**Portafolio Módulo 5: Fundamentos de Arquitectura y Modelamiento de Datos**

El siguiente Documento consolida el trabajo realizado durante el módulo 5 en el cual se desarrolla cada uno de los ejercicios indicados en clases y los que se deben desarrollar como término de cada clase.

**Clase M5 AE1: Arquitectura de Datos**

**Diseña una arquitectura de datos mínima para una APP:**

**De las bases de datos:**

Usuarios: ID, nombre, fono, dirección, fecha\_pedido, hora\_pedido

Pedidos: Nropedido, tamaño,

Restaurantes: nombre\_rest, dirección, tipo\_de\_producto, solicitud, pago

Productos:

Medios de pago:

Repartidor: IDrepartidor, vehículo, fecha\_retiro\_rest, hora\_retiro\_rest, hora\_entrega, propina, calificación

Ubicación: Restaurant, cliente, repartidor

**Fuentes:**

**Almacenamiento:** bases de datos Relacionales para datos estructurados, como usuarios, restaurantes, medios de pago y Repartidor y NoSQL, para datos que requieren mayor dinamismo, como los pedidos y Ubicación (del repartidor) en un data Lake que se pueda usar para ampliar el producto (captar más clientes para la app).

**Procesamiento:** los datos de clientes, restaurantes, medios de pago y repartidor se cargarían en batch. Los datos de pedido, pago, y entregas deben ser en tiempo real.

**Acceso:** Los accesos deben ser segmentados de acuerdo con el perfil de cada usuario.

Ejemplo: El Restaurant ve la mayor cantidad de información, pero el repartidor solo tiene acceso a los datos de retiro y entrega

**Seguridad:**

La visualización para el restaurant mediante API, con acceso a Dashboard e información histórica de su restaurante

La visualización para el repartidor es mediante una APP del pedido, de sus entregas y montos recibidos.

Para el cliente mediante una APP que muestra la información de sus datos, su pedido actual e histórico.

**Flujo de la implementación:**

* Establecer acuerdos con restaurantes
* Seleccionar plataforma para gestión de pedidos
* Configurar perfiles de restaurantes en la plataforma
* Integración de menú y sistema de pago
* Reclutamiento y capacitación de repartidores
* Establecer rutas y zonas de cobertura
* Lanzamiento gradual y pruebas del servicio
* Recopilar feedback de clientes
* Optimizar el servicio basado en feedback
* Finalizar implementación con todo en funcionamiento

**Gobernanza:** La gobernanza establece las estructuras de las bases de datos, donde se almacenan y quienes tienen acceso a ellas. De esta forma se protegen los datos y se cumple con las políticas establecidas y las leyes vigentes.

**Flexibilidad:** El hecho de trabajar con un sistema híbrido con la información almacenada en Data Lakes permite poder adaptarse a las necesidades que se pueden requerir, obteniendo información histórica y en tiempo real para mejorar la toma de decisiones.

**Escalabilidad:** Trabajar con bases de datos NoSQL permiten crecer en forma horizontal, pudiendo expandir las bases si es necesario.

**Diagrama de Flujo**:

Diagrama

El contenido generado por IA puede ser incorrecto.

En el diagrama de flujo se muestra la relación entre los proceso y datos del sistema de delivery para restaurantes, donde el cliente genera un pedido, que es ingresado al Restaurant, donde se ha establecido un menú a ofertar. Generado el pedido es entregado al repartidor, quien obtiene el pedido con los datos de los clientes y realiza la entrega.

**Preguntas a la clase:**

**¿Cuáles son los riesgos de no tener gobernanza de datos en una organización?**

El no tener gobernanza de datos pone en riesgo la información de la organización, al no establecer responsabilidades, estructura y estrategia para responder a los objetivos de la organización. Hay riesgo de pérdida de datos y de que los datos no sean fidedignos para la toma de decisiones.

**¿Cómo influye la interoperabilidad en el éxito de una arquitectura de datos?**

La interoperabilidad permite que las diferentes unidades puedan compartir información y poder manejar los datos de manera unificada y con datos consistentes y confiables.

**M5 AE2: Enfoques para el almacenamiento y gestión de los datos.**

**Mapa Comparativo entre Data Lake, Data Warehouse y DataMart:**

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | Data Lake | Data Warehouse | Data Mart |
| Tipo de datos | Crudos (en su formato original). Puede ser estructurado, semiestructurado o no estructurado. | Estructurado, limpio y preparado para el análisis. | Estructurado y resumido. Es un subconjunto de un Data Warehouse. |
| Aplicación del esquema | Se aplica "esquema en lectura". Los datos se cargan sin un esquema predefinido; se definen al consultarlos o analizarlos. | Se aplica "esquema en escritura". El esquema (estructura de las tablas) se define antes de cargar los datos. | Se aplica "esquema en escritura" ya que es una porción del Data Warehouse. |
| Casos de uso ideales | Análisis exploratorio, Machine Learning, Ciencia de Datos. Ideal para científicos de datos que necesitan datos crudos y a gran escala para descubrir patrones. | Business Intelligence (BI), Reportes y análisis de datos históricos. Perfecto para analistas y ejecutivos que necesitan informes de rendimiento para la toma de decisiones. | Análisis específico por departamento o área de negocio. Ideal para marketing, ventas o finanzas que necesitan análisis rápidos y enfocados a sus métricas. |
| Nivel de madurez técnica | Alto. Requiere personal técnico especializado (científicos de datos, ingenieros de datos) para limpiar y transformar los datos antes de su uso. | Medio. Requiere ingenieros de datos para la configuración inicial de los pipelines de ETL (Extract, Transform, Load), pero el acceso para los analistas es más sencillo. | Bajo. Al ser más pequeño y enfocado, su implementación es más simple y los usuarios finales pueden acceder fácilmente con herramientas de BI. |
| Costo y tiempo de implementación | Bajo costo inicial de almacenamiento, ya que guarda datos sin procesar. El tiempo para la preparación de los datos puede variar. | Alto costo inicial, tanto en infraestructura como en personal especializado. El tiempo de implementación es largo (meses o años). | Costo y tiempo relativamente bajos, ya que se enfoca en un área de negocio. Es la opción más rápida de las tres. |

**Casos Reales:**

**Data Lake:** Monitoreo de sensores, machine learning, integración de sistemas

**Data Warehouse:** Análisis financieros en Banca, Retail, e-commerce

**DataMart:** Departamento de una organización, como Logística, RRHH, etc.

**Preguntas a la Clase:**

**¿Por qué no existe un enfoque de datos único que sirva a todas las organizaciones?**

Porque las organizaciones son distintas y con diferentes tamaños y necesidades de datos.

**¿Qué consecuencias trae trabajar con múltiples Data Marts no gobernados?**

Puede traer pérdida de información, duplicidad, mayor costo y datos aislados.

**¿En qué casos un Data Lake se transforma en un "pantano de datos"?**

Cuando no hay gobernanza, cuando no hay control sobre los datos. Se pueden almacenar datos innecesarios e inútiles.

**Diagnóstico de Arquitectura en una empresa ficticia.**

**Caso:** Una entidad de educación trabaja de manera separada la información de los estudiantes la que se va entregando de manera parcelada y en la medida que se considere necesario. Existe un sistema de matrículas, otro del historial académico y otre de prácticas y titulación, donde cada unidad debe solicitar información a la otra en caso de surgir incongruencias en la información compartida.

**Problemas:** Actualmente, la entidad trabaja con silos de información, lo que está provocando inconsistencia en la información, falta de trazabilidad, pérdida de tiempo y dependencia entre unidades para dar respuesta a las consultas necesarias, lo que, además se traduce en un mal servicio a los estudiantes y entre los departamentos.

**Solución Propuesta: DataWarehouse**

La implementación de un DataWarehouse permitirá consolidar la información en un solo repositorio de datos, donde se podrán realizar consultas de manera directa, pudiendo realizar trazabilidad de los datos y estandarizando los reportes, lo que permitirá hacer un mejor seguimiento al historial académico de los estudiantes y poder tomar decisiones con información más precisa, como identificar los bajos rendimientos, posibilidades de deserción, tasas de retención, etc.

**Esquema de la implementación:**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.**

**Preguntas a la Clase:**

**¿Cuál es la diferencia entre esquema de lectura y esquema de escritura?**

La diferencia radica en el tipo de almacenamiento definido, en el Data Warehouse los datos están estructurados (esquema de escritura), por lo que los datos se consultan de manera más rápida, en el Data Lake los datos se almacenan en forma bruta sin formatos (esquema de lectura), por lo que la lectura puede ser más lento.

**¿Cuándo tiene Sentido implementar primero un Data Mart antes que un Data Warehouse?**

Cuando los recursos son limitados, cuando la cantidad de datos no requiere una gran implementación o si la cultura de la empresa no está preparada para la implementación de un sistema integral de los datos.

**M5 AE3: Calidad de los Datos:**

**Diagnóstico de Calidad de una Base de Datos**

**Revisa la Base de Datos entregada: Base de datos de 500 registros de nombre, correo**

En la base de datos entregada para análisis hay 500 filas, que incluye los siguientes campos:

nombre: contiene todos los datos con formato Camel Case, se podría asegurar que los datos estén ingresados de esta forma.

correo: contiene todos los datos, se puede validar que contenga un dominio (@ y .extensión).

edad: tiene datos en blanco, debe contener números enteros, formato INT (se visualizan errores ingresados con texto: veinticinco). Se sugiere convertir la palabra veinticinco a número: 25 y completar los blancos con el promedio.

telefono: se debe validar una forma estándar, en su formato: +codigonumero, ejemplo: +56912345678. Se sugiere dar el formato y los blancos dejar como 0.

ciudad: si bien contiene todos los datos, están ingresados con diferentes formatos, por lo que se debe normalizar la forma de registrar, Camel Case, con acentos. Idealmente el ingreso debe ser validado con cuadros de selección.

fecha\_registro: contiene datos en blanco, se debe establecer el formato de ingreso Ej: 2024-07-20 (aaaa-mm-dd). Y completar los blancos con la fecha inicial del periodo a evaluar, por ejemplo 2023-01-01.

**La evaluación de datos entrega la siguiente información:**

|  |  |  |
| --- | --- | --- |
| Blancos en edad | 248 | 49,6% |
| Error en edad | 127 | 25,4% |
|  |  |  |
| Blancos en teléfono | 230 | 46,0% |
| Error en Teléfono | 121 | 24,2% |
|  |  |  |
| Blancos en ciudad | 0 | 0,0% |
| errores en ciudad | tipográfico |  |
|  |  |  |
| blancos en fecha de registro | 334 | 66,8% |
| error en fecha de registros | 0 |  |

**2. Elige 5 dimensiones de Calidad a evaluar:**

**Completitud:** existen campos en blanco en diferentes columnas. Para corregir, de debe definir la forma como se completarán los datos faltantes o si se eliminarán las filas en que falte algún dato.

**Unicidad:** en la base de datos no se visualizan datos duplicados en la totalidad de los datos, se debe evaluar si algún dato duplicado en particular es un error, como un nombre, correo o teléfono.

**Precisión:** se deben validar los datos que requieran precisión para poder corregir lo que corresponda

**Consistencia:** hay error de consistencia en la edad, debido a que algunas están en texto y otras en número. Se deben unificar los formatos de ingreso, por ejemplo, el número de teléfono.

**Validez:** Se debe evaluar cuales datos tienen validez y cuales pueden estar llevando a un error. Por ejemplo, si las fechas deben estar dentro de un rango establecido.

**Sugerencias de mejora:**

**nombre:** se debe asegurar que los datos estén ingresados con el mismo formato: Camel Case.

**correo:** se sugiere validar que contenga un dominio (@ y .extensión), formato: correo@dominio.ext.

**edad:** debe contener números enteros, formato INT.

**telefono:** se debe definir una forma estándar en su formato, se sugiere separar el código de país del número. Ejemplo: +56 912345678.

**ciudad:** se debe normalizar la forma de registrar, Camel Case, con acentos. Idealmente el ingreso debe ser validado con cuadros de selección.

**fecha\_registro:** se debe establecer el formato de ingreso e idealmente acotar a las fechas válidas. Ejemplo: 2024-07-20 (aaaa-mm-dd).

**Caso de análisis: Gestión de tickets**

**Resumen Ejecutivo**

El presente informe detalla el análisis de calidad de la base de datos de gestión de tickets, centrándose en las dimensiones de Completitud, Unicidad, Validez, Consistencia y Actualidad. Se han identificado varias áreas de mejora que impactan la fiabilidad de los datos y, consecuentemente, la eficiencia operativa y la toma de decisiones. A continuación, se presentan los hallazgos clave y propuestas de mejora concretas.

**1. Metodología de Evaluación**

Se utilizaron las siguientes cinco dimensiones de calidad de datos, con sus respectivas métricas:

* **Completitud:** Porcentaje de valores nulos/vacíos en campos clave.
* **Unicidad:** Porcentaje de registros duplicados basados en ticket\_id.
* **Validez:** Porcentaje de valores fuera de formatos/rangos esperados (ej., estado, prioridad, fechas).
* **Consistencia:** Porcentaje de inconsistencias entre campos relacionados (ej., estado y fecha\_cierre).
* **Actualidad:** Porcentaje de tickets "Abiertos" inactivos por un período prolongado.

**2. Hallazgos y Cuantificación de Impacto**

**2.1. Completitud**

* **Hallazgo:** Se encontró que el **10%** de los tickets carecen de un agente asignado, y el **5%** tienen la descripción vacía.
* **Impacto:** Dificulta el enrutamiento y la resolución de problemas, y limita la capacidad de los agentes para comprender el contexto del cliente sin información detallada.

**2.2. Unicidad**

* **Hallazgo:** Se detectó un **0.5%** de ticket\_id duplicados en la base de datos.
* **Impacto:** Riesgo de gestionar el mismo problema múltiples veces, generando confusión, uso ineficiente de recursos y posibles duplicaciones en la comunicación con el cliente.

**2.3. Validez**

* **Hallazgo 1:** El **2%** de los estados de ticket no corresponden a los valores permitidos ("Abierto", "Cerrado", "En Proceso", "Pendiente").
* **Hallazgo 2:** El **1%** de los registros muestran una fecha\_cierre que es anterior a su fecha\_apertura.
* **Impacto:** Distorsiona los informes de rendimiento, impide el análisis de flujos de trabajo correctos y lleva a cálculos erróneos sobre los tiempos de ciclo.

**2.4. Consistencia**

* **Hallazgo:** El **3%** de los tickets marcados como "Cerrados" tienen el campo fecha\_cierre nulo.
* **Impacto:** Imposibilidad de medir con precisión el tiempo de resolución para estos tickets, afectando la evaluación del rendimiento del servicio y la planificación.

**2.5. Actualidad**

* **Hallazgo:** Un **15%** de los tickets "Abiertos" no han registrado ninguna actividad o actualización en los últimos 30 días.
* **Impacto:** Indica posibles problemas de estancamiento en la resolución de casos, lo que puede llevar a insatisfacción del cliente y una percepción de ineficiencia en el servicio.

**3. Propuestas de Mejora y Acciones Correctivas**

Para abordar los problemas de calidad de datos identificados, se proponen las siguientes acciones:

**3.1. Mejoras en la Completitud**

* **Acción 1:** Implementar **validaciones obligatorias** en la interfaz de creación y edición de tickets para asegurar que los campos agente y descripcion no puedan guardarse como nulos/vacíos.
* **Acción 2:** Realizar una **limpieza de datos retroactiva** para identificar y completar los campos faltantes, contactando a los agentes responsables o clientes si es necesario para obtener la información.

**3.2. Mejoras en la Unicidad**

* **Acción 1:** Configurar una **restricción de unicidad (unique constraint)** a nivel de base de datos en el campo ticket\_id para prevenir la inserción de duplicados.
* **Acción 2:** Desarrollar un script o proceso para **identificar y fusionar/eliminar registros duplicados** existentes, priorizando la preservación de la información más completa y reciente.

**3.3. Mejoras en la Validez**

* **Acción 1:** Establecer **listas desplegables o catálogos cerrados** para los campos estado y prioridad en la interfaz de usuario, evitando la entrada de valores no válidos.
* **Acción 2:** Implementar **validaciones a nivel de aplicación y/o base de datos** que aseguren que fecha\_cierre siempre sea igual o posterior a fecha\_apertura.
* **Acción 3:** Capacitar a los agentes sobre los formatos y rangos de datos esperados para campos críticos.

**3.4. Mejoras en la Consistencia**

* **Acción 1:** Automatizar una regla de negocio que, al cambiar el estado de un ticket a "Cerrado", **requiera automáticamente el registro de una fecha\_cierre válida**.
* **Acción 2:** Ejecutar un proceso de limpieza para **revisar y corregir manualmente** los tickets "Cerrados" que actualmente tienen fecha\_cierre nula.

**3.5. Mejoras en la Actualidad**

* **Acción 1:** Implementar **alertas automáticas** para los gerentes o agentes cuando un ticket "Abierto" exceda un umbral de inactividad (ej., 15 o 30 días).
* **Acción 2:** Establecer **revisiones periódicas (ej., semanales)** de tickets antiguos por parte de los supervisores para asegurar su seguimiento o cierre.
* **Acción 3:** Considerar una política de **cierre automático o escalado** para tickets que superen un tiempo máximo sin actividad.

**Conclusión**

Abordar estas deficiencias en la calidad de los datos no solo mejorará la fiabilidad de la información, sino que también optimizará la operación del sistema de tickets, facilitará la toma de decisiones basada en datos precisos y, en última instancia, mejorará la experiencia del cliente. Se recomienda priorizar la implementación de las validaciones a nivel de sistema para prevenir futuros problemas.

**M5 AE4: Introducción al Modelamiento Multidimensional:**

**Diseño de un modelo en estrella para un negocio real o ficticio:**

**Esquema de Datos para un Sistema de Ventas:**

Diagrama

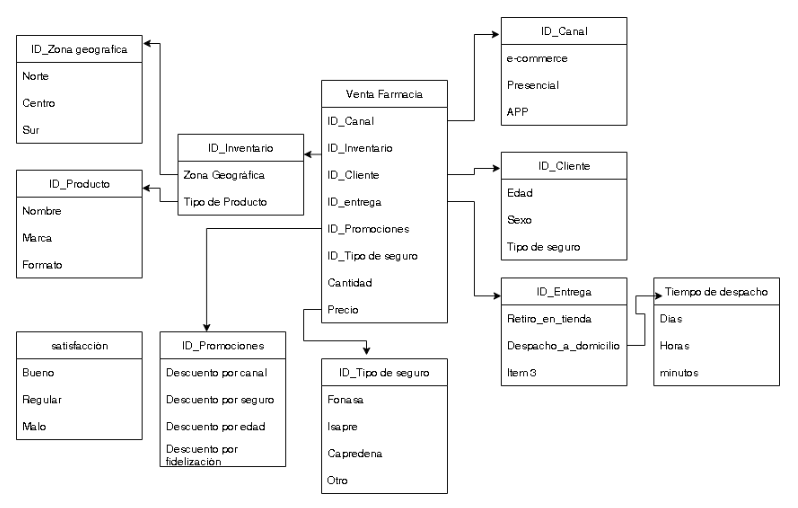
El contenido generado por IA puede ser incorrecto.

En el esquema se visualizan las bases de datos para un sistema de ventas, con la tabla de hecho: **Venta** y las tablas de dimensión o foráneas: **Cliente, Fecha, Vendedor, Producto**, las cuales van a poder consultadas para dar respuestas a las consultas que se requieran. Cada tabla de dimensión tiene una jerarquía, estableciendo que las ramas no son complejas ya que no hay mayor granularidad de los datos.

Lo que caracteriza al esquema estrella es que permite dar respuestas rápidas a las consultas, sin que estas sean muy complejas, permitiendo entregar información simple para la toma de decisión.

**Comparación y selección de esquemas para un caso de negocio:**

**Esquema de datos para caso FarmaciaSalud+**



Dada la diversidad de canales, la cantidad de sucursales y el manejo de inventarios, **se recomienda optar por un esquema híbrido** que permita:

Mantener dimensiones como Cliente, Producto y Tiempo desnormalizadas para optimizar la velocidad de consulta en análisis frecuentes.

Normalizar dimensiones jerárquicas y con alta granularidad como Geografía y Promociones/Campañas para facilitar el mantenimiento, evitar redundancias y escalar con facilidad.

Centralizar los hechos en una tabla que contenga métricas relevantes para ventas, inventarios, promociones, tiempos de despacho y satisfacción del cliente.

**Preguntas a la clase:**

**¿Qué diferencia clave hay entre un modelo OLTP y un modelo OLAP?**

OLTP se usa para la gestión de transacciones en tiempo real, operaciones pequeñas y con datos normalizados. OLAP se usa para análisis de datos y poder responder a preguntas complejas y con datos históricos, suelen ser datos desnormalizados.

**¿En qué situaciones es más conveniente usar un esquema copo de nieve?**

El esquema copo de nieve las tablas de dimensiones se subdividen en subdividen en tablas más pequeñas, se usan para reducir las redundancias de datos y para resolver relaciones complejas.

**¿Por qué es importante definir jerarquías dentro de las dimensiones?**

Permiten mejorar las consultas, al tener los datos estructurados, permite consultas más rápidas y fáciles de entender, debido a que se refieren a un contexto específico.