

CODER HOUSE



PROYECTO FINAL **SOL**

Nicolas Carlos Yacob

INDICE

<u>1. INTRODUCCIÓN</u>	3
<u>2. OBJETIVOS</u>	3
OBJETIVO GENERAL:	3
OBJETIVOS ESPECÍFICOS:	3
<u>8. SITUACIÓN PROBLEMÁTICA</u>	3
<u>12. MODELO DE NEGOCIO</u>	4
<u>13. DIAGRAMA ENTIDAD RELACION</u>	5
<u>14. LISTADO DE TABLAS CON DESCRIPCIÓN DE ESTRUCTURA</u>	6
<u>15. SCRIPTS DE CREACIÓN DE CADA OBJETO DE LA BASE DE DATOS</u>	9
VISTAS IMPLEMENTADAS	9
FUNCIONES IMPLEMENTADAS	9
PROCEDIMIENTOS ALMACENADOS	9
TRIGGERS IMPLEMENTADOS.....	9
<u>16. SCRIPTS DE INSERCIÓN DE DATOS</u>	10
<u>17. INFORMES GENERADOS EN BASE A LA INFORMACIÓN DE LA BASE</u>	10
<u>18. HERRAMIENTAS Y TECNOLOGÍAS USADAS</u>	11

1. INTRODUCCIÓN

El emprendimiento familiar Polipiel se especializa en la fabricación y comercialización de productos personalizados y estándar, como manteles, fundas de sillas, paneras, cubre sommier y otros artículos de decoración. Utiliza materiales como cuerina, ecocuero, telas y PVC, que son adquiridos a través de proveedores locales.

La venta de estos productos se realiza principalmente mediante plataformas digitales, incluyendo redes sociales, sitios de comercio electrónico y WhatsApp. La empresa también gestiona la distribución de sus productos a través de transportes que garantizan la entrega directa a sus clientes.

El presente proyecto desarrolla un modelo de base de datos diseñado para optimizar la gestión integral de Polipiel. Este modelo busca organizar y facilitar el control sobre clientes, ventas, productos, materias primas y empleados, asegurando una mejora en la eficiencia operativa del negocio.

2. OBJETIVOS

Objetivo General:

3. Diseñar e implementar un modelo de base de datos que permita gestionar de manera eficiente las operaciones de Polipiel, optimizando la organización de la información y facilitando la toma de decisiones estratégicas.

Objetivos Específicos:

4. Centralizar el registro de clientes y sus transacciones, permitiendo un seguimiento detallado del historial de compras y cambios.
5. Mejorar el control del stock de materias primas y productos terminados, asegurando una adecuada relación con proveedores y empleados responsables.
6. Generar reportes de ventas por canales y métodos de pago, obteniendo información clave para evaluar el rendimiento comercial.
7. Automatizar procesos relacionados con la actualización de precios, validación de stock y registro de cambios en el estado de las ventas, reduciendo errores y aumentando la productividad.

8. SITUACIÓN PROBLEMÁTICA

Polipiel enfrenta múltiples desafíos en su operación diaria debido al manejo manual y descentralizado de la información, lo cual genera ineficiencias en:

9. **Trazabilidad:** Dificultades para registrar y seguir transacciones de ventas, así como los cambios realizados en los pedidos.
10. **Gestión de Stock:** Ausencia de un sistema automatizado que permita verificar la disponibilidad de materiales necesarios para la producción y garantizar que no haya desabastecimiento.
11. **Control Financiero:** Limitaciones para analizar el desempeño comercial de los canales de venta y los métodos de pago, lo que dificulta identificar las áreas de mayor rentabilidad.

Estas problemáticas se traducen en tiempos de respuesta más lentos, posibles errores en el manejo de inventarios y menor capacidad de análisis estratégico.

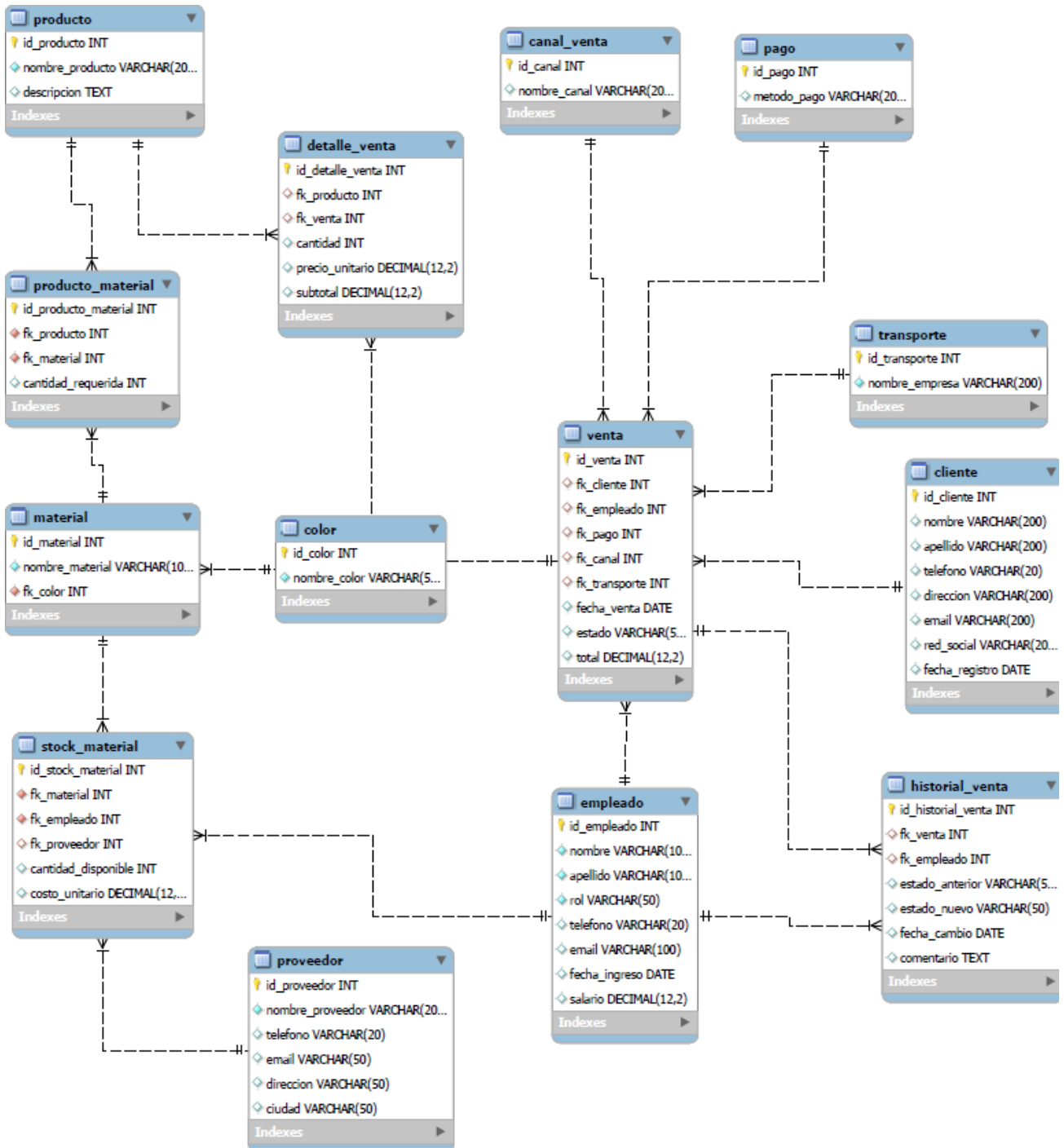
12. MODELO DE NEGOCIO

El modelo de negocio de Polipiel se basa en la **fabricación y venta directa de productos de decoración** personalizados y estándar. Sus pilares principales incluyen:

- **Fabricación Personalizada:** Adaptación de los productos a las necesidades específicas de los clientes.
- **Canales de Venta Digitales:** Uso de redes sociales, sitios de comercio electrónico y WhatsApp como principales herramientas para promocionar y vender productos.
- **Proveedores Locales:** Adquisición de materias primas de alta calidad, promoviendo el desarrollo del mercado local.
- **Distribución Directa:** Entrega de los productos a través de servicios de transporte que garantizan un contacto directo con el cliente.

Este modelo se complementa con un enfoque de optimización operativa mediante la implementación de una base de datos que respalde las actividades clave de la empresa.

13. DIAGRAMA ENTIDAD RELACION



14. LISTADO DE TABLAS CON DESCRIPCIÓN DE ESTRUCTURA

1. cliente

Descripción: Almacena información de los clientes.

Columnas:

- `id_cliente`: INT (PK, AUTO_INCREMENT) - Identificador único de cada cliente.
- `nombre`: VARCHAR(200) - Nombre del cliente.
- `apellido`: VARCHAR(200) - Apellido del cliente.
- `telefono`: VARCHAR(20) - Teléfono del cliente.
- `direccion`: VARCHAR(200) - Dirección del cliente.
- `email`: VARCHAR(200) - Correo electrónico del cliente.
- `red_social`: VARCHAR(200) - Red social del cliente.
- `fecha_registro`: DATE - Fecha de registro del cliente.

2. pago

Descripción: Métodos de pago disponibles.

Columnas:

- `id_pago`: INT (PK, AUTO_INCREMENT) - Identificador único del método de pago.
- `metodo_pago`: VARCHAR(200) - Nombre del método de pago.

3. transporte

Descripción: Información de las empresas de transporte.

Columnas:

- `id_transporte`: INT (PK, AUTO_INCREMENT) - Identificador único del transporte.
- `nombre_empresa`: VARCHAR(200) - Nombre de la empresa de transporte.

4. canal_venta

Descripción: Información de los canales de venta.

Columnas:

- `id_canal`: INT (PK, AUTO_INCREMENT) - Identificador único del canal de venta.
- `nombre_canal`: VARCHAR(200) - Nombre del canal de venta.

5. empleados

Descripción: Almacena datos de los empleados.

Columnas:

- `id_empleado`: INT (PK, AUTO_INCREMENT) - Identificador único del empleado.
- `nombre`: VARCHAR(100) - Nombre del empleado.
- `apellido`: VARCHAR(100) - Apellido del empleado.
- `rol`: VARCHAR(50) - Rol del empleado en la empresa.
- `telefono`: VARCHAR(20) - Teléfono del empleado.
- `email`: VARCHAR(100) - Correo electrónico del empleado.
- `fecha_ingreso`: DATE - Fecha de ingreso del empleado.
- `salario`: DECIMAL(12,2) - Salario del empleado.

6. proveedor

Descripción: Información de los proveedores.

Columnas:

- `id_proveedor`: INT (PK, AUTO_INCREMENT) - Identificador único del proveedor.
- `nombre_proveedor`: VARCHAR(200) - Nombre del proveedor.
- `telefono`: VARCHAR(50) - Teléfono del proveedor.
- `email`: VARCHAR(50) UNIQUE - Correo electrónico del proveedor.
- `direccion`: VARCHAR(50) - Dirección del proveedor.
- `ciudad`: VARCHAR(50) - Ciudad del proveedor.

7. producto

Descripción: Almacena información de los productos.

Columnas:

- `id_producto`: INT (PK, AUTO_INCREMENT) - Identificador único del producto.
- `nombre_producto`: VARCHAR(200) - Nombre del producto.
- `descripcion`: TEXT - Descripción del producto.

8. color

Descripción: Almacena información de los colores.

Columnas:

- `id_color`: INT (PK, AUTO_INCREMENT) - Identificador único del color.
- `nombre_color`: VARCHAR(50) - Nombre del color.

9. material

Descripción: Almacena información de los materiales.

Columnas:

- `id_material`: INT (PK, AUTO_INCREMENT) - Identificador único del material.
- `nombre_material`: VARCHAR(100) - Nombre del material.
- `fk_color`: INT (FK) - Identificador único del color.

10. venta

Descripción: Información de las ventas realizadas.

Columnas:

- `id_venta`: INT (PK, AUTO_INCREMENT) - Identificador único de la venta.
- `fk_cliente`: INT (FK) - Referencia al cliente.
- `fk_empleado`: INT (FK) - Referencia al empleado.
- `fk_pago`: INT (FK) - Referencia al método de pago.
- `fk_canal`: INT (FK) - Referencia al canal de venta.
- `fk_transporte`: INT (FK) - Referencia al transporte.
- `fecha_venta`: DATE - Fecha de la venta.
- `estado`: VARCHAR(50) - Estado de la venta.
- `total`: DECIMAL(10,2) - Total de la venta.

11. detalle_venta

Descripción: Detalles de los productos vendidos en cada venta.

Columnas:

- `id_detalle_venta`: INT (PK, AUTO_INCREMENT) - Identificador único del detalle de venta.
- `fk_producto`: INT (FK) - Referencia al producto.
- `fk_venta`: INT (FK) - Referencia a la venta.
- `cantidad`: INT - Cantidad de producto vendido.
- `precio_unitario`: DECIMAL(10,2) - Precio unitario del producto.
- `subtotal`: DECIMAL(10,2) - Subtotal de la venta de ese producto.

12. historial_venta

Descripción: Registro del historial de cambios en el estado de las ventas.

Columnas:

- `id_historial_venta`: INT (PK, AUTO_INCREMENT) - Identificador único del historial.
- `fk_venta`: INT (FK) - Referencia a la venta.
- `fk_empleado`: INT (FK) - Referencia al empleado que hizo el cambio.
- `estado_anterior`: VARCHAR(50) - Estado anterior de la venta.
- `estado_nuevo`: VARCHAR(50) - Nuevo estado de la venta.
- `fecha_cambio`: DATE - Fecha del cambio de estado.
- `comentario`: TEXT - Comentarios sobre el cambio.

13. producto_material

Descripción: Almacena la relación entre productos y materiales requeridos.

Columnas:

- `id_producto_material`: INT (PK, AUTO_INCREMENT) - Identificador único.
- `fk_producto`: INT (FK) - Referencia al producto.
- `fk_material`: INT (FK) - Referencia al material.
- `cantidad_requerida`: INT - Cantidad requerida de materiales usados para el producto.

14. stock_materiales

Descripción: Información del inventario de materiales.

Columna:

- `id_stock_materiales`: INT (PK, AUTO_INCREMENT) - Identificador único del stock del material.
- `fk_material`: INT (FK) - Referencia al material.
- `fk_empleado`: INT (FK) - Empleado encargado.
- `fk_proveedor`: INT (FK) - Proveedor de material.
- `cantidad_disponible`: INT - Cantidad disponible en stock.
- `costo_unitario`: DECIMAL(12,2) - Costo unitario del material.

15. SCRIPTS DE CREACIÓN DE CADA OBJETO DE LA BASE DE DATOS

Vistas Implementadas

1. Ventas Totales por Cliente

Esta vista organiza y presenta las ventas acumuladas por cada cliente, permitiendo identificar a los más frecuentes o aquellos que generan mayores ingresos. Es una herramienta esencial para el análisis de rendimiento comercial.

2. Ventas por Canal de Venta

Agrupar las ventas según el canal utilizado (por ejemplo, tienda física, online, etc.), mostrando tanto la cantidad de transacciones como los ingresos generados por cada canal. Esto facilita el análisis de cuáles canales son más rentables.

Funciones Implementadas

1. Validar Formato de Email

Esta función verifica si los correos electrónicos ingresados cumplen con el formato correcto. Ayuda a mantener la calidad de los datos y evitar errores en los registros de clientes.

2. Validar Disponibilidad de Materiales

Permite comprobar si hay suficiente stock de un material antes de generar órdenes de producción. Así se evita planificar pedidos que no puedan cumplirse debido a falta de inventario.

Procedimientos Almacenados

1. Actualizar Precios de Detalle de Venta

Facilita la actualización de precios de productos en las órdenes de venta, aplicando incrementos o rebajas porcentuales de forma automática. También recalcula subtotales y registra los cambios aplicados, asegurando coherencia en la información.

2. Generar Reporte de Ventas Mensual

Genera un informe con el detalle de las ventas realizadas en un mes específico, desglosado por producto. Incluye cantidades vendidas, ingresos totales y fechas de las transacciones, proporcionando una visión clara del rendimiento mensual.

Triggers Implementados

1. Registrar Cambios en el Estado de Ventas

Cada vez que se actualiza el estado de una venta, este trigger guarda un registro en una tabla de auditoría, especificando el estado anterior, el nuevo estado y otros datos relevantes. Esto asegura la trazabilidad de las operaciones.

2. Validar Stock Antes de Confirmar Detalles de Venta

Este mecanismo impide que se confirme una venta si el stock disponible del producto es insuficiente. Si no hay suficientes materiales para cumplir con la solicitud, el sistema genera un error y bloquea la operación, protegiendo la integridad de los datos y evitando inconsistencias en el inventario.

Conclusión

Estas implementaciones fortalecen la operatividad y confiabilidad del sistema de Polipiel, proporcionando herramientas que no solo simplifican la gestión diaria, sino que también contribuyen a decisiones basadas en datos. Cada elemento está diseñado pensando en garantizar la eficiencia y la calidad de la información procesada.

16. SCRIPTS DE INSERCIÓN DE DATOS

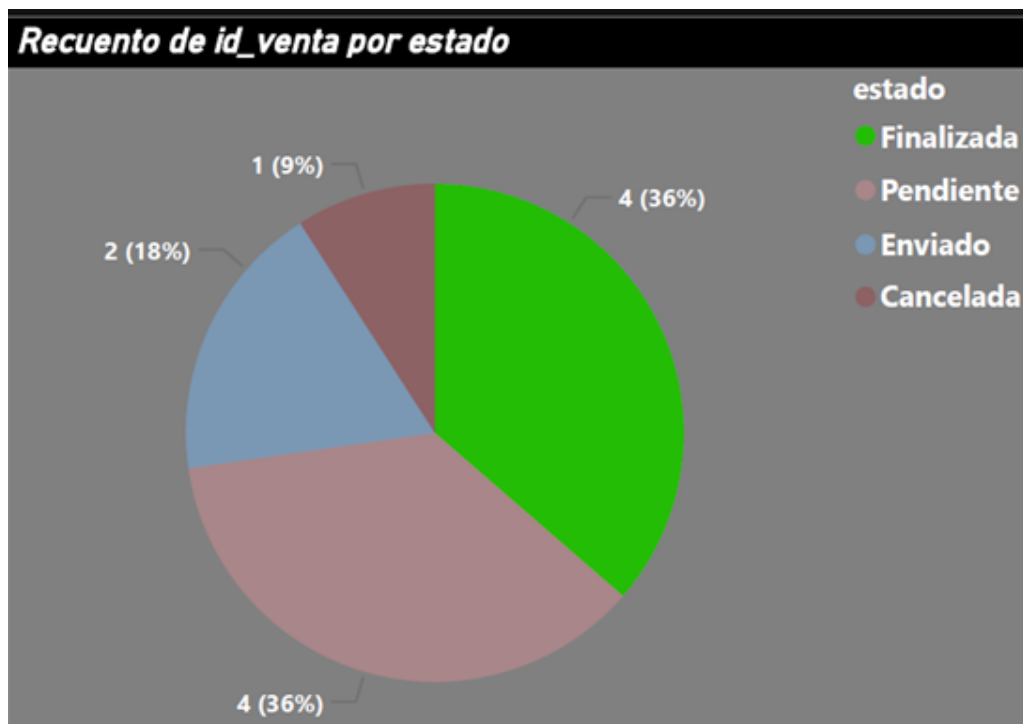
Se realizó una simple inserción de datos ficticios obtenidos desde IA (<https://claude.ai/>), a continuación se adjunta un print de pantalla:

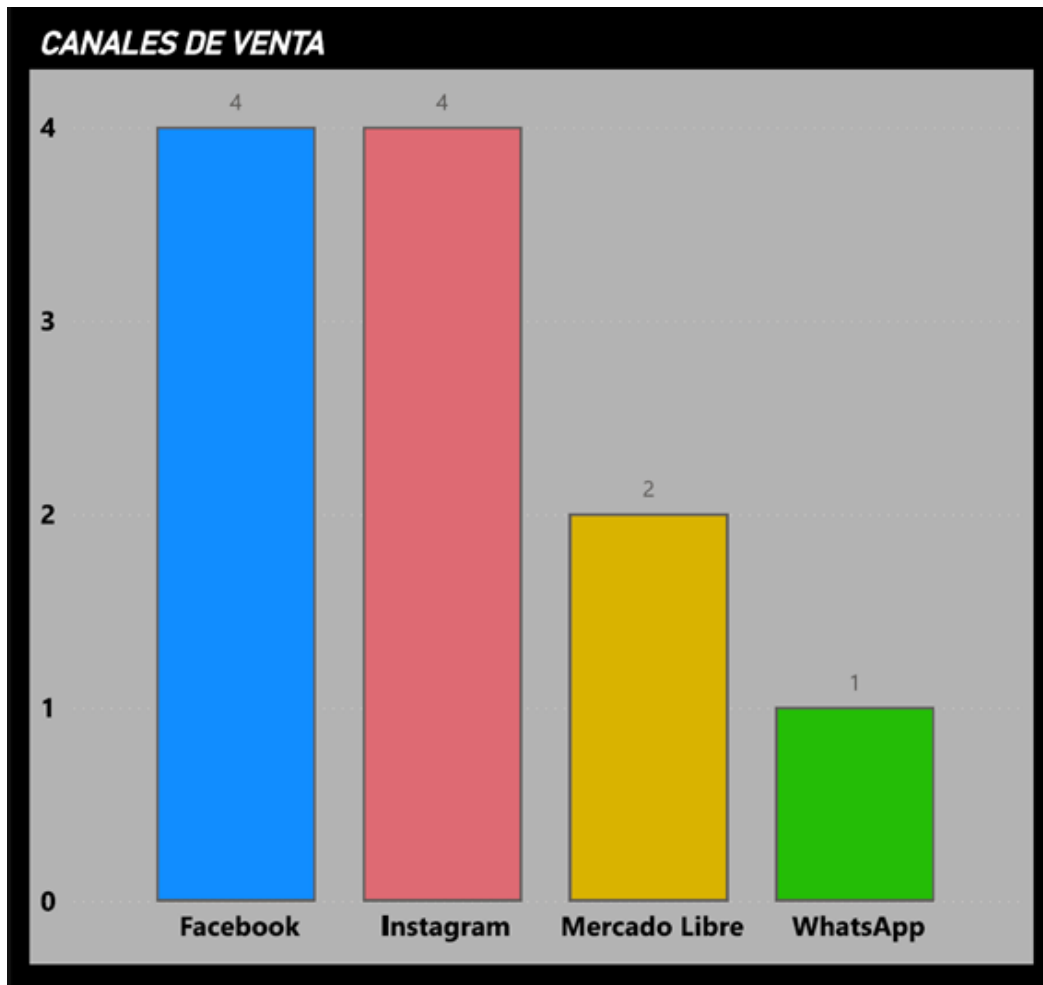
```
INSERT INTO cliente (id_cliente, nombre, apellido, telefono, direccion, email, red_social, fecha_registro) VALUES
(1, 'Sofia', 'Lopez', '15222546', 'Av. Siempre Viva 742', 'sofia.lopez@gmail.com', 'Instagram', '2024-01-10'),
(2, 'Lucas', 'Martinez', '15133122', 'Calle Falsa 123', 'lucas.martinez@gmail.com', 'Facebook', '2024-01-12'),
(3, 'Carla', 'Gomez', '15466455', 'Pasaje Verde 456', 'carla.gomez@gmail.com', 'WhatsApp', '2024-01-15'),
(4, 'Ricardo', 'Perez', '15794613', 'Boulevard Azul 789', 'ricardo.perez@gmail.com', 'Mercado Libre', '2024-01-18'),
(5, 'Mariana', 'Rodriguez', '15641385', 'Calle Rosa 321', 'mariana.rodriguez@gmail.com', 'Instagram', '2024-01-20'),
(6, 'Gabriel', 'Sánchez', '15555586', 'Calle Azul 123', 'gabriel.sanchez@gmail.com', 'Instagram', '2024-01-25'),
(7, 'Juliana', 'Castro', '15369654', 'Av. Roja 456', 'juliana.castro@gmail.com', 'Facebook', '2024-01-26'),
(8, 'Daniel', 'Vega', '15133326', 'Ruta 32', 'daniel.vega@gmail.com', 'WhatsApp', '2024-01-27'),
(9, 'Laura', 'Duarte', '15112211', 'Calle Verde 789', 'laura.duarte@gmail.com', 'Mercado Libre', '2024-01-28'),
(10, 'Pablo', 'Fernández', '15431212', 'Barrio Norte 111', 'pablo.fernandez@gmail.com', 'Instagram', '2024-01-29');

-- Tabla de pagos
INSERT INTO pago (id_pago, metodo_pago) VALUES
(1, 'Efectivo'),
(2, 'Tarjeta Debito'),
(3, 'Tarjeta Credito'),
(4, 'Transferencia Electrónica');

-- Tabla de transporte
INSERT INTO transporte (id_transporte, nombre_empresa) VALUES
(1, 'OCA'),
(2, 'Credifin'),
(3, 'Transporte Carolina'),
(4, 'OCASA'),
(5, 'Correo Argentino'),
(6, 'Andreani');
```

17. INFORMES GENERADOS EN BASE A LA INFORMACIÓN DE LA BASE





18. HERRAMIENTAS Y TECNOLOGÍAS USADAS

- MySQL (Motor de bases de datos versión: latest)
- MySQL Workbench (Interfaz gráfica)
- Claude.ai (para otorgar datos ficticios)
- Power BI (dashboard)