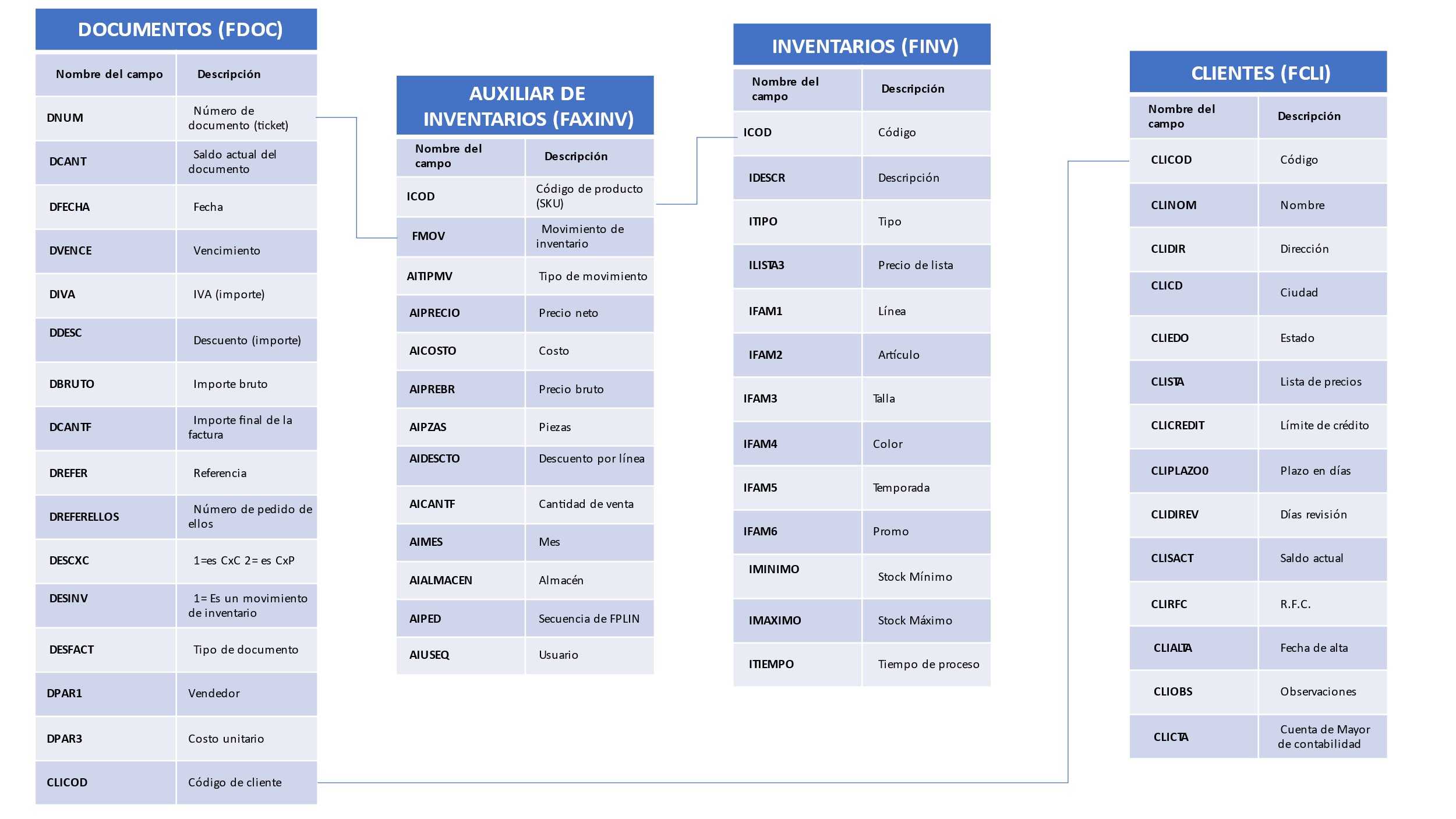
1. **Construcción de query en SQL**

**Instrucciones:** Con el diagrama de base de datos compartido, crea un query para obtener la información de ventas del cliente, incluyendo los siguientes campos:

* # de ticket
* Fecha
* Monto (DCANT) con y sin IVA
* Vendedor
* Cliente
* Productos comprados (SKU)
* Descripción
* Cantidad (unidades)
* Almacén que vendió
* Precio de lista
* Talla
* Color
* Temporada



**Entregable:** Código escrito con la consulta solicitada. Puedes asumir que las tablas ya están creadas en la base de datos.

1. **Desarrollo de visualizaciones en Business Intelligence**

**Instrucciones:** Con el archivo “Fuente para dashboard”, desarrolla un dashboard que permita responder la mayor cantidad de preguntas de la siguiente lista, para dar una recomendación al cliente acerca de su operación.

**Preguntas:**

1. ¿Qué materiales me voy a quedar sin inventario y en qué fecha me quedo sin inventario?
2. ¿Cuáles productos son los más importantes en venta?
3. ¿Cuáles productos tengo riesgo de quedarme con mucho inventario al final del año?
4. ¿Hay mucha variación entre la venta de años pasados y el forecast?
5. ¿Cuál sería la cobertura del inventario contra la versión 1 del forecast y con la versión 2 del forecast?
6. Después de haber presentado esta primera versión del dashboard, ¿qué pasos recomendarías seguir para liberarlo para el uso en el día a día por parte del cliente?
7. ¿Cómo esperarías que fuera la interacción entre IBSO y el cliente por los siguientes 12 meses?

**Entregable:** Dashboard (En Tableau / Power BI / Qlik) con las tablas, gráficas y/o visualizaciones que justifiquen los resultados y recomendaciones propuestas. El dashboard debería seguir una estructura lógica y deberá permitir al usuario entenderlo sin mucha capacitación.

**Referencias para estructura del dashboard:**

<https://medium.com/@mokkup/8-essential-dashboard-design-principles-for-effective-data-visualization-40653c5fd135>

Tableau: <https://data.ucop.edu/support-training/tableau-files/building_effective_dashboards.pdf>

1. **Ejercicios – Python**

**Instrucciones:** Con las preguntas mostradas, trata de desarrollar los códigos para responder el mayor número de ejercicios usando Python.

**Entregable:** Códigos en Python mostrando los resultados de los ejercicios.

**Ejercicio 1:**

1. Crea un diccionario llamado datos\_climaticos donde cada clave es el nombre de una ciudad y el valor es una lista de temperaturas (en grados Celsius) registradas a lo largo de una semana.
2. Para cada ciudad, calcula la temperatura promedio, la temperatura máxima y la mínima de la semana.
3. Determina cuál fue la ciudad con la temperatura promedio más alta y la más baja durante la semana.

**Ejercicio 2:**

Asigna a cada letra minúscula un valor, desde 1 para la 'a' hasta 26 para la 'z'.

Crea una función que pida al usuario una cadena de letras minúsculas y responde la suma de los valores de las letras en la cadena. (Ejemplo: hola = 8 + 15 + 12 + 1 = 36).

Si el usuario te da un número o una letra mayúscula, pídele que lo cambie (Input: Hola. Output: Cambia a minúscula la letra “H” en la posición 1. Input: int2. Output: Cambia el número en la posición 4 por una letra minúscula).

**Ejercicio 3:**

1. Extraer la información del csv **Prueba\_Promociones**

2. ⁠Generar un código donde el usuario pueda ingresar las siguientes variables

- fecha inicio (convertir a datetime)

- ⁠fecha fin (convertir a datetime)

- ⁠categoría (validar que sea string)

- ⁠uso (validar que sea string)

- ⁠sku (no permitir al usuario avanzar si no ingreso un valor string en el campo de SKU)

- ⁠% (validar que sea número decimal)

- ⁠inventario inicial (validar que sea entero)

3. Generar una nueva columna donde se coloque el # de semana correspondiente de la fecha

4. ⁠Con él % ingresado por el usuario, impactar ese porcentaje como crecimiento (columna de piezas) para todos los datos que cumplan con las siguientes condiciones:

- estén dentro del rango de fecha seleccionado. Si el usuario no coloca fecha fin entonces desde fecha inicio hasta el final, si el usuario no coloca fecha inicio entonces desde la fecha fin hasta el principio y si no coloca ninguna, a todas las fechas disponibles

- ⁠la columna **modelo** sea diferente a “real”

- ⁠la columna **uso** sea igual al valor ingresado por el usuario, en caso de que esté vacío ese campo, la columna categoría sea igual al valor ingresado por el usuario. En caso de que ambas vengan vacías aplicar a todo

5. Una vez aplicado el incremento %, generar un nuevo dataframe en donde solo venga información del SKU seleccionado y se pueda visualizar el consumo de inventario, es decir, el inventario inicial colocado por el usuario será la cantidad inicial el 01-01-2024 y de ahí se tendrá que ir consumiendo (restando) la venta proyectada (**modelo** diferente de real). Se deberá imprimir la primera fecha en la que se vuelve negativo el inventario

**Ejercicio 4: Recomendador de libros**

Fuente: <https://developer.nytimes.com/docs/books-product/1/overview>

**Elementos a utilizar:**

1. Crea una cuenta de Developer en la página de New York Times para tener un API key (<https://developer.nytimes.com/>)
2. Usa el “Books API”.

**¿Qué debe poder hacer tu código?**

1. Pedirle al usuario decidir qué lista de “Best Sellers” quiere consultar.
2. Poder escoger si quiere ver los “Best Sellers” actuales o de alguna fecha en específica.
3. Poder escoger un precio específico del libro que quiere adquirir.
4. Poder escoger un rango de edades dirigido para el libro.

**Resultado:**

1. Poder verlo de manera estructurada, poniendo la información clave del libro que estás recomendado para la información que te dio el usuario.
2. Poder acceder rápidamente a la reseña generada por el NYT acerca del libro recomendado.
3. Para los Best Sellers actuales, decirle al usuario dónde lo puede comprar.
4. Poder mostrar toda esta información en una aplicación / pantalla / interfaz gráfica amigable para que el usuario pueda interactuar con los resultados.
5. **Examen técnico**

Explica con el mayor detalle las respuestas a estas preguntas (45 minutos).

1. La información del ERP del cliente vive en un data center controlado por un proveedor, con reglas que no permiten adicionar campos, columnas, construir tablas o extraer información. ¿Qué pasos harías para pasar de la información en el ERP, cruzar y transformarla, y mandarla a una herramienta de Business Intelligence con actualizaciones automáticas?

Lo primero que se debe de hacer es identificar puntos de acceso autorizados para extraer datos del ERP. Una vez garantizando el acceso se debe diseñar un proceso de extracción y transformación de datos que cumpla con las reglas establecidas. Este proceso debe ser automatizado utilizando herramientas de ETL y scripts de python, con un cronograma definido para las actualizaciones automáticas en la herramienta de BI. Se debe monitorear continuamente el proceso para realizar ajustes según sea necesario y garantizar la seguridad y el cumplimiento de los requisitos del cliente.

1. El cliente de la pregunta 1 está evaluando adquirir un servidor físico o en la nube para poder alojar ahí un espejo de la información de su ERP. ¿Qué recomendarías en cuanto a costo o inversión y el rendimiento esperado dadas las características de cada una de las opciones?

Habría que evaluar las necesidades del cliente. Personalmente en la mayoría de los casos yo recomendaría un servidor en la nube, ya que tiene ciertas ventajas que lo vuelven generalmente una mejor opción:

* Menor inversión - Al no tener que adquirir el servidor físico el cliente se ahorra el precio del hardware, mantenimientos, electricidad y la necesidad de tener que garantizar instalaciones con las condiciones necesarias para garantizar su funcionamiento. Siendo generalmente un gasto menor desde el inicio en comparación con un servidor físico.
* Mayor escalabilidad - Los cloud services nos dan la facilidad de aumentar o disminuir los recursos de nuestro servidor según nuestra necesidad. Esto es especialmente útil si nuestro sistema no siempre consume la misma cantidad de recursos; es decir, tenemos la ventaja de solo pagar lo que utilizamos.
* Mejor adaptabilidad - Normalmente las empresas que ofrecen estos servicios, nos brindan soporte según nuestras necesidades. Podemos trabajar de la forma más básica simplemente utilizando su infraestructura (IaaS), o contratar manos extras que nos ayuden con el desarrollo y hasta administración de nuestro software (SaaS y PaaS)

Por estas razones es que a menos que se requiera un control total y condiciones específicas, recomendaría un servidor en la nube la mayoría de veces.

1. Un cliente solamente cuenta con un servidor de producción donde viven los datos, ¿qué sugerirías para no afectar el desempeño de esta al conectar BI?

Para evitar afectar el rendimiento del servidor de producción al conectar una herramienta de Business Intelligence (BI), se recomienda establecer un entorno de prueba o desarrollo separado para realizar pruebas y ajustes antes de implementar cambios en producción. Además, se debe optimizar las consultas y cargas de datos para minimizar la carga en el servidor, programar extracciones y actualizaciones durante horas de baja actividad, limitar consultas en tiempo real al servidor y monitorear de forma proactiva el rendimiento para detectar y abordar problemas rápidamente. Estas medidas ayudarán a garantizar un acceso eficiente a los datos para informes y análisis sin afectar negativamente el rendimiento del servidor de producción.

* 1. ¿Cuáles serían las implicaciones (tiempo, conocimiento, administración, costo) de esta sugerencia?

Implementar las sugerencias propuestas implica una inversión inicial de tiempo y recursos para configurar un entorno de prueba o desarrollo separado, así como para optimizar consultas y cargas de datos. Requiere un conocimiento técnico sólido en BI, bases de datos y administración de servidores para implementar y mantener estas soluciones de manera efectiva, junto con la comprensión de las mejores prácticas de rendimiento y optimización. La administración continua de estos entornos, incluida la supervisión del rendimiento y la resolución de problemas, es necesaria para garantizar el funcionamiento óptimo. Aunque puede haber un aumento inicial en los costos, a largo plazo, estas medidas pueden resultar en ahorros al reducir el riesgo de interrupciones en el servidor de producción y optimizar el rendimiento del sistema, lo que contribuye a una implementación eficiente y un beneficio en las decisiones del cliente al poder medir su sistema.

1. Un query que escribiste regresa 10,000 filas únicamente, pero su ejecución toma entre 3 y 4 horas. Por lo mismo, muchas veces no llegan los resultados completos a BI o se pierde la conexión mientras esto ocurre. ¿Qué harías?

Se deben considerar varias acciones. En primer lugar, se debe realizar una revisión exhaustiva de la consulta para identificar posibles oportunidades de optimización, como la mejora de índices, la simplificación de joins y la reducción del volumen de datos recuperados. Además, es crucial evaluar la infraestructura para garantizar que el servidor de base de datos tenga los recursos adecuados para manejar la carga de trabajo. Si es posible, dividir la consulta en partes más pequeñas puede ayudar a reducir el tiempo de ejecución y evitar problemas de conectividad al enviar resultados parciales a BI. Otra estrategia efectiva puede ser implementar procesos de extracción incremental en lugar de extraer toda la información cada vez, especialmente si los datos cambian con poca frecuencia. Además, utilizar herramientas de almacenamiento en caché puede ayudar a evitar ejecuciones repetidas de consultas. Este enfoque iterativo de monitoreo y ajuste continuo garantizará una mejora progresiva en el rendimiento de la consulta y evitará problemas de conectividad en BI debido a tiempos de ejecución prolongados.

1. Te busca un cliente para decirte que los resultados que ve en los tableros no le hacen sentido (ayer si mostraba resultados correctos). ¿Qué pasos harías para darle una respuesta al cliente?

Es una situación que requiere principalmente de análisis del sistema y sus partes. Inicialmente, realizaría un análisis exhaustivo de la arquitectura de datos, revisando la integridad de los procesos de extracción, transformación y carga (ETL) para detectar posibles fallos. Investigaría cualquier cambio reciente en la fuente de datos o en los algoritmos de procesamiento que podrían haber afectado los resultados. Además, verificaría la configuración y lógica de los tableros, asegurándose de que los filtros, métricas y dimensiones estén correctamente definidos y aplicados. Utilizando herramientas de monitoreo y registro, identificaría cualquier anomalía en el flujo de datos y realizaría pruebas exhaustivas para validar la integridad y precisión de los resultados. Finalmente, proporciona una solución técnica al cliente, documentando los hallazgos y recomendando medidas correctivas y preventivas para evitar futuras discrepancias en los resultados del tablero.

1. ¿Qué mejores prácticas sugerirías para garantizar la seguridad de la información del cliente, ya sea en un servidor físico o en la nube?

Habría que tomar medidas preventivas que cumplan con la normatividad en el manejo de datos. Esto incluye el uso de encriptación para proteger los datos tanto en reposo como en tránsito, así como la aplicación de un sólido control de acceso y gestión de privilegios para limitar el acceso solo a usuarios autorizados. Además, se deben aplicar actualizaciones y parches de seguridad regularmente, implementar sistemas de monitoreo y detección de intrusiones, realizar copias de seguridad periódicas de los datos y almacenarlas de forma segura fuera del sitio. Las auditorías de seguridad regulares son clave para evaluar y mejorar continuamente la postura de seguridad, mientras que la educación y concientización de los empleados sobre las mejores prácticas de seguridad también desempeñan un papel fundamental. Finalmente, es crucial cumplir con las regulaciones de seguridad de datos pertinentes y asegurar las instalaciones físicas donde se encuentran alojados los servidores. Estas medidas en conjunto garantizan un entorno seguro para la información del cliente, protegiéndola contra amenazas internas y externas.

1. El director comercial de una empresa nos pide un proyecto para poder visualizar información en un tablero de BI con actualizaciones cada 15 minutos con información proveniente de distintas fuentes:
   1. ERP (SQL Server)
   2. CRM (API)
   3. Catálogos de información (Excel)
   4. Google Analytics
   5. Google Ads
   6. Shopify
   7. Facebook Ads

Prepara una propuesta técnica que incluya tiempos, costos y herramientas a utilizar para entregarle al director comercial este tablero. En caso de que haya información que necesites para hacer esta propuesta y no la tengas, genera supuestos para los que sería válida la propuesta.

Para poder trabajar este proyecto, tomó como supuestos los siguientes puntos:

* Las APIs necesarias para acceder a los datos del CRM y otras fuentes externas están disponibles y son accesibles para integración.
* Los datos de Google Analytics, Google Ads, Shopify y Facebook Ads pueden ser accesados ​​a través de sus APIs, principalmente la de Google
* Los catálogos de información en formato Excel están bien estructurados y se pueden importar y procesar. De otro modo habría que definir una norma en los formatos que trabaje la empresa.
* El tablero de BI requerirá administración y mantenimiento, mismo que puede ofrecerse como soporte a largo plazo para la empresa.

Con esta propuesta, el director comercial tendrá un tablero de BI completo y actualizado con datos relevantes de todas las fuentes mencionadas, lo que le permitirá tomar decisiones informadas y estratégicas en tiempo real.

* Análisis de requisitos: Reunión inicial con el director comercial para comprender completamente sus necesidades y expectativas, yo sugeriría trabajarlo bajo una metodología Agile y garantizar este punto por medio de un Product Owner.
* Diseño de la arquitectura de datos: Diseño de una arquitectura que permita integrar y consolidar los datos de todas las fuentes mencionadas. Esto puede incluir el diseño de una base de datos intermedia y la configuración de conexiones y procesos de extracción, transformación y carga (ETL) para cada fuente de datos. Para este punto sería fundamental contar con un equipo técnico liderado por un Scrum Master con experiencia en el desarrollo de procesos ETL e Análisis de datos
* Desarrollo del tablero de BI: Desarrollo del tablero de BI utilizando una herramienta adecuada como Tableau, Power BI o Looker. Este tablero mostrará visualizaciones en tiempo real de los datos provenientes de todas las fuentes mencionadas, con actualizaciones automáticas cada 15 minutos.
* Integración de fuentes de datos: Implementación de procesos ETL para extraer datos del ERP (SQL Server), CRM (a través de API), catálogos de información (archivos Excel), Google Analytics, Google Ads, Shopify y Facebook Ads. Se establecerán conexiones seguras y eficientes para garantizar la actualización oportuna de los datos.
* Pruebas y ajustes: Realización de pruebas exhaustivas para verificar la integridad y precisión de los datos en el tablero de BI. Se realizan ajustes según sea necesario para garantizar que las visualizaciones sean precisas y relevantes para el usuario final.
* Implementación y despliegue: Implementación del tablero de BI en el entorno de producción y despliegue para uso del director comercial y otros usuarios autorizados.
* Documentación: Entrega de documentación detallada sobre su funcionamiento y mantenimiento. Se sugiere una capacitación al personal técnico de la empresa, y soporte a largo plazo para dar soluciones en incidentes y mantenimientos.

Tomando en cuenta un escenario promedio, estas son mis estimaciones para tiempo y costo:

* Tiempo - Se estima que el proyecto tomará aproximadamente 2 meses para completarse, considerando el análisis de requisitos, diseño de la arquitectura, desarrollo del tablero, integración de fuentes de datos, pruebas y ajustes, implementación y despliegue. Este plazo puede variar poco dependiendo del tamaño del proyecto y los recursos destinados, planeando completarlo en 2 sprints de un mes cada uno, sujeto al análisis del proyecto.
* Costos: Los costos del proyecto incluirán el tiempo y recursos del equipo de desarrollo, licencias de software para la herramienta de BI, posibles costos de adquisición de datos externos (por ejemplo, Google Analytics), y cualquier otro gasto relacionado con la infraestructura de servidores y almacenamiento de datos. Este es muy variable, y podrá ser proporcionado una vez que se defina el tamaño del equipo técnico, necesidades de hardware y software, así como las cuotas promedio de las APIs de google por su utilización.