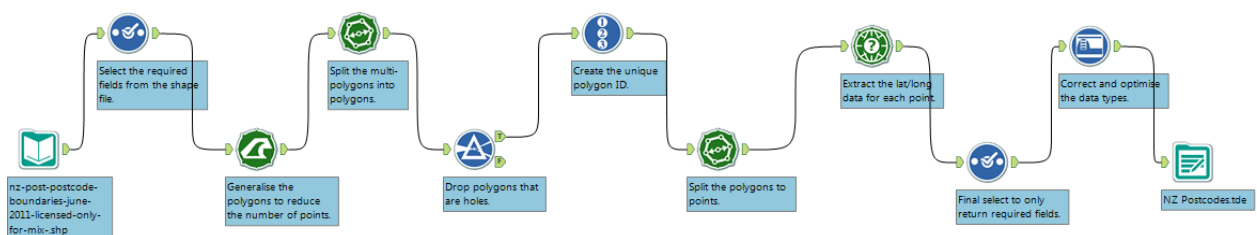
Esta API está disponible para desarrolladores en Python y C/C++/Java tanto en Windows como en Linux. Puede encontrar más información sobre la API aquí:

[http://onlinehelp.tableau.com/current/pro/desktop/es-es/extracting\\_TDE\\_API.html](http://onlinehelp.tableau.com/current/pro/desktop/es-es/extracting_TDE_API.html)

## Creación de extracciones a través de herramientas de terceros

Los desarrolladores de muchas herramientas de terceros utilizaron la API de extracción de datos para incorporar el almacenamiento en formato nativo TDE a sus aplicaciones. Entre estas aplicaciones, se incluyen plataformas de análisis, como Adobe Marketing Cloud, y herramientas ETL, como Alteryx e Informatica.

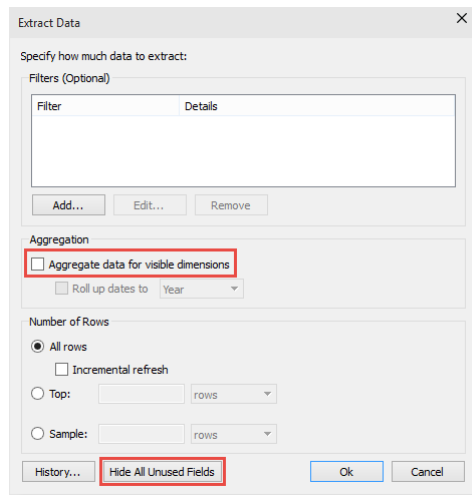
Si necesita preparar datos complejos, las herramientas como Alteryx e Informatica se pueden utilizar eficazmente para las etapas de ETL y, luego, almacenar los datos preparados directamente en un archivo TDE para utilizar en Tableau Desktop.



## Extracciones agregadas

El uso de extracciones agregadas siempre mejora el rendimiento. Incluso si trabaja en Teradata o Vertica con una enorme cantidad de datos, la extracción puede implicar mejoras, siempre que los datos se agreguen y filtren adecuadamente. Por ejemplo, puede filtrar los datos si solo le interesan los más recientes.

Puede definir la extracción con antelación eligiendo los campos que desee y activando la casilla de verificación “Añadir datos para dimensiones visibles” en el cuadro de diálogo Extraer datos de Tableau Desktop. Otra opción, después de hacer el análisis y crear el dashboard, cuando esté listo para publicar, es volver al cuadro de diálogo Extraer datos y hacer clic en el botón Ocultar campos no utilizados. Entonces, al extraer los datos se obtiene el mínimo absoluto necesario para crear la vista.



La creación de extracciones agregadas es una técnica eficaz cuando cuenta con una gran cantidad de bases de datos, pero debe crear vistas resumidas que consulten todo el conjunto de datos. Por ejemplo, es posible que tenga mil millones de registros de datos con transacciones detalladas, que representan 10 años de ventas. Desea comenzar mostrando la tendencia general de ventas de los 10 años. La consulta para esta vista inicial será potencialmente lenta, puesto que debe consultar todas las mil millones de filas. Al crear una extracción que esté agregada en el nivel de año, podemos reducir los esfuerzos de consulta necesarios para la visualización, puesto que solo tendremos 10 números en la extracción. Por supuesto, este es un ejemplo muy simplificado. En la realidad, tendrá más dimensiones que tiempo, pero es útil reducir significativamente la cantidad de registros que deben consultarse en el momento de la visualización.

Podemos crear libros de trabajo muy sofisticados, con varios niveles de detalle, mediante la creación de varias extracciones agregadas, cada una configurada para dar soporte a un nivel de detalle específico, o a través de la combinación de extracciones agregadas y conexiones en tiempo real. Por ejemplo, es posible que tenga un conjunto de vistas resumidas iniciales que usen una extracción con una alta agregación, pero, cuando explora para ver los detalles, utiliza filtros de acción de las vistas resumidas de otra hoja que se conecta mediante una conexión en tiempo real. Esto implica que las vistas resumidas serán más rápidas, ya que no deben rastrear todo el conjunto de base de datos. Pero tampoco debemos extraer toda la base de datos para el correcto funcionamiento de las acciones de exploración en detalle. Además, la conexión en tiempo real funcionará rápidamente, porque, en el nivel de exploración, solo estamos accediendo a un conjunto pequeño de registros.

De este modo, puede hacer combinaciones y agregaciones en diferentes niveles para resolver casi cualquier problema de rendimiento y obtener los resultados lo más rápido posible. Como Tableau hace un uso eficaz de la memoria, mejorar el rendimiento de esta manera suele ser relativamente fácil y permite ejecutar distintas extracciones al mismo tiempo.

### Optimización de extracciones

Tableau Server no solo optimiza las columnas físicas de la base de datos, sino también las columnas adicionales creadas en Tableau. Estas columnas incluyen los resultados de cálculos determinísticos, como la manipulación y la concatenación de cadenas, en los que el resultado, los grupos y los conjuntos nunca cambiarán. Los resultados de los cálculos no determinísticos, como los que incluyen parámetros o agregaciones (como sumas o promedios) que se calculan en tiempo de ejecución, no se pueden almacenar.

Un usuario puede actualizar una extracción después de agregar solo dos filas de datos y observar que el tamaño de la extracción aumentó de 100 MB a 120 MB. Este incremento de tamaño se debe a que para la optimización se crean columnas adicionales con valores de campos calculados porque es menos costoso almacenar datos en el disco que recalcularlos cada vez que se necesitan.

Algo a tener en cuenta es que si hace copias duplicadas de una conexión en una extracción de datos, debe asegurarse de que todos los campos calculados existan en la conexión que seleccione para las opciones “Optimizar” y “Actualizar”. De lo contrario, Tableau no mostrará campos que considere en desuso. Un buen método es definir todos los campos calculados de la fuente de datos principal y copiarlos, según sea necesario, en las demás conexiones y solamente actualizar y optimizar la extracción de la fuente de datos principal.

*Advertencia: un miembro confiable de nuestro equipo de desarrollo comentó que, a pesar de que es posible crear múltiples conexiones con un único archivo TDE, nunca fue desarrollado para ello. Puede causar muchos problemas si las conexiones no están sincronizadas con la estructura del archivo TDE. Aconseja no hacerlo.*

### Actualización de extracciones

En Tableau Desktop, para actualizar una extracción, se hace una selección en el menú (menú Datos > [su fuente de datos] > Extraer > Actualizar), que actualiza los datos y agrega cualquier fila nueva. Pero en Tableau Server, durante o después del proceso de publicación, se puede adjuntar un programa definido por un administrador para actualizar la extracción automáticamente. El incremento más pequeño permitido para el programa es de 15 minutos; el programa puede consistir en hacer una actualización a la misma hora todos los días, todas las semanas, etc. Puede establecer una “ventana móvil” para actualizar continuamente los datos y ver solo los más recientes.

Nota: si desea actualizar sus datos con mayor frecuencia que cada 15 minutos, debería considerar conectarse a datos activos o establecer una base de datos de informes sincronizados.

Puede elegir dos programas de actualización para una misma extracción:

- Una actualización incremental solo agrega filas y no incluye cambios de las filas existentes.
- Una actualización completa descarta la extracción vigente y genera una nueva a partir de la fuente de datos.

### ¿Qué sucede si la actualización tarda más que el incremento?

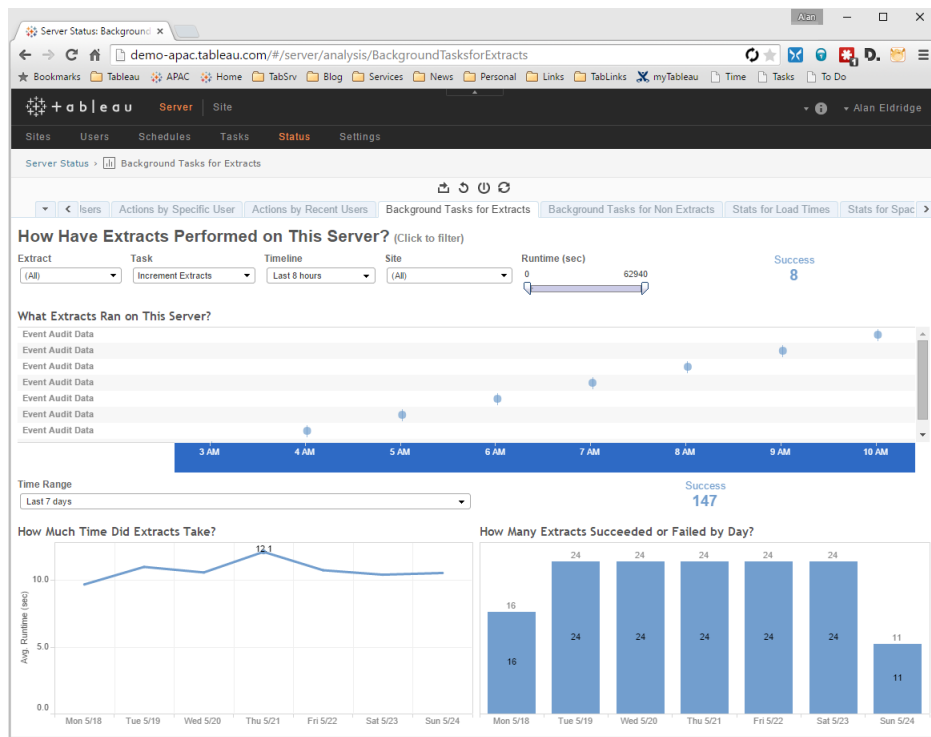
Si la duración de la actualización de la extracción tarda más tiempo que el incremento, se omiten las actualizaciones programadas. Por ejemplo, se establece un programa para actualizar los datos cada hora, pero la cantidad de datos es tan grande que la actualización lleva una hora y media. En ese caso:

- La primera actualización comienza a la 01:00 y acaba a las 02:30.
- La próxima actualización está programada para comenzar a las 02:00, pero, debido a que la primera está todavía en ejecución, se omite.
- La siguiente actualización comienza a las 03:00 y acaba a las 04:30.

### Mantenimiento de extracciones

Las pantallas de mantenimiento muestran qué tareas de fondo se están ejecutando, además de las que se ejecutaron durante las últimas 12 horas. Se usa codificación por colores para mostrar el estado de esas tareas. Las pantallas de mantenimiento están disponibles para los administradores y, con los permisos adecuados, para algunos usuarios, que pueden estar autorizados a iniciar una actualización ad hoc

de una extracción. Además, por ejemplo, si se va a cargar una base de datos, puede establecer un desencadenador para que inicie una extracción después de que la base de datos se haya cargado.



También puede actualizar un libro de trabajo de manera incremental o completa con la herramienta de línea de comandos `tabcmd` si usa Tableau Server o la línea de comandos `Tableau.exe` si usa Tableau Desktop. Si tiene requisitos de programación complicados, puede invocarlos desde una herramienta de programación externa, como el Programador de tareas de Windows. Este enfoque es necesario si desea un ciclo de actualización más corto que el mínimo de 15 minutos permitido en la interfaz de Tableau Server.

Por último, para los usuarios de Tableau Online, puede utilizar el cliente de sincronización para mantener actualizadas las fuentes de datos de las instalaciones mediante un programa definido en el servicio en línea. Puede encontrar más información sobre esta utilidad aquí:

[http://onlinehelp.tableau.com/current/online/es-es/qs\\_refresh\\_local\\_data.htm](http://onlinehelp.tableau.com/current/online/es-es/qs_refresh_local_data.htm)

## Administración de datos

Otra forma de conectarse a fuentes de datos es mediante el servidor de datos de Tableau Server, aunque este no sea exactamente una fuente de datos. El servidor de datos admite tanto conexiones en tiempo real como extracciones de datos y proporciona varias ventajas en comparación con las conexiones de datos independientes:

- Dado que los metadatos se almacenan de manera centralizada en Tableau Server, varios libros de trabajo y varios autores o analistas pueden compartirlos. Los libros de trabajo conservan una referencia a la definición de los metadatos centralizados y, cada vez que se abren, controlan si se hicieron cambios. En caso afirmativo, se pide al usuario que actualice la copia insertada en el libro de trabajo. Esto significa que los cambios de lógica de negocios solo deben realizarse en un lugar y pueden propagarse a todos los libros de trabajo dependientes.
- Si la fuente de datos es una extracción de datos, se puede usar en múltiples libros de trabajo. Sin el servidor de datos, cada libro de trabajo tendrá su propia copia local de la extracción.



Esto reduce la cantidad de copias redundantes que, a su vez, reduce el espacio de almacenamiento necesario en el servidor y los procesos de actualización de copias.

- Si la fuente de datos es una conexión en tiempo real, no es necesario instalar los controladores en todos los equipos, sino que solo deben instalarse en Tableau Server. El servidor de datos actúa como proxy para las consultas de Tableau Desktop.

## ¿El problema es mi entorno?

Hay momentos en los que un libro de trabajo funciona bien en evaluaciones de un único usuario, pero, cuando se implementa en Tableau Server (o en Tableau Online) para que lo consuman muchos usuarios, funciona mal. La siguiente sección identifica áreas en las que las divergencias entre situaciones de uno o más usuarios pueden marcar una diferencia.

### Actualice

El equipo de desarrollo de Tableau trabaja continuamente para mejorar el rendimiento y la capacidad de uso de nuestro software. En algunas ocasiones, la actualización a la última versión de Tableau Server puede producir mejoras significativas en el rendimiento, sin necesidad de implementar cambios en el libro de trabajo.

Consulte la página de Notas de la versión de Tableau y actualice a la última versión disponible que pueda:

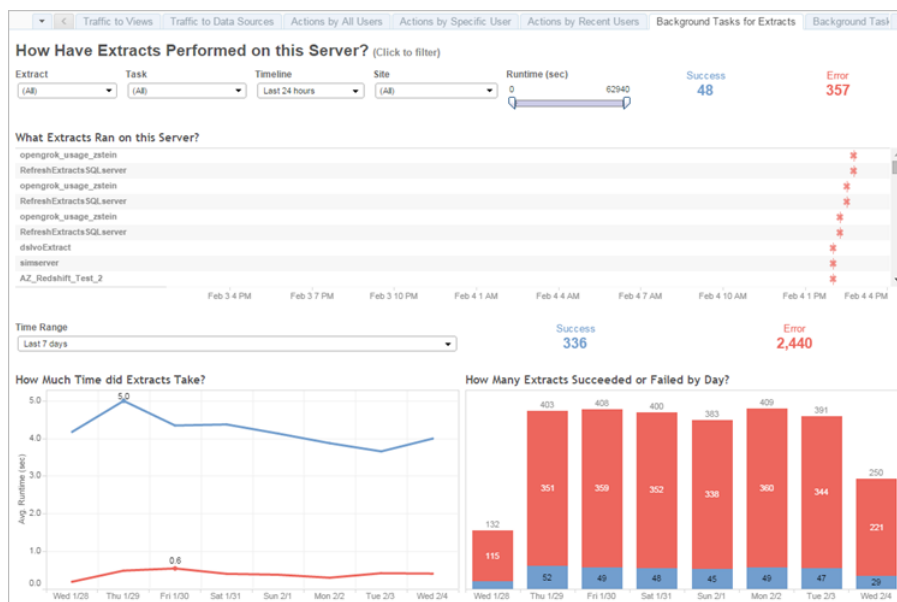
<http://www.tableau.com/es-es/support/releases>

### Pruebe Tableau Desktop en el servidor

En algunas ocasiones, es posible que se encuentre en una situación en la que tiene un libro de trabajo con un buen rendimiento en Tableau Desktop en su propia estación de trabajo, pero su rendimiento es deficiente en Tableau Server. Al abrir el libro de trabajo en una copia de Tableau Desktop instalada en una máquina de Tableau Server, es posible determinar si existe algún problema con su libro de trabajo o si hay un problema de configuración del servidor. A través de este método se puede identificar si es un problema de incompatibilidad con el controlador o si hay un inconveniente con la red, como un enrutamiento incorrecto, o con la configuración de DNS o el proxy.

### Separe las actualizaciones y las cargas de trabajo interactivas

Si el rendimiento del servidor es reducido, use la vista administrativa Tareas de fondo para ver sus programas de tareas de actualización vigentes.



Si puede programar actualizaciones para los horarios de bajo consumo, hágalo. Si su configuración de hardware lo permite, también puede trasladar los procesos del componente de segundo plano a un nodo de trabajo dedicado.

## Supervise y ajuste su servidor

Tableau Server viene con varias vistas para administradores, con el fin de ayudar a supervisar la actividad en Tableau Server. Las vistas se ubican en la tabla Análisis en la página Mantenimiento del servidor:

Views	Analysis
Traffic to Views	View count, viewers, and viewer behavior for published views.
Traffic to Data Sources	Data source usage, users, and user behavior for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.

Puede encontrar más información sobre estas vistas en el siguiente enlace:

<http://onlinehelp.tableau.com/current/server/es-es/adminview.htm>

Además, puede crear vistas administrativas personalizadas conectándose a la base de datos PostgreSQL que forma parte del repositorio de Tableau. Puede encontrar instrucciones aquí:

[http://onlinehelp.tableau.com/current/server/es-es/adminview\\_postgres.htm](http://onlinehelp.tableau.com/current/server/es-es/adminview_postgres.htm)

Compruebe el límite de tiempo de espera de la sesión VizQL

El límite predeterminado de tiempo de espera de la sesión VizQL es de 30 minutos. Aunque la sesión VizQL esté inactiva, aún consume memoria y ciclos de CPU. Si puede utilizar un límite inferior, use tabadmin para cambiar la configuración de vizqlserver.session.expiry.timeout.

[http://onlinehelp.tableau.com/current/server/es-es/reconfig\\_tabadmin.htm#ida6864815-db67-4a51-b8b6-c93617582091](http://onlinehelp.tableau.com/current/server/es-es/reconfig_tabadmin.htm#ida6864815-db67-4a51-b8b6-c93617582091)

Evalúe su configuración de proceso

Tableau Server se divide en seis componentes diferentes llamados procesos del servidor. Aunque la configuración predeterminada se diseñó para que funcione en un amplio rango de situaciones, usted puede modificarla para lograr diferentes objetivos de rendimiento. Específicamente, puede controlar en qué equipos se ejecutan los procesos y cuántos procesos se ejecutan. Consulte las pautas para mejorar el rendimiento del servidor en implementaciones de uno, dos y tres equipos:

[http://onlinehelp.tableau.com/current/server/es-es/perf\\_extracts\\_view.htm#idd21e8541-07c4-420f-913e-92dcf5f0c34](http://onlinehelp.tableau.com/current/server/es-es/perf_extracts_view.htm#idd21e8541-07c4-420f-913e-92dcf5f0c34)

## Infraestructura

### 64 bits

Las versiones de Tableau Server anteriores a Tableau 10 pueden ejecutarse en sistemas operativos de Microsoft de 32 bits. Sin embargo, se recomienda encarecidamente a los clientes que instalen la versión de 64 bits en sus entornos de producción. Las instalaciones de 32 bits solo se recomiendan para instalaciones de desarrollo y prueba.

A partir de Tableau 10, Tableau Server solo está disponible como aplicación de 64 bits.

## Agregue más CPU/RAM

Independientemente de si ejecuta Tableau Server en una máquina o en varias, la regla general es que más núcleos de CPU y más RAM le brindarán mejor rendimiento. Asegúrese de cumplir con los requisitos recomendados de hardware y software (<http://onlinehelp.tableau.com/current/server/es-es/requ.htm#ida4d2bd02-00a8-49fc-9570-bd41140d7b74>) de Tableau Server y consulte Cuándo agregar trabajadores y reconfigurar ([http://onlinehelp.tableau.com/current/server/es-es/distrib\\_when.htm#idfdd60003-298e-47e6-8133-3d2490e21e07](http://onlinehelp.tableau.com/current/server/es-es/distrib_when.htm#idfdd60003-298e-47e6-8133-3d2490e21e07)) para evaluar si debe agregar equipos adicionales.

TabMon (mencionado anteriormente en este documento) es una herramienta excelente para recopilar datos de uso que lo puedan ayudar en el proceso de planeamiento de capacidad.

## No descarte la E/S

Algunas acciones de Tableau implican una E/S intensiva (p. ej., la carga, la creación y la actualización de extracciones de datos) y se beneficiarían del uso de SSD, en lugar de discos giratorios. Russell Christopher publicó varios artículos fantásticos en su blog Tableau Love, en los que se explora el impacto de la cantidad de núcleos, la velocidad de la CPU y la E/S por segundo sobre el rendimiento general. Aunque este experimento se realizó con AWS, es aplicable a todos los entornos:

<http://bit.ly/1f17oa3>

<http://bit.ly/1f17n50>

## Infraestructura física y virtual

Actualmente, muchos clientes están implementando Tableau Server en infraestructuras virtuales. La virtualización siempre implica una sobrecarga al rendimiento, por lo que no será tan rápida como la instalación física. Sin embargo, la moderna tecnología de hipervisores redujo drásticamente esta sobrecarga.

Cuando realice instalaciones en máquinas virtuales, es importante asegurarse de que Tableau Server cuente con memoria RAM y recursos de CPU dedicados. Si compite por recursos con otras máquinas virtuales en el host físico, esto puede tener un impacto significativo en el rendimiento.

Un buen recurso para configurar las implementaciones virtuales es el informe de VMWare denominado “Deploying Extremely Latency-Sensitive Applications in vSphere 5.5” (Implementación de aplicaciones extremadamente sensibles a la latencia en vSphere 5.5). En la página 15 se proporciona una lista general de las prácticas recomendadas para las aplicaciones sensibles a la latencia:

<http://www.vmware.com/latam.html>

## Navegador

Tableau hace un uso intensivo de JavaScript, de modo que la velocidad del intérprete de JavaScript del navegador afecta la velocidad de la representación. Todos los navegadores modernos tienen un desempeño similar a pesar de ser un campo que cambia rápidamente. Sin embargo, merece la pena plantearse si la versión y el rendimiento de su navegador pueden estar afectando su experiencia.

## Conclusión

Como alguien muy sabio me dijo una vez (y que me dio algunos consejos para crear este documento): “Diles lo que les vas a decir, díselo, y entonces vuelve a decírselo”. Un consejo muy sabio, desde luego.

Por eso, nuevamente, aquí les presento los puntos clave que espero que hayan incorporado gracias a la lectura de este documento:

- No hay soluciones mágicas. El rendimiento puede ser lento por una gran cantidad de razones. Recopile datos que lo ayuden a identificar dónde hay demoras. Luego, concentre sus mejoras en el área más costosa y pase al área siguiente, y así sucesivamente. Deténgase cuando sea lo suficientemente rápido o cuando la relación esfuerzo-recompensa ya no sea viable.
- Muchos dashboard son lentos porque su diseño es deficiente. Simplifique. No muestre demasiado a la vez. Utilice diseños de análisis guiado cuando sea posible.
- No trabaje con datos que no necesite. Utilice filtros, oculte campos que no use y emplee agregaciones.
- No luche con el motor de datos. Es inteligente y hace lo posible para ayudarlo, por lo que debe confiar en que generará consultas eficaces.
- Cuanto más limpios estén sus datos y mejor concuerden con la estructura de sus preguntas, más rápido se ejecutarán sus libros de trabajo. Y usted será más feliz.
- Las extracciones son una forma sencilla y rápida de ejecutar con mayor velocidad la mayoría de los libros de trabajo. Si no necesita datos en tiempo real y no está trabajando con miles de millones de filas de datos, pruébelos.
- Las cadenas y las fechas son lentas, mientras que los números y los valores booleanos son rápidos.
- A pesar de que este documento está basado en los mejores consejos de muchas personas inteligentes, sus recomendaciones son solo recomendaciones. Deberá comprobar cuáles mejorarán el rendimiento en su caso particular.
- Cuando esté trabajando con pequeños volúmenes de datos, mucho de esto no será relevante. Puede manipularlos con la fuerza bruta y una mala técnica. Sin embargo, no será dañino seguir estas recomendaciones en todos sus libros de trabajo, puesto que nunca se sabe si sus datos incrementarán en algún momento futuro.
- La práctica hace al maestro.

Ahora, póngase en marcha y genere muchos libros de trabajo eficaces.