

1. Diseño

En lugar de quedarnos con solo 8 tablas, expandimos el modelo a 9 entidades para no solo gestionar los pedidos, sino también el inventario y los pagos.

El modelo se divide en tres áreas principales:

Catálogos (Lo Fijo): Incluye las Categorías (ej. Bebidas), los Productos (lo que vendemos), los Meseros (quién atiende) y las Mesas (dónde se sientan).

Inventario y Costos (La Cocina): Aquí están los Ingredientes (la materia prima) y la tabla Recetas, que conecta cada producto con los ingredientes que lleva y la cantidad exacta. ¡Esto es clave para calcular costos!

Transacciones (El Movimiento): Lo principal es la tabla Comandas (el encabezado del pedido: quién y dónde), el Detalle_Comanda (la lista de lo que se pidió) y los Pagos (cómo y cuánto se pagó, incluyendo la propina).

Relaciones Clave: Tuvimos que usar dos tablas intermedias para manejar relaciones de muchos a muchos (M:N):

Un Producto usa muchos Ingredientes, y un Ingrediente está en muchos Productos (resuelto por Recetas).

Una Comanda tiene muchos Productos, y un Producto está en muchas Comandas (resuelto por Detalle_Comanda).

2. Normalización: Estructura 3FN Garantizada

Nuestra base de datos está en Tercera Forma Normal (3FN). Esto significa que está súper bien organizada, es flexible y evitamos la pesadilla de duplicar información.

¿Cómo lo logramos? El secreto estuvo en el área de costos. No guardamos el costo unitario de la carne en la tabla de productos finales, sino en la tabla de Ingredientes. Luego, usamos la tabla Recetas para "armar" el producto. Si el precio de la carne sube, solo cambiamos un registro en Ingredientes, ¡y todos los costos de los productos que usan carne se actualizan automáticamente en nuestros reportes!

3. Integridad y Reglas del Juego (CHECKs)

Para asegurar que nadie meta datos raros o incoherentes, usamos restricciones de integridad:

3.1 Integridad Referencial (ON DELETE):

ON DELETE RESTRICT (Lo más común): Lo usamos en todas las relaciones importantes, como entre Comandas y Meseros, o Productos e Ingredientes. Por ejemplo, no puedes borrar un Ingrediente si todavía está siendo usado en una Receta activa.

ON DELETE CASCADE (Borrado en cadena): Se aplicó en la tabla Detalle_Comanda. Si se cancela y borra una Comanda, automáticamente se borra el detalle de los artículos que tenía.

3.2 Los 3 Chequeos Mínimos (CHECK):

Cada tabla tiene reglas de validación de datos:

Reglas de Rango y Positivo: En Productos, forzamos que el precio sea mayor a cero. En Recetas, la cantidad_necesaria de un ingrediente tiene que ser obligatoriamente mayor a cero. En Pagos, la propina debe ser mayor o igual a cero.

Reglas de Dominio y Lista: En Comandas, el estado solo acepta valores predefinidos como 'Abierta', 'Pagada' o 'Cancelada'. En Pagos, el metodo_pago solo puede ser 'Efectivo', 'Tarjeta', 'Transferencia' o 'QR'.

Reglas de Formato y Lógica: En Mesas, validamos que la capacidad sea un número lógico (entre 1 y 12). También usamos chequeos para que los nombres de las categorías comiencen con mayúscula (INITCAP).

4. Las 5 Consultas Estratégicas

Las consultas demuestran la utilidad del diseño, uniendo múltiples tablas para responder preguntas de negocio cruciales:

Ingresos por Categoría: Unimos Detalle_Comanda, Productos y Categorías para saber si ganamos más vendiendo 'Platos Fuertes' o 'Bebidas'.

Mesero Estrella (Propinas): Conectamos Pagos con Meseros para identificar quién está generando la mayor propina, lo que indica un mejor servicio.

Costo de la Hamburguesa: Una consulta profunda que une Productos, Recetas e Ingredientes para calcular el costo real de una sola Hamburguesa Clásica. ¡Crucial para el margen de ganancia!

Top Ventas: Contamos las unidades vendidas en Detalle_Comanda para saber cuáles son los 5 productos más populares, informando las decisiones de inventario y promoción.

Mesas Ocupadas: Combinamos Mesas y Comandas para ver qué mesas están ocupadas y si su comanda está 'Abierta', 'En Preparación' o 'Lista', ayudando al flujo de trabajo del salón.

5. Manual de Instalación Rápida

Para que el sistema funcione en cualquier máquina con PostgreSQL, solo sigue esta secuencia:

Crea la BD: Abre tu cliente SQL y crea una nueva base de datos (ej. restaurante_db).

Estructura: Ejecuta el script creacion_bd_restaurante.sql para levantar todas las 9 tablas con sus reglas.

Datos de Prueba: Ejecuta el script datos_prueba_restaurante.sql para insertar meseros, productos, recetas, comandas de ejemplo y un pago.

Prueba Funcional: Ejecuta el script consultas_restaurante.sql para ver el resultado de las 5 consultas clave y validar que la BD esté funcionando correctamente.