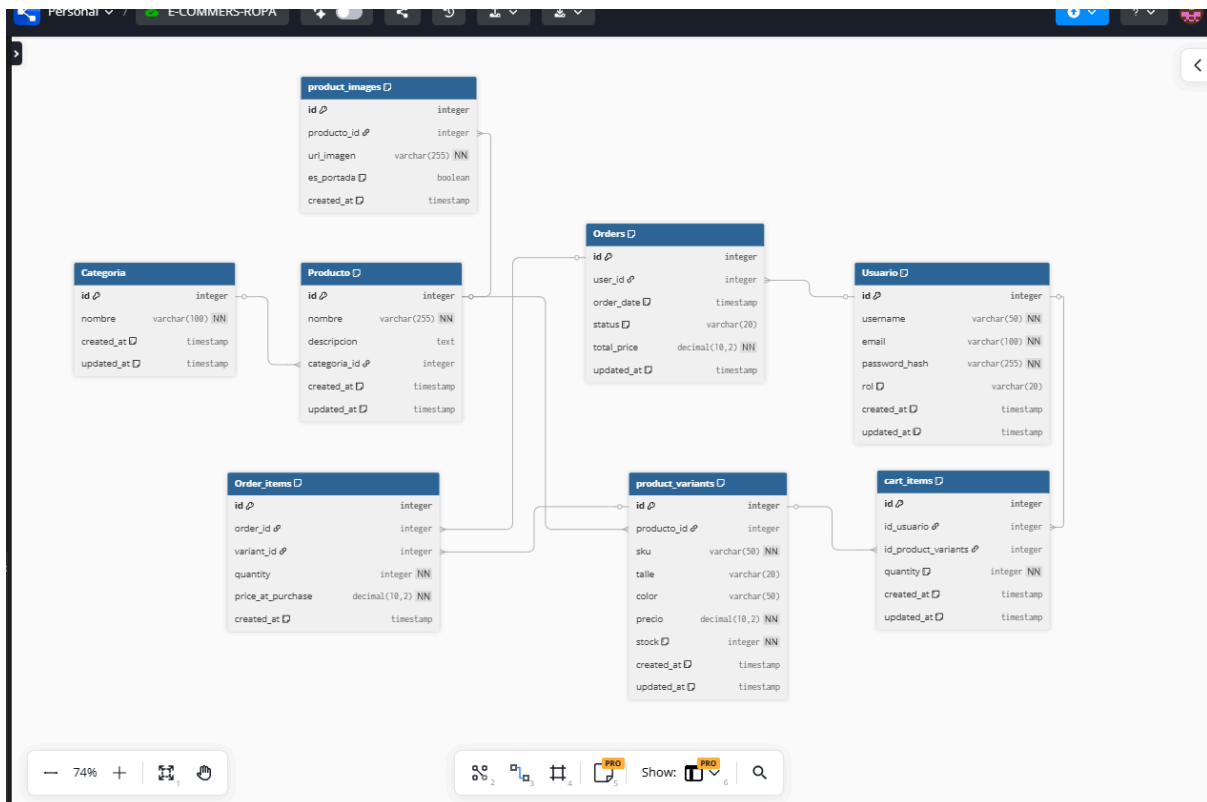


# Documentación del Modelo de Datos: E-commerce de Indumentaria

## 1. Introducción

Este proyecto implementa una base de datos relacional robusta para una tienda de ropa en línea. El diseño se centra en la **escalabilidad**, la **integridad referencial** y la **trazabilidad histórica** de las ventas.

[Ver Diagrama de Entidad-Relación Interactivo en DBDiagram.io](#)



## 2. Diccionario de Datos

### Módulo de Catálogo e Inventario

Este bloque gestiona la jerarquía de productos, asegurando que el stock se maneje de forma atómica.

Tabla	Descripción	Campos Clave

<b>Categoría</b>	Clasificación de productos (ej. remeras, pantalones).	nombre
<b>Producto</b>	Información base del artículo (catálogo general).	categoria_id, nombre
<b>product_variants</b>	<b>Nivel de Stock real.</b> Cruza producto con talla y color.	sku (único), talla, color, precio, stock
<b>product_images</b>	Galería multimedia vinculada a los productos.	url_imagen, es_portada

- Decisión de Diseño:** Se separó **Producto** de **product\_variants** para permitir que un mismo modelo tenga múltiples combinaciones de tallas y colores con inventarios independientes.

## Módulo de Usuarios y Sesión

Gestión de clientes y persistencia de sus preferencias de compra.

Tabla	Descripción	Campos Clave
<b>Usuario</b>	Registro de clientes y administradores.	email, password_hash, rol
<b>cart_items</b>	Persistencia del carrito de compras antes de la orden.	id_usuario, id_product_variants

## Módulo de Ventas (Transaccional)

El núcleo del negocio. Mantiene la integridad de lo vendido.

Tabla	Descripción	Campos Clave
<b>Orders</b>	Cabecera del pedido (información global).	user_id, status, total_price

<b>Order_items</b>	Detalle de productos dentro de cada pedido.	price_at_purchase
--------------------	---	-------------------

- **Decisión de Diseño:** El campo `price_at_purchase` en `Order_items` es vital para el **historial financiero**. Registra el precio al momento de la venta, permitiendo cambios futuros en los precios del catálogo sin alterar la contabilidad pasada.

---

### 3. Estrategia de Auditoría (Timestamps)

Todas las tablas incluyen los campos `created_at` y `updated_at`.

- **created\_at:** Registra el momento exacto de creación del registro.
- **updated\_at:** Permite auditar la última modificación (cambios de stock, actualizaciones de estado de pedido o cambios de contraseña).

---

### 4. Optimización de Consultas (Índices)

Para garantizar un rendimiento óptimo bajo carga, se han implementado los siguientes índices estratégicos:

- **Búsqueda de Productos:** `idx_producto_nombre` permite búsquedas textuales rápidas.
- **Logística de Inventario:** `idx_variants_sku` y `idx_variants_producto` optimizan la actualización de stock.
- **Acceso de Usuarios:** `idx_usuario_email` asegura que el proceso de login sea de tiempo constante.
- **Gestión de Pedidos:** `idx_orders_status` y `idx_orders_fecha` facilitan la creación de dashboards administrativos y reportes de ventas mensuales.

---

### 5. Integridad Referencial

El modelo utiliza **Foreign Keys (FK)** en todas sus relaciones para evitar "datos huérfanos". Por ejemplo, no se puede eliminar una categoría si existen productos asociados a ella, y no se puede borrar un usuario que tenga pedidos registrados.

### 6. Decisiones de Arquitectura y Normalización

**Cumplimiento de las Formas Normales (3NF):** El diseño se normalizo hasta la Tercera Forma Normal para eliminar la redundancia de datos y proteger la integridad referencial.

- **1NF:** Se eliminaron grupos repetidos (como talles en una sola celda), moviéndolos a la tabla `product_variants`.
- **2NF:** Todas las columnas dependen de la clave primaria de su respectiva tabla.
- **3NF:** No existen dependencias transitivas; los datos de las categorías, por ejemplo, viven exclusivamente en su tabla y se referencian mediante llaves foráneas.