

Optimización en Vidrierías

Cristian Dejoy(921135), Julián López(938189), Harrison Carrillo(901344)

Universidad Minuto De Dios

Proyecto Bases De Datos

13 de marzo de 2025

1. Introduccion

En este proyecto, se busca optimizar el manejo de inventarios y cotizaciones en una vidriería. Utilizando una base de datos, se podrá dar acceso a los clientes para verificar los productos disponibles en nuestros inventarios y consultar sus costos. Con esta base de datos, se espera lograr una mayor eficiencia y un ahorro significativo de tiempo al permitir que los clientes realicen consultas sin la necesidad de visitar un punto físico.

2. planteamiento del problema

En la mayoría de vidrerías no se utiliza ningún sistema de inventarios o bases de datos, lo cual hace que no sea tan eficiente a la hora de atender un cliente que busca un tipo de cristal en específico. Se implementaría una base de datos para así tener mejor manejo de un inventario y darle la oportunidad al cliente que tenga acceso a ellos y poder el mismo cotizar sus productos.

3. Objetivos

3.1 objetivo general

Diseñar e implementar una base de datos optimizada para la gestión eficiente de inventarios, que permita al dueño actualizar, borrar, y editar información lo cual facilite a los clientes realizar consultas detalladas sobre productos específicos, mejorando la toma de decisiones.

3.2 Objetivos específicos

- Crear un algoritmo para calcular los precios de los distintos tipos de cristales que se manejan
- crear la Base de datos en la cual se guardara el inventario del almacén
- Crear la interfaz final para la vista usuario al cual tenga acceso tanto al inventario como a la cotización de precios

4. Metodologia

Fases del diseño de la base de datos:

1. Análisis de requerimientos: Se identificó que las vidrierías no contaban con un sistema organizado para manejar inventarios ni cotizaciones.
2. Diseño conceptual: Se definieron las entidades principales y sus relaciones mediante un diagrama entidad-relación.
3. Diseño lógico: Se tradujo el modelo conceptual en un esquema lógico que será implementado en un sistema gestor de base de datos
4. Diseño físico: Se especificaron los tipos de datos y restricciones necesarias para implementar la base de datos.

Herramientas utilizadas:

- MySQL: Para la implementación del sistema de gestión de base de datos.
- Visual Studio Code: Como entornos de Desarrollo

5. IDENTIFICACIÓN DE ENTIDADES, ATRIBUTOS Y RELACIONES**Identificación de entidades principales:**

- Cliente
- Pedido
- Producto
- Venta
- DetallePedido
- Factura
- Cotización

Atributos:

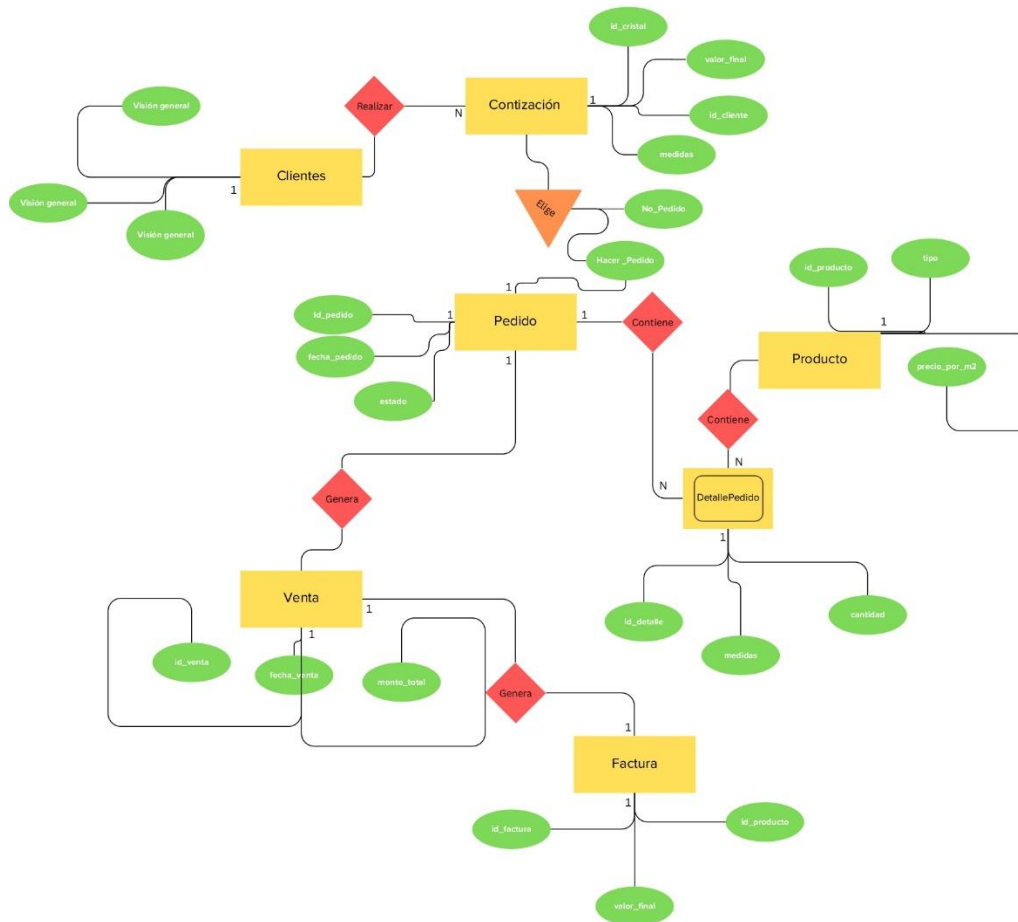
- Cliente: id_cliente (PK), nombre, dirección, contacto
- Producto: id_producto (PK), tipo, medidas, precio_por_m2
- Pedido: id_pedido (PK), fecha_pedido, estado
- DetallePedido: id_detalle (PK), cantidad, medidas, id_producto (FK)
- Venta: id_venta (PK), fecha_venta, monto_total
- Factura: id_factura (PK), valor_final, id_venta (FK)
- Cotización: id_cristal (PK), medidas, valor_final

Relaciones entre las entidades y su cardinalidad :

- Un cliente puede realizar muchos pedidos (1:N).
- Un pedido contiene muchos productos a través de DetallePedido (N:M).
- Una venta genera una factura (1:1).
- Un pedido puede generar una venta (1:1).
- La cotización está asociada a un producto específico (1:1).

6. DIAGRAMA ENTIDAD-RELACIÓN

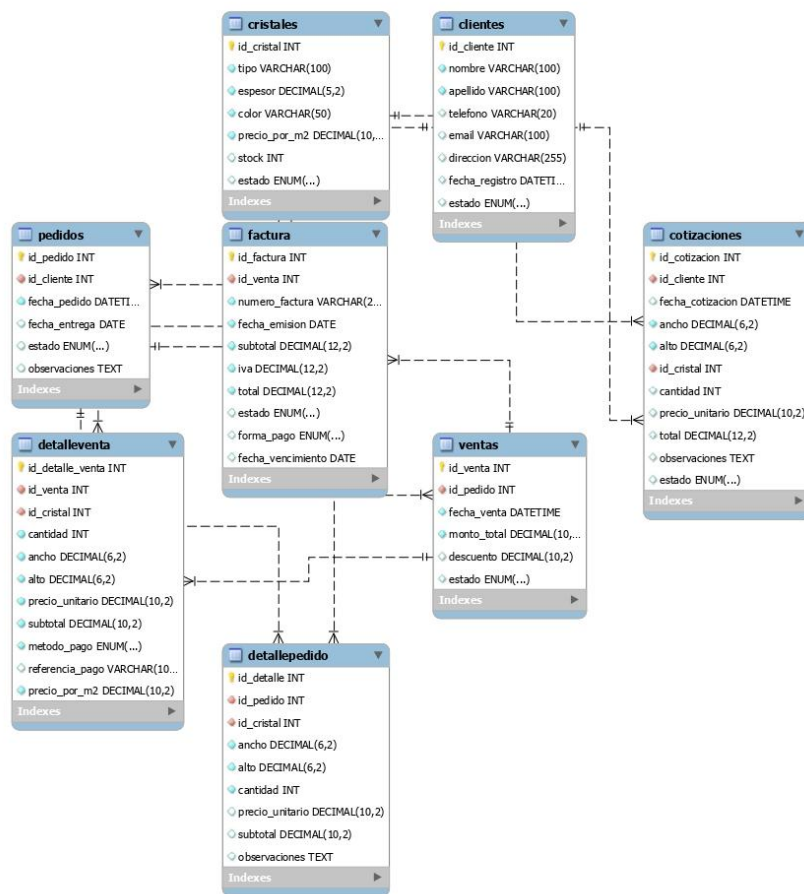
DIAGRAMA ENTIDAD-RELACION CRISTALERIA CULIAN-AGOGO



7. TRANSFORMACIÓN A MODELO LÓGICO

Conversión de entidades y relaciones a tablas:

Cada entidad fue convertida en una tabla SQL con sus respectivos atributos. Las relaciones entre entidades también fueron traducidas, en su mayoría, a claves foráneas o tablas intermedias.



Definición de claves primarias (PK) y claves foráneas (FK):

Para garantizar la integridad de los datos, se identificaron y declararon claves primarias únicas para cada tabla, así como claves foráneas que permiten relacionarlas adecuadamente:

- clientes (id_cliente PK)
- cristales (id_cristal PK)
- cotizaciones (id_cotizacion PK, FK id_cliente, FK id_cristal)
- pedidos (id_pedido PK, FK id_cliente)
- detallepedido (id_detalle PK, FK id_pedido, FK id_cristal)
- ventas (id_venta PK, FK id_pedido)
- detalleventa (id_detalle_venta PK, FK id_venta, FK id_cristal)
- factura (id_factura PK, FK id_venta)

Las claves primarias permiten identificar de forma única cada registro, mientras que las claves foráneas aseguran la integridad entre las entidades relacionadas. Además, se utilizaron tipos de datos adecuados (VARCHAR, DECIMAL, DATETIME, ENUM, TEXT, etc.) para representar correctamente la información.

Aplicación de normalización:

Se aplicaron las tres primeras formas normales:

1FN: Eliminación de grupos repetitivos y estructuración tabular.

2FN: Eliminación de dependencias parciales entre atributos y clave primaria.

3FN: Eliminación de dependencias transitivas entre atributos no clave.

8. IMPLEMENTACIÓN Y PRUEBAS EN UN SGBD

Ejecución de scripts SQL en MySQL:

```

1 • CREATE TABLE cristales (
2     id_cristal int NOT NULL AUTO_INCREMENT,
3     tipo varchar(100) NOT NULL,
4     espesor decimal(5,2) NOT NULL,
5     color varchar(50) NOT NULL,
6     precio_por_m2 decimal(10,2) NOT NULL,
7     stock int DEFAULT '0',
8     estado enum('disponible','agotado') DEFAULT 'disponible',
9     PRIMARY KEY (id_cristal)
10 ) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Se crearon las tablas en MySQL utilizando sentencias CREATE TABLE, aplicando restricciones como PRIMARY KEY, FOREIGN KEY, NOT NULL, entre otras.

Inserciones de prueba (DML):

Se realizaron inserciones de prueba con datos simulados para verificar el correcto funcionamiento de las relaciones entre tablas, por ejemplo:

Inserción de datos:

```

INSERT INTO clientes (nombre, apellido, telefono, email, direccion)
VALUES ('Carlos', 'Ramírez', '3123456789', 'carlos@mail.com', 'Calle 10 #23-45');

```

Consultas JOIN para verificar relaciones:

```

SELECT clientes.nombre, pedidos.fecha_pedido
FROM clientes
JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente
WHERE pedidos.estado = 'completado';

```

Pruebas con funciones de agregación:

```

SELECT estado, COUNT(*) AS total
FROM pedidos
GROUP BY estado;

```

9. Conclusiones

- Se diseñó e implementó una base de datos optimizada para la gestión de vidrierías, facilitando el manejo de cotizaciones, pedidos, inventario y ventas.
- Se mejoró el acceso a la información tanto para clientes como para administradores, permitiendo obtener cotizaciones exactas y consultar disponibilidad de cristales.

- La estructura relacional del sistema, con tablas vinculadas mediante claves foráneas, garantiza una gestión coherente y segura de la información.
- El sistema es escalable y puede integrarse con interfaces web o móviles en fases futuras del proyecto.
- Se recomienda continuar con el desarrollo de una interfaz gráfica y módulos de reportes.

10. Bibliografía

-MySQL Documentation. Recuperado de: <https://dev.mysql.com/doc/>

- Elmasri, R., & Navathe, S. (2016). Fundamentals of Database Systems. Pearson.
- Stack Overflow. Foros de ayuda técnica para SQL y diseño de bases de datos.
<https://stackoverflow.com/>
- Apuntes de clase de Bases de Datos – Universidad Minuto de Dios.
- YouTube. Tutoriales de implementación de bases de datos en MySQL, consultas JOIN y normalización.