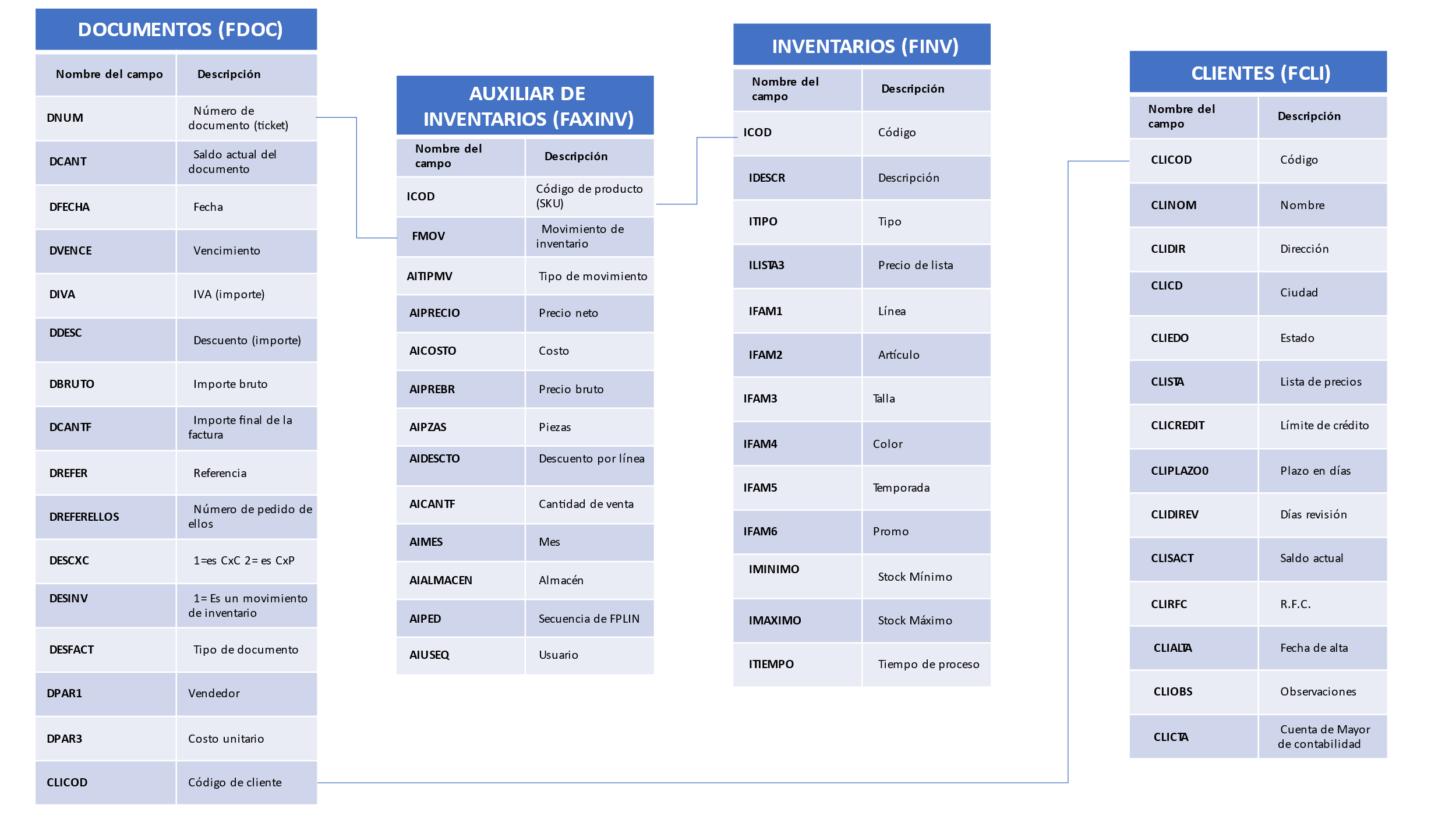
1. **Construcción de query en SQL**

**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.

SELECT

FDOC.DNUM AS '# de ticket',

FDOC.DFECHA AS 'Fecha',

FDOC.DCANTF AS 'Monto con IVA',

(FDOC.DCANTF - FDOC.DIVA) AS 'Monto sin IVA',

FDOC.DPAR1 AS 'Vendedor',

FCLI.CLINOM AS 'Cliente',

FAXINV.ICOD AS 'SKU',

FINV.IDESCR AS 'Descripción',

FAXINV.AIPZAS AS 'Cantidad',

FAXINV.AIALMACEN AS 'Almacén que vendió',

FINV.ILISTA3 AS 'Precio de lista',

FINV.IFAM3 AS 'Talla',

FINV.IFAM4 AS 'Color',

FINV.IFAM5 AS 'Temporada'

FROM FDOC

JOIN FCLI ON FDOC.CLICOD = FCLI.CLICOD

JOIN FAXINV ON FDOC.DNUM = FAXINV.FMOV

JOIN FINV ON FAXINV.ICOD = FINV.ICOD

WHERE FDOC.DCANTF > 0;

Este query une varias tablas (FDOC, FCLI, FAXINV, FINV) para tener una vista de las transacciones de ventas, incluyendo detalles del cliente, del producto y del vendedor, siempre y cuando el monto de la transacción sea mayor a cero (FDOC.DCANTF > 0)

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?

R: **Recopilación de Feedback:** Organizar sesiones de revisión con usuarios finales.

Implementación de Mejoras: Realizar ajustes en el dashboard basados en el feedback recibido.

**Capacitación de Usuarios:** Proveer formación detallada para garantizar una correcta utilización del dashboard.

**Pruebas de Validación:** Verificar la funcionalidad y rendimiento del dashboard ajustado.

**Lanzamiento y Monitoreo:** Implementar el dashboard para uso diario y monitorear su desempeño.

7. ¿Cómo esperarías que fuera la interacción entre IBSO y el cliente por los siguientes 12 meses? R: Se tendría que llevar a cabo en una serie de etapas claves diseñadas para optimizar la implementación y el uso continuo de las soluciones de Business Intelligence:

**Implementación y Configuración Inicial**: Comenzará con la instalación de herramientas de BI y la configuración de sistemas, seguido de sesiones de capacitación intensiva para los usuarios.

**Integración y Optimización**: Enfocada en asegurar la conexión efectiva entre las herramientas de BI y los sistemas existentes del cliente. Durante esta etapa, se llevarán a cabo revisiones regulares para ajustar procesos según sea necesario.

**Evaluación y Ajustes**: Se medirá el impacto de las implementaciones de BI y se realizarán modificaciones basadas en el rendimiento observado y el feedback del cliente.

**Consolidación y Planificación de Expansión**: Se confirmará la funcionalidad completa de todas las herramientas de BI y se identificarán oportunidades para ampliar el uso de BI en el negocio del cliente.

**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.
4. # Ejercicio 1: Análisis de datos climáticos de varias ciudades
5. # Paso 1: Creando un diccionario llamado 'datos\_climaticos'
6. # Este diccionario almacenará las temperaturas registradas en varias ciudades durante una semana
7. datos\_climaticos = {
8. "Ciudad de México": [23, 25, 22, 24, 26, 23, 22],
9. "Guadalajara": [18, 17, 19, 21, 20, 18, 19],
10. "Monterrey": [30, 32, 31, 29, 30, 31, 33],
11. "Cancún": [15, 14, 16, 17, 16, 15, 14]
12. }
13. # Paso 2: Inicializando variables para almacenar resultados
14. # Estas variables nos ayudarán a identificar las ciudades con la temperatura promedio más alta y más baja
15. ciudad\_temp\_max = ""
16. ciudad\_temp\_min = ""
17. max\_temp\_promedio = float('-inf')  # Usamos '-inf' para asegurarnos de que cualquier temperatura la supere
18. min\_temp\_promedio = float('inf')   # Usamos 'inf' para asegurarnos de que cualquier temperatura la supere
19. # Paso 3: Iterando sobre cada ciudad en el diccionario
20. for ciudad, temperaturas in datos\_climaticos.items():
21. # Calculando la temperatura promedio de la semana
22. temp\_promedio = sum(temperaturas) / len(temperaturas)
24. # Determinando la temperatura máxima y mínima de la semana
25. temp\_max = max(temperaturas)
26. temp\_min = min(temperaturas)
28. # Imprimiendo los resultados de la ciudad actual
29. print(f"{ciudad}: Promedio = {temp\_promedio:.2f}°C, Máxima = {temp\_max}°C, Mínima = {temp\_min}°C")
31. # Actualizando la ciudad con la temperatura promedio más alta
32. if temp\_promedio > max\_temp\_promedio:
33. max\_temp\_promedio = temp\_promedio
34. ciudad\_temp\_max = ciudad
36. # Actualizando la ciudad con la temperatura promedio más baja
37. if temp\_promedio < min\_temp\_promedio:
38. min\_temp\_promedio = temp\_promedio
39. ciudad\_temp\_min = ciudad
40. # Paso 4: Imprimiendo los resultados finales
41. print("\nResultados finales:")
42. print(f"La ciudad con la temperatura promedio más alta es {ciudad\_temp\_max} con {max\_temp\_promedio:.2f}°C.")
43. print(f"La ciudad con la temperatura promedio más baja es {ciudad\_temp\_min} con {min\_temp\_promedio:.2f}°C.")

Texto

Descripción generada automáticamente

**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 2: Asignar valores a letras minúsculas y sumar los valores de una cadena

# Paso 1: Crear una función para asignar valores a las letras y sumar los valores en una cadena

def suma\_valores\_cadena(cadena):

    # Diccionario para asignar valores a las letras de la 'a' a la 'z'

    valores\_letras = {chr(i): i - 96 for i in range(97, 123)}

    # chr(i) convierte el número ASCII en su carácter correspondiente

    # 97 es el valor ASCII de 'a', y 123 es el valor ASCII de '{' (no incluido)

    suma\_total = 0  # Inicializar la suma total

    for index, letra in enumerate(cadena):

        # Verificar si el carácter es una letra minúscula

        if letra.islower():

            # Sumar el valor de la letra a la suma total

            suma\_total += valores\_letras[letra]

        # Verificar si el carácter es una letra mayúscula

        elif letra.isupper():

            # Avisar al usuario que hay una letra mayúscula y necesita cambiarla

            print(f"Cambia a minúscula la letra '{letra}' en la posición {index + 1}.")

            return  # Detener la función hasta que el usuario ingrese una cadena válida

        # Verificar si el carácter es un número

        elif letra.isdigit():

            # Avisar al usuario que hay un número en la cadena y necesita cambiarlo

            print(f"Cambia el número en la posición {index + 1} por una letra minúscula.")

            return  # Detener la función hasta que el usuario ingrese una cadena válida

        else:

            # Si hay cualquier otro carácter no permitido

            print(f"Caracter inválido '{letra}' en la posición {index + 1}.")

            return

    # Retornar la suma total de los valores de las letras si todo está correcto

    return suma\_total

# Paso 2: Solicitar al usuario una cadena de letras minúsculas

cadena\_usuario = input("Introduce una cadena de letras minúsculas: ")

# Paso 3: Llamar a la función y mostrar el resultado si la cadena es válida

resultado = suma\_valores\_cadena(cadena\_usuario)

if resultado is not None:  # Asegurarse de que la función no retornó por un error de entrada

    print(f"La suma de los valores de las letras en la cadena '{cadena\_usuario}' es {resultado}.")



**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.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **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?

R: Utilizar una API del ERP para extraer datos y procesarlos en un data warehouse externo con herramientas ETL , permitiendo transformaciones y análisis sin afectar el sistema original.

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?

**R:** Recomendaría servidores en la nube por su escalabilidad, flexibilidad y menor costo de mantenimiento comparado con servidores físicos, ideal para empresas que buscan adaptabilidad y eficiencia.

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?
   1. ¿Cuáles serían las implicaciones (tiempo, conocimiento, administración, costo) de esta sugerencia?**R:** Implementar un sistema de replicación de datos para separar la carga de análisis BI del servidor de producción, minimizando el impacto sobre el rendimiento operativo.
2. 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?

**R:** Revisar y optimizar el query utilizando alguna herramienta disponible, también podría considerar técnicas como la indexación y partición de datos para mejorar el rendimiento.

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? **R:** Realizar una auditoría de los procesos ETL y las configuraciones de dashboards para asegurar la integridad y la actualidad de los datos mostrados.
2. ¿Qué mejores prácticas sugerirías para garantizar la seguridad de la información del cliente, ya sea en un servidor físico o en la nube? **R:** Focalizarse en la encriptación de datos, controles de acceso estrictos, y auditorías regulares para asegurar la protección de la información tanto en servidores físicos como en la nube.
3. 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 valida la propuesta.

R:

**Herramientas y Tecnologías Propuestas:**

* **Power BI:** para la creación y visualización del tablero de BI.
* **Microsoft SQL Server Integration Services (SSIS):** para la integración y transformación de datos del ERP y Excel.
* **API Connectors:** para la integración de datos desde CRM, Google Analytics, Google Ads, Shopify, y Facebook Ads.
* **Azure SQL Database:** como base de datos centralizada para consolidar y gestionar los datos antes de su visualización.

**Etapas del Proyecto:**

1. **Planificación y Diseño (2 semanas):**
   * Definición de requerimientos con stakeholders.
   * Diseño del esquema de la base de datos y arquitectura de la solución.
2. **Implementación de Integración de Datos (4 semanas):**
   * Configuración para la extracción y carga de datos.
   * Desarrollo de conectores API para fuentes externas.
3. **Desarrollo del Tablero de BI (3 semanas):**
   * Creación de modelos de datos en Power BI.
   * Diseño y configuración de visualizaciones e informes.
4. **Pruebas y Ajustes (2 semanas):**
   * Pruebas de integración y rendimiento.
   * Ajustes basados en feedback de usuarios preliminares.
5. **Despliegue y Capacitación (1 semana):**
   * Despliegue del tablero en el entorno de producción.
   * Capacitación de usuarios y entrega de documentación.

**Costos Estimados (en USD, para México):**

* Licencias de Software y Servicios Cloud: $5,000 USD (estimado anual).
* Desarrollo y Implementación: $20,000 USD (basado en 100 horas de trabajo a $200 USD/hora).
* Mantenimiento y Soporte: $2,000 USD (estimado anual).

**Supuestos:**

* Se asume acceso administrativo a todas las fuentes de datos requeridas.
* Se presume que todas las plataformas (CRM, Google Ads, Shopify, etc.) ofrecen APIs compatibles para la integración.

**Entregables:**

**Dashboards Personalizados**

* **Adaptados a Roles Específicos:** Configuración de dashboards personalizados para diferentes departamentos, asegurando que cada usuario acceda a la información más relevante y valiosa para su función.

**Interactividad y Funcionalidad Intuitiva**

* **Controles Interactivos:** Implementación de filtros y herramientas de selección que permiten a los usuarios ajustar las visualizaciones a sus necesidades específicas de forma autónoma.

**Capacitación y Soporte**

* **Capacitación Integral:** Proporcionar sesiones de formación práctica y material de apoyo detallado para facilitar la comprensión y el manejo eficaz del dashboard.
* **Soporte Continuo:** Establecimiento de un servicio de soporte técnico accesible para resolver dudas y garantizar la continuidad operativa del sistema.

**Documentación Clara y Accesible**

* **Guías y Tutoriales:** Entrega de manuales de usuario y videos tutoriales claros para facilitar la autogestión y el aprendizaje continuo de los usuarios.

**Actualizaciones y Mejoras**

* **Mantenimiento Proactivo:** Plan de actualizaciones regulares basado en feedback de usuarios, asegurando que el sistema evoluciona en línea con las necesidades del negocio y las tendencias del mercado.