Lección 3: Calidad de los Datos

1. Revisión de Zonas y Flujo Arquitectónico

La arquitectura de datos definida en las lecciones anteriores establece un flujo de datos a través de diferentes zonas (Raw, Trusted, Curated) hasta llegar al Data Warehouse y Data Marts. Para garantizar el valor y la confiabilidad de los datos a lo largo de este flujo, es fundamental implementar un plan integral de calidad de datos.

Este plan debe adaptarse a las características específicas de cada zona y a los requisitos particulares del negocio de e-commerce y retail digital.

2. Marco de Calidad de Datos

2.1. Dimensiones de Calidad

Para evaluar y monitorear sistemáticamente la calidad de los datos, se utilizarán las siguientes dimensiones:

Dimensión	Descripción	Relevancia para E-commerce
Exactitud	Grado en que los datos representan correctamente el valor real	Crítica para precios, inventario, información de clientes
Completitud	Presencia de todos los valores necesarios	Importante para perfiles de clientes, atributos de productos
Consistencia	Coherencia entre datos relacionados	Vital para inventario vs. ventas, precios en diferentes canales
Actualidad	Grado de actualización de los datos	Crucial para inventario, precios, promociones
Unicidad	Ausencia de duplicados	Esencial para catálogo de productos, base de clientes
Validez	Conformidad con reglas de negocio y formatos	Necesaria para direcciones, información fiscal, códigos de productos
Integridad	Mantenimiento de relaciones entre datos	Importante para relaciones pedido-cliente, producto-categoría

2.2. Niveles de Aplicación

La calidad de datos debe aplicarse en diferentes niveles:

- **Nivel de campo**: Validación de tipos de datos, formatos, rangos
- **Nivel de registro**: Validación de relaciones entre campos
- **Nivel de tabla/colección**: Validación de agregaciones, distribuciones
- **Nivel de sistema**: Validación de consistencia entre sistemas y dominios de datos

3. Controles, Métricas e Indicadores por Zona

3.1. Zona Raw (Data Lake)

Controles:

- Verificación de completitud de archivos
- Validación de estructura básica
- Detección de anomalías en volumen o frecuencia

Métricas e Indicadores:

- Recuento de registros vs. sistema fuente
- Porcentaje de archivos recibidos completos
- Tiempo de ingesta
- Tamaño de datos vs. promedio histórico

Implementación Técnica:

```
# Ejemplo de control de completitud
def validate_file_completeness(file_path, expected_count):
    actual_count = get_record_count(file_path)
    if actual_count < expected_count * 0.95:  # Tolerancia del 5%
        raise QualityException(f"File {file_path} has {actual_count} records,
expected {expected_count}")
```

3.2. Zona Trusted (Data Lake)

Controles:

- Validación de tipos de datos
- Detección de valores nulos en campos obligatorios
- Validación de rangos y formatos
- Detección de duplicados
- Verificación de integridad referencial básica

Métricas e Indicadores:

- Porcentaje de valores nulos por campo
- Porcentaje de valores fuera de rango
- Número de duplicados identificados
- Tasa de rechazo de registros
- Distribución estadística de valores clave

Implementación Técnica (usando Great Expectations):

```
# Definición de expectativas para datos de productos
expectations = {
    "product_id": {
        "not_null": True,
        "unique": True
    },
    "price": {
        "not_null": True,
        "min_value": 0.01,
        "max_value": 99999.99
    },
    "stock_quantity": {
        "not_null": True,
        "min_value": 0
    },
    "category_id": {
        "not_null": True,
        "in_reference": "categories.category_id"
    }
}
```

3.3. Zona Curated (Data Lake)

Controles:

- Validación de reglas de negocio complejas
- Verificación de agregaciones y cálculos
- Validación de consistencia entre dominios
- Detección de anomalías en tendencias

Métricas e Indicadores:

- Consistencia de KPIs entre diferentes agregaciones
- Desviación de métricas respecto a periodos anteriores
- Completitud de dimensiones de análisis
- Coherencia de jerarquías

Implementación Técnica:

```
-- Verificación de consistencia entre ventas diarias y mensuales
SELECT
    DATE_TRUNC('month', date) AS month,
    SUM(daily_sales) AS sum_daily,
    monthly_sales,
    ABS(SUM(daily_sales) - monthly_sales) / monthly_sales AS deviation
FROM
    curated.daily_sales d
JOIN
    curated.monthly_sales m ON DATE_TRUNC('month', d.date) = m.month
GROUP BY
    DATE_TRUNC('month', date),
    monthly_sales
HAVING
    ABS(SUM(daily_sales) - monthly_sales) / monthly_sales > 0.01; -- Tolerancia del
1%
```

3.4. Data Warehouse

Controles:

- Validación de integridad referencial completa
- Verificación de conformidad dimensional
- Validación de consistencia temporal
- Verificación de balances y reconciliaciones

Métricas e Indicadores:

- Integridad de relaciones entre hechos y dimensiones
- Coherencia de jerarquías dimensionales
- Precisión de cálculos de métricas clave
- Freshness de datos en tablas de hechos

Implementación Técnica:

```
-- Verificación de integridad referencial
SELECT
    COUNT(*) AS orphaned_records
FROM
    fact_sales s
LEFT JOIN
    dim_product p ON s.product_id = p.product_id
```

```
WHERE
    p.product_id IS NULL;
```

3.5. Data Marts

Controles:

- Validación de métricas específicas de negocio
- Verificación de cálculos departamentales
- Validación de KPIs con fuentes alternativas

Métricas e Indicadores:

- Precisión de métricas de negocio
- Coherencia con informes externos
- Usabilidad de datos para análisis

4. Proceso de Monitoreo y Remediación

4.1. Monitoreo Proactivo

Se implementará un sistema de monitoreo proactivo que incluya:

1. **Dashboards de calidad**: Visualización en tiempo real de métricas clave de calidad
2. **Alertas automáticas**: Notificaciones cuando las métricas superen umbrales definidos
3. **Reportes periódicos**: Informes diarios/semanales sobre el estado de la calidad de los datos
4. **Perfiles de datos**: Generación automática de perfiles estadísticos para detectar anomalías

Implementación Técnica:

- Plataforma de observabilidad: Grafana, Datadog
- Gestión de alertas: PagerDuty, Opsgenie
- Perfilado de datos: AWS Glue DataBrew, Informatica Data Quality

4.2. Proceso de Remediación

Se establecerá un proceso estructurado para abordar problemas de calidad:

1. **Detección**: Identificación automática o manual de problemas
2. **Clasificación**: Categorización por severidad e impacto
3. **Investigación**: Análisis de causa raíz
4. **Corrección**: Implementación de soluciones
    - Corrección en origen (preferible)
    - Corrección en flujo
    - Corrección en destino (último recurso)
5. **Verificación**: Validación de la efectividad de la corrección
6. **Prevención**: Mejora de controles para evitar recurrencia

Workflow de Remediación:

```
graph TD
    A[Detección de Problema] --> B[Clasificación]
    B --> C{Severidad}
    C -->|Alta| D[Alerta Inmediata]
    C -->|Media| E[Ticket Prioritario]
    C -->|Baja| F[Ticket Regular]
    D --> G[Análisis Causa Raíz]
    E --> G
    F --> G
```

```
    G --> H[Determinar Solución]
    H --> I{Tipo de Corrección}
    I -->|En Origen| J[Corregir Sistema Fuente]
    I -->|En Flujo| K[Modificar Transformación]
    I -->|En Destino| L[Corregir Datos Destino]
    J --> M[Verificar Corrección]
    K --> M
    L --> M
    M --> N[Actualizar Documentación]
    N --> O[Mejorar Controles Preventivos]
```

4.3. Gestión de Incidentes de Calidad

Se implementará un sistema formal de gestión de incidentes que incluya:

- **Registro centralizado**: Documentación de todos los incidentes de calidad
- **Clasificación de severidad**: Basada en impacto al negocio
- **SLAs de resolución**: Tiempos máximos de respuesta según severidad
- **Métricas de incidentes**: Seguimiento de frecuencia, tiempo de resolución y efectividad

5. Integración del Plan de Calidad en la Arquitectura

5.1. Puntos de Control en el Flujo de Datos

El plan de calidad se integrará en puntos estratégicos del flujo de datos:

1. **Pre-ingesta**: Validación antes de aceptar datos en el sistema
2. **Post-ingesta**: Verificación después de almacenar en zona Raw
3. **Pre-transformación**: Validación antes de procesar hacia zona Trusted
4. **Post-transformación**: Verificación después de cada etapa de transformación
5. **Pre-carga**: Validación antes de cargar en Data Warehouse
6. **Post-carga**: Verificación de integridad después de cargar

5.2. Herramientas y Tecnologías

La implementación del plan de calidad utilizará las siguientes herramientas:

Función	Herramientas Recomendadas
Perfilado de datos	AWS Glue DataBrew, Informatica Data Quality
Validación	Great Expectations, Deequ, DBT tests
Monitoreo	Datadog, Grafana, Prometheus
Observabilidad de datos	Monte Carlo, Databand, Acceldata
Gestión de incidentes	JIRA, ServiceNow
Linaje y metadatos	Apache Atlas, Collibra

5.3. Implementación en Infraestructura

La calidad de datos se implementará como componentes transversales en la arquitectura:

- **Servicios de validación**: Microservicios dedicados a validar datos
- **Componentes de monitoreo**: Agentes para recopilar métricas
- **Capa de observabilidad**: Dashboards y alertas
- **Catálogo de reglas**: Repositorio centralizado de reglas de calidad
- **API de calidad**: Interfaces para integrar validaciones en pipelines

6. Métricas de Éxito del Plan de Calidad

Para evaluar la efectividad del plan de calidad, se utilizarán las siguientes métricas:

1. **Reducción de incidentes**: Disminución en número y severidad de incidentes relacionados con datos
2. **Tiempo de detección**: Reducción en tiempo para detectar problemas
3. **Cobertura de calidad**: Porcentaje de dominios de datos con controles implementados
4. **Satisfacción de usuarios**: Percepción de confiabilidad de los datos
5. **Costo de mala calidad**: Estimación de costos evitados por mejora en calidad

7. Caso de Estudio: Calidad en Datos de Inventario

Para ilustrar la aplicación del plan de calidad, consideremos el flujo de datos de inventario:

7.1. Controles en Zona Raw

- Verificación de recepción de actualizaciones diarias
- Validación de formato de archivo
- Comprobación de integridad estructural

7.2. Controles en Zona Trusted

- Validación de SKUs contra catálogo maestro
- Detección de cantidades negativas o anormalmente altas
- Verificación de ubicaciones contra maestro de almacenes

7.3. Controles en Zona Curated

- Cálculo y validación de métricas de rotación
- Reconciliación con datos de ventas
- Detección de anomalías en tendencias de stock

7.4. Controles en Data Warehouse

- Validación de agregaciones por categoría, región, etc.
- Verificación de consistencia con dimensiones de productos
- Reconciliación con datos financieros

8. Implementación Progresiva

Se recomienda implementar el plan de calidad de forma progresiva:

Fase 1 (0-3 meses)

- Establecer perfiles de datos para fuentes principales
- Implementar controles básicos en zonas Raw y Trusted
- Configurar monitoreo de completitud y validez

Fase 2 (3-6 meses)

- Extender controles a zona Curated
- Implementar validaciones de reglas de negocio
- Configurar dashboards de calidad

Fase 3 (6-12 meses)

- Integrar controles en Data Warehouse y Data Marts
- Implementar detección avanzada de anomalías
- Automatizar procesos de remediación

9. Conclusiones

Un plan robusto de calidad de datos es esencial para garantizar que la arquitectura de datos propuesta cumpla su objetivo de facilitar la toma de decisiones basada en información confiable. La implementación de controles adaptados a cada zona del flujo de datos, junto con procesos de monitoreo y remediación, asegurará que los datos que lleguen a los usuarios finales sean precisos, completos y actualizados.

Para el contexto específico de e-commerce y retail digital, donde las decisiones comerciales dependen críticamente de datos actualizados sobre inventario, precios, comportamiento de clientes y tendencias de ventas, la calidad de datos no es solo un requisito técnico sino un imperativo de negocio.