

Visualización de Datos y BI

Bases de Datos 1 (CC3088) - 2026

Visualización de Datos y BI

Semestre 01, 2026

El problema con las tablas

Una empresa tiene esta tabla de ventas del último trimestre.

id	fecha	producto	categoria	cantidad	precio	vendedor
1	2026-01-03	Laptop Pro	Electrónica	2	5000.00	Ana
2	2026-01-05	Mouse Óptico	Accesorios	15	120.00	Carlos
3	2026-01-07	Monitor 27"	Electrónica	3	1800.00	Ana
...						

La tabla tiene 4,800 filas.

¿Cuál categoría genera más ingresos? ¿Hay una tendencia de crecimiento?

¿Quién es el mejor vendedor?

Lo que el cerebro humano necesita

El ser humano no procesa tablas de miles de filas de forma natural.

Procesa patrones, tendencias y comparaciones visualmente.

Una gráfica de barras comunica en 2 segundos lo que una tabla de 4,800 filas no comunica en 5 minutos.

La visualización de datos transforma números en comprensión.

La distancia entre datos y decisiones

Las bases de datos almacenan hechos.

Los hechos por sí solos no generan decisiones.

Para decidir hace falta: contexto, comparación, tendencia, y claridad.

La visualización de datos crea ese puente.

Business Intelligence

Business Intelligence (BI) es el conjunto de procesos, tecnologías y herramientas que transforman datos crudos en información útil para tomar decisiones de negocio.

No es solo hacer gráficas.

Es un flujo completo que va desde los datos hasta la acción.

El flujo BI

Datos crudos



Almacenamiento (base de datos)



Transformación y limpieza



Análisis y consultas



Visualización



Decisión

Ya conocemos las primeras dos etapas.

En esta clase se cierran las últimas tres.

Componentes del ecosistema BI

| Componente | ¿Qué hace? | Ejemplos | | --- | --- | --- | | Base de datos / Data Warehouse | Almacena los datos | PostgreSQL, MySQL, Redshift | | ETL | Extrae, transforma y carga datos | dbt, Airbyte, scripts SQL | | Herramienta de BI | Visualiza y analiza | Metabase, Tableau, Power BI, Looker | | Dashboard | Muestra el estado del negocio en tiempo real | Creado en la herramienta de BI |

BI operacional vs analítico

| | BI Operacional | BI Analítico | | --- | --- | --- | | Frecuencia | Tiempo real / diario | Semanal / mensual | | Usuarios | Equipo de operaciones | Gerencia, analistas | | Preguntas | ¿Qué está pasando ahora? | ¿Por qué pasó? ¿Qué tendencia hay? | | Ejemplo | Pedidos pendientes de hoy | Crecimiento de ventas por trimestre |

En esta clase construiremos ambos tipos.

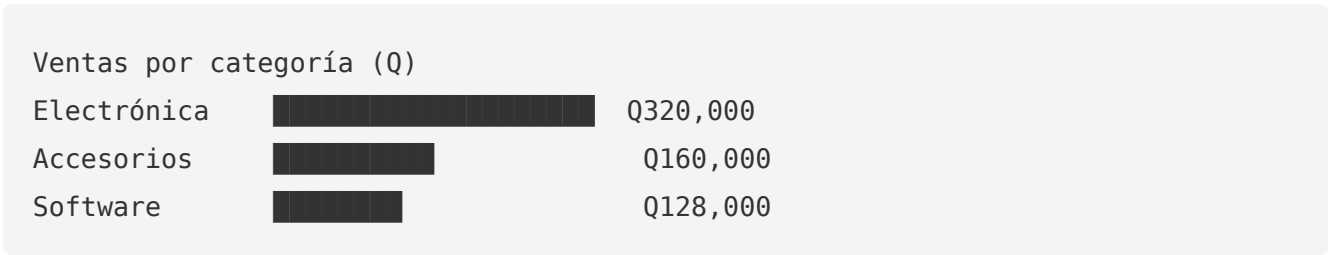
Tipos de visualización

Elegir el gráfico correcto es tan importante como tener los datos correctos.

Gráfico de barras

Cuándo usar: comparar valores entre categorías.

Eje X: categorías. Eje Y: valores numéricos.

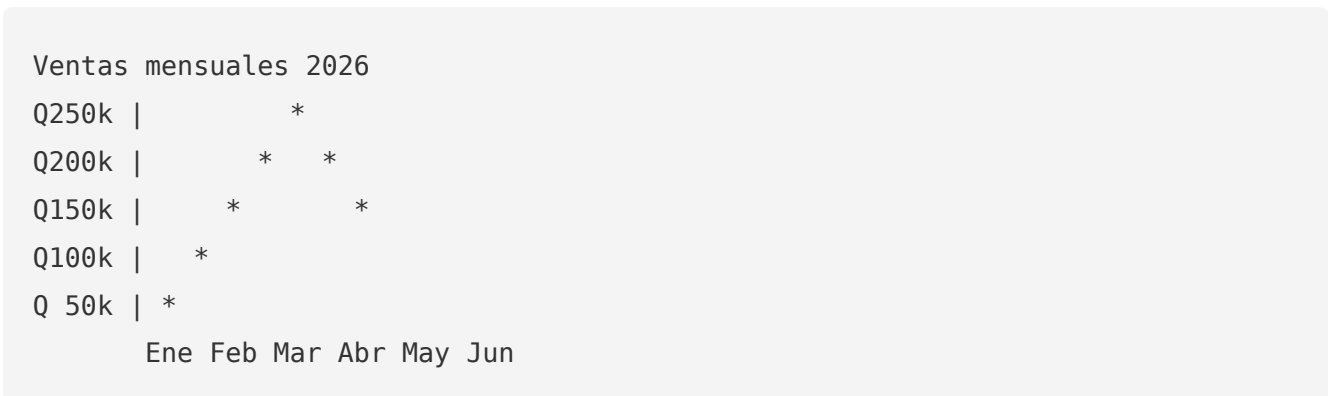


Preguntas que responde: ¿cuál categoría vende más? ¿cómo se comparan entre sí?

Gráfico de líneas

Cuándo usar: mostrar tendencias a lo largo del tiempo.

Eje X: tiempo. Eje Y: valor que cambia.



Preguntas que responde: ¿hay crecimiento? ¿dónde están los picos y caídas?

Gráfico de pastel / dona

Cuándo usar: mostrar proporciones de un total.

Solo cuando hay pocas categorías (máximo 5-6) y las diferencias son claras.

Preguntas que responde: ¿qué fracción del total representa cada parte?

Tabla de datos

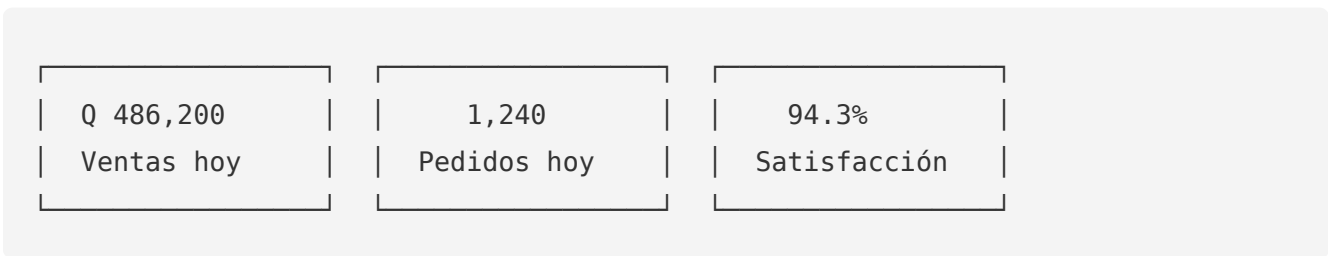
Cuándo usar: cuando se necesita el detalle exacto, no solo la tendencia.

No reemplaza a las gráficas — las complementa.

Útil para: listas de pedidos, ranking de vendedores, inventario crítico.

Número / KPI

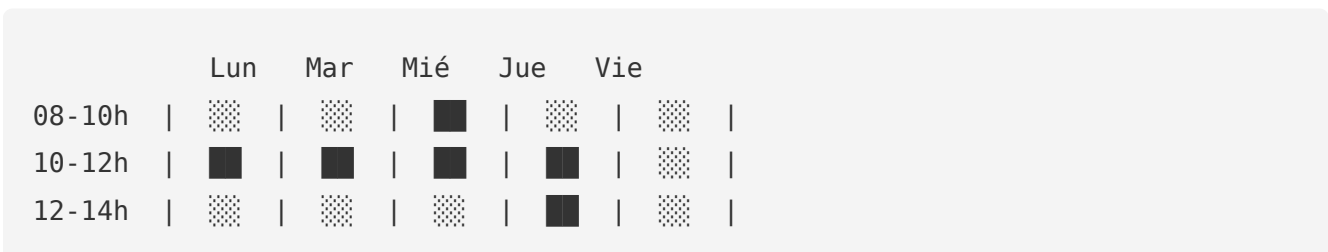
Cuándo usar: mostrar una métrica clave de un solo vistazo.



Los KPIs son el primer elemento que ve cualquier persona al abrir un dashboard.

Mapa de calor

Cuándo usar: comparar dos dimensiones simultáneamente.



Preguntas que responde: ¿en qué día y hora hay más pedidos?

Guía para elegir el gráfico correcto

| Pregunta | Gráfico | | --- | --- | | ¿Cuánto? / ¿Cuántos? (una métrica) | KPI / número | | ¿Cuál categoría es mayor? | Barras | | ¿Cómo cambia en el tiempo? | Líneas | | ¿Qué proporción del total? | Pastel / dona | | ¿Cómo se distribuye? | Histograma | | ¿Hay correlación entre dos variables? | Dispersión (scatter) | |

¿Dónde está? | Mapa | | ¿Cuál es el detalle exacto? | Tabla |

Metabase

Metabase es una herramienta de Business Intelligence open source.

Permite conectar directamente a una base de datos y crear visualizaciones sin escribir código.

También soporta SQL nativo para usuarios avanzados.

¿Por qué Metabase?

- **Open source** — gratuito para uso propio.
- **Conecta a PostgreSQL, MySQL, SQLite** y más de 20 motores.
- **No requiere programación** para consultas básicas — interfaz visual.
- **Soporta SQL** para consultas avanzadas.
- **Dashboards interactivos** con filtros y parámetros.
- **Se despliega con Docker** en menos de 5 minutos.

Instalación con Docker

```
docker run -d \  
  -p 3000:3000 \  
  --name metabase \  
  metabase/metabase
```

Abrir en el navegador: `http://localhost:3000`

Seguir el wizard de configuración inicial.

Conectar a la base de datos

En Metabase: Configuración → Bases de datos → Agregar base de datos.

| Campo | PostgreSQL | MySQL | | --- | --- | --- | | Tipo | PostgreSQL | MySQL | | Host | localhost | localhost | | Puerto | 5432 | 3306 | | Base de datos | nombre_db | nombre_db | | Usuario | usuario | usuario | | Contraseña | clave | clave |

Metabase se conecta con un rol de solo lectura — no necesita permisos de escritura.

Preguntas en Metabase

En Metabase, cada visualización parte de una **Pregunta** (Question).

Hay dos formas de crear una pregunta: editor visual y SQL nativo.

Editor visual — sin SQL

Metabase traduce las selecciones del usuario en SQL automáticamente.

```
Tabla: ventas
Métricas: Suma de total
Agrupación: Por categoría
Filtro: fecha > hace 30 días
Ordenar: de mayor a menor
Límite: 10
```

Genera internamente:

```
SELECT categoria, SUM(total) AS suma_total
FROM ventas
WHERE fecha > NOW() - INTERVAL '30 days'
GROUP BY categoria
ORDER BY suma_total DESC
LIMIT 10;
```

Resultado: seleccionar el tipo de gráfico

Con el resultado en mano, Metabase sugiere el tipo de gráfico más apropiado.

Se puede cambiar entre: barras, líneas, pastel, tabla, número, mapa.

El eje X y Y se configuran con un clic.

Editor SQL — para consultas avanzadas

Para consultas que el editor visual no puede expresar: JOINS, subconsultas, funciones de ventana.

```
SELECT
  v.vendedor,
  COUNT(*)           AS total_pedidos,
  SUM(v.total)       AS ingresos_totales,
  AVG(v.total)       AS ticket_promedio,
  MAX(v.total)       AS venta_maxima
FROM ventas v
WHERE v.fecha >= DATE_TRUNC('month', NOW())
GROUP BY v.vendedor
ORDER BY ingresos_totales DESC;
```

El resultado se visualiza igual que en el editor visual.

Parámetros en SQL

Metabase permite agregar filtros interactivos a las consultas SQL con doble corchete:

```
SELECT categoria, SUM(total) AS ingresos
FROM ventas
WHERE fecha BETWEEN {{fecha_inicio}} AND {{fecha_fin}}
[[AND vendedor = {{vendedor}}]]
GROUP BY categoria
```

```
ORDER BY ingresos DESC;
```

{{fecha_inicio}} → Metabase genera un selector de fecha en el dashboard.

[[AND vendedor = {{vendedor}}]] → el filtro es opcional — si no se llena, se ignora.

Dashboards

Un dashboard es una colección de preguntas organizadas en una sola pantalla.

Muestra el estado del negocio de un vistazo.

Principios de un buen dashboard

- **Una sola historia:** el dashboard debe responder una pregunta de negocio específica.
- **KPIs primero:** los números más importantes arriba, al centro.
- **De lo general a lo específico:** resumen arriba, detalle abajo.
- **Máximo 6-8 visualizaciones:** más es confusión, no información.
- **Filtros globales:** que afecten todas las tarjetas al mismo tiempo.

Estructura típica de un dashboard de ventas

Ventas del mes	Pedidos del mes	Ticket promedio
Ventas por día (líneas)		
Top vendedores (tabla)	Por categoría (barras)	

Filtros globales en el dashboard

Se puede agregar un filtro de fecha o categoría que afecte a todas las tarjetas simultáneamente.

El usuario selecciona "enero 2026" y todos los gráficos se actualizan.

No hay que crear un dashboard separado para cada período.

Actualización automática

Los dashboards de Metabase pueden configurarse para actualizarse cada 1, 5, 10 o 60 minutos.

Los datos siempre vienen directamente de la base de datos — no hay caché desactualizado.

Caso de uso 1: Dashboard de ventas

```
-- KPI: ventas del mes actual
SELECT SUM(total) AS ventas_mes
FROM ventas
WHERE DATE_TRUNC('month', fecha) = DATE_TRUNC('month', NOW());

-- Tendencia: ventas por día del mes actual
SELECT DATE(fecha) AS dia, SUM(total) AS total_dia
FROM ventas
WHERE DATE_TRUNC('month', fecha) = DATE_TRUNC('month', NOW())
GROUP BY DATE(fecha)
ORDER BY dia;

-- Top 5 productos del mes
SELECT producto, SUM(cantidad) AS unidades, SUM(total) AS ingresos
FROM ventas
WHERE DATE_TRUNC('month', fecha) = DATE_TRUNC('month', NOW())
```

```
GROUP BY producto
ORDER BY ingresos DESC
LIMIT 5;
```

Cada consulta se convierte en una tarjeta del dashboard.

El resultado

| Tarjeta | Tipo | Insight | | --- | --- | --- | | Ventas del mes | KPI | Q486,200 | | Ventas por día | Líneas | Tendencia de crecimiento los viernes | | Top 5 productos | Barras horizontales | Laptop Pro lidera por amplio margen |

Caso de uso 2: Dashboard operacional

Un gerente de operaciones necesita saber el estado de los pedidos en tiempo real.

```
-- Pedidos pendientes por estado
SELECT estado, COUNT(*) AS cantidad
FROM pedidos
WHERE estado IN ('pendiente', 'en_proceso', 'listo_para_envio')
GROUP BY estado;

-- Tiempo promedio de preparación por sucursal
SELECT sucursal,
       AVG(EXTRACT(EPOCH FROM (fecha_entrega - fecha_pedido)) / 60) AS minutos_promedio
FROM pedidos
WHERE fecha_pedido::date = CURRENT_DATE
      AND fecha_entrega IS NOT NULL
GROUP BY sucursal
ORDER BY minutos_promedio;

-- Pedidos atrasados (más de 30 minutos sin entregar)
SELECT id, cliente, sucursal, fecha_pedido,
       EXTRACT(EPOCH FROM (NOW() - fecha_pedido)) / 60 AS minutos_espera
```

```
FROM pedidos
WHERE estado != 'entregado'
      AND fecha_pedido < NOW() - INTERVAL '30 minutes'
ORDER BY minutos_espera DESC;
```

Por qué funciona en Metabase

La consulta de pedidos atrasados se actualiza cada 1 minuto.

El equipo de operaciones ve en tiempo real qué pedidos necesitan atención.

No hay que escribir un reporte ni esperar al final del día.

Caso de uso 3: Análisis de clientes

```
-- Segmentación RFM básica: Recencia, Frecuencia, Monto
SELECT
  cliente_id,
  MAX(fecha)           AS ultima_compra,
  COUNT(*)             AS num_compras,
  SUM(total)           AS total_gastado,
  NOW()::date - MAX(fecha)::date AS dias_desde_ultima_compra
FROM ventas
GROUP BY cliente_id
ORDER BY total_gastado DESC;

-- Clientes que no han comprado en 90 días
SELECT c.nombre, c.email,
       MAX(v.fecha) AS ultima_compra,
       NOW()::date - MAX(v.fecha)::date AS dias_inactivo
FROM clientes c
JOIN ventas v ON v.cliente_id = c.id
GROUP BY c.id, c.nombre, c.email
HAVING NOW()::date - MAX(v.fecha)::date > 90
ORDER BY dias_inactivo DESC;
```

La tabla de clientes inactivos en el dashboard permite al equipo de marketing lanzar campañas de reactivación.

Herramientas de BI: panorama general

| Herramienta | Tipo | Características | Para quién | | --- | --- | --- | --- | | Metabase | Open source | Fácil, SQL + visual, Docker | Equipos técnicos, startups | | Tableau | Comercial | Muy visual, drag & drop, costoso | Analistas de negocio | | Power BI | Comercial (Microsoft) | Integración con Excel/Azure | Empresas con ecosistema Microsoft | | Looker | Comercial (Google) | LookML, data modeling avanzado | Empresas grandes | | Grafana | Open source | Métricas de sistemas, tiempo real | DevOps, infraestructura | | Apache Superset | Open source | Similar a Metabase, más configurable | Equipos con recursos técnicos |

¿Por qué Metabase en este curso?

- Gratuito y open source.
- Se conecta directamente a las bases de datos que ya conocen.
- Soporta SQL avanzado — todo lo aprendido en el curso es usable.
- Desplegable con Docker — sin infraestructura compleja.
- Estándar real en el mercado — muchas empresas en Guatemala lo usan.

Buenas prácticas de visualización

Usar el gráfico correcto

Un gráfico de pastel con 12 categorías es ilegible.

Un KPI que muestra una suma sin contexto es inútil.

La visualización debe responder una pregunta específica.

Agregar contexto

Un número solo no comunica nada.

Q486,200 → Q486,200 (+12% vs mes anterior)

El cambio porcentual, la meta, el promedio histórico — eso es contexto.

Mantener los dashboards simples

Cada elemento en el dashboard debe tener una razón para estar ahí.

Si hay que explicar qué muestra un gráfico, el gráfico está mal diseñado.

Un usuario nuevo debe entender el dashboard en menos de 30 segundos.

Los datos deben ser confiables

Un dashboard construido sobre datos inconsistentes o duplicados genera decisiones incorrectas.

La calidad de los datos (normalización, constraints, integridad referencial) que se aprendió en este curso es el fundamento de cualquier sistema BI.

Garbage in, garbage out.

¿Qué puede salir mal?

- **Conectar Metabase con un usuario con permisos de escritura** → Metabase solo necesita SELECT; si alguien borra datos desde el dashboard es un problema de permisos

- **Dashboards sin filtro de fecha** → una consulta sin WHERE fecha > ... puede devolver millones de filas y colapsar la base de datos
- **KPIs sin denominador** → "1,200 pedidos hoy" sin comparación no dice nada; siempre agregar contexto (meta, período anterior, promedio)
- **Gráficos de pastel con más de 5 categorías** → imposibles de leer; usar barras
- **Actualización cada 1 minuto en consultas pesadas** → puede saturar la base de datos; ajustar la frecuencia según el costo de la consulta
- **Un dashboard para todo** → un dashboard que intenta responder 20 preguntas no responde ninguna bien; crear dashboards específicos por audiencia
- **No usar índices en las columnas que filtran los dashboards** → si el dashboard filtra siempre por fecha, esa columna debe tener índice
- **Confundir correlación con causalidad** → un gráfico puede mostrar que dos métricas suben juntas sin que una cause la otra; las visualizaciones muestran patrones, no explicaciones