

INFORME: Proyecto Integrador: Arquitectura de Datos

Introducción

El presente proyecto tiene como objetivo diseñar una arquitectura de datos robusta, escalable y gobernada que permita integrar diversas fuentes, asegurar la calidad, almacenar eficientemente y preparar la información para análisis avanzado en una empresa de e-commerce y retail digital.

Lección 1 - Arquitectura de Datos

Resumen

Se identificaron fuentes de datos clave: ventas online (estructurado), inventario y logística (estructurado), redes sociales y comportamiento web (no estructurado), y fotografías de productos (no estructurado).

La arquitectura propuesta es modular y por capas:

- **Ingesta:** extracción y captura de datos mediante batch y streaming.
- **Integración:** limpieza y transformación con procesos ETL/ELT.
- **Almacenamiento:** zonas diferenciadas de Data Lake, Data Warehouse y Data Marts.
- **Calidad y Gobierno:** controles de calidad, catálogos y linaje.
- **Consumo:** acceso mediante BI, dashboards y modelos de IA.

Se aplican principios de gobernanza, escalabilidad y flexibilidad para asegurar un sistema confiable y adaptable.

Lección 2 - Enfoques para Almacenamiento y Gestión

Resumen

Se definieron las zonas del Data Lake:

- **Raw/Bronze:** datos originales sin procesar.
- **Trusted/Silver:** datos validados y limpios.
- **Curated/Gold:** datos preparados para análisis.

Estas zonas alimentan al Data Warehouse, que organiza los datos para el consumo analítico, y a los Data Marts, enfocados en áreas de negocio específicas.

Se seleccionaron tecnologías compatibles con el contexto cloud y on-premise, priorizando escalabilidad y seguridad.

Lección 3 - Calidad de los Datos

Resumen

Se diseñó un plan integral para el aseguramiento de calidad con controles y métricas como:

- Completitud
- Consistencia
- Unicidad
- Validez
- Actualidad

Se definió un proceso de monitoreo continuo que permite detectar anomalías y activar mecanismos de remediación automatizados o manuales, integrando estos controles en el flujo arquitectónico.

Lección 4 - Modelamiento Multidimensional

Resumen

Se seleccionó el área de ventas como foco del modelo OLAP. Se diseñó un esquema dimensional tipo estrella, compuesto por:

- **Tabla de hechos:** FactVentas (ventas, cantidad, monto, descuentos).
- **Dimensiones:** Cliente, Producto, Tiempo, Canal, Ubicación.

Se priorizó la desnormalización para optimizar la consulta y asegurar rapidez en análisis multidimensionales, alineado con las zonas de almacenamiento y calidad definidas.

Diagrama Integrador Final

El diagrama integrador refleja el flujo completo y la interacción entre capas y zonas, mostrando:

- **Fuentes de datos** heterogéneas alimentando la capa de ingesta (batch y streaming).
- La capa de integración que realiza limpieza y transformación (ETL/ELT).
- Las zonas del Data Lake (Raw, Trusted, Curated) para almacenar datos en diferentes etapas de madurez.
- El Data Warehouse y Data Marts para almacenamiento analítico.
- El módulo de Calidad y Gobernanza con controles y catálogo.
- El modelo dimensional OLAP para consumo analítico mediante BI, dashboards y modelos avanzados.

Este diseño asegura un flujo transparente, gobernado y escalable, permitiendo que los datos transiten con calidad garantizada hacia los usuarios finales para la toma de decisiones.

Conclusión

El proyecto integra todas las etapas necesarias para construir una arquitectura moderna de datos, desde la ingesta hasta el consumo, pasando por un riguroso aseguramiento de calidad y una gobernanza efectiva.

El diseño modular y escalable permite la incorporación de nuevas fuentes y tecnologías, asegurando la flexibilidad y el crecimiento futuro de la plataforma analítica.

La documentación, diagramas y modelos generados constituyen una base sólida para la implementación y evolución del sistema de datos en la empresa.

✓ Lección 1 - Arquitectura de Datos

Objetivo

Diseñar un esquema arquitectónico robusto para la **integración, almacenamiento y consumo de datos**, que soporte múltiples fuentes, asegure la calidad y aplique principios de **gobierno, escalabilidad y flexibilidad**.

1 Fuentes de datos relevantes

En la empresa de e-commerce y retail digital, se identifican al menos tres fuentes **estructuradas y no estructuradas** clave para la analítica:

Tipo de fuente	Ejemplo	Descripción	Frecuencia
Estructurada (SQL/CSV)	Base de datos transaccional de ventas online (OLTP)	Contiene datos de pedidos, pagos y devoluciones.	Tiempo real / diario
Estructurada (API)	Sistema de inventario y logística	Información de stock, envíos, tiempos de entrega y devoluciones.	Tiempo real / cada hora
No estructurada (JSON/Texto)	Datos de redes sociales y comportamiento web	Comentarios de clientes, clics, páginas vistas, interacción con campañas.	Streaming continuo
No estructurada (Imagen)	Fotografías de productos	Imágenes para análisis visual (ej. detección de defectos, tendencias).	A demanda

2 Diseño de arquitectura escalable y modular por capas

Propuesta de arquitectura **moderna y escalable** con cinco capas:

Capa	Función	Tecnologías posibles	Buenas prácticas
Ingesta	Conectar y extraer datos desde las fuentes (batch y streaming).	Apache Nifi, Airflow, Kafka, AWS Glue	Conectores dedicados, formatos
Integración	Limpiar, transformar, estandarizar y unificar datos.	Spark, Databricks, Flink	Procesos ETL/ELT documentados
Almacenamiento	Guardar datos en distintas zonas según madurez: Raw/Bronze, Refined/Silver, Curated/Gold.	Data Lake (S3, GCS), DW (BigQuery, Snowflake), Data Marts	Organizar por nivel de calidad y uso
Calidad y Gobernance	Asegurar completitud, consistencia y confiabilidad.	AWS Glue Data Catalog, Apache Atlas	Controles automáticos, linaje, catálogos
Consumo	Exponer datos para usuarios y modelos analíticos.	Power BI, Tableau, Looker, APIs	Dashboards, reportes, modelos predictivos

3 Principios de gobierno, escalabilidad y flexibilidad

- **Gobierno de datos:** Roles y permisos claros (RBAC), catálogo y diccionario, cumplimiento normativo (GDPR, Ley de Protección de Datos).
- **Escalabilidad:** Arquitectura cloud-native con escalado automático, separación almacenamiento-cálculo para optimizar costos.
- **Flexibilidad:** Soporte para datos estructurados y no estructurados, integración modular para incorporar nuevas fuentes sin rediseñar.

4 Diagrama arquitectónico

Para este proyecto se usa un diagrama generado con la librería **Graphviz** en Python, ideal para notebooks como Google Colab porque se puede crear y visualizar sin salir del entorno.

El diagrama representa:

- **Fuentes:** Tres fuentes principales (Ventas Online, Comportamiento Web, Redes Sociales), agrupadas horizontalmente para mayor claridad visual.
- **Capas:** Ingesta → Integración ETL/ELT → Almacenamiento (Data Lake, Data Warehouse, Data Marts) → Calidad y Gobierno → Consumo (Dashboards / BI y Data Science / IA).
- **Flujo de datos:** Se representa con flechas que muestran cómo los datos se mueven y transforman desde la captura hasta el consumo.

Características del diagrama:

- Se usa `rank='same'` para alinear horizontalmente nodos del mismo nivel (ej. fuentes y consumo).
- Los nodos tienen formas y colores que ayudan a diferenciar funciones (carpetas para fuentes, cilindros para almacenamiento, cajas para procesamiento, notas para consumo).
- El diseño facilita la comprensión del flujo y la modularidad del sistema.

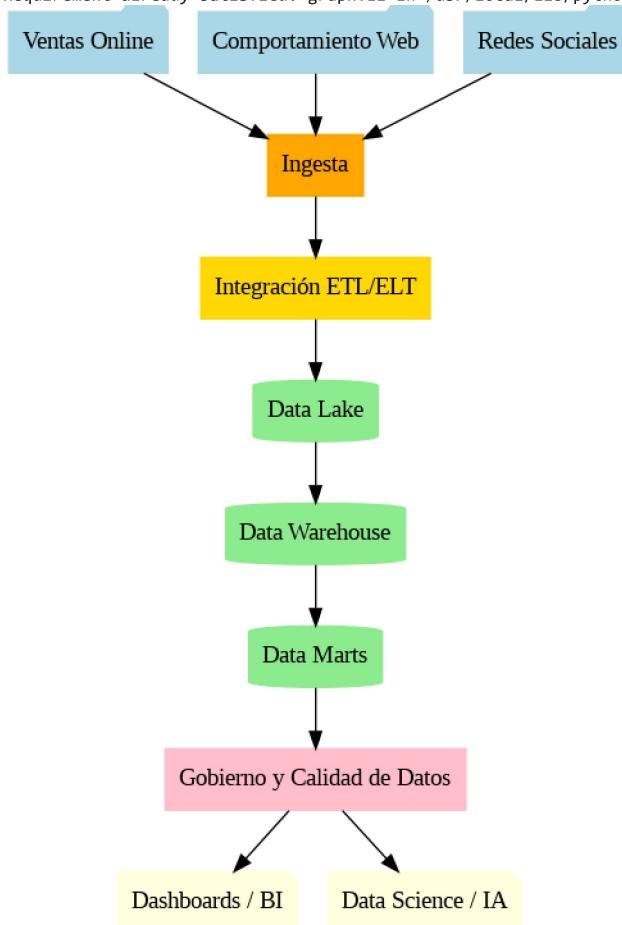
Haz doble clic (o ingresa) para editar

```

1 # [1] Instalar Graphviz (solo necesario la primera vez que corras el notebook)
2 !apt-get install -qq graphviz
3 !pip install graphviz
4
5 # [2] Importar librería
6 from graphviz import Digraph
7 from IPython.display import Image
8
9 # [3] Crear el diagrama con agrupación horizontal para fuentes y consumo
10 dot = Digraph(comment='Arquitectura de Datos')
11
12 # Nodos Fuentes
13 dot.node('F1', 'Ventas Online', shape='folder', style='filled', color='lightblue')
14 dot.node('F2', 'Comportamiento Web', shape='folder', style='filled', color='lightblue')
15 dot.node('F3', 'Redes Sociales', shape='folder', style='filled', color='lightblue')
16
17 # Agrupar fuentes en misma fila horizontal
18 with dot.subgraph() as s:
19     s.attr(rank='same')
20     s.node('F1')
21     s.node('F2')
22     s.node('F3')
23
24 # Capas intermedias
25 dot.node('I1', 'Ingesta', shape='box', style='filled', color='orange')
26 dot.node('I2', 'Integración ETL/ELT', shape='box', style='filled', color='gold')
27 dot.node('A1', 'Data Lake', shape='cylinder', style='filled', color='lightgreen')
28 dot.node('A2', 'Data Warehouse', shape='cylinder', style='filled', color='lightgreen')
29 dot.node('A3', 'Data Marts', shape='cylinder', style='filled', color='lightgreen')
30 dot.node('C1', 'Gobierno y Calidad de Datos', shape='box', style='filled', color='pink')
31
32 # Nodos consumo
33 dot.node('BI', 'Dashboards / BI', shape='note', style='filled', color='lightyellow')
34 dot.node('DS', 'Data Science / IA', shape='note', style='filled', color='lightyellow')
35
36 # Agrupar nodos consumo en misma fila horizontal
37 with dot.subgraph() as s:
38     s.attr(rank='same')
39     s.node('BI')
40     s.node('DS')
41
42 # Conexiones
43 dot.edges([('F1','I1'), ('F2','I1'), ('F3','I1')])
44 dot.edge('I1', 'I2')
45 dot.edge('I2', 'A1')
46 dot.edge('A1', 'A2')
47 dot.edge('A2', 'A3')
48 dot.edge('A3', 'C1')
49 dot.edge('C1', 'BI')
50 dot.edge('C1', 'DS')
51
52 # [4] Renderizar y mostrar la imagen en Colab
53 dot.render('arquitectura_datos', format='png', cleanup=True)
54 Image('arquitectura_datos.png')
55

```

→ Requirement already satisfied: graphviz in /usr/local/lib/python3.11/dist-packages (0.21)



Lección 2 – Enfoques para almacenamiento y gestión

Objetivo

Definir y justificar las estrategias de almacenamiento y gobernanza de los datos, alineadas al diseño arquitectónico elaborado en la Lección 1.

1 Análisis del esquema arquitectónico diseñado

En la Lección 1 se diseñó una arquitectura modular y escalable con las siguientes capas principales: fuentes, ingestión, integración, almacenamiento, calidad y consumo.

- El **almacenamiento** está estructurado en diferentes niveles de madurez de datos: Data Lake, Data Warehouse y Data Mart.
- El flujo de datos va desde fuentes heterogéneas hacia un Data Lake que captura datos en su estado más crudo, luego a un Data Warehouse para datos organizados y estructurados, y finalmente a Data Marts para consumo específico y analítico.
- Las prácticas de gobernanza se aplican para asegurar calidad, seguridad y trazabilidad.

Esta base permite ahora definir con detalle las zonas de almacenamiento y las políticas de gestión y gobernanza para cada una.

2 Definición de zonas de almacenamiento en Data Lake y su relación con Data Warehouse y Data Mart

Zonas del Data Lake

- **Raw (Bronze):**
 - Aquí se almacenan los datos **crudos**, tal como llegan desde las fuentes.
 - No se aplican transformaciones ni limpiezas, se conserva el dato original para auditoría y trazabilidad.
 - Formatos comunes: JSON, CSV, Parquet, Avro.
 - Ejemplo: logs de eventos, archivos JSON de redes sociales, extractos completos de bases operacionales.

- **Trusted (Silver):**
 - Datos que ya pasaron por procesos de limpieza, filtrado y validación básica.
 - Se corrigen errores evidentes y se estandarizan formatos.
 - Los datos son semi-estructurados y confiables para análisis exploratorios y procesos posteriores.
- **Curated (Gold):**
 - Datos refinados y enriquecidos, listos para análisis avanzados y consumo en BI.
 - Normalizados y modelados según requerimientos de negocio.
 - Esta zona es la fuente para alimentar el Data Warehouse y los Data Marts.

Relación con Data Warehouse y Data Mart

- El **Data Warehouse (DW)** consume datos principalmente de la zona **Curated** del Data Lake, estructurándolos en modelos multidimensionales o normalizados para consultas rápidas y consistentes.
- Los **Data Marts** son subconjuntos especializados del DW, orientados a áreas específicas (ventas, logística, marketing) o tipos de análisis.
- Así se garantiza un almacenamiento jerarquizado, con datos confiables y modelados para distintos tipos de usuarios y necesidades.

3 Tecnologías y servicios sugeridos para cada zona

Zona	Tecnologías / Servicios recomendados	Comentarios
Raw (Bronze)	Amazon S3, Google Cloud Storage, Azure Data Lake Storage, HDFS	Almacenamiento barato y escalable, formatos abiertos.
Trusted (Silver)	AWS Glue, Databricks, Apache Spark, Azure Data Factory	Procesamiento y transformación de datos en batch o streaming.
Curated (Gold)	Snowflake, Google BigQuery, Amazon Redshift, Azure Synapse	Bases optimizadas para consulta analítica, soporte SQL.
Data Warehouse	Google BigQuery, Snowflake, Amazon Redshift, Azure Synapse	Modelo integrado y normalizado para consultas empresariales.
Data Marts	PostgreSQL, Redshift Spectrum, Google BigQuery datasets, Datamarts en DW	Subconjuntos específicos para análisis departamentales.

4 Prácticas de gobernanza y gestión de datos

Trazabilidad y linaje

- Mantener un registro de origen, transformaciones y destino de cada dato mediante sistemas de **Data Lineage** (ej. Apache Atlas, AWS Glue Data Catalog).
- Versionar los datasets y mantener copias históricas para auditoría.

Seguridad y acceso

- Implementar controles de acceso granular basados en roles (RBAC) para proteger datos sensibles.
- Cifrado en reposo y en tránsito (TLS, KMS).
- Auditorías periódicas de acceso y modificaciones.

Calidad y monitoreo

- Establecer reglas automáticas para validar completitud, unicidad, consistencia, precisión y actualidad.
- Integrar sistemas de alertas tempranas ante anomalías o violaciones de calidad.

Disponibilidad y recuperación

- Replicación de datos y backups regulares para asegurar la disponibilidad y resiliencia.
- Definir SLA claros para recuperación ante fallas (RTO, RPO).

Documentación y catálogo

- Documentar metadatos, diccionarios y definiciones de negocio en un catálogo centralizado.
- Facilitar el descubrimiento y entendimiento de los datos por parte de usuarios.

Resumen

La arquitectura definida se apoya en un Data Lake con zonas Raw, Trusted y Curated, que alimentan un Data Warehouse y Data Marts específicos. Cada zona cuenta con tecnologías adecuadas para su función y un esquema de gobernanza que asegura seguridad, trazabilidad y calidad.

Esta estrategia robusta permite manejar el crecimiento y diversidad de datos en la organización, facilitando un análisis confiable y oportuno.

▼ Lección 3 – Calidad de los Datos

Objetivo

Diseñar un plan de aseguramiento de calidad de los datos integrado a la arquitectura definida en las lecciones previas, que permita garantizar datos confiables y preparados para el análisis.

1 Revisión de zonas y flujo arquitectónico (Lección 2)

Recordando de la lección previa, las zonas de almacenamiento definidas:

- **Raw (Bronze):** Datos crudos, sin procesar.
- **Trusted (Silver):** Datos limpios y validados parcialmente.
- **Curated (Gold):** Datos refinados y listos para consumo analítico.
- **Data Warehouse y Data Marts:** Datos estructurados y modelados para reportes y análisis.

El flujo de datos va desde la ingesta hacia estas zonas, con procesos de limpieza, transformación y validación en cada paso.

2 Controles, métricas e indicadores de calidad en cada etapa

Zona	Controles de Calidad	Métricas e Indicadores
Raw (Bronze)	- Integridad del archivo (formatos válidos). - Compleción básica (datos no nulos esenciales). - Captura de metadatos (fecha, origen, versión).	- % de archivos corruptos o incompletos. - % de registros con campos vacíos o nulos. - Tiempo de retraso en la ingesta (latencia).
Trusted (Silver)	- Validaciones de unicidad y duplicados. - Consistencia entre campos (ej. fechas coherentes). - Validación de formatos y tipos.	- % de duplicados detectados. - % de errores de consistencia. - % de registros corregidos o rechazados.
Curated (Gold)	- Reglas de negocio específicas. - Integridad referencial (FK, relaciones). - Calidad semántica (ej. categorización correcta).	- % de cumplimiento de reglas de negocio. - % de violaciones de integridad referencial. - Nivel de precisión de clasificación o etiquetas.
Data Warehouse y Data Marts	- Consistencia con fuentes originales. - Disponibilidad y accesibilidad.	- Tiempo de refresco de datos. - SLA de disponibilidad y tiempos de respuesta.

3 Proceso de monitoreo y remediación

Monitoreo continuo

- Implementar pipelines que validen los datos al llegar a cada zona, con reportes automáticos de métricas de calidad.
- Utilizar herramientas como **Great Expectations**, **Deequ** o servicios nativos en la nube para calidad de datos.
- Dashboards que muestren métricas clave (ej. porcentaje de errores, volumen procesado, latencia).

Alerta y notificación

- Configurar alertas para desviaciones o caídas en la calidad de datos.
- Notificaciones automáticas a equipos responsables para revisión.

Remediación

- Definir procesos automáticos para corregir errores comunes (ej. imputación, eliminación de duplicados).
- Escalar casos complejos a equipos de datos para intervención manual.
- Registrar las acciones tomadas para auditoría y mejora continua.

4 Integración del plan de calidad en la arquitectura general

- El plan de calidad se inserta como una capa transversal en el flujo de datos, especialmente en las transiciones entre las zonas Raw → Trusted → Curated.
- La capa de **Gobierno y Calidad de Datos** (definida en Lección 1) centraliza el control y monitoreo, interactuando con todas las zonas.
- Se integran herramientas de calidad dentro de la capa de integración y almacenamiento para validar datos antes de que avancen a la siguiente etapa.
- Los reportes de calidad están disponibles en la capa de consumo para usuarios técnicos y de negocio.
- Se documentan las reglas de calidad, excepciones y procesos de remediación en el catálogo de datos para transparencia y gobernanza.

Resumen

El plan de calidad abarca controles específicos en cada zona del Data Lake y Data Warehouse, con métricas claras y procesos automáticos de monitoreo y corrección. Este plan asegura que los datos que lleguen al modelo dimensional estén limpios, confiables y alineados a los requisitos del negocio.

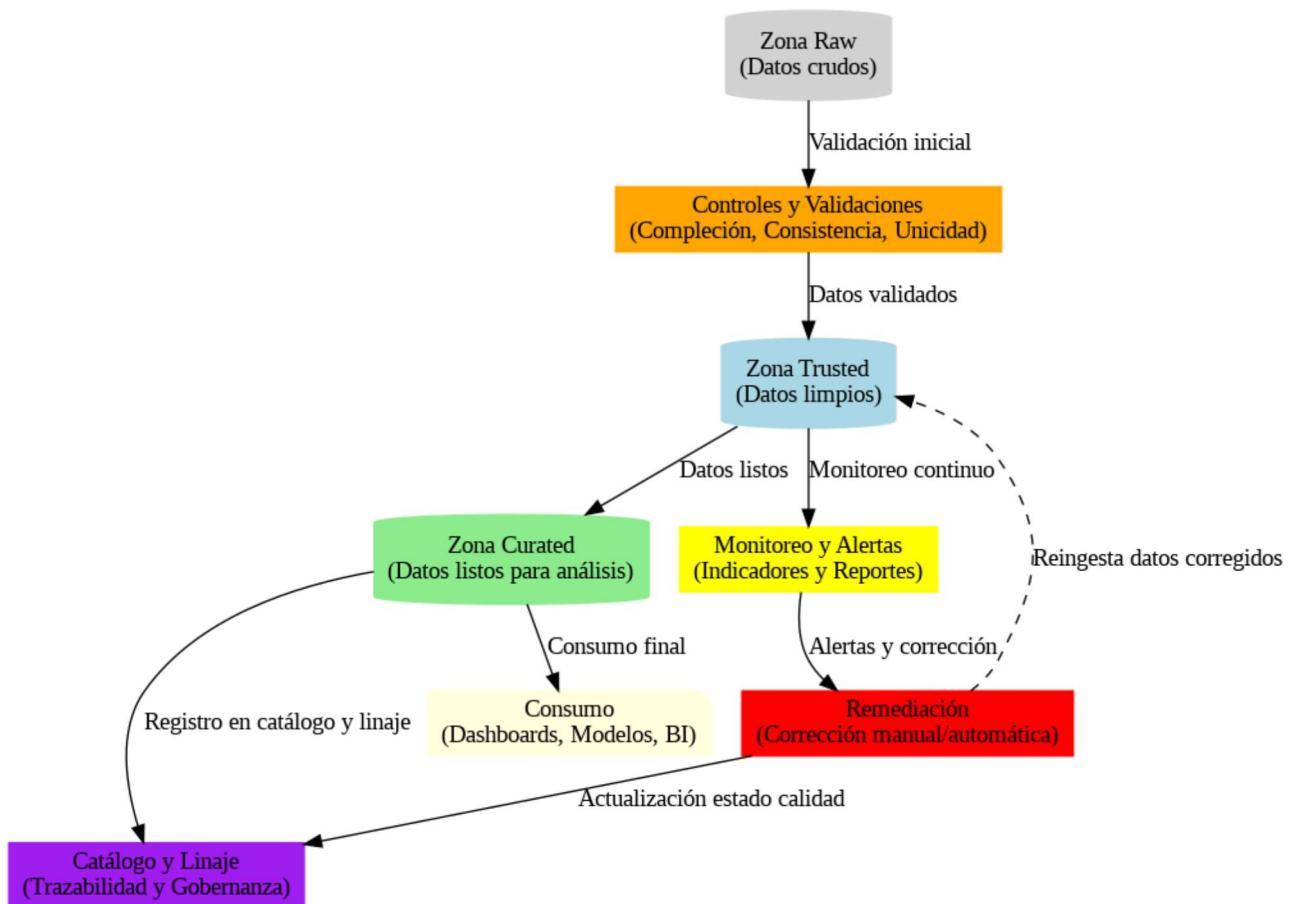
Plan de aseguramiento y remediación de calidad de datos

El siguiente diagrama muestra el proceso de control, monitoreo y remediación de calidad aplicado en el flujo de datos, integrando las zonas definidas en el Data Lake y el Data Warehouse.

Descripción del flujo

1. **Zona Raw (datos crudos):** Los datos llegan inicialmente en su forma original, sin procesar, provenientes de múltiples fuentes estructuradas y no estructuradas.
2. **Controles y Validaciones:** En esta etapa se aplican reglas automáticas para evaluar la completitud, unicidad, consistencia y validez de los datos. Solo los datos que cumplen con estos criterios avanzan.
3. **Zona Trusted (datos limpios):** Aquí se almacenan los datos validados y listos para análisis posteriores, con garantías de calidad inicial.
4. **Monitoreo y Alertas:** Se realiza un seguimiento continuo de la calidad mediante indicadores y reportes. Si se detectan desviaciones o errores, se generan alertas para tomar acción.
5. **Remediación:** El proceso incluye correcciones automáticas y manuales sobre los datos afectados, además de la reingesta de los datos corregidos para asegurar que la calidad se mantenga.
6. **Zona Curated (datos listos para análisis):** Los datos depurados y consolidados se almacenan en esta zona para su uso final en análisis, modelos predictivos y reportes.
7. **Catálogo y Linaje:** Toda la información sobre la calidad, transformaciones y estado de los datos queda registrada para trazabilidad y gobernanza, facilitando auditorías y control.
8. **Consumo:** Los usuarios de negocio, científicos de datos y sistemas de inteligencia acceden a los datos confiables para la toma de decisiones y generación de insights.

```
1 from graphviz import Digraph
2 from IPython.display import Image
3
4 dot = Digraph('ControlCalidadDatos', format='png')
5
6 # Nodos principales
7 dot.node('Raw', 'Zona Raw\n(Datos crudos)', shape='cylinder', style='filled', color='lightgray')
8 dot.node('Trusted', 'Zona Trusted\n(Datos limpios)', shape='cylinder', style='filled', color='lightblue')
9 dot.node('Curated', 'Zona Curated\n(Datos listos para análisis)', shape='cylinder', style='filled', color='lightgreen')
10
11 dot.node('QA', 'Controles y Validaciones\n(Compleción, Consistencia, Unicidad)', shape='box', style='filled', color='orange')
12 dot.node('Mon', 'Monitoreo y Alertas\n(Indicadores y Reportes)', shape='box', style='filled', color='yellow')
13 dot.node('Rem', 'Remediación\n(Corrección manual/automática)', shape='box', style='filled', color='red')
14 dot.node('Cat', 'Catálogo y Linaje\n(Trazabilidad y Gobernanza)', shape='box', style='filled', color='purple')
15 dot.node('Cons', 'Consumo\n(Dashboards, Modelos, BI)', shape='note', style='filled', color='lightyellow')
16
17 # Flujo principal
18 dot.edge('Raw', 'QA', label='Validación inicial')
19 dot.edge('QA', 'Trusted', label='Datos validados')
20 dot.edge('Trusted', 'Mon', label='Monitoreo continuo')
21 dot.edge('Mon', 'Rem', label='Alertas y corrección')
22 dot.edge('Rem', 'Trusted', label='Reingesta datos corregidos', style='dashed')
23
24 dot.edge('Trusted', 'Curated', label='Datos listos')
25 dot.edge('Curated', 'Cat', label='Registro en catálogo y linaje')
26 dot.edge('Curated', 'Cons', label='Consumo final')
27
28 # Feedback al catálogo y gobernanza
29 dot.edge('Rem', 'Cat', label='Actualización estado calidad')
30
31 dot.render('control_calidad_datos', cleanup=True)
32 Image('control_calidad_datos.png')
```



Lección 4 – Modelamiento multidimensional

Objetivo

Diseñar un modelo OLAP (Online Analytical Processing) coherente con la arquitectura de datos y las estrategias de calidad implementadas, para facilitar análisis rápidos y efectivos.

1 Selección del área clave de negocio y datos disponibles

Área seleccionada: Ventas Online

Justificación:

- Es un área fundamental para el e-commerce y retail digital, donde se concentra información crítica para la toma de decisiones comerciales, marketing y logística.
- Datos disponibles de ventas incluyen: transacciones, productos, clientes, tiempo, canales de venta y geografía.

2 Diseño del modelo dimensional alineado con la arquitectura

Tipo de modelo elegido: **Modelo Estrella**

- **Hechos:** Tabla central que contiene medidas numéricicas (ventas, cantidad, ingresos, descuentos).
- **Dimensiones:** Tablas relacionadas que contienen atributos para análisis (cliente, producto, tiempo, canal, ubicación).
- Elegimos el modelo estrella por su simplicidad y eficiencia en consultas OLAP, adecuado para dashboards y análisis rápidos.

Relación con las zonas de almacenamiento

- El modelo dimensional se construye en la zona **Curated (Gold)** del Data Lake y se implementa en el Data Warehouse.
- Los datos provienen de fuentes limpias y validadas (con calidad garantizada en las etapas anteriores).

3 Diagrama del modelo dimensional

Elementos:

- **Tabla de Hechos:**

Nombre	Medidas clave
FactVentas	CantidadVendida, MontoVenta, Descuento, Costo

- **Tablas de Dimensiones:**

Nombre	Atributos principales
DimCliente	ClienteID, Nombre, Edad, Género, SegmentoCliente
DimProducto	ProductoID, NombreProducto, Categoría, Marca, Precio
DimTiempo	Fecha, Día, Mes, Trimestre, Año
DimCanal	CanalID, NombreCanal, TipoCanal
DimUbicacion	UbicacionID, País, Región, Ciudad

4 Decisiones de modelamiento y criterios analíticos

- **Desnormalización:**

- Se optó por un modelo estrella que favorece la desnormalización de dimensiones para mejorar rendimiento en consultas analíticas, sacrificando algo de redundancia.
- Cada dimensión contiene atributos fácilmente consultables sin necesidad de joins complejos.

- **Jerarquías:**

- En la dimensión Tiempo se definen jerarquías naturales (Día → Mes → Trimestre → Año) para análisis en diferentes niveles temporales.
- En la dimensión Ubicación, se establecen jerarquías geográficas (Ciudad → Región → País).

- **Criterios analíticos:**

- Se priorizan métricas clave para medir desempeño comercial (ventas netas, volumen vendido, descuentos aplicados).
- Las dimensiones permiten segmentar análisis por cliente, producto, tiempo, canal de venta y ubicación geográfica, facilitando insights en marketing, inventarios y operaciones.

- **Integración con arquitectura:**

- Los datos usados para poblar las tablas dimensionales provienen de la zona Curated del Data Lake, garantizando que estén limpios y validados.
- El modelo está pensado para alimentar dashboards y modelos predictivos en la capa de consumo.

Resumen

El modelo dimensional estrella para Ventas Online es eficiente y sencillo, alineado con las zonas de almacenamiento y los controles de calidad definidos previamente. Provee la base analítica para la toma de decisiones en el negocio, facilitando análisis rápidos y multidimensionales.

Diagrama gráfico

El siguiente diagrama estrella muestra la relación entre la tabla de hechos y las dimensiones:

```

1 !pip install graphviz
2
3 from graphviz import Graph
4 from IPython.display import Image
5
6 g = Graph('EsquemaEstrella', format='png', engine='neato')
7 g.attr(overlap='false')
8
9 # Nodo central - Tabla de Hechos
10 g.node('HV', 'HECHOS_VENTAS\n(id_tiempo, id_producto, id_cliente, id_canal, id_ubicacion,\ncantidad, monto_total, descuento, costo_envio)', shape='box', style='filled', color='lightblue')
11
12 # Dimensiones alrededor
13 g.node('TI', 'DIM_TIEMPO\n(id_tiempo, dia, mes, trimestre, anio)', shape='box', style='filled', color='lightyellow')
14 g.node('PR', 'DIM_PRODUCTO\n(id_producto, nombre, categoria, marca, precio)', shape='box', style='filled', color='lightyellow')
15 g.node('CL', 'DIM_CLIENTE\n(id_cliente, nombre, sexo, edad, segmento)', shape='box', style='filled', color='lightyellow')
16 g.node('CA', 'DIM_CANAL\n(id_canal, canal, tipo_envio)', shape='box', style='filled', color='lightyellow')
17 g.node('UB', 'DIM_UBICACION\n(id_ubicacion, pais, region, ciudad)', shape='box', style='filled', color='lightyellow')
18
19 # Conexiones hechos - dimensiones
20 g.edge('HV', 'TI')
21 g.edge('HV', 'PR')
22 g.edge('HV', 'CL')
23 g.edge('HV', 'CA')
24 g.edge('HV', 'UB')
25
26

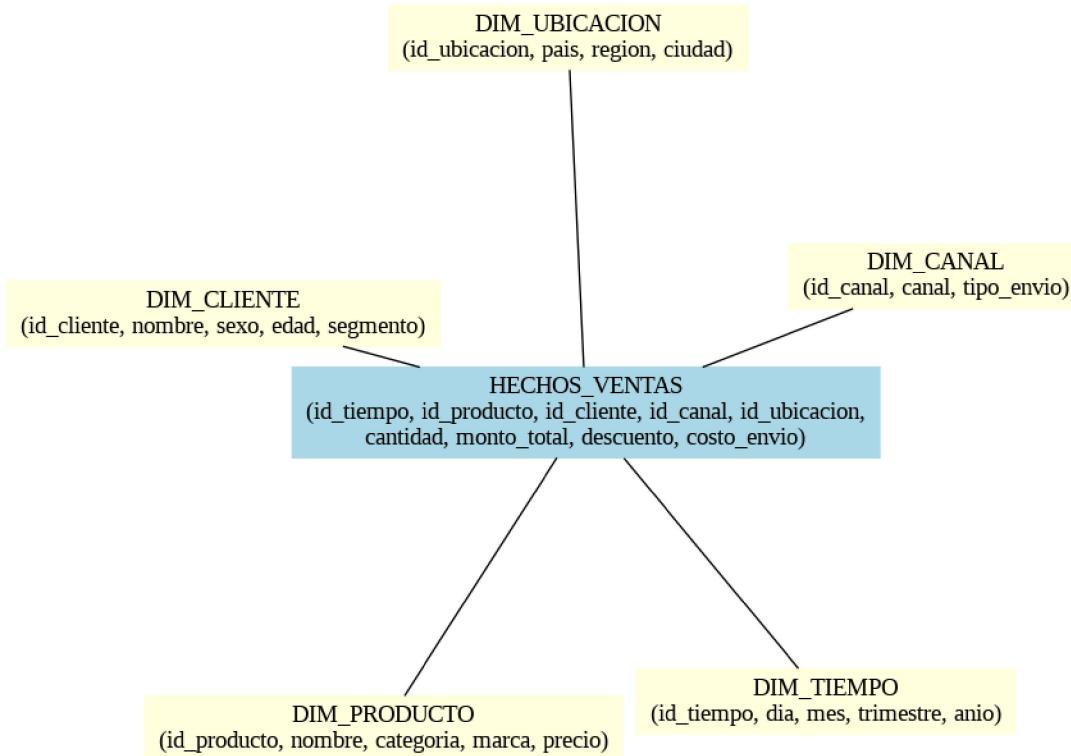
```

```

27 # Render y mostrar imagen
28 g.render('esquema_estrella', cleanup=True)
29 Image('esquema_estrella.png')
30

```

→ Requirement already satisfied: graphviz in /usr/local/lib/python3.11/dist-packages (0.21)



▼ Diagrama Integrador Final Unificado

Este diagrama representa de forma visual e integral la arquitectura completa diseñada para el proyecto de Arquitectura de Datos, mostrando el flujo de los datos desde las fuentes hasta el consumo analítico final, incluyendo las capas de calidad y gobernanza, así como el modelo dimensional.

Componentes del diagrama

- Fuentes de Datos:** Se incluyen diversas fuentes heterogéneas que alimentan el sistema:
 - **Ventas Online e Inventario y Logística** (datos estructurados).
 - **Redes Sociales e Imágenes de Productos** (datos no estructurados).
- Capa de Ingesta:** Los datos se capturan mediante procesos batch (por lotes) y streaming (en tiempo real), asegurando la flexibilidad para distintos tipos y volúmenes de datos.
- Capa de Integración:** A través de procesos ETL/ELT se realiza la limpieza, transformación y unificación de los datos, preparando la información para su almacenamiento organizado.
- Zonas del Data Lake:** Los datos pasan por distintas etapas de madurez:
 - **Raw/Bronze:** datos crudos, sin procesar.
 - **Trusted/Silver:** datos validados y limpios.
 - **Curated/Gold:** datos depurados, listos para análisis y consumo.
- Data Warehouse y Data Marts:** Se almacenan los datos procesados para consultas analíticas, con Data Marts especializados para áreas específicas del negocio.
- Capa de Calidad y Gobernanza:** Esta capa implementa controles automáticos y manuales para asegurar la calidad (integridad, consistencia, completitud) y mantiene el catálogo y linaje de los datos, facilitando la trazabilidad y cumplimiento normativo.
- Modelo Dimensional (OLAP):** El consumo final se estructura mediante un modelo dimensional tipo estrella, con una tabla de hechos (FactVentas) y tablas de dimensiones (Cliente, Producto, Tiempo, Canal, Ubicación), facilitando análisis eficientes y flexibles.

Flujo general

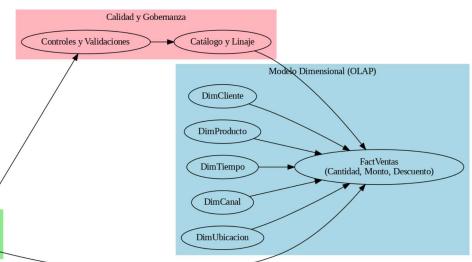
Los datos se originan en las fuentes y son ingeridos en la arquitectura mediante mecanismos batch o streaming. Tras la integración y transformación, avanzan a las zonas del Data Lake para su almacenamiento progresivo. Luego se trasladan al Data Warehouse y Data Marts, donde se aplican controles de calidad y gobernanza. Finalmente, los datos limpios y confiables son consumidos por usuarios y sistemas analíticos a través del modelo dimensional diseñado.

```
1 from graphviz import Digraph
2 from IPython.display import Image
3
4 #dot = Digraph('ArquitecturaDatosIntegrada', format='png', engine='dot')
5 #dot.attr(rankdir='LR', size='10,7')
6
7 # Crear diagrama
8 dot = Digraph('ArquitecturaDatosIntegrada', format='png', engine='dot')
9 dot.attr(rankdir='LR', size='12,8', dpi='300')
10
11 # Fuentes de datos
12 with dot.subgraph(name='cluster_fuentes') as c:
13     c.attr(style='filled', color='lightgrey', label='Fuentes de Datos')
14     c.node('F1', 'Ventas Online\n(Estructurado)', shape='folder', color='blue')
15     c.node('F2', 'Inventario y Logística\n(Estructurado)', shape='folder', color='blue')
16     c.node('F3', 'Redes Sociales\n(No estructurado)', shape='folder', color='blue')
17     c.node('F4', 'Imágenes Productos\n(No estructurado)', shape='folder', color='blue')
18
19 # Capa Ingesta
20 with dot.subgraph(name='cluster_ingesta') as c:
21     c.attr(style='filled', color='lightyellow', label='Capa de Ingesta')
22     c.node('I1', 'Batch ETL')
23     c.node('I2', 'Streaming')
24
25 # Capa Integración
26 with dot.subgraph(name='cluster_integracion') as c:
27     c.attr(style='filled', color='lightgoldenrodyellow', label='Capa de Integración')
28     c.node('P1', 'ETL/ELT\nLimpieza y Transformación')
29
30 # Zonas Data Lake
31 with dot.subgraph(name='cluster_lake') as c:
32     c.attr(style='filled', color='lightcyan', label='Data Lake')
33     c.node('DL1', 'Raw/Bronze\n(Datos crudos)')
34     c.node('DL2', 'Trusted/Silver\n(Datos validados)')
35     c.node('DL3', 'Curated/Gold\n(Datos listos)')
36
37 # Data Warehouse y Data Mart
38 with dot.subgraph(name='cluster_warehouse') as c:
39     c.attr(style='filled', color='lightgreen', label='Data Warehouse / Data Marts')
40     c.node('DW', 'Data Warehouse')
41     c.node('DM', 'Data Marts')
42
43 # Calidad y Gobernanza
44 with dot.subgraph(name='cluster_calidad') as c:
45     c.attr(style='filled', color='lightpink', label='Calidad y Gobernanza')
46     c.node('QA', 'Controles y Validaciones')
47     c.node('CAT', 'Catálogo y Linaje')
48
49 # Modelo Dimensional
50 with dot.subgraph(name='cluster_modelo') as c:
51     c.attr(style='filled', color='lightblue', label='Modelo Dimensional (OLAP)')
52     c.node('F', 'FactVentas\n(Cantidad, Monto, Descuento)')
53     c.node('C', 'DimCliente')
54     c.node('P', 'DimProducto')
55     c.node('T', 'DimTiempo')
56     c.node('CA', 'DimCanal')
57     c.node('UB', 'DimUbicacion')
58
59 # Conexiones entre capas y zonas
60 dot.edge('F1', 'I1')
61 dot.edge('F2', 'I1')
62 dot.edge('F3', 'I2')
63 dot.edge('F4', 'I1')
64
65 dot.edge('I1', 'P1')
66 dot.edge('I2', 'P1')
67
68 dot.edge('P1', 'DL1')
69 dot.edge('DL1', 'DL2')
70 dot.edge('DL2', 'DL3')
71
72 dot.edge('DL3', 'DW')
73 dot.edge('DW', 'DM')
74
```

```

75 dot.edge('DM', 'QA')
76 dot.edge('QA', 'CAT')
77
78 dot.edge('DM', 'F')
79 dot.edge('CAT', 'F')
80
81 # Conexiones modelo dimensional (estrella)
82 dot.edge('C', 'F')
83 dot.edge('P', 'F')
84 dot.edge('T', 'F')
85 dot.edge('CA', 'F')
86 dot.edge('UB', 'F')
87
88 dot.render('arquitectura_integrada', cleanup=True)
89 Image('arquitectura_integrada.png')
90

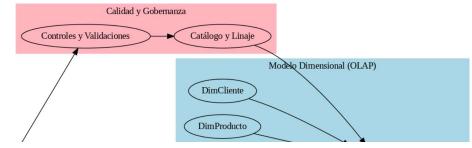
```



```

1 #Descargar la imagen para mejor visualización
2 from IPython.display import Image
3 Image(filename='arquitectura_integrada.png')

```



Fuentes de Datos

Ventas Online (Estructurado)

Inventario y Logística (Estructurado)

Imagenes Productos (No estructurado)

Redes Sociales (No estructurado)

Capa de Ingesta

Batch ETL

Streaming

ETL/ELT Limpieza y Transformación

Limpieza y Transformación

Data Lake

Raw/Bronze (Datos crudos)

Trusted/Silver (Datos validados)

Curated/Gold (Datos listos)

Data Warehouse / Data Marts

Data Warehouse

Data Marts

Calidad y Gobernanza

Controles y Validaciones

Catálogo y Lineaje

Modelo Dimensional (OLAP)

DimCliente

DimProducto

DimTiempo

DimCanal

DimUbicacion

FactVentas (Cantidad, Monto, Descuento)

Cantidad, Monto, Descuento