

Conversación con Gemini

Exported: 15/6/2026 10:03:54

Jordi H.

Has dicho tengo un contenedor ejecutando mongodb 8.2.1 ¿tiene instalado ya el driver odbc?

Gemini

Ir directo al grano: **No, por defecto el contenedor oficial de MongoDB no incluye el driver ODBC.** Aunque suene un poco contraintuitivo, los motores de base de datos y los drivers que permiten conectarse a ellos son piezas de software distintas. El contenedor de MongoDB trae el **servidor** (el proceso mongod) y herramientas de shell, pero los drivers ODBC/JDBC suelen instalarse en el lado del **cliente** (la aplicación o servidor que quiere leer los datos). Aquí te explico por qué y qué opciones tie

1. ¿Por qué no viene instalado?

- Filosofía de Docker: Las imágenes oficiales buscan ser lo más ligeras posible. Incluir drivers ODBC (que dependen de gestores como unixODBC y librerías adicionales) engordaría la imagen innecesariamente para la mayoría de los usuarios.
- Arquitectura: El driver ODBC de MongoDB es técnicamente un "conector" que traduce consultas SQL a la API de MongoDB. Normalmente, esto se instala en la máquina que tiene el software de BI (como Power BI, Tableau o Excel) o el servidor de aplicaciónes.

2. Diferencia entre Driver y Servicio

Es importante no confundir el driver con el **MongoDB Connector for BI**.

- MongoDB 8.0+: MongoDB ha mejorado mucho sus capacidades nativas, pero para usar ODBC sigues necesitando el componente intermedio que "entienda" SQL.
- El Driver: Es una librería .so (en Linux) o .dll (en Windows) que permite a aplicaciones externas hablar con el clúster.

3. ¿Qué hacer si realmente lo necesitas dentro del contenedor?

Si tienes un caso de uso muy específico donde el contenedor de MongoDB debe actuar como cliente de sí mismo vía ODBC, tendrías que crear un Dockerfile personalizado: Dockerfile

CODE

```
FROM mongodb/mongodb-community-server:8.2.1-ubuntu22.04

USER root
RUN apt-get update && \
  apt-get install -y unixodbc odbcinst && \
  # Aquí descargarías e instalarías el .deb del driver oficial de MongoDB
  # ...
USER mongodb
```

¿Tu intención es conectar alguna herramienta externa (como Excel o Power BI) al contenedor? Si es así, puedo explicarte cómo configurar el driver en tu sistema operativo host para que apunte a la IP del contenedor.

Jordi H.

Has dicho si, eso es

Gemini

¡Perfecto! Entonces el enfoque cambia totalmente: **no necesitas tocar nada dentro del contenedor**. Para conectar una herramienta externa (Excel, Power BI, Tableau) a tu MongoDB en Docker, el "puente" se construye en tu **máquina local** (el Host). Aquí tienes los pasos para dejarlo funcionando

1. Mapeo de Puertos

Primero, asegúrate de que al levantar tu contenedor hayas expuesto el puerto de MongoDB. Sin esto, el driver ODBC no "verá" la base de datos desde fuera de Docker

- El comando debió ser algo como: `docker run -p 27017:27017 ...`
- Si usas Docker Desktop, el host al que te conectarás será localhost.

2. Instalación del Driver (En tu PC)

Debes descargar e instalar el driver oficial en el sistema operativo donde corre tu herramienta de análisis:

1. Ve al Centro de Descargas de MongoDB.
2. Descarga la versión compatible con tu SO (ej. el MSI para Windows de 64 bits).
3. Instálalo siguiendo el asistente.

3. Configuración del DSN (Data Source Name)

Una vez instalado el driver, debes configurarlo para que sepa dónde está el contenedor:

1. Abre el Administrador de orígenes de datos ODBC en tu Windows.
2. Ve a la pestaña System DSN (o User DSN) y haz clic en Add.
3. Selecciona MongoDB ODBC Driver de la lista.
4. Rellena los datos de conexión:
 - Server: localhost (si el contenedor está en la misma PC).
 - Port: 27017.
 - Database: El nombre de tu base de datos.
 - Mechanism: SCRAM-SHA-1 o SCRAM-SHA-256 (si tienes autenticación activada).

Un detalle crucial: El "SQL" en MongoDB

MongoDB es NoSQL (documentos JSON). Los drivers ODBC modernos suelen requerir que la base de datos tenga un esquema definido o que uses el **Atlas SQL Interface** (si estás en la nube) o el **BI Connector** (si es On-Premise/Docker)

- Si usas Power BI: A veces es más sencillo usar el conector nativo de MongoDB que ya viene en Power BI Desktop, en lugar de pelearte con ODBC
- Si usas Excel: ODBC sigue siendo la mejor ruta.

¿Qué herramienta específica vas a usar para conectarte? (Si me dices cuál es, puedo darte el truco exacto para que no te des errores de esquema)

Jordi H.

Has dicho esto es un lío. todo empezó porque mi servidor linux tiene un contenedor con una Community edition de mongodb. cuando la consulto con powerBI tiene un rendimiento muy lento si es via mongosql (un a pasarela que está a punto de estar obsoleta). he podido comprobar que mis compañeros que tienen el servid

or en windows les va más rápido si usan los conectores de powerbi como si la base de datos fuese mysql. a partir de ahí, he estado investigando cómo conectar powerbi a mi mongo en linux para aumentar el rendimiento. cómo

Gemini

Te entiendo perfectamente, estás en medio de la transición tecnológica de MongoDB. El "lío" viene de que MongoDB está retirando el **BI Connector (mongosqld)** en favor de nuevas arquitecturas, y la diferencia de rendimiento que ves con tus compañeros de Windows tiene una explicación técnica muy clara. Aquí tienes el análisis de por qué a ellos les va más rápido y cómo puedes replicarlo (o mejorarlo) en tu entorno Linux.

1. El secreto de tus compañeros: El Driver ADO.NET / ODBC de "Nueva Generación"

Si tus compañeros conectan Power BI a MongoDB como si fuera MySQL y les va rápido, es porque probablemente están usando el **MongoDB ODBC Driver con soporte SQL** (que internamente mapea el esquema) o el conector nativo de Power BI que utiliza el protocolo **Atlas SQL / Interface VPC**. La diferencia de velocidad es que mongosqld es un "proxy" que traduce cada consulta sobre la marcha, lo cual es ineficiente. Los conectores modernos crean un esquema relacional virtual mucho más ligero.

2. Tu situación: MongoDB Community en Linux + Power BI

Al usar la **Community Edition**, no tienes acceso nativo a "Atlas SQL" (que es la solución de alto rendimiento de MongoDB). Para mejorar el rendimiento en tu caso, tienes tres caminos.

Opción A: Usar el conector nativo de MongoDB en Power BI (Recomendado)

Power BI ya tiene un conector específico para MongoDB.

- Ventaja: No necesitas mongosqld.
- Cómo hacerlo: En Power BI, ve a Obtener Datos -> Base de datos -> MongoDB Atlas.
- El truco: Aunque diga "Atlas", a veces funciona con instancias locales si tienes habilitada la autenticación y el puerto abierto. Sin embargo, suele requerir el VPC Peering o el BI Connector en versiones antiguas.

Opción B: Instalar el Driver ODBC en tu Windows (Donde está Power BI)

Esta es la forma de replicar lo que hacen tus compañeros:

1. En tu PC (Windows): Instala el MongoDB ODBC Driver.
2. Configuración: Configura un DSN que apunte a la IP de tu servidor Linux y al puerto 27017.
3. En Power BI: Elige "ODBC" como origen de datos. Esto suele ser más rápido que mongosqld porque el driver gestiona mejor el "aplanamiento" de los documentos JSON a tablas.

Opción C: La vía "Profesional" (Arquitectura de alto rendimiento)

Si el volumen de datos es grande y Power BI se arrastra, la solución que se usa en producción no es conectar directamente el BI a la base de datos transaccional, sino:

- Exportar a Parquet/CSV: Usar un pequeño script en Python en tu Linux que extraiga los datos de Mongo y los guarde en archivos planos. Power BI lee estos archivos de forma instantánea.
- Réplica de lectura: Si decides seguir con conexión directa, asegúrate de que Power BI ataque a una "Secondary Node" de un Replica Set, para que las consultas pesadas de BI no bloqueen tu base de datos de producción.

Mi recomendación:

Lo que ves como "magia" en Windows es simplemente que el **Driver ODBC de MongoDB para Windows** hace un trabajo de traducción mucho más eficiente que el viejo binario mongosqld de Linux. **Prueba esto hoy**

1. Desactiva/ignora mongosqld.
 2. Instala el driver ODBC oficial en tu Windows.
 3. Conéctate directamente al puerto 27017 de tu contenedor Linux.
 4. Si Power BI te pide un formato, intenta usar el conector de MongoDB Atlas (aunque sea Community), ya que Microsoft y MongoDB han optimizado ese túnel recientemente
- ¿Te da algún error específico de "Timeout" o de "Esquema no encontrado" al intentar conectar directamente?

Jordi H.

Has dicho me dice que ha detectado community edition y que no

Gemini

¡Ahí está el muro! Te has topado con la **segmentación comercial** de MongoDB. Lo que sucede es que el conector nativo de Power BI (el que dice "MongoDB Atlas") está diseñado para enviar consultas a un motor llamado **Atlas SQL Interface**. Ese motor solo vive en la versión Cloud (Atlas) o en la versión Enterprise (On-Premise). Cuando el conector detecta que tu contenedor es **Community Edition**, "cierra la puerta" porque no encuentra ese motor SQL avanzado. Para saltarte este bloqueo y obtener el rendimiento que tienen tus compañeros sin pagar la versión Enterprise, tienes estas dos rutas re

Opción 1: El "Engaño" mediante ODBC (Lo que hacen tus compañeros)

Tus compañeros probablemente no usan el conector que dice "MongoDB", sino que usan el **Driver ODBC**. Al configurar un DSN en Windows, el driver hace de "traductor" y Power BI cree que está hablando con una base de datos SQL normal.

1. En tu Windows: Instala el MongoDB ODBC Driver.
2. Configura el DSN: En el Administrador de Orígenes de Datos ODBC de Windows, crea un nuevo "System DSN" usando ese driver. Apunta a la IP de tu Linux y al puerto 27017
3. En Power BI: No elijas "MongoDB". Elige "ODBC". Selecciona el DSN que creaste.

Opción 2: El "Puente" moderno (Sin mongosqld)

Si mongosqld te va lento es porque es un proceso único que intenta emular un servidor MySQL entero. Si quieres rendimiento de verdad en Linux Community, la tendencia actual es usar un **conector de flujo de datos**

- Vía Python/Script: Un pequeño script que haga un find() en Mongo y lo escupa a un archivo .parquet o .csv en una carpeta compartida. Power BI lee archivos Parquet a una velocidad absurda comparado con cualquier drive
- Vía Superset o Metabase: Si no estás casado con Power BI, estas herramientas tienen conectores nativos para Mongo Community que vuelan

¿Por qué a ellos les va "rápido" en Windows?

Es muy probable que su servidor de Windows tenga instalado el **MongoDB Connector for BI** como un servicio de Windows, que está mejor optimizado que el binario de Linux, o que estén usando una versión del driver ODBC que hace "sampling" (muestreo) de datos. Cuando el driver hace sampling, no lee toda la base de datos para entender el esquema, sino solo los primeros 100 documentos. Eso hace que la conexión inicial sea instantánea. **Mi consejo:** Olvida el conector nativo de Power BI que dice "MongoDB". Instala el **Driver** en tu PC, **configura el DSN apuntando a tu Linux y usa el conector ODBC**** en Power BI. ¿Quieres que te ayude a configurar los parámetros de ese DSN para que no te de error de "Schema"?

Jordi H.

Has dicho estoy probando todo el rato la opción 1.

Gemini

Si estás atascado en la **Opción 1 (ODBC)** y te sigue dando problemas de rendimiento o conexión con la versión Community, es muy probable que el cuello de botella esté en la forma en que el driver intenta "adivinar" la estructura de tus datos JSON para presentarlos como tablas SQL. Aquí tienes los 3 puntos clave para que la Opción 1 funcione de verdad con tu contenedor Lin

1. El "Sampling" es el secreto del rendimiento

Cuando configuras el DSN en Windows, hay una pestaña de **"Schema"** o **"Sampling"**.

- El problema: Si dejas que el driver escanee toda la colección de MongoDB para crear el esquema, Power BI se quedará "pensando" eternamente (lento)
- La solución: Configura el Sampling Limit a un número bajo (ej. 100 o 1000 documentos). Esto hace que el driver solo lea una pequeña muestra para saber qué columnas existen, acelerando la conexión inicial drásticamente

2. No uses el conector "MongoDB" en Power BI

Es confuso, pero una vez creado el DSN en tu Panel de Control de Windows, en Power BI debes elegir el "ODBC", no el que se llama "MongoDB".

- Al elegir ODBC, selecciona el nombre (DSN) que creaste.
- Si te pide credenciales, selecciona "Database" (o "Windows" si no tienes usuario en Mongo, aunque lo ideal es usar "Database" con el usuario/pass de tu contenedor)

3. Revisa los puertos del Contenedor

Si el driver ODBC en Windows no conecta al Linux, asegúrate de que:

- El contenedor se lanzó con -p 27017:27017.
- El firewall de Linux permite tráfico de entrada en el puerto 27017.
- En el archivo de configuración de Mongo dentro del contenedor (mongod.conf), el bindIp incluya 0.0.0.0 (o al menos la IP del bridge de Docker)

Si después de esto te sigue saliendo el mensaje de "Community Edition no soportado":

Significa que Power BI está intentando usar su "lógica interna" de Atlas. Para romper ese vínculo:

1. Borra el origen de datos actual en Power BI (Archivo -> Opciones y configuración -> Configuración de origen de datos)
2. Limpia todos los permisos de MongoDB.
3. Intenta de nuevo desde cero usando exclusivamente el conector genérico ODBC.

¿En qué paso exacto de la configuración del DSN te estás quedando? (¿Te falla el "Test Connection" en el panel de control de Windows o te falla al cargar los datos en Power BI?)

Jordi H.

Has dicho el error que sale es Data source error occurred. SQLSTATE: 01000 NativeError: 444 Error message: ODBC: ERROR [01000] Controlador devuelto no válido (o error en la devolución) SQL_DRIVER_ODBC_VER : 03.

Ese error (SQL_DRIVER_ODBC_VER: 03.80) es un clásico "error de saludo" entre Power BI y el driver. Básicamente, Power BI intenta hablar un "dialecto" de SQL moderno (ODBC 3.8) y el driver que tiene instalado o la configuración del DSN le responde algo que no entiende o que considera inconsistente. Como estás usando **MongoDB Community** en un contenedor Linux, lo más probable es que el driver esté intentando conectar, pero al no encontrar el motor de SQL nativo de MongoDB (que solo está en Atlas/Enterprise), se queda en un "limbo" técnico. Aquí tienes los 3 ajustes quirúrgicos para eliminar ese error.

1. Cambia el "Cursor Library" en la configuración del DSN

Este error suele ocurrir porque el driver intenta usar las librerías del gestor de Windows en lugar de las suyas propias.

1. Abre el Administrador de Orígenes de Datos ODBC (64 bits).
2. Configura tu DSN de MongoDB.
3. Busca una pestaña llamada "Advanced" o "Options".
4. Busca la opción "Driver MS Cursor Library" y asegúrate de que esté en "Off" o "No". Queremos que el driver de MongoDB maneje la comunicación, no Windows.

2. Fuerza el modo de compatibilidad (Workaround de SQL)

El error 01000 a veces indica que Power BI está intentando leer metadatos que el driver no sabe dar.

- En la configuración del DSN, busca una casilla que diga "Enable SQL Queries" o "Ignore Unknown Types".
- Si ves una opción de "Fetch Size", bájala a 100.

3. El problema del "Schema": El paso que te falta

A diferencia de MySQL, MongoDB no tiene tablas. El driver ODBC tiene que "inventárselas". Si no has configurado el **Schema Mode**, el driver devuelve un error al intentar conectar porque no sabe qué responderle a Power BI sobre la estructura de las tablas. En la configuración del DSN:

1. Ve a la pestaña Schema.
2. Selecciona "Local" (no Cloud/Atlas).
3. Asegúrate de que el Sampling Limit esté activo (pon 500 documentos).
4. Haz clic en un botón que suele decir "Refresh" o "Generate Schema". Si este botón falla aquí, Power BI nunca funcionará.

¿Por qué a tus compañeros sí les funciona?

Si ellos usan el conector de MySQL para conectar a MongoDB, es porque tienen instalado en sus servidores Windows un componente llamado **MongoDB Connector for BI (mongosqld.exe)**. Este componente actúa como un "traductor": por un lado habla con MongoDB y por el otro se hace pasar por un servidor MySQL. Por eso a ellos les funciona rápido; Power BI cree que está hablando con MySQL y usa el driver de MySQL, que es mucho más estable que el de MongoDB. **Mi recomendación** es:

Si el driver ODBC oficial te sigue escupiendo ese error de versión, la solución más estable para tu entorno Linux es **levantar un segundo contenedor** con la imagen oficial de `mongodb/mongodb-bi-connector`.

1. El contenedor A corre MongoDB.
2. El contenedor B corre BI Connector (apuntando al A).
3. Tú desde Windows conectas Power BI usando el conector de MySQL apuntando a la IP del contenedor B.

¿Te gustaría que te pase el comando docker run para levantar ese "traductor" de MySQL y acabar con el lío del ODBC?

Jordi H.

Has dicho estoy configurando el dbeaver con el conector JDBC para mongo. me he encallado en la JDBC url

Gemini

¡DBeaver es una excelente elección! Al usar el driver **JDBC**, te saltas muchos de los problemas de "capas" de Windows, pero la URL de conexión de MongoDB tiene su propia lógica. En DBeaver, para una instancia a **Community** en Docker, la estructura estándar es est

1. La estructura de la URL

Si estás configurando la URL manualmente en la pestaña "Main", debe verse así: `jdbc:mongodb://<IP_DE_TU_LINUX>:27017/<NOMBRE_DATABASE`

2. Casos comunes (y por qué podrías estar encallado)

Dependiendo de cómo tengas configurado tu contenedor, añade estos parámetros al final de la URL (después del ?)

- Si tienes autenticación (Usuario/Pass):

`jdbc:mongodb://tu_ip:27017/mi_db?authSource=admin`

(Es vital el `authSource=admin` si tu usuario se creó en la base de datos admin, que es lo normal).

- Si quieres que DBeaver "entienda" el esquema (Modo SQL):

DBeaver suele usar un driver que intenta mapear JSON a tablas. Si ves que no te aparecen tablas, asegúrate de que en la configuración del driver esté activado el "Document ID" o el "Flattening"

3. Cómo rellenar los campos en DBeaver (Paso a paso)

En lugar de escribir la URL a mano, te recomiendo usar los campos separados en la pestaña **"General"**:

1. Host: La IP de tu servidor Linux (ej. 192.168.1.50 o localhost si haces túnel SSH).
2. Port: 27017.
3. Database: El nombre de la base de datos que quieres consultar.
4. User / Password: Tus credenciales de MongoDB.
5. Pestaña "Driver Properties":
 - Busca la propiedad `authSource` y ponle `admin`.
 - Busca `ssl` y asegúrate de que esté en `false` (a menos que hayas configurado certificados en el contenedor).

4. Un "truco" de rendimiento para DBeaver

DBeaver a veces intenta leer miles de documentos para generar el esquema inicial y se queda congelado.

- Ve a Edit Connection -> Driver Properties.
- Busca una propiedad llamada `maxSharedConnectionSize` o similar (depende de la versión del driver JDBC) y limita el escaneo

¿Te está dando un error de **"Connection Refused"** o de **"Authentication Failed"**? Si me pasas el mensaje exacto de DBeaver, te digo qué parámetro te falta en la URL
