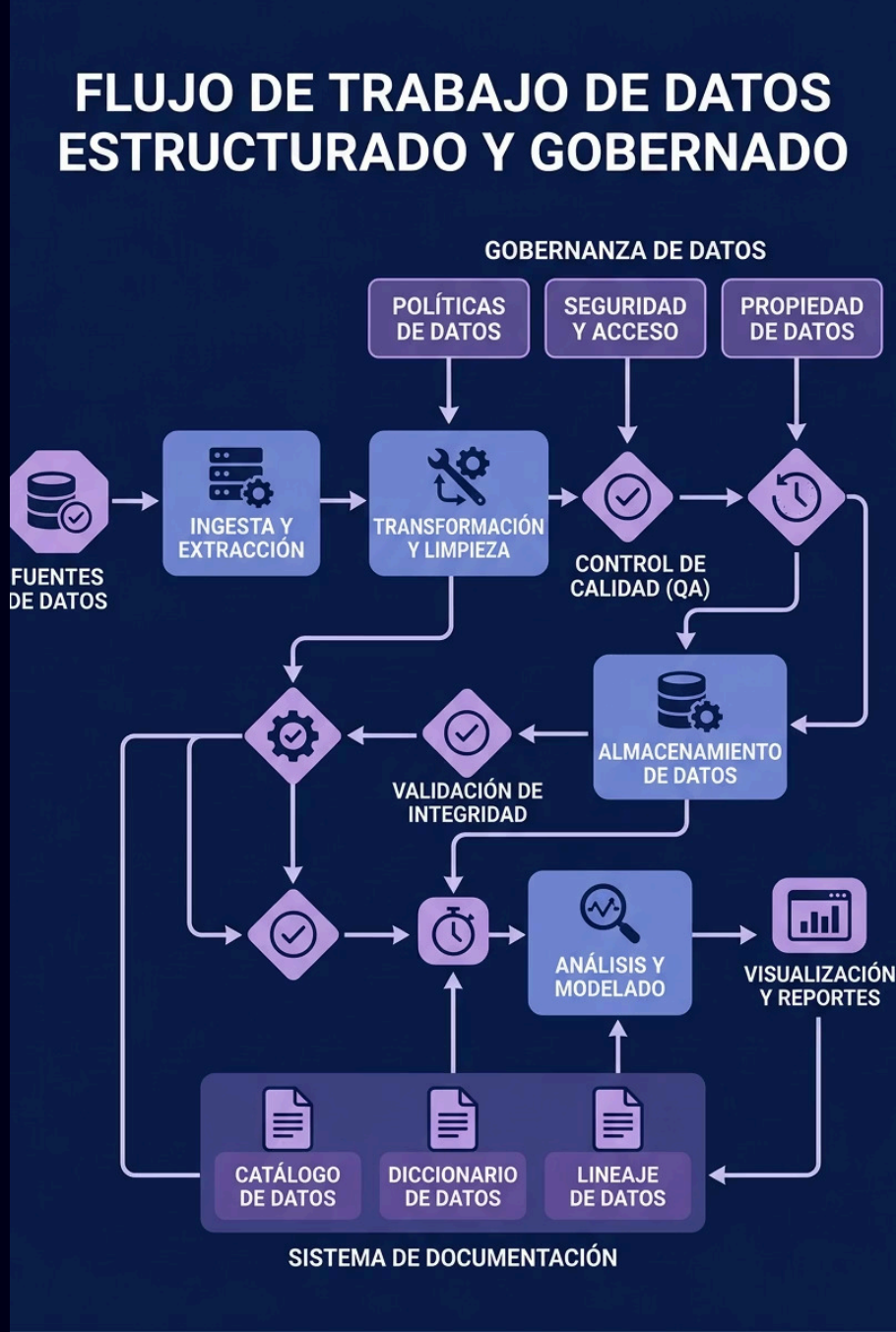
Dateneo®

Gobierno y calidad de datos

Cómo garantizar datos confiables y gobernados en pipelines modernos

Una guía práctica para ingenieros de datos que buscan implementar procesos robustos de gobierno, validación y documentación de datos.



¿Qué es el Gobierno de datos?

Definición

Conjunto de **procesos, roles, políticas y herramientas** para asegurar que los datos sean **confiables** (calidad validada), **seguros** (accesos controlados) y **disponibles** (ubicables y reutilizables).

Objetivos principales

- Garantizar **confianza** en los datos
- Mejorar la **toma de decisiones** basadas en evidencia
- Cumplir con **regulaciones** (GDPR, ISO, etc.)

Roles clave

- **Data owners:** responsables funcionales del dataset
- **Data stewards:** guardianes de la calidad y definición
- **Ingenieros de datos:** desarrollan procesos y pipelines acordes a las políticas de gobierno

El gobierno de datos no es solo una cuestión técnica, sino también organizacional y cultural que requiere compromiso a todos los niveles.

Gobierno aplicado a pipelines de datos

Ownership claro

Cada dataset debe tener un responsable definido que:

- Conoce su estructura y significado
- Aprueba cambios estructurales
- Responde ante incidentes

Catálogo de datos

Documentación centralizada que incluye:

- Datasets, columnas y descripciones
- Reglas de negocio aplicables
- Herramientas: Glue Data Catalog, DataHub, OpenMetadata

Linaje de datos

Trazabilidad que permite:

- Entender transformaciones y flujos
- Column-level lineage vs. table-level lineage

Compliance y accesos

Control granular sobre:

- Quién puede ver cada dato
- Quién puede modificar información
- Automatización de políticas

Calidad de datos

¿Qué es un dato?

Representaciones simbólicas

Los datos son representaciones de hechos, eventos o entidades que ocurren en el mundo real o virtual.

Ejemplos concretos

- **Evento:** milímetros de agua caídos en una tormenta
- **Hecho:** monto transferido en una operación bancaria
- **Entidad:** color de la pintura de un automóvil

Finalidad

Los usamos para:

- Comprender la realidad
- Anticipar lo que puede ocurrir
- Intervenir para lograr objetivos

> "Los datos son la materia prima para construir conocimiento, pero solo si representan correctamente la realidad que intentan describir."

¿Por qué Importa la calidad de datos?

La calidad de datos es fundamental porque:

- Permite tomar **decisiones confiables** basadas en evidencia real
- Es la base para **productos de datos robustos** (modelos ML, dashboards, APIs)
- Previene **errores costosos** derivados de inconsistencias o datos incompletos
- Optimiza la **eficiencia operativa** al reducir reprocesos y validaciones manuales
- Mejora la **confianza** de los usuarios en los sistemas de información

⚠ El costo del error

Según estudios, las organizaciones pierden entre el 15-25% de sus ingresos por problemas de calidad de datos.

"Garbage in, garbage out"

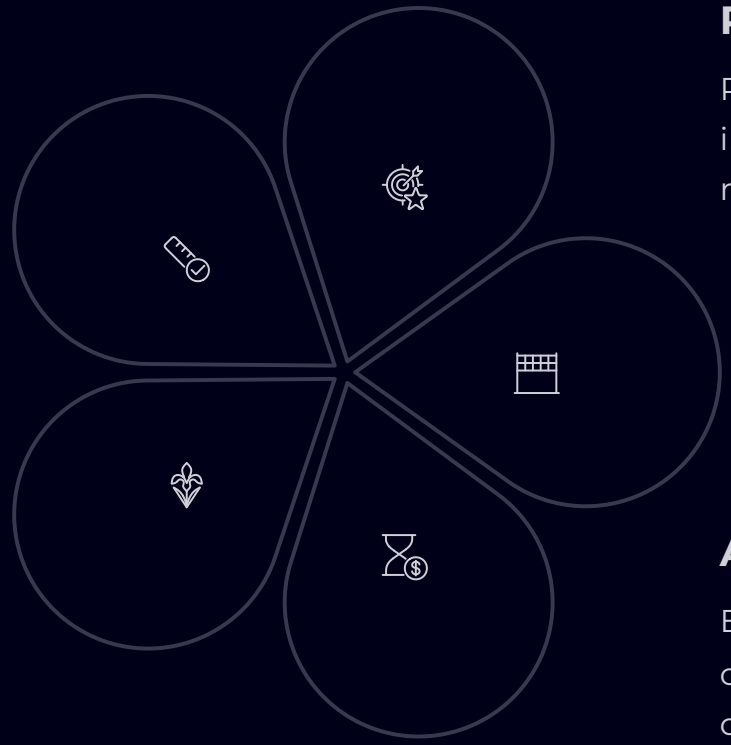
Dimensiones de la calidad de datos

Consistencia

Los datos no se contradicen entre sí ni con otras fuentes. Ejemplo: el total de ventas coincide en diferentes informes.

Relevancia

Sirven para el propósito y decisiones que se necesitan tomar. Ejemplo: incluyen las variables necesarias para segmentar clientes.



Precisión

Reflejan exactamente la realidad que intentan representar. Ejemplo: el precio registrado es el precio real cobrado.

Completitud

No faltan datos críticos para el análisis o proceso. Ejemplo: todos los clientes tienen información de contacto.

Actualidad

Están actualizados según lo esperado por el caso de uso. Ejemplo: datos de inventario con menos de 1 hora de antigüedad.

Estas dimensiones deben evaluarse en el contexto específico de cada organización y caso de uso.

Preguntas para hacernos sobre calidad

¿Qué?

La calidad de datos es un indicador de si los datos sirven para tomar decisiones confiables y precisas.

¿Cuándo?

Se debe validar en **múltiples momentos**:

- En la **fuente de origen** (ideal)
- Durante el **desarrollo** de pipelines
- En **producción**, de forma continua
- Antes de **decisiones críticas**

¿Cómo?

1. **Análisis exploratorio** para entender patrones
2. **Definir reglas** basadas en expectativas del negocio
3. **Validar** de forma manual o con herramientas
4. **Automatizar** las validaciones en pipelines
5. **Medir y reportar** niveles de calidad

Reglas de calidad básicas

1

Claves Primarias

Deben ser **únicas** y **no nulas**.

```sql
-- Ejemplo SQL para validar
SELECT id, COUNT(*)
FROM tabla
GROUP BY id
HAVING COUNT(*) > 1
```

2

Claves Foráneas

Deben **existir** en la tabla primaria correspondiente.

```sql
-- Validar integridad referencial
SELECT DISTINCT a.id_cliente
FROM pedidos a
LEFT JOIN clientes b ON a.id_cliente = b.id
WHERE b.id IS NULL
```

3

Valores predefinidos

Cumplir con **listas de valores** o **umbrales** establecidos.

```sql
-- Validar valores dentro de lista
SELECT estado, COUNT(*)
FROM pedidos
WHERE estado NOT IN ('Pendiente','Aprobado','Rechazado')
GROUP BY estado
```

4

Totales origen vs destino

Verificar **conteos** y **agregaciones** entre tablas relacionadas.

```sql
-- Comparar totales
SELECT SUM(monto) FROM transacciones_origen;
SELECT SUM(monto) FROM transacciones_procesadas;
```

Reglas de calidad avanzadas

Reglas de negocio complejas

Validaciones que reflejan la lógica específica del dominio:

- El descuento no puede exceder el 30% para clientes regulares
- Un usuario no puede tener más de 5 sesiones simultáneas
- La suma de porcentajes debe ser exactamente 100%

Detección de outliers

Identificar valores atípicos mediante:

- Comparación con promedios históricos
- Desviaciones estándar
- Métodos estadísticos como IQR
- Algoritmos de detección de anomalías

Validación de frescura (SLA)

Asegurar que los datos están actualizados según lo requerido:

- Tiempo transcurrido desde última actualización
- Cumplimiento de ventanas de actualización

Coherencia entre fuentes

Validar consistencia cuando el mismo dato aparece en múltiples sistemas:

- Reconciliación entre data warehouse y sistemas operacionales
- Verificación cruzada entre canales

Acciones sobre las validaciones

"Si validamos y no hacemos nada, no sirve."

Integrar en Pipelines

Incorporar las validaciones directamente en el código de los pipelines de datos utilizando frameworks como Great Expectations, dbt tests o Soda Core.

Automatizar ejecución

Programar la ejecución recurrente de validaciones como parte del ciclo de vida de los datos, no como un proceso aislado o manual.

Alertar y comunicar

Configurar sistemas de alertas en tiempo real para problemas críticos y generar informes automáticos de calidad para stakeholders.

Acción correctiva

Implementar procesos de corrección automáticos o manuales bien definidos para cada tipo de problema detectado.

Tratar los problemas de calidad de datos como fallas del sistema, no como "cuestiones de datos" aisladas.

Equilibrio en las validaciones

Riesgos del exceso

Demasiadas validaciones pueden ser contraproducentes:

- Generan **sobrecarga computacional** innecesaria
- Aumentan la **complejidad** del mantenimiento
- Producen **fatiga de alertas** que lleva a ignorarlas
- Ralentizan los **tiempos de procesamiento**

Estrategias de optimización

Para un enfoque balanceado:

- **Priorizar validaciones** según impacto en el negocio
- Evitar **redundancia** en los controles
- Aplicar **muestreo** para validaciones costosas
- Implementar **validaciones incrementales** que solo revisen datos nuevos
- Usar **validaciones asincrónicas** para casos no críticos

Buscar el punto óptimo donde la validación maximiza la confianza sin comprometer el rendimiento.

Mejora continua de la calidad

Evaluar métricas

Monitorear indicadores clave de calidad y su evolución en el tiempo.

Documentar cambios

Mantener registro de evolución de las reglas y su justificación.

Incorporar feedback

Recopilar retroalimentación de usuarios finales sobre problemas de datos.

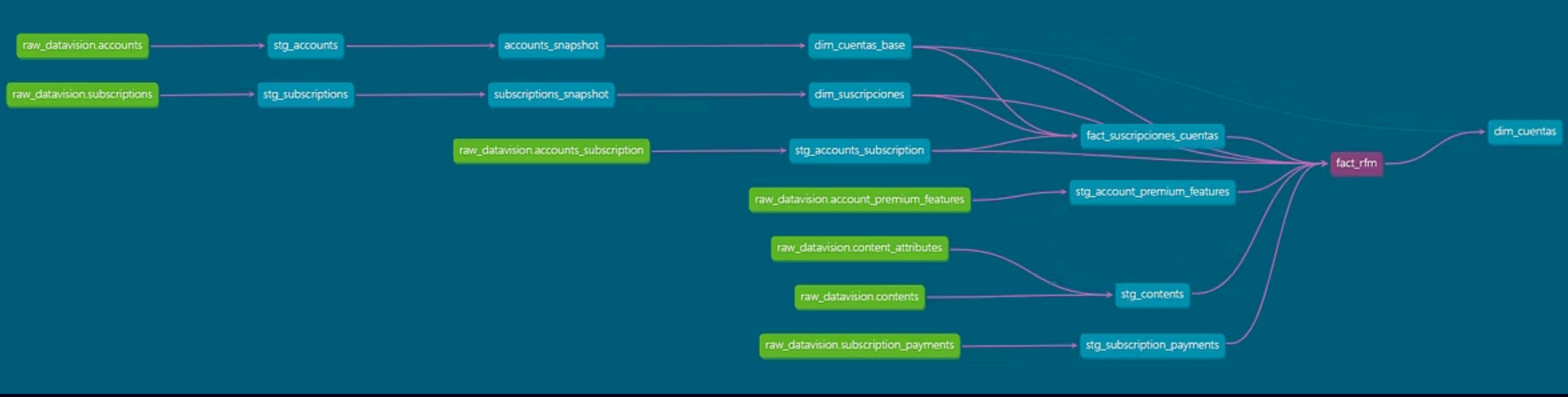
Optimizar reglas

Ajustar umbrales y eliminar validaciones que ya no aportan valor.

Añadir nuevas validaciones

Incorporar controles para cubrir casos de uso emergentes.

La calidad de datos no es un proyecto con fecha de finalización, sino un proceso iterativo de mejora constante.



Linaje de datos

¿Qué es?

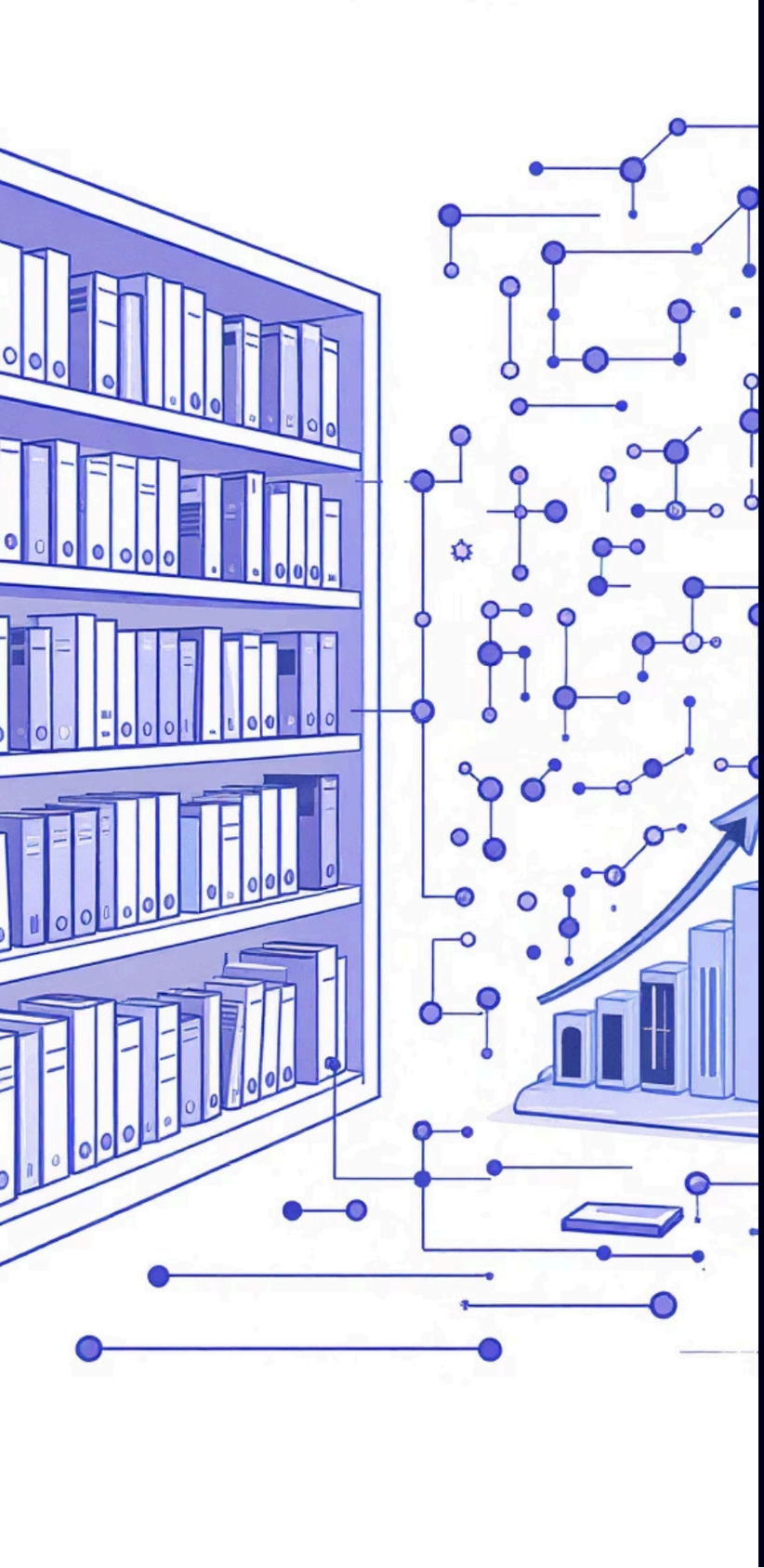
Trazabilidad completa de cómo los datos se mueven y se transforman a lo largo de su ciclo de vida, desde su origen hasta su consumo final.

Tipos de linaje

- **A nivel tabla:** muestra relaciones entre datasets completos
- **A nivel columna:** detalla transformaciones de campos específicos (más valioso pero más complejo)

¿Por qué importa?

- **Diagnóstico de errores:** identificar dónde se originó un problema
- **Análisis de impacto:** evaluar consecuencias antes de modificar un modelo
- **Auditoría y compliance:** demostrar origen y transformaciones de datos sensibles
- **Confianza:** entender de dónde viene cada dato y cómo se calculó
- **Optimización:** detectar duplicaciones o ineficiencias en el procesamiento



Catálogo de datos

Es una herramienta que:

- Centraliza metadatos, documentación, *ownership* y linaje.
- Permite buscar, descubrir y entender *datasets* de forma rápida.
- Habilita la colaboración entre equipos y la gobernanza de datos.

Opciones Open Source

- **OpenMetadata:** moderno, *column-level lineage*, taxonomías, *owners*.
- **DataHub:** muy usado, buena integración con *pipelines* y CI/CD.
- **Amundsen:** interfaz simple, fuerte en descubrimiento.

Opciones pagas

- **Alation:** robusto en gobernanza y *compliance*.
- **Collibra:** muy usado en grandes organizaciones con requisitos complejos.
- **Informatica Axon:** fuerte en metadatos empresariales y *workflows* de aprobación.

Nota sobre AWS Glue Data Catalog

⚠ **Glue NO es un catálogo de datos completo:**

- ✅ Almacena metadatos técnicos (tablas, columnas, tipos).
- ❌ No permite definir *owners* ni dominios de negocio.
- ❌ No gestiona glosarios ni búsqueda de términos.
- ❌ No habilita vistas colaborativas ni *governance* avanzado.

Se usa más como repositorio técnico que como catálogo de datos corporativo.

Conclusiones

El Gobierno es un habilitador

No es burocracia sino una forma de **empoderar** a los equipos con datos confiables y accesibles.

La calidad requiere compromiso

Debe ser parte integral del **diseño de pipelines**, no una consideración posterior.

El equilibrio es clave

Encontrar el balance entre **rigurosidad** y **agilidad** para cada contexto específico.

Los datos son el activo más valioso de las organizaciones modernas, y su gobierno, calidad y trazabilidad son **responsabilidad compartida** de todos los ingenieros de datos.