

(FarmaciaDBManager 1.0.0)

Aquí se presentan las tablas que componen esta base de datos y su estructura, en dicha versión, sin agregar procedimientos almacenados, disparadores y demás mejoras para la misma, sin corregir posibles errores de estructura y demás.

Tablas de la base de datos:

1- Roles:

Define los perfiles de usuario dentro del sistema, controlando qué acciones puede realizar cada persona según su función en la farmacia.

Campos

- **RollID:** Identificador único del rol. Es un número autoincremental que sirve como clave primaria.
- **NombreRol:** Nombre del rol (Administrador, Cajero, Farmacéutico).
- **Descripcion:** Texto opcional para describir con más detalle la función del rol.
- **Activo:** Indicador lógico (1 = activo, 0 = inactivo) que permite desactivar roles sin borrarlos físicamente.

Relaciones

- Se relaciona con la tabla Usuarios mediante la clave foránea RollID.

Ejemplo de roles iniciales

- **Administrador** → Maneja todo el sistema.
- **Cajero** → Procesa ventas en el punto de venta.
- **Farmacéutico** → Administra inventario y control de medicamentos.

2- Usuarios:

Almacena la información de las personas que pueden acceder al sistema. Cada usuario está asociado a un rol, lo que define los permisos y funciones que tendrá dentro de la farmacia.

Campos

- **UsuarioID:** Identificador único del usuario. Es un número autoincremental y clave primaria.
- **Nombres:** Nombres del usuario.
- **Apellidos:** Apellidos del usuario.
- **Username:** Nombre de usuario único con el que se inicia sesión.
- **PasswordHash:** Contraseña del usuario almacenada en forma encriptada/hasheada.
- **Email:** Correo electrónico del usuario (opcional).
- **Telefono:** Número de contacto del usuario (opcional).
- **RollID:** Referencia al rol asignado al usuario. Determina qué acciones puede realizar en el sistema.

- **Activo:** Indicador lógico (1 = activo, 0 = inactivo) que permite desactivar usuarios sin borrarlos.
- **FechaCreacion:** Fecha y hora en la que el usuario fue registrado en el sistema.
- **UltimoAcceso:** Fecha y hora del último inicio de sesión del usuario.

Relaciones

- Se relaciona con la tabla **Roles**, a través del campo RolID. Esto asegura que cada usuario pertenezca a un rol válido.

Ejemplo de uso

- Un cajero podría tener RolID = 2 y solo acceder al módulo de ventas.
- Un administrador tendría RolID = 1 y control total sobre el sistema.

3- CategoríasProductos:

Organiza los productos en categorías para facilitar su clasificación, búsqueda y control. Esto permite agrupar los medicamentos o artículos según su tipo o función (analgésicos, antibióticos, vitaminas).

Campos

- **CategoriaID:** Identificador único de la categoría. Es un número autoincremental y clave primaria.
- **NombreCategoria:** Nombre de la categoría (Analgésicos, Jarabes, Suplementos).
- **Descripcion:** Texto opcional para ampliar detalles sobre la categoría.
- **Activo:** Indicador lógico (1 = activo, 0 = inactivo) que permite mantener categorías históricas sin eliminarlas.

Relaciones

- Se utiliza como referencia en la tabla Productos, permitiendo que cada producto esté asociado a una categoría específica.

Ejemplo de uso

- **Categoría:** “Antibióticos” → Productos como Amoxicilina, Ciprofloxacino.
- **Categoría:** “Vitaminas” → Productos como Vitamina C, Multivitamínicos.

4- Productos:

Almacena la información de los medicamentos o artículos disponibles en la farmacia. Es la tabla central para el control de inventario, ventas y compras.

Campos

- **ProductID:** Identificador único del producto. Es un número autoincremental y clave primaria.
- **NombreProducto:** Nombre del medicamento o producto (ejemplo: "Paracetamol 500mg").
- **CategorialD:** Identificador de la categoría a la que pertenece el producto. Relaciona con la tabla **CategoriasProductos**.
- **Presentacion:** Indica el formato o presentación del producto (caja de 20 tabletas, frasco de 120 ml).
- **PrecioUnitario:** Precio de venta por unidad del producto.
- **StockActual:** Cantidad actual disponible en el inventario.
- **StockMinimo:** Cantidad mínima permitida antes de generar alertas de reabastecimiento.
- **Activo:** Indicador lógico (1 = activo, 0 = inactivo) que permite desactivar productos sin eliminarlos.
- **FechaCreacion:** Fecha y hora en que el producto fue registrado en el sistema.
- **FechaModificacion:** Fecha y hora de la última modificación realizada al producto.

Relaciones

- CategorialD conecta cada producto con su respectiva categoría en **CategoriasProductos**.
- Este producto puede relacionarse con las tablas de **Ventas, Compras y Lotes** para controlar su movimiento.

Ejemplo de uso

- Producto: "Ibuprofeno 400mg"
 - Categoría: Analgésicos.
 - Presentación: Caja de 10 tabletas.
 - Precio unitario: 120.00.
 - Stock actual: 50.
 - Stock mínimo: 10.

5- Lotes:

Registra los diferentes lotes de cada producto, permitiendo llevar control de fechas de vencimiento, cantidades y trazabilidad del inventario.

Campos:

- **LoteID:** Identificador único del lote (numérico, autonumérico, clave primaria).
- **ProductoID:** Relación con el producto al que pertenece este lote (clave foránea hacia *Productos*).
- **NumeroLote:** Código o número del lote tal como viene en la factura o empaque del proveedor.
- **Cantidad:** Número de unidades disponibles en este lote. Por defecto inicia en 0.
- **FechaVencimiento:** Fecha de caducidad del lote (dato muy importante en farmacias).
- **FechaIngreso:** Momento en que se registró el lote en el sistema.
- **Activo:** Indicador lógico (1 = activo, 0 = inactivo), para marcar si un lote sigue vigente o fue descartado.

Relaciones:

- Cada lote está asociado a un producto mediante ProductoID.

Es importante porque en farmacias o almacenes no basta con controlar el stock general de un producto, sino que también se necesita identificar qué lote se está vendiendo, ya que cada uno puede tener diferente fecha de vencimiento.

Qué representa un lote

- Un lote es un grupo de productos idénticos que provienen de la misma producción o compra y comparten la misma fecha de vencimiento y número de lote.
- Ejemplo:
 - Producto: Paracetamol 500mg – pote de 20 pastillas.
 - Lote #A123 → 100 potes, fecha de vencimiento: 2026-02-01
 - Lote #B456 → 50 potes, fecha de vencimiento: 2026-04-15

Así se puede diferenciar de dónde viene cada unidad y qué fecha caduca primero.

Por qué es importante en el sistema

1. Control de caducidad:

- Permite vender siempre los productos que vencen primero y controlar los vencidos.

2. Trazabilidad:

- Si hay un problema o retiro de productos, sabes exactamente qué lotes fueron afectados.

3. Gestión de stock detallada:

- No solo controlas el stock total de un producto, sino por lote, lo que evita vender productos vencidos.

Cómo se relaciona con otros módulos

- **Productos:** Cada lote pertenece a un producto específico (ProductoID).
- **Compras:** Al registrar compras, se crean lotes con la cantidad y número de lote del proveedor.
- **Ventas:** Cuando se realiza una venta, se descuenta la cantidad del lote que vence primero (FIFO – First In First Out).

Analogía simple

El lote es como un cajón específico de productos:

- Se tiene 3 cajones de la misma pastilla, cada uno con diferente fecha de vencimiento.
- El sistema necesita saber de qué cajón se toma cada pastilla al vender.
- Así nunca se vende productos vencidos y se puede llevar control histórico.

Flujo de actualización del stock con lotes

1. Producto existente:

- Producto: Paracetamol 500mg
- StockActual = 100

2. Nuevo lote ingresado:

- Lote #A124
- Cantidad = 50

3. Registro del lote:

- Se crea un nuevo registro en la tabla **Lotes** con ProductoID = Paracetamol, Cantidad = 50 y su FechaVencimiento correspondiente.

4. Actualización del stock del producto:

- StockActual del producto = StockActual previo + Cantidad del lote nuevo
- $100 + 50 = 150$

Detalles importantes

- StockActual en la tabla Productos siempre representa el total disponible sumando todos los lotes activos.
- Cada venta descontará del stock de los lotes siguiendo la regla FIFO (primero se venden los lotes que vencen antes).
- Esto permite mantener inventario preciso y control de caducidad.

Cómo funciona el stock al vender

- Tabla Productos → StockActual refleja el total disponible sumando todos los lotes activos.
- Tabla Lotes → Cantidad refleja cuántas unidades quedan por lote.

Cuando se registra una venta:

1. Se identifica el producto que se está vendiendo.
2. Se buscan los lotes activos del producto, generalmente siguiendo la regla FIFO (First In First Out):
 - a. Se venden primero las unidades del lote que vence antes.
3. Se descuenta la cantidad vendida del lote correspondiente.
4. Se actualiza StockActual en Productos, reduciéndolo en la misma cantidad total vendida.

Ejemplo práctico

- Producto: Paracetamol 500mg
- StockActual = 150
- Lotes activos:
 - Lote A123 → 100 unidades, vence 2026-02-01
 - Lote B124 → 50 unidades, vence 2026-04-01

Venta: 30 unidades

- Se descuentan primero de Lote A123 → $100 - 30 = 70$
- Lote B124 sigue en 50
- StockActual del producto = $150 - 30 = 120$

Detalles importantes

- **Tabla Productos → StockActual:** total disponible.
- **Tabla Lotes → Cantidad:** control detallado por lote.
- **Ventas** siempre afectan primero a los lotes según caducidad o criterio definido.

6- Ventas:

Registra cada transacción de venta realizada en la farmacia, incluyendo la fecha, el total, el usuario que la realizó y el cliente si es necesario. Es el registro principal que conecta productos, clientes y usuarios en el sistema de ventas.

Campos

- **VentaID:** Identificador único de la venta (clave primaria, autonumérico).
- **FechaVenta:** Fecha y hora en que se realizó la venta.
- **Total:** Monto total de la venta en moneda local.
- **UsuarioID:** Referencia al usuario que registró la venta, vinculado a la tabla Usuarios.
- **ClienteID:** Referencia al cliente al que se le realizó la venta (puede ser nulo si es venta sin registro de cliente).
- **Estado:** Indica si la venta está pendiente, completada o anulada. Valor por defecto: "Pendiente".

Relaciones

- Se relaciona con Usuarios mediante UsuarioID.
- Se relaciona de forma indirecta con Clientes mediante ClienteID.
- Cada venta tiene detalles de productos vendidos en la tabla DetalleVenta, que se asocia a esta mediante VentaID.

Detalles importantes

- No almacena el detalle de los productos ni la cantidad vendida; eso queda en DetalleVenta.
- Al registrarse una venta, se debe actualizar:
 - Cantidad en los lotes (descontando según regla FIFO).
 - StockActual en la tabla Productos.

7- DetallesVentas

Registra los productos que se venden en cada transacción, incluyendo la cantidad, el lote (si aplica) y el precio unitario. Permite calcular el total de la venta y controlar el stock de manera detallada por producto y por lote.

Campos

- **DetalleID:** Identificador único de cada detalle de venta.
- **VentaID:** Referencia a la venta a la que pertenece este detalle, vinculada a la tabla Ventas.
- **ProductoID:** Referencia al producto vendido, vinculada a la tabla Productos.
- **LoteID:** Referencia opcional al lote del producto vendido, vinculada a la tabla Lotes. Permite controlar stock por lote.
- **Cantidad:** Número de unidades vendidas de ese producto en esta venta.
- **PrecioUnitario:** Precio de venta por unidad en el momento de la venta.

Relaciones

- Cada detalle está asociado a una venta específica (VentaID).
- Cada detalle está asociado a un producto (ProductoID).
- Opcionalmente, se puede asociar a un lote específico (LoteID) para descontar stock correctamente.

Detalles importantes

- Al registrar un detalle de venta:
 - Se descuenta la cantidad del lote correspondiente (si se especifica LoteID).
 - Se actualiza el StockActual del producto en la tabla Productos.
- Permite calcular el total de la venta sumando (Cantidad * PrecioUnitario) de todos los detalles.
- Ventas es padre de DetallesVentas, es decir, DetallesVentas tendrá el VentaID de Ventas para indicar que pertenece a ese mismo recibo por así decirlo.

Resumen del flujo de una venta

1. **Se registra la venta** → se crea un registro en **Ventas** con fecha, usuario, cliente y estado.
2. **Se agregan los productos vendidos** → por cada producto se crea un registro en **DetalleVenta**:
 - a. Se indica ProductoID
 - b. Se indica Cantidad
 - c. Se indica PrecioUnitario
 - d. Opcionalmente, se indica LoteID si quieres controlar stock por lote.
3. **Stock**
 - a. Se descuenta la cantidad vendida del lote correspondiente (si hay) en **Lotes**.
 - b. Se actualiza el **StockActual** del producto en **Productos**.
4. **Cálculo total**

- a. Se suman (Cantidad * PrecioUnitario) de todos los registros de DetalleVenta asociados a la venta → eso da el Total en la tabla Ventas.

Ejemplo concreto de funcionalidad

- **Producto A** → 100 unidades en stock
 - Lote L1 → 60 unidades
 - Lote L2 → 40 unidades
- **Producto B** → 50 unidades en stock
 - Lote L3 → 50 unidades

Venta 1:

- 30 unidades de Producto A
- 10 unidades de Producto B

En el sistema:

- Producto A: Se descuentan 30 unidades del lote que vence primero (L1) → L1 queda 30, L2 sigue 40
- Producto B: Se descuentan 10 unidades de Lote L3 → L3 queda 40
- StockActual en Productos:
 - Producto A → $100 - 30 = 70$
 - Producto B → $50 - 10 = 40$
- DetalleVenta guarda dos registros: uno por cada producto, con cantidad y precio unitario
- $Ventas.Total = (30 * PrecioUnitarioA) + (10 * PrecioUnitarioB)$

Ventas = Tabla principal / padre

- Cada registro en Ventas representa una venta completa, con fecha, usuario, cliente y total.
- No guarda información de los productos individuales, solo la venta en general.

DetallesVentas = Tabla hija / secundaria

- Cada registro en DetalleVenta representa un producto específico dentro de esa venta.
- Se vincula con la venta mediante VentaID.
- Permite registrar cantidad, precio unitario y lote de cada producto vendido.

Relación

- Tipo: **Uno a Muchos (1:N)**
 - Una venta (1) puede tener muchos detalles de venta (N).
 - Cada detalle pertenece a una sola venta.

Esto permitiendo:

- Registrar varias categorías de productos en la misma venta (analgésicos, jarabes, etc.)
- Calcular correctamente el total de la venta sumando todos los detalles
- Controlar el stock por lote y por producto

8- Proveedores

Almacena la información de los proveedores de productos farmacéuticos que suministran la farmacia. Permite relacionar compras con cada proveedor y llevar un historial de contacto y actividad, ya incluyendo que se provea de una propia sucursal y demás, en este caso puede ser que necesitemos agregar más controles.

Campos

- **ProveedorID:** Identificador único del proveedor (clave primaria, autonumérico).
- **NombreProveedor:** Nombre de la empresa o persona que suministra los productos.
- **Contacto:** Nombre de la persona de contacto dentro del proveedor (opcional).
- **Email:** Correo electrónico del proveedor (opcional).
- **Telefono:** Número de contacto telefónico del proveedor (opcional).
- **Activo:** Indicador lógico para marcar si el proveedor sigue activo (1 = activo, 0 = inactivo).
- **FechaCreacion:** Fecha en que se registró el proveedor en el sistema, por defecto la fecha actual.

Detalles importantes

- Permite asociar órdenes de compra con proveedores específicos.
- Facilita el envío de pedidos, consultas o seguimiento de facturación.
- Mantiene trazabilidad de las compras históricas para auditorías o reportes.

9- OrdenCompra

Registra cada orden de compra realizada a los proveedores, indicando quién la creó, la fecha, el total y el estado de la orden. Permite llevar control del proceso de adquisición de productos para el inventario de la farmacia.

Campos

- **OrdenCompraID:** Identificador único de la orden de compra (clave primaria, autonumérico).
- **ProveedorID:** Referencia al proveedor que suministra los productos, vinculada a la tabla **Proveedores**.
- **UsuarioID:** Referencia al usuario que creó la orden de compra, vinculada a la tabla **Usuarios**.
- **FechaOrden:** Fecha y hora en que se creó la orden de compra, por defecto la fecha actual.
- **Total:** Monto total de la orden de compra.
- **Estado:** Indica el estado de la orden: Pendiente (por recibir), Completada (recibida), Anulada (cancelada).

Relaciones

- Se relaciona con **Proveedores** mediante ProveedorID.
- Se relaciona con **Usuarios** mediante UsuarioID.
- Cada orden de compra tiene detalles en **DetalleCompra**, donde se registran los productos y cantidades adquiridas.

Detalles importantes

- Muy parecido o lo mismo que dije en ventas, el campo Estado reemplaza la necesidad de un campo Activo, y también toda la funcionalidad es similar en cuestión de OrdenCompra y DetalleCompra.
- Permite generar reportes de compras, controlar stock al recibir productos y llevar trazabilidad de las adquisiciones.

10- DetalleCompra

Registra cada producto incluido en una orden de compra, indicando la cantidad, el precio unitario y, opcionalmente, el lote al que pertenece. Permite controlar el inventario detalladamente y calcular el total de la orden de compra.

Campos

- **DetalleCompraID:** Identificador único de cada detalle de compra (clave primaria, autonumérico).
- **OrdenCompraID:** Referencia a la orden de compra a la que pertenece este detalle, vinculada a la tabla **OrdenCompra**.
- **ProductoID:** Referencia al producto adquirido, vinculada a la tabla **Productos**.
- **LoteID:** Referencia opcional al lote asignado al producto comprado, vinculada a la tabla **Lotes**, si es un nuevo lote hay que crearlo primero (OJO PARA REVISIÓN SOBRE ESTOS DETALLES, YA QUE ME CONFUNDE :)).
- **Cantidad:** Número de unidades compradas de ese producto en esta orden.
- **PrecioUnitario:** Precio de compra por unidad del producto en el momento de la compra.

Relaciones

- Cada detalle está asociado a una orden de compra específica (OrdenCompraID).
- Cada detalle está asociado a un producto (ProductoID).
- Opcionalmente, se puede asociar a un lote específico (LoteID) para controlar stock y trazabilidad.

Detalles importantes

- Al registrar un detalle de compra:
 - Se puede **actualizar la cantidad del lote** correspondiente en la tabla Lotes.
 - Se **actualiza StockActual** en Productos sumando las unidades compradas.
- Permite calcular el **total de la orden de compra** sumando (Cantidad * PrecioUnitario) de todos los detalles.
- Facilita generar reportes de compras históricas, análisis de proveedores y control de inventario.

Situación en funcionamiento

Cuando recibes productos de un proveedor:

- Cada envío normalmente **viene con un número de lote asignado por el proveedor**.
- Ese lote puede ser **nuevo** para tu farmacia o uno que ya existía si es un reabastecimiento.

Cómo se maneja en la base de datos

1. Crear registro de Lote (Lotes)

- Si el lote ya existe → solo se actualiza la cantidad del lote.
- Si es un lote nuevo → se crea un registro en la tabla **Lotes** con:
 - ProductoID
 - Número de lote
 - Cantidad
 - Fecha de vencimiento
 - Fecha de ingreso

2. Registrar DetalleCompra

- Se vincula el detalle de la compra con el **LoteID** correspondiente.
- Cantidad comprada → se suma al stock del lote y al stock total en Productos.

Flujo general

1. Recibes 100 pastillas de Paracetamol, lote **L003**
2. Revisas si L003 existe en la tabla **Lotes**:
 - No → lo creas con 100 unidades
 - Sí → sumas 100 unidades a la cantidad existente
3. Creas un registro en **DetalleCompra**:
 - ProductoID → Paracetamol
 - LoteID → L003
 - Cantidad → 100
 - PrecioUnitario → según lo que pagaste
4. **Productos.StockActual** → se actualiza sumando las 100 unidades

11- Clientes

Almacena la información de los clientes de la farmacia, permitiendo registrar sus datos personales y mantener un control de clientes activos para ventas, reportes y comunicación. Es fundamental para garantizar la identificación correcta de cada cliente y la trazabilidad de las transacciones, para el funcionamiento del sistema no es obligatorio registrar clientes, ya que las facturas se pueden emitir con simples datos del mismo.

Campos:

1. **ClienteID** → Identificador único de cada cliente.
2. **Nombres** → Primer nombre(s) del cliente.
3. **Apellidos** → del cliente.
4. **TipoDocumento** → Indica el tipo de documento de identificación: Cédula, Pasaporte, etc.
5. **Documento** → Número del documento de identificación del cliente.
6. **Telefono** → Número de contacto del cliente.
7. **Direccion** → Dirección del cliente.
8. **Activo** → Indica si el cliente está activo (1) o eliminado lógicamente (0).
9. **FechaCreacion** → Fecha en que se registró el cliente en el sistema.
10. **FechaModificacion** → Fecha en que se modificó por última vez la información del cliente.

Relaciones

- Cada cliente puede estar vinculado a múltiples **Ventas**, permitiendo llevar historial de compras y estadísticas.
- La combinación de **TipoDocumento + Documento** asegura que cada cliente sea único en la base de datos.

Detalles importantes:

- Documento de identificación es muy importante si se busca almacenar clientes, garantiza que no haya duplicados de clientes y permite identificar correctamente a cada persona al registrar ventas, devoluciones o consultas sobre historial de compras.
- El campo Activo permite “eliminar” clientes sin borrar sus registros, manteniendo integridad en ventas históricas.
- La información opcional como teléfono y dirección ayuda en contacto o envíos, pero no es obligatoria para registrar la venta mínima.
- FechaCreacion y FechaModificacion facilitan control de cambios en los registros de clientes, aunque en general puede que no lo necesitemos, se puede eliminar en dado caso diría yo.

12- VersionDB

Registra la versión actual de la base de datos y la fecha en que se aplicó, permitiendo llevar un control de cambios, actualizaciones y demás. Es útil o sirve para identificar en qué estado se encuentra la base en cada entorno, agregue esta tabla después de ver la varios consejos de como versionar una base de datos, y entre ello estaba el agregar una tabla así, lo cual por el momento es simbólico sin una función real o funcionalmente operativa dentro de la tabla

Campos

- **VersionID:** Identificador único de la versión.
- **NumeroVersion:** Número de la versión de la BD (1.0, 1.1, 2.0).
- **Descripcion:** Explicación si se quiere de los cambios introducidos en esta versión ("Se agregaron las tablas de Lotes y DetalleCompra").
- **FechaAplicacion:** Fecha y hora en que esta versión fue registrada en el sistema.
- **AplicadoPor:** Nombre del usuario que aplicó la actualización (ejemplo: "Admin" o "DesarrolladorX").

Relaciones

- Esta tabla es independiente: no necesita claves foráneas porque su propósito es documental.
- Puede relacionarse indirectamente con los **Usuarios** (si se guarda el UsuarioID de quien aplicó la versión), pero no es necesario tal vez.

Detalles importantes

- Permite llevar control de **versionado interno de la base de datos**, asegurando trazabilidad de los cambios.
- Con esto se puede implementar un proceso profesional de versionado de base de datos junto con los scripts SQL.

Esto es lo que se tiene en cuestión de la base de datos en su versión 1.0, sujeto a muchos cambios y mejoras, ya que solo son tablas y sus relaciones, aquí faltaría todos los procedimientos almacenados, disparadores y demás lógica en la base de datos que permita su integridad y funcionalidad como debería, de igual forma hay muchas cosas que deben manejarse en el backend y no en la base de datos, también falta corregir errores o detalles que no se tomaron en cuenta a la hora de realizar las tablas, por ejemplo en ventas agregue ClienteID, pero solo lo agregue y ya, creo que debería estar como una clave foránea me parece, para asegurar la existencia de si se usa ya que es opcional por ahora, todos esto se determina para las siguientes versiones.