

# SESIÓN 2: GOBERNANZA Y MODELADO DE DATOS

El modelado de datos es una técnica fundamental que permite representar cómo se organizan, almacenan y relacionan los datos dentro de un sistema. Su objetivo es proporcionar una visión estructurada y lógica de los datos antes de que estos sean implementados físicamente.

**R** por Kibernum Capacitación S.A.

# ¿Qué es el modelado de datos?

El modelado de datos es una técnica fundamental que permite representar cómo se organizan, almacenan y relacionan los datos dentro de un sistema. Su objetivo es proporcionar una visión estructurada y lógica de los datos antes de que estos sean implementados físicamente.

Esta representación se realiza a través de diagramas o esquemas que describen:

- Entidades: objetos o conceptos del negocio (ej. Clientes, Productos, Pedidos).
- Atributos: propiedades de cada entidad (ej. nombre, precio, fecha).
- Relaciones: cómo se conectan las entidades entre sí.
- Reglas de negocio: restricciones como "un cliente puede tener varios pedidos".

# Niveles de modelado

El modelado de datos se realiza en varios niveles, que van desde lo más abstracto hasta lo más técnico. Cada nivel cumple un propósito específico:

| Etapas del Modelo | Descripción  |
|-------------------|--|
| Modelo Conceptual | Representación abstracta de los datos, centrada en los requerimientos del negocio. Suele crearse en colaboración con usuarios no técnicos.           |
| Modelo Lógico     | Traducción técnica del modelo conceptual. Define entidades, relaciones, claves primarias y foráneas, sin considerar un gestor específico.            |
| Modelo Físico     | Implementación directa en un sistema de gestión de base de datos (SGBD), con detalles técnicos como tipos de datos, índices, tablas y restricciones. |

# Diseño de datos

El diseño de datos es la etapa siguiente al modelado. Su función es traducir los modelos (conceptual, lógico y físico) en estructuras reales dentro de un sistema de almacenamiento como MySQL, PostgreSQL o Snowflake. Si el modelo lógico define que habrá una tabla de clientes con ciertos campos...

El modelado es como crear los planos de un edificio; el diseño es la construcción real con materiales concretos.

# ¿Por qué es importante el modelado y diseño?

Tanto el modelado como el diseño permiten asegurar varios principios esenciales en la gestión de datos:

| Beneficio                                 | Impacto   |
|---|---|
| Integridad de los datos                   | Garantiza que los datos sean correctos, coherentes y respeten las reglas del negocio.                     |
| Evitar duplicidad                         | Mejora el rendimiento y evita errores por redundancia innecesaria.  |
| Facilitar el acceso y el análisis         | Permite que las consultas y reportes sean más simples, rápidos y precisos.                                |
| Documentar el conocimiento organizacional | Hace explícita la estructura y lógica de los datos, facilitando el traspaso de información entre equipos. |

El modelado y diseño de datos son fases críticas en cualquier proyecto de ingeniería de datos. Permiten construir sistemas coherentes, escalables y alineados con los objetivos del negocio.

# Modelado de datos: del concepto a la implementación

El modelado de datos es una técnica esencial en el diseño de sistemas de información. Permite estructurar, representar y comprender cómo se organizan los datos dentro de una organización antes de ser implementados en un sistema de almacenamiento.

El modelado se realiza en tres niveles progresivos:

1. Modelado conceptual
2. Modelado lógico
3. Modelado físico

A continuación, profundizaremos un poco en cada uno con ejemplos, representaciones y sus beneficios.

# Modelado Conceptual

El modelado conceptual es la primera etapa en el proceso de diseño. Tiene un enfoque abstracto y está orientado al negocio. Su objetivo es identificar qué datos existen, cómo se relacionan y qué reglas de negocio los gobiernan, sin preocuparse por detalles técnicos.

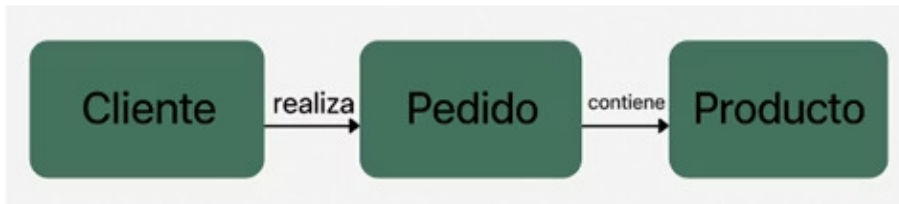
Características:

- Orientado al negocio.
- Utiliza lenguaje común para usuarios técnicos y no técnicos.
- Representa entidades, relaciones generales y reglas de negocio.

Ejemplo: Veamos un ejemplo sencillo de un sistema de ventas

- Entidades: Cliente, Pedido, Producto.
- Relaciones:
  - Un cliente realiza muchos pedidos.
  - Un pedido contiene varios productos.

**Representación típica:**



# Modelado Lógico

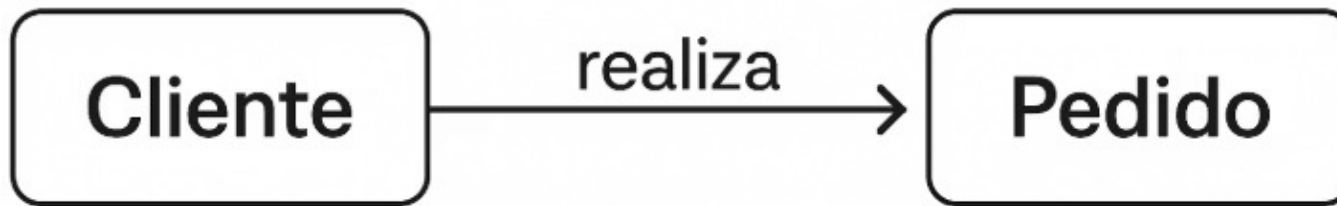
El modelado lógico toma el modelo conceptual y lo traduce en un diseño estructurado, con entidades detalladas, atributos, claves primarias y foráneas, pero sin comprometerse aún con una tecnología específica.

Características:

- Traduce el modelo conceptual a tablas.
- Define atributos, claves primarias y foráneas.
- Mantiene independencia del SGBD.

Ejemplo (Cliente-Pedido):

**Modelo Conceptual:**



**Modelo Lógico:**

- Tabla Cliente:
  - ID\_Cliente (PK)
  - Nombre
  - Dirección
- Tabla Pedido:
  - ID\_Pedido (PK)
  - Fecha
  - Monto
  - ID\_Cliente (FK → Cliente)



# Visualización del Modelo Lógico

| Clientes        |        |
|-----------------|--------|
| ID_Cliente (PK) | Nombre |
| 1               | Juan   |
| 2               | María  |

| Pedidos        |           |
|----------------|-----------|
| ID_Pedido (PK) | Fecha     |
| 101            | 1/11/2024 |
| 102            | 2/11/2024 |
| 103            | 3/11/2024 |

# Modelado Físico

El modelado físico es la etapa final, donde el modelo lógico se transforma en un diseño implementable en un sistema de base de datos (SGBD) o DBMS por Database Management System (sus siglas en inglés).

Características:

- Implementación real en un SGBD.
- Define tipos de datos, índices, claves, collation, storage engine, etc.
- **Ejemplo en SQL (MySQL):**

```
CREATE TABLE Clientes (  
  ID_Cliente INT PRIMARY KEY,  
  Nombre VARCHAR(50) NOT NULL,  
  Telefono VARCHAR(15)  
);
```

```
CREATE TABLE Pedidos (  
  ID_Pedido INT PRIMARY KEY,  
  Fecha DATE NOT NULL,  
  Total DECIMAL(10, 2),  
  ID_Cliente INT,  
  FOREIGN KEY (ID_Cliente) REFERENCES Clientes(ID_Cliente)  
);
```

# Relación entre los tres niveles

| Nivel del modelo | Audiencia principal      | Enfoque                                | Dependencia tecnológica |
|------------------|--------------------------|--|-------------------------|
| Conceptual       | Negocio, analistas       | Qué datos existen y cómo se relacionan | No                      |
| Lógico           | Diseñadores de BD, TI    | Cómo se estructuran los datos          | No                      |
| Físico           | DBA, ingenieros de datos | Cómo se implementan los datos          | Sí (depende del SGBD)   |

# Normalización de una base de datos

La normalización es un proceso en el diseño de bases de datos relacionales que busca:

- Eliminar la redundancia de datos (evitando la repetición innecesaria de información).
- Minimizar las anomalías en la inserción, actualización y eliminación de datos.
- Garantizar una estructura eficiente para almacenar y recuperar datos.

Se logra mediante la aplicación de formas normales, que son reglas para organizar los datos. Las formas normales más comunes son:

1. Primera Forma Normal (1FN): Los datos deben estar en formato tabular, sin grupos repetidos ni valores múltiples en una sola celda.
2. Segunda Forma Normal (2FN): Todos los atributos que no son clave deben depender completamente de la clave primaria.
3. Tercera Forma Normal (3FN): Ningún atributo que no sea clave debe depender de otro atributo que no sea clave.

# Formas normales (resumen)

|     | Requisitos:  |
|-----|--|
| 1FN | Todos los atributos deben ser atómicos                         |
| 2FN | Cumple 1FN + sin dependencias parciales (en claves compuestas) |
| 3FN | Cumple 2FN + sin dependencias transitivas                      |

# Ejemplo aplicado de normalización

Tabla inicial (no normalizada):

| ID_Cliente | Nombre | ID_Pedido | Fecha | Ciudad     | Código_Postal |
|------------|--------|-----------|-------|------------|---------------|
| 1          | Juan   | 101       | 45597 | Santiago   | 12345         |
| 2          | María  | 102       | 45598 | Valparaíso | 54321         |
| 1          | Juan   | 103       | 45599 | Santiago   | 12345         |

Problemas detectados:

1FN: Cumple con la Primera Forma Normal porque los datos son tabulares.

2FN: Cumple con la Segunda Forma Normal al dividir en las tablas Clientes y Pedidos:

Clientes contiene ID\_Cliente, Nombre, Ciudad y Código\_Postal.

Pedidos contiene ID\_Pedido, Fecha y la clave foránea ID\_Cliente.

Pero hay un problema de dependencia transitiva:

En la tabla Clientes, Código\_Postal depende de Ciudad, no directamente de ID\_Cliente.

# Solución: aplicar normalización hasta 3FN

Primera Forma Normal (1FN): La tabla ya cumple con esta regla, ya que no hay grupos repetidos ni valores múltiples en una celda.

Segunda Forma Normal (2FN): Dividimos en Clientes y Pedidos, eliminando dependencias parciales:

Clientes (2FN):

| ID_Cliente | Nombre | Ciudad     | Código_Postal |
|------------|--------|------------|---------------|
| 1          | Juan   | Santiago   | 12345         |
| 2          | María  | Valparaíso | 54321         |

Pedidos (2FN):

| ID_Pedido | Fecha     | ID_Cliente |
|-----------|-----------|------------|
| 101       | 1/11/2024 | 1          |
| 102       | 2/11/2024 | 2          |
| 103       | 3/11/2024 | 1          |

# Tercera Forma Normal (3FN)

Problema Persistente: En la tabla Clientes, Código\_Postal depende de Ciudad, no directamente de ID\_Cliente. Esto crea una dependencia transitiva.

Tercera Forma Normal (3FN): Creamos una nueva tabla para separar la información de la ciudad y el código postal.

Clientes (3FN):

| ID_Cliente | Nombre | Ciudad     |
|------------|--------|------------|
| 1          | Juan   | Santiago   |
| 2          | María  | Valparaíso |

Ciudades:

| Ciudad     | Código_Postal |
|------------|---------------|
| Santiago   | 12345         |
| Valparaíso | 54321         |

Beneficios de normalizar:

- Mejora el rendimiento del sistema.
- Facilita la escalabilidad.
- Reduce duplicidad y errores.
- Facilita el mantenimiento y el control de versiones.



# Metas y principios del modelado de datos

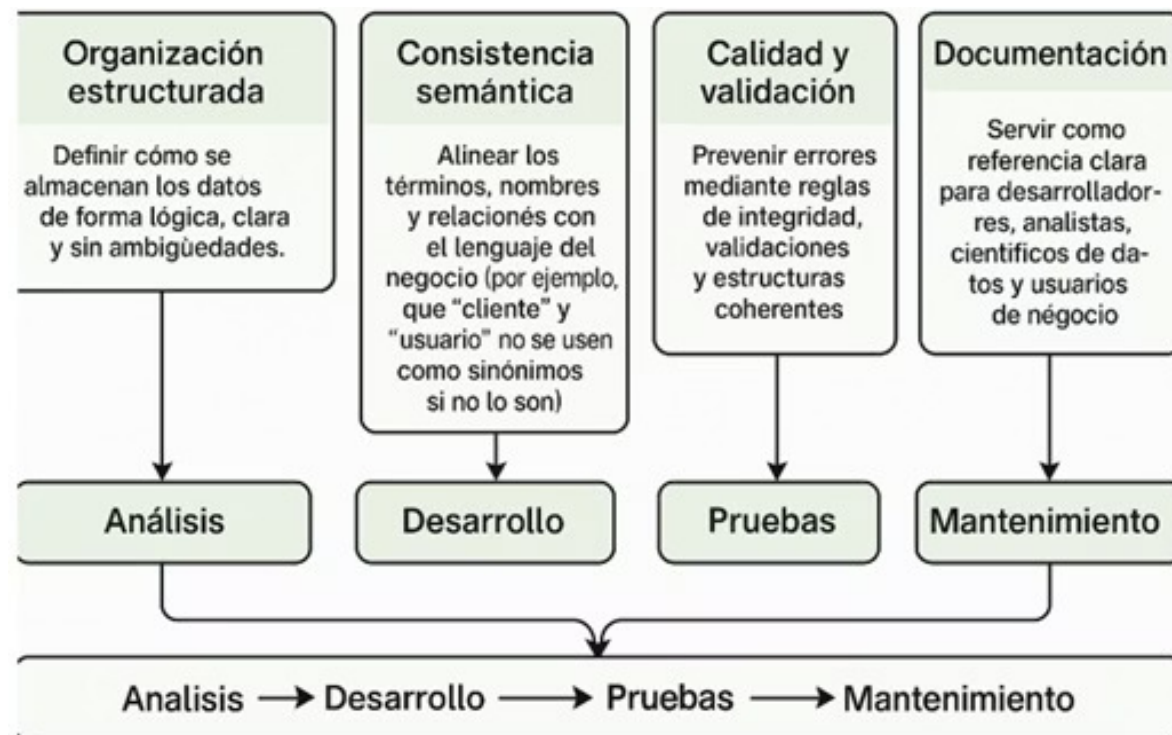
El modelado de datos no es solo un ejercicio técnico, sino una estrategia fundamental para alinear la estructura de los datos con los objetivos del negocio. Cuando una organización define correctamente sus modelos de datos, está creando una base sólida para el análisis, la automatización, la calidad de la información y la toma de decisiones.

A continuación, exploramos las metas principales del modelado de datos, y cómo cada una contribuye al éxito de los proyectos de datos en entornos reales.

# Metas del modelado de datos

| Meta                       | Descripción   |
|----------------------------|---|
| Organización estructurada  | Definir cómo se almacenan los datos de forma lógica, clara y sin ambigüedades.  |
| Consistencia semántica     | Alinear los términos, nombres y relaciones con el lenguaje del negocio (por ejemplo, que "cliente" y "usuario" no se usen como sinónimos si no lo son). |
| Calidad y validación       | Prevenir errores mediante reglas de integridad, validaciones y estructuras coherentes.  |
| Facilitar el mantenimiento | Permitir que los modelos puedan ser modificados o extendidos sin grandes impactos.  |
| Documentación              | Servir como referencia clara para desarrolladores, analistas, científicos de datos y usuarios de negocio.   |

Observa estas metas representadas en un flujo visual, el cual nos ayuda a comprender cómo se relaciona cada una con las distintas etapas del proceso.



# ¿Por qué son importantes las metas del modelado de datos?

El modelado de datos no es solo una herramienta técnica; es una base estratégica que permite que los datos se comuniquen de forma coherente entre distintos sistemas, equipos y procesos.

Veamos un caso práctico que ilustra por qué esto es tan relevante:

## Caso práctico: problemas de fragmentación por falta de modelado

Una empresa de logística utiliza tres sistemas diferentes, cada uno con su propia forma de referirse al mismo concepto:

- Sistema de ventas: usa el término "Cliente"
- Sistema de envíos: se refiere al mismo registro como "Receptor"
- Sistema contable: lo denomina "Persona jurídica"

A simple vista parece una diferencia menor de nombres... pero tiene grandes implicancias.

# Problemas generados por la falta de un modelo unificado

- Duplicidad de datos: El mismo cliente aparece varias veces con nombres distintos.
- Errores operativos: Las facturas no coinciden con los destinatarios reales.
- Análisis fragmentado: Es imposible hacer reportes cruzados entre áreas porque no hay correspondencia entre los registros.

Esto genera pérdida de tiempo, aumento de costos operativos y toma de decisiones basadas en datos incompletos o erróneos.

# Solución a través del modelado de datos

Un modelo de datos bien diseñado y documentado resuelve este tipo de situaciones:

| Beneficio    | Impacto directo en la organización   |
|--------------|--|
| Consistencia | Todos los sistemas adoptan el mismo término estandarizado (ej. "Cliente").       |
| Autonomía    | Los equipos pueden comprender y reutilizar los datos sin depender de un experto. |
| Eficiencia   | Se reduce el tiempo dedicado a limpieza de datos y se minimizan los errores.     |

Esto no es solo teoría: empresas como FedEx, DHL y otras grandes operadoras logísticas aplican modelos de datos estandarizados para integrar procesos de ventas, despacho, CRM y facturación.



# Principios clave del modelado

A la hora de diseñar modelos de datos, existen principios rectores que ayudan a tomar decisiones de diseño sostenibles, escalables y comprensibles para todo el equipo.

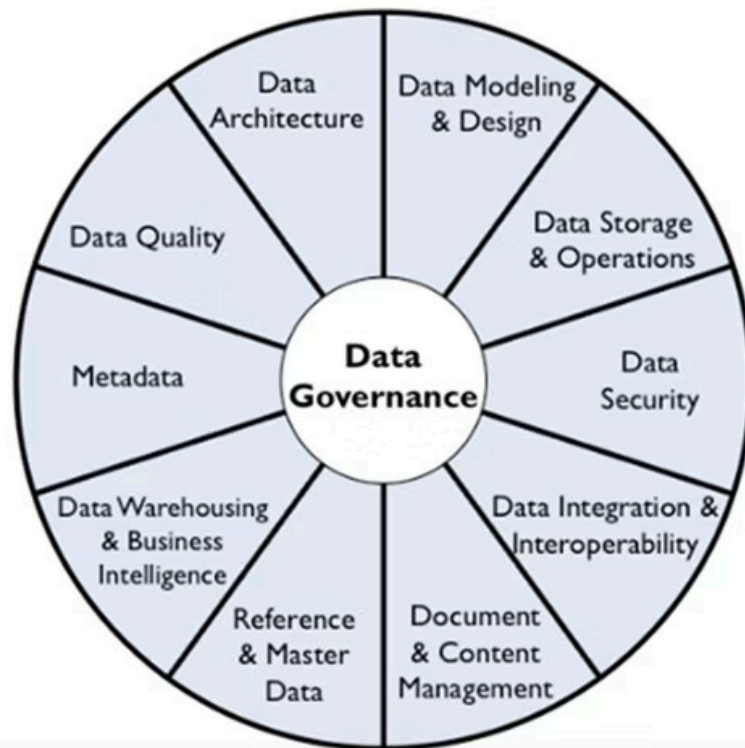
- Simplicidad antes que complejidad innecesaria: Un modelo de datos no debe ser más complicado de lo necesario. Si puedes representar algo en una tabla clara, evita estructuras rebuscadas.
- Normalización donde sea útil: La normalización (como vimos antes) mejora la integridad y evita duplicaciones. Sin embargo, si la desnormalización mejora el rendimiento en consultas frecuentes, puede aplicarse conscientemente.
- Reutilización de entidades comunes: Si varios sistemas manejan "Clientes", usa una definición común. Esto facilita la interoperabilidad y el mantenimiento del modelo.
- Visibilidad para los stakeholders del negocio: El modelo debe estar documentado y disponible. No sirve de nada un modelo "perfecto" si nadie lo entiende ni lo puede revisar.

# Gobernanza del modelo de datos

En cualquier organización basada en datos, no basta con diseñar modelos bien estructurados. También es necesario garantizar su control, mantenimiento y evolución en el tiempo. Aquí es donde entra en juego la gobernanza del modelo de datos, un conjunto de prácticas, roles y procesos que aseguran que los modelos sean:

- Coherentes: siguen una lógica común en toda la organización.
- Documentados: cuentan con descripciones claras, diagramas y definiciones.
- Versionados: es posible rastrear cambios a lo largo del tiempo.
- Reutilizables: se diseñan pensando en su uso en múltiples sistemas o contextos.
- Validados: tanto por áreas técnicas como por áreas de negocio.

La gobernanza no se trata solo de normas: es una capa de organización y sostenibilidad sobre los activos de datos.



**Crédito Imagen 6: Imágenes de la rueda DAMA-DMBOK v2**

Observa la rueda DAMA-DMBOK que vimos en la sesión anterior, que muestra en el centro, profundizaremos un poco en lo que representa.

# ¿Qué representa la rueda DAMA-DMBOK?

Esta figura, conocida como la rueda de DAMA-DMBOK v2, es uno de los marcos de referencia más utilizados a nivel internacional para la gestión de datos. Nos muestra que la gobernanza no es un área aislada, sino una función transversal que coordina, regula y da coherencia a todas las demás disciplinas de la gestión de datos.

Cada sección de la rueda representa un área crítica:

| Área                      | Rol clave dentro del ecosistema de datos                           |
|---------------------------|--|
| Data Architecture         | Define las estructuras técnicas y lógicas para soportar los datos. |
| Data Modeling & Design    | Especifica cómo se organizan y conectan los datos.                 |
| Data Storage & Operations | Administra cómo se almacenan, respaldan y mantienen los datos.     |
| Data Security             | Protege los datos contra accesos no autorizados o pérdidas.        |



# Áreas críticas de la gestión de datos (continuación)

| Área                                     | Rol clave dentro del ecosistema de datos  |
|--|---|
| Data Integration & Interoperability      | Permite que sistemas distintos compartan y usen datos de forma fluida.                      |
| Document & Content Management            | Gestiona archivos, documentos digitales y contenidos asociados a los datos.                 |
| Reference & Master Data                  | Controla los datos clave compartidos por múltiples sistemas (como "Clientes", "Productos"). |
| Data Warehousing & Business Intelligence | Facilita el análisis de datos históricos y la toma de decisiones basada en datos.           |
| Metadata                                 | Asegura que los datos sean comprensibles mediante definiciones y descripciones.             |
| Data Quality                             | Evalúa y mantiene la precisión, completitud y confiabilidad de los datos.                   |

Todos estos componentes dependen de una gobernanza efectiva para funcionar de forma integrada y sostenible. Si falla la gobernanza, se rompen los estándares, los roles se diluyen y los modelos pierden validez.

# ¿Por qué es clave gobernar los modelos?

Sin una estrategia de gobernanza, los modelos de datos pueden volverse inconsistentes, obsoletos o contradictorios entre áreas. Esto trae consecuencias como:

- Retrabajo entre equipos (cada uno "reinventa" el modelo).
- Fallos de interpretación (los datos significan cosas distintas según el área).
- Dificultades para auditar o mantener sistemas de análisis o reporting.
- La falta de gobernanza no solo afecta a los datos, afecta a la toma de decisiones.

# Ejemplos de buenas prácticas de gobernanza

Las organizaciones maduras en gestión de datos suelen aplicar prácticas como:

| Buena práctica                      | Descripción   |
|-------------------------------------|---|
| Repositorio centralizado de modelos | Una fuente única de verdad para acceder, consultar y modificar modelos.       |
| Control de versiones                | Cada cambio queda registrado, documentado y puede revertirse si es necesario. |
| Validaciones colaborativas          | Los modelos se revisan en conjunto por equipos técnicos y de negocio.         |
| Estándares de nomenclatura          | Las entidades y atributos siguen reglas claras y consistentes.                |
| Documentación accesible             | Diagramas, glosarios y descripciones disponibles para todos los usuarios.     |

## Roles clave en la gobernanza del modelo

| Rol                   | Responsabilidad principal   |
|-----------------------|---|
| Data Architect        | Diseña la arquitectura de datos y valida modelos de alto nivel.     |
| Data Modeler          | Implementa y documenta los modelos conceptuales, lógicos y físicos. |
| Data Steward          | Asegura la calidad del modelo desde el punto de vista del negocio.  |
| Equipo de BI/Análisis | Consume los modelos para generar reportes, dashboards e insights.   |

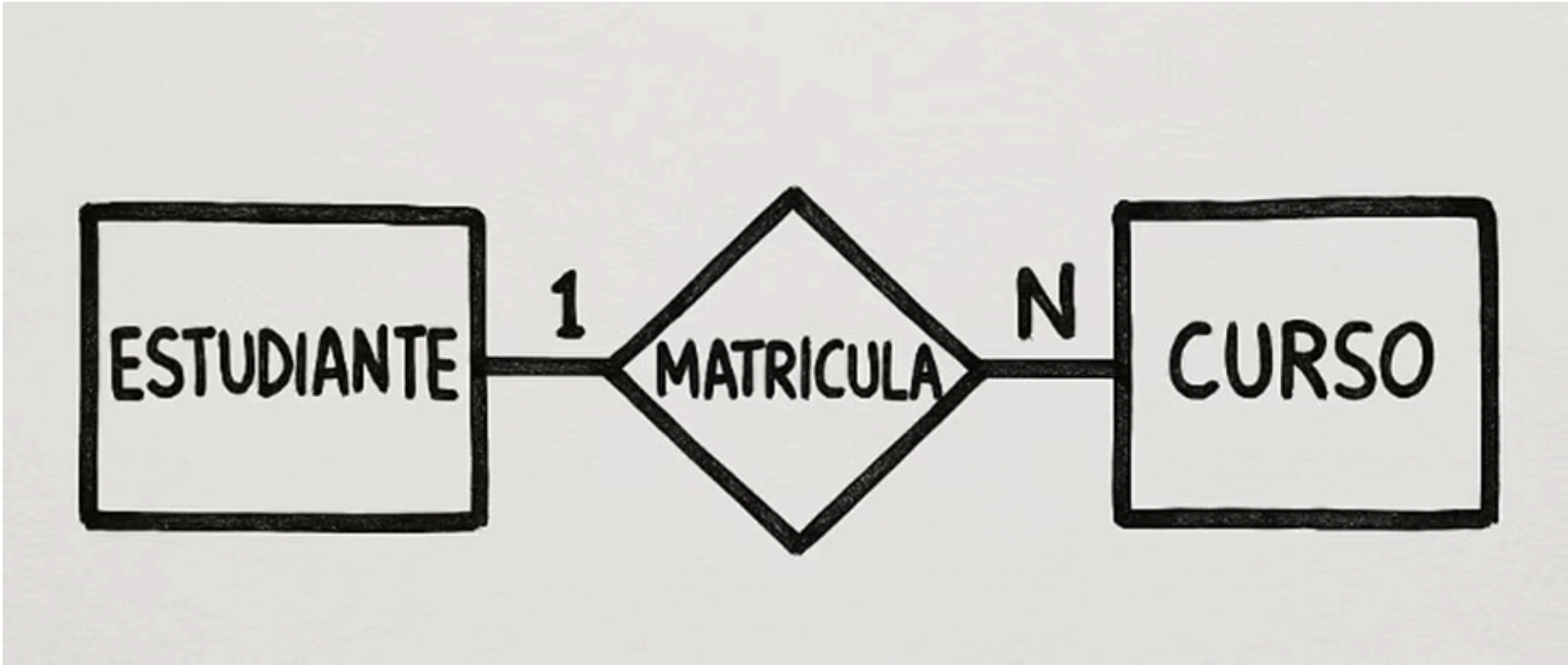
# Actividades de modelado de datos

El modelado de datos no es una tarea puntual, sino un proceso estructurado que implica una serie de actividades secuenciales, desde el levantamiento inicial de requerimientos hasta la documentación final. Estas actividades permiten construir modelos robustos, alineados con los objetivos del negocio, la estructura de la organización y las buenas prácticas de ingeniería de datos.

- Levantamiento de requerimientos: entrevistas, documentación, análisis de procesos.
- Modelado conceptual: entidades, relaciones, sin detalles técnicos.
- Modelado lógico: definición de atributos, tipos de datos, cardinalidades.
- Normalización: eliminación de redundancias y mejora de integridad.
- Modelado físico: adaptación al motor de base de datos específico.
- Validación: revisión con usuarios de negocio y técnicos.
- Documentación y publicación: incluir definiciones, reglas y diagramas.

A continuación, profundizaremos en cada una de estas etapas.

1. Levantamiento de requerimientos Es la etapa inicial y más crítica: aquí se busca comprender qué datos se necesitan, por qué y cómo se usarán. ¿Qué implica? • Entrevistas con usuarios y áreas funcionales. • Revisión de procesos existentes. • Análisis de documentación actual (formularios, reportes, catálogos). • Identificación de reglas del negocio y actores clave. Ejemplo práctico: En una universidad, se identificaría que se gestionan estudiantes, cursos, docentes, matrículas y calificaciones.
2. Modelado conceptual Aquí se traduce el conocimiento del negocio a una representación abstracta, sin entrar aún en detalles técnicos. ¿Qué se diseña? • Entidades (objetos del negocio, como Estudiante, Curso). • Relaciones (por ejemplo, “un estudiante se matricula en varios cursos”). • Cardinalidades básicas. Objetivo: Lograr un lenguaje común entre técnicos y usuarios del negocio. Imagen: Diagrama entidad-relación simple.



3. Modelado lógico Es la traducción técnica del modelo conceptual. Aquí se incorporan los detalles que permitirán implementar el modelo en cualquier sistema, sin atarse aún a una tecnología específica. Incluye: • Definición de atributos y tipos de datos genéricos. • Identificación de claves primarias y foráneas. • Especificación de cardinalidades y restricciones lógicas. Ejemplo: La entidad Estudiante tendrá atributos como ID\_Estudiante, Nombre, FechaNacimiento.

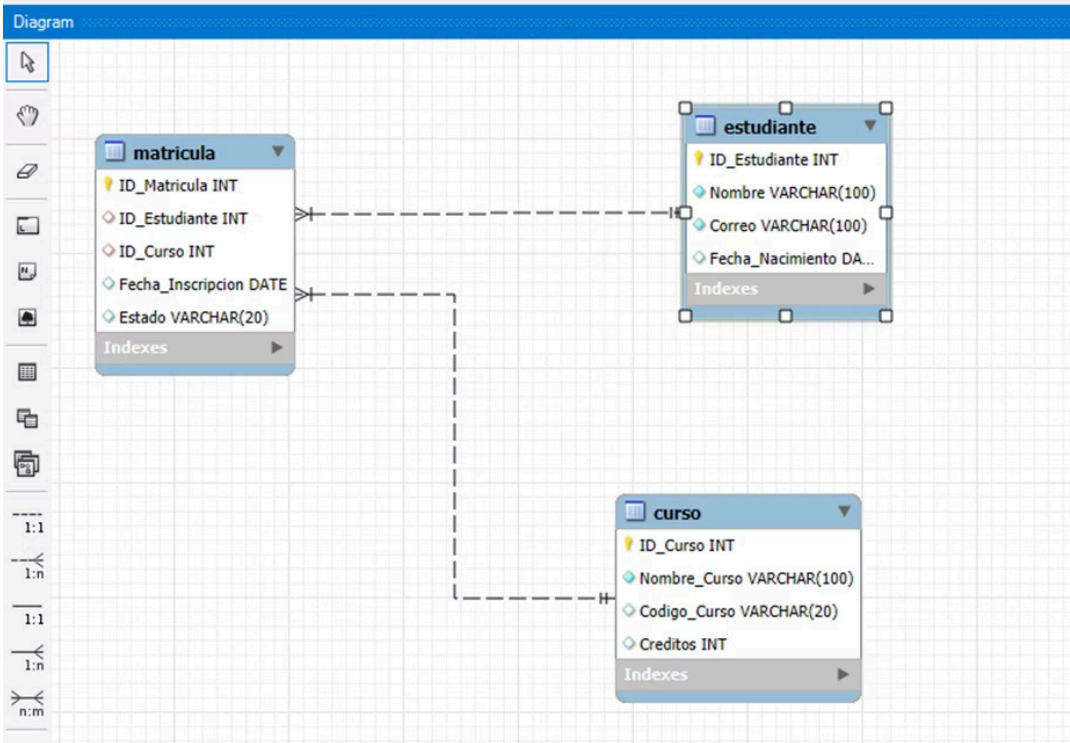


Diagrama con tablas y relaciones. ¿Identifican la diferencia entre el modelo conceptual y el modelo lógico? En el modelo lógico hay más detalles como los tipos de datos, claves primarias y foráneas, y la forma concreta en que se implementan las relaciones entre entidades.

4. Normalización Recordemos que la normalización es una técnica para reducir redundancia, mejorar integridad y facilitar el mantenimiento de los datos. Se aplica sobre el modelo lógico antes de pasar al físico. Fases típicas: Forma Normal Propósito clave 1FN Todos los atributos deben ser atómicos 2FN Cumple 1FN + sin dependencias parciales (en claves compuestas) 3FN Cumple 2FN + sin dependencias transitivas

| Forma Normal | Propósito clave  |
|--------------|--|
| 1FN          | Todos los atributos deben ser atómicos                         |
| 2FN          | Cumple 1FN + sin dependencias parciales (en claves compuestas) |
| 3FN          | Cumple 2FN + sin dependencias transitivas                      |

Consejo: La normalización mejora la calidad del modelo, pero debe equilibrarse con el rendimiento

| Tabla No Normalizada |        |           |       |            |               | Versión Normalizada |        |            |               |
|----------------------|--------|-----------|-------|------------|---------------|---------------------|--------|------------|---------------|
| ID_Cliente           | Nombre | ID_Pedido | Fecha | Ciudad     | Código_Postal | ID_Cliente          | Nombre | Ciudad     | Código_Postal |
| 1                    | Juan   | 101       | 45597 | Santiago   | 12345         | 1                   | Juan   | Santiago   | 12345         |
| 2                    | Maria  | 102       | 45598 | Valparaíso | 54321         | 2                   | Maria  | Valparaíso | 54321         |
| 1                    | Juan   | 103       | 45599 | Santiago   | 12345         |                     |        |            |               |

| ID_Pedido | Fecha     | ID_Cliente |
|-----------|-----------|------------|
| 101       | 1/11/2024 | 1          |
| 102       | 2/11/2024 | 2          |
| 103       | 3/11/2024 | 1          |

Comparación visual entre una tabla “no normalizada” y su versión normalizada.

5. Modelado físico En esta etapa se adapta el modelo lógico a un sistema de gestión de base de datos (SGBD) específico (MySQL, PostgreSQL, Snowflake, etc.). ¿Qué se define? • Tipos de datos específicos (VARCHAR, INT, DATE...). • Índices y particiones. • Restricciones (NOT NULL, UNIQUE, CHECK...). • Parámetros de almacenamiento. Ejemplo: El campo Nombre se implementa como VARCHAR(100) en PostgreSQL.

```
MySQL localhost:33060+ ssl escuela_db_ejemplo SQL > CREATE TABLE Estudiante (
-> ID_Estudiante INT PRIMARY KEY,
-> Nombre VARCHAR(100) NOT NULL,
-> Correo VARCHAR(100) NOT NULL,
-> Fecha_Nacimiento DATE
-> );
Query OK, 0 rows affected (0.0660 sec)
MySQL localhost:33060+ ssl escuela_db_ejemplo SQL > CREATE TABLE Curso (
-> ID_Curso INT PRIMARY KEY,
-> Nombre_Curso VARCHAR(100) NOT NULL,
-> Codigo_Curso VARCHAR(20) UNIQUE,
-> Creditos INT
-> );
Query OK, 0 rows affected (0.0158 sec)
MySQL localhost:33060+ ssl escuela_db_ejemplo SQL > CREATE TABLE Matricula (
-> ID_Matricula INT PRIMARY KEY,
-> ID_Estudiante INT,
-> ID_Curso INT,
-> Fecha_Inscripcion DATE,
-> Estado VARCHAR(20),
-> FOREIGN KEY (ID_Estudiante) REFERENCES Estudiante(ID_Estudiante),
-> FOREIGN KEY (ID_Curso) REFERENCES Curso(ID_Curso)
-> );
Query OK, 0 rows affected (0.0174 sec)
```

Migración del modelo lógico al físico en MySQL (MySQL Shell)

6. Validación Aquí se revisa el modelo completo con usuarios de negocio, analistas y equipos técnicos para asegurar su corrección y utilidad. Validación cruzada: • ¿Representa bien los procesos reales? • ¿Los nombres son claros y coherentes? • ¿Cumple con las reglas del negocio? • ¿Es viable desde el punto de vista técnico? Resultado: Aprobación o ajustes al modelo antes de implementarlo.

7. Documentación y publicación Una vez validado, el modelo debe ser formalmente documentado y publicado para que otros equipos lo comprendan y reutilicen. Elementos que debe incluir: Elemento Descripción Glosario de entidades Definición clara de cada entidad y su propósito. Diccionario de datos Descripción de cada atributo, tipo, formato, restricciones. Diagramas visuales ERD o diagramas lógicos con relaciones. Versionado y autores Quién lo creó, cuándo, cambios realizados.

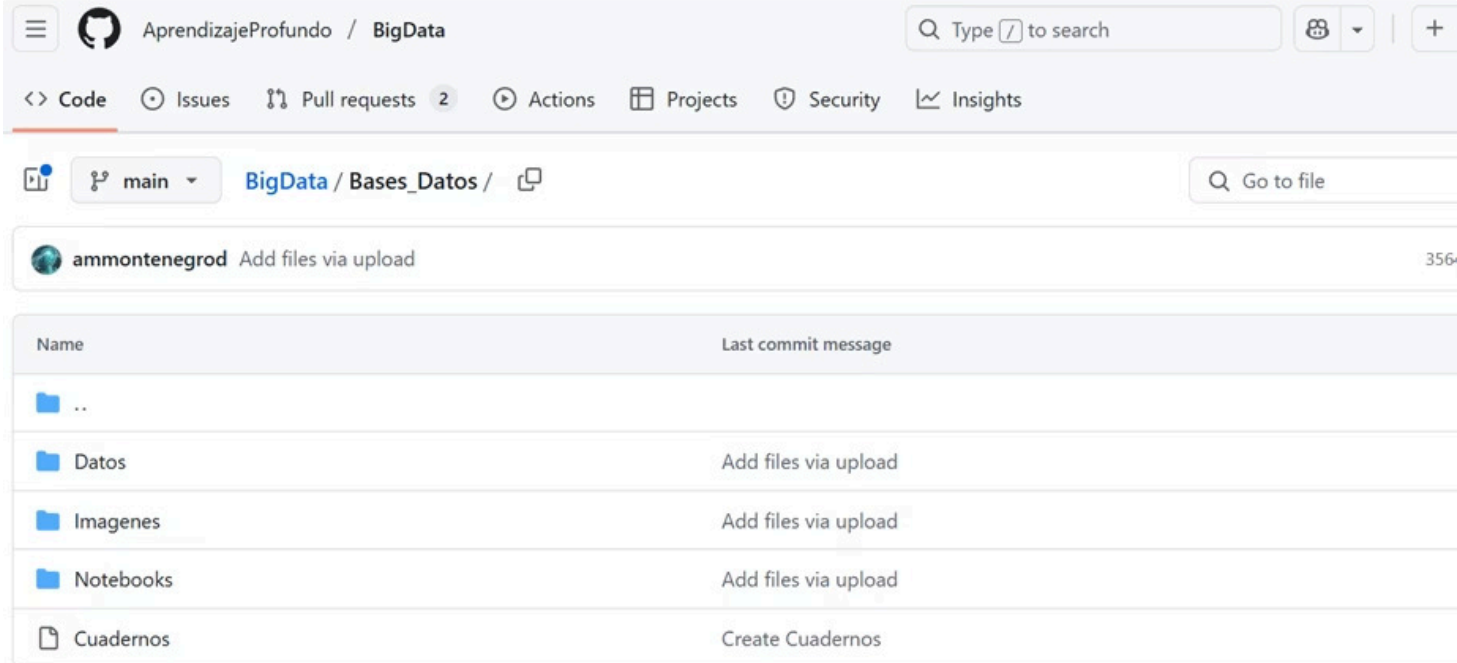


Imagen 12: Captura de un repositorio de modelos en GitHub, puedes revisar el repositorio [aquí](#).

# Herramientas de modelado de datos

Existen herramientas específicas que permiten diseñar, documentar y mantener modelos de datos. Algunas de las más populares incluyen:

| Herramienta            | Tipo              | Características principales                           |
|------------------------|-------------------|---|
| ER/Studio              | Comercial         | Potente en modelado lógico y físico                   |
| Lucidchart             | Online (freemium) | Visualización simple de modelos ER                    |
| MySQL Workbench        | Gratuita          | Diagramas ER con integración a bases MySQL            |
| dbdiagram.io           | Gratuita/Online   | Modelado colaborativo rápido                          |
| PowerDesigner          | Comercial         | Amplia cobertura de ciclo de vida de datos            |
| Draw.io (diagrams.net) | Gratuita          | Ideal para diagramas conceptuales o bocetos iniciales |

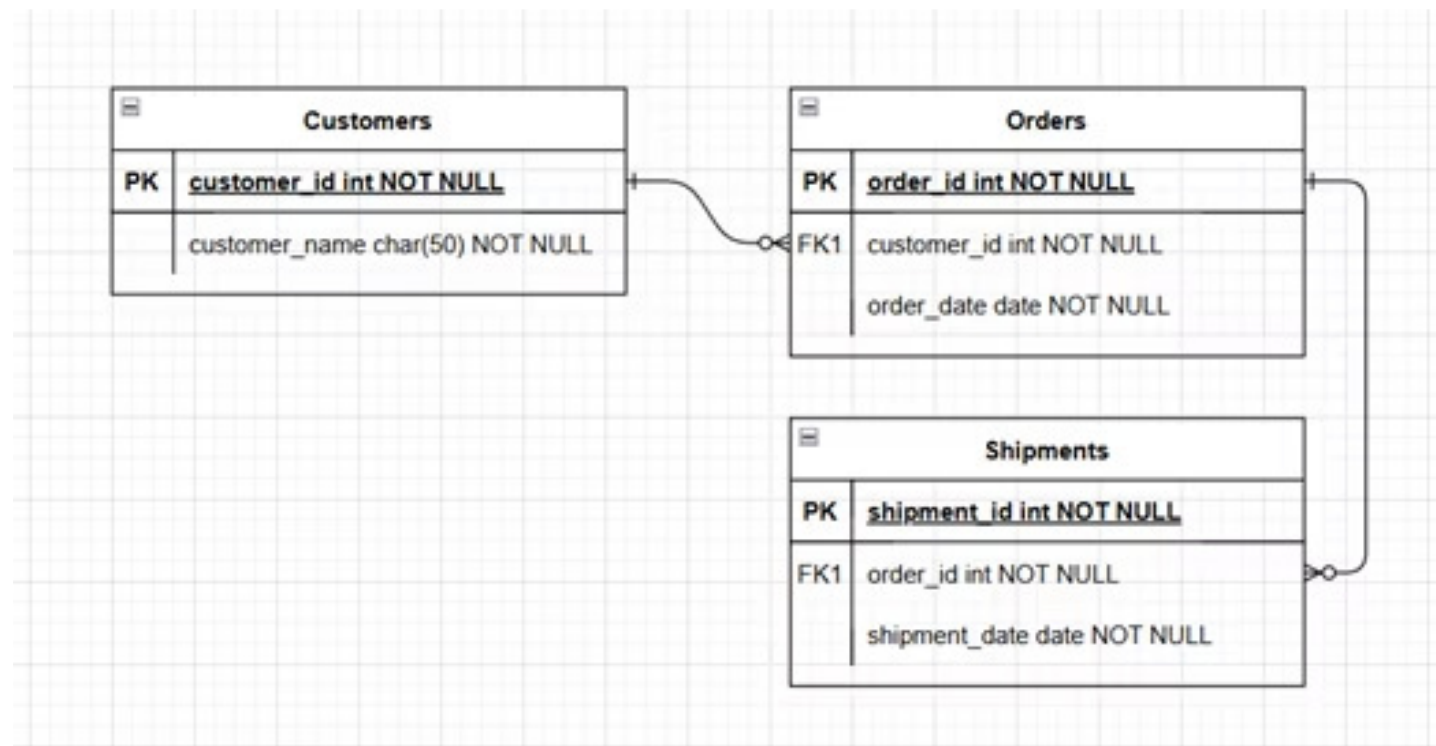


Imagen 13: [Diagrama ER creada en app.diagrams.](#)



# Mejores prácticas (DAMA-DMBOK)

El **DAMA-DMBOK** (Data Management Body of Knowledge) es uno de los marcos más reconocidos a nivel mundial para la gestión de datos. En el ámbito del **modelado y diseño de datos**, establece una serie de buenas prácticas que permiten mantener modelos consistentes, validados y alineados con los objetivos del negocio.

Estas prácticas no solo ayudan a construir mejores bases de datos, sino que también **fortalecen la comunicación entre los equipos técnicos y funcionales**, evitando malentendidos y retrabajos.

## Principales buenas prácticas recomendadas por DAMA-DMBOK

| Práctica                                       | Descripción   | Ejemplo / Sugerencia   |
|--|---|--|
| Revisión constante con stakeholders            | Involucrar a usuarios de negocio, analistas y técnicos en iteraciones del modelo.             | Reuniones quincenales para revisar modelos en evolución.                               |
| Versionamiento controlado del modelo           | Cada cambio debe quedar registrado con fecha, autor y propósito.                              | Usar herramientas como Git o repositorios internos de modelos (ej: dbdocs.io).         |
| Nomenclatura estándar y reglas de consistencia | Establecer un lenguaje común en nombres de entidades, atributos y relaciones.                 | Prefijos como ID_, uso consistente de singular o plural.                               |
| Documentación centralizada                     | Mantener un diccionario de datos accesible a todos los equipos.                               | Documentar definiciones, formatos, orígenes y reglas de validación.                    |
| Validaciones técnicas y de negocio             | Validar el modelo antes del paso a producción tanto desde lo técnico como desde lo funcional. | Revisión de integridad referencial, claves, reglas de negocio aplicadas.               |
| Separación de modelos lógico y físico          | Diseñar primero en forma abstracta (modelo lógico), luego adaptarlo al SGBD (modelo físico).  | Evita decisiones apresuradas como tipos de datos o nombres reservados desde el inicio. |

**“El modelado es tanto una actividad técnica como una conversación entre áreas. Si no se entiende en ambos mundos, no es un buen modelo.”**

Esta frase destaca un principio fundamental del DAMA-DMBOK: **un modelo de datos no es solo un diseño técnico, sino también un artefacto de comunicación entre áreas**. Si los usuarios de negocio no lo comprenden, o los técnicos no lo implementan correctamente, el modelo pierde valor.

# Datos maestros (MDM)

La gestión de datos maestros (Master Data Management, MDM) es una de las disciplinas más críticas dentro del ciclo de vida de los datos, especialmente cuando trabajamos con múltiples sistemas, aplicaciones o fuentes que hacen referencia a las mismas entidades clave del negocio, como clientes, productos, empleados o ubicaciones.

## ¿Qué busca el MDM?

El objetivo central de MDM es garantizar que ciertos datos fundamentales y compartidos estén siempre:

| Criterio   | Descripción   |
|------------|---|
| Uniformes  | No existen diferencias en nombres, formatos o atributos entre sistemas.     |
| Precisos   | No contienen errores, omisiones o datos desactualizados.                    |
| Accesibles | Están disponibles para los equipos que los necesitan, cuando los necesitan. |
| Gobernados | Cuentan con procesos de control, validación y trazabilidad.                 |

¿De qué sirve diseñar modelos de datos increíbles si luego cada sistema los interpreta a su manera?

La gestión de datos maestros no es solo para empresas grandes. Incluso en proyectos más simples, mantener definiciones consistentes es clave para evitar errores y tomar mejores decisiones.

### ¿Qué son los "datos maestros"?

Se denominan **datos maestros** a aquellos que representan **entidades clave del negocio** que deben ser consistentes y confiables en toda la organización. No cambian con frecuencia y se usan transversalmente en muchas áreas.

Ejemplos comunes de dominios de datos maestros:

| Dominio        | Atributos típicos                                     |
|----------------|---|
| Clientes       | Nombre, RUT, dirección, correo, historial de compra   |
| Productos      | Código SKU, nombre del producto, categoría, proveedor |
| Proveedores    | RUT o ID fiscal, nombre, condiciones de pago          |
| Ubicaciones    | Región, comuna, dirección, código postal              |
| Empleados (HR) | ID interno, nombre, cargo, unidad, fecha de ingreso   |



# ¿Qué pasa cuando falla el MDM?

Cuando una organización no implementa una estrategia de MDM, puede enfrentar los siguientes **problemas críticos**:

- **Facturación duplicada** por registros repetidos de clientes.
- **Errores de inventario** por códigos de producto inconsistentes.
- **Problemas de cumplimiento legal** (ej. en normativas tributarias o protección de datos).
- **Análisis poco confiables**, con reportes contradictorios entre áreas.

**Ejemplo real:** Dos áreas de una empresa tienen registrado al mismo cliente con distintos nombres ("**Juan Pérez**" y "**J. Pérez**"). Resultado: se generan dos facturas, se rompe la trazabilidad del historial de compras y se pierde control sobre promociones activas.

## ¿Cómo contribuye el modelado de datos al MDM?

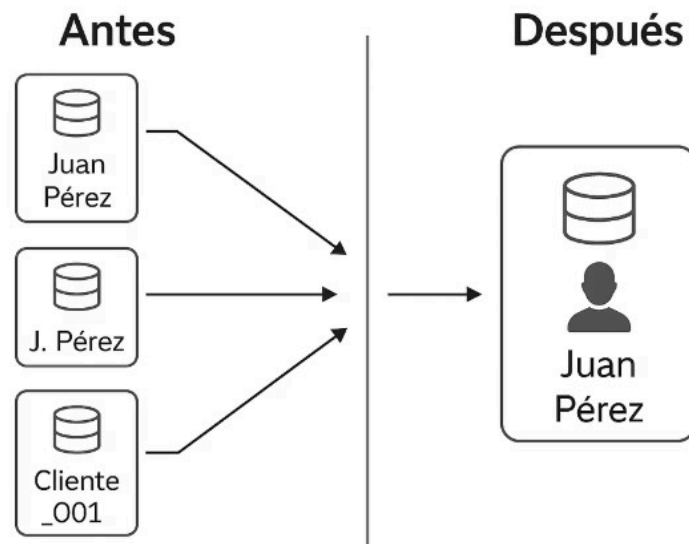
Desde la perspectiva del modelado de datos, es **esencial definir correctamente las entidades maestras** y prever su reutilización, consistencia y control en todos los sistemas donde se usen.

Recomendaciones de modelado para apoyar el MDM:

| Recomendación                                | Descripción   |
|--|---|
| Definir entidades maestras como tablas clave | Tablas como Cliente, Producto, Ubicación deben ser reutilizadas, no duplicadas.               |
| Establecer claves únicas (ej: RUT, SKU)      | Asegura la identificación unívoca de los registros.   |
| Usar relaciones foráneas con otras tablas    | Impide que se creen datos huérfanos o inconsistentes.   |
| Validar atributos con reglas de negocio      | Por ejemplo, que un RUT tenga un formato específico o que una comuna pertenezca a una región. |
| Consolidar orígenes en una vista central     | En sistemas complejos, usar procesos ETL para unificar datos dispersos.                       |

# ¿Qué pasa cuando falla el MDM?

## Visión unificada con MDM: del caos a la coherencia



Representación de Unificación de datos de cliente mediante MDM: del registro fragmentado a una vista única y coherente.

### ¿De qué sirve diseñar modelos de datos increíbles si luego cada sistema los interpreta a su manera?

La gestión de datos maestros **no es solo para empresas grandes**. Incluso en proyectos más simples, mantener definiciones consistentes es clave para evitar errores y tomar mejores decisiones.

### ¿Dónde se conecta esto con lo que hemos aprendido?

En esta sesión hablamos de modelado, gobernanza, nomenclatura, calidad, y ahora todo eso se junta. Porque **un modelo de datos sin gestión de maestros puede estar bien diseñado, pero lleno de errores** si nadie se preocupa de mantener un registro único, actualizado y coherente de cada entidad clave.

MDM es una práctica que actúa como puente entre la teoría del modelado y la realidad del negocio.