💡 Entendimiento del negocio

### **📦 Sistema de Gestión de Abastecimiento**

#### **🧠 Contexto del Negocio**

Una empresa del sector retail requiere mejorar el control sobre sus procesos de abastecimiento. Actualmente, la gestión de productos, proveedores y órdenes de compra se realiza de forma dispersa, dificultando la trazabilidad, la eficiencia operativa y la toma de decisiones basada en datos.

#### **🎯 Objetivo del Sistema**

Diseñar una base de datos transaccional (OLTP) que permita:

* Registrar productos y sus respectivas categorías.
* Gestionar los proveedores que abastecen los productos.
* Controlar las órdenes de compra realizadas a cada proveedor.
* Llevar un historial detallado de los productos comprados, cantidades y precios unitarios.

#### **📌 Beneficios Esperados**

* Reducción de errores humanos en el registro de compras.
* Acceso rápido a la información de productos y proveedores.
* Mejora en la toma de decisiones logísticas y de inventario.
* Base sólida para desarrollar futuros dashboards o sistemas automatizados.

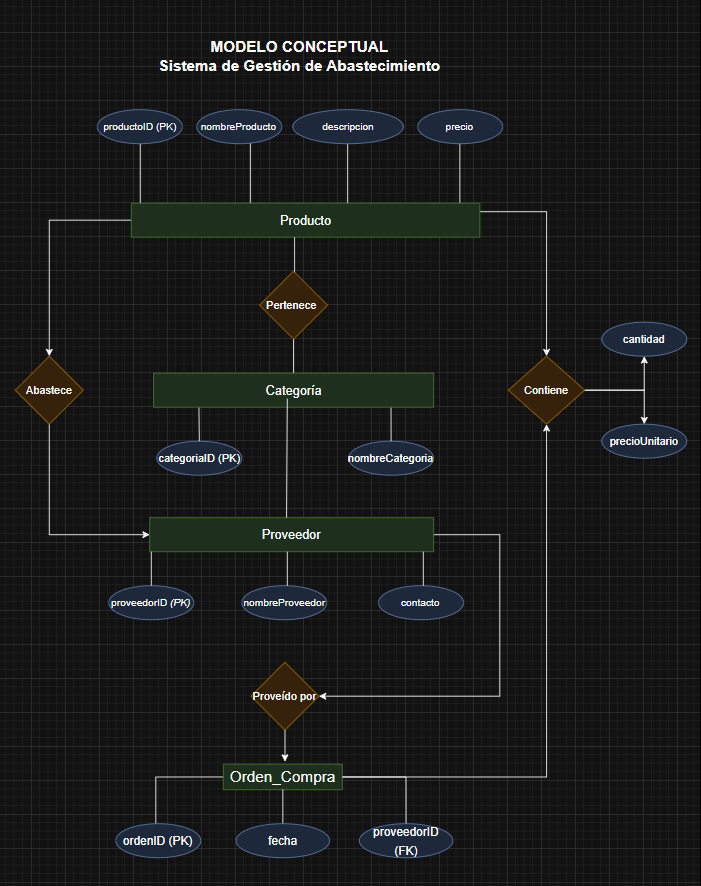
#### **👤 Usuarios del Sistema**

* Jefes de compras
* Auxiliares logísticos
* Coordinadores de inventario

#### **🧱 Proceso a Modelar**

1. Registro de productos y asignación a una categoría.
2. Asociación de productos a uno o varios proveedores.
3. Registro de órdenes de compra realizadas.
4. Detalle de cada producto incluido en la orden, con su cantidad y precio.

⛓️ Modelo Conceptual (ER)



## **📄 Resumen del Modelo Conceptual**

## **Sistema de Gestión de Abastecimiento**

En este modelo conceptual se representa un sistema transaccional enfocado en gestionar el abastecimiento de productos desde proveedores, incluyendo la categorización de productos y el control de órdenes de compra.

### 

| **Elemento** | **Cantidad** | **Detalle** |
| --- | --- | --- |
| **Entidades** | 4 | Producto, Categoría, Proveedor, Orden\_Compra |
| **Relaciones** | 4 | Pertenece, Abastece, Proveído por, Contiene |
| **Atributos** | 13 | Atributos directos de las entidades + 2 atributos en una relación |

### **✅ Resultado final del modelo**

### **🧩 Descripción general**

* Cada **producto** tiene información como nombre, precio y descripción, y **pertenece a una categoría**.
* Los **proveedores** ofrecen productos, generando una relación **N:M** entre productos y proveedores (*Abastece*).
* Las **órdenes de compra** son emitidas por proveedores, y cada una **contiene varios productos**, con atributos específicos como **cantidad y precio unitario** (*Contiene*).
* Las relaciones han sido representadas con **rombos**, siguiendo la notación estándar del modelo Entidad-Relación.

### **🎯 ¿Por qué es un buen modelo?**

* Sigue principios de **normalización**.
* Claramente separa entidades de relaciones.
* Representa bien las **cardinalidades reales** (1:N y N:M).
* Es una base perfecta para pasar al modelo lógico y luego físico en SQL.

Modelo Lógico (Tablas SQL)



## **🧱 Modelo Lógico**

## **Sistema de Gestión de Abastecimiento**

### **🎯 Propósito**

El modelo lógico representa la **estructura relacional detallada** del sistema de abastecimiento planteado en el modelo conceptual. En esta fase, se definen las **tablas, campos, claves primarias y claves foráneas**, necesarias para construir una base de datos transaccional eficiente y normalizada.

### **📌 Características del modelo**

* Se compone de **6 tablas principales**, que representan las entidades y relaciones más relevantes del sistema.
* Cada tabla incluye los campos con sus respectivos tipos de dato (VARCHAR, INT, DECIMAL, DATE).
* Se definen claramente las **claves primarias (PK)** y **claves foráneas (FK)** para garantizar la integridad referencial.
* Las relaciones de **muchos a muchos (N:M)** se resuelven mediante **tablas intermedias** con claves primarias compuestas.

### **🧩 Estructura general**

| **Tabla** | **Descripción** |
| --- | --- |
| Producto | Almacena los productos que se gestionan. |
| Categoria | Define la categoría o tipo de cada producto. |
| Proveedor | Registra la información de los proveedores. |
| Orden\_Compra | Representa cada orden generada para abastecer productos. |
| Producto\_Proveedor | Relación N:M entre productos y proveedores. |
| Detalle\_Orden | Relación N:M entre órdenes y productos, incluyendo cantidad y precio unitario. |

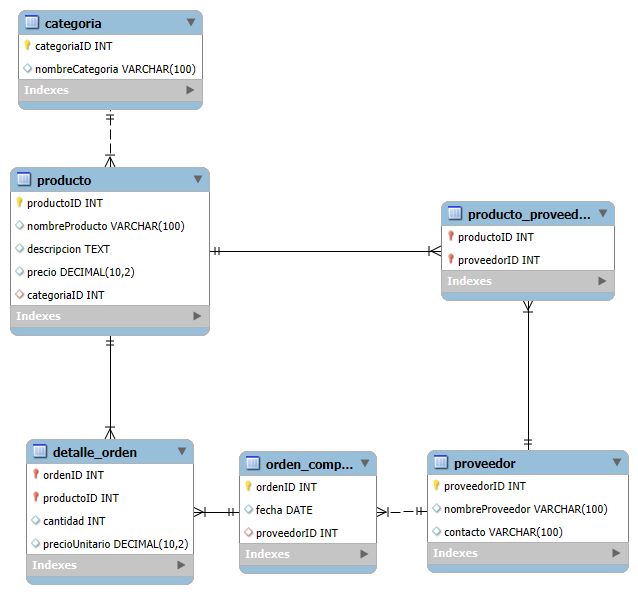
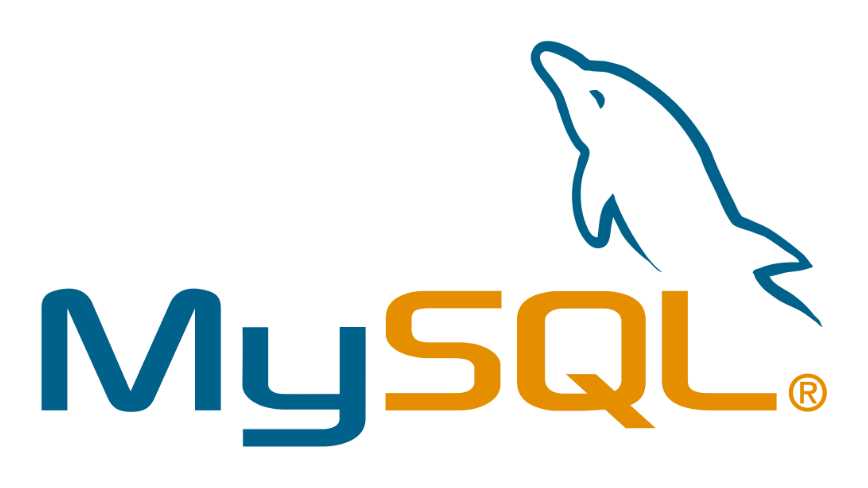
### **🧠 Resultado esperado**

Este modelo permite:

* Consultas claras sobre qué productos están en cada orden.
* Saber quién provee cada producto.
* Seguir la trazabilidad desde la orden hasta el proveedor.
* Usarse como base directa para el diseño físico y su ejecución en MySQL u otro SGBD.

es perfecto estopy llevando toda la trazabilidad en el archivo de drive doc aunque el proyecto se va a presentar en el .drawio ya que la idea es llevar esto a git hub.

Modelo Físico BD



## **🧩 Análisis del Modelo Lógico de Base de Datos**

### **📌 Proyecto: Sistema de Gestión de Abastecimiento**

El presente modelo lógico fue diseñado para estructurar una base de datos transaccional (OLTP) orientada a la gestión de productos, proveedores y órdenes de compra en un entorno de abastecimiento empresarial. Este modelo permite controlar eficientemente la trazabilidad entre productos, sus proveedores y las órdenes generadas, garantizando integridad y escalabilidad del sistema.

### **🔸 Estructura General**

El modelo está conformado por **6 entidades principales**, conectadas mediante relaciones **1:N** y **N:M**, con claves primarias (PK) bien definidas y claves foráneas (FK) que garantizan la integridad referencial.

| **Entidad** | **Descripción** |
| --- | --- |
| **Producto** | Contiene los productos disponibles. Se relaciona con una categoría y con órdenes. |
| **Proveedor** | Registra información básica de los proveedores. |
| **Categoria** | Clasificación de los productos. |
| **Orden\_Compra** | Encabezado de las órdenes emitidas, con fecha y proveedor responsable. |
| **Detalle\_Orden** | Detalle de productos por orden, con cantidad y precio unitario. |
| **Producto\_Proveedor** | Relación N:M entre productos y proveedores. |

### **🔹 Relaciones Lógicas**

1. **Producto – Categoria** (categoriaID):  
   * Relación 1:N: Una categoría puede tener muchos productos.
2. **Producto – Detalle\_Orden** (productoID):  
   * Relación 1:N: Un producto puede aparecer en muchas órdenes.
3. **Orden\_Compra – Detalle\_Orden** (ordenID):  
   * Relación 1:N: Una orden puede tener múltiples productos asociados.
4. **Orden\_Compra – Proveedor** (proveedorID):  
   * Relación 1:N: Un proveedor puede tener múltiples órdenes asignadas.
5. **Producto – Proveedor** (Tabla puente Producto\_Proveedor):  
   * Relación N:M: Un producto puede tener varios proveedores, y un proveedor puede distribuir varios productos. Esta relación se representa mediante una clave compuesta.

### **🔧 Consideraciones Técnicas**

* Se utilizaron tipos de datos apropiados: INT para identificadores, VARCHAR para textos, DECIMAL(10,2) para precios y DATE para fechas.
* La tabla Detalle\_Orden y Producto\_Proveedor manejan claves primarias **compuestas**, permitiendo controlar relaciones sin necesidad de identificadores artificiales adicionales.
* El modelo fue diseñado siguiendo las buenas prácticas de **normalización (hasta 3FN)**, minimizando redundancias y optimizando la integridad.

### **✅ Beneficios del Modelo**

* **Escalable**: Puede extenderse fácilmente a inventarios, almacenes, facturación, etc.
* **Modular**: Separación clara entre entidades maestras (producto, proveedor) y transaccionales (orden).
* **Íntegro**: Asegura consistencia mediante claves foráneas y relaciones bien definidas.