

Caso de Negocio: *TechStore* - Gestión de Inventario y Ventas

TechStore es una cadena de tiendas que vende productos tecnológicos en varias ciudades. Cada sucursal tiene su propio inventario, lo que significa que la cantidad de productos en stock puede variar de una sucursal a otra. Para mejorar la gestión del inventario y las ventas, *TechStore* quiere desarrollar un sistema de base de datos que administre la información de productos, sucursales, clientes, empleados y transacciones de ventas.

Requerimientos del Sistema:

1. **Productos:**
 - Los productos deben ser catalogados con información como nombre, precio y categoría.
 - Cada producto puede estar disponible en múltiples sucursales, pero la cantidad en stock varía.
2. **Sucursales:**
 - *TechStore* tiene múltiples sucursales, cada una con su dirección y detalles específicos.
 - Cada sucursal tiene su propio inventario de productos.
3. **Clientes:**
 - Los clientes realizan compras en las sucursales.
 - Se necesita almacenar la información de contacto de los clientes.
4. **Empleados:**
 - Los empleados trabajan en una sucursal específica y gestionan las ventas.
5. **Ventas:**
 - Cada venta se realiza en una sucursal, es gestionada por un empleado y es realizada por un cliente.
 - Cada venta incluye uno o más productos, con la cantidad vendida de cada producto.

Tareas:

1. **Identificación de Entidades:** Identifica las entidades principales que se necesitan en el sistema, tales como *Producto*, *Sucursal*, *Empleado*, etc.
2. **Relaciones entre Entidades:** Define las relaciones entre las entidades. (Por ejemplo: Un cliente puede realizar múltiples ventas, cada venta incluye uno o más productos, etc.). Considera preguntas como:
 - ¿Cómo se relaciona un producto con una sucursal?
 - ¿Cómo se registran las ventas y los productos vendidos?
3. **Diagrama Entidad-Relación (ERD):** Crea un diagrama que represente visualmente las entidades y sus relaciones.

Observaciones:

Minimización de Redundancia / Normalización: Asegurar que el diseño del ERD minimice la redundancia de datos. La redundancia ocurre cuando la misma información se almacena en varios lugares, lo que puede llevar a inconsistencias. Por ejemplo, si la dirección de un cliente se almacena en múltiples tablas y esta cambia, se tendría que actualizar en todos esos lugares, lo que es propenso a

errores. Para evitar esto, el diseño debe garantizar que la información se almacene de manera centralizada y que las relaciones entre las tablas se usen para acceder a la información relacionada. Asegúrate de que la estructura de la base de datos minimice la redundancia. Piensa en cómo estructurar los datos de manera que evites duplicaciones innecesarias.

Consideraciones sobre la cantidad de stock: ¿Dónde debe almacenarse la cantidad de stock para cada producto en cada sucursal?