INFORME FINAL – DISEÑO DE ARQUITECTURA DE DATOS MODERNA

1. INTRODUCCIÓN

El presente informe describe el desarrollo e implementación de una arquitectura de datos moderna para la gestión, procesamiento y análisis de información proveniente de múltiples fuentes. El objetivo principal fue diseñar un flujo de datos eficiente que garantizara calidad, integridad y disponibilidad para su posterior explotación analítica.

El proyecto integra procesos ETL, almacenamiento estructurado, modelado multidimensional y visualización de datos, siguiendo las mejores prácticas de ingeniería de datos.

2. DISEÑO DE LA ARQUITECTURA

La arquitectura propuesta sigue un enfoque moderno y escalable, basado en un flujo datalake → ETL → data warehouse → datamart → visualización:

- Capa de ingesta: Recepción de archivos CSV crudos (clientes, productos y ventas) en el datalake.

- Capa de procesamiento: Aplicación de scripts ETL para diagnosticar, limpiar, transformar y unificar los datos.

- Capa de almacenamiento: Organización en un modelo multidimensional optimizado para análisis OLAP.

- Capa de visualización: Representación de métricas clave mediante herramientas gráficas.

3. PROCESOS ETL Y ALMACENAMIENTO

Se desarrollaron scripts Python para:

1. Diagnóstico de calidad: Antes de aplicar la limpieza, se ejecuta un script de diagnóstico sobre cada dataset en datos_crudos. El diagnóstico estima completitud, validez (numérica, de fecha y de texto) y duplicados por columna; además, evalúa reglas de negocio simples (por ejemplo, cantidad >= 0, monto >= 0, fechas en rango). Los resultados se guardan en reportes_datos_crudos/.
   Tras la limpieza y transformación, se repite el diagnóstico sobre datos_procesados para cuantificar la mejora (por ejemplo, la columna fecha pasa a 100% de validez).
   Este control bidireccional asegura trazabilidad del cambio y habilita umbrales de calidad que pueden automatizar decisiones (continuar/fallar el pipeline).
2. Limpieza de datos: eliminación de duplicados, validación de tipos y formatos (especialmente fechas), normalización de texto.
3. Transformación: creación de dimensiones (clientes, productos,tiempo) y tabla de hechos (ventas).

4. Carga: almacenamiento en formato limpio dentro de la carpeta de datos procesados y preparación para su inserción en el datamart.El almacenamiento final sigue un modelo en estrella, permitiendo consultas rápidas y agregaciones eficientes.

4. MODELO MULTIDIMENSIONAL

El modelo implementado se compone de:

- Tabla de hechos: hecho_ventas (id_venta, id_cliente, id_producto, id_tiempo, cantidad, total).

- Dimensiones:

- dim_cliente (id_cliente, nombre_cliente,edad, ubicación, categoria)

- dim_producto (id_producto,nombre_producto,categoria,proveedor)

- dim_tiempo (id_tiempo,fecha,anio,mes,dia)

Este diseño facilita análisis como ventas por período, por región, por categoría de producto, etc.

5. EVALUACIÓN DE CALIDAD DE DATOS

Se aplicaron diagnósticos automáticos sobre completitud, validez y duplicados:

- Completitud: 100% en todos los campos tras limpieza.

- Validez: Corrección de formatos de fecha y validación de rangos temporales.

- Duplicados: Eliminación de registros repetidos por identificador.

El campo más conflictivo inicialmente fue fecha, pero se corrigió hasta alcanzar el 100% de validez.

6. VISUALIZACIÓN Y ANÁLISIS

Se generaron visualizaciones usando Matplotlib y Tabulate para:

- Ventas por año según categoría.

- Ventas mensuales por año y categoría

Las métricas y gráficos permiten a la dirección identificar tendencias, estacionalidad y oportunidades de negocio.

7. CONCLUSIONES Y RECOMENDACIONES

- La arquitectura implementada asegura un flujo de datos limpio, estructurado y listo para análisis avanzado.

- El modelo multidimensional permite ampliar fácilmente las dimensiones y métricas.

- Se recomienda en el futuro:

1. Automatizar la ingesta con pipelines programados.
2. Integrar almacenamiento en la nube para escalabilidad.
3. Incorporar dashboards interactivos con herramientas como Power BI o Tableau.

Arquitectura de datos