

1. Introducción

El proyecto consiste en el diseño e implementación de una base de datos relacional para una tienda e-commerce de tamaño pequeño a mediano. El objetivo principal es administrar de manera eficiente la información crítica del negocio, incluyendo usuarios, productos, categorías, stock, pedidos, pagos y movimientos de inventario.

La base de datos constituye el núcleo funcional sobre el cual se apoya el sistema transaccional, permitiendo asegurar la integridad de la información y la consistencia entre las distintas operaciones comerciales.

2. Objetivo del Proyecto

El objetivo del proyecto es construir un modelo de datos que soporte todas las operaciones fundamentales de un e-commerce: gestión de catálogo, control de inventario, registro de ventas, procesamiento de pagos y generación de indicadores analíticos.

El diseño contempla distintas aristas cross-funcionales del negocio:

- **Contable:** registro de ventas, pagos, totales por pedido y trazabilidad de transacciones.
- **Logística:** movimientos de stock, entradas y salidas, asociación entre productos y categorías.
- **Analítica:** información estructurada para obtener métricas como ventas por producto, margen por categoría, evolución temporal de la demanda y rotación de inventario.

La base de datos busca garantizar integridad, optimizar consultas y permitir la escalabilidad del sistema a futuro.

3. Situación Problemática

En una tienda e-commerce sin una base de datos estructurada, los distintos procesos suelen quedar desintegrados entre sí: registros de ventas en planillas, inventario desactualizado, falta de control sobre el stock disponible y registros duplicados de clientes o productos. Estos problemas generan errores operativos, pérdidas económicas y dificultades para analizar el rendimiento del negocio.

Las brechas principales identificadas son:

1. **Falta de trazabilidad del stock:** es difícil saber cuántas unidades hay disponibles en tiempo real o cuáles fueron los movimientos que afectaron el inventario.
2. **Pedidos inconsistentes:** sin un esquema centralizado existe riesgo de duplicación de órdenes, totales mal calculados o pagos sin asociar.
3. **Información fragmentada:** catálogo, clientes y ventas suelen gestionarse por separado, dificultando obtener métricas confiables.
4. **Ausencia de integridad:** no hay validaciones entre entidades (por ejemplo, un pedido puede asociarse a un usuario inexistente si no hay FK).

La implementación de una base de datos relacional resuelve estas brechas asegurando integridad, consistencia y centralización de la información.

4. Modelo de Negocio

La tienda e-commerce es una organización digital que vende productos físicos a clientes finales. Su flujo operativo puede resumirse en:

1. El usuario se registra en la plataforma.
2. Navega el catálogo, compuesto por categorías y productos.
3. Realiza un pedido agregando productos al carrito.
4. El sistema calcula totales según cantidad y precio unitario.
5. El comprador confirma el pedido y realiza el pago.
6. El personal de la tienda procesa la orden y actualiza el inventario mediante movimientos de stock.
7. El sistema registra el pago y cierra la operación.

Este modelo implica la interacción de los siguientes actores:

- **Cliente:** crea pedidos, realiza pagos.
- **Administrador / Logística:** administra productos, stock, categorías.

- **Sistema:** calcula totales, mantiene relaciones y validaciones.

El sistema debe soportar consultas analíticas (ventas por día, stock disponible, productos más vendidos), y mantener un registro consistente de todas las operaciones.

5. Diagrama Entidad–Relación (E-R)

(Descripción para el PDF — si querés te genero una imagen gráfica también)

Entidades principales:

- **Usuarios**
- **Categorías**
- **Productos**
- **Pedidos**
- **ItemPedido**
- **Pagos**

Relaciones:

- Un **Usuario** realiza **muchos Pedidos** (1:N).
- Un **Pedido** contiene **muchos Ítems** (1:N).
- Un **Producto** puede aparecer en muchos ítems de pedido (1:N).
- Una **Categoría** agrupa muchos **Productos** (1:N).
- Un **Producto** registra **muchos Movimientos de Stock** (1:N).
- Cada **Pedido** tiene asociado un **Pago** (1:1).

Cardinalidades clave:

- usuarios (1) — (N) pedidos
 - pedidos (1) — (N) item_pedido
 - productos (1) — (N) item_pedido
 - categorias (1) — (N) productos
 - productos (1) — (N) stock_movimientos
-

6. Listado de Tablas (con campos, tipos, PK y FK)

Tabla: usuarios

Campo	Tipo	Abrev.	Clave
id_usuario	INT AI	id_usr	PK
nombre	VARCHAR(100)	nom	
apellido	VARCHAR(100)	ape	
email	VARCHAR(150)	mail	UNIQUE
password_hash	VARCHAR(255)	pwd	
fecha_creacion	DATETIME	f_crea	DEFAULT

Tabla: categorías

Campo	Tipo	Abrev.	Clave
id_categoria	INT AI	id_cat	PK
nombre	VARCHAR(80)	nom	UNIQUE
descripcion	TEXT	desc	

Tabla: productos

Campo	Tipo	Abrev.	Clave
id_producto	INT AI	id_prod	PK
nombre	VARCHAR(150)	nom	
descripcion	TEXT	desc	
precio	DECIMAL(10,2)	precio	
id_categoria	INT	id_cat	FK → categorias.id_categoria
sku	VARCHAR(50)	sku	UNIQUE

Índices:

- idx_categoria (id_categoria)
-

Tabla: pedidos

Campo	Tipo	Abrev	Clave
id_pedido	INT AI	id_pedido	PK
id_usuario	INT	id_usr	FK → usuarios.id_usuario
total	DECIMAL(12,2)	tot	
estado	ENUM('carrito','confirmado','enviado','cancelado')	est	
fecha	DATETIME	fecha	DEFAULT

Índices:

- idx_usr_estado (id_usuario, estado)
-

Tabla: item_pedido

Campo	Tipo	Abrev	Clave
id_item	INT AI	id_item	PK
id_pedido	INT	id_ped	FK → pedidos.id_pedido
id_producto	INT	id_prod	FK → productos.id_producto
cantidad	INT	cant	
precio_unit	DECIMAL(10,2)	p_unit	

Índices:

- idx_ped_prod (id_pedido, id_producto)

Tabla: pagos

Campo	Tipo	Abrev	Clave
id_pago	INT AI	id_pago	PK
id_pedido	INT	id_ped	FK (UNIQUE) → pedidos.id_pedido
monto	DECIMAL(12,2)	monto	
metodo	VARCHAR(50)	met	
fecha_pago	DATETIME	f_pago	
estado	VARCHAR(50)	est_p	

6. Diagrama E-R

