

1. Introducción

Este proyecto consiste en el diseño y creación de una base de datos relacional para la gestión integral de un comercio de óptica.

La propuesta permite almacenar y relacionar información sobre clientes, profesionales médicos de la zona, empleados, proveedores, productos, ventas, detalle de ventas y recetas.

Esta base de datos está pensada para servir como soporte en procesos administrativos, logísticos, contables y de análisis de datos, facilitando la generación de reportes para la toma de decisiones, siendo lo más practica posible para su uso diario, a priori. (con el uso es posible que haga modificaciones, evaluando su practicidad diaria mayormente).

2. Situación Problemática

En la gestión diaria de la óptica, la información se almacena en distintos documentos físicos o planillas de cálculo. Esto genera problemas como:

- Dificultad para consultar el historial de compras de un cliente.
- Falta de control en el stock de productos.
- Imposibilidad de vincular ventas con recetas emitidas por profesionales.
- Dificultad para generar reportes de facturación, ventas por producto o desempeño de empleados.

La implementación de una base de datos relacional permite resolver estas brechas, unificando toda la información en un solo lugar, con relaciones claras entre entidades y eliminando redundancias.

3. Modelo de Negocio

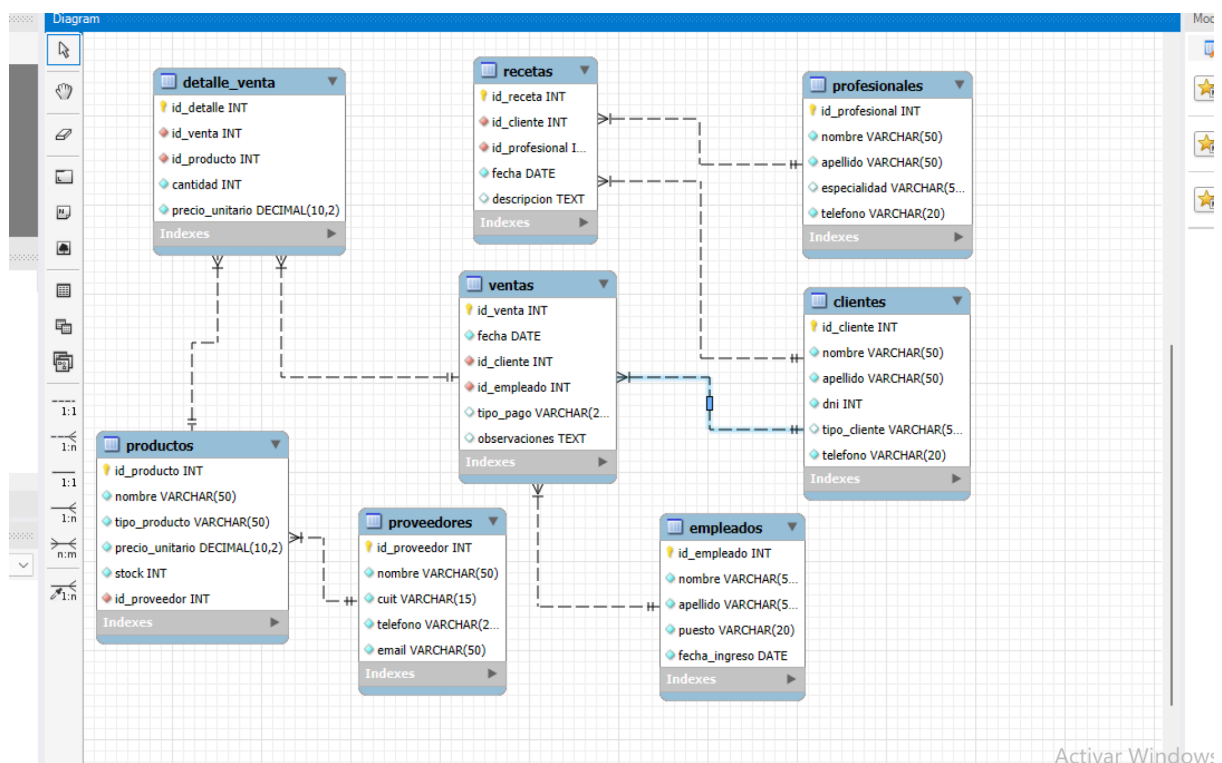
Tomo como modelo de negocio una óptica minorista que vende productos ópticos a medida, tales como anteojos recetados, lentes de contacto y accesorios.

El negocio recibe recetas de profesionales médicos externos, atiende tanto a clientes particulares como afiliados a PAMI, y trabaja con proveedores para el abastecimiento de productos.

La base de datos permitiría:

- Registrar clientes, diferenciando su tipo (particular o PAMI).
- Mantener un listado de profesionales médicos asociados.
- Registrar empleados y sus funciones.
- Controlar el inventario de productos.
- Registrar ventas y sus detalles.
- Asociar recetas a clientes y profesionales.

4. Diagrama Entidad-Relación



5. Listado de Tablas

Productos:

id_producto (PK)	INT
nombre	VARCHAR (50)
tipo_producto	VARCHAR (50)
precio_unitario	DECIMAL (10,2)
stock	INT
id_proveedor (FK)	INT

Clientes:

id_cliente (PK)	INT
nombre	VARCHAR (50)
apellido	VARCHAR (50)
dni	INT
tipo_cliente	VARCHAR (50)
telefono	VARCHAR (50)

Profesionales:

id_profesional (PK)	INT
nombre	VARCHAR (50)
apellido	VARCHAR (50)
especialidad	VARCHAR (50)
telefono	VARCHAR (50)

Recetas:

id_receta (PK)	INT
id_cliente (FK)	INT
id_profesional (FK)	INT
fecha	DATE
descripcion	TEXT

Empleados:

id_empleado (PK)	INT
nombre	VARCHAR (50)
apellido	VARCHAR (50)
puesto	VARCHAR (20)
fecha_ingreso	DATE

Proveedores:

id_proveedor (PK)	INT
nombre	VARCHAR (50)
cuit	VARCHAR (15)
telefono	VARCHAR (20)
email	VARCHAR (50)

Ventas:

id_venta (PK)	INT
fecha	DATE
id_cliente (FK)	INT
id_empleado (FK)	INT
tipo_pago	VARCHAR (20)
observaciones	TEXT

Detalle ventas:

id_detalle (PK)	INT
id_venta (FK)	INT
id_producto (FK)	INT
cantidad	INT
precio_unitario	DECIMAL (10,2)

6. Enlace al repositorio GtiHub con el script.

https://github.com/julianmagdalena/comercio_optica_julian_magdalena/blob/main/OPTICASQL.sql

7. Vistas

- `clientes_top`
Objetivo: identificar los clientes con mayor cantidad de compras registradas.
Tablas involucradas: `clientes`, `ventas`.
- `proveedor_facturacion`
Objetivo: calcular la facturación total asociada a cada proveedor.
Tablas involucradas: `detalle_venta`, `productos`, `proveedores`.
- `productos_top`
Objetivo: listar los productos más vendidos, medidos en unidades.
Tablas involucradas: `productos`, `detalle_venta`.
- `stock_cristales`
Objetivo: mostrar el stock actual únicamente de los productos de tipo "Cristal".
Tablas involucradas: `productos`.
- `empleados_sin_ventas`
Objetivo: detectar empleados que no tienen ventas registradas.
Tablas involucradas: `empleados`, `ventas`.

8. Funciones

- `total_vendido_x_empleado(id_empleado)`
Objetivo: calcular el total vendido por un empleado.
Tablas involucradas: `ventas`, `detalle_venta`.
- `condicion_cliente(id_cliente)`
Objetivo: devolver nombre, apellido y condición (particular o PAMI) de un cliente.
Tablas involucradas: `clientes`.

9. Stored Procedures

- crear_venta
Objetivo: registrar una nueva venta en la tabla `ventas` y devolver el `id_venta` generado.
Tablas involucradas: `ventas`.
- agregar_detalle_venta
Objetivo: insertar una línea de detalle en una venta existente, con producto, cantidad y precio.
Tablas involucradas: `detalle_venta`.

10. Triggers

- trg_detalle_venta_after_ins
Objetivo: actualizar el stock automáticamente cuando se inserta un nuevo detalle de venta (resta la cantidad vendida).
Tablas involucradas: `detalle_venta`, `productos`.
- trg_detalle_venta_after_del
Objetivo: restaurar el stock cuando se elimina un detalle de venta (devuelve la cantidad al inventario).
Tablas involucradas: `detalle_venta`, `productos`.

*6. Enlace al repositorio GtiHub con el script.

https://github.com/julianmagdalena/comercio_optica_julian_magdalena/blob/main/OPTICASQL.sql