ARQUITECTURA

1. ETL y Data Integration:
- Utilizar un proceso ETL para extraer datos de las fuentes F1, F2 y F3. Transformar los datos según sea necesario y carga los datos consolidados en un Data Lake o Data Warehouse.

- Para F1 (CRM Propietario):
  Integración de Datos:
  Integrar datos del CRM con otros sistemas. Puede hacerse con APIs, conectores específicos o procesos de extracción de datos programados.

- Para F2 (RDBMS SQL Server):
  Optimización de Consultas:
  Debido a que F2 almacena transacciones, asegurarse de optimizar las consultas SQL para garantizar un rendimiento eficiente, especialmente si la tabla de transacciones es grande.

- Para F3 (RDBMS PostgreSQL):
  Diseño de Esquema:
  Asegurarse de tener un diseño de esquema eficiente para las transacciones de los productos restantes. Considerar las relaciones y la normalización adecuada.

1.1 Conexiones Seguras:
- Utilizar conexiones seguras (por ejemplo, HTTPS) para la extracción de datos desde las fuentes. Asegurarse de que las conexiones estén cifradas y autenticadas.

1.2 Validación de Entradas:
- Implementar validaciones rigurosas de las entradas para prevenir ataques de inyección, especialmente si se extraen datos de fuentes externas.

1.3 Seguridad del Servidor ETL:
- Asegurarse de que el servidor ETL esté configurado de forma segura. Aplicar actualizaciones de seguridad y seguir las mejores prácticas de configuración.

2. Data Lake / Data Warehouse:
- Almacenar los datos consolidados en un Data Lake o Data Warehouse centralizado. Usar herramientas como Amazon Redshift, Google BigQuery, Apache Hive o Snowflake.

2.1 Cifrado de Datos:
- Utilizar cifrado para almacenar datos tanto en reposo como en tránsito.

2.2 Control de Acceso:
- Implementar controles de acceso rigurosos para limitar el acceso a los datos almacenados. Definir roles y permisos de manera granular.

2.3 Auditoría:
- Establecer un sistema de auditoría para registrar y supervisar las actividades relacionadas con los datos almacenados.

2.4 Calidad de Datos:
- Implementar procesos de limpieza y validación para garantizar la precisión de la información demográfica y de contacto.

3. SQL Access Layer:
- Implementar una capa de acceso SQL que permita a los usuarios del área operativa realizar consultas mediante lenguaje SQL estándar. Utilizar herramientas como Apache Drill, Presto o el propio sistema de consultas SQL del Data Warehouse elegido.

3.1 Firewalls y Filtrado de Tráfico:
- Configurar firewalls y realizar filtrado de tráfico para permitir solo conexiones seguras y autorizadas a la capa de acceso SQL.

3.2 Autenticación y Autorización:
- Implementar mecanismos fuertes de autenticación y autorización para garantizar que solo usuarios autorizados puedan acceder a los datos.

4. Herramientas de Ciencia de Datos:
- Utilizar herramientas de ciencia de datos como Jupyter Notebooks con bibliotecas como scikit-learn, TensorFlow o PyTorch para aplicar algoritmos de detección de patrones como clustering.

- Considerar el uso de Apache Spark para realizar transformaciones complejas y análisis avanzado de datos.

4.1 Entornos Seguros:
- Asegurarse de que los entornos de ciencia de datos estén configurados de manera segura. Limitar el acceso y la ejecución de código no autorizado.

4.2 Seguridad de Modelos:
- Protege los modelos de ciencia de datos y los resultados de análisis. Limitar el acceso a los modelos entrenados y a los datos de salida.

5. Orquestación y Automatización:
- Utilizar herramientas de orquestación como Apache Airflow para automatizar y programar la ejecución de procesos ETL, actualizaciones de datos y tareas de limpieza.

5.1 Credenciales Seguras:

- Gestionar de forma segura las credenciales utilizadas en el proceso de orquestación. Evitar almacenar credenciales en texto plano y utilizar soluciones de administración de secretos.

5.2 Monitoreo Continuo:
- Implementar un sistema de monitoreo continuo para detectar actividades anómalas o intrusiones en el proceso de orquestación.

6. Seguridad:
- Implementar medidas de seguridad robustas en todos los niveles como el cifrado de datos, control de acceso a nivel de usuario, auditoría de actividades y cumplimiento de regulaciones de privacidad.

6.1 Canales Seguros:
- Asegurar que la comunicación entre los diferentes componentes del sistema sea segura y esté cifrada.

6.2 Autenticación Mutua:
- Considerar la autenticación mutua entre componentes críticos para garantizar la identidad de cada parte del flujo de datos.

7. Monitoreo y Mantenimiento:
- Establecer un sistema de monitoreo para supervisar el rendimiento del sistema, la integridad de los datos y detectar posibles problemas. Realizar mantenimiento regular para garantizar la eficiencia y calidad de los datos.

8. Documentación y Metadatos:
- Mantener una documentación completa de la arquitectura, los procesos ETL, la estructura de datos y los metadatos. Facilitar a los usuarios del área operativa y al equipo de ciencia de datos entender y utilizar los datos de manera efectiva.

8.1 Capacitación del Personal:
- Proporcionar capacitación continua a los usuarios y al personal involucrado en el manejo de datos sobre prácticas seguras y la importancia de la seguridad de los datos.

8.2 Concientización de Amenazas:
- Mantener al personal informado sobre las amenazas de seguridad actuales y promover una cultura de seguridad.

9. Escalabilidad:
- Diseñar la arquitectura para ser escalable, de manera que pueda manejar crecimiento en volumen de datos y demandas de usuarios.

10. Gobierno de Datos:

- Establecer políticas y normativas de gobierno de datos que incluyan la seguridad de extremo a extremo, calidad de los datos, la estandarización y la gestión de cambios. Asegurase de que se sigan prácticas de seguridad en todas las fases del ciclo de vida de los datos.

10.1 Plan de Respuesta a Incidentes:
- Desarrollar un plan de respuesta a incidentes que incluya procedimientos para manejar y mitigar posibles violaciones de seguridad.

10.2 Cumplimiento Normativo:
- Asegurarse de cumplir con los requisitos legales y normativos relacionados con la seguridad y privacidad de los datos.

---

1. Metadata Management:

1.1 Catálogo de Metadatos:
- Utilizar un catálogo de metadatos centralizado que almacene información sobre las fuentes de datos, definiciones de campos, transformaciones, y cualquier otra información relevante. Utilizar herramientas como Apache Atlas, Collibra o AWS Glue DataBrew.

1.2 Lineage:
- Implementar seguimiento de linaje para entender cómo los datos fluyen a través del sistema desde la fuente hasta el destino. Esto ayuda a entender las dependencias y el impacto de los cambios.

1.3 Versionado de Metadatos:
- Implementar un sistema de versionado para los metadatos. Cada vez que haya cambios en la definición de datos o procesos, actualizar el catálogo de metadatos y registrar la versión correspondiente.

1.4 Auditoría de Metadatos:
- Configurar la auditoría en el catálogo de metadatos para registrar quién hizo qué cambios y cuándo.

2. Control de Cambios del Pipeline:

2.1 Control de Versiones del Código:
- Utilizar sistemas de control de versiones (por ejemplo, Git) para controlar el código de tus procesos ETL, scripts y configuraciones.

2.2 Despliegues Controlados:
- Establecer prácticas de despliegue controladas para tus procesos ETL. Utiliza entornos de desarrollo, prueba y producción para minimizar los riesgos.

2.3 Registro de Cambios:

- Llevar un registro de los cambios en el pipeline, incluyendo modificaciones en las transformaciones, esquemas de datos y conexiones a fuentes. Utilizar un sistema de control de cambios o un registro de cambios específico.

3. Almacenamiento de Datos de Metadatos:

3.1 Base de Datos de Metadatos:
- Almacenar los metadatos en una base de datos dedicada, preferiblemente de fácil consulta. Se puede utilizar bases de datos relacionales o NoSQL según las necesidades. PostgreSQL, MySQL o MongoDB son opciones comunes.

3.2 Data Lake:
- Almacenar información detallada de linaje y metadatos en un Data Lake para permitir análisis avanzados. Se puede utilizar servicios como AWS Lake Formation o Azure Data Lake Storage.

3.3 Servicios de Catálogo de Datos:
- Algunas plataformas y servicios de nube ofrecen servicios integrados de catálogo de datos que facilitan el almacenamiento y gestión de metadatos. Ejemplos incluyen AWS Glue DataBrew y Azure Purview.

4. Documentación:

4.1 Wiki o Documentación Interna:
- Mantener documentación interna actualizada que describa los cambios en el pipeline, estructuras de datos, y cualquier otra información relevante. Se puede utilizar wikis internos o herramientas de documentación colaborativa.

5. Automatización y Monitorización:

5.1 Automatización del Registro:
- Automatizar la actualización del catálogo de metadatos y del registro de cambios en cada ejecución del pipeline.

5.2 Alertas y Monitoreo:
- Configurar alertas y monitoreo para detectar cualquier anomalía en el registro de cambios o en la actualización de metadatos.

6. Seguridad:

6.1 Seguridad de Metadatos:
- Implementar medidas de seguridad para proteger la integridad y confidencialidad de los metadatos, incluyendo controles de acceso y cifrado si es necesario.